

**O'ZBEKISTON RESPUBLIKASI
OLIV VA O'RTA TA'LIM VAZIRLIGI**

QARSHI MUHANDISLIK IQTISODIYOT INSTITUTI

GEOLOGIYA VA KONCHILIK fakulteti

Konchilik ishi yo'nalishi KI – 113-16 guruh talabasi

Yo'ldoshev Ruslanning "Axborot texnologiyalari" fanidan

**"Oracle ma'lumotlar bazasini SQL tili yordamida boshqarish"
mavzusidagi**

REFERAT

Bajardi:

R. Yo'ldoshev

Qabul qildi:

N.Xamrayev

Qarshi-2017

Mavzu: Oracle ma'lumotlar bazasini SQL tili yordamida boshqarish

Reja

1. Ma'lumotlar bazasi haqida umumiy tushunchalar.
2. SQL asoslari
3. Jadvallar bilan ishlash
4. SELECT so'rov operatori
5. Bir necha jadvallar bilan ishlash
6. So'rovlarda guruhlash va funksiyalar

MBBTning nazariy asoslari.

Informatsion tizimlarni yaratish bo'yicha jadal harakatlar ma'lumotlar hajmining tez suratlar bilan oshib borishi sharoitida 60 yillar boshida maxsus "Ma'lumotlar bazasini boshqarish tizimi" (MBBT) deb ataluvchi dasturiy kompleksning yaratilishiga olib keldi.

MBBT asosiy xususiyatlari - bu protseduralar tarkibi bo'lib, ular faqat ma'lumotlarni kiritish va saqlashda ishlatilmasdan, ularning strukturasi ham tasvirlaydi. Ma'lumotlarni o'zida saqlab va MBBT ostida boshqariladigan fayl, oldin ma'lumotlar banki deb atalib, keyinchalik esa "Ma'lumotlar bazasi" deb yuritila boshladi.

Ma'lumotlarni boshqarish tizimi, quyidagi xossalarga ega:

- ✓ fayllar to'plami mantiqiy kelishuvni quvvatlaydi;
- ✓ ma'lumotlar ustida ish yuritish tili bilan ta'minlaydi;
- ✓ har xil to'xtalishlardan keyin ma'lumotlarni qayta tiklaydi;
- ✓ MBBT bir necha foydalanuvchilarning parallel ishlashini ta'minlaydi.

MBBT funksiyalari tarkibiga yanada aniqroq qilib quyidagilar qabul qilingan:

- ✓ **Tashqi xotirada bevosita ma'lumotlarni boshqarish.**

Bu funktsiya MBga bevosita kiruvchi ma'lumotlarni saqlash uchun kerakli strukturani ta'minlab tashqi xotiraga qo'shadi. MBBT ishlatishda mavjud fayl tizimi imkoniyatlari aktiv ravishda ishlatiladi. Rivojlantirilgan MBBTda foydalanuvchi istalgan holda MBBT fayl tizimini ishlatayapdimi bu haqda bilishi shart emas, va agar ishlatsa, u holda fayllar tashkil qilingan bo'ladi. Xususiy holda MBga berkitilgan ob'ektlarni MBBT quvvatlaydi.

- ✓ **Tezkor xotirani bufer bilan boshqarish**

MBBT odatda ancha katta hajmdagi MB bilan ish yuritadi. Bu hajm odatda tezkor xotiraning mumkin bo'lgan hajmidan yetarli darajada katta bo'ladi. Ma'lumki, agar ma'lumotlarning biror elementiga murojaat qilish kerak bo'lsa tashqi xotira bilan aloqa o'rnatiladi, lekin barcha tizim tashqi xotira qurilmasi tezligida ishlaydi. Bu tezlikni oshirishning amaliy yagona usulilaridan biri bu operativ xotiraga ma'lumotlarni buferizatsiya qilishdir.

- ✓ **Tranzaktsiya bilan boshqarish**

Tranzaktsiya – bu qaralayotgan MBBT MB ustida ketma-ket operatsiyalarni bajararishidir, ya'ni ma'lumotlar bilan manipulyatsiya qilib ketma-ket operatsiyalar yordamida MBBTga ta'sir etishdir. Tranzaktsiya ma'lumotlar bazasini bir butun holatdan ikkinchi bir butun holatga o'tkazadi, yoki agar ma'lum sababga ko'ra tranzaktsiyaning biror holati bajarilmaydigan bo'lsa yoki tizimda biror xatolik yuz

bersa, ma'lumotlar bazasi boshlang'ich holatiga qaytadi. MBning mantiqiy butunligini quvvatlash uchun tranzaktsiya tushunchasi kerak.

✓ **Jurnalizatsiya**

MBBT ga bo'lgan asosiy talablardan biri bu tashqi xotirada ma'lumotlarning ishonchli saqlanishidir. Ma'lumotlarning ishonchli saqlanishi deganda har qanday apparatli yoki dasturli to'xtab qolishdan (sboydan) keyin MBBT MBning oxirgi holatini qayta tiklashi tushuniladi. Odatda apparatli to'xtab qolish holati ikki xil bo'ladi: yengil to'xtab qolish, ya'ni bunda kompyuter ishlashi kutilmaganda to'xtashi (masalan, elektr toki manbaining o'chishi), ikkinchisi qattiq to'xtab qolish, bu tashqi xotirada ma'lumotlarning yo'qolib ketishi bilan xarakterlanadi. Dasturli to'xtab qolishlarga quyidagilarni keltirishi mumkin: MBBTning to'satdan buzilishi bilan ishni tugatishi (dastur xatosi bo'yicha yoki qaysidir apparatning to'xtab qolishi natijasida) yoki foydalanuvchi dasturining avariya bilan tugallanishi bo'lib natijada ayrim tranzaktsiyalar tugallanmasdan qoladi. Har qanday holda ham MB qayta tiklash uchun qo'shimcha ma'lumotlarni joylashtirish kerak. Boshqacha qilib aytganda MB da ma'lumotlarning butunligini saqlash uchun saqlanadigan ma'lumotlarning to'liqligi talab qilinadi. Ma'lumotlarning ishlatilayotgan qismi qayta tiklanishi uchun alohida ishonchli saqlanishi lozim. Bunda to'liq ma'lumotlarni quvvatlash uchun keng tarqalgan usullardan biri MB ning o'zgartirish jurnalini olib borish usuli ishlatiladi.

Jurnal – MBning asosiy qismi bo'lib hisoblanadi va u barcha rivojdagi MBBT da “jurnal zapisi utverjdeniy” (protokol Write Ahead Log - WAL) deb nomlanadi. Qattiq to'xtab qolishdan keyin MBni qayta tiklash uchun jurnal va MBning arxiv nusxasi ishlatiladi. Arxiv nusxa - bu MBning to'liq nusxasi bo'lib, jurnalni to'ldirish momentidan boshlanadi

SQL asoslari

SQL (Structured Queru Language) tili srukturalashgan so'rov tili deyilib, u ma'lumotlar bazasi bilan aloqa o'rnatish uchun xizmat qiladi. SQL tili 70 yillar oxirida IBM firmasi tomonidan Sytem r nomli ma'lumotlar bazasini boshqarish tizimining tajribaviy loyihasini ishlab chiqish doirasida yaratildi. Keyinchalik IBM firmasi tomonidan Sytem r ga yaqin bo'lgan yana ikkita tizim- SQL/DS va DB2 tizimlari ishlab chiqildi. Bu tilning xalqoro standarti 1986 yili ishlab chiqildi va u 1989 yilga kelib yanada kengaytirildi, lekin uning to'liq xalqaro standarti 1992 yil qabul qilindi. 1995 yilga kelib SQL92 standarti yangi komponentalar bilan to'ldirildi. Birinchi bolib SQL dan foydalanish Oracle MBBTda ishlatildi.

SQL quyidagi komponentalarni o'z ichiga oladi:

- administrativ ma'lumotlar vositasi;
- tranzaksiyalar bilan boshqarish vositasi;
- DLL (Data Definition Language) ma'lumotlarni aniqlash operatorlari;
- DML (Data Manipulation Language) ma'lumotlar ustida manipulyatsiya qilish (murakkab harakatlar bajarish) operatorlari;
- DQL (Data Query Language) ma'lumotlar bazasiga so'rovlar tili.

Administrativ ma'lumotlar - ma'lumotlar bazasini yaratish, unga kirishga ruxsat berish va parollarni o'zgartirishda kerak bo'ladi.

Tranzaksiya - bu ma'lumotlar bilan manipulyatsiya qilib ketma-ket operatsiyalar yordamida MBBTga ta'sir etishdir. Tranzaksiya yoki to'liq bajariladi va ma'lumotlar bazasini bir butun holatdan ikkinchi bir butun holatga o'tkazadi, yoki agar ma'lum sababga ko'ra tranzaksiyaning biror holati bajarilmaydigan bo'lsa yoki sistemada biror xatolik yuz bersa, ma'lumotlar bazasi boshlang'ich holatiga qaytadi.

Ma'lumotlarni aniqlash operatorlari - ma'lumotlar bazasi jadvalini tuzish, o'zgartirish yoki o'chirish, indeks tashkil qilish (ma'lumotlarni tez izlab topish vositasi), har xil foydalanuvchilar bilan ishini tashkil qilish va boshqalarni amalga oshiradi.

Ma'lumotlar ustida manipulyatsiya qilish operatorlari - ma'lumotlar bazasiga yozuvni qo'shish, o'chirish yoki yangilash imkonini beradi.

Jadvallar bilan ishlash.

Ma'lumotlar bazasining jadvallari massiv ko'rinishida bo'lib, ular satr va ustunlardan iborat bo'ladi. Jadval tuzilgandan keyin uni to'ldirishga kirishish mumkin.

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>)], ...);

<table-name > - jadval nomi

<column name> - maydon nomi

<data type>[(<size>) maydon turi va ulchami

Argument qiymati kattaligi ma'lumot turiga bog'liqdir. Agar siz maxsus ko'rsatmasangiz, tizim avtomatik qiymatni o'rnatadi.

