

B.T. Ergashev. K.Z. Abidov

MA'LUMOTLAR BAZASINI BOSHQARISH TIZIMLARI

O'quv qo'llanma



**O'ZBEKISTON RESPUBLIKASI OLIY VA O'RTA
MAXSUS TA'LIM VAZIRLIGI**

**BUXORO MUHANDISLIK-TEXNOLOGIYA
INSTITUTI**

B.T. Ergashev, K.Z. Abidov

**MA'LUMOTLAR
BAZASINI BOSHQARISH
TIZIMLARI**

**BUXORO – 2021
«DURDONA» NASHRIYOTI**

32.81я7

004.65(075)

E 75

B.T. Ergashev, K.Z. Abidov

Ma'lumotlar bazasini boshqarish tizimlari [Matn] : /o'quv qo'llanma/ B.T. Ergashev, K.Z. Abidov – Buxoro: "Sadriddin Salim Buxoriy" Durdona nashriyoti, - 2021. – 320 b.

UO'K 004.65(075)

BBK 32.81я7

Mazkur o'quv qo'llanma "Ma'lumotlar bazasini boshqarish tizimlari" fanidan namunaviy dastur asosida yozilgan bo'lib, unda ma'lumotlar bazasini boshqarish tizimlarining umumiy tushunchalari, ularning turlari, klassifikatsiyasi, ma'lumotlar bazasining modellari va ularni me'yorlashtirish shakllari, ma'lumotlar bazasi bilan ishlaydigan tillarning komponentalari hamda so'rovlarni tashkil etish texnologiyasi, shuningdek bilimlar bazasi bilan ishlash tartibi xususida ma'lumotlar keltirilgan. O'quv qo'llanma texnika yo'nalishidagi oliy o'quv yurtlari talabalari uchun mo'ljallangan bo'lib, undan tayanch doktorantlar, muhandis-texnik xodimlar ham foydalanishlari mumkin.

Данное учебное пособие составлено на основании примерной программы дисциплины «Системы управления базами данных» и содержит основные понятия системы управления базами данных, их типы, классификацию, модели базы данных и формы их реализации, компоненты языков работающие с базами данных и технологию организации запросов, а также порядок работы с базами знаний.

Учебное пособие предназначено для студентов технических направлений высших учебных заведений, а также ими могут пользоваться опорные докторанты и инженерно-технические работники.

This textbook had written based on a standard program on the subject "Database Management Systems", which includes general concepts of database management systems, their types, classification, database models and forms of their standardization, information on the components of languages working with the database and the technology of organizing queries, as well as the procedure for working with the database.

The textbook has intended for students of technical universities and can used by basic doctoral students, engineers and technicians.

TAORIZCHILAR

J. Jumaev

– fizika-matematika fanlari nomzodi, Buxoro davlat universiteti "Axborot texnologiyalari" kafedrası dotsenti..

O.K. Salieva

– texnika fanlari nomzodi, Buxoro muhandislik-texnologiya instituti "Texnologik jarayonlarni boshqarishning axborot-kommunikatsiya tizimlari" kafedrası dotsenti.

Mazkur o'quv qo'llanma Oliy va o'rta maxsus ta'lim vazirligi Buxoro Muhandislik-texnologiya instituti kengashining 2020 yil 26-oktyabr 3-sonli qaroriga asosan nashr qilishga ruxsat etilgan.

ISBN 978-9943-7368-3-2

© B.T. Ergashev, K.Z. Abidov

MUNDARIJA

Kirish.....	15
I. MA'LUMOTLAR VA BILIMLAR BAZASI. MA'LUMOTLAR BAZASINING ASOSIY TUSHUNCHALARI	
1.1. Ma'lumotlar bazasi va axborot tizimlari.....	17
1.2. Bilimlar bazasi va uning komponentlari.....	18
1.3. Bilimlar bazasidan foydalanishning texnologiyasi.....	19
1.4. Ma'lumotlar bazasi.....	22
1.5. Asosiy tushuncha va ta'riflar.....	25
1.6. Ma'lumotlar bazasiga qo'yiladigan asosiy talablar.....	29
II. Predmet soha tavsiflash usullari	
2.1. Predmet sohani tavsiflash.....	32
2.2. Ma'lumotlar bazasini yaratish bosqichlari.....	36
2.3. "Mohiyat-aloqa" usuli.....	42
2.4. "Mohiyat-aloqa" diagrammasi. Mohiyatlar klassifikatsiyasi.....	45
2.5. Predmet sohani infologik, datalogik va jismoniy modellari.....	53
III. MA'LUMOTLAR MODELLARI	
3.1. Ma'lumotlarning iyerarxik modellari.....	57
3.2. Ma'lumotlarning to'r (tarmoqlangan) modellari.....	61
3.3. Ma'lumotlarning relyatsion modellari.....	64
IV. RELYATSION MA'LUMOTLAR BAZASI. RELYATSION ALGEBRA VA RELYATSION HISOBLASH ELEMENTLARI	
4.1. Relyatsion modellarda operatsiyalar bajarish.....	69
4.2. Munosabatlar va bog'lanishlar.....	74
4.3. Relyatsion algebra va relyatsion hisoblash elementlari.....	77
V. MA'LUMOTLAR BAZASINI ME'YORLASHTIRISH. MA'LUMOTLAR-NING ME'YORIY SHAKLLARI NAZARIYASI	
5.1. Ma'lumotlar bazani me'yorlashtirish haqida umumiy ma'lumot.....	80

5.2.	Atributlarning funktsional bog'liqligi.....	88
5.3.	Ma'lumotlar bazasida kalit tushunchasi.....	91
5.4.	Me'yoriy shakllar.....	92

VI. MA'LUMOTLAR BAZASINI MANTIQUIY LOYIHALASH

6.1.	Relyatsion ma'lumotlar bazasini loyihalash asoslari.....	100
6.2.	Ma'lumotlarni turlarga guruhlash.....	105
6.3.	Axborot tizimlarini loyihalashning zamonaviy CASE-vositalari.....	108
6.3.1.	Axborot tizimlarini loyihalashda CASE-texnologiyalar o'rni.....	108
6.3.2.	CASE-vositalarini o'zlashtirish va joriy etish texnologiyasi.....	112
6.3.3.	Zamonaviy CASE-vositalar xarakteristikallari.....	113
6.3.4.	Lokal CASE-vositalari.....	116
6.3.5.	Ob'ektga mo'ljallangan CASE-vositasi.....	119
6.3.6.	Konfiguratsiyali boshqaruv vositasi.....	120
6.3.7.	Hujjatlashtirish vositasi.....	121
6.3.8.	Testlash vositasi.....	122
6.3.9.	CASE-vositalarining taqqosiy xarakteristikallari.....	123

VII. MA'LUMOTLAR BAZASINI BOSHQARISH TIZIMLARI

7.1.	Zamonaviy BTLari va ularning arxitekturalari.....	126
7.2.	MBBTlarini standartlashtirish va rivojlanish istiqbollari.....	139
7.3.	MS Access.....	143
7.4.	MS SQL Server.....	145
7.5.	IBM kompaniyasining serverli MBBTlari.....	146
7.6.	My SQL.....	147
7.7.	Taqsimlangan ma'lumotlar bazasi va uni qayta ishlash tizimlari arxitekturasi.....	149

VIII. SQL tili

8.1.	SQL tili haqida tushuncha.....	156
8.2.	Ma'lumotlar bazasini yaratish.....	165

8.3.	DDL operatorlari.....	170
8.4.	Ma'lumotlarni qayta ishlash so'rovlari.....	177
8.5.	Sodda so'rovlar.....	181
8.6.	Hisoblanadigan ustunlar. Murakkab so'rovlar.....	189

IX. INTERBASE DA MA'LUMOTLAR BAZASINI YARATISH

9.1.	InterBase da ishlash asoslari.....	199
9.2.	InterBase da yangi ma'lumotlar bazasini yaratish.....	202
9.3.	Jadvallarni yaratish.....	203
9.4.	Interbaseda ma'lumotlar bilan ishlash.....	208
9.5.	Generatorlar.....	215
9.6.	Triggerlar.....	216
9.7.	Indeksar va cheklovlar.....	218
9.8.	Saqlanuvchi prosteduralar.....	221

X.DELPHIDA MA'LUMOTLAR BAZASI BILAN ISHLASH

10.1.	MB bilan ishlovchi instrumental vositalar va komponentlar.....	224
10.2.	DataBase Desktop yordamida MBni hosil qilish.....	224
10.3.	BDE yordamida MB ga ulanish va ma'lumotlarni diagrammada tasvirlash.....	227
10.4.	Rave Reports hisobot generatori	230
10.5.	InterBASE va DBExpress palitrasi komponentlari yordamida InterBASE ma'lumotlar bazasi bilan ishlash.....	237
10.6.	XML-fayli ma'lumotlarini jadval ko'rinishida ifodalash.....	241
10.7.	Hisoblanadigan maydonlar.....	243

XI. PHPDA MA'LUMOTLAR BAZASI BILAN ISHLASH

11.1.	PHP haqida ma'lumot.....	246
11.2.	PHPda o'zgaruvchilar.....	247
11.3.	Shartli operatorlar.....	249
11.4.	Ma'lumotlarni uzatishda GET va POST metodlarni ishlatilishi.....	250
11.5.	PHP va MySQL bog'liqligi.....	252

11.6. PHP yordamida ma'lumotlar bazasini boshqarish.....	255
--	-----

XII. MA'LUMOTLAR BAZASI ADMINISTRATSIYASI

12.1. MB ni himoyalash bosqichlari.....	259
12.2. IBConsole dasturida foydalanuvchilar bilan ishlash.....	260
12.3. GRANT va REVOKE buyruqlari.....	263

XIII. ORACLE MA'LUMOTLAR BAZASINI BOSHQARISH

TIZIMI

13.1. ORACLE ma'lumotlar bazasi haqida qisqacha tarixiy ma'lumot.....	266
13.2. ORACLE MBBTda ishlatiladigan asosiy tushunchalar...	268
13.3. ORACLE MBBTning asosiy ob'ektlari.....	273
13.4. ORACLEda ma'lumot toifalari.....	274
13.5. Tasavvurlar, ketma-ketlik va sinonimlar.....	279
13.6. PL/SQLda dasturlash	286
13.7. PL/SQL tili alfaviti va muhit komandalari.....	289
13.8. Procedura, funksiya va paketlar.....	295
13.9. ORACLEda triggerlarning ishlatilishi.....	301

XIV. NOSQL MA'LUMOTLAR BAZASI

14.1. NoSQL ma'lumotlar bazasi haqida asosiy tushunchalar.....	306
14.2. NoSQL ma'lumotlar bazasining kelib chiqish sabablari.....	307
14.3. NoSQL bazalarining toifalari.....	309
14.4. NoSQL ma'lumotlar bazasidagi ma'lumotlar bilan ishlash.....	310
14.5. NoSQLning afzalliklari.....	311
Mustaqil bajarish uchun savol va masalalar.....	313
Adabiyotlar.....	316

ОГЛАВЛЕНИЕ

Введение.....	15
I. БАЗЫ ДАННЫХ И ЗНАНИЙ. ОСНОВНЫЕ ПОНЯТИЯ	
БАЗЫ ДАННЫХ	
1.1. Базы данных и информационные системы.....	17
1.2. Базы знаний и ее компоненты.....	18
1.3. Технология использования базы знаний.....	19
1.4. База данных.....	19
1.5. Основные понятия и определения.....	22
1.6. Основные требования к базам данных.....	25
II. МЕТОДЫ ОПИСАНИЯ ПРЕДМЕТНОЙ ОБЛАСТИ	
2.1. Описание предметной области	32
2.2. Этапы создания базы данных.....	36
2.3. Метод «Сущность - связь».....	42
2.4. Диаграмма «Сущность - связь».....	45
2.5. Инфологическая, даталогическая и физическая модели предметной области.....	53
III. МОДЕЛИ ДАННЫХ	
3.1. Иерархические модели данных.....	57
3.2. Сетевые модели данных.....	61
3.3. Реляционные модели данных.....	64
IV. РЕЛЯЦИОННАЯ БАЗА ДАННЫХ. РЕЛЯЦИОННАЯ АЛГЕБРА И РЕЛЯЦИОННОЕ ВЫЧИСЛЕНИЕ	
4.1. Выполнение операции в реляционных моделях..	69
4.2. Отношения и связи	74
4.3. Реляционная алгебра и элементы реляционного вычисления	77
V. НОРМАЛИЗАЦИЯ БАЗЫ ДАННЫХ. ТЕОРИЯ НОРМАЛЬНЫХ ФОРМ БАЗЫ ДАННЫХ.	
5.1. Основные сведения о нормализации базы данных.....	54
5.2. Функциональные связи атрибутов.....	88
5.3. Понятие ключ в базе данных.....	91

5.4.	Нормативные виды.....	91
VI. ЛОГИЧЕСКОЕ ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ		
6.1.	Основы проектирования реляционной базы данных.....	100
6.2.	Группировка данных по видам.....	105
6.3.	Современные CASE-средства проектирования информационных систем.....	108
6.3.1.	Место CASE-технологии в проектировании информационных систем.....	108
6.3.2.	Технология освоения и внедрения CASE-средств.....	112
6.3.3.	Характеристики современных CASE-средств....	113
6.3.4.	Локальные CASE-средства.....	116
6.3.5.	Объектно-ориентированные CASE-средства	119
6.3.6.	Средства управления конфигурацией.....	120
6.3.7.	Средства документирования.....	121
6.3.8.	Средства тестирования.....	122
6.3.9.	Сравнительные характеристики CASE-средств...	123
VII. СИСТЕМЫ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ		
7.1.	Современные СУБД и их архитектуры.....	126
7.2.	Перспективы стандартизации и развития СУБД	139
7.3.	MS Access.....	143
7.4.	MS SQL Server.....	145
7.5.	Серверные СУБД компании IBM.....	146
7.6.	My SQL.....	147
7.7.	Распределенная база данных и ее архитектура систем обработки.....	149
VIII. Язык SQL		
8.1.	Понятие о языке SQL	156
8.2.	Создание базы данных.....	165
8.3.	Операторы DDL	170
8.4.	Запросы обработки данных	177
8.5.	Простые запросы	181

8.6.	Вычисляемые столбцы. Сложные запросы.....	189
------	---	-----

IX. СОЗДАНИЕ БАЗ ДАННЫХ В INTERBASE

9.1.	Основы работы в InterBase.....	199
9.2.	Создание новой базы данных в InterBase.....	202
9.3.	создание таблиц	203
9.4.	Работа с данными в Interbase.....	208
9.5.	Генераторы.....	215
9.6.	Триггеры.....	216
9.7.	Индексы и ограничения	218
9.8.	Сохраняемые процедуры	221

X. РАБОТА С БАЗАМИ ДАННЫХ В DELPHI

10.1.	Инструментальные средства и компоненты работы с БД.....	224
10.2.	создание БД с помощью DataBase Desktop	227
10.3.	Соединение с БД с помощью BDE и изображение данных в диаграмме	230
10.4.	Генератор отчетов в Rave Reports.....	234
10.5.	Создание компонентов палитры InterBASE и DBExpress при работе с базами данных InterBASE	237
10.6.	Выражение данных файла XML в табличной форме	241
10.7.	Вычисляемые поля	243

XI. РАБОТА С БАЗАМИ ДАННЫХ В PHP

11.1.	Сведения о PHP	246
11.2.	Переменные в PHP.....	247
11.3.	Условные операторы.....	249
11.4.	Использование методов GET и POST для передачи данных.....	250
11.5.	Связь PHP и MySQL	252
11.6.	Управление базами данных с помощью PHP	255

XII. АДМИНИСТРАЦИЯ БАЗЫ ДАННЫХ

12.1.	Этапы защиты базы данных	259
12.2.	Работа с пользователями в программе IBConsole	260
12.3.	Команды GRANT и REVOKE	263

XIII. СИСТЕМА УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ

ORACLE

13.1.	Краткая история баз данных ORACLE	266
13.2.	Основные понятия СУБД ORACLE	268
13.3.	Основные объекты СУБД ORACLE	273
13.4.	Типы данных в ORACLE.....	274
13.5.	Представления, последовательности и синонимы.....	279
13.6.	Программирование на языке PL/SQL	286
13.7.	Алфавит языка PL/SQL и команды оболочки.....	289
13.8.	Процедуры, функции и пакеты	295
13.9.	Использование триггеров в ORACLE.....	301

XIV.БАЗЫ ДАННЫХ NOSQL

14.1.	Основные понятия о базы данных nosq	306
14.2.	Причины появление баз данных NoSQL	307
14.3.	Основные типы NoSQL базы данных.....	309
14.4.	Работа с данными NoSQL базы данных.....	310
14.5.	Преимущества NoSQL.....	311
	Вопросы и задачи для самостоятельного выполнения.....	313
	Литература.....	316

CONTENTS

Introduction	15
I. DATABASES AND KNOWLEDGE. DATABASE CONCEPTS	
1.1. Databases and information systems	17
1.2. Knowledge base and its components	18
1.3. Knowledge base technology	19
1.4. Database.....	22
1.5. Basic concepts and definitions	25
1.6. Basic requirements for databases	29
II. METHODS FOR DESCRIBING THE SUBJECT AREA	
2.1. Description of the subject area	32
2.2. Stages of creating a database	36
2.3. Method "Entity relationship diagram"	42
2.4. Entity relationship diagram.....	45
2.5. Infological, datalogical and physical models subject area	53
III. DATA MODELS	
3.1. Hierarchical data models	57
3.2. Network Data Models	61
3.3. Relational data models	64
IV. RELATIONAL DATABASE. RELATIONAL ALGEBRA AND RELATIONAL COMPUTATION	
4.1. Performing an operation in relational models	69
4.2. Relationships and connections	74
4.3. Relational algebra and the elements of relational computing	77
V. DATABASE NORMALIZATION. THE THEORY OF NORMAL FORMS DATABASE	
5.1. Basics of Database Normalization	80
5.2. Functional relationships of attributes	88
5.3. The concept of a key in a database	91
5.4. Regulatory types	92

VI. LOGICAL DATABASE DESIGN

6.1.	Relational Database Design Basics	100
6.2.	Data grouping by types	105
6.3.	Modern CASE-tools for designing information systems	108
6.3.1.	Place of CASE-technology in the design of information systems	112
6.3.2	Technology of mastering and implementation of CASE-tools	113
6.3.3	Characteristics of modern CASE-tools	116
6.3.4.	Local CASE-tools	116
6.3.5.	Object-Oriented CASE Tools	119
6.3.6.	Configuration controls	120
6.3.7.	Documentation Tools	121
6.3.8.	Testing tools	122
6.3.9.	Comparative characteristics of CASE tools	123

VII. DATABASE MANAGEMENT SYSTEMS

7.1.	Modern DBMS and their architectures	126
7.2.	Prospects for standardization and development of DBMS	139
7.3.	MS Access	143
7.4.	MS SQL Server	145
7.5.	Server DBMS of IBM company	146
7.6.	My SQL	147
7.7.	Distributed database and its architecture of processing systems	149

VIII. THE SQL LANGUAGE

8.1.	Understanding the SQL language	156
8.2.	Database creation	165
8.3.	DDL Operators	170
8.4.	Data processing requests	177
8.5.	Simple queries	181
8.6.	Calculated Columns. Complex queries	189

IX. CREATION OF DATABASES IN INTER BASE

9.1.	Basics of working in Inter Base	199
9.2.	Creation of a new database in Inter Base	202
9.3.	Creating tables	203
9.4.	Working with data in Inter base	208
9.5.	Generators	215
9.6.	Triggers	216
9.7.	Indices and Limitations	218
9.8.	Stored Procedures	221

X. WORKING WITH DATABASES IN DELPHI

10.1.	Tools and components for working with the database	224
10.2.	Creating a database using Data Base Desktop	227
10.3.	Connecting to a DB using BDE and displaying data in a diagram	230 234
10.4.	Report generator in Rave Reports	237
10.5.	Creation of Inter BASE and DB Express palette components when working with Inter BASE databases	241
10.6.	Expression of XML file data in tabular form	243
10.7.	Calculated Fields	232

XI. WORKING WITH DATABASES IN PHP

11.1.	About PHP	246
11.2.	Variables in PHP	247
11.3.	Conditional statements	249
11.4.	Using the GET and POST Methods to Transfer Data.....	250
11.5.	PHP and MySQL communication	252
11.6.	Base management	255

XII. DATABASE ADMINISTRATION

12.1.	DB protection steps	259
12.2.	Working with users in the IBConsole program	260
12.3.	GRANT and REVOKE commands	263

XIII. ORACLE DATABASE MANAGEMENT SYSTEM

13.1.	A brief history of the ORACLE database.....	266
13.2.	Basic concepts used in ORACLE MBBT.....	268
13.3.	The main objects of ORACLE MBBT	273
13.4.	Data categories in ORACLE.....	274
13.5.	Sections, Sequences and Synonyms.....	279
13.6.	Programming in PL/SQL.....	286
13.7.	PL/SQL language alphabet and environment commands.....	289
13.8.	Procedures, functions and packages	295
13.9.	Use of triggers in ORACLE.....	301

XIV.NOSQL DATABASE

14.1.	Basic knowledge of NoSQL database	306
14.2.	The origins of the NoSQL database	307
14.3.	Categories of NoSQL databases	309
14.4.	Working with data in the NoSQL database.....	310
14.5.	Advantages of NoSQL.....	311
	Questions and issues for independent completion ...	313
	Literatures	316

KIRISH

Hozirgi vaqtda jamiyat rivojlanishining ko'p jabhalarida ulkan hajmdagi axborot bilan ishlashga to'g'ri keladi. Ayniqsa axborot oqimlarining tobora ortib borishi, ma'lumotlarining tez o'zgarishi kabi holatlar insoniyatni bu ma'lumotlarni o'z vaqtida qayta ishlash choralarini qidirib topishga undaydi. Natijada, axborot texnologiyalarida bu muammolarni samarali yechish usullari yaratildi va u ma'lumotlar bazasi yo'nalishi deb nomlandi.

Ma'lumotlar bazasi (MB), bu o'zaro bog'langan va tartiblangan ma'lumotlar majmuasi bo'lib, u ko'rilayotgan ob'ektlarning xususiyatlarini, holatini, va ob'ektlar o'rtasidagi munosabatni ma'lum sohada tavsiflaydi.

Katta hajmli ma'lumotlar matematik asosga ega bo'lgan MBda saqlanadi. MBda ma'lumotlarni saqlash, uzatish va qayta ishlash uchun maxsus ma'lumotlar bazasini boshqarish tizimlari (MBBT) yaratilgan. Natijada ulkan hajmdagi, turli tuzilmali axborotni saqlashning ishonchli, arzon va samarali imkoniyatlari vujudga keldi. Axborot bilan ishlashda MBBTdan foydalanish quyidagi imkoniyatlarni beradi:

- turli formatdagi katta hajmli axborotni ancha ratsional usulda saqlash;
- ma'lumotlarga tez kira olish imkoniyatiga ega bo'lish;
- yopiq ma'lumotlarga kafolatli cheklovlar qo'yish;
- kompyuter tarmog'ida axborot bilan ishlash uchun "mijoz-server" arxitekturasiga asoslangan dasturlar yaratish.

Ushbu o'quv qo'llanmada MBBT yordamida ma'lum bir soha bilan bog'liq bo'lgan MBni loyihalashtirish, shu bilan bir qatorda taqsimlangan MBBTlari masalalari ham alohida ko'rib o'tiladi.

"Ma'lumotlar bazasini boshqarish tizimlari" fani bo'yicha tayyorlangan ushbu o'quv qo'llanmada quyidagi masalalar yoritilgan:

- ma'lumotlar banki va bazasi haqida umumiy tushunchalar va ularda ishlashni tashkil etish;

- MBning asosiy konseptsionalari va ularning tahlili;

- MBni loyihalashtirish va ularni loyihalashtirish prinsiplarini tashkil etish;

- ma'lumotlar modellarini yaratish va tadbiq etish yo'llari;

- MBning me'yoriy shakllari va ma'lumotlar bazani me'yorlashtirish;

- so'rovlarni tashkil qilish va ularni tadbiq etish yo'llari;

- amaliyotdan kelib chiqib, texnik va dasturiy vositalar yordamida MBni yaratish texnologiyalari;

- SQL tili;

- zamonaviy ma'lumotlar bazasini boshqarish tizimlari asosida ishlovch axborot tizimlarini yaratish texnologiyasi;

- Delphi va PHP dasturiy vositalari yordamida ma'lumotlar bazasi bilan ishlash.

I. MA'LUMOTLAR VA BILIMLAR BAZASI. MA'LUMOTLAR BAZASINING ASOSIY TUSHUNCHALARI

1. 1. Ma'lumotlar bazasi va axborot tizimlari

Ko'pgina masalalarni yechish asosida axborotlarni qayta ishlash yotadi. Axborotlarni qayta ishlashning yengillashtirish maqsadida axborot tizimlari yaratiladi. Avtomatlashgan axborot tizimlar (AT) deb, shunday tizimlarga aytamizki, ularning tarkibida texnik vositalar, jumladan shaxsiy kompyuterlar ishtrok etadi. ATlarni keng ma'noda axborotni qayta ishlaydigan ixtiyoriy tizimni tushunish mumkin. Tadbqiq etish sohasiga qarab, ATlar ishlab chiqarish sohasida, talim sohasida, sog'liqni saqlash sohasida, harbiy sohada va boshqa sohalarda ishlatiladigan tizimlarga ajratish mumkin.

ATlarni maqsadli ishlatilishiga qarab bir qancha kategoriyalarga ajratishimiz mumkin. Jumladan boshqariluvchi, axborot qidiruv, axborot ma'lumotnoma va boshqa tizimlarga ajratiladi.

Axborot tizimlarni tom ma'noda ba'zi bir amaliy masalalarni yechishda ishlatiladigan apparat dasturiy vositalari majmuasi sifatida ham tushunamiz. Masalan, tashkilotlarda ishlatiladigan kadrlarni hisobga olish va kuzatish, omborxona material va tovarlarni hisobga olish va nazorat qilish, buxgalter masalarini yechish va boshqalar.

ATlari asosida ma'lumotlar bazasi yotadi. MB degenda, ma'lumotlarni shunday o'zaro bog'langan to'plamini tushunamizki, u mashina xotirasida saqlanib, maxsus ma'lumotlar bazasini boshqarish tizimini to'ldirilishi, o'zgartirilishi, takomillashtirilishi mumkin.

Aniq ma'lumotlarni (masalani) hal qilishda inson real dunyoni u yoki bu sohasi bilan cheklanadi. Bunday hollarda faqat ba'zi bir ob'ektlarni o'rganishgina qiziqish o'yg'otadi. Bunday ob'ektlarni majmuasini predmet soha deyiladi.

Obyektlarni sinflarga ajratish deyilganda, barcha ob'ektlar to'plamini birorta norasmiy belgi bo'yicha qism to'plamlarga ajratishni tushunamiz. MB ni ko'pligini hisobga olib, uni sinflarga ajratish belgilari xilma – xil. Hozirgi kunda MBni quyidagi sinflari ko'p ishlatiladi:

1. MB ma'lumotlarni tasvirlash shakliga qarab: video, audio, multimedia guruhlariga ajratish mumkin.

2. Video MB ma'lumotlarini ko'rinishiga qarab o'z navbatida matnli

va grafik tasvirli bo'ladi.

3. Matnli MB ma'lumotlarni strukturalashganiga qarab strukturalashgan, qisman strukturalashgan va strukturalashmagan MBga bo'linadi.

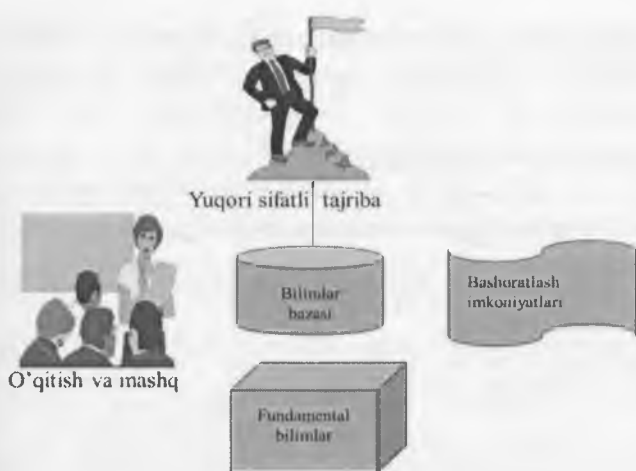
4. Strukturalashgan MB o'z navbatida ma'lumotlarni modeliga qarab: iyerarxik, tarmoqli, relyasion, ob'ekli relyasion, ob'ektga yo'naltirilgan MBga bo'linadi. Bundan tashqari strukturalashgan MBlari strategik va dinamik shuningdek, markazlashgan va taqsimlangan MBga bo'linadi. MBni foydalanuvchilar soniga qarab: bitta va ko'p foydalanuvchili MBga bo'lamiz va ular ma'lumotlarni saqlanishiga qarab operatsion va analitik bo'ladi[8].

1.2. Bilimlar bazasi va uning komponentlari

Iqtisodiy faoliyatni boshqarishning avtomatlashtirilgan axborot texnikasini axborotli ta'minlashda sun'iy aql sohasining qo'llanishi katta qiziqish uyg'atadi, bu sohadagi yutuqlarni amalga oshirishning shakllaridan biri ekspertli tizimlarni yuqori malakali mutaxassislar, ekspertlar bilimlarini tizimli jamlash, umumlashtirish, tahlil qilish va baholashga asoslangan maxsus kompyuter tizimlarini yaratashdan iboratdir. Ekspertli tizimda bilimlar bazasidan foydalaniladi, unda aniq muammo soha haqidagi bilimlar beriladi.

Bilimlar bazasi—bu, ba'zi bir muammo sohada murakkab vazifalarning yechimini topish uchun tahlil va xulosalarni yaratuvchi modellar, qoidalar va omillar (ma'lumotlar)ning majmuidir.

Axborotli ta'minlanishning alohida, yaxlit tuzilishi ko'rinishidan ajratilgan va tashkil qilingan muammo soha haqidagi bilimlar oshkora bo'ladi va bilimlarning boshqa turlaridan, masalan umumiy bilimlardan ajratiladi. Ma'lumotlar bazasi shunchaki rasmiy (matematik) mantiq asosida, balki tajriba, dalillar, evristikalar asosida ham mulohazalarni bajarishga imkon beradi, ya'ni ular inson mantiqiga yaqinlashtirilgandir.



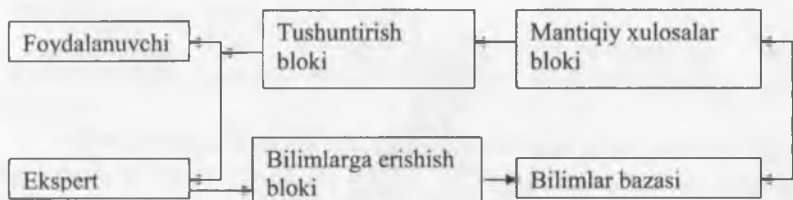
1.2.1-rasm. Bilimlar bazasining ishlatilishi.

Sun'iy aql sohasidagi ishlab chiqishlar murakkab, oddiy bo'lmagan vazifalarni yechish uchun ba'zi bir tor muammo soha haqidagi yuqori malakali maxsus bilimlardan katta hajmlarda foydalanish maqsadiga egadir.

Ma'lumotlar bazasi ekspertli tizimning asosi bo'ladi, uni qurish jarayonida jamlanadi. Bilimlar fikrlash va vazifalarni yechishning yaqqol usulini qurishga imkon beruvchi oshkora ko'rinishda aks ettiriladi, qarorlar qabul qilishni soddalashtiruvchi sifatida tashkil qilingan. Ekspertli tizimning xabardorligini asoslovchi bilimlar bazasi muassasa, bulim mutaxassislari bilimlarini va mutaxassislar guruhlarini tajribasini o'zida jamlaydi hamda institutsional bilimlar (malakali, yangilanayotgan strategiyalar, usullar, qarorlarning yig'indisi) dan iborat bo'ladi.

1.3. Bilimlar bazasidan foydalanishning texnologiyasi

Ma'lumotlar bazasining mazmuni foydalanuvchi tomonidan samarali boshqaruv qarorlarini olish uchun tatbiq etiladigan tuzilish va uning faoliyat yuritishi tarzida 1.3.1-rasmda keltirilgan



1.3.1-rasm. Bilimlar bazasidan foydalanishning texnologiyasi

Ekspert –bu, aniq muammo sohada samarali qarorlarni topa oluvchi mutaxassisdir.

Bilimlarni bilish bloki bilimlar bazasining jamlanishi, bilimlar va ma’lumotlarning zamonaviylashtirish bosqichini aks ettirada. Bilimlar bazasi malakali mutaxassisning fikrlash darajasidagi yuqori sifatli tajribadan foydalanish imkoniyatini aks ettiradi. U ekspertli tizimni biznes va buyutrmachining zaruriyatlariga muvofiq ravishda rentabelli qiladi.

Mantiqiy xulosalar bloki qoidalarning dalillar bilan taqqoslanishini amalga oshirib, xulosalar zanjirini yaratadi. Ishonchsiz ma’lumotlar bilan ishlashda noaniq mantiq, ishonchlilikning kuchsiz koeffitsiyentlari, ishonch o’lchovlarining past darajada va hokazolar shakllantiriladi.

Tushuntirish bloki foydalanuvchi tomonidan bilimlar bazasidan foydalanish texnologiyasini “nima uchun?” savoliga javob berish imkoniyatiga ega, u yoki bu xulosalarga olib keluvchi qadamlarning bilimlar jamlanishi sur’ati bilan belgilanadi.

Inson faoliyatining kasbiy jihatdan shakllantirilishi, demakki, EHM asosida avtomatlashtirilishi mumkin bo’lgan sohasi, bu, inson tomonidan jamlangan bilimlarning bir qismidir. Jamlangan bilimlar tarkibidagi katta qatlamni yakka tartibda jamlanuvchi begonalashtirilmaydigan bilimlar tashkil qiladi. Va, nihoyat, barcha qolgan bilimlarning umumiy hajmidagi zo’rg’a farqlanadigan – bu, bilimlarni shakllantirishdir.

Bilimlarni tarkiblashtirish yoki shaklga tushirish ularni tatbiq etishning turli usullariga asoslangan. Zamonaviy tizimlarda eng ommaviy usul dalillar va qoidalaridan foydalanishdir. Ular ba'zi bir muammo sohada jarayonlarni bayon qilishning tabiiy usulini ta'minlaydi.

Qoidalar, tavsiya, ko'rsatmalar strategiyalarini taqdim etishning rasmiy usulini ta'minlaydi, ular muammo bilimlarni ushbu sohadagi vazifalarni yechish bo'yicha ishlash yo'llari ichida jamlangan tajribali (empirik) assotsiatsiyalardan vujudga kelgan hollarda ham to'g'ri keladi. Qoidalar ko'pincha «Agar... shunda...» ko'rinishida bo'ladi.

Bilimlar bazasida muammo sohasini bayon qilish bilimlarni taqdim etish va tashkil qilish, vazifalarni shakllantirish, qayta shakllantirish va yechish usullarini ishlab chiqishni ko'zda tutadi. Muammo soha tushinchasi alomatlar yordamida taqdim etiladi. Masalan, bank tizimi uchun bu mijoz, fond vositasi, operatsiya, vazifa bo'lishi mumkin. Alomatli tushunchalar o'rtasidagi munosabat aniqlanadi, tushunchalar bilan manipulyatsiya qilish uchun turli xildagi (mantiqiy yoki tajriba natijasida olingan) strategiyalar qo'llaniladi. Bilimlarni taqdim etish, ularni turkumlashtirish tushunchalarini, murakkab, oddiy bo'lmagan vazifalarni tanlashni ko'zda tutadi. Shuning uchun ham bilimlar bazasidagi qoidalar yoki murakkab yoki ko'p hajmli bo'ladi.

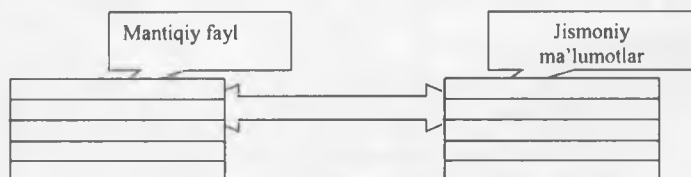
Bilimlar bazasi tamoyillarining rivojlanishi sun'iy aql tizimi sohalaridagi tadqiqotlar va yutuqlarga bog'liq. Bilimlar bazalarini qo'llash sohalari va ular asosidagi tizimlar kengayadi. Bilimlar bazalarining butun bir spektri portativ tizimlar uchun hajmi bo'yicha kichikdan tortib to murakkab va qimmatbaho avtomatlashtirilgan axborot tizimlaridan foydalanuvchi kasb egalari uchun mo'ljallangan va qudratga ega holda yaratiladi. Bilimlarning juda katta bazalari markazlashtirilgan saqlash joylarida saqlanadi, foydalanuvchi tomonidan ularga kirish turli tizimlar, darajalar, ko'lamlarning tarmog'i orqali amalga

oshiriladi. Bilimlar bazalarini ishlab chiqishdagi muvaffaqiyatlar, ularga ommaviy foydalanuvchi kirishini osonlashtiradi, bu ularni faol tijorat mahsuloti sifatida paydo bo'lishiga yordam beradi.

1.4. Ma'lumotlar bazasi

Ma'lumotlar bazasi tushunchasi 60-yillarning oxirlarida paydo bo'ldi. Bungacha ma'lumotlarni qayta ishlash sohasida ma'lumotlar fayli va ma'lumotlarni o'lchash to'g'risida gapirilar edi. Dasturchilar ma'lumotlarini shunday tashkil qilar edilarki, u faqat qaralayotgan masala uchungina o'rinli bo'lardi. Har bir yangi masalani hal qilishda ma'lumotlar qaytadan tashkil qilinir va bu hol yaratilgan dasturlardan foydalanishni qiyinlashtirir edi. Shundan so'ng ma'lumotlar bazasi konseptsiyasi sekin-asta, avtomatlashtirilgan axborot tizimining rivojlanishi jarayonida pishib bordi va uning rivojlanishini bir nechta bosqichlarga ajratish mumkin.

Birinchi bosqichda (60-yillar boshlarida) ma'lumotlar 1-rasm^{da} ko'rsatilgani kabi qayta ishlangan.



1.4.1-rasm. Ma'lumotlar tuzilmasi

Bu bosqichning dastur ta'minoti asosan eslab qoluvchi qurilmaga kiritish-chiqarish ishini va ma'lumotlarni qayta ishlashda katta bo'lmagan ba'zi bir yordamchi qurilmalarni yuritishni amalga oshirar edi. Ma'lumotlarni tashkil qilishni esa amaliy dasturlarni yozishda o'ylashga to'g'ri kelar edi va bu oddiy usul bilan yechim ma'lumotlarni oddiy fayllar ketma-ketligi ko'rinishida tashkil qilish bilan amalga oshirilar edi.

Ikkinchi (60-yillar oxiri) bosqich. Fayllar tabiati va ular saqlanadigan qurilmalarning o'zgarishi bilan birinchi bosqichdan farqlanadi. Bu bosqichda ma'lumotlarni ketma-ket to'plamlar ko'rinishida tashkil etishnigina emas, balki ulardan tug'ridan-tug'ri foydalanishning ham imkoni bo'ldi.

Ma'lumotlarni qayta ishlash vositalarining rivojlanishi bilan amaliy dasturlarning ma'lumotlarni saqlash apparat vositalari o'zgarishiga, o'sishiga, saqlanayotgan ma'lumotlarga yangi ob'ekt va yangi bog'likliklar qo'shilishiga bog'liqmas bo'lishi maqsadga muvofiq ekanligi ma'lum bo'ldi. Axborot texnologiyalarning rivojlanishi va axborot oqimlarining tobora ortib borishi, ma'lumotlarning tez uzgarishi kabi holatlar **ma'lumotlar bazasi (MB)** ni vujudga keltirdi.

Ma'lumotlar bazasi rivojlanishining **3-bosqichida (70-yillar boshida)** yuqorida aytganlarni amalga oshirishga urinishlar bo'ldi. Ushbu davrda ma'lumotlar bazasini boshqarishning dastur ta'minoti (MBBT) paydo bo'ldi. MBni boshqarish uchun ma'lumotlar bazasini boshqarish tizimlari ishlatiladi (MBBT).

Ma'lumotlarni modellar ko'rinishida ifodalanishining kiritilishi **to'rtinchi bosqichning** farqli xususiyatidir. Bu paytda dasturchilarning ma'lumotlar tug'risidagi tasavvurlari kabi, ma'lumotlarning fizik saqlanishi va tashkil etilishi ham o'zgaradi, ma'lumotlar modeli o'zgarmasdan qolishi yoki ular qayta ishlashning qo'shimcha usullarini yoki ma'lumotlarning qo'shimcha turlarini kiritish maqsadida uzgartirilishi mumkin.

Ma'lumotlar bazasini boshqarish tizimi(MBBT) deb - MB-ni yaratish, ishlatish va ko'pchilik foydalanuvchilar tomonidan birgalikda foydalanishni ta'minlovchi tillar va dasturiy ta'minotlari to'plamiga aytiladi.

Ma'lumotlar bazasi(MB) deb - ob'ektlar holatini, ularning qaralayotgan soha predmetidagi munosabatini akslantiruvchi ma'lumotlar to'plamiga aytiladi.

Ma'lumotlar bazasi – bu o'zaro bog'langan va tartiblangan ma'lumotlar majmuasi bulib, u ko'rilayotgan ob'ektlarning xususiyatini, holatini va ob'ektlar o'rtasidagi munosabatni ma'lum sohada tavsiflaydi.

Ma'lumotlar bazasi (date base) – bu kompyuterning tashqi xotirasida saqlanadigan, har qanday jismoniy, ijtimoiy, statistik, tashkiliy va boshqa ob'ektlar, jarayonlar, holatlarning o'zaro bog'liq va tartiblashtirilgan majmuidir.

MB - diskda saqlanadigan va maxsus ravishda tashkil qilingan ma'lumotlar turkumlarining jamlanmasidir.

MB - kompyuter yordamida qidirib topilishi va qayta ishlatish mumkin bo'lgan tarzda tartibga solingan ma'lumotlar to'plamidir. Masalan, maqolalar, hisob-kitob, telefon raqamlari, guruh jurnali ma'lumotlari va h.k.

Ma'lumotlarni saqlash, uzatish va qayta ishlash uchun MBni yaratish, so'ngra undan keng foydalanish bugungi kunda dolzarb bo'lib qolmoqda. Darhaqiqat, hozirgi kunda inson hayotida MBda kerakli axborotlarni saqlash va undan oqilona foydalanish juda muhim rol o'ynaydi. Chunki, jamiyat taraqqiyotining qaysi jabhasiga nazar solmaylik o'zimizga kerakli ma'lumotlarni olish uchun, albatta, MBga murojaat qilishga majbur bo'lamiz.

Ma'lumotlar bazasi ob'ekt yoki ob'ektlar to'plami holatini ularning xossalari va o'zaro munosabatlarini aks ettiruvchi nomlangan ma'lumotlar to'plamini ifodalaydi.

Ma'lumotlar bazasini ob'ektning axborot modeli deb qarash mumkin.

Ma'lumotlar bazasi, ma'lumotlar to'plamini cheksiz ko'p marta ishlata olish uchun kerak.

MBni boshqarishda keng miqyosda ishlatilishi quyidagi faktorlar bilan aniqlangan:

- operativligi (ma'lumotlarga tez va operativ kirish imkoniyati bilan);

- o'ng'ayligi (istalgan talabga javob ola olishligi, ma'lumotlarni takomillashtirish va o'zgartiruvchi effektiv usullarning mavjudligi);

- himoyalanganligi va yaxlitligi (ma'lumotlarga ruxsatsiz kirish mumkin emasligi hamda kompyuter texnik vositalarining nosozligidan himoyalanganligi).

Ma'lumotlarni qayta ishlash masalalarini yechish usullarning takomillashishi, ma'lumotlar banki tushunchasini ham keltirib chiqardi.

Ma'lumotlar banki (MBN) – MB, MBBT va ularni amalga oshiruvchi texnik vositalar majmuidir.

Ma'lumotlar banki(MBN) - ma'lumotlarni markazlashtirilgan holda saqlash va jamoa bo'lib foydalanishning avtomatlashtirilgan tizimidir. Uning tarkibiga texnik vositalar, MB yoki ularning majmualari, MBBT hamda amaliy dasturlar paketi kiradi.

Ma'lumotlarni o'zida saqlab va MBBT ostida boshqariladigan fayl oldin «ma'lumotlar banki» deb atalib, keyinchalik esa "ma'lumotlar bazasi" deb yuritila boshlandi.

MBN istalgan ma'lumotlar to'plamiga tegishli bo'lishi mumkin, jumladan mustaqil fayllar, MB va axborot izlovchi tizimlar. Ma'lumotlar bazasidan va bankidan foydalanish o'zaro bog'langan ma'lumotlar to'plamlariga kirishni ulardan foydalanishni avtomatlashtirish, ma'lumotlarni yangilash, keraksizlarini yo'qotishni avtomatlashtirishni ta'minlaydi, jumladan effektiv paketli holatni va teleprotssessorli ma'lumotlarni ishlashni ta'minlaydi.

1.5. Asosiy tushuncha va ta'riflar

MB ob'ektlar holatini, ularning qaralayotgan soha predmetidagi munosabatini akslantiruvchi ma'lumotlar to'plami hisoblanadi.

MB (date base) – bu o'zaro bog'langan va tartiblangan ma'lumotlar majmuasi bo'lib, u ko'rilayotgan ob'ektlarning xususiyatini, holatini va ob'ektlar o'rtasidagi munosabatni ma'lum sohada tavsiflaydi.

MB, kompyuterning tashqi xotirasida saqlanadigan, har qanday jismoniy, ijtimoiy, statistik, tashkiliy va boshqa ob'ektlar, jarayonlar, holatlarning o'zaro bog'liq va tartiblashtirilgan majmuasidir.

MB, diskda saqlanadigan, maxsus ravishda tashkil qilingan ma'lumotlar turkumlarining jamlanmasidir.

Ma'lumotlar banki (MBN) – MB, MBBT va ularni amalga oshiruvchi texnik vositalar majmuasidir.

MB, bu maxsus formatga ega bo'lgan muayyan tuzilmali fayl bo'lib, u yozuv deb ataladigan bir xil turdagi qismlardan iborat bo'ladi.

Yozuv-o'zaro bog'langan ma'lumotlarning bir qismidir.

Har bir yozuv maydon deb ataladigan bo'laklardan tashkil topadi. Maydon ma'lumotlarning eng kichik qismi hisoblanadi. U ma'lumotlarning, imkoni boricha, qisqa to'plamidan iborat bo'lishi lozim. Ammo, u foydalanuvchi uchun ma'nosini yo'qotadigan darajada kichik bo'laklarga bo'linishi mumkin emas. Har bir maydon, o'zi ifodalaydigan ma'lumotlariga ko'ra biror nomga ega bo'ladi. Fikrimizni misol bilan ifodalashga harakat qilamiz. Masalan, ta'lim muassasida o'qiydigan 1 - kurs talabalarini informatika, matematika va huquqshunoslik fanlaridan o'zlashtirish darajasini aniqlovchi bazani yaratishimiz kerak. Ma'lumotlarni jadval shaklida taqdim qilamiz. Bu jadvalni MB ning bir qismi deb olamiz. Unda 6 ta yozuv bo'lib, ularning har biri 7 ta maydondan iborat. Mazkur maydonlarning har biri, mos ravishda 'Familiyasi', 'Ismi', 'Guruh raqami', 'Informatika %', 'Matematika%' 'Huquqshunoslik%' deb nomlangan.

1.5.1-jadval

Imtihon jadvali

T	Familiyasi	Ismi	Guruhi №	Informatika %	Matematika %	Huquq-shunoslik %
1	Abdullaev	Mahmud	11	80	100	100
2	Dadaboev	Rahim	16	60	80	80
3	Axmedov	Adham	12	80	80	80
4	Sobirov	Ilhom	18	100	100	100
5	Jabborov	Islom	18	80	80	80
6	Halimov	Rustam	14	60	80	80

Jadval yaratilgan bo'lsa, unga qo'shimcha qatorlarni qo'shib, qiymat kiritishimiz mumkin. Ammo, lozim deb topilmasa, unga yangi ustun qo'shish imkoniyati mavjud emas.

Misol tariqasida oliy o'quv yurtining aniq fakul'tetida tahsil olayotgan biror guruh talabalari to'g'risidagi ma'lumotlar bitilgan jadvalni tahlil etamiz:

1.5.2-jadval

Guruh talabalari

Familiyasi	Ismi	Tug'ilgan sanasi	Guruhi	Turar joyi	Qiziqqan fani
Ochilov	Alisher	2.05.1978	5-M	A.Somiy,4	Matem.
Qobulov	Farhod	2.12.1982	6-E	I.Sino,1	Adabiyot
Aminov	San'at	3.06.1980	5-M	6 mkr-2,12	Tarix
Tolipov	Jasur	24.05.1979	6-E	Beruniy,2	Iqtisod

Ushbu jadvalda to'rtta yozuv bo'lib, ularning har biri 6 ta maydondan iborat. Mazkur maydonlarning har biri, mos ravishda "Familiyasi", "Ismi", "Tug'ilgan sanasi", "Guruhi", "Turar joyi" va "Qiziqqan fani" deb nomlangan.

Jadvalga aniq ma'lumotlar kiritildi va har bir yozuv aniq ma'lumotlardan iborat. Bu ma'lumotlar birlamchi hisoblanadi.

Ma'lumotlar elementi o'rtasidagi o'zaro aloqani ifodalovchi ma'lumotlar to'plami kortej deyiladi. Lekin, dastur ta'minotining har xil tizimlarida bu ibora o'rnida ma'lumotlar elementlari

guruhi, satr, segment, yozuv kabi soʻzlar ishlatiladi. Hozirgi vaqtda umumiy qabul qilingan ibora yozuvdir.

Maydonlar turli xil qiymatlarni qabul qiladi va ularning baʼzi biri yozuvda muhim ahamiyatga ega hisoblanadi. Masalan, biror bir oliy taʼlim muassasi xodimlari haqidagi maʼlumotlar toʻplamini tuzash talab etilsin (1.5.3-jadval). Unda "Tartib raqami", "Familiyasi", "Ismi", "Tugʻilgan sanasi", "Lavozimi", "Ish staji" va "Ishlaydigan boʻlimi" singari maydonlar mavjud boʻlsin. "Ishlaydigan boʻlimi" maydonidagi qiymatlar har doim takrorlanmasligi uchun shartli ravishda qisqa belgilash, yaʼni kodlash (FIZ-fizika-matematika fakul'teti, XIM-kimyo fakul'teti, ATM-axborot texnologiyalari markazi va h-zo singari) kiritilgan taqdirda ish yetarli darajada osonlashadi.

Juda koʻp axborotlarda davlat klassifikatori ishlatiladi va boshqa holatlarda kodlarni ishlatish, yaratuvchining xohishiga koʻra amalga oshiriladi.

Jadvalda har bir yozuv oʻzining bosh kalitiga ega boʻlishi va uning qiymati yagona boʻlishi kerak. Bosh kalit ikkita xossaga ega boʻladi:

1.Ustundagi qiymatning yagonaligi, yaʼni takrorlanmasligi.

1.5.3-jadval

Oʻqituvchilar jadvali

T	Familiyasi	Ismi	Tugilgan yili	Lavozimi	Ish staji	Ishlaydigan boʻlimi
1.	Axmedov	Ikrom	1954	Dekan	20	FIZ
2.	Sobirov	Akram	1949	Dekan muovin	15	FIZ
3.	Jumanazarov	Siroj	1968	Boʻlim boshligi	10	ATM
4	Sattorov	Jamol	1956	Oʻqituvchi	12	XIM

2. Kalitdagi qiymat imkon darajasida oʻzgarmasligi lozim. Bizning misolimizda bosh kalit "Tartib raqami" maydonidir. Har bir MB quyidagi muhim xossalarga ega boʻlishi lozim.

MB yaxlitligi. Ma'lumotlar yaxlitligi yoki butunligi deganda ma'lumotlarni alohida olingan tarkibiy qismlarining o'zaro mutanosibligi tushuniladi.

MB ning mustaqilligi. MB o'z ichki tuzilmasiga ega bo'lgan mustaqil ob'ekt bo'lib, u amaliy dasturlarga bog'liq emas. Ma'lumotlarni ma'lumot tashuvchilarga yozish va ularga murojaat qilish uslublari amaliy dasturiy mahsulotlarga bog'liq emas. MBga har qanday amaliy dasturiy mahsulot orqali murojaat qilish mumkin va bu vaqtda MB mustaqil ravishda amaliy dasturlarga bog'liq bo'lmagan holda o'z uslublarini qo'llaydi [8].

1.6. Ma'lumotlar bazasiga qo'yiladigan asosiy talablar

MB ni tuzishda quyidagi talablarga rioya qilish lozim:

- unumdorlik;
- ortiqchalik;
- butunlik;
- xavfsizlik;
- maxfiylik;
- eng kam sarf va himoya.

- *Unumdorlik.* Foydalanuvchini MB bilan bog'lovchi, maxsus ishlab chiqilgan ilovalar foydalanuvchi bilan muloqotni qanoatlantiradigan vaqtni ta'minlab berishi zarur. Bundan tashqari MB tizimiga kelib tushgan ma'lumotlarni qayta ishlashni ta'minlashi zarur.

- *Ortiqchalik.* MBda har bir ma'lumot faqat bir joyda saqlanishi va u takrorlanmasligi lozim. Agar ma'lumot faqat bir joyda saqlansa, ular qiymati o'zgartirilganda MB yaxlitligi ta'minlanadi. Chunki, u boshqa joyda takrorlanmasligi sababli boshqa aynan shunday qiymatni o'zgartirishga ehtiyoj yo'q. MB ning unumdorligi ortiqcha ma'lumotlarning bo'lmashligiga bog'liq bo'ladi.

- *Butunligi.* Agar MB tarkibida ko'p foydalanuvchilar ishlatadigan ma'lumotlar bo'lsa, unda ma'lumotlar elementlari

va ular orasidagi bog'lanish mumkin qadar buzilmasligi kerak. Ma'lumotlarni saqlash, ularni yangilash, ma'lumotlarni qayta ishlash tartibi shunday bo'lishi kerakki, mabodo tizimda biror o'zgarish bo'lib qolsa, ma'lumotlarni yuqotishsiz qaytadan tiklash mumkin bo'lsin. Bulardan tashqari, ma'lumotlarni har xil tizimli o'zgarishlardan saqlash uchun ularning qiymatlarini biror aniq o'lchovlarga mos kelishligini ta'minlashda oldindan ko'zda tutilgan butunligini tekshirish amalga oshiriladi.

- *Xavfsizlik va maxfiylik.* Ma'lumotlar yashirilishi va maxfiy saqlanmoqligi lozim. U yo'qotilmasligi yoki o'g'irlanmasligi kerak. Ma'lumotning yashash chidamligini oshirish uchun uni texnik yoki dasturiy o'zgarishlardan, tabiiy va sun'iy vaziyatlardan va yomon niyatda foydalanishlardan saqlamoq lozim.

Ma'lumotlarning xavfsizligi deganda, ma'lumotlarni tasodifiy yoki ularga bila turib kirishga haqqi yo'q shaxslardan himoya qilinishi tushuniladi.

Maxfiyligi esa ayrim shaxslarning yoki takshilotlarning qachon va qanday miqdorda boshqa shaxslarga yoki tashkilotlarga ma'lumotlarni berish huquqi bilan aniqlanadi.

- *Eng kam sarf.* (Minimal harajatlar) MBni tuzish va undan foydalanishdagi xarajatlarni kamaytirish maqsadida, tashkil qilishni shunday usullari tanlanadiki, ular vaqt va xarajatlarni imkon darajada kamaytirsin.

- *Qidiruv imkoniyatlari.* MBdan foydalanuvchi MBga, turli xil shakldagi ma'lumotlarni olish uchun murojaat qilishi mumkin. Tizim oldindan rejalashtirilmagan turli talablarni qayta ishlashni ta'minlashi yoki shunday javoblarni tuza olishi kerak.

- *O'tgan zamon bilan bog'lanish.* Tashkilot ixtiyoriy vaqtda oldingi, eski ma'lumotlarini ko'ra olishi lozim.

- *Kelajak bilan bog'lanish.* Kelajak bilan bog'lanishni tasavvur etish muhimdir. Kelajakda ma'lumotlar va ularni saqlash muhiti ko'p yo'nalishlar bo'yicha o'zgaradi. Har qanday tijorat tashkiloti vaqt o'tishi bilan o'zgarishlarga duchor bo'ladi. Ayniqsa, bu o'zgarishlar ma'lumotlarni qayta ishlaydigan foydalanuvchilar

uchun qimmatlidir. Oddiy o'zgarishlarni amalga oshirish uchun talab qilinadigan juda katta xarajatlar, bu tizimlarning rivojlanishiga ko'chli to'siq bo'ladi. MBni sinchiklab o'rganishda muhim masalalardan biri, MBni shunday loyihalash kerakki, ularning o'zgarishini amaliy dasturlarni o'zgartirmasdan turib bajarish mumkin bo'lsin.

- *Sozlash.* Ma'lumotlarning unumdorligini yaxshilash maqsadida uning bazasini qayta qurish – MBni sozlash deyiladi.

MBni sozlash va ishlashga administrator javobgar bo'ladi. U mumkin qadar, amaliy dasturlarning butunligini saqlab qolgan holda MBga qanday o'zgarishlarni kiritish yetarliligini malakali aniqlay bilishi lozim[1].

Nazorat savollari

- 1) Bilimlar bazasiga ta'rif bering.
- 2) Bilimlar bazasining asosiy komponentlariga nimalar kiradi?
- 3) Ma'lumotlar bazasidan foydalanishning texnologiyasini sxematik tasvirlang?
- 4) Bilimlarga erishish bloki nimalarni aks ettiradi?
- 5) Mantiqiy xulosalar bloki nimalarni yaratadi?
- 6) Amalda ekspert tizimlaridan qanday foydalaniladi?
- 7) Ekspert tizimlaridagi bilimlar bazasi qanday maqsadlar uchun muljallangan?
- 8) Axborot ta'minoti deganda nimani tushunasiz?
- 9) Axborot tizimlariga qo'yiladigan talablar haqida tushuncha bering.
- 10) MB ning rivojlanish bosqichlarini haqida ma'lumot bering.
- 11) MBBTga ta'rif bering.
- 12) MB haqida tushuncha bering.
- 13) Ma'lumotlar banki iborasiga ta'rif bering.
- 14) Ma'lumotlar bazasiga qo'yiladigan asosiy talablar nimalardan iborat?
- 15) Yozuv haqida tushuncha bering.
- 16) Maydon va unga qo'yiladigan talablarni ayting.

II. PREDMET SOHANI TASVIRLASH USULLARI

2.1. Predmet sohani tavsiflash

Hozirgi kunda insonning ish faoliyatida MBda kerakli ma'lumotlarni saqlash va undan oqilona foydalanish juda muhim ahamiyatga ega. Sababi: jamiyatning qaysi jabhasiga nazar solmaylik kerakli ma'lumotlarni olish uchun MBga murojaat qilishga to'g'ri kelishini ko'ramiz. Demak, MBni yaratish va undan foydalanishni tashkil qilish, axborot almashuv texnologiyasining eng dolzarb muammolaridan biriga aylanib borayotgani davr taqozasidir. Shuning uchun, MBni yaratishga e'tibor qaratilmoqda. Amaliyotda MBlari asosan chegaralangan predmet soha uchun yaratiladi. Shu sababli, bir tashkilot uchun bir qancha MBlar yaratilishi mumkin.

Predmet sohasi - bu boshqarishni tashkil qilish va uni avtomatlashtirish maqsadida o'rganilishi lozim bo'lgan ob'ektiv dunyoning ma'lum bir qismidir.

Predmet soha ma'lumotlaridan foydalanish uchun ularning mazmuniy berilishi - semantikasi berilishi lozim. Ma'lumotlar semantikasini tasvirlashning asosiy vositasi - tabiiy tildir. Ba'zan, formallashtirilgan tillardan ham foydalanish mumkin.

Agar MBning vazifasi faqat ma'lumotlarni saqlash bo'lganida, uning tasnifi juda oddiy bo'lar edi. Ammo, uning boshqa har xil ma'lumotlar elementlari bilan aloqasini ta'minlashi kerakligi, MB tasnifini murakkablashtiradi. Shuning uchun, MBni yaratishda ikkita muhim shartni hisobga olmoq zarur:

Birinchidan, ma'lumotlar turi, ko'rinishi, ularni qo'llaydigan dasturlarga bog'liq bo'lmasligi lozim. Ya'ni, MBga yangi ma'lumotlarni kiritganda yoki ma'lumotlar turini o'zgartirganda, dasturlarni o'zgartirish talab etilmasin.

Ikkinchidan, MBdagi kerakli ma'lumotni bilish yoki izlash uchun biror dastur tuzishga hojat qolmasin.

Shu sababli, MBni tashkil etishda ma'lum qonun va qoidalarga amal qilish talab qilinadi.

Ma'lumotlar hajmi oshib borishi bilan masalalarni hal etish murakkablashadi. Yuzaga kelgan muammo ob'ekt va ma'lumotlarni tizimga solish yo'li bilan hal qilinadi.

Ob'ekt - bu mavjud va farqlanishi mumkin bo'lgan narsadir.

Ob'ektlarga tegishli bir qator ma'lumotlar borki, ularning to'plami MB bo'la oladi. Masalan, har bir ta'lim muassasasi ob'ekt bo'lsa, ulardagi o'quvchilar haqidagi ma'lumotlar to'plami MBga misol bo'la oladi.

Har qanday jiddiy MBning yaratilishi uning loyihasini tuzishdan boshlanadi. MB ni loyihalashtirishda faoliyat sohasi tahlil qilinadi va barcha foydalanuvchilar talablari o'rganib chiqiladi. Odatda, MBni loyihalash vazifasi bir kishiga yoki guruhga yuklatiladi. Ularga tashkilot tomonidan asosiy ish jarayonlarini tushunadigan va qiymatlarga kompyuter orqali ishlov berish haqida tushunchaga ega bo'lgan xodim biriktiriladi. MB loyihalovchisining asosiy vazifasi ob'ektlar va ularni tavsiflovchi parametrlarni tanlash, ma'lumotlar orasidagi bog'liqliklarni o'rnatishdan iborat. MBni yaratish jarayonida, foydalanuvchi ma'lumotlarni turli belgilar bo'yicha tartiblashga va belgilarning turli birikmalari bo'yicha zarur ma'lumotlarni (tanlanmani) tez topish uchun imkoniyatlar yaratilishiga harakat qiladi. Bunday ishlarni ma'lumotlar tuzilmalangan (strukturalangan) bo'lgandagina bajarish mumkin.

Tuzilmalash - bu ob'ektlar va ma'lumotlarning o'zaro bog'lanishini tasvirlash

usullari haqidagi kelishuvni kiritishdir.

1-misol. Tuzilmalanmagan ma'lumotlar.

Shaxsiy ish №16493; Aliev Karim ergashevich; tug'ilgan sanasi 1 yanvar 1979 yil; Shaxsiy ish №16498; Boqiev Dilmurod Rahmatullaevich; tug'ilgan sanasi 5 dekabr 1985 yil; Shaxsiy ish №16595; Zokirov Anvar Rashidovich; tug'ilgan sanasi 15 may 1984 yil.

2-misol. Tuzilmalangan ma'lumotlarning ko'rinish 2.1.1-jadvalda keltirilgan.

Ma'lumotlarni saqlash - kompyuterning muhim vazifalaridan biri hisoblanadi. MB esa bu jarayonda ishtirok etuvchi eng muhim omil sanaladi.

2.1.1-jadval

Tuzilmalangan ma'lumotlar

SHaxsiy ishi №	Familiyasi	Ismi	Otasining ismi	Tug'ilgan sana
16493	Aliev	Karim	Ergashevich	01.01.79
16498	Boqiev	Dilmurod	Rahmatullaevich	05.12.85
16595	Zokirov	Anvar	Rashidovich	15.05.84

Hozirgi kunda MBlarining aksariyat ko'pchiligi jadvalsimon tuzilishga ega. Jadvalsimon tuzilishda kerakli ma'lumotlar qator va ustunlarning kesishishi bilan aniqlanadi. MBda ustunlar - maydonlar, qatorlar - yozuvlar deyiladi. Maydonlar MB tuzilmasini tashkil qilsa, yozuvlar esa unda saqlanuvchi ma'lumotni o'zida aks ettiradi.

Maydon yoki rekvizit – ma'lumotlarni tashkil etishning oddiy birligi bo'lib, u axborotning alohida, bo'linmas birligidir.

MB tuzilmasining asosiy elementlari — bu maydonlardir. Ular bir qancha xususiyatlarga ega. Xususiyatlarga bog'liq holda maydonga qanday axborot kiritish mumkin va undagi axborot bilan qanday vazifani bajarsa bo'ladi singari ishlar bajariladi.

Masalan, "Narx" maydonida joylashgan ma'lumotlarni qo'shish yo'li bilan ma'lum bir natija olishimiz mumkin. Lekin, 'Telefon nomeri' deb nomlangan maydondagi ma'lumotlarni (telefon nomerlari raqam bilan yozilgan bo'lsa ham) qo'shish mumkin emas. Demak, bu maydonlar turli xususiyatlarga ega va har xil tiplarga taaluqlidir.

Har qanday maydonning asosiy xususiyati uning uzunligi hisoblanadi. Maydonga qancha ma'lumot joylashishi uning uzunligiga bog'liq. Maydon uzunligi ramziy belgilar bilan

ifodalanadi. Ramziy belgilar baytlar bilan kodlanadi hamda maydon uzunligini shartli ravishda baytlarda o'lchash mumkin. Maydonning noyob xususiyatlaridan biri — uning nomidir. MBning bir ob'ekti bir xil nomdagi ikki maydonga ega bo'la olmaydi, chunki kompyuter ularning tarkibidan adashib ketadi. Ammo, maydonning nomidan tashqari — imzo deb nomlangan yana bir xususiyati bor. Imzo — ustunning sarlavhasini ifodalovchi axborotdir. Agar imzo berilmagan bo'lsa ham, uni maydon nomi bilan almashtirmaslik lozim. Turli maydonlarga bir xil imzo berish mumkin. Bu kompyuterning ishiga xalaqit bermaydi, chunki maydonlar turli nomlarga ega bo'lib qolmoqda.

Turli tipdagi maydonlardan har xil maqsadda foydalaniladi va ular turli xususiyatlarga egadir.

Matnli maydonning asosiy xususiyati uning hajmidir.

Raqamli maydon raqamli ma'lumotlarni kiritish uchun xizmat qiladi va u ham o'lchamga ega. Raqamli maydonlar har xil bo'ladi. Masalan, butun va haqiqiy sonlarni kiritish uchun. Oxirgi holatda maydon o'lchamida sonning o'nlik qismini o'lchami ham bo'ladi.

Vaqt yoki sanani kiritish uchun maydonlar sana\vaqt ko'rinishiga egadir.

Ikki xil qiymatga ega bo'lgan mantiqiy ma'lumotlarni kiritish uchun esa (Ha yoki Yo'q, 0 yoki 1 va h.k.) maxsus tip—mantiqiy maydon xizmat qiladi. Bunday maydonning o'lchami 1 baytga teng. Bu mantiqiy qiymatni ifodalash uchun etarlidir.

Zamonaviy MB o'zida faqat raqam va harflarnigina emas, balki rasm, musiqali klip va videoyozuvlarni ham saqlashi mumkin. Bunday ob'ektlar uchun mo'ljallangan maydon — OLE ob'ekti maydoni deb ataladi.

Yozuv — mantiqan bog'langan maydonlardir. Har bir yozuvda yagona yozuv kaliti mavjud bo'lib, ular dastlabki (birlamchi) va ikkilamchi bo'lishi mumkin.

Dastlabki kalit (DK) – yozuvni ma'no jihatdan birxillashtiruvchi bir yoki bir necha maydonlardir. Dastlabki kalit bir maydondan iborat bo'lsa u oddiy, agar bir necha maydonli bo'lsa – tarkibli kalit hisoblanadi.

Ikkilamchi kalit (IK) – dastlabki kalitdan farqli maydon bo'lib, u yagona emas. Agar dastlabki kalitning mazmuniga ko'ra faqat bitta yozuv nusxasi topilsa, ikkinchi kalit bo'yicha bir necha nusxa topilishi mumkin.

Yangi yaratilgan MBda hech qanday ma'lumot yo'q deb tasavvur qilamiz. Ammo, unda baribir ma'lumot bor. Bu ma'lumot MBning tuzilmasi, ya'ni maydonlar to'plamidir. Ular bu bazaga nima va qanday holatda yozilishini aniqlab beradi [1].

2.2. Ma'lumotlar bazasini yaratish bosqichlari

Ma'lumotlar modeli va modellash Model (lat. modulus – o'lchov, me'yor) ob'ekt, ob'ektlar majmuasi yoki biron jarayonni matematik munosabatlar, tekst, jadvallar yoki boshqa yo'l bilan formal ta'riflashdir. U, ob'ekt yoki ob'ektlar majmuasining obrazi yoki namunasidir. Masalan, Erning modeli globus, osmon va undagi yulduzlar modeli planetariy ekrani, odam suratini shu surat egasining modeli deyish mumkin. Model o'rganilayotgan hodisa va jarayonni iloji boricha to'la aks ettirishi zarur.

Modelning taqribiylik xarakteri turli ko'rinishda namoyon bo'lishi mumkin. Masalan, tajriba o'tkazish mobaynida foydalaniladigan asboblarning aniqligi, olinayotgan natijaning aniqligiga ta'sir etadi. Samolyotning ob-havo sharoitini hisobga olmay tuzilgan uchish jadvali, aeroflot ishining taqribiy modelini ifodalaydi va h-zo.

Modellash – bilish ob'ektlari (fizik hodisa va jarayonlar) ni ularning modellari yordamda tadqiq qilish, mavjud narsa va hodisalarning modellarini yasash va o'rganishdan iborat.

Modellashtirish uslubidan hozirgi zamon fanida keng foydalanilmoqda. U ilmiy-tadqiqot jarayonini osonlashtiradi, ba'zi hollarda esa murakkab ob'ektlarni o'rganishning yagona vositasiga aylanadi. Modellashtirishning, ayniqsa, mavhum ob'ektlarni, olis-olislarda joylashgan ob'ektlarni, juda kichik hajmli ob'ektlarni o'rganishda ahamiyati kattadir. Modellashtirish uslubidan fizika, astronomiya, biologiya, iqtisod fanlarida ob'ektning ma'lum xususiyat va munosabatlarini aniqlashda foydalaniladi.

MBni yaratish bosqichlari. MBni yaratishda quyidagi bosqichlarni ajratib ko'rsatish mumkin:

1. *Muammoning qo'yilishi.* Ushbu bosqichda MBni yaratish uchun vazifa shakllantiriladi. Unda bazaning tarkibi, nima uchun ishlatilishi, yaratish maqsadi batafsil bayon etiladi. Shuningdek, ushbu MBda qanday turdagi ishlarni bajarish mo'ljallanayotganligi (tashlash, qo'shish, ma'lumotlarni o'zgartirish, hisobotni ekranda chiqarish yoki chop etish va h-zo) sanab o'tiladi.

2. *Obyektning tahlili.* Ushbu bosqichda MB qanday ob'ektlardan tuzilishi mumkinligi va ularning xususiyatlari, ya'ni ob'ekt qanday parametrlar bilan aniqlanishi ko'rib chiqiladi. Barcha ma'lumotlarni alohida yozuvlar yoki jadvallar ko'rinishida tasvirlash mumkin. Shundan so'ng, har bir alohida yozuv birligining turi (matnli, sonli va h-zo) aniqlanadi.

3. *Model sintezi.* Ushbu bosqichda yuqoridagi tahlil asosida MB modeli tanlanadi (Relyatsion, iyerarxik, tarmoqli va h-zo). Modelning yaratilayotgan MB talablariga javob berishligi, qo'yilgan masalani echish imkoniyatiga ega bo'lishligi singari xususiyatlari ko'rib chiqiladi.

4. *MB jadvallarining tuzilmasini yaratish.* (Hozirgi kunda relyatsion model eng ko'p ishlatilayotganligi bois MB tuzilmasini yaratishni jadval nuqtai nazaridan ko'rib chiqamiz). Tanlangan modelga muvofiq MB sxemasi, jadvallar, jadval maydonlari,

jadvallar va tugunlar orasidagi bog'lanishlar yaratiladi.

MBning yangi jadvalini shakllantirish, uning tuzilmasini yaratishdan boshlanadi. Ushbu jarayon maydonlar nomlarini, ularning turlari va o'lchamlarini aniqlashni o'z ichiga oladi.

5. *Axborotni tasvirlash usullari, dasturiy uskunasi.* MB sxemasi yaratilgandan so'ng dasturiy mahsulotga bog'liq holda axborotni tasvirlash usulini aniqlab olish kerak. Ko'pchilik MBBTda ma'lumotlarni ikki xil ko'rinishda tasvirlash mumkin:

- shakllardan foydalanib;
- shakllardan foydalanmasdan;

Shakl - foydalanuvchi tomonidan MBga ma'lumotlarni kiritish va ma'lumotlarni tasvirlash uchun yaratilgan ekran grafik interfeysidir.

6. *Obyektning kompyuter modeli sintezi va uni yaratish texnologiyasi.* Tanlab olingan dasturiy mahsulotning uskunaviy imkoniyatlarini ko'rib chiqib, kompyuterda MBni bevosita yaratishga kirishish mumkin. MBning kompyuter modelini yaratish jarayonida har qanday MBBT uchun tipik bo'lgan ayrim bosqichlarni ajratib ko'rsatish mumkin:

- MBBTni ishga tushirish, MBning yangi faylini yaratish yoki oldindan yaratilgan bazani ochish;
- dastlabki jadvalni yoki jadvallarni yaratish;
- interfeysni yaratish;
- MBni to'ldirish.

MBni to'ldirish ikki ko'rinishda olib boriladi: Bunda sonli va matnli maydonlarni jadval ko'rinishida, MEMO va OLE turidagi maydonlarni shakl ko'rinishida to'ldirish lozim.

7. *Yaratilgan MB bilan ishlash.* MB bilan ishlash deganda quyidagi imkoniyatlar nazarda tutiladi:

- kerakli axborotlarni izlash;
- ma'lumotlarni saqlash;
- ma'lumotlarni tanlab olish;
- chop etish;

- ma'lumotlarni o'zgartirish va to'ldirish.

MBda ishlash tamoyillaridan quyidagilar eng muhim hisoblanadi:

1. *Ma'lumotlarni kiritish va tahrirlash.* Deyarli barcha MBBTlar jadvallarga ma'lumotlar kiritish va ularni tahrirlashga imkon beradi. Bu ishlarni ikki usulda bajarish mumkin:

- jadval ko'rinishda taklif etiladigan standart shakllar yordamida;
- foydalanuvchi tomonidan maxsus yaratilgan ekranli shakllar yordamida.

2. *Jadvaldagi ma'lumotlarga ishlov berish.* MB jadvallaridagi ma'lumotlarga so'rovlar berish yo'li bilan yoki maxsus ishlab chiqilgan dasturni bajarish orqali ishlov berish mumkin. So'rov deganda yozuvlarni tanlash uchun beriladigan ko'rsatma tushuniladi. So'rovni bajarish natijasida ma'lumotlar to'plami (dinamik to'plam)dan iborat vaqtinchalik jadval hosil bo'ladi. Dinamik to'plamning yozuvlari bir yoki bir necha jadvaldan iborat maydonlarni o'z ichiga olishi mumkin. So'rov asosida hisobot yoki shaklni tuzish mumkin.

3. *MBdan axborotni chiqarish.* Har qanday MBBT kompyuter ekraniga yoki chop etish qurilmasiga maxsus shaklda MBdagi ma'lumotlarni chiqarishga imkon beradi. MBBT bilan ishlayotgan foydalanuvchi chop etish qurilmasiga ma'lumotlarni chiqarish uchun hisobotlarni tuzishning maxsus vositalaridan foydalanish imkoniyatiga ega bo'ladi.

Hozirgi ko'pchilik MB zaminida jadvallar yotadi. Shuning uchun, qo'yilgan masalada ishlatiladigan ma'lumotlarni jadvallar orqali ifodalash, keyin esa so'rov, shakl va boshqa ob'ektlarni yaratish mumkin. Bunday yondashish nisbatan sodda masalalarni avtomatlashtirishda qo'l kelishi mumkin. Murakkabroq masalalarda MBni qo'yilgan talablarga javob beradigan holga keltirish uchun yaratilgan ob'ektlarni qayta-qayta o'zgartirish lozim bo'ladi. Shu sababli, boshlang'ich

bosqichda bo'lajak MBni loyihalashga vaqtni ayamagan ma'qul. Loyihalash jarayonida quyidagi savollar yoritiladi:

- MB qanday talablarga javob berishi kerak?
- Bu talablarni bajarish uchun qanday masalalar echilishi kerak?
- Buning uchun qanday ma'lumotlar ishlatiladi?
- Ularning tuzilmasi va tipi masalalar orasidagi qanday bog'liqlik bor va h-zo.

MB – MBBT vositalari orqali mashina "tashuvchi"da hosil qilingan mantiqiy bog'langan ma'lumotlar yig'indisidir. MB o'zaro bog'langan ob'ektlarning mantiqiy modelini aks ettiradi va u aniq predmet sohasini o'zida namoyon etadi. MB ko'p foydalanuvchilarga zarur bo'lgan ma'lumotlarni o'z ichiga oladi. Bir vaqtning o'zida bir necha foydalanuvchilar umumiy MBga kirishi mumkin. Bu imkoniyat MBBT kompyuterlar tarmog'iga ulanganda va ko'pchilik foydalanadigan MB yaratilganda mumkin bo'ladi.

MBBT tarmoq orqali turli foydalanuvchilar MB ma'lumotlaridan foydalanganda ma'lumotlar himoyalanganligini ta'minlaydi hamda ular tomonidan kiritilayotgan ma'lumotlar cheklanishlarini kuzatib boradi. Bir vaqtning o'zida bir necha mijozlar ma'lumotlardan foydalanganda ularning muvofiqligi, himoyalaniishi avtomatik ravishda amalga oshiriladi.

