

**O‘ZBEKISTON RESPUBLIKASI AXBOROT TEXNOLOGIYALARI VA
KOMMUNIKATSIYALARINI RIVOJLANTIRISH VAZIRLIGI**

**TOSHKENT AXBOROT TEXNOLOGIYALARI UNIVERSITETI
FARG‘ONA FILIALI**

TATU FF ilmiy kengashi
tomonidan ma’qullandi.

Bayonnoma № _____

«____» _____ 201__ yil.

KOMPYUTER TIZIMLARI KAFEDRASI

**MA’LUMOTLAR BAZASINI BOSHQARISH
TIZIMLARI**

fanidan

AMALIY MASHG‘ULOTLARNI

bajarish uchun

O‘quv-uslubiy ko‘rsatma

Farg‘ona shahri

O'quv-uslubiy ko'rsatma Ma'lumotlar bazasini boshqarish tizimlari faniga oid amaliy mashg'ulotlar o'rin egallagan.

Ushbu o'quv-uslubiy qo'llanma 5330500 – Kompyuter injiniringi bakalavriat yo'nalishi Ma'lumotlar bazasini boshqarish tizimlari fani ishchi dasturi asosida tayyorlangan va unga mos keladi.

O'quv-uslubiy qo'llanma kafedra o'quv-uslubiy seminarida ko'rib chiqildi

Bayonnoma № _____

«_____» _____ 201__ yil.

Tuzuvchilar: _____

kat.o'q. A.Dilshodov

ass.G'.G.Pulatov

ass.S.Qodirov

Taqrizchi: _____ dots. A.A. Xolmurzayev

Mundarija:

1. Predmet soxa bo'yicha tizimli taxlil bajarish.
2. Ma'lumotlar bazasi infomantiqiy modeli.
Moxiyat-aloqa modeli.
3. Ma'lumotlar bazasi jadvallarini MS Access da yaratish instrumental vositalari va ma'lumotlar bilan turli amallar bajarish.
4. Ma'lumotlar bazasida modifik, bog'langan jadvallardan foydalanish.
5. Ma'lumotlar bazasi sxemasini normallashtirish.
Normal formalar 1NF, 2NF, ZNF.
6. Bog'langan jadvallar uchun forma yaratish. SQL tili. SQL tili yordamida ma'lumotlar bazasi yaratish.
7. Ma'lumotlar bazasi xisobotini yaratish. SQL tili yordamida so'rovlar tizimini yaratish.
8. MS Access muxitida SQL tilida ma'lumotlar bazasini yaratish
9. SQL tilidagi boshqaruvchi so'rovlar yordamida ma'lumotlar bazasini modifikatsiyalash va jadvallarni yaratish.

1 – AMALIY MASHG'ULOT.

Mavzu: Predmet soha bo'yicha tizimli taxlil bajarish.

Ishdan maqsad: Ma'lum bir predmet soxa uchun ma'lumotlar bazasini taxlil qilishni o'rganish.

Qo'llanish va predmet sohasi

MB ni loyihalashtirishda asosan ikkita masala yechiladi:

- predmet sohasi ob'ektlarini qanday qilib ma'lumotlar modellarining abstrakt obektlari shaklida ifodalash. Ayrim hollarda bu masalaga ma'lumotlar bazasini mantiqiy loyihalash masalasi deyiladi.
- ma'lumotlar bazasiga so'rovlarning bajarilish effektivligini qanday ta'minlash. Bu masalaga ma'lumotlar bazasini fizik loyihalash masalasi deyiladi.

Ixtiyoriy turdagi MBni loyihalashtirishning birinchi bosqichdagi predmet sohasini tahlil qilish bo'lib, u axborot tuzilmasini (kontseptual sxemalar) tuzish bilan yakunlanadi. Bu bosqichda foydalanuvchining so'rovlari tahlil qilinadi, axborot ob'ektlari va uning xarakteristikalarini tanlanadi, hamda o'tkazilgan tahlil asosida predmet sohasi tuzilmalashtiriladi. Predmet sohasini tahlil qilishni uch bosqichga bo'lish maqsadga muvofiqdir:

- kontseptual talablar va axborot ehtiyojlarini tahlil qilish;
- axborot ob'ektlari va ular orasidagi aloqalarni aniqlash;
- predmet sohasining kontseptual modelini qurish va MBni kontseptual sxemasini loyihalashtirish.

Kontseptual talablar va axborot ehtiyojlarini tahlil qilishda quyidagi masalalarni hal qilish kerak:

- foydalanuvchilarning MBga bo'lgan talablarini tahlil qilish;
- MBdan o'rin olishi lozim bo'lgan axborotlarga ishlov berish bo'yicha mavjud masalalarini aniqlash;
- kelajakda hal qilinishi lozim bo'lgan masalalarni aniqlash;
- tahlil natijalarini hujjatlashtirish.

Misol. TATU FF talabalarini hisobga olish uchun MBni ishlab chiqish taklif qilinayapti. Predmet sohasining tahlil qilish:

- 1.TATU FF da qancha talaba ta'lim oladi?
- 2.TATU FF da qancha fakul'tet va kafedralar bor?
- 3.Fakul'tet va kurslar bo'yicha talabalar qanday taqsimlanagan?
- 4.Har bir kursda har bir mutaxassislik bo'yicha qancha fanlar o'qitiladi?
- 5.MBdagi axborotlar qancha (qanday) tez almashtirib turiladi?
- 6.TATU FF da qancha o'qituvchi bor?
- 7.Boshqa shahardan kelgan qancha talaba yotokxonada va xususiy xonadonlarda yashaydi?
- 8.Ma'ruza va amaliy mashg'ulotlar o'tkazish uchun qancha auditoriyalar, hamda qancha laboratoriyalar bor?
- 9.O'qitilayotgan fanlar orasida qanday ketma-ketlik mavjud?

10. Hozirgi paytda 1-9 punktlarda ko'rsatilgan axborotlardan qanday foydalaniladi (dars jadvali, imtihon, sinov va boshqalar) va qanday ishlatish rejalashtirilmoqchi?

11. MBdan kim foydalaniladi, nechta odam va bir kunda necha marta?

Predmet sohasini tahlil qilishning ikkinchi pog'onasi axborot ob'ektlarini tanlash, har bir ob'ekt uchun zarur xossalarni berish, ob'ektlar orasidagi aloqalarni, axborot ob'ektlariga qo'yiladigan cheklashlarni va axborot ob'ektlari orasidagi aloqalarning turlarini aniqlashdan iborat.