Misol uchun talabalar jadvalini yaratishni ko'rib chiqamiz:

```
CREATE TABLE TALABA (  
    ID            INTEGER,  
    ID_TALABA    INTEGER,  
    FIO          VARCHAR(45) ,  
    QURS        CHAR(1) ,  
    FIRED       VARCHAR(1) ,  
    STIPEND     FLOAT,  
    ID_FAK      INTEGER,  
    ID_GR       INTEGER,  
    ADRES       VARCHAR(80) ,  
    SOST        VARCHAR(1) ,  
    GR          VARCHAR(30) ,  
    BUYRUQ_LOG  VARCHAR(500) ,  
    OPER_DAT    DATE ,  
    ACTIV1     VARCHAR(1) ,  
    ACTIV2     VARCHAR(1) ,  
    PS_SER     VARCHAR(2) ,  
    PS_NUM     VARCHAR(7) ,  
    GRUPPA     CHAR(1) ,  
    BUYRUQ     CHAR(5) ,  
    DAT1       DATE ,  
    DAT2       DATE ,  
    PERIOD     CHAR(9) ,  
    ASCII     CHAR(32) ,  
    MANZIL     CHAR(140) ,
```

VIL CHAR (2) ,
 TUMAN CHAR (2) ,
 HARBIY CHAR (1) ,
 JINSI CHAR (1) ,
 OILA CHAR (1) ,
 OTASI VARCHAR (80) ,
 ONASI VARCHAR (80) ,
 T_SANA DATE) ;

Talaba jadvali

Sn um	Fam	Ismi	Born_ dat
1	Ataboev	Nuriddin	1975
2	Ikromov	Elyor	1989
3	Mamada lieva	Zulhum or	1989

ID - Qator identifikatsiya rakami,
 ID_TALABA - Xar bir talabaga belgalangan tartib rakami,
 FIO - Talabaning familiyasi,
 QURS - Kursi,
 FIRED - Katorni o'chirilganlik belgisi,
 STIPEND - Stipendiya,
 ID_FAK - Fakultet kodi,
 ID_GR - Gurux kodi,
 ADRES - Manzili,
 SOST - Xolat belgisi,
 GR - Gurux,
 BUYRUQ_LOG - Buyruk mazmuni,
 OPER_DAT - Operatsiya sana,
 PS_SER - Pasport seriya,
 PS_NUM - Pasport nomeri,
 GRUPPA - Gurux belgisi,
 BUYRUQ - Buyruk nomeri,
 DAT1 - Boshlangich sana,
 DAT2 - Oxirgi sana,
 PERIOD - Davr,

ASCII - Nazorat summa,
 VIL - Viloyat ,
 TUMAN - Tuman ,
 HARBIY - Narbiyga alokadorlik belgisi ,
 JINSI - Jinsi,
 OILA - Oilalilik belgisi,
 OTASI - Otasi,
 ONASI - Onasi,
 T_SANA - Tugilgan sanasi

ID	CODE	FIO	FAK_C...	ISH_OST	SUMMA_DEB	SUMMA_KRED	STIP1
110 027	218-C 161106	Гофоров Ханидилло	7	0,00	0,00	0,00	<null>
110 928	133-C 170907	Мадрахимов Расул Кучкоревич	00	<null>	<null>	<null>	<null>
110 966	133-C 170907	Тургунов Абдугаффор Абдивалиевич	00	<null>	<null>	<null>	<null>
111 191	133-C 170907	Сулаймонова Машхура Косимовна	00	<null>	<null>	<null>	<null>
111 196	140-C 021007	Юлдашева Розияхон кодировна	7	<null>	<null>	<null>	<null>
111 200	133-C 170907	Кучкарова Нафисахон Асадиллаевна	00	<null>	<null>	<null>	<null>
110 979	133-C 170907	Тухтасинов Акбарали Мирзакаримович	00	<null>	<null>	<null>	<null>
111 219	133-C	Рахимбердиева(Шокирова) Мухаббатхон Акбарали	7	<null>	<null>	<null>	<null>
111 302	133-C 170907	Юсупова Лазиза Зокиржонова	00	<null>	<null>	<null>	<null>
111 360	133-C 170907	Бадалбаев Жаётулло Ахмадуллаевич	00	<null>	<null>	<null>	<null>
111 386	133-C 170907	Абдукаримова Сохибабону Хайитбой кизи	00	<null>	<null>	<null>	<null>
111 418	133-C 170907	Ортикова Малика Обобакировна	00	<null>	<null>	<null>	<null>
111 526	133-C 170907	Ахунов Иномидин Камолдинович	00	<null>	<null>	<null>	<null>
111 541	133-C 170907	Буванатова Зарнигор Дилмуродовна	00	<null>	<null>	<null>	<null>
111 550	166-C 021107	Рисов Фаррух Рахматжонович	00	<null>	<null>	<null>	<null>
111 561	166-C 021107	Сойилов Рашидбек Маликжонович	00	<null>	<null>	<null>	<null>
111 649	133-C 170907	Абдурахмонова Юлдузхон Зохидажон кизи	00	<null>	<null>	<null>	<null>

Masalan Oracle ma'lumotlar bazasida yangi hodimlar bo'yicha jadval yaratamiz.


```
Oracle SQL*Plus
File Edit Search Options Help

Connected to:
Oracle7 Server Release 7.3.4.0.0 - Production
With the distributed, replication and parallel query options
PL/SQL Release 2.3.4.0.0 - Production

SQL> create table xodim (
2  pasnom      char(12) not null          -- паспорт серияси ва номери
3  ,famil      char(30) not null         -- фамилияси
4  ,ismi       char(20) not null        -- исми
5  ,otasi      char(10)                 -- отасининг исми
6  ,manzili    varchar2(50) not null    -- турар жойи
7  ,uquv       varchar2(30) not null    -- таномлаган уқув даргохи
8  ,lavozim    char(20) not null        -- мансаби
9  ,maoshi     number(15,2) not null    -- лавозим маоши
10 );

Table created.

SQL> |
```

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 >; [4]

<table-name > - jadval nomi

Masalan: Talaba jadvalining butunligicha o'chirish

DROP TABLE talaba;

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>;
```

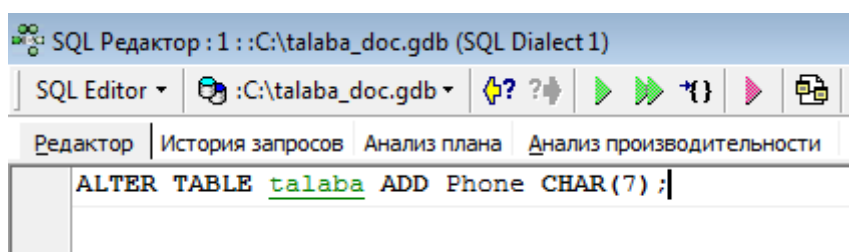
<table-name > - jadval nomi

<column name> - maydon nomi

<data type>[(<size>)] maydon turi va o'lchami [4]

Masalan:

```
ALTER TABLE talaba ADD Phone CHAR(7);
```



ER	PS_NUM	GRUPPA	BUYRUQ	DAT1	DAT2	PERIOD	ASCII	MANZIL	VIL	TUMAN	HARBIY	JINSI	OILA	OTASI	ONASI	T_SANA	PHONE
	1940986	<null>	<null>	24.08.2006 18:06	<null>	Талабалар	<null>	<null>	Ан	Бу	Й	Э	У	<null>	<null>	13.05.1989 00:00	<null>
	1950165	<null>	<null>	24.08.2006 18:06	<null>	Талабалар	<null>	<null>	Ан	Ас	Й	Э	У	<null>	<null>	06.02.1989 00:00	<null>
▶	1925624	<null>	<null>	24.08.2006 18:06	<null>	Талабалар	<null>	<null>	Ан	Жа	Й	А	Т	<null>	<null>	29.08.2019 00:00	<null>
	1643066	<null>	<null>	24.08.2006 18:06	<null>	Талабалар	<null>	<null>	Ан	Па	Й	А	Т	<null>	<null>	02.06.2019 00:00	<null>
	1879855	<null>	<null>	24.08.2006 18:06	<null>	Талабалар	<null>	<null>	Ан	Хо	Й	А	Т	<null>	<null>	30.11.2019 00:00	<null>
	1751012	<null>	<null>	24.08.2006 18:06	<null>	Талабалар	<null>	<null>	Ан	Ху	Й	А	Т	<null>	<null>	27.12.2019 00:00	<null>
	1858970	<null>	<null>	24.08.2006 18:06	<null>	Талабалар	<null>	<null>	Ан	Ан	Й	А	Т	<null>	<null>	30.07.2019 00:00	<null>
	1958240	<null>	<null>	24.08.2006 18:06	<null>	Талабалар	<null>	<null>	Ан	Ол	Й	Э	У	<null>	<null>	01.02.2019 00:00	<null>
	1958240	<null>	<null>	24.08.2006 18:06	<null>	Талабалар	<null>	<null>	Ан	Бу	Й	А	Т	<null>	<null>	14.05.2019 00:00	<null>
	1888502	<null>	<null>	24.08.2006 18:06	<null>	Талабалар	<null>	<null>	Ан	Бу	Й	Э	У	<null>	<null>	29.04.2019 00:00	<null>
	1651864	<null>	<null>	02.10.2006 18:06	<null>	Талабалар	<null>	<null>	Ан	Бу	Й	А	Т	<null>	<null>	17.03.1986 00:00	<null>
	1534038	<null>	<null>	02.10.2006 18:06	<null>	Талабалар	<null>	<null>	Ан	Ку	Й	Э	У	<null>	<null>	16.06.1986 00:00	<null>
	1913359	<null>	<null>	02.10.2006 18:06	<null>	Талабалар	<null>	<null>	Ан	Жа	Й	А	Т	<null>	<null>	02.01.1989 00:00	<null>
	1819186	<null>	<null>	02.10.2006 18:06	<null>	Талабалар	<null>	<null>	Ан	Ку	Й	А	Т	<null>	<null>	22.05.1989 00:00	<null>
	1507460	<null>	<null>	02.10.2006 18:06	<null>	Талабалар	<null>	<null>	Ан	Па	Й	А	Т	<null>	<null>	23.03.1986 00:00	<null>
	1492829	<null>	<null>	02.10.2006 18:06	<null>	Талабалар	<null>	<null>	Ан	Жа	Х	Э	У	<null>	<null>	13.03.1986 00:00	<null>
	1494133	<null>	<null>	02.10.2006 18:06	<null>	Талабалар	<null>	<null>	Ан	Из	Х	Э	У	<null>	<null>	11.04.1985 00:00	<null>
	1838112	<null>	<null>	02.10.2006 18:06	<null>	Талабалар	<null>	<null>	Ан	Жа	Й	Э	У	<null>	<null>	08.02.1986 00:00	<null>
	1290665	<null>	<null>	02.10.2006 18:06	<null>	Талабалар	<null>	<null>	Ан	Ан	Й	А	Т	<null>	<null>	30.05.1972 00:00	<null>
	2267273	<null>	<null>	02.10.2006 18:06	<null>	Талабалар	<null>	<null>	Фа	Ол	Й	А	Т	<null>	<null>	24.05.1987 00:00	<null>
	1791393	<null>	<null>	02.10.2006 18:06	<null>	Талабалар	<null>	<null>	Ан	Жа	Й	А	Т	<null>	<null>	30.04.1988 00:00	<null>

Bu erda talaba jadvaliga yangi maydon phone (telefon raqami qo'shilmoqda).

Jadvallarga cheklov qo'yish.

Cheklanishlarni kiritish.

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

Masalan:

```
CREATE TABLE talaba
( SNum    integer NOT NULL UNIQUE,
  FAM    char (15),
  ISM    char (15),
  Born_dat numeric (4,0)); [4]
```

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. Bizning ma'lumotlart bazamizda har bir buyurtmachi bitta sotuvchiga birlashtirilgan. 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.

Birlamchi kalitlar cheklanishlari.

SQL birlamchi kalitlarni to'g'ridan to'g'ri birlamchi kalit (PRIMARY KEY) cheklanishi orqali ta'riflaydi. PRIMARY KEY jadvalni yoki ustunlarni cheklashi mumkin. Bu cheklanish UNIQUE cheklanishi kabi ishlaydi, jadval uchun faqat bitta birlamchi kalit (ixtiyoriy sondagi ustunlar uchun) aniqlanishi mumkin bo'lgan xoldan tashqari. Birlamchi kalitlar NULL qiymatga ega bo'lishi mumkin emas. [4]

Misol:

```
CREATE TABLE Talaba
( SNum    integer NOT NULL PRIMARY KEY,
  FAM    char (10),
  ISMI    char (10),
```

Born_dat decimal);

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 ifodalaridan iboratdir. [4]

Misol: talaba jadvali Born_dat ustuniga kiritilayotgan qiymat 1990 dan kichik bo'lish sharti.

