

**O'ZBEKISTON RESPUBLIKASI ALOQA, AXBOROTLASHTIRISH  
VA TELEKOMMUNIKATSIYA TEXNOLOGIYALARI DAVLAT  
QO'MITASI**

**TOSHKENT AXBOROT TEXNOLOGIYALARI UNIVERSITETI  
SAMARQAND FILIALI**

«Telekommunikasiya texnologiyalari va kasb ta'limi» fakulteti

«Axborot ta'lim texnologiyalari» kafedrası

**Umarov A.A.**

**SQL texnologiyalar**  
fanidan

**Amaliyot mashg'ulotlarini o'tish uchun  
USLUBIY KO'RSATMA**

SAMARQAND -2015

Umarov A.A. SQL texnologiyalari fanidan amaliyot mashg'ulotlarini ot'kazish uchun uslubiy ko'rsatma - Samarqand, TATU SF, 2015. – 57 bet.  
5521900 – «Informatika va axborot texnologiyalari», 5522200 – “Telekommunikatsiya”,  
5140900 – «Kasbiy ta'lim» (Informatika va axborot texnologiyasi), 5811200 - «Axborot servisi" ta'lim yo'nalishlarining 3-bosqich bakalavriat talabalari uchun.

Taqrizchilar: Djumanov O.I. – SamDU, dotsent.  
Xo'jayorov I.SH. – TATU SF, katta o'qituvchisi

Uslubiy ko'rsatma TATU Samarqand filiali o'quv-uslubiy kengashining 2015 yil \_\_\_ - \_\_\_\_\_ № \_\_\_ qarori bilan nashrga tavsiya qilingan.

Ilmiy-uslubiy kengash raisi: \_\_\_\_\_

## MUNDARIJA

Kirish .....	4
1-Mavzu. <b>SQL tilida ma'lumotlar bazasining relyatsion modelini yaratish</b> ....	6
1. Ma'lumotlar bazasi to'g'risida umumiy ma'lumotlar.	
2. SQL tilida relyatsion modelini yaratish.	
2 -Mavzu. <b>SQL tilida so'rovlarni loyihalashtirish</b> .....	12
1. So'rovlar to'g'risida umumiy tushunchalar.	
2. SQL tilida so'rovlarni loyihalashtirish.	
3-Mavzu. <b>SQL tilida ma'lumotlar bazasini boshqarish</b> .....	18
1. Ma'lumotlar bazasini boshqarish to'g'risida umumiy tushunchalar.	
2. SQL tili va ma'lumotlar tiplari.	
4-Mavzu. <b>SQL tilida ma'lumotlar bazasi funksiyalari bilan ishlash</b> .....	23
1. Ma'lumotlar bazasi funksiyalari to'g'risida umumiy tushunchalar.	
2. SQL tilida MB funksiyalari bilan ishlash.	
5-Mavzu. <b>SQL tilida protseduralar va ulardan foydalanish</b> .....	28
1. Protseuralar to'g'risida umumiy tushunchalar.	
2. SQL tilida protseduralar va ulardan foydalanish.	
6-Mavzu. <b>SQL tilida jadvallarni loyihalashtirish. Jadvallarni yaratish, o'zgartirish va o'chirish. Jadvallar haqida ma'lumot olish</b> .....	33
1. Jadvallarni loyihalashtirish.	
2. SQL tilida jadvallarni indekslar o'rnatish orqali yaratish.	
7-Mavzu. <b>SQL tilida ma'lumotlarni boshqarish. Bazadan ma'lumotlarni olish. Ma'lumotlarni qo'shish, o'zgartirish va o'chirish</b> .....	40
1. SQL tili yordamida ma'lumotlarni boshqarish.	
2. Murakkab so'rovlar yordamida MB dan ma'lumotlarni olish.	
8-Mavzu. <b>SQL tilida tranzaksiyalar va blokirovkalar. Tranzaksiya va blokirovka tushunchasi. Aniq joylashtirilgan tranzaksiyalar</b> .....	44
1. SQL muhitida tranzaksiya tushunchasi.	
2. SQL muhitida tranzaksiyalarni boshqarish.	
Nazorat savollari .....	49
Adabiyotlar .....	50
SQL texnologiyalarni fanini o'rganish uchun zaruriy terminlar va ularning izohlari .....	51

## KIRISH

Hozirgi kunda inson faoliyatida ma'lumotlar bazasi (MB) kerakli axborotlarni to'plash, saqlash va qayta ishlashda muhim o'rin tutmoqda. Chunki jamiyat ishlab chiqarishining qaysi jabhasiga nazar solmaylik o'zimizga kerakli ma'lumotlarni olish uchun, albatta, MBga murojaat qilishga majbur bo'layapmiz. Shuning uchun ham MB bilan muloqot qilish axborot almashinuv texnologiyasining eng dolzarb muammolaridan biriga aylanib borishi hozirgi davr taqozasidir.

Axborotli texnologiyalarning taraqqiyoti va axborot oqimlarining tobora ortib borishi insoniyatni ma'lumotlarni o'z vaqtida qayta ishlash choralarining yangi uslublarini qidirib topishga majbur qilmoqda. Ma'lumotlarni saqlash, uzatish va qayta ishlash uchun MBni yaratish, so'ngra undan oqilona foydalanish bugungi kunning dolzarb masalalaridan biriga aylandi.

Shuni ham e'tiborga olish kerakki, ma'lumotlar bazasi dasturiy mahsulotlari rivojlanishi vaqt o'tishi bilan tobora standartlashib bormoqda. Bu jarayon har xil kompyuter muhitlarida faoliyat ko'rsatuvchi axborot tizimlarini yaratishda yagona standart til yaratish zaruriyatini keltirib chiqardi. Standart til bir buyruqlar to'plamini bilgan foydalanuvchilarga ularni shaxsiy kompyuter, tarmoq avtomatlashtirilgan ishchi joyi (AIJ) yoki katta EHMda ishlashlaridan qat'iy nazar ma'lumotlarni yig'ish, saqlash, izlash va uzatishga imkon beradi. Ana shunday standart tillardan biri SQL tilidir.

SQL (Structured Query Language, odatda "sikvel" deb o'qiladi) va ma'nosi "Tarkiblangan so'rovlar tili". Bu til relyatsion ma'lumotlar bazalari bilan ishlashga imkon beradigan tildir. SQL tilining xususiyati shundan iboratki u ma'lumotlarni qayta ishlash protseduralariga emas natijalariga yo'naltirilgan tildir. SQL tilining o'zi ma'lumotlar qayerda joylashgani, qanday indekslarni qo'llash kerakligini aniqlaydi, ya'ni bu komponentalarni ma'lumotlar bazasi so'rovlarida ko'rsatish shart emas.

SQL tilining o'zi dastlab IBM kompaniyasining DB2 MBBTni yaratish jarayonida ishlab chiqilgan va keng ko'lamda RISC protsessorli kompyuterlarda UNIX operatsion tizimi muhitida qo'llangan.

Shu bilan birga mustaqil bo'lmasdan PL/SQL va Transact-SQL kabi ichki dasturlash tillari bilan birgalikda ishlatilgan. 1986 yilda ANSI (American National Standart Institute) SQL tilining mustaqil rasmiy standartini ishlab chiqdi va 1992 yil bu standart kengaytirildi. SQL tili 30 ga yaqin operatorlarga ega bo'lib, rusumlariga qarab ba'zi hollarda ko'proq, ba'zilarida esa kamroq bo'lishi mumkin.

SQL tili "Mijoz-Server" texnologiyasiga asosida ishlab, foydalanuvchining (Mijozning) so'rovlari maxsus ma'lumotlar serverlarida (Server) qayta ishlanadi va foydalanuvchi komputerlariga faqat so'rov natijalari qaytariladi.

Relyatsion modellarda Server bilan muloqot qilish uchun standart sifatida SQL tili tanlangan. Shuning uchun hamma zamonaviy relyatsion MBBTda (DB2, Oracle, Ingres, Informix, Sybase, Progress, DB2) va hattoki norelyatsion MBBT (masalan, Adabas)da "Klient\_Server" texnologiyasi va SQL tilidan foydalanadilar.

Hozirgi vaqtda ikki turdagi SQL mavjud: Interaktiv va Joylashtirilgan. Ko'p hollarda ikkala tur bir xil ishlaydi, lekin ikki xil foydalaniladi.

**Interaktiv SQL** ma'lumotlar bazasining o'zida faoliyat ko'rsatadi va foydalanuvchi ilovalarini bajarish uchun ishlatiladi. SQLning bu turida buyruqni kiritish zahotiy oq buyruq bajariladi va natija ekranda paydo bo'ladi.

**Joylashtirilgan SQL** esa boshqa tilda yaratilgan dastur ichiga joylashtirilgan SQL buyruqlaridan iborat.

SQL tili quyidagi tarkibiy qismlardan iborat:

- **DDL** ( Ma'lumotlarni Ta'riflash Tili ) - ANSI da ma'lumotlar sxemasini ta'riflash tili, ob'ektlarni (jadvallar, indekslar, tasavirlar va hokazo) yaratuvchi buyruqlar to'plamidan iborat.

- **DML** (Ma'lumotlarni O'zgartirish Tili) - bu ixtiyoriy daqiqada jadvallarda qanday qiymatlar saqlanishini aniqlovchi buyruqlar majmuasidir.

- **DCL** (Ma'lumotlarni Boshqarish Tili) foydalanuvchiga aniq ob'ektlarga ta'sir o'tkazishga ruxsat berish yoki bermaslikni aniqlovchi buyruqlardan iborat.

SQL tilini dasturlash uchun yaratilgan algoritmik til deb hisoblab bo'lmaydi. U eng avvalo, mantiqiy-axborotlar tilidir. Uning yordamida relyatsion MBda saqlanadigan ma'lumotlarni tavsiflash, izlash, o'zgartirish va olish mumkin.

Uslubiy ko'rsatmai Toshkent Axborot Texnologiyalari Universiteti qoshidagi Oliy o'quv yurtlararo ilmiy uslubiy kengashida ko'rib chiqilgan va tavsiya qilingan dastur asosida ishlab chiqildi.

# 1-Mavzu: SQL TILIDA MA'LUMOTLAR BAZASINING RELYATSION MODELINI YARATISH.

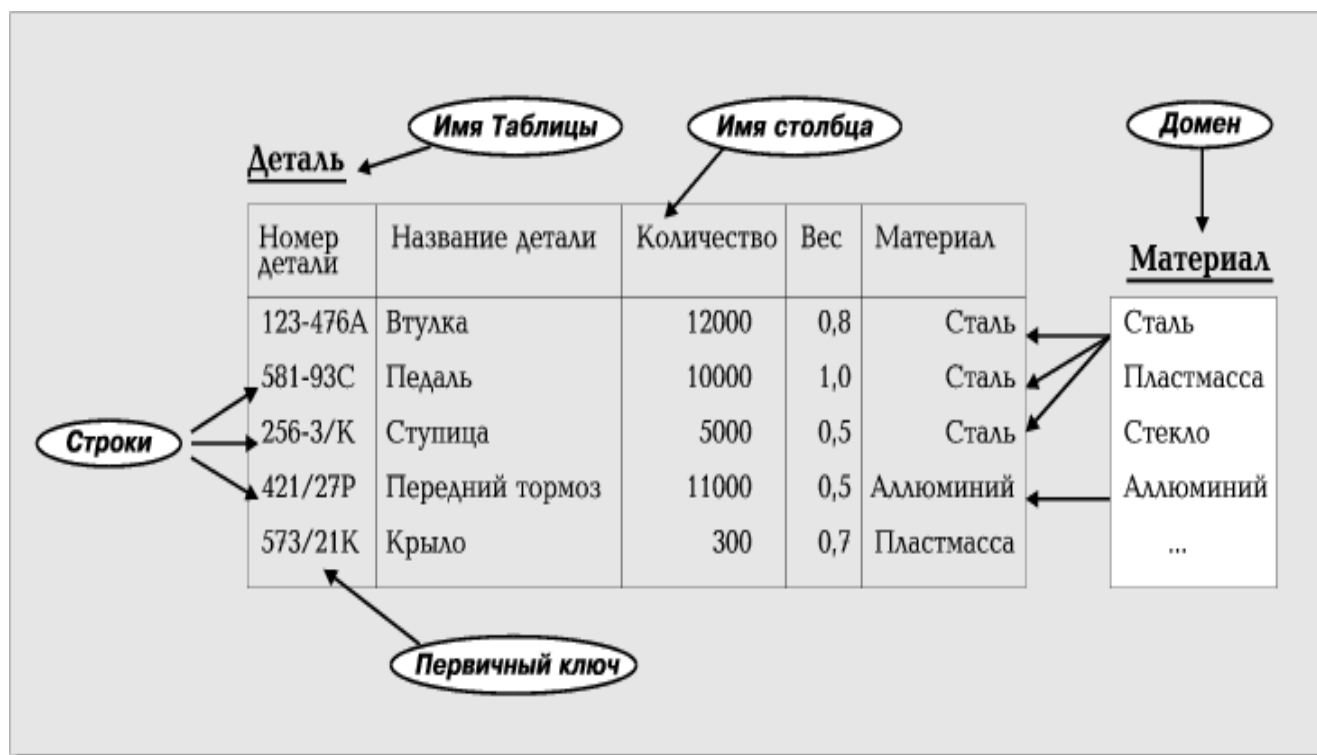
1. Ma'lumotlar bazasi to'g'risida umumiy ma'lumotlar.
2. SQL tilida relyatsion modelini yaratish.

## 1. Ma'lumotlar bazasi to'g'risida umumiy ma'lumotlar.

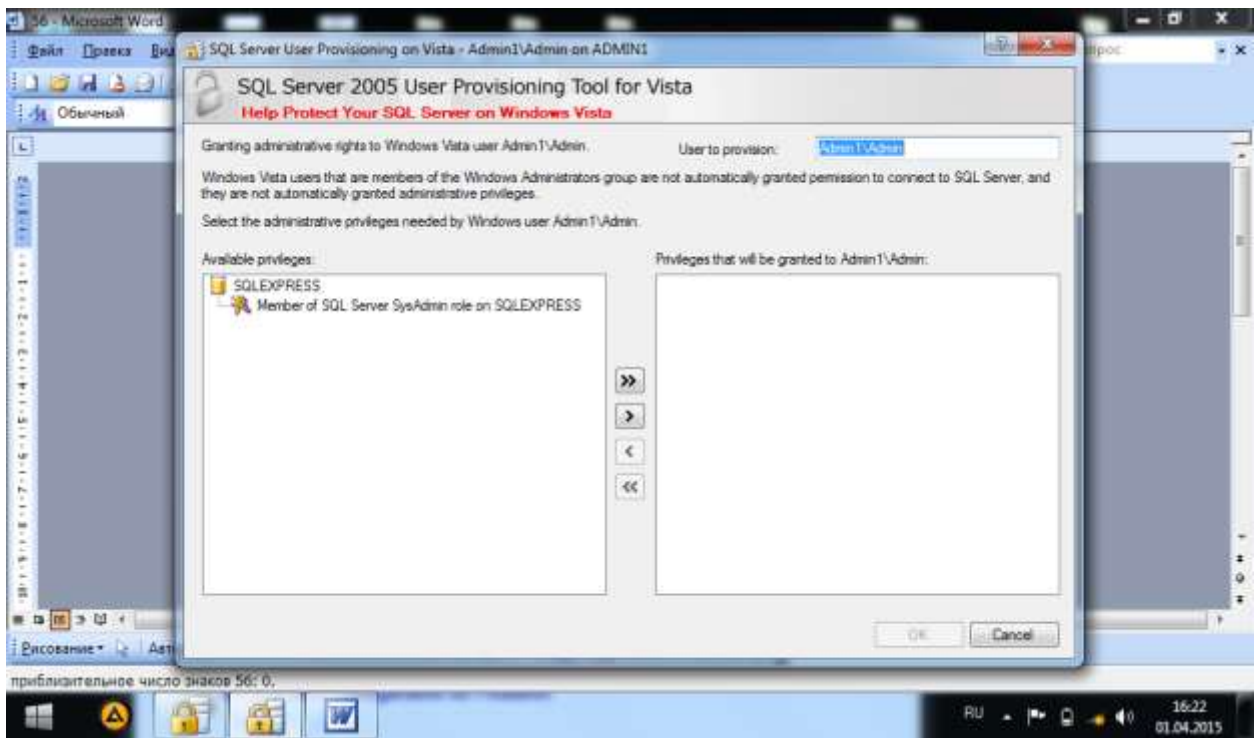
MB tushunchasi fanga kirib kelgunga qadar, dasturchilar ma'lumotlarni shunday tashkil qilar edilarki, u faqat qaralayotgan masala uchungina o'rinli bo'lardi. Keyingi yangi masala uchun ma'lumotlar qayta tashkil qilinadi va bu hol ish unumdorligining sifatsiz bo'lishiga olib kelar edi. Shuning uchun ham ma'lumotlarni qayta ishlashning unumdor usullarini ishlab chiqish zarurati paydo bo'ldi.

Umuman olganda har qanday axborot tizimining maqsadi real muhit ob'ektlari haqidagi ma'lumotlarga ishlov berishdan iborat. Keng ma'noda ma'lumotlar bazasi - bu aniq predmet sohaga talluqli bo'lgan real muhitning ob'ektlari haqidagi ma'lumotlar to'plamidir. Predmet soha deganda, ma'lumotlar bazasini tashkil qilish uchun o'rganilayotgan real muhitning ma'lum bir qismi tushuniladi. Masalan, tashkilot, zavod, kollej, oliy o'quv yurti va boshqalar.

MBni tashkil etishda ma'lum qoida va qonunlarga amal qilish lozim bo'ladi. Bundan buyon axborot so'zini ma'lumot so'zidan farqlaymiz, ya'ni, axborot so'zini umumiy tushuncha sifatida qabul qilib, ma'lumot deganda aniq bir belgilangan narsa yoki hodisaning sifatlarini nazarda tutamiz.



1- rasm. Ma'lumotlar bazasi tuzilmasi.



2-rasm. SQL tilida ro'yxatdan o'tish

Munosabat ustunlari atributlar deb ataladi va ularga nomlar beriladi. Munosabat atributlarining nomlardan iborat ro'yxatiga munosabat sxemasi deyiladi. Masalan, TALABA jadvalining munosabat sxemasini quyidagicha yozish mumkin: TALABA(DTM, FISH, JINSI, TYIL, MANZILI, TEL).

Ma'lumotlarning relyatsion modeli ikki o'lchamli jadvallarda saqlanadigan ma'lumotlarning o'zaro bog'iq majmuasidir. Relyatsion model kuchli nazariy fundamentga ega bo'lib, u matematik munosabatlar nazariyasiga asoslangan. Ma'lumotlarning relyatsion modeli konseptsiyasi 1970 yilda E. Kodd tomonidan taklif qilingan bo'lib, u ma'lumotlarni tavsiflash va tasvirlashning amaliy dasturlarga bog'liq bo'lmasligini ta'minlaydi.

## 2. SQL tilida relyatsion modelini yaratish.

**Relatsion (jadval) ko'rinishidagi ma'lumotlar bazasi:** Relatsion yoki jadval ko'rinishidagi ma'lumotlar bazasi hayotda keng qo'llaniladigan ma'lumotlar bazasi bo'lib, Access ham relatsion ma'lumotlar bazasi bilan ishlaydi. Relatsion ma'lumotlar bazasi- bu shunday ma'lumotlar bazasiki, unda ma'lumotlar ikki o'lchovli jadvallarda saqlanadi va ular o'rtasida o'rnatilgan bog'lanish qayta ishlash samaradorligini oshiradi hamda bir xil ma'lumotlarning takrorlanishini kamaytiradi.

### Relatsion ma'lumotlar bazasining ko'rinishi.

		USTUNLAR		
		1-maydon	2-maydon	3-maydon
S A T R L A R	1-yozuv			
	2-yozuv			
	3-yozuv			
	4-yozuv			
	5-yozuv			

3-rasm. Relatsion ma'lumotlar bazasi

**Maydon-** ma'lumotlar bazasida saqlanadigan obyektning bitta parametri (xarakteriskasini) kiritish uchun mo'ljallangan jadvalning elementi (ustuni).

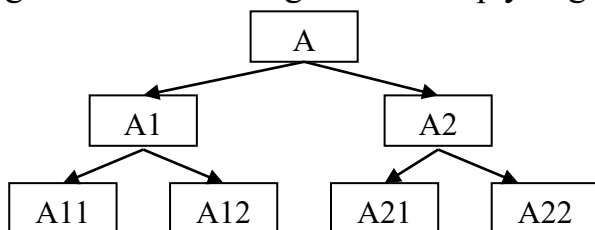
MBni yaratish jarayonida ma'lumotlarni bevosita kompyuterning tashqi qurilmalarida tashkil etish lozim boladi. Ma'lumotlar modeli – bu ma'lumotlarning o'zaro bog'langan tuzilishlari va ular ustida bajariladigan amallar to'plamidir. Ma'lumotlar modeli quyidagi tarkibiy qismlardan iborat:

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

2. Ma'lumotlar tuzilishida bajarilishi mumkin bo'lgan amallar. Ma'lumotlar modeli ma'lumotlarni aniqlash tili (ЯОД) va ma'lumotlar bilan amallar bajarish tili (ЯМД) bilan ishlash imkoniyatiga ega bo'lishi lozim.

3. Ma'lumotlar modeli MBning yaxlitligini saqlash va himoya qilish vositalari bilan ta'milangan bo'lishi kerak.

Ierarxik modelda kalit bo'yicha kirish odatda, faqat boshqa ob'ektlarga tobe bo'lmagan eng yuqori pog'onadagi ob'yektgagina taalluqli bo'lishi mumkin. Boshqa ob'ektlarga kirish modelning eng yuqori pog'onasidagi ob'yektdan aloqalar bo'yicha amalga oshiriladi. Uning ko'rinishi quyidagicha:



*4-rasm. Ma'lumotlarning ierarxik modeli ko'rinishi.*

Masalan, 3-rasmdagi A12 tugunidan ma'lumotni olish uchun oldin A tugun, keyin A1 tugun va undan keyin A12 tugun topiladi.

Tarmoqli modellarda esa kalit bo'uyicha bevosita ixtiyoriy ob'yektga kirish (uning modelda qaysi pog'onada joylashishidan qat'iy nazar) ta'minlanishi mumkin. Tarmoqli modellarda ob'ektlar tuzilmasi ko'pincha chiziqli va kamroq hollarda ierarxik bo'ladi. Chiziqli tuzilmaga ega bo'lgan ob'yekt faqat oddiy va kalit atributlardan iboratdir. Tarmoqli model ko'rinishi quyidagicha.

Ma'lumotlarning relyatsion modeli asosida "munosabat" tushunchasi yotib, u unglizcha "relatsion" so'zidan olingan. Quyidagi qoidalarga amal qilgan holda, munosabatlarni ikki o'lchovli jadval ko'rinishda tasvirlash mumkin:

- 1) ma'lumotlar jadval yacheykalrida tuzilishi jihatidan bo'linmas bo'lishi kerak;
- 2) bitta ustundagi ma'lumotlar bir xil turga ega bo'lishi lozim;
- 3) har bir ustun ynikal nomga ega bo'lishi kerak (ustunlarning takrorlanishiga yo'l qo'yilmaydi);
- 4) ustunlar jadvalda ma'lum tartibda joylashishi lozim;
- 5) jadval qatorlari ixtiyoriy tartibda joylashishi mumkin;



Yozuv,  
Segment,  
Kortej

Birlamchi kalit		Domen			
DTM	FISH	JINSI	TYIL	MANZILI	TEL
0573051	Amriev Bahodir	E	101284	Navoiy	2401813
0573092	Sultonova Guzal	A	120185	Samarqand	2331224
0573186	<b>Davriyev Kamol</b>	<b>E</b>	<b>010984</b>	<b>Farg'ona</b>	<b>2101095</b>
0673219	Soriev Vaxob	E	260584	Toshkent	3864415
0573321	Xamzaev Sharof	E	210780	Buxoro	1201680

Ob'ekt

5-rasm. Ma'lumotlarning relyatsion modeli ko'rinishi

Har qanday munosabat (jadval) kompyuter xotirasida fayl ko'rinishida yoziladi. Ularning orasida quyidagi moslik mavjud:

Fayl	Jadval	Munosabat	Mazmuni
Yozuv Maydon	Satr Utun	Kortej Atribut	Mazmun nusxasi Atribut

### Relatsion ma'lumotlar bazasi asosida jadvallar ko'rinishda MB hosil qilish.

Quyidagi ko'rinishdagi jadvallar berilgan bo'lsin:

Sotuvchilar (Salespeople) jadvali:

SNUM	FNAME	SNAME	CITY	COMM
1001	Sobirov	Raxim	Navoiy	12%
1002	Qosimov	Sodiq	Samarqand	11%
1004	Muammadiev	Lapas	Farg'ona	11%
1007	Razzoqov	Asam	Samarqand	15%
1003	Asqarov	Nodir	Namangan	10%

Bu erda: Snum – har bir sotuvchi unikal raqami (kodi),

Fname – sotuvchi sarifi,

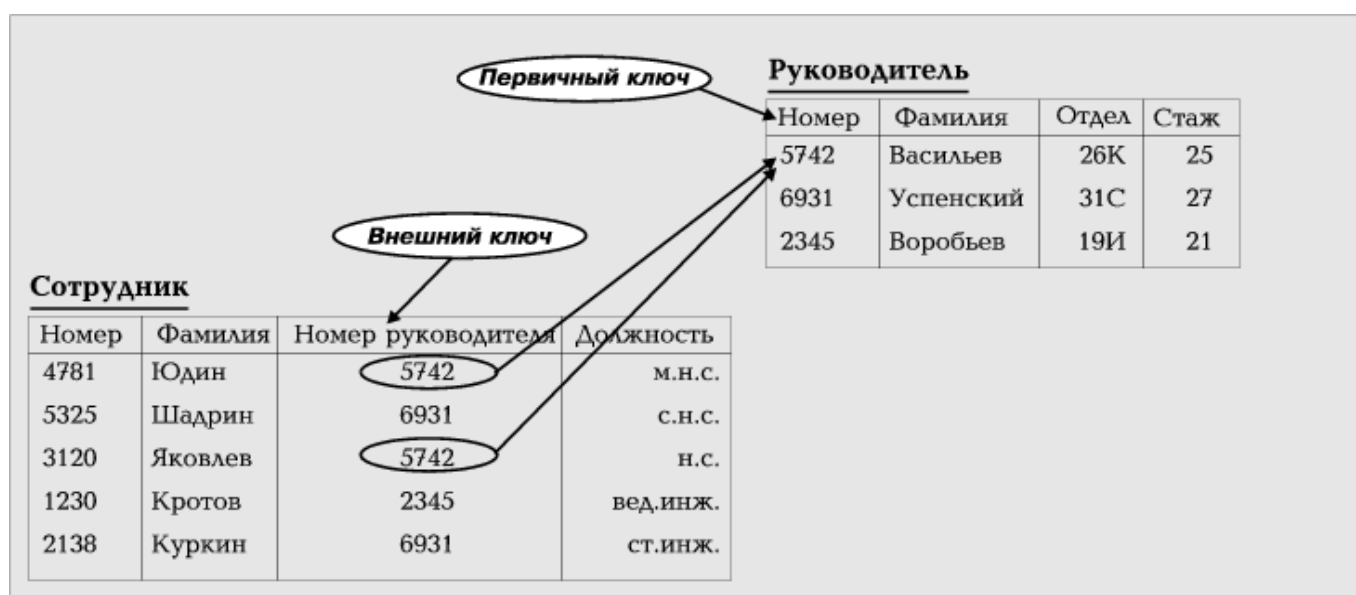
Sname – sotuvchi ismi,

City – sotuvchi manzili (yashah shahri),

Comm – sotuvchilarning ustama ish haqi.

Buyurtmachilar (Customers) jadvali:

CNUM	FNAME	CNAME	CITY	RATING	SNUM
2001	Shoqosimov	Rasul	Navoiy	100	1001
2002	G'aniev	Olqor	Asaka	200	1003
2003	Abdullaev	Lutfulla	Samarqand	200	1002
2004	Murodov	Umid	Samarqand	300	1002
2006	Salimov	Akram	Jizzax	100	1001
2008	Burxonov	Toxir	Samarqand	300	1007
2007	Po'latov	Jalil	Farg'ona	100	1004



6-*rasm. Ma'lumotlar bazasini bir-biri bilan bog'lash.*

Ma'lumotlarning relyatsion bazasidagi munosabatlar ustida bajariladigan asosiy amallar sakkizta bo'lib, ular quyidagilardan iborat:

- To'plamlar ustidagi ananaviy amallar, ya'ni to'plamlarning birlashmasi (yig'indisi), kesishmasi (ko'paytmasi), to'ldiruvchisi (ayrmasi), dekart ko'paytmasi, bo'lishmasi.
- Maxsus relyatsion amallar, ya'ni proektsiyalash, bog'lanish (qo'shilish), birlashtirish (ulab qo'yish) va tanlash.

Bu erda: CNum – har bir buyurtmachining unikal raqami (kodi),

Fname – buyurtmachinig harifi,

CName – buyurtmachi ismi,

City – buyurtmachi manzili (yashash shahri),

Rating – buyurtmachining boshqalardan ustunlik darajasini ko'rsatuvchi raqami.

Ma'lumotlarning relyatsion bazasidagi munosabatlar ustida bajariladigan asosiy amallar sakkizta bo'lib, ular quyidagilardan iborat:

- To'plamlar ustidagi ananaviy amallar, ya'ni to'plamlarning birlashmasi (yig'indisi), kesishmasi (ko'paytmasi), to'ldiruvchisi (ayrmasi), dekart ko'paytmasi, bo'lishmasi.
- Maxsus relyatsion amallar, ya'ni proektsiyalash, bog'lanish (qo'shilish), birlashtirish (ulab qo'yish) va tanlash.

(reyingi),

SNum – buyurtmachiga tayinlangan sotuvchi raqami (kodi).

Buyurtmalar (Orders) jadvali:

ONUM	AMT	ODATE	CNUM	SNUM
3001	1860	10/03/2013	2008	1007
3003	76719	10/03/2013	2001	1001
3002	19010	10/03/2013	2007	1004
3005	51605	10/03/2013	2003	1002
3006	10981	10/03/2013	2008	1007
3009	17132	10/04/2013	2002	1003
3007	7575	10/04/2013	2004	1002
3008	47230	10/05/2013	2006	1001
3010	13099	10/06/2013	2004	1002
3011	98918	10/07/2013	2006	1001

Bu erda: Onum – har bir sotib olingan mahsulotning unikal raqami (kodi),

AMT – sotib olingan mahsulot summasi qiymati,

Odate – sotib olish sanasi,

CNum – sotib oluvchi buyurtmachining raqami (kodi),

SNum – sotuvchining raqami (kodi).

### Uyda bajarish uchun topshiriqlar

1. Ma'lumotlar bazasi tushunchasi.
2. Ma'lumotlarning ierarxik modeli tushunchasi.
3. Ma'lumotlarning tarmoqli modeli tushunchasi.
4. Ma'lumotlarning relyatsion modeli tushunchasi.

### Adabiyotlar

1. Кириллов В.В. Основы проектирования реляционных баз данных. Учебное пособие. СПб.ИТМО,1994.
2. Дейт К. Введение в системы баз данных. -М.Наука,1980 г.
3. Кузнецов С.Д. Введение в стандарты языка баз данных SQL.-М. 1998г.

### Internet ma'lumotlari

1. [www.sql.ru](http://www.sql.ru)
2. [www.mysql.ru](http://www.mysql.ru)

## 2-Mavzu: SO'ROVLARNI LOYIHALASHTIRISH.

1. So'rovlar to'g'risida umumiy tushunchalar.
2. SQL tilida so'rovlarni loyihalashtirish.

### 1. So'rovlar to'g'risida umumiy tushunchalar.

SQL tilida jadvallar bilan ishlashda eng ko'p ishlatiladigan buyruq SELECT (tanlash) operatoridir. Biz SELECT operatori yordamida so'rov beramiz, u esa bizga kerakli ma'lumotlarni jadvallardan izlab topadi va jadval sifatida taqdim etadi. Bu natijaviy jadval esa boshqa SELECT operatori tomonidan qayta ishlanishi mumkin. Bu operator sintaksisi quyidagicha:

```
SELECT ALL <sxema yoki ustun> DISTINCT < * >  
FROM <sxema yoki jadval>  
WHERE <izlash sharti>  
GROUP BY <sxema yoki ustun>  
HAVING <izlash sharti>  
ORDER BY <tartiblash yozuvlari>;
```

SELECT operatorini yaratishdagi birinchi qoida, SELECT ifodasi o'z ichiga albatta, hech bo'lmaganda bitta, FROM ifodasini olishi kerak. Qolgan ifodalar kerak hollarda ishlatiladi. Ikkinchi qoida, SELECT kalit so'zidan so'ng chiqarilishi lozim bo'lgan ustunlar ro'yxati ko'rsatilishi lozim.

Uchinchi qoida, FROM kalit so'zidan so'ng so'rovni bajarish uchun kerakli jadvallar nomi yoziladi. Misol uchun sotuvchlar jadvalidagi hamma yozuvlarni ekranga chiqarish lozim bo'lsin. Unda SELECT operatorini quyidagicha yozishimiz mumkin:

```
SELECT Snum, Sname, City, Comm FROM Salespeople;  
yoki SELECT * FROM Salespeople;
```

Komanda	Amalga oshiriladigan jarayon	Izox
SELECT	Ma'lumotlar bazasidan ma'lumotlarni tanlash	Tanlash
INSERT	Ma'lumotlar bazasiga ma'lumot qo'shish	Qo'yosh
UPDATE	Ma'lumotlar bazasidagi ma'lumotlarni yangilash	Yangilash
DELETE	Ma'lumotlar bazasidan ma'lumotlarni o'chirish	O'chirish
GRANT	Foydalanuvchiga imkoniyat qo'shish	Ruxsat berish
REVOKE	Foydalanuvchi imkoniyatlarini cheklash	Bekor qilish
COMMIT	Joriy tranzaksiyani belgilash	O'rnatish


7-rasm. SQL tilinig komandalari.

WHERE kalit so'zidan so'ng kerakli ma'lumotlarni izlash sharti yoziladi.

GROUP BY ifodasi guruhlar asosida natijaviy so'rovlarni yaratishga imkon beradi. HAVING ifodasi GROUP BY ifodasi bilan birgalikda ishlatilib, unda guruhlarini qaytarish sharti yoziladi.

ORDER BY ifodasi natijaviy ma'lumotlarni olishda ularni qanday tartiblash yo'nalishini bildiradi. SQL tili muhitida har bir foydalanuvchi maxsus identifikatsiyali nom va murojlat identifikatoriga (ID) ega bo'ladi. Ma'lumotlar bazasiga berilgan buyruq ma'lum foydalanuvchi bilan yoki boshqacha aytganda maxsus murojaat identifikatori bilan bog'lanadi. SQL ma'lumotlar bazasidagi ID ruxsat – bu foydalanuvchi nomi va SQL buyrug'i bilan bog'langan murojaat identifikatoriga ilova qiluvchi maxsus kalit so'z USER dir.

Foydalanuvchini tizimda qayd etish, bu kompyuter tizimiga kirish huquqini olish uchun foydalanuvchi bajarishi kerak bo'lgan aniq protseduradir. Bu protsedura foydalanuvchi bilan qaysi murojaat IDsi bog'lanishini lozimligini aniqlaydi. Odatda har bir ma'lumotlar bazasidan foydalanuvchi o'zining alohida IDsiga ega bo'lishi kerak va IDsini qayd qilish jarayonida u MBning haqiqiy foydalanuvchisiga aylanadi. SQL tizimida ko'p topshiriqlarga ega foydalanuvchilar bir necha murojaat ID lari bilan qayd qilinishi yoki bir necha foydalanuvchi bitta murojaat ID sidan foydalanishlari ham mumkin.



```

===== SQL Execution Log =====
|
| SELECT snum, sname, sity, comm
| FROM Salespeople;
|
| =====
|      snum      sname      city      comm
| -----
|      1001      Peel      London      0.12
|      1002      Serres     San Jose     0.13
|      1004      Motika     London      0.11
|      1007      Rifkin     Barcelona    0.15
|      1003      Axelrod    New York     0.10
|
| =====
  
```

8-rasm.SELECT komandasining ishlatilishi.

SELECT operatori MB jadvallaridan natijaviy to'plam olish uchun mo'ljallangan ifodadir. Biz SELECT operatori yordamida so'rov beramiz, u bo'lsa ma'lumotlar natijaviy to'plamini qaytaradi. Bu ma'lumotlar jadval shaklida qaytariladi. Bu jadval keyingi SELECT operatori tomonidan qayta ishlanishi mumkin va xokazo.

**SELECT FILED1, (FIELD2 - FIELD3) "CONST" ... FROM ...**

FROM jumlası "jadval spetsifikatorlari ", Ya'ni so'rovni tashkil qiluvchi jadvallar nomini o'z ichiga oladi. Bu jadvallar so'rov asosini tashkil qiluvchi jadvallar deyiladi.

Misol: Hamma xizmatchilarning nomlari, ofislari va ishga olish sanalari ro'yxatini xosil qilish.

**SELECT NAME, REP\_OFFICE, HIRE\_DATE FROM SALESREPS**

operatora SELECT qaytaruvchi ustunlar ixtiyoriysi hisoblanuvchi, Ya'ni natijada mustaqil ustun sifatida tasvirlanuvchi matematik ifoda bo'lishi mumkin.

Misol: Har bir ofis uchun shaharlar, regionlar va sotuvlar rejasi qanchaga ortig'i yoki kami Bilan bajarilganligi ro'yxati.

**SELECT CITY, REGION, (SALES-TARGET) FROM OFFICES**

Har bir xizmatchi uchun rejadagi sotuvlar xajmini haqiqiy sotuvlar xajmining 3% foiziga oshirish!

**SELECT NAME, QUOTA, (QUOTA +((SALES/100)\*3)) FROM SALESREPS**

Ba'zida ustunlardan biri izlash shartiga bog'liq bo'lmagan qiymat qaytarishi kerak bo'ladi!

Masalan: Har bir shahar uchun sotuvlar xajmlari ro'yxatini chiqaring.

**SELECT CITY, 'Has sales of', SALES FROM OFFICES**

'Has sales of' bu konstantalar ustunidir.

Ba'zida ma'lumotlarni tanlashda qaytariluvchi qiymatlar xosil bo'ladi.

Bu xol yuz bermasligi uchun DISTINCT operatoridan foydalanish lozim. Masalan, quyidagicha:

**SELECT DISTINCT MGR FROM OFFICES**

***SELECT operatori WHERE sharti***

Endi WHERE ifodasidan foydalanib ba'zi so'rovlarni ko'rib chiqamiz: Sotuvlar haqiqiy xajmi rejadan oshgan ofislarni ko'rsating.

**SELECT CITY, SALES, TARGET FROM OFFICES WHERE SALES > TARGET**

**Zdes WHERE SALES > TARGET, znachit, esli SALES bolshe TARGET!**

Identifikatori 105 ga teng bo'lgan xizmatchi nomi haqiqiy va rejadagi sotuvlar xajmini ko'rsating:

**SELECT SALES, NAME, QUOTA FROM SALESREPS WHERE  
EMPL\_NUM = 105**

**Zdes WHERE EMPL\_NUM = 105, oznachaet, EMPL\_NUM ravno 105!**

Agar izlash sharti ROST(TRUE), bo'lsa qator natijaviy to'plamga qo'shiladi, agar izlash sharti LOJNO(FALSE), qator natijaviy to'plamga qo'shilmaydi, agar NULL bo'lsa ham natijaviy to'plamdan chiqariladi! O'z ma'nosiga ko'ra WHERE, keraksiz yozuvlarni chiqarib, kerakligimni qoldiruvchi filtr sifatida ishlatiladi!

Asosiy izlash shartlari "predikatlar", beshta. Ularni ko'rib chiqamiz:

1. Solishtirish, Ya'ni bir shart natijasi ikkinchisi bilan solishtiriladi. Birinchi so'rov kabi.

2. Qiymatlar diapazoniga tegishlilikni tekshirish. Masalan berilgan qiymat diapazonga kiradimi yo'qmi.

3. To'plam elementiligini tekshirish. Masalan, ifoda qiymati to'plamdagi biror qiymat Bilan ustma ust tushadimi.

4. Shablonga moslikni tekshirish. Ustundagi satrli qiymat shablonga mos keladimi.

5. NULL qiymatga tenglikka tekshirish.

Solishtirish amallari maydon va konstantalarni solishtirish amallarini o'z ichiga olishi mumkin: 1988 yilgacha ishga olingan hamma xizmatchilar nomlarini toping.

**SELECT NAME FROM SALESREPS**

**WHERE HIRE\_DATE <TO\_DATE('01.06.1988','DD/MM/YYYY')**

TO\_DATE('01.06.1988','DD/MM/YYYY') - PL/SQL Oracle sana Bilan ishlash standart funktsiyasi.

Yoki arifmetik ifodalarni o'z ichiga olishi mumkin: Haqiqiy sotuvlar xajmi rejaning 80 foizidan kam bo'lgan ofislar ro'yxatini chiqaring.

**SELECT CITY, SALES, TARGET FROM OFFICES**

**WHERE SALES < (0.8 \* TARGET)**

Ko'p xollarda izlash birlamchi kaliti bo'yicha konstantalar bilan solishtirish so'rovlaridan foydalaniladi, masalan shahar telefon tarmog'i abonent, axir ikkita bir xil nomerlar mavjud emas!

Bundan tashqari har bir foydalanuvchi SQL ma'lumotlar bazasida nima qilish mumkinligini ko'rsatuvchi imtiyozlarga egadir. Bu imtiyozlar vaqt o'tishi bilan o'zgarishi, ya'ni eskilari o'chirilib, yangilari qo'shilishi mumkin. SQL tizimi imtiyozlari bu ob'ektlarga berilgan imtiyozlardir. Bu shuni bildiradiki, foydalanuvchi o'z buyruq'ini ma'lumotlar bazasining aniq ob'ekti ustida bajarishi mumkin. Ob'ekt imtiyozlari bir vaqtning o'zida foydalanuvchilar va jadvallar bilan bog'liq, ya'ni

imtiyoz ma'lum foydalanuvchiga aniq ko'rsatilgan jadvalga nisbatan o'rnatiladi. Ixtiyoriy turdagi jadvalni yaratgan foydalanuvchi shu jadval egasidir. Bu uni bildiradiki, foydalanuvchi shu jadvalda hamma imtiyozlarga ega va imtiyozlarini shu jadvalning boshqa foydalanuvchilariga ham berishi mumkin.

Agar SELECT operatori o'zining tarkibida bitta yoki bir nachta boshqa SELECT operatorlarini saqlasa, bunday so'rovlarga ichma-ich joylashgan yoki murakkab so'rovlar deyiladi.

## 2. SQL tilida so'rovlarni loyihalashtirish.

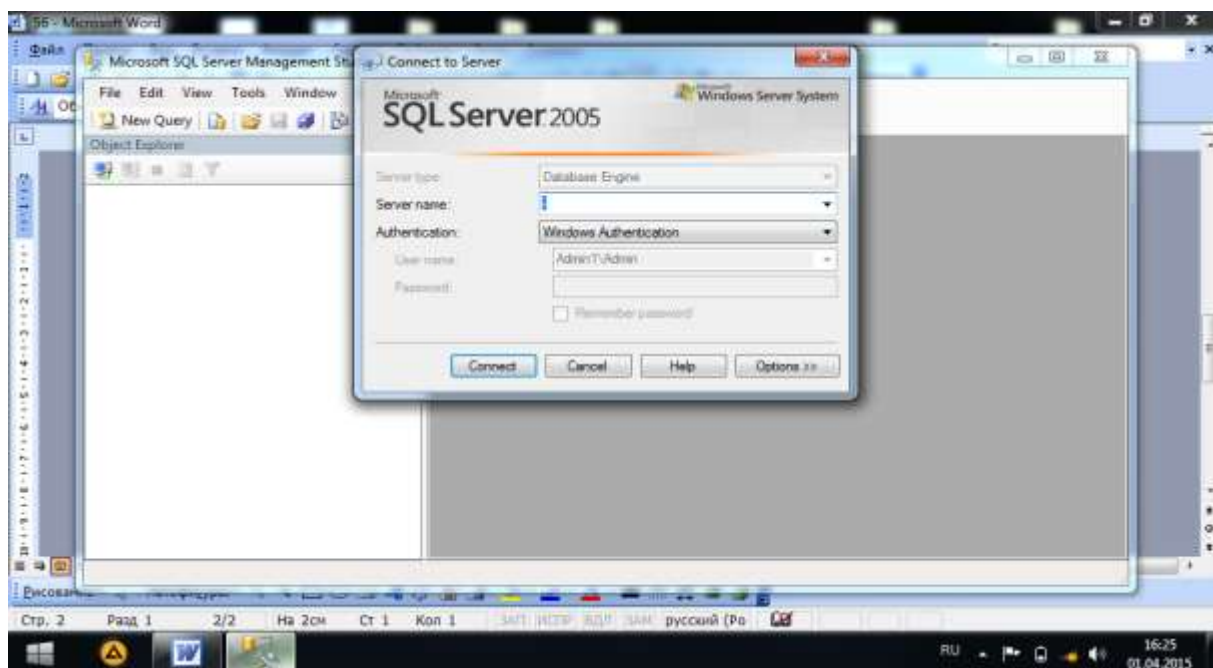
**SQL tilida so'rovlarni yaratish.** Qo'yilgan masalani echish uchun quyidagi ishlarni amalga oshiramiz. Berilgan topshiriqlar ro'yxatidan birorta topshiriq matnini

```

===== SQL Execution Log =====
|
| SELECT snum, sname, sity, comm
| FROM Salespeople;
|
| =====
|      snum      sname      city      comm
| -----
|      1001      Peel      London    0.12
|      1002      Serres    San Jose   0.13
|      1004      Motika    London     0.11
|      1007      Rifkin    Barcelona  0.15
|      1003      Axelrod   New York   0.10
|
| =====
  
```

olamiz. Masalan, Buyurtmalar jadvalinig qiymatlarini matnga chiqaring. Bu topshiriqni bajarish uchun kerakli biyruqni beramiz:

*9-rasm. INSERT komandasining ishlatilishi*



*10-rasm. SQL server oynasini yuklash*



```
SELECT ONUM, AMT, ODATE, CNUM, SNUM FROM ORDERS; yoki  
SELECT * FROM ORDERS;
```

Olingan natijani (jadvalni) hisobotda aks ettiramiz.

Buyruq bajarilishi jarayonida amalga oshgan jarayonlar to'g'risida qisqacha tavsiflar beramiz. Masalan, ushbu topshiriq bajarilishi mobaynida quyidagi jarayonlar amalga oshadi. So'ngra operatorida ko'rsatilgan ustun nomlaridan foydalanib, bizning holatimizda ONUM, AMT, ODATE, CNUM, SNUM qiymatlarini ekranga chiqaradi. Agarda operatorida \* ko'rsatilgan bo'lsa, jadvaldagi mavjud hamma ustun qiymatlari ekranga chiqariladi.

SELECT operatoridan foydalanib, oddiy va murakkab so'rovlar tashkillashtiring. So'rovlarni tashkillashtirishda SQL tilining mantiqiy operatorlari: ALL, SOME, ANY, EXISTS, IN, BETWEEN, LIKE kabi turlarini ishlatib. So'rovlarni tashkillashtirishda quyidagi so'rov variantlaridan foydalaning:

1. Sotuvshilar jadvalinig qiymatlarini matnga chiqaring.
2. Buyurtmachilar jadvalinig qiymatlarini matnga chiqaring.
3. Buyurtmalar jadvalinig qiymatlarini matnga chiqaring.
4. 10.03.2013 yilda xarid qilgan buyurtmachilarni aniqlang.
5. 10.03.2013 yil ishlagan sotuvchilarni aniqlang.
6. 10.03.2013 yilda sotilgan mahsulotlar ro'yxatini matnga chiqaring.
7. ALL operatoridan foydalanib, Samarqand shahrida yashovchi har bir buyurtmachidan reytinglari yuqori bo'lgan buyurtmachilarni toping.
8. ANY operatoridan foydalanib, bir shaharda buyurtmachilari bilan birga yashovchi sotuvchilarni aniqlang.

### **Uyda bajarish uchun topshiriqlar**

1. SQL tilida so'rov tushunchasi.
2. SQL tilidagi qanday so'rov turlari bor.
3. SQL tili so'rovlaridagi ALL kalit so'zi qanday vazifasi bor.
4. SQL tili so'rovlaridagi SOME kalit so'zi tushunchasi.
5. SQL tili so'rovlaridagi EXISTS kalit so'zi tushunchasi.
6. SQL tili so'rovlaridagi IN kalit so'zi tushunchasi.

### **Adabiyotlar**

1. Кириллов В.В. Основы проектирования реляционных баз данных. Учебное пособие. СПб.ИТМО,1994.
2. Дейт К. Введение в системы баз данных. -М.Наука,1980 г.
3. Кузнецов С.Д. Введение в стандарты языка баз данных SQL.-М. 1998г.

### **Internet ma'lumotlari**

1. [www.sql.ru](http://www.sql.ru)
2. [www.mysql.ru](http://www.mysql.ru)

### 3-Mavzu: **MA'LUMOTLAR BAZASINI BOSHQARISH.**

1. Ma'lumotlar bazasini boshqarish to'g'risida umumiy tushunchalar.
2. SQL tilida ma'lumotlar bazasini loyihalash.

#### **1. Ma'lumotlar bazasini boshqarish to'g'risida umumiy tushunchalar.**

MBni boshqarish deganda, mavjud yozuvlarni o'zgartirish, jadvallardan keraksiz yozuvlarni olib tashlash, MB jadvallariga yangi yozuvlarni qo'shish, jadvallarga yangi ustunlar qo'shish (keraksiz ustunlarni olib tashlash), jadval nomlarini o'zgartirish kabi amallar kiradi.

SQL muhitida tashkillashtirilgan MB jadvallaridagi ma'lumotlarni o'zgartirish uchun UPDATE (o'zgartirish) buyrug'idan foydalaniladi. Bu buyruqda jadval nomini ko'rsatishdan tashqari, SET (aniq ustun qiymatlarini bildiradi) kalit so'zii ham ishlatiladi. Masalan, hamma buyutmachilarning reytingini 200 ga o'zgartirish uchun quyidagi buyruqni berish mumkin:

```
UPDATE Customers SET rating = 200;
```

Bu buyruqda jadvaldagi hamma qator qiymatlarini emas, ayrim qatorlarni tanlab o'zgartirish ham mumkin. Bunun uchun predikatlardan foydalanamiz. Masalan, sotuvchi Peel xizmat ko'rsatadigan barcha buyutmachilarning reytingini 200ga o'zgartirish uchun, quyidagi buyruqni beramiz:

```
UPDATE Customers SET rating = 200 WHERE snum = 1001;
```

Siz UPDATE buyrug'ida ifodalardan foydalanishingiz ham mumkin. Masalan, barcha sotuvchilarning ustamalarini ikki barobar oshirmoqchisiz. Bunun uchun quyidagicha buyruq berishingiz mumkin:

```
UPDATE Salespeople SET comm = comm * 2;
```

Bu buyruq yordamida jadvalga NULL qiymatlarni ham kiritish mumkin. Masalan, London shahrida yashovchi buyurtmachilarning reytinglari o'rniga NULL qiymat kiritish lozim bo'lsin. Unda quyidagi buyruqni beramiz:

```
UPDATE customers SET rating = NULL WHERE city = 'London';
```

SQL muhitida tashkillashtirilgan MB jadvallaridan ma'lumotlarni olib tashlash uchun DELETE (olib tashlah) buyrug'idan foydalaniladi. Masalan, Sotuvchilar jadvalidagi ma'lumotlarning hammasini olib tashlash uchun quyidagi buyruqdan foydalanishimiz mumkin:

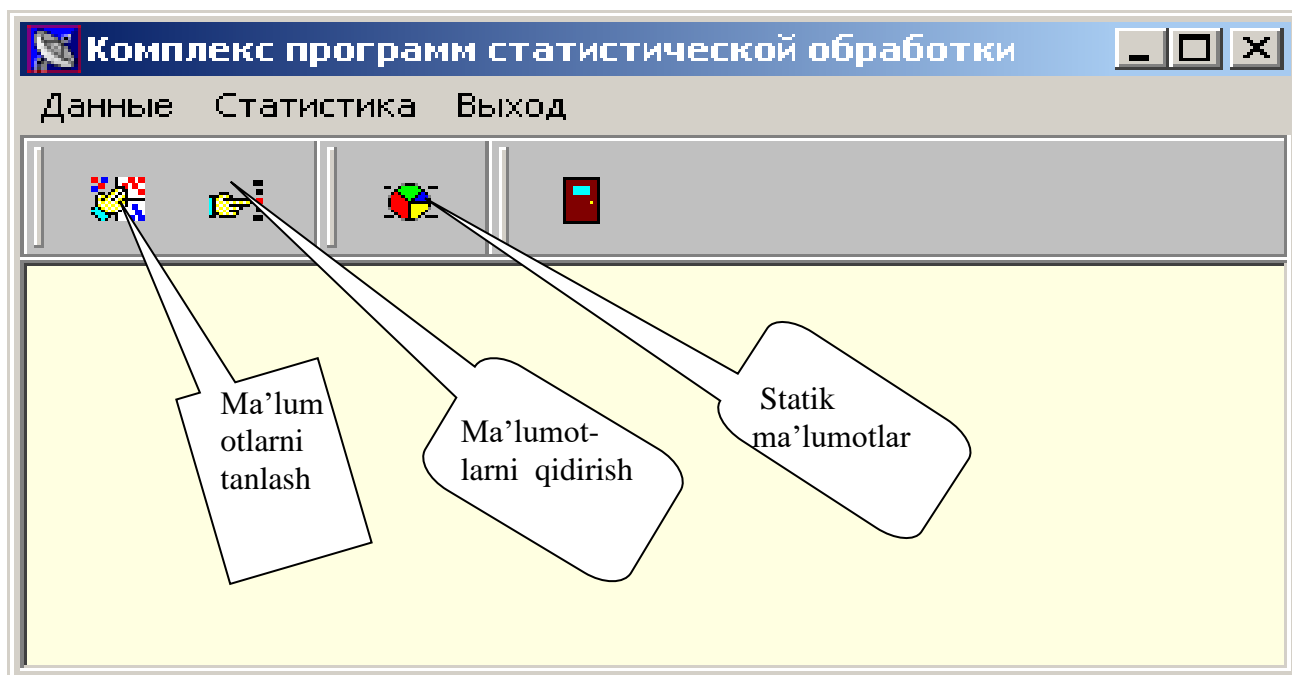
```
DELETE FROM Salespeople;
```

Ana endi bo'sh jadvalni butunlay olib tashlash mumkin:

```
DROP TABLE Salespeople;
```

MB jadvalliridan ko'rsatilgan qatorlarni ham olib tashlash mumkin. Masalan, Sotuvchilar jadvalidan sotuvchi Rozzoqovni olib tashlash uchun quyidagi buyruqni beramiz:

```
DELETE FROM Salespeople WHERE snum = 1007;
```



11-rasm. Dasturning asosiy oynasi.

Bu buyruq yordamida jadvaldan bir qancha qatorlarni birdaniga olib tashlash ham mumkin. Masalan, Sotuvchilar jadvalidan barcha London shahrida yashovchi sotuvchilarni olib tashlash uchun quyidagicha buyruq beriladi:

```
DELETE FROM Salespeople WHERE city = 'London';
```

Mavjud jadvallarga qatorlar qo'shish INSERT INTO buyrig'i orqali amalga oshiriladi. Bu buyruq formati quyidagicha:

```
INSERT INTO table_name (field_name1, field_name2,...)
values('content1', 'content2',...);
```

Bu buyruq table\_name jadvalining field\_name\_N maydonlariga content\_N qiymat o'rnatilgan yozuv qo'shadi. Masalan, biz tel\_numbr jadvaliga (fio, adres va telefon) qymatlarini kiritmoqchi bo'lsak, quyidagi byuruqni berishimiz lozim:

```
INSERT INTO tel_numbr(fio, address, tel)
values('Vali Ismoilov', 'Furqat ko'ch., uy.18', '354-54-54');
```

Ko'pgina hollarda MBdagi jadval nomlarini o'zgartirishga to'g'ri keladi. Buni quyidagi formatli buyruqdan foydalanib amalga oshirsa bo'ladi:

```
ALTER TABLE table_name_old RENAME table_name_new;
```

bu erda,

table\_name\_old – jadvalning eski nomi;

table\_name\_new – jadvalning yangi nomi.

Misol uchun search jadvali nomini search\_en nomiga o'zgartirish kerak bo'lsin. Buning uchun quyidagi buyruq beriladi:

```
ALTER TABLE search RENAME search_en;
```

MB jadvaliga ustunlar qo'shish uchun quyidagi formatli buyruqdan foydalanamiz:

```
ALTER TABLE table_name ADD field_name parametrs
```

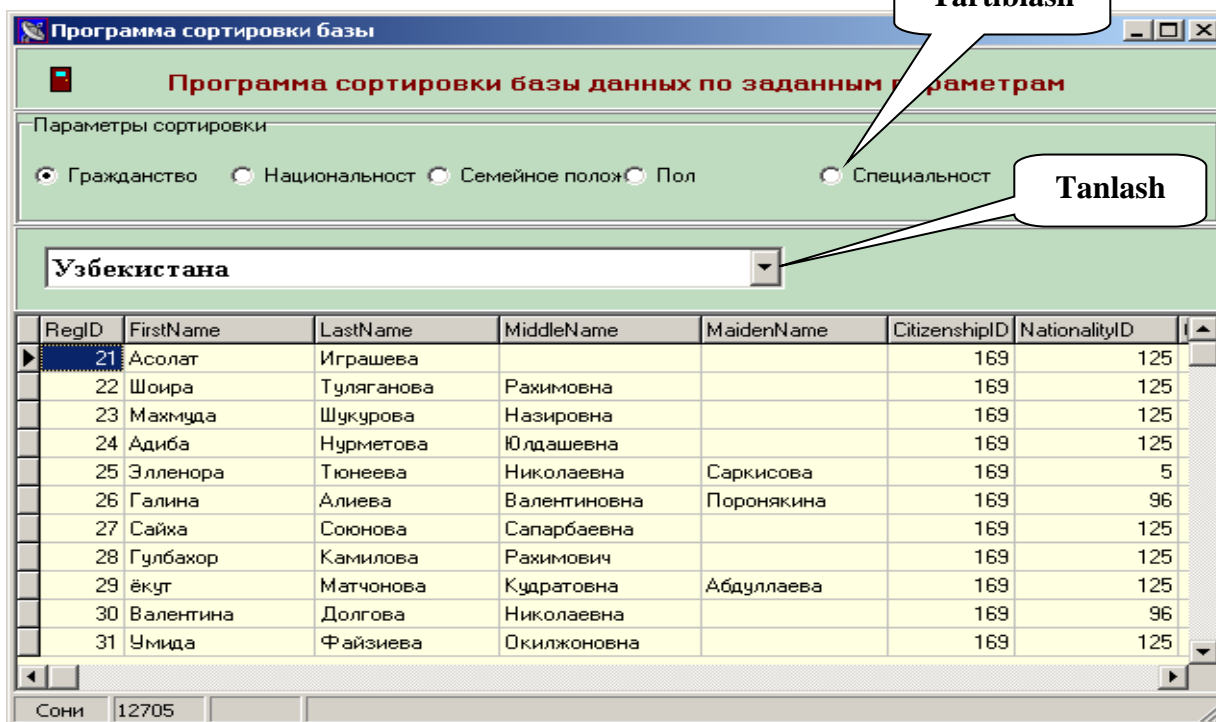
bu erda, table\_name – yangi ustun qo'shiladigan jadval nomi;

field\_name – qo'shilayotgan ustun nomi;

parametrs – qo'shilayotgan ustunni tasvirlovchi parametrlar.

Masalan, my\_friends nomli jadvalga adress\_2 nomli matn qiymatlariga ega ustun qo'shishimiz kerak bo'lsin. Unda quyidagicha buyruq beriladi:

```
ALTER TABLE my_friends ADD adress_2 TEXT;
```



12-rasm. Ma'lumotlarni tartiblash tanlash formasi.

Odatda yangi ustun jadval oxiriga qo'shiladi. Agar ustun jadval boshiga qo'shilishi kerak bo'lsa, qo'shilayotgan ustun parametrlaridan so'ng FIRST kalit so'zini yozish kerak:

```
ALTER TABLE my_friends ADD adress_2 TEXT FIRST;
```

Agar ustun jadval boshi yoki oxiriga emas, balki ma'lum ustundan keyin qo'yilishi lozim bo'lsa, AFTER kalit so'sidan foydalanamiz:

```
ALTER TABLE my_friends ADD adress_2 TEXT AFTER adress_1;
```

Bu misolda yangi adress\_2 ustuni adress\_1 ustunidan keyin qo'yiladi.

Agar jadvalga bitta emas, bir necha ustun qo'shish kerak bo'lsa har bir ustun uchun ADD field\_name parametrs yozivuni vergul orqali har bir ustun uchun yozish kerak. Masalan:

```
ALTER TABLE my_friends ADD adress_2 TEXT,  
ADD adress_3 TEXT, ADD adress_4 TEXT;
```

Agar jadvalga bir ustundan so'ng, birdaniga ikki ustun qo'shish lozim bo'lsa, quyidagicha amalga oshirish mumkin:

```
ALTER TABLE my_frends ADD adress_2 TEXT AFTER adress_1,  
ADD adress_3 TEXT AFTER adress_2";
```

Bu misolda yangi ustunlar adress\_1 va adress\_3 ikkinchi ustundan so'ng qo'shiladi.

MB jadvallaridan satrlarni o'chirish uchun DELETE buyrug'idan foydalanamiz. Bu buyruq formati quyidagicha:

```
DELETE FROM table_name WHERE (ifoda) ;
```

Bu erda ifoda - oddiy mantiqiy ifoda.

Masalan, FIO, adres va telefon raqamini o'z ichiga olgan jadvaldan biror-bir yozuvni (FIO: Karim Mansurovich bolgan kishini) olib tashlash uchun:

```
DELETE FROM tel_numb WHERE (fio='Karim Mansurivuch');
```

buyrug'ini berashimiz kerak.

Agar bir nechta parametr bo'yicha o'chirish lozim bo'lsa, buyruq quyidagicha beriladi:

```
DELETE FROM tel_numb WHERE (fio='Karim Mansurovich' & tel='54-54-54');
```

## 2. SQL tili va ma'lumotlar tiplari

Ma'lumotlar tiplari-SQL tilida quyidagi asosiy ma'lumotlar tiplari ishlatilib, ularning formatlari har xil MBBT lar uchun farq qilishi mumkin:

INTEGER - butun son (odatda 10 tagacha qiymatli raqam va ishora).

SMALLINT - "qisqa butun" (odatda 5 tagacha qiymatli raqam va ishora).

DECIMAL(p, q) - o'nli son, p raqam va ishoradan iborat ( $0 < p < 16$ ). O'nli nuqtadan so'ng raqamlar soni q orqali beriladi ( $q < p$ , agar  $q = 0$  bo'lsa, tashlab yuborilishi mumkin).

FLOAT - haqiqiy son 15 ta qiymatli raqam va butun darajadan iborat. Daraja MBBT tipi bilan aniqlanadi (masalan, 75 yoki 307).

CHAR(n) - uzunligi o'zgarmas, n ga teng bo'lgan simvulli qator ( $0 < n < 256$ ).

VARCHAR(n) - uzunligi o'zgaruvchi, n simvoldan oshmagan simvulli qator ( $n > 0$  va har xil MBBT larda har xil lekin 4096 dan kam emas).

DATE - maxsus komanda orqali aniqlanuvchi formatdagi sana (Subase da ko'zda tutilgan bo'yicha yy/mm/dd); sana maydonlari bizning eramizdan oldin bir necha mingyilliklardan boshlanuvchi va bizning eramiz beshinchi- o'ninchi mingyilligi bilan cheklangan

haqiqiy sanalarni o'z ichiga olishi mumkin.

**TIME** -maxsus komanda orqali aniqlanuvchi formatdagi vaqt (ko'zda tutilgan bo'yicha hh.mm.ss).

**DATETIME** - sana va vaqt kombinatsiyasi. (Sybase da **TIMESTAMP** ).

**MONEY** -maxsus komanda orqali aniqlanuvchi formatdagi pul. Format o'z ichiga pul birligi simvoli (\$, rub, ...) va uning joylashuvi (suffiks yoki prefiks), kasr qism aniqligi va pul qiymatini ko'rsatish shartlarini oladi.

**DELETE FROM Salespeople WHERE city = 'Farg'ona';**  
Topshiriq bajarilishidan oldingi jadval holati quyidagicha:

SNUM	FNAME	SNAME	CITY	COMM
1001	Sobirov	Raxim	Navoiy	12%
1002	Qosimov	Sodiq	Samarqand	11%
1004	Muammadiev	Lapas	Farg'ona	11%
1007	Razzoqov	Asam	Samarqand	15%
1003	Asqarov	Nodir	Namangan	10%

Topshiriq bajarilganidan keyingi jadval holati quyidagicha bo'ladi:

SNUM	FNAME	SNAME	CITY	COMM
1001	Sobirov	Raxim	Navoiy	12%
1002	Qosimov	Sodiq	Samarqand	11%
1007	Razzoqov	Asam	Samarqand	15%
1003	Asqarov	Nodir	Namangan	10%

### Uyda bajarish uchun topshiriqlar

1. SQL tilida jadval nomlarini o'zgartirish.
2. SQL tilida jadval yozuvlarini o'zgartirish.
3. SQL tilida jadvaldan yozuvlarini olib tashlash.
4. SQL tilida jadvalga yangi yozuvlarini qo'shish uchun qanday buyruq beriladi?
5. SQL tilida jadvaldan ustunlarni olib tashlash uchun qanday buyruq beriladi?

### Adabiyotlar

1. Murodov O'.M. SQL texnologiyalar fanidan maruzalar matni. TATU Samarqand filiali, 2012.
2. Кириллов В.В. Основы проектирования реляционных баз данных. Учебное пособие. СПб.ИТМО,1994.

#### 4-Mavzu: MA'LUMOTLAR BAZASI FUNKSIYALARI BILAN ISHLASH.

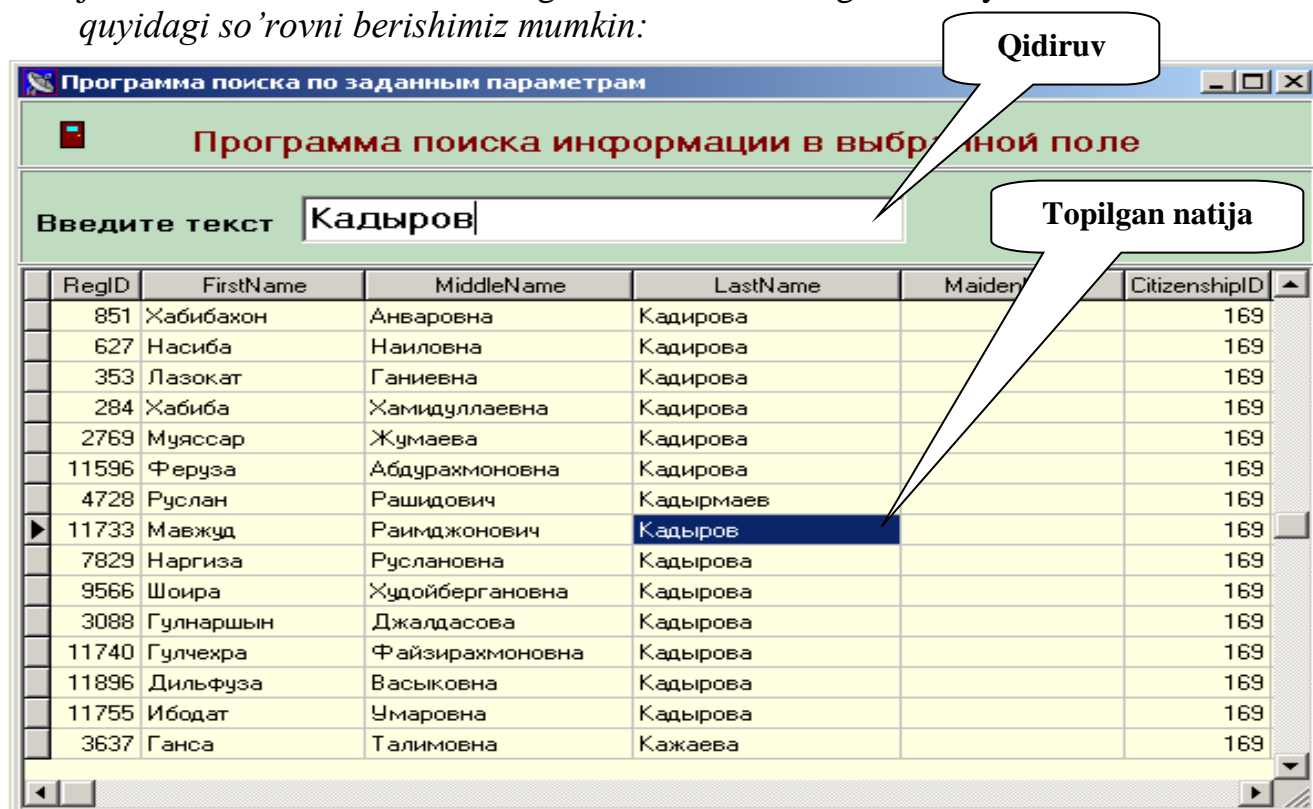
1. Ma'lumotlar bazasi funksiyalari to'g'risida umumiy tushunchalar.
2. SQL tilida MB funksiyalari bilan ishlash.

##### 1. Ma'lumotlar bazasi funksiyalari to'g'risida umumiy tushunchalar.

SQL tili so'rovlarida MBdan ma'lumotlarni olish uchun bir qancha turdagi funksiyalar ishlatiladi. Shunday funksiyalaran biri agregat funksiyalari hisoblanib, bu funksiyalar jadval barcha qiymatlaridan bitta qiymatni olish imkonini beradi. Bular quyidagilar:

- COUNT - jadvaldagi qatorlar sonini aniqlaydi.
- SUM - maydon qiymatlarining umumiy yig'indisini hisoblaydi.
- AVG - tanlangan maydon qiymatlarning o'rta arifmetik qiymatini hisoblaydi.
- MAX - tanlangan maydon qiymatlarning eng kattasini aniqlaydi.
- MIN - tanlangan maydon qiymatlarning eng kichigini aniqlaydi.

Agregat funksiyalarini ishlatish usullarini ko'rib o'tamiz. Masalan, Buyrtmalar jadvalidan hamma sotib olingan mahsulotlarning umumiy bahosini bilish uchun *quyidagi so'rovni berishimiz mumkin:*



13-rasm. Ma'lumotlar bazasidan qidirish.

SELECT SUM (amt) FROM Orders;

Agregat funksiyasini ishlatishda kalit so'zlardan ham foydalanish mumkin. Ana shunday kalit so'zlardan biri GROUP BY kalit so'zidir. Bu kalit so'z maxsus maydonlarda boshqa maydon terminlaridan foydalangan holda agregat funksiyalarini ishlatish imkoniyatini beradi. Bu SELECT operatorida agregat funksiya va maydon nomlarini bitta SELECT buyrug'ida ifodalash imkoniyatini yaratadi. Masalan, siz har bir sotuvchining eng qimmat qiymatda sotgan mahsulotini ko'rmoqchisiz. Buni siz har bir sotuvchini alohida tanlab (SALESPEOPLE jadvalidan), uning sotgan mahsulotlari ichidan eng qimmatini topish yo'li bilan ham aniqlashingiz mumkin.

Ammo GROUP BY kalit so'zi buni bitta SELECT operatorida birlashtirib, engil bajarish imkonini beradi:

```
SELECT snum, MAX (amt) FROM Orders GROUP BY snum;
```

GROUP BY funksiyasini bir nechta maydonlar ishtirokida ham ishlatsa bo'ladi.

Masalan, siz har bir sotuvchining har bir kundagi eng ko'p savdo summasini ko'rmoqchi bo'lsangiz, quyidagicha so'rov berishingiz mumkin:

```
SELECT snum, odate, MAX (amt) FROM Orders GROUP BY snum, odate;
```

SQL tilida agregat funksiyalardan tashqari yana bir qancha funksiyalar ham ishlatiladi va ular quyidagicha sinflashtiriladi:

- matematik funksiyalar;
- qatorli funksiyalar;
- sana-vaqt funksiyalari;
- tizimni konfiguratsiya funksiyalari;
- tizim funksiyalari;
- xavfsizlik tizimi funksiyalari;
- ma'lumotlarni boshqarish funksiyalari;
- statistika funksiyalari.

SQL tilida matematik funksiyalaridan quyidagilar ishlatiladi:

ABS (son) – sonning absolyut qiymatini hisoblaydi;

ISNUMERIC (ifoda) – ifodaning sonli turga mansubligini aniqlaydi;

SING(son) – sonning ishorasini aniqlaydi;

RAND(utun son) – 0 va 1 oralig'ida berguli siljiydigan favqulotdagi sonni hisoblaydi;

ROUND(son, aniqlik) – ko'rsatilgan aniqlikda sonni yaxlitlaydi.

POWER(son, daraja) – sonni ko'rsatilgan darajaga ko'taradi;

SORT(son) – sondan kvadrat ildiz chiqaradi;

SIN(burchak) – radianlarda ifodalangan burchak sinusini hisoblaydi;

COS(burchak) – radianlarda ifodalangan burchak kosinusini hisoblaydi;

EXP(son) – sonning eksponentasini hisoblaydi;

LOG(son) – sonning natural logarifmini hisoblaydi;

Qatorlar bilan ishlash funksiyalari:

LEN(qator) – qatordagi belgilar sonini hisoblaydi;

LTRIM(qator) – qator boshidagi bo'sh joylar (probel)larni o'chiradi;

RTLIM(qator) - qator oxiridagi bo'sh joylar (probel)larni o'chiradi;

LEFT(qator,miqdor) – qatorning chap tomonidan, ko'rsatilgan miqdordagi belgilarni qaytaradi;

RIGHT(qator, miqdor) - qatorning o'ng tomonidan, ko'rsatilgan miqdordagi belgilarni qaytaradi;

LOWER(qator) – ko'rsatilgan qator belgilarini kichik harflarga o'tkazadi;

UPPER(qator) - ko'rsatilgan qator belgilarini katta harflarga o'tkazadi;

STR(son) – son qiymatini belgi formatiga konvertirlashni amalga oshiradi;

SUBSTRING(qator,indeks, uzunlik) – qator uchun ko'rsatilgan uzunlikdagi qator qismini, aniqlashtirilgan indeksdan boshlab qaytaradi.

Tizim funksiyalari:



GETDATE() –tizim faol sanasini qaytaradi;  
ISDATE(qator) – qatorning ko'rsatilgan sana formatiga moslini aniqlaydi;  
DAY(sana) – ko'rsatilgan sana raqamini qaytaradi;  
MONTH(sana) – ko'rsatilgan sanadagi oy raqamini qaytaradi;  
YEAR(sana) – ko'rsatilgan yil raqamini qaytaradi;  
DATAADD(tur,raqam,sana) – sanaga ko'rsatilgan tur birligidagi raqamni  
qo'sadi (yil, oy, kun, saot va boshqalar).

Xavfsizlik tizimi funksiyalari:

SQL muhitida foydalanuvchilarning ma'lumotlarga nisbatan xavfsizligini ta'minlash uchun bir qancha funksiyalar mavjud. Shulardan asosiylarini ko'rib o'tamiz.

SQL Serverda xavfsizlikni o'rnatish uchun quyidagi funksiyalar ishlatiladi:

- GRANT - ob'yekt uchun bajarilishi mumkin bo'lgan vazifalarni, buyruq orqali bajarishga ruxsat berishni o'rnatadi;
- REVOKE – ob'yekt va buyruqlar uchun o'rnatilgan imtiyozlarni bekor qiladi;
- DENY – ob'yekt bilan har qanday amal bajarishni taqiqlaydi (shunu aytish kerakki, REVOKE buyrig'i faqat imtiyozlarni olib tashlayi).

Foydalanuvchiga ruxsat olish funktsiyalarini quyidagi buyruqlar amalga oshiradi:

CREATE DATABASE — ma'lumotlar bazasini yaratish huquqi;

CREATE DEFAULT — jadval ustunlari uchun standart qiymatlarni o'rnatish huquqi;

CREATE PROCEDURE — saqlanadigan protseduralarni yaratish huquqi;

CREATE ROLE — jadval ustunlari uchun qoidalarni o'ranish huquqi;

CREATE TABLE — jadvallarni yaratish huquqi;

CREATE VIEW — ob'ekt tasvirlarini yaratish huquqi;

BACKUP DATABASE — zahira nusxalarni yaratish huquqi;

BACKUP TRANSACTION — tranzaksiyalar zaxira nusxasini yaratish huquqi.

## 2. SQL tilida MB funksiyalari bilan ishlash.

BETWEEN ifodasi bu qiymatlar diapazoniga tegishlilikni tekshirishdir. Ifoda sintaksisi quyidagicha:

**---** tekshirilayotgan ifoda **-----** BETWEEN **-----** quyi ifoda AND yuqori ifoda

**- NOT -**

NOT ifodasi shartni teskarisiga o'giradi, Ya'ni tegishli emas ma'noni bildiradi.

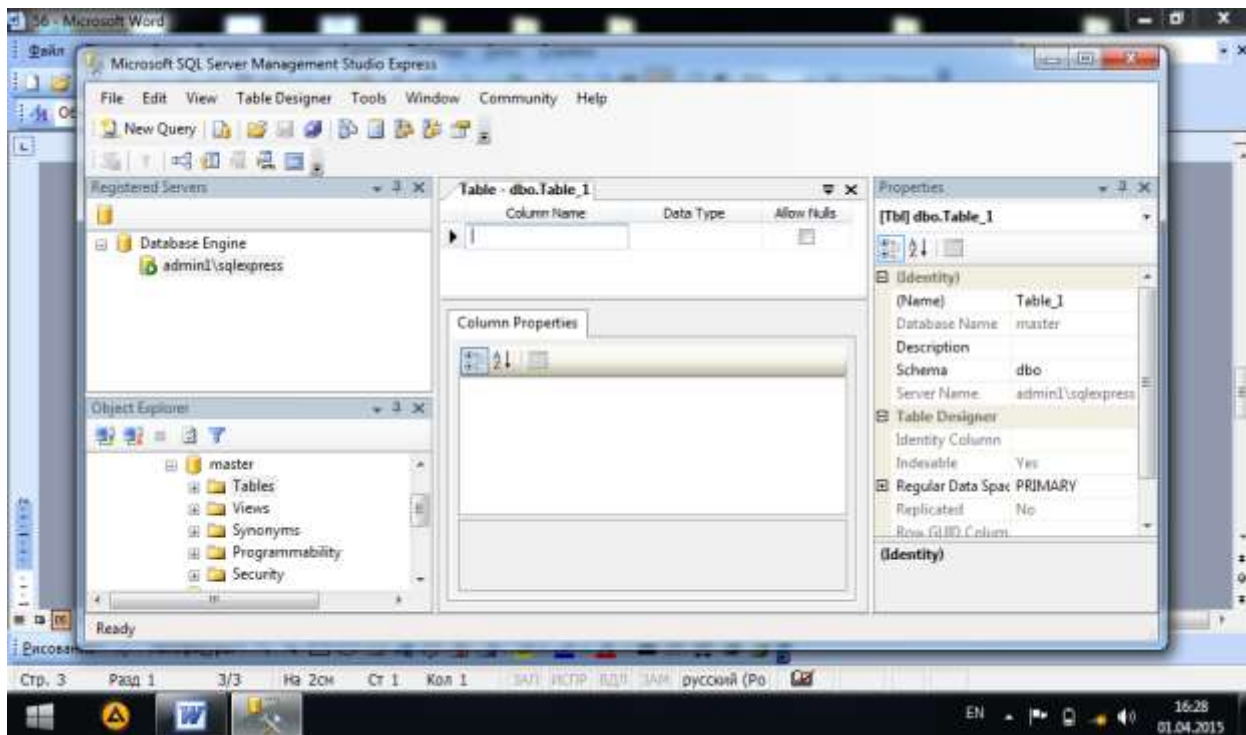
Misol: Narxi har xil diapazonga mos keluvchi buyurtmalarni topish.

**SELECT ORDER\_NUM, AMOUNT**

## FROM ORDERS

**WHERE AMOUNT BETWEEN 20.000 AND 29.999**

NOT ifodasi yordamida berilgan diapazonga tegishlilikni tekshirish mumkin, masalan: Sotuvlar haqiqiy xajmlari rejaning 80 dan 120 protsentigacha bo'lgan diapazonga tushmaydigan xizmatchilar ro'yxatini chiqarish.



14-rasm. SQL tilida jadval tuzish

**SELECT NAME, SALES, QUOTA**

**FROM SALESREPS**

**WHERE SALES NOT BETWEEN (0.8 \* QUOTA) AND (1.2 \* QUOTA)**

Ifoda IN to'plamga tegishlilikni tekshiradi. Komanda sintaksisi quyidagicha:

--- tekshirilayotgan ifoda ----- IN ----- (-- const -----)

- NOT - -- , -----

1990 yil iyun oyining har xil kunlarida qilingan hamma buyurtmalarni aniqlash.

**SELECT ORDER\_NUM, ORDER\_DATE, AMOUNT**

**FROM ORDERS**

**WHERE ORDER\_DATE IN (TO\_DATE('14.06.1990','DD/MM/YYYY'),  
TO\_DATE('08.06.1990','DD/MM/YYYY'),**

**TO\_DATE('29.06.1990','DD/MM/YYYY'),  
TO\_DATE('04.06.1990','DD/MM/YYYY'))**

## ***Operator LIKE***

LIKE ifodasi sintaksisi SQL92 standarti bo'yicha quyidagi ko'rinishga ega:

**--- IMYA STOLBTSA ----- LIKE (shablon) -----**  
**NOT ESCAPE (o'tkazish nomi)**

Sodda so'rov bajaramiz: "Apelsin" kompaniyasi uchun kredit limitini ko'rsatish:

```
SELECT COMPANY, CREDIT_LIMIT  
FROM CUSTOMERS  
WHERE COMPANY = 'Apelin'
```

Quyidagicha '%' shablonli LIKE operatorini qo'llaymiz:

```
SELECT COMPANY, CREDIT_LIMIT  
FROM CUSTOMERS  
WHERE COMPANY LIKE '%n'
```

Bu xolda LIKE '%n' operatori 'n' harfiga tugaydigan hamma yozuvlarni ko'rsatadi, agar '%' shablona birinchi kelsa:

### **Uyda bajarish uchun topshiriqlar**

1. SQL tilidagi qanday agregat funksiyalarni bilasiz?
2. COUNT agregat funksiyasi qanday vazifani bajaradi?
3. SUM agregat funksiyasi qanday vazifani bajaradi?
4. AVG agregat funksiyasi qanday vazifani bajaradi?
5. MAX agregat funksiyasi qanday vazifani bajaradi?
6. MIN agregat funksiyasi qanday vazifani bajaradi?
7. SQL tilidagi matematik funksiyalarga misol keltiring.
8. SQL tilidagi qatorli funksiyalarga misol keltiring.
9. SQL tilidagi sana-vaqt funksiyalariga misol keltiring.
10. SQL tilidagi tizimni konfiguratsiya funksiyalariga misol keltiring.

### **Adabiyotlar**

1. Murodov O'.M. SQL texnologiyalar fanidan maruzalar matni. TATU Samarqand filiali, 2012.
2. Кириллов В.В. Основы проектирования реляционных баз данных. Учебное пособие. СПб.ИТМО,1994.
3. Дейт К. Введение в системы баз данных. -М.Наука,1980 г.
4. Кузнецов С.Д. Введение в стандарты языка баз данных SQL.-М. 1998г.

### **Internet ma'lumotlari**

1. [www.sql.ru](http://www.sql.ru)

## 5-Mavzu: SQL tilida protseduralar va ulardan foydalanish

Reja:

1. Protседuralar to'g'risida umumiy tushunchalar.
2. Foydalanuvchilar va ularning imtiyozlari.

### 1. Protседuralar to'g'risida umumiy tushunchalar.

SQL tilida protseduralardan foydalanish dasturlar tuzish samaradorligini oshiradi. Saqlanuvchi protseduralar (stored procedure) – bu SQL buyruqlar to'plamidan iborat bo'lib, bu buyruqlar to'plamini SQL SERVER bir marta kompilyatsiya qiladi. Protседuralarning keyingi ishlatilishida saqlangan protseduralar kompilyatsiya qilinmaydi. Bu protseduralar xuddi algoritmik tillardagi kabi kirish parametrlaridan iborat bo'lishi ham mumkin.

Saqlanuvchi protseduralar SQL tilida quyidagi buyruq yordamida yaratiladi:

```
CREATE PROCEDURE <protsedura nomi>  
    [(% birinchi parametr ma'lumoti turi)] ...]  
AS SQL-operatorlari;
```

Saqlanuvchi protseduralarning ikki turi mavjud: foydalanuvchi protseduralari va tizimli protseduralar.

Foydalanuvchi protseduralari SQL SERVERlarida qo'llanilib, serverni boshqarish, MB va foydalanuvchilar haqidagi ma'lumotlarni olish uchun ishlatiladi.

Tizimli protseduralar esa, amaliy dasturlarni bajarish uchun yaratiladi. Amaliy dasturlar hech bo'lmaganda bitta modulni o'zida saqlashi kerak. Modul (MODULE) biror bir algoritmik tilda tuzilgan, uzoq muddat saqlanadigan ob'yektdir.

Modul - modul nomidan (module name), algoritmik til bo'limidan (language clause), modul bo'limi huquqidan (module authorization clause), kursorlarni tavsiflash (declare cursor) va bir yoki bir nechta protsedura (procedure) lardan tashkil topadi.

Modul sintaksisi quyidagicha:

```
<Module> ::= <module name clause>  
            <language clause>  
            <module authorization clause>  
            [<declare cursor>...]  
            < procedure > ...
```

```
<language clause> ::= LANGUAGE { COBOL | FORTRAN | PASCAL | PLI }
```

```
<module authorization clause> ::= AUTHORIZATION <module authorization  
                                identifier>
```

```
<module authorization identifier> ::= <authorization identifier>;
```

Modullarni yaratishda quyidagi sintaktik qoidalarga rioya qilish lozim bo'ladi:

Har bir aniqlangan kursorda (cursor declare) hech bo'lmaganda bitta modul (module) va bu modulda hech bo'lmaganda bitta protsedura (procedure) mavjud bo'lishi kerak, hamda bu protsedura ochish operatori (open statement) va tavsiflashda (cursor declare) e'lon qilinadigan kursor nomini (cursor name) o'zida aks ettishi ozim. Amaliy dastur bittadan ortig modul bilan ishlamasligi kerak.

Protsedura o'z navbatida potsedura nomidan (procedure name), parametrlar tavsifi (parameters declaration) va hech bo'lmaganda bitta SQL operatoridan (SQL statement) tashkil topadi.

Moduldan tashkil topgan amaliy dastur potseduraga murojaat qilish uchun CALL operatoridan foydalanadi. CALL operatori potsedura nomidan (procedure name), parametr qiymatalri ketma-ketligidan, son va ma'lumotlar turidan iborat. Protseduraga murojaat protsedurada mavjud bo'lgan SQL operatorlarini bajarishni ta'minlaydi.

SQL tilida protseduralar quyidagicha yaratiladi.

```
<procedure> ::=PROCEDURE <procedure name>  
                <parameter declaration>...  
                <SQL statement>;
```

Bu erda,

```
<parameter declaration>::=<parameter name> <data type>  
                        | <SQLCODE parameter>
```

```
<SQLCODE parameter> ::=SQLCODE
```

```
<SQL statement> ::=<close statement>  
                | <commit statement>  
                | <delete statement positioned>  
                | <delete statement searched>  
                | <fetch statement>  
                | <insert statement>  
                | <open statement>  
                | <rollback statement>  
                | <select statement>  
                | <update statement positioned>  
                | <update statement searched>
```

Protseduralarni yaratishda quyidagi sintaktik qoidalarga amal qilish lozim:

1. Protsedura nomi modulda ishtirok etadigan boshqa protsedura nomlaridan farq qilishi lozim.
2. Protsedura parametrlari ham boshqa protsedura parametrlaridan farq qilishi lozim.
3. Har bir parametr nomi (parameter declaration) tavsifida ko'rsatilgan bo'lishi ozim.
4. Agar SQL operatoridagi ustun nomi (column names) (parametr declation) tavsifida ko'rsatilgan parametr nomi bilan mos tushsa, bunday ustun nomlari (column specification) oldiga kvalifikator (qualifier) qo'yiladi.
5. Til bo'limida (language clause) modulda ishlatiladigan algoritmik til nomi ko'rsatiladi. Har bir algoritmik tilni ishlatishda o'ziga xos qoidalarga rioya qilishga to'g'ri keladi. Masalan til bo'limida Pascal tili ishlatilsa, unda:

1) SQLCODE parametrining turi INTEGER bo'lishi kerak;

- 2) Har qanday ishlatiladigan ma'lumot turlari (data type) CHAR, INTEGER va REAL bo'lishi talab qilinadi;
- 3) Agar (parameters declaration) tavsifida berilgan parametr turi (data type) INTEGER yoki REAL bo'lsa, shu parametrlarga mos keluvchi parametr turi ham INTEGER yoki REAL bo'lishi kerak.

Tizimli protseduraga misol keltiramiz. MBdagi detallarni hajm jihatidan katta kichikligiga qarab Detallar jadvalidan izlash lozim bo'lsin. Buni quyidagi rekursiv protsedura yordamida amalga oshiramiz.

```

GET LIST(Kiritilayotgan_detal);
CALL RECURSION(Kiritilayotgan_detal);
RETURN;
RECURSION: PROC(Katta_detal) RECURSIVE;
    DCL Katta_detal CHAR(30);
    DCL Kichik_detal CHAR(30);
EXEC SQL DECLARE C KURSOR FOR
    SELECT Detal_raqami FROM Detallar
    WHERE Asosiy_detal=Katta_detal
    AND Detal_raqami>Kichik_detal
    ORDER BY Asosiy_detal;
EXEC SQL CLOSE C;
CALL RECURSION (Kichik_detal);
END;
END;
```

## 2. Foydalanuvchilar va ularning imtiyozlari.

SQL muxitida har bir foydalanuvchi maxsus identifikatsion nom, murojlat identifikatoriga (ID) ega. Ma'lumotlar bazasiga yuborilgan komanda ma'lum foydalanuvchi bilan yoki boshqacha aytganda maxsus murojaat identifikatori bilan bog'lanadi. SQL ma'lumotlar bazasida ID ruxsat – bu foydalanuvchi nomi va SQL komanda bilan bog'langan murojaat identifikatoriga ilova qiluvchi maxsus kalit so'z USER dan foydalanishi mumkin.

Registratsiya bu kompyuter tizimiga kirish huquqini olish uchun foydalanuvchi bajarishi kerak bo'lgan protseduradir. Bu protsedura foydalanuvchi bilan qaysi murojaat ID si bog'lanishini aniqlaydi. Odatda har bir ma'lumotlar bazasidan foydalanuvchi o'zining ID sigaga ega bo'lishi kerak va registratsiya jarayonida haqiqiy foydalanuvchiga aylanadi. Lekin ko'p masalalarga ega foydalanuvchilar bir necha murojaat ID lari bilan registratsiyadan o'tishlari 1ki bir necha foydalanuvchi bitta murojaat ID sidan foydalanishlari mumkin.

**Imtiyozlar-**Har bir foydalanuvchi SQL ma'lumotlar bazasida nima qilish mumkinligini ko'rsatuvchi imtiyozlarga egadir. Bu imtiyozlar vaqt o'tishi bilan o'zgarishi Ya'ni eskilari o'chirilib Yangilari qo'shilishi mumkin. SQL imtiyozlar bu ob'ekt imtiyozlaridir. Bu shuni bildiradiki foydalanuvchi berilgan komandani ma'lumotlar bazasining biror ob'ekti ustida bajarishi mumkin. Ob'ekt imtiyozlari bir vaqtning o'zida foydalanuvchilar va jadvallar bilan bog'liq. Ya'ni imtiyoz ma'lum

foydalanuvchiga ko'rsatilgan jadvalda, asos jadvalda yoki tasavvurda beriladi. Ixtiyoriy turdagi jadvalni yaratgan foydalanuvchi shu jadval egasidir. Bush uni bildiradiki foydalanuvchi bu jadvalda hamma imtiyozlarga ega va imtiyozlarini shu jadvalning boshqa foydalanuvchilariga uzatishi mumkin.

Foydalanuvchiga tayinlash mumkin bo'lgan imtiyozlar:

- **SELECT** Bu imtiyozga ega foydalanuvchi jadvallarda so'rovlar bajarishi mumkin.

- **INSERT** Bu imtiyozga ega foydalanuvchi jadvalda INSERT komandasini bajarishi mumkin.

- **UPDATE** Bu imtiyozga ega foydalanuvchi jadvalda UPDATE komandasini bajarishi mumkin. Bu imtiyozni jadvalning ayrim ustunlari uchun cheklab qo'yishingiz mumkin.

- **DELETE** Bu imtiyozga ega foydalanuvchi jadvalda DELETE komandasini bajarishi mumkin.

- **REFERENCES** Bu imtiyozga ega foydalanuvchi jadvalning ustunidan (yoki ustunlaridan) ajdod kalit sifatida foydalanuvchi tashqi kalit aniqlashi mumkin. Siz bu imtiyozni ayrim ustunlar uchun berishingiz mumkin.

Bundan tashqari siz ob'ekt nostandart imtiyozlarini uchratasiz, masalan **INDEX (INDEKS)** –jadvalda indeks yaratish huquqini beruvchi, **SYNONYM (SINONIM)**- ob'ekt uchun sinonim yaratish huquqini beruvchi va **ALTER (IZMENIT)**- jadvalda ALTER TABLE komandasini bajarish xquqini beruvchi. SQL Mexanizm foydalanuvchilarga bu imtiyozlarni GRANT komandasi yordamida beradi.

**GRANT Komandasi-** GRANT komandasining 4 formati mavjud bo'lib, ulardan biri konkret ob'ekt ustidan, konkret foydalanuvchilarga konkret imtyozlar berish bo'lib, quyidagi ko'rinishga ega:

```
GRANT privilege ON [creator.]tablename TO userid, ... [WITH GRANT OPTION]
```

Bu erda

-privilege – tayinlanayotgan imtiyozlar ro'yxati,

-tablename - jadval nomi,

-userid – imtyozlar olgan foydalanuvchilar ro'yxati.

Masalan: GRANT SELECT, INSERT ON Orders TO Adrian, Diane;

Ma'lum foydalanuvchilarga imtiyozlarni SQL Central da ikki usul Bilan tayinlash mumkin. Birinchidan Users & Groups papkasini tanlash va ma'lum foydalanuvchi xossalarini ro'yxatdan chaqirish (sichqoncha o'ng klavishasini bosish va menyu Properties punktini tanlash). So'ngra Permissions qo'shimcha sahifasida

kerakli jadvalni tanlab imtiyozni o'rnatish. Ikkinchidan Tables yoki Views papkasida ma'lum jadval yoki tasavvur xossalar oynasini chaqirish, so'ngra Permissions qo'shimcha sahifasiga o'tish va GRANT tugmasi yordamida kerakli foydalanuvchini tanlab, imtiyozni o'rnatish.

GRANT UPDATE (City, Comm) ON Salespeople TO Diane; - bu Diane ga Salepeople jadvalining City va Comm ustunlari qiymatlarini o'zgartirish huquqini beradi yoki GRANT REFERENCES (CName, CNum) ON Customers TO Stephen; - bu komanda Stephen ga CNum va CName ustunlarini o'zining jadvallaridagi ixtiyoriy tashqi kalitlarga nisbatan ajdod kalit sifatida ishlatish huquqini beradi. Stephen (CName, CNum) yoki (CNum, CName) usutunlarni, jadvalarining ikki ustuni bilan tashqi kalit yordamida mos kelgan ikki -ustunli ajdod kalit sifatida aniqlashi mumkin. YOki u maydonga individual murojaat qilish uchun ajratilgan tashqi kalitlar yaratishi mumkin.

### **Uyda bajarish uchun topshiriqlar**

1. SQL tilidagi protseduralarga tavsif bering?
2. Foydalanuvchi va tizim protseduralarining farqi nimada?
3. Protseduralarning formal va haqiqiy parametrlari deganda nimani tushunasiz?

### **Adabiyotlar**

1. Murodov O'.M. SQL texnologiyalar fanidan maruzalar matni. TATU Samarqand filiali, 2012.
2. Кириллов В.В. Основы проектирования реляционных баз данных. Учебное пособие. СПб.ИТМО,1994.
3. Дейт К. Введение в системы баз данных. -М.Наука,1980 г.
4. Кузнецов С.Д. Введение в стандарты языка баз данных SQL.-М., 1998г.
5. Шкарина Л. Язык SQL. Учебный курс. Санкт-Петербург. 2001.

### **Internet ma'lumotlari**

1. [www.sql.ru](http://www.sql.ru)
2. [www.mysql.ru](http://www.mysql.ru)
3. [www.firststeps.ru/sql](http://www.firststeps.ru/sql)



## 6-Mavzu. SQL tilida jadvallarni loyihalashtirish. Jadvallarni yaratish, o'zgartirish va o'chirish. Jadvallar haqida ma'lumot olish

1. Jadvallarni loyihalashtirish.
2. SQL tilida maydon qiymatlarini tekshirish (CHECK cheklanishi).

Jadvallar CREATE TABLE komandasi bilan yaratiladi. Bu komanda qatorlarsiz bo'sh jadval yaratadi. CREATE TABLE komandasi jadval nomini va jadval o'zini ma'lum tartibda ko'rsatilgan ustunlar nomlari ketma – ketligi ta'rifi ko'rinishida aniqlaydi. U ma'lumotlar tiplari va ustunlar o'lchovini aniqlaydi. Har bir jadval juda 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.

Bundan buyon quyida keltirilgan 3 ta jadvaldan iborat ma'lumotlar bazasini ko'ramiz.

Jadval Sotuvchilar (Salepeople):

Snu	Sname	City	Comm	SNum	– har bir sotuvchi unikal nomeri, m
11	Alisher	Jomboy	0.12		SName – sotuvchi nomi, City – sotuvchi adresi ( shahar ),
12	Rustam	Urgut	0.13		Comm – sotuvchilarning o'nli shakldagi komission foydasi.
14	Nodir	Toyloq	0.11		

Jadval Buyurtmachilar (Customers):

Cnum	Cname	City	Rating	Num	CNum – har bir buyurtmachi unikal nomeri,
21	Shuxrat	Chelak	100	11	CName – buyurtmachi nomi, City – buyurtmachi adresi ( shahar ),
22	G'ayrat	Toyloq	200	13	
23	Ulug'bek	Urgut	200	12	

Rating – buyurtmachining boshqalardan ustunlik darajasini ko'rsatuvchi kod ( reyting ),

SNum – shu buyurtmachiga tayinlangan sotuvchi nomeri.

### Jadval Buyurtma (Orders):

Onum	AMT	Odate	CNum	Snum	ONum – har bir sotib olish unikal nomeri, AMT – sotib olish summasi qiymati, ODate – sotib olish sanasi, CNum – sotib oluvchi buyurtmachi nomeri, SNum – sotuvchining nomeri.
38	4723 .00	1990/10/05	26	11	
310	1309 .95	1990/10/06	24	12	

Misol uchun sotuvchilar jadvalini yaratishni ko‘rib chiqamiz:

#### **CREATE TABLE Salepeople**

**( SNum integer,  
SName char (10),  
City char (10),  
Comm decimal );**

Jadvallarni o‘chirish- Jadvalni o‘chirish imkoniga ega bo‘lish uchun, jadval egasi (Ya’ni yaratuvchisi) bo‘lishingiz kerak. Faqat bo‘sh jadvalni o‘chirish mumkin. Qatorlarga ega bo‘lgan, to‘ldirilgan jadvalni o‘chirish mumkin emas, Ya’ni jadval o‘chirishdan oldin tozalangan bo‘lishi kerak. Jadvalni o‘chirish komandasi quyidagi ko‘rinishga ega:

**DROP TABLE < table name >;**

Masalan: **DROP TABLE Salepeople;**

Jadvalni yaratilgandan so‘ng o‘zgartirish- Jadvalni o‘zgartirish uchun ALTER TABLE komandasidan foydalaniladi. Bu komanda jadvalga Yangi ustunlar qo‘shish, ustunlarni o‘chirish, ustunlar kattaligini o‘zgartirish, hamda cheklanishlarni qo‘shish va olib tashlash imkoniyatlariga ega. Bu komanda ANSI standarti qismi emas, shuning uchun har xil tizimlarda har xil imkoniyatlarga ega.

Jadvalga ustun qo‘shish uchun komandaning tipik sintaksisi:

**ALTER TABLE <table name> ADD <column name>  
<data type> <size>;**

**Masalan:**

**ALTER TABLE Salepeople ADD Phone CHAR(7);**

Jadvallar uchun cheklanishlar- jadval yaratayotganingizda (yoki uni o‘zgartirayotganingizda), siz maydonlarga kiritilayotgan qiymatlarga cheklanishlar o‘rnatishingiz mumkin. Bu xolda SQL cheklanishlarga to‘g‘ri kelmaydigan hamma qiymatlarni rad etadi. Cheklanishlar ikki asosiy turi mavjud: - ustun va jadval cheklanishlari. Ularning farqi shundaki ustun cheklanishi stolbtsa faqat ayrim ustunlarga qo‘llanadi, jadval cheklanishi bo‘lsa bir yoki bir necha ustunlar guruxiga qo‘llanadi. Ustun cheklanishi ustun nomi oxiriga ma’lumotlar tipidan so‘ng va

verguldan oldin qo'yiladi. Jadval cheklanishi jadval nomi oxiriga so'nggi dumaloq verguldan oldin qo'yiladi. Cheklanishlar hisobga olingan CREATE TABLE komandasi sintaksisi:

```
CREATE TABLE < table name >  
  ( <column name> <data type> <column constraint>,  
    <column name> <data type> <column constraint> ...  
    <table constraint> ( <column name>  
      [, <column name> ]... );
```

Maydonga bo'sh (NULL) qiymatlar kiritilishi oldini olish uchun CREATE TABLE komandasida NOT NULL cheklanishi ishlatiladi. Bu cheklanish faqat har xil ustunlar uchun o'rnatiladi.

Masalan, shu narsa aniqki, birlamchi kalitlar hech qachon bo'sh bo'lmasliklari kerak, shuning uchun Salepeople jadvalini quyidagicha yaratish mumkin:

```
CREATE TABLE Salepeople  
  ( Snum   integer NOT NULL,  
    Sname  char (10),  
    city   char (10),  
    comm   decimal);
```

Ko'p xollarda ustunga kiritilgan qiymatlar bir biridan farq qilishi kerak. Agar ustun uchun UNIQUE cheklanishi o'rnatilsa, bu ustungsha mavjud qiymatni kiritishga urinish rad etilapdi. Bu cheklanish bo'sh bo'lmaydigan (NOT NULL) debe'lon qilingan maydonlarga qo'llanishi mumkin.

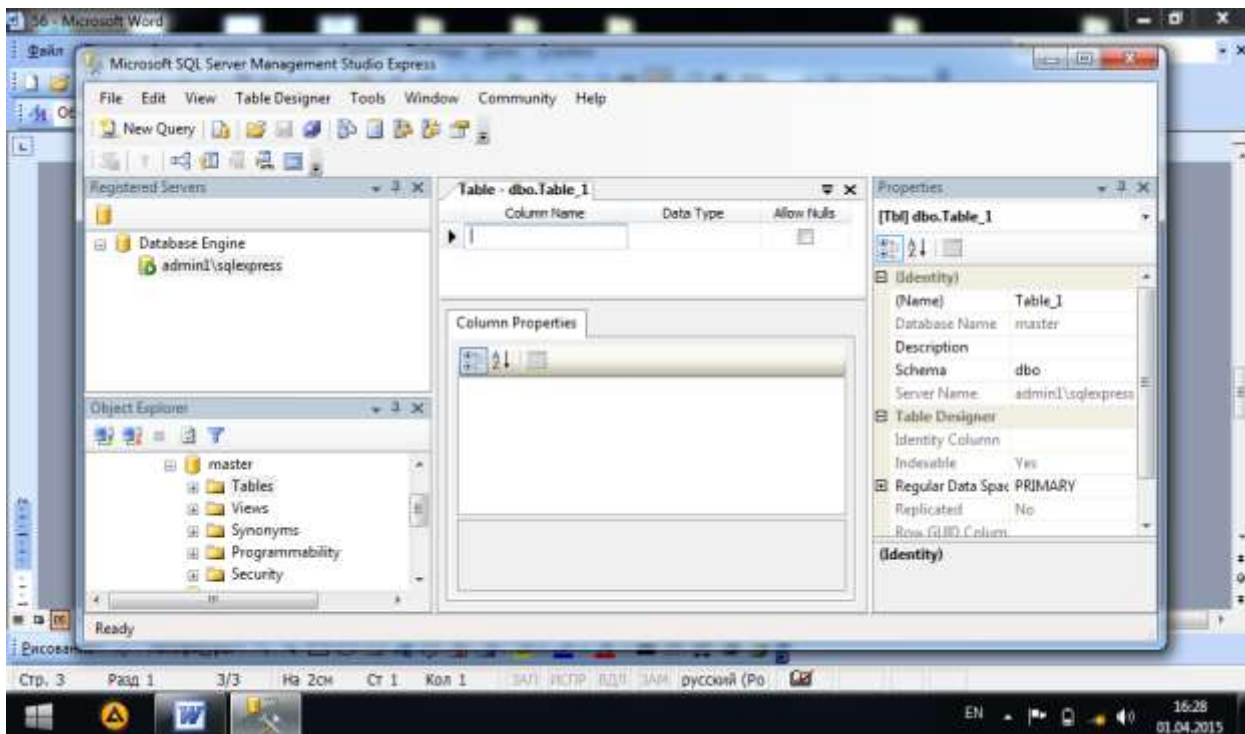
Masalan:

```
CREATE TABLE Salepeople  
  ( SNum   integer NOT NULL UNIQUE,  
    SName  char (10),  
    City   char (10),  
    Comm   decimal);
```

Unikalligi talab qilinadigan maydonlar(birlamchi kalitlardan tashqari) kandidat kalitlar yoki unikal kalitlar deyiladi.

Jadval cheklanishi UNIQUE maydonlar guruxiga o'rnatilishi mumkin. Bu bir necha maydonlar qiymatlari kombinatsiyasi unikalligini ta'minlaydi. Bixning ma'lumotlart bazamizda har bir buyurtmachi bita sotuvchiga biriktirilgan. Ya'ni Buyurtmachilar jadvalida buyurtmachi nomeri (cnum) va sotuvchi nomeri (snum) kombinatsiyasi unikal bo'lishi kerak. Bu cheklanishni UNIQUE (cnum, snum)

yordamida, Customers jadvalini yaratishda kiritish mumkin. Bu ustunlar uchun NOT NULL cheklanishini kiritish zarurdir.



15-rasm. SQL tilida jadval tuzish

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, faqat jadval uchun faqat bitta birlamchi kalit (ixtiyoriy sondagi ustunlar uchun ) aniqlanishi mumkin bo‘lgan xoldan tashqari. Birlamchi kalitlar NULL qiymatga ega bo‘lishi mumkin emas.

Misol:

```
CREATE TABLE Salepeople
```

```
( SNum integer NOT NULL PRIMARY KEY,
```

```
  SName char (10),
```

```
  City char (10),
```

```
  Comm decimal);
```

PRIMARY KEY cheklanishi qiymatlar unikal kombinatsiyasini tashkil qiluvchi bir necha maydonlar uchun qo‘llanishi mumkin. Masalan PRIMARY KEY cheklanishini juftliklar uchun qo‘llash mumkin:

```
CREATE TABLE Namefield
```

```
( firstname char (10) NOT NULL,
```

```
  lastname char (10) NOT NULL
```

```
  city char (10),
```

```
  PRIMARY KEY ( firstname, lastname ));
```

## 2. SQL tilida maydon qiymatlarini tekshirish (CHECK cheklanishi).

Maydon qiymatlarini tekshirish (CHECK cheklanishi)- CHECK cheklanishi jadvalga kiritilayotgan ma'lumot qabul qilinishidan oldin mos kelishi lozim bo'lgan shart kiritishga imkon beradi. CHECK cheklanishi CHECK kalit so'zi ko'rsatilgan maydondan foydalanuvchi predikat ifodapdan iboratdir.

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 ));
```

CHECK cheklanishidan maydonga ma'lum qiymatlarini kiritishdan ximoya qilib, xatolar oldini olish uchun foydalanish mumkin. Masalan maxsulotni sotish shaxobchalariga ega bo'lgan shaharlar faqat London, Barselona, San Xose va Nu York bo'lsin.

```
CREATE TABLE Salepeople
```

```
( SNum integer NOT NULL PRIMARY KEY,  
SName char(10) NOT NULL UNIQUE,  
City char(10) CHECK (City IN ('London', 'New York', 'San Jose',  
'Barselona'))),  
Comm decimal CHECK ( Comm < 1 ));
```

CHECK jadval cheklanishi sifatida kelishi mumkin. Bu shartga bir necha maydon kiritishga imkon beradi.

Masalan:

```
CREATE TABLE Salepeople
```

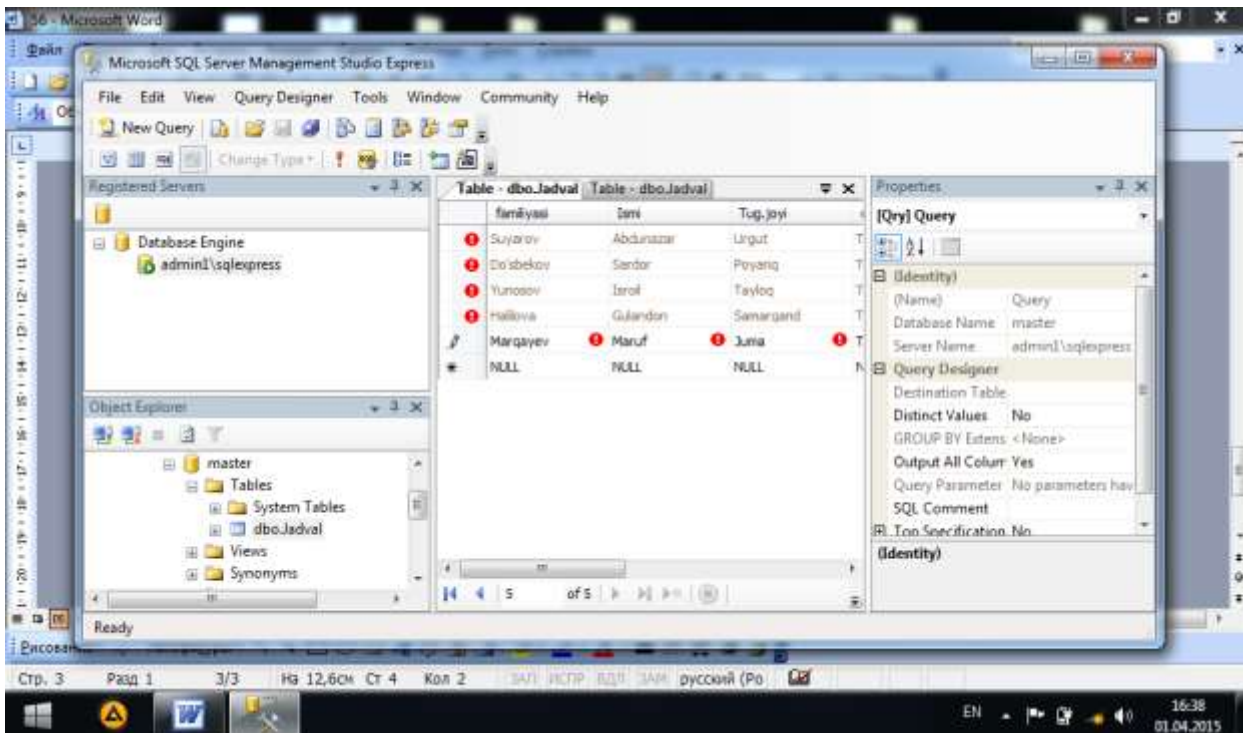
```
( SNum integer NOT NULL PRIMARY KEY,  
SName char(10) NOT NULL UNIQUE,  
City char(10),  
Comm decimal,  
CHECK (Somm < .15 OR City = 'Barcelona'));
```

Ko'zda tutilgan qiymatlarni o'rnatish-biror bir maydon uchun qiymat ko'rsatmagan xolda jadvalga satr qo'shsangiz, SQL bunday maydonga kiritish uchun ko'zda tutilgan qiymatga ega bo'lishi kerak, aks xolda komanda rad etiladi. Eng

umumiy koʻzda tutilgan qiymat NULL qiymatdir. CREATE TABLE komandasida koʻzda tutilgan qiymat DEFAULT operatori orqali, ustun cheklanishi sifatida koʻrsatiladi. 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 ));
```



16-rasm. SQL tili jadvalni toʻldirish

Maʼlumotlar yaxlitligini taʼminlash- jadval bir maydonidagi hamma qiymatlar boshqa jadval maydonida aks etsa, birinchi maydon ikkinchisiga ilova qiladi deyiladi. Bu ikki maydon orasidagi bogʻliqlikni koʻrsatadi. Masalan, buyurtmachilar jadvalida har bir buyurtmachi, sotuvchilar jadvalida oʻziga 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 bita maydondan iborat boʻlishi shart emas. Birlamchi kalit kabi, tashqi kalit bita modul sifatida qayta ishlanuvchi bir necha maydonlarga ega boʻlishi mumkin. Maydon tashqi kalit boʻlsa ilova qitlayotgan jadval Bilan maʼlumi usulda bogʻliqdir. Tashqi kalit har bir qiymati (satri), ajdod kalitning bita va faqat bitta qiymatiga( satriga) ilova qilishi kerak. Bu xolda tizim ilovali yaxlit xolatda deyiladi

Shu bilan birga ajdod kalit qiymati tashqi kalit 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 kalitmaydonlariga kiritmaslikdir. FOREIGN KEY cheklanishi sintaksisi:

### **FOREIGN KEY <column list> REFERENCES**

**<pktable> [<column list>]**

Birinchi ro'yxat komanda 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) ); \***

### **Uyda bajarish uchun topshiriqlar**

1. SQL tilida jadval tuzing.
2. SQL tilida jadvalning loyihasini tuzing.
3. SQL tilida maydon tushunchasi.

### **Adabiyotlar**

1. Murodov O'.M. SQL texnologiyalar fanidan maruzalar matni. TATU Samarqand filiali, 2012.
2. Кириллов В.В. Основы проектирования реляционных баз данных. Учебное пособие. СПб.ИТМО,1994.
3. Дейт К. Введение в системы баз данных. -М.Наука,1980 г.
4. Кузнецов С.Д. Введение в стандарты языка баз данных SQL.-М., 1998г.
5. Шкарина Л. Язык SQL. Учебный курс. Санкт-Петербург. 2001.

### **Internet ma'lumotlari**

1. [www.sql.ru](http://www.sql.ru)
2. [www.mysql.ru](http://www.mysql.ru)
3. [www.firtsteps.ru/sql](http://www.firtsteps.ru/sql)

7-Mavzu. **SQL tilida ma'lumotlarni boshqarish. Bazadan ma'lumotlarni olish. Ma'lumotlarni qo'shish, o'zgartirish va o'chirish**

1. SQL tili yordamida ma'lumotlarni boshqarish.
2. Murakkab so'rovlar yordamida MB dan ma'lumotlarni olish.

SQL tilining asosiy xususiyatlaridan biri shundaki, u bir necha jadvallarni o'zaro bog'lay oladi va ular ichidan kerakli malumotlarni izlab topadi. Bu ko'rinishdagi amallarni jadvallarni birlashtirish amallari deyiladi. Birlashtirish amallari yordamida bir nechta jadvalardan bitta so'rov yordamida ma'lumotlarni olish imkoniyatiga ega. Buning uchun jadval nomlari va uning ustunlaridan foydalaniladi.

Umuman olganda MB jadvalining nomi jadval nomi, nuqta va ustun nomidan tashkil topadi. Masalan:

Salespeople.snum

Salespeople.city

Orders.odate

Jadvallarni birlashtirish uchun ham SELECT operatori yordamida amalga oshiriladi. Masalan: bir shaxarda yashovchi sotuvchilar va buyurtmachilarning o'zaro mosligini ta'minlash lozim bo'lsin. Buni quyidagi so'rovlar asosida tashkil qilish mumkin.

```
SELECT Customers.cname, Salespeople.sname, Salespeople.city
FROM Salespeople, Customers
WHERE Salespeople.city = Customers.city
```

<b>cname</b>	<b>sname</b>	<b>city</b>
Shavkat	Maxmud	Samarqand
Rustam	Nodir	Jizzax
Bobur	Anvar	Buxoro
Ulug'bek	G'ulom	Xorazm
Mashrab	Azizbek	Tayloq
Jasur	Sadulla	Urgut

Bitta sotuvchi xizmat ko'rsatadigan hamma buyurtmachilarni bilish uchun quyidagi so'rovni bajarish mumkin.

```
SELECT Customers.cname, Salespeople.sname
FROM Customers, Salespeople
WHERE Salespeople.snum= Customers.snum;
```

<b>cname</b>	<b>sname</b>
Shavkat	Maxmud
Mashrab	G'ulom
Bobur	Azizbek
Ulug'bek	Nodir
Rustam	Anvar
Jasur	Sadulla



Yuqoridagi ko'rib o'tilgan misollarda ishlatilgan predikatlar tenglik amali yordamida bajariladi va shuning uchun ular o'zaro tenglik birlashtirishlari deb yuritiladi. Umuman olganda ma'lumotlarni birlashtirishda istalgan relyatsion amallardan foydalanish mumkin. Masalan: bir shaharda yashovchi barcha buyurtmachilarni ularga xizmat ko'rsatuvchi, retingi 200 dan kam bo'lgan sotuvchilar bilan birgalikda hamda bu sotuvchi nomlarini alfavit tartibida matnga chiqarish talab etilsin. Buni quyidagi so'rov asosida amalga oshiriladi.

```
SELECT sname, cname FROM Salespeople, Customers
WHERE sname<cname AND rating<200;
```

<b>sname</b>	<b>cname</b>
Maxmud	Shavkat
G'ulom	Mashrab
Azizbek	Bobur
Nodir	Ulug'bek
Anvar	Rustam
Sadulla	Jasur

Jadvaldagi ustunlardagi ma'lumotlar tartiblanmagan ma'lumot to'plamidan iborat. SQL tilida jadval qiymatlarini tartiblangan holda chop etish imkoniyatlari mavjud. Buning uchun ORDER BY predikatida ASC (o'sish) va DESC (kamayish) kalit so'zlari ishlatiladi. Masalan: buyurtmachining tartiblangan raqamlari (cnum) orqali buyurtmalarni qabul qilish jarayonini quyidagicha tasvirlash mumkin.

```
SELECT * FROM Orders
ORDER BY cnum DESC;
```

<b>onum</b>	<b>amt</b>	<b>odate</b>	<b>cnum</b>	<b>snum</b>
3001	10600.60	10/03/2014	2008	1007
3006	12500.20	21/03/2014	2007	1008
3003	25000.25	05/01/2015	2006	1009
3008	60200.80	09/02/2015	2004	1010
3009	45000.40	20/03/2015	2009	1003
3011	54000.70	24/03/2015	2001	1002

Ikki va indan ortiq jadvallarni bitta so'rov operatorida birlashtirib jadval hosil qilinadi. Ayrim hollarda jadvallarning o'zini o'zi bilan birlashtirishga to'g'ri keladi.

## 2. Murakkab so'rovlar yordamida MB dan ma'lumotlarni olish.

SQL tilidagi so'rovlar boshqa so'rovlarni ham boshqarish imkoniyatiga ega. Bunday boshqarishlar quyi so'rovlar yordamida amalga oshiriladi, yani so'rovlarni ichma-ich joylashtirish mumkin. Bunday so'rovlarda ichki so'rovlar bitta qiymatni generatsiya qilib, bu qiymatlar tashqi so'rov tekshirishlarida ishtirok etadi. Masalan: sotuvchining ismini bilamiz (Anvar), ammo uning identifikatsion raqamini (snum) ni bilmaymiz bunda Anvarning barcha buyurtmalari matnga chiqarishimiz kerak bo'ladi. Buni quyida so'rov orqali amalga oshiriladi:

```
SELECT * FROM orders
      WHERE snum= (SELECT snum FROM Salespeople
                  WHERE sname = 'Anvar');
```

Onum	Amt	Odate	Cnum	Snum
3002	20600.20	10/03/2015	2007	1004
3005	28900.70	18/03/2015	2009	1006
3007	36800.80	25/03/2015	2012	1009

Ayrim hollarda quyi so'rovlar bitta qiymatni berishi uchun DISTINCT kalit so'zidan foydalanish mumkin. Masalan: Jasur (cnum=2001) ga xizmat ko'rsatuvchi sotuvchini va unga xarid tartiblarini aniqlashi zarur bo'lsin. Unda quyidagi so'rov amalga oshiriladi.

```
SELECT * FROM Orders
      WHERE snum = (SELECT DISTINCT snum FROM Orders)
      WHERE cnum = 2001;
```

Onum	Amt	Odate	Cnum	Snum
3003	15600.20	10/01/2015	2001	1001
3008	17800.20	10/02/2015	2001	1001
3011	19500.50	10/03/2015	2001	1001
3012	29400.40	10/03/2015	2001	1001

SQL tilidagi agregat funksiyalar bir qiymatli miqdorlarni hisoblaganliklari uchun quyi so'rovlarda ishlash mumkin: masalan, 4-oktyabrdagi xarid qilingan va bahosi o'rtacha summadan yuqori bo'lgan mahsulotlarni ko'rish uchun quyidagi so'rovlar bajariladi.

```
SELECT * FROM Orders
      WHERE amt > (SELECT AVG(amt) FROM Orders)
      WHERE odate = 10/03/2015
```

Quyida so'rovlarni tashkilshtirishda maxsus operatorlardan (IN, BETWEEN, LIKE, IS NULL) dan ham foydalanish mumkin.

<b>Onum</b>	<b>Amt</b>	<b>Odate</b>	<b>Cnum</b>	<b>Snum</b>
3002	15600.20	10/04/2015	2007	1004
3005	17800.20	10/04/2015	2003	1002
3006	19500.50	10/04/2015	2008	1007
3009	29400.40	10/04/2015	2002	1003
3008	24600.10	10/04/2015	2006	1001
3010	23900.50	10/04/2015	2004	1002
3011	29200.50	10/04/2015	2005	1006
3012	31300.80	10/04/2015	2009	1007

Quyida so'rovlarni tashkilshtirishda ifodalardan ham foydalanamiz. Masalan, cnum maydoni miqdori 1000 va snum Serres maydoni qiymati miqdoridan yuqori bo'lgan barcha buyurtmachilarning ro'yxati matnga chiqarilsin:

```
SELECT * FROM Customers
WHERE cnum = (SELECT snum+1000 FROM Salespeople
WHERE sname = 'Serres');
```

<b>cnum</b>	<b>cname</b>	<b>city</b>	<b>rating</b>	<b>snum</b>
2002	Alisher	Samarqand	200	1003
2003	Rustam	Jizzax	300	1005
2005	G'anisher	Buxaro	200	1007

Hamma jadvallarda satr qiymatlari SQLda INSERT buyrug'i yordamida kiritiladi. INSERT buyrug'i quyidagi forematda bo'lishi mumkin.

```
INSERT INTO <table name/ view name>[(column [, column]...)]
VALUES (<value> [,<value>]...);
```

Masalan, sotuvchilar jadvaliga satr kiritish uchun quyidagi buyruqdan foydalanish mumkin.

```
INSERT INTO Salespeople VALUES (11, 'Peel', 'London', .12);
```

Bu buyruqlardan ko'rinib turibdiki, INTO kalit so'zi jadvalga qiymatlar kiritishda ishlatiladi.

## 8-Mavzu. SQL tilida tranzaksiyalar va blokirovkalar. Tranzaksiya va blokirovka tushunchasi. Aniq joylashtirilgan tranzaksiyalar

1. SQL muhitida tranzaksiya tushunchasi.
2. SQL muhitida tranzaksiyalarni boshqarish.

**SQL muhitida tranzaksiya tushunchasi.** SQL tilida tranzaksiya deb, ma'lumotlarni tiklashga nisbatan ajralmas bo'lgan operatorlar ketma-ketligiga aytiladi. SQL tilidagi har bir chaqirish moduli tranzaksiyadir. SQL tili tranzaksiyalari biror-bir modulning protseduralarini bajarishdan boshlanadi. COMMIT yoki ROLLBACK operatorining bajarilishi bilan tugaydi. Agar tranzaksiya ROLLBACK operatori bilan tugasa, protsedurada barcha qilingan amallar bekor qilinadi.

Har bir tranzaksiyaning "faqat o'qish" yoki "o'qish va yozish" tartiblari mavjud. Tranzaksiya tartiblari SETTRANSACTION operatori yordamida o'rnatiladi. Jimlik qoidasiga nisbatan "o'qish va yozish" tartibi o'rnatiladi. "Faqat o'qish" tartibi doimo saqlanadigan bazaviy ma'lumotlarga qo'llaniladi.

Har bir SQL tranzaksiyasi himoyalangan darajasiga ega: READUNCOMMITTED, READCOMMITTED, REPEATABLEREAD yoki SERIALIZABLE. SQL tranzaksiyasi himoyalangan darajalari bajarilayotgan tranzaksiyaning boshqa parallel bajarilayotgan tranzaksiyalarga ta'sir etish darajasini aniqlaydi. Tranzaksiyaning aniq darajasini o'rnatish uchun SETTRANSACTION operatoridan foydalaniladi. Jimlik qoidasiga nisbatan SERIALIZABLE tartibi o'rnatiladi.

Himoyalangan darajalari tranzaksiyalarning parallel bajarilishida yuz berishi mumkin bo'lgan hodisalarni aniqlaydi. Quyidagi ko'rinishdagi hodisalar bo'lishi mumkin:

*t*

1. P1 ("Dirtyread" - "Yomon o'qish"): T1 tranzaksiya qatoni yaratadi. Keyin T2 tranzaksiya T1 COMMIT amalini bajarmasdan bu qatoni o'qiydi. Shundan so'ng T1 ROLLBACK amalini bajarsa, T2 tranzaksiya umuman mavjud bo'lmagan qatoni o'qigan bo'lib chiqadi.
2. P2 ("Non-repeatableread" - "Takrorlanmagan o'qish"): T1 tranzaksiya qatoni o'qiydi. Shundan so'ng T2 tranzaksiya bu buyruqlar qatorini o'zgartiradi yoki olib tashlaydi va COMMITni bajaradi. Shundan so'ng T1 shu qatoni yana o'qishga harakat qiladi, ammo bu qator birinchi holatdagi qator emas yoki olib tashlangani uchun topolmaydi.
3. P3 ("Phantom" - "Fantom"): T1 tranzaksiya biror-bir shartni qanoatlantiradigan N qatoni o'qiydi. Shundan so'ng T2 tranzaksiya bu qatorlar ichidan bir yoki bir nechta qator shartlarini generatsiya qiladi. Agar shu ishlardan keyin T1 o'qishni qaytarsa, u butunlay boshqa qatorlarga ega bo'ladi,

To'rtala himoyalani sh daragalari P1,P2 va P3 hodisalarga nisbatan quyidagicha ta'sirga ega:

Himoyalani sh	PI	P2	P3
READUNCOM	Mumkin	Mumkin	Mumkin
READCOMMIT	Mumkin	Mumkin	Mumkin
REPEATABLE	Mumkin	Mumkin	Mumkin
SERIALIZABL	Mumkin	Mumkin	NjntfiEi

MB bilan ish jarayonida ma'lumotlar butligi muhim o'rin tutadi. Ma'lumotlar butligi deganimizda, ma'lumotlarning to'g'riligi va mazmunan qarma-qarshi ma'noga ega emasligi tushuniladi. Masalan, "O'qituvchi" jadvalidagi har bir o'zgarish "Yuklama" jadvalida ham qayd etilishi kerak. O'qituvchining "Yuklama" jadvalida qayd etilmasligi ma'lumotlar butligining buzilishiga olib keladi.

Ko'pchilik hollarada MBning ma'lumotlari butligini saqlashni tashkillashtirish uchun tranzaksiyalardan foydalaniladi.

Umuman olganda tranzaksiya - bu mantiqan bo'linmaydigan ish birligi.

Bu jarayonda:

- yoki tranzaksiyaga kiruvchi barcha amallar MBda aks etadi;
- yoki bu amallar umuman bajarilmaydi.

Ttranzaksiyaning bu xususiyati butlik shartining buzilmasligini ta'minlaydi.

Ko'pgina MBBTda tranzaksiyalarning ikkita modeli ishlatiladi:

1. Tranzaksiyalarning avtomatik bajarilish modeli.
2. Tranzaksiyalarning bajarilishini boshqarish modeli.

Ttranzaksiyalarning avtomatik bajarilish modelida, tranzaksiya avtomatik ravishda ishga tushadi va quyidagi usullardan biri bilan tugaydi:

- COMMIT yoTiqnomasi bilan, bunda MBdagi o'zgarishlar doimiy bajariladigan bo'ladi va yangi tranzaksiya COMMIT buyryg'idan so'ng boshlanadi.
- ROLLBACK yoTiqnomasi bilan, bunda tranzaksiyada bajarilgan barcha o'zgarishlar bekor bo'ladi va yangi tranzaksiya ROLLBACK buyryg'idan so'ng boshlanadi.

Ttranzaksiyalarning bajarilishini boshqarish modeli SUBD Sybase dasturida qo'llanilib, quyidagi yoTiqnomalardan foydalaniladi:

1. BEGIN TRANSACTION yoTiqnomasi, tranzaksiyaning boshlanishini bildiradi.
2. COMMIT TRANSACTION yoTiqnomasi, tranzaksiyaning muvaffaqiyatli tuganini bildiradi. Bunda yangi tranzaksiya avtomatik ravishda ishga tushmaydi.
3. SAVE TRANSACTION yoTiqnomasi, bunda tranzaksiya ichida saqlash nuqtasi tashkillashtiradi va saqlangan saqlash nuqtasiga nom berish imkoniyati yaratiladi.

ROLLBACK yoTiqnomasi, bunda barcha tranzaksiyadagi amallar bekor qilinadi va MB holati tranzaksiyadan oldingi holatga qaytariladi.

Shunday qilib, tranzaksiya - bu MBga tugallangan murojaat bo'lib quyidagi to'rtta shartning bajarilishini kafolatlaydi:

- *Bo'linmaslik* (atomarnost) - tranzaksiya boshi va oxiriga ega bo'lgan bo'linmas blok. Bu blok yoki to'liqligicha bajariladi, yoki umuman bajarilmaydi;
- *Kelishuvchanlik* - tranzaksiya tugaganidan so'ng, hamma ob'yektlar kelishganlik holatini saqlab qoladi;
- *Himoyalanganlik* - har bir tranzaksiya jarayoni boshqa tranzaksiya ishiga ta'sir ko'rsatmaydi.
- *Doimiylik* - tranzaksiya jarayonida bajarilgan barcha o'zgarishlar doimiylik xarakteruga ega.

### **SQL tilida tranzaksiya jaravoniga misol.**

SQL tilida har bir tranzaksiya bitta to'liq jarayonni bajarishiga tuhuncha holsil

qildik. Ana shunday jarayonga misol keltiramiz:

“Ta'rinlovchilar” jadvalidagi Sx raqamini Sy raqamiga o'zganirish lozim bo'lsin. Sx va Sy - berilgan aniq parametr.

```
TRANEX: PROC OPTIONS (MAIN); /* 7'tranzaksiyaga misol*/
EXEC SQL WNEVER SQLERROR GO TO UNDO;
GET LIST (SX,SY);
EXEC SQL UPDATE S
      SET TA'MINLOVCHI RAQAMI = SY WHERE
      TA'MINLOVCHI RAQAMI = SX;
EXEC SQL UPDATE SP
      SET TA'MINLOVCHI RAQAMI = SY WHERE
      TA'MINLOVCHI RAQAMI = SX;
EXEC SQL COMMIT;
      GO TO'FINISH;
UNDO: EXEC SQL ROLLBACK;
FINISH: RETURN;
END TRANEX;
```

Misolimizdan ko'rinib turibdiki, bu tranzaksiya jarayonida ikkita jadval ustuda o'zgarishlar amalga oshirilayapti. Demak, tranzaksiya deganimizda bitta amalni emas, balki amallar ketma-ketligini tushunish lozim.

### **2. SQL muhitida tranzaksiyalarni boshqarish.**

SQL tilida tranzaksiyalarni maxsus operatorlar yordamida boshqarish imkoniyati

mavjud. Shulardan biri tranzaksiya parametrlarini o'rnatish operatori bo'lib. uni

yo'zish formati quyidagicha:

```
<set transaction statement> ::=
  SET TRANSACTION transaction mode>
  [ { <comma> transaction mode> }... ]
```

transaction mode> ::=  
    <isolation level>  
    | <transaction access mode>  
    | «diagnostics size>

«isolation level» ::=  
    ISOLATION LEVEL «level of isolation»  
  
    <level of isolation> ::=  
    READ UNCOMMITTED | READ COMMITTED | REPEATABLE  
    READ | SERIALIZABLE

transaction access mode>  
    READ ONLY | READ WRITE

diagnostics size> ::=  
    DIAGNOSTICS SIZE <number of conditions>  
        <number of conditions> ::= <simple value specification>

Bu yerda:

1. Agar himoya darajalari ko'rsatilmasa, himoya darajasi SERIALIZABLE deb tushumniladi.
2. Agar ruxsat tartibi READWRITE kalit so'zi bilan belgilansa, unda himoyalaniş darajasi READUNCOMMITTED bo'lmasligi kerak.
3. Agar ruxsat tartibi va himoyalaniş darajasi READUNCOMMITTED deb ko'rsatilsa, unda beriladigan ruxsat tartibi READONLY deb tushuniladi, aks hollarda ruxsat tartibi READWRITE bo'ladi.

Eo'pchilik hollarda tranzaksiyalarning bajarilish jarayonida MB jadvallari stmkturasi buzilishining oldini olish uchun tranzaksiyalarga faqat o'qish tartibini o'matish mumkin. Buning uchun quyidagi operator ishlatiladi:

SET TRANSACTION READ ONLY;

Bu operator tranzaksiya jarayoni boshlanishidan oldin ko'rsatiladi.

Masalan, EXEC SQL SET TRANSACTION READ ONLY;

buyurtmalarni qabul qilgan sotuvchini aniqlamoqchisiz. Bu ikki so'rovni bir-biridan farq qilishi uchun matn kiritish yo'li bilan tashkillashtirishimiz mumkin:

```
SELECT a.snum, sname, onum, 'Highest on', odate FROM Salespeople a,  
      Orders b WHERE a.snum = b.snum  
      AND b.amt = (SELECT MAX (amt) FROM Orders c  
                  WHERE c.odate = b.odate)
```

UNION

```
SELECT a.snum, sname, onum, 'Lowest on', odate FROM Salespeople a,  
      Orders b WHERE a.snum = b.snum  
      AND b.amt = (SELECT MIN (amt) FROM Orders c  
                  WHERE c.odate = b.odate);
```

10	Peel	30	High	o	10/05/
10	Peel	30	Low	o	10/05/
10	Peel	30	High	o	10/06/
10	Serre	30	High	o	10/03/
10	Serre	30	Low	o	10/04/
10	Serre	30	Low	o	10/06/
10	Axel	30	High	o	10/04/
10	Rif	30	Low	o	10/03/

Shu paytgacha UNION kalit so'zi yordamida birlashtirilgan so'rov natijalari qanday tartibda matnga chiqarilishi to'g'risida gapirmadik. Birlashtirilgan natijalarni ORDER BY kalit so'zi yordamida tartiblashtirish mumkin. Yoqoridagi misolni tartub raqamlariga nisbatan tartiblashni ko'rib o'tamiz.

```
SELECT a.snum, sname, onum, 'Highest on', odate FROM Salespeople a.
Orders b WHERE a.snum = b.snum AND b.amt = (SELECT MAX (amt)
FROM Orders c WHERE c.odate = b.odate)
```

UNION

```
SELECT a.snum, sname, onum. 'Lowest on', odate FROM Salespeople a.
Orders b WHERE a.snum = b.snu AND b.amt = (SELECT MIN (amt)
FROM Orders c WHERE c.odate = b.odate)
```

ORDER BY 3;

1007	Rifkin	3001	Lowest	on	10/03/199
1002	Serres	3005	Highest	on	10/03/199
1002	Serres	3007	lowest	on	10/04/199
1001	Peel	3008	Highest	on	10/05/199
1001	Peel	3008	Lowest	on	10/05/199
1003	Axelrod	3009	Highest	on	10/04/199
1002	Serres	3010	Lowest	on	10/06/199
1001	Peel	3011	Highest	on	10/06/199

### Nazorat savollari

1. SQL muhitida bir nechta jadvallardan ma'lumotlar olish deganda nimani tushunasiz?
2. Jadvallarni birlashtirish qanday amalga oshiriladi?
3. UNION kalit so'zining vazifasi nima?

### Adabiyotlar

1. С. Д. Кузнецов. Введение в стандарты языка баз данных SQL. Центр Информационных Технологий, 1998.
2. Sh. Nazirov, A. Ne'matov, R. Qobulov. Ma'lumotlar bazasini dasturlash chuqurlashtirilgan kursi. Toshkent, "Sharq", 2007.

1. [www.scil.iu](http://www.scil.iu)



## SQL TEXNOLOGIYALARI FANIDAN JORIY BAXOLASH UCHUN NAZORAT SAVOLLARI

1. Ma'lumotlar bazasi nima?
2. Ma'lumotlarning ierarxik modeli deganda nimani tushunasiz?
3. Ma'lumotlarning tarmoqli modeli deganda nimani tushunasiz?
4. Ma'lumotlarning relyatsion modeli deganda nimani tushunasiz?
5. Birlamchi kalit nima?
6. Jadval munosabatni ifodalash uchun qanday talablar bajarilishi kerak?
7. Qanday mantiqiy operatorlarni bilasiz?
8. Agregat funksiya deganda nimani tushunasiz?
9. Predikat yoki funksiya deganda nimani tushunasiz?
10. SQL tizimida xavfsizlikni kim tashkillashtiradi?
11. SQL tizimida xavfsizlik necha bosqichda ta'munlanadi?
12. Foydalanuvchilarga qanday imtiyozlar beriladi?
13. Ma'lum rolga ega foydalanuvchilar deganda kimlarni tushunasiz?
14. SQL tilining mantiqiy operatorlari nima uchun ishlatiladi?
15. Qanday mantiqiy operatorlarni bilasiz?
16. Agregat funksiya deganda nimani tushunasiz?
17. Predikat yoki funksiya deganda nimani tushunasiz?
18. SQL tizimida xavfsizlikni kim tashkillashtiradi?
19. SQL tizimida xavfsizlik necha bosqichda ta'munlanadini?
20. Foydalanuvchilarga qanday imtiyozlar beriladi?
21. Ma'lum rolga ega foydalanuvchilar deganda kimlarni tushunasiz?
22. SQL tilidagi funkstiyalar qanday sinflarga bo'linadi?
23. Matematik funkstiyalarga qaysi funkstiyalar kiradi?
24. Qator funkstiyalariga qaysi funkstiyalar kiradi?
25. Tizim funkstiyalariga qaysi funkstiyalar kiradi?
26. Xavfsizlik funkstiyalariga qaysi funkstiyalar kiradi?
27. Foydalanuvchiga qanday imtiyozlar beriladi?
28. Foydalanuvchi imtiyozlarini kim o'rnatadi?
29. Foydalanuvchi huquqlari qanday buyruq orqali o'rnatiladi?
30. Tizim huquqini o'rnatuvchi qanday buyruqlarni bilasiz?
31. Foydalanuvchi SQL tizimiga qanday qo'shiladi?
32. Foydalanuvchi SQL tizimidan qanday olib tashlanadi?
33. SQL muhitida relyatsion MB arxitekturasi deganda nimani tushunasiz?
34. MBni belgilariga qarab qanday sinflarga ajratish mumkin?
35. Yaratilgan MB jadvallariga ma'lumotlar qanday kitiladi?
36. Jadvallarga nima uchun indekslar tashkillashtiriladi?
37. So'rov nima va u nima uchun ishlatiladi?
38. MBdan ma'lumotlarni tanlab olish deganda nimani tushunasiz?
39. So'rovlar yordamida yangi ob'yektlar hosil qilish mumkinmi?
40. SQL muhitida bir nechta jadvallardan ma'lumotlar olish deganda nimani tushunasiz?
41. Jadvallarni birlashtirish qanday amalga oshiriladi?

42. UNION kalit so'zining vazifasi nima?
43. Birlashtirilgan jadvalardagi ma'lumotlarni chop etishda ular qanday tartiblanadi?
44. Jadval psevdonimlari nima va ular qachon ishlatiladi?
45. SQL muhitidagi tranzaksiya nima?
46. Tranzaksiyalarning bajarilish jarayonida ma'lumotlarning himoyalanih darajasi deganda nimani tushunasiz?
47. Tranzaksiyalar MB jadvallari strukturasi buzmasligi uchun qanday tartiblar o'rnatiladi?
48. SQL muhitida blokirovka nima?
49. Blokirovkalarining qanday turlarini bilasiz?
50. SQL so'rovlarida "blokirovka birliklari" nimani bildiradi?
51. SQL muhitidagi tasvir nima?
52. Tasvirlarni qo'lalashdagi ustunliklar deganda nimani tushunasiz?

### **Adabiyotlar**

1. Макарова Н.В. и др. Информатика. /Под ред. Н.В. Макаровой. -М.: Финансы и статистика, 2003. 768 с.
2. Кириллов В.В. Основы проектирования реляционных баз данных. Учебное пособие. СПб.ИТМО,1994.
3. Дейт К. Введение в системы баз данных. -М.Наука,1980 г.
4. Кузнецов С.Д. Введение в стандарты языка баз данных SQL.-М. 1998г.
5. Sh.A. Nazirov, R.V. Qobulov. SQL va ma'lumotlar bazalarini keyingi dasturlash.Toshkent, 2006.
6. Шкарина Л. Язык SQL. Учебный курс. Санкт-Петербург. 2001.
7. Кузнецов С.Д. Введение в стандарты языка баз данных SQL.-М. 1998.
8. Sh. Nazirov, A. Ne'matov, R. Qobulov. Ma'lumotlar bazasini dasturlash chuqurlashtirilgan kursi. Toshkent, "Sharq", 2007.
9. К. Дейт. Руководство по реляционной СУБД DB2. –М.: Финансы и статистика. 1988.
10. О. Л. Голицина, Н. В. Максимов, И. И. Попов. Базы данных. –М.: 2005.
11. Кириллов В.В. Основы проектирования реляционных баз данных. Учебное пособие. СПб. ИТМО, 1994.
12. С. К. Ганиев, М.М. Каримов, К.А. Ташев. Ахборот хавфсизлиги. -Т., "Алоқачи", 2008.
13. Murodov O'.M. SQL texnologiyalar fanidan maruzalar matni. TATU Samarqand filiali, 2012.

### **Internet ma'lumotlari**

1. [www.mysql.ru](http://www.mysql.ru)
2. [www.sql.ru](http://www.sql.ru)
3. [www.intuit.ru.department/database](http://www.intuit.ru.department/database)
4. [www.firststeps.ru/sql](http://www.firststeps.ru/sql)

## **SQL TEXNOLOGIYALARI FANINI MUSTAQIL O'RGANISH UCHUN ZARURIY TERMINLAR VA ULARNING IZOHLARI**

Ma'lumotlar bazasi – minimal xatoliklarda saqlanadigan, bir qancha foydalanuvchilar tomonidan foydalanishga mo'ljallangan, ob'ektlarning o'zaro aloqasi va holatini aks ettiradigan, biror-bir predmet sohaga aloqador bir-biriga bog'liq ma'lumotlar to'plami.

Ma'lumotlar banki – Jamoa bolib ko'p maqsadlarda ishlatishga mo'ljallangan va markazlashgan holda tashkillastiriladigan tashkiliy, texnik, dasturiy hamda lingvistik jihatdan maxsus yaratilgan ma'lumotlat tizimi.

MB topologiyasi – MB komponentalarining fizik tashuvchilardagi taqsimlanish sxemasi, xuddi shunday turli xil hisoblash tugunlarida joylashgan komponentalar sxemasini ham ifodalaydi.

Saqlash nuqtasi – MBdagi tranzaksiya ishining yozilish vaqt momenti. Tranzaksiyalarda ish oraliq nuqtalari sifatida bir qator saqlash nuqtalari qo'llanilishi mumkin.

Tranzaksiya - Ma'lumotlar bazasi ustida bajariladigan amallar ketma-ketligidan iborat bo'lib, MBni bir holatdan qarama-qarshi bo'lmagan boshqa holatga o'tkazadi va bu jarayon bitta "hodisa" ko'rinishida ifodalanadi.

Trigger – saqlanadigan protseduraning asosiy turi bo'lib, UPDATE, INSERT yoki DELETE operatorlari yordamida yuz beradigan o'zgarishlarni avtomatik ravishda bajaradi.

Tashqi sxema - Ma'lumotlarning foydalanuvchi yoki amaliy dastur tasviridagi ko'rinishi.

Ichki sxema – Ma'lumotlarning fizik tuzilishi.

Hujjat – Hujjatli tizimlardagi ierarxik tuzilishga ega bo'lgan, elementlari fiksirlangan uzunlikdagi, odatda matnli maydonlardan yoki aniq belgilar ketma-ketligidan iborat, mantiqan pragraflarga (PAR,SEGM), ilovalarga (SENT), so'zlarga (WORD) bo'linadigan ma'lumotlar agregati.

Mantiqiy yozuv – tashqi xotira bilan ma'lumotlar almashinuvida amaliy dasturlar tomonidan bir butun birlik sifatida qabul qilinadigan ma'lumotlar agregatining yoki elementlarning nomlangan (identifikatsiyalashgan) majmuasi. Yozuv – bu quyidagi xarakteristikalariga bog'liq holda tartiblashtirilgan: turlariga qarab mos ravishda xotirada joylashtiriladigan ma'lumotlar maydonining (elementlarining) o'zaro bog'liq majmuasi.

Fizik yozuv –kiritish-chiqarish buyruqlarida bir butun birlik sifatida o'qiladigan yoki yoziladigan ma'lumotlar majmuasi.

Ma'lumotlarning ierarxik modeli – MB predmet sohasini ierarxik daraxt ko'rinishida tasvirlaydigan, tugunlari vertikal munosabatlarda “nasldan-avlodlarga” ko'rinishda aks ettiriladigan model. MBda ma'lumotlarni izlash ana shu yo'nalishda amalga oshadi. Juda taniqli, ierarxik model asosida yaratilgan MBBT, bu 1968 yilda yaratilgan IBM kompaniyasining Informatsion Managment System (IMS) dasturidir. Bu dasturning yutug'i quyidagilardan iborat: 1. MB tuzilishi kompaniya tizilishiga o'xshash; 2. “Avlod-dan-naslga” munosabati ishlatiladi. Masalan, “A Bga qarashli”; 3. IMS dasturida munosabatlar bir yozuv boshqa yozuvni ko'rsatadigan ko'rsatkichlar asosida tashkillashtirilgan. “Avlod” va “nasl” yozuvlarini alohida disklarda joylashtirish mumkin.

Import (zagruzka, download) – MBBT utilitasi (buyrug'i, funksiyasi) bo'lib, operatsion tizim uchun kommunikativ formatda tasvirlangan MB ma'lumotlaridan tashkil topgan fayl yozuvlarini o'qish uchun ishlatiladi.

Inventirlashtirilgan fayl (ro'yxat) – kalit qiymatlari bo'yicha yozuvlarni ixtiyoriy ravishda tez izlashga mo'ljallangan fayl. Bu fayl, o'zaro bog'liq bo'lmagan

tartiblashgan kalitar ro'yxati (indekslar) asosida tashkillashtiriladi. Kalit-asosiy faylning aniq maydoni yozuv qiymatlaridan tashkil topadi.

Indeks - yozuv adresini aniqlashda ishlatiladigan jadval.

Informatsiyali baza – Informatsion tizimda foydalaniladigan va aniq bir predmet soha holatini aks ettiradigan ma'lumotlar. Informatsion baza ikkita komponentadan tashkil topadi: 1) ma'lumotlar o'z yozuvlari kolleksiyasidan va 2) shu ma'lumotlar tavsiflaridan - metama'lumotlardan. Ma'lumotlar tavsiflardan ajratilgan, shu bilan birga ma'lumotlar tavsiflariga murojaatsiz ishlatilmaydi.

Mijoz – Foydalanuvchi yoki MBBTni o'rnatuvchilar tomonidan MBBTga nisbatan tashqi yozilgan dastur. Mijoz-dastur ilovasi MBBT "ustida" ishlaydigan ko'rinishda tashkillashtiriladi va MBBT komponentalari bilan amallar bajarish ushuncha tashqi interfeyslarga murojaat qiladi.

Kalit – identifikatsiyalash yoki yozuv adresini aniqlash ushuncha foydalaniladigan qiymat (ma'lumot elementi).

Birlamchi (bosh) kalit – qiymati yozuvni bir qitmatli aniqlaydigan kalit.

Ikkilamchi (al'ternativli) kalit – umumiy xossalarga ega bo'lgan bir guruh yozuvlarni identifikatsiyalaydigan kalit. Ma'lumotlar majmuasi bir nechta ikkilamchi kalitlarga ega bo'lishi mumkin. Bunday zaruriyat kalitlarga mos keluvchi yozuvlarni topish jarayonlarini optimallashtirish amaliy zaruriyatidan kelib chiqadi.

Ulangan (tarkibli) kalit – bir hechta ma'lumot elementlaridan tashkil topib, birgalikda ma'lumotlar majmuasidagi har bir yozuvning unikal identifikatsiyalashuvini ta'minlaydi.

Halqali struktura – ro'yxatli ma'lumotlar tuzilishi bolib, oxirgi element birinchi elementni ko'rsatadi, shunday qilib halqani tashkil qiladi.

Hazorat nuqtasi – MB ma’lumotlarini yangilash amallaridagi tizim buferi holati bo’lib, fizik fayllardagi MB holatini faol kesh xotira holati bilan moslash amali.

Malumotlar bazasi konsepsiyasi - Ma’lumotlarni qayta ishlash va saqlashning integrellashgan informatsion texnologiyasi bo’lib, uning asosida hamma saqlanadigan ma’lumotlar ichidan kerakli va shu dastur talabiga javob beradigan shaklda tanlab olishni qayta ishlaydigan dastur mexanizmi yotadi.

Kommutativli format – operatsion tizim fayllaridagi ma’lumotlarni tasvirlash usullaridan biri bo’lib, ma’lumotlar faylini operatsion tizimlardan MBga import qilishda informatsiyalar birligini (yozuvlar, agregatlar, maydonlar) xatosiz tanishni ta’minlaydi. Kommutativli formatlardagi fayllar MB mazmunini transportirovka qilishda va arxivlarda saqlashda ishlatishga mo’ljallangan. Kommutativli formatlarga misol sifatida FoxPro oilaviy tizimlarida ishlatiladigan asosiy usullarni aytish mumkin: 1) SDF (ma’lumotlarning tizimli formati), bunda MB jadvallarining har bir qatori matnli fayllarning yozuvlari ko’rinishida chiqarilib, ulardagi har bir element MB tavsiflariga mos keluvchi tur va uzunlikka ega bo’ladi. 2) Ajratgichli usullar, bunda har element qo’shnisidan ajratgich-belgilar (“terminatoplar”) yordamida ajratiladi. Hujjatlarni qayta ishlashga mo’ljallangan tizimlarda ancha murakkab formatlar ishlatiladi, masalan, agregat va ma’lumot elementlarini rekursiv tasvirlashda ishlatiladigan ISO 2709 sandarti ko’rinishi.

MBning mantiqiy strukturasi – MBning fizik bosqichdagi ixtiyoriy tasvirlanishi.

Mantiqiy fayl – Fayllarning amaliy masalalardagi tasvirlanishi, bunda fayl yozuvlari mantiqiy yozuvlardan tashkil topib, xotirada tasvirlangan ma’lumotlar strukturasi bilan farq qilishi mumkin.

Ruxsat usuli – operatsion tizimlarda mavjud bo’lgan, odatda operativ va tashqi xotira o’rtasida, ma’lumotlar almashinuvining tashkiliy usuli. Masalan, to’g’ridan-to’g’ri, ketma-ket, indeksli ruxsat usullari.

Ma'lumotlar bazasi modeli – ma'lumotlar bazasini boshqarishda mantiqiy va fizik aspekt chegaralarini aniqlashga imkon berivchi usullar majmui (ma'lumotlar bog'liqmasligi); ma'lumotlar mazmunini umumiy tushunishda dasturchilar va foydalanuvchilarni imkoniyatlar bilan ta'minlash vositalari (kommunikabellik); katta yozuvlar majmuasida (umumiy holda har xil turdagi ma'lumotlar) bir turdagi amallarni bajarish imkoniyati bilan yuqori bosqichdagi til tushunchalari bilan ta'minlash (to'plamlarni qayta ishlash) vositalari.

Ma'lumotlar modeli – abstrakt formal bosqichlarda ob'ekt va ular aloqalarini aniq usullarda tasvirlash bilan ta'minlaydigan bazali vositalar.

Datalogik model – aniq MBBTda infologik model asosida yaratiladigan va tasvirlanadigan hamda tavsiflash tili orqali ma'lumotlarni aks ettiruvchi tavsif.

Infologik (konseptual)model – MBni loyihalashtirish ustida ishlaydigan, hamma kishilar uchun tushunarli tabiiy til, matematik ifodalar, jadvallar, graflar va boshqa vositalardan foydalanib amalga oshiriladigan ma'lumotlar predmet sohasining tavsifi.

Fizik model – MBBT tashqi xotira qurilmalarida ma'lumotlar joylashishini va ularni izlashlarni aniqlovchi vosita.

Paradigmalarni modellashtirish - Ma'lumotlar strukturasi bosqichida ob'ekt aloqalarini ko'rsatishni aniqlaydigan shartli usul. Ana shu huqtai hazardan ierarxik, tarmoqli, relyatsion, ob'ektlashgan, relyatsion-ob'ektlashgan, hujjatli va boshqa turdagi modellarni farqlashadi.

Ma'lumotlarning mantiqan (fizik) bog'liqmasligi – ma'lumotlarning mantiqiy (fizik) strukturasi fizik (mantiqiy) strukturaga strukturani o'zgartirmasdan o'zgartiradigan tizim xosasi.

Normallashtirish – murakkab strukturali ma'lumotlarni (hujjatlarni) ikkilik jadvallar (munosabatlar) ko'rinishida tasvirlash.

Munosabat (relyatsion) – relyatsion MB jadvallarida saqlanadigan (jadval qatorlarida saqlanadigan) ma'lumotlar agregati, yoki ma'lumotlar bazasi so'rovlarini bajarish jarayonida virtual ravishda yaratiladigan ma'lumotlar agregati.

MB foydalanuvchilari – MBBT vositalari yotdamida MBga murojaat qiluvchi shaxs yoki dastur.

Predmet soha – real hayot abstrakt yoki aniq tusunchalar bilan akslanadigan va foal foydalanuvchilar uchun qishiqish uyg'otadigan hamda o'zaro aloqalari fiksirlanadigan ob'ektlar majmuasi.

Tasvir (View) – virtual jadvallarni hosil qilish vositasi bo'ib, bir yoki bir nechta jadvallarga bo'lgan so'rovlar (SELECT operatori natijasi) natijasini o'zida saqlaydi. Aniq foydalanuvchi uchun tasvir MB oddiy jadvallari ko'rinishida bo'ib, bu jadvallar ustuda SELECT, INSERT, UPDATE va DELETE operatoplarini bajarishi mumkin. Tasvir ma'lumotlarni qayta ishlashda va filtirlashda ishlatiladi.

Ma'lumotlar bazasini loyihalash – tasvirlashlarning o'zaro bog'liqligini tartivlashtirgan formallashtirish jarayoni – ya'ni predmet sohasining shunday modeliki, bu modelda MBda saqlanadigan ob'ektlarning aloqalari o'rnatiladi (fiksirlanadi) hamda shu ma'lumotlar orqali tavsiflanadi.

Taqsimlangan ma'lumotlar bazasi - ma'lumotlar bazasi majmuasi bo'lib, har biri alohida boshqariladi va qayta ishlanadu hamda ma'lumotlarni bo'laklashi ham mumkin.

Relyatsion algebra – munosabatlar ustuda bajariladigan amallarni o'z ichiga olgan algebra (til).

Relyatsion ma'lumotlar bazasi – munosabatlardan tashkil topgan ma'lumotlar bazasi. Bu munosabatlardagi foydalanuvchi uchun ruxsat berilgan barcha ma'lumotlar jadvallar ko'rinishida tashkillashtiriladi, bu jadvallar unikal nomga ega bo'lib, ustun va qatorlardan tashkil topadi, ustun va qator kesishuvida qaymatlar,



ma'lumotlar, joylashadi hamda ma'lumotlar ustida bajariladigan amallar ana shu jadvallar bilan amallar bajarilishiga olib kelinadi. Jadvallar asosan serverda joylashadi. Server – MBBT vazifasini bajaruvchi dasturdir. Serverda ma'lumotlarni aniqlash, ma'lumotlarni yozish, o'qish, olib tashlash, kontseptual, ichki, tashqi sxemalarni qo'llb-quvvatlash, so'rovlarni dispetchirlashtirish va optimallashtirish, ma'lumotlarni himoy qilish kabi vazifalar amalga oshiriladi.

Ma'lumotlarning tarmoqli modeli (model CODASIL) – CODASIL tomonidan taklif qilingan ierarxik modelning modifikatsiyasi. Bu modelda bitta yozuv bir nechta "avlod-nasl" munosabatlarida qatnashishi mumkin. Bunday munosabatlar tarmoqli modellarda to'plam deb yuritiladi. 1970 yillarda dastiriy ta'minotni erkin ishlab chiquvchilar Cullinet kompaniyasining IDMS tarmoqli modelini joriy ettilar. Tarmoqli ma'lumotlar bazasi quyidagi imkoniyatlarga ega: 1) Moslanuvchanlik. "Avlod-nasl" to'plam munosabatlarida tuzilishi ierarxik modellardagidan murakkabroq bo'lgan tuzilshdagi ma'lumotlarni ham saqlqah mumkin. 2) Standartlasuv. CODASIL standarti paydo bo'ldi. 3) Tezishlash. Tarmoqli modellar murakkabligiga qaramasdan ma'lumotlarni qayta ishlashda tezligi ierarxik modellar tezligidan kam bo'lmadi.

Kamchiligi shunda ediki, ma'lumotlar strukturasi o'zgartirish uchun butun ma'lumotlar bazasi strukturasi o'zgartirishga to'g'ri keladi. Munosabatlar va yozuvlar tuzilsishini oldindan or'natish lozim.

Ma'lumotlar basazini boshqarish tizimi (MBBT) – Ko'pchilik foydalanuvchilar tomonidan ishlatishga mo'ljallangan, MBni yaratish, boshqarish va birgalikda ishlatish uchun mo'ljallangan dastur va tillardan iborat majmua.