Axborot ob'ektlarini tanlayotganda quyidagi savollarga javob berishga harakat qilinadi:

1. MBda saqlanishi lozim bo'lgan ma'lumotlarni qanday sinflarga ajratish mumkin?

2. Har bir ma'lumotlar sinfiga qanday nom berish mumkin?

3. Har bir ma'lumotlar sinfi uchun qanday eng muhim tasifnomalarni (foydalanuvchining nuqtai nazaridan) ajratish mumkin?

4. Tanlangan tasifnomalar to'plamlariga qanday nomlarni berish mumkin?

Axborot ob'ektlarini aniqlash -itaratsion jarayon. U axborotlar oqimining tahlili va iste'molchilar bilan suhbat o'tkazish asosida amalga oshiriladi. Axborot ob'ektlarning tasifnomalari ham xuddi shu usullar bilan aniqlanadi. Keyin axborot ob'ektlari orasidagi aloqalar aniqlanadi. Bu jarayon borishida quyidagi savollarga javob berishga harakat qilinadi:

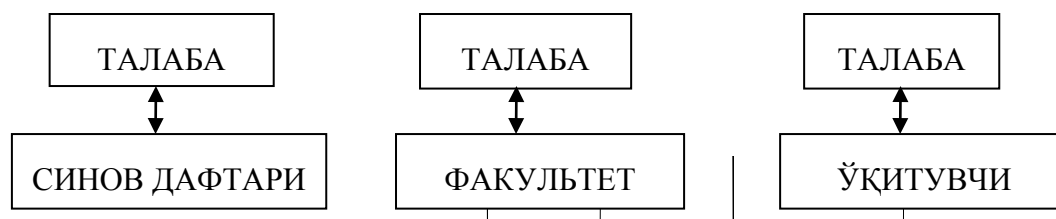
1. Axborot ob'ektlari orasidagi aloqalar turi qanday?

2. Har bir tur aloqalarga qanday nom berish mumkin?

3. Keyinchalik foydalanish mumkin bo'lgan qanday turdagi aloqalar bo'lishi mumkin?

4. Aloqa turlarining biror bir kombinatsiyasi ma'noga egami?

Axborot ob'ektlarining o'zaro aloqasiga misol sifatida TALABA, SINOV_Daftari, ФАКУЛЬТЕТ О'QITUVCHI va axborot ob'ektlari va ular o'rtasidagi o'zaro aloqa quyidagi ra: Бирга бир n. Бирга кўп Кўпга кўп



2 – AMALIY MASHG'ULOT

**Mavzu: Ma'lumotlar bazasi infomantiqiy modeli.
Mohiyat-aloha modeli.**

Ishdan maqsad: Ma'lumotlar bazasining infomantiqiy modelini tuzishni o'rganish va mohiyat – aloqa modelining grafik diagrammasini tashkil qilish bo'yicha tushuncha va ko'nikmalar xosil qilish.

Infomantiqiy model qurilishi

Yuqoridagi ob'ekt va atributlarni analiz qilgan holda loyihalana yotgan ma'lumotlar bazasini mohiyatini belgilash va relyatsion bazani yaratishni hal qilish, uning infomantiqiy modelini tuzish.

Sterj mohiyatga quidagini ko'rsatish mumkin:

1. Yaratuvchilar(Yaratuvchi_kodi, Yaratuvchi). Bu mohiyat asosi kishilar haqida ma'lumot saqlash, nashrni tayyorlashda qatnashganlar (mualliflar, tuzuvchilar, titul redaktorlari, tarjimonlar va rassomlar). Bunday birlashish yo'lga qo'tiladiki, turli xil yaratuvchilar haqida ma'lumot birgina domendan(familiya va ismi) tanlanadi va ma'lumotlarni bir hillikdan istisno qiladi(bir kishi turli xil nashrlarda turli xil rolni o'ynaydi). Misol uchun, S.Ya.Marshak tomonidan yozilgan sh'er(Ahmoq chichqoncha haqida ertak) va peysa(O'n ikki oy), Dj.Bayron, R.Bernsa, G.Geyne tarjimasi va she'riy to'plam tuzuvchisi. Yaratuvchi familiya va ismi(intsiali) juda katta bo'lishi mumkin(M.E.Saltkov-Shedrin, Fransua Rene de Shatobrin, Osten Jyul Jan-Batist Ippolit va hokazo) va turli nashrlarda ko'p marta takrorlanishi mumkin, shuning uxhun ularni raqamlash va shu raqamlarga bog'lash. Buning uchun butun raqamli atribut kiritiladi "Yaratuvchi_kodi", ma'lumotlar baza yangi muallif, tarjimon yoki boshqa yaratuvchi kiritilganda avtomat ravishda bir birlikka ko'payadi.

Shunday yaratiladi: Nashr_kodi, Nashr_turi, Xarakter_kodi, Til_kodi, Bilet_raqami, Pereplet_raqami, Joy_kodi va Nashr_kodi, o'zgartiriladigan birdan to o'ninchi atributgacha.

2. Nashiryot(Nashiryot_kodi, Nomi, Shahar).

3. Sarlavha(Sarlavha_kodi, Sarlavha). Bunday belgilash ma'lumotlar mohiyatini kamayishiga va bir biriga qarshi chiqishi ehtimolligini kamayishiga (uzun matnli ma'lumotlar kiritish zarurati tug'dirmaydi turli hil tomli to'plamlar, qayta nashrlar, o'quv qo'llanmalar va boshqalar uchun)

4. Nashr_turi(Nashr_turi, Tur_nomi)

5. Qayta nashr xarakteri(Xarakter_kodi, qayta nashr_xarakteri)

6. Tili(Til_kodi, Til, Qisqatmalar). Tildan tashqari hamma uchun qabul qilingan qisqartmalar(angl.,isp.,nem.,fr.), agar u yaratigan bo'lsa.

7. Joyi (Joy_kodi, Xona_raqami, Stellaj_raqami, Javon_raqami). Mohiyatni kodi hamma joy, kitoblar saqlanadigan devor orqasida

8. Kitobxonlar(Bilet_raqami, Familiya, Ismi, Otasini ismi,Adres,Telefon).

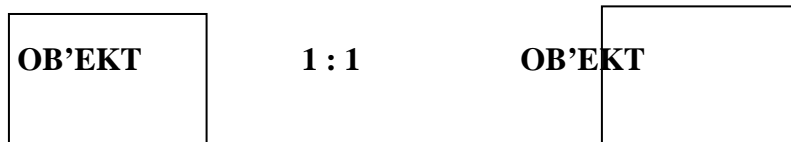
Informatsion ob'ektlar aloqasi

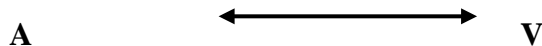
Informatsion o'bektlarni ajratishdan keyin loyihalashning keyingi qadami ular orasida aloqani aniqlash bo'lib hisoblanadi. Aloqa ikki informatsion ob'ektlar orasida o'rnatiladi. Bor aloqalar qoidaga binoan informatsion ob'ektlar bilan aniqlanadi.