MBBT va MBlarining asosiy xususiyatlari quyidagilardan iborat:

- ma'lumotlarni bir marta kiritish va ularni muvofiqlashtirishning oddiyligini ta'minlovchi modelning turli ob'ektlarida ma'lumotlarning takrorlanishining yo'qligi;
- ma'lumotlarni bir-biriga zid emasligi;
- MBning yaxlitligi;
- ko'p jihatli kirishning mumkinligi.
- ma'lumotlarni har xil tanlash va ulardan foydalanuvchining turli topshiriqlar va ilovalarda foydalanishi;

- avariya holatlari, apparat va dastur buzilishi yoki foydalanuvchining xatolari paytida ma'lumotlarni himoya qilish va tiklash;

- mijozlarning beruxsat kirishlarini cheklash vositalari yordamida himoya qilish;

- ma'lumotlarni qayta o'zgartirmay MB tuzilmalarini modifikatsiya qilish

- imkoniyati;

- MB tuzilmalarini modifikatsiyalashda dasturlarni o'zgartirmasdan, ma'lumotlar mustaqilligini ta'minlash;

- mashina "tashuvchisi"dagi MBni joylashtirishni hajm-vaqt tavsifini yaxshilash maqsadida qayta tashkil etish;

- mijozga har qanday talabi bo'yicha mavjud axborotlarni MBdan chiqarib berishni va foydalanuvchi uchun qulay hisobot shakllari ko'rinishida taqdim etishni ta'minlovchi yuqori darajali talablar tilining mavjudligi.

Foydalanuvchi tomonidan MBBTni tanlash mezonlari.
Foydalanuvchining MBBT ni tanlashi quyidagi omillar bilan belgilanadi:

- mavjud texnikaviy va dasturiy ta'minlanish, operativ va disk xotiralari;

- tanlangan ma'lumotlar modelining turi, predmet sohasining xususiyati, axborotlar mantiqiy modelining topologiyasi;

- foydalanuvchining ishlab chiqilayotgan ilovalarga ehtiyojlari;

- ma'lumotlarga ishlov berishda mahsuldorlik ko'rsatkichiga qo'yiladigan talablar;

- MBBTda zaruriy vazifalar vositalarining mavjudligi;

- mijozlarning malakasi, darajasi va MBBTda ishlash uchun muloqot va ishlash vositalari mavjudligi.

MBBTni o'rnatish. MBBT dastur mahsuloti bo'lib, amaliy dasturlar paketi ko'rinishida etkazilib beriladi. U shakllari,

zahirilari va operatsion tuzilishi, shuningdek vazifalar to'plash uchun bo'lgan talablari hisobga olingan holda kompyuterga o'rnatilishi (installashtirilishi) zarur.

Bosqichma – bosqich joriy etish jarayoni. MBBTni o'rnatgach, MBni kompyuterda yaratish, jumladan MB tuzilmalarini joylashtirish, ma'lumotlarni kiritish, shuningdek MBBT funktsional imkoniyatlari nazarda tutilgan har qanday harakatlarni bajarish mumkin. Shaxsiy kompyuterlarga mo'ljallangan MBBT lar etarlicha ixcham bo'ladi. Bu foydalanuvchi ilovalarini ishlab chiqishning ilk bosqichlaridayoq MBning alohida qismlarini yaratishga kirishishga imkon beradi. Bunday MBni ishlab chiqish chuqurlashgani sayin, u oson kengayishi va modifikatsiyalanishi mumkin. MB bilan ishlash texnologiyasi xodimlar tomonidan tez o'zlashtirilsa, MBBTni bosqichma – bosqich joriy etish osonlashadi [1].

2.3. "Mohiyat-alloqa" usuli

Loyihalashda o'rganilgan ma'lumotlar, foydalanuvchilar takliflari va ish jarayonini tasavvur qilishga asoslangan holda MBning boshlang'ich xomaki nusxasi tayyorlanadi. Tabiiy til orqali matematik formulalar, jadvallar, grafiklar va barcha kishilarga tushunarli bo'lgan vositalardan foydalanib tayyorlangan MB loyihasi ma'lumotlarning infologik modeli deb ataladi. Infologik modellashtirishning maqsadi MBda saqlanadigan ma'lumotlarni yig'ish va taqdim qilishni inson tushunadigan tilda ifodalashdir. Shuning uchun, ma'lumotlarning infologik modelini tabiiy tilga yaqin usulda tavsiflashga harakat qilinadi. Infologik modelning asosiy qurish elementi mohiyat, ular orasidagi bog'lanishlar va xossalari (atributlari) hisoblanadi.

Mohiyat-MBda saqlanishi kerak bo'lgan ma'lumot, ixtiyoriy bir-biridan farq qiluvchi ob'ektdir. Insonlar, joy, samolyotlar, reyslar, ta'm, rang va h-zolar mohiyat bo'lishi mumkin.

Mohiyatning tipi va nusxasi mavjud. Bir butun majmua sifatida qaralayotgan shaxslar, fanlar, hodisalar yoki g'oyalarni mohiyat tipi tushunchasi deb qarashimiz mumkin. Majmuadagi aniq narsalar mohiyat nusxasiga tegishli bo'ladi. Masalan, 'Shahar' mohiyat tipi bo'lsa, 'Buxoro' esa mohiyat nusxasi hisoblanadi.

Atribut-mohiyatning nomlangan tavsifidir. Nomlash aniq bir mohiyat tipi uchun takrorlanmas, ammo turli xil mohiyat tiplari uchun bir xil bo'lishi mumkin. (Masalan, "Rang" atributi "It", "Avtomobil", "Gul", "Tutun" va h-zo singari mohiyatlar uchun aniqlanishi mumkin). Atribut mohiyatda qanday ma'lumotlar saqlanishi kerakligini aniqlash uchun ishlatiladi. Masalan, 'Avtomobil' mohiyati uchun "Tip", "Tur", "Raqami", "Rang" va h-zolar atributga misol bo'la oladi. Bu yerda tip va nusxa o'rtasida farq mavjudligi namoyon bo'ladi. "Rang" atributi tipi juda ko'p qiymat yoki nusxaga ega bo'ladi:

"Qizil", "Ko'k", "Sariq", "Oq" va h-zo.

Mohiyatning har bir nusxasiga atributning birgina qiymati taqdim qilinadi.

Mohiyatning tipi va atributi orasidagi mutloq farq mavjud emas. Mohiyat tipiga bogliq holda atribut o'z o'rniga ega bo'lishi mumkin. Boshqa hollarda atribut mustaqil mohiyat sifatida qatnashishi mumkin. Masalan, avtomobil zavodi uchun 'rang' ishlab chiqarilayotgan mahsulot atributi bo'lsa, rang ishlab chiqaruvchi fabrikada 'rang' mohiyat tipi hisoblanadi.

Kalit-atributlarning kam sonli to'plami bo'lib, ular qiymatlari bo'yicha mohiyatning kerakli nusxasini topish mumkin. To'plamdan biror bir atributni chiqarganda, qolgan atributlar yordamida mohiyatning mutanosibligini belgilay olmaslik, kamlik tushunchasini bildiradi. Masalan, biror bir oliy ta'lim muassasi xodimlari haqidagi ma'lumotlar to'plami 'Xodim' mohiyati bo'lsa, unda 'tabel rakami' kalit hisoblanadi.

Bog'lanish-ikki yoki undan ortiq mohiyatlarni birlashtirishdir. Chunki, MBni tashkil qilishning shartlaridan biri,

mohiyatlar orasida o'zaro bog'liqlikni tashkil qilib, mohiyatning biror-bir qiymati orqali boshqa mohiyatni izlashdan iborat bo'ladi.

Mavjud MB yuzlab, hattoki minglab mohiyatlarni umumlashtiradi va nazariy jihatdan mohiyatlar orasida millionlab bog'lanishlarni tashkil qilish mumkin. Ko'plab bog'lanishlarning mavjudligi infologik modelning murakkabligini belgilaydi.

Bog'lanishlar tavsifi va modellashtirish tillari. Infologik modelni qurishda ER diagramma (inglizcha Entity-relationship mohiyat-bog'lanish) tilidan foydalanish mumkin. Unda mohiyat to'g'ri turtburchak, birlashtirish romb yoki oltiburchak, atributlar oval, bog'lanishlar esa yo'nalishga ega bo'lmagan va yuqorisida bog'lanish darajasi qo'yilgan chiziq shaklida ifodalanadi. Bog'lanish darajasi 1 yoki 'ko'p' so'zini bildiruvchi M harfi bilan belgilanadi.

Ikki mohiyat, masalan A va B orasida 4 xil bog'lanish bo'lishi mumkin.

Birinchi tip-"birga-bir bog'lanish" (1:1). Har bir vaqtda A mohiyatning har bir nusxasiga B mohiyatning 1 yoki 0 nusxasi mos keladi.



2.3.1-rasm. Birga-bir bog'lanish

Talaba stipendiya olmasligi yoki odatdagidek olishi yoki yuqori turlardan birini olishi mumkin.

Ikkinchi tip-"birga-ko'p bog'lanish" (1:M). A mohiyatning bir nusxasiga B mohiyatning 0 yoki 1 yoki bir necha nusxasi mos keladi.



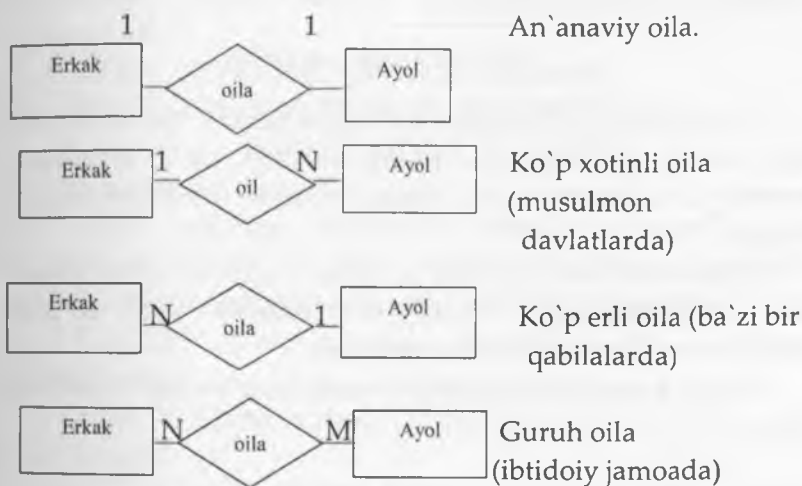
2.3.2-rasm. Birga-ko'p bog'lanish

Xonadon bo'sh bo'lishi yoki unda 1 yoki bir necha kishi yashashi mumkin.

Ikki mohiyat orasida yana quyidagicha bog'lanishlar bo'lishi mumkin. "Ko'pga-bir" (M:1) va "Ko'pga-ko'p" (M:N).

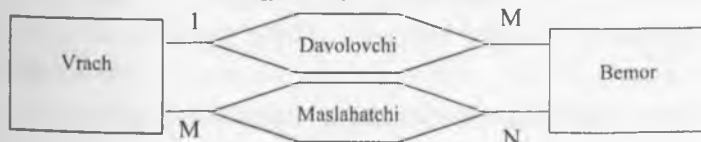
2.4. "Mohiyat-aloha" diagrammasi. Mohiyatlar klassifikatsiyasi

1-misol. Ayol va erkak mohiyatlari orasida dunyo bo'yicha "Oila" deb ataluvchi 4 xil bog'lanish mavjud bo'lishi mumkin.



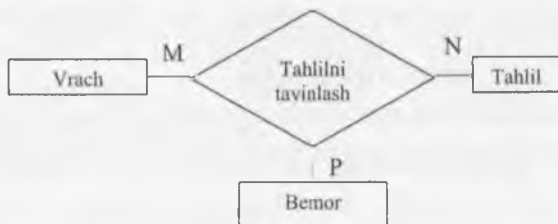
2.4.1-rasm. Ayol va erkak mohiyatlarining bog'lanishi

2-misol. Vrach va bemor mohiyatlari orasida bir necha bog'lanish mavjud bo'lishi mumkin.



2.4.2-rasm. Vrach va bemor mohiyatlarining bog'lanishi

Bemor bitta davolovchi vrachga ega bo'lgan holda bir necha maslahatchi vrachga murojaat qilishi, vrach esa bir necha bemorni davolashi hamda bir necha bemorga maslahatchi bo'lishi mumkin.

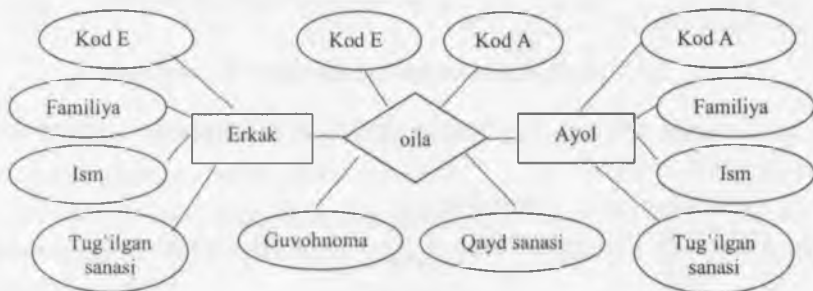


2.4.3-rasm. Trenar bog'lanish

Vrach bir necha bemorga bir necha tahlilni tayinlashi, bir necha vrach tomonidan bir necha bemorga tahlil tayinlashi, bemorga bir necha vrach tomonidan bir necha tahlil tayinlanishi mumkin.

Mohiyatlar orasidagi bog'lanishlar yuqorida keltirilganlar bilan chegaralanmaydi. Mohiyatlar orasida juda ko'plab murakkab bog'lanishlar bo'lishi mumkin.

Yuqori darajali bog'lanishlar semantikolari ba'zan murakkab bo'ladi.



2.4.4-rasm. Oila bog'lanishining ER diagrammasi

Keltirilgan misolda ko'rgazmalilikni oshirish uchun bog'lanishlarda mohiyat atributi va birlashtirishlar, barcha ER

diagrammalarda keltirilmadi. Agar mohiyatlar atributi ham bog'lanishlarda tasvirlansa ER diagramma murakkab ko'rinishga ega bo'lar edi. Masalan, oila bog'lanishida asosiy atributlarni bergan holda ER diagrammani quramiz (2.4.4-rasm).

Agar ER diagrammada mohiyatlarda ishtirok etuvchi barcha atributlarni tasvirlasak tushunish qiyinlashadi. Shuning uchun, katta bo'lmagan modellarda va modellarning qismlarida ER diagramma tilidan foydalaniladi. Ko'pchilik hollarda kamroq ko'rgazmali, ammo ko'proq ifodalanuvchi infologik model tili qo'llaniladi. Unda mohiyat va birlashtirishlar quyidagicha tasvirlanadi.

MOHIYAT (atribut1, atribut2, ..., atributn)

BOG'LANISH [MOHIYAT S1, MOHIYAT S2, ...]

(atribut1, atribut2, ..., atributn)

S-bog'lanish darajasi.

Kalit tarkibiga kiruvchi atributlar chiziqcha bilan belgilanadi. Vrach va bemor sifatida keltirilgan mohiyatlar misolida bog'lanishlar quyidagicha ifodalanishi mumkin.

VRACH (Vrach-raqami, Familiya, Ism, Ota-ismi, Mutaxassisligi)

BEMOR (Qayd-raqami, O'rin-raqami, Familiya, Ism, Ota-ismi, Manzil,

Tug'ilgan-sana, Jinsi)

DAVOLOVCHI-VRACH [Vrach1, BemorM]

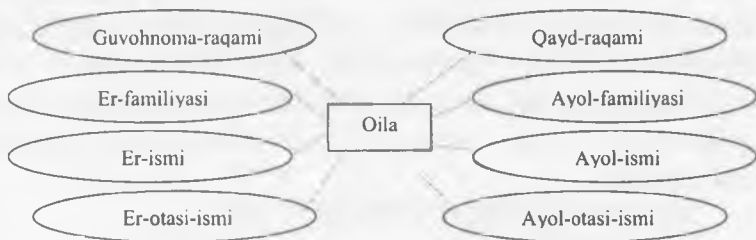
(Vrach-raqami, Qayd-raqami)

MASLAHATCHI [VrachM, BemorN]

(Vrach-raqami, Qayd-raqami)

Mohiyatlar orasidagi bog'lanishlarni aniqlash uchun eng kamida mohiyatning o'zini aniqlash lozim. Ammo, bu ancha murakkab masala hisoblanadi. Chunki turli xil faoliyat sohalarida bir obyektning o'zi mohiyat yoki atribut yoki birlashtirish bo'lishi mumkin.

3-misol. Fuqarolik holatlarini qayd qilish (ZAGS) bo'limlarida faqat oilani qayd qilish, tug'ilish va o'lim holatlarida murojaat qilgan kishilar qaydi yuritiladi. Shuning uchun, an'anaviy oila shakllangan mamlakatlarda ZAGS bo'limlari yagona mohiyat asosida ma'lumotlarni qayd qiladi.



2.4.5-rasm. Oila mohiyati uchun ER diagramma

OILA (Guvohnoma-raqami, er-familiyasi, er-ismi, er-otasi-ismi,

er-tug'ilgan-sana, Ayol-familiyasi, Ayol-ismi, Ayol-otasi-ismi, Ayol-tug'ilgan-sana, ..., Qayd-sanasi, Qayd-manzili)

4-misol. Ko'p xotinlilik ruxsat berilgan mamlakat ZAGS bo'limida qaydni yuritishda mohiyatlar orasidagi bog'lanishni ko'rib chiqamiz. Agar nikohlarni qayd qilishda "Oila" mohiyatini ishlatsak bir necha xotini bor er haqidagi ma'lumotlar takrorlanadi.

2.4.1-jadval

Mohiyatlar bog'lanishi jadvali

Qayd raqami	Er ismi		Xotin ismi		Qayd sanasi
1-YUB 154417	Ibrohim		Fotima		01.03.92
1-YUB 155811	Ibrohim		Go'zal		17.08.98
1-YUB 172116	Ibrohim		Xadicha		13.04.04
1-YUB 171701	Ahmad		Amina		12.01.93
1-YUB 170110	Yusuf		Karima		14.06.93
1-YUB 170197	Yusuf		Olima		15.06.99

Takrorlanishni yo'qotish uchun qo'shimcha "Er" nomli mohiyat yaratamiz.

ER (KodM, Familiya, Ism, Otasi-ismi, Tug'ilgan-sana).

Er haqidagi barcha ma'lumotlar quyidagi jadvalda tasvirlanadi.

2.4.2-jadval

Mohiyat atributlari

Kod M	Er ismi			Tug. sanasi
111	Ibrohim			22.03.70
112	Ahmad			12.11.73
113	Yusuf			04.06.71

Endi "Oila" mohiyatda "Er" mohiyatining mos tavsifini havola sifatida kiritamiz. Birinchi keltirilgan jadval ikkita jadvalga ajratiladi.

2.4.3-jadval

Oila mohiyatining tavsifi

Qayd raqami	Kod E	Xotin ismi		Qayd sanasi
1-YUB 154417	111	Fotima		01.03.92
1-YUB 155811	111	Go'zal		17.08.98
1-YUB 172116	111	Xadicha		13.04.04
1-YUB 171701	112	Amina		12.01.93
1-YUB 170110	113	Karima		14.06.93
1-YUB 170197	113	Olima		15.06.99

Ayollar haqidagi ma'lumotlarni alohida jadvalda tasvirlashimiz va keyin esa "Oila" jadvalida ayol ma'lumotlari o'rniga "KodA" atributni qo'yishimiz mumkin.

Bu mohiyatlarning ER diagramma bog'lanishi 2.4.6-rasmda keltirilgan.

5-misol. Biror bir tashkilotda "Xodim" mohiyati mavjud va ularning oilaviy juftligi haqidagi ma'lumotlar zarur bo'lib qoldi. er va xotin shu tashkilotda ishlaydi.

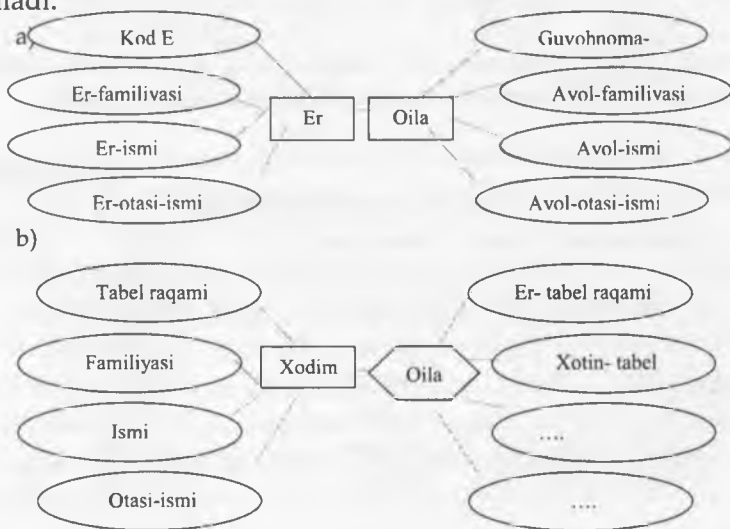
"Xodim" mohiyati:

XODIM (Tabel-raqami, Familiya, Ism, Otasi-ismi, Tug`ilgan-sana, ...).

Bu erda "Oila" mohiyati boshqacha bo'ladi. OILA [Xodim1, Xodim2]

(Er-tabel-raqami, Xotin-tabel-raqami, ...)

ER diagramma bog'lanishida chiziq mohiyat, atribut, birlashtirish va boshqa ob'ektlarni bog'laydi. Matnda esa mohiyatlarning o'zaro bog'lanishni ifodalaydi. Agar bu o'zaro bog'liqlik atributlarga ega bo'lsa, u holda u birlashtirish deb ataladi.



2.4.6-rasm. a) Oila va b) xodim mohiyatlari uchun ER diagramma

Mohiyatlar klassifikatsiyasi. K.Deyt mohiyatlarning uchta sinfi mavjudligini qayd qiladi. Ular prujinali, birlashmali va tavsiflidir.

Negiz mohiyat (Negiz)-bu mustaqil mohiyatdir. Ko'rib chiqilgan misollarda to'rtburchak ichida qayd qilingan "Talaba", "Xonadon", "Er", "Vrach", "Oila" va boshqalar negiz hisoblanadi.

Birlashma mohiyat (birlashma)-bu ikki yoki ko'proq mohiyatlar yoki mohiyatlar nusxasi orasidagi "ko'pga-ko'p" turdagi bog'lanishdir. Birlashma to'la huquqli mohiyat sifatida qaraladi.

- ular xuddi negiz mohiyat singari boshqa birlashmalarda ishtirok etishi mumkin.

- ular nafaqat bog'lanishni ko'rsatish uchun kerakli kalit atributlar to'plamiga, balki ixtiyoriy sondagi atributlarga ega bo'lish xossasiga ega. Masalan, 1 va 4-misollardagi "Oila" birlashmasi "KodM", "KodA" va "Er-tabel-raqami", "Ayol-tabel-raqami" kalit atributlar hamda "Guvohnoma-raqami", "Qayd-sanasi", "Qayd-joyi", "Qayd-raqami" singari aniqlovchi atributlarga ham ega bo'ladi.

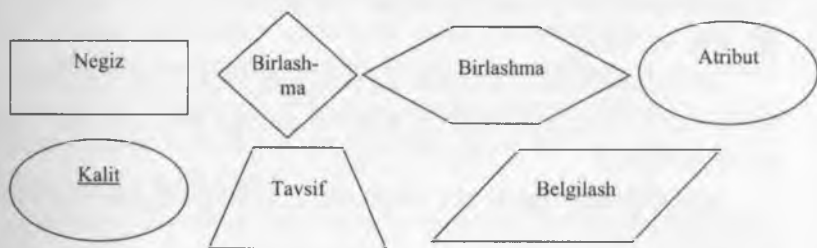
Tavsifli mohiyat (tavsif)-bu ikki mohiyat orasidagi "ko'pga-bir" yoki "birga-bir" turdagi bog'lanishlardir. Tavsifning asosiy maqsadi qaralayotgan faoliyat sohasida boshqa bir mohiyatni aniqlash yoki tasvirlashdan iborat. Haqiqiy dunyo mohiyatlari ba'zan ko'p qiymatli xossalarga ega bo'lganligi uchun ularga ehtiyoj tug'iladi. Ba'zan er bir necha xotinga ega bo'lishi, qayta nashr qilingan kitobning bir qancha tavsiflari (tuzatish, to'ldirish, qayta ishlash, ...) va h-zolar bo'lishi mumkin.

Tavsif tavsiflanuvchi mohiya'tga bevosita bog'liqdir. Agar er dunyodan o'tsa, ayol uning xotini darajasidan ayriladi.

Tavsifni yozish uchun yangi ifoda ishlatiladi.

Tavsif (atribut1, atribut2, ...) {tavsiflanuvchi mohiyatlar ro'yxati}

ER diagrammada tavsif uchun trapetsiya shakli ishlatiladi.



2.4.7-rasm. Kengaytirilgan ER diagramma tili elementlari

Belgilangan mohiyatlar yoki belgilash-bu ikki mohiyat orasidagi "ko'pga-bir" yoki "birga-bir" turdagi bog'lanish bo'lib, ular tavsifdan farqli ravishda belgilanadigan mohiyatga bog'liq bo'ladi.

Xodimni tashkilotning bir necha bo'limiga tayin qilish misolini ko'rib chiqamiz. (xodim bir bo'limda yoki bir vaqtning o'zida bir necha bo'limda qayd qilinishi yoki birorta ham bo'limda qayd qilinmasligi mumkin)

Tayinlash birlashmasi tasvirini tuzamiz.

Bo`limlar (Bo`lim-raqami, Nomi, ...)

Xodim (Tabel-raqami, familiya, ...)

Tayinlash [Bo`limN, XodimN] (Bo`lim-raqami, Tabel-raqami, Tayinlash-sanasi)

Har bir xodim albatta bir bo`limga tayinlanishi lozim degan shart asosida "Xodim" tasvirini tuzamiz.

Bo`limlar (Bo`lim-raqami, Nomi, ...)

Xodimlar (Tabel-raqami, familiya, ..., Bo`lim-raqami, Tayinlash-sanasi)

Bu misolda xodim o`z mustaqilliligiga ega. (Agar bo`lim yo`qotilsa, bundan shu bo`limga tayinlangan xodimlar ham yo`qotilishi kelib chiqmaydi). SHuning uchun, ular bo`lim tavsifi bo`la olmaydi va belgilash deb nomlanadi.

Katta hajmli va takrorlanuvchi matnlar uchun belgilashlar ishlatiladi. Masalan, talabalar o`rganayotgan fanlar 'kodlashtirishlari', tashkilot va bo`limlari nomi, mahsulotlar ro`yxati va h-zo.

Belgilashni tasviri tashqi ko`rinishda tavsifdan farq qiladi va unda belgilanadigan mohiyat figurali qavsga emas, balki kvadrat qavsga olinadi.

BELGILASH (atribut1, atribut2, ...) [Belgilangan mohiyatlar ro`yxati]

Belgilash to`la huquqli mohiyat sifatida qaralmaydi va bu hech qanday xatolikka olib kelmaydi.

Belgilash va tavsif to`liq mustaqil mohiyat hisoblanmaydi, chunki ular qandaydir boshqa mohiyatni 'belgilashi' yoki 'tavsiflashi' uchun ushbu mohiyatni bo`lishini taqoza qiladi. Ammo, ular xususiy holda mohiyat bo`lishi va xossaga ega bo`lishi, birlashmada, belgilashda ishtirok etishi hamda o`z tavsifiga (ancha quyi darajada) ega bo`lishi mumkin. Har bir tavsifning nusxasi tavsiflanuvchi mohiyatning nusxasi bilan albatta bog`lanishi shart. Ammo, tavsiflanuvchi mohiyatning

ba'zi bir nusxalari bog'lanishga ega bo'lmashligiga yo'l qo'yiladi. Haqiqatan ham bu oilaviy munosabatga tegishli bo'lsa, u holda 'er' mohiyat 'erkak' mohiyatga (xotini bo'lmagan er yo'q) almashtirilishi lozim.

Endi negiz mohiyat, mohiyat sifatida birlashma ham, belgilash ham, tavsif ham emas. Bunday mohiyatlar mustaqil mavjud bo'ladi, ba'zan esa ular boshqa mohiyatni belgilaydi. Masalan, xodimlar bo'limlarni belgilaydi [2].

2.5. Predmet sohani infologik, datalogik va jismoniy modellari

Ma'lumotlarning infologik modeli-biror bir kompyuter va MBBT ga bog'liq bo'lmagan, umumlashtirilgan jarayon sohasi tavsifidir. (ish jarayonlari, qiymatlar to'plami, qiymatlar tipi, uzunligi, o'zaro bog'liqligi va h-zo)

Insonga mo'ljallab tavsiflangan model, qiymatlarni saqlash uchun xizmat qiladigan jismoniy soha parametrlariga bog'liq bo'lmaydi. Ushbu model tavsifi kompyuter uchun emas, balki insonlarga mo'ljallanadi. Infologik model jarayon sohasini to'liq akslantirganligi uchun ish jarayonida biror bir o'zgarish sodir bo'lmasa o'zgartirilmaydi.

Infologik jihatidan ma'lumotlarni tizim xotirasida ifodalash shakliga bog'liq bo'lmagan holda mazmuni bilan bog'liq savollar qoraladi, ya'ni:

1. Tizimdagi real dunyoning qaysi ob'ektlari va jarayonlari haqidagi ma'lumotlarni yig'ish va qayta ishlash talab etilgan?
2. Ularda qaysi asosiy xarakteristkalar va o'zaro aloqalar inobatga olinadi?
3. Axborot tizimiga ob'ektlar va jarayonlar haqidagi qaysi tushunchalar kiritiladi hamda ularning qaysi xarakteristka va o'zaro aloqalarini aniqlashtirish zarur?

Shunday qilib, infologik loyihalashda tizimni axborot talabini tashkil etuvchi real dunyo, ya'ni predmet soha aniqlanadi.

Infologik model kompyuterda masalani echish jarayonlarini inson tushunchasiga muvofiq ravishda tasvirlash uchun xizmat qiladi. Bu model ma'lumotlarni saqlash parametrlariga bog'liq emas. Bunday modelni qurish uchun juda ko'plab uslublar mavjud. Ular graf modellari, semantik tarmoqlar, "mohiyat-bog'lanish" modeli va h-zolardir.

Ma'lumotlarning datalogik modeli - Infologik model MBBT uchun "tushunarli" bo'lgan kompyuterga yo'naltirilgan datologik modelda o'z aksini topishi lozim.

Datologik jihatda axborotni axborot tizimida tasvirlash savollari qaraladi. Datologik loyihalashda ma'lumotlarni qabul qilish, saqlash va qayta ishlash vositalarining imkoniyatlaridan kelib chiqqan holda tizimda axborot yordamida ma'lumotlarni mos ifodalash shakllari ishlab chiqiladi hamda axborotni tasvirlash va o'zgartirish model va metodlari keltiriladi.

Datologik modelda infologik model ma'lumotlari tuzilmasi aniq bir modelda (ierarxiya, tarmoqsimon, relyatsion) ko'riladi. Hozirgi kunda ma'lumotlar datalogik modeli relyatsion talab asosida yaratiladi.

Ma'lumotlarning jismoniy modeli- Ma'lumotlarni kompyuterda biror bir MBBTda jismoniy tashkil qilinishi va ma'lumotlar kiritilishi jismoniy modelni tashkil qiladi. Jismoniy modelda datalogik modelda yaratilgan tuzilma bevosita tanlangan MBBTda ko'riladi.

MBni qurish masalasi ikki nuqtai nazardan qaraladi, ya'ni infologik va datologik nuqtai nazardan. Infologik qarashda ma'lumotlar mashina xotirasida tasvirlanish uslublaridan qat'iy nazar, ularning mazmuniy berilishi o'rganiladi.

Datologik modelda ma'lumotlarni mashina xotiralarida tasvirlanishi o'rganiladi.

Foydalanuvchi uchun MB	MB administratori
Ishlatilayotgan MBBT model-lari va ta'riflari	Ma'lumotlarning infologik modeli – Biror bir EHM va MBBT ga bog'liq bo'lmagan, umumlashtirilgan jarayon sohasi tavsifi (ish jarayonlari, qiymatlar to'plami, qiymatlar tipi, uzunligi, o'zaro bog'liqligi va h-zo)
	Ma'lumotlarning datologik modeli – Aniq bir EHM uchun mo'ljallab yozilgan tavsifi.
	Ma'lumotlarning jismoniy modeli - Saqlanadigan ma'lumotlar tavsifi.

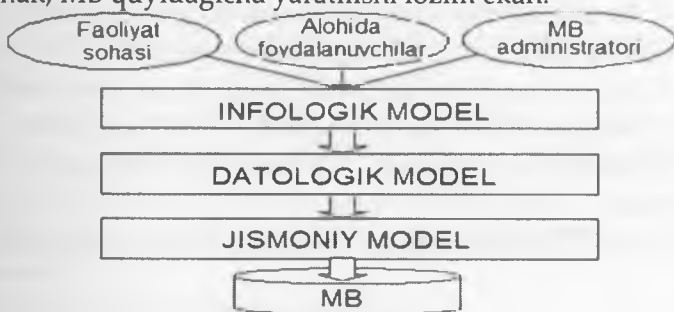
2.5.1-rasm. Ma'lumotlarni loyihalashning modellari

Datologik va jismoniy modellar kompyuterga yo'naltirilgan hisoblanadi.

Kompyuter uchun mo'ljallangan modellar aniq bir MBBTga mo'ljallab yozilishi lozim. MBBT tavsiflangan modellarga asoslangan holda qiymatlarga kirish imkonini yaratadi.

Faoliyat sohasi (MBda akslantirilgan haqiqiy dunyo qismi)

Demak, MB quyidagicha yaratilishi lozim ekan:



2.5.2-rasm. Ma'lumotlar bazasini loyihalashning 3 bosqichli arxitekturasi

Yuqoridagi ishlar amalga oshirilgandan so'ng MBBTda MB yaratiladi.

Uch bosqichli (infologik, datologik va jismoniy) arxitektura saqlanayotgan ma'lumotlarning mustaqilligini ta'minlaydi va ular foydalanilayotgan amaliy dasturlarga bog'liq bo'lmaydi. MB

administratori (MBA) zaruriyat tug'ilganda ma'lumotlarni boshqa tashqi xotiraga ko'chirishi yoki jismoniy modelga o'zgartirish kiritib, uning jismoniy tuzilmasini o'zgartirishi mumkin. MBA tizimga yangi foydalanuvchilarni biriktirishi yoki datalogik modelga o'zgartirish kiritib yangi amaliy dasturlarni qo'shishi mumkin. Tizimning datalogik va jismoniy modeliga qo'shilgan yangi tavsiflar eski foydalanuvchilarga ta'sir qilmaydi va ular o'zgartirishni sezmasligi ham mumkin. Shunday qilib, ma'lumotlarning mustaqilligi mavjud dasturlarga o'zgartirish kiritmagan holda tizimni takomillashtirish hamda rivojlantirish imkonini beradi [8].

Nazorat savollari

1. Mohiyat, atribut, kalit va bog'lanish tushinchalarini ta'riflang.
2. Birga-bir bog'lanish deganda nimani tushinasiz?
3. Birga-ko'p bog'lanishga misol keltiring?
4. Ko'pga-bir bog'lanish qanday amalga oshiraladi?
5. Ko'pga-ko'p bog'lanishga misol keltiring?
6. Mohiyatlar klassifikatsiyasi xususida ma'lumot bering.
7. Negiz mohiyat (negiz) deganda nimani tushinasiz?
8. Birlashma mohiyat (birlashma) deganda nimani tushinasiz?
9. Tavsifli mohiyat (tavsif) deganda nimani tushinasiz?

III. MA'LUMOTLAR MODELLARI

3.1. Ma'lumotlarning iyerarxik modellari

Istalgan MB yadrosini ma'lumotlar modeli tashkil qilib, u ma'lumotlar to'plamidan, yaxlitlikni ta'minlovchi cheklanishlardan va ma'lumotlar bilan manipulyatsiya qilish operatsiyalaridan tashkil topadi.

Ma'lumotlarni o'zaro bog'liqligini tashkil qilish, saqlash, ularni to'liq namoyon eta bilish, ularga murojaat qilish uslubi va kerakli vaqtda olish mexanizmini bayon etilishi ma'lumotlar modelidir.

Ma'lumotlar modeli quyidagi tarkibiy qismdan iborat:

1. *Foydalanuvchining MBga munosabatini namoyish etishga mo'ljallangan ma'lumotlar tuzilmasi.*

2. *Ma'lumotlar tuzilishida bajarilish mumkin bo'lgan operatsiyalar.* Ular ko'rib chiqilayotgan ma'lumotlar modeli uchun ma'lumotlar tilining asosini tashkil etadi. Ma'lumotlarni aniqlash tili va ma'lumotlar bilan amallar bajarish tilining turli operatsiyalari yordamida bu tuzilma bilan ishlash imkoniga ega bo'lish zarur.

3. *Yaxlitlikni nazorat qilish uchun cheklashlar.* Ma'lumotlar modeli uning yaxlitligini saqlash va himoya qilishga imkon beruvchi vositalar bilan ta'minlangan bo'lishi lozim.

MBBTda har bir MB modeli quyidagi xususiyatlari bo'yicha tavsiflanadi:

- ma'lumotlar tuzilmalarining turi;
- ma'lumotlar ustida bajariladigan amallar;
- butunlikning cheklanganligi.

MBni yaratishga kirishishdan oldin ma'lumotlarni tasvirlash modelini tanlab olish kerak. U quyidagi talablarga javob berishi lozim:

- axborotni ko'rgazmali tasvirlash;
- axborotni kiritishda soddalik;

- axborotni izlash va tanlashda qulaylik;
- boshqa MBga kiritilgan ma'lumotdan foydalanish imkoniyatining mavjudligi;

- MBning ochiqligini ta'minlash (yangi ma'lumotlar va maydonlar qo'shish, ularni olib tashlash imkoniyatlari va h-zo).

MBda ma'lumotlar shunday tashkil etilgan bo'lishi lozimki, foydalanuvchilar va amaliy dasturlar, ularni olishi va qayta ishlashi mumkin bo'lsin. MBda ma'lumotlarni tashkil etilishiga muvofiq ko'plab mutaxassislar bugungi kunda MBni qurishga uch xil yondashish mumkin deb hisoblashadi:

- daraxtsimon (iyerarxik) modellar;
- tarmoqli (turli) modellar;
- relyatsion modellar.

Bulardan tashqari yana MBning binar munosabatlar, semantik, fayl, ob'ektga yo'naltirilgan, analitik model kabi bir qator turlari mavjud. Shuning uchun, biz MB tuzilmasi modellarini umumiy tarzda quyidagi guruhlariga ajratamiz:

- relyatsiongacha;
- relyatsion;
- relyatsiondan keyin.

Asosiy qo'llanilib kelingan modellarga qisqacha to'xtalib o'tamiz.

Ma'lumotlarning iyerarxik modellari. Ma'lumotlarning iyerarxik modeli o'zaro bog'liq ob'ektlarning majmuasidir. Ikki ob'ektning aloqasi ularning bir-biriga bo'ysunishini aks ettiradi. Daraxtsimon (iyerarxik) modelda ob'ektlar yozuvlar ko'rinishida ifodalanadi. Unda hamma yozuvlar shajarali tasnifli ko'rinishda bog'langan. Iyerarxik MBda har bir ma'lumot quyi bug'inga ega va quyi bug'in bevosita o'zi bo'ysunuvchi yuqori bug'imga ega.

MBning iyerarxik modeli asosini ag'darilgan daraxt(graf) tashkil etadi. Shuning uchun, iyerarxik modelda ma'lumotlar daraxtsimon bog'lanish



ko'rinishda saqlanadi deyiladi. Ushbu model sath, tugun, bog'lanish kabi parametrlar bilan tavsiflanadi. Uning ishlash tamoyili shundayki, quyi sathdagi bir necha tugunlar bog'lanish yordamida yuqoriroq sathdagi bitta tugun bilan bog'langan bo'ladi. Tugun - bu ierarxiyaning berilgan sathida joylashgan elementning axborot modelidir. Sodaroq qilib aytganda daraxt shoxlarga ajraladi va har bir shox o'z navbatida yana shoxga ajralishi mumkin.

Bog'lanishlar tarkibi MBda qat'iy qayd qilinadi va bog'lanishlar soni cheklangan bo'ladi. Bog'lanishlarni o'zgartirish tarkiblarini o'zgartirishga va berilganlarni qaytadan kiritishga olib keladi.

Birinchi iyerarxik MBBTlari mahsulot yetishtiruvchi kompaniyalar ishlab chiqarishini rivojlantirish sohasida qo'llanilgan edi. Masalan, agarda avtomobil kompaniyasi bir xil modeldagi 10000 ta mashina, boshqa bir modeldagi 5000 ta mashina ishlab chiqarmoqchi bo'lsa, u o'zining yetkazib beruvchilariga qancha detallarni buyurtma berishi lozim. Bu savolga javob berish uchun mashina qanday qismlardan tashkil topgan va har bir qism qanday detallardan iborat ekanligini aniqlashimiz lozim. Masalan, mashina dvigatel, korpus va yurish qismlaridan iborat; dvigatel klapan, silindr va boshqa qismlardan iborat. Tarkibiy qismlar ro'yxati bilan ishlash xuddi kompyuter uchun mo'ljallanganday tuyuladi. Tarkibiy qismlar ro'yxati o'z tabiatiga ko'ra iyerarxik tuzilmani tashkil etadi. Bunday tuzilmaga ega bo'lgan ma'lumotlarni saqlash uchun iyerarxik model ishlab chiqildi. Bunday modelda har bir MBdagi yozuv aniq detalni ifodalaydi. Yozuvlar orasidagi merosxo'rlik munosabati o'rnatilgan bo'lib, har bir qism unga tegishli detal bilan bog'langan.

MBdagi ma'lumotlarga murojaat qilish uchun dastur quyidagi ishlarni bajara olishi lozim:

- berilgan tartib bo'yicha aniq detalni topish;

- birinchi merosxurlik bo'yicha pastga o'tish;
- o'zi merosxo'r bo'lgan qismga o'tish, ya'ni yuqoriga o'tish;

- yon tomonga o'tish

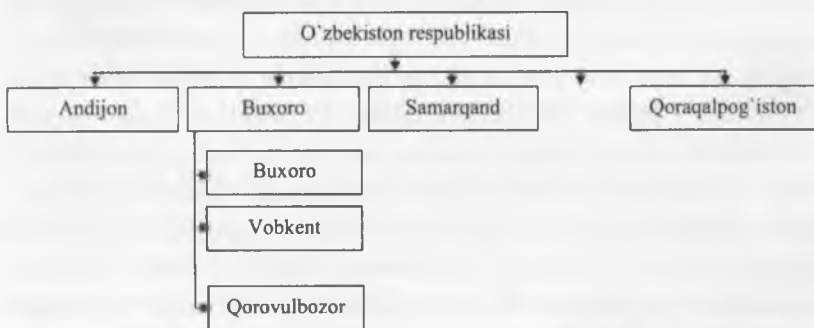
Bunday holda iyerarxik MBdagi ma'lumotlarni o'qish uchun yozuvlar bo'yicha yuqoriga, pastga, o'ngga, yon tomonga o'tish talab etiladi.

Eng ommaviy iyerarxik MBBT bo'lgan Information Management System (IMS) 1968 yilda IBM kompaniyasi tomonidan yaratilgan edi.

Iyerarxik model xususiyatlari:

- ma'lumotlar iyerarxik tarkibda tashkil qilingan;
- turli tarkiblarni tanlashda ma'lumotlarni nusxalash lozim;
- asosiy ishlash birligi-yozuv;
- ishlash ildiz yozuvdan boshlanadi.

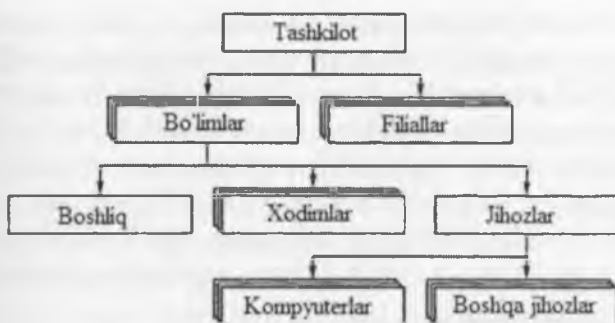
1-misol. Davlatlar, ular tarkibidagi viloyatlar hamda viloyat tumanlari va h-zolar.



3.1.1-rasm. Mamlakatning mintaqaviy strukturasi

Bu ro'yxatni tumanga bo'ysunuvchi mahalla yig'inlari, mahalla yig'inlariga qarashli aholi yashash joylari tartibida davom ettirish mumkin.

2-misol. Tashkilot, uning bo'lim va filiallari, xodimlar, jihozlar va h.z.



3.1.2-rasm. Tashkilotning shajaraviy strukturası

Daraxtsimon turdagi MB birinchi va ikkinchi avlod EHM lari yordamida tashkil qilingan. Bunday MBdan tegishli axborotni olish uchun eng avval yuqorigi xususiyatga murojaat qilinadi. Keyin yuqoridan pastga harakat qilish orqali tegishli ma'lumotlarni olish mumkin. Bu usulning kamchiligi, ma'lumotlarni saqlash uchun katta hajm talab qilinadi va tegishli ma'lumotlarni olishga ko'p vaqt sarflanadi [2].

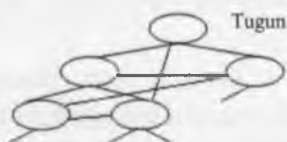
3.2. Ma'lumotlarning to'r (tarmoqlangan) modellari

Agar ma'lumotlar o'rtasidagi munosabatda yaratilgan element bittadan ortiq boshlang'ich elementga ega bo'lsa, bu munosabatni endi iyerarxik model yordamida bayon etib bo'lmaydi. Masalan, buyurtmalar haqidagi ma'lumotlarni MB da saqlashda, u uchta turli munosabatlarda ishtirok etadi. Bular buyurtma bergan mijoz bilan bog'lanish, buyurtmani qabul qilgan xodim bilan bog'lanish va buyurtma berilgan tovar bilan bog'lanishdir. Bu ma'lumotlar tuzilmasi IMS ning qat'iy ierarxiyasiga mos kelmaydi.

Buyurtmalarga ishlov berishga o'xshash ilovalarga bog'liq ravishda yangi tarmoqli ma'lumotlar modeli ishlab chiqildi. U ierarxik modelning rivojlantirilgani bo'lib, bunda bitta yozuv bir nechta munosabatda ishtirok etishi mumkin. Tarmoqli modelda bunday munosabatlar to'plamlar deb aytiladi.

IBM kompaniyasi tarmoqli MBBT ni ishlab chiqmadi va uning o'rniga IMS imkoniyatlarini kengaytirdi. 1970-yillarga kelib IDMS kompaniyasining Cullinet, Total kompaniyasining Cincom va Adabas MBBTlari ommaviylashib ketdi. 1971-yilda ma'lumotlar tizimi tili konfrentsiyasida tarmoqli MBni rasmiy standarti e'lon qilindi. U hozirda CODASYL modeli nomi bilan tanilgan. Bunday Mblari 3 chi avlod EHMLarida qo'llanilgan. Tegishli ma'lumotlar olish uchun istalgan yo'nalish bo'ylab borish mumkin bo'lgan.

Iyerarxik modelda ikki pog'onadagi elementlar bog'langan bo'lsa, unday ma'lumotlar tarmoqli modelda ifodalangan deyiladi. Tarmoqli modellarda ham ob'ektlar daraxtsimon modellardagi kabi yozuvlar ko'rinishida tasvirlanadi. Ob'ektlarning o'zaro aloqalari yozuvlar o'rtasidagi aloqalar sifatida tavsiflanadi. U ham tugun, sath, bog'lanish kabi asosiy parametrlarga ega. Lekin semantik tarmoq modelida turli sathdagi elementlar orqali "erkin", ya'ni, "har biri hamma bilan" ma'noli bog'lanish qabul qilingan.



Tarmoqsimon yoki to'rsimon ko'rinishdagi MBdagi ma'lumotlar ham iyerarxik ko'rinishdagidek tasvirlanadi, ammo undan farqi ixtiyoriy bug'in bevosita boshqa bug'inga ham bog'lanishi mumkin.

1-misol. Internet tarmog'i.

Tarmoqli tizim iyerarxik tizimga nisbatan ancha qulayliklarga ega. Chunki, bu tizimda ma'lumot (fayl) lar o'rtasida ko'p yo'nalishli aloqalarni o'rnatish mumkin. Bunday aloqalar foydalanuvchiga kerakli ma'lumot (fayl) ni shajaraning yuqori sathlaridagi fayllarga murojaat qilmasdan topish imkonini beradi. Masalan, do'kondagi mahsulotlar jadvali bilan mahsulotlar narxi jadvallari orasida alohida bog'liqlik o'rnatilishi, kerakli mahsulot narxini topish imkonini beradi.

Bir pog'onadagi ma'lumotlarning boshqa pog'onadagi ma'lumot bilan ikki yoki undan ortiq marta bog'lanadigan

turiga ma'lumotlarning tarmoqli modeli deyiladi.

Ma'lumotlarning tarmoqli modeli iyerarxik modeliga nisbatan mashina ish muhitida turli predmet sohalari uchun axborotlarni aks ettiruvchi umumiy vosita hisoblanadi. Ko'plab predmet sohalari ma'lumotlari o'rtasidagi aloqalar tarmoqli ko'rinishga ega. Bu esa ma'lumotlarning iyerarxik modeliga ega bo'lgan MBBTdan foydalanishni cheklab qo'yadi. Tarmoqli modellar, shuningdek, ma'lumotlarning iyerarxik aloqasini ham aks ettirishga imkon beradi. Tarmoqli modellar bilan ishlash texnologiyasi foydalanuvchi uchun qulaydir. Chunki, ma'lumotlarga kirish amalda hech qanday cheklashlarga ega emas va bevosita har qanday bosqichdagi ob'ektlarga kirish imkoni mavjud.

Dasturchi nuqtai - nazaridan tarmoqli MBga murojaat qilish ierarxik MBga murojaat qilishga o'xshashdir. Amaliy dastur bu modeldagi MBda quyidagi amallarni bajara olishi kerak edi:

- kalit bo'yicha aniq yozuvni topish (masalan, mijoz nomeri);
- aniq to'plam bo'yicha birinchi oqimga o'tish (mijoz tomonidan berilgan birinchi buyurtma);
- aniq to'plam bo'yicha navbatdagi oqimga o'tish (shu mijoz amalga oshirgan ikkinchi buyurtma);
- boshqa to'plam bo'yicha yuqoriga o'tish (buyurtmani qabul qilgan xodim).

Bu holda ham dasturchi MBdan ma'lumotlarni izlashi talab etiladi. Bunda nafaqat yo'nalish bo'yicha, balki talab qilingan munosabat bo'yicha ham izlash talab qilinadi.

Tarmoqli MB iyerarxik MBga nisbatan qator afzalliklarga ega bo'lib ular quyidagilardir.

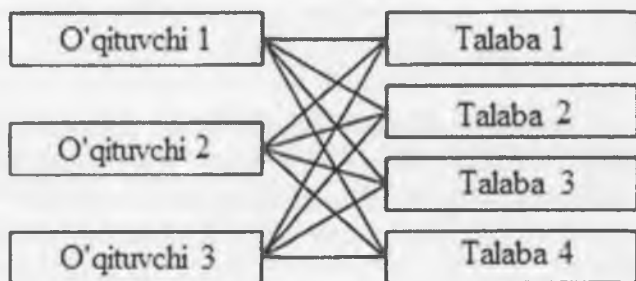
❖ Silliqlik. Ko'plik munosabatlarini tarmoqli MB iyerarxik tuzilmaga nisbatan oddiyroq tarzda saqlaydi.

❖ Standartlashtirilganligi. CODASYL standartini paydo bo'lishi bilan tarmoqli modelning ommalashuvi oshdi. Digital Equipment Corporation va Data General kabi minikompyuterlar

ishlab chiquvchilar ham tarmoqli MBBTlarini tadbiq qildilar.

❖ Tezkorlik. O'zining ancha murakkabligiga qaramasdan tarmoqli MB ierarxaik MB bilan raqobat qila oladigan darjada tezkorlikka ega. To'plamlar jismoniy MB yozuvlariga ko'rsatkich sifatida ifodalangan. Ayrim tizimlarda esa administratorlar munosabat to'plamlari asosida ma'lumotlarni klasterlash imkoniga ega.

2-misol. Ta'lim muassasasidagi o'qituvchilar va talabalar orasidagi bog'liqlik 3.2.1-rasmda keltirilgan.



3.2.1-rasm. Ma'lumotlarning to'r modeli

Albatta, tarmoqli MB ham kamchiliklarga ega edi. Iyerarxik MB singari tarmoqli MB ham juda qat'iydir. Munosabatlar to'plami va yozuvlar tuzilmasi oldindan aniqlangan bo'lishi lozim. MB tuzilmasini o'zgartirish odatda butun MB ni qayta qurishga olib keladi.

Iyerarxik ma'lumotlar bazasi singari tarmoqli MB ham dasturchi instrumenti edi. Masalan, "Asme Motors kompaniyasi qanday tovarga tez-tez buyurtma beryapti?" degan savolga javob berish uchun dasturchi MBdan ma'lumotlarni izlovchi dastur yozishi kerak edi. Bunday so'rovlarni amalga oshirish haftalar yoki oylarni talab etardi. Dasturda kerakli ma'lumot paydo bo'lganda esa u foydasiz bo'lib qolar edi [2].

3.3. Ma'lumotlarning relyatsion modellari

Iyerarxik va tarmoqli modellarning kamchiliklarini bartaraf etish ustida izlanishlar olib borildi. Natijada, 1970 yil IBM

firmasining xodimi E.F. Kodd tomonidan munosabatlarning matematik nazariyasi asosida relyatsion model g'oyasi ishlab chiqildi va u hozirgi paytda eng ko'p tarqalgan model hisoblanadi.

Relyatsion (lotin tilidagi relatio-munosabat so'zidan olingan) modelda ma'lumotlarni saqlash, uni tashkil etuvchi qismlari orasidagi munosabatlarga asoslangan. Ma'lumotlarning jadval ko'rinishida berilishi ma'lumotlarning relyatsion modeli deyiladi. Relyatsion modelli MBdagi malumotlar oddiy ikki o'lchovli jadvallarda saqlanadi va jadvallar o'zaro bir-biri bilan bog'lanadi. Ular orasidagi o'rnatilgan bog'liqlik qayta ishlash samaradorligini oshiradi hamda bir xil ma'lumotlarning takrorlanishini kamaytiradi.

Jadvallarda ustunlar maydonlar yoki atributlar, satrlar yozuvlar deb ataladi. Atributlar MB ning tuzilmasini, yozuvlar esa, unda joylashgan ma'lumotlarni tashkil etadi.

XODIM mohiyati (jadval)	Bo'lim atributi (ustun sarlovhas)	Mohiyat xemasi (sarlovha qatori)

3.3.1-rasm. "Xodim" mohiyati (jadval) va uning xususiyatlari

Jadvaldagi ma'lumotlar manzili satr va ustunlar kesishmasi bilan aniqlanadi.

Atribut – jadval ustunlari

Yozuv – atributlar yig'indisi yoki jadval satr

Jadval yacheykasi – atribut va yozuv kesishmasi

Relyatsion modelda MB bo'yicha har bir jadval takrorlanmas nomga ega bo'ladi. Jadvallarga esa birqancha talablar qo'yiladi:

❖ jadval yacheykasi qiymati tuzilma bo'yicha bo'linmasdir. Har bir yacheyka

❖ qiymatlarning faqat bir qismini olishi mumkin. Bu xossa ko'pincha ma'lumotning bo'linmaslik printsipli ham deyiladi;

❖ bir atribut (ustun) qiymatlari bir tipga ega;

❖ har bir yozuv (qator, satr) takrorlanmasdir;

❖ atributlar (ustunlar) ixtiyoriy tartibda joylashtiriladi;

❖ yozuvlar (qatorlar, satrlar) jadvalga ixtiyoriy tartibda joylashtiriladi;

❖ atributlar (ustunlar) takrorlanmas nomga ega.

Jadvalning tuzilishi ustunlarning yig'indisi bilan belgilanadi. Jadvalning har bir satrida tegishli ustunga mos keluvchi bittadan mazmun joylashgan bo'lib, unda ikkita bir xil satr bo'lishi mumkin emas. Ustunlar soni chegaralangan bo'lsada, satrlarning umumiy soni chegaralanmagan. Jadvalda ustun va satrlarning joylashish tartibi ahamiyatga ega emas.

Yozuvlarning yagonaligini ta'minlash uchun jadvalning bir yoki bir necha maydonlaridan iborat bo'lgan kalitlardan foydalaniladi. MBBT ga qo'yiladigan talablardan biri katta hajmdagi axborotlar ichidan yozuvlarni tezroq topish imkoniyatidir. Kalitli maydonlar bu ishni bajarishni tezlashtirishga imkon beradi.

Ma'lumotlar relyatsion asoslarining ayrim afzallik (ustunlik)lari:

- Oddiyligi (Soddalik). Ko'pchilik ma'lumotlar tuzilishini taqdim etishda ikki o'lchovli jadvallardan foydalanish eng oddiy usul hisoblanadi.

- Ixchamligi (Moslashuvchanlik). Proektsiyalash va bog'lash operatsiyalari munosabatlarni kesmoq va yopishtirmoqqa yo'l qo'yadiki, unda amaliy dasturchilar har xil fayllarni kerakli formada olishi mumkin.

- Muayyanligi (Aniqlilik). Matematik aniq usullar bilan MB manipulyatsiya qilinadi. Munosabatlar o'z tabiatiga ko'ra muayyan ma'noga ega bo'ladilar va matematik muayyan usullar

bilan manipulyatsiya qilish, munosabatlar algebrasi va hisoblab chiqiladigan munosabatlarga asoslangan.

- Maxfiyligi. Maxfiylik nazorati soddalashtiriladi. Har bir munosabat uchun kirish imkoniyati haqliligi beriladi. O'ziga xos maxfiylik ma'nosida ko'rsatkichlarning kirish imkoniyatini tekshirish huquqi talab qilinadi. Agar kirish imkoniyati huquqi buzilmagan bo'lsa, amaliy dasturlardan ma'lumotlarni kiritish, ishlov berish mumkin bo'ladi.

- Bog'langanligi. Relyatsion tasavvurlar turli munosabatlar atributlarini o'zaro aloqadorligi to'g'risida aniq ko'rinish beradi.

- Oddiy boshqarilishi. Ikki o'lchovli jadvallarni jismoniy joylashtirish sodd bo'ladi. Xotirani jismoniy tashkillashtirishning yangi vositalarini ixtiro qilish natijasida joylashtirishning yangi optimal imkoniyatlari paydo bo'lmoqda.

- Ma'lumotlar mustaqilligi. MBga yangi atributlar va munosabatlarni qo'shilishi. Ma'lumotlardan foydalanish usullari ham o'zgaruvchan. Yangi kortejlar qo'shilishi va eskilari chiqarib yuborilishi mumkin. Xuddi shu narsa ma'lumotlar elementiga ham tegishli. MBni me'yoriy shaklda tashkil etishda mustaqil dasturiy ta'minoti bilan ma'lumotlarni restrukturizatsiya qilish amaliy dasturlarni o'zgartirilishini talab qilmaydi. MBda o'zgarishlar amalga oshirilsa, ya'ni, mavjud atribut bir jadvaldan boshqasiga o'tkazilsa hamda amaliy dasturlar tasvir (view) jadvallar bilan bog'langan bo'lsa, bu amaliy dasturlarni o'zgartirishni talab qilmaydi.

- Ma'lumotlar bilan manipulyatsiya qilish tili. Relyatsion MB uchun yagona SQL tilidan foydalaniladi.

Relyatsion model MB tuzilmasini soddalashtirishga bo'lgan urinish natijasida vujudga keldi. Dastlabki relyatsion MBBTlarida Kodd modelining ayrim asosiy qismlari amalga oshirilmadi va bu bo'shliq keyinchalik to'ldirildi. Relyatsion konseptsianing ommalashuvi bilan o'zi relyatsion bo'lmagan ko'pgina MBlari ham relyatsion deb atala boshlandi. Shu sababli, "relyatsion" atamasining noto'g'ri qo'llanilishiga javoban, 1985-yilda Kodd maqola yozdi va unda relyatsion nomiga da'vogarlik qiluvchi ixtiyoriy MB qanoatlantirishi kerak bo'lgan 12 qoidani

keltirdi. Hozirgi vaqtgacha Koddning 12 qoidasi relyatsion MBBT ta'rifini hisoblanib kelyapti.

Umumiy, sodda qilib aytganda:

Relyatsion MB deb, foydalanuvchi ishlatadigan barcha ma'lumotlar jadval ko'rinishida tashkil etilgan va bu ma'lumotlar ustida bajariladigan amallar esa jadvallar ustida bajariladigan amallar deb tushuniladigan MBga aytiladi.

Iyerarxik tarmoqlangan model ma'lumotlarini jadvallar shaklida ham ifodalash mumkin. Natijada, bu modellar ma'lumotlari tahlil qilingan holda relyatsion model shakliga olib kelinadi [2].

Nazorat savollari

1. Ma'lumotlar modeli deganda nimani tushunasiz?
2. Ma'lumotlar modeli guruhlarini ayting?
3. Ma'lumotlarning iyerarxik modeli deganda nimani tushunasiz?
4. Ma'lumotlarning iyerarxik modeliga misol keltiring.
5. Tugun deganda nimani tushinasiz?
6. Iyerarxik model xususiyatlarini izohlang.
7. Ma'lumotlarning tarmoqli modeli deganda nimani tushunasiz?
8. Ma'lumotlarning tarmoqli modeliga misol keltiring.
9. Tarmoqli tizimning iyerarxik tizimga nisbatan qulayliklar qanday?
10. Tarmoqli model xususiyatlari deganda nimani tushinasiz?
11. Modellarda ma'lumotlar tuzilishini tushintirib bering.
12. Ob'ektlarning modellardagi aloqasini izohlang.
13. Iyerarxik va tarmoqli modellarning xususiyatlari.
14. Ma'lumotlarning relyatsion modeli deganda nimani tushunasiz?
15. Ma'lumotlarning relyatsion modeliga misol keltiring.
16. Relyatsion model xususiyatlarini izohlab bering?
17. Relyatsion modellarda ma'lumotlar tuzilishi tushintiring.
18. Ma'lumotlar relyatsion asoslarining ayrim afzalliklarini izohlang.

IV. RELYATSION MA'LUMOTLAR BAZASI. RELYATSION ALGEBRA VA RELYATSION HISOBLASH ELEMENTLARI

4.1. Relyatsion modellarda operatsiyalar bajarish

Jadval-munosabat relyatsion modellarning universal ob'ekti hisoblanadi. Bu, relyatsion modelni ta'minlovchi turli MBBT lardagi ma'lumotlarni birxillashtirish imkonini beradi. Ularni ishlash operatsiyalari munosabatlar algebrasi va relyatsion hisob-kitoblarning universal apparatidan foydalanishga asoslangan. Shuning uchun, MBni tuzishda relyatsion yondashuv munosabatlar nazariyasining terminlari qo'llaniladi.

RMBning asosiy tushunchalari jadval, munosabat, ob'ekt, mohiyat, atribut, yozuv, kalitlar va bog'lanish hamda ma'lumotlar tipi, domen va kortej hisoblanadi.

Eng oddiy ikki o'lchamli jadval munosabat sifatida belgilanadi.

Munosabat - bu jadval. RMBni "jadvalga asoslangan" MB deb atash mumkin.

MBda jadvallarni ob'ekt yoki mohiyat sifatida ham qarash mumkin.

Ob'ekt deyilganda har qanday jarayon, moddiy va nomoddiy xususiyatga ega voqealar tushuniladi.

Mohiyat - MBda saqlanishi kerak bo'lgan ma'lumot, bir-biridan farq qiluvchi ixtiyoriy ob'ektlardir. Insonlar, joy, fanlar, shaharlar, ta'm, rang va h-zolar mohiyat bo'lishi mumkin. Mohiyatning tipi va nusxasi mavjud.

Atribut - mohiyatning nomlangan tavsifidir (atribut ba'zan maydon ham deb ataladi yoki uni jadval ustuni ham deyish mumkin). Nomlash aniq bir mohiyat tipi uchun takrorlanmas, ammo turli xil mohiyat tiplari uchun bir xil bo'lishi mumkin. (masalan "rang" atributi "it", "avtomobil", "gul", "tutun", "suyuqlik" va boshqa mohiyatlar uchun aniqlanishi mumkin).

Atribut mohiyatda qanday ma'lumotlar saqlanishi kerakligini aniqlash uchun ishlatiladi. Masalan "avtomobil" mohiyati uchun "tip", "tur", "raqami", "rang" va boshqa atributlari olinishi mumkin. Bu yerda ham tip va nusxasi o'rtasida farq mavjudligi namoyon bo'ladi. 'Rang' atributi tipi juda ko'p qiymat yoki nusxaga ega bo'ladi, ya'ni qizil, ko'k, sariq, oq va hokozo. Ammo mohiyatning har bir nusxasiga atributning birgina qiymati taqdim qilinadi.

Mohiyatning tipi va atributi orasidagi mutloq farq mavjud emas. Mohiyat tipiga bog'liq holda atribut o'z o'rniga ega bo'lishi mumkin. Boshqa hollarda atribut mustaqil mohiyat sifatida qatnashishi mumkin. Masalan, avtomobil zavodi uchun "rang" ishlab chiqarilayotgan mahsulot atributi bo'lsa, rang ishlab chiqaruvchi fabrikada "rang" mohiyat tipi hisoblanadi.

Yozuv - atributlar yig'indisi yoki jadvalning satridir

Kalit - atributlarning kam sonli to'plami bo'lib, ular qiymatlari bo'yicha mohiyatning kerakli nusxasini topish mumkin. Har bir mohiyat kamida bitta kalitga ega bo'ladi. "Talaba" mohiyati uchun "Reyting daftarchasi raqami" atributi yoki "Ismi familiyasi" va "Yashash manzili" atributlar to'plami kalit hisoblanadi.

Relyatsion MBda birlamchi (dastlabki) va ikkilamchi kalitlar mavjud bo'ladi.

Birlamchi kalit - mohiyatda takrorlanmas va bo'sh bo'lmagan qiymatga yoki qiymatlarga ega bo'lgan va mohiyat nusxalarini bir qiymatli akslantiruvchi atribut yoki atributlar to'plamidir.

Ikkilamchi kalit — mazmuni turli satrlarda takrorlanishi mumkin bo'lgan kalitdir. Ular bo'yicha ikkinchi kalitning bir xil mazmunli satrlari guruhi izlab topiladi.

Bog'lanish - ikki yoki undan ortiq mohiyatlarni birlashtirish. Buning natijasida mohiyatning biror-bir qiymati orqali boshqa mohiyatni izlash amalga oshiriladi.

Jadvallarni bog'lashda atributlar bog'lanadi. Jadvaldagi boshqa jadvalga bog'lovchi atributlar ikkilamchi kalit hisoblanadi.

Ma'lumotlar tipi - Relyatsion MBda ham ma'lumotlar tipi xuddi dasturlash tilidagidek bo'lib, ular aynan bir xil tushuncha sifatida qaraladi. Odatda, zamonaviy relyatsion MBda harfiy, sonli, bitli qatorlar, maxsus sonli qiymatlar('pul' singarilar) hamda maxsus 'temporal' (sana, vaqt, vaqtinchalik interval) qiymatlar saqlanadi. Relyatsion tizimda abstrakt tipli qiymatlarni qo'llash rivojlantirilmoqda.

Domen. Domen tushunchasi ko'pchilik MBBTlarida mavjud emas. Ammo, Oracle MBBTda u qo'llaniladi. Domenni ma'lumot qabul qiladigan qiymatlar to'plami deb qarashimiz mumkin. Bir yoki bir necha atribut bir xil mazmunga ega bo'lsa, ular domen deyiladi. Masalan, "ism" domeni alfavit harflari qatoridan iborat bo'lib, u ma'lum bir belgilarni qabul qiladi. (Masalan, uning tarkibiga kiradigan yumshatish belgisi mavjud, ammo ism yumshatish belgisi bilan boshlanmaydi). Bundan tashqari bir necha ma'lumotlar bir domen tarkibida bo'lishi mumkin. Agar ikki xil ma'lumot bir domen tarkibida bo'lsa, u holda ularni solishtirish mumkin. Masalan, talabalar bileti raqami va guruh raqami qiymatlari sonli bo'lsada, ularni solishtirish o'rinsiz.

Kundalik turmushda jadvallar bilan ish ko'ramiz. Jadval sarlavhasi munosabat tasviri bo'lsa, qatorlari esa munosabat korteji nusxalari hisoblanadi. Satrlar ko'pincha kortej ham deb ataladi. Umumiy qilib aytganda tegishli atribut mazmuniga ega bo'lgan jadval ustini domen deyiladi. Turli atributlar mazmuniga ega bo'lgan satrlar esa kortej deb ataladi.

Xulosa qilib aytganda:

MB o'zaro bog'langan jadvallardan tashkil topadi.

Jadval nomga ega bo'lib, nom bir MB doirasida takrorlanmasdir.

Jadvalning har bir atributi qabul qiladigan ma'lumotlariga mos ravishdagi nomga ega bo'ladi. Har bir jadval doirasida nom takrorlanmasdir. Har bir atribut tipga ega va faqat shu tipga tegishli bo'lgan qiymatlarni qabul qiladi. Har bir atribut axborotning alohida, bo'linmas birligi sifatidagi qiymatni qabul qilishi lozim. Jadvalda atributlar soni cheklangan bo'ladi.

O'zaro mantiqan bog'langan atributlar yig'indisi yozuvni tashkil qiladi. Yozuvning tuzilishi uning tarkibiga kiruvchi, har biri oddiy ma'lumotga ega atributlar ketma-ketligi bilan belgilanadi. Yozuv nomga va tipga ega emas. Jadvalda yozuvlar soni mantiqan chegaralanmagan.

Jadval satrini ifodalovchi ma'noga ega bo'lgan atribut yoki atributlar to'plami jadval kaliti deyiladi. Jadvallarda asosan birlamchi va ikkilamchi kalitlar bo'ladi.

Jadval kaliti qiymati jadval ichida takrorlanmasa va bo'sh bo'lmasa, u birlamchi kalit deyiladi. Har bir jadval o'z birlamchi kalitiga ega bo'lishi zarur.

Birlamchi kalit bevosita jadval atributlaridan olinsa tabiiy, agar qo'shimcha atribut sifatida kiritilsa sun'iy deyiladi.

Agar birlamchi kalit bir atributdan iborat bo'lsa oddiy, bir necha atributlar birlashmasidan iborat bo'lsa murakkab deyiladi.

Jadval kaliti qiymati jadval ichida takrorlansa va boshqa bir jadval bilan bog'lanishni tashkil qilsa, u ikkilamchi kalit deyiladi. Agar birlamchi kalitning qiymatiga ko'ra faqat bitta yozuv nusxasi topilsa, ikkilamchi kalit bo'yicha bir necha nusxa topilishi mumkin. Bir jadvalda bir nechta ikkilamchi kalit bo'lishi yoki u ikkilamchi kalitga ega bo'lmasligi mumkin. Ikkilamchi kalit bo'sh qiymat qabul qilishi mumkin.

1-misol. Relyatsion modelning har bir tushunchasini "Xodimlar" – mohiyati (ob'ekti) misolida ko'rib chiqamiz:

Xodimlar		Atribut	munosabat sxemasi		
Tabel nomer	Familiya	Ism	Otasining ismi	Bo'lim	lavozimi
100	Abdullaev	Abdulla	Abdullaevich	2 00	Boshliq
101	Ismoilov	G'ani	Komilovich	1 00	Injener
102	Karimov	Bo'labo	To'raevich	2 00	injener

Atribut, maydon
 domen, ustun
 qiymati

Kortej, yozuv, satr
 satrli sonli ma'lumotlar tipi

4.1.1-rasm. "Xodimlar" relyatsion modeli

Har bir ustun o'z nomiga ega bo'ladi va ustun nomi odatda jadvalning ustki qismida ko'rsatiladi. Uning nomi joriy jadvalda boshqa ustun uchun takrorlanmasligi kerak, biroq bu nom boshqa jadvalning ustun nomi sifatida ishlatilishi mumkin. Ya'ni, turli jadvallar bir xil nomli ustunga ega bo'lishi mumkin.

Ixtiyoriy jadval kamida bitta ustunga ega bo'ladi, ustunlar jadvalda ularni yaratish tartibida joylashadi. Satrlar ustunlardan (atributlar) farqli ravishda nomga va joylashish tartibiga ega emas. Ularning miqdori chegaralanmagan.

Ixtiyoriy jadval birlamchi kalit deb ataladigan ustunga yoki ustunlarga ega bo'ladi va undagi qiymatlar har bir satrni takrorlanmas qiymat bilan identifikatsiyalaydi.

Rasmda keltirilgan misolda birlamchi kalit bu "tabel nomeri" ustunidir

Relyatsion MBda ishtirok etadigan jadvallar bir biri bilan bog'langan bo'ladi. MBda jadvallarni bir-biri bilan bog'lash uchun bir jadvalning ikkilamchi kaliti boshqa jadvalning birlamchi kaliti bilan bog'lanadi.

Jadvalning bir-biri bilan bog'lanish tuzilmasi bog'lanish sxemasi deyiladi.

MB doimo o'zgarib turadi: unga yangi yozuvlar, borlariga esa yangi elementlar qo'shiladi.

MBni kengaytirish yangi ma'lumotlar uchun qo'shimga jadval tuzib, uni MBga qo'shish orqali amalga oshiriladi.

Ma'lumotlarga kirish istalgan kerakli munosabatlar orasidagi bog'lanishlar orqali bo'ladi. Axborotni izlashda barcha atributlardan foydalanish mumkin.

Relyatsion modelda barcha ma'lumotlar foydalanuvchiga jadval shaklida havola qilinadi va MB ustidagi barcha amallar jadvallar ustidagi amallarga olib kelinadi. Har bir jadval real dunyodagi ob'ektlar (mohiyat) tipini aks ettiradi, har bir satri esa ob'ektning aniq nusxasini bildiradi.

Relyatsion modeldagi asosiy tushuncha 'munosabat' (relation) hisoblanadi va uning asosiy xossasi quyidagicha:

- o qiymatlarning bazali qismi munosabat (relations) sifatida ifodalanadi.

- o jadvalga ta'luqli jarayonlar faqat munosabatlar ifodasi (relations closure) doirasida ifodalanadi [3].

4.2. Munosabatlar va bog'lanishlar

Munosabat (jadval)- doimiy sondagi o'zaro mantiqiy bog'langan atributlar (ustunlar) va o'zgaruvcha sondagi kortejlar(satrlar) to'plamidir.

Munosabat matematik kontsepsiya bo'lib, ko'plikning ikki elementi orasidagi mosligidir. Shunday qilib, relyatsion modelning ildizi matematika sohasiga tegishlidir. Ammo bizning nuqtai nazarimizcha, munosabat bu ba'zi bir maxsus xossalarga ega bo'lgan jadvaldir. Relyatsion model-qiymatlarni tashkil qilishni jadval ko'rinishida ifodalashdan iborat.

Relyatsion modeldagi relyatsion bog'langan ikki jadvallar orasidagi mantiqiy aloqalar jadval munosabatlarining bir xil

o'ziga xos xususiyatlarining mazmun jihatidan tengligiga ko'ra o'rnatiladi.

Jadvallarni bog'lashda bir jadvaldagi yozuvning bir yoki bir necha atributi ikkinchi jadvaldagi yozuvning bir yoki bir necha atributi bilan bog'lanadi. Bog'lanishda asosan kalit deb ataluvchi atributlar ishtirok etadi. Kalitlar birlamchi yoki ikkilamchi bo'lishi mumkin.

Jadvallar orasidagi munosabatlarning to'rtta turi mavjud, ular: "birga-bir", "birga-ko'p", "ko'pga-bir" va "ko'pga-ko'p".

Bu munosabatlar haqida 4-mavzuda batafsil ma'lumot berib o'tilgan edi. SHunday bo'lsada, ularning har biriga qisqaroq to'xtab o'tamiz.

Birga-bir munosabatning ma'nosi shuki, bitta jadvalning har bir yozuvi boshqa jadvalning faqat bitta yozuviga to'g'ri keladi (mos keladi). Misol uchun, agar ikkita jadvalni ko'radigan bo'lsak, ulardan birida korxona ishchilari to'g'risida ma'lumotlar, ikkinchisida esa-pasport hujjati ma'lumotlari bo'lsa, u holda bu jadvallar orasida birga-bir munosabat mavjud. Chunki, bir jadvaldagi bir yozuv uchun ikkinchi jadvalda faqat bitta yozuv bo'lishi mumkin.

Birga-ko'p munosabatning ma'nosi shuki, bitta jadvalning har bir yozuvi boshqa jadvalning bir necha yozuviga to'g'ri keladi (mos keladi). U MBda ko'p uchraydigan munosabatlar turidir. Berilgan munosabatlar turini ko'rsatish uchun, korxona mijozlari va ular bergan buyurtmalar to'g'risida ma'lumotlar bo'lgan jadvallarga murojaat qilishimiz mumkin. Boshqa misol sifatida korxona va unda ishlaydigan xodimlar orasidagi munosabatlar ko'rilishi mumkin. Xuddi shunday munosabatlar kompyuter va unga kiruvchi komponentlar orasida mavjud va h-zo.

Ko'pga-bir munosabat avval ko'rilgan tur, birga-ko'p munosabat bilan o'xshash. Ob'ektlar orasidagi munosabatlar turi sizning nuqtai nazaringizga bog'liq. Misol, agar siz buyurtmalar

va mijozlar orasidagi munosabatlarni ko'rsangiz, u holda ko'p-bir munosabatni olasiz.

Ko'pga-ko'p munosabat ancha murakkab hisoblanadi. Ikkita jadvallar orasida ko'pga-ko'pga munosabat hosil bo'ladi, qachonki:

- birinchi jadvaldagi bitta yozuv ikkinchi jadvaldagi bittadan ortiqroq yozuv bilan bog'lanishi mumkin bo'lsa;

- ikkinchi jadvaldagi bitta yozuv birinchi jadvaldagi bittadan ortiqroq yozuv bilan bog'lanishi mumkin bo'lsa.

Relyatsion modelda ko'pga-ko'p munosabat ko'pga-bir yoki birga-ko'p munosabat orqali bog'lanadi. Buning uchun qo'shimcha jadval tuziladi va bu jadval ikkita jadvalni bog'lashga xizmat qiladi.