```
CREATE TABLE talaba
```

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

```
FAM char(10) NOT NULL UNIQUE,
```

```
ISMI char(10),
```

```
Born_dat decimal CHECK (Born_dat < 1990 ));
```

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 ko'zda tutilgan qiymat NULL qiymatdir. CREATE TABLE komandasida ko'zda tutilgan qiymat DEFAULT operatori orqali, ustun cheklanishi sifatida ko'rsatiladi. Masalan:

```
CREATE TABLE talaba
```

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

```
Fam char(10) NOT NULL UNIQUE,
```

```
Ismi char(10) ,
```

```
Summa decimal default '20000',
```

```
Born_dat decimal CHECK ( Comm < 1990 ));
```

Ma'lumotlar yaxlitligini ta'minlash.

Jadval bir maydonidagi hamma qiymatlar boshqa jadval maydonida aks etsa, birinchi maydon ikkinchisiga ilova qiladi deyiladi. Bu ikki maydon orasidagi bog'liqlikni ko'rsatadi. Masalan, buyurtmachilar jadvalida har bir buyurtmachi, sotuvchilar jadvalida o'ziga biriktirilgan sotuvchiga ilova qiluvchi SNum

maydoniga ega. Bir maydon ikkinchisiga ilova qilsa tashqi kalit, u ilova qilayotgan maydon ajdod kalit deyiladi. Buyurtmachilar jadvalidagi SNum maydoni tashqi kalit, sotuvchilar jadvalidagi SNum - ajdod kalitdir.

Tashqi kalit bitta maydondan iborat bo'lishi shart emas. Birlamchi kalit kabi, tashqi kalit bitta modul sifatida qayta ishlanuvchi bir necha maydonlarga ega bo'lishi mumkin. Maydon tashqi kalit bo'lsa ilova qilayotgan jadval bilan ma'lum usulda bog'liqdir. Tashqi kalit har bir qiymati (satri), ajdod kalitning bitta va faqat bitta qiymatiga(satriga) ilova qilishi kerak. Bu 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 kalit maydonlariga kiritmaslikdir. FOREIGN KEY cheklanishi sintaksisi:

```
FOREIGN KEY <column list> REFERENCES  
<pktable> [<column list>] [4]
```

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 talabalar jadvaliga ilova qiluvchi tashqi kalit sifatida e'lon qilingan id_talaba maydoniga ega bo'lgan stipendiya jadvalini yaratamiz:

```
CREATE TABLE stip  
( CNum integer NOT NULL PRIMARY KEY,  
  id_talaba integer,  
  oy char (10) ,  
  Summa numeric (10,2) ,  
  Kasaba numeric (10,2),  
  Qo'lga numeric (10,2),  
  FOREIGN KEY (SNum) REFERENCES stip (id_talaba );
```

SQL da monepulyatsiya.

Hamma satrlar SQLda INSERT komandasi yordamida kiritiladi. INSERT quyidagi formatlar biriga ega bo'lishi mumkin:

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

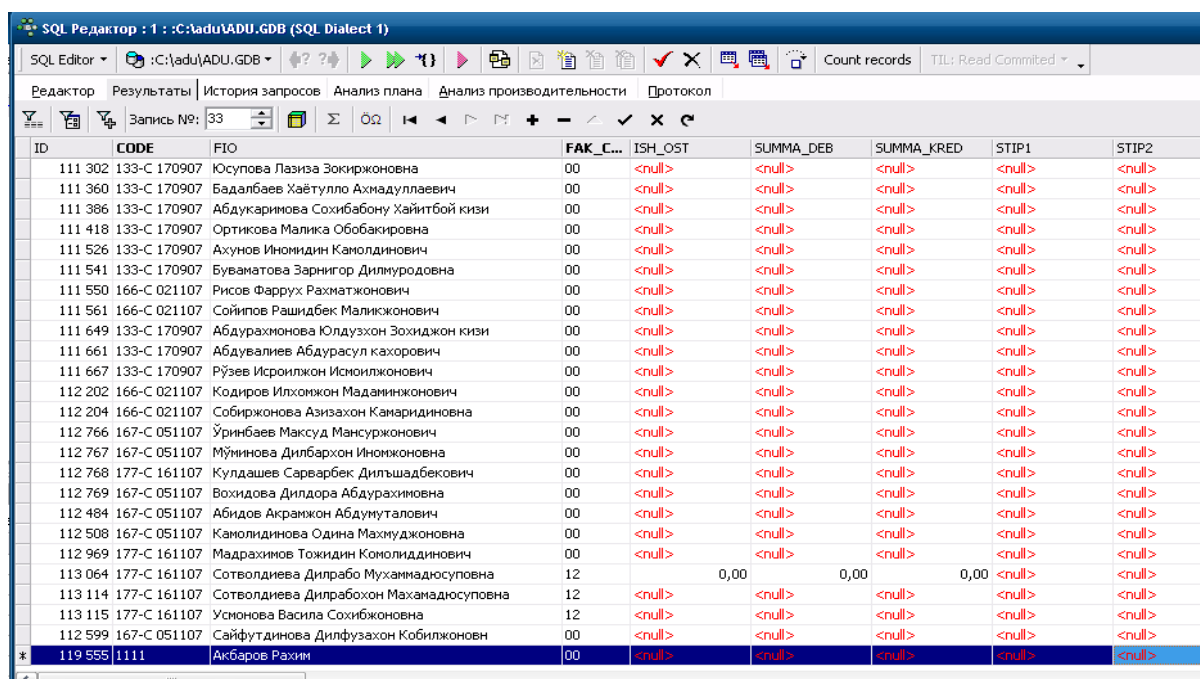
yoki

```
INSERT INTO <table name | view name> [(column [,column] ...)] [4]
```

Ostki so'rov;

Masalan, talabalar jadvaliga satr kiritish uchun quyidagi shartdan foydalanishingiz mumkin:

```
INSERT INTO talaba (id, FIO, qurs)  
VALUES (14, 'Akbarov Sarvar', '4');
```



ID	CODE	FIO	FAK_C...	ISH_OST	SUMMA_DEB	SUMMA_KRED	STIP1	STIP2
111 302	133-C 170907	Юсупова Лазиза Зокиржоновна	00	<null>	<null>	<null>	<null>	<null>
111 360	133-C 170907	Бадалбаев Хаётулло Ахмадуллаевич	00	<null>	<null>	<null>	<null>	<null>
111 386	133-C 170907	Абдукаримова Сохибабону Хайитбой кизи	00	<null>	<null>	<null>	<null>	<null>
111 418	133-C 170907	Ортикова Малика Обобакировна	00	<null>	<null>	<null>	<null>	<null>
111 526	133-C 170907	Ахунов Иномидин Канолдинович	00	<null>	<null>	<null>	<null>	<null>
111 541	133-C 170907	Буванатова Зарнигор Дилмуродовна	00	<null>	<null>	<null>	<null>	<null>
111 550	166-C 021107	Рисов Фаррух Рахматжонович	00	<null>	<null>	<null>	<null>	<null>
111 561	166-C 021107	Сойипов Рашидбек Маликжонович	00	<null>	<null>	<null>	<null>	<null>
111 649	133-C 170907	Абдурахмонова Юлдузхон Зокиржон кизи	00	<null>	<null>	<null>	<null>	<null>
111 661	133-C 170907	Абдувалиев Абдурасул кахорович	00	<null>	<null>	<null>	<null>	<null>
111 667	133-C 170907	Рўзев Исроилжон Исмоилжонович	00	<null>	<null>	<null>	<null>	<null>
112 202	166-C 021107	Кодиров Илхомжон Маданинжонович	00	<null>	<null>	<null>	<null>	<null>
112 204	166-C 021107	Собиржонова Азизахон Канаридиновна	00	<null>	<null>	<null>	<null>	<null>
112 766	167-C 051107	Ўринбаев Максуд Мансуржонович	00	<null>	<null>	<null>	<null>	<null>
112 767	167-C 051107	Мўминова Дилбархон Иномжоновна	00	<null>	<null>	<null>	<null>	<null>
112 768	177-C 161107	Кулдашев Сарварбек Дильшадбекович	00	<null>	<null>	<null>	<null>	<null>
112 769	167-C 051107	Вохидова Дилдора Абдурахимовна	00	<null>	<null>	<null>	<null>	<null>
112 484	167-C 051107	Абидов Акрамжон Абдумуталович	00	<null>	<null>	<null>	<null>	<null>
112 508	167-C 051107	Канолдинова Одина Махмуджоновна	00	<null>	<null>	<null>	<null>	<null>
112 969	177-C 161107	Мадрохимов Тождин Комолиддинович	00	<null>	<null>	<null>	<null>	<null>
113 064	177-C 161107	Сотволдиева Дилрабо Мухаммадјусуповна	12		0,00	0,00	0,00	<null>
113 114	177-C 161107	Сотволдиева Дилрабохон Махаммадјусуповна	12	<null>	<null>	<null>	<null>	<null>
113 115	177-C 161107	Усмонова Василя Сохибжонковна	12	<null>	<null>	<null>	<null>	<null>
112 599	167-C 051107	Сайфутдинова Дилфузхон Кобилжоновн	00	<null>	<null>	<null>	<null>	<null>
* 119 555	1111	Акбаров Рахим	00	<null>	<null>	<null>	<null>	<null>

Siz nom kiritish uchun ustunlar ko'rsatishingiz mumkin. Bu nomlarni ixtiyoriy tartibda kiritishga imkon beradi. Masalan:

```
INSERT INTO Stip (Id_talaba, Oy, Summa, Kasaba, Summa, Qulga )  
VALUES (11, 'май' ,50666, 1, 506,66);
```

E'tibor bering bu erda birinchi kavsdagi xar bir uzgaruvchi maydonga ikkinchi kavsdagi mos qiymatlar qabul qilinadi.

Satrlarni o'chirish.

Satrlarni jadvaldan DELETE komandasi bilan o'chirish mumkin. U alohida qiymatlarni emas faqat satrlarni o'chiradi. DELETE quyidagi formatga ega:

DELETE FROM <table name | view name>

[**WHERE** search-condition];

Masalan, talabalar jadvalidagi hamma satrlarni o'chirish uchun, quyidagi shartni kiritish mumkin:

DELETE FROM talaba;

Ma'lum satrlarni o'chirish uchun predikatdan foydalaniladi. Masalan, jadvaldan

Yuldashev Abrorni o'chirish uchun:

DELETE FROM Talaba WHERE id=55;

Komandasi ishlatiladi.

ID	ID_TALABA	FIO	QURS	FIRE	STIPEND	ID_FAK	ID_GR	ADRES
38	<null>	Шокиров Донёр Достонович	4	N	<null>	1	1	Шарк Юлдуз КФЙ С,
39	<null>	Юлдашев Козим Мансурович	4	N	<null>	1	1	Катортол КФЙ мирза
40	<null>	Алимова Саида Комилжоновна	4	N	<null>	1	1	Андижон шаҳар пушк
41	<null>	Рузиматова Шоира Ёркинбековна	4	N	<null>	1	1	Олтиарик тумани Чи
42	<null>	Халилова Махлиё Содикжоновна	4	N	<null>	1	1	Жанубий оланушук ш
43	<null>	Юлдашева Махсума Ариповна	4	N	<null>	1	1	Мустақиллик 26 уй
44	<null>	Сиддикова Зарифа Уткировна	4	N	<null>	1	1	Ёркишлок манадалие
32	<null>	Атабоева Дилдора Нодирбековна	4	N	<null>	1	1	Ширмонбулок кишлог
33	<null>	Холмирзаев Шерали Икромович	4	N	<null>	1	1	Куштепа ЖК 16
34	<null>	Абидова Махлуза Тохировна	4	N	<null>	1	1	Ойим киш. 49
22	<null>	Джаппоров Илс Илхомович	4	N	<null>	1	1	<null>
23	<null>	Шералиев Икром Исломович	4	N	<null>	1	1	<null>
24	<null>	Омонова Гулзира Акбаралиевна	4	N	<null>	1	1	<null>
25	<null>	Туйчиева Хуррият Акбаралиевна	4	N	<null>	1	1	<null>
35	<null>	Мусабоева Зулайхо Дилшодбековна	4	N	<null>	1	1	Куштепа ж-х 49
36	<null>	Тошматова Мархабо Аскарловна	4	N	<null>	1	1	Маданият киш Корин
26	<null>	Таджибаева Махлиё Зухритдиновна	4	N	<null>	1	1	<null>
27	<null>	Таштанкулова Наргиса Толиповна	4	N	<null>	1	1	<null>
28	<null>	Джурабоева Максуда Мансуровна	4	N	<null>	1	1	<null>
29	<null>	Кодиров Уткир Азамович	4	N	<null>	1	1	<null>
30	<null>	Бойматова Оязим Донёровна	4	N	<null>	1	1	<null>
31	<null>	Каюмов Элёр Кодирович	4	N	<null>	1	1	<null>
37	<null>	Буриев Хикматилло Хамидович	4	N	<null>	1	1	Абдуллабий КФЙ
45	<null>	Хусанова Гулрухсор Манажоновна	4	N	<null>	1	1	Ширмонбулок кишлог
46	<null>	Хушназарова Дилноза Рустамовна	4	N	<null>	1	1	Кипчок 1-тор куча 37
47	<null>	Эралиева Эргашхон Абдулхамидовна	4	N	<null>	1	1	Куштепа кишлоги бо
48	<null>	Джабборова Динара Уткировна	4	N	<null>	1	1	Дустлик ЖК кутарма
49	<null>	Косимова Сурайё Ёркиновна	4	N	<null>	1	1	Пастки кишлок МФЙ
50	<null>	Искандарбеков Дилшодбек Искандарбекович	4	N	<null>	1	1	Хортум КФЙ улугбек
51	<null>	Рузимбетов Ёркинбек Хабибуллаевич	4	N	<null>	1	1	Бустон шаҳри Бобур-
52	<null>	Рузимбетов Ёркинбек Хабибуллаевич	4	N	<null>	1	1	Бустон шаҳри Бобур-
53	<null>	Муллажоновна Гулмира Абдуллаевна	4	N	<null>	1	1	Дурмон кушлоги 6
54	<null>	Исмоилова Дилноза Шокиржоновна	4	N	<null>	1	1	Кумгузор
56	<null>	Akbarov Sarvar	4	<null>	<null>	<null>	<null>	<null>

Maydon qiymatlarini o'zgartirish.

Bu o'zgartirish UPDATE komandasi yordamida bajariladi. Bu komandada UPDATE ifodasidan so'ng jadval nomi va SET ifodasidan so'ng ma'lum ustun uchun o'zgartirish ko'rsatiladi. UPDATE ikki formatga ega. Ulardan birinchisi:

```
UPDATE <table name | view name>
```

```
SET column = expression [, column = expression] ...
```

```
[WHERE search-condition]
```

bu erda expression - bu ustun | ifoda | konstanta | o'zgaruvchi.

Masalan, stipeniya jadvalidan barcha talabalarning stipendiya miqdorlarini 61200ga almashtirish komandasi.

```
UPDATE stip SET summa = 61200;
```

SNUM	ID_TALABA	OY	SUMMA	KASABA	QULGA
1	1	январ	61 200,00	1,00	506,66
2	2	январ	61 200,00	1,00	506,66
3	3	январ	61 200,00	1,00	506,66
4	4	январ	61 200,00	1,00	506,66
5	5	январ	61 200,00	1,00	506,66
6	6	январ	61 200,00	1,00	506,66
7	7	январ	61 200,00	1,00	506,66
8	8	январ	61 200,00	1,00	506,66
9	9	январ	61 200,00	1,00	506,66
10	10	январ	61 200,00	1,00	506,66
11	1	Феврал	61 200,00	1,00	506,66
12	2	Феврал	61 200,00	1,00	506,66
13	3	Феврал	61 200,00	1,00	506,66

Ma'lum satrlarni o'zgartirish uchun DELETE dagi kabi predikatdan foydalanish kerak. Masalan talaba jadvalidan Halilova Maxliyo Sodiqjonovni manzilini o'zgartirish uchun quyidagicha komanda ishlatiladi.