Informatsion ob'ektlar aloqasi har xil turda bo'lishi mumkin:

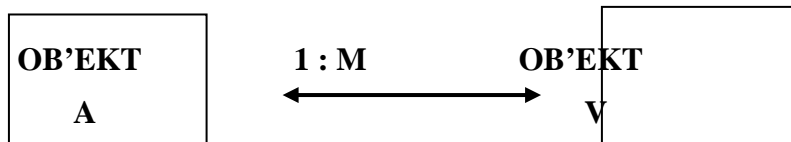
- Bir – birqiyamatli (1:1)
- Bir – ko'pqiyamatli (1:M)
- Ko'p – ko'pqiyamatli (N:M)

Bir – birqiyamatli aloqa bo'ladi, agar birinchi (A) ob'ektning har bir nusxasiga ikkinchi (V) ob'ektning faqat bitta nusxasi mos kelsa va teskarisiga, agar ikkinchi (B) ob'ektning har bir nusxasiga birinchi (A) ob'ektning faqat bitta nusxasi mos kelsa. Bunday ob'ektlarni bittaga birlashtirish juda oson. Ular ikkita mavjud ob'ektlar rkvizitlarini birlashtirib yagona strukturani tashkil qiladi.

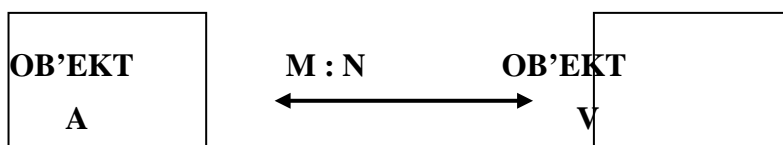




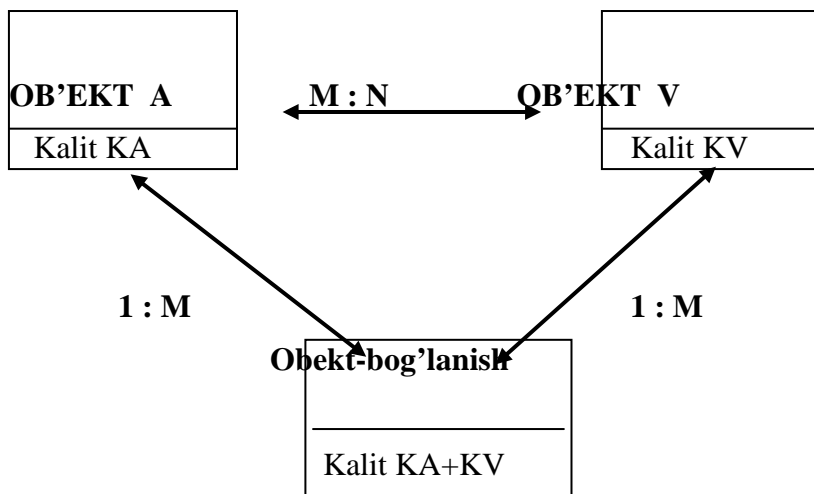
Bir – ko'pqiyimatli aloqa (1:M) - bu shunday aloqaki unda bitta (A) ob'ektning har bir nusxasiga ikkinchi (V) ob'ektning bir necha nusxasi mos keladi va teskarisiga, ikkinchi (B) ob'ektning har bir nusxasiga bitta (A) ob'ektning faqat bitta nusxasi mos kelsa. Bunday aloqada (A) ob'ektiasosiy ob'ekt, (B) ob'ekt esa qaram deyiladi.



Ko'p – ko'pqiyimatli aloqa (N:M) – bu shunday aloqaki unda bitta (A) ob'ektning har bir nusxasiga ikkinchi (V) ob'ektning bir necha nusxasi mos keladi va teskarisiga, ikkinchi (B) ob'ektning har bir nusxasiga birinchi (A) ob'ektning bir necha nusxasi mos kelsa.



Relyatsion ma'lumotlar bazasida ko'pga ko'p bog'lanishlarni bevosita bajarib bo'lmaydi. Shuning uchun, agar shunday bog'lanishlar aniqlangan bo'lsa, qo'shimcha bog'lanish ob'yektini kiritish yo'li bilan ularni o'zgartirish kerak bo'ladi. Ob'yektning boshlang'ich ma'lumotlari shu ob'yekt bilan birga-ko'p turidagi bog'lanish orqali bog'lanadi. Bog'lovchi ob'yekt boshlang'ich ob'yektar kalitlaridan hosil bo'lgan kalitga ega bo'lishi kerak.



Mohiyat – har xil ob'ekt (ob'ekt -bu biz ularni bir biridan farqlaymiz), uning ma'lumotlari ma'lumotlar bazasida saqlanadi. Mohiyat odam, joy, samalyot, reyslar, gullar va boshqalar bo'lishi mumkin. Mohiyat turi va mohiyat nusxasi tushunchalari mavjud, ularni bir biridan farqlash kerak. Mohiyat turi tushunchasi bir jinsli narsalar, hodisalar, shaxslar uchun qaraladi. Mohiyat nusxasi tushunchasi aniq narsalar to'plami sifatida qaraladi. Masalan, mohiyat turi SHAHAR bo'lishi mumkin, uning nusxasi esa – TOSHKENT, MOSKVA bo'lishi mumkin.

Mohiyatlar klassifikatsiyasi

Mohiyatlarning uchta asosiy turi aniqlangan:

- ✓ **Sterjenli**
- ✓ **Assotsiativ**
- ✓ **Tavsifiy**

shuningdek assotsiativ mohiyatning qisim sinfi bo'lgan – **belgilash**.

Sterjenli mohiyat (Sterjen) – bu mustaqil mohiyat (to'liq tarifi quyida ko'rsatilgan).

Assotsiativ mohiyat (Assotsiatsiya) – bu “ko'pdan-ko'pga” (“birdan-ko'pga” va hakoza) ko'rinishidagi bog'lanishlar ya'ni ikki yoki undan ortiq mohiyatlar yoki mohiyatlar hushasi orasidagi aloqa. Assotsiatsiyalar to'laqonli mohiyatlar sifatida ko'rib chiqiladi:

- ✓ Ular huddi sterjenli mohiyatlar singari boshqa assotsiatsiyalar va manolarda qatnasha oladi;
- ✓ Ular o'z hususiatlariga ega bo'lishi mumkin, ya'ni ular nafaqat aloqani ko'rsatishga kerak bo'ladigan kalit atributlariga ega bo'lib qo'lmasdan, balki aloqani tavsiflovchi istalgancha boshqa atributlar soniga ega bo'lishi mumkin. Masalan, “Nikoh” kuidagi kalit atributlarni o'z ichiga oladi “Kod_M”, “Kod_J” va “Erning tabel raqami”, “Xotinning tabel raqami”, shuningdek aniqlovchi atributlar “Guvohlik raqami”, “Qaydnoma muddati”, “Qaydnoma manzili”, “Nikoh uyi kitobidagi qaydnoma raqami” va hakoza.