2-misol. Jadvallar orasidagi bog'lanishlarni "Xodim" MBning ba'zi bir jadvallari misolida ko'rib chiqamiz:

Jadvallardagi tushunchalar va bog'lanishlar quyidagicha:

Jadval nomi: "Lavozim", "Xodim", "Bo'lim", "Pasport".

Atributlar:

- "Lavozim" jadvalida "id", "Lavozim".
- "Xodim" jadvalida "id", "Familiya", "Lavozim", "Bo'lim", "Staj" va h-zo.
- "Bo'lim" jadvalida "id", "Bo'lim".
- "Pasport" jadvalida "id", "Seriya", "nomer", "Jinsi", "Tug_sana", "Ber_sana", "Amal_sana" va h-zo.

Birlamchi kalitlar:

- "Lavozim" jadvalida "id".
- "Xodim" jadvalida "id".
- "Bo'lim" jadvalida "id".
- "Pasport" jadvalida "id"

Ikkilamchi kalitlar:

- "Lavozim" jadvalida yo`q.
- "Xodim" jadvalida "Lavozim", "Bo'lim".
- "Bo'lim" jadvalida yo`q.
- "Pasport" jadvalida yo`q.

Bog'lanishlar:

- "Lavozim" jadvalining birlamchi kaliti "id" va "Xodim" jadvalining ikkilamchi kaliti "Lavozim" birga-ko'p bog'langan.

- "Bo'lim" jadvalining birlamchi kaliti "id" va "Xodim" jadvalining ikkilamchi kaliti "Bo'lim" birga-ko'p bog'langan.

- "Xodim" jadvalining birlamchi kaliti "id" va "Pasport" jadvalining ikkilamchi kaliti "id" birga-bir bog'langan [3].

4.3. Relyatsion algebra va relyatsion hisoblash elementlari

Relyatsion MBda har bir jadval munosabat hisoblanadi va ular ustida operatsiyalar esa relyatsion algebraga asoslanadi. Faqatgina jadvallarni yaratish, unga qiymat kiritish, atributlarni tavsiflash va ularni qayta nomlash operatsiyalari relyatsion algebraga tegishli emas.

Relyatsion algebra nazariyasida munosabat to'plam, jadval qatorlari kortej, ustunlari esa atribut sifatida qaraladi. Relyatsion algebra Kodd tomonidan aniqlangan ikkita guruhga bo'lingan 8 ta operatoridan iborat.

Birinchi guruhga to'plamlar ustida bajariluvchi an'anaviy amallar kiradi: birlashtirish (U), kesishma ($\cap$), ayirish ($-$) va dekart ko'paytirish (*). Bunda barcha amallarda operandlar ixtiyoriy to'plam deb emas, balki munosabatlar deb qaraladi.

Ikkinchi guruh maxsus relyatsion amallarni tashkil etadi: tanlash, proektsiya, birlashtirish va bo'lish.

Ushbu amallarning munosabatlarda qo'llash natijalarini batafsil ko'rib chiqamiz.

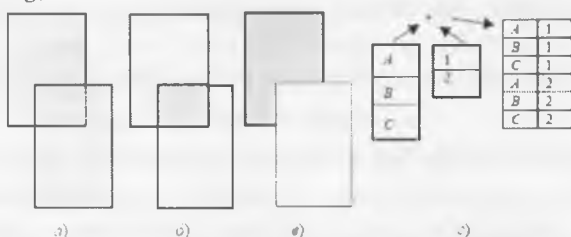
Birlashtirish (U). Berilgan ikkita munosabatdan biriga va ikkalasiga ham tegishli bo'lgan kortejlardan iborat yangi munosabatni qaytaradi. (4.3.1-rasm, a).

Kesishma ($\cap$). Bir vaqtning o'zida ikkala munosabatga ham tegishli bo'lgan barcha kortejlardan iborat yangi

munosabatni qaytaradi. (4.3.1-rasm, b).

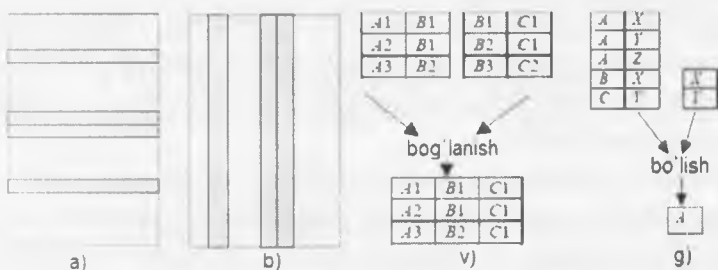
Ayirish (-). Berilgan ikkala munosabatdan faqat birinchisiga tegishli bo'lgan kortejlardan iborat yangi munosabatni qaytaradi. (4.3.1-rasm, v).

Dekart ko'paytirish (*) - berilgan ikkala jadvaldagi kortejlarning kombinatsiyalaridan iborat yangi jadval hosil qiladi (4.3.1-rasm, g).



4.3.1-rasm. To'plamlar ustida amallar

a-birlashtirish, b - kesishma, v- ayirish, g - dekart ko'paytirish



4.3.2-rasm. Maxsus relyatsion bog'lanish.

a-tanlash, b - proektsiya, v- ulash, g - bo'lish

Tanlash - berilgan jadvaldagi ma'lum shartni qanoatlantiruvchi barcha kortejlardan iborat yangi jadval hosil qiladi. Bu algebralik *cheklanish* hisoblanadi. (4.3.2- rasm, a).

Proektsiya - berilgan jadvaldagi ba'zi kortejlarni istisno (chiqarib tashlash) qilib qolgan kortejlardan (podkortej) yangi munosabat hosil qiladi (4.3.2- rasm, b).

Ulash - berilgan ikkala jadvalda umumiy qiymatga ega bo'lgan kortejlarning ulanishidan iborat yangi jadval hosil qiladi.

Natijaviy jadvalda umumiy qiymat faqat bir marta qatnashadi. Bunday ulash tabiiy ulash deb ataladi (4.3.2- rasm, v).

Bo'lish – berilgan binar va unar ikkita jadval uchun unar jadvalning barcha qiymatlari bilan moslashgan binar jadvaldagi bitta atributning qiymatlaridan iborat jadval (4.3.2- rasm, g).

Jadval ustidagi har amal natijasi jadvaldan iborat bo'ladi. Ushbu relyatsion xususiyat *yopiqlik* xossasi deb ataladi.

Biror amal natijasi boshqa amal uchun boshlang'ich ma'lumot sifatida qo'llanilishi mumkin. Masalan, birlashtirish proyeksiyasini olish, yoki ikkita tanlanma ulanmasini olish mumkin. Bunday ifodalar murakkab hisoblanadi.

Har bir munosabat sarlavhaga, tanaga, kalitlar zahirasiga ega bo'ladi. Relyatsion amallarni bajarishda atribut nomlarini yaratilishi va potentsial kalitlar o'zgarishiga e'tibor qilish kerak.

Munosabatlar ustidagi to'plamlar nazariyasining an'anaviy operatsiyalarini qo'shimcha ravishda jadval nuqtai nazaridan boshqacha tavsiflab o'tamiz:

1. Munosabat cheklovi (Tanlash) - tanlangan qator munosabatlari cheklov shartlarini qanoatlantiradigan – operandli (boshlang'ich munosabatli) yangi munosabat yaratishdir.

2. Munosabat tasviri (Proektsiya) - aniq ustun munosabatlari tanlangan – operandli yangi munosabat yaratishdir.

3. Munosabatlarni birlashtirish – barcha kortejlari munosabatlari saqlangan – operandli yangi munosabat yaratishdir. Bu holatda operandlar bir xil atributlarga ega bo'lishi lozim.

1-misol. Tsexlardan avvalgi oylarda olingan yangi mahsulotlar haqida kelgan hisobotlarni birlashtirish lozim. U "Mahsulot chiqarish" umumiy jadvaliga o'xshash quyidagi ma'lumotlardan tashkil topgan: "Tsex nomeri", "Mahsulot kodi", "Chiqarilgan sana", "Soni". Hisobotlarni birlashtirish uchun:

Mahsulot chiqarish (Tsex nomeri, Mahsulot kodi, Chiqarilgan sana, Soni) kortejiga

Yangi mahsulot (Tsex nomeri, Mahsulot kodi, Chiqarilgan sana, Soni)

kortejlari qo'shiladi.

4. Munosabatlar kesishuvi – solishtiriladigan operandlar uchun umumiy qatorga ega bo'lgan yangi munosabat yaratishdir. Bu holatda operandlar bir xil atributlarga ega bo'lishi lozim.

2-misol. Munosabatlar kesishuvini cheklov va tasvir (proektsiya) operatsiyalarinig bajarilishi orqali ko'ramiz.

Bizda atributlari o'zaro mos tushuvchi munosabatlar mavjud bo'lsin, ya'ni:

Imtihon vedomosti(Guruh, Zachetka (Reyting daftarcha), Talaba f.i.o., Fan, Sana, Baho).

4.3.1-jadval

Informatika fanidan imtihon vedomosti

Guruh	Sinov daf.	Talaba f.i.o.	Fan	Sana	Baho
1	1	Karimov I.B.	Informatika	15.02.2013	A'lo
1	2	Olimov S.T.	Informatika	15.02.2013	Qoniqarli
1	3	Sobirov U.S.	Informatika	15.02.2013	Qoniqarli
1	4	Toshev G.M.	Informatika	15.02.2013	Yaxshi
1	5	Alieva G.G.	Informatika	15.02.2013	A'lo

4.3.2-jadval

Matematika fanidan imtihon vedomosti

Guruh	Sinov daf	Talaba f.i.o.	Fan	Sana	Baho
1	1	Karimov I.B.	Matematika	17.02.2013	Yaxshi
1	2	Olimov S.T.	Matematika	17.02.2013	Qoniqarli
1	3	Sobirov U.S.	Matematika	17.02.2013	Qoniqarli
1	4	Teshaeva G.M.	Matematika	17.02.2013	Yaxshi
1	5	Alieva G.G.	Matematika	17.02.2013	A'lo

Fanlardan "a'lo" baho olgan talabalar haqidagi ma'lumotni yuqoridagi ustunlarga ega bo'lgan jadval shaklida olishimiz kerak. ("A'lo" baho cheklovi orqali)

4.3.3-jadval

Informatika fanidan imtihon vedomosti

Guruh	Zachetka	Talaba f.i.o.	Fan	Sana	Baho
1	1	Karimov I.B.	Informatika	15.02.2013	A'lo
1	5	Alieva G.G.	Informatika	15.02.2013	A'lo

4.3.4-jadval

Matematika fanidan imtihon vedomosti

Guruh	Zachetka	Talaba f.i.o.	Fan	Sana	Baho
1	5	Alieva G.G.	Matematika	17.02.2013	A'lo

Jadvallarni solishtirish orqali fanlardan "a'lo" baho olgan talabalar haqidagi ma'lumotni olishimiz mumkin. (Tasvir orqali va so'ngra kesishuv)

4.3.5-jadval

A'lochilar ro'yxati

Zachetka	Talaba f.i.o.
5	Alieva G.G.

5. Munosabatlar farqi – 1-operanda mavjud, ammo 2-operandda mavjud bo'lmagan qatorga ega bo'lgan yangi munosabat yaratishdir. Bu holatda operandlar bir xil atributlarga ega bo'lishi lozim.

3-misol. Tsexlardan avvalgi oylarda olingan yangi mahsulotlar haqida kelgan hisobotlar (1-misol) asosida oxirgi kvartalda chiqarilgan yangi mahsulot haqida ma'lumot tayyorlash kerak.

Buning uchun "Mahsulot chiqarish" munosabati "Chiqarilgan sana" ma'lumotiga "sana o'tgan kvartal oxirgi sanasidan

kichik" shartini qo'yamiz.

Ushbu cheklov natijasini boshlang'ich jadvalga qo'yamiz.

Keyin ushbu jadval "Chiqarilgan sana" ma'lumotiga "sana o'tgan kvartal birinchi sanasidan kichik" shartini qo'yamiz.

Ushbu cheklov natijasini oxirgi jadvalga qo'yamiz.

Boshlang'ich va oxirgi jadvallar farqi, ko'zlangan natijani beradi.

6. Munosabatlarni ko'payishi – 1 chi va 2 chi operandlarning barcha atributlari bo'lgan hamda qatorlar juft-juft bog'langan yangi munosabat yaratishdir. Yangi munosabat kortejlari quvvati soni 1 chi va 2 chi operandlarning kortejlari quvvatiga teng bo'ladi. Bu holatda atributlar to'plami kesishmasligi lozim.

7. Munosabatlarni bo'lish – 2-operandda mavjud bo'lmagan, ammo 1-operandda mavjud bo'lgan atributlar hamda 2-operandga mos tushuvchi 1-operand kortejlari bo'lgan yangi munosabat yaratishdir. Bu operatsiyani bajarish uchun 2-operand 1-operand bilan mos keluvchi atributlarga ega bo'lishi lozim.

8. Munosabatlarni qo'shilishi – operandlar (boshlang'ich munosabatlar) kortejlari birlashishi orqali olingan kortejlar bo'yicha yangi munosabat yaratishdir.

Tabiiy va shart bo'yicha munosabatlarni qo'shilishi mavjud.

Shart bo'yicha munosabatlarni qo'shishda birlashishi orqali olingan yangi qator shartni qanoatlantirishi tekshiriladi.

Agar shart qanoatlantirilsa, olingan qator natija munosabatga qo'shiladi.

Tabiiy munosabatlarni qo'shishda birlashishi orqali olingan yangi qator uchun shart mavjud emas. Bu holatda operandlar bir xil atributlarga ega bulishi lozim.

Mahsulot chiqarish (Tseh nomeri, Mahsulot kodi, Chiqarilgan sana, Soni) kortejiga

Yangi mahsulot (Tseh nomeri, Mahsulot kodi, Chiqarilgan sana, Soni) kortejlari qo'shiladi [8].

Nazorat savollari

1. Ma'lumotlarning relyatsion modeli deganda nimani tushunasiz?

2. Ob'ekt iborasiga ta'rif bering.

3. Domen deganda nimani tushinasiz?

4. Kortej tushunchasini izohlab bering.

5. Atribut xususiyatlarini ochib bering.

6. Yozuv deganda nimani tushinasiz.

7. Kalit maydonlarining qanday xususiyatlari mavjud?

8. Birlamchi kalit deganda nimani tushuinasiz?

9. Ikkilamchi kalit birlamchi kalitdan nima bilan farq qiladi?

10. Munosabat tushinchasi qanday ma'noga ega?

11. Mohiyat iborasi nimani anglatadi?

12. Ma'lumotlarning qanday tiplarini bilasiz?

13. Relyatsion modelning ikki asosiy xossasini tushintiring.

14. Relyatsion MB jadvallariga qanday talablar qo'yiladi?

15. Ma'lumotlar relyatsion asoslarning afzalliklarini sanab o'ting?

16. Relyatsion MBning asosiy parametrlari ajratib ko'rsating?

V. MA'LUMOTLAR BAZASINI ME'YORLASHTIRISH. MA'LUMOTLARNING ME'YORIY SHAKLLARI NAZARIYASI

5.1. Ma'lumotlar bazani me'yorlashtirish haqida umumiy ma'lumot

MBni yaratish predmet sohasini tavsiflash asosida amalga oshiriladi. Bunday tavsif predmet sohasini ifodalovchi hujjatlar majmuasiini, aniq ob'ektlar, jarayonlar va hodisalar, MBda saqlanishi kerak bo'lgan axborotlar butun turkumini qamrab olishi kerak. MBni yaratishda uning tuzilishi aniqlanishi lozim. MB tuzilishini aniqlashdan avval axborot ehtiyojlari - MBga joylashtirilishi lozim bo'lgan axborot majmuasi aniqlanadi. Shundan so'ng, MBni loyihalashga kirishish lozim. MB ni loyihalashning asosiy maqsadi uning mantiqiy tuzilishini belgilashdan iboratdir. Loyihalash natijasida relyatsion bazaning tuzilishi, ya'ni relyatsion jadvallari, ularning tarkibi va mantiqiy aloqadorliklari aniqlanadi. Chunki, baza o'zaro mantiqiy aloqadorlikdagi sanoqsiz jadvallardan tuzilgan bo'ladi. Jadvalning tuzilishi esa ustunlar tarkibi, ularning ketma-ketligi, har bir ustun ma'lumotlarining turi va o'lchami, shuningdek, jadval kaliti bilan aniqlanadi.

MBda saqlanayotgan ma'lumotlar bilan ishlovchi samarali ilovani hosil qilish ham loyihalash orqali tashkil qilingan ma'lumotlar tuzilmasiga bog'liq. Yaxshi tashkil qilingan ma'lumotlar tuzilmasi:

- oddiy va foydalanuvchiga tushunarli bo'lgan ilova bilan ma'lumot kiritishni bajarish (amalg oshirish);
- kerakli bo'lgan ma'lumotni MBdan tez topish;
- MBni ortiqcha o'sib ketishga olib kelmaydigan ko'rinishda saqlash;

- dastur ta'minotini ishlab chiqish va kuzatishni soddalashtirish imkonini beradi.

MBni loyihalashda eng qulay MB tuzilmasi to'g'risidagi masalani hal qilishimiz zarur. Bundan qo'yilgan asosiy maqsadlar:

- jadvaldagi ma'lumotlarga tez etishishni ta'minlash;
- kiritishdagi xatolar sababi va kompyuter diski maydonidan noratsional foydalanish sababi bo'lishi mumkin bo'lgan, keraksiz ma'lumotlarni qaytarilishiga yo'l qo'ymaslik;
- ma'lumotlar butligini shunday ta'minlash kerakki, bitta ob'ekt o'zgartirilganda avtomatik ravishda ular bilan bog'liq bo'lgan obyektlarni o'zgarishi ro'y bersin.

Relyatsion MBda ma'lumotlarni relyatsion talabi darajasida tashkil qilishda me'yorlashtirish qo'llaniladi. Me'yorlashtirish kelgusida relyatsion modelni takomillashtirish jarayonini namoyon qiladi. U tahliliy modelni tuzgandan so'ng bajariladi va uning tashkiliy tuzilish darajasini oshirish uchun mo'ljallangan.

Me'yorlashtirish nima ekanligini o'rganishdan oldin me'yoriy bo'lmagan shakl tushunchasini o'rganib chiqamiz.

Me'yoriy bo'lmagan shakl (MBSH) – bu bir yoki bir nechta takrolanuvchi ma'lumotlar guruhiga ega bo'lgan jadval.

Takrolanuvchi guruh tushunchasi. Takrorlanuvchi guruh ER – diagrammalardagi munosabatning ko'p qiymatli atributiga mos keladi. Takrorlanuvchi guruh (repeating group) - bu har bir maydonida bir nechta qiymat joylashgan atributdir.

1-misol. Masalan, xodimlar munosabatida (jadvalida) farzandlar nomini va tug'ilgan kunlarini saqlash zarur bo'lsin. Har bir xodimda bir nechta farzand bo'lishi mumkin. Shuning uchun, farzandlar tug'ilgan kuni va ismlari takrorlanuvchi guruhni tashkil etadi.

Ushbu jadvalda ko'rinib turganidek, har bir satrning ikkita ustunida (F_ism, F_tug) bittadan ko'p qiymat joylashgan (5.1.1-jadval).

Bu jadvaldan "qaysi bolalarning tug'ilgan kunlari ustma-ust tushadi" degan ma'lumotni aniq olishning iloji yo'q. Faraz qilaylik, farzand nomini kelish tartibi bilan tug'ilgan sana kelish tartibini solishtiramiz desak ham pozitsiyalar mos kelmasligi mumkin. Bundan tashqari, "qaysi xodimlarning farzandlari 16 yoshgacha", "qaysi xodimlarning farzandlari ismlari bir xil", "xodimning nechta farzandi bor" singari ma'lumotlarni ham aniq olishning iloji yo'q.

5.1.1-jadval

"Xodimlar" jadvali ma'lumotlari

ID	Fam	F_ism	F_tug
101	Karimov Olim	Ahad, Gulchehra	11.1.92, 15.5.94
102	Abdullaev Sobir	Nilufar, Nigina, Zarina	12.2.92, 5.10.95, 17.06.01
103	Aslonova Salima	Ma'suma	10.5.04
104	Obidov Ikrom	Olim	7.4.06
105	Xo'jaeva Karima	Madina, Ali	10.21.05, 18.1.08
106	Bo'ronova Guli		

Mazkur jadvaldan kerakli ma'lumotni izlab topish juda ham qiyin. Vergullarni ishlatilishi, ma'lumotlarni kelish tartibi bo'yicha solishtirish orqali kerakli ma'lumotni olish mumkin, ammo bu ma'lumotlarga ishlov berish operatsiyalarini ko'payib ketishiga olib keladi.

Takrorlanuvchi guruhlarini bartaraf etish. Takrorlanuvchi guruhlarini bartaraf etishning ikkita usuli mavjud: noto'g'ri va to'g'ri.

Noto'g'ri usul. Birinchi usulda takrorlanuvchi guruhlar har bir takrorlanuvchi qiymat uchun jadvalga alohida ustun qo'shish orqali bartaraf etiladi.

2-misol. Yuqorida keltirilgan jadvalda F_ism va F_tug ustunlarini uchta ustunga ajratamiz (5.1.2-jadval).

Bu jadval izlash talablarini qanoatlantiradi. Ammo, unda bir qator muammolar vujudga kelgan.

- Jadval har bir xodim uchun faqat uchtagacha farzandi bilan chegaralangan. Unda xodimning 4-farzandi haqida ma'lumotni saqlashning imkoni yo'q.

- Xodimlarning farzandi yo'q yoki uchtdan kam bo'lgan xodimlarda jadvalning ko'pgina qismi bekor qoladi.

- Aniq bir farzandni izlash amali murakkablashadi.

5.1.2-jadval

"Xodimlar" jadvalining noto'g'ri usuldagi tuzilishi

ID	Fam	F_ism1	F_ism2	F_ism3	F_tug1	F_tug2	F_tug3
101	Karimov Olim	Ahad	Gulnora		11.1.92	15.5.94	
102	Abdullaev Sobir	Nilufar	Nigina	Zarina	12.2.92	5.10.95	17.06.01
103	Aslonova Salima	Ma'suma			10.5.04		
104	Obidov Ikrom	Olim			7.4.06		
105	Xo'jaeva Karima	Madina	Ali		10.21.05	18.1.08	
106	Bo'ronova Guli						

To'g'ri usul. Bu usulda boshlang'ich jadvaldagi biror atribut yoki atributlar kalit sifatida belgilanadi. Keyin takrorlanuvchi guruhlar joriy jadvaldan olinib alohida jadvalga o'tkaziladi. Bunda takrorlanuvchi guruhdagi har bir qiymat kalit nusxasi bilan birga yangi jadvalga o'tkaziladi. Yangi jadvalga o'tkazish har bir takrorlanuvchi guruh uchun bajariladi. Agar yaratilgan jadvalda ham takrorlanuvchi guruhlar mavjud bo'lsa, ular uchun ham to'g'ri yoki noto'g'ri usul qo'llaniladi.

3-misol. Keltirilgan misolda farzand ismlari va tug'ilgan kunlari haqidagi ma'lumotlar yangi jadvalga o'tkaziladi.

Xodimlar 5.1.3-jadval

ID	Fam
101	Karimov Olim
102	Abdullaev Sobir
103	Aslonova Salima
104	Obidov Ikrom
105	Xo`jaeva Karima
106	Bo`ronova Guli

Farzand 5.1.4-jadval

ID	F ism	F tug
101	Ahad	11.1.92
101	Gulnora	15.5.94
102	Nilufar	12.2.92
102	Nigina	5.10.95
102	Zarina	17.06.01
103	Ma`suma	10.5.04
104	Olim	7.4.06
105	Madina	10.21.05
105	Ali	18.1.08

*Me`yorlashtirish* - MBdagi ma`lumotlar ortiqchaligini yo`qotish, takrorlanmaslik va o`zaro bog`lanishlarni tashkil qilish jarayonidir.

Me`yorlashtirishning asosiy maqsadi – shunday MBni yaratishga qaratilganki, unda ma`lumotlar bir marta qayd qilinadi, ya`ni ma`lumotlar ortiqcha yoki takrorlanishi mumkin emas. Bu amal saqlanadigan ma`lumotlarni qarama – qarshiligini bartaraf etish uchun zarur.

Me`yorlashtirishni talab darajasida tashkil qilish uchun dekompozitsiya (bo`laklarga ajratish) qo`llaniladi.

Dekompozitsiya - MB jadvallarini o`zaro bog`langan sodda va ixcham bir nechta jadvallarga ajratishdir.

Dekompozitsiyalashda MB jadvallari soni oshib ketadi, ammo MB ixcham tuzilmaga ega bo`ladi. Ular me`yorlashtirilgan bo`lishi lozim.

5.2. Atributlarning funktsional bog`liqligi

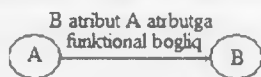
Ma`lumotlarni me`yorlashtirish shakllari nazariyasi bo`yicha relyatsion MB sxemasi tuzilmali va semantik axborotlarga ega bo`ladi. Tuzilmali axborotlar munosabatlarning paydo bo`lishiga bog`liqdir. Semantik axborotlar esa sxemada ifodalangan munosabatlar atributlari orasidagi ko`plab funktsional aloqalarda ifodalanadi. Biroq, ba`zi funktsional aloqalar qo`shimcha ta`sirlar

yoki MBni modifikatsiyalashda kelib chiqadigan noto'g'ri shakl (anomaliya)lar tufayli qo'llanilmasligi mumkin. Shu munosabat bilan taklif etilayotgan sxemaning to'g'ri yoki noto'g'riligi haqidagi savol to'g'iladi. Noto'g'ri funktsional aloqalari bo'lmagan sxema to'g'ri hisoblanadi. Aks holda, dekompozitsiya (bo'laklarga ajratish) deb nomlangan tadbirga murojaat qilishga to'g'ri keladi. Unda ko'plab munosabatlar biri ikkinchisiga almashtiriladi (ularning soni ortib ketadi). Bu tadbirning maqsadi — noto'g'ri shakliy bog'liqlik (anomaliyalar)ni bartaraf etish bo'lib, u me'yorlashtirish jarayonining mohiyatini tashkil etadi. Boshqacha aytganda, me'yorlashtirish — berilgan sxema (yoki munosabatlar yig'indisi)ni munosabatlari ko'proq oddiy va doimiy tuzilmaga ega bo'lgan boshqa sxema bilan qadamba-qadam almashtirish jarayonidir.

Dekompozitsiyada yo'qotishlarsiz bog'liqliklarni saqlab qolish, ushbu sxemadagi barcha ehtimol tutilgan funktsional bog'liqliklarni bilishni talab qiladi.

Me'yorlashtirish uchun bir necha bog'liqlik tushunchalari ishlatiladi, ya'ni:

- *Funktsional bog'liqlik:* Agar ixtiyoriy vaqtda A atributning har bir qiymatiga B atributning mavjud qiymatlaridan yagona bir qiymat mos kelsa, u holda A va B atributlar orasidagi bog'liqlik funktsional bog'liqlik deyiladi.



- Masalan, bir daryo faqat bir dengizga qo'yiladi va bir dengizga bir necha daryo qo'yilishi mumkin.

- *To'liq funktsional bog'liqlik:* Agar B atribut qiymati A tarkibiy atributning qiymatiga funktsional bog'liq, ammo A tarkibiy atribut qiymatining biror bir qismiga funktsional bog'liq bo'lmasa, u holda A tarkibiy atribut bilan B atribut to'liq funktsional bog'liq deyiladi.

- *Ko'p qiymatli funktsional bog'liqlik:* Agar ixtiyoriy vaqtda A atributning har bir qiymatiga B atributning ko'p qirrali mos qiymati mavjud bo'lsa, u holda A atribut B atributni bir jinsli aniqlaydi va bu ko'p qiymatli funktsional bog'lanish deyiladi.

Masalan, talabalar o'qiyotgan fanlar va olgan baholar. Baho atributi bir nechta (1, 2, 3, 4, 5) qiymatlarga ega. Bundan tashqari talaba o'qiyotgan fanlar soni ham chegaralangan.

5.2.1-jadval

Ko'p qiymatli funktsional bog'liqlikka misol "O'qish jadvali"

Fan	O'qituvchi	Darslik
Dasturlash	Abdullaev I.	Liberti Dj. Osvoy samostoyatelno C++
Dasturlash	Rahmonov U.	Forsayt R. Pascal dlya vsekh
Dasturlash	Qobilov.R.	Liberti Dj. Osvoy samostoyatelno C++
Dasturlash	Kovalev N.	Forsayt R. Paskal dlya vsekh
Dasturlash	Rahmonov U.	Faysman K. Prof. programmirovaniye na Paskale

Jadvalda ko'p qiymatli bog'liqlik bor "Fan-O'qituvchi": Fan (Dasturlash misolida) bir nechta o'qituvchi tomonidan o'qilishi mumkin (misolda Abdullaev, Rahmonov, Qobilov, Kovalev). Yana boshqa bog'liqlik ham bor "Fan-Darslik": informatikani o'tishda "Pascal dlya vsekh", "Osvoy samostoyatelno C++" va "Professionalnoe programmirovaniye na Paskale" darsliklari ishlatiladi. Bunda "O'qituvchi" va "Darslik" funktsional bog'lanmagan, bu esa ortiqchalikka olib keladi (yangi darslikni yozish uchun yana ikkita yangi qator yozish kerak bo'ladi). Bu jadvalni ikkiga ajratganda ish yaxshilanadi: (Fan- O'qituvchi va Fan-Darslik)

- *Tranzitiv funktsional bog'liqlik:* Agar B atribut A atributga, C atribut B atributga funktsional bog'lik bo'lsa, u holda A atribut C atributga tranzitiv funktsional bog'liq deyiladi. O'z navbatida A atribut B atributga funktsional bog'liq bo'lmaydi.

Bir necha atributning o'zaro bog'liq emasligi: Bir necha atribut o'zaro bog'liq emas deyiladi, qachonki ulardan birortasi ham boshqa atributga funktsional bog'liq bo'lmasa.

Kalit bo'lmagan atribut. Birlamchi kalit tarkibiga kirmagan har bir atribut kalit bo'lmagan atribut deyiladi.

Munosabatlar sxemasining (jadval ma'lumot kiritish, o'zgartirish va o'chirish uchun qulay xossaga ega bo'lgan) relyatsion variantini qurish munosabatlar sxemasini me'yorlashtirish orqali amalga oshiriladi [2]

5.3. Ma'lumotlar bazasida kalit tushunchasi

Kalit haqida yuqoridagi mavzularda aytib o'tilganlarga quyidagi qo'shimcha ma'lumotlarni ham keltiramiz.

Har bir munosabat kalitga (birlamchi kalit), ya'ni kortejni bir qiymatli identifikatsiyalovchi atributga (oddiy kalit) yoki atributlar to'plamiga (tarkibiy yoki murakkab kalit) ega. Qaralayotgan munosabatda kalit bo'lmagan atribut yoki atributlar guruhi boshqa munosabatda kalit bo'lsa, bu atribut yoki atributlar guruhi tashqi kalit (ikkilamchi kalit) deb ataladi.

Agar biror jadval tashqi kalitga ega bo'lsa, u holda:

- mos birlamchi kalitga ega bo'lgan jadval bilan mantiqan bog'langan bo'ladi;
- bu bog'lanish birga ko'p xarakterga ega bo'ladi.

Kalitni ishlatishda uzun matnli va qiymatli atributni olish tavsiya etilmaydi (butun sonli atributlarni ishlatish foydaliroqdir). Masalan, talabalar identifikatsiyasi uchun unikal (qaytarilmaydigan yagona ma'nosida) imtihon daftarchasi nomerini ishlatish mumkin. Kalit sifatida familiyasini, guruh nomerini va boshqa qo'shimcha atributlarni olish mumkin. Ammo, bunday hollarda ikkita bitta familiyali talaba chiqib qolishi ham mumkin. Imkon darajasida murakkab kalitlardan foydalanmaslik maqsadga muvofiq hisoblanadi. Agar tabiiy

oddiy kalit mavjud bo'lmasa, yangi sun'iy oddiy kalit qo'llash lozim.

Agar C mohiyati ikkita A va B mohiyatni bog'lasa, u holda A va B mohiyatlar birinchi kalitga mos tashqi kalitni tashkil etishi kerak.

Agar B mohiyati ikkinchi A mohiyatni belgilasa, u holda A mohiyat birinchi kalitga mos tashqi kalitni tashkil etishi kerak.

Har qanday o'zaro bog'lanishlarda bo'lgan mohiyatlarni belgilash uchun (prujinali, xarakteristik, belgilash assotsatsiya kabi) yangi birlashtirilgan termin 'Maqsad' yoki 'Maqsadli mohiyat' ishlatiladi.

Assotsatsiyalarni ko'rsatish usulini tanlash muammosini qarashda MB da 'Qaysi kalit tashqi kalit?' degan savolga javob olish lozim. Keyinchalik, har bir tashqi kalit uchun uchta savolni echish kerak bo'ladi:

- Ishlatilayotgan tashqi kalit aniqlanmagan qiymat qabul qilishi mumkinmi (Null-qiymat)?
- Tashqi kalitga murojaat qilayotgan maqsadli mohiyatni o'chirishga harakat bo'lganda nima yuz berishi mumkin?
- Tashqi kalitga murojaat qilayotgan maqsadli mohiyat birinchi kalitni yangilashga harakat qilinganda nima yuz berishi mumkin?

Shunday qilib, har bir tashqi kalit uchun MBni loyihalovchi loyihada tashqi kalitni tashkil qiluvchi atribut yoki atributlar kombinatsiyasini yuqoridagi savollarga muvofiq tashkil etishi kerak.

5.4. Me'yoriy shakllar

Relyatsion MB me'yorlashtirish shakllari nazariyasida quyidagi me'yoriy shakllar ketma – ketligi ajratib ko'rsatiladi:

- birinchi me'yoriy shakl – 1MSh (1NF);
- ikkinchi me'yoriy shakl – 2MSh (2NF);
- uchinchi me'yoriy shakl – 3MSh (3NF);

- Boys-Kodd me'yoriy shakli – BKMSH (BCNF);
- to'rtinchi me'yoriy shakl – 4MSH (4NF);
- beshinchi me'yoriy shakl yoki proektsiya – ulash me'yoriy shakli – 5MSH (5NF yoki PJ/NF).

Har bir me'yoriy shakl uchun bir qator talablar mavjud bo'lib, jadval ushbu talablarga javob bersa, jadval ushbu me'yoriy shaklga ega deyiladi.

Me'yoriy shakllarning asosiy xossalari:

- har bir navbatdagi me'yoriy shakl qaysidir ma'noda oldingisiga qaraganda yaxshi;
- navbatdagi me'yoriy shaklga o'tganda oldingi me'yoriy shakl xossalari saqlab qolinadi.

Umuman olganda me'yorlashtirilmagan jadvallar, ya'ni takrorlanuvchi ma'lumotlar guruhiga ega jadvallar relyatsion MBga kiritilmaydi.

Me'yorlashtirish jarayoni bir me'yoriy shakldan boshqasiga o'tish amallaridan iborat bo'lib, unda har bir navbatdagi jadval oldingisiga nisbatan yaxshi xossalarga ega bo'ladi.

Quyida eng muhim me'yoriy shakllarni ko'rib chiqamiz.

Birinchi me'yoriy shakl (1MSH): Agar jadval atributlarining har biri bo'linmas (atomar) qiymat qabul qilsa, u holda jadval birinchi me'yoriy shaklga ega deyiladi.

Takrorlanuvchi guruhlariga ega bo'lmagan jadval birinchi me'yoriy shaklga ega bo'ladi. Jadval birinchi me'yoriy shaklga ega bo'lishi uchun undagi takrorlanuvchi guruhlarni bartaraf etish lozim. 5.1.1 misolda keltirilgan "Fam" atributi qiymatlari ham takrorlanuvchi guruh hisoblanadi. Shuning uchun, bu ko'rinishdagi atributlar uchun familiya, ism va otasining ismi ma'lumotlarini uchta atribut orqali ifodalash maqsadga muvofiq hisoblanadi. Ammo, amaliyotda shu shakldagi ma'lumotga ko'pincha e'tibor berishmaydi.

Birinchi me'yoriy shaklning quyidagicha ta'rif ham mavjud:

Agar jadvalning birorta ham qatori har bir atributi bo'yicha

birtadan ortiq qiymat qabul qilmasa va kalit atributlarining birortasi ham bo'sh bo'lmasa, u holda jadval birinchi me'yoriy shaklga ega deyiladi.

Kategoriya	Tovarlar
Kitoblar	Urush va tinchlik, Alifbe
O'yinchoqlar	Yula

Bu misolda katalardan biri ikkita elementdan iborat, ya'ni "Urush va tinchlik", "Alifbe". Shuning uchun u atomar emas.

Uni quyidagi jadvalda tuzatib, 1NF shaklni hosil qilamiz.

Kategoriya	Tovarlar
Kitoblar	Urush va tinchlik
Kitoblar	Alifbe
O'yinchoqlar	Yula

Ikkinchi me'yoriy shakl (2MSh): Jadval ikkinchi me'yoriy shaklda deyiladi, qachonki u birinchi me'yoriy shaklga ega bo'lsa hamda birlamchi kalitga ega bo'lib, kalit bo'lmagan atributlar birlamchi kalitga to'liq funksional bog'liq bo'lsa.

Bir nechta misollarni ko'rib chiqamiz.

1-misol. Quyidagi jadvalda tovarlar, ularning kategoriyalari, chegirmalar va uning vaqti haqidagi ma'lumotlar keltirilgan. Bu jadval 1-me'yoriy shaklda 1NF. "Tovar" atributi birlamchi kalit bilan to'liq funksional bog'liqlikda emas. Shuning uchun u ikkinchi me'yoriy shaklda bo'la olmaydi.

Kategoriya	Sana	Chegirma	Tovar
Kitoblar	10.10.2008	10%	PHP for dummies
Noutbuklar	11.10.2008	20%	Acer
Kitoblar	10.10.2008	10%	Windows XP

Ikkinchi me'yoriy shaklga keltirish uchun uni quyidagicha dekompozitsiya qilamiz (pastki jadvallar).

Korpus №	Komendant tabel №
1	100
2	102

Kategoriya	Sana	Chegirma
Kitoblar	10.10.2008	10%
Noutbuklar	11.10.2008	20%

↓

Kategoriya	Tovar
Kitoblar	PHP for dummies
Noutbuklar	Acer
Kitoblar	Windows XP

2-misol. Quyida 1-me'yoriy shaklda bo'lgan, ammo 2-me'roy shaklda bo'lmagan jadval namunasi keltirilgan.

Auditoriyalar (*korpus№, auditoriya№, maydon kv.m, korpus komendantining tabel nomeri №*); **Primary key** (*korpus №, auditoriya№*);

Korpus №	Auditoriya №	Maydon kv. m	Komendant Tabel №
1	417	25	100
2	315	30	102
1	412	25	100

Ushbu jadval 1 me'yoriy shaklda. "Komendant tabel №" atributi kalit bilan to'liq funksional bog'liq emas. Shuning uchun u 2 me'yoriy shaklda emas. Quyidagicha dekompozitsiyani bajarib uni 2-MSH ga o'tkazamiz.

<i>Korpus №</i>	<i>Auditoriy a №</i>	<i>Maydo n kv. m</i>
1	417	25
2	315	30
1	412	25

Uchinchi me'yoriy shakl (3MSh). Jadval uchinchi me'yoriy shaklda deyiladi, qachonki u birinchi va ikkinchi me'yoriy shaklga ega bo'lsa, hamda uning kalitsiz atributlari boshqa bir kalitsiz atributlarga funktsional bog'liq bo'lmasa.

Uchinchi me'yoriy shaklning quyidagicha ta'rifi ham mavjud. Jadval uchinchi me'yoriy shaklda deyiladi, qachonki u birinchi va ikkinchi me'yoriy shaklga ega bo'lsa hamda birlamchi kalit tarkibiga kirmagan, ammo birlamchi kalit bilan tranzitiv funktsional bog'liqlikda bo'lgan atributlarga ega bo'lmasa.

Uchinchi me'yoriy shaklning umumiy ta'rifi. Jadval uchinchi me'yoriy shaklda deyiladi, qachonki u birinchi va ikkinchi me'yoriy shaklga ega bo'lsa, hamda birlamchi kalit tarkibiga kirmagan atributlarining birortasi ham kalit imkoniyatli atribut bilan tranzitiv funktsional bog'liqlikda bo'lmasa.

3-misol. Quyida 2 me'yoriy shakldagi jadval namunasi keltirilgan.

<i>Tabel №</i>	<i>Familiya</i>	<i>Ism</i>	<i>Otchestvo</i>	<i>Mansab kodi</i>	<i>Oklad</i>
100	Toshev	Hamid	Axmedovich	1	300
101	Salomov	Shavkat	Po'latovich	2	250

Bu jadval 2 me'yoriy shaklda. "Oklad" atributi "Mansab kodi" atributi bilan tranzitiv funktsional bog'liq bo'lganligi sababli u 3 me'yoriy shaklda emas. Uni quyidagicha dekompozitsiyalab 3-MSH ga o'tkazamiz.

<i>Tabel №</i>	<i>Familiya</i>	<i>Ism</i>	<i>Otchestvo</i>	<i>Mansab kodi</i>	<i>Mansab kodi</i>	<i>Oklad</i>
00	Toshev	Hamid	Axmedovich	1	1	300
101	Salomov	Shavkat	Po'latovich	2	2	250

Dekompozitsiya natijasida ushbu jadval 3-me'yoriy shaklga keldi.

MB tarkibiga kiruvchi har bir jadval uchinchi me'yoriy shaklga ega bo'lsa, bu MB ishlatish uchun yaroqli hisoblanadi. Shuning uchun, amaliyotda ko'pchilik MBlar uchinchi me'yoriy shaklgacha olib kelinadi.

MB jadvallari shunday loyihalashtiriladiki, bu jadvallarning birortasida ham kalit bo'lmagan atributlar qisman bo'lsada tranzitiv bog'liqlikka ega bo'lmasligi lozim. Bu singari muammolar mavjud bo'lishi mumkinligi uchun relyatsion tizim nazariyotchilar Kodd va Boys 3MSh uchun anchagina qat'iy tarifni taklif qilishdi.

Boys-Kodd me'yoriy shakli barcha mumkin bo'lgan kalitlarni ham hisobga olgan holda funktsional bog'liqlikka asoslangan. Bundan tashqari, bu shaklda 3MSh bilan solishtirilganda anchagina qat'iy cheklov ko'rib chiqilgan.

Boys-Kodd me'yoriy shakli (BKMSH): Jadval Boys-Kodd me'yoriy shaklida deyiladi, qachonki atributlari orasidagi har qanday funktsional bog'lanish kalit hisoblangan atributlar bilan to'liq funktsional bog'lanishga olib kelsa.

Boys-Kodd shaklining quyidagicha ta'rifi ham mavjud: Jadval Boys-Kodd me'yoriy shaklida deyiladi, qachonki faqat va faqat uning har bir determinanti potentsial kalit hisoblansa.

Bir atribut yoki atributlar guruhi determinant hisoblanadi, qachonki boshqa atribut unga to'liq funktsional bog'lanishda bo'lsa.

3MSh va BKMSH ning farqi:

○ B atribut birlamchi kalit bo'lsa, A atribut esa mumkin bo'lgan kalit bo'lishi shart bo'lmasa, A atributning B atributga funktsional bog'lanishga 3MSh da ruxsat etiladi.

○ A atribut faqat mumkin bo'lgan kalit bo'lsagina bunday bog'lanishga BKMSH da ruxsat etiladi.

Keyingi beshinchi me'yoriy shakllarda jadvallar atributlari orasida nafaqat funktsional bog'lanishlar, balki ko'p qiymatli funktsional bog'liqliklar ham hisobga olinadi. Uni tavsiflash uchun to'liq dekompozitsiya tushunchasidan foydalaniladi.

To'liq dekompozitsiya – jadvalni barcha bog'liqliklarini saqlab qolgan holda, o'zaro bog'langan sodda va ixcham bir nechta jadvallarga ajratishdir.

To'liq dekompozitsiyada bo'laklarga ajratilgan jadvallarda soxta qatorlar mavjud bo'lmasligi va uning bog'lanishlari, bo'lingan jadval ichidagi bog'lanishlar bilan to'liq bir-biriga mos kelishi lozim. Agar bo'laklar birlashtirilsa boshlang'ich jadval hech qanday yo'qotishlarsiz yoki ortiqcha ma'lumotlarsiz tiklanishi lozim.

Beshinchi me'yoriy shakl (5MSh): Jadval beshinchi me'yoriy shaklda deyiladi, qachonki har bir to'liq dekompozitsiya barcha proektsiyalari mumkin bo'lgan kalitlarni saqlasa. Birorta to'liq dekompozitsiyaga ega bo'lmagan jadval ham beshinchi me'yoriy shaklda bo'ladi.

To'rtinchi me'yoriy shakl (4MSh): Agar to'liq dekompozitsiya ikkita proektsiya birlashmasidan iborat bo'lsa, beshinchi me'yoriy shaklning xususiy holi bo'ladi [8].

Nazorat savollari

1. MBni loyihalash deganda nimani tushunasiz?
2. MBni loyihalashning asosiy maqsadi nimadan iborat?
3. MB tuzilmasini yaxshi tashkil qilish qanday imkoniyat yaratadi?
4. Jadvallar orasidagi qanday bog'lanishlar mavjud?

5. Takrorlanuvchi guruhlar tushunchasi haqida tushuncha bering.

6. Takrorlanuvchi guruhlarni yo'qotish usullari haqida tushuncha bering.

7. Takrorlanuvchi guruhlarni yo'qotishning noto'g'ri usulini izohlang.

8. Takrorlanuvchi guruhlarni yo'qotishning to'g'ri usulini izohlang.

9. Me'yorlashtirishni asosiy maqsadini tushintiring?

10. Dekompozitsiyalashni asosiy maqsadi nimalardan iborat?

11. Funktsional bog'liqlik deganda nimani tushinasiz?

12. To'liq funktsional bog'liqlik nimani anglatadi?

13. Ko'p qiymatli funktsional bog'liqlikni qanday tushinasiz?

14. Tranzitiv funktsional bog'liqlik qanday bo'ladi?

15. Bir necha atributning o'zaro bog'liq emasligi qanday nomlanadi?

16. Kalit bo'lmagan atribut haqida tushuncha bering.

17. Kalitlar xususiyatlarini tavsiflang.

18. Birlamchi kalit xususiyatlarini tavsiflang.

19. Ma'lumotlarni me'yorlashtirish shakllari nazariyasi haqida tushuncha bering.

20. Birinchi me'yoriy shakl (1MSh) ta'rifini keltiring?

21. Ikkinchi me'yoriy shakl (2MSh) ta'rifini keltiring?

22. Uchinchi me'yoriy shakl (3MSh) ta'rifini keltiring?

23. Boys-Kodd me'yoriy shakli (BKMSH) qanday bo'ladi?

24. To'rtinchi me'yoriy shakl (4MSh) qanday bo'ladi?

25. Me'yoriy bo'lmagan shakl qanday bo'ladi?

VI. MA'LUMOTLAR BAZASINI MANTIQUIY LOYIHALASH

6.1. Relyatsion ma'lumotlar bazasini loyihalash asoslari

Mbni ishlatishdan maqsad predmet soha ma'lumotlarini saqlash va ularni qayta ishlashdan iboratdir. MBni tuzish uni loyihalashdan boshlanadi. Loyihalashda predmet soha ma'lumotlari guruhlanadi va ular guruh shaklida MBda saqlanadi. Relyatsion MBda guruh sifatida jadval nazarda tutiladi. Demak, loyihalashda MB jadvallari tanlangan relyatsion MBBT talablariga muvofiq me'yorlashtirilgan holda yaratilishi lozim.

Loyihalashda ikkita yondashuvdan foydalanish mumkin.

Birinchi yondashuvda dastlab baza yaratishdan maqsad ularni echish bo'lgan asosiy masalalar hamda masalalarning ma'lumotlarga ehtiyoji aniqlanadi.

Ikkinchi yondashuvda predmet sohasi o'rganiladi, uning ma'lumotlari tahlildan o'tkaziladi va predmet sohasining bir turdagi ob'ektlari belgilanadi.

Amaliyotda har ikkala yondashuvning birikmasi ma'qulroqdir. Chunki, odatda dastlabki bosqichda barcha masalalar haqida to'liq ma'lumotlar bo'lmaydi, natijada MBni loyihalash va yaratishni barcha masalalarni tuzish bajarilgunga qadar kechiktirishga to'g'ri keladi. Bu texnologiya qo'llanilsa, ishlab chiqish jarayonining xohlagan bosqichida MBga o'zgartirishlar kiritish va uning tuzilishini ilgari kiritilgan ma'lumotlariga ziyon etkazmasdan turib modifikatsiya qilish (yangilash)ga imkon beradi. Bu texnologiya MBdan turli axborotlarni olish zaruriyati haqidagi dastlabki ma'lumotlardan foydalanishni ko'zda tutadi.

MBni yaratish uchta asosiy bosqichdan iborat:

- kontseptual;
- mantiqiy;

- jismoniy.

MBni kontseptual loyihalash – ma'lumotlarni biror bir jismonan aspektda tadbiq qilishni ko'zlamagan holda, tashkilotda foydalanilayotganligiga muvofiq modelini yaratish.

Kontseptual loyihalashda predmet sohasi ma'lumotlarining axborot-mantiqiy modeli qurilishi (loyihalanishi) lozim. Bu modelni ishlab chiqish predmet sohasini tekshirish natijasida olingan tavsifga asoslanadi. Dastlab predmet sohaning MB bo'lg'usi foydalanuvchilarining kontseptual talablari o'rganiladi va aniqlashtiriladi. MB saqlashi kerak bo'lgan ma'lumotlar, foydalanuvchilarning zarur so'rovlari, masalalari hamda ilovalarni bajarishni ta'minlashi lozim bo'lgan predmet sohasi ma'lumotlari tarkibi va tuzilishi o'rganiladi. Bu ma'lumotlar odatda turli hujjatlar, qoidalar, ishlov berish shartlari asosida tahlil qilinadi. Ba'zi bir foydalanuvchilarning xususiy talablari umumiy talablarga zid bo'lsa, ular e'tiborga olinmaydi. Aniqlangan ma'lumotlarning tahlili, axborot ob'ektlarni ajratish uchun qo'llanuvchi rekvizitlarning funktsional bog'liqligini bilish hamda ob'ektlar orasidagi tuzilmaviy aloqalarni aniqlash axborot-mantiqiy modelini qurishga imkon beradi. Axborot-mantiqiy model axborot ob'ektlari bog'liqligining ierarxiyasini aks ettiruvchi namunali ko'rinishda berilish lozim. Natijada predmet sohani ma'lumotlarining, ularning jismoniy saqlash usuli ko'rsatilmagan shakldagi o'zaro bog'langan kontseptual modeli yaratiladi.

Bu modelda ma'lumotlar tashkilotda qanday tartibda mavjud va ishlov berilayotgan bo'lsa, xuddi shu holati bo'yicha tavsiflanadi. Bu tavsifning foydalanuvchi ishlatayotgan holatga muvofiqligi doimiy ravishda tekshirib boriladi.

MB ni mantiqiy loyihalash – ma'lumotlarni kontseptual modeli bo'yicha maqsadli, MBBT tipini hisobga olgan holda biror bir jismonan aspektga tadbiq qilish modelini yaratish.

Ishlab chiqilgan kontseptual model bog'lanishlarini tanlangan aniq MBBTga tadbiq qilishning imkoni bo'lmاسligi mumkin. Shuning uchun, kontseptul model tanlangan aniq MBBT uchun qaytadan ishlab chiqiladi va u mantiqiy model deyiladi.

Mantiqiy loyihalashning maqsadi predmet sohaning o'rganilayotgan qismi kontseptul modeliga asoslangan holda aniq bir MBBT uchun mantiqiy bog'lanishlarga ega bo'lgan mantiqiy modelni qurishdan iboratdir.

Bu bosqichda aniq bir MBBT uchun MB jadvallari, jadvallarining tuzilishi va jadvallar orasida bog'lanishlar yaratiladi. Jadvallarni yaratishda me'yorlashtirish talablarining bajarilishiga rioya qilinadi.

Hozirgi vaqtda ko'pchilik MB relyatsion modelga muvofiq loyihalashtiriladi. Shuning uchun, loyihalashda Koddning 12 ta qoidasiga rioya qilish talab qilinadi.

Koddning 12 qoidasi. Kodd relyatsion modeli quyidagi asosiy tushunchalarga asoslanadi: jadval, munosabat, satr, ustun, birlamchi kalit, tashqi (ikkilamchi) kalit. Kodd relyatsion MB bilan ishlovchi MBBT uchun 12 ta qoida ishlab chiqdi. Ular relyatsion MB tushunchasini rasmiy ta'rifi hisoblanadi. Relyatsion MBBT qanoatlantirishi lozim bo'lgan 12 qoida quyidagicha:

1. *Axborot qoidasi.* MBdagi barcha ma'lumotlar mantiqiy darajada ifodalanishi kerak bo'lib, ular faqat jadvalda saqlanadigan qiymat ko'rinishida ifodalanishi lozim.

2. *Kafolatlangan murojaat qilish qoidasi.* Relyatsion MBdagi barcha va har bir ma'lumot elementiga mantiqiy murojaat jadval nomi, birlamchi kalit va ustun nomi orqali amalga oshirilishi lozim.

3. *Haqiqiy bo'lmagan qiymatni qo'llab-quvvatlash qoidasi* (bo'sh qiymatni mavjud bo'lishi qoidasi). Haqiqiy relyatsion MBda bo'sh qiymatni qo'llab-quvvatlash amalga oshirilishi lozim. Bu

qiymat nol uzunlikdagi qiymatlar satri, bo'sh probel simvol satri va nol soni yoki ixtiyoriy boshqa sondan farq qilishi kerak va biror ma'lumotlar tiliga bog'liq bo'lmasligi lozim.

4. *Relyatsion modelga asoslangan dinamik katalog qoidasi.* Mantiqiy darajadagi MB tavsifi shunday ko'rinishda ifodalanishi lozimki, mos huquqga ega bo'lgan foydalanuvchilar relyatsion til orqali asosiy ma'lumotlar bilan ishlaganlarida ularni qo'llash imkoniyatiga ega bo'lsinlar.

5. *Relyatsion tizimda turli tillarni va foydalanuvchi bilan ishtirakdagi turli rejimlarni qo'llab-quvvatlashi lozim.* Relyatsion MBBTda hech bo'lmaganda bitta biror bir qat'iy aniqlangan sintaksisli simvollar satri ko'rinishdagi operatorlarga ega bo'lgan til bo'lishi lozim. Bu til quyidagi elementlarni qo'llab-quvvatlashi lozim:

- ma'lumotlarni aniqlash;
- ko'rinishlarni aniqlash;
- ma'lumotlar ustida amal bajarish;
- yaxlitlik shartlari;
- ma'lumotlarga murojaat qilish huquqlari;
- tranzaktsiya chegarasi (boshlanishi, tugallanishi).

6. *Ko'rinishlarni o'zgartirish qoidasi.* Nazariy jihatdan o'zgartirish mumkin bo'lgan barcha ko'rinishlarni o'zgartirish imkoniyati.

7. *Qo'shish, o'zgartirish va o'chirish qoidasi.* Munosabatlar bilan ishlashda bir operand orqali nafaqat ma'lumotlarni o'qish imkoniyati, balki ularni qo'shish, o'zgartirish va ko'chirish imkoniyatlari ham bo'lishi lozim.

8. *Jismoniy ma'lumotlarga bog'liqlilik qoidasi.* ma'lumotni saqlash yoki ularga murojaat qilish uslubida ixtiyoriy o'zgartirish bo'lishidan kat'iy nazar ma'lumotlar bilan ishlovchi amaliy dasturlar va utilitalar faqat mantiqiy darajadagi ma'lumotlar bilan ishlashi lozim.

9. *Mantiqiy ma'lumotlarga bog'liqmaslik qoidasi.* Tayanch jadvaldagi ma'lumotlarga ixtiyoriy o'zgartirish kiritilganda ham ular bilan ishlovchi amaliy dasturlar va utilitalar o'zgartirilmasiligi lozim.

10. *Yaxlitlik shartining bog'liqmasligi qoidasi.* Yaxlitlik shartini aniq relyatsion MB uchun maxsus tilda, amaliy dasturda emas, balki MBBTni o'zida aniqlash imkoni bo'lishi lozim.

11. *Tarqalishga bog'liqmaslik qoidasi.* Relyatsion MBBT biror bir mijozning talablariga bog'liq bo'lmasligi lozim.

12. *Yagonalik qoidasi.* Agarda relyatsion MBda quyi darajadagi til (bir martada bitta yozuvni qayta ishlovchi) mavjud bo'lsa, uning yuqori darajadagi relyatsion tilini ishlatish lozim.

Birinci qoida, oldinroq keltirilgan relyatsion MBni noformal aniqlanishini eslatadi. 2-qoida esa, MBdan ma'lumotlarni izlashda birlamchi kalitni vazifasini ko'rsatadi. Jadval nomi talab qilingan jadvalni, ustun nomi talab qilingan ustunni, birlamchi kalit esa izlanayotgan ma'lumot elementi saqlanayotgan satrni topish uchun xizmat qiladi. 3-qoida esa relyatsion MBda bo'sh qiymatni mavjud bo'lishi zarurligini anglatadi. 4-qoida relyatsion MB o'zini o'zi tavsiflashi lozimligini anglatadi. 5-qoida, MBBTda hech bo'lmasa bitta (SQL) mavjud bo'lgan relyatsion MBtilini ishlatilishini talab qiladi. Bunday til MBBT ning barcha asosiy funktsiyalarini – MBni himoyalash, ma'lumotlarni o'qish va kiritish, MB himoyasini amalga oshirish ishlarini bajaradi. 6-qoida ko'rinishlarga bag'ishlangan bo'lib, ular orqali turli foydalanuvchilarga MB tuzilmasini turlicha ko'rinishda ko'rsatadi. 7-qoida, MB tabiatiga ko'ra to'plamga mo'ljallanganligini anglatadi. Bu qoida qo'shish, o'chirish va o'zgartirish amallarini satrlar to'plami ustida bajarilishini talab qiladi. Faqat bitta satr ustida amal bajaradigan operatorlardan foydalanishni ta'qiqlaydi.

Shunday qilib, foydalanuvchi mantiqiy modelga asoslangan holda predmet soha MBdan kontseptual talablarini

qanoatlantiruvchi ma'lumotlarini oladi va ularga ishlov berishni talab qiladi. Mantiqiy model foydalanuvchining tashqi modeli hisoblanadi. U qanchalik yaxshi loyihalashtirilgan bo'lsa, shunchalik darajada predmet sohani aniq va to'liq akslantiradi hamda bu predmet sohaga tegishli avtomatlashtirilgan axborot tizimi yuqori darajada aniq va to'liq ishlaydi.

MB ni jismoniy loyihalash. Predmet soha ma'lumotlarini MBBT yordamida kompyuter jismoniy xotirasiga kiritish, saqlash hamda kerakli ma'lumotlarni olish va ma'lumotlarga ishlov berishda ma'lum bir usullarni qo'llashni tashkil qilishdir.

Mantiqiy model yaratilgandan so'ng u aniq bir MBBT yordamida kompyuter jismoniy xotirasiga kiritiladi. Endi kompyuter jismoniy xotirasida mantiqiy modelning tasviri hosil bo'ladi. U orqali predmet soha ma'lumotlarini kiritish, saqlash va kerakli ma'lumotlarni olish hamda ma'lumotlarga ishlov berish mumkin. Ma'lumotlarni olish hamda ularga ishlov berishda ma'lum bir usullar qo'llaniladi. Bu jismoniy yoki ichki model hisoblanadi.

6.2. Ma'lumotlarni turlarga guruhlash

Jadvalli ma'lumotlar manzili satr va ustunlar kesishmasi bilan aniqlanadi. MBda ustunlar-atributlar yoki maydonlar, satrlar esa yozuvlar deb ataladi. Yozuvlar MBning tuzilmasini, maydonlar esa, unda joylashgan ma'lumotlarni tashkil etadi.

Maydonlar - MB tuzilmasining asosiy elementlaridir. Ular ma'lum xususiyatlarga ega bo'ladilar. Har qanday maydonning asosiy xususiyati uning uzunligidir. Maydon uzunligi undagi belgilar soni bilan ifodalanadi.

Maydonning yana bir xususiyati, uning nomidir. Interfeysda maydonda uning nomidan tashqari yana imzo xususiyati ham mavjud.

Imzo - ustunning sarlavhasida aks ettiriladigan axborotdir.

Uni maydon nomi bilan aralashtirib yubormaslik lozim.

Agar imzo berilmagan bo'lsa, sarlavhada maydon nomi yozib qo'yiladi. Turli tipdagi maydonlar turli maqsadlarda ishlatiladi va turli xossalarga ega bo'ladi.

Har bir maydon tipga ega. Tiplar sonli, harfiy, grafik, sana, mantiqiy va hokazo bo'lishi mumkin. Maydon tipiga muvofiq unda ma'lumotlar ifodalanadi.

MS Access MBBT misolida maydonlarning xususiyati bilan tanishib chiqamiz:

1. *Oddiy matn maydoni*. Belgilar soni 255 dan oshmasligi kerak.

2. *MEMO - katta o'lchamli matn maydoni*. Belgilar soni 65535dan oshmasligi shart. Oddiy matn va MEMO maydonida hisob ishlarini bajarib bo'lmaydi.

3. *Sonli maydon*. Sonli ma'lumotlarni kiritishga xizmat qiladi va hisob ishlarini bajarishda foydalaniladi. Bu maydon 1,2,4,8 va 16 baytli bo'lishi mumkin.

4. *Sana va vaqt maydoni*. Bu maydon sana va vaqtni bichimlangan holda saqlab qo'yish imkonini beradi (01.06.01 20:29:59). 8bayt o'lchamga ega.

5. *"Pul birligi" nomi bilan ataluvchi maydon*. Bu maydondan hisob kitob ishlarini yuritishda foydalaniladi.

6. *Hisoblagich maydoni*. Bu maydon 4 bayt uzunlikka va avtomatik ravishda ma'lum songa oshib borish xususiyatiga ega. Ushbu maydondan yozuvlarni nomerlashda foydalanish qulaydir.

7. *Mantiqiy amal natijasini saqlovchi maydon*. Bu maydon 'rost' (true) yoki 'yolg'on' (false) qiymatni saqlaydi. Maydon o'lchami 1bayt.

8. *OLE-nomi bilan yurituvchi maydon*. Bu maydon Excel jadvalini, Word hujjatini, rasm, ovoz va boshqa shu kabi ma'lumotlarni ikkilik sanoq sistemasida saqlaydi. Maydon o'lchami 1 Gbaytgacha.

9. *Giperssilka maydoni*. Bu maydon belgi va sonlardan iborat bo'lib, biror fayl yoki saytga yo'l ko'rsatadi.

10. *Qiymatlar ro'yxatidan iborat bo'lgan maydon*. Bu maydon bir qancha qiymatlardan iborat bo'lgan ro'yxatdan tanlangan aniq bir qiymatni saqlaydi.

Xuddi shuningdek, MySQL, MS SQL Server, Oracle MBBTlarida uziga xos tiplar qo'llaniladi. Barcha relyatsion MBBTd ishlatiladigan tiplar SQL tili standartini qanoatlantirishi lozim.

Nazorat savollari

1. Ishlab chiqish usullari haqida tushuncha bering.
2. MBni kontseptual loyihalash texnologiyasini izohlag.
3. MBni mantiqiy loyihalash haqida ma'lumot bering.
4. MBni jismoniy loyihalash deganda nimani tushinasiz?
5. Predmet sohasi ma'lumotlarining axborot-mantiqiy modelini ishlab chiqish qanday amalga oshiriladi?
6. MBning mantiqiy tuzilishini aniqlash deganda nimani tushunasiz?
7. MBning jadvallarini tuzish deganda nimani tushunasiz?
8. Axborot qoidasiga izoh bering.
9. Kafolatlangan murojaat qilish qoidasi deganda nimani tushunasiz?
10. Haqiqiy bo'lmagan qiymatni qo'llab-quvvatlash qoidasiga izoh bering.
11. Relyatsion modelga asoslangan dinamik katalog qoidasiga izoh bering.
12. Relyatsion tizim turli tillarni va foydalanuvchi bilan ishdaydigan turli rejimlarni qo'llab-quvvatlashi lozim deganda nimani tushunasiz?
13. Ko'rinishlarni o'zgartirish qoidasiga izoh bering.
14. Qo'shish, o'zgartirish va o'chirish qoidasiga izoh bering.
15. Jismoniy ma'lumotlarga bog'liqmaslik qoidasiga izoh bering.

16. Mantiqiy ma'lumotlarga bog'liqmaslik qoidasiga izoh bering.

17. Yaxlitlik shartining bog'liqmasligi qoidasiga izoh bering.

18. Tarqalishga bog'liqmaslik qoidasiga izoh bering.

19. Yagonalik qoidasiga izoh bering.

6.3. Axborot tizimlarini loyihalashning zamonaviy CASE-vositalari

6.3.1. Axborot tizimlarini loyihalashda CASE-texnologiyalar o'rni

Oxirgi o'n yillikda dasturlash texnikasida yangi yo'nalish paydo bo'ldi - CASE (Computer-Aided Software/System Engineering) – kompyuter yordamida axborot tizimlarining dasturiy ta'minotini ishlab chiqish degan ma'noni bildiradi. Dastlab CASE iborasining ma'nosi faqat dasturiy ta'minot bo'lgan bo'lsa, hozirgi vaqtga kelib yangi ma'noni ya'ni, murakkab avtomatlashtirilgan axborot tizimlarini to'liq ishlab chiqish jarayoni ma'nosini anglatadi. Endi CASE-vositalari iborasi ostida axborot tizimlarini yaratish va joriy etish jarayonini qo'llab-quvvatlovchi dasturiy vositalar tushuniladi. Ushbu jarayonga, talabning ifodalanishi va tahlili, amaliy dastur ta'minoti va ma'lumotlar bazasini loyihalash, kod generatsiyasi, testlash, hujjatlashtirish, sifatni ta'minlash, konfiguratsiyali boshqaruv, loyihani boshqarish va boshqa jarayonlar kiradi.

CASE-vositalari ko'rgazmali grafikli texnikaga asoslangan, unda axborot tizimlarining turli toifadagi modellarini ifoda etish uchun sxema va diagrammalar ishlatiladi. Ko'rgazmali grafika foydalanuvchilar va dasturchilarga ishning mohiyatini va uni bajarish ketma-ketligini osongina tushinishiga yordam beradi. Dastlab modellarni ifoda etish uchun sxema va diagrammalar qo'lda, qog'ozda bajarilar, ya'ni avtomatlashtirilmagan edi, ushbu

jarayonni avtomatlashtirish, axborot tizimlarini yaratish va joriy etish maqsadi CASE-vositalarini yaratilishiga asosiy sabab bo'ldi.

CASE-vositalari tizimli dastur ta'minoti va texnik vositalar bilan birga axborot tizimlarini ishlab chiqishning to'liq muhitini tashkil etadi.

CASE-vositalarining asosiy maqsadi, axborot tizimlarini ishlab chiqish bosqichlarida loyihalash bilan kodlashni alohida ajratishdir. CASE-texnologiyalarini ishlatish bilan axborot tizimlari dasturiy ta'minoti hayotiy davrining barcha bosqichlari, xususan tahlil etish va loyihalash bosqichlari ko'proq o'zgaradi. Ko'pgina mavjud CASE-vositalari strukturali yoki ob'ktga mo'ljallangan tahlil va loyihalash metodologiyasiga asoslangan. Ularda tizimning o'zgarishi, dasturiy vositalarning arxitekturasini, tizim modellarining o'zaro aloqasi va tashqi talablar diagramma yoki tekst ko'rinishida ifodalanadi.

Bunday metodologiyalar loyihlashtirilayotgan tizimni mustahkam va ko'rgazmali ifodalanishini ta'minlaydi, ya'ni uning umumiy tahlilidan boshlab keyinchalik ko'p sonli darajada iyerarxik struktura ko'rinishida detallashtiradi.

CASE-texnologiyalar barcha turdagi axborot tizimlarini qurishda qo'llaniladi, lekin quyidagi sohalarida ular mustahkam o'rin egallaydi:

- savdo-sotiq ishlari bo'yicha axborot tizimlarini ishlab chiqish, ularda CASE faqat axborot tizimini ishlab chiqishda emas, balki strategik rejalashtirish, moliyani boshqarish, firma siyosatchilarini aniqlash, personalni o'qitish masalalarini echishga yordam beruvchi tizim modelini yaratishda ham qo'llaniladi (bu yo'nalish biznes-tahlil degan nomga ega bo'ldi);

- tizimli va boshqaruvchi axborot tizimlarini ishlab chiqish. CASE-texnologiyalar asosan ish unumdorligini oshirish maqsadida ushbu yo'nalishga faol joriy qilinadi.

Strukturali metodologiyani avtomatlashtirish, tizimli va dasturli injeneriyaga zamonaviy usullarni qo'llash borasida CASE-vositalar quyidagi asosiy ustunliklarga ega:

- avtomatik nazorat, avvalambor loyihani nazorat qilish vositasi evaziga yaratilayotgan axborot tizimi sifatini yaxshilaydi;
- kutilayotgan natijani barvaqt baholashga uchun bo'lajak tizimning prototipini qisqa muddatda yaratishga imkon beradi;
- loyihalash va ishlab chiqish jarayonini tezlashtiradi;
- dasturchini qiyin ishdan ozod etadi, uning dastur tuzish borasidagi ijodiy faoliyatini yaxshilashga vaqt va imkon yaratadi;
- ishning rivojlanishini qo'llab quvvatlaydi;
- ish jarayonida komponentlardan takror foydalanish imkoniyatini yaratadi.

Shuni tushinish zarurki, ushbu vositalar asoslangan bazaviy texnologiyani tushunmasdan turib, CASE-vositalarini muvoffaqiyatli qo'llab bo'lmaydi. O'z-o'zidan dasturiy CASE-vositalar axborot tizimlarini loyihalash va joriy etish jarayonlarini avtomatlashtiruvchi vositalar hisoblanadi. Axborot tizimlarining metodologiyasini tushunmasdan turib, CASE-vositalarini qo'llab bo'lmaydi.

Axborot tizimlarining dasturiy ta'minotini ishlab chiquvchi ko'pgina dasturchilar CASE-texnologiyalariga murojaat qilishadi. Ilg'or texnologiyalar ko'zdan kechirilganida 1000 dan ortiq firmalarning anketa savol javobi natijalariga ko'ra CASE-texnologiyalar bir maromda ishlovchi axborot texnologiyalari qatoriga tushdi. So'rov o'tkazilganlardan yarmi o'zining uchdan bir loyihasida CASE-texnologiyalarini qo'llab, 85 foizi yutuqli natijalarga erishdi. Lekin CASE-vositalarining barcha potentsial imkoniyatlariga qaramasdan, joriy etishda omadsizlikka uchragan vaqtlari ham bo'lgan, unga asosiy sabab qilib quyidagilarni ko'rsatish mumkin:

- CASE-vositalar tez orada o'zining effektini bermasligi mumkin, oradan qanchadir vaqt o'tishi talab etiladi;

- CASE-vositalarini yaratishga nisbatan uni joriy etish uchun real sarf xarajatlar oshib ketadi;
- CASE-vositalarni joriy etish jarayoni muvoffaqiyatli yakunlanganidan so'nggina, foyda olish imkoniyati ta'minlanadi.

CASE-vositalarining muvoffaqiyatli joriy etilishi jarayondagi vosita turi, loyiha xarakteristikalar, foydalanuvchining ishlash tajribasiga bog'liq. Ba'zi analitiklarning fikricha CASE-vositalari bir yoki ikki yillik tajribadan o'tganidan so'ng yaxshi samara beradi. Boshqalarning fikricha, axborot tizimi hayotiy davri tadbiq etilishi mobaynida turli ta'sirlar o'zini nomoyon etishi mumkin, ya'ni tizimning texnologik yaxshilanishiga sarflanadigan harajatlar uni tadbiq etish harajatlarini pasayishiga sabab bo'ladi.

CASE-vositalarini muvoffaqiyatli joriy etish kaliti tashkilot tayyorgarligiga bog'liq bo'lib quyidagilarni o'z ichiga oladi:

- Texnologiya. Mavjud imkoniyatlarning chegaralanganligini, eskirganligini tushinish va yangi texnologiyalarni qabul qilish qobiliyati;
- Madaniyat. Yangi jarayonlarni joriy etishga tayyorligi va dasturchilar bilan foydalanuvchilarning o'zaro hamjihatlikda bo'lishi;
- Boshqaruv. Muhimroq bosqichlarga va joriy etish jarayoniga nisbatan rahbarlik va tashkilotchilik.

CASE-vositalaridan foydalanmoqchi bo'lganlar tizimni tadbiq etish uchun tezkor sarf xarajatlarga tayyor bo'lishi, vositaning eskirishi bilan yangi versiyani o'rnatishi, yangi personalni o'qitishi va mavjud personallarning kvalifikatsiyasini oshirib borishi talab etiladi.

CASE-vositalari muvoffaqiyatli joriy etilganida quyidagi foydalar ta'minlanadi:

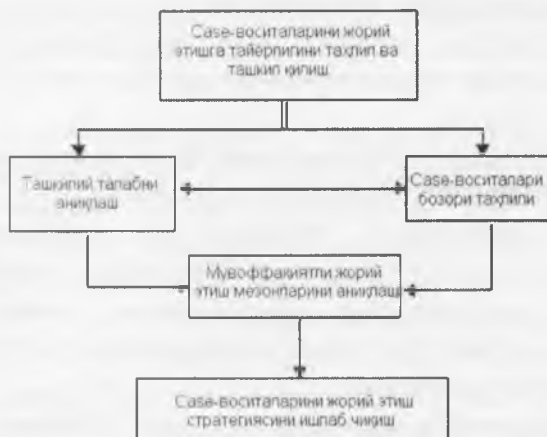
- dasturiy ta'minotni ishlab chiqish va kuzatish jarayonini texnologik qo'llab quvvatlanishi;

- ishlab chiqarish, mahsulotning sifati, standartga mosligi, talab darajasida hujjatlashtrilishi;
- axborot texnologiyalariga bog'liq bo'lgan faoliyatga nisbatan diqqatni oshirilishi;
- foydalanuvchilar va dasturchilar o'rtasidagi aloqani yaxshilanishi.

6.3.2. CASE-vositalarini o'zlashtirish va joriy etish texnologiyasi

CASE-vositalarini o'zlashtirish va eoriy etishning zamonaviy texnologiyalari asosan IEEE standarti bazasiga tayanadi. CASE-vositalarini joriy etish jarayoni quyidagi bosqichlardan tashkil topgan:

- CASE-vositalariga bo'lgan talabni aniqlash;
- CASE-vositalarini baholash va tanlash;
- mustahkam loyihani ishlab chiqish;
- CASE-vositalarini amaliy joriy etish.



6.3.1-rasm. CASE-vositalariga bo'lgan talablar.

Yirik axborot tizimlari loyihasini tuzishda CASE-vositalarini tanlash quyidagi mezonlarga bog'liq holda amalga oshiriladi:

1. Axborot tizimi hayotiy davri ciklini va uning rivojlanishini ta'minlash;

2. Loyihani yaxlitligini ta'minlash va uning holatini nazorat qilib borish;

3. Apparat-dasturiy platforma va MBBTlarga bog'liq bo'lmasligi;

4. Ochiq arxitektura;

5. O'zbekiston texnik qo'llab-quvvatlash sifati va narxi, muvoffaqiyatli ishlatish tajribasi;

6. O'zlashtirish va ishlatishning soddaligi.

Loyihani tahlil etish natijasida aynan biror bir vosita uni to'liq tuzish uchun etarli bo'lmasligi va bir nechta CASE-vositalarini bir vaqtda ishlatish zarur bo'lishi ham mumkinligini aytish kerak.

Xulosa qilib aytganda, CASE-vositalarini qo'llashdan oldin aynan qaysi biri yoki qaysilari unga mos kelishini aniqlash, u o'z samarasini berishiga ishonch hosil qilish zarur, aks holda juda katta iqtisodiy zarar etishi mumkin [3].

6.3.3. Zamonaviy CASE-vositalar xarakteristikalar

Zamonaviy CASE-vositalari oddiy tahlil va hujjatlashtirishdan tortib dasturiy ta'minot hayotiy davrini to'liq qamrab olgan to'liq masshtabli avtomatlashtirish vositasiga qadar ko'p sonli axborot tizimlarini loyihalash texnologiyasini o'z ichiga oladi.

CASE-vositalar qatorida imkoniyatlari cheklangan arzon tizimlar va operatsion muhit, platfomalarga bog'liq bo'lgan juda qimmat tizimlar ham uchraydi. Zamonaviy dastur vositalari bozorida 300 dan ortiq turli CASE-vositalari sanab o'tilgan, ulardan eng kuchlilarini ilg'or firmalar ishlatmoqda.

Dasturiy ta'minot hayotiy davrini ta'minlovchi kompleks CASE-vositalar tarkibiga quyidagi komponentlar kiradi:

- CASE-vositalarining asosi - repozitori. U loyiha versiyasi va uning komponentlarini saqlaydi hamda turli dasturchilardan kelayotgan xabarlarini sinxronizatsiyalaydi;

- tahlil qilish va loyihalashning grafik vositalari – axborot tizimlari modelini qurishda iyerarxik bog'langan diagrammalarni yaratish va tahrirlash uchun xizmat qiladi;

- 4 GL oilasiga mansub bo'lgan dastur vositalari;
- konfiguratsiyali boshqarish vositasi;
- hujjatlashtirish vositasi;
- testlash vositasi;
- loyihani boshqarish vositasi;
- reinjining vositasi (dastur kodi orqali o'chirilgan asosiy loyihani modelini tiklash imkoni).

Barcha zamonaviy CASE-vositalar asosan turi va kategoriyasi bo'yicha klassifikatsiyalanadi. Turi bo'yicha klassifikatsiyalash deganda dasturiy ta'minot hayotiy davrining u yoki bu jarayoniga CASE-vositalarini funksional mo'ljallanishi tushuniladi.