```
UPDATE talaba SET adres='Jalolquduq Janubiy olamushuk '  
WHERE id = 42 ;
```


ID	ID_TALABA	FIO	QURS	FIRE	STIPEND	ID_FAK	ID_GR	ADRES
38	<null>	Шокиров Донёр Достонович	4	N	<null>	1	1	Шарк Юлдуз ҚФЙ С, Зунунов 114
39	<null>	Юлдашев Козим Мансурович	4	N	<null>	1	1	Катортол ҚФЙ мирзаев 7
40	<null>	Алимова Саида Комилжоновна	4	N	<null>	1	1	Андижон шаҳар пушкин 15
41	<null>	Рузиматова Шоира Ёркинбековна	4	N	<null>	1	1	Олтиарик тумани Чинортаги куча 13
42	<null>	Халилова Махлиё Содиқоқовна	4	N	<null>	1	1	Jalolquduq Janubiy olamshuk
43	<null>	Юлдашева Махсума Ариповна	4	N	<null>	1	1	Мустақиллик 26 уй
44	<null>	Сиддикова Зарифа Уткировна	4	N	<null>	1	1	Ёркишлок мағдалиев кучаси
32	<null>	Атабоев Дилдора Нодирбековна	4	N	<null>	1	1	Ширнонбулок кишлоги Худойберганаов 13
33	<null>	Холмирзаев Шерали Иқромович	4	N	<null>	1	1	Куштена ЖК 16
34	<null>	Абидова Махпуза Тоҳировна	4	N	<null>	1	1	Ойим киш, 49
22	<null>	Джалпоров Илёс Илхомович	4	N	<null>	1	1	<null>
23	<null>	Шералиев Иқром Исломович	4	N	<null>	1	1	<null>
24	<null>	Ононова Гулзира Акбаралиевна	4	N	<null>	1	1	<null>
25	<null>	Туйчиева Хуррият Акбаралиевна	4	N	<null>	1	1	<null>
35	<null>	Мусабекова Зулайхо Дилшодбековна	4	N	<null>	1	1	Куштена ж-х 49
36	<null>	Тошматова Мархабо Аскарвна	4	N	<null>	1	1	Маданият киш Кориниёзий 16
26	<null>	Таджибаева Махлиё Зухритдиновна	4	N	<null>	1	1	<null>
27	<null>	Таштанкулова Наргиса Толиповна	4	N	<null>	1	1	<null>
28	<null>	Джурабоева Максуда Мансурвна	4	N	<null>	1	1	<null>
29	<null>	Кодиров Уткир Азамович	4	N	<null>	1	1	<null>
30	<null>	Бойматова Оязим Донёровна	4	N	<null>	1	1	<null>
31	<null>	Каюнов Элёр Кодирович	4	N	<null>	1	1	<null>
37	<null>	Буриев Хикматилло Хамидович	4	N	<null>	1	1	Абдуллабий ҚФЙ
45	<null>	Хусанова Гулрухсор Мамажоновна	4	N	<null>	1	1	Ширнонбулок кишлоги янги маҳалла 3

Bir paytni o'zida bir necha maydon qiymatlarini almashtirish quyidagicha komanda yordamida almashtiriladi

SELECT so'rov operatori

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.

Operator SQL92 standartiga ko'ra quyidagi ko'rinishga ega:

SELECT -- ALL ----- sxema , ustun ----

-- DISTINCT -- ---- * -----

FROM -- sxema , Jadval .. -----

WHERE -- izlash sharti -----

GROUP BY -- sxema , ustun -----

HAVING -- izlash sharti -----

ORDER BY - tartiblash spetsifikatori -----

OFFICES jadvalidagi hamma yozuvlarni qaytaruvchi sodda so'rov ko'ramiz.

SELECT * FROM OFFICES [4]

SELECT yordamida ma'lumotlarni tanlash

SELECT operatori albatta "qaytariluvchi ustunlar ro'yxati " ni o'z ichiga olishi kerak.

Misol: Hamma talabalar familiyasi , ismi va shartnoma summalarini ruyxatini xosil qilish

SELECT Fam, Ismi, Pasport FROM Talaba

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

Bu xol yuz bermasligi uchun DISTINCT operatoridan foydalanish lozim.

Masalan, quyidagicha:

SELECT DISTINCT ID_talaba from Stip

SELECT operatori WHERE sharti

Endi WHERE ifodasidan foydalanib ba'zi so'rovlarni ko'rib chiqamiz:

Talabalar jadvalidan 2 kursda ukiydigan talabalarni ruyxatini xosil kilamiz.

Select fam, Ismi, passport from talaba where qurs=2

Identifikatori 4 ga teng bo'lgan talabaning familiyasi , ismi , passport malumotlarini ko'rsating:

Select fam, Ismi, passport from talaba where Snum=4

Agar izlash sharti TRUE, bo'lsa qator natijaviy to'plamga qo'shiladi, agar izlash sharti FALSE bo'lsa, 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, kerakligini qoldiruvchi filtr sifatida ishlatiladi!

SQL mantiqiy operatorlari.

BETWEEN va IN Operatorlari

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

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

Misol: Tug'ilgan yili xar xil diapozonga mos keluvchi talabalarni topish

Select fam, ismi , born_dat from talaba where born_dat between 1985 and 1987

NOT ifodasi yordamida berilgan diapazonga tegishlilikni tekshirish mumkin, masalan: talabalarning tug'ilgan yili 1985 dan 1987 gacha tushmaydigan talabalar ro'yxati

Select fam, ismi , born_dat from talaba where born_dat not between 1985 and 1987

Ikkita konkret yilda tug'ilgan talabalarni aniqlash.

Select fam, ismi , born_dat from talaba where born_dat in(1985,1987)

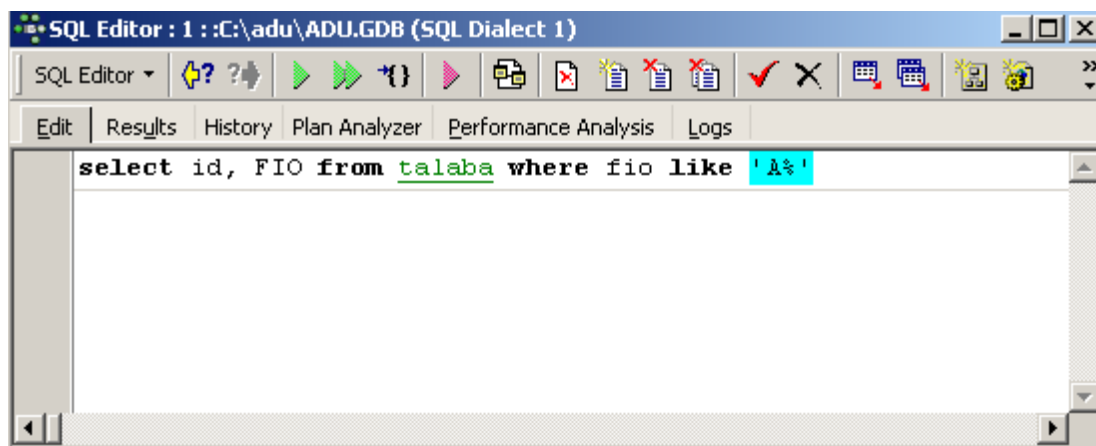
NOT IN yordamida diapazonga "tegishli emaslikni " tekshirish mumkin.

Operator LIKE

Quyidagicha '%' shablonli LIKE operatorini qo'llaymiz:

Select fam, ismi , born_dat from talaba where Fam like 'A%'

Bu erda familiyasi A xarfi bilan boshlanadigan talabalar ko'rsatiladi.



Agar faqat bitta simvol ixtiyoriy bo'lsa '_' shabloni qo'llanadi. Masalan:

Select * from talaba where FIO like 'M_minov %'

ID	CODE	FIO	FAK_C...	DA
102 289	0	Муминов Мухаммад	00	01.
102 344	0	Муминов Равшан	00	01.
109 764	0	Муминов Аброр (диф тенг)	8	23.
112 767	167-C 051107	Мўминова Дилбархон Иномжоновна	00	<P>

Bu erda Mo'minov yoki Muminov familiyasi barcha talabalar ro'yxati aks ettiriladi.

Operator IS NULL

Talaba jadvalidan familiyasi yozilmagan talabalar ro'yxati chikarilsin

Select * from talaba where FIO is null

ID	CODE	FIO	FAK_C...	DATA_POST
111 843	00	<null>	00	<null>
111 844	00	<null>	00	<null>
111 845	00	<null>	00	<null>
113 163	00	<null>	00	<null>
113 164	00	<null>	00	<null>
113 165	00	<null>	00	<null>
113 166	00	<null>	00	<null>
113 167	00	<null>	00	<null>
113 168	00	<null>	00	<null>
113 169	00	<null>	00	<null>
113 170	00	<null>	00	<null>

NOT sharti

Talabalar jadvalidan FIO maydonida fakat familiya yozilgan qatorlarni chiqarish

Select * from talaba where FIO is not null

ID	CODE	FIO	FAK_C...	DATA_POST
102 303	0	Ҳамидова Назокат	00	01.09.2003 00:00
102 304	0	Ҳолмуродов Абдусалом	00	01.09.2003 00:00
102 305	0	Умурзаков Шохруҳ	00	01.09.2003 00:00
102 306	0	Сидиков Донёрбек	00	01.09.2003 00:00
102 307	0	Ахмадбеков Махсудбек	00	01.09.2003 00:00
102 309	0	Нишонава Хилола	00	01.09.2003 00:00
102 310	0	Ваҳобов Авазбек	00	01.09.2003 00:00
102 311	0	Исаев Баходир	00	01.09.2003 00:00
102 312	0	Мамазиёва Хилола	00	01.09.2003 00:00
102 313	-C	Болтабоева Муборак	7	01.09.2003 00:00
102 314	0	Раҳимова Моҳира	00	01.09.2003 00:00
102 315	0	Кучкарова Чуллоной	00	01.09.2003 00:00
102 316	0	Атабоев Ҳуршид	00	01.09.2003 00:00
102 317	0	Қосимова Мастура	00	01.09.2003 00:00
102 318	0	Раззоков Элёр	00	01.09.2003 00:00

Yozuvlarni tartiblash, **ORDER BY** jumlasini

Oldin ko'rilgan so'rovlarda natijalar ixtiyoriy tartibda olingan edi. Agar o'quvchilar ro'yxatini alfabit tartibida yoki tovarlar narxini kamayish tartibida chiqarish zarur bo'lsachi? Buning uchun

SELECT operatori tarkibida ORDER BY ifodasi ko'zda tutilgan.

Avval quyidagi misolni ko'ramiz:

Select * from talaba order by FIO

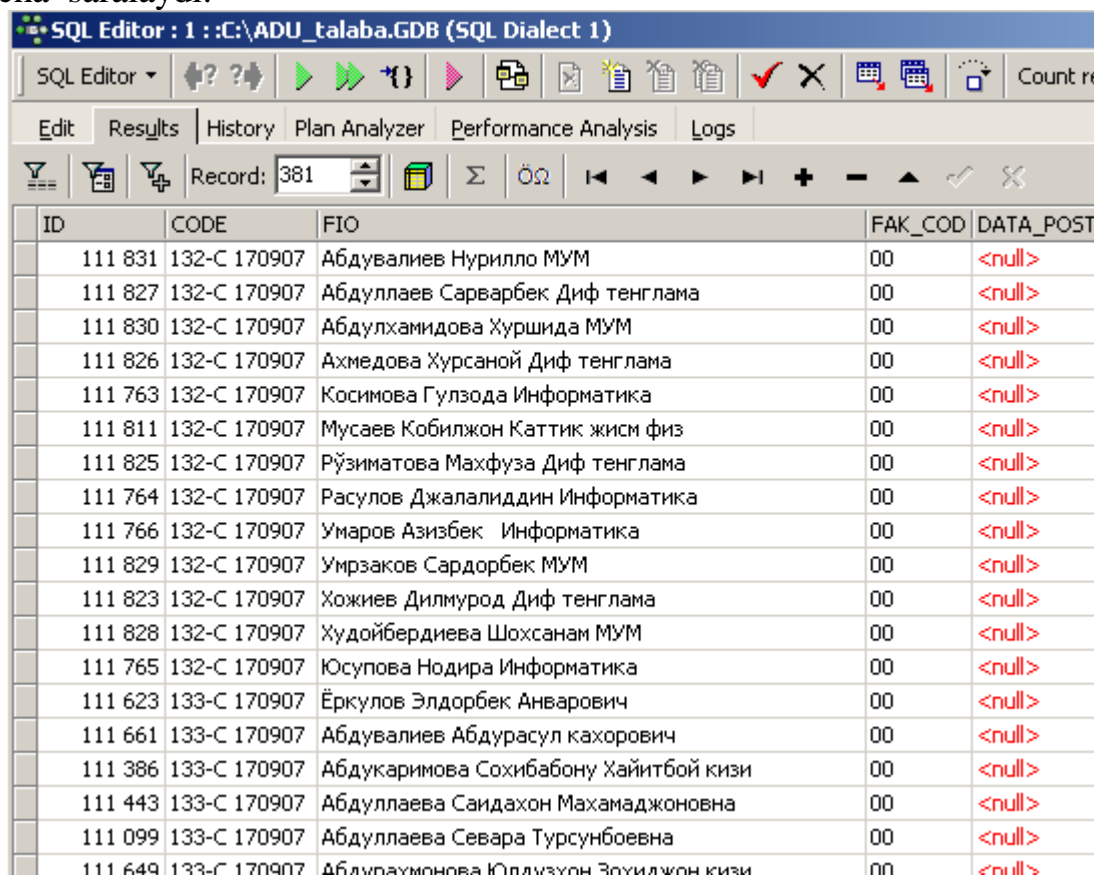
ID	CODE	FIO	FAK
111 661	133-C 170907	Абдувалиев Абдурасул кахорович	00
111 831	132-C 170907	Абдувалиев Нурилло МУМ	00
111 741	177-C 161107	Абдувалиева Наргизахон Равшанбековна	00
101 149	0	Абдугофурова Азиза	00
111 386	133-C 170907	Абдукаримова Соҳибабону Хайитбой кизи	00
108 855	192-C 021006	Абдуқодиров Бобурмирзо	00
100 967	0	Абдуллаев Абдуқодир	00
113 062	180-C 271107	Абдуллаев Қудратилло Набиқонович МУМ	00
111 827	132-C 170907	Абдуллаев Сарварбек Диф тенглама	00
100 985	0	Абдуллаев Шоят	00
101 043	0	Абдуллаева А	00
109 659	192-C 021006	Абдуллаева Қомилжон	00
101 163	0	Абдуллаева Машхура	00
111 443	133-C 170907	Абдуллаева Саидахон Махамаджоновна	00
111 099	133-C 170907	Абдуллаева Севара Турсунбоевна	00
101 194	0	Абдуллаева Шахноза	00
111 597	167-C 051107	Абдуллажонов Илхонжон Иброҳимжон ўгли	00
111 830	132-C 170907	Абдулхамидова Ҳуршида МУМ	00
109 795	0	Абдуманнонова Наргиза (мат)	7
101 052	0	Абдуолимова X	00

Bu erda talabalar ro'yxatini familiya bo'yicha saralaydi.

ORDER BY ifodasidan keyin kelgan ustun ASOSIY kalitdir, undan keyingi ustunlar ikkinchi darajali kalitlardir. Yozuvlarni o'sish hamda kamayish bo'yicha tartiblash mumkin.

Select * from talaba order by code , FIO

Bu erda talabalar ruyxatini birinchi buyruqlar kodi keyin familiya buyicha saralaydi.



The screenshot shows a SQL Editor window with a table of student records. The table has five columns: ID, CODE, FIO, FAK_COD, and DATA_POST. The records are sorted by CODE in ascending order. The FIO column contains the full names of the students in Cyrillic script. The FAK_COD column contains the value '00' for all records. The DATA_POST column contains the value '<null>' for all records.

ID	CODE	FIO	FAK_COD	DATA_POST
111 831	132-C 170907	Абдувалиев Нурилло МУМ	00	<null>
111 827	132-C 170907	Абдуллаев Сарварбек Диф тенглама	00	<null>
111 830	132-C 170907	Абдулхамидова Хуршида МУМ	00	<null>
111 826	132-C 170907	Ахмедова Хурсаной Диф тенглама	00	<null>
111 763	132-C 170907	Косимова Гулзода Информатика	00	<null>
111 811	132-C 170907	Мусаев Кобилжон Каттик жисм физ	00	<null>
111 825	132-C 170907	Рўзиматова Махфуза Диф тенглама	00	<null>
111 764	132-C 170907	Расулов Джалалиддин Информатика	00	<null>
111 766	132-C 170907	Умаров Азизбек Информатика	00	<null>
111 829	132-C 170907	Умрақов Сардорбек МУМ	00	<null>
111 823	132-C 170907	Хожиев Дилмурод Диф тенглама	00	<null>
111 828	132-C 170907	Худойбердиева Шохсанам МУМ	00	<null>
111 765	132-C 170907	Юсупова Нодира Информатика	00	<null>
111 623	133-C 170907	Ёркулов Элдорбек Анварович	00	<null>
111 661	133-C 170907	Абдувалиев Абдурасул кахорович	00	<null>
111 386	133-C 170907	Абдукаримова Сохибабону Хайитбой кизи	00	<null>
111 443	133-C 170907	Абдуллаева Саидахон Махамаджоновна	00	<null>
111 099	133-C 170907	Абдуллаева Севара Турсунбоевна	00	<null>
111 649	133-C 170907	Абдурахмонова Юлдузхон Зокиржон кизи	00	<null>

Stipendiya olgan talabalar xajmlarini DESC predikatini kullab Kamayish tartibida chiqaramiz. O'sish tartibida chiqarish uchun ASC predikati qo'llaniladi. Bu predikat ko'zda tutilgan bo'lib, uni ko'rsatish shart emas. Agar ustun xisoblanuvchi bo'lib, nomga ega bo'lmasa uning tartib nomerini ko'rsatish mumkin.

Bir necha jadvallar bilan ishlash.

Jamlashtirish relyatsion ma'lumotlar bazasi operatsiyalaridan biri bo'lib, jadvallar orasidagi aloqani belgilaydi va ulardan ma'lumotni bitta komanda yordamida ajratishga imkon beradi. Har xil jadvallarda bir xil nomli ustunlar bo'lishi mumkin bo'lgani uchun, kerakli ustun uchun jadval nomi prefiksi ishlatiladi.

Jamlashda jadvallar FROM ifodasidan so'ng ro'yxat sifatida tasvirlanadi. So'rov predikati ixtiyoriy jadval ixtiyoriy ustuniga tegishli bo'lishi mumkin. Jamlash eng soddasi bu dekart ko'paytmasi, uni quyidagicha bajarish mumkin:

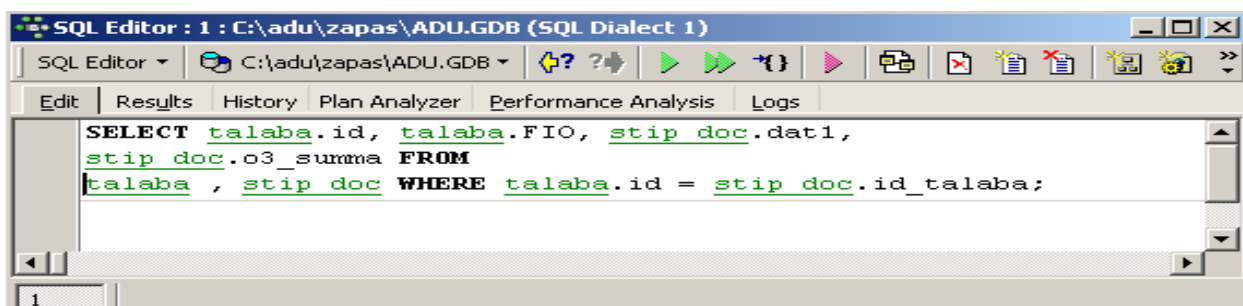
```
SELECT talaba.*, Stip.* FROM talaba, Stip;*
```

Lekin bu erda hosil bo'lgan jadval keraksiz ma'lumotlarga ega. Keraksiz satrlarni olib tashlash uchun WHERE jumlasidan foydalaniladi.

Masalan: Talabalar umumiy ma'lumoti stipendiya bilan birgalikda aks ettirilishi uchun quyidagicha komanda ishlatiladi:

```
SELECT talaba.id, talaba.FIO, stip_doc.dat1, stip_doc.o3_summa  
FROM talaba , stip_doc  
WHERE talaba.id = stip_doc.id_talaba;
```

Bu erda xar bir talaba uchun mos oylar va stipendiya miqdorlari ro'yxati chiqariladi.



The screenshot shows a window titled "SQL Editor : 1 : C:\adu\zapas\ADU.GDB (SQL Dialect 1)". The editor contains the following SQL query:

```
SELECT talaba.id, talaba.FIO, stip_doc.dat1,  
stip_doc.o3_summa FROM  
talaba , stip_doc WHERE talaba.id = stip_doc.id_talaba;
```

The interface includes a menu bar with "Edit", "Results", "History", "Plan Analyzer", "Performance Analysis", and "Logs". A toolbar with various icons is located below the menu bar. The query text is displayed in a monospaced font with syntax highlighting.

ID	FIO	DAT1	O3_SUMMA
100 074	Рахимова Халима	01.09.2006 00:00	30 158,00
100 074	Рахимова Халима	01.03.2007 00:00	30 158,00
100 074	Рахимова Халима	01.09.2007 00:00	<null>
100 074	Рахимова Халима	01.03.2008 00:00	<null>
100 075	Комилова Гулнора	01.09.2006 00:00	20 106,00
100 075	Комилова Гулнора	01.03.2007 00:00	30 158,00
100 075	Комилова Гулнора	01.09.2007 00:00	<null>
100 075	Комилова Гулнора	01.03.2008 00:00	<null>
100 076	Юлдашев Акмал	01.09.2006 00:00	30 158,00
100 076	Юлдашев Акмал	01.03.2007 00:00	30 158,00
100 076	Юлдашев Акмал	01.09.2007 00:00	<null>
100 076	Юлдашев Акмал	01.03.2008 00:00	<null>
100 077	Гофурова Феруза	01.09.2006 00:00	30 158,00
100 077	Гофурова Феруза	01.03.2007 00:00	30 158,00

Tenglikka asoslangan predikatlardan foydalanuvchi jamlanmalar, tenglik bo'yicha jamlanma deb atalib, jamlanmalarning eng umuiy ko'rinishidir. Shu bilan birga ixtiyoriy relyatsion operatoridan foydalanish mumkin.

Sodda joylashtirilgan ostki so'rovlar.

SQL yordamida so'rovlarni bir birining ichiga joylashtirishingiz mumkin. Odatda ichki so'rov qiymat xosil qiladi va bu qiymat tashqi predikat tomonidan tekshirilib, to'g'ri yoki noto'g'riligi tekshiriladi.

Misol: Ma'lumotlar bazasidan talaba jadvalidan eng oxirgi kiritilgan talabaning stipendiyalari ro'yxati Buni quyidagicha amalga oshirish mumkin:

```
SELECT talaba.id, talaba.FIO, stip_doc.dat1, stip_doc.o3_summa
FROM talaba , stip_doc
WHERE talaba.id = stip_doc.id_talaba and talaba.id = (select max(id)
from talaba)
```

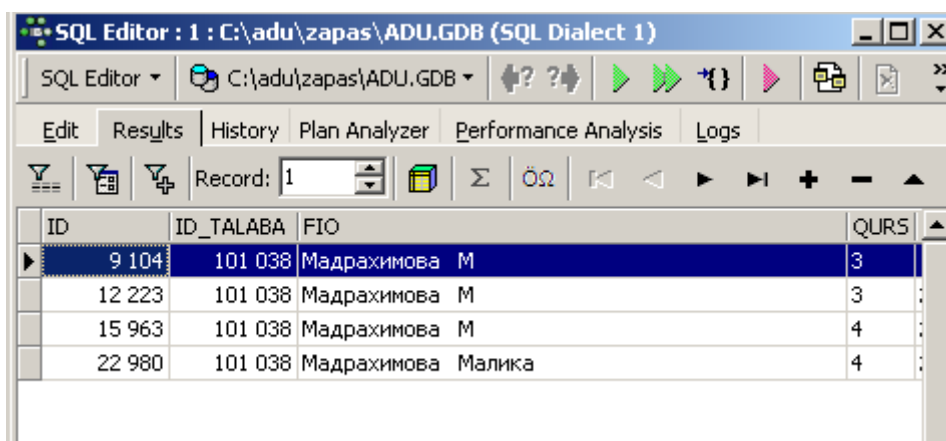
ID	FIO	DAT1	O3_SUMMA
113 238	Маъмурова З 122-С	01.03.2008 00:00	<null>

Avval ichki so'rov bajariladi, so'ngra uning natijasi tashqi so'rovni xosil qilish uchun ishlatiladi (SNum ostki so'rov natijasi bilan solishtiriladi).

Ostki so'rov bitta ustun tanlashi lozim, bu ustun qiymatlari tipi predikatda solishtiriladigan qiymat tipi bilan bir xil bo'lishi kerak. Siz ba'zi xollarda ostki so'rov bitta qiymat xosil qilishi uchun DISTINCT operatoridan foydalanishingiz mumkin.

Misol: Talabalik kodi 101038 nomerli bo'lgan talabani stipendiyalari to'g'risidagi xamma ma'lumotlarni olish lozim bo'lsin. Buning uchun quyidagi so'rov ishlatiladi.

```
SELECT * FROM stip_doc  
WHERE id_talaba = ( SELECT distinct id_talaba FROM  
stip_doc WHERE id_talaba = 101038 );
```



ID	ID_TALABA	FIO	QURS
9 104	101 038	Мадрахимова М	3
12 223	101 038	Мадрахимова М	3
15 963	101 038	Мадрахимова М	4
22 980	101 038	Мадрахимова Малика	4

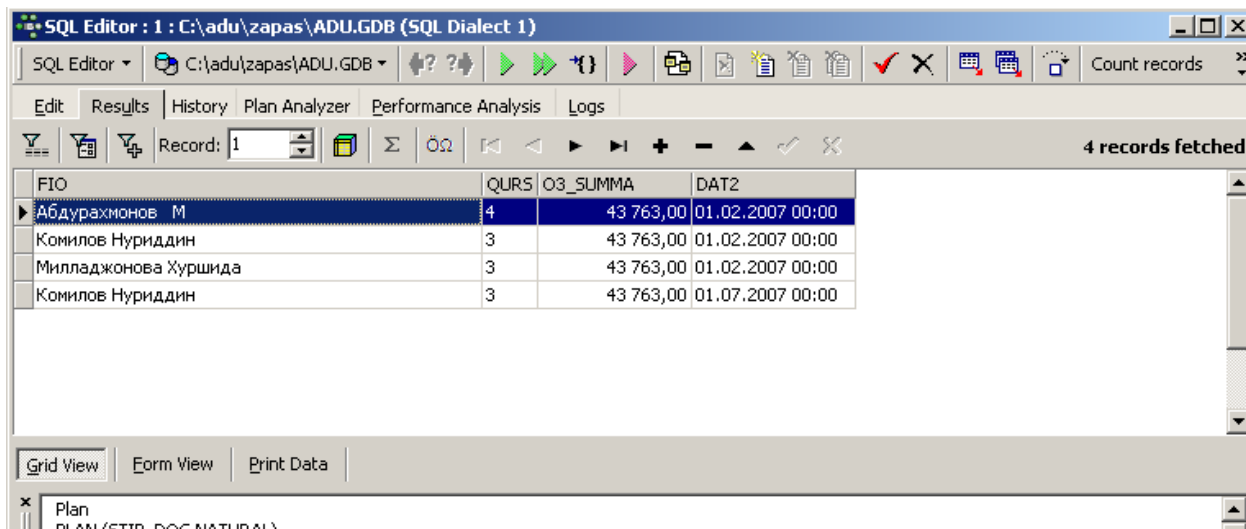
Bu xolda ostki so'rov faqat bitta 101038 qiymat chiqaradi, lekin umumiy xolda bir necha qiymatlar bo'lishi mumkin va ular ichidan DISTINCT faqat bittasini tanlaydi.

Ixtiyoriy sondagi satrlar uchun avtomatik ravishda bitta qiymat xosil qiluvchi funksiya turi - agregat funksiya bo'lib, undan ostki so'rovda foydalanish mumkin.

Masalan, Stipendiya jadvalidan 01.07.2007 davrida eng ko'p stipendiya olgan talabalar ro'yxatini chiqarish uchun quyidagi so'rov ishlatiladi:

```
SELECT stip_doc.fio, stip_doc.qurs, stip_doc.o3_summa, stip_doc.dat2  
FROM stip_doc  
WHERE stip_doc.o3_summa = ( SELECT max(o3_summa) FROM
```

stip_doc where dat2='01.07.2007');

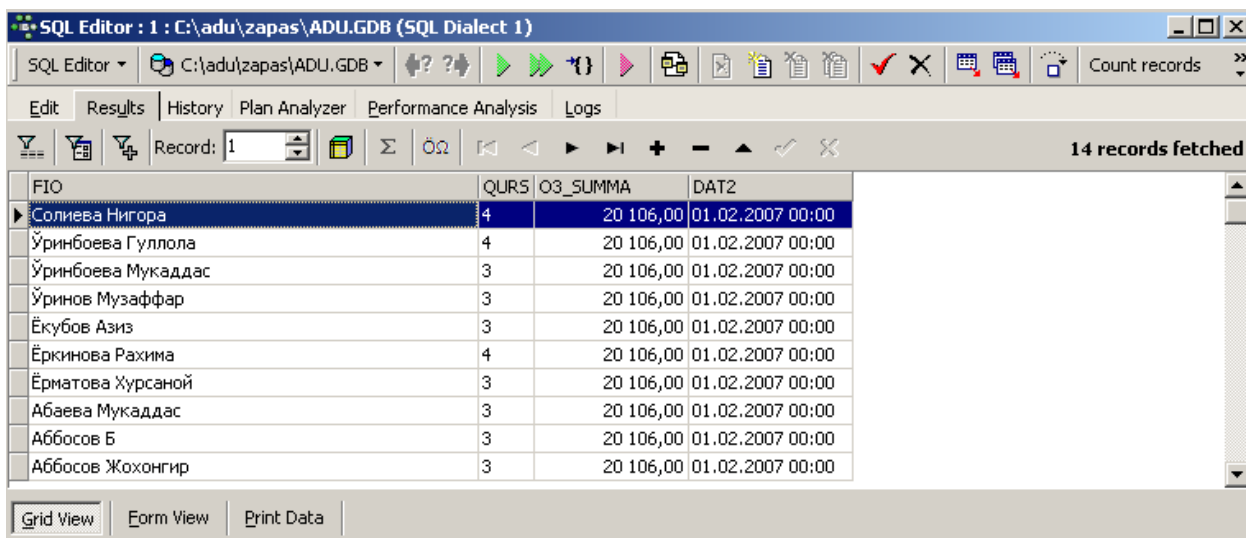


The screenshot shows the SQL Editor interface with a query result table. The table has four columns: FIO, QURS, O3_SUMMA, and DAT2. The first three records have a QURS value of 3 and an O3_SUMMA of 43 763,00, with a DAT2 of 01.02.2007 00:00. The fourth record has a QURS value of 3 and an O3_SUMMA of 43 763,00, with a DAT2 of 01.07.2007 00:00. The interface also shows a toolbar with various icons and a status bar indicating '4 records fetched'.

FIO	QURS	O3_SUMMA	DAT2
Абдурахмонов М	4	43 763,00	01.02.2007 00:00
Комилов Нуриддин	3	43 763,00	01.02.2007 00:00
Милладжоновна Хуршида	3	43 763,00	01.02.2007 00:00
Комилов Нуриддин	3	43 763,00	01.07.2007 00:00

Stipendiya jadvalidan 01.07.2007 davrida eng kam stipendiya olgan talabalar ro'yxatini chiqarish uchun esa quyidagi so'rov ishlatiladi:

```
SELECT stip_doc.fio, stip_doc.qurs, stip_doc.o3_summa, stip_doc.dat2  
FROM stip_doc  
WHERE stip_doc.o3_summa = ( SELECT min(o3_summa) FROM  
stip_doc where dat2='01.07.2007' );
```



The screenshot shows the SQL Editor interface with a query result table. The table has four columns: FIO, QURS, O3_SUMMA, and DAT2. The first record has a QURS value of 4 and an O3_SUMMA of 20 106,00, with a DAT2 of 01.02.2007 00:00. The remaining 13 records have a QURS value of 3 and an O3_SUMMA of 20 106,00, with a DAT2 of 01.02.2007 00:00. The interface also shows a toolbar with various icons and a status bar indicating '14 records fetched'.

FIO	QURS	O3_SUMMA	DAT2
Солиева Нигора	4	20 106,00	01.02.2007 00:00
Ўринбоева Гуллола	4	20 106,00	01.02.2007 00:00
Ўринбоева Мукаддас	3	20 106,00	01.02.2007 00:00
Ўринов Музаффар	3	20 106,00	01.02.2007 00:00
Ёкубов Азиз	3	20 106,00	01.02.2007 00:00
Ёркинова Рахима	4	20 106,00	01.02.2007 00:00
Ёрматова Хурсаной	3	20 106,00	01.02.2007 00:00
Абаева Мукаддас	3	20 106,00	01.02.2007 00:00
Аббосов Б	3	20 106,00	01.02.2007 00:00
Аббосов Жохонгир	3	20 106,00	01.02.2007 00:00

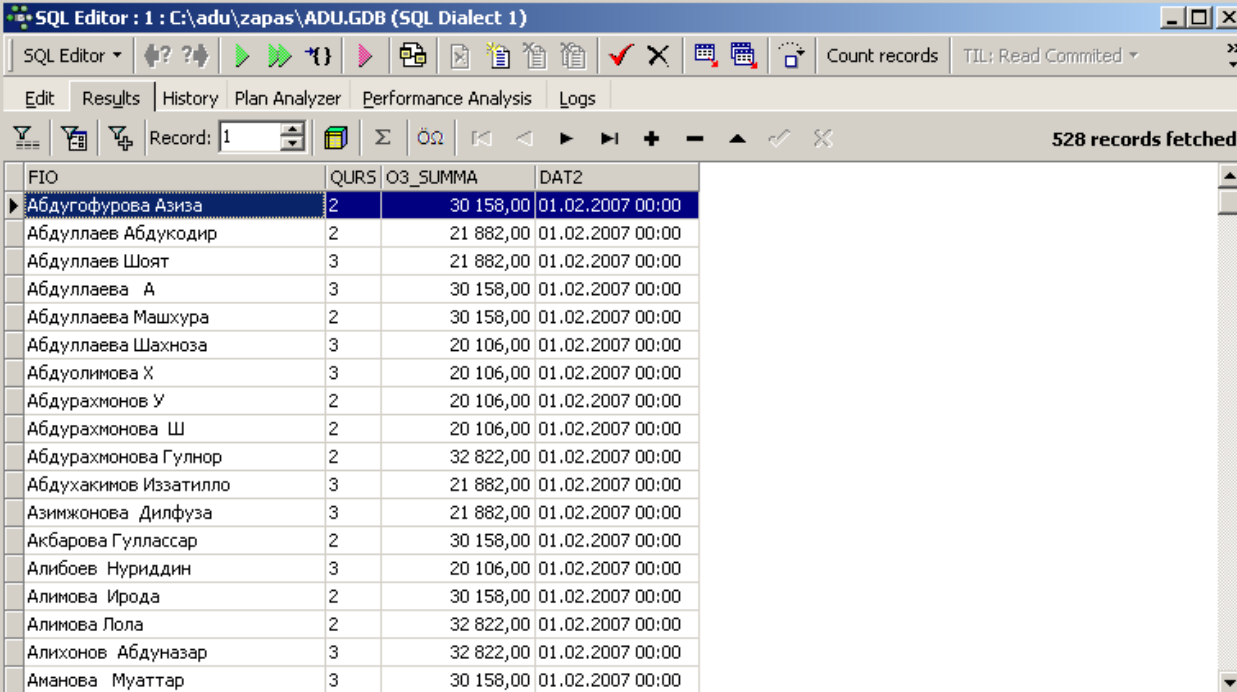
Shuni nazarda tutish kerakki guruxlangan agregat funktsiyalar GROUP BY ifodasi terminlarida aniqlangan agregat funktsiyalar bo'lsa ko'p qiymatlar xosil qilishi mumkin.

Agar ostki so'rov IN operatoridan foydalanilsa, ixtiyoriy sondagi satrlar xosil qilish mumkin.

Misol: Matematika fakultetidagi barcha gurux talabariini stipendiyalari to'grisidagi ma'lumotni olish so'rovnomasi quyidagicha yoziladi.

```
SELECT stip_doc.fio, stip_doc.qurs, stip_doc.o3_summa, stip_doc.dat2  
FROM stip_doc WHERE id_gr in ( SELECT id FROM sp_gruppa  
where id_fak=7);
```

id_gr –gruppa kodi, id_fak – fakultet kodi

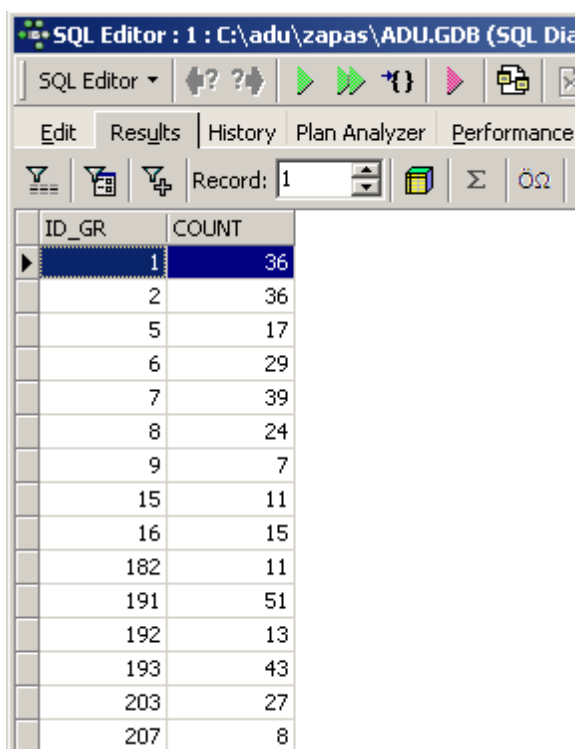


FIO	QURS	O3_SUMMA	DAT2
Абдугофурова Азиза	2	30 158,00	01.02.2007 00:00
Абдуллаев Абдукодир	2	21 882,00	01.02.2007 00:00
Абдуллаев Шоят	3	21 882,00	01.02.2007 00:00
Абдуллаева А	3	30 158,00	01.02.2007 00:00
Абдуллаева Машхура	2	30 158,00	01.02.2007 00:00
Абдуллаева Шахноза	3	20 106,00	01.02.2007 00:00
Абдуолимова Х	3	20 106,00	01.02.2007 00:00
Абдурахмонов У	2	20 106,00	01.02.2007 00:00
Абдурахмонова Ш	2	20 106,00	01.02.2007 00:00
Абдурахмонова Гулнор	2	32 822,00	01.02.2007 00:00
Абдухакимов Иззатилло	3	21 882,00	01.02.2007 00:00
Азимжонов Дилфуза	3	21 882,00	01.02.2007 00:00
Акбарова Гулласар	2	30 158,00	01.02.2007 00:00
Алибоев Нуриддин	3	20 106,00	01.02.2007 00:00
Алимова Ирода	2	30 158,00	01.02.2007 00:00
Алимова Лола	2	32 822,00	01.02.2007 00:00
Алихонов Абдуназар	3	32 822,00	01.02.2007 00:00
Аманова Муаттар	3	30 158,00	01.02.2007 00:00

Bu natijani jamlanma orqali xosil qilish mumkin. Lekin odatda ostki so'rovli so'rovlar tezroq bajariladi. Siz ostki so'rov SELECT jumlasida ustunga asoslangan ifodadan foydalanishingiz mumkin. Bu relyatsion operatorlar yordamida yoki IN yordamida amalga oshirilishi mumkin. Siz ostki so'rovlarni HAVING ichida ishlatishingiz mumkin. Bu ostki so'rovlar agar ko'p qiymatlar qaytarmasa xususiy agregat funktsiyalaridan yoki GROUP BY yoki HAVING operatorlaridan foydalanishi mumkin.

Misol: Matematika fakultetining '01.07.2007' davrda guruxlar buyicha stipendiya tayyorlangan talabalar sonini aniqlash so'rovnomasi.