Tavsifiy mohiyat (Tavsifnoma) bu ikki mohiyat orasidagi “ko'pdan- birga” yoki “birdan-birga” aloqa turi (Assotsiatsiyaning xususiy holati). Tavsifnomaning yagona maqsadi ba'zi boshqa mohiyatni tasvirlash yoki aniqlashdan iborat. Ularga zarurat tugilishining sababi xaqiqiy dunyodagi mohiyat ba'zan ko'p ma'noli xususiyatga ega. Er bir necha xotinga, kitob bir necha qayta tahrir tavsifiga (To'g'irlangan, to'ldirilgan, qayta ishlangan) ega bo'lishi mumkin va hakoza.

Tavsifnoma mohiyati butunlay tavsiflanayotgan mohiyatga bog'liqdir: agar eri olamdan o'tsa ayollar xotinlik mavqeini yoqotadi.

Tavsifnomani yozish uchun IMT umumiy hollarda quyidagi ko'rinishga ega bo'lgan yangi gapdan foydalanadi:

Tavsifnoma (atribut 1, atribut 2,...) {Tavsiflanayotgan mohiyatlar ro'yxati}.

ER-diagramm tilida, tavsifnoma trapetsiya shaklida ko'rsatiladi.



Belgilab olinadigan mohiyat yoki belgilash – bu ikki mohiyat orasidagi “ko'pdan-birga” yoki “birdan-birga” aloqa turidir va u tavsifnomadan belginlangan mohiyatga bog'lik emasligi bilan farq qiladi.

Xodimlarni turli tashkilotlar bo'limiga qabul qilish bilan bo'g'liq misolni ko'rib chiqamiz,

Qat'iy qoidalar bo'lmaganda (xodim bir vaqtda bir nechta bo'limda qayd qilinishi yoki hech qaysi bo'limda qayd qilinmasligi mumkin) tarifni assotsiatsiyada yaratish kerak.

Hisobga olish:

Bo'limlar (bo'lim raqami, bo'lim nomi,...)

Xizmatchilar (tabel raqami, familiyasi,.....)

Qaydnoma [Bo'limlarM, HizmatchilarN] (bo'lim raqami, bo'lim nomi, tabel raqami, qaydnoma muddati).

Shunga qaramasdan, agar har bir xodim biror bir bo'limga kiritilishi shart bo'lsa, tarifni xodimlar belgisi bilan yaratish mumkin:

Bo‘limlar (bo‘lim raqami, bo‘lim nomi,...)

Xizmatchilar (tabel raqami, familiyasi, bo‘lim raqami, qaydnoma muddati,)
[bo‘limlar]

Ushbu misolda xodimlar mustaqil mavjuddir (agar bo‘lim yoq qilinsa bu shu bo‘limdagi xodimlar ham yoqotilishi kerar degani emas.) Shuning uchun ular bo‘lim tavsifnomalari bo‘lishi va belgilash deb atalishi mumkin emas .

Belgilar takrorlanuvchi qiymatli katta matnli atributlarni saqlash uchun foydalaniladi: talabalar o‘rganayotgan fanlar ”kodifikatorlari”, tashkilotlar nomi va ularning bo‘limlari, mollar ro‘yxati va xakozalar.

Ko‘rinishidan belgilash tarifi tavsif tarifidan faqat qavslari bilan farq qiladi, belgilangan mohiyatlar katta qavsda emas , balki kvadrat qavslarda yoziladi:

BELGILASH(atribut 1, atribut 2,...)[BELGILANGAN MOHIYATLAR RO‘YXATI].

ER-diagrammalar tilida tavsifnoma paralleloiped bilan ko‘rsatiladi



Belgilashlar va **tavsifnomalar** butunlay mustaqil mohiatlar emas, chunki ular “belgilanadigan” va “tavsiflanadigan“ boshqa mohiyat borligini taxmin qiladi. Shunga qaramasdan ular baribir mohiyatlar xususiy holatini ko‘rsatadi va albatta xususiyatlarga ega bo‘lishi mumkin, assotsiatsiyalar va belgilarda qatnashishi mumkin hamda o‘zlarining (ancha past darajadagi) xususiy tavsifnomalariga ega bo‘lishi mumkin. Tavsifnomalarning barcha nusxalari tavsiflanayotgan mohiyatning biror bir nusxasi bilan albatta bog‘lanishi kerak.

Endi sterjenli mohiyatni na assotsiatsiya, na belgi na tavsifnoma bo‘lmagan mohiyat sifatida qayta ko‘rib chiqamiz. Bunday mohiyatlar boshqa mohiyatlarni bildirsa ham ular mustaqil ravishda mavjud .

3 – AMALIY MASHG‘ULOT

Mavzu: Ma’lumotlar bazasi jadvallarini MS Access da yaratish instrumental vositalari va ma’lumotlar bilan turli amallar bajarish.

Ishdan maqsad: Ma’lumotlar bazasi jadvallarini MS Access da yaratish instrumental vositalari va ma’lumotlar bilan turli amallar bajarish ko‘nikmasini hosil qilish.

Qisqacha nazariy ma’lumot

Ma’lumki, ma’lumotlar bazasi (MB) tushunchasi fanga kirib kelgunga qadar, ma’lumotlardan turli ko‘rinishda foydalanish juda qiyin edi. Hozirgi kunga kelib bir necha turdagi ma’lumotlar bazasini boshqarish tizimlari (MBBT) yaratilganki ular o‘z dasturlash tillariga ham ega bo‘lib, ular yordamida ma’lumotlar bazasini yaratish va shu bazadagi ma’lumotlar ustida ish yuritish imkoni mavjud. Ularga Clipper, Paradox, FoxPro, Oracle, Access va boshqalarni misol keltirish mumkin.

Bu MBBT ichida Access eng sodda va foydanuvchi uchun qulay bo‘lgan, Microsoft Office tarkibiga kiruvchi dasturdir.

Microsoft Access MBBT relyatsion modellar asosiga qurilgan bo'lib, unda tuziladigan MB jadvallar ko'rinishida aks etadi.

2. MS Accessning asosiy ob'ektlari

Access MBBT oynasi oddiy bo'lib, bu oynada yana dastur ishlatadigan yettita ob'ekt oynalarini ishlatish mumkin.

Tablitsa - ma'lumotlar bazasining asosiy ob'ekti. Unda ma'lumotlar jadval ko'rinishida saqlanadi.

Zapros - baza ma'lumotlarini qayta ishlash uchun maxsus struktura. Zapros orqali ma'lumotlarga so'rovlar tayyorlanadi va qayta ishlanadi.

Forma - bazaga yangi ma'lumotlarni kiritish yoki mavjud ma'lumotlarni ekranda ko'rsatish va ularni taxrirlashni ta'minlaydi.

Otchet - baza ma'lumotlarini kerakli formada chop etishni ta'minlaydi.