Turi bo'yicha klassifikatsiyalash asosan CASE-vositalarining komponentlari bilan mos keladi va quyidagi asosiy turlarga ajratiladi:

- tahlil qilish vositasi (Upper Case) – predmet sohasi bo'yicha modellarni qurish va tahlil etishga mo'ljallangan. Ularga Design/IDEF, BPwin, CASE-vositalari misol bo'la oladi;

- tahlil qilish va loyihalash vositasi (Middle Case) – kengroq tarqalgan loyihalash metodologiyasini qo'llash bilan birga sohaga moslashtirilgan loyihalarni yaratish uchun ishlatiladi. Ular yordamida komponentlar, tizim interfeysi, tizim arxitekturasi, ma'lumotlar algoritmi va strukturasi ishlab chiqiladi. Vantage Team Builder, Desiner/2000, Silverrun, Pro IV, Case-Analitik CASE-vositalari shular jumlasidandir;

- ma'lumotlar bazasini loyihalash vositalari – keng tarqalgan MBBTlari uchun ma'lumotlarni modellashtirish va MB

sxemalarini generatsiyasini (SQL tilida) ta'minlaydi. Ularga Erwin, S-Designor, DataBase Designer (Oracle) CASE-vositalarini misol keltirishimiz mumkin, MBni loyihalash vositalari Vantage Team Builder, Designer/2000, Silverrun va Pro-IV vositalari tarkibida ham mavjud.

- dastur ilovalarini ishlab chiqish vositalari – unga 4 GL vositasi oilasiga mansub Uniface, Jam, Power Builder, Developer/2000, NewEra, SQL Windows, Delphi va boshqa vositalar hamda Vantage Team Builder, Pro-IV, Silverrun tarkibiga kiradigan kod generatorlari kiradi;

- reinjiniring vositasi – dastur kodlari va MB sxemalarini tahlil etadi hamda turli modellar va loyihalar asosini shakllantiradi. MB sxemalarini tahlil etish va shakllantirish vositasi Vantage Team Builder, Pro-IV, Silverrun, Designer/2000, Erwin va S-Designor tarkibida mavjud. Dastur kodlarini tahlil qilish borasida ob'ektga mo'ljallangan CASE-vositalari keng tarqalgan, ular S++ tilida reinjiniring amalini bajaradi. Rotional Rose, Object Team Builder ob'ektga yo'naltirilgan CASE-vositalari shular jumlasidandir.

Kategoriya bo'yicha klassifikatsiyalash deganda bajaradigan vazifasi bo'yicha tadbiq etilishini darajasi tushuniladi va ko'proq avtonom masalalarni echadigan alohida lokal vositalarni o'z ichiga oladi. Ushbu CASE-vositalarini quyidagi belgilar yordamida klassifikatsiyalash mumkin:

- qo'llaniladigan metodologiya, modellari va ma'lumotlar bazasi;

- MBBT bilan integratsiya darajasi;

- ruxsat etilgan platformasi yoki operatsion tizimi.

Hozirgi vaqtda jahon bozorida quyidagi CASE-vositalariga bo'lgan talab oshib bormoqda:

Lokal CASE-vositalar:

- Vantage Team Builder;

- Designer/2000;

- Silverrun;
- ERwin+BPwin;
- S-Designor;
- Case Analitik;

Ob'ektga mo'ljallangan CASE-vosita: Rotional Rose.

Ushbu lokal hamda ob'ektga mo'ljallangan CASE-vositalarining har biriga qisqacha to'xtalib o'tamiz.

6.3.4. Lokal CASE-vositlari

Vantage Team Builder integrallashgan dasturiy mahsuloti - dasturiy ta'minot hayotiy davrining kaskadli modelini tadbiq etish va to'liq qo'llab-quvvatlash uchun mo'ljallangan. Vantage Team Builder quyidagi vazifalarni bajaradi:

- ma'lumotlar diagrammasini, "mavjudlik-alohadorlik" tizimini, ma'lumotlar strukturasi, dasturlar sxemasi va ekran formalari ketma-ketligini loyihalash;
 - tizim arxitekturasini diagrammasini loyihalash;
 - 4 GL tilida dastur kodini generatsiyalash, MB jadvallarini yaratish uchun SQL kodini generatsiyalash;
 - C tilida dasturlash;
 - loyiha konfiguratsiyasi va versiyasini boshqarish;
 - standart shablonlar yordamida hujjatlarni loyihalash va generatsiya qilish;
 - CDIF formatidagi loyiha ma'lumotlarini import va eksport qilish.
 - Vantage Team Builder asosan barcha UNIX platformalarida (Colaris, SCO UNIX, AIX, HP-UX) ishlatiladi.
- Designer/2000 - sohalar bo'yicha turli modellar va diagrammalarni ishlab chiqishning grafik interfeysini ta'minlaydi. Uning tarkibiga quyidagi komponentlar kiradi:
- Repository Administrator – repozitorni boshqarish vositasi. Unda dasturlarni yaratish va o'chirish, turli foydalanuvchilar tomonidan ma'lumotlarga ruxsat etilishini

boshqarish, ma'lumotlarni impor va eksport qilish imkoni mavjud;

- Repository Object Navigator – repozitor va uning barcha elementlariga ruxsat etish vositasi.

- Process Modeller – biznes jarayonlar reinjiningi konsepciyasi asosida faoliyatni tahlil etish va modellashirish vositasi;

- Systems Modeller – axborot tizimlarining axborot modellari diagrammalarini qurish vositalarini o'z ichiga oladi jumladan, "mavjudlik-aloadorlik", funksional iyerarxik, ma'lumotlar oqimi diagrammalari;

- Systems Designer – axborot tizimlarini loyihalash vositalari to'plami. Unda relyatsion ma'lumotlar bazasi strukturasini qurish vositasi, ma'lumotlar o'rtasidagi aloqa diagrammalari, PL/SQL tilidagi dastur strukturasini ko'rsatuvchi diagrammalar mavjud;

- Server Generator – Oracle MBBT ob'ektlarini (jadval, indekslar, kalitlar va boshqalar) tasvirlash generatori. Oracle mahsulotlarida ma'lumotlar bazasi generatsiyasi Informix, DB/2, MS SQL Server, SysBase MBBTlarida amalga oshirilish mumkin.

- Forms Generator (Oracle formalari uchun dastur generatori) – generatsiya qilinadigan dasturlar ekran formalari, ma'lumotlarni nazorat qilish vositasi formalari, yaxlit bir butunlikni tekshirish va avtomatik javob ko'rsatish formalari;

- Repository Reports – standart hisobotlar generatori. U axborot strukturasini o'zgartirish, o'zbek tiliga o'girish kabi imkoniyatlarni beradi.

Designer/2000 mahsulotlarini API (Application Programming Interface) interfeysini ishlatib boshqa vositalarga moslashtirish yoki integratsiyalash mumkin. Dastur ilovalarini Oracle mahsulotlaridan tashqari Visual Basic dasturiga ham generatsiya qilish imkoni mavjud.

Designer/2000 vositasini Windows va Windows NT operatsion tizimlarida ishlatish mumkin.

Silverrun - Computer Systems Advisers (CSA) firmasining mahsuloti bo'lib, dastur ta'minoti hayotiy davrining spiral modeliga mo'ljallangan bo'lib, biznes-klass axborot tizimlarini loyihalash va tahlil etish uchun ishlatiladi. Ixtiyoriy metodologiyani qo'llab turli axborot modellarini qurishda ishlatilishi mumkin. Silverrun modulli strukturaga ega bo'lib, har biri mustaqil mahsulot hisoblanadigan to'rtta moduldan tuzilgan.

1. Biznes jarayonlarni qurish moduli – bunda ma'lumotlar diagramma ko'rinishida bo'lib, tashkilot yoki yaratilayotgan axborot tizimlarini modellashtirishni amalga oshiradi.

2. Ma'lumotlarni konseptual modellashtirish moduli - "mavjudlik-alohadorlik" ma'lumotlari modelini qurishni ta'minlaydi.

3. Relyaion modellashtirish moduli – relyatsion ma'lumotlar bazasida "mavjudlik-alohadorlik" modelini joriy etish uchun xizmat qiladi.

4. Ishchi guruh menedjeri – barcha axborot modellarini saqlash uchun ma'lumotlar lug'ati sifatida qo'llaniladi.

Silverrun vositasini Windows, Macintosh va OS/2 platformalarda ishlatish mumkin.

ERwin – IDEF1X metodologiyasini asosida ma'lumotlar bazasini konseptual modellashtirish vositasi. ERwin ma'lumotlar bazasi sxemalarini loyihalashni amalga oshirib, uning yozilishini MBBTlarga generatsiyalaydi va reinjiniring vazifasini bajaradi. Erwin 4 GL oilasiga mansub dasturlash tillarida dastur ilovalarini ishlab chiqish imkoniyatiga egadir. ERwin/OPEN versiyasi Power Builder va SQL Windows dastur vositalariga to'liq mos keladi va loyihalashtirilgan MBni eksport qilish qobiliyatiga ega.

ERwin Model Mart tarmoq uchun mo'ljallangan versiyasi MB va dastur ilovalarini ishchi guruhlarda kelishilgan holda loyihalash imkonini yaratadi.

BPwin – IDEF0 metodologiyasini tadbiq etuvchi funksional modellashtirish vositasidir.

S-Designor – relyatsion MBni loyihalovchi vosita bo'lib, vazifasiga ko'ra ERwin vositasiga o'xshab ketadi, faqat unda diagrammalarning ishlatilishi farq qiladi. S-Designor ma'lumotlarni modellashtirishning standart metodologiyasini qo'llaydi va ma'lumotlar bazasini Oracle, Informix, DB/2, MS SQL Server, SysBase MBBTlarga generatsiyalash imkoniyatini beradi.

CASE-Analitik – hozirgi kunda eng raqobatbardosh Case-vositalaridan biri bo'lib, funksional modellashtirish va ma'lumotlarni diagrammalarda qurishni amalga oshiradi. Uning asosiy vazifalariga:

- DFD (Data Flow Diagram – ma'lumotlar oqimi diagrammalari)ni qurish va tahrirlash;
- diagrammalar va loyihaning to'liqligini tahlil etish;
- loyiha bo'yicha turli hisobotlarni olish;
- standart talablarga mos holda hujjatlar maketini generatsiya qilish.

CASE-Analitik dastur vositasi ERwin vositasi bilan ma'lumot almashinish imkoniga ega va Windows tizimida ishlashga mo'ljallangan [3]

6.3.5. Obyektga mo'ljallangan CASE-vositasi

Rotional Rose – Rotional Software Corporation (AQSH) firmasining Case-vositasi bo'lib, dasturiy ta'minotni loyihalash va avtomatlashtirish bosqichlarini tahlil etishga mo'ljallangan. Shu bilan birga u loyiha hujjatlari va turli tillardagi kodlarni generatsiyalash imkoniga ham ega. Rotional Rose ob'ektga mo'ljallangan tahlil va loyihalash metodologiyasini qo'llaydi. Ushbu metodologiya UML (Unified Modeling Language) tizimida ham qo'llanilgan bo'lib, u ob'ektlarni modellashtirishning universal dastur vositasi hisoblanadi.

Hozirda ushbu dastur vositasi ob'ektga mo'ljallangan tahlil va loyihalash borasida standart dasturga aylanib bormoqda.

Rotional Rose vositasining aniq varianti dastur kodlarini generatsiya qiladigan tillar (C++, Smalltalk, Power Builder, Ada, SQL Windows) orqali aniqlanadi. Rotional Rose/S++ versiyasi loyiha hujjatlarini diagramma ko'rinishida ishlab chiqishni va dastur kodlarini S++ da generatsiya qilishni amalga oshiradi.

Rotional Rose yangi loyihalar yaratishda dastur komponentlarini takror ishlatish imkonini beradi va reinjining vositasini qo'llab quvvatlaydi. Rotional Rose asosida modelning mantiqiy va fizik tuzilishini aniqlovchi turli diagrammalarni qurish yotadi. Ular qatoriga sinflar, holatlar, ssenariylar, modullar va jarayonlar diagrammalari kiradi.

Rotional Rose tarkibida 6 ta asosiy komponent yotadi:

1. Repoizitor.
2. Foydalanuvchining grafik interfeysi.
3. Loyihani ko'rish vositasi (Browser).
4. Loyihani nazorat qilish vositasi.
5. Statik ma'lumotlarni yig'ish va hujjatlarni generatsiya qilish vositalari.
6. Kod generatori va C++ uchun analizator, ular yordamida reinjining (dastur kodi bo'yicha loyiha modelini tiklash) ta'minlanadi.

Rotional Rose vositasini Windows, Unix, Solaris, SunOS, HP UX operatsion tizimlarida qo'llash mumkin [3]

6.3.6. Konfiguratsiyali boshqaruv vositasi

Konfiguratsiyali boshqaruv maqsadi – axborot tizimlarini ishlab chiqish va ishlashini kuzatib borish jarayonini nazorat qilish va boshqarishni ta'minlashdir. Buning uchun axborot tizimi holati, uning komponentlari va o'zgarishlar haqidagi aniq ma'lumot doimiy vaqt oralig'ida zarur bo'ladi.

Konfiguratsiyali boshqaruv masalasini yechish uchun komponent holati identifikatsiyasini ta'minlovchi quyidagi usul va vositalar qo'llaniladi:

- barcha komponentlarni ro'yxatga olish va tizimni to'liq modifikatsiya qilish;
- komponentlarga kiritilayotgan o'zgarishlarni nazorat qilish;
- tizim tuzilishi va vazifasini o'rganish va nazorat qilish;
- tizim xarakteristikasini yaxshilash va bajaradigan vazifalarini rivojlantirishni boshqarish.

Eng keng tarqalgan konfiguratsiyali boshqaruv vositasiga misol qilib AQShning Intersolv firmasi mahsuloti sanaladigan PVCS vositasi va uning tarkibidagi bir nechta mustaqil dastur mahsulotlarini: PVCS Version Manager, PVCS Tracker, PVCS Configuration Builder va PVCS Notifylarni aytishimiz mumkin.

6.3.7. Hujjatlashtirish vositasi

Axborot tizimlarini ishlab chiqish jarayonida hujjatlarni yaratish uchun turli vositalar ya'ni, hisobotlarni shakllantirish vositalari va nashriyot tizimi komponentlari ishlatiladi.

Odatda hujjatlashtirish vositasi CASE-vositalarning tarkibida mavjud bo'ladi. Lekin, ba'zi paketlar hujjatlashtirishda qo'shimcha servis xizmatlarini taqdim etishgani uchun ularni ishlatish zaruriyati tug'iladi. Hozirgi kunda bunday paketlardan SoDA (Software Document Automation) hujjatlashtirish vositasi keng doirada qo'llanilmoqda.

SoDA mahsuloti axborot tizimlari hayotiy davrining barcha jabhalarida hujjat loyihasini ishlab chiqishni avtomatlashtirish uchun mo'ljallangan. U loyihani ishlab chiqish jarayonida hosil bo'ladigan turli axborotlarni avtomatik tarzda tortib olib, chiqarilayotgan hujjatga qo'shish orqali hujjatni tayyorlash imkonini beradi.

Shu bilan birga loyiha bilan hujjatlar mosligi va hujjatlarning o'zaro aloqasi nazorat qilinadi hamda ularning o'z vaqtida yangilanishi ta'minlanadi. Natijaviy hujjat avtomatik tarzda ko'plab manbalar orqali shakllantiriladi.

Paketda hujjat shablonlarini tayyorlash uchun grafik redaktor mavjud bo'lib, u kerakli stil, fon, shriftni o'rnatish, sarlavhalarni joylashuvini aniqlash, axborotlarni turli manbalardan olib joylashtirish uchun zahira joyini belgilash kabi imkoniyatlarni beradi.

SoDA FrameBuilder nashriyot bazasida shakllangan bo'lib, hujjatlashtirish va tahrirlashning to'liq kompleks vositalarini o'z ichiga oladi.

SoDA tizimi ishining oxirgi natijasi bu tayyor hujjat yoki kitob hisoblanadi. Hujjat generatsiya qilinganidan so'ng, u SoDA (Frame Builder) formatidagi fayl sifatida xotiraga saqlanadi. Hujjat yoki hujjat qismini chop etishni faqat SoDA tizimi orqali amalga oshirish mumkin.

6.3.8. Testlash vositasi

Testlash deganda dasturni bajarilish jarayonida undagi xatolikni topish tushuniladi. Regressiyali testlash - bu dasturga o'zgartirishlar kiritilganidan yoki takomillashtirilganidan keyingi bo'ladigan testlashdir.

Rivojlangan testlash vositalaridan biri QA (Quality Works) – integrallashgan va ko'p platformali muhitga ega bo'lib, regressiyali testlar hamda ixtiyoriy darajadagi avtomatlashtirilgan testlarni ishlab chiqishga mo'ljallangan.

QA dasturiy ta'minot hayotiy davrining ixtiyoriy pallasida testlashni boshlashga ruxsat beradi, bundan tashqari testlash jarayonini rejalashtirish va boshqarish, ilovalardagi o'zgarishlarni ko'rsatish va testni takroran ishlatish imkonini ham beradi.

6.3.9. CASE-vositalarining taqqosiy xarakteristikalari

Shuni yana bir marotaba ta'kidlash lozimki, CASE-vositalari alohida o'rganilganida ularning hech biri mustaqil ravishda axborot tizimini yaratish va joriy etish muammolarini to'liq yecha olmaydi. Faqatgina dasturiy ta'minot hayotiy davrini to'liq qo'llab quvvatlovchi komplekslashgan instrumental vositalarni taqqoslashimiz mumkin. Ushbu vositalar ichidan zaruriylarini birgalikda qo'llagan holda yutuqli natijalarga erishishimiz mumkin. Hozirgi vaqtda axborot tizimlarini yaratishda "Argussoft Compani" firmasining komplekslashgan instrumental vositalarini qo'llash yaxshi samara bermoqda, uning tarkibiga quyidagilar kiradi:

- Silverrun CASE-vositasi;
- JAM ilovalarni ishlab chiqish vositasi;
- QA testlash vositasi;
- Tuxedo tranzaksiya menedjeri;
- SE Companion loyihani rejalashtirish va boshqarishning kompleks vositasi;
- PVCS konfiguratsiyali boshqarishning kompleks vositasi;
- Rotional Rose ob'ektga mo'ljallangan CASE-vositasi;
- SoDA hujjatlashtirish vositasi.

Boshqa kompleks vositalarga misol qilib quyidagilarni keltiramiz:

"DataX/Florin" va "LANIT" firmalarining Vantage Team Builder for Uniface+Uniface vositasi;

"FORS" firmasining kompleks vositalari hisoblanadigan:

- o Designer/2000, ERwin, BPwin Case-vositalari;
- o Developer/2000, OraclePowerObjects ilovalarni ishlab chiqish vositalari;
- o Explain SQL sozlash va optimizatsiyalash vositasi;
- o SQL Watch, DB Vision, SQL Spy administratsiyalash vositalari;
- o OracleBook hujjatlashtirish vositasi.

"CENTURA" firmasining kompleks vositalari:

- ERwin, BPwin Case-vositalari;
- SQL Windows va Team Windows ilovalarni ishlab chiqish vositalari;
- SQL Bench “klien-server” ilovalarini optimizatsiya qilish va testlash vositasi;
- Quest va Crystal Reports tadbiq etish, kuzatish va hisobotlar tayyorlash vositalari.

Barcha sanab o'tilgan kompleks dasturiy vositalar dasturiy ta'minot hayotiy davrini to'liq qo'llab quvvatlaydi va muammolarni hal etadi. Boshqa mezonlarga e'tibor beradigan bo'lsak, bu yerda quyidagilarni alohida ajratib ko'rsatishimiz mumkin:

1. Loyihaning butunligini ta'minlash va uning holatini nazorat qilish bo'yicha – “FORS” firmasining kompleks vositalari va Vantage Team Builder for Uniface+Uniface vositasi qo'llanilsa maqsadga muvofiq bo'ladi.

2. Platforma, operatsion tizim va MBBTga bog'liq bo'lmasligi darajasi bo'yicha – “Argussoft Kompani” firmasining kompleks vositalari, ular aniq bir konkret platformaga mo'ljallanmagan, ixtiyoriy platformada ishlash imkoniga ega. Bundan tashqari Vantage Team Builder for Uniface+Uniface vositasi ham alohida platforma talab etmaydi.

3. Ochiq arxitekturaga egaligi bo'yicha - “Argussoft Kompani” firmasining kompleks vositalari va Vantage Team Builder for Uniface+Uniface vositasi ustun turadi.

4. Texnik qo'llab-quvvatlash sifati bo'yicha - “Argussoft Kompani” firmasining kompleks vositalarini aytish mumkin. Chunki, hozirgi kunda ushbu vositani texnik va metodik jihatdan o'qitish tizimi yaxshi yo'lga qo'yilgan.

5. O'zlashtirish va ishlatishning oddiyligi va qulayligi bo'yicha - “Argussoft Kompani” va “CENTURA” firmalarining kompleks vositalari oldingi o'rinlarda turadi.

Yuqorida keltirilgan taqqosiy xarakteristikalaridan xulosa qilib quyidagilarni aytishimiz mumkin:

- dasturiy ta'minot hayotiy davrining spiral modeliga

mo'ljallangan yirik loyihalarda - "Argussoft Kompani" firmasining kompleks vositalarini qo'llash tavsiya etiladi;

- dasturiy ta'minot hayotiy davrining koskadli modeliga mo'ljallangan yirik loyihalarda - Vantage Team Builder for Uniface+Uniface vositasi qo'llash zarur;

- Oracle MBBTni ishlatishga mo'ljallangan yirik loyihalarda - "FORS" firmasining kompleks vositalarini qo'llash mumkin;

- o'rta hajmli loyihalarda - "Argussoft Kompani" firmasining kompleks vositalari va Power Builder, Delphi, SQL Windows dastur ilovalarini ishlab chiquvchi vositalar bilan birga ERwin, BPwin, CASE Analitik, S-Designor, CASE-vositalarini qo'llash tavsiya etiladi;

Ob'ektga mo'ljallangan usulda loyiha tuzishda - Rotional Rose ob'ektga mo'ljallangan CASE-vositasi va dastur ilovalarini ishlab chiqishda unga mos uchta vositadan birini tanlash zarur.

Nazorat savollari

1. CASE – vositalarining klassifikatsiyasini keltiring.
2. CASE so'zi ingliz tilida nimani anglatadi?
3. CASE-vositalar qanday sohalarda qo'llaniladi?
4. CASE-vositarni joriy etish texnologiyasini izohlang.
5. CASE-trxnologiyalarga qo'yilgan talablarni sanab o'ting.
6. Zamonaviy CASE-vositalarining xarakteristikalarini belgilang.
7. Local CASE-vositalar haqida ma'lumot bering.
8. Ob'yektga mo'ljallangan CASE-vositalarining xarakteristikalarini izjhlab bering.
9. CASE-vositalarining konfiguratsiyasini boshqarishda qanday usullar qo'llaniladi?
10. CASE-texnologiyalarida testlash vositalari haqida ma'lumot bering.

VII. MA'LUMOTLAR BAZASINI BOSHQARISH TIZIMLARI

7.1. Zamonaviy MBBT lari va ularning arxitekturalari

Zamonaviy MBBT va uning vazifasi va imkoniyatlari haqida oldingi mavzularda tavsif berib o'tilgan edi. Ushbu tavsifga qo'shimcha quyidagilarni keltiramiz.

MBBTda har bir MB modeli quyidagi xususiyatlari bo'yicha tavsiflanadi:

- ma'lumotlar tuzilmalarining turi;
- ma'lumotlar ustida bajariladigan amallar;
- butunlikning cheklanganligi.

MBBTning asosiy xususiyati faqatgina qiymatlarni kiritish va saqlashni amalga oshiruvchi vositalar bo'lmasdan, uning tuzilmasini ham berilishidir. MBBT boshqaruvi ostidagi qiymatlar va ular tavsifi saqlanadigan fayllarni MB deb atash qabul qilingan.

Biror bir dastur uchun maxsus tashkil qilingan fayldagi ma'lumotlarga ishlov berishga nisbatan MBBT orqali ma'lumotlarga ishlov berish bir oz ko'proq vaqtni oladi. Hozirgi kunda kompyuterlarning texnik imkoniyatlari yuqori bo'lganligi uchun bu vaqt sezilmaydi. Ammo, MBga biror o'zgarish kiritilsa u dasturiy mahsulotlarga ta'sir qilmaydi. MBBT har bir foydalanuvchiga qiymatlarga kirish imkoniyatini yaratib beradi. Foydalanuvchi MBBTning quyidagi xususiyatlari haqida tushunchaga ega bo'lmasligi mumkin:

- qiymatlarning xotirada jismoniy joylashtirilishi va ularning tasvirlanishi;
- so'ralgan ma'lumotlarni izlash mexanizmi;
- bir vaqtning o'zida bir xil ma'lumotlarni bir necha foydalanuvchi so'rovida tasvirlanishi va o'zgartirishda vujudga keladigan muammolar echimi;

- ruxsat berilmagan kirishlar va nomutanosib o'zgartirishlarda himoyalash usullari;

- MBni ishchi holatda doimo saqlash va MBBTning boshqa ko'pgina xizmatlari.

Oldingi mavzularda ta'kidlab o'tilganidek, MB tushunchasi avvalo ko'p masalalarni bir nechta foydalanuvchilar tomonidan echishga qaratilgan. Shu sababli, zamonaviy MBBTlarida ko'p foydalanuvchili texnologiyaning mavjudligi muhim hisoblanadi. Turli vaqtlarda bunday texnologiyalarni ro'yobga chiqarish hisoblash texnikasining asosiy xarakteristikalari va dasturiy ta'minotning rivojlanishi bilan bog'liq bo'lgan. Ushbu texnologiyalarni xronologik ketma – ketlikda ko'rib chiqamiz.

Markazlashgan arxitektura. MBning ushbu texnologiyasida MB, MBBT va dasturiy ta'minot (ilovalar) bitta kompyuterda (meynfreydm yoki personal kompyuterda, 7.1.1-rasm) joylashgan bo'ladi. Bu usul uchun tarmoqni qo'llab – quvvatlash zarur emas va barcha vazifalar avtonom shaklda bajariladi. Ishlash tartibi va tuzilishi quyidagicha:

- MB fayllar to'plami shaklida kompyuterning qattiq diskida joylashadi;

- shu kompyuterning o'zida MBBT va MB bilan ishlovchi dastur o'rnatiladi;

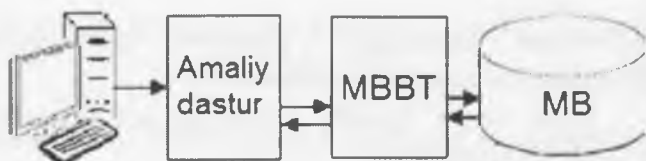
- foydalanuvchi dasturni ishga tushiradi va dastur interfeysi yordamida ma'lumotlarni o'qish/yangilash uchun MB ga murojaat etadi;

- MBga murojaat MBBT orqali bajariladi va MBBT da MBning jismoniy tuzilmasi haqidagi barcha ma'lumotlar saqlanadi;

- MBBT foydalanuvchi so'rovlarini bajarish uchun ma'lumotlarga murojaatni ta'minlab beradi;

- MBBT natijalarni ilovaga beradi;

- ilova esa o'zida mavjud foydalanuvchi interfeysi yordamida so'rov natijasini tasvirlab beradi.



7.1.1-rasm. Markazlashgan arxitektura

Bunday arxitektura DB2, Oracle, Ingres MBBTlarining birinchi versiyalarida qo'llanilgan.

Ko'p foydalanuvchili ishlash texnologiyasi yoki mul'tiprogrammalar rejimi (protssessor va tashqi qurilma baravar ishlaydi - masalan, biror foydalanuvchining amaliy dasturi tashqi xotiradan ma'lumotni o'qiyotgan vaqtda boshqa foydalanuvchi dasturi protssessor tomonidan bajariladi) vaqtini taqsimlash rejimi yordamida (foydalanuvchi dasturlari bajarilishi uchun vaqt kvantlari taqsimlab beriladi) amalga oshiriladi. Bunday texnologiya katta EHMLar (IBM-370, ES-1045, ES-1060) davrida keng qo'llanilgan. Bu texnologiyaning asosiy kamchiligi shundaki foydalanuvchilar soni oshsa, unumdorlik keskin pasayadi.

"fayl-server" arxitekturasi. Qo'yilgan masalalar murakkabligining oshishi, shaxsiy kompyuterlar va lokal hisoblash tarmoqlarining paydo bo'lishi, yangi "fayl-server" arxitekturasi paydo bo'lishiga zamin yaratdi. Tarmoq orqali murojaat qilish mumkin bo'lgan MBning ushbu arxitekturasida tarmoqdagi bitta kompyuter server sifatida ajratiladi va unda MBning fayllari joylashtiriladi. Foydalanuvchilarning so'roviga mos holda fayllar fayl-serverdan foydalanuvchilarning ishchi stantsiyalariga uzatiladi va u yerda ma'lumotlarni qayta ishlash bilan bog'liq asosiy ishlar bajariladi.

Bunda markaziy server ma'lumotlarni qayta ishlashda qatnashmasdan faqat fayllarni saqlovchi vazifasini bajaradi (7.1.2-rasm).

Bu texnologiyani ishlash tartibi quyidagicha:

- MB fayllar to'plami shaklida maxsus ajratilgan kompyuterning (fayl server) qattiq diskida joylashtirilgan bo'ladi;
- lokal tarmoq mavjud bo'lib, undagi har bir mijoz

kompyuterida MBBT va MB bilan ishlash uchun dastur o'rnatiladi;

- o har bir mijoz kompyuterda foydalanuvchilar dasturini ishga tushirish imkoniga ega bo'ladi. Ular dasturdagi foydalanuvchi interfeysi yordamida MB dagi ma'lumotlarni o'qish/yangilash uchun so'rovlarni amalga oshiradi;

- o MBga berilgan barcha so'rovlar MBBT orqali amalga oshadi va MBBT o'zida fayl serverda joylashgan MBni jismoniy tuzilmasi haqidagi ma'lumotlarni saqlaydi;

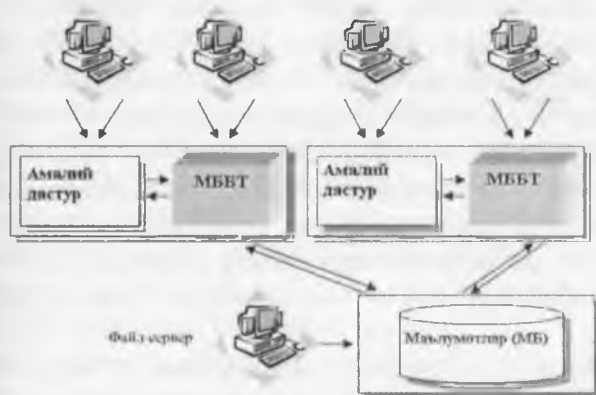
- o MBBT fayl serverda joylashgan ma'lumotlarga murojaatni amalga oshiradi. Buning natijasida MB fayllarining bir qismi mijoz kompyuterga nusxalanadi va qayta ishlanadi. Natijada foydalanuvchi so'rovini bajarish ta'minlanadi (ma'lumotlar ustidagi zarur amallar bajariladi);

- o zarur bo'lganda (ma'lumotlar o'zgartirilganda) MBni yangilash uchun ma'lumotlar orqaga, ya'ni fayl-serverga uzatiladi;

- o MBBT natijalarni dasturga qaytaradi;

- o dastur foydalanuvchi interfeysi yordamida so'rov natijalarini tasvirlaydi.

"Fayl-server" arxitekturasida dBase va Microsoft Access kabi mashhur stol MBBTlari ishlaydi.



7.1.2-rasm. 'Fayl-server' arxitekturası

Adabiyotlarda bu arxitekturaning quyidagi asosiy kamchiliklari ko'rsatiladi:

- Foydalanuvchilar bir vaqtda bitta ma'lumotga murojaat etsa unumdorlik keskin pasayadi, chunki joriy ma'lumot bilan ishlayotgan foydalanuvchi o'z ishini tugatishini kutish zarur. Aks holda, bir foydalanuvchi tomonidan kiritilgan o'zgartish ikkinchi foydalanuvchining amallari natijasida yo'qotilishi mumkin.

- MBga murojaat etganda barcha hisoblash yuklamasi mijoz dasturi zimmasiga tushadi. Chunki, jadvaldan ma'lumot tanlash so'rovi berilganda MBning to'la jadvali mijoz mashinasiga nusxalanadi va tanlash mijoz tomondan bajariladi. Shu tarzda mijoz kompyuteri va tarmoqning resurslari nooptimal tarzda sarflanadi. Natijada, tarmoq trafigi ko'payadi va foydalanuvchi kompyuterining apparat quvvatiga bo'lgan talablari ham oshadi.

- Qoidaga ko'ra, alohida yozuvlar bilan ishlash uchun mo'ljallangan navigatsion yondoshuv qo'llaniladi.

- Fayl-serverdagi MBning alohida jadvallariga dasturdan foydalanmasdan instrumentlar vositasi yordamida o'zgartishlar kiritish juda oson (masalan, Paradox va dBase fayllariga Borland firmasining Database Desktop utiliti yordamida); bu imkoniyat yana shunisi bilan osonlashadi, bunday MBBTlarda MB tushunchasi jismoniy emas ko'proq mantiqiy tushuncha hisoblanadi. Chunki, MB deganda diskning alohida kataloglarida joylashgan alohida jadvallar to'plami tushuniladi. Ma'lumotlarni o'g'irlash, ma'lumotlarni qasddan o'zgartirish yoki bilmasdan xato qiymatlarni kiritish nuqtai nazaridan bularning barchasi xavfsizlikni past darajasini bildiradi. Tranzaktsiya apparatining yaxshi rivojlantirmaganligi, bir yozuvning o'ziga bir vaqtning o'zida bir necha foydalanuvchining murojaat qilishi natijasida ma'lumotlarning ma'no va murojaat butunligigiga putur etkazadi.

"Mijoz - server" arxitekturasini. 'Mijoz - server' texnologiyasini qo'llash tarmoqqa birlashtirilgan kompyuterlarga

asoslanadi va bu kompyuterlardan biri maxsus boshqaruv funktsiyalarini bajaradi (tarmoq serveri bo'ladi).

'Mijoz – server' arxitekturasida funktsiyalarni foydalanuvchi dasturi (mijoz deb ataluvchi) va server funktsiyalariga ajratadi. Mijoz–dastur MB saqlanayotgan serverga SQL tilidagi so'rovni jo'natadi. Masofadagi server so'rovni qabul qiladi va faol bo'lgan SQL-serverga beradi. SQL-server – bu masofadagi MBni boshqaruvchi maxsus dastur. SQL-server so'rovni talqin qiladi, bajaradi, so'rov natijalarini rasmiylashtiradi va mijoz–dasturiga uzatadi. Mijoz kompyuterining resurslari so'rovni bajarishda ishtirok etmaydi. Mijoz kompyuter faqat server MBga so'rov jo'natadi va natijani qabul qiladi. Keyin natijani zarur shaklda talqin qiladi va foydalanuvchiga tasvirlab beradi. Mijoz dasturga so'rovni bajarish natijasi jo'natiladi, ya'ni tarmoq orqali faqat mijozga zarur bo'lgan ma'lumotlar jo'natiladi. Natijada tarmoq yuklamasi pasayadi. Qolaversa, so'rov MB saqlanayotgan joyda (serverda) bajariladi va katta hajmli ma'lumotlar paketini tarmoq orqali jo'natish zarur bo'lmaydi. Bundan tashqari, SQL-server eng kam vaqt sarflanishi uchun so'rovni optimallashtiradi. Bunday tizim arxitekturasida 7.1.3-rasmda tasvirlangan.

Bularning barchasi tizimni tezligini oshiradi va so'rov natijasini kutish vaqtini qisqartiradi. Server tomonidan so'rovlar bajarilganda ma'lumotlarni xavfsizlik darajasi ancha yuqori bo'ladi. Chunki, ma'lumotlarni butunlik qoidasi serverdagi MBda aniqlanadi. Bu qoidalar ushbu MB dan foydalanuvchi barcha dasturlar uchun yagona hisoblanadi. Shu tarzda, butunlikni qo'llab-quvvatlashda qarama-qarshi qoidalarni vujudga kelishi oldi olinadi. SQL-serverlar tomonidan qo'llab-quvvatlanadigan ishonchli tranzaksiyalar apparati bitta ma'lumotni bir vaqtda turli foydalanuvchilar tomonidan o'zgartirilishiga yo'l qo'ymaydi va MBda avariya tugagan o'zgarishlarni bekor qilish imkoniyati mavjud.

"Mijoz – server" arxitekturasida quyidagicha qurilgan:

- o MB fayllar to'plami shaklida maxsus kompyuterning (tarmoq serveri) qattiq diskida joylashadi.

- MBBT ham tarmoq serverida joylashadi.
- Mijoz kompyuterlaridan iborat bo'lgan lokal tarmoq mavjud bo'lib, har bir kompyuterda MB bilan ishllovchi mijoz dasturi o'rnatilgan.
- Har bir mijoz kompyuterda foydalanuvchilar dasturni ishga tushirish imkoniga ega. Dasturning foydalanuvchi interfeysi yordamida foydalanuvchilar ma'lumotlarni tanlash/yangilash uchun serverdagi MBga murojaat qiladi. Murojaat qilish uchun maxsus SQL so'rovlar tili ishlatiladi, ya'ni serverga faqat so'rov matni jo'natiladi.
- MBBTda serverdagi MBning jismoniy tuzilmasi haqidagi ma'lumotlar saqlanadi.
- MBBT serverdagi ma'lumotlarga murojaatni qayd qiladi va serverda ma'lumotlarni qayta ishlash amallari bajariladi va natija mijoz kompyuteriga yuboriladi. Shu tarzda MBBT natijalarni mijoz dasturiga yuboradi.
- Dastur esa o'zidagi foydalanuvchi interfeysi yordamida natijalarni tasvirlab beradi.



7.1.3-rasm. 'Mijoz-server' arxitekturası

Bajariladigan funksiyalar server va mijoz o'rtasida qanday taqsimlanganini ko'rib chiqamiz.

- Mijoz dasturi funksiyalari:
 - so'rovlarni serverga jo'natish;
 - serverdan olingan so'rov natijalarini talqin qilish.
 - natijalarni biror shaklda foydalanuvchiga ko'rsatish (foydalanuvchi interfeysi).
- Server funksiyalari:
 - mijoz –dasturlardan so'rovlarni qabul qilish.
 - so'rovlarni talqin qilish.
 - MB ga so'rovlarni optimallashtirish va bajarish;
 - natijalarni mijoz-dasturga jo'natish;
 - xavfsizlik tizimini va murojaatni chegaralash;
 - MB butunligini boshqarish;
 - ko'p foydalanuvchili ish rejimi stabilligini ta'minlash.

“Sanoat” MBBTlari “mijoz – server” arxitekturasiga asoslanadi. Sanoat MBBT lari deb atalashiga sabab, aynan shu tipdagi MBBTlar o'rta va yirik masshtabli korxona, tashkilot va banklarning axborot tizimlarining faoliyatini ta'minlab beradi. Sanoat MBBT lariga MS SQL Server, Oracle, Gupta, Informix, Sybase, DB2, InterBase va boshqalar misol bo'ladi.

Tartibga ko'ra, SQL – serverga bitta xodim yoki xodimlar guruhi (SQL – server administratori) tomonidan xizmat ko'rsatiladi. Ular MBning jismoniy xarakteristikalarini boshqaradi, MB turli komponentlarini rostlaydi va qayta aniqlaydi, optimallashtiradi, yangi MBlarini yaratadi, mavjud MBlarini o'zgartiradi hamda turli foydalanuvchilar vakolatlarini belgilaydi.

Bu arxitekturaning “fayl-server” arxitekturaga nisbatan afzalliklarini ko'rib chiqamiz:

- Tarmoq trafigi keskin kamayadi.
- Mijoz dasturlarining murakkabligi kamayadi (asosiy yuklama server qismiga beriladi), buning natijasida mijoz

kompyuterlarining apparat quvvatlariga bo'lgan talab pasayadi.

- o Maxsus dasturiy vosita — SQL-serverni mavjudligi – loyihaviy va dasturlash masalasining kattagina qismini yechilishini ta'minlaydi.

- o MB butunligi va xavfsizligi oshadi.

Bu arxitekturaning kamchiligi sifatida apparat va dasturiy ta'minot hamda turli joylardagi ko'p sonli mijoz kompyuterlardagi mijoz-dasturlarni yangilash uchun zarur bo'ladigan yuqori moliyaviy xarajatni aytish mumkin. Shunga qaramasdan "mijoz-server" arxitekturasini amalda o'zini oqladi va hozirgi paytda joriy arxitektura asosidagi ko'pgina MBlari mavjud va ishlatilmoqda.

"Fayl – server" va "mijoz – server" arxitekturalari farqi.
Quyidagi misol orqali tizimlar tahlilini amalga oshiramiz.

1- misol. Firmada buyurtmachi va buyurtma haqidagi ma'lumotlar "Manzil" va "Hisob" nomli jadvallarda saqlanadi. Jadvallar quyidagi atributlardan tashkil topgan:

Manzil (buyurtmachi, ism, familiya, shahar, ko'cha, uy)

Hisob (sana, pul miqdori, to'langan, buyurtmachi)

"Manzil" jadvalida 100000 ta, "Hisob" jadvalida esa 300000 ta yozuv mavjud.

"Hisob" jadvalida "Manzil" jadvalining "buyurtma" qiymati saqlanadi. So'rov matni quyidagicha:

```
SELECT a.ism, a.familiya, a.shahar, a.ko'cha, a.uy, c.sana, c.pul  
miqdori
```

```
FROM manzil a, hisob c
```

```
WHERE c.buyurtmachi=a.buyurtmachi
```

```
AND c.sana=a.oxirgi to'lov muddati buyurtmachi
```

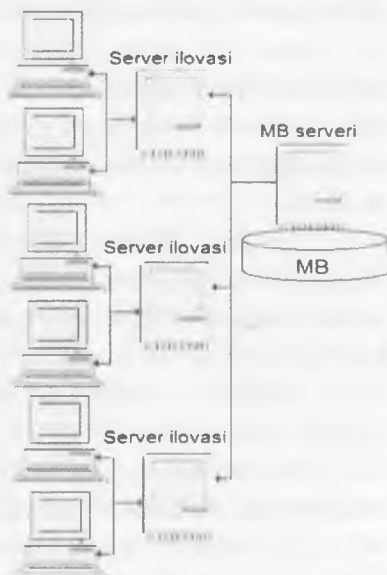
```
AND c.to'langan=IS NULL
```

"Fayl – server" arxitektura ishlatilsa "Manzil" jadvalidan 100000 ta, "Hisob" jadvalidan esa 300000 ta yozuvni, jami 20 MBga yaqin ma'lumotni "mijoz" kompyuteriga ko'chirishga to'g'ri keladi.

“Mijoz – server” arxitektura ishlatilsa soʻrovga serverda ishlov beriladi va shartni bajaruvchi, eng koʻpi bilan 1000 ga yaqin yozuvni, yaʼni taxminan 10 Kb ga yaqin maʼlumotni “mijoz” kompyuteriga koʻchirishga toʻgʻri keladi. Bu tarmoq yukini “fayl – server” arxitekturasiga nisbatan 2000 martaga yaqin engillashtiradi.

“Mijoz – server” arxitekturasi katta hajmli maʼlumotlarga tez ishlov bera oladi va bir vaqtning oʻzida koʻp sonli foydalanuvchilar bilan ishlay oladi. Bundan tashqari, u foydalanuvchilarning MBga murojaatini toʻxtatib turmasdan, MB nusxasini ola oladi. Masalan, mobil aloqa mijozlarini, aloqasini tungi 2 dan 3 gacha toʻxtatib, MB zahirasini olinishini tasavvur qiling.

Uch zvenoli (koʻp zvenoli) arxitektur. Uch zvenoli (baʼzida koʻp zvenoli) arxitektura (N-tier yoki multi-tier) “mijoz-server” texnologiyasining rivojlangan va takomillashtirilgan shaklini ifodalaydi. “Mijoz – server” arxitekturasi bilan tanishib, u ikkita zvenodan tashkil topgan: birinchi zveno – mijoz dasturi, ikkinchi zveno – MB serveri + MBning oʻzi degan xulosaga kelinadi. Uch zvenoli arxitektura bu butunicha biznes-logika (ishbilarmon-mantiq), oldin mijoz dastur boʻlgan zveno alohida ilovalar zvenosiga kiritiladi. Bunda mijoz dasturi sifatida faqat foydalanuvchi interfeysi qoladi.



Unda mijoz ilovasi sifatida Web-brauzer ishlatiladi.

7.1.4-rasm. Uch zvenoli arxitektura

Uch zvenoli arxitekturadan

foydalanganda qanday o'zgarish yuz beradi? Unda endi biznes – logika o'zgarganda mijoz dasturlarini o'zgartirish va barcha foydalanuvchilar uni yangilashi zarur emas. Bundan tashqari foydalanuvchi kompyuterining apparatiga bo'lgan talab keskin pasayadi.

Uch zvenoli arxitekturaning ishlash tartibi quyidagicha qurilgan:

- MB fayllar to'plami shaklida maxsus ajratilgan kompyuterining (tarmoq serveri) qattiq diskida joylashadi.

- MBBT ham tarmoq serverida joylashadi.

- Maxsus ajratilgan ilovalar serveri mavjud bo'lib, unda ishbilarmon doiraning (biznes – logika) dasturiy ta'minoti joylashadi.

- Ko'p sonli mijoz kompyuterlari mavjud va ularda "kichik mijoz" deb ataluvchi mijoz dasturi o'rnatilgan bo'lib, ular foydalanuvchi interfeysini hosil qiladi.

- Har bir mijoz kompyuterida foydalanuvchilar kichik - mijoz dasturini ishga tushirish imkoniga ega. Bu dastur yaratgan foydalanuvchi interfeysi yordamida foydalanuvchilar ilovalar serveridagi ishbilarmon doiraning dasturiy ta'minotiga murojaat etadi.

- Ilovalar serveri foydalanuvchi talabini tahlil qiladi va MB so'rovlarni hosil qiladi. Buning uchun maxsus SQL –so'rovlar tili qo'llaniladi, ya'ni tarmoq orqali ilovalar serveridan MB serveriga faqat so'rov matni jo'natiladi.

- MBBTda serverdagi MBning jismoniy tuzilmasi haqidagi ma'lumotlar saqlanadi.

- MBBT serverdagi ma'lumotlarga murojaatni bajaradi va uning natijalari ilovalar serveriga jo'natiladi.

- Ilovalar serveri natijani mijoz dasturiga (foydalanuvchiga) jo'natadi.

- Ilova foydalanuvchi interfeysi yordamida so'rov natijalarini tasvirlab beradi.

Cartner Group tadqiqotlariga muvofiq "mijoz-server" da oʻrtacha 700 tagacha mijoz mavjud boʻladi. Mijozlarning hammasi ham serverga bogʻlanmagan taqdirda ham bir tizimda zoʻriqish vujudga keladi.

Koʻp zvenoli arxitektura sharofati bilan zoʻriqish taxminiy ravishda teng holda server ilovalariga yuklanadi. Bundan tashqari, server ilovalari serverga xato qoʻllanmalar tushishini oldini oladi. Baʼzi bir hollarda bir guruh mijozlar bir vaqtning oʻzida bir xilda soʻrov berishadi. Serverdan qiymatlar bir marta berilishi kerak. Oʻzgarishlarni buferlashtirishdan foydalanish orqali qiymatlar MB serveridan yuklanadi. Ishchi guruhning server ilovasi orqali maʼlumotga ishlov beriladi va maʼlum bir vaqt oʻtgandan soʻng toʻplangan oʻzgarishlar yana orqaga, MB serveriga qaytariladi. Mijoz interfeysi dasturlari server ilovalarda joylashgan boʻladi.

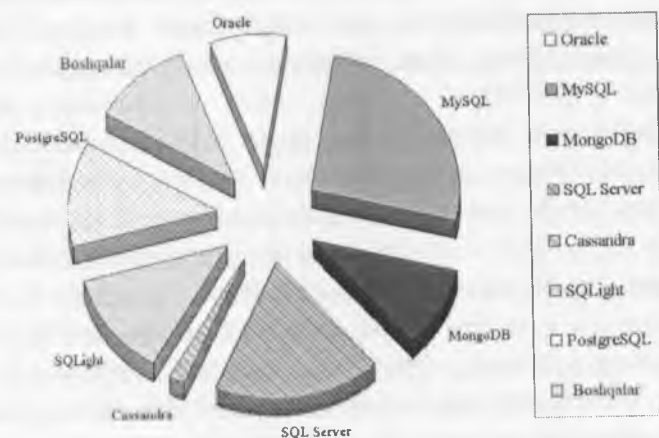
Koʻpgina mutaxassislar MBBTlarini ikkita katta sinfga ajratishadi: "stol" va "server".

Stol MBBT. Stol MBBTlari nisbatan kichik masalalar (qayta ishlanadigan maʼlumotlarning katta boʻlmagan miqdori, kam sonli foydalanuvchilar) uchun ishlatiladi. Shuni inobatga olgan holda bunday MBBTlari nisbatan sodda arxitekturaga ega, xususan, fayl-server rejimida ishlaydi va MBBTlarning barcha funktsiyalarini (masalan, tranzaktsiya jurnali yuritilmaydi, ishdagi xatoliklarda avtomatik ravishda MBni tiklash mavjud emas va h-zo) qoʻllab – quvvatlamaydi. Shunday boʻlsada, bu tizimlar keng qoʻllanish sohasiga ega. Bular, baʼzi bir kichik davlat korxonalari, taʼlim sohasi, xizmatlar sohasi, kichik-oʻrta biznes va h-zo sohalaridir. Bu tizimlar maʼlumotlar hajmi juda ham katta boʻlmagan, maʼlumotlarni oʻzgartirish tebranishlari etarlicha yuqori emas, tashkilot asosan kichik radiusni qamrab oladigan (asosan bir yoki bir necha binoni qamrab oladi), foydalanuvchilar soni birtadan 15 tagacha boʻlgan sohalar uchun qulaydir.

Bu turdagi tizimlarning boshlang'ich tipi dBase dir. Keyinchalik dBase III – PLUS (Achton-Tate firmasi), Clipper (Nantucket Inc. firmasi), FoxPro (Fox Software firmasi), FoxBase+ (Fox Software firmasi), Visual FoxPro (Microsoft firmasi) tizim-kompilyatorlari yaratildi. Eng keng miqyosda PARADOX (Borland International firmasi) MBBT ishlatilgan. Oxirgi yillarda keng miqyosda ishlatilayotgan MBBT Microsoft Access dir.

Serverli MBBT. Yirik tashkilotlarda “fayl-server” texnologiyasi asosida ishlovchi serverli MBBTlaridan foydalaniladi. Serverli MBBTlarini yaratishda dunyoning 3 ta yirik korporatsiyasi yetakchilik qilishmoqda. Ular Oracle, Microsoft va IBM. Bu korporatsiyalarning MBBT sotuvi bo'yicha 2017 yilda olingan ma'lumoti 7.1.5-rasmdagi diagrammada tasvirlangan.

Eng ko'p tarqalgan “mijoz-server” texnologiyali MBBTlari My SQL, MS SQL Server (Microsoft kompaniyasi), Postgre SQL, Oracle (Oracle kompaniyasi), SQLight hisoblanadi [1].



7.1.5-rasm. MBBTlarnig ommaviylashganlik darajalari

7.2. MBBTlarini standartlashtirish va rivojlanish istiqbollari

Zamonaviy MBBTlari fayl tizimlari kamchiliklariga chek qo'yish maqsadida ishlab chiqilgan. MBBTning ishlab chiqishda amal qiluvchi quyidagi printsiplari mavjud:

- *ma'lumotlarning mustaqilligi*;
 - *universallik*. Foydalanuvchining mantiqiy - tasavvurlarini aks ettirish uchun MBBT kontseptual modelini qo'llab-quvvatlashning kuchli vositalariga ega bo'lish;
 - *mutanosiblik*. MBBT dasturiy va apparat ta'minotini rivojlantirishda ish qobiliyatini saqlab qolish;
 - *ma'lumotlarning me'yordan ortiq emasligi*. Fayl tizimlaridan farqli ravishda MB integratsiyalashgan ma'lumotlarning yagona majmuasini mujassamlashtirish;
 - *ma'lumotlarni himoya qilish*. MBBT ruksatsiz kirishlarda himoya qilishni ta'minlash;
 - *ma'lumotlarning yaxlitligi*. MBBT MBning foydalanuvchilar tomonidan buzilishining oldini olish;
 - *bir vaqtdagi ishlarni boshqarish*. MBBT MBni yoppasiga foydalanish rejimidagi nomuvofiqliklardan saqlashi zarur. MBning muvofiqlashgan holatlarda ta'minlash uchun foydalanuvchilarning barcha talablari (tranzaktsiyalari) belgilangan tartibda bajarilishi lozim;
 - *ma'lumotlar mustaqilligi eng muhim xususiyat*. Chunki, u ma'lumotlar integratsiyasi, me'yordan ortiq bo'lmaslik, yoppasiga foydalanish va himoya hamda yaxlitlikni ta'minlash kabi boshqa xususiyatlarning mavjudligiga ta'sir etadi.
- Bu printsiplarga yana quyidagilarni qo'shish lozim:
- MBBT universal bo'lishi, ya'ni u yagona mantiqli va jismoniy asosidagi ma'lumotlarning turli modellarini ta'minlash;
 - MBBT ham markazlashgan, ham taqsimlangan MBni qo'llab-quvvatlashi zarur. Hozirgi paytda hisoblash tarmoqlari va

ma'lumotlarni taqsimlangan holda ishlash ommalashmoqda. Ma'lumotlarning lokal mustaqilligi muhim bo'lgan hududiy taqsimlangan tashkil etishlarga mo'ljallangan MBBT — taqsimlangan ma'lumotlar bazasini boshqarish tizimi (TMBBT) bo'lishi zarur.

Axborot texnologiyalarini qo'llash sohalarining kengayishi zamonaviy MBBTlar oldiga foydalanuvchilar va axborotlar tizimi o'rtasidagi o'zaro hamkorlikni osonlashtirishga qaratilgan bir qator yangi talablarni qo'ydi. Bu talablarni quyidagi vazifalarni hal etish orqali amalga oshirishi mumkin:

- o foydalanuvchining (amaliy dasturlardan) MB bir tizimidan boshqa tizimiga o'tishi uchun bir xillashtirilgan interfeysni ta'minlash;

- o ba'zi ma'lumotlar modeli uchun ma'lumotlar bilan amal bajarish tilining standart kichik "ko'pligi"ni ajratish. Bu ma'lumotlar yangi ishlab chiqilayotgan MBBTlarida nazarda tutilgan foydalanuvchilarga mo'ljallangan. Ularda quyidagilar ko'zda tutilgan: MBni bir model ichida bir xilda taqdim etish, muloqot qilish va dasturlashtirishning hech bo'lmaganda yetarlicha namoyon bo'ladigan kesishuvlarga ega birxillashtirilgan vositalari. Qurilishiga ko'ra farqlanadigan kompyuterlar (platformalari) yoki ma'lumotlarning bir xil modelini ta'minlovchi MBBT bilan turli operatsion tizim muhitlarida ishlayotgan foydalanuvchi nuqtai nazaridan, uning bu MBBTlari bilan o'zaro aloqasi shu tariqa birxillashtirilgan bo'lishi zarur;

- o MB muhitida ishlovchi amaliy dasturlarni avtomatik loyihalashning yanada rivojlangan vositalarini yaratish;

- o LISP, Prolog, Python, R Programming kabi sun'iy idrok tizimlari uchun instrumental vosita sifatida foydalaniladigan dasturlar tizimiga ega bo'lgan MBBTlarining interfeyslarini rivojlantirish;

- o dasturlashtirishning an'anaviy tizimlaridan MBBTga

murojaat qilishning bir xilligini ta'minlash. Bu MBga murojaat qilishlardan foydalanuvchi amaliy dasturlarning safarbarligiga zamin yaratadi.

Interfeyslar tizimi asosida ko'rsatib o'tilgan talablarni bajarishga olib keluvchi ikki yo'l ko'zga tashlanadi. Bular standartlashtirish bo'yicha xalqaro tashkilot (ISO) ning faoliyati va System Application Architecture (SAA) tizimlarini qo'llash arxitekturasining kontseptsiyasidir. SAA IBM firmasi tomonidan tavsiya etilgan. Bunda ISO ning faoliyati, ma'lumki, turli tuzilishdagi ma'lumotlarni ishlashning ochiq tizimi uchun muammolarning juda keng qatlamini qamrab oladi. SAA kontseptsiyasi esa, ko'proq xususiy vazifalarning echimi bo'lib, IBM firmasining faqat uchta sinfi uchun e'lon qilingan. Biroq, ushbu yo'nalishlar belgilab berilgan bir qator vazifalar va echimlar, ko'p jihatdan bir-biriga muvofiqlashtirilib turiladi.

MBBT sohasida standartlashtirish bo'yicha xalqaro faoliyat 1989 yilda SQL MBning tili xalqaro standarti ishlab chiqilgandan so'ng, ISO ni ishi asosan quyidagi ikki yo'nalishda olib borildi:

- o ochiq tizimlarda taqsimlangan ishlov tamoyillarini rivojlantirish. Bu MBBT uchun tranzaktsiyalar (shuningdek, qaydnomalar va xizmatlar) taqsimlangan ishlov berish va kompyuter tarmog'idagi yo'qotilgan MBga kirish kontseptsiyasini ishlab chiqishga olib keladi;

- o korxonalar yoki tashkilotlar axborot zahiralarning lug'at tizimlari uchun tamoyillar va interfeyslar tizimini belgilash.

Bu yo'nalishda qilinayotgan ishlar SQL MB tilini takomillashtirish va kengaytirishni talab qiladi. Bu ham mavjud standartga qo'shimchalar kiritish shaklida, ham SQL2 va SQL3 yangi standartlarining loyihalari ko'rinishida amalga oshirilmoqda.

Ma'lumotlarni boshqarishning o'lchov modelini standartlashtirishda standartlar tizimining shunday tuzilishini

belgilashi kerakki, undan MBBT sohasida ham ishlab chiqilgan, ham ishlab chiqilayotgan istiqbolli standartlar samarali joy olishi mumkin bo'lsin. O'lchov modelini ishlab chiqish MBBT standartlarini NDL (tarmoq turi), SQLflam MB ko'rinishlariga tegishli bo'lgan sohalarda, shuningdek, yo'qotilgan MBga kirish va axborot zahiralari lug'ati tizimida rejalashtirilgan rivojlantirish asosida olib borilmoqda. Umuman, yangi model ochiq tizimlarning o'zaro aloqasi o'lchov modeliga tegishli bo'lgan standartlarga asoslanadi. MBBTning o'lchov modeli MBBTni standartizatsiya qilish jihatlarini shunday tasnif qiladiki, unda MBBT o'z funksional xususiyatlari va loyihalashning turli bosqichlarida ma'lumotlarni ta'riflash darajasiga bog'liq holda ko'rib chiqiladi.

Amaldagi vazifalar bo'yicha tasniflash ma'lumotlarni boshqarishga xizmat ko'rsatishni to'rt qiymatini ajratishga imkon beradi: bazisli, ma'lumotlar bazalari, instrumental va foydalanuvchining talablariga xizmat ko'rsatish.

Bazisli qatlam ma'lumotlar blokini ularning mazmunini xususiyatlashtirmasdan saqlash va chiqarib olish uchun xizmat ko'rsatadi. MB larining qatlami ochiq hisob tizimi ichida MB bilan ishlash uchun asosiy xizmatlarni taqdim etadi. Bu xizmatlar, ochiq, tizimlar o'zaro ta'sirining o'lchov modelini ta'riflovchi xizmatlarga aloqador emasligi nazarda tutiladi. Instrumental qatlam bir turda jonli ma'lumotlar to'plamini, turli ochiq. hisob tizimlarida lokallashgan, taqsimlangan MBga kirishni ta'minlash uchun xizmatlar to'plamini taqdim etadi. Foydalanuvchining talablariga xizmat ko'rsatish qatlami oxirgi foydalanuvchining hisoblash tizimi bilan o'zaro ta'sirini engillashtiradigan ma'lumotlarni boshqarishga xizmat ko'rsatishni ta'riflashga mo'ljallangan. ekran interfeysi va ekran pochta tizimi shunday xizmatlarga misol bo'la oladi. Ma'lumotlar ularda mavjud ta'rif darajasiga ko'ra, amaliy, lug'atli, lug'atni va axborotlar tizimi uchun asosiy tushunchalarni saqlovchi

ma'lumotlarga bo'linadi. Bu tasniflarning uyg'unligi standartlar tizimining tuzilishini belgilab beradi.

MB tilini yaratish bo'yicha ishlar dasturi quyidagilarni o'z ichiga oladi: MB kontseptual tillari va o'qish asosini aniqlash va tahlil etish. MBning kontseptual tillari bo'yicha tavsiyalarni baholash uchun metodologiya ishlab chiqish, shu o'ringa nomzod sifatida tavsiya etilgan MB tillarini baholash, kontseptual darajadagi MB tillari foydalanuvchisining asosiy imkoniyatlarini tanlab olish. Bu sohadagi eng sezilarli natijalar — standartlar sifatida NDL tarmoq va SQL relyatsion ko'rinishdagi MB tillarining qabul qilinishi bo'ldi.

Korxona axborot resurslari lug'atining tizimlarini yaratish bo'yicha ishlar loyihalashtirilmoqda. Bunday tizim foydalanuvchiga ixtisoslashgan MBni taqdim etadi. Ularga tizimni qo'llashiga qarab MB, ma'lumotlar to'plami, dasturlar va hokazolar kiradi. Standartning asosini lug'at bilan qo'llab-quvvatlanadigan, ma'lumotlar modelini ta'riflash, lug'at ma'lumotlari bilan amallar bajarish uchun buyruq, tili va tizimdan foydalanuvchiga muloqotli kirishning ekranli interfeysi tashkil etiladi.

Bundan tashqari yadroni kengaytirish, masalan, amaliy dasturlar uchun interfeyslar taalluqli bo'lgan modullar deb nomlangan standartlar tavsiya etiladi. Bunday modullilik ko'rsatilgan standartni axborot zahiralari lug'ati tizimida foydalanishning yangi usullari paydo bo'lishiga qarab qengaytirish uchun yo'l ochib qo'yadi.

7.3. MS Access

MS Access MBBT da MB, bog'lanishlar, so'rovlar, formalar, hisobotlar va turli ob'ektlar jismoniy nuqtai nazardan bitta faylda joylashadi. Bu faylni ixtiyoriy kompyuterga ko'chirib o'tkazib, belgilangan ishlarni bajarish mumkin. MS Access MBBT asosan "Fayl-server" arxitekturasida ishlaydi.

Loyihasi yaratilgan MBni kiritishga o'tishdan oldin qattiq diskda MB faylini saqlash uchun ixtiyoriy nom (masalan BAZA) bilan fayl yaratiladi. (MS Word, MS Excel va MS PowerPoint larni o'rganishda hujjatlar bilan, ularni saqlamasdan ishlash mumkin, ammo MS Access da bu hol mumkin emas.)

Microsoft Access ishga tushiriladi.

Asboblari panelida "создать базу данных" tugmasi bosiladi yoki "Файл" menyusida "Создать" buyrug'i tanlanadi. Hosil bo'lgan oynaning o'ng tomonidan joylashgan qismdan "Новая база данных" buyrug'i tanlanadi.

"Файл новая базы данных" muloqot oynasi ochiladi, uning "Имя файла" bo'limida yaratilayotgan MBga nom beriladi (masalan db1) va xotirada saqlash uchun papka manzili ko'rsatiladi. Natijada MB oynasiga ega bo'lamiz:

MB oynasi MB ochilganda chiqariladi. Undan jadvallar, formalar va MBning boshqa ob'ektlari ochiladi.

MBning oynasi quyidagi elementlarga ega:

Sarlavha satri. Ochilgan MB ismini chiqaradi.

Tugmalar. "Создать", "Открыть", "Конструктор" va h. k. Tugmalar ma'lum oynada va rejimda ob'ektni ochadi.

Ob'ekt oynalari (ob'ektlarni tanlash yorliqlari). "Таблица", "Форма" va h-zo. Ob'ektlar tugmalari ochilish yoki o'zgarishi mumkin bo'lgan ob'ektlar ro'yxatini chiqaradi.

Ob'ektlar ro'yxati. Foydalanuvchi tanlaydigan ob'ektlar ro'yxatini chiqaradi.

MBni kiritishni bevosita boshlashdan oldin MBning ba'zi bir asosiy ob'ektlarning tavsiflariga to'xtaymiz:

Таблица (jadval) – bu ma'lumotlarni yozuv (satri) lar va maydon (ustun)lar ko'rinishida saqlovchi ob'ekt. Odatda har bir jadval bitta aniq masala bo'yicha ma'lumotlarni saqlash uchun ishlatiladi.

Форма (shakl) – asosan ma'lumotlarni chiqarish va kiritish uchun ishlatiladigan interfeys. Formada jadval maydonlaridagi

ma'lumotlarni kiritish, tasvirlash va o'zgartirishda ishlatiladigan boshqarish elementlarini joylashtirish mumkin.

Запрос (so'rov) – kerakli ma'lumotlarni bitta yoki bir nechta jadvaldan olish imkonini beradi.

Отчеты (hisobotlar) – ma'lumotlarni chop etish uchun mo'ljallangan MS Accessning MB ob'ekti.

Ish jadvalni yaratishdan boshlanadi. Jadvalda ma'lum tur, masalan mijozlar yoki tovarlar ro'yxatining ma'lumotlarini saqlovchi yozuv saqlanadi. Jadvalning asosiy qismi maydon bo'ladi.

Maydon – bu ma'lum bir turdagi, masalan, xodimlar familiyasiga, ma'lumotlarga ega bo'lgan jadval elementi. Jadval rejimida maydonni ifodalash uchun ustun yoki yacheyka ishlatiladi, bu holda maydon ismi jadval ustuni sarlavhasi bo'ladi.

Yozuv – ma'lum ob'ekt haqidagi ma'lumotlarning to'la to'plami. Jadval rejimida yozuv satr ko'rinishida tasvirlanadi [1].

7.4. MS SQL Server

MS SQL Server MBBT Microsoft firmasi mahsuloti.

Shu kunga qadar MS SQL Server-2000, MS SQL Server-2005, MS SQL Server-2008, MS SQL Server-2012, MS SQL Server-2016 versiyalari yaratilgan.

MS SQL Server-2008 ga XML tili, HTTP protokolini qo'llash kiritilgan. Mobil aloqa va mobil texnikalarining geometrik progressiya bo'yicha rivojlanishi, web texnologiyalarining rivojlanishi, internetga mobil aloqa orqali kirishning taraqqiy etishi, NET texnologiya, NET Framework va Visual Studio ning NET bo'yicha ishlashi vujudga kelishi natijasida MBBT larni ham qayta ko'rib chiqishni taqoza qildi. Ushbu sohalaridagi taraqqiyot imkoniyatlari MS SQL Server da o'z in'okisini topgan. MS SQL Server ham katta hajmli MB bilan ishlashga, yaxshi himoya vositalariga, keng imkoniyatlarga va minglab foydalanuvchilarga

bir vaqtning o'zida xizmat qilish quvvatiga hamda har qanday web xizmatlarni qo'llash imkoniga ega.

MS SQL Server ham xuddi Oracle MBBT singari, bir biri bilan bog'langan hamda har biri qisman mustaqil bo'lgan, bir necha yirik dasturiy majmualar birlashmasidan tashkil topgan. Har bir dasturiy majmua qaysidir sohaga tegishli ishlarni bajarish uchun o'z xizmatlarini tavsiya qiladi. MS SQL Server ham Oracle singari xizmatlarni va MBBTning eng yuqori ish jarayonini taklif qiladi.

MS SQL Server da ham ANSI SQL standartiga muvofiq keluvchi SQL tili qo'llaniladi. Dasturiy majmua tarkibida Transact-SQL (T-SQL) interektiv interpretator mavjud.

MS SQL Server Microsoft Office bilan chambarchas bog'langan bo'lib, ma'lumotlarni MS Excel va MS Word interfeyslariga chiqarish va bu interfeyslardan MBga ma'lumot kiritish imkoniyatlari mavjud.

Xizmatlar bozorida MS SQL Server Oracle bilan raqobatlasha oladigan MBBT hisoblanadi. Ammo, shu soha mutaxassislarining fikriga ko'ra MS SQL Serverda himoyalaniish darajasi Oracle ga nisbatan bir oz bo'shroq hisoblanadi. MS SQL Server muhiti administratori ish jarayoni Oraclega nisbatan bir oz soddaroq. Bu uning katta yutug'i hisoblanadi. MS SQL Server dasturchilar uchun qulay muhitni tashkil qilib beradi va unda web xizmatlarga muvofiq ma'lumotlar bilan ishlash imkoniyatlari mavjud [14].

7.5. IBM kompaniyasining serverli MBBTlari

IBM kompaniyasi DB2 va Informix MBBT lari bilan ham mashhurdir. Hozirgi kunda DB2 Universal Database, DB2 Personal Edition, DB2 Enterprise 9 hamda Informix Dynamic Server, Informix Dynamic Server Express, Informix Extended Parallel Server va h-zo MBBTlar ko'p muassasalarda ishlatilmoqda.

Umumiy server MB hisoblangan DB2 Universal Database, mul'timedia va Web ni bog'lagan holda qo'llovchi ob'ektga yo'naltirilgan MBBT bo'lib, u barcha turdagi (bir va simmetrik ko'p protsessorli, umumiy parallel tizimli) protsessorga ega bo'lgan kompyuterlarda hamda manfreymlarda ishlatiladi. DB2 Universal Database yangi biriktirilgan vosita bo'lib, u o'z bazasiga Oracle, Microsoft, Sybase va Informix singari MB ma'lumotlarini olishni yengillashtiradi. Bundan tashqari, DB2 Universal Database real vaqtga muvofiq analitik ishlov berish (OLAP), qarorlarni qabul qilish tizimini qo'llash va turli xil oddiy kengaytmalardan (DB2 extenders) foydalanish vositasiga ega. DB2 Universal Database ko'pchilik muhim platformalarda qo'llaniladi va ularda buyurtmachilarga zarur bo'lgan qayishqoqlikni beradi.

IBM kompaniyasi DB2 va Informix MBBTlari ham Oracle va MS SQL Server singari imkoniyatlarga ega hamda foydalanuvchilarning talablarini to'liq qanoatlantira oladi.

7.6. My SQL

My SQL Mixael Videnius tomonidan ishlab chiqilgan bo'lib, u Hughes Technologies Mini SQL (mSQL) traditsiyasiga muvofiq katta bo'lmagan va tez ishlovchi relyatsion MBBT hisoblanadi.

My SQL paketining eng muhim xislatlari:

- o ko'p oqimlilik. Bir vaqtning o'zida berilgan so'rovlarni qo'llash;
- o bir o'tishda ko'p qiymatlarni bog'lashda aloqalarni optimallashtirish;
- o o'zgaruvchi va bir xilli uzunlikdagi yozuvlar;
- o boshlang'ich paketda ODBC drayver mavjudligi;
- o jadvalda 16 tagacha kalitga ega bo'lishlik. Har bir kalit 15 tagacha maydonga ega bo'lishi mumkin;
- o imtiyoz va parol bo'yicha qayishqoq tizimga egalik;
- o CREATE operatori kalit maydonlarni va maxsus

maydonlarni qo'llaydi;

- o 1 va 4 bayt uzunlikdagi sonlarni (int, float, double, fixed), o'zgaruvchi uzunlikdagi matnlarni va vaqt ko'rinishlarini qo'llash;

- o C va Perl tillari mavjud bo'lgan interfeys;

- o oqimlarga asoslangan tez xotira tizimi;

- o jadvallarni tekshirish va tuzatish utiliti (isamchk);

- o barcha ma'lumotlar ISO8859_1 formatida saqlanadi;

- o qatorlar bilan ishlash operatorlari harf belgi registriga e'tibor bermaydi;

- o jadvallarga va jadvalning alohida ustunlariga psevdanim qo'llash mumkinligi;

- o har bir ustun so'zsiz qiymatga egaligi. Har qanday ustunlar birlashmasiga INSERT ni qo'llash mumkin;

- o kalitlarni va maydonlarni qo'shish hamda olib tashlash singari amallar bilan birga jadvallarni oson boshqarish mumkin.

Hozirgi kunda katta hajmga ega bo'lmagan va juda ko'plab foydalanuvchilari bo'lmagan, ko'proq Web texnologiyalar qo'llaniladigan sohalarda PHP tili orqali murojaat qilinadigan ma'lumotlar bazasi My SQL MBBTda tashkil qilinmoqda.

Mavzu bo'yicha quyidagi xulosalarni keltirish mumkin:

1. Ko'plab ma'lumotlarni jadvallar asosida tartibga solish mumkin. Bu jadvallar o'zaro bog'lanishi kerak bo'ladi.

2. Hozirda ma'lumotlarning relyatsion modeli keng qo'llanilmoqda. Bunday modellar foydalanuvchi uchun qulay bo'lgan jadval ko'rinishiga ega. Relyatsion model ma'lumotlarni shakl o'lchovli jadvalda tasvirlaydi.

3. MBBT yordamida baza tuziladi va qayta ishlanadi. Xususan unda jadvallar, formalar, so'rovlar va hisobotlarni yaratishimiz mumkin [11].

7.7. Taqsimlangan ma'lumotlar bazasi va uni qayta ishlash tizimlari arxitekturası

Ma'lumotlarni taqsimlangan qayta ishlashga o'tish. MBni hisoblash tizimlarida qo'llanilishi natijasida ma'lumotlarni qayta ishlashning eski usullari o'rniga yangi usullar paydo bo'lishiga sabab bo'ldi. Yangi usulda har bir ilova uchun alohida ma'lumotlar to'plami aniqlanadi va qo'llab – quvvatlanadi. Eski usulda esa barcha ma'lumotlar markazlashgan holda aniqlangan va qo'llab quvvatlangan.

So'nggi paytlarda tarmoq orqali ma'lumot almashish texnologiyalari shiddat bilan rivojlanmoqda. Bular Internet tarmog'i, mobil va simsiz hisoblash vositalari hamda "intellektual" qurilmalardir. Tarmoq orqali ma'lumot almashish shiddat bilan rivojlanayotganligi, MBga uzoq masofalardan murojaat qilish ehtiyoji oshishiga olib keldi. Uzoq masofalardan MBga murojaat qiluvchi foydalanuvchilar ehtiyojini tez qondirish maqsadida taqsimlangan MB yaratildi. Taqsimlangan MBBT axborot texnologiyalarining MB sohasidagi eng katta yutuqlardan biri hisoblanadi. Taqsimlangan MB texnologiyasi ma'lumotlarni qayta ishlashning markazlashgan turidan, unga teskari bo'lgan markazlashmagan turiga o'tishga zamin yaratmoqda.

Taqsimlangan MB bilan bog'liq muammolarni tahlil qilishdan oldin taqsimlangan MB nima ekanligini aniqlab olish zarur.

Taqsimlangan MB – bu o'zaro mantiqan bog'langan va kompyuter tarmog'ida jismoniy jihatdan tarqoq joylashgan taqsimlangan ma'lumotlar to'plamidir.

Taqsimlangan MB. Ma'lumotlarni qayta ishlashning bu kesishuvchi va hatto takrolanuvchi ma'lumotlar saqlanadigan bir nechta serverlardan foydalanishni talab etadi. Bunday MB bilan ishlash uchun taqsimlangan MBBT ishlatiladi.

Taqsimlangan MBBT- taqsimlangan MBlarini boshqarish uchun mo'ljallangan va foydalanuvchilarga taqsimlangan ma'lumotlarga oson murojaat usulini ta'minlab beruvchi dasturiy majmuadir.

Zamonaviy taqsimlangan MBBT arxitekturası.
Taqsimlangan MBBT bir qancha fragmentlarga bo'lingan va mantiqiy jihatdan yagona bo'lgan MB dan iborat. MBning har bir fragmenti alohida MBBT boshqaruvchi ostida va o'zaro aloqa tarmog'i orqali bog'langan bitta yoki bir nechta kompyuterda saqlanadi. Tarmoqdagi har bir tugun foydalanuvchilarning lokal saqlanayotgan ma'lumotlarga bo'lgan so'rovlarini o'zi mustaqil qayta ishlash (ya'ni, har biri ma'lum darajadagi avtonomlikka ega) hamda tarmoqning boshqa kompyuteridagi ma'lumotlarni qayta ishlash imkoniga ega.

Foydalanuvchilar taqsimlangan MBga dastur yordamida murojaat etadi. Dasturlar boshqa tugundagi ma'lumotlarga murojaatni talab etmaydigan (lokal) va talab etadigan (global) dasturlarga bo'linadi. Taqsimlangan MBBTda kamida bitta global dastur mavjud bo'ladi kerak. Shuning uchun, ixtiyoriy taqsimlangan MBBT quyidagi xarakteristikalarga ega bo'lishi kerak.

- o mantiqan bo'lingan ma'lumotlar to'plami mavjud bo'lishi kerak;

- o saqlanadigan ma'lumotlar bir qancha fragmentlarga bo'lingan;

- o ma'lumotlar fragmentining replikatsiyasi mavjud bo'lishi mumkin;

- o fragmentlar va ularning nusxalari turli tugunlarga taqsimlanadi;

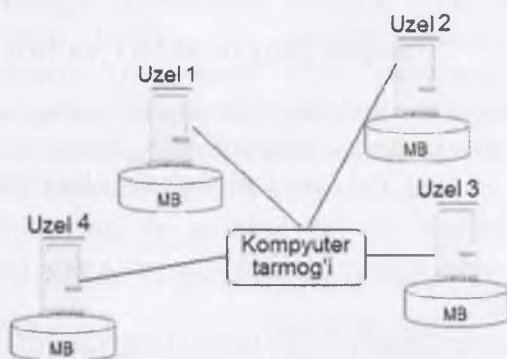
- o tugunlar bir – biri bilan tarmoq orqali bog'lanadi;

- o har bir tugundagi ma'lumotga murojaat MBBT boshqaruvi ostida amalga oshadi;

- MBBT har bir tugundagi lokal dasturlarni avtonom ishlashini qo'llab – quvvatlash imkoniga ega;
- har bir tugundagi MBBT kamida bitta global dasturni qo'llab-quvvatlaydi.

Misol tariqasida 7.7.1-rasmdagi ko'rsatilgan taqsimlangan MB topologiyasidagi kabi tizimning har bir tugunida o'zining xususiy lokal MB mavjud bo'lishi shart emas.