```
SELECT id_gr ,count(fio) FROM stip_doc where dat2='01.07.2007'
GROUP BY id_gr
HAVING id_gr in ( SELECT id FROM sp_gruppa where id_fak=7);
```



ID_GR	COUNT
1	36
2	36
5	17
6	29
7	39
8	24
9	7
15	11
16	15
182	11
191	51
192	13
193	43
203	27
207	8

2.5. So'rovlarda guruhlash va funksiyalar.

Agregat funksiyalar qo'llanishi

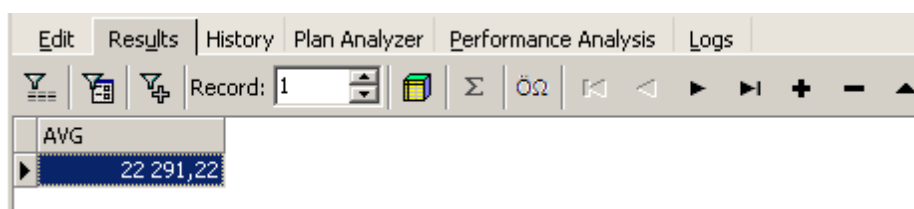
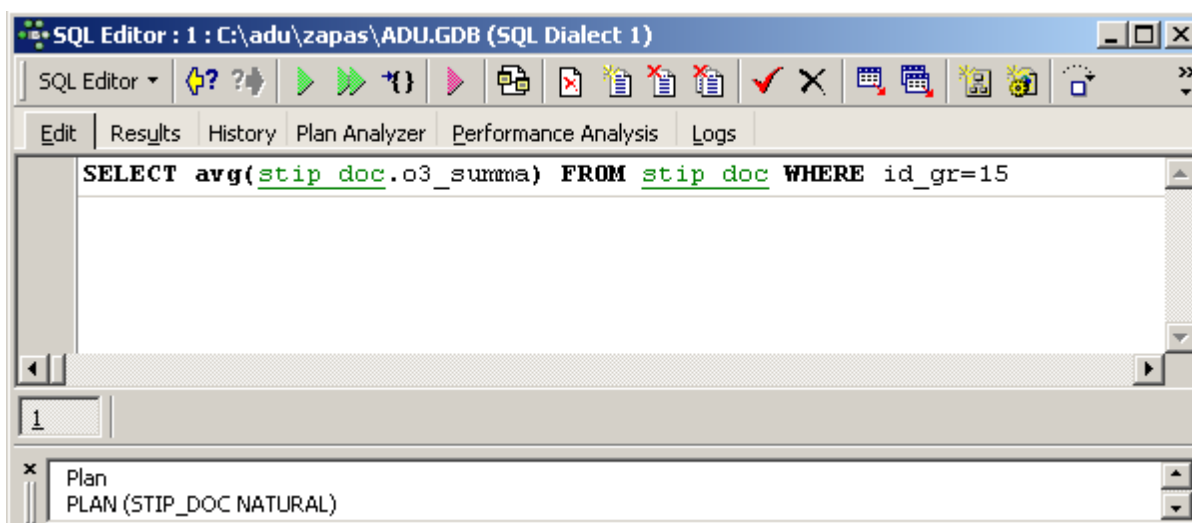
Agregat (yoki STATIK) funksiyalar, sonli yoki hisoblanuvchi ustunlar bilan ishlaydi. Agregat funksiya argumenti butun ustun bo'lib, bitta qiymat qaytaradi.

Bu funksiyalarni ko'rib chiqamiz:

- SUM() - Ustundagi hamma qiymatlar summasini hisoblaydi.
- AVG() - Ustundagi hamma qiymatlar o'rtasi qiymatini hisoblaydi.
- MIN() - Ustundagi hamma qiymatlar eng kichigini aniqlaydi.
- MAX() - Ustundagi hamma qiymatlar eng kattasini aniqlaydi.
- COUNT() - Ustundagi qiymatlar sonini hisoblaydi.
- COUNT(*) - So'rov natijalari jadvalidagi satrlar sonini hisoblaydi.

Agregatlash argumenti bo'lib ustun nomidan tashqari ixtiyoriy matematik ifoda xizmat qilishi mumkin. Misol uchun quyidagi so'rovda: 15-gurux bo'yicha stipendiya summasi qanchaligini aniqlash so'rovnomasi.

SELECT avg(stip_doc.o3_summa) FROM stip_doc WHERE id_gr=15



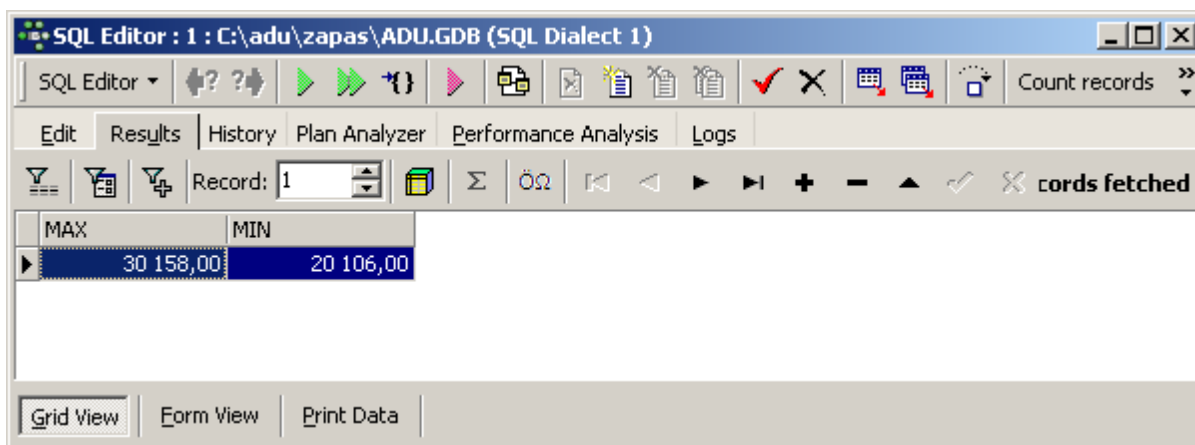
Ustunlar summasini hisoblab ko'ramiz. SUM() funksiyasini qo'llaymiz, buning uchun ustun albatta sonli bo'lishi kerak! Masalan, quyidagicha:

15 gurux bo'yicha jami xisoblangan stipendiya summasini aniqlash so'rovnomasi.

SELECT sum(stip_doc.o3_summa) FROM stip_doc WHERE id_gr=15

Ekstremumlarni topish funksiyalari yani MIN(), MAX() funksiyalarini ko'ramiz. Bu funksiyalar sonli ustunlar, sanalar va satrli o'zgaruvchilar bilan ishlaydi. Eng sodda qo'llanishi sonlar bilan ishlash. Masalan quyidagi so'rov beramiz: 15 gurux buyicha eng ko'p va eng kam stipendiya miqdorini aniqlash bo'yicha so'rovnoma.

SELECT max(stip_doc.o3_summa),min(stip_doc.o3_summa) FROM stip_doc WHERE id_gr=15



Bu sonlarni o'z ichiga olgan ustunlardir. Yana bir so'rov beramiz: Eng oxirigi davrdagi stipendiya sanasi?

SELECT max(stip_doc.dat2) FROM stip_doc WHERE id_gr=15

Satrlar bilan ishlaganda har xil SQL serverlardagi kodirovkalar har xil natija berishi mumkin. Yozuvlar sonini sanash uchun COUNT() qo'llanadi. Bu funksiya son qiymat qaytaradi

Masalan: 15 gurux talabalar soni qancha ?

SELECT COUNT(FIO) FROM talaba WHERE id_gr=15

COUNT(*) funktsiyasi qiymatlar sonini emas, satrlar sonini hisoblaydi.

Quyidagicha yozish mumkin: 1,2,3 kurs talabalar sonini aniqlash so'rovnomasi.

SELECT COUNT(FIO) FROM talaba WHERE qurs<3

Agregatlar va ma'lumotlarni guruxlash

Agregat funksiyalar jadval uchun natijaviy satr xosil qiladi. Masalan:

Buyurtma o'rtacha narxi qancha?

SELECT AVG(AMOUNT)

FROM ORDERS

Masalan, oraliq natijani topish lozim bo'lsin. Bu holda guruxlanishli so'rov yordam beradi. Ya'ni SELECT operatorining GROUP BY ifodasi. Avval GROUP BY ifodasi qatnashgan quyidagi so'rovni ko'ramiz:

Har bir guruh bo'yicha o'rtacha stipendiya miqdori qancha?

SELECT id_gr, avg(stip_doc.o3_summa) FROM stip_doc where id_fak=7 group by id_gr

ID_GR	AVG
1	27 906,93
2	25 320,33
5	26 883,14
6	23 745,52
7	23 239,09
8	23 456,67
9	23 972,15
15	22 291,22
16	23 791,73
182	29 152,80
191	23 062,47
192	22 425,69
193	21 274,84
203	23 084,41
207	24 414,00

id_gr maydoni (guruh kodi) bu xolda guruxlash maydonidir, Ya'ni id_gr maydonning hamma qiymatlari guruxlarga ajratiladi va har bir gurux uchun **avg(stip_doc.o3_summa)** ifodasi hisoblanadi!

Har bir shahocha uchun sotuvlarning rejalashtirilgan xajmi diapazoni qancha?

**SELECT REP_OFFICE, MIN(QUOTA), MAX(QUOTA)
FROM SALESREPS
GROUP BY REP_OFFICE**

Bu erda **REP_OFFICE** shahobcha kodi bo'lib, guruxlash mana shu maydon bo'yicha bajariladi.

Yana bir so'rov: Har bir shahobchada qancha xizmatchi ishlaydi ?

**SELECT REP_OFFICE, COUNT(*)
FROM SALESREPS
GROUP BY REP_OFFICE**

Guruxlash va HAVING yordamida ajratish

Shart bo'yicha satrlarni ajratish uchun WHERE ifodasidan foydalangan edik.

Shart bo'yicha guruxlarni ajratish uchun HAVING operatori mavjuddir. Uning

sintaksisi WHERE operatori bilan bir xil va ulardan birgalikda foydalanish mumkin. Quyidagi so'rovni ko'ramiz:

Buyurtmalar umumiy narxi \$300 dan ortiq xizmatchilar uchun buyurtma o'rtacha narxi qanchaga teng?

```
SELECT REP, AVG(AMOUNT)
FROM ORDERS
GROUP BY REP
HAVING SUM(AMOUNT) > 300
```

Ko'rinib turibdiki HAVING SUM(AMOUNT) > 300 ifodasi satrlarni guruxlash sharti sifatida kelmoqda.

Agar SUM(AMOUNT) > 300 sharti yolg'on bo'lsa, bu guruh natijaviy to'plamdan chiqariladi. Agar rost bo'lsa gurux natijaviy to'plamga kiradi!

Yana bir misol ko'raylik: Ikki va undan ortiq xizmatchiga ega har bir offisning hamma xizmatchilari uchun rejadagi va haqiqiy sotuvlar umumiy xajmini hisoblash.

```
SELECT CITY, SUM(QUOTA), SUM(SALESREPS.SALES)
FROM OFFICES, SALESREPS
WHERE OFFICE = REP_OFFICE
GROUP BY CITY
HAVING COUNT(*) >= 2
```

Bu misolda WHERE va HAVING ifodalari o'z funksiyalarini bajaradilar. Shunga e'tibor berish kerakki HAVING ifodasida agregat funksiyalardan foydalaniladi.