Stranitsy - ma'lumotlar bazasini Internetga joylashtirishda Web sahifalarini yaratish uchun ishlatiladi.

Makros - makrokomanda to'plami bo'lib baza operatsiyalarini birlashtirishda ishlatiladi.

Moduly - protsedura dasturlar to'plami. Bu standart protseduralar Vizual Basic tilida yozilgan bo'lib ularni dasturchi kengaytirish imkoniyatiga ega.

MBni loyihalash predmet sohasini tavsiflash asosida amalga oshiriladi. MB uchun zarur bo'lgan bu tavsiflash predmet sohasini ifodalovchi ob'ekt va jarayonlar haqidagi ma'lumotlarni o'z ichiga oladi.

Ma'lumotlar bazasini yaratish uchun avvalambor uning strukturasi ishlab chiqish zarur. Buning uchun MBda maydonning quyidagi elementlarini aniqlash kerak bo'ladi:

maydon nomi (imya polya);

maydon turi (tip dannyx);

maydon o'lchami (razmer polya);

maydonga tavsif (opisanie);

maydon xossalari;

maydon kaliti (klyuch);

Maydon turlari quyidagilar bilan aniqlanadi:

“Tekstovyy”- simvulli (matnli) qiymatlarni bildirib, uning uzunligi 255 tagacha bo'lishi mumkin;

“CHislovoy” sonli qiymatlarni bildirib, uning diapazoni -10307 +10308 ;

“Data/vremya” sana/vaqt qiymatini bildiradi;

“Logicheskiy” mantiqiy qiymatni bildirib, “True” va “False” qiymatlarni qabul qiladi;

“Polya Memo” memo (izohli) bo'lib qatorli qiymatlarni qabul qiladi, har bir qator 255 tagacha simvol

olishi mumkin. Memo ma'lumotlari maxsus fayllarda saqlanadi.

“OLE ob’ekt maydoni” tashqi ob’ektlarni joylashtirish uchun xizmat qiladi. Masalan: fotografiya,

videoyozuv va muzika fragmentlari.

Tablitsa ma’lumotlar bazasining asosiy ob’ekti bo’lib, boshqa ob’ektlar jadvalsiz ishlay olmaydi.

MB oynasidan Tablitsa ob’ekti ishga tushiriladi.

Instrumentlar panelidan Sozdats (New) buyrug’i beriladi. Ekranida yangi jadval nomli muloqot darchasi chiqadi.

Agar jadval strukturasi o’zingiz tuzmoqchi bo’lsangiz muloqot darchasidan Konstruktor rejimi tanlanib, Ok tugmasi bosiladi. Jadval strukturasi kiritish uchun darcha chiqadi.

Jadval strukturasi maydon nomi, turi, uning tavsifi va xossalari ketma-ket beriladi.

4. So’rovlar va ularni loyihalash

Access o’zida bir necha so’rov vositalarini mujassamlashtirganki, bu vositalar yordamida MB dan berilgan so’rovnoma asosida javoblar olish mumkin. MB dan kerakli ma’lumotlarni ajratib olishda Zaproso ob’ekti muhim o’rin tutadi.

Filtrlar yordamida ham ma’lumotlarni ajratib olish mumkin. Zaproso ob’ekti so’rovlarni saqlash va ularni qayta ishlatish imkonini beradi.

Access Zaproso ob’ekti ikki xil so’rovnoma tayyorlash usuliga ega:

- master yordamida;
- konstruktor yordamida.

5. Formalar yaratish

Ma’lumotlarni kiritishda formadan foydalanish ko’p qulayliklarni beradi. Accessda yaratilgan har bir forma ma’lumotlarni kiritish uchun maxsus maydonlar, oddiy tanlash tugmachalari va buyruq tugmachalarini o’z ichiga oladi.

MB oynasi obektlar panelidan Formy (Forms) bo’limi ishchi rejimga o’tkaziladi. Kerakli jadval yoki zapros tanlanadi.

6. Accessning standart funktsiyalari.

Access MBBTda bir qancha standart funktsiyalar mavjud. Ularni ishlatish uchun konstruktor rejimiga kirib, kerakli maydon belgilanadi. Sichqonchaning o’ng tugmachasi bir marta chiqillatib, keyin u yerdan “obrabotka sobyitiy” bo’limiga kiriladi. Birinchi oynadan “Funktsii”, keyin “Vstroennyye funktsii” bo’limi tanlanadi.

Ikkinchi oynada funktsiyalarning quyidagi ro’yhati chiqadi: vse, massivы, preobrazovanie, baza dannыx, finansovыe, matematicheskie, upravlenie, statisticheskie, tekstovыe va boshqa.

MS Access dasturida SQL tili asosida jadvallarni yaratish. Jadvallar CREATE TABLE buyrug’i bilan yaratiladi. Bu buyruq qatorlarsiz bo’sh jadval yaratadi. U jadval nomini, ma’lum tartibda ko’rsatilgan ustunlar nomlari ketma - ketligi, ma’lumotlar turlari va ustunlar o’lchovini aniqlaydi.

CREATE TABLE buyrug’ining umumiy yozilishi:

CREATE TABLE <jadval nomi>

(<ustun nomi> <ma’lumot turi>[(<ustun o’lchovi>)],

<ustun nomi> <ma'lumot turi>[(<ustun o'lchovi, ...)];

Jadval yaratishda va ular ustida ish yuritishda quyidagi 2 ta jadvaldan iborat ma'lumotlar bazasini

misol sifatida qaraymiz.

Sotuvchilar (Salepeople):

Snum Sname City Comm

11 Peel London 0.12

12 Serres San Jose 0.13

14 Motika London 0.11

SNum - xar bir sotuvchi unikal nomeri,

SName - sotuvchi nomi,

City - sotuvchi adresi (shaxri),

Comm - sotuvchilarning o'qli shakldagi komission foydasi.

Buyurtmachilar (Customers):

Cnum Cname City Rating SNum

21 Hoffman London 100 11

22 Giovanni Rome 200 13

23 Liu SanJose 200 12

CNum - xar bir buyurtmachi unikal nomeri,

CName - buyurtmachi nomi,

City - buyurtmachi adresi (shaxri),

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

SNum - shu buyurtmachiga tayinlangan sotuvchi nomeri.

Misol uchun sotuvchilar jadvalini yaratish:

```
CREATE TABLE Salepeople
```

```
( SNum integer,
```

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

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

```
Comm decimal )
```

Nazorat savollari

1.Qanday MBBTlarini bilasiz?

2.MB deganda nimani tushinasiz?

3.Accessning qanday ob'ektlarini bilasiz?

4.Tablitsa strukturasi qanday tashkil etiladi?

5.MBda maydon va yozuv tushunchasi nimani anglatadi?

6.So'rovlar qanday tashkil etiladi?

4 – AMALIY MASHG'ULOT

Mavzu: Ma'lumotlar bazasida modifik, bog'langan jadvallardan foydalanish.