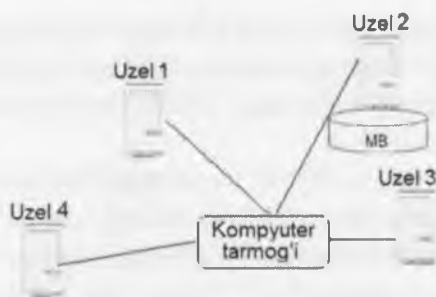
Taqsimlangan MBBT bilan taqsimlangan qayta ishlash vositalari o'rtasidagi farqni anglash muhim hisoblanadi.



7.7.1-rasm. Taqsimlangan MBBT topologiyasi

Taqsimlangan qayta ishlash. Markazlashgan MB ni qayta ishlashni turli kompyuterlardan turib amalga oshirilishidir.

Taqsimlangan MBBTni ta'riflashda bu tizim jismoniy jihatdan tarmoqda taqsimlangan ma'lumotlar bilan ishlashi asosiy hisoblanadi. Agar ma'lumotlar markazlashgan holda saqlanayotgan bo'lsa, u holda hatto ixtiyoriy foydalanuvchi bu ma'lumotlarga tarmoqdagi ixtiyoriy kompyuterdan murojaat etayotgan bo'lsa ham, bu tizim taqsimlangan qayta ishlashni qo'llab quvvatlovchi hisoblanadi, ammo taqsimlangan MBBT sifatida qaralmaydi. Taqsimlangan qayta ishlash topologiyasi sxemasi 7.7.2-rasmda tasvirlangan. Bu sxemada tasvirlangan 2-tugundagi markazlashgan MBni 7.7.1-rasmdagi MB bilan solishtiring



7.7.2-rasm. Taqsimlangan qayta ishlash topologiyasi.

Taqsimlangan MBBT va Internet

Taqsimlangan MBga kompyuter tarmog'i orqali kiriladi va u bir biridan ancha masofada joylashgan bir necha MBdan tashkil topgan bo'ladi. Demak, bu MBga bevosita internet orqali kirish lozim bo'ladi. Bu MB larining ish jarayonini tashkil qilish va ularga oson kirishni ta'minlash vazifasini taqsimlangan MBBT bajaradi.

Taqsimlangan MBBT tarkibi (eng kamida) quyidagi komponentlardan tashkil topgan bo'lishi lozim:

- o tarmoq tizimini tashkil qiluvchi kompyuter ishchi stantsiyalari (saytlar yoki tugunlar). Taqsimlangan MB asbob-uskunalariga bog'liq bo'lmisligi lozim;

- o har bir ishchi stantsiyaning tarmoq uskunalari va dasturiy ta'minoti komponentalari;

- o tarmoq komponentalari barcha saytlarning bir-biri bilan o'zaro hamkorlikda harakat qilishini va qiymat almashishini ta'minlaydi. Bu komponentlar (kompyuterlar, operatsion tizimlar, tarmoq va h-zolar) turli xil muassasalar tomonidan taqdim qilinadi. Shuning uchun, taqsimlangan MB funktsiyalari turli xil platformalarda ishlashi maqsadga muvofiq bo'ladi;

- o qiymatlarni bir ishchi stantsiyadan boshqasiga o'tkazuvchi kommunikatsiya qurilmalari. Taqsimlangan MBBT

kommunikatsiya qurilmalariga bog'liq bo'lmasligi lozim, ya'ni u turli xil kommunikatsiya qurilmalarini qo'llab-quvvatlashi lozim;

- o so'rov bajariladigan har bir kompyuterda mavjud bo'lgan va dasturiy komponentni ifodalovchi tranzaktsiya protsessori (transaction processor, TR). Tranzaktsiya protsessori (lokal va uzoqdagi) so'rov ilovalari ma'lumotlarini qabul qiladi va unga ishlov beradi. Tranzaktsiya protsessorini ilovalar protsessori (application processor, AR) yoki tranzaktsiya menedjeri (transaction manager, TM) deb ham atashadi;

- o saytda joylashgan qiymatlarni chaqirib olish har bir kompyuterda mavjud bo'lgan va dasturiy komponentni ifodalovchi ma'lumotlar protsessori (data processor, DP). Ma'lumotlar protsessorini ma'lumotlar menedjeri (data manager, DM) deb ham atashadi. Ma'lumotlar protsessori hatto markazlashgan MBBTni ifodalashi mumkin.

7.8.1-jadval

Taqsimlangan MBBT ning afzalliklari va kamchiliklari

Afzalliklari	Kamchiliklari
Tuzilmaning akslantirilishi	Murakkablikning oshishi
Alohida-alohidalik va lokal avtonomlik	Narxning oshishi
Qiymatlarga kirish imkonini oshishi	Himoyalani muammosi
Ishonchlikning oshishi	Qiymatlar yaxlitligini nazorat qilishning qiyinlashuvi
Unumdorlikni oshishi.	Standartlarning mavjud emasligi
Iqtisodiy foyda	Tajribaning yo'qligi
Tizimning modulliligi	MB ga ishlov berish protseduralarining murakkablashuvi

Ma'lumotlar hamda ularga ishlov berish funksiyalari kompyuter tarmoqlari tizimlarining bir nechta saytlarlariga joylashtirilgan bo'lib, ushbu ma'lumotlarni saqlash va mantiqan bog'langan ma'lumotlarga ishlov berish funksiyalari ishini

boshqarish vazifasini taqsimlangan MBBT bajaradi. Foydalanuvchi MB qaysi uzelda joylashganini bilmaydi.

Taqsimlangan MBBT quyidagi funktsional imkoniyatlarga ega bo'lishi lozim:

- foydalanuvchi yoki amaliy dasturlar hamda taqsimlangan MB doirasida boshqa MBBT bilan bog'lanishni ta'minlovchi amaliy interfeys;
- so'rovlarni tahlil qilishda ishonchlilik tekshiruvi;
- So'rovni, qaysi komponentlari lokal, qaysilari esa taqsimlangan ekanligini aniqlash uchun o'zgartirish;
- kirishning eng yaxshi strategiyasini aniqlash kafolati bo'yicha so'rovni optimallashtirish;
- qiymatlarning lokal va uzoqdagi qismlarini aniqlash imkonini beruvchi akslantirish;
- qiymatlarning doimiy saqlanadigan o'rnidan o'qish/yozishni ta'minlovchi kirish/chiqish interfeysi;
- qiymatlarni foydalanuvchiga taqdim qilish yoki amaliy dasturga uzatish uchun tayyorlash formatlari;
- xavfsizlik, ya'ni qiymatlarni lokal va uzoqdagi MBlar uchun maxfiylikni ta'minlash;
- ishdan chiqqan hollarda qiymatlarni tiklash va kirish imkonini yaratishni kafolatlash uchun zahira nusxalarni olinishi;
- bir vaqtning MBBTning MBlari qismlariga kirish va qiymatlar yaxlitligi kafolatini ta'minlovchi parallel bajarilishni boshqarish;
- qiymatlarni bir turgun holatdan boshqasiga o'tishini ta'minlovchi tranzaktsiyalarni boshqarish. Bundan tashqari lokal va uzoqdagi tranzaktsiyalarni sinxronlash hamda bir necha taqsimlangan segmentlarga berilgan tranzaktsiyalar ham qamrab olinadi [2].

Nazorat savollari

1. MBBTning asosiy xususiyatlari haqida tushuncha bering.
2. MBning "mijoz – server" arxitekturasida server funktsiyalari haqida tushuncha bering.

3. MBning "mijoz – server" arxitekturasining "fayl-server" arxitektura nisbatan afzalliklari.

4. MBning uch zvenoli (ko'p zvenoli) "mijoz – server" arxitekturasini.

5. MBBTdagi ma'lumotlar xususiyatlari haqida tushuncha bering.

6. Integrallashgan dasturlar deganda nimani tushunasiz?

7. Integrallashgan dasturlar vazifasi nimadan iborat?

8. Paketli dasturlar deganda nimani tushunasiz?

9. Paketli dasturlar vazifasi nimadan iborat?

10. MBBT samaradorligi deganda nimani tushunasiz?

11. MBBTning xususiyatlari nimalardan iborat?

12. MBBT arxitekturasini yoki yadrosi qanday modullardan tashkil topgan.

13. MBBTlarini standartlashtirish va rivojlanish istiqbollari haqida tushuncha bering?

14. MS Access haqida tushuncha bering?

15. MS Access oynasi elementlarini izohlang?

16. Oracle MBBT tarixi haqida ma'lumot bering?

17. TEMP va ROLL1 jadval fazolari haqida tushuncha bering?

18. MS SQL Server MBBT haqida tushuncha bering.

19. IBM kompaniyasi server MBBTlari haqida tushuncha bering

20. Taqsimlangan MBBT topologiyasini izohlang?

21. Taqsimlangan MBBT qanday xarakteristikalariga ega bo'lishi kerak?

22. Ma'lumotlarni taqsimlangan qayta ishlash deganda nimani tushinasiz?

23. Taqsimlangan qayta ishlash topologiyasini izohlang.

24. Taqsimlangan MBBTning asosiy komponentlari haqida tushuncha bering?

25. Taqsimlangan MBBTlarning afzalliklari va kamchiliklari.

26. Taqsimlangan MBBTlar qanday funktsional imkoniyatlarga ega bo'lishi kerak ?

VIII. SQL TILI

8.1. SQL tili haqida tushuncha

SQL tili tarixi. *SQL (Structured Query Language)* — Strukturalashgan so'rovlar tili — relyatsion MB bilan ishlashda qo'llaniladigan so'rovlar tilidir.

Relyatsion modelning tarixi (va bilvosita SQL tarixi ham) 1970 yil E.F.Koddni (bu paytda u IBM korporatsiyasining San Xosedagi tadqiqot markazida ishlagan) maqolasi chiqqan davrdan boshlanadi. 1974 yil shu laboratoriyada ishlovchi D. Chamberlen "Structured English Query Language" yoki SEQUEL deb nomlangan tilni e'lon qiladi. 1976 yil bu tilning qayta ishlangan SEQUEL/2 versiyasi yaratildi va u rasmiy ravishda SQL deb atalgan. Hozirgi kunda SQL qisqartmasini ba'zilar 'sikvel' deb talaffuz etadi. Biroq rasmiy ravishda u "es-kyu-el" deb o'qilishi kerak.

SQL tili relyatsion algebra paydo bo'lgandan keyin paydo bo'ldi va uning birinchi prototipi IBM Research kompaniyasi tomonidan 70 yillar oxirida yaratilgan. Bu til birinchi IBM System R nomli MBBT tarkibiga kiritilgan. Keyinchalik bu til ko'pgina tijorat MBBT tarkibida qo'llanilgan va keng tarqalganligi sababli vaqt o'tishi bilan relyatsion MBBT larda ma'lumotlar ustida amallar bajaruvchi tillarning norasmiy standarti bo'lib qoldi. SQL tilining birinchi rasmiy standarti 1989 yil qabul qilingan. Ko'pgina MBBT lar ushbu standartni qo'llab – quvvatlaydi. Biroq MB bilan bog'liq axborot texnologiyalarining rivojlanishi va ba'zi talablarning paydo bo'lishi birinchi SQL standartini qayta ishlash va kengaytirishni taqozo etdi.

1992 yil oxirida SQL tilining yangi xalqaro standarti (SQL/92 yoki SQL2) qabul qilindi. Unda ham ba'zi kamchiliklar aniqlangan, biroq shunga qaramasdan SQL/89 ga nisbatan aniq va to'liqroq hisoblanadi. Ko'pgina MBBT ishlab chiqaruvchilar

o'z mahsulotlarini SQL2 standartini qanoatlantiradigan qilib o'zgartirdilar.

1999 yil SQL3 deb atalgan yangi standart paydo bo'ldi. Agar SQL1 va SQL2 standartlari bir-biridan miqdor jihati bilan farq qilgan bo'lsa, SQL3 standarti sifat jihatlari bilan farqlanadi. SQL3 ga murakkab tuzilmaga ega ma'lumotlar tipini ishlatish imkonini beradigan yangi ma'lumotlar tipi kiritilgan. Bu tipni ob'ektga mo'ljallanganlik darajasi yuqori hisoblanadi. SQL tilini to'la qonli an'anaviy dasturlash tillari tarkibiga kiritib bo'lmaydi. Chunki, unda dastur bajarilishini boshqaruvchi va boshqa ko'pgina an'anaviy operatorlar yo'q. Unda faqat MBda saqlanayotgan ma'lumotlarga murojaat qiluvchi operatorlar mavjud. SQL tili o'rganish uchun juda oson. SQL:

- o bu noprotsedura til. Shuning uchun, unda ma'lumotni qanday olish emas, balki qanday ma'lumot olish kerakligi ko'rsatiladi. Boshqacha aytganda, SQL tili ma'lumotlarga murojaat usulini ko'rsatishni talab etmaydi. Boshqa zamonaviy tillar kabi SQL tili operatorlarning mustaqil formatiga ega. Ya'ni, operatorlarni yozishda operatorlarni alohida elementlari ekrandagi ma'lum o'rinlarda joylashishi bilan bog'liq emas.

- o buyruqlar ingliz tilining odatdagi so'zlaridan iborat kalit so'zlardan tashkil topgan, masalan, CREATE TABLE (jadval yaratish), INSERT (kiritish), SELECT (tanlash). Masalan, Staff jadvalini yaratish buyrug'i quyidagicha bo'ladi:

```
CREATE TABLE Staff (staffNo VARCHAR(S), IName  
VARCHAR(15), salary DECIMAL(7,2));
```

Axborot texnologiyalari sanoatida SQL tili ommaviy tarqalishi jihatidan eng oldingi o'rinlarda turadi. IBM kompaniyasining ikkinchi darajali tadqiqot loyihasining bajarilishi natijasida paydo bo'lgan SQL tili hozirgi vaqtda MBBTlar faktori sifatida keng tarqaldi.

Xo'sh, SQL nima? Nima uchun u bunchalik muhim?

U nima ish qiladi va u qanday ishlaydi? Agarda SQL -

haqiqiy standart bo'lsa, nima uchun MBning juda ko'p versiyalari va dialektlari mavjud. SQL Server, Oracle, Interbase, Paradox MBBTlar bir - biridan nimasi bilan farq qiladi? SQL haqiqatan ham kompyuterlar va lokal tarmoq uchun shunchalik muhimmi?

SQL tilining rivojlanish tarixi:

1970 Doktor Kodd relyatsion MB modeleni yaratdi.

1974 IBM kompaniyasi System/R loyihasini ishlashga kirishdi.

1974 SEQUEL tili haqidagi birinchi maqola chiqdi.

1978 System/R loyihasinining tajribaviy tadbig'i.

1979 Oracle kompaniyasi birinchi relyatsion MBBTni yaratdi.

1981 Relational Technology kompaniyasi Ingres relyatsion MBBT ni yaratdi.

1981 IBM kompaniyasi SQL/DS relyatsion MBBT ni yaratdi.

1982 ANSI SQL tilini standartlash ko'mitasini tashkil qildi.

1983 IBM kompaniyasi DB2 relyatsion MBBTni yaratishini e'lon qildi.

1986 ANSI SQL1 standartini qabul qildi.

1986 Sybase kompaniyasi tranzaktsiyalarga ishlov beruvchi relyatsion MBBT yaratdi.

1987 ISO SQL1 standartini qo'llab-quvvatladi.

1988 Microsoft va Ashton-Tate kompaniyalari OS/2 operatsion tizimi uchun SQL Server relyatsion MBBTni yaratishini e'lon qildi.

1989 Unumdorlikni aniqlovchi TPC (TPC-A) birinchi test matbuotda e'lon qilindi.

1990 Unumdorlikni aniqlovchi TPC-B test matbuotda e'lon qilindi.

1991 SQL Access Group kontsertsiumi MBga kirish spetsifikatsiyasini chop qildi.

1992 Microsoft kompaniyasi ODBC protokoli spetsifikatsiyasini chop qildi.

1992 ANSI SQL2 standartini qabul qildi.

1992 Unumdorlikni aniqlovchi TPC-C (OLTP tizimi uchun) test matbuotda e'lon qilindi.

1993 Ma'lumotlarni saqlash xizmatlarining birinchi tizimi topshirildi.

1993 ODBC protokolini qo'llovchi birinchi dasturiy mahsulotlar topshirildi.

1994 Unumdorlikni aniqlovchi TPC-D (qarorlar qabul qilish tizimlari uchun) test matbuotda e'lon qilindi.

1994 Parallel ishlovlarni qo'llovchi server MB topshirildi.

1996 OLAP MB ga ruxsat va OLAP tizimlari unumdorligini aniqlovchi test API-funktsiya standarti chop qilindi.

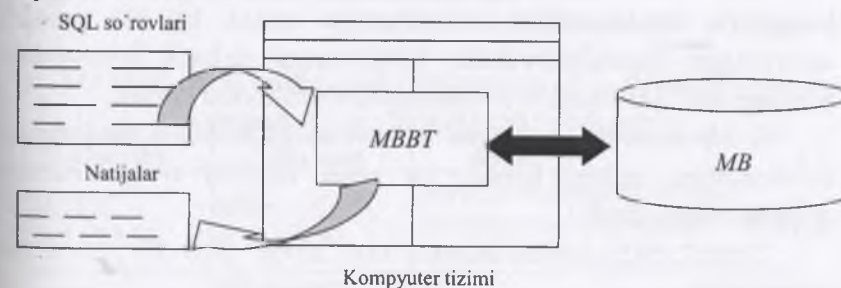
1997 IBM kompaniyasi boshqa platformalarda ishlovchi DB2 Universal Database relyatsion MBBT ni yaratdi.

1997 Yetakchi MBBT ishlab chiqaruvchilar Java-texnologiyasini qo'llab-quvvatlashlarini e'lon qilishdi.

1988 Microsoft kompaniyasi korporativ MBni qo'llovchi Windows NT platformasi uchun SQL Server 7 MBBTni yaratdi.

1998 Oracle kompaniyasi Internet muhitida ishlashni ta'minlovchi Oracle 8i MBBTni yaratdi.

SQL tili. SQL - kompyuter MB da saqlanuvchi ma'lumotlarni qayta ishlash va o'qish uchun mo'ljallangan instrument bo'lib, u faqat relyatsion MB bilan ishlaydi. Barcha relyatsion MBBTlar SQL tilini tushunadi.



8.1.1-rasm. SQL tilida ishlash

Bu sxemaga muvofiq hisoblash tizimi muhim ma'lumotlar saqlanuvchi MBga ega bo'ladi. Agarda foydalanuvchi MBdan ma'lumotlarni o'qimoqchi bo'lsa, u buni MBBTdan SQL yordamida so'raydi. MBBT so'rovga ishlov beradi, talab qilingan ma'lumotlarni topadi va uni foydalanuvchiga uzatadi.

Ma'lumotlarni olish yoki ma'lum bir shartlar buyicha ishlov berilgan ma'lumotni MBdan olish buyrug'i so'rov deb aytiladi.

SQL bir qator imkoniyatlarga ham ega, ya'ni:

1. *Ma'lumotlar tuzilmasini tuzish.* SQL foydalanuvchilarga MB tuzilmasini tuzish, o'zgartirish hamda MB elementlari o'rtasida aloqalarni o'rnatish imkoniyatini beradi.

2. *Ma'lumotlarni o'qish.* SQL foydalanuvchi yoki dasturga MB da saqlanuvchi ma'lumotlarni o'qish va ulardan foydalanish imkonini beradi.

3. *Ma'lumotlarni qayta ishlash.* SQL foydalanuvchiga yoki dasturga MBni o'zgartirish, ya'ni unga yangi ma'lumotlar qo'shish, mavjud ma'lumotlarni o'zgartirish va o'chirish imkonini beradi.

4. *MBni himoyalash.* SQL yordamida MB foydalanuvchilarini, undagi ma'lumotlarni o'qish va o'zgartirish imkoniyatlarini chegaralab qo'yish mumkin. Bu orqali ruxsat berilmagan foydalanuvchilardan axborotlar himoyalanadi.

5. *Ma'lumotlardan birgalikda foydalanish.* SQL ma'lumotlardan birgalikda foydalanishni koordinatsiya qiladi, bu esa paralel ishlayotgan foydalanuvchilar bir-birlariga xalaqit bermasdan MBdagi ma'lumotlardan foydalanishlari imkonini beradi.

6. *Ma'lumotlar yaxlitligini ta'minlash.* SQL MBni yaxlitligini ta'minlashga imkon beradi va unda noo'rin o'zgartirishlar qilishni oldini oladi.

Demak, SQL MBBT bilan o'zaro aloqa qiluvchi yetarlicha kuchli tildir.

SQL tilining vazifasi. SQL tili foydalanuvchining relyatsion MB bilan muloqat qilishi uchun mo'ljallangan bo'lib, quyidagi 3 ta qismdan iborat:

DDL (Data Definition Language) – ma'lumotlarni aniqlash tili. MBni (jadvallarni, indekslarni va h.k.) yaratish va uning sxemasini tahrirlash uchun mo'ljallangan.

8.1.1-jadval

Ma'lumotlarni aniqlash operatorlari DDL
(ma'lumotlarni aniqlash tili)

Operator	Ma'nosi	Amal
CREATE TABLE	Jadval yaratish	MBda yangi jadval yaratish
DROP TABLE	Jadvalni o'chirish	MBdan jadvalni o'chirish
ALTER TABLE	Jadvalni o'zgartirish	Mavjud jadval tuzilmasini o'zgartirish yoki joriy jadval uchun o'rnatilgan butunlik cheklanishlarini o'zgartirish
CREATE VIEW	Tasvir yaratish	Biror SQL –so'roviga mos virtual jadvalni yaratish

DCL (Data Control Language) – ma'lumotlarni boshqarish tili. Foydalanuvchilarning MB ob'ektlariga murojaatini chegaralash operatorlaridan iborat.

8.1.2-Jadval

Ma'lumotlarni boshqarish operatorlari DCL (Data Control Language) – ma'lumotlarni boshqarish

Operator	Ma'nosi	Amal
ALTER DATABASE	MBni o'zgartirish	Butun MBga tegishli bo'lgan ob'ektlar va cheklanishlar to'plamini o'zgartirish
ALTER DBAREA	MB saqlash sohasin o'zgartirish	Avval yaratilgan saqlash sohasini o'zgartirish
ALTER	Parolni	Butun MB parolini o'zgartirish

PASSWORD	o'zgartirish	
CREATE DATABASE	MBni yaratish	Barcha parametrlarini ko'rsatgan holda yangi MBni yaratish
CREATE DBAREA	Saqlash sohasini yaratish	Yangi saqlash sohasini yaratish va unda ma'lumotlar joylashtirishga ruxsat etishi
DROP DATABASE	MBni o'chirish	Mavjud MBni o'chirish (bunday amal bajarish vakolatiga ega bo'lgan foydalanuvchilar uchun)
DROP DBAREA	MB saqlash sohasini o'chirish	Mavjud saqlash sohasini o'chirish (agar unda joriy vaqtda faol ma'lumotlar joylashmagan bo'lsa)
GRANT	Vakolat berish	MB ba'zi ob'ektlari ustida bir qator amallarni bajarish huquqini berish
REVOKE	Huquqdan mahrum qilish	Biror ob'ektga yoki ob'ekt ustida bajariladigan ba'zi amallarni bajarish huquqidan mahrum qilish

DML (Data Manipulation Language) – ma'lumotlarni qayta ishlash tili. MB jadvallariga o'zgartirishlar kiritish uchun mo'ljallangan.

8.1.3 -jadval

Ma'lumotlarni manipulyatsiyalash operatorlari DML (Data Manipulation Language)

Operator	Ma'nosi	Amal
DELETE	Satrni o'chirish	Fil'tr shartini qanoatlantiruvchi muvofiq bir yoki bir necha satrni o'chiradi.
INSERT	Satrni qo'yish	Jadvalga bitta satrni qo'yish. Bu operatorning boshqa modifikatsiyalarida biror jadvalning bir necha satri yoki so'rov natijasi jadvalga qo'yilishi mumkin.
UPDATE	Satrni yangilash	Bitta yoki fil'tr shartini qanoatlantiruvchi bir necha ustunlaridagi bitta yoki ko'proq ustunlardagi qiymatlarni almashtirish

MB bilan ishlovchi ixtiyoriy til foydalanuvchiga quyidagi imkoniyatlarni yaratishi lozim:

- o tuzilmasini to'la tavsiflagan holda MBni va jadvallarini yaratish;

- o ma'lumotlar ustida manipulyatsiya amallarini bajarish, masalan, jadvallarga ma'lumotlarni kiritish, tahrirlash, va o'chirish;

- o oddiy va murakkab so'rovlarni bajarish.

Bundan tashqari, MB bilan ishlovchi til yuqoridagi amallarni bajarish uchun foydalanuvchilardan kam urinishlarni talab qilishi, hamda buyruqlarining sintaksisi va tuzilishi o'rganish uchun oson va tushunarli bo'lishi kerak. Nihoyat, bu til universal bo'lishi kerak. Bu bir MBBTdan boshqasiga o'tganda buyruqlarni bir xil tuzilmasi va sintaksisidan foydalanishni ta'minlaydi. SQL tili bu talablarni barchasini qanoatlantiradi.

SQL turlari yoki rejimlari. Hisoblash jarayonini boshqarish uchun SQL tili ikki usulda qo'llanilishi mumkin. Birinchi usulda interaktiv ishlash nazarda tutiladi. Bunda foydalanuvchi SQL operatorlarini terminaldan beradi. Ikkinchi usulda protsedurali tildagi dasturga SQL tili operatorlari kiritiladi. Bu usullar ba'zi adabiyotlarda MB bilan ishlash texnologiyasi yoki rejimi yoki SQL turlari deb ataladi.

Interaktiv rejimda MB bilan ishlash uchun foydalanuvchi muloqat rejimida ishlaydi, ya'ni SQL tilidagi so'rovni kiritadi va natijani oladi va h.k.

Ikkinchi usulda SQL so'rovlari boshqa dasturlash tillarida yaratilgan dastur tarkibiga kiritiladi. Natijada, MB bilan amaliy dasturlar orqali ishlash ta'minlanadi. Bu erda qo'shimcha dasturiy vosita kerak bo'ladi. U dasturlash tili bilan SQL operatorlari o'rtasidagi interfeysni ta'minlab beradi.

So'rov SQL tili operatorlari yordamida yaratiladi. Operatorlar so'zlar deb ataluvchi alohida ma'noli qismlardan tashkil topgan bo'lib, uning sintaksisi SQL tilining standartida belgilab berilgan.

8.1.4 -jadval

DQL (Data Query Language) so'rov tili

Operator	Ma'nosi	Amal
SELECT	Satrni tanlash	Relyatsion algebraning barcha amallarini bajaruvchi operator bo'lib so'rovga mos natijaviy jadvalni hosil qiladi.

SQL tilida "jadval", "satr" va "ustun" atamaları ishlatiladi.

Tranzaktsiyalarni boshqarishda 8.1.5-jadvalda ko'rsatilgan operatorlar islatiladi.

8.1.5 jadval

Tranzaktsiyalarni boshqarish operatorlari

Operator	Ma'nosi	Amal
COMMIT	Tranzaktsiyani tugatish	Tranzaktsiyani tashkil qiluvchi, ma'lumotlarni qayta ishlovchi murakkab va o'zaro bog'langan amallarni tugatish
ROLLBACK	Tranzaktsiyani bekor qilish	Tranzaktsiya bajarilishi natijasida yuz bergan o'zgarishlarni bekor qilish
SAVEPOINT	Tranzaktsiya bajarilishida oraliq nuqta saqlash	MBni oraliq holatini saqlash. Bu keyinchalik shu holatga qaytish uchun zarur bo'ladi.

Nazorat savollari

1. MBBT bilan ishlashda SQL tilining roli haqida tushuncha bering?
2. SQL tilining asosiy vazifasi nimalardan iborat?
3. SQL tili MBda boshqarish vazifasini bajaradimi?
4. SQL tili barcha MBBTlarida ishlatiladimi?
5. SQL tili va uning yaratilish tarixi haqida tushuncha bering.
6. SQL tilining dialektlari haqida tushuncha bering.
7. SQLni ishlash sxemasi haqida tushuncha bering.
8. SQL tilining qanday funktsional imkoniyatlari mavjud.
9. SQL tili bajaradigan funktsiyalari.
10. SQL tili imkoniyatlari

11. Ma'lumotlarni aniqlash operatorlari DDL haqida tushuncha bering.

12. DQL - so'rov tili haqida tushuncha bering.

13. Ma'lumotlarni manipulyatsiyalash operatorlari DMP haqida tushuncha bering [9].

8.2. Ma'lumotlar bazasini yaratish

Nomlar. MBdagi har bir ob'ektga takrorlanmas nom beriladi. Nomlar SQL operatorlarida ishlatiladi va operator MBdagi qaysi ob'ekt ustida amallar bajarishi lozimligini ko'rsatadi. ANSI/ISO standartida jadvallar, ustunlar va foydalanuvchilar nomga ega bo'lishi belgilangan. Ko'pgina SQLni qo'llab quvvatlovchi MBBTlarida "birlamchi kalit—ikkilamchi kalit" munosabatlari, protseduralar, funktsiyalar va triggerlar kabi ob'ektlar ham nomlanadi.

Jadval nomlari. Agarda operatorlarda jadval nomi ko'rsatilgan bo'lsa, SQL operatorlar orqali ko'rsatilgan jadvalga murojaat qiladi. Boshqa foydalanuvchilar tomonidan aniqlangan jadvalga murojaat qilish uchun jadvalning to'liq nomidan foydalanish kerak bo'ladi. To'liq nom egasining nomi va nuqta bilan ajratilgan jadval nomidan tashkil topadi. Masalan, SAM nomli foydalanuvchining BIRTHDAYS jadvali to'liq nomi quyidagicha yoziladi:

SAM.BIRTHDAYS

Jadvalning to'liq nomidan, SQLni barcha operatorlarida, uning oddiy nomi o'rnida foydalanish mumkin.

Ustun nomlari. Agarda operatorga ustun nomi berilsa, SQL bu operatorda ko'rsatilgan jadvallarning qaysi birida, ushbu ustun mavjudligini o'zi aniqlaydi. Agarda operatorda turli jadvallarning bir xil nomdagi ikkita ustuni ishlatilsa, u holda bu ustunlarni to'liq nomidan foydalanishimizga to'g'ri keladi. Ustunning to'liq nomi nuqta bilan ajratilgan ustunni o'zida saqlovchi jadval nomi va ustun nomi(oddiy nomi)dan iborat. Masalan, SALESREPS jadvalining SALES ustuni to'liq nomi quyidagicha aniqlanadi: SALESREPS.SALES.

Ma'lumotlar tipi. Relyatsion MB da har bir ma'lumot tipiga muvofiq saqlanadi. ANSI/ISO SQL1 va SQL2 standartlarida berilgan tiplar ro'yxati quyida keltirilgan (8.3.1-jadval).

SQL1 da quyidagi tiplar ishlatiladi:

- O'zgarmas uzunlikdagi belgili satrlar – bu ma'lumotlar
- tipiga mansub bo'lgan ustunlarda odatda, insonlar, kompaniyalar nomlari, manzillar va boshqalar saqlanadi.
- Butun sonlar. Bunday tipdagi ustunlarda sonlar saqlanadi. Butun sonlar bundan tashqari identifikatorlarni saqlash vazifasini ham bajaradi.
- Kasrli butun sonlar. Bunday ma'lumotlar tipidagi ustunlarda kasr qism aniq hisoblanishi kerak bo'lgan kasr sonlar saqlanadi. Bundan tashqari bu ustunlarda pulli qiymatlar ham saqlanadi.
- Haqiqiy sonlar. Bunday tipdagi ustunlarda taxminiy sonlarni saqlash mumkin. Bunday tiplar orqali juda katta diapazondagi satrlar saqlanadi, lekin hisoblashlarda xatolik kelib chiqishi mumkin.

8.3.1-jadval

Tip	Tavsifi
CHAR (uzunlik)	o'zgarmas uzunlikdagi belgilar satri
CHARACTER(uzunlik)	
VARCHAR(uzunlik)	o'zgaruvchan uzunlikdagi belgilar satri
NCHAR/(uzunlik)	o'zgarmas uzunlikdagi lokallashtirilgan simvollar satri. Milliy alfavitni qo'llaydi.
INTEGER	butun son
INT	
SMALLINT	kichik butun son
BIT(uzunlik)	o'zgarmas uzunlikdagi bitlar satri
BIT VARYING(uzunlik)	o'zgaruvchan uzunlikdagi bitlar satri
NUMERIC(aniqlik, daraja)	masshtabli butun son
DESIMAL(aniqlik, daraja)	kasrli son
DEC(aniqlik, daraja)	
FLOAT(aniqlik, daraja)	

REAL	kichik aniqlikdagi kasr son
DOUBLE PRECISION	yuqori aniqlikdagi kasr son
DATE	kalendar, kun
TIME(aniqlik)	vaqt
TIME STAMP(aniqlik)	kun va vaqt
INTERVALSQL SQL	vaqt intervali

Qo'shimcha ma'lumotlar tiplari. Ko'pgina MBBTlarida SQL1 standartida aniqlangan tiplardan tashqari ko'pgina boshqa tiplar ham mavjud. Quyida ulardan asosiylari ko'rsatilgan:

- O'zgaruvchi uzunlikdagi belgili simvollar uchun ko'pgina MBBTlarda VARCHAR tipi aniqlangan. Bu tipli ma'lumotlarda berilgan diapazonda uzunligi o'zgaruvchi belgili satrlar saqlanadi. SQL1 standartida esa o'zgarmas uzunlikdagi satrlar aniqlangan edi. Ular o'ng tomondan bo'sh joy bilan to'ldiriladi.

- Pul kattaliklari. Ko'pgina MBBTlarda MONEY yoki CURRENCY tipida pul kattaliklari mavjud. Pul kattaliklarini qo'llanishini asosiy sababi ma'lumotlarni ekranga belgilangan formatda chiqarishdir.

- Kun va vaqt. Kun va vaqt kattaligidagi qiymatlarni qo'llab—quvvatlash turli MBBTlarda keng tarqalgan. Bu ma'lumotlar tipidagi qiymatlar ustida turli operatsiyalarni bajarish mumkin. SQL2 standartida DATE, TIME, TIME STAMP, INTERVAL kabi ma'lumotlar tiplari aniqlangan. Ular orqali vaqt sekundning o'ndan bir qismigacha aniqlikda ifodalanadi.

- Mantiqiy berilganlar tipi. Ayrim MBBTlari mantiqiy tiplarni qo'llab—quvvatlaydi (TRUE yoki FALSE)

Ma'lumotlar tiplarini turlicha aniqlash. SQL qo'llaniladigan turli MBBTlarda ma'lumotlar tiplarini turlicha berilishi, dasturlarning bir MBBTidan ikkinchisiga ko'chirilishiga xalaqit beradi. Ma'lumotlar tiplarini turlicha berilishiga misol qilib, kun va vaqtni olish mumkin. Masalan, DB 2 MBBTda quyidagi 3 ta ma'lumotlar tiplari mavjud:

- DATE - bu tip oy kunni ifodalaydi.

- TIME – bu tipda sutka vaqtlari ifodalanadi.masalan, “12:30 p.m”;
- TIME STAMP – bu tipda vaqtning aniq holati nanosekundgacha aniqlikda ifodalanadi.

Aniq vaqt va holatlarni satrli o'zgarmaslar orqali ham ifodalash mumkin. Bundan tashqari vaqt qiymatlari ustida arifmetik operatsiyalarni ham bajarish mumkin.

MYSQL tilida jadval maydonlarining quyidagi tiplari mavjud :

8.3.2-jadval

Tip	Tavsifi
Butun sonli ma'lumotlar tiplari	
TINYINT	-128 dan +127 gacha sonlarni saqlashi mumkin
SMALLINT	-32 768 dan 32 767 gacha
MEDIUMINT	-8 388 608 ... 8 388 607
INT	Diapazoni -2 147 483 648 ...2 147 483 647
BIGINT	Diapazoni -9 223 372 036 854 775 808 ... 9 223 372 036 854 775 807
Ratsional sonlar tiplari	
FLOAT	Kichik aniqlikdagi o'zgaruvchan vergulli son
DOUBLE	Ikkilangan aniqlikdagi o'zgaruvchan vergulli son
REAL	DOUBLE sinonimi
DECIMAL	Satr shaklida saqlanuvchi kasr son
NUMERIC	DECIMAL sinonimi
Binar ma'lumotlar	
TINYBLOB	255 ta simvol saqlash mumkin
BLOB	65535 simvolli satr
MEDIUMBLOB	16 777 215 gacha bo'lgan simvol saqlashi mumkin
LOB	4 294 967 295 SIMVOL
Sana va vaqt	
DATE	yyyy-mm-dd formatdagi sana
TIME	ss:mm:ss formatdagi vaqt
DATETIME	yyyy-mm-dd ss:mm:ss formatdagi sana va vaqt
TIMESTAMP	timestamp formatdagi sana va vaqt

MBni yaratish. MB loyihasi to'liq tuzib bo'lingach kompyuter xotirasida ushbu loyiha sxemasi joylashtiriladi. eng avvalo, MB biror nom bilan yaratiladi. MBni yaratish uchun quyidagi formatga ega bo'lgan SQL operatori ishlatiladi:

Create {database | schema} < fayl nomi>

[user 'foydalanuvchi nomi' [password 'parol']]

[page_size [=] butun]

[length [=] butun [page [s]]]

[Default character set belgilar_to`plami]

[<ikkilamchi fayl>];

[] – shart bo'lmagan element; { } – mumkin bo'lgan elementlar.

<ikkilamchi fayl> = file 'fayl nomi' [<ma'lumot_fayllar>]

[ikkilamchi fayl]

<ma'lumot_fayllar> = length [=] butun [page [s]] | starting [at [page]]

butun [ma'lumot_fayllar]

Bunda:

- *<fayla nomi>* – MB saqlanadigan fayl spetsifikatsiyasi;
- *user 'foydalanuvchi nomi'* – foydalanuvchi serverga bog'langanda tekshiriladi;
- *password 'parol'* – foydalanuvchi nomi bilan birga tekshiriladi;
- *page_size [=] butun* – foydalanuvchi MB o'lchovi, baytda 1024 (so'zsiz), 2048, 4096 yoki 8192;
- *default character set – None* so'zsiz holatida MBda qo'llanilayotgan belgilar to'plamini aniqlash;
- *file '<fayl nomi>'* – MB joylashadigan bir yoki bir necha fayllar nomi;
- *starting [at [page]]* – agar MB bir necha fayllarni band qilsa, u holda bu ifoda ko'rsatilgan faylda MB qaysi sahifasidan boshlab joylashishini aniqlaydi;
- *length [=] butun [page [s]]* – sahifalarda fayl uzunligi. So'zsiz holatda sahifada 75, minimum 50, maksimum holatda disk fazosi bilan chegaralanadi.

Ko'p faylli MBda eng birinchi fayl birlamchi, qolganlari esa ikkilamchi deyiladi. Masalan:

```
Create database 'D:\BD\base.gdb'  
file 'D:\BD\base.gd1' starting at page 1001  
length 500  
file 'D:\BD\base.gd2'.
```

Bu yerda 3 ta fayldan iborat, uzunligi 1000 bo'lgan birlamchi *base.gdb* fayl, uzunligi 500 bo'lgan *base.gd1* va uzunligi noaniq bo'lgan *base.gd2* fayllar bo'lgan *d:\bd\base.gdb* MB aniqlanadi.

Agar ikkilamchi fayllar uchun uzunlik ko'rsatilmagan bo'lsa, u holda u qaysi sahifadan boshlanishini ko'rsatish lozim.

Sahifa o'lchovi baytlarda ko'rsatiladi, masalan:

```
Create database 'base.gdb' page_size 4096.
```

Sahifalar o'lchovini oshishi, indekslar chuqurligi kamayishi, uzun yozuvlarni solishtirib o'qish amallarini kamayishi hisobiga MB bilan ishlashni tezlashtirish mumkin. Agar so'rovlar ko'p bo'lmagan yozuvlarni qaytarsa, sahifa to'liqligicha solishtirib o'qib olinishi va unda ko'p ortiqcha yozuvlar bo'lishi mumkinligi oqlanmagan.

8.3. DDL operatorlari

SQL ning DDL (Data Definition Language) ma'lumotlarni aniqlash tili MB ning sxema, domenlar, jadvallar, tasvirlar va indekslar kabi ob'ektlarini yaratish va o'chirish uchun ishlatiladi.

Quyida SQL da ma'lumotlarni aniqlash tilining asosiy operatorlari nomi keltirilgan.

CREATE SCHEMA	ALTER TABLE
DROP SCHEMA	DROP TABLE
CREATE DOMAIN	CREATE VIEW
ALTER DOMAIN	DROP VIEW
DROP DOMAIN	CREATE SYNONYM
CREATE TABLE	DROP SYNONYM

Bu yerda:

- o CREATE – yaratish;

- ALTER – mavjud ob'ektni modifikatsiyalash;
- DROP – mavjud ob'ektni o'chirish.
- SCHEMA - sxema.
- DOMAIN – domen.
- TABLE – jadval.
- VIEW – tasvir yoki ko'rinish.
- SYNONYM – jadval nomi uchun sinonim.

Bu operatorlar kontseptual sxemaning tarkibiga kiruvchi tuzilmalarni yaratish, tahrirlash va o'chirish uchun ishlatiladi. Ba'zi MBBTlarda quyidagi operatorlar ham mavjud:

CREATE INDEX

DROP INDEX

INDEX - ustun uchun indeks.

Jadvallar yaratish. Jadvallar CREATE TABLE buyrug'i bilan yaratiladi. Bu buyruq qatorlarsiz bo'sh jadval yaratadi. CREATE TABLE buyrug'ida jadval nomi va jadvalning ustunlari ko'rsatilgan tartibda ketma – ket beriladi. Unda har bir ustundagi ma'lumotlarning tiplari va ustunlar o'lchovi ko'rsatiladi. Har bir jadval hech bo'lmaganda bitta ustunga ega bo'lishi kerak.

CREATE TABLE komandasi sintaksisi:

CREATE TABLE <table-name >

(<column name> <data type>[(<size>)],
<column name> <data type>[(<size>)], ...);

Argument qiymati kattaligi ma'lumot turiga bog'liqdir. Agar siz maxsus ko'rsatmasangiz, tizim avtomatik qiymatni o'rnatadi.

Misol uchun sotuvchilar jadvalini yaratishni ko'rib chiqamiz:

CREATE TABLE Salepeople

(SNum integer,
SName char (10),
City char (10),
Comm decimal);

Cheklanishlarni kiritish. Ko'p hollarda ustunga kiritilgan qiymatlar bir biridan farq qilishi kerak. Agar ustun uchun UNIQUE cheklanishi o'rnatilsa, bu ustunga takrorlanuvchi qiymatlarni kiritishga urinish rad etiladi. Bu cheklanish bo'sh bo'lmaydigan (NOT NULL) deb e'lon qilingan maydonlarga qo'llanishi mumkin.

Masalan:

```
CREATE TABLE Salepeople  
  ( SNum   integer NOT NULL UNIQUE,  
    SName  char (10),  
    City   char (10),  
    Comm   decimal);
```

Qiymatlari takrorlanmaydigan maydonlar (birlamchi kalitlardan tashqari) nomzod kalitlar yoki unikal kalitlar deyiladi.

Jadval cheklanishi UNIQUE maydonlar guruhiga o'rnatilishi mumkin. Bu bir necha maydonlar qiymatlari kombinatsiyasi unikalligini ta'minlaydi. Bizning MBda har bir buyurtmachi bitta sotuvchiga biriktirilgan. Ya'ni buyurtmachilar jadvalida buyurtmachi nomeri (SNum) va sotuvchi nomeri (SNum) kombinatsiyasi unikal bo'lishi kerak. Bu cheklanishni UNIQUE (SNum, SNum) yordamida, Customers jadvalini yaratishda kiritish mumkin. Bu ustunlar uchun NOT NULL cheklanishini kiritish zarurdir.

Birlamchi kalitlar cheklanishlari. SQL birlamchi kalitlarni to'g'ridan to'g'ri birlamchi kalit (PRIMARY KEY) cheklanishi orqali ta'riflaydi. PRIMARY KEY jadvalni yoki ustunlarni cheklashi mumkin. Bu cheklanish UNIQUE cheklanishi kabi ishlaydi, jadval uchun faqat bitta birlamchi kalit (ixtiyoriy sondagi ustunlar uchun) aniqlanishi mumkin bo'lgan holdan tashqari. Birlamchi kalitlar NULL qiymatga ega bo'lishi mumkin emas.

1-misol.

```
CREATE TABLE Salepeople
```

```
( SNum integer NOT NULL PRIMARY KEY,
  SName char (10),
  City char (10),
  Comm decimal);
```

Maydon qiymatlarini tekshirish (CHECK cheklanishi). CHECK cheklanishi jadvalga kiritilayotgan ma'lumot, qabul qilinishidan oldin berilgan shartni qanoatlantirishini tekshirish uchun tekshirilayotgan shartni kiritishni ta'minlaydi.

2-misol. Salepeople jadvali Comm ustuniga kiritilayotgan qiymat 1 dan kichik bo'lish sharti.

```
CREATE TABLE Salepeople
( SNum integer NOT NULL PRIMARY KEY,
  SName char(10) NOT NULL UNIQUE,
  City char(10),
  Comm decimal CHECK (Comm < 1));
```

Ko'zda tutilgan qiymatlarni o'rnatish. Agar qiymat kiritilishi shart bo'lgan biror bir maydonga qiymat kiritmasdan, satrni jadvalga qo'shmoqchi bo'lsangiz, buyruq rad etiladi. Satrga eng ko'p takrorlanib kiritiladigan qiymatlarni avvaldan kiritish buyrug'ini berish mumkin. Bu, ko'zda tutilgan qiymat DEFAULT operatori orqali, ustun cheklanishi sifatida ko'rsatiladi. Masalan,

```
CREATE TABLE Salepeople
( SNum integer NOT NULL PRIMARY KEY,
  SName char(10) NOT NULL UNIQUE,
  City char(10) DEFAULT 'New York',
  Comm decimal CHECK ( Comm < 1 ));
```

Eng umumiy ko'zda tutilgan qiymat NULL qiymatdir.

Ma'lumotlar yaxlitligini ta'minlash. Jadvalning bir maydonidagi hamma qiymatlar boshqa jadval maydonida aks etsa, birinchi maydon ikkinchisiga ilova qiladi deyiladi. Bu ikki maydon orasidagi bog'liqlik ko'rsatiladi. Masalan, buyurtmachilar jadvalida har bir buyurtmachi, sotuvchilar jadvalida o'ziga biriktirilgan sotuvchiga ilova qiluvchi SNum

maydoniga ega. Bir maydon ikkinchisiga ilova qilsa tashqi kalit, u ilova qilayotgan maydon ajdod kalit deyiladi. Buyurtmachilar jadvalidagi SNum maydoni tashqi kalit, sotuvchilar jadvalidagi SNum - ajdod kalitdir.

Tashqi kalit bitta maydondan iborat bo'lishi shart emas. Birlamchi kalit kabi, tashqi kalit bitta modul sifatida qayta ishlanuvchi bir necha maydonlarga ega bo'lishi mumkin. Maydon tashqi kalit bo'lsa, u ilova qilayotgan jadval bilan ma'lum usulda bog'liqdir. Tashqi kalit har bir qiymati (satri), ajdod kalitning bitta va faqat bitta qiymatiga (satriga) ilova qilishi kerak. Bu holda tizim ilovali yaxlit holatda deyiladi. Shu bilan birga ajdod kalit qiymati tashqi kalitning bir necha qiymatlariga ilova qilishi mumkin.

Cheklanish FOREIGN KEY. SQL ilovali yaxlitlikni FOREIGN KEY yordamida ta'minlaydi. Tashqi kalit vazifasi ajdod kalitda ko'rsatilmagan qiymatlarni tashqi kalit maydonlariga kiritmaslikdir. FOREIGN KEY cheklanishi sintaksisi:

```
FOREIGN KEY <column list> REFERENCES  
      <pktable> [<column list>]
```

Birinchi ro'yxat buyruq tomonidan o'zgartiriluvchi ustunlar ro'yxatidir. pktable - bu ajdod kalitli jadval. Ikkinchi ustunlar ro'yxati bu ajdod kalitni tashkil qiluvchi ustunlardir.

Misol uchun 'Sotuvchilar' jadvaliga ilova qiluvchi tashqi kalit sifatida e'lon qilingan SNum maydoniga ega bo'lgan buyurtmachilar jadvalini yaratamiz:

```
CREATE TABLE Customers  
  ( CNum integer NOT NULL PRIMARY KEY,  
    CName char(10),  
    City char(10),  
    SNum integer,  
    FOREIGN KEY (SNum) REFERENCES Salepeople (SNum) );
```

Tasvirlarni yaratish. Jadval va jadvallardagi ma'lumotlarni olish uchun tasvirlardan foydalaniladi. Tasvir - bu bo'sh, virtual

jadvaldir. So'rov vaqtida uning ustunlari tasvirda ko'rsatilgan jadvaldan to'ldiriladi. Tasvirlarni ishlatishdan maqsad jadvallarni yoki jadvallarning ba'zi bir ustunlarini foydalanuvchilardan yashirishdir. Bundan tashqari, tasvir orqali bir necha jadvallarda berilgan ma'lumotlarni yagona jadval shaklida foydalanuvchiga taqdim qilishdir. MB jadvallari o'zgartirilsa, bu o'zgarishlar amaliy dasturlarga ta'sir qilmasligini ta'minlash, tasvirlarni ishlatish orqali amalga oshiriladi. Shuning uchun, amaliy dasturlar orqali xech qachon to'g'ridan to'g'ri jadvalga emas, balki uning tasviriga bog'lanishni amalga oshirish maqsadga muvofiq hisoblanadi.

Tasvirlar CREATE VIEW operatori bilan e'lon qilinadi va uning sintaksisi quyidagicha:

```
CREATE VIEW tasvir_nomi  
  [(ustun[,ustun] ...)]  
  AS so'rov osti  
  [WITH CHECK OPTION];
```

Masalan, Talaba jadvalidan 1 va 3 tartib raqami bilan berilgan talabalar ro'yxatini alohida Talaba1 nomli virtual jadval shaklida foydalanuvchiga taqdim qilish mumkin:

```
CREATE VIEW Talaba1 AS SELECT id, stfam, stname  
  FROM Talaba WHERE guruh='1' AND guruh='3';
```

Masalan, 8.5-mavzu 1-misoldagi student, mark_st va exam_st jadvallaridan talabalarning fanlardan olgan baholarini yagona 'Talaba_fan' nomli jadval shaklida foydalanuvchiga taqdim qilish mumkin:

```
CREATE VIEW Talaba_fan  
  SELECT E.stfam, E.stname, M.id_ex, M.mark FROM student  
  E, mark_st M  
  WHERE E.id_st = M.id_st
```

Tasvirni o'chirish:

```
DROP VIEW tasvir_nomi;
```

Jadvalni o'chirish. Jadvalni o'chirish uchun DROP TABLE buyrug'i ishlatiladi. Bo'sh bo'lmagan jadvallarni o'chirishga ruxsat berilmaydi. Shuning uchun, jadvalni o'chirishdan oldin uning qiymatlarini o'chirish talab qilinadi. Uning sintaksisi:

DROP TABLE <table name>;

Masalan: DROP TABLE Salepeople;

Jadvalni o'zgartirish. Jadvalni o'zgartirish uchun ALTER TABLE buyrug'i ishlatiladi. Bu buyruq orqali jadvalga yangi ustunlar qo'shish va mavjud ustunlarni olib tashlash, ustunlar kattaliklarini o'zgartirish, cheklashlarni qo'yish va mavjud cheklashlarni olib tashlash mumkin. Uning sintaksisi:

ALTER TABLE <table name> ADD <column name>
<data type> <size>;

Masalan:

ALTER TABLE Salepeople ADD Phone CHAR(7);

SQL tilida protsedura va triggerlar. Ba'zi bir MBBT larida protseduralar yaratishga imkon beruvchi SQL tili ishlatiladi. Bu protseduralar ham xuddi dasturlash tillaridagi protseduralar singari vazifani bajaradi. Protsedura matni avvaldan yaratiladi, kompilyatsiya qilinadi va saqlanadi. Keyin esa kerakli vaqtda chaqirilib, ishlatiladi. Protseduralar to'plami paket shaklida jamlanib saqlanadi.

Trigger ham xuddi protsedura singari yaratiladi, kompilyatsiya qilinadi va saqlanadi. Ammo, u kerakli paytda chaqirilmaydi, balki kerakli vaziyatda avtomatik ravishda o'zi ishlaydi.

Nazorat savollari

1. MB SQL tilining qaysi buyrug'i orqali yaratiladi?
2. CREATE DATABASE operatorining vazifasi haqida tushuncha bering.
3. SQL tilida ma'lumotlarning qanday tiplari mavjud?
4. SQLning DDL qismi haqida ma'lumot bering.

5.MBda jadvallar SQL tilining qaysi buyrug'i orqali yaratiladi?

6.CREATE TABLE operatorining vazifasi haqida tushuncha bering.

7.Salepeople jadvalini yaratish matnini tushuntirib bering.

8.Cheklanishlarni kiritish haqida tushuncha bering.

9.Birlamchi kalitlar cheklanishlari haqida tushuncha bering.

10.Maydon qiymatlarini tekshirish (*CHECK cheklanishi*) haqida tushuncha bering.

11.Ko'zda tutilgan qiymatlarni o'rnatish haqida tushuncha bering.

12.Ma'lumotlar yaxlitligini ta'minlash haqida tushuncha bering.

13.Cheklanish FOREIGN KEY haqida tushuncha bering.

14.SQL tilining Drop Table operatori haqida tushuncha bering.

8.4. Ma'lumotlarni qayta ishlash so'rovlari

INSERT ma'lumotlarni kiritish operatori.

INSERT INTO jadval_nomi [(<ustunlar ro'yxati >)]
VALUES (<qiymatlar ro'yxati >)

Bunday sintaksis jadvalga faqat bitta satr kiritish imkonini beradi. Agar satrdagi barcha ustunlarga qiymat kiritilayotgan bo'lsa, so'rovda barcha ustunlar nomini ko'rsatish zarur emas.

Masalan, BOOKS jadvaliga yangi kitob ma'lumotlari kiritiladi

INSERT INTO books (isbn, titl, autor, coautor, yearizd, pages)

VALUES ('5-88782-290-2', 'Аппаратные средства IBM PC. Энциклопедия', 'Guk M.', '', 2000, 816)

Bu kitob muallifi faqat bitta va muallifdosh (soavtor) mavjud emas, biroq ustunlar ro'yxatida COAUTOR ustuni ham ko'rsatilgan. Shuning uchun VALUES bo'limida bu ustunga mos

qiymatni ko'rsatish zarur. Misolda bu maydon uchun bo'sh satr (") ko'rsatilgan. Bu muallifdosh yo'qligini bildiradi. Shuningdek, bu erda aniqlanmagan NULL qiymatini ko'rsatish ham mumkin.

Satrdagi barcha ustunlarga qiymat kiritishda ustunlar ro'yxatini ko'rsatish zarur emas. Bunda faqat qiymatlar ro'yxatini ko'rsatish etarli bo'ladi. Bunday holda operator ko'rinish quyidagicha shaklda bo'ladi:

```
INSERT INTO books VALUES ('5-88782-290-2',  
'Аппаратные средства IBM PC. Энциклопедия.', Guk M.,  
'', 2000, 816)
```

Misolda keltirilgan ikkala operator ham bir xil amalni bajaradi.

Shuningdek, to'liq miqdorda bo'lmagan qiymatlarni ko'rsatish mumkin. Ya'ni, qiymatlar qatorida muallifdoshni ko'rsatmaslik mumkin, chunki joriy kitobda muallifdosh yo'q. Biroq bunda qiymat kiritiladigan ustun nomlarini quyidagicha shaklda ko'rsatish kerak bo'ladi:

```
INSERT INTO books ( isbn, titl, autor, yearizd,pages)  
VALUES ('5-88782-290-2', 'Аппаратные средства IBM PC.  
Энциклопедия'. 'Guk M.', 2000, 816)
```

Bu holda COAUTHOR ustuniga NULL qiymati yoziladi.

Agar jadvalni yaratishda ustun yoki atributga majburiy qiymat (NOT NULL) belgisi qo'yilgan bo'lsa, u holda INSERT operatorida joriy ustunning har bir satriga kiritiladigan qiymat ko'rsatilishi kerak. Shuning uchun, agar jadvalning hamma ustuni majburiy qiymatli bo'lsa, u holda har bir yangi kiritiladigan satrda barcha ustun uchun qiymat mavjud bo'lishi kerak va bunda ustunlar ro'yxatini ko'rsatish shart emas. Aks holda jadvalda kamida bitta majburiy qiymatli bo'lmagan ustun bo'lsa, u holda albatta ustunlar ro'yxatini ko'rsatish shart bo'ladi.

Qiymatlar ro'yxatida maxsus funktsiyalar va ifodalar ko'rsatilishi ham mumkin. Bunda, ushbu funktsiyalarning

qiymatlari ma'lumotlarni kiritish holatida hisoblangan bo'lishi zarur.

Ma'lumotlarni kiritish operatori birdaniga bir necha satrlarni kiritish imkoniga ham ega. Bunda, qiymatlar satri boshqa bir jadvaldan tanlab olinadi. Masalan, talabalar haqidagi jadval mavjud bo'lsin. Unda talabalarning familiyasi, adresi, uy telefoni va tug'ilgan sanasi ko'rsatilgan bo'lsin. U holda, bitta operator yordamida ularni kutubxonaning kitobxonlariga aylantirish mumkin:

```
INSERT INTO READER (FIO_studenta, Adres, Telefon, Data_rojd)
```

```
SELECT (FIO_studenta, Adres, Telefon, Den_rojd) FROM STUDENT
```

DELETE o'chirish operatori. Ma'lumotlarni o'chirish operatori jadvaldan shartni qanoatlantiruvchi bir yoki bir neta satrlarni o'chirishi mumkin.

```
DELETE FROM jadval_nomi[WHERE tanlash_sharti]
```

Agar satrlarni tanlash sharti ko'rsatilmasa, u holda jadvaldagi barcha satrlar o'chiriladi. Natijada ma'lumotlarga ega bo'lmagan bo'sh jadval hosil bo'ladi.

Agar jadvaldan oldingi sessiya natijalarini o'chirish kerak bo'lsa, u holda R1 jadvalidagi barcha satrlar o'chiriladi:

```
DELETE FROM R1
```

WHERE qismidagi shart ifodasi xuddi SELECT operatoridagi fil'trlash shartiga o'xshash bo'ladi. Bu shart jadvaldan qaysi satrlar o'chirilishi kerakligini aniqlaydi.

Masalan, talaba Maxmudova A.V. o'chirilishi kerak bo'lsa, quyidagi so'rov beriladi:

```
DELETE FROM R2 WHERE FIO = "Maxmudova A.V."
```

WHERE qismida biror shart ko'rsatilishi mumkin. Masalan, agar jadvaldan o'zlashtirmagan talabalarni o'chirish kerak bo'lsin. Oliy ta'lim qonuniga ko'ra oxirgi sessiyada ikkita va undan ortiq fandan ikki baho olgan talaba o'zlashtirmagan

hisoblanadi. U holda tanlab olish sharti ikkita va undart ko'p ikki baho olgan talabalarni va ikkita undan ko'p imtihonlarni topshirmagan talabalarni aniqlashi kerak. Bunday talabalarni aniqlash uchun R1 jadvalidan ikki baholi va baho ko'rsatilmagan satrlar tanlab olinishi, keyin olingan natija 'FIO' ustuni bo'yicha guruhlanishi kerak. Keyin har bir guruhdagi satrlar soni aniqlanadi (bu har bir talabaning olgan ikki baholari bilan topshirmagan imtihonlar sonini bildiradi) va ikkitadan ko'p satrga ega ustunlar tanlab olinadi. endi ushbu murakkab bo'lgan konstruktsiyani SQL tilida yozamiz va u sodda ko'rinishga ega bo'lishini ko'ramiz.

DELETE FROM R2 WHERE R2.FIO IN (SELECT R1.FIO FROM R1 WHERE Otsenka = 2 OR Otsenka IS NULL GROUP BY R1.FIO HAVING COUNT(*) >= 2

DELETE operatsiyasini bajarishda jadvaldan o'chiriladigan satrlar WHERE qismida ko'rsatilmasa jadvalning barcha qiymatlari o'chiriladi.

Ma'lumotlarni manipulyatsiyalash operatsiyalarining barchasi MBning butunligi tushunchasi bilan bog'langan. Manipulyatsiyalash amallari sintaktik jihatdan to'g'ri bo'lsada butunlik talablari tufayli bajarilmasligi mumkin.

Ma'lumotlarni o'zgartirish operatori (UPDATE). Ma'lumotlarni o'gartirish operatori o'zgarish yuz berganda va mos holda bu o'zgarishni MBda akslantirish uchun ishlatiladi.

UPDATE jadval_nomi **SET** ustun_nomi = yangi_qiymat [**WHERE** tanlash_sharti]

Bu yerda ham WHERE qismi DELETE operatoridagi kabi ko'rsatilishi shart emas. U DELETE operatoridagi kabi bir xil vazifani bajaradi va o'zgartirish amali bajariladigan satrlarni tanlash imkonini beradi. Agar tanlash sharti (WHERE qismi) ko'rsatilmagan bo'lsa, u holda, o'zgartirish amali jadvalning barcha satrlari uchun bajariladi.

Masalan, talaba Fozilova K. E. MBBT fanidan "2" baho oldi, keyin uni "3" bahoga qayta topshirgan bo'lsin. Bu holatga mos holda R1 jadvalini o'zgartirish amali quyidagi operator bilan amalga oshiriladi:

```
UPDATE R1 SET R1.Otsenka = 3
```

```
WHERE R1.FIO = 'Fozilova K.E.' AND R1.Distsiplina = 'MBBT'
```

Qanday holatlarda bir nechta satrlarni o'zgartirish zarurati paydo bo'ladi? Bu kam uchraydigan masala emas. Masalan, agar guruhlar jadvalidagi guruhlarini kursini bittaga oshirish zarur bo'lsa, quyidagi o'zgartirish amalini bajarish mumkin. Guruhlar jadvali quyidagicha sxemaga ega bo'lsin:

```
R4 = <Guruh, Kurs>
```

```
UPDATE R4 SET R4.Kypc = R4.Kypc + 1
```

8.5. Sodda so'rovlar

SELECT operatori. So'rov SELECT operatori yordamida bajariladi va uning sintaksisi quyidagicha:

```
SELECT [ALL/DISTINCT] <atributlar ro'yxati>/*
```

```
FROM <jadvallar ro'yxati >
```

```
[WHERE <tanlash sharti >]
```

```
[ORDER BY <atributlar ro'yxati >]
```

```
[GROUP BY <atributlar ro'yxati >]
```

```
[HAVING <shart>]
```

```
[UNION< SELECT operatorli ifoda>]
```

Kvadrat qavslarda operatorni yozishda qatnashishi shart bo'lmagan elementlar ko'rsatilgan.

Barcha so'rovlar **SELECT** kalit so'zi bilan boshlanadi. Bu so'zdan keyin so'rov natijasi sifatida qaytariladigan ma'lumot elementlarini ko'rsatish lozim. Bu elementlar vergul bilan ajratilgan holda beriladi. So'rov natijasi alohida vaqtinchalik jadvalda saqlanadi va ro'yxatni har bir elementi uchun so'rov natijalari jadvalida bitta ustun hosil qilinadi. Natija jadvali

ustunlari qaytariladigan ustunlar ro'yxati elementlari tartibi bo'yicha tartiblanadi. Qaytariladigan qiymat o'zida quyidagilarni ifodalashi mumkin.

- o Ustun nomi. So'rov uchun murojaat qilinayotgan jadval ustunlaridan biri nomi ko'rsatiladi. Bu holda bu ustunning har bir satri qiymati olinadi va so'rov natijasiga chiqariladi.

- o O'zgarmas. So'rov natijasining barcha satrlari bir xil qiymat bilan chiqariladi.

- o Ifoda. Ifodada berilgan formula orqali hisoblash bajarilib olingan qiymatlar so'rov natijasiga chiqariladi.

ALL kalit so'zi natijaga shartni qanoatlantiruvchi barcha satrlar, shuningdek takrorlanuvchi satrlar ham kirishini bildiradi.

DISTINCT kalit so'zi natijaga takrorlanuvchi satrlar kiritilmasligini bildiradi. Keyin boshlang'ich jadvaldagi atributlar ro'yxati ko'rsatiladi. Bu atributlar natijaviy jadvalga kiritiladi.

Yulduzcha (*) belgisi natijaga so'rov uchun murojaat qilinayotgan jadval ustunlarining barcha ustunlari kiritilishini bildiradi.

FROM kalit so'zidan keyin so'rov uchun murojaat qilinayotgan jadval nomi beriladi. Murakkab so'rovlarda ba'zan jadvallar bir nechta bo'lishi mumkin va ular spetsifikatorlari bilan beriladi. Spetsifikatorlar bir-biridan vergul bilan ajratiladi. Har bir jadval spetsifikatori so'rov natijasiga chiqariladigan ma'lumotlarni o'zida saqlovchi jadvalni ifodalaydi. Bunday jadvallar so'rov jadvallari deyiladi. Chunki, so'rov natijalari jadvalida saqlanadigan barcha ma'lumotlar ulardan olinadi.

WHERE tanlash ifodasi kalit so'zidan keyin jadval satrlarini tanlab olish sharti ko'rsatiladi. Bunda natijaviy jadvalga *WHERE* ifodasidagi shart rost qiymat qabul qiladigan satrlar kiritiladi.

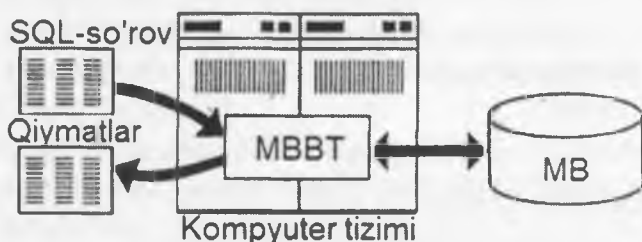
ORDER BY kalit so'zi natijaviy jadval satrlarini ko'rsatilgan ustunlar ro'yxati bo'yicha tartiblash amalini bildiradi.

GROUP BY kalit so'zidan keyin guruxlanadigan atributlar ro'yxati ko'rsatiladi.

HAVING kalit soʻzidan keyin GROUP BY kalit soʻzi yordamida tuzilgan guruhlarini har bir guruhiga qoʻyiladigan shartlar koʻrsatiladi.

FROM, WHERE va ORDER BY kalit soʻzlari SQL tilining boshqa manipulyatsiyalash operatorlarida ham shu tarzda ishlatiladi.

Soʻrovlar yaratishni misol orqali koʻrib oʻtamiz.



8.6.1-rasm. SQL soʻrovni uzatish mexanizmi

1-misol. Talabalarning fanlardan olgan baholari jadvallarini tuzing va soʻrovlarni amalga oshiring.

1. Talabalar roʻyxati jadvali

Student

student – jadval nomi;

id_st – tartib raqami, birlamchi kalit;

Stfam – talabalar familiyasi ;

Sname – talabalar nomi.

id_st	Stfam	Sname
1	Xasanov	Salim
2	Karimov	Bobur
3	Jabborov	Soli

2. Talabalar imtihon topshirgan fanlar roʻyxati jadvali

exam_st

exam_st – jadval nomi;

id_ex – tartib raqami, birlamchi kalit;

Subject – fanlar nomi;

Date – fandan imtihon topshirilgan sana.

id_ex	Subject	Date
1	Matematika	10.05.2009
2	Fizika	15.05.2009
3	Informatika	25.05.2009

3. Talabalarning imtihon topshirgan fanlardan olgan baholari jadvali

mark_st

id_ex	id_st	mark
2	2	3
3	1	2
1	1	4
3	2	5
2	1	3
1	2	2
2	3	4
1	3	5
3	3	3

mark_st – jadval nomi

id_ex – fan tartib raqami

id_st – talaba tartib raqami. Subject

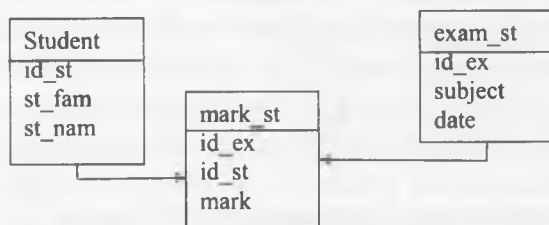
fanlar nomi

mark – talabaning fandan imtihon

bahosi

(id_ex, id_st) - birlamchi kalit.

4. Masala uchun ER-diagramma



Barcha talabalar ro'yxatini tanlash va tasvirlash.

SELECT * FROM student

yoki

```
SELECT id_st, stfam, stname FROM student
```

Agar ushbu so'rovga ORDER BY stfam ifodasi qo'shilsa, u holda ro'yxat familiya bo'yicha tartiblanadi.

```
SELECT * FROM student ORDER BY stfam
```

So'zsiz tartiblash o'sish bo'yicha bajariladi. Agar kamayish bo'yicha tartiblash kerak bo'lsa, u holda oxirgi ifodadagi atribut nomidan keyin DESC so'zi qo'shiladi..

```
SELECT * FROM student ORDER BY stfam DESC
```

Kodi birga teng bo'lgan talaba olgan baholar ro'yxatini tanlab olish va tasvirlash

```
SELECT id_st, mark FROM mark_st Where id_st = 1
```

Imtihonlardan kamida bitta 2 yoki 3 baho olgan talabalar kodini tanlab olish.

WHERE so'zidan keyin solishtirish amallarini (<, >, =, <> va h.k.) va mantiqiy operatorlar qatnashgan ifodalarni joylashtirish mumkin.

```
SELECT id_st, mark FROM mark_st WHERE (mark >= 2)  
AND (mark<=3)
```

SQL tilida shart ifodalarini tuzish uchun solishtirish va mantiqiy operatorlardan tashqari yana bir qator maxsus operatorlar qo'llaniladi. Bu operatorlar dasturlash tillarida mavjud emas. Bu operatorlar:

IN – biror qiymatlar to'plamiga tegishlilikini tekshirish;

BETWEEN – biror qiymatlar diapozoniga tegishlilikini tekshirish;

LIKE – namuna bilan mosligini tekshirish;

IS NULL – qiymat mavjudmasligini tekshirish.

IN operatori biror qiymatlar to'plamiga tegishlilikni tekshirish uchun ishlatiladi.

Quyidagi so'rov oxirgi keltirilgan misoldagi so'rov natijalarini beradi (Imtihonlardan kamida bitta 2 yoki 3 baho olgan talabalar kodini tanlab oladi)

SELECT id_st, mark FROM mark_st WHERE mark IN (2,3)
Xuddi shunday natijani BETWEEN operatoridan foydalanib olish mumkin:

SELECT id_st, mark FROM mark_st WHERE mark BETWEEN 2 AND 3

Familiyalari A harfi bilan boshlanuvchi talabalar ro'yxatini tanlab olish.

Bunday holatda LIKE operatoridan foydalanish qulay hisoblanadi. LIKE operatori faqat simvolli maydonlar uchun qo'llaniladi va maydon qiymati operatorida ko'rsatilgan namunaga mosligini tekshirish imkonini yaratadi. Namuna quyidagi maxsus simvollardan tashkil topadi:

_ (tagiga chizish belgisi) – bitta ixtiyoriy simvolni bildiradi;

% (foiz belgisi) – ixtiyoriy miqdordagi simvollar ketma – ketligini bildiradi.

SELECT id_st, stfam, stname FROM student WHERE stfam LIKE 'A%'

Agregat funksiyalar. Ko'pincha ustunlardagi minimal, maksimal yoki o'rtacha qiymatlarni hisoblashga to'g'ri keladi. Masalan, keltirilgan jadvallarda o'rtacha bahoni hisoblash mumkin. Bunday hisoblashni bajarish uchun SQL tilining maxsus agregat funksiyalari mavjud:

MIN() – ustundagi minimal qiymat;

MAX() – ustundagi maksimal qiymat;

SUM() – ustundagi qiymatlar summasi;

AVG() – ustundagi qiymatlarni o'rtachasi;

COUNT() – ustundagi NULL dan farqli qiymatlar miqdori.

Quyidagi so'rov talabalar olgan baholarning o'rtachasini hisoblaydi.

SELECT AVG(mark) FROM mark_st

Albatta, agregat funksiyalarini WHERE so'zi bilan birgalikda qo'llash mumkin. Quyidagi so'rov 100 kodli

talabaning imtihonlardan olgan baholarining o'rtachasini hisoblaydi:

SELECT AVG(mark) FROM mark_st WHERE id_st = 100

Navbatdagi so'rov 10 kodli talabaning o'rtacha bahosini hisoblaydi:

SELECT AVG(mark) FROM mark_st WHERE id_ex = 10

SQL tili ko'rib o'tilgan mexanizmiga qo'shimcha ravishda agregat funksiyalarni to'la jadval uchun emas, balki guruhlangan qiymatlar uchun qo'llash imkonini beruvchi vosita, GROUP BY mavjud. Masalan, har bir talabani imtihonlardan olgan baholarining o'rtachasini aniqlash mumkin. Buning uchun quyidagi so'rov beriladi:

SELECT id_st, AVG(mark) FROM mark_st GROUP BY id_st

Bu imkoniyat ham odatdagidek WHERE so'zi bilan birgalikda qo'llanilishi mumkin. Bu so'rovni bajarishda MBBT avval jadvaldan WHERE ifodasidagi shartni qanoatlantiruvchi satrlarni tanlab oladi, keyin tanlangan satrlarni guruhlash va agregatlash amalini bajaradi.

Quyidagi so'rov har bir talabaning 100 kodli imtihondan olgan baholarining o'rtachasini aniqlaydi.

**SELECT id_st, AVG(mark) FROM mark_st
WHERE id_ex = 100 GROUP BY id_st**

Ko'rib turganimizdek guruhlash bittadan ko'p maydon bo'yicha bajarilishi mumkin. GROUP BY sektsiyasiga ega bo'lgan so'rovlar uchun quyidagi muhim cheklanish mavjud: bunday so'rovlar natijasi tarkibiga guruhlash bajarilgan ustunlar va agregatlash natijasiga ega bo'lgan ustunlar kiradi.

SQL tilining imkoniyatlaridan biri, masalan, ixtiyoriy matnni so'rov tarkibiga kiritish mumkin. Bunga misol keltiramiz:

**SELECT 'O'rtacha ball =', AVG(mark) FROM mark_st
WHERE id_ex = 10**

Ushbu so'rov natijasida foydalanuvchi faqat oddiy sonlarni emas, balki matnni ham ko'radi [8].

Bir necha jadvallardan ma'lumotlarni tanlash uchun SQL ni qo'llash. Shu paytgacha faqat bitta jadvaldan ma'lumotlarni tanlash misollari keltirildi. Bir necha jadvallardan ham ma'lumotlarni tanlab olish mumkin va ba'zi holatlarni misollar orqali ko'rib o'tamiz.

Relyatsion model talabiga ko'ra, ma'lumotlar tanlab olinadigan jadvallar birga-bir yoki birga-ko'p shaklda bir biri bilan bog'langan (8.5 mavzu 1- misoldagi student, mark_st va exam_st jadvallari ER-diagrammasiga qarang).

mark_st jadvali **id_ex** maydoni bo'yicha **exam_st** jadvali bilan bog'langan.

mark_st jadvali **id_st** maydoni bo'yicha **student** jadvali bilan bog'langan.

Masalan, talabalar ro'yxatini imtihonlardan olgan baholari bilan birgalikda tanlash zarur bo'lsin. Buning uchun quyidagi so'rov beriladi:

```
SELECT student.stfam, student.stname, mark_st.id_ex,  
mark_st.mark
```

```
FROM student, mark_st WHERE student.id_st =  
mark_st.id_st
```

Keltirilgan ko'p jadvalli so'rov bir jadvalli so'rovdan quyidagilar bilan farq qiladi.

1. FROM seksiyasida ikkita jadval ko'rsatilgan.

2. Jadvallar soni bitta ko'p, shuning uchun ko'rsatilgan maydonlar nomining bir qiymatliligi yo'qoladi. Masalan, ko'p hollarda maydonni FROM da ko'rsatilgan jadvallar ro'yxatidagi qaysi jadvaldan olish noma'lum bo'lib qoladi. Maydon nomlarining ko'p qiymatliligini bataraf etish uchun maydon nomida perefiks - jadval nomi qo'shimcha qilinadi. Jadval nomi maydon nomidan nuqta bilan ajratiladi.

3. WHERE ifodasida jadvallarni solishtirish sharti ko'rsatiladi.

Ko`rinib turganidek jadval nomidan iborat prefiksdan foydalanish so`rovni murakkablashtiradi. Bunday murakkablikni bartaraf etish uchun psevdonim ishlatiladi. Yuqoridagi so`rovni quyidagicha yozish mumkin:

```
SELECT E.stfam, Estname, M.id_ex, M.mark FROM student  
E, mark_st M  
WHERE E.id_st = M.id_st
```

Nazorat savollari

1. Tranzaksiyalarni boshqarish operatorlari haqida tushuncha bering.

2. Ma`lumotlarni boshqarish operatorlari DMP va DQL haqida tushuncha bering.

3. SELECT operatori haqida tushuncha bering.

4. IN, BETWEEN, LIKE, IS NULL operatorlari haqida tushuncha bering. Ularni ta`riflang va misollar keltiring.

5. SELECT operatoriga oid misollar tuzing.

6. SELECT operatori yordamida bir nechta jadvallardan ma`lumotlarni tanlash haqida tushuncha bering.

7. INSERT operatorini qo`llash bo`yicha misol keltiring.

8. INSERT operatorining sintaksisi haqida tushuncha bering.

9. UPDATE operatori qanday vazifani bajaradi?

10. DELETE operatorining ishlatilishini misolda tushintiring.

8.6. Hisoblanadigan ustunlar. Murakkab so`rovlar

Hisoblanadigan ustunlar. SQL so`rovi bevosita MBdagi qiymatlardan iborat ustunlardan tashqari MBda saqlanuvchi qiymatlar asosida hisoblanadigan ustunlardan ham tashkil topadi. Hisoblanadigan ustunni olish uchun natija sifatida qaytariladigan qiymatlar ro`yxatiga ifodani qo`yish lozim. Oldingi qismlarda aytib o`tilganidek, ifoda qo`shish, ayirish,

ko'paytirish, bo'lish operastiyalaridan tashkil topadi. Ifodalarda qavslarni ham ishlatish mumkin. Arifmetik ifodalarda qatnashuvchi ustunlar , albatta sonli qiymatlarni saqlashi lozim. Matnli ma'lumotlar ustida qo'shish, ko'paytirish, ayirish yoki bo'lish amallari bajarilsa xatolik haqida xabar chiqariladi.