Ishdan maqsad: Talabalarda ma'lumotlar bazasida modifik, bog'langan jadvallardan foydalanish ko'nikmasini xosil qilish

Infologik modellashtirishning maqsadi - tuziladigan ma'lumotlar bazasida shakllanishi mumkin bo'lgan ma'lumotlarni tasvirlash va yig'ish usullarini odamlar uchun ayniqsa tabiiy ta'minlashdir. SHuning uchun ma'lumotlarning infologik modelini tabiiy tilga mos qilib qurishga harakat qilinadi.

Infologik modelni qurishning asosiy konstruktiv elementlari bu:

- mohiyat;
- bog'lanish;
- xossalalar (atributlar)dir.

Mohiyat -bu ma'lumotlari ma'lumotlar bazasida saqlanishi kerak bo'lgan biror real yoki tasavvur qilingan ob'ekt bo'lib, aniq ma'noga va nomga egadir, u yagona bo'lishi kerak. Mohiyat turini uning nusxasi bilan farq qilish kerak. Mohiyat nomi uning nusxasiga emas, turiga beriladi. Mohiyat nusxasi -bu aniq bir xil turdagi narsalar, hodisalar va boshqalardir.

Masalan, "O'quvchi" mohiyatida "O'quvchi" mohiyat turining nomi, mohiyat nusxasi esa aniq bir o'quvchidir.

Masalan, Axmedov, Toshmatov va boshqa. "SHahar" mohiyatida "SHahar" mohiyat nomi, uning nusxasi esa aniq bir shaharlardir. Masalan, Toshkent, Samarqand, Buxoro va boshqa.

Bog'lanish -bu ikki yoki undan ortiq mohiyatlarning bir-biri bilan o'zaro bog'lanishidir. Bog'lanish faqat ikkita har xil mohiyatlar orasida mavjud bo'ladi.

Atribut(xossa) - mohiyatni xarakterlovchi nomlardir. U o'zida yagona murakab bo'lmagan strukturani tasvirlab, mohiyat holatini xarakterlaydi. Masalan, "O'quvchi" mohiyati atributi -kod, familiya, ism, yosh va boshqalar bo'lishi mumkin.

Yana bir misol: AVTOMOBIL mohiyati atributi bu - tur, marka, nomer belgisi, rangi va boshqalar. Ular ko'pqiymatlidir. Masalan, rang atributi turi ko'p qiymatli yoki nusxada: qizil, ko'k, oq, va boshqalar. Atribut o'zi mustaqil mohiyat bo'lib qatnashishi ham mumkin. Masalan, avtomobil zavodi uchun rang - bu faqat ishlab chiqarish mahsuloti atributi, lak kraska fabrikasi uchun esa rang - mohiyat turidir.

Entity-Relyatsion modeli

Ma'lumotlar bazasini loyihalashda har xil semantik modellar ham ishlatiladi. Ulardan eng ko'p tarqalganlaridan biri - ER modelidir. Bu model inglizcha "Entity-relation" deyilib, ma'nosi "Mohiyat-bog'lanish" demakdir.

Bu model 1976 yil Piter CHen tamonidan kiritilgan bo'lib u o'ziga bir qator grafik diagrammalarni oluvchi bir necha turdagi komponentalarni birlashtirgan.

ER modeli diagrammasida mohiyat odatda to'rtburchak shaklida tasvirlanib, uning ichiga mohiyat nomi yoziladi.

ER - modelining asosiy komponentalari mohiyat, bog'lanish va atribut bo'lib hisoblanadi.

Infomlogik modelni qurishda ER diagramma tili ishlatiladi.

Ikki mohiyat A va B o'rtasida to'rtta bog'lanish turi bo'lishi mumkin:

1. Birga-bir bog'lanish (1:1):

Har bir vaqt momentida A mohiyatning har bir (nusxasiga) vakiliga B mohiyatning vakili 1 yoki 0 bilan.

2. Birga-ko'p bog'lanish (1:M): A mohiyatning bitta vakiliga B mohiyatning bir necha vakili mos keladi.

A 1 AB M B

3. Ko'pga -bir bog'lanish (M:1)

A M AB 1 B

4. Ko'pga -ko'p bog'lanish (M:N)

A M AB N

5 – AMALIY MASHG'ULOT

Mavzu: Ma'lumotlar bazasi sxemasini normallashtirish.

Normal formalar 1NF, 2NF, 3NF.

Ishdan maqsad: Ma'lumotlar bazasi sxemalarini normallashtirish va ularning turlari bo'yicha bilim va ko'nikmalar xosil qilish.

Qisqacha nazariy ma'lumot.

Ma'lumotlar bazasini normallashtirish

Normallashtirish – bu ma'lumotlarni qo'shishda, o'zgartirishda va o'chirishda eng yaxshi xususiyatlarga ega ikki yoki undan ortiq bo'laklarga jadvalni bo'lish. Normallashtirishning asosiy maqsadi ma'lumotlar bazasini olishga qaratilganki, unda har bir dalil (fakt) faqat bir joyda uchraydi, ya'ni ma'lumotlar ortib ketmaydi. Bu faqat xotiradan tejam foydalanish maqsadida qilinmay, balki saqlanayotgan ma'lumotlar orasida qaramaqarshiliklarni bartaraf qilish uchundir.

Har bir jadval relyatsion ma'lumotlar bazasida shunday shartlarni qoniqtiradiki, unga ko'ra jadvalning har bir ustun va satrining kesishish joyida har doim yagona atomar qiymat joylashadi va hech qachon ko'p miqdorda xuddi shunday qiymatlar bo'lishi mumkin emas. Shu shartni qoniqtiruvchi har qanday jadval normallashtirilgan deyiladi. Umuman olganda normallashtirilmagan jadvallar, ya'ni takrorlanuvchi ma'lumotlar guruhiga ega jadvallar relatsion ma'lumotlar bazasiga kiritilmaydi.

Har qanday normallashtirilgan jadval avtomat ravishda **birinchi normal formada**, qisqacha **1NF**, deb hisoblanadi. Shunday qilib, umuman olganda, "normallashtirilgan" va "1NF da joylashgan" bitta ma'nosi anglatadi. Lekin amaliyotda "normallashtirilgan" atamasi ko'proq tor ma'noda islatiladi – "to'liq normallashtirilgan", ya'ni proyektda normallashtirishning hech qaysi prinsipi buzilmayapti degan ma'noda.

Keyingi pog'onadagi normallashtirishlarni ko'rib chiqamiz – **ikkinchi normal forma (2NF)**, **uchinchi normal forma (3NF)** va hokazo. Umuman olganda, jadval 1NF da bo'lsa va undan tashqari yana bir qo'shimcha shartni qoniqtirsa, uning ma'nosi keyinroq ko'rib o'tiladi. Jadval 3NF da deyiladi, qachonki u 2NF da bo'lsa va yana bir qo'shimcha shartni qoniqtirsa va hokazo.