Quyidagi so'rovda oddiy hisoblanuvchi ustun keltirilgan .

Har bir ofis haqida u joylashgan shahar region va rejadan oshgan, rejaga etishmaydigan, bajarilgan savdo hajmi haqida ma'lumot chiqaring.

```
SELECT city, region, (sales-target)
FROM offices
```

city	region	sales- target
-----	-----	-----
Denver	Wester	\$113,958
	n	.00
New	Easter	\$117,637
york	n	.00
Chicago	Easter	\$64,968.
	n	00
Atlanta	Easter	\$17,911.
	n	00
Los	Wester	\$110,915
angeles	n	.00

Bu so'rovni bajarish uchun OFFICES jadvalining har bir satri uchun bitta natija satri tuziladi. So'rov natijasining birinchi 2 ta ustuni qiymati bevosita OFFICES jadvalidan olinadi. 3-ustun so'rov natijasining har bir satri uchun OFFICES jadvali ustunlarining joriy satrlari qiymatlari asosida hisoblanadi. quyida hisoblanuvchi ustun ishlatiladigan boshqa so'rovlarga misollar keltirilgan .

Xar bir tovarning umumiy qiymatini chiqaring.

```
SELECT mfr_id, product_id, description,
(qty_on_hand*price)
FROM products
```

mfr_id	product_id	descreption	(qty_on_hand*price)
REI	2A45C	Ratchet Link	\$79.00
ACI	41004	Widget Remover	\$2,750.00
QSA	XK47	Reduser	\$355.00
BIC	41672	late	\$180.00
IMM	779C	900-ibBrase	\$1,875.00
ACI	41003	Size 3 widget	\$107,00
ACI	41004	Size 4 Widget	\$117.00
BCI	41003	Handle	\$652.00

Agarda xar bir xizmatchining rejadagi savdo hajmi uning bajargan savdo hajmining 3 % iga oshirilsa, uning rejadagi savdo hajmi qanchaga teng bo'ladi.

```
SELECT name, quota, (quota*(0.03*sales))
FROM salesreps
```

name	Quota	(quota*(0.03*sales))
Bill Adams	\$350,000.00	\$361,037.33
Mary Jones	\$300,000.00	\$311,781.75
Sue Smith	\$350,000.00	\$364,221.50
Sam Slark	\$275,000.00	\$283,997.36
Bob Smith	\$200,000.00	\$204,277.82
Dan Roberts	\$300,000.00	\$309,170.19
Tom Snyder	NULL	NULL
Larry Fitch	\$350,000.00	\$360,855.95

DISTINCT kalitli so'zi. Agarda o'qish so'rovining qaytariladigan qiymatlar ustida jadvalning birlamchi kalitini ko'rsatsak so'rov natijasining har bir satri unikal bo'ladi.

Ofisning barcha menedjerlari haqida ma'lumot chiqaring

SELECT MGR FROM OFFICES

So'rov natijasi quyidagicha bo'ladi:

MGR

108

106

104

105

108

SELECT DISTINCT MGR FROM OFFICES

MGR

106

104

105

108

Bu so'rov quyidagi tarzda bajariladi. Oldin barcha natijaviy satrlar generastiya qilinadi, undan keyin esa boshqasini takrorlaydigan ustunlar o'chiriladi. Agarda **DISTING** kalit so'zi ko'rsatilmagan bo'lsa takrorlanadigan satrlar o'chirilmaydi.

Qiymatni biror diapazonga tegishlilik shartini tekshirish (BETWEEN).

1989 yilning oxirgi kvartalida berilgan barcha buyurtmalarni toping
**SELECT ORDER_NUM, ORDER_DATE, MFR, PRODUCT, AMOUNT
FROM ORDERS
WHERE ORDERS_DATE BETWEEN '01-OCT-89' AND '31-DEC-89'**

Natija:

ORDER_ NUM	ORDER_ DATE	MFR	PRO DUCT	AMOUNT
112961	17-DEC-89	REI	2A44L	\$31,500.00
112968	12-OCT-89	ACI	41004	\$3,276.00
112963	17-DEC-89	ACI	41004	\$702.00
112983	27-DEC-89	ACI	41004	\$15,000.00
112979	12-OCT-89	ACI	41007	\$760.00
112992	04-NOV-89	ACI	41008	\$R,100.00
112975	12-OCT-89	REI	2A44G	\$3,276.00
112987	31-DEC-89	ACI	41004	\$27,500.00

Narxi ko'rsatilgan diapazonda yotuvchi buyurtmalar haqidagi ma'lumotlarni chiqaring.

```
SELECT ORDER_NUM, AMOUNT FROM ORDERS
WHERE AMOUNT BETWEEN 30000.00 AND 39999.99
```

Natija:

ORDER_NU	AMOUNT
112961	\$1,500.00
113069	\$31,350.00

```
SELECT ORDER_NUM, AMOUNT
FROM ORDERS
```

```
WHERE AMOUNT BETWEEN 40000.00 AND 49999.99
```

ORDER_NUM	AMOUNT
113045	\$45,000.00

Bajargan savdo hajmi 80% dan – 120% gacha diapazonda yotmaydigan xodimlar ro'yxatini chiqaring.

```
SELECT NAME, SALES, QUOTA FROM SALESREPS
```

```
WHERE SALES NOT BETWEEN (0.8*QUOTA) AND
(1.2*QUOTA)
```

Natija:		
NAME	SALES	QUOTA
-----	-----	-----
MARY JONES	\$392,475.00	\$300,000.00
SUE SMITH	\$474,050.00	\$350,000.00
BOB SMITH	\$142,594.00	\$200,000.00
NANCY	\$186,042.00	\$300,000.00
ANGELLI		

WHERE kalit so'zi. **WHERE** kalit so'zi **SELECT** operatori yordamida ma'lumotlarni ajratib olishda qiymatlar bo'yicha shart kiritish uchun ishlatiladi. Quyidagi operatorni kiritamiz:

SELECT * FROM REGIONS WHERE Zip = 8324

Natijada bitta qator tanlanadi va Zip ustunidagi qiymati 8324 ga teng bo'ladi.

Keyingi keltirilayotgan operatorlarda **REGIONS** jadvalidagi qatorlarni sanash **REGION** ustuni "Gijduvan" bilan boshlanadigan shart asosida bo'ladi.

SELECT COUNT(*) FROM REGIONS WHERE Region LIKE 'Gijduvan%'

Natijada 1 qiymati qayd etiladi.

SELECT operatorining **WHERE** kalit so'zi bilan keladigan yana bir nechta variantlarini ko'ramiz.

SELECT COUNT(*) FROM REGIONS WHERE Zip BETWEEN 8300 AND 8320

Bu erda Zip maydoni 8300 dan 8320 gacha diapazonda bo'lgan qatorlar soni aniqlanadi.

SELECT * FROM REGIONS WHERE Zip IN (8300,8310,8320)

Bu erda zip maydonining qiymati **IN** so'zidan keyin turgan qiymatlardan biriga teng bo'lgan qatorlar tanlanadi.

SELECT MAX(Zip) FROM REGIONS WHERE Region IS NOT NULL

Bu operatorida REGIONS jadvalining Zip maydonidagi eng katta qiymatga ega bo'lgan qator Region maydonidagi qandaydir qiymatga ega bo'lgan qatorlar orasidan tanlanadi.

```
SELECT * FROM REGIONS
```

```
WHERE Region LIKE 'Gijduvan%' AND City = 'Gishti'
```

Bu erda Region maydoning qiymati "Gijduvan" bilan boshlanadigan va City maydoning qiymati "Gishti" ga teng bo'lgan qatorlar tanlanadi.

Birlashtirish. WHERE kalit so'zining yana bir qo'llanish sohalaridan biri jadvallarni mantiqiy birlashtirishdan iborat. Birlashtirish amali ikki etapda amalga oshiriladi: avval FROM kalit so'zi yordamida SELECT operatoridagi jadvallar birlashtiriladi, keyin esa WHERE kalit so'zi orqali jadvallarni kerakli maydon bo'yicha birlashtiriladi.

Quyidagi misolni ko'ramiz:

```
SELECT STAFF.LastName, STAFF.FirstName,  
STAFF.FatherName, REGIONS.City FROM STAFF, REGIONS  
WHERE STAFF.Zip = REGIONS.ZIP
```

Bu operator bajarilishi natijasida to'rtta maydondan iborat qator chiqariladi: Jo'raev, Azamat, Nosirovich, Gishti. Bu erda STAFF va REGION jadvallari Zip ustuni bo'yicha birlashtirilgan. WHERE shartida tenglikdan chap tomonda turgan jadval tashqi (outer), o'ng tomonda turgan jadval esa ichki (inner) jadval deyiladi. Yuqoridagi keltirilgan birlashtirish ichki birlashtirish (inner join) deyiladi.

Bundan tashqari birlashtirish tashqi (outer join) bo'lishi ham mumkin.

Bundan tashqari shartga bog'liq bo'lmagan qatorlar ham keltiriladi, birlashtirish shartiga mos kelmagan barcha maydonlar NULL qiymatini beradi.

Tashqi va ichki birlashtirishning ANSI sintaksisi quyidagi ko'rinishga ega.

```
SELECT Table1.Column1, Table2.Column2
```

FROM Table1 **OUTER** [**INNER**] **JOIN** Table2
ON Table1.Column1 = Table2.Column3

Ichma-ich so'rovlar. Ichma-ich so'rovlar — bu shunday **SELECT** operatorlariki, **WHERE** kalit so'zidan keyin ham **SELECT** operatori ishlatiladi.

Misol:

```
SELECT * FROM STAFF  
WHERE PosID IN (SELECT PosID FROM POSS  
WHERE PosLevel BETWEEN 1 AND 3)
```

Bu operatorida oldin ichki so'rov bajariladi, unda lavozim identifikatori 1,2 yoki 3 ga teng bo'lgan yozuvlar tanlanadi, so'ngra **STAFF** jadvalidan **PosID** maydonining qiymati oldingi so'rovda aniqlangan naborga mos keladigan qatorlar tanlanadi.

GROUP BY amali. **SQL** da **GROUP BY** amali orqali ma'lum bir maydonlar bo'yicha guruxlash amalga oshiriladi. Odatda bu amal **SELECT** operatorining oxirgi maydoni sifatida matematik funkstiya ishlatilganda qo'llaniladi.

Misol:

```
SELECT DEPS.DeptFullName, SUM(STAFF.Salary)  
FROM DEPS, STAFF  
WHERE STAFF.DepID = DEPS.DeptID  
GROUP BY DEPS.DeptFullName
```

Bu operatorida xodimlar maoshining summasi bo'limlar bo'yicha guruxlanib hisoblanadi. Bunda **STAFF** va **DEPS** jadvallari birlashtirilayapti. Chunki **STAFF**da bo'limning faqat identifikatori saqlanadi.

Yana bitta guruxlash amaliga doir misol ko'ramiz:

```
SELECT LastName, COUNT(*) FROM STAFF  
GROUP BY LastName
```

Bu operator yordamida **STAFF** jadvalidagi bir familiyalı xodimlar soni aneqlanadi.

HAVING kalit so'zi. **HAVING** kalit so'zi **GROUP BY** qatnashgan so'rov bajarilgandan keyin qatorlarni tanlash uchun ishlatiladi. Uning ishlashi **WHERE** ga o'xshab ketadi. **WHERE** faqat dastlabki jadval bilan ishlaydi, **HAVING** esa **SELECT** bajarilgandan keyingi natija bilan ishlaydi.

HAVING kalit so'zi ishlatilishiga doir misol keltiramiz:

```
SELECT LastName, COUNT(*) FROM STAFF
```

```
GROUP BY LastName
```

```
HAVING LastName LIKE 'A%'
```

Bu operatorida "A" harfi bilan boshlanuvchi birfamiliyalilar soni hisoblanadi. Bu misolni boshqacha yozish ham mumkin:

```
SELECT LastName, COUNT(*) FROM STAFF
```

```
WHERE LastName LIKE 'A%'
```

```
GROUP BY LastName
```

ORDER BY amali. **ORDER BY** amali tanlangan naborni tartiblashtirish uchun ishlatiladi.

Masalan:

```
SELECT * FROM STAFF
```

```
ORDER BY LastName, FirstName, FatherName
```

Bu so'rov bajarilishi natijasida **STAFF** jadvali familiya, ism va sharif bo'yicha tartiblangan holda chiqariladi.

Ustun psevdonimlari. Matematik funkstiyalarning natijalarini chiqarish uchun ustunni nomlash mumkin. Bunday nomlar ustun psevdonimlari deyiladi (column alias).

Masalan:

```
SELECT LastName, COUNT(*) AS CountOfNames FROM  
STAFF
```

```
GROUP BY LastName
```

Natijaviy naborda ikkinchi maydonning nomi, ya'ni familiyalar soni joylashgan maydonning nomi **CountOfNames** deb nomlanadi. **WHERE** yoki **GROUP BY** da psevdonimlar ishlatilmaydi. Ularda faqat real maydon nomlari ko'rsatiladi.

Jadval psevdonimlari. Har sira jadvalning to'liq nomini SELECT buyrug'ining har-xil qismlarida ko'rsatmasdan uning qisqartirilgan nomini ishlatish mumkin. Bunday qisqartirilgan nom jadval psevdonimi (table alias) deyiladi. U FROM kalit so'zi yordamida yaratiladi:

```
SELECT S.LastName, S.FirstName, S.FatherName, R.City  
FROM STAFF S, REGIONS R WHERE S.Zip = R.ZIP
```

Nazorat savollari

1. Agregat funksiyalar haqija ma'lumot bering.
2. Hisoblanadigan ustunlarga misol keltiring.
3. Qaytariladigan qiymatlar (DISTING kalitli so'zi) ni izohlang.
4. Qiymatni biror diapazonga tegishlilik shartini tekshirishga doir so'rov tuzing.
5. WHERE kalit so'zi ishlatiladigan so'rov tuzing;
6. Birlashtirishga so'rov tuzing.
7. Ichma-ich so'rovlarga misol keltiring.
8. GROUP BY amali qaerlarda ishlatiladi ?
9. HAVING kalit so'zi qandaq hollarda ishlatiladi?
10. ORDER BY amali qachon ishlatiladi ?
11. Ustun psevdonimlarini qo'llash texnologiyasini izohlang?
12. Jadval psevdonimlari nima uchun ishlatiladi ?

IX. INTERBASEDA MA'LUMOTLAR BAZASINI YARATISH

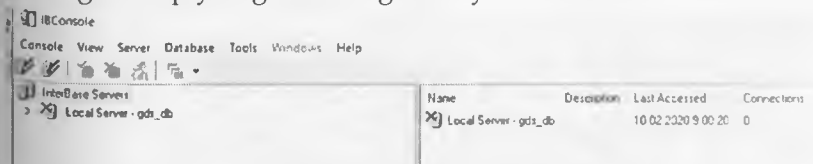
9.1. InterBaseda ishlash asoslari

SQL tilining poydevori ANSI spetifikastiyasi hisoblanadi. Bu spetifikastiyaning komandalari SQLni qo'llovchi barcha ma'lumotlar bazasida bajariladi. Har bir serverli MBBT o'zining xususiy dialektini hosil qilgan holda ANSI spetifikastiyasiga qo'shimchalar kiritib uni kengaytirmoqda. Bu mavzuda biz asosan InterBase 6.5 MBBTda ishlashni ko'rib chiqamiz. Asosiy e'tibor mana shu MBBT uchun SQL tili dialektining mohiyatiga qaratiladi.

Agar ishlash uchun boshqa bir SQL-server ishlatiladigan bo'lsa, u holda quyidagilarni e'tiborga olish kerak: Sybase va Microsoft SQL Server MBBT oladigan bo'lsak, ularning SQL sintaksisi ANSI standartidan birmuncha farq qiladi. InterBase va Oracle MBBTda esa ANSI standart operatorlariga kamroq darajada qo'shimcha kiritilgan. Shu sababli bu mavzuda ko'rilayotgan namunalar Sybase va Microsoft SQL Serverda noto'g'ri bajarilishi mumkin.

Biz ko'rayotgan namunalarni SQL Explorer dasturida ham bajarish mumkin. InterBase MBBTni o'rganayotganligimiz uchun yaxshisi bu namunalarni InterBase SQL-serverida bajarib ko'ramiz.

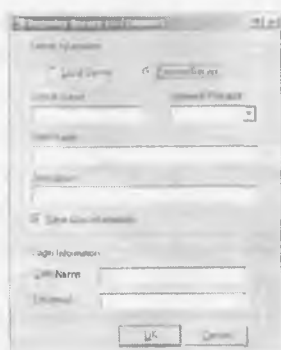
IBConsole ilovasi. InterBaseda barcha administratsiya vositalari va ma'lumotlar bazasi bilan ishlash **IBConsole ilovasi** yordamida bajariladi. Bu ilova bilan ishlashga kirishish oldidan InterBase serverini ishga tushirish lozim. **IBConsole ilovasi** ishga tushirilganda quyidagi rasmdagidek oyna ochiladi



9.1.1-rasm. IBConsole ilovasi

Serverga ulanish. Yuqoridagi rasmda oynaning chap qismida qayd qilingan serverlar ro'yxati joylashgan. Hozircha bu ro'yxat bo'sh, birorta ham server qayd qilinmagan. Serverga ulanish quyidagi tartibda amalga oshiriladi.

1. Server - > Register buyrug'ini bajarining yoki daraxtsimon strukturaning o'zak elementi InterBase Serversga sichqoncha tugmasini ikki marta bosning. Natijada quyidagi ko'rinishdagi oyna ochiladi:



9.1.2-rasm. Serverga ulanish

Ushbu dialog oynasiga qaysi serverga ulanish kerakligi haqidagi ma'lumot kiritiladi:

Local Server - lokal serverga;

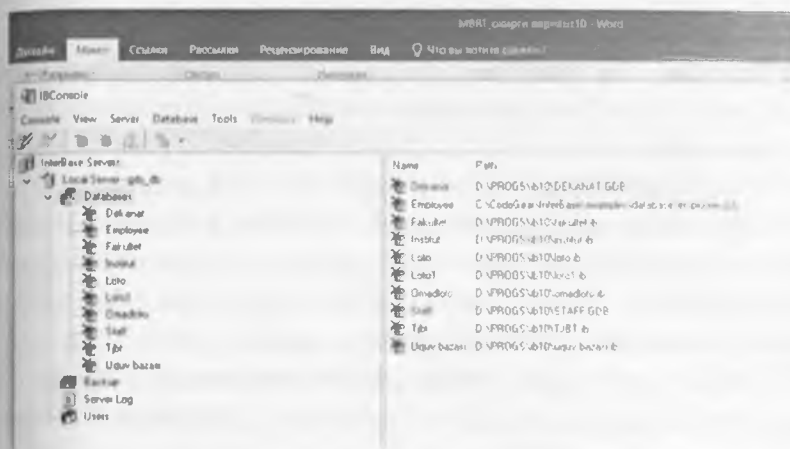
Remote Server – uzoqdagi serverga

Lokal server – foydalanuvchi kompyuterida joylashgan server, uzoqdagi serverga esa bog'lanish tarmoq orqali amalga oshiriladi. Agar uzoqdagi server tanlangan taqdirda **Server name** maydoniga serverning Network Protocol ga mos keluvchi tarmoq nomi yoki adresi ko'rsatiladi.

2. Sichqoncha bilan **Local Server** ni belgilaymiz.

Login Information bo'limiga sichqoncha yordamida belgilab, unga foydalanuvchi va parolni kiriting. InterBase da to'liq administrator huquqini beruvchi standart foydalanuvchi

sifatida **SYSDBA** belgilangan, parol sifatida esa **masterkey** tanlangan. Bu qiymatlarni keyinchalik o'zgartirish mumkin. OK tugmasi bosilganda quyidagi ekran hosil bo'ladi:



9.1.3- rasm. Lokal serverga ulanish bajarilgan hol

Mavjud ma'lumotlar bazasini qayd qilish. Databases yordamida mavjud MBni qayd qilish mumkin yoki yangi MB yaratish mumkin. Dastlab mavjud MB qayd qilishni ko'ramiz.

1. **Database >- Register** buyrug'ini bajaring va sichqoncha bilan uch nuqtalik joyga ko'rsatkichni keltirib chap tugmani bosamiz.

2. InterBase MBBTning MB .gdb kengaytmaga ega bo'ladi. Qayd qilingan MB sifatida employee.gdb ni tanlaymiz. U \ProgramFiles\Common Files\Borland Shared\Data papkasida joylashgan.

3. OK tugmasini bosamiz.

Natijada daraxtsimon strukturaning **Database** elementiga bo'yso'nuvchi employee.gdb elementi paydo bo'ladi. U ham o'z novbatida bo'yso'nuvchi xususiy elementlarga ega. Ma'lumotlar bazasining elementlariga domenlar (**Domains**), jadvallar (**Tables**), indekslar (**Indexes**), ko'rinishlar (**Views**), saqlanuvchi

prosteduralar (**Stored Procedures**), tashqi funkstiyalar (**External Functions**), istisno (**Exceptions**), BLOB-filtrlar (**BLOB Filters**) va rollar (**Roles**) kiradi.

9.2. InterBaseda yangi ma'lumotlar bazasini yaratish

InterBase **MBBT**da ma'lumotlar bazasini yaratishni "Xodimlar" bo'limi uchun ma'lumotlar bazasini yaratish misolida ko'rib o'tamiz.

"Xodimlar" dasturi uchun **staff.gdb** ma'lumotlar bazasini yaratishni ko'rib o'tamiz. **IBConsole** dasturini ishga tushiramiz. Bu dasturdagi **Interactive SQL** oynasini ochamiz. Bu oyna yordamida **SQL** buyruqlarni bajarish mumkin. Bu buyruqlarni **DELPHI** ilovalari orqali ham bajarish mumkin.

«Xodimlar» dasturi uchun ma'lumotlar bazasini yaratishni avval aloxida katalog ochishdan boshlaymiz. Bu katalogda dastur fayllari va ma'lumotlar bazasi joylashadi. «Xodimlar» daturining ma'lumotlar bazasini nomi **STAFF.GDB** bo'lsin. Ma'lumotlar bazasini va uning elementlarini yaratishdan oldin **Interactive SQL** oynasiga quyidagi buyruqni kiritamiz:

```
SET SQL DIALECT 3
```

Agar bu buyruqni bajarish paytida xatolik xaqidagi ma'lumot chiqarilsa, bu boshka bir ma'lumotlar bazasi aktiv holda ekanligidan dalolat beradi. Bunday xolda **Database >- Disconnect** buyrug'i yordamida uni passiv xolga keltiriladi. Ma'lumotlar bazasini yaratish uchun **CREATE DATABASE** ko'rinishdagi **SQL** buyrug'i ishlatiladi. **Staff.gdb** ma'lumotlar bazasini yaratamiz. Buning uchun quyidagi **SQL** buyruqni kiritamiz.

```
CREATE DATABASE "d:\programs\staff\base\staff.gdb"  
USER "SYSDBA" PASSWORD "rnasterkey"  
PAGE_SIZE = 4096  
LENGTH = 10000 PAGES
```

Natijada `e:\programs\staff\base` katalogida `staff .gdb` ma'lumotlari bazasi yaratiladi. Bu ma'lumotlar bazasining o'lchami $4 \text{ Kbayt} * 10000 = 40 \text{ Mbayt}$ dan oshmaydi.

Lokal ma'lumotlar bazasini yaratishda `SET SQL DIALECT` va `CREATE DATABASE` buyruqlar ishlatilmaydi. Lokal ma'lumotlar bazasidan farqli ravishda serverli ma'lumotlar bazasida serverli ma'lumotlar bazasining barcha ob'ektlari bitta faylda joylashgan bo'ladi.

Interactive SQL oynasida `CREATE DATABASE` buyrug'i bajarilganda so'ng yaratilgan baza aktiv holatga keladi. Bundan keyingi barcha `SQL` buyruqlar shu ma'lumotlar bazasiga ko'llaniladi. Bu jarayon ma'lumotlar bazasini `DISCONNECT` buyrug'i yordamida passiv xolatga keltirguncha davom etadi. Ma'lumotlar bazasiga ulanish uchun `SQL` ning `CONNECT` buyrug'i ishlatiladi

9.3. Jadvallar yaratish

Jadvallarni yaratishdan oldin InterBase da qanday tiplar ishlatilishi bilan tanishib utamiz:

BLOB Katta o'lchamli ma'lumotlar, ya'ni grafik, matn, raqamli tovushni saqlash uchun ishlatiladi;

CHAR(n) 1 dan 32765 simvolgacha bo'lgan matnli informastiyani saqlash uchun;

DATE 01.01.01 00 dan 11.01.5941 gacha sana va vaqt;

DECIMAL (p,S) 15 razryadli, p parametriga umumiy razryadlar soni, s ga esa veguldan keyingi raqamlar soni to'g'ri keladi;

DOUBLE PRECISION $1,7 \times 10^{-308}$ dan $1,7 \times 10^{308}$ gacha bo'lgan 15 razryadli siljuvchi vergulli sonlar;

FLOAT $2,4 \times 10^{-38}$ dan $3,4 \times 10^{38}$ gacha bo'lgan etti razriyadli siljuvchi vergulli sonlar;

INTEGER -2147483648 dan 2147483647 gacha bo'lgan butun sonlar;

NUMERIC (p,s) 15 razryadli **DECIMAL** ga o'xshash va aniq ko'rsatilgan o'lchamlarda bo'lishi shart bo'lmagan sonlar;
SMALLINT -32768 dan 32767 o'lchamgacha bo'lgan kichik butun sonlar;

VARCHAR(n) 1 dan 32765 gacha o'zgaruvchi matn tipli kattaliklar saqlanadi.

Lokal jadvallarda **DECIMAL**, **DOUBLE PRECISION** va **VARCHAR** tipli kattaliklar ishlatilmaydi. Jadval yaratishda SQLning **CREATE TABLE** operatori ishlatiladi. Oltita jadval yaratamiz.

- **STAFF** — xodimlar va ular haqidagi ayrim individual ma'lumotlar.
- **REGIONS** — korxona atrofidagi rayonlar va ularning pochta indeksleri.
- **DEPS** — korxonaning bo'limlar ro'yxati.
- **POSS** — lavozimlar ro'yxati.
- **JOBS** — xodimning mexnat faoliyati haqidagi ma'lumotlar saqlanadigan jadval.
- **FAMILY** — Xodimlarning oila a'zolari haqidagi ma'lumotlar saqlanadigan jadval.

Endi bu jadvallarni **Interactive SQL** oynasi yordamida yaratamiz:

```
CREATE TABLE STAFF  
( ID integer NOT NULL PRIMARY KEY,  
  LastName varchar(30) NOT NULL,  
  FirstName varchar(30) NOT NULL,  
  FatherName varchar(30),  
  Zip integer,  
  Street varchar(30),  
  House varchar(10),  
  Tel varchar(20),  
  TaxCode numeric(12,0) DEFAULT 0 NOT NULL,  
  TabNum varchar(10),
```

BirthDate date NOT NULL,
DepID smallint NOT NULL,
PosID smallint NOT NULL,
BornPlace varchar(50),
Salary numeric(15,2) DEFAULT 200.0 NOT NULL,
PasspNum varchar(20),
PasspDate date,
Photo blob)

Jadvaldagi maydonlarning qo'llanilishi xaqida tushuncha beramiz.

STAFF. ID — takrorlanmaydigan yozuvli identifikator bo'lib, bilamchi kalit qurishda ishlatiladi.

LastName, FirstName va FatherName — familiya, ism va sharif.

Zip, street, House, Tel — mos ravishda pochta indeksi, shahar, ko'cha, uy va kvartira nomeri, telefon nomeri .

TaxCode — nalog to'lanuvchi identifikastion nomer

TabNum — korxona reestri bo'yicha xodimning tabel nomeri.

BirthDate — tug'ilgan sanasi.

DepID va POSID — korxona va lavozim identifikatorlari.

BornPlace — tug'ulgan joyi.

PasspNum va PasspDate — pasport nomeri va berilgan vaqti.

Photo — fotografiya.

PRIMARY KEY atributi birlamchi kalitga mos keladi.

Ba'zi qatorlar NOT NULL atributi bilan tugaydi. Bu maydonning bo'sh qoldirilmasligini anglatadi. Xususan birlamchi kalitli maydonlar albatta NOT NULL atributiga ega bo'lishi kerak. Bundan tashqari maydonni yaratishda DEFAULT kalit so'zi va undan keyin keladigan mos qiymatlardan ham foydalanish muikin. Bu qiymat jadvalga yangi yozuv

qo'shilganda avtomatik kiritiladi. Masalan, jadvalga yangi yozuv qo'shilganda Salary maydoniga 200 qiymati kiritiladi.

```
CREATE TABLE REGIONS  
( Zip integer NOT NULL PRIMARY KEY,  
Area varchar(30),  
Region varchar(30),  
City varchar(20) NOT NULL)
```

Bu jadval quyidagi ma'lumotlarni saqlashga mo'ljallangan: pochta indeksi, region nomi, rayon nomi va aholi punktining nomi.

```
CREATE TABLE DEPS  
( DeptID integer NOT NULL PRIMARY KEY,  
DeptFullName varchar(100) NOT NULL,  
DeptShortName varchar(10) NOT NULL,  
ParentDeptID integer DEFAULT 0 NOT NULL,  
Parents varchar(100))
```

Bu jadval qatorlariga quyidagi informatsiyalar kiritiladi: identifikator, bo'limning to'liq va qisqartirilgan nomi. Iyerarxik strukturali bo'limlarning o'zak elementi uchun ParentDeptID maydoniga 0 kiritiladi, boshqalariga esa "ota-ona" bo'lim identifikatori qo'yiladi. Parents maydoniga barcha "ota-ona" bo'limlarning identifikatorlar ketma-ketligi kiritiladi. Masalan, 1 identifikatorli bo'lim mavjud bo'lsin va unda 3 identifikatorli bo'yso'nuvchi bo'lim mavjud bo'lsin, u holda Parents maydoniga 1,3 qiymatlar kiritiladi. Ierarxiya yuqorisida joylashgan bo'lim uchun Parents maydoniga NULL qiymati kiritiladi.

```
CREATE TABLE POSS  
( PosID integer NOT NULL PRIMARY KEY,  
PosFullName varchar(100) NOT NULL,  
PosShortName varchar(20) NOT NULL,  
PosLevel smallint DEFAULT 1 NOT NULL)
```

Bu jadval qatorlariga quyidagi informatsiyalar kiritiladi: identifikator, lavozim nomining to'liq va qisqartirilgan nomi,

shuningdek lavozimning darajasi haqidagi ma'lumot. Son qancha kichik bo'lsa, lavozim shuncha yuqori.

```
CREATE TABLE JOBS  
( EmpID integer NOT NULL,  
StartDate date NOT NULL,  
StopDate date.  
Organization varchar(100) NOT NULL,  
Dep varchar(100),  
Pos varchar(100) NOT NULL,  
StopCauses varchar(100),  
CurOrg char(1),  
InOrderNum varchar(10),  
OutOrderNum varchar(10))
```

Bu jadval qatorlariga quyidagi informatsiyalar kiritiladi: xodim identifikatori, mehnat faoliyatining boshlangan va tugagan sanasi, tashkilot nomi, bo'lim nomi, lavozim, ishdan bo'shalish/ko'chirilish sababi, hozirgi tashkilotning ishga olinganlik yoki bo'shatilganlik haqidagi buyruq nomeri.

```
CREATE TABLE FAMILY  
( EmpID integer NOT NULL,  
kin varchar(S) NOT NULL,  
BirthDate date,  
KinNarae varchar(50) NOT NULL)
```

Bu jadval qatorlariga quyidagi informatsiyalar kiritiladi: xodim identifikatori, qarindoshlik aloqasining turi (xotini, eri, o'g'li, qizi, va b.), tug'ilgan sanasi va qarindoshning ismi.

Jadvallarni o'chirish. Jadvalni o'chirish operatori juda oddiy:

DROP TABLE jadval nomi

Agar jadval boshqa jadvallar bilan tashqi kalitlar orqali bog'langan bo'lsa, DROP TABLE buyrug'i bajarilmaydi.

Indekslar yaratish. Indeks yaratish uchun SQL buyruqlaridan CREATE INDEX buyrug'i ishlatiladi. Birlamchi va tashqi kalitlarni aniqlashda unga mos indekslar avtomatik tarzda yaratiladi, boshqa hollarda esa mustaqil yaratish zarur.

1. STAFF jadvali uchun indeks yaratamiz va bunda qatorlar familiya, ism va sharif bo'yicha saralansin.

```
CREATE INDEX STAFF_INDEX1 ON STAFF  
(LastName,FirstName,FatherHame)
```

2. Bu jadval uchun yana ikkita indeks yaratamiz: nalog to'lovchining kodi ustuni va tabel nomeri ustuni bo'yicha.

```
CREATE UNIQUE INDEX STAFF_INDEX2 ON STAFF  
(TaxCode);
```

```
CREATE UNIQUE INDEX STAFF_INDEX3 ON STAFF  
(TabNum)
```

UNIQUE kalit so'zi bu indeksda qiymatlarning takrorlanishiga yo'l qo'yilmasligini bildiradi.

3. Endi JOBS jadvali uchun indekslar yaratamiz:

```
CREATE INDEX JOBS_INDEX1 ON JOBS (EmpID,  
StartDate)
```

Bu indeks qatorlarning ish faoliyati boshlangan sana bo'yicha saralanishini ta'minlash uchun ishlatiladi.

9.4. Interbaseda ma'lumotlar bilan islash

Jadvalga yangi qator qo'shish uchun INSERT operatori ishlatiladi.

Masalan, staff. gdb ma'lumotlar bazasining REGIONS jadvaliga bir juft qator qo'shamiz:

```
INSERT INTO REGIONS  
VALUES(8320,'Buxoro','','Mustaqillik');  
INSERT INTO REGIONS  
VALUES(8324,'Buxoro obl.','Gijduvon','Gishti' |
```

Birinchi INSERT operatoridan keyin ";" simvoli qo'yilgan bo'lib, bu InterBase da SQL operatorlarini ajratuvchi belgi

hisoblanadi. Bu belgi InterBase da an'anaviy ishlatiladi. Ajratuvchi belgini SET TERM buyrug'i yordamida boshqa belgi bilan almashtirish mumkin.

Yuqoridagi SQL operatorlari qanday vazifani bajarishini ko'rib o'taylik. Birinchi INSERT operatori REGIONS jadvaliga Buxoro shaxri va Mustaqillik ko'chasi ma'lumotlarini kiritadi, ikkinchi INSERT operatori esa Buxoro oblast, G'ijduvon tuman, G'ishti aholi punkti haqidagi ma'lumotlarni kiritadi. Bu operatorda INSERT yordamida barcha maydonlarga yozuv kiritiladi.

Endi ma'lumotlarni INSERT operatori yordamida ajratib kiritishni ko'rib o'tamiz. STAFF jadvaliga bitta kator kiritamiz (sana formati mm/dd/gg ko'rinishda):

```
INSERT INTO STAFF(ID,LastName,FirstName,FatherName,  
Zip,BirthDate,DepID,PosID,TabNum)
```

```
VALUES(0, 'Juraev', 'Azamat', 'Nosirovich',8324;
```

```
'30/08/1988',0,0,'0' )
```

Natijada STAFF jadvaliga 0 identifikatorli yozuv kiritiladi.

COMMIT va ROLLBACK buyruqlari. Ma'lumotlar

bazasiga qilingan barcha o'zgartirishlar to'plami (Masalan, INSERT buyrug'i yordamida) **tranzakstiya** deb ataladi.

Tranzakstiya tushunchasi faqat SQL-serverlar bilan ishlaganda mavjud bo'ladi, lokal ma'lumotlar bazasi bilan ishlaganda esa

ishlatilmaydi. Barcha o'zgarishlarni bazaga yozish uchun

COMMIT buyrug'i ishlatiladi. Bu buyruq bajarilmaguncha

ma'lumotlar bazaga joylashtirilmaydi. Bazaga joylashtirishni

bekor qilish uchun esa ROLLBACK buyrug'i ishlatiladi. Bazaga joylashtirilib bo'lingan o'zgartirishlarni bekor qilib bo'lmaydi.

Ma'lumotlarni tanlash. Jadvaldagi ma'lumotlarni olish

uchun SELECT buyrug'i ishlatiladi. Oddiy formada bu buyruq quyidagi sintaksisga ega:

```
SELECT * FROM REGIONS
```

1. Interactive SQL oynasida bu buyruqni bajarganda REGIONS jadvalidan tanlangan katorlar to'plami oynaning pastki qismida ekranga chiqariladi. "*" simvoli jadvaldagi barcha maydonlardagi ma'lumotlar ekranga chiqarilishini bildiradi.

2. Bu buyruqni biroz o'zgartiramiz:

```
SELECT Zip, City FROM REGIONS
```

Bu buyruq bajarilganda faqat **zip** va **City** maydonlaridagi ma'lumotlar ekranga chiqiriladi.

3. Quyidagi buyruqni kiritamiz:

```
SELECT Area FROM REGIONS
```

Natijada ekranga fakat Area maydonidagi ma'lumotlar chiqariladi

4. Oldingi SELECT buyrug'ini modifikastiyalashtiramiz:

```
SELECT DISTINCT Area From REGIONS
```

Bu buyruqda DISTINCT kalit so'zi ishlatilyapti va u Area maydonida bir xil qiymatlarni takrorlanishiga yo'l qo'ymaydi. Buning natijasida Buxoro yozuvli qator bir marta chiqariladi.

Birlashtirish. WHERE kalit so'zining yana bir funksiyalaridan biri jadvallarni mantiqiy birlashtirishdan iborat. Birlashtirish amali ikki bosqichda amalga oshiriladi: avval FROM kalit so'zi yordamida SELECT operatoridagi jadvallar birlashtiriladi, keyin esa WHERE kalit so'zi orqali jadvallarni kerakli maydon bo'yicha birlashtiriladi.

Quyidagi misolni ko'ramiz:

```
SELECT STAFF.LastName, STAFF.FirstName,  
STAFF.FatherName, REGIONS.City FROM STAFF, REGIONS  
WHERE STAFF.Zip = REGIONS.ZIP
```

Bu operator bajarilishi natijasida to'rtta maydondan iborat qator chiqariladi: Jo'raev, Azamat, Nosirovich, Gishti. Bu erda STAFF va REGION jadvallari Zip ustuni bo'yicha birlashtirilgan. WHERE shartida tenglikdan chap tomonda turgan jadval tashqi (outer), o'ng tomonda turgan jadval esa ichki (inner) jadval

deyiladi. Yuqoridagi keltirilgan birlashtirish ichki birlashtirish (inner join) deyiladi.

Bundan tashqari birlashtirish tashqi (outer join) bo'lishi ham mumkin.

Bundan tashqari shartga bog'liq bo'lmagan qatorlar ham keltiriladi, birlashtirish shartiga mos kelmagan barcha maydonlar NULL qiymatini beradi.

Tashqi va ichki birlashtirishning ANSI sintaksisi quyidagi ko'rinishga ega.

```
SELECT Table1.Column1, Table2.Column2  
FROM Table1 OUTER [INNER] JOIN Table2  
ON Table1.Column1 = Table2.Column3
```

Ichma-ich so'rovlar. Ichma-ich so'rovlar — bu shunday SELECT operatorlariki, WHERE kalit so'zidan keyin ham SELECT operatori ishlatiladi. Misol:

```
SELECT * FROM STAFF  
WHERE PosID IN (SELECT PosID FROM POSS  
WHERE PosLevel BETWEEN 1 AND 3)
```

Bu operatorlarda oldin ichki so'rov bajariladi, unda lavozim identifikatori 1,2 yoki 3 ga teng bo'lgan yozuvlar tanlanadi, so'ngra STAFF jadvalidan PosID maydonning qiymati oldingi so'rovda aniqlangan naborga mos keladigan qatorlar tanlanadi.

GROUP BY amali. SQL da GROUP BY amali orqali ma'lum bir maydonlar bo'yicha guruxlash amalga oshiriladi. Odatda bu amal SELECT operatorining oxirgi maydoni sifatida matematik funktsiya ishlatilganda qo'llaniladi.

Misol:

```
SELECT DEPS.DeptFullName, SUM(STAFF.Salary)  
FROM DEPS, STAFF  
WHERE STAFF.DepID = DEPS.DeptID  
GROUP BY DEPS.DeptFullName
```

Bu operatorlarda xodimlar maoshining summasi bo'limlar bo'yicha guruhlanib hisoblanadi. Bunda STAFF va DEPS

jadvallari birlashtirilayapti. Chunki STAFFda bo'limning faqat identifikatori saqlanadi.

Yana bitta guruhlash amaliga doir misol ko'ramiz:

```
SELECT LastName, COUNT(*) FROM STAFF  
GROUP BY LastName
```

Bu operator yordamida STAFF jadvalidagi bir familiyalı xodimlar soni aneqlanadi.

HAVING kalit so'zi. HAVING kalit so'zi GROUP BY qatnashgan so'rov bajarilgandan keyin qatorlarni tanlash uchun ishlatiladi. Uning ishlashi WHERE ga o'xshab ketadi. WHERE faqat dastlabki jadval bilan ishlaydi, HAVING esa SELECT bajarilgandan keyingi natija bilan ishlaydi.

HAVINGkalit so'zi ishlatilishiga doir misol keltiramiz:

```
SELECT LastName, COUNT(*) FROM STAFF  
GROUP BY LastName  
HAVING LastName LIKE 'A%'
```

Bu operatorida "A" harfi bilan boshlanuvchi birfamiliyalilar soni hisoblanadi. Bu misolni boshqacha yozish ham mumkin:

```
SELECT LastName, COUNT(*) FROM STAFF  
WHERE LastName LIKE 'A%'  
GROUP BY LastName
```

ORDER BY amali. ORDER BY amali tanlangan jadvalni tartiblashtirish uchun ishlatiladi.

Masalan:

```
SELECT * FROM STAFF  
ORDER BY LastName, FirstName, FatherName
```

Bu so'rov bajarilishi natijasida STAFF jadvali familiya, ism va sharif bo'yicha tartiblangan holda chiqariladi.

Ustun psevdonimlari. Matematik funkstiyalarning natijalarini chiqarish uchun ustunni nomlash mumkin. Bunday nomlar ustun psevdonimlari deyiladi (column alias).

Masalan: SELECT LastName, COUNT(*) CountOfNames
FROM STAFF GROUP BY LastName

Natijaviy naborda ikkinchi maydonning nomi, ya'ni familiyalar soni joylashgan maydonning nomi CountOfNames deb nomlanadi. WHERE yoki GROUP BY da psevdonimlar ishlatilmaydi. Ularda faqat real maydon nomlari ko'rsatiladi.

Jadval psevdonimlari. Har sira jadvalning to'liq nomini SELECT buyrug'ining har-xil qismlarida ko'rsatmasdan uning qisqartirilgan nomini ishlatish mumkin. Bunday qisqartirilgan nom jadval psevdonimi (table alias) deyiladi. U FROM kalit so'zi yordamida yaratiladi:

```
SELECT S.LastName, S.FirstNarae, S.FatherName, R.City  
FROM STAFF S, REGIONS R WHERE S.Zip = R.Zip
```

Ma'lumotlarni o'zgartirish. Jadvallarda saqlanayotgan ma'lumotlarni o'zgartirish uchun SQL UPDATE buyrug'i ishlatiladi. Bu buyriqni ishlatilishiga doir misol ko'ramiz:

```
UPDATE STAFF SET Zip = 8320
```

Bu operatororda STAFF jadvalining barcha qatorlari uchun Zip maydoniga 8320 kiritilayapti.

Yana bir misol:

```
UPDATE STAFF SET Salary = Salary + 100  
WHERE PosID IN (SELECT PosID FROM POSS  
WHERE PosLevel<3)
```

Bu misolda Salary maydoni 100 ga oshiriladi, agar lavozim darajasi 3 dan oshmasa.

Misol:

```
UPDATE STAFF SET TabNum = '00001', PosID = 2  
WHERE ID = 0
```

Bu operatororda ID = 0 bo'lgan qatordagi TabNum maydoniga '00001' va PosID maydoniga 2 qiymatlar kiritilayapti.

Qatorni o'chirish. Qatorni ochirish uchun DELETE buyrug'i ishlatiladi. Masalan, FAMILY jadvalidagi barcha qatorlarni o'chirish uchun quyidagi operatorni bajarish kerak:

```
DELETE FROM FAMILY
```

Shuningdek ma'lum shartni qanoatlantiruv yozuvlarni o'chirish uchun WHERE kalit so'zini qo'llash mumkin
DELETE FROM FAMILY WHERE ID = 0

Nazorat savollari

1. IBConsole ilovasi haqida ma'lumot bering.
2. Serverga ulanish qanday amalga oshiriladi?
3. Mavjud ma'lumotlar bazasini qayd qilish haqida ma'lumot bering.
4. InterBase da ma'lumotlar bazasini yaratish haqida ma'lumot bering.
5. Jadval qanday yaratiladi?
6. Jadvallarni o'chirish qanday amalga oshiriladi?
7. Indeks larni yaratish texnologiyasini tushintiring.
8. Ma'lumotlarni kiritish qanday amalga oshiriladi?
9. COMMIT va ROLLBACK buyruqlari qanday vazifani bajaradi ?
10. Ma'lumotlarni ko'rish qanday amalga oshiriladi ?
11. Ustunlardagi ifodalar haqida ma'lumot bering.
12. Jadvallarni birlashtirish usulini izohlab bering.
13. Ichma-ich so'rovlar so'rovlarga misol keltiring.
14. GROUP BY amali qanday vazifani bajaradi?
15. HAVING kalit so'zi ishlatilishiga misol keltiring.
16. ORDER BY xizmat so'zi qanday vazifani bajaradi?
17. Ma'lumotlarni o'zgartirish qaysi operator orqali amalga oshiriladi?
18. Qatorni o'chirish o'cherish operatorining ishlatilishiga misol keltiring.

9.5. Generatorlar

Generator – bu ketma –ket oshib boruvchi son bo'lib, GEN_ID() funkstiyasi yordamida jadvaldagi maydonga kiritiladi. Generatorlar ko'pincha birlamchi kalitga ega bo'lgan maydonlarga ularning takrorlanmaydigan son qiymatini xosil qilishda ishlatiladi.

InterBase ma'lumotlar bazasida ixtiyoriy sondagi generatorlar bo'lishi mumkin. Generatorlar o'z qiymatini tranzaktsiya vaqtida o'zgartiradi.

staff.gdb ma'lumotlar bazasi uchun uchta generator yaratamiz. Bu generatorlar STAFF, DEPS va POSS jadvallarining birlamchi kalitli maydonlari uchun qiymat hosil qiladi.

Buning uchun quyidagi buyruqni bajarish kerak:

```
CREATE GENERATOR STAFF_ID_GEN;
```

```
CREATE GENERATOR DEPS_ID_GEN;
```

```
CREATE GENERATOR POSS_ID_GEN;
```

Natijada ma'lumotlar bazasida uchta generator yaratiladi.

Bunga ishonch hosil qilish uchun **IBConsole** dasturining daraxtsimon strukturasi kuyidagi bandni tanlash kerak:

```
Servers > Local Server > Databases > STAFF.GDB >
```

Generators

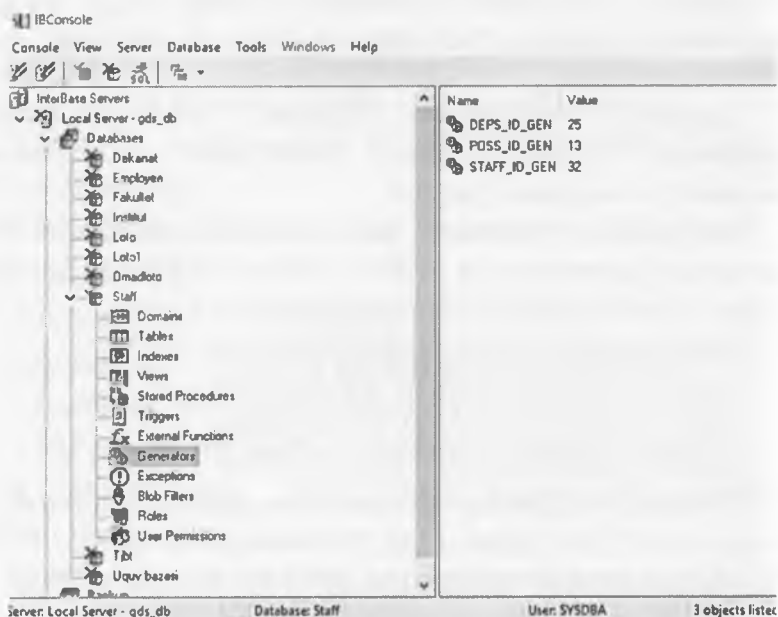
InterBasening eski versiyalarida generatorni to'g'ridan-to'g'ri o'chirib bo'lmaydi, shuning uchun generatorni o'chirishda boshqa xil usuldan foydalanamiz. Gap shundaki, foydalanuvchi tomonidan yaratilgan jadvallardan tashqari sistema tomonidan yaratilgan xizmatchi jadvallar xam mavjud bo'lib, generatorlar shu jadvalga bog'langan bo'ladi. Bu jadvallarni ko'rish uchun **IBConsole** dasturida quyidagi buyruqni bajarish kerak:

```
View > System Data
```

Yaratilgan generatorga murojaatlar RDB\$GENERATORS xizmatchi jadvaliga kiritiladi. Shu sababli qandaydir generatorni,

masalan, POSS_ID_GEN generatorini o'chirish uchun quyidaga buyruqni bajarish yetarli:

```
DELETE FROM RDB$GENERATOR  
WHERE RDB$GENERATOR_NAME = 'POSS_ID_GEN'
```



9.6.5-rasm. IBConsole dastur oynasida "Generators" ob'ektlarining ko'rinishi

Jadvalga yangi qator qo'shilganda generatorning qiymatini kandaydir yo'l bilan oshirish kerak bo'ladi. Bunday vazifani trigger yordamida amalga oshirish mumkin.

9.6. Triggerlar

Trigger- bu SQL tilidagi dastur bo'lib, aniq bir jadval ustidan aniq bir amalni bajaradi. Triggerlar jadvalda yangi qator yaratish, o'zgartirish va qatorni o'chirish paytlarida ishga tushishi mumkin.

1. STAFF jadvali yozuviga mos keladigan FAMILY va JOBS jadvallaridagi yozuvlarni o'chiradigan trigger yaratamiz:

```

SET TERM ^ ;
CREATE TRIGGER STAFFDelete FOR STAFF
BEFORE DELETE
AS BEGIN
DELETE FROM FAMILY
WHERE EmpID = OLD.ID;
DELETE FROM JOBS
WHERE EmpID = OLD.ID;
END ^

```

Triggerlarni tashkil qiluvchi ayrim elementlarni ko'rib o'tamiz. BEFORE kalit so'zi triggerning mos amallardan (INSERT, UPDATE yoki DELETE) oldin chakirilib bajarishini bildiradi. AFTER kalit so'zi ishlatilganda esa aksincha amaldan keyin trigger chaqiriladi.

OLD o'zgaruvchisi o'chirilayotgan yoki o'zgartirilayotgan qatorning ko'rsatilgan maydonga mos keladigan qiymatini bildiradi. Bu o'zgaruvchiga qarama-qarshi bo'lgan yana bitta NEW nomli maxsus o'zgaruvchi mavjud bo'lib, bu o'zgaruvchi INSERT yoki UPDATE buyruqlari bajarilayotganda yangi qiymatni ifodalaydi. SET TERM operatori yordamida dasturdagi operatorlar orasiga qo'yiladigan ajratgich belgi o'zgartiriladi. Avtomatik tarzda ";" belgi qabul qilinadi.

2. Endi STAFF, DEPS va POSS jadvallari uchun yangi qator qo'yganda GEN_ID funkstiyasini chaqiruvchi va kalit maydonga yoziladigan sonning qiymatini bittaga oshiruvchi triggerni yaratamiz.

```

SET TERM ^ ;
CREATE TRIGGER STAFFInsert FOR STAFF
BEFORE INSERT POSITION 0
AS BEGIN
NEW.ID = GEN_ID(STAFF_ID_GEN, 1);
END ^
CREATE TRIGGER DEPSInsert FOR DEPS
BEFORE INSERT POSITION 0
AS BEGIN

```

```

NEW.DeptID = GEN_ID(DEPS_ID_GEN/ 1);
END ^
CREATE TRIGGER POSSInsert FOR POSS
BEFORE INSERT POSITION 0
AS BEGIN
NEW.PosID = GEN_ID(POSS_ID_GEN, 1);
END ^

```

Bu erda POSITION kalit so'zi bitta jadval bilan bog'langan bir nechta triggerning bajarilish tartibini ifodalaydi. Yuqoridagi POSITION 0 bo'lgan holda trigger INSERT amali bog'liq barcha triggerlar orasida birinchi bo'lib bajariladi.

9.7. Indekslar va cheklovlar

InterBase da indeks yaratishda CREATE INDEX buyrug'i ishlatiladi. SELECT buyrug'i yordamida tez-tez ishlatilib turiladigan maydonlar uchun indekslar yaratilsa, amallarning bajarilishi tezlashadi. Bundan tashqari haddan ziyod ko'p indekslar yaratilsa ham teskari natija berishi, ya'ni ma'lumotlarni ajratishni tezlashtirish o'rniga sekinlashtirishi mumkin. InterBase da indeksli jadvallar bilan ishlashni tezlashtirish uchun maxsus foydali mexanizm, ya'ni vaqtincha indekslarni ajratish amali ishlatiladi. Bu esa modifikastiyalashgan jadvallarning ishlashini tezlashtiradi. Indeksni vaqtincha o'chirib qo'yish uchun quyidagi buyruq shlatiladi:

```
ALTER INDEX indeks nomi INACTIVE
```

Indeksni qayta yaratish uchun quyidagi buyruq ishlatiladi:

```
ALTER INDEX indeks nomi ACTIVE
```

Indeks ulanganda u yangidan tuziladi.

Cheklovlar. Cheklov (constraint) deb jadval maydonining olish mumkin bo'lgan ma'lumotlarini aniqlash mexanizmiga aytiladi. Cheklovlar CREATE TABLE yoki ALTER TABLE ADD buyruqlari orqali qo'yilishi mumkin. Agar cheklov indeklar bilan birgalikda ishlatiladigan bo'lsa, indeksni yopishdan oldin ALTER TABLE DROP buyrug'i yordamida cheklovni bekor qilish lozim.

Birlamchi kalit. Bunday cheklov qo'yilgan maydondagi ma'lumotlar takrorlanmaydi, ya'ni maydonda ikkita bir xil qiymatli ma'lumot yozilishiga yo'l qo'yilmaydi. Masalan, REGIONS jadvaliga birlamchi kalit qo'yish quyidagicha amalga oshiriladi:

```
CREATE TABLE REGIONS  
( Zip integer NOT NULL, PRIMARY KEY,  
...)
```

Bu berilgan holatda birlamchi kalit zip ustuniga tegishli. Bu amalni boshqacha ko'rinishda ham amalga oshirish mumkin:

```
CREATE TABLE REGIONS  
( Zip integer NOT NULL,
```

...

```
City varchar(20) NOT NULL,  
PRIMARY KEY (Zip))
```

Bu usul jadval yaratishda qo'llaniladi. Bundan tashqari birlamchi kalitni jadval strukturasini o'zgartirish orqali ham yaratish mumkin.

```
ALTER TABLE REGIONS  
ADD PRIMARY KEY (Zip)
```

Tashqi kalitlar. Tashqi kalit (foreign key) jadvalning tashqi kalit o'rnatilgan maydonidagi qiymatlar boshqa tashqi deb ataladigan jadvalda mavjud bo'lishi shart. Birlamchi kalitdan uning farqi shundaki, tashqi kalitli maydondagi qiymatlar takrorlanishi mumkin. Tashqi jadvaldagi maydon qiymatlari esa takrorlanmas bo'lishi kerak. Jadvalga tashqi kalitning qo'shilishi avtomatik tarzda undagi kalit maydonda ikkilamchi indeksni hosil qiladi.

Birlamchi kalitga o'xshash tarzda tashqi kalitni ham CREATE TABLE buyrug'i yordamida yaratish mumkin yoki jadval mavjud bo'lsa ALTER TABLE ADD FOREIGN KEY buyrug'i yordamida hosil qilish mumkin.

1. STAFF jadvali uchun tashqi kalit yaryatamiz:

```
ALTER TABLE STAFF  
ADD FOREIGN KEY (PosID) REFERENCES POSS
```

Bu cheklov PosID maydoniniga tashqi kalit o'rnatadi va POSS jadvalining shunga mos maydoniga murojaat qilinadi. Bu shuni anglatadiki, STAFF jadvaliga kiritilayotgan identifikatorlar POSS jadvalining unga mos maydonida mavjud bo'lishi shart va uni POSS jadvalidan o'chirib bo'lmaydi. Bunday biriktirib qo'yish murojaatli butunlik (referential integrity) deyiladi.

2. Shunga o'xshash tashqi kalitni DepID maydoni uchun ham yaratamiz:

```
ALTER TABLE STAFF
```

```
ADD FOREIGN KEY (DepID) REFERENCES DEPS(DeptID)
```

Bu misolda STAFF va DEPS jadvallaridagi maydonlarning nomlari har xil. Shuning uchun DEPS jadvalining nomidan so'ng maydon nomi keltirilgan.

3. Novbatdagi buyruqda STAFF jadvali pochta indeksi uchun tashqi kalit hosil qilinmoqda:

```
ALTER TABLE STAFF
```

```
ADD FOREIGN KEY (Zip) REFERENCES REGIONS
```

Qiymatlarni tekshirish. Qiymat bo'yicha cheklovlar jadval yaratish vaqtida yoki yaratib bo'lingandan so'ng ham qo'yilishi mumkin. Buning uchun CHECK operatori ishlatiladi. Masalan, STAFF jadvali yaratish vaqtida uning Zip maydoni uchun quyidagicha cheklov kiritish mumkin:

```
CREATE TABLE
```

```
(...
```

```
CHECK (Zip BETWEEN 1 AND 99999))
```

1. Endi bu buyruqni ALTER TABLE buyrug'i yordamida yaratamiz:

```
ALTER TABLE STAFF
```

```
ADD CONSTRAINT INVALID ZIP CHECK
```

```
(Zip BETWEEN 1 AND 99999)
```

2. Oylik maosh uchun cheklov kiritamiz:

```
ALTER TABLE STAFF
```

```
ADD CONSTRAINT INVALID SALARY CHECK (Salary >= 0)
```

Nomning qarama-qarshi ma'nosiga e'tibor bersak, cheklov uchun tanlangan nom (masalan, "INVALID ZIP" frazasi nomunosib pochta manzilini bildiradi

Ma'lumot berishdagi bunday nomlar foydalanuvchi o'rnatilgan cheklovning buzilganligi haqidagi muammoning manbaini aniqlay olishi uchun kerak

3. JOBS jadvali uchun cheklov kiritamiz. Undagi so'ngi ishlab turgan tashkilotni aniqlovchi maydon 0 yoki 1 qiymat qabul qilishi kerak:

```
ALTER TABLE JOBS
```

```
ADD CONSTRAINT INVALID CURORG CHECK (CurOrg  
IN ('1', '0'))
```

9.8. Saqlanuvchi prosteduralar

Saqlanuvchi prostedura (stored procedure) – bu ixtiyoriy hajmdagi kompilyastiyalangan SQL tilidagi dastur bo'lib, boshqa ob'ektlar bilan birgalikda ma'lumotlar bazasida saqlanadi. Saqlanuvchi prosteduralar ikkita asosiy turga bo'linadi: tanlash prosteduralari (select procedure) va bajariluvchi prosteduralar (executable procedure). Tanlash prostedurasi SELECT operatorida jadval o'rnida ishlatiladi va ma'lumotlar to'plamini ifodalaydi. Bajariluvchi prostedura uchun bu narsa shart emas. Saqlanuvchi prosteduralar CREATE PROCEDURE buyrug'i yordamida yaratiladi va DROP PROCEDURE buyrug'i yordamida o'chiriladi.

Tanlash prosteduralari. Staff .gdb ma'lumotlar bazasida bo'ysotuvchi bo'limlar ro'yxatini hosil qiluvchi protsedurani yaratamiz. Bunda shuni e'tiborga olish kerakki, korxona iyerarxik strukturali bo'limlarga ega.

DEPS jadvalining ParentDeptID maydonida tegishli bo'limning identifikatori ko'rsatiladi. Agar tegishli bo'lim mavjud bo'lmasa, bo'lim ierarxiyaning yuqorisida bo'ladi va ParentDeptID maydoniga 0 qiymati kiritiladi.

```
SET TERM ^;
```

```
CREATE PROCEDURE ChildDeptsList (pDeptID integer)
```

```

RETURNS (rDeptFullName varchar(100))
AS BEGIN
FOR SELECT DeptFullName FROM DEPS
WHERE ParentDeptID =:pDeptID
INTO :rDeptFullName
DO SUSPEND;
END

```

Bu prostedurada yuqori bo'limning identifikatori bo'lgan pDeptID parametr uzatiladi va natija sifatida bo'yso'nuvchi korxonalar ro'yxati qaytariladi. Tanlash prostedurasida qaytariluvchi ma'lumotlar RETURNS kalit so'zi orqali aniqlanadi. Kiritiluvchi parametrlar esa prostedura nomidan keyin yoziladi.

Natijaviy qatorni hosil qilish uchun FOR SELECT . . . DO konstrukstiyasi ishlatiladi. SUSPEND buyrug'i chiqariladigan ma'lumotlarni buferga o'qib oladi.

Кучанинг номини берганда унда яшовчи хodimлар рўйхатини олиш дастурини тузамиз.

```

SET TERM ^ ;
CREATE PROCEDURE "TANLASH"
( "KO_CHA" VARCHAR(30))
RETURNS
("FAM" VARCHAR(25), "ISM" VARCHAR(25), "OTASI"
VARCHAR(25))
AS begin
FOR SELECT LASTNAME, FIRSTNAME, FATHERNAME
FROM STAFF
WHERE (street="KO_CHA") INTO : "FAM", : "ISM", : "OTASI"
DO BEGIN SUSPEND; END
End ^
SET TERM ; ^
Protsedurani chaqirish uchun
SELECT * FROM tanlash( 'Алпомиш' )
кўринишида сўров ёзилади

```

Bajariluvchi prostedurlar. Bajariluvchi prosteduralarda RETURNS operatori ishlatilmasligi mumkin. Xodim haqidagi ma'lumotlarni o'chirish prostedurasini tuzamiz.

```
SET TERM ^;
```

```
CREATE PROCEDURE DeleteEmp (pEmpID integer)
```

```
AS BEGIN
```

```
DELETE FROM STAFF WHERE ID = :pEmpID;
```

```
END ^
```

ID qiymati 7 ga teng qatorni ochirish uchun quyidagi buyruq beriladi:

```
EXECUTE PROCEDURE DeleteEmp 7
```

"Talaba" nomli jadvalga yangi yozuv kiritish uchun saqlanuvchi procedura tuzamiz:

```
CREATE PROCEDURE "TALABAINSERT"
```

```
( "F" VARCHAR(15), "T" VARCHAR(15), "T" DATE, "S"  
INTEGER)
```

```
AS BEGIN
```

```
INSERT INTO "TALABA" ("FAM","ISM","TSANA","STEP")
```

```
VALUES (:F",:"T",:"T",:"S");
```

```
END ^
```

Nazorat savollari

1. InterBaseda ma'lumotlar bazasini yaratish qanday amalga oshiriladi ?
2. Generatorlar haqida ma'lumot bering.
3. Triggerlar qanday hollarda ishlatiladi.
4. Indeksar qanday hollarda yaratiladi.
5. Cheklovlarni ishlatilishiga misol keltiring.
6. InterBaseda saqlanuvchi prostedura strukturasi qanday ko'rinishda b'ladi?
7. Saqlanuvchi prosteduralarda ishlatiladigan operatorlarni izohlang.
8. Tanlash prosteduralari haqida ma'lumot bering.
9. Bajariluvchi prostedurlaraga misol keltiring.

X. DELPHIDA MA'LUMOTLAR BAZASI BILAN ISHLASH

10.1. MB bilan ishlovchi instrumental vositalar va komponentlar

Delphida MB bilan ishlovchi vositalarni umuman 2 ta katta guruhga bo'lish mumkin:

1. Instrumental vositalar;
2. Komponentlar.

MB bilan amallarni bajarish uchun Delphi tizimida qo'yidagi instrumental vositalarni ishlatiladi:

1. **Borland DataBase Engine (BDE);**
2. **BDE Administrator-** BDE ning turli xil parametrlarini o'rganish uchun utilita;
3. **Database Desktop-** jadvallarni, SQL-so'rovlarni va QBE-so'rovlarni hosil qiluvchi va tahrirlovchi programma;
4. **SQL Explorer** - MB provodnigi. U MBni ko'rish va tahrirlash imkonini beradi;
5. **SQL Builder** - SQL-so'rovlarni vizual tuzishni tashkil qiluvchi programma;
6. **SQL Monitor** - Masofaviy MBga SQL-so'rovlarning bajarilish tartibini tekshirib boruvchi programma;
7. **Data Pump** - MB orasida axborotni uzatish uchun mo'ljallangan programma;
8. **InteBase Windows Interactive SQL (WISQL)-** masofaviy MBni boshqarish uchun programma.
9. **InterBase Server Manager** - masofaviy serverdan boshqarish uchun programma.
10. **SQL Links-** masofaviy sanoat MBBTga murojaat uchun drayverlar. Masalan, Microsoft SQL Server yoki Orasle. Delphi bilan tekinga tarqatiladigan **InterBase** sanoat serveriga **SQL Links** drayverini ishlatmasdan ham tashkil etish mumkin.

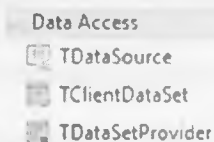
11. **Losal InterBase Server** - Borland InterBase SQL serverining lokal versiyasi.

12. **InterBase Server for Windows** - Borland InterBase SQL serverining ko'p foydalanuvchili versiyasi.

MB bilan ishlovchi komponentlar. MB ilovalarini hosil qiluvchi komponentlarni ko'ramiz. Bu komponentlar ham boshqa komponentlarga o'xshab vizual va novizual komponentlarga bo'linadi.

MB bilan ishlash imkonini beruvchi komponentlar Data Access, Data Controls, DataSnap, Desicion Cube, QReport va InterBase komponentlar palitrasida joylashgan. Ayrim komponentlar masofaviy MB bilan "klient-server" arxitekturasida ishlash uchun mo'ljallangan. Quyida Delphi uchun MBga mo'ljallangan komponentlar palitrasi keltirilgan.

Data Access sahifasida ma'lumotlarga murojaatni tashkil qilishga mo'ljallangan novizual komponentlar joylashgan (10.1.1-rasm):



10.1.1-rasm. Data Access sahifasining komponentlari.

Data Source - axborot manbasi;

table - MB jadvaliga asoslangan axborot to'plami;

Query - SQL-so'rovga asoslangan axborot to'plami;

StoredProc - serverning saqlanuvchi prostedurasini chaqirish;

Database - MB bilan bog'lanish

Session - MB bilan ishlovchi joriy seans;

Batchmove - yozuvlar guruhi ustida amal bajarish;

UpdateSQL - SQL-so'rovga asoslangan axborot to'plamining modifikastiyasi;

NestedTable - ichma-ich jadval;

Data Controls sahifasidagi ma'lumotlarni boshqarish uchun mo'ljallangan vizual komponentlar joylashgan (10.1.1-rasm):

DBGrid - setka (to'r);
 DBNavigator - navigastiyali
 interfeys;

Dbtext - yozuv;
 DBEdit - 1 qatorli muharrir;
 DBMemo - ko'p qatorli muharrir;
 DBImage - grafik tasvir;
 DBListbox - oddiy ro'yxat;
 DBCombobox

kombinastiyalashgan ro'yxat;

DBCheckbox - mustaqil o'tkazgich;
 DBRadioGroup - bog'liq
 o'tkazgichlar guruhi;

DBLookupListBox - boshqa axborotlar

to'plamining maydoni bo'yicha shakllanadigan oddiy ro'yxat;

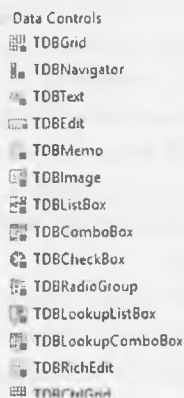
DBLookupComboBox - boshqa axborotlar to'plamining
 maydoni bo'yicha shakllanadigan kombinastiyalashgan ro'yxat;

DBRichEdit - to'la funkstional matn muxarriri;

DBCtrlGrid - modifikastiyalashgan to'r;

DBChart - diagramma.

DataSnap Client sahifasida masofaviy MBni boshqarish
 uchun mo'ljallangan komponentlar joylashgan (10.1.3-rasm):



10.1.2-rasm. Data Controls
 sahifasining komponentlari.



10.1.3-rasm. Data Snap Client sahifasining komponentlari.

ClientDataSet - axborotni klient darajasida kiritish;

DCOMConnection - ko'p oqimli ilovalar uchun masofaviy
 server bilan bog'lanish;

SocketConnection - Windows coketi orqali masofaviy server bilan bog'lanish;

DataSetProvider - axborotni kiritish provayderi;

SimpleObjectBroker - oddiy ob'ekt brokeri;

WebConnection – web-server bilan bog'lanish;

CorbaConnection - CORBA-klientni ulash.

fastreport sahifasining komponentlari:



10.2.DataBase Desktop yordamida MBni hosil qilish

DataBase Desktop odatda, Delphining asosiy menyusidagi Tools bo'limida keltirilgan bo'ladi. Agar u erda bo'lmasa, u holda Tools | Configure Tools komandasi orqali DataBaseDesktopni chaqirish mumkin. U chaqirilgach, uning oynasi ekranga chiqadi. Windows 10 uchun Internetdagi maxsus saytlardan olib o'rnatish mumkin.

Endi Paradox7 MBBT yordamida MB jadvalini hosil qilishni ko'ramiz. Paradox7 da MB bu alohida katalog bo'lib, unda kengaytmasi .db bo'lgan fayllar-jadvallar joylashadi. Shuning uchun yangi katalog ochib qo'yish kerak bo'ladi (masalan, "provodnik" dasturi yordamida). So'ngra File| New komandasi beriladi. Bunda 3 xil variantli qism menyu chiqadi:

QBE Query ...	So'rovlarni vizual hosil qilish va ularni faylga yozish
SQL File	So'rovlarni SQL tilida hosil qilish va uni faylga yozish
Table	Yangi jadval hosil qilish

Bu yerdan Tableni tanlash kerak. Bu holda kichikroq dialog oynasi ochiladi. Undan Paradox7 ni tanlaymiz. So'ngra hosil bo'lgan yangi oynadan biz jadvalimizning tuzilishini, ya'ni undagi maydonlar va ularning toifalarini va boshqa ko'pgina axborotni kiritishimiz lozim bo'ladi.

Maydonlarni boshqarish. Har bir hosil qilinadigan jadval uchun maydonlar nomi, ya'ni identifikatori aniqlanishi lozim. Maydon nomi 25 ta simvolgacha bo'lishi va birinchi simvoli probel bo'lmasligi kerak. So'ngra bu maydonning toifasi aniqlanadi.

Buning uchun Type bo'limiga o'tiladi va sichqonchaning o'ng tugmasi bosiladi. Maydon toifasi sifatida quyidagilardan birini berish mumkin:

Kalit maydonlar eng oxirgi ustunida «*» belgisi bilan tugashi lozim.

10.2.1-jadval

Paradox7 MBBTdagi maydon toifalari

Belgi - lanish i	Ro'yxatda berilishi	Tavsifi
A	Alpha	1 dan 24 gacha bo'lgan ixtiyoriy ASCII simvollaridan tashkil topgan matnaviy maydon
N	Number	-10^{307} - 10^{308} bo'lgan haqiqiy sonlar
\$	Money	Pul birligi ko'rinishidagi musbat yoki manfiy sonlar
S	Short	$-32\,767 \div +32\,767$ gacha bo'lgan butun sonlar
I	Long Integer	$-2\,147\,483\,648 \div +2\,147\,483\,647$ gacha bo'lgan butun sonlar
#	BCD	BCD (Binary Coded Decimal) formulasidagi sonlar
D	Date	Sana ko'rinishidagi qiymatlar
T	Time	Vaqt ko'rinishidagi qiymatlar

@	Timestamp	Vaqt va sanani saqlovchi qiymatlar
M	Memo	Chegaralanmagan uzunlikdagi matnlarni saqlaydi
F	Formatted memo	Chegaralangan uzunlikdagi formatlangan matnlarni saqlaydi
G	Graphic	.bmp, .pex, .tif, .gif yoki .eps formatli fayllardagi rasmlarni .vmr formatga ko'chiradi
O	OLE	OLE toifasidagi ma'lumotlar. Bu tasvirlar, tovushlar, hujjatlar
L	Logical	Mantiqiy maydon. True yoki false qiymatiga ega
T	Autoincrement	Avtomatik ravishda 1 butun songa oshiradi
B	Binary	MB fayllardagi ikkilik axborotni saqlaydi. Bularda tovush yoki ixtiyoriy boshqa ma'lumotlar bo'lishi mumkin
Y	Bytes	Binary dan farqli tomoni fayllarda emas jadvallarda saqlaydi.

Jadval xossalarini aniqlash. Table propertiesda jadval xossalari aniqlanadi. Undan quyidagilarni tanlash mumkin:

1.Validity checks -qiymatlarning to'g'riligini tekshirish.

Unda maydonni quyidagi xarakteristikalarini berish mumkin:

required field	bu indikorda maydonning har bir yozuvi bor bo'lgan maydonlar belgilanadi
Minimum	Minimal qiymat. Bu xossani sonli maydonlarga berish foydali
Maximum	Maksimal qiymat. Bu xossani sonli maydonlarga berish foydali.
default	Jamlik bo'yicha qiymat. Bu xossani sonli va mantiqiy maydonlar uchun berish foydali
picture	Axborotni kiritish uchun shablon. Bu xossani telefon nomer uchun berish mumkin
assist	Bu tugmacha picture shablonini hosil qilishga yordam beruvchi dialog oynani chaqiradi

2. **Table Lookup** - ko'rish jadvali.

Bu bo'lim jadvalning maydonini boshqa jadvalning qandaydir maydoni bilan bog'laydi.

3. **Secondary Indexes** - ikkilamchi indekslar.

Bu bo'lim keyingi ishlarda oson bo'lishi uchun ikkilamchi indekslar hosil qiladi. Yangi ikkilamchi indeks hosil qilish uchun Define-tugmasi bosiladi. So'ng indexed fields oynasidan maydonlar ro'yxati tanlanadi. Indeks shakllangach, OK tugmachasi bosiladi va yangi oynaga indeksning nomi kiritiladi.

4. **Referential Integrity** - ishoralar darajasidagi yaxlitlik.

Nazorat savollari

5. Delphi dasturi MBga qanday ulanadi?
6. MB bilan ishlovchi vositalarga nimalar kiradi?
7. MB bilan ishlovchi komponentlarga nimalar kiradi?
8. Maydonlarni boshqarish qanday amalga oshiriladi?
9. Jadvalning qaysi xossalarini aniqlash kerak bo'ladi?

10.3.BDE yordamida MB ga ulanish va ma'lumotlarni diagrammada tasvirlash

Ma'lumotlar bazasiga BDE texnologiyasi bo'yicha ulanish uchun BDE politrasi joylashgan komponentlar ishlatiladi. Bu texnologiya bo'yicha ulanishni namoyish qilish uchun DBDEMOS MBda joylashgan ta'minlovchilar (customer.db) va byurtmalar (orders.db) haqidagi ma'lumotlar joylashgan jadvallardan foydalanamiz. Buning uchun quyidagi ketma-ketlikdagi amallarni bajaramiz.

1. Kompyuter qattiq diskida papka yaratish.
2. Delphi ni ishga tushirib yangi loyiha yaratamiz.

3. Formaga BDE politrasidan Ttable komponentini, Data Access politrasidan TdataSource komponentasini, Data Control politrasidan TDBGrid va TDBNavigator komponentalarini keltirib qo'yamiz. Komponentlar xususiyatlarini quyidagicha o'zgartirami:

```
Table1.Name := OrdersTable;
```

Name o'zgargandan keyin komponentaning nomi o'zgaradi va uning xossalarini quyidagiday o'zgartiramiz.

```
OrdersTable.DatabaseName := DBDEMOS;
```

```
OrdersTable.TableName := orders.db;
```

```
DataSource1.Name := OrdersSource;
```

```
OrdersSource.DataSet := OrdersTable;
```

```
DBGrid.Name := OrdersGrid;
```

```
OrdersGrid.DataSource := OrdersSource;
```

```
DBNavigator.Name := DBNav;
```

```
DBNav.DataSource = OrdersSource;
```

```
DBNav.ShowHint := True;
```

4. Modulga unit1 nomini loyiha fayliga esa bdedemo nomini berib faylni oldin yaratilgan papkaga saqlaymiz.

5. OrdersTable komponentasining Active xususiyatiga true qiymatini o'rnatamiz. SHundan so'ng OrdersGrid komponentasi orders.db jadvali ma'lumotlarini ekranga chiqaradi.

6. Ilovani ishga tishiramiz va DBNavigator ning imkoniyatlarini o'rganib chiqamiz.

7. Jadval ustunlaridagi yozuvlarni o'zbek tiliga o'girishuchun OrdersGrid komponentasidan foydalanamiz. Buning uchun bu komponentning Columns xususiyatini tanlaymiz va ochilgan oynada Add All Fields belgisiga sichqoncha strelkasini keltirib chap knopkasini bosamiz. Natijada oynaga maydon nomlari qo'yiladi. Kerakli maydonlani tanlab, ob'ekt inspektorining Title xususiyatidagi Caption xususiyatiga kiramiz va o'zbek tilidagi nom yozamiz.

8. Ilovani ishga tushirib natijalarni ko'ramiz.

9. Diagramma yaratish uchun formaga TDBChart komponentasin keltirib qo'yamiz. Uning xususiyatlarin quyidagicha o'rnatamiz.