Har bir normal forma qandaydir ma'noda oldingisiga qaraganda ancha chegaralangan, lekin ma'qulroqdir. Bu shunga bog'liqlik, "**N-chi normal forma**" ega

bo'lgan ba'zi yoqimsiz tomonlariga “(N+1)-chi normal forma” ega emas. N-chi normal forma ga qarab (N+1)-chi normal formaga qo'yilgan qo'shimcha shartning umumiy ma'nosi ana shunday yoqimsiz tomonlarni yo'qotishdan iborat.

Normallashtirish nazariyasi jadval maydonlari orasidagi u yoki bu bog'liqlikning borligiga asoslanadi. Bunday bog'liqliklarning ikki turi aniqlangan:

- funksional
- ko'p qiymatli

Funksional bog'liqlik: Berilgan ixtiyoriy vaqtda A maydonning har bir turli qiymatiga mos ravishda albatta B maydonning har bir turli qiymatidan faqat bitta qiymati bo'lsa, jadvalning B maydoni huddi shu jadvalning A maydoniga funksional bog'langan deyiladi. Aytib o'tish kerakki, A va B maydonlar tashkil qiluvchilar bo'lishi mumkin.

To'liq funksional bog'liqlik: B maydon A tashkil qiluvchiga to'liq funksional bog'liq bo'ladi, agar u A ga funksional bog'liq bo'lsa va A maydonning boshqa qiymatlariga bog'liq bo'lmasa.

Ko'p qiymatli bog'liqlik: A maydon huddi shu jadvalning B maydonini ko'p qiymatli aniqlaydi, agar A maydonning har bir qiymatiga mos ravishda B maydonning aniqlangan qiymatlari bor bo'lsa.

Ko'p funksional bog'liqlikga misol “O'qish jadvali”

Tartib	Muallim	Darslik
Dasturlash	Abdullayev I.R.	Liberti Dj . Osvoy samostoyatelno C++
Dasturlash	Raxmonov U.K.	Forsayt R. Pascal dlya vsex
Dasturlash	Kovalev N.N.	Liberti Dj. Osvoy samostoyatelno C++
Dasturlash	Kovalev N.N.	Forsayt R. Pascal dlya vsex
Dasturlash	Raxmonov U.K.	Faysman K. Professionalnoye programmirovaniye na Paskale

Jadvalda ko'p qiymatli bog'liqlik bor “Tartib-Muallim”: tartib (Dasturlash misolida) bir nechta muallim tomonidan o'qilishi mumkin (misolda Abdullayev, Raxmonov, Kovalev). Yana boshqa bog'liqlik ham bor “Tartib-Darslik”: informatikani o'tishda “Pascal dlya vsex”, ”Osvoy samostoyatelno C++” va “Professionalnoye programmirovaniye na Pascale” darsliklari ishlatiladi. Bunda Muallim va Darslik funksional bog'lanmagan, bu esa ortiqchalikka olib keladi (yangi darslikni yozish uchun yana ikkita yangi qator yozish kerak bo'ladi). Bu jadvalni ikkiga ajratganda ish yaxshilanadi: (Tartib-Muallim va Tartib-Darslik)

Normal formalar

Birinchi normal forma:

Jadval birinchi normal formada deyiladi , qachonki hech bir undagi qator istalgan maydonda bittadan ortiq bo'lmagan ma'noga ega bo'lmasa va birorta kalit maydoni bo'sh bo'lmasa ;

Ikkinchi normal forma:

Jadval ikkinchi normal formada deyiladi , qachonki agar u birinchi normal formadagi ifodalarni ,maydonlarni qanoatlantiradi va birlamchi kalit bo'lmaganda hamda birlamchi kalitga to'liq funksional qaram bolmaganda ;

Uchinchi normal forma:

Jadval uchinchi normal formada deyiladi, qachonki u 2NF ning barcha shartlarini qanoatlantirsa va birorta ham uning kalitsiz maydonlari boshqa bir kalitsiz maydonlar bilan bog'liq bo'lmasa;

Boysa-Kodd normal formasi

Jadval Boysa-Kodd normal (BKNF) formasida bo'ladi, agar maydonlar orasidagi har qanday funktsional bog'lanish to'liq funktsional bog'lanishga ega bo'lsa.

Beshinchi normal forma:

Jadval beshinchi normal formada (*5NF*) bo'ladi, qachonki har bir to'liq dekompozitsiya barcha proektsiyalari mumkin bo'lgan kalitlarni saqlasa. Birorta to'liq dekompozitsiyaga ega bo'lmagan jadval ham beshinchi normal formada (*5NF*) bo'ladi.

To'rtinchi normal forma agar to'liq dekompozitsiya ikkita proektsiya birlashmasidan iborat bo'lsa beshinchi normal formaning xususiy holi bo'ladi.

6 – AMALIY MASHG'ULOT

Mavzu: Bog'langan jadvallar uchun forma yaratish. SQL tili. SQL tili yordamida ma'lumotlar bazasi yaratish.

Ishdan maqsad: SQLda jadvallar yaratish bo'yicha ko'nikma hosil qilish. SQL tili va u yordamida ma'lumotlar bazasini yaratishni o'rganish.

Qisqacha nazariy ma'lumot.

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.

SQL da ma'lumotlar bazasini yaratish.

Create database <MB nomi> - MB nomi berilganda ma'lumotlar bazasini yaratish so'rovi.

Drop database <MB nomi> - MB nomi berilgan bazani o'chirish komondasi

Jadvallar bilan ishlash

Jadvallarni yaratish.

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):

Snum	Sname	City	Comm
11	Peel	London	0.12
12	Serres	San Jose	0.13
14	Motika	London	0.11

SNum – har bir sotuvchi unikal nomeri,

SName – sotuvchi nomi,

City – sotuvchi adresi (shahar),

Comm – sotuvchilarning o'nli shakldagi komission foydasi.

Jadval Buyurtmachilar (Customers):

Cnum	Cname	City	Rating	SNum
21	Hoffman	London	100	11
22	Giovanni	Rome	200	13
23	Liu	SanJose	200	12

CNum – har bir buyurtmachi unikal nomeri,

CName – buyurtmachi nomi,

City – buyurtmachi adresi (shahar),

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

SNum – shu buyurtmachiga tayinlangan sotuvchi nomeri.

Jadval Buyurtma (Orders):

Onum	AMT	Odate	CNum	Snum
38	4723.00	1990/10/05	26	11
310	1309.95	1990/10/06	24	12

ONum – har bir sotib olish unikal nomeri,

AMT – sotib olish summasi qiymati,

ODate – sotib olish sanasi,

CNum – sotib oluvchi buyurtmachi nomeri,

SNum – sotuvchining nomeri.

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

Cheklanishlarni kiritish.

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 etiladi. 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

7 – AMALIY MASHG'ULOT

Mavzu: Ma'lumotlar bazasi xisobotini yaratish. SQL tili yordamida so'rovlar tizimini yaratish.

Ishdan maqsad: SQL tili yordamida ma'lumotlar bazasida turli so'rovlar yaratishni o'rganish.

Qisqacha nazariy ma'lumot.

Maydonlar qiymatlarini kiritish, o'chirish va o'zgartirish

Qiymatlarni kiritish.

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] ...)]**
ostki so'rov;

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

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

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

```
INSERT INTO Salepeople (Sname, Comm, SNum) VALUES ('Peel', .12, 11);
```

E'tibor bering City ustuni tashlab yuborilgan, chunki unga ko'zda tutilgan qiymat kiritiladi.

Siz INSERT komandasidan bir jadvaldan qiymat tanlab, so'rov bilan ishlatish uchun, ikkinchisiga joylashishda foydalanishingiz mumkin. Buning uchun siz VALUES ifordasini (oldingi misoldagi) mos so'rovga almashiringiz kerak:

```
INSERT INTO Londonstaff
```

```
SELECT * FROM Salespeople WHERE City = 'London';
```

Satrlarni o'chirish.

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

DELETE FROM <table name | view name>

[WHERE search-condition];

Masalan, Sotuvchilar jadvalidagi hamma satrlarni o'chirish uchun, quyidagi shartni kiritish mumkin: **DELETE FROM Salepeople;**

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

DELETE FROM Salepeople WHERE SNum = 13;

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.

Ikkinchi variant:

UPDATE <table name> SET column = expression, ... [FROM table-list]

[WHERE search-condition]

Masalan, hamma buyurtmachilar baxosini 200 ga o'zgartirish uchun quyidagini kiritishingiz mumkin: **UPDATE Customers SET Rating = 200;**

Ma'lum satrlarni o'zgartirish uchun DELETE dagi kabi predikatdan foydalanish kerak. Masalan Peel (imeyущego SNum=11) sotuvchining hamma buyurtmachilari uchun bir xil o'zgartirishni quyidagicha kiritish mumkin выполnit:

UPDATE Customers SET Rating = 200 WHERE SNum = 11;

SET vergul bilan ajratilgan ixtiyoriy sondagi ustunlarga qiymat tayinlashi mumkin. Ixtiyoriy jadval satrlari uchun qiymat tayinlanishi mumkin, lekin bir vaqtning o'zida faqat bitta satrga qiymat tayinlanadi. Masalan:

UPDATE Salepeople SET SName = 'Gibson', City = 'Boston', Comm = .10

WHERE SNum = 14;

Siz UPDATE komandasining SET jumlasida skalyar ifodalardan o'zgartirilayotgan maydon ifodasiga qo'shgan xolda foydalanishingiz mumkin. Masalan:

UPDATE Salepeople SET Comm = Comm * 2;

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 -----

Birinchi qoida, SELECT ifodasi o‘z ichiga albatta FROM ifodasini olishi kerak. Qolgan ifodalar kerak bo‘lsa ishlatiladi.

SELECT ifodasidan so‘ng so‘rovda qaytariluvchi ustunlar ro‘yxati yoziladi.

FROM ifodasidan so‘ng so‘rovni bajarish uchun jadvallar nomi yoziladi.

WHERE ifodasidan so‘ng agar ma‘lum satrlarni qaytarish lozim bo‘lsa, izlash sharti yoziladi.

GROUP BY ifodasi guruxlarga ajratilgan natijaviy so‘rov yaratishga imkon beradi.

HAVING ifodasidan guruxlarni qaytarish sharti yoziladi va GROUP BY bilan birga ishlatiladi.

ORDER BY ifodasi ma‘lumotlar natijaviy to‘plamini tartiblash yo‘nalishini aniqlaydi.

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

SELECT *(vse!) FROM OFFICES

SELECT yordamida ma‘lumotlarni tanlash

SELECT Operatori albatta "qaytariluvchi ustunlar ro‘yxati " ni o‘zichiga olishi kerak, Ya‘ni:

SELECT FILED1, FIELD2, FIELD3 ... FROM ...

FILED1, FIELD2, FIELD3 qaytariluvchi ustunlar ro‘yxati bo‘lib, ma‘lumotlar ketma ketligi shu tartibda qaytariladi!

Ya‘ni "qaytariluvchi ustunlar ro‘yxati" hisoblanuvchi ustunlar va konstantalarni o‘zichiga olishi mumkin. **SELECT FILED1, (FIELD2 - FIELD3) "CONST" ... FROM ...**

FROM jumlasini "jadval spetsifikatorlari ", Ya‘ni so‘rovni tashkil qiluvchi jadvallar nomini o‘z ichiga oladi. Bu jadvallar so‘rov asoini 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 kamini 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 konstantalarnisolishtirish 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 abonentlari, axir ikkita bir xil nomerlar mavjud emas!

Mantiqiy operatorlar

BETWEEN va IN Operatorlari

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

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

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.

```
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 -----

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'))
```

Sanalar bilan shu tarzda ishlanadi.

To'rtta konkret xizmatchilar tomonidan olingan hamma buyurtmalarni aniqlash.

```
SELECT ORDER_NUM, REP, AMOUNT FROM ORDERS  
WHERE REP IN (107, 109, 101, 103)
```

NOT IN yordamida diapazonga "tegishli emaslikni" tekshirish mumkin.

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 '%' shablони birinchi kelsa:

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

Ba'zida '%' shablони o'rniga '*' belgisi qo'llanadi znak, masalan v MS SQL uchun, c:\>dir *.exe!

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

```
SELECT COMPANY, CREDIT_LIMIT FROM CUSTOMERS  
WHERE COMPANY LIKE 'Ap_lsin'
```

8– AMALIY MASHG'ULOT

Mavzu: MS Access muxitida SQL tilida ma'lumotlar bazasini yaratish.

Ishdan maqsad: MS Access muxitida SQL tilida ma'lumotlar bazasini yaratish.

МО жадваллари билан ишлаш жараёни

1. **МОБТ** ойнасининг пастки қисмида **поля номера записи** (номер ёзиш майдони) бўлиб, бунда майдонга ўтиш тугмалари бор (*жадвал бўйича силжишни амалга оширади*).

2. Ҳар бир ёзув чап томонида **ёзув маркери** (маркер записи) тугмачасига эга. Шу тугмани боссак, ёзув ажратилиб кўринади ва нусха олишга тайёрланади.

3. Ажратилган ёзувда сичқонча ўнг тугмасини боссак, **контекст меню** мулоқот ойнаси чиқади ва унинг буйруқлари орқали ёзув устида иш бажарилади.

4. Жадвалнинг чап томони юқори қисмида турган маркер **жадвал маркери** дейилади. Уни боссак, бутун жадвал ажратилиб кўринади