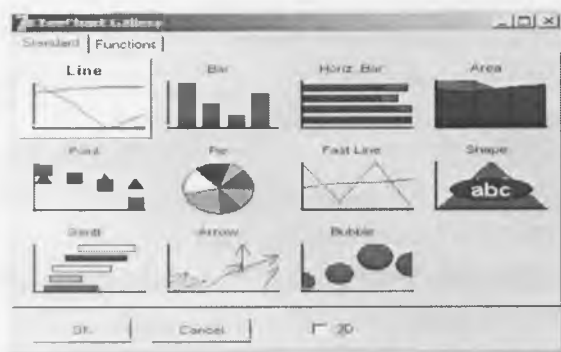
```
DBChart1.Name := DataChart;
```

```
DataChart.ShowHint := True;
```

10. Diagrammaga sichqoncha strelkasini keltirib, o'ng knopkasini bosamiz va ochilgan konteks menyusidan Edit Chart ni tanlaymiz.

11. Ma'lumotlarning yangi diagramma qatorini kiritish uchun Editing DataChart oynasiga kirib undagi Chart vkladkasidan Series sahifasini tanlaymiz. Sichqoncha strelkasini Add tugmasiga keltirib bosamiz.

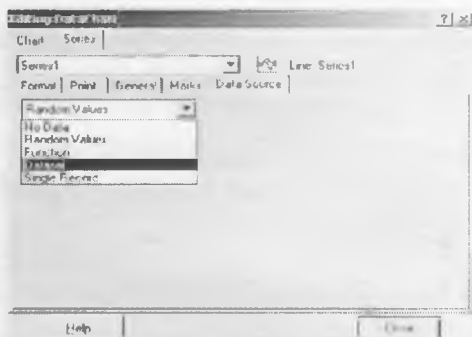
12. Natijada ochilgan TeeChart Gallery oynasidagi Standard sahifasidan Line diagramasini tanlaymiz.



10.3.1-rasm. TeeChart Gallery oynasi

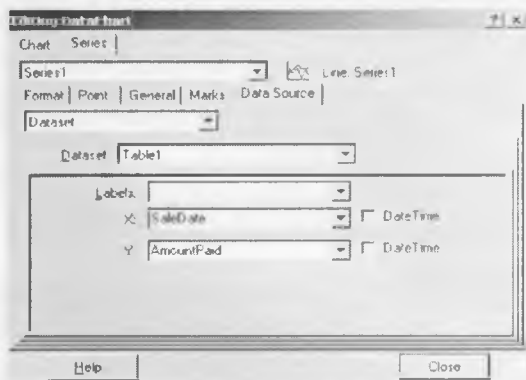
13. Editing DataChart oynasin Close tugmasi yordamida yopamiz.

14. Diagrammani MB jadvali bilan bog'lash uchun konteks menyudan yana Edit Chart ni tanlaymiz. Editing DataChart oynasidan Series vkladkasini tanlaymiz. Undagi ro'yxatdan Series1 ni tanlaymiz. DataSource sahifasiga o'tib, undan DataSet qatorini tanlaymiz.



10.3.2-rasm. Editing DataChart oynasi

15. Shundan so'ng DataSet elementi uchun ro'yxatdan OrdersTable qiymatini tanlaymiz. X elementi uchun SaleDate, Y elementi uchun AmountPaid qiymatini tanlaymiz. SaleDate va AmountPaid lar orders.db jadvalining maydonlaridir. Bu ma'lumotlar ekranda tasvirlanadi.



10.3.3-rasm. Editing DataChart oynasida qiymatlarni tanlash

Vaqtni formada aks ettirish uchun formaga System politrasiidan Timer komponentasini keltirib qo'yamiz. Interval xususiyati uchun hodisalar chastotasining qiymatini 1 dan 64435 gacha bo'lgan qiymatlarni kiritish mumkin. Timer

komponentasida bitta hodisa mavjud bo'lib u OnTimer orqali o'rnatiladi. OnTimer hodisasiga sichqonchani keltirib ikki marta bosiladi va begin va end orasiga

TimeLabel1.Caption:= TimeToStr(Time);
kodini yozamiz. Bunda TimeLabel1 yangi kiritilgan TLabel komponentasi.

10.4. Rave Reports hisobot generatori

Ma'lumotlar bazasidagi ma'lumotlar uchun hisobot tayyorlab uni printerda chop qilish uchun dastlab biror bir MBBTda MB yaratamiz va quyidagi amallarni bajaramiz.

1. Ishchi diskda HISOBOT nomli papka yaratamiz.

2. DELPHIni ishga tushiramiz va yangi loyiha yaratamiz. Formaga va modulga nom beramiz.

3. Formaga bazaga ulanish uchun kerakli komponentlarni keltirib qo'yib, oldingi darslarda bajarilgandek MBga ulanamiz.

4. Rave Reports utiliti hisobotlar tayyorlash uchun ishlatiladi. Bu utilit bilan ishlovchi komponentalar Rave politrasida joylashgan. Unga ulanishdan oldin MB jadvallarini aktiv holga keltirish lozim. Formaga RvDataSetConnection komponentasini keltirib qo'yamiz.

5. RvDataSetConnection komponentasining DataSet xususiyatidan pechatga chiqarmoqchi bo'lgan jadvalni (Adotable1) tanlaymiz.

6. Hisobotni ishlab chiqish uchun Rave Reports (Tools - Rave Designer) utilitini ishga tushiramiz.

7. Bu utilitning oynasi ochilgandan so'ng File — New Data Object buyrug'ini bajaramiz.

8. Buyruq bajarilishi natijasida ochilgan **Data Connections** oynasidan Direct Data View elementini tanlaymiz va NEXT tugmasini bosamiz.

9. Keyingi qadamda Data Connections oynasidan RvDataSetConnection1 elementini tanlaymiz va Finish

tugmasini bosamiz.

10. Rave Reports utiliti interfeysi oynasining o'ng tomonida hisobot loyihasining daraxtsimon strukturasi joylashgan. Undagi Data View Dictionary tuguniga sichqoncha strelkasini keltirib bosamiz va natijada u yoyiladi. U erdan DataView1 tugunini tanlaymiz. Bu tugun ham yoyilib, unda Adotable1 jadvalining maydonlari ochiladi.

11. Hisobotni qadam ba qadam tayyorlovchi Rave Reports ustasini ishga tushirish uchun Tools — Report Wizards — Simple Table buyrug'ini bajaramiz.

12. Simple Table oynasidan DataView1 ni tanlaymiz va Next tugmasini bosamiz.

13. Novbatdagi qadamda ochilgan Simple Table oynasidan chiqarilishi kerak bo'lgan maydonlarga bayroqcha o'rnatamiz va Next tugmasini bosamiz.

14. Keyingi qadamda Report Title qatoriga hisobotning nomini kiritamiz. Masalan, "Talabalar haqida ma'lumot".

15. Novbatdagi qadamda hisobot materiallarining shirifti tanlanadi va Generate tugmasi bosiladi. Natijada hisobot ustasi hisobot loyihasini avtomatik tarzda tayyorlab beradi. Kerak bo'lganda loyihadagi ob'ektlarni o'zgartirish mumkin.

16. Novbatdagi qadamda loyiha ishchi papkaga yozib qo'yiladi. Rave Reports utilitining hisobot fayli rav kengaytmaga ega bo'ladi.

17. Hisobotning qanday darajada yaratilganligini File — Execute Report buyrug'i orqali tekshirib ko'rish mumkin.

18. Rave Reports bilan ishni tugatgandan so'ng DELPHI daturiga qaytamiz.

19. Formaga Rave politrasi dan TrvProject komponentini keltirib qo'yamiz.

20. Bu komponentning ProjectFile xususiyatiga yaratilgan hisobot faylining nomini o'rnatamiz. U ".rav" kengaytmali bo'ladi.

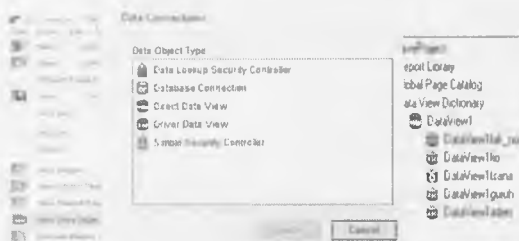
21. Formaga Tbutton ya'ni tugma komponentini keltirib qo'yamiz va unga

RvProject1.Execute; kodni yozamiz.

22. Daterni ishga tushirib natijalarni tekshirib ko'ramiz.



10.4.1-rasm. Delphi formasida Rave Report komponentlarining joylashuvi



10.4.2-rasm. Rave Report hisobot generatorida hisobotni yaratish bosqichlari



10.4.3-rasm. Rave Reportda hisobot yaratish bosqichlari



10.4.4-rasm. Rave Reportda hisobot yaratish bosqichlari

10.5. INTERBASE va DBEXPRESS palitrasi komponentlari yordamida INTERBASE ma'lumotlar bazasi bilan ishlash

INTERBASE politrasi komponentlari bilan ishlash.

Delphining INTERBASE palitrasi komponentlari yordamida INTERBASE MBBTga ulanish mumkin. InterBase komponentlaridan foydalanish quyidagi imkoniyatlarni yaratadi:

- API InterBasedan to'g'ridan-to'g'ri foydalanish evaziga tezlikning oshishi;
- yaxshilangan tranzakciyadan foydalanish;
- MBning holati haqida sistema jadvallariga murojaat qilmasdan to'g'ridan - to'g'ri ma'lumot olish;
- MBga beriladigan so'rovlarni kuzatib borish (TIBSQLMonitor)

Bu komponentlar yordamida INTERBASEga ulanish quyidagicha amalga oshiriladi.

1. Ishchi diskda papka yaratish.

2. InterBaseni ishga tushirib Employee.gdb ma'lumotlar bazasi borligini va uning aliasini aniqlaymiz.

3. DELPHIning ishga tushiramiz va yangi loyiha yaratamiz.

4. Interbase palitrasi komponentlari orasidan IBDatabase komponentini formaga keltirib qo'yamiz. IBDatabase xususiyatini quyidagicha o'zgartiramiz.

IBDatabase1.LoginPrompt:= false.

Formaga IBTransaction komponentini keltirib qo'yamiz. IBTransaction komponenti InterBase serveriga ulanishda tranzaksiya bilan boshqarish vositalarini inkapsulyatsiya qiladi. Uning xususiyati qiymatini quyidagicha beramiz.

IBTransaction1.DefaultDatabase := IBDatabase1

5. Novbatdagi bosqichda formaga IBTable komponentini qo'yamiz. IBTable ning xususiyatlarini quyidagicha o'rnatamiz.

IBTable1.Database := IBDatabase1

IBTable1.DefaultTransaction := IBTransaction1

IBTable1.TableName := Employee

IBTable1.Active := True

6. Formaga DataSource komponentini qo'yamiz. Uning xususiyatlarini quyidagicha o'rnatamiz.

DataSource1.DataSet := IBTable1

7. Ma'lumotlarni tasvirlash uchun formaga DBGrid komponentini qo'yamiz va uning xususiyatlarini quyidagicha o'rnatamiz.

DBGrid1.DataSource := DataSource1

8. Loyihani kompilyatsiya qilamiz natijalarni tekshirib ko'ramiz.

9. MBga so'rov berishni tekshirib ko'rish uchun formaga IBQuery komponentini qo'yamiz. Uning xususiyatlarini quyidagicha o'rnatamiz.

IBQuery1.Database := IBDatabase1

IBQuery1.SQL := "select * from EMPLOYEE"

IBQuery1.Active := true

10. Formaga DataSource komponentini qo'yamiz va uning xususiyatlarini quyidagicha o'zgartiramiz.

DataSource2.DataSet := IBQuery1

DBGrid1.DataSource := DataSource2

11. Interbase MB haqida ma'lumot olish uchun formaga IBDatabaseInfo va Memo komponentlarin qo'yamiz.

IBDatabaseInfo ning xususiyatini quyidagidek o'zgartiramiz.

IBDatabaseInfo1.Database := IBDatabase1

Form1 ning OnActivate() metodiga quyidagi kodni yozamiz.

Memo1.Lines.Clear();

Memo1.Lines.Add(IBDatabaseInfo1.Version);

Memo1.Lines.Add(IBDatabaseInfo1.DBFileName);

Memo1.Lines.Add(IBDatabaseInfo1.PageSize);

Dasturni bajarib natijalarni tekshirib ko'ramiz.



10.5.1-rasm. Interbase MBBT ga ulanish uchun Delphi formasiga komponentlarni joylashtirish

DBExpress palitrasi komponentlari yordamida InterBASE ma'lumotlar bazasiga ulanish. DBExpress palitrasi komponentlari yordamida InterBASE ma'lumotlar bazasiga "Mijoz-server" arxitekturasi bo'yicha ulanishni namoyish qilamiz. Ma'lumotlar bazasiga dbExpress texnologiyasi bo'yicha ulanish dinamik bibliotekalar ko'rinishidagi maxsus drayverlar orqali amalga oshiriladi.

1. Ishchi diskda yangi papka yaratamiz.
2. IBConsole daturini ishga tushirib unda demonstracion InterBase\examples\Database\Employee.gdb MB borligini tekshirib ko'ramiz.

3. Delphini ishga tushirib formaga dbExpress palitrasida joylashgan SQLConnection komponentini qo'yamiz.

4. Uning xususiyatlarini quyidagicha o'rnatamiz.

SQLConnection1.ConnectionName := IBConnection

SQLConnection1.LoginPrompt := false

SQLConnection1.driver := Interbase;

SQLConnection1.driver.database:=

C:\CodeGear\InterBase\examples\databases\employee.gdb

5. Formaga SQLTable komponentini qo'yamiz va uning xususiyatlarini quyidagicha o'rnatamiz.

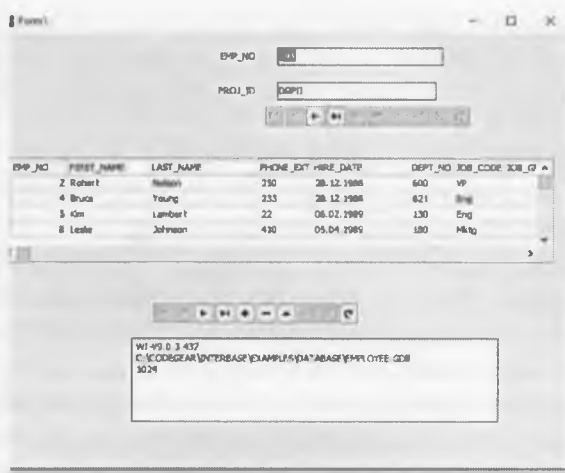
SQLTable1.SQLConnection := SQLConnection1

SQLTable1.TableName := EMPLOYEE_PROJECT

SQLTable1.Active :=true

EMPLOYEE_PROJECT jadvalidagi ma'lumotlarni tasvirlash uchun jadvalga ikkita Tlabel va ikkita Tedit komponentlarini qo'yamiz.

Labelning Caption xususiyatiga maydonlarning nomlarini (EMP_NO va PROJ_ID) beramiz. DataSource1 va DBNavigator1 komponentlarini bir-biriga bog'laymiz. DataSource1ni SQLTable1 ga ulaymiz. DBNavigator1ning orqaga harakatlantiradigan strekalarini ta'qiqlash kerak, chunki DBEXPRESS texnologiyasida kursor bir tomonlama harakatlanadi.



10.5.2-rasm. Delphi formasiga komponentlarni joylashtirish

10.6. XML-fayli ma'lumotlarini jadval ko'rinishida ifodalash

DELPHIda XML formatida saqlanayotgan ma'lumotlarni jadval ko'rinishida ifodalovchi ilova yaratamiz. Manba fayli sifatida Delphi distributivi tarkibiga kiruvchi biolife.xml faylini olamiz. U c:\Program Files (x86)\Common Files\CodeGear Shared\Data\ papkasida joylashgan.

1. Ishchi diskda xmldemo nomli papka yaratamiz.
2. Delphi ni ishga tushirib yangi loyiha yaratamiz.
3. XMLformatidagi ma'lumotlar manbasi bilan ishlash uchun XML Mapper utiliti yordamida transformacion (o'zgartiriladigan) fayllar yaratish kerak. XML Mapper utilitini ishga tushirish uchun Tools —

XML Mapper buyrug'ini bajarish kerak.

4. XML Mapping Tools oynasida File — Open punkti tanlanadi. biolife.xml faylini ochib uni xmldemo papkaga saqlaymiz. Buning uchun (File — Save — XML Document) buyrug'ini bajaramiz.

5. XML Mapper utiliti oynasining chap qismida Document View tanlanganiga ishonch hosil qilish kerak. Unda biolife.xml hujjatining daraxtsimon strukturasi aks etadi. XML hujjatining

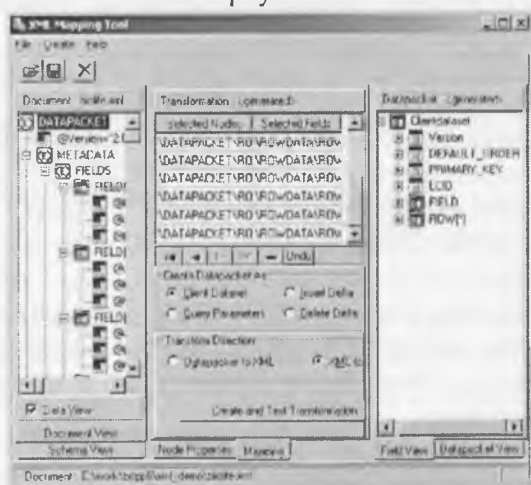
yuqori tuguni (DATAPACKET) ga sichqoncha strelkasini keltirib o'ng tugmasini bosamiz va konteks menyusidan ketma-ket ravishda Select All va Create Datapacket from XML punktlarini tanlaymiz.

6.XML Mapper utilitining chap oynasida Mapping vkladkasi tanlanganiga ishonch hosil qilib Create and Test Transformation knopkasini bosamiz (1-rasm).

7.Transformacion faylni yaratilgan papkaga saqlaymiz. Buning uchun File — Save — Transformation buyrug'ini bajaramiz va transformacion fayl uchun **ToBiolifeDp.xtr** nomni beramiz.

8. Yaratilgan papkaga ma'lumotlar paketi faylini ham yozib qo'yamiz. Bu ham XML fayl bo'lib, mijoz ma'lumotlari bilan provayder komponenti o'rtasidagi ma'lumotlar almashinuvini ta'minlaydi. Saqlash uchun File—Save—Datapacket buyrug'ini bajaramiz va faylning nomi sifatida packet.xml nomini kiritamiz.

9.XML Mapper bilan ishni tugatamiz va Delphi loyihasiga qaytamiz. Formaga quyidagi komponentlarni keltirib qo'yamiz: XMLTransformProvider, ClientDataSet, DataSource (komponentlar palitrasing Data Access sahifasidan) va XML fayl ma'lumotlarini tasvirlash uchun DBGrid komponentini Data Controls sahifasidan keltirib qo'yamiz.



10.6.1-rasm. XML Mapping Tool oynasi

10. Komponentlar xususiyatlarini quyidagicha tartibda o'rnatamiz:

```
XMLTransformProvider1.XMLDataFile := <...>\biolife.xml  
XMLTransformProvider1.TransformationRead.TransformationFile :=  
<...>\ToBiolifeDp.xtr  
ClientDataSet1.ProviderName := XMLTransformProvider1  
ClientDataSet1.FileName := <...>\packet.xml  
DataSource1.DataSet := ClientDataSet1  
DBGrid1.DataSource := DataSource1
```

11. Loyihani yaratilgan papkaga xmldemo nomi bilan saqlaymiz. Keyin ClientDataSet1.Active := true xususiyatini o'rnatamiz. Natijada dbgrid1 komponentasi paket faylining sarlovhasini ko'rsatadi.

12. DBGrid1 jadvali biolife.xml fayli tarkibidagi ma'lumotlarni ko'rsatishi uchun

```
ClientDataSet1.Active := false;  
ClientDataSet1.FileName := <..>\biolife.xml;  
ClientDataSet1.Active := true;
```

xususiyatini o'rnatamiz. Shundan so'ng dasturni bajarishga buyruq bo'ramiz.

10.7. Hisoblanadigan maydonlar

Ma'lumotlar bazasi jadvalida qiymatlari boshqa ustunlar qiymatlari asosida hisoblab hosil qilinadigan maydon yaratish mumkin. Bunday maydonni yaratish quyidagicha amalga oshiriladi.

1. Ishchi diskda dbsdemo nomli papka yaratamiz.

2. DELPHI ni ishga tushirib yangi loyiha yaratamiz. Formani DBSDemo nom bilan nomlaymiz. Loyihani papkaga saqlaymiz.

3. Ma'lumotlar bazasining vizual bo'lmagan komponentlarini joylashtirish uchun ma'lumotlar modulini yaratamiz (File — New — Data Module). Ma'lumotlar moduliga **dataunit** nomini beramiz va saqlaymiz. Modulga TTable va TDataSource komponentlarini keltirib qo'yamiz va ularning xususiyatlarini quyidagicha o'rnatamiz.

```

Table1.Name := OrderTable;
OrderTable.DatabaseName := DBDEMOS;
OrderTable.TableName := orders.db;
OrderTable.Active := True;
DataSource1.Name := OrderDataSource;
OrderDataSource.DataSet := OrderTable;

```

4. Formaga TDBGrid komponentini qo'yamiz va unga quyidagi xususiyatlarni o'rnatamiz:

```

DBGrid1.Name := OrderGrid;
OrderGrid.DataSource := MD.OrderDataSource;
OrderGrid.Options.dgEditing := False

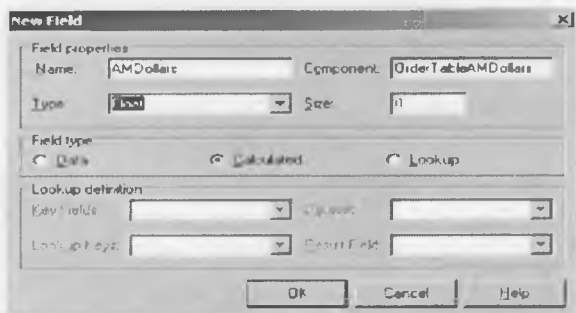
```

5. Iloovani ishga tushirib uning to'g'ri ishlashini tekshirib ko'ramiz.

6. Jadvalga yana bitta maydon kiritamiz. Bu maydon AmountPaid maydonidagi rubl'da berilgan qiymatlarni dollorda ifodalasin. Bunday maydonlar hisoblanadigan maydonlar deyiladi. Buning uchun OrderTable komponenti ustiga strelkani keltirib, chap knopkani ikki marta bosamiz va maydonlar redaktori Field Editor ni chaqiramiz.

7. MD.OrderTable oynasiga strelkani keltirib New Field ni tanlaymiz va chap knopkani bosamiz.

8. New Field oynasidagi Name qatoriga maydonning nomi «AMDollars»ni kiritamiz, Type qatoriga maydon tipi «Float»ni kiritamiz, Field Type elementi uchun Calculated ni tanlaymiz va OK.



10.7.1-rasm. Hisoblanadigan maydonni o'rnatish

9. Yaratilgan maydon OrderGrid da ko'rinishi uchun OrderGrid ni tanlaymiz va Columns Editor buyrug'ini bajaramiz.

Editing OrderGrid.Columns oynasida Add ni tanlaymiz. Ob'ektlar inspektorining FieldName xususiyatiga «AMDollars»ni kiritamiz.

10. Object TreeView utiliti yordamida OrderTable jadvalidan AMDollars maydonini topib OnGetText hodisasiga kod kiritamiz:

```
Text:=FloatToStr(OrderTable.FieldByName('AmountPaid').AsFloat/30);
```

Nazorat savollari

1. Delphida ma'lumotlar bazasi bilan ishlovchi qanday komponentlar mavjud?

2. InterBasi MBBT ga qaysi komponentlar yordamida ulanish mumkin?

3. QReports politrasi komponentlari yordamida hisobot yaratish haqida tushuncha bering.

4. Hisoblanuvchi maydonlar qanday yaratiladi?

5. DELPHIda XML formatida saqlanayotgan ma'lumotlarni jadval ko'rinishida ifodalovchi ilova qanday yaratiladi?

6. Delphida Interbase MB haqida ma'lumot olish qanday amalga oshiriladi?

XI. PHPDA MA'LUMOTLAR BAZASI BILAN ISHLASH

11.1. PHP haqida ma'lumot

PHP tiliga 1995 yil asos solingan. Shu davrda Lerdorf nomli dasturchi Perl/CGI ko'rinishidagi ssenariyni tuzdi. Bu ssenariyni u o'zining sahifasiga kiruvchilar sonini aniqlash maqsadida tuzdi. Shu dastur PHP tilining yaratilishiga turtki bo'ldi.

1997 yildan boshlab u PHP deb atala boshladi. PHP tilini (Huper Text Processor-PHP) texnik nuqtai nazaridan qaraganda server tomonidan turib ishlaydigan ssenariylar tilidir. PHP tili dasturchilarga o'zining qulayligi, ishonchliligi bilan manzur bo'lgan. PHP tilining asosiy ustunligidan biri uni to'g'ridan-to'g'ri html ning ichiga joylashtira olishimizdir.

Quyidagi misolda PHP bilan html teglari aralashib keladi:

```
<html>
<title><?php print "Bu PHP"; ?>/title>
<body>
  <?php
    print "<h3> Yana PHP </h3>";
  ?>
  <h4>Bu HTML</h4>
  <?php
    print "<h1>Yana PHP" </h1>;
    print "<h6>2-satr</h6>";
  ?>
</body>
</html>
```

PHP tilidagi dasturlar ishlashi uchun kompyuterga Apache nomli web server o'rnatilishi kerak. Odatda Apache, PHP, MySQL uchalasi birgalikda ishlatiladi. PHP ga o'tish jarayoni **escaping to PHP** deb yuritiladi.

PHP ga o'tishning 4 ta varianti bor:

1. Standart teglar;
2. Qiska teglar;
3. Script tegi;
4. ASP stilidagi teglar.

Ko'pchilik hollarda dasturchilar standart tegdan foydalanadi. U quyidagicha:

```
<?php  
    print "Bu PHP";  
?>
```

Ko'rinib turganidek PHP ning boshlanishi **<?php** bilan ochiladi, **?>** bilan yopiladi.

Qisqa teglar quyidagicha yoziladi:

```
<?  
    print "Bu PHP";  
?>
```

Lekin standart teglardan foydalangan maqsadga muvofiq. Chunki serverda qisqa teglar ajratilgan bo'lishi mumkin. Shuning uchun turli xil xatoliklar bo'ladi.

Agarda html redaktorlar php kodni xato deb qabul qilsa, u holda quyidagi script tegi yordamida php kod qo'shiladi. Bu holat kamdan-kam hollarda yuz beradi.

```
<script language="php">  
    <?php  
        print "Bu PHP";  
    ?>  
</script>
```

11.2. PHPda o'zgaruvchilar

PHP da o'zgaruvchilar \$ (dollar) belgisi bilan boshlanadi. Bu belgidan keyin lotin alifbosining harflari, keyin arab raqamlari va **"_"** belgilar ketma-ketligi bo'lishi mumkin. Lekin \$ belgisidan keyin birdaniga raqam bilan boshlash mumkin emas. Masalan, \$b, \$a_3, \$abbos. Xato o'zgaruvchilar: \$20, \$, A, \$92.

PHP da o'zgaruvchilarning nomlari katta, kichikligi farqlanadi. Masalan, \$A#\$a - bular alohida o'zgaruvchilar.

O'zgaruvchiga qiymat berish uchun teng "=" belgisi ishlatiladi.

```
<?php
    $a=10;  $s='maktab';
    echo $a;
?>
```

PHP da o'zgaruvchilar qabul qilgan qiymatlarining tipi bo'yicha aniqlanadi. PHP kuchli tiplanmas dasturlash tillari sinfiga kiradi. Masalan, Perl? Basiclar kabi.

Masalan, quyidagi dasturni yozsak;

```
<?php
    echo '100'+1;
?>
```

Dastur ishlaganda natija 101 bo'ladi. PHP translyatori satrli kattalikni avtomatik ravishda songa aylantirib son bilan qo'shadi.

```
<?php
    $a='maktab';
    $b='keldim';
    $A="$a,$b";// maktab,keldim
    echo $A;
?>
```

Bu script bajarilgandan keyin **maktab, keldim** hosil bo'ladi. Qo'shtirnoq ichida maxsus boshqaruvchi belgilar ham berilishi mumkin.

```
\r-enter;
\n-yangi satr;
\"-\"-chiziqcha belgisi;
\-$-dollar belgisi;
\\-\ belgisi.
```

Yuqoridagi misolni boshqacharoq ko'rinishda yozamiz:

```
<?php
```

```

$a='maktab';
$b='keldim';
$A=$a.''. $b.'!';// maktab,keldim!
echo $A;
?>

```

Oldingi dasturning natijasi bilan bir xil. Faqat satrlarni qo'shish uchun maxsus "." belgisi ishlatilgan.

Agarda bir necha o'zgaruvchilarga bir xil qiymat berish kerak bo'lsa, quyidagicha yozish mumkin:

```

$a=$c=10.35;

```

11.3. Shartli operatorlar

Dasturlash tillaridagi kabi PHP da ham shartli jarayonlarni dasturlash uchun shartli operatorlar mavjud. Buni quyidagi misolda ko'ramiz:

```

<?php
$a=25; $b=5*5;
if ($a==$b)
    echo '$a va $b teng';
else
    echo '$a va $b teng emas';
?>

```

Bu yerda if else ko'rinishidagi shartli operator ishlatilgan. Uning umumiy ko'rinishi quyidagicha:

```

if (shart)
    ifoda1;
else
    ifoda2;

```

if dan keyin tenglik tekshiriladi. Bunda "==" teng ma'nosida, "!=" teng emasligini bildiradi.

Shart tekshirish operatori quyidagi ko'rinishda ham ishlatilishi mumkin:

```

<?php
$a=25; $b=5*5;
if ($a==$b)

```

echo '\$a va \$b teng';

?>

Agar shartning aks holda qismi qatnashmasa yuqoridagicha ishlatiladi.

Biror shart bajarilganda bitta ifoda emas, bir nechta operator bajariladigan bo'lsa, {} qavslardan foydalaniladi

11.4. Ma'lumotlarni uzatishda GET va POST metodlarni ishlatilishi

GET metodi orqali uzatish. Buning uchun forma, gipermurojaat yoki to'g'ridan-to'g'ri URL dan foydalanish mumkin. Masalan, \$a=77 qiymatni uzatish uchun URL ga quyidagicha yozish mumkin:

[HTTP://talaba/filetest1.php?a=77](http://talaba/filetest1.php?a=77)

Agar ikkita o'zgaruvchilarga \$a=77 va \$b=10 qiymatlarni uzatmoqchi bo'lsak URL ni quyidagicha yozamiz:

[HTTP://talaba/filetest1.php?a=77&b=10](http://talaba/filetest1.php?a=77&b=10)

GET metodi orqali uzatilayotgan o'zgaruvchilar maxsus global massivda joylashtiriladi. Bu massiv nomi \$_GET

	a	b
\$_GET =	77	10

O'zgaruvchilarni qabul qilib olish quyidagicha amalga oshiriladi:

`$a=$_GET['a'], $b=$_GET['b']`

Gipermurojaat orqali uzatish uchun faylga murojaatni quyidagicha yozamiz:

echo " Yangi qiymat "

POST metodi orqali uzatish. Bu metod orqali uzatish uchun forma yaratamiz va unda o'zgaruvchilar uzatilayotgan faylning nomini ko'rsatamiz.

```
<form action= "filetest3.php" method= "post" name="form1"
target="_blank">
  <p>Ismingizni kiriting: <br> <input name="name"
type="text" size="20"></p>
  <p>Familiyani kiriting: <br> <input name="lastname"
type="text" size="20"></p>
  <p> <input name= "submit" type= "submit" value= "Uzatish"
> </p>
</form1>
```

Novbatnagi bosqichda yangi filetest3.php faylini ochib, unga quyidagi kodlarni kiritamiz:

```
$name=$_POST['name'];
$lastname=$_POST['lastname'];
echo "Oldingi sahifada kiritilgan ism: $name va familiya
$lastname"
```

Dasturni ishga tushirib tekshirib ko'ramiz.

Agar birorta saytga murojaat qilmoqchi bo'lsak, shu saytdagi sahifani ko'rish uchun so'rov yuboramiz. Bunda bizning brouzer biz murojaat qilgan saytning serveriga turli xil ma'lumotlarni uzatadi. Bu ma'lumotlar \$_server nomli super massivda saqlanadi.

	HTTP_REFERER	HTTP_USER_AGENT	REMOTE_ADDR
	http://talaba	Opera/9.23Windows10	127.0.0.1

\$_SERVER =

Saytga kiruvchi mehmonlarning IP adresi va boshqa ma'lumotlarini bilmoqchi bo'lsak, quyidagicha dastur kodini yozamiz:

Echo "Sizning IP adresingiz:

```
".$_SERVER['REMOTE_ADDR'];
```

Echo "Sizning brauzer va OS:

".\$_SERVER['HTTP_USER_AGENT'];

Echo "Sizning adresingiz:".\$_SERVER['HTTP_REFERER'];

include instruksiyasi

summa.php fayli

<?php

include "settings.php";

\$c = \$a+\$b;

echo "O'zgaruvchilar yig'indisi - \$c";

?>

settings.php fayli

<?

\$a = 10;

\$b=15;

?>

14.4.1-rasm.Include instruksiyasining ishlatilishi

include instruksiyasi dasturni alohida modullarga bo'lib, ularni birgalikda ishlatishga imkon beradi. Dastur faylini boshqa boshqa bir dastur fayliga bog'lash uchun quyidagicha amalga oshiriladi:

Include "settings.php";

11.5.PHP va MySQL bog'liqligi

PHP tili turli ma'lumotlar bazasi bilan bog'lana oladi, bu esa uning yutuqlaridan biri hisoblanadi. Bu bo'limda MySQL ma'lumotlar bazasi haqida so'z boradi. MySQL keng tarqalgan bepul ma'lumotlar bazasi bo'lib, uni PHP tili yordamida ishlatish mumkin. MySQL ma'lumotlar bazasini boshqarish tizimida ishlash uchun SQL tili buyruqlaridan foydalaniladi. Har bir ma'lumotlar bazasi o'zining dasturiga ega bo'lgani kabi MySQL ham dastur interfeysiga ega. MySQL dasturi o'rnatilgandan so'ng mysql.exe fayli yordamida konsol oynali dastur ishga tushadi.

Bazadan ma'lumotlarni o'qib olish umumiy protsedurasini keltiraiz.

```
<?php  
MBBTga boglanish  
SQL buyruqlari  
MBBT tizimidan chiqish  
?>
```

Ushbu dasturiy ta'minotni olish va u haqda ma'lumotlarga ega bo'lish uchun <http://www.mysql.org> yoki www.mysql.com saytiga murojaat qilish mumkin. MySQL bilan ishlashdan oldin ro'yxatdan o'tish kerak.

```
Mysql [-h tugun] [-u foydalanuvchi_nomi]  
[ma'lumotlar_bazasi_nomi]
```

Kvadrat qavslar ularning qo'yilishi muhim emasligini bildiradi. Tugun – bu MySQL ishlab turgan kompyuter nomi. Agar bu parametr ko'rsatilmagan bo'lsa, u holda MySQL lokal kompyuterda ishlab turibdi deb hisoblanadi. Agar foydalanuvchi nomi ko'rsatilmagan bo'lsa, u holda kompyuter yuklanganda ro'yxatdan o'tgan foydalanuvchi ishlatayotganligi tushuniladi. Agar ma'lumotlar bazasi nomi ko'rsatilmasa, MySQL mavjud bazasi ishlab to'rganligini bildiradi. [p] parametri faqatgina MySQLda ishlash uchun parol so'ragan taqdirdagina ko'rsatiladi. Agar mavjud bo'lgan ma'lumotlar bazasiga bog'lanish uchun USE buyrug'idan foydalaniladi. Masalan cars bazasiga bog'lanish uchun "USE cars" buyrug'i ishlatiladi. Shundan so'ng ekranda baza tanlanganligi haqidagi xabar chiqadi: Database changed.

Agar ro'yxatdan o'tish paytida baza MySQL serverida baza tanlanmagan bo'lsa va USE buyrug'idan oldin boshqa buyruqlar ishlatilgan bo'lsa, u holda quyidagi xatolik ekranda paydo bo'ladi.

```
ERROR 1046:p No Database Selected
```

Ma'lumotlar bazasini bilan ishlashdan oldin ma'lumotlar bazasini o'zini yaratish va undan so'ng jadvallar yaratish

mumkin. Ma'lumotlar bazasini yaratish uchun SQL buyrug'i CREATE DATABASE buyrug'idan foydalanish mumkin.

```
CREATE DATABASE cars;
```

MySQL bu so'rovni qayta ishlagandan so'ng quyidagi xabarni chiqaradi.

```
Query Ok, 1 row affected (0.07 sec)
```

Jadval yaratish uchun SQL tilining CREATE TABLE buyrugidan foydalaniladi. Masalan quyidagicha

```
CREATE TABLE Equipment
```

```
(Equip_ID INT UNSIGNED NOT NULL AUTO  
INCREMENT PRIMARY KEY, Equip INT UNSIGNED);
```

PHP yordamida MySQL ma'lumotlar bazasiga boglanish.

PHP da mysql_connect funkstiyasi ssenariyni MySQL bilan bog'laydi.

```
mysql_connect ("localhost" , "username" , "password");
```

Bu funkstiya uchta parametrga ega. Bu parametrlar yuqorida ko'rib o'tildi. Bu parametrlarni jimlik qoidasi buyicha ishlatish uchun funktsiya argumentlari ko'rsatilmaydi.

```
$db=mysql_connect();
```

Aniq bazaning o'ziga bog'lanish uchun mysql_select_db() funksiyasidan foydalaniladi.

```
mysql_select_db("cars");
```

MySQL serveriga so'rovlar mysql_query funkstiyasi yordamida beriladi. So'rov avval biror bir satr o'zgaruvchisiga o'zlashtirilgan bo'lishi kerak.

```
$query="select * from mashina"
```

```
$result=mysql_query($query);
```

Ko'pchilik hollarda natijalovchi ma'lumotlarda sartlar sonini bilish kerak bo'ladi. Buning uchun mysql_num_rows funkstiyasidan foydalaniladi.

```
$num_rows=mysql_num_rows($result);
```

Ma'lumotlar to'plamidan alohida satrlarni turli usullar yordamida ajratib olish mumkin. Masalan mysql_fetch_array

funksiyasi yordamida, keyingi satrning ma'lumotlarini olish mumkin. Agar yozuv boshqa mavjud bo'lmasa, false qiymatni beradi.

11.6. PHP yordamida ma'lumotlar bazasini boshqarish

PhpMyAdmin yordamida MB bilan ishlash uchun dastlab WEB serverni ishga tushiriladi. PhpMyAdmin oynasini ochiladi. Login va parolni kiritiladi. Yangi MB (Yordamchi) ni yaratiladi. Unda tasks (masalalar) nomli jadval yaratiladi. Uning strukturasi quyidagicha ko'rinish oladi.

#	Имя	Тип	Сравнение	Атрибуты	Null	По умолчанию	Дополнительно	Действие
1	id_task	int(11)			Нет	Нет	AUTO_INCREMENT	
2	text	varchar(255)	utf8_general_ci		Нет	Нет		
3	status	int(1)			Нет	0		

11.6.1-rasm. MySQL oynasining ko'rinishi

My SQL imkoniyatlaridan foydalanib bir nechta yozuv kiritamiz.

	id_task	text	status
☐	1	Машинани ювиш	0
☐	2	Атрофии тозалаш	0
☐	3	Видеодарс езиш	1
☐	4	Дастур тузиш	0
☐	5	111	1
☐	6	Масалани оптималлаштириш	0

11.6.2-rasm. MySQL oynasida ma'lumotlarning ko'rinishi

PHP skript yordamida MBga ulanish uchun Adobe Dreamweaver yordamida quyidagi dasturni tuzamiz.

```
<?php
```

```

// MBga ulanish
mysql_connect('localhost' , 'root', "");
mysql_select_db('Yordamchi');
// POST ni qabul qilish
if (isset ($_POST['task']))
{
$task=trim($_POST['task']);
    if ($task != "")
    {
        $task=mysql_real_escape_string($task);
        $sql= "insert into tasks (text) values ('$task')";
        mysql_query($sql);
    }
    header('location: index.php');
    exit();
}
else if (isset ($_POST['id_task']))
{
$id_task=trim($_POST['id_task']);
$id_task=mysql_real_escape_string($id_task);
$sql= "update tasks set status='1' where id_task='$id_task'";
mysql_query($sql);
// echo $sql;
header('location: index.php');
exit();
}
?>
<!DOCTYPE HTML>

```

```

<html>
<head>
<title>PHP da MB bilan ishlash </title>
<meta http-equiv="Content-Type" content="text/html;
charset=UTF-8" />
</head>
<body>
<h1> TODO </h1>
<form method="post">
    <input type="text" name=task placeholder="Nima qilish
kerak"/>
<input name="" type="submit" value="Masala kiritish">
    </form>
<hr/>
<?php
$result=mysql_query('select * from tasks order by status');
while ($row= mysql_fetch_assoc($result))
{
    if ($row['status']==0)
    { echo '<form method="post">';
      echo $row['text'];
      echo ' ';
      echo '<input name="id_task" type="hidden"
value="'. $row['id_task']. "' >';
      echo '<input type="submit" value="Bajarildi">';
      echo '</form>';          }
    else      {
      echo '<s>'. $row['text']. '</s>';
      echo '<br/><br/>';          } }

```

?>

</body>

</html>

Dasturdan ko'rinib turibdiki, MBdan qatorlarni o'qish, MBga qator kiritish va o'zgartirish uchun SQL so'rovlardan, formalardan va PHP operatorlaridan foydalaniladi. Dasturni bajarishga yuboramiz va turli xil qiymatlar kiritib uning ishlashini tekshirib ko'ramiz.

http://127.0.0.1/

TODO

Masala kiritish

Matnning uzunligi

Atrofingizni tozalang

Dastur tugadi

Masalani optimallashtiring

Biladim

11..6.3-rasm. Brouzer oynasida dastur natijasining ko'rinishi

Nazorat savollari

1. PHP da ozgaruvchilar qanday yoziladi?
2. Qanday web serverlarni bilasiz?
3. MYSQL da jadval qanday yaratiladi?
4. HTML da forma komponentalaridan qaysilarini bilasiz?
5. Form tegining ishlashini izohlab bering.
6. SQL so'rovlar PHP da qaysi operator yordamida yoziladi?
7. PHP da ECHO buyrug'ining vazifasi
8. PHP da if operatorlariga misol keltiring.
9. PHP da MBga bog'lanish operatorlarini izohlang.

XII. MA'LUMOTLAR BAZASI ADMINISTRATSIYASI

12.1. MBni himoyalash bosqichlari

Ma'lumotlarni himoyalashni ikki bosqichda tashkil qilish mumkin: ilova bosqichida va ma'lumotlar bazasi bosqichida.

Ilova bosqichida himoyalash deganda foydalanuvchi interfeysining qaysidir elementlariga kirishni cheklash tushuniladi (masalan, knopkalarga yoki menyu buyruqlariga). Bu bilan ma'lumotlar bazasiga boshqa ilovalardan kirish foydalanuvchi nomi va parolidan tashqari hech narsa bilan cheklanmagan.

Ma'lumotlar bazasi bosqichida himoyalash deganda tizimga kiritilgan himoya vositalari yordamida ma'lumotlarga kirish va uni qayta ishlash uchun cheklov qo'yish tushuniladi. Bunda foydalanuvchilarga berilgan vakolatlar haqidagi ma'lumot to'g'ridan-to'g'ri serverning o'zida saqlanadi va bazaga ulangan barcha ilovalar uchun umumiy hisoblanadi.

Ma'lumotlarni ishonchli himoyalashni ta'minlash uchun ma'lumotlar bazasi bosqichida himoyalashni qo'llash tavsiya etiladi. Lekin amalda himoyani tashkil etishning aralash usuli ko'p ishlatiladi.

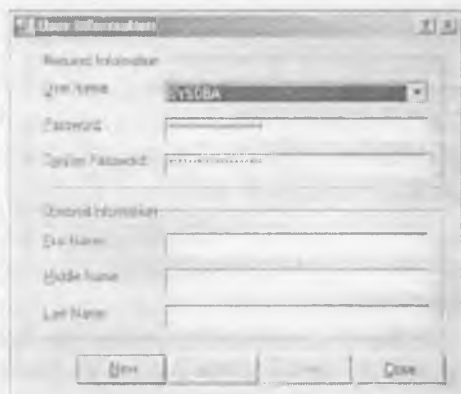
InterBase MBBTda ma'lumotlar himoyasini tashkil qilishda markaziy rolni maxsuslashtirilgan isc4.gdb nomli MB bajaradi. U MBBT o'rnatilgan katalogda joylashgan. Unda ma'lumotlarga kirish va InterBase server xizmatidan foydalanish huquqiga ega bo'lgan barcha foydalanuvchilar haqidagi ma'lumotlar saqlanadi. Undagi yozuvlar hisob yozuvlari (account) deb ataladi. Har bir shunday yozuv foydalanuvchi nomi va uning parolidan iborat. Foydalanuvchi nomi 31 simvoldan (harflar registri ahamiyatga ega emas), parol sakkiz simvoldan (registr hisobga olinadi) iborat bo'lishi mumkin.

Ma'lumotlar bazasi ustida har qanday masalani echish yoki ma'murlash uchun eng avval SQL-serverga ulanish kerak. Buning uchun har bir foydalanuvchi o'z nomini va parolini kiritishi kerak. Kiritilgan nom va parolning to'g'riligi isc4.gdb ma'lumotlar bazasida tekshiriladi. Agar mos hisob yozuvi topilsa, MBga ulanish bajariladi.

12.2. IBConsole dasturida foydalanuvchilar bilan ishlash

InterBasening ixtiyoriy serverida SYSDBA foydalanuvchi ro'yxatga olingan. Uning jimlikdagi paroli **masterkey**. Bu foydalanuvchiga maxsus hisob yozuvi mos keladi va u **SQL**-serverning himoya tizimini e'tiborga olmaydi hamda ma'murlashga doir ixtiyoriy topshiriqni bajara oladi. InterBase o'rnatilgandan so'ng faqatgina SYSDBA foydalanuvchi mavjud bo'ladi va u boshqa foydalanuvchilarni ro'yxatga olishi kerak. `isc4.gdb` MBga ma'lumot kiritish, o'chirish va ma'lumotlarni o'zgartirishni faqat SYSDBA qila oladi. Buning uchun **IBConsole** dasturi ishlatiladi.

Maslahat. SYSDBA foydalanuvchining parolini birinchi imkoniyatdayoq **IBConsole** dasturi yordamida o'zgartirish qat'iy talab qilinadi. InterBase foydalanuvchilarini **IBConsole** dasturi yordamida ma'murlash **User Information** (rasm 12.1) dialog oynasi orqali amalga oshiriladi. Bu oynaga kirish faqat InterBase serveriga ulangandan keyingina mumkin bo'ladi.



12.2.1-rasm.. "User Information" dialog oynasi

Ushbu oynani ochish uchun quyidagi ikki usuldan biridan foydalanish mumkin.

1. **IBConsole** dasturi yuqori menyusidan **Server** elementini tanlash va undagi **User Security** buyrug'ini ishga tushirish;

2. **IBConsole** dasturi daraxtsimon oynasining chap tomonidan registrastiya qilingan server ustiga sichqoncha strelkasini keltirib, o'ng knopkasi bosiladi va konteks menyusidan **User Security** tanlanadi.

Yangi foydalanuvchini registrastiya qilish. Serverda yangi foydalanuvchini registrastiya qilish uchun **User Information** oynasi ochiladi va unda quyidagi ishlar bajariladi

1. Sichqoncha knopkasini **New** (yangi) ustiga bosing;
2. **User Name** maydoniga foydalanuvchi nomini kiriting
3. Foydalanuvchi parolini **Password** va **Confirm Password** maydonlariga kiriting;
4. Hohlasangiz familiya, ism va otchestvodan iborat qo'shimcha informastiyani **Optional Information** bo'limiga kiritishingiz mumkin.

5. isc4 . gdb MBga yangi foydalanuvchi haqidagi ma'lumotni kiritish uchun sichqoncha yordamida **Apply** knopkasini bosing.

Ogohlantirish. InterBase da foydalanuvchi nomi va parolida probel ishlatishga ruxsat berilmaydi.

Foydalanuvchi haqidagi ma'lumotlarni o'zgartirish. Foydalanuvchi haqidagi informastiyani o'zgartirish uchun **User Information** (12.2.1- rasm) oynasida quyidagi harakatlar bajariladi.

1. Dialog oynasining yuqori tomonida ochilgan ro'yxatdan foydalanuvchini tanlang.
2. **User Name** dan boshqa ixtiyoriy maydonni qiymatlarini o'zgartiring. Agar **Password** maydonidagi parol o'zgartirilsa, uni ikkinchi marta **Confirm Password** maydonida o'zgartiring.
3. O'zgartirishlarni saqlash uchun **Apply** tugmasiga sichqoncha yordamida bosing

Izoh. Foydalanuvchi nomini o'zgartirish mumkin emas. Uning birdan bir yo'li foydalanuvchni o'chirib tashlab, keyin esa boshqa nom bilan qayta yaratish.

Foydalanuvchini o'chirish. Foydalanuvchi haqidagi informastiyani o'chirish uchun **User Information** oynasida quyidagi harakatni bajarish kerak.

1. **User Name** ochiluvchi ro'yxatdan foydalanuvchini tanlang

2. **Delete** knopkasiga sichqoncha yordamida bosing. Natijada ekranda amalni tasdiqlash haqida so'rov paydo bo'ladi. Agar bu so'rovga javob sifatida **OK** ni bosilsa, uholda joriy foydalanuvchi o'chadi. Bu foydalanuvchi serverning ma'lumotlar bazalariga boshqa ulana olmaydi.

Ogohlantirish. SYSDBA o'zini o'zi o'chirishi mumkin bo'lsada, bunday qilishga qat'iy ravishda tavsiya qilinmaydi. Unday holda boshqa foydalanuvchilar haqidagi ma'lumotlarni o'zgartirib bo'lmay qoladi. SYSDBA foydalanuvchisini o'chirgandan so'ng isc4 . gdb MBni qayta tiklash uchun InterBase qayta o'rnatishga to'g'ri keladi.

Kirish huquqi. Yangi foydalanuvchi avtomatik tarzda hech qanday huquqga ega bo'lmaydi. Foydalanuvchi (SYSDBA dan tashqari) unga mos huquqlar berilmaguncha birorta ham ob'ektga kirish huquqiga ega bo'lmaydi. InterBase serveriga kirish uchun vakolatlar rollar (role) orqali va GRANT nomli SQL buyrug'i yordamida belgilanadi.

Rollar. Rol — bu vakolatlarning nomlangan nabori bo'lib, keyinchalik uni MBning ixtiyoriy foydalanuvchisiga belgilash mumkin. Rolni yaratish uchun

"CREATE ROLE rol nomi" buyrug'i ishlatiladi.

Rolni o'chirish uchun **"DROP ROLE rol nomi"** buyrug'i ishlatiladi.

MBga ulanishda rol ko'rsatilishi kerak.

Quyida CONNECT buyrug'ida rolni ishlatishga misol keltirilgan.

CONNECT 'c:\staff\base\staff.gdb'

USER 'user1' PASSWORD 'pssw1' ROLE 'role1'

12.3. GRANT va REVOKE buyruqlari

GRANT buyrug'i foydalanuvchilarga, rollarga va boshqa ob'ektlarga kirish huquqini va rolni belgilash uchun ishlatiladi. Bu buyruqning sintaksisini "Xodimlar" dasturi misolida ko'ramiz.

Faraz qilaylik, staff.gdb MBga quyidagi vakolatlarga ega bo'lgan uch xil kategoriyadagi foydalanuvchilar ulanishi kerak.

1. Kiritish, o'chirish va barcha ma'lumotlarni taxrirlash huquqiga ega bo'lgan rol yaratamiz va uni FullR deb belgilaymiz.

CREATE ROLE FullR

2. FAMILY jadvali uchun to'liq vakolat, STAFF jadvali bilan ishlashda faqat manzil haqidagi ma'lumotlarni taxrirlash (zip, street, House, Tel ustunlari), boshqa jadvallar uchun faqat ko'rishga ruxsat beruvchi chiefR roli. CREATE ROLE ChiefR

3. Barcha ma'lumotlarni faqat ko'rish mumkin. Bu kategoriya uchun ReadR rolini yaratamiz CREATE ROLE ReadR

Birinci kategoriyaga kadrlar bo'limi xodimlari kirishi mumkin, ikkinchi kategoriyaga bo'lim boshliqlari, uchinchi kategoriyaga qolgan barcha ishlovchi xodimlar kirsin (12.3.1-rasm).



12.3.1-rasm. IBConsole oynasi

Bu rollarning har biri uchun vakolatlar belgilaymiz.

4. Dastlab FullR roli uchun barcha jadvallar va FullList ko'rinish uchun to'liq vakolatlar beramiz:

```
GRANT ALL ON DEPS TO FullR;  
GRANT ALL ON FAMILY TO FullR;  
GRANT ALL ON JOBS TO FullR;  
GRANT ALL ON POSS TO FullR;  
GRANT ALL ON REGIONS TO FullR;  
GRANT ALL ON STAFF TO FullR;  
GRANT ALL ON FullList TO FullR
```

5. Keyin ChiefR roli uchun FAMILY jadvaliga to'liq vakolat, STAFF jadvaliga chegaralangan vakolat va boshqa jadvallarga faqat ko'rish uchun vakolatlar beramiz:

```
GRANT SELECT ON DEPS TO ChiefR;  
GRANT ALL ON FAMILY TO ChiefR;  
GRANT SELECT ON JOBS TO ChiefR;  
GRANT SELECT ON POSS TO ChiefR;  
GRANT SELECT ON REGIONS TO ChiefR;  
GRANT SELECT, UPDATE (Zip,Street,House,Tel) ON  
STAFF TO ChiefR;  
GRANT SELECT ON FullList TO ChiefR
```

Bu yuqoridagi holda FAMILY jadvali uchun to'liq vakolat ALL belgilandi. Bu GRANT buyrug'ini kengaygan formada quyidagicha ham yozsa bo'ladi:

```
GRANT SELECT, DELETE, INSERT, UPDATE ON FAMILY  
TO chief
```

6. Barcha jadvallarni faqat o'qish uchun vakolatni ReadR roliga beramiz:

```
GRANT SELECT ON DEPS TO ReadR;  
GRANT SELECT ON FAMILY TO ReadR;  
GRANT SELECT ON JOBS TO ReadR;  
GRANT SELECT ON POSS TO ReadR;  
GRANT SELECT ON REGIONS TO ReadR;
```

GRANT SELECT ON STAFF TO ReadR;
GRANT SELECT ON FullList TO ReadR .

7. Faraz qilaylik IBConsole dasturi yordamida quyidagi foydalanuvchilar qayd qilingan bo'lsin: BOSS - tashkilot rahbari, PD1 va PD2 — kadrlar bo'limi xodimlari, CHIEF1 va CHIEF2 – bo'lim boshliqlari, USER1 va USER2- ishchi xodimlar

8. U holda bu foydalanuvchilarga mos rollarni belgilash uchun quyidagi buyruqlar ishlatiladi:

GRANT FullR TO PD1, PD2;

GRANT ChiefR TO CHIEF1, CHIEF2;

GRANT ReadR TO BOSS, USER1, USER2

9. Saqlanuvchi prosedurasi DeleteEmp ga FullR rolini belgilash mumkin

GRANT EXECUTE ON PROCEDURE DeleteEmp TO FullR

REVOKE buyrug'i. Rollar, foydalanuvchilar yoki MB ob'ektiga berilgan vakolatlarni bekor qilish uchun REVOKE buyrug'i ishlatiladi. Masalan ChiefR rolidan STAFF jadvalini taxrirlash huquqini olish uchun quyidagi buyruqni bajarish kerak:

REVOKE UPDATE (Zip,Street,House,Tel) ON STAFF FROM ChiefR

Nazorat savollari

1. MBni himoyalashning qanday bosqichlari bor?
2. IBConsole dasturida foydalanuvchilar bilan ishlash qanday amalga oshiriladi?
3. Yangi foydalanuvchini registratsiya qilishni tushintiring.
4. Foydalanuvchi haqidagi ma'lumotlar qanday o'zgartiriladi?
5. Foydalanuvchini o'chirish qanday amalga oshiriladi?
6. Kirish huquqini berishni tushintiring.
7. Rollar qanday beriladi?
8. GRANT va REVOKE buyruqlarini tushintiring.

XIII. ORACLE MA'LUMOTLAR BAZASINI BOSHQARISH TIZIMI

13.1. ORACLE ma'lumotlar bazasi haqida qisqacha tarixiy ma'lumot

ORACLE korporatsiyasi ma'lumotlar bazasini boshqarish dasturiy ta'minotini yetkazib berish bo'yicha yetakchi kompaniyalardan biri hisoblanadi.

1977 yil Larri Ellison, Bob Mayner va Ed Outslar Software Development Laboratories (SDL) kompaniyasiga asos soldilar.

1979 yil SDL nomini Relational Software, Inc. (RSI) ga o'zgartirdi va Oracle v2 nomli MBBTni ishlab chiqardi. Bu versiya tranzaksiyani qo'llamas edi, lekin SQLning asosiy funksiyalarini amalga oshirar edi. Bu birinchi relyatsion ma'lumotlar bazasini boshqaruvchi tijorat tizimi edi. RSI bozor munosabatlarini hisobga olib birinchi versiyani version 2 deb atadi. Birinchi versiya assemblerda yozilgan bo'lib RSX-11 operacion tizimida ishlar edi.

1982 yilda RSI nomini yana o'zgartirdi va Oracle Systems deb atadi.

1983 yilda "C" tilida yozilgan yangi 3-versiya ishlab chiqildi. Bu versiya tranzaksiyani qo'llab quvvatlovchi funksiyalar bilan to'ldirilgan edi. Bu versiya meynfreym kompyuterlari bilan bir qatorda minikompyuterlar va shaxsiy kompyuterlarda ham ishlay olardi.

1984 yilda amallarni parallel bajara olish vositalarga ega bo'lgan 4- versiya ishlab chiqildi.

1985 yilda mijoz-server arxitekturasida ishlay oladigan 5-versiya ishlab chiqildi. Shu bilan bir qatorda taqsimlangan so'rovlarni qo'llash imkoniyati paydo bo'ldi.

1986 yilda 5.1 versiya ishlab chiqildi.

1988 yilda 6 versiya ishlab chiqildi. Bunda bloklash va

qaynoq zahiralash imkoniyatlari kiritildi.

1989 yilda 6.2 versiya ishlab chiqildi va unga tranzaksiyalarni operativ qayta ishlash (OLTP) ni qo'llash vositalari kiritildi.

1992 yilda 7 versiya ishlab chiqildi. Unda havola butunligini saqlash, saqlanuvchi proceduralar va triggerlarni qo'llash imkoniyati kiritildi.

1997 yilda 8 versiya ishlab chiqildi. Unda oldingi versiyalarga nisbatan yuqori ishonchlilik, katta hajmdagi ma'lumotlarni va ko'p sonli foydalanuvchilarni qo'llash imkoniyati yaratildi. Bu versiyadan boshlab ORACLE ob'ektno-relyatsion MBBTga aylandi.

1998 yilda 8i Release 1 (8.1.5) versiya ishlab chiqildi. Undagi «i» yozuvi Internetda ishlash imkoniyatini qo'llashni ifodalaydi. Bu versiyadan boshlab MBBTga Java virtual mashinasi (JVM) kiritildi. Javada administratsiya ishlarini bajarish va dasturlash imkoniyati paydo bo'ldi.

2001 yilda 9i Release 1 (9.0.1) versiya ishlab chiqildi. Bu versiyada XML hujjatlarni to'g'ridan to'g'ri qayta ishlash vositalari ishga tushirildi. Shuningdek replikatsiya, Oracle RAC texnologiyasi, OLAP, Data Mining va yana boshqa juda ko'plab funksiyalarni qo'llash imkoniyati kiritildi.

2004 yilda 10g Release 1 (10.1.0) versiyasi ishlab chiqildi. Uning nomidagi «g» harfi «grid» («tarmoq») ni bildiradi va bu bilan grid-hisoblash ko'zda tutiladi.

2007 yilda 11g Release 1 (11.1.0.6) versiyasi ishlab chiqildi. Unga Apache, IIS veb serverlar, PHP, Perl va boshqa ilovalarni qo'llash imkoniyati kiritildi.

2013 yilda 12c (12.1.0.1) versiyasi yaratildi. Uning nomidagi «c» harfi ingliz tilidagi **cloud** (bulut) so'zidan olingan bo'lib bulutli texnologiyalar bilan ishlashga moslashgan.

2018 yilda 18c versiyasi yaratildi.

2019 yilda 19c versiyasi yaratildi.

13.2. ORACLE MBBTda ishlatiladigan asosiy tushunchalar

Oracle kompaniyasi o'z yo'nalishini RMBBT yaratishga qaratdi. Uning mahsulotlari dunyoning barcha etakchi yirik muassasalarda qo'llanilmoqda. Bundan tashqari uzoq yillar mobaynida kompaniya Oracle RMBBT bilan bog'liq holda ishlovchi dasturiy mahsulotlarni yaratish uchun turli xil vositalarni ham taklif etmoqda. Ular orasida Disaigner, Case – tizimi va Devoloper juda mashhurdir.

Oracle kompaniyasi ham boshqa MBBT yaratuvchilari singari o'z mahsulotlari imkoniyatlarini kengaytirib va takomillashtirib bormoqda. Oracle 8 mahsulotidan boshlab dasturiy mahsulotlariga ob'ektga yo'naltirilgan dasturlash vositalari – sinflar (tiplar), metodlar, kolleksiya, taqdimot va identifikator singarilar kiritildi. Web texnologiyalarining barcha imkoniyatlari dasturiy mahsulotlarga kiritib borilmoqda, ayniqsa ular Designer, Devoloper, SQL Devoloper va boshqa dasturiy mahsulotlarga keng tadbiq qilingan. Hozirgi kunda Oracle kompaniyasi MBBT bozorida yetakchilik qilmoqda.

Oracle, bir biri bilan bog'langan hamda har biri qisman mustaqil bo'lgan, bir necha yirik dasturiy majmualar birlashmasidan tashkil topgan. Har bir dasturiy majmua qaysidar sohaga tegishli ishlarni bajarish uchun o'z xizmatlarini tavsiya qiladi. Oracle qiymatlar lug'atini qo'llab-quvvatlash, ko'p darajali himoyalaniish, qiymatlar butunligini kafolatlash, parallel kirishlarni tashkil qilish va tillar aloqasini ta'minlashning kuchli vositalariga ega. SYSTEM jadval fazosida saqlanuvchi qiymatlar lug'atiga SYS va SYSTEM tavsifi bo'yicha xizmat ko'rsatadi. Oracle foydalanuvchilarni aniqlashni OT vositalari yordamida, qayd qilishni esa MB darajasida belgilashni taklif qiladi. Ular rolini (ma'lumotlardan foydalanish huquqi) aniqlash va rioya qilinishini hamda huquqlarini belgilashni o'z zimmasiga oladi.

Ma'lumotlardan foydalanish huquqini belgilash ixtiyoriy jadvalning atributigacha tushirilgan. Resurslardan foydalanish yo'nalishlarga muvofiq tartibga solinadi. Birlamchi va kalitlar uchun butunlik deklarativ butunlik mexanizmi orqali ta'minlanadi. Buning uchun kalitni ko'rsatish kifoya qilinadi (butunlikni kafolatlovchi maxsus dasturlarni yozish zaruriyati yo'qoladi). Oracle har bir birlamchi kalit uchun takrorlanmas noyob indeksni shakllantiradi. Foydalanuvchi qo'shimcha ravishda tashqi kalitlar uchun ham indeksni shakllantirish buyurtmasini berishi mumkin. Oracle SQL da qabul qilingan cheklov (constraints) va triggerlarni talab darajasida ishlashini ta'minlaydi. Odatda, blokirovka yashirin tarzda, qator darajasida amalga oshiriladi, ammo blokirovkani blok darajasida ham tashkil qilish mumkin.

Блокировка – to'siq. Joriy amallar bajarilganga qadar, keyingi amallar bajarilishini taqiqlash. Umumiy tarzda ishlatiladigan vositalar tekshirilgan holda kirishni tashkil qilish mexanizmi.

Транзакция. MB ni o'zgartirish so'rovi. Foydalanuvchi MB dan ma'lumotni qabul qilishi, unga ishlov bergandan keyin javob ma'lumotini uzatishi.

Буферизация-буферlash. Asosiy xotiraning maxsus maydonlari (buferlar) bo'lib, asosiy xotira va tashqi qurilmalar orasidagi ma'lumot almashish uchun ma'lumotlar vaqtinchalik shu joyda saqlanadi.

Транзаксийalar iurnali. MBBTdan tushayotgan tranzaktsiyalarni qayd qiluvchi jurnal.

Бекор qilish (omkam)-ortga qaytarish. Tranzaktsiya vaqtida o'zgartirilgan ma'lumotlarni o'zgarganligini bekor qilib, avvalgi holatiga qaytarish amali.

Oracle da ANSI SQL standartiga muvofiq keluvchi SQL tili qo'llaniladi. Dasturiy majmua tarkibida SQL*Plus interaktiv interpretator mavjud. Bundan tashqari protsedura, funksiya va

triggerlarni yozish uchun kuchli protsedurali PL/SQL tili ham mavjud.

Hozirgi kunda Koddning 12 ta qoidasiga to'liq javob beruvchi birorta ham MBBT mavjud emas, ammo Oracle bu qoidalarining eng asosiylariga eng ko'proq javob bera oladigan MBBT hisoblanadi.

MBni MBBTda saqlash uchun jadval fazosi ajratiladi.

Jadval fazosi (tablespace). Jadval yoki indeks singari mantiqiy ob'ekt yoki segment saqlanuvchi jismoniy tuzilmadir.

Segment – yozuv yoki uning qismi.

OT da jadval fazosi fayl ko'rinishida saqlanadi. SYSTEM jadval fazosida qiymatlar lug'ati saqlanadi.

Qiymatlar lug'ati, ma'lumotlar lug'ati (словарь данных) MB ob'ektlarining tavsifi va tuzilishini saqlovchi tizim jadvali.

TEMP jadval fazosida vaqtinchalik ob'ektlar (tartiblash yoki yaratishda hosil bo'lgan jadvallar yoki indekslar) saqlanadi.

ROLL1 jadval fazosida bekor qilinganlar jurnali segmenti saqlanadi.

Tranzaktsiya jurnali bevosita fayl ko'rinishida saqlanadi va u uchun jadval fazo talab qilinmaydi.

Jadvallar (Tables) – Ma'lumotlar bazasini tashkil qiluvchi asosiy jadvallar. Jadvallar satr va ustunlardan tashkil topadi. Bitta yoki bir nechta jadvallar jadval sohasini (Tablespace) ichida saqlanadi.

Blok (block) – (TMBBT) ORACLE ma'lumotlarni saqlashni eng kichik birligi. U blokni o'zi (ma'lumotlar yoki PL/SQL –kodi) va axborotlar sarlavhasidan iborat. Blokni o'lchami 2 dan 16 Kbaytgacha bo'ladi.

Bufer (buffer) – ma'lumotlarni saqlash uchun ishlatiladigan birorta operativ xotira hajmi. Buferda ishlatishga mo'ljallangan yoki yaqin orada ishlatilgan ma'lumotlar joylashadi. Ko'pgina hollarda buferda qattiq diskda saqlanayotgan ma'lumotlarni nusxalari joylashadi. Buferdagi ma'lumotlar o'zgartirilishi

mumkin va disk yozib qo'yilishi yoki vaqtincha saqlovchi buferga joylashtirilishi mumkin. Buferlar to'plami (guruhi) ma'lumotlarni kesh buferni tashkil qiladi. Shuningdek buferda o'zgarishlar jurnalidagi vaqtinchalik yozuvlar saqlanishi mumkin. Ular keyinchalik diskka yoziladi (o'zgarishlar jurnali buferi).

Iflos bufer (Dirty buffer) – ma'lumotlari o'zgartilgan bufer. Xarakteristikalarini dinamik jadvallari (Dynamic Performance Tables) – ORACLE nusxasi ishga tushganda avtomatik ravishda yaratiladigan jadval va u shu nusxani xarakteristikalarini saqlash uchun ishlatiladi. Ular o'z ichida bog'lanishlar, kiritish(chiqarish), muhit parametrlarini boshlang'ich qiymatlari haqidagi axborotlarni saqlaydi.

Indeks (Index) – Indekslar ustun yoki ustunlar guruhi bo'yicha yaratiladi. Indeks ma'lumotlarni tezkor va samarali topish imkonini beradi. Jadvallarga murojat indekslangan ustun (yoki ustunlar) bo'yicha amalga oshiriladi.

Klaster (Cluster) – Klaster bu jadvallar guruhi (to'plami) bo'lib birgalikda ishlatadigan umumiy ustunga ega bo'ladi va fizik nuqtai nazardan bitta jadvalday saqlanadi. Agar ma'lumotlar ikki yoki bir nechta jadvallardan bir vaqtda umumiy ustundagi ma'lumotlar bo'yicha qidiralayotgan bo'lsa, klasterlar katta samara berishi mumkin. Jadvallarga ular klasterni qismi bo'lsa ham alohida murojat qilishi mumkin. Klasterlarni strukturasi bog'langan ma'lumotlarga bir vaqtda murojat qilinganda kiritish va chiqarish xarajatlarini kamaytirish imkonini beradi.

Tasavvur (View) – ma'lumotlar bazasida mavjud bo'lishi shart bo'lmagan, lekin alohida foydalanuvchini so'rovini bajarish vaqtida yaratiladigan virtual jadvaldir. Tasavurlar va jadvallar bilan bajariladigan amallarni (so'rovlar qurish, olib tashlash, tiklash) cheklanmagan sonda bajarish mumkin.

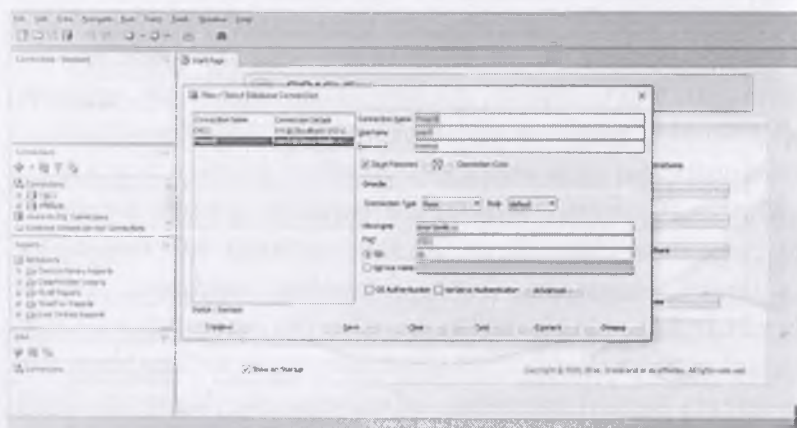
Sinonimlar (Synonyms) – Ma'lumotlar bazasi ob'ektlari uchun alternativ nomlar.

Ketma – ketliklar (Sequences) – Ma’lumotlarni kesh buferida saqlanuvchi raqamlar ketma – ketligini yaratish uchun ishlatiladi.

Ma'lumotlar lug'ati (Data Dictionary) – Ma'lumotlar bazasidagi axborotlarni ta'minlash uchun ishlatiladigan jadvallar to'plami.

Sxema (Schema) - ma'lumot baza ob'ektlarining majmuasi.

Toza bufer (clean buffer) – Ichidagi ma'lumotlar o'zgartirilmagan bufer. Undagi ma'lumot o'zgarmagani uchun uni qattiq diskka yozishga zarurat yo'q.



13.1-rasm. ORACLE ma'lumotlar bazasiga ulanish

Umuman olganda Oracle yaxshi himoya vositalariga, ma'lumotlar butunligini ta'minlash kafolatiga, to'liq tiklanish xususiyatiga, juda katta hajmli MB bilan ishlash hamda minglab foydalanuvchilarga bir vaqtning o'zida xizmat qilish quvvatiga va boshqa keng imkoniyatlarga ega bo'lgan MBBTdir. U kuchli SQL*Plus va PL/SQL tili bilan ta'minlangan. Oracle taqsimlangan MB bilan ishlay oladi va to'liq tiklanish xususiyatiga hamda har qanday web xizmatlarni qo'llash imkoniga ega.

Oracle dunyodagi eng yirik MBBTni ishlab chiquvchi kompaniyadir. Bugungi kunda Oracle MBBT 80 dan ortiq

operatsion tizimlarda ishlaydi. Oracle yangi versiyalari platformaga bog'liq bo'lmagan holda, Web texnologiya asosida ishlash xususiyatiga ega. Dunyoning 100 dan ortiq mamlakatlarida 40000 dan ortiq xodimlari kompaniyaga milliardlab dollar foyda keltirishadi.

13.3. ORACLE MBBTning asosiy ob'ektlari

ORACLE asosiy ob'ektlariga quyidagilar kiradi: jadvallar, tasavvurlar va foydalanuvchi.

Foydalanuvchi (USER) – ob'ekt bo'lib, u ORACLE ni boshqa ob'ektlarini yaratish va ishlatishga, hamda server funksiyalarini bajarishni so'rash imkoniyatiga ega. Bunday funksiyalar soniga sessiyalarni tashkil qilish, ma'lumot bazasini xolatini o'zgartirish va boshqalar kiradi. ORACLE foydalanuvchisi bilan sxema (SCHEMA) bog'langan.

Sxema foydalanuviga tegishli bo'lgan ma'lumotlar bazasi ob'ektlarini mantiqiy to'plami (jadvallar, ketma ketliklar, saqlanuvchi dasturlar). Sxema faqat bitta ega foydalanuvchiga tegish bo'lib, faqat u bu ob'ektlarni yaratish va olib tashlashga javobgardir. Sxema quyidagi ob'ektlarni saqlashi mumkin: klasterlar, ma'lumot baza bog'lanishlari, triggerlar, tashqi procedura bibliotekasi, indekslar, paketlar, ketma ketliklar, saqlanuvchi funksiya va proceduralar, sinonimlar, jadvallar, tasavvurlar.

Jadvalar (TABLE) relyatsion ma'lumotlar bazasini asosini tashkil qiladi. Relyacion ma'lumotlar bazasida barcha axborotlar jadvallarda saqlanadi. Jadvallarni to'la nomi sxema nomi va jadval nomidan tashkil topadi.

Rol (ROLE) – Foydalanuvchilar yoki boshqa rollarga taqdim etiladigan imtiyozlarni majmuasi. Jadval sohasi (TABLESPACE) - ma'lumot bazasini nomlangan qismi bo'lib, jadvallar, indekslar va boshqa ob'ektlar uchun xotirani taqsimlash uchun ishlatiladi. Ma'lumotlarni qayta ishlash algoritmlarin dasturlash uchun,

ma'lumot bazasini yaxlitligini dinamik ta'minlash mexanizmini amalga oshirish uchun ORACLE quyidagi ob'ektlarni ishlatadi: Procedura (Procedure), funksiya (FUNCTION), paket (PACKAGE), trigger (TRIGGER) kabi ob'ektlarni ishlatiladi.

13.4. ORACLEda ma'lumot toifalari

SQL ning 1992 yilgi standartlarida sanab o'tilgan barcha toifalar Oracle tomonidan ta'minlanadi.

Simvolli satrlar. **CHARACTER** toifasi fiksirlangan uzunlikdagi satrlarni saqlash uchun ishlatiladi. Satrlarni saqlash uchun uzunlik parametrida ko'rsatilgan soha rezerv(zahira) qilib bo'yiladi. Zarur bo'lganda qisqa satr probellar bilan to'ldiriladi.

Sintaksisi: **CHARACTER**[(uzunlik)], **CHAR**[(uzunlik)]

Agar satr uzunligi oshkor holda ko'rsatilmasa, u 1 teng deb hisoblanadi. Uzunlik parametrini maksimal qiymati- 256 simvoldan iborat.

Misol: str1 **CHAR**(10), Str2 **CHARACTER**

VARCHAR toifasi o'zgaruvchan uzunlikdagi satrlarni saqlash uchun ishlatiladi. Satrlarni saqlash uchun real zarur bo'lgan soha rezerv(zahira) qilib olinadi.

Sintaksisi:

VARCHAR[(uzunlik)], **CHAR VARYING** [(uzunlik)], **CHARACTER**

VARYING [(uzunlik)]

Agar satr uzunligi oshkor berilmagan bo'lsa, u 1 teng deb hisoblanadi. Uzunlik parametrini maksimal qiymati - 4000 simvoldan iborat.

Misol: varstr1 **VARCHAR**(10)

VARCHAR2 toifasi [faqat ORACLE uchun]. O'zgaruvchan uzunlikdagi satrlarni saqlash uchun ishlatiladi. Satrlarni saqlash uchun real zarur bo'lgan soha rezerv(zahira) qilib olinadi. **VARCHAR2** toifani kiritishga sabab firma ORACLE keyin

versiyalarida bu toifani o'zgarimasligini ta'kidlaydi, vaholanki VARCHAR toifasi SQL standartlar talablariga mos keladi.

Sintaksisi: **VARCHAR2**(uzunlik)

Satr uzunligi oshkor ko'rsatilishi shart. Uzunlik parametrini minimal qiymati – 1 simvol. Uzunlik parametrini maksimal qiymati – 4000 simvol.

Misol: str1ora VARCHAR2(10)

Bu ikki toifa o'rtasida farq bor: VARCHAR o'zgaruvchan uzunlikdagi satrlarni saqlash uchun ishlatiladi. Satrlarni saqlash uchun real zarur bo'lgan soha rezerv (zahira) qilib olinadi.

CHAR toifadagi ma'lumotlar fiksirlangan uzunlikka ega va 255 tagacha simvoldan tashkil topishi mumkin. Vaholanki, VARCHAR2 o'zgaruvchan uzunlikka ega va 4000 tagacha simvolga ega bo'lishi mumkin.

Ixtiyoriy CHAR(255) toifali ustun qiymati 255 bayt disk sohasini egallaydi. Bu farqni jadval strukturasi loyixalashda, ma'lumotlarni saqlash uchun joy ajratishda va satrlarni SQL – ifodalar solishtirishda hisobga olish kerak. Agar ikkita satr har xil uzunlikka ega bo'lsa, ularni solishtirishda qisqaroq satr ikkinchi uzun satr uzunligicha probel bilan to'ldiriladi. Milliy alfavitni ishlatadigan simvolli ma'lumotlarni saqlash uchun NCHAR va NVARCHAR2 toifalari mo'ljallangan.

Sonli toifalar

INTEGER toifasi. INTEGER toifasi – 2^{31} dan 2^{31} gacha diapazondagi butun sonlarni tasvirlash uchun ishlatiladi.

Sintaksisi: INTEGER, INT

Misol: varint1 INTEGER, varint2 INT

NUMBER toifasi [faqat Oracle uchun]. NUMBER toifasi berilgan aniqlikda sonlarni tasvirlash uchun ishlatiladi.

Sintaksisi: NUMBER [(aniqlik [,masshtab])]

Agar aniqlik parametrini qiymati oshkor holda ko'rsatilmasa, u 38ga teng deb hisoblanadi. Masshtab parametrini qiymati sukut bilan 0 teng deb olinadi. Aniqlik parametrini

qiymati 1 dan 38 gacha o'zgarishi mumkin, masshtab parametrlarini qiymati – 84 dan 127 gacha o'zgarishi mumkin.

Masshtabni manfiy qiymatlarini ishlatish o'nli nuqtani yuqori razryadlar tomon siljishini anglatadi. Masalan, NUMBER (7, -3) minglar xonasigacha yaxlitlashni anglatadi.

Misol: varcounter NUMBER

Quyidagi jadvalda 123456.789 sonini har xil aniqlik va masshtabda tasvirlanishi keltirilgan.

Aniqlanishi	Amallar bajarishdagi ko'rinishi
Number	123456
Number(7,1)	123456.7
Number(5,2)	Ma'lumotlarda hato
Number(9,3)	123456.789
Number(7,-2)	123500

Boshqa MBBT bilan mos ta'minlash uchun Oracle DECIMAL, DOUBLE_PRECISION, NUMERIC, DEC va REAL ma'lumot toifalarini ham ishlatadi.(ta'minlaydi). Sonli toifalarni barchasi real ravishda Oracle ichki bir xil formatida saqlanadi. DECIMAL va NUMERIC toifalari NUMBER toifasiga to'la ekvivalent.

Sintaksisi:

DECIMAL [(aniqlik [,masshtab])], DEC [(aniqlik [,masshtab])]

NUMERIC [(aniqlik [,masshtab])]

Misol:

vardec DEC, vardec2 DEC(5), vardec3 DECIMAL(8,3), varnum NUMERIC

ROWID toifasi

ROWID – maxsus ma'lumot toifasi bo'lib, jadvaldagi yozuvga ko'rsatkichlarni tasvirlash(qo'yish) uchun xizmat qiladi.

Jadvalda satr yaratilganda unga darhol ROWID tayinlanadi va u ma'lumotlar qayta tashkil qilingunga yoki uni olib tashlangunga qadar o'zgarmasdan qoladi. ROWID ishlatish jadvaldagi satrga murojaatni eng tezkor usulidir.

Bitli satrlar

RAW toifasi [faqat Oracle uchun] – O'zgaruvchan uzunlikdagi ikkilik satrlarni saqlash uchun ishlatiladi. RAW toifani CHAR, VARCHAR2 toifadan farqi shundan iboratki, Oracle simvolli toifalar uchun klient bilan server orasida ularni uzatayotganda ma'lumotlarni avtomatik almashtiradi. Oracle RAW toifasidagi ma'lumotlarni o'noltilik ko'rinishda chiqaradi.

Sintaksisi: RAW [(uzunlik)]

Uzunlik parametri baytlarda o'lchanadi. Uzunlik parametri maksimal qiymati – 2000 bayt.

Misol: bitarray1 RAW(10)

LONG RAW toifasi[faqat Oracle uchun]. O'zgaruvchi uzunlikdagi katta bitli satrlarni saqlash uchun ishlatiladi.

Sintaksisi: LONG RAW [(uzunlik)]

Uzunlik parametri baytlarda o'lchanadi. Agar satr uzunligi oshkor holda berilgan bo'lmasa, u 2 megabaytga teng deb hisoblanadi. Uzunlik parametrini maksimal qiymati – 2gigabayt simvoldan iborat. LONG RAW toifadagi o'zgaruvchilar uchun indeksni qurish mumkin emas.

Misol: bitarray1 RAW(10)

Sana va vaqt toifalari

DATE toifasi [faqat Oracle uchun] – bu toifa sana va vaqt saqlash uchun ishlatiladi. Eramizdan avvalgi 1 yanvar 4712 dan to 31 dekabr 4712 eramizgacha sanalarga ruxsat beriladi. SQL va PL/SQL da DATE (sana) toifalarini qiymatini yaratish uchun odatda kiritilgan funksiyalar TO_DATE ("sanani simvolli satri", "sana_formati") ishlatiladi.

Vaqtни belgilamasdan sana aniqlansa, sukut bilan yarim kechasi vaqti qabul qilinadi. SYSDATE funksiyasi joriy sana va

vaqtni chiqaradi. Funkciyani qiymati Oracle serveri ishlayotgan kompyuter operacion tizimi vositalari bilan aniqlanadi.

Sintaksisi: DATE Misol: birthday DATE

Sana va vaqtni saqlash uchun maxsus toifani (borligi) mavjudligi sana va vaqt maxsus arifmetika amallarini bajarish imkonini ta'minlaydi. DATE toifasidagi o'zgaruvchiga butun sonni qo'shish Oracle tomonidan etarli kechki sanani aniqlash deb, ayirish esa oldingi sana aniqlash deb talqin qilinadi.

Misollar:

```
SQL> SELECT SYSDATE FROM dual;
```

SYSDATE

16-09-2011

```
SQL> SELECT SYSDATE+10 "sysd+10" FROM dual;
```

sysd+10

26-09-2011

```
SQL> SELECT SYSDATE-10 "sysd-10" FROM dual;
```

sysd-10

06-09-2011

LOB – ob'ektlari

BLOB toifasi [faqat Oracle uchun] o'lchami 4 gigabayt bo'lgan ikkilik ma'lumotlarni saqlash uchun ishlatiladi. Katta ikkilik ob'ektlari bilan ishlash uchun DBMS_LOB paketi ishlatiladi.

Sintaksisi: BLOB Misol: doc BLOB

CLOB toifasi [faqat Oracle uchun] bir baytli kodirovkani ishlatadigan o'lchami 4 gigabaytgacha o'zgaruvchan uzunlikdagi simvolli ma'lumotlarni saqlash uchun ishlatiladi.

Sintaksisi: CLOB Misol: cdoc CLOB

NCLOB toifasi [faqat Oracle uchun] bitta yoki ko'p baytli kodirovkani ishlatadigan, o'lchami 4 gigabaytgacha bo'lgan simvolli ma'lumotlarni saqlash uchun ishlatiladi.

Sintaksisi: NCLOB Misol: ncdoc NCLOB

BFILE toifasi [faqat Oracle uchun] MBBTga nisbatan tashqi fayllarda joylashgan, ikkilik ma'lumotlar ko'rsatkichlarini saqlash uchun ishlatiladi. Fayllarni o'zi fayl tizimida saqlanadi

Sintaksisi: BFILE Misol: File_doc BFILE

Nazariy savollar

1. LOB – ob'ektlarni sanab bering.
2. Sana va vaqt toifalari ishlatishni tushuntirib bering.
3. Sonli toifalar haqida ma'lumot bering.

13.5. Tasavvurlar, ketma-ketlik va sinonimlar

Tasavvur – bir yoki bir necha jadvallardan yoki tasavvurlardan olinadigan, server tomonidan dinamik ravishda ta'minlanadigan nomlanadigan tanlashdir. Oracle da tasavvurni aniqlash operatorini sintaksisi:

```
CREATE [OR REPLACE] [{FORCE|NO FORCE}]  
VIEW[sxema_nomi.]      tasavvur_nomi      [(al'ternativ_nom  
[al'ternativ_nom...])] AS so'rov WITH {READ ONLY|CHECK  
OPTION[CONSTRAINT yaxlitlikni_cheklash]}
```

OR REPLACE kalit so'z eski tasavvurni yangi bilan majburiy almashtirishni kerakligini ko'rsatadi.

FORCE | NO FORCE (sukut bilan olinadi) asosiy jadvallarni va tasavvurlarni mavjudligi va tasavvur yaratuvchi foydalanuvchida asosiy jadvallarga murojat uchun ruxsatlar bo'lishi majburiyligini ko'rsatadi. So'rov parametri ixtiyoriy to'g'ri so'rovni anglatadi. Bu so'rovlarda ORDER BY kalit so'z yoki FOR UPDATE konstruktsiya bo'lmasligi kerak.

WITH READ ONLY kalit so'z asosiy (bazaviy) jadvallar uchun ko'rsatilgan tasavvur bilan ma'lumotlarni takomillashtirishni ta'qiqlashni ko'rsatadi. CONSTRAINT kalit so'zi tekshirish uchun ishlatiladigan cheklash nomini aniqlaydi. Faqat o'qish uchun cheklanishli tasavvurni yaratishga doir misol:

```
SQL>CONNECT ul/ulpsvf@educ;
```

```
Connected.
```

```
SQL>CREATE TABLE Tab2 (At1 NUMBER, At2  
NUMBER);
```

TABLE created.

```
SQL>CREATE VIEW Vul AS SELECT * FROM Tab2 WITH  
READ ONLY;
```

View created.

```
SQL>GRANT INSERT ON Tab2 TO u2;
```

Grant succeeded.

```
SQL>GRANT INSERT ON Vul TO u2;
```

Grant succeeded.

```
SQL>CONNECT u2/u2psw@educ;
```

Connected.

```
SQL>INSERT INTO ul.Vul VALUES(1,2);
```

```
INSERT INTO ul.VUI VALUES (1,2) * ' ERROR at line 1:  
ORA-01732: data manipulation operation not legal on this view
```

```
SQL> INSERT INTO ul.Tab2 VALUES(1,2);
```

Faqat o'qish uchun cheklanishli tasavvurni yaratish protokoli.

Tasavvurni olib tashlash

Tasavvurni olib tashlash uchun DROP VIEW komandasi ishlatiladi. Tasavvurni olib tashlash uchun uni egasi bo'lishi kerak yoki DROP ANY VIEW imtiyoziga ega bo'lish kerak. Komandani sintaksisi quyidagicha:

```
DROP VIEW [sxema_nomi.]tasavvur_nomi
```

Tasavvurni olib tashlashganda, unga murojat ob'ektlar olib tashlanmaydi, ular xaqiqiy bo'lmay qoladi. Olib tashlanayotgan tasavvurlarga beriladigan imtiyozlar bekor qilinadi. Misol: vu3 Tasavvurni olib tashlashga misol.

```
SQL>DROP VIEW vu3;
```

View dropped.

Uzoqlashgan Oracle ma'lumot bazasi bilan bog'lanish yaratish

Uzoqlashgyn mv'lumot bazasi bilan bog'lanish o'rnatish uchun SQL -operatori CREATE DATABASE LINK komandasi ishlatiladi. Bunda lokal va uzoqlashgan ma'lumot bazasida maxsus dasturiy ta'minot o'rnatilgan bo'lishi kerak. Uzoqlashgan ma'lumot bazasi bilan bog'lanish operatorini sintaksisi quyidagicha:

```
CREATE [PUBLIC] DATABASE LINK MB bilan
bog'lanish_nomi [CONNECT TO foydalanuvchi_nomi
IDENTIFIED BY foydalanuvchi_paroli] USING 'bog'lanish
_satri'
```

Misol: sun_ora_link bog'lanishi aniqlangan misolni ko'rib chiqamiz. Bu bog'lanish ul foydalanuvchi hisobot yozuvini ulpsw parolini ishlatadi. sunora bog'lanish satri bilan aniqlanagan. Uzoqlashgan ma'lumotlar bazasi bilan muvaffaqiyatli bog'lanish o'rnatilgandan keyin, uni jadvallariga so'rovlar bajarish mumkin.

```
SQL> CREATE DATABASE LINK sun_ora_link
CONNECT TO ul IDENTIFIED BY ulpsw USING 'sunora';
```

Database link created.

```
SQL> SELECT * FROM Tab1@sun_ora_link;
```

A
t1
1
2

Foydalanuvchidan ul foydalanuvchini Tab1 jadvali joylashganini berkitish (скрыть) uchun sinonimdan foydalanish mumkin. Misol: Quyidagi misolda uzoqlashgan ma'lumot bazasi bilan oshkormas holda bog'lanishni ta'minlash uchun va undan foydalanish so'rovini listingi keltirilgan.

```
SQL> CREATE SYNONYM suntab1 FOR
ul.Tab1@sun_ora_link;
```

Synonym created.

```
SQL> SELECT * FROM suntab1;
```

A

t1

1

2

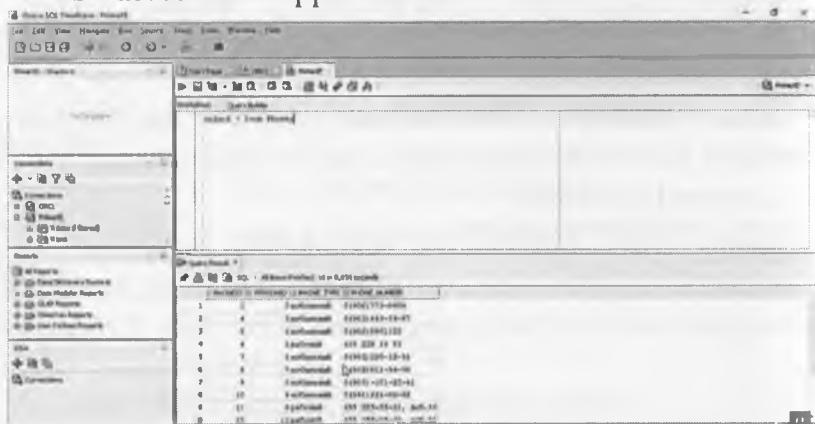
DROP DATABASE LINK komandasi. Bu komanda uzoqlashgan ma'lumot bazasi bilan bog'lanishni uzush uchun ishlatiladi. Bu komanda sintaksisi:

```
DROP [PUBLIC] DATABASE LINK MB bog'lanish_nomi
```

Misol: Sun_ora_link nomli uzoqlashgan ma'lumotlar bazasi bilan bog'lanish olib tashlash.

```
SQL> DROP DATABASE link sun_ora_link;
```

Database link dropped.



13.2-rasm. ORACLE MBBTga SQL Developer yordamida ulanish va so'rov yuborish

Ketma – ketlik

Takrorlanmaydigan butun sonlarni generatsiya qiladigan ma'lumotlar bazasi ob'ekti ketma ketlik deyiladi. Ketma – ketlik natijasida hosil qilingan sonlar odatda birlamchi kalit qiymati sifatida ishlatiladi. Ketma-ketlik natijasida hosil qilingan sonlar doim o'sishi, yoki ma'lum chegargacha o'sishi yoki chegaraga

etgandan boshlab yana kaytadan o'sishi mumkin. Shuningdek orttirmani qiymatini ham berish mumkin.

Ketma-ketlikni aniqlash operatori sintaksisi:

```
CREATE SEQUENCE [sxema_nomi.] ketmaketlik_nomi  
[INCREMENT BY orttirma] [START WITH boshlang'ich_qiymat]  
[MAXVALUE eng_katta_qiymat|NOMAXVALUE][MINVALUE  
eng_kichik_qiymat|NOMINVALUE]  
[CYCLE|NOCYCLE][CACHE  
elementlar_soni|NOCACHE][ORDER|NOORDER]
```

INCREMENT BY parametri ketma-ket nomer orasidagi intervalni beradi. START WITH kalit so'z boshlang'ich qiymat parametri orqali birinchi generatsiya qilingan nomerni beradi. Agar bu parametr berilmasa, o'suvchi ketma ketlik uchun boshlang'ich generatsiya qiladigan nomer MINVALUE qiymatiga (kamayuvchi uchun MAXVALUE) teng qilib olinadi.

Seq1 ketma - ketlikni yaratishni ko'rib chiqamiz. Ketma - ketlikni boshlang'ich elementi 2 ga teng deb aniqlanagan, eng_katta_qiymat eng_kichik_qiymat parametri mos ravishda 3 va 1 teng deb aniqlangan.

```
SQL> CREATE SEQUENCE Seq1 MAXVALUE 3  
MINVALUE 1 START WITH 2;
```

Sequence created.

```
SQL> SELECT Seq1.NEXTVAL FROM dual;
```

NEXTVAL

2

```
SQL> SELECT Seq1.NEXTVAL FROM dual;
```

NEXTVAL

3

```
SQL> SELECT Seq1.NEXTVAL FROM dual;
```

ERROR: ORA-08004: sequence SEQ1.NEXTVAL exceeds MAXVALUE and cannot be instantiated no rows selected

Oldingi misolni o'zgartirib Seq2 ketma-ketligini yaratamiz. Ketma ketlikni boshlang'ich elementi, eng_katta_qiymat, eng_kichik_qiymat parametri ham oldingi misoldagi kabi, faqat qo'shimcha qilib sikl parametri 2 ga teng, va keshlash parametri 2 ga teng deb aniqlangan. (agar shunday qilib keshlash parametri aniqlanmasa, xatolik paydo bo'ladi, chunki CACHE kalit so'zini elementlar_soni parametri MAXVALUE va MINVALUE kalit so'zlar bilan beriladigan parametrlarini ayirmasidan katta bo'lmasligi kerak).

```
SQL> CREATE SEQUENCE Seq2 MAXVALUE 3  
MINVALUE 1 START WITH 2 CYCLE CACHE 2;
```

Sequence created.

```
SQL> SELECT Seq2.NEXTVAL FROM dual;
```

NEXTVAL

2

```
SQL> SELECT Seq2.NEXTVAL FROM dual;
```

NEXTVAL

3

```
SQL> SELECT Seq2.NEXTVAL FROM dual;
```

NEXTVAL

1

DROP SEQUENCE komandasi ketma ketlikni olib tashlash uchun ishlatiladi. Komanda sintaksisi:

```
DROP SEQUENCE [sxema_nomi.]ketma_ketlik_nomi
```

Misol: Ketma ketlikni olib tashlash misol

```
SQL> DROP SEQUENCE Seq1;
```

Sequence dropped.

ORACLE da sinonimlar yaratish

Sinonim alternativ nomlash uchun ishlatiladigan ma'lumot baza ob'ekti hisoblanadi. Odatda sinonim jadvallar, tasavvurlar, umumiy foydalanishga mo'ljallangan (sxema prefiksni ko'rsatmagan holda) ketma ketliklarni yoki uzoqlashgan ma'lumotlar bazasiga murojatni berkitish uchun yaratiladi.

Sintaksisi:

```
CREATE [PUBLIC] SYNONYM
```

```
[sxema_nomi.]sinonim_nomi FOR
```

```
[sxema_nomi2.]ob'ekt_nomi [@MBga bog'lanish nomi]
```

Misol: RefCodes jadvali foydalanuvchi Administrator tomonidan berilgan va quyidagi komandalar bilan to'ldirilgan bo'lsin.

```
CREATE TABLE RefCodes(At1 NUMBER, At2  
VARCHAR2(5));
```

```
INSERT INTO RefCodes VALUES(1,'Text1');
```

```
INSERT INTO RefCodes VALUES(2,'Text2');
```

SYSTEM foydalanuvchi u1 foydalanuvchiga, foydalanuvchi Administrator sxemasidan RefCodes jadvali murojat qilgan holda sinonim yaratadi. Bunda sinonimni to'g'riligiga qaramasdan, sinonim murojat qilayotgan ob'ekt, foydalanuvchi u1 taqiqlangan (nedostupen).

Sinonim yaratishga doir misol

```
SQL> CONNECT system/manager@sunora;
```

Connected.

```
SQL> CREATE SYNONYM ul.RefCodes FOR  
Administrator.RefCodes;
```

Synonym created.

```
SQL> CONNECT ul/ulpsw@sunora;
```

Connected.

```
SQL> SELECT * FROM RefCodes;  
SELECT * FROM REFCODES * ERROR at -line 1: ORA-  
00942: TABLE or view does not exist
```

Nazorat savollari

1. Ketma – ketliklar qanday yaratiladi?
2. Sinonimlar qanday yaratiladi?
3. Uzoqlashgan ma'lumotlar bazasidagi jadvalning ustida qanday qilib amallar bajariladi.

13.6. PL/SQLda dasturlash

PL/SQL da dastur tuzilishi (strukturasi)

PL/SQL da dasturi odatda 3 ta blokdan tashkil topadi: tavsiflash bloki, bajarish bloki va favquloda vaziyatlarni qayta ishlash bloki. Bajarish bloki BEGIN va END operator qavslarini ishlatish bilan strukturalashtirili mumkin. PL/SQL da dastur sintaksis quyidagicha rasmiylashtirilishi mumkin

DECLARE

operatorlar...

BEGIN

operatorlar...

EXCEPTION

operatorlar...

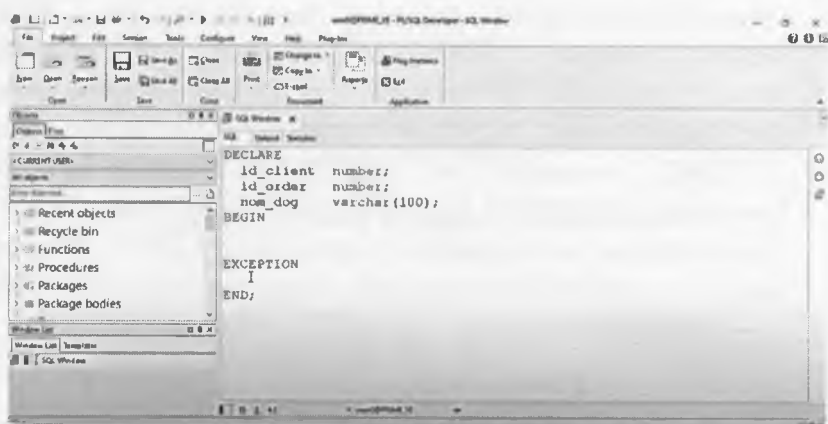
END;

Bunday ko'rinishdagi blok anonim blok deyiladi.

Bloklar haqida

DECLARE blokida o'zgaruvchilar konstantalar va foydalanuvchilar tomonidan aniqlanadigina ma'lumotlar tavsiflanadi (e'lon qilinadi) tavsiflanadigan. Birinchi BEGIN operatori asosiy dasturni jismini boshlanishini belgilaydi. Dastur jisimga boshqa BEGIN va END operatorlar qavslari bilan

chegaralangan boshqa bloklar kiritilishi mumkin. EXCEPTION blokida dasturda favquloda vaziyatlarni kayta ishlov uchun dastur kodini fragmenti beriladi. Oxirgi END operatori dastur jismini oxirini ko'rsatadi. Anonim blokda DECLARE va EXCEPTION bloklari bo'lmazligi mumkin, lekin BEGIN va END operator qavslari bilan chegaralangan operatorlar bloki bo'lishi kerak. Ba'zi nostandart hollarlarda (vyrojdennom) faqat NULL kelishi mumkin.



13.3-rasm. PL/SQL da dastur tuzilishi

PL/SQL dasturini ixtiyoriy qismida izoh (sharh) keltirish mumkin. «--» simvol bilan boshlanadigan va joriy satr oxirigacha davom etadigan ixtiyoriy matn izoh hisoblanadi. Ko'p satrli izohlar « /* » va « */ » simvollar orasiga olib yoziladi. Dasturda izohlar keltirish dasturlash amaliyotini yaxshi uslubiyati hisoblanadi. SHuningdek dasturlarni yozishda ularni o'qishni engillashtirish maqsadida satr boshiga nisbatan siljitib yozish ham, yaxshi uslubiyat hisoblanadi.

DECLARE bloki oldida har xil dasturiy vositalar uchun o'zgaruvchilar muhitini o'rnatish komandalari kelishi mumkin [8].

O'zgaruvchilar konstantalar va toifalar

PL/SQL da dasturi DECLARE va BEGIN operatorlari bilan cheklangan blokida o'zgaruvchilar, toifalar (tiplar) va konstantalar tavsiflanadi. Ixtiyoriy o'zgaruvchi yoki konstanta PL/SQL da mumkin bo'lgan toifalardan biriga ega bo'ladi. PL/SQL da ma'lumot toifalari yuqorida ko'rib chiqilgan SQL dagi ma'lumot toifalari (VARCHAR2, NUMBER, DATE, BOOLEAN va boshqalar) bilan (ba'zi bir muhim bo'lmagan farqlarni e'tiborga olmasa, ma'lumotlarni maksimal uzunligidagi farq, PL/SQL uchun maxsus ma'lumot toifalari PLS_INTEGER, BINARY_INTEGER va boshqalar) bir xil deb hisoblasa bo'ladi. Shu bilan birga PL/SQLda murakkab ma'lumot toifalari (RECORD, massivlar va PL/SQL -jadvallari) ishlatiladi. Konstanta CONSTANT kalit so'z bilan identifikatsiyalanadi va o'zgaruvchidan uni qiymatini o'zgartirishga harakat qilinganda, xatolik chiqarish bilan farq qiladi. O'zgaruvchiga qiymat berish «:=» operator bilan amalga oshiriladi.

O'zgaruvchiga qiymat berish

O'zgaruvchilarga qiymat berish barcha dastur tillarida muhim o'rin egallaydi. ORACLE da quyidagi konstruksiyalar mavjud:

O'zgaruvchi_nomi o'zgaruvchi_toifasi:=qiymat;

O'zgaruvchi_nomi:=qiymat;

Misol: Dasturni e'lon kilish bo'limi **DECLARE**

Sounter **number:=0;**

O'zgaruvchini initsializatsiya qilinayapti

Today **date:=sysdate;**

O'zgaruvchiga tizim sanasi berilayapti

Name **varchar2(25);**

O'zgaruvchi e'lon qilinayapti, lekin qiymat berilmayapti

RECORD murakkab toifasi

RECORD murakkab ma'lumot toifasi yozuvlarni saqlash va qayta ishlash uchun mo'ljallangan. Har bir yozuv atributlarga ega bo'lib, ularga boshlang'ich qiymat berish (initsializatsiya qilish) yozuvni e'lon qilishda amalga oshirilishi mumkin. Qiymat berishda o'zgaruvchi nomi va atribut nomi nuqta bilan ajratilib birgalikda berilishi kerak.

RECORD o'zgaruvchi toifasini kursorlar yordamida ma'lumotlarni tanlab olishda ishlatish qulay [22].

Nazorat savollari:

1. O'zgaruvchiga qiymat qanday o'zlashtiriladi?
2. Strukturalar qanday e'lon qilinadi?
3. Qiymat berish operatorini aytib bering?

13.7. PL/SQL tili alfaviti va muhit komandalar

- Barcha katta va kichik lotin harflari;
- Raqamlar: 0..9;
- Matematik operatorlar:
- ** - darajaga ko'tarish;
- *, / -ko'paytirish va bo'lish;
- +, -, || - qo'shish, ayirish, ulash amallari;

SET SERVEROUTPUT va SET ECHO - foydalanuvchi terminaliga chiqarish rejimini aniqlash komandalar.

Procedura DBMS_OUTPUT.PUT_LINE procedurasi ma'lumotlarni foydalanuvchi terminaliga chiqarishni taminlab beradi. Belgi (simvol) "/" - anonim blok matnini yakunlanganini va komandalarni tahlil qilish va bajarish kerakligini, belgilaydi.

Misol: O'zgaruvchilarni aniqlash va 2 va 3 sonlarni logarifmini hisoblash dasturini tuzamiz.

```
SQL> SET SERVEROUTPUT ON;  
SQL> SET ECHO ON;  
SQL> DECLARE
```

```

Header1 CONSTANT VARCHAR2(20) := 'Ikkinchi logarifmi
teng';
Header2 CONSTANT VARCHAR2(20) := 'Uchchi logarifmi
teng';
Arg NUMBER := 2; -- Bu yerda argumentni qiymati beriladi
-- Bajariluvchi blok
BEGIN
DBMS_OUTPUT.PUT_LINE(Header1 || LN(Arg));
Arg := Arg+1;
DBMS_OUTPUT.PUT_LINE(Header2 || LN(Arg));
END;
Natija:
Ikkinchi logarifmi teng
0.6931471805599453094172321214581765680814
Uchchi logarifmi teng
1.09861228866810969139524523692252570466
PL/SQL procedure successfully completed.

```

Shartli IF operatori

Umumiy xolda IF operatorini formati quyidagicha:

```

IF shart1
THEN operatorlar1; -- birinchi tarmoq
[ELSIF shart2 THEN operatorlar2;] -- ikkinchi tarmoq
.....
[ELSE operatorlarN;] -- alternativ tarmoq
END IF;
IF-THEN konstruksiyasi
IF shart1
THEN operatorlar;
END IF;
Misol:
IF l_date > '10-sep-10'
THEN

```

```
l_salary := l_salary *1.15; -- maosh 15 % oshiriladi.
```

```
END IF;
```

Dasturda IF - THEN operatorlarini bir-biriga joylashtirish mumkin.

Masalan:

```
IF l_date > '10-sep-10'
```

```
THEN
```

```
IF name = 'Ivanov'
```

```
THEN
```

```
l_salary := l_salary *1.15; --maosh 15 % oshiriladi.
```

```
END IF;
```

```
END IF;
```

IF-THEN-ELSE konstruksiyasi

```
IF shart1
```

```
THEN
```

```
operatorlar;
```

```
ELSE
```

```
operatorlar;
```

```
END IF;
```

Masalan:

```
IF l_date > '10-sep-10'
```

```
THEN l_salary:=l_salary*1.15; -- maosh 15% oshadi
```

```
ELSE l_salary:=l_salary*1.05; -- maosh 5% oshadi
```

```
END IF;
```

```
IF name = 'Ivanov'
```

```
THEN l_salary:=l_salary *1.15; --maosh 15% oshadi
```

```
ELSIF name='Aliev'
```

```
THEN l_salary:=l_salary*1.10; --maosh 10% oshadi
```

```
ELSE l_salary:=l_salary*1.05; -- maosh 5% oshadi
```

```
END IF;
```

Case operatori

Oracle 9i boshlanib case operatori PL/SQL tarkibiga kiritildi.

Uning formati quyidagicha:

CASE o'zgaruvchi

WHEN ifoda_1 THEN qiymat_1;

WHEN ifoda_2 THEN qiymat_2;

WHEN ifoda_3 THEN qiymat_3;

WHEN ifoda_4 THEN qiymat_4;

.....

WHEN ifoda_n THEN qiymat_n;

ELSE qiymat_(n+1);

END CASE;

Masalan,

DECLARE v NUMBER :=1;

BEGIN

CASE v

WHEN 1 THEN DBMS_OUTPUT.PUT_LINE('Bir');

WHEN 2 THEN DBMS_OUTPUT.PUT_LINE('Ikki');

WHEN 3 THEN DBMS_OUTPUT.PUT_LINE('Uch');

WHEN 4 THEN DBMS_OUTPUT.PUT_LINE('To'rt');

WHEN 5 THEN DBMS_OUTPUT.PUT_LINE('Besh');

END CASE;

END;

Sikllar

PL/SQL tilida 3 xil toifali sikllar ishlatiladi;

- LOOP sikl operatori;
- While sikl operatori;
- For sonli sikl operatori;

LOOP sikl operatori

LOOP sikl operatori formati:

LOOP

operatorlar;

EXIT [when shart];

END LOOP;

Masalan:

DECLARE v_counter **DINARY_INTEGER:=1;**

BEGIN

LOOP -- temp_Tab1t jadvaliga schotchik

-- qiymatlarini kiritamiz

INSERT INTO temp_TABLE **VALUES**(v_counter,'loop
index');

v_counter:=v_counter+1;

EXIT WHEN v_counter >50;

END LOOP;

END;

While sikl operatori.

Bu operatorni formati quyidagicha:

WHILE shart **LOOP**

operator1;

operator2;

.....

operatorN;

END LOOP;

Masalan:

DECLARE v_counter **BINARY_INTEGER:=1;**

WHILE v_counter <=50 **LOOP**

INSERT INTO temp_TABLE **VALUES**(v_counter, 'loop
index');

v_counter:=v_counter+1;

END LOOP;

END;

For sikl operatori

Bu siklda qaytarilishlar soni ma'lum. Siklni formati quyidagicha:

```
FOR «sikl_schetchigi» IN[REVERSE] «quyi_chegara»..  
«yuqori_chegara»
```

```
LOOP
```

```
operatorlar_ketma_ketligi;
```

```
END LOOP;
```

Bu erda: sikl schetchigi – oshkormas xolda yaratiladigan indeksli o'zgaruvchi; quyi chegara va yuqori chegara iteratsiyalar (takrorlanishlar) sonini ko'rsatadi; Operatorlar ketma – ketligi sikl mazmunini tashkil qiladi; sikl chegaralari bir marta hisoblanib qaytarishlar sonini aniqlaydi. Schetchik quyi chegaradan yuqori chegaragacha 1 qo'shib o'zgaradi. (sikl tugagunga qadar); schetchik(indeks) yoki sikl parametri oshkormas holda BINARI_INTEGER toifali deb e'lon qilinadi. Sikldan oldin uni e'lon qilish shart emas.

Misol: jadvalini 1..50 sonlar bilan to'ldirish.

```
BEGIN
```

```
FOR v_counter IN 1..50 LOOP
```

```
INSERT
```

```
INTO
```

```
temh_tadle
```

```
VALUES(v_counter,'loop_index');
```

```
END LOOP;
```

```
END;
```

FOR siklida REVERSE kalit so'zi ishlatilsa sikl indesi (teskari tartibda) yuqori chegaradan quyi chegaragacha o'zgaradi. Bunda chegaralarni joylashishi uzgarishsiz qoladi, ya'ni pastki chegara birinchi ko'rsatiladi.

```
BEGIN
```

```
FOR v_counter IN REVERSE 10..50 LOOP
```

```
//v_counter 50 dan boshlanidi va 1 ga kamayib boradi
```

```
NULL;
```

```
END LOOP;
```

END;

Sikl diapazonlari (yuqori va quyi chegaralar) sonli ifodalar bilan berilishi mumkin. Masalan:

DECLARE

v_lowvalue NUMBER:=10;

v_highvalue NUMBER:=40;

BEGIN

FOR v_counter IN REVERSE v_lowvalue..v_highvalue
LOOP

INSERT INTO temp_Tab1e VALUES(v_counter,'dinavic');

END LOOP;

END;

Nazorat savollari

1. Til alfaviti va muhit komandalari sanab bering.
2. Shartli IF operatori ishlash jarayoniga misol keltiring.
3. Case operatori ishlab jarayonini izohlang.
4. Har bir sikl operatorlari ishlash jarayoniga misollar keltiring.

13.8. Prosedura, funksiya va paketlar

PL/SQL tilida procedura, funksiya va paketlar keng ishlatiladi. Prosedura (procedure) aniq amallarni bajarishga mo'ljallangan birgalikda ishlatiladigan SQL va PL/SQL, tillarining operatorlari, o'zgaruvchilar va toifalar to'plamidan iborat dastur.

Funksiyalar(functions) aniq amallarni bajarishga mo'ljallangan birgalikda ishlatiladigan SQL va PL/SQL, tillarining operatorlari, o'zgaruvchilar va toifalar to'plamidan iborat dastur.

Prosedura va funksiyani farqi katta emas. Funksiya har doim chaqiruvchi dasturga qiymat qaytaradi. Prosedura bunday qilmaydi.

Paketlar (packages) bu procedura, funksiya, o'zgaruvchi va SQL operatorlar kolleksiyasidan iborat bo'lib, birgalikda

guruhlangan va yagona dastur moduli ko'rinishida saqlanadi. Procedura va funksiyalar ma'lumot baza ob'ektlari bo'lib keladi.

Foydalanuvchi protsedura va funksiyalarini yaratish

Protsedura va funksiyalar – ma'lumotlar bazasi ob'ektlaridir. Ular Create komandasi bilan yaratiladi va Drop komandasi bilan yo'qotiladi. Protsedura yoki funksiyalarni yaratishda ma'lumot ob'ekti nomlari, parametrlarni toifalari va ro'yxati hamda PL/SQL tilida kodlangan dasturni ishlash mantiqi aniqlangan bo'lishi kerak. Protsedura va funksiyalar yaratish uchun **Create procedure** tizim imtiyozlariga ega bo'lish kerak. Yangi protsedura va funksiya nomi aniqlangandan so'ng, uning nomini, toifasini va parametrlarini ko'rinishini berish kerak. Har bir parametr uchun odatda uni ko'rinishi IN, OUT, IN OUT kalit so'zlar bilan ko'rsatiladi.

IN ko'rinishli parametrdagi parametrni qiymati dasturga murojat qilganda aniqlanadi va dasturda o'zgartirilmaydi.

OUT parametr ko'rinishi, dasturni ishlash jarayonida parametr qiymatini o'zgartirish mumkinligini anglatadi.

IN OUT parametriga chaqirishda qiymat tayinlanadi va u dastur tanasida (jismida) o'zgartirilishi mumkin. Funksiya ta'rifida, protsedura ta'rifiga qo'shimcha, funksiya qaytarayotgan (chiqarayotgan) qiymatni toifasi ko'rsatiladi. Funksiya qiymatini qaytarishni RETURN operatori bajaradi [22].

Oracle procedurasini aniqlash operatorini sintaksisi

```
Create [or replace] procedure protsedura_nomi  
[(parametr_nomi[{IN | OUT | IN OUT }]) ma'lumot_toifasi  
[(parametr_nomi [{IN | OUT | IN OUT}]) ma'lumot_toifasi ...]]'  
{IS | AS} PL/SQL_da_dastur;
```

OR REPLACE kalit so'zi procedurani eski matnini shartsiz almashtirishni ko'rsatadi. Agar OR REPLACE kalit so'zi ko'rsatilmasa va procedura aniqlansa, procedura kodini eski

qiymatini almashtirish bo'lmaydi va xatolik haqida axborot chiqariladi. Ma'lumotlarni aniqlash bloki AS (yoki IS foydalanuvchini tanlashiga qarab) kalit so'zidan keyin boshlanadi.

Misol: Jadvalga sonli parametr va joriy sanaga bog'liq funksiya qiymatlarini kiritish procedurasini yarating.

Masalani echish uchun Tab1 jadvali yaratilgan bo'lsin:

Create table Tab1 (At1 number, At2 date);

Dastur listingi:

SQL> CREATE OR REPLACE PROCEDURE InsRec(Arg1
in number) AS

Coeff constant number:=0.5;

BEGIN

INSERT INTO Tab1 **VALUES**(coeff*Arg1, sysdate)

END;

procedure created

Procedurani bajarish

Procedurani bajarilishi natijasida ma'lumotlar bazasidagi bo'ladigan o'zgarishlarni quyidagi dastur kodi orqali bajarib ko'rish mumkin.

SQL> BEGIN

InsRec(240)

END;

/PL/SQL procedure successfully completed.

SQL> SELECT * FROM Tab1;

Nazariy savollar:

1. Procedura yaratish buyrug'ini tushuntirib bering.
2. Mavjud procedurani o'zgartirish buyrug'ini tushuntirib bering.
3. Proceduralar qanday vazifani bajaradi.

ORACLEda funktsiyani aniqlash operatori

```
CREATE [OR REPLACE] FUNCTION funksiya_nomi
[(parametr_nomi[{IN|OUT|INOUT}]      ma'lumot_toifasi
[,parametr_nomi
[{IN|OUT|INOUT}]  ma'lumot_  toifasi...])] RETURN
ma'lumot_toifasi
{IS|AS} RL/SQL_da_dastur;
```

Misol: Funksiya parametrida berilgan intervalga tushuvchi sanalarga mos keluvchi atribut qiymatlarini yig'indisini hisoblash funksiyasi yarating.

Yechish: Tab1 jadvali yaratilgan va to'ldirilgan bo'lsin.

```
CREATE TABLE Tab1 (At1 NUMBER, At2 DATE);
INSERT INTO Tab1 VALUES(5, SYSDATE);
INSERT INTO Tab1 VALUES(6, SYSDATE);
INSERT INTO Tab1 VALUES(7, SYSDATE+1);
```

Dastur listingi:

```
SQL> CREATE OR REPLACE FUNCTION SumRecInt
(Arg1 IN DATE, Arg2 IN DATE) RETURN NUMBER AS
SumVar NUMBER := 0;
BEGIN
SELECT Sum(At1) INTO SumVar FROM Tab1
WHERE At2 BETWEEN Arg1 AND Arg2;
RETURN SumVar;
END;
```

Function created.

Funksiyadan foydalanish

```
SQL> BEGIN
```

```
DBMS_OUTPUT.PUT_LINE(SumRecInt(SYSDATE-1/2,
SYSDATE+1/2));
END;
```

PL/SQL procedure successfully completed.

Procedura va funksiya o'chirib tashlash uchun mos ravishda quyidagi komandalardan foydalanamiz.

DROP PROCEDURE «procedura_nomi»;

DROP FUNCTION «funksiya_nomi»;

Paketlar

Paket ob'ekt sifatida olib qaralganda ikki qismdan tashkil topadi.

- Paket spesifikasiyasi (interfeysi);
- Paket tanasi (jismi).

Paket specifikaciyasida procedura va funksiyalar, global o'zgaruvchilar, konstantalar, toifalar, tashqi ilovalar kirishi uchun kursor tavsiflar saqlanadi.

Paket tanasida barcha proceduralar, funksiyalar va o'zgaruvchilar aniqlanadi. Paket tanasida aniqlangan, lekin uning spesifikasiyasida tavsiflanmagan procedura, funksiyalar, o'zgaruvchilar lokal hisoblanadi.

```
CREATE [OR REPLACE] PACKAGE paketa_nomi (IS|AS)  
PL/SQL_da_paketa_specifikaciyasi;
```

Misol: Konstantalar, funksiyalar va proceduralar tavsifidan tashkil topgan paket specifikaciyasini yarating.

```
SQL> CREATE OR REPLACE PACKAGE PACAA AS
```

```
PACAA_CONST CONSTANT NUMBER := 1.2;
```

```
FUNCTION MULCONST(Arg1 NUMBER) RETURN  
NUMBER;
```

```
PROCEDURE AUDITMUL;
```

```
END;
```

Package created.

Oracle paketini bajariluvchi qismini (tanasini) aniqlash operatorini sintaksisi

```
CREATE [OR REPLACE] PACKAGE BODY
```

```
Paketa_nomi (IS|AS) RL/SQL_da_paket_specifikaciyasi
```

Yuqorida keltirilgan listing uchun paket tanasini yaratamiz.

Faraz qilamiz, mulconst paket funksiyasi argumentni paket konstantasiga kupaytirishni bajarsin.

AUDITMUL procedura esa mulconst funksiyasiga murojatlar faktini fiksirlasin. Mulconst jadvalini yozuvlarini jadvaldagi qiymatlarni murojat schetchigiva joriy sanasi bo'lsin. Paket tanasi yaratilgunga qadar TabAUD jadvalini mos ravishda atribut toifalari yaratilgan bo'lsin.

```
SQL> CREATE OR REPLACE PACKAGE BODY PACAA AS
  PACAA_COUNT NUMBER := 0;
  FUNCTION MULCONST(Arg1 NUMBER) RETURN
NUMBER IS
  BEGIN
    AUDITMUL;
    RETURN Arg1*PACAA_CONST;
  END;
  PROCEDURE AUDITMUL IS
  BEGIN
    PACAA_COUNT := PACAA_COUNT + 1;
    INSERT INTO TabAUD VALUES(PACAA_COUNT,
SYSDATE);
    COMMIT;
  END;
END;
Package body created.
```

Paketdan foydalanish

```
SQL> BEGIN
  DBMS_OUTPUT.PUT_LINE(PACAA.PACAA_CONST);
END;
/1.2 PL/SQL procedure successfully completed.
SQL> BEGIN
  DBMS_OUTPUT.PUT_LINE(PACAA.PACAA_COUNT);
END;
/DBMS_OUTPUT.PUT_LINE(PACAA.PACAA_COUNT);
```

* ERROR at line 2: ORA-06550: line 2, column 30: PLS-00302: component

'PACAA_COUNT' must be declared ORA-06550: line 2, column 3:

PL/SQL: Statement ignored

SQL> BEGIN

DBMS_OUTPUT.PUT_LINE(PACAA.MULCONST(111));

END;

/133.2 PL/SQL procedure successfully completed.

SQL> SELECT * FROM TabAUD;

Nazariy savollar:

1. Paketlar qanday yaratiladi?
2. Funksiyalar yaratish haqida gapirib bering.
3. Yaratilgan paketlardan foydalanish qanday amalga oshiriladi?

13.9. ORACLEda triggerlarning ishlatilishi

Ma'lumotlar bazasi triggerlari – bu procedura bo'lib, aniq hodisalar paydo bo'lishi bilan ishga tushadi, hodisalar asosan ma'lumot jadvallarini modifikatsiyalash (takomillashtirish) olib tashlash yoki qo'shish amallarini bajarishi bilan bog'liq bo'ladi. Triggerni ishga tushirishni boshqaradigan hodisa mantiqiy shartlar ko'rinishida tasvirlanadi. Triggerni shartlariga mos hodisa vujudga kelganda, Oracle serveri avtomatik ravishda triggerni ishlatib yuboradi.

Trigger jadvalidagi ma'lumotlarni o'zgartirish Insert, Delete va Update amallaridan biri bajarilganda ishga tushadi.

Trigger kodi, Insert, Delete va Update amallarigacha, yoki undan keyin bajarilishi mumkin.

Masalan, trigger foydalanuvchini amallar bajarish huquqini tekshirish uchun trigger ishga tushishi kerak bo'lsa, unda albatta amallarni bajarguncha (Before kalit so'zi) bo'ladigan triggerni ishga tushirish kerak.

Agar trigger ma'lumotlarni auditoriya yozuvlari yaratishda ishlatilsa, unda triggerlarni amallarni bajarishdan keyin (After kalit so'zi) ishga tushirilganidan foydalanish mumkin. Trigger kodi butun jadval bilan amal bajarishi mumkin, yoki amal bajarilayotgan har bir satr bilan ishlashga moslashgan bo'lishi mumkin. Shunga qarab triggerlar operatorli triggerlarga yoki satrli triggerlar bo'linadi. Operatorli triggerlar butun jadval bilan amal bajaruvchi qoidalarini tekshirish uchun ishlatiladi.

Satrli triggerlar satrlarni qo'shishda (kiritishda) butunligini tekshirish uchun ishlatiladi. Satrli triggerlarni ishga tushirishni qo'shimcha mantiqiy shart bilan oydinlashtirish mumkin.

Triggerlarni aniqlash

Oracle'da triggerlarni aniqlash operatori quyidagicha sintaksisga ega:

```
CREATE [OR REPLACE] TRIGGER [sxema_nomi] trigger
nomi
{BEFORE | AFTER} [OR {INSERT|DELETE|UPDATE [OF
ustun_nomi {
ustun_nomi...}]}} ON [sxema_nomi.]
{jadval_nomi | tasavur_nomi} [FOR
EACH ROW] [When shart]
PL\SQL_da_dastur_spetsifikatsiyasi.
```

- OR REPLACE parametrlari eski trigger matnini shartsiz almashtirish kerakligini anglatadi. Agar OR REPLACE ko'rsatilmasa va trigger aniqlansa, eski trigger matnini almashtirish bajarilmaydi va xatolik haqida xabar chiqariladi.

- Before yoki After kalit so'zi trigger kodini bajarishini triggerlarni ishga tushirishini boshlab beruvchi (initsializatsiya qiluvchi) operatorlardan oldin yoki keyin ishga tushirishini bildiradi.

- INSERT, DELETE yoki UPDATE kalit so'zlari triggerlarni ishga tushiruvchi aniq operatorlar hisoblanadi. Zarur bo'lmagan OR kalit so'zi triggerlarni ishga tushiradigan qo'shimcha operatorni ulaydi. ON kalit so'zi trigger bilan bog'liq jadval nomini beradi.

- Zaruriy bo'lmagan FOR EACH ROW kalit so'zi satrli triggerni aniqlaydi. Zaruriy bo'lmagan WHEN kalit so'zi trigger ishga tushishini boshlanishini aniqlovchi hodisa sohasini toraytiradigan qo'shimcha mantiqiy shartni beradi (aniqlaydi).

new - procedurasi (old qo'shimchasi) faqat triggerlar uchun ishlatish mumkin. INSERT yoki UPDATE komandasi uchun ustunni yangi qiymatini belgilaydi.

Shuningdek qayta tiklash (UPDATE) va olib iashlash (DELETE) uchun: old prefiks ishlatiladi. Uning ma'nosi UPDATE yoki DELETE komandalarini bajargunga qadar ustun qiymatini belgilaydi [8].

Xatoliklarni qayta ishlash

Oracle'da favqulotda vaziyatlarni qayta ishlashni ko'rib o'tamiz. PL/SQL dasturi avariyali holatda yakunlanganda xatolikni aniqlash uchun RAISE_APPLICATION_ERROR procedurasi ishlatiladi. Uning yordamida foydalanuvchi tomonidan aniqlanadigan 1000 tagacha xatolikni qayta ishlash mumkin. Bu xatolikni nomerlari diapozoni 20 000 dan 20 999gacha.

RAISE_APPLICATION_ERROR procedurasi chaqirishi favquloda (maxsus) vaziyatni generatsiya qmlishga olib keladi va procedura dastur bajarilishini to'xtatishga olib keladi. Shu bilan birga, chaqirgan dastur muhitiga, xatolikni toifasi (tipini) ifodalovchi nomerli va matnli xabar chiqariladi.

Triggerni yaratish doir misollar

Talaba kodini hosil qilish va kiritish uchun trigger yaratamiz. Buning uchun dastlab ketma-ketlik tashkil qilamiz.

```
CREATE SEQUENCE stud_seq
```

Endi students jadvaliga yangi qator kiritish uchun trigger yaratamiz:

```
CREATE TRIGGER st_keys  
BEFORE INSERT ON students  
FOR EACH ROW  
BEGIN
```

```
SELECT stud_seq.NEXTVAL INTO :NEW.cod_st FROM  
dual;
```

```
END;
```

Novbatdagi masalada talabani jadvaldan o'chirish uchun trigger yaratamiz. Faraz qilaylik **archive_students** va **students** jadvallari mavjud.

```
CREATE TRIGGER st_del  
BEFORE DELETE ON students  
FOR EACH ROW  
BEGIN  
INSERT INTO archive_students  
VALUES(:OLD.cod_st, :OLD.name_st, :OLD.born,  
:OLD.phone);  
END;
```

Masala: Agar jadvalga atributni kiritilayotgan qiymati o'rta qiymatdan haddan tashqari kattalashib ketsa, ishga tushadigan triggerlarni yaratamiz. Haddan tashqari kattalikni o'lchovi sifatida, muhandislik amaliyotida keng qo'llanadigan «uchta sigma» qoidasini qo'llaymiz. Faraz qilaylik Tab1 jadvali yaratilgan va to'ldirilgan bo'lsin

```
CREATE TABLE Tab1(At1 NUMBER);  
INSERT INTO TAB1 VALUES(1);  
INSERT INTO TAB1 VALUES(3);  
INSERT INTO TAB1 VALUES(5);  
SQL> CREATE OR REPLACE TRIGGER TRIG_TB1  
BEFORE INSERT ON Tab1 FOR EACH ROW  
DECLARE  
StatAvg NUMBER;  
StatStd NUMBER;  
StatN NUMBER;  
BEGIN  
SELECT COUNT(At1), SUM(At1), STDDEV(At1) INTO  
StatN, StatAvg, StatStd FROM Tab1;  
IF (ABS(StatAvg-StatN*(.new.At1))/(SQRT(StatN)*StatStd)  
>3) THEN
```

```
RAISE_APPLICATION_ERROR(-20002, 'Haddan ziyod
chetlashuv');
END IF;
END;
/Trigger created.
```

Triggerning bajarilishi

```
SQL> INSERT INTO Tab1 VALUES(4);
1 row created.
SQL> INSERT INTO Tab1 VALUES(7);
INSERT INTO Tab1 VALUES(7)
* ERROR.at line 1: ORA-20002: Haddan ziyod chetlashuv
ORA-06512: at
"U1.TRIG_TB1", line 9 ORA-04088: error during execution of
trigger
'U1.TRIG_TB1'
```

```
SQL> SELECT * FROM Tab1;
```

A
T1
1
2
3
4

Nazariy savollar:

1. Trigger qanday yaratiladi?
2. Trigger qanday ishlatiladi?
3. Triggerning vazifasi nimalardan iborat?

XIV. NOSQL MA'LUMOTLAR BAZASI

14.1. NoSQL ma'lumotlar bazasi haqida asosiy tushunchalar

Nosql (inglizcha **not only SQL** – faqat SQL emas) termini ma'lumotlar bazasini tashkil etishga qaratilgan bir qator yondoshuvlarni anglatadi va SQL tilidan foydalangan holda ma'lumotlarga kirish imkoniyatidan foydalanilgan an'anaviy relyatsion ma'lumotlar bazasi modelidan sezilarli farq qiladi. Bu miqyoslilik va qulaylik muammosini hal qilishda bo'linmaslik (atomar) va muvofiqlilik xususiyatini qo'llashga asoslangan ma'lumotlar bazasida ishlatiladi.

Bu g'oya o'z-o'zidan chiqqan yangi g'oya emas. Dastabki kompyuterlardayoq norelyatsion ma'lumotlar omborlaridan foydalanish boshlangan. Norelyatsion ma'lumotlar bazasi meynfreymlar davrida rivojlandi. 2000 yillarning boshida Google o'zining keng miqyosli va izchil izlash tizimini va ilovalarini yaratdi. Ular GMail, Google Maps, Google Earth tizimlari bo'lib, miqyoslilik va katta hajmdagi ma'lumotni parallel qayta ishlashga asoslangan.

Relyatsion ma'lumotlar bazasi tranzaksion tizimning ACID talablari asosida tashkil etilgan. Bu atama **atomicity**(atomar), **consistency**(izchillik), **isolation**(izolyatsiya), **durability**(ishonchlilik) so'zlaridan olingan. Shunga o'xshash NoSQL ma'lumotlar bazalari ham BASE talablariga tayanadi. Bu talablar quyidagilar:

- **basic availability** (bazaviy yaroqlilik)-har bir so'rov kafolatgan bo'lishi kerak.
- **soft state** (moslashuvchan holat) – ma'lumotlar mosligini ta'minlash uchun tizim holati vaqti-vaqti bilan ma'lumot kiritilmasa ham o'zgarib turishi mumkin.

- **eventual consistency** (oxir-oqibatda muvofiqlashish) - ma'lumotlar ma'lum bir vaqt uchun nomuvofiq bo'lishi mumkin, lekin bir muncha vaqt o'tgach muvofiqlashadi.

NoSQL qisqartmasi 1998-yilda yaratilgan. Ba'zilar NoSQLni SQLni siqib chiqarish uchun yaratilgan kamsitish atamasi deb hisoblashadi. Aslida, atama faqat SQL emas. Ushbu g'oya shundan iboratki, har ikkala texnologiya ham birlashishi mumkin va ularning har biri o'z o'rnini egallaydi. NoSQL-harakati so'nggi bir necha yil ichida yangiliklar bo'ldi, chunki Web 2.0 rahbarlarining ko'pchiligi NoSQL texnologiyasini qabul qildilar. Facebook, Twitter, Digg, Amazon, LinkedIn va Google kabi kompaniyalar NoSQL-dan foydalana boshlashdi.

14.2. NoSQL ma'lumotlar bazasining kelib chiqish sabablari

Relyatsion (SQL) bazalar insoniyatni qoniqtirmay qolganligini sababi nimada? Hozirgi vaqtda biz ma'lumotlar saqlanish texnologiyalarini 4 xil yo'nalishda rivojlanayotganligini ko'rishimiz mumkin.

Birinchi yo'nalish - saqlanuvchi ma'lumotlarni hajmini ortishi. Bugungi kunga kelib ma'lumotlar hajmi tasavvur qilish qiyin darajada kattalashib ketdi. Faqatgina 2009, 2010 yillarda bazalarda saqlangan ma'lumotlar undan avval butun insoniyat tarixidagi saqlangan ma'lumotlardan ham ortib ketdi.

Ikkinchi yo'nalish - ma'lumotlarni bir biri bilan bog'liqligi. Ma'lumotni har bir bo'lagini boshqa joylarda saqlanuvchi boshqa ma'lumotlar bilan qanday bo'lsa ham bog'liqligi bor. Internetdagi sahifalarni boshqa sahifalarga havolalari bo'ladi. Teglar har xil manbalardagi belgilangan ma'lumotlarni birlashtiradi.

Uchinchi yo'nalish - zaif tuzilgan ma'lumotlarni ishlatilishi. Bunga oddiy misol: do'konidagi molni ta'riflanishi. Agarda avval erkaklar ko'ylagini tarflashga 5-6 ta ustun (o'lchami, rangi,

matosi, rasmi) yetgan bo'lsa, hozirda bunday tariflar soni bir qancha o'nlab ustunlardan tashkil topgan bo'lishi mumkin. Bunda har xil ko'ylaklar uchun har xil "parametrlar" ishlatiladi. Bunday holatlarda molning xossalari haqidagi ma'lumot saqlanadigan jadval tuzilishini avvaldan aniqlash juda qiyin bo'ladi.

To'rtinchi yo'nalish - arxitektura. O'tgan asrning saksoninchi yillarida yagona katta kompyuter (mainframe) va yagona ma'lumotlar bazasi odatiy arxitektura hisoblanar edi. No'qsoninchi yillarda, klient-server arxitekturasi rivoj topdi. Yangi asrda esa, har biri o'z ma'lumotlar bazasiga ega bo'lgan web-xizmatlar va boshqa taqsimlangan yechimlar faol ravishda ishlatilmoqda. Bunday sharoitlarda esa relyatsion ma'lumotlar bazalarining unumdorligi keskin ravishda pasayib ketishi aniqlandi. Ko'pgina web saytlar uchun unumdorlik hozircha yetarli bo'lsa ham, sotsial tarmoqlar yoki izlash xizmatlari uchun SQL ma'lumotlar bazalaridan foydalanish asossiz ekanligi aniqlandi. Klassik relyatsion ma'lumotlar bazalari juda katta hajmdagi ma'lumotlar bilan yuqori "nagruzka" tartibida ishlashda bir qancha muammolarga duch keladi. Shuning uchun, ertami kechmi taqsimlangan yechimga o'tish ehtiyoji kelib chiqadi. Shunday ekan, yakka ma'lumotlar bazasini taqsimlangan tizimda izolyatsiyalashtirishga erishishni umuman ilojisi bo'lmaydi. CAP teoremasiga binoan taqsimlangan yechimning har qanday ko'rinishida 3 ta xossadan faqatgina 2 tasiga erishish mumkin: ma'lumotlarni muvofiqligi (**consistency**) - tizimning har bir tugunida vaqtni bir bo'lagida ma'lumotlar bir biri bilan hamohang bo'lishi, yetishuvchanlik (**availability**) - taqsimlangan tizimga berilgan har qanday so'rov to'g'ri javob bilan yakunlanadi. Taqsimlanishga turg'un (**partition tolerance**) - taqsimlangan tizimni bir qancha "izolyatsiyalangan" bo'limlarga bo'linishi har bir bo'limdan keluvchi javobni noto'g'ri bo'lishiga olib kelmaydi. NoSQL konsepsiyasi shulardan ikkitasini ya'ni,

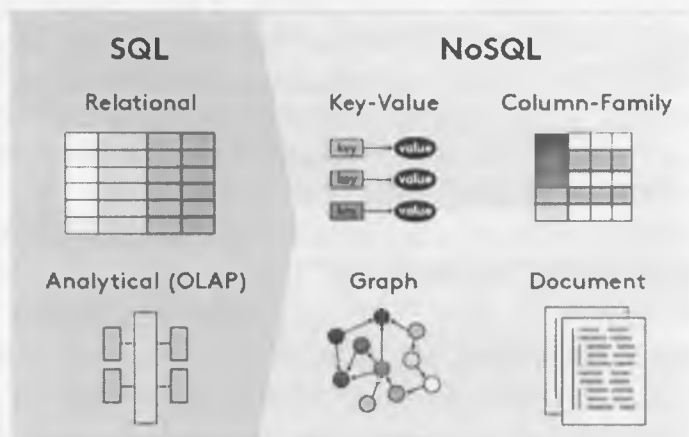
yuqori darajadagi "availability" va ma'lumotlarni taqsimlanishini ta'minlaydi. Bu yondashuv o'ziga munosib bo'lgan holatlar uchun maxsus yechim hisoblanadi [4].

14.3. NoSQL bazalarining toifalari

NoSQL bazalarining 4ta toifasi mavjud.

Birinchi toifa bu Key-Value (Kalit-Qiymat) turdagi bazalar. Bular o'zining tuzilishi jihatidan juda oddiy bazalar. Aslini olganda bular juda katta xesh-jadvallardir. Unda har bir kalitga alohida qiymat belgilangan bo'ladi. Bunday bazalar juda katta miqdordagi ma'lumotlar bilan ancha tez ishlashi mumkin, ammo ularni so'rovlar tilida kamchiligi mavjud.

Key-value bazalarga misol tarzida Dynomite, Voldemort, Tokyo va Redislarni keltirish mumkin.



14.1-rasm. NoSQL ma'lumotlar bazasi toifalari

Ikkinchi toifa BigTable klonlari. BigTable bu Google tarafidan o'z ehtiyojlari uchun ishlab chiqarilgan ma'lumotlar bazasi hisoblanadi. Bu baza juda katta uch o'lchamli baza hisoblanadi: ustunlar, qatorlar va vaqt belgilari. Bunday arxitektura ulkan unumdorlikga yetishish imkonini beradi,

undan tashqari u bir qancha kompyuterlarga joylashtirilishi mumkin. Ammo bu relyatsion baza emas, va unda relyatsion bazaning xususiyatlari yo'q.

BigTableda joinlar, qiyin so'rovlar yo'q. Google korxonasi BigTableni tarqatmaydi, shuning uchun ham bozorda bu bazani klonlari paydo bo'ldi. Bular: Hadoop, Hypertable va Cassandra.

Uchinchi toifa – bu hujjat (dokument) larga mo'ljallangan ma'lumotlar bazalari. Bunday bazalari bir-oz Key-Value bazalariga o'xshab ketadi, ammo bu holatda, ma'lumotlar bazasiuning qiymatlari nimaligi haqida xabardor bo'ladi. Odatda, qiymat sifatida biron bir xujjat yoki tuzilishiga so'rov berish mumkin bo'lgan ob'ekt bo'ladi. Bunday bazalarga misol qilib: CouchDB va MongoDBlarni keltirish mumkin.

To'rtinchi toifa -graflarga asoslangan ma'lumotlar bazalari. Bunday bazalar ob'ektlar orasidagi har xil qiyin bog'liqliklarni ta'minlashga qaratilgan bo'ladi hamda ularni asosida graflar nazariyasi yotadi. Bunday bazalar bir birlari bilan havolalar orqali bog'langan, tugunlar to'plami hisoblanadi. Bunda tugunlar va havolalar bir qancha xususiyat (atribut)larga ega bo'lishi mumkin. Misol tarzida: Neo4j, AllegroGraph, Sones graphDBlarni keltirish mumkin.

Bulardan tashqari yana beshinchi toifa ham mavjud, ammo u odatda NoSQL hisoblanmaydi. Gap ob'ektlarga asoslangan bazalar haqida ketayapti. Bunday bazalar eng avvalo OOP dasturlash "paradigma"sini ta'minlash uchun mo'ljallangan. Ularni OOP dasturlash tillaridan foydalanilganda ishlatish juda oson.

14.4. NoSQL ma'lumotlar bazasidagi ma'lumotlar bilan ishlash

NoSQL bazasidagi ma'lumotlar bilan ishlashni bir qancha "mexanizm"lari mavjud.

1. RESTful interfeyslar. Qisqacha qilib aytganda, bu

interfeys internetning asosiy protokoli bo'lmish - HTTP ga o'xshab ketadi. Bu protokolda ishlash mumkin bo'lgan har bir ob'ekt o'zinining qaytarimas manziliga ega bo'ladi. Ushbu manzilga murojaat qilib, biz shu ob'ekt haqidagi ma'lumotlarni olishimiz, tuzishimiz, o'zgartirishimiz va o'chirishimiz mumkin. Bunda serverda hech qanday holat saqlanmaydi, ya'ni har bir so'rov boshqa so'rovlardan mustaqil ravishda ishlaydi.

2. SQLdan farq qiluvchi so'rov tillari

- GQL - Google BigTablening SQL-ga o'xshash tili
- SPARQL - Semantik webning so'rovlar tili
- Gremlin - graflar bo'yicha o'tish tili
- Sones Graph Query Language - Sones Graph so'rovlar tili

Ma'lumotlar boshqaruvi qanday amalga oshiriladi?

Bu savolga ham javob, RESTful API yordamida ma'lumotlarni olish, o'zgartirish, tuzish va o'chirish mumkin. Yana bir yondashuvda esa ma'lumotlar boshqaruvi uchun maxsus API ishlatilishi ko'zda tutiladi. Masalan, Google BigTable da, DataStore API, Neo4j da esa - graphDatabase API ishlatiladi. Bundan tashqari, NoSQL bazalarida ma'lumotlarni JSON (javascript object notation) yoki Trift formatlariga serializatsiya qilinishi ham mumkin.

14.5. NoSQLning afzalliklari

NoSQL ning eng asosiy afzalligi bu unumdorlik hisoblanadi. Barcha NoSQL bazalar o'z yo'nalishi bo'yicha relyatsion bazalardan afzal hisoblanadi. Agarda yaqin o'tmishda har qanday holatlar uchun ham faqatgina relyatsion bazalar ko'zda tutilgan bo'lsa, bugunga kelib ahvol o'zgardi. Har bir konkret holat uchun, o'ziga yarasha ma'lumotlar saqlanishini tanlash lozim.

Ba'zi holatlarda bir vaqtni o'zida bir qancha bazalar bo'lishi kerakligi ehtiyoji kelib chiqadi. Masalan, web-dasturlarda

MongoDB ma’umotlar saqlanishi uchun asosiy baza bo’lib xizmat qilsa, Redis yordamida foydalanuvchining so’rovlarini keshlash amalga oshiriladi. Natijada biz juda yuqori unumdorlikga va dasturchilar uchun ham qulay bo’lgan interfeyslarga ega bo’lgan tizimga erishamiz. NoSQL bazalarining yana bir afzalligi shundaki, ularing juda ko’p turlari Open Source loyihalar hisoblanadi. Bu esa, dasturlash tarafidan juda qulay.

Example NoSQL databases



14.2-rasm. NoSQL ma’lumotlar bazasini boshqarish tizimlari

Monopoliyaning tugashi

Hozirgi vaqtga kelib ma’lumotlarni saqlashda relyatsion bazalarning so’zsiz peshqadamligi yakuniga etdi. Bugunda dasturchi boshqacha yondashuvlardan ham xabardor bo’lishi kerak va u ishlayotgan vazifaga eng mos keluvchi yechimni to’g’ri tanlashni ham bilishi kerak. Albatta, har qanday yechim uchun ham faqatgina NoSQLdan foydalanish kerak degan fikr mutlaqo noto’g’ri. Kichik va o’rta tizimlar uchun relyatsion ma’lumotlar bazasi NoSQL yechimdan ko’ra afzalroq bo’lishi mumkin. Ammo katta ma’lumotlar va taqsimlangan serverlar ishlatilgan holatda NoSQL relyatsion ma’lumotlar bazalariga qaraganda ancha afzalliklarga ega. NoSQL ma’lumotlar bazasi dunyosi juda qiziq, bu erda ko’pgina original va innovatsion yechimlarni topish mumkin [4].

Mustaqil bajarish uchun savol va masalalar

1. Sxemasi quyida ko'rsatilgan jadval uchun imtihonlardan 3 bahosi bo'lgan talabalarni chiqarish so'rovini tuzing.

Jadval sxemasi: **Student**(id_s, FIO, tugilgan_sana, guruh);
Baholash(id_b, id_s, Fan, baho).

2. SQL tilida TABL1 nomli MB jadvalini yaratish buyrug'ini yozing. Bu jadval quyidagicha ustun xarakteristikalariga ega bo'lishi kerak.

A — butun; B — simvolli (25 simvol); C — butun (Tabl2 jadvalidagi D maydon bilan bog'langan);

Cheklov: A *ustun* — birlamchi kalit.

3. Sxemasi quyida ko'rsatilgan jadval uchun imtihonlardan faqat 4 va 5 bahosi bo'lgan talabalarni chiqarish so'rovini tuzing.

Jadval sxemasi: **Student**(id_s, FIO, tugilgan_sana, grupp);
Baholash(id_b, id_s, Fan, baho).

4. So'rovda chiqarilayotgan matn qanday funksiya yordamida bosh harflarga yoki mayda harflarga aylantiriladi?

5. Sxemasi quyida ko'rsatilgan jadval uchun imtihonlardan faqat 5 bahosi bo'lgan talabalarni chiqarish so'rovini tuzing.

Jadval sxemasi: **Student**(id_s, FIO, tugilgan_sana, grupp);
Baholash(id_b, id_s, Fan, baho).

6. SQL tilida TABL1 jadvalga xarakteristikalari quyida ko'rsatilgan ustunni qo'shish buyrug'ini yozing. Ustunning nomi — D; ma'lumot tipi - satrli, 10 simvol.

7. Sxemasi quyida ko'rsatilgan jadval uchun imtihonlardan 2 va undan ortiq fanlardan qoniqarsiz baho olgan talabalarni chiqarish so'rovini tuzing.

Jadval sxemasi: **Student**(id_s, FIO, tugilgan_sana, grupp);

Baholash(id_b, id_s, Fan, baho).

8. Quyidagi so'rovni izohlab bering:

SELECT * FROM EXAM_MARKS WHERE MARK NOT
IN (4, 5);

9. MySQL MBBT uchun takrorlanuvchi strukturaga ega bo'lgan saqlanuvchi procedura tuzing.

10. Sxemasi quyida ko'rsatilgan jadval uchun imtihonlardan faqat 5 bahosi bo'lgan 4-15 TJBAKT guruhi talabalarini chiqarish so'rovini tuzing.

Jadval sxemasi: **Student**(id_s, FIO, tugilgan_sana, grupp);
Baholash(id_b, id_s, fan, baho, sana).

11. Quyidagi sxemadan 2018 yil 12 va 14 dekabr kunlari imtihon olingan fanlar ro'yxatini chiqaruvchi so'rov yarating.

Student(id_s, FIO, tugilgan_sana, grupp);
Baholash(id_b, id_s, id_fan, baho, sana);
Fanlar(id_fan, fan, soat).

12. Sxemasi quyida ko'rsatilgan jadval uchun imtihonlardan faqat 5 bahosi bo'lgan 4-15 TJBAKT guruhi talabalarini chiqarish so'rovini tuzing.

Jadval sxemasi: **Student**(id_s, FIO, tugilgan_sana, grupp);
Baholash(id_b, id_s, fan, baho, sana).

13. Sxemasi quyida ko'rsatilgan jadval uchun fizika fani imtihonidan faqat 4 va 5 bahosi bo'lgan 2-15 TJBAKT guruhi talabalarini aniqlash so'rovini tuzing.

Jadval sxemasi: **Student**(id_s, FIO, tugilgan_sana, grupp);
Baholash(id_b, id_s, fan, baho, sana).

14. Fizika fanidan 700-17 TJB talabalarining olgan o'rtacha balini aniqlash so'rovini tuzing.

15. Quyidagi so'rovning ishlashini tushuntirib bering
SELECT * FROM Fan WHERE fan_kodi
NOT IN (SELECT fan_kodi FROM baholash).

16. SQL tilida TABL1 nomli MB jadvalini yaratish so'rovini yozing. Bu jadval quyidagicha ustun xarakteristikalariga ega bo'lishi kerak: A — butun; B — simvolli (50 simvol) bo'sh emas; C

— butun (Tabl2 jadvalidagi E maydon bilan bog'langan); D-sana;
Cheklov: *A ustun* — birlamchi kalit.

17. Quyidagi so'rovning ishlashini tushuntirib bering

SELECT * FROM Student WHERE [№ reyting daftari] IN

(SELECT [№ reyting daftari] FROM Baholash WHERE
natija>=4)

18. MySQLda k=0 bo'lganda talabalar ro'yxatini ko'rsatilgan
guruh bo'yicha va k=1 bo'lganda ko'rsatilgan manzil bo'yicha
chiqaradigan saqlanuvchi procedura tuzing.

19. Jadvalga yangi yozuv kiritilganda kalit maydon
qiymatini hosil qiladigan trigger tuzing.

20. MySQL MBBT da magazindan 10000 so'mdan 100000
so'mgacha tovar xarid qilinganda 10% skidka va 100000 so'mdan
ko'p xarid qilinganda 20% skidka qilishni hisoblovchi
saqlanuvchi procedura tuzing.

21. Sxemasi quyida keltirilgan jadval uchun avgust 1996
yilda tug'ilgan 704-17 TJB guruhi talabalarini tanlash SQL
buyrug'ini yozing. **STUDENT**(FIO, Tug'ilgan sanasi, № grupp).
Sanani "01-dek-2018" ko'rinishda chiqarilsin.

22. MySQL MBBT uchun takrorlanuvchi strukturaga ega
bo'lgan saqlanuvchi procedura tuzing

ADABIYOTLAR

1. M.X.Hakimov, S.M. Gaynazarov. Berilganlar bazasini boshqarish tizimlari. Oliy o'quv yurtlari uchun darslik. Toshkent: "Fan va texnologiya" nashriyoti, 2015 yil, 648 bet.

2. Хомоненко А.Д. Базы данных. Учебник для высших учебных заведений / Под ред. проф. А. Д. Хомоненко. — 6-е изд., доп. - СПб.: КОРОНА-Век, 2009. - 736 с.

3. Н.П. Стружкин, В.В. Годин. Базы данных. Проектирование. Учебник. М.: Юрайт, 2016. - 478 с.

4. С.А.Мартишин, В.Л.Симонов, М.В. Храпченко. Базы данных. Практическое примечание СУБД SQL и NoSQL. Учебное пособие - М.: Форум, Инфра-М, 2016. - 368 с.

5. Э.В. Фуфаев, Д.Э. Фуфаев. Базы данных. Учебное пособие- М.: Академия, 2016. - 320 с.

6. Документация к PostgreSQL 9.5.15 The PostgreSQL Global Development Group Перевод на русский язык, 2015-2018 гг.: Компания «Постгрес Профессиональный».-2335 с.

7. П. Лузанов, Е. Рогов, И. Лёвшин. PostgreSQL для начинающих. 117036, г. Москва, ул. Дмитрия Ульянова, д. 7А. 2017 г.

8. Ржеуцкая С.Ю. Базы данных. Язык SQL. Учебное пособие. Вологда: ВоГТУ, 2010. 159 с.

9. Дейт К. Дж. SQL и реляционная теория. Как грамотно писать код на SQL. – Пер. с англ. –СПб.: Символ-Плюс, 2010.

10. Кузнецов С.Д. Основы баз данных. Учебное пособие. М.: Интернет-Университет Информационных Технологий, 2017г. - 488 с.

11. Кузнецов М.В. MySQL-5.-СПб.: БХВ Петербург, 2010.-1024 с.

12. П.В. Бураков, В.Ю. Петров. Введение в системы баз данных. Учебное пособие. СПб.-2010 г.

13. О.Л. Голицына, Н.В. Максимов, И.И. Попов. Базы данных. М.: Форум-Инфра-М, 2007г.

14. Жилинский А. MS SQL сервер 2008. СПб.: БХВ-Петербург, 2009 г.

15. Джеймс Перри, Джеральд Пост. "Введение в Oracle 10g". Пер. с англ.- М.: ООО «И.Д. Вильямс», 2006.-704 с.

16. Виктор Гольцман. MySQL-5.0, Санкт-Петербург; 2010.

17. Гурвиц Г.А. Microsoft® Access 2010. Разработка приложений на реальном примере. — СПб.: БХВ-Петербург, 2010.—496 с

18. Ицик Бен Ган. Microsoft SQL Server-2008. Основы T-SQL. СПб.: «БХВ Петербург», 2009

19. Зрюмов, Е. А. Базы данных для инженеров [Текст] : учебное пособие. Алт. гос. техн. ун-т им. И. И. Ползунова, – Барнаул : Изд-во АлтГТУ, 2010.–131 с.

20. Виейра, Роберт. Программирование баз данных Microsoft SQL server-2005 для профессионалов. Пер. с англ. М.: - ООО «И.Д. Вильямс», 2008. -1072с.

21. Рибанов А.А. Инструментальные средства автоматизированного проектирования баз данных: Учебное пособие и варианты заданий к лабораторным работам по дисциплине «Базы данных» / ВолгГТУ, Волгоград, 2007.

22. Steve Fogel .Oracle® Database Administrator's Guide 11g Release 2 (11.2) E25494-07. Copyright 2015, Oracle and/or its affiliates.

23. A. Sattorov. Ma'lumotlar bazasini boshqarish sistemasi (ACCES WINDOWS – 9X/2000 uchun). T., «Fan va texnologiya», 2006, 304-bet

24. Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer Widom. DATABASE SYSTEMS. The Complete Book. Second Edition. Department of Computer Science Stanford University. © 2009 by Pearson Education Inc. Pearson Prentice Hall. Pearson Education, Inc. Upper Saddle River, NJ 07458.

25. Jefrey A. Hoffer, Mary B. Prescott, Fred R. McFadden. Modern database management. Copyright 2016 by Pearson Education, Inc. Upper Saddle River, New Jersey 07458.

26. Д. Крёнке. Теория и практика построения баз данных. 8-е издание. Москва, Санкт-Петербург, Киев, Харьков, Минск, 2003 г.

B.T. Ergashev, K.Z. Abidov

MA'LUMOTLAR BAZASINI BOSHQARISH TIZIMLARI

Muharrir:

Texnik muharir:

Musahhih:

Sahifalochi:

G'.Murodov

G.Samiyeva

M.Raximov

M.Arslonov



Nashriyot litsenziyasi AI № 178. 08.12.2010. Original –
maketdan bosishga ruxsat etildi: 05.07.2021. Bichimi 60x84.
Kegli 16 shponli. « Palatino Linotype» garn. Ofset bosma
usulida. Ofset bosma qog'ozi. Bosma tabog'i 20 Adadi 100.
Buyurtma № 85.



«Sharq-Buxoro» MCHJ bosmaxonasida chop etildi.
Buxoro shahar O'zbekiton Mustaqilligi ko'chasi, 70/2 uy.
Tel: 0(365) 222-46-46



ISBN978-9943-7368-3-2



9 789943 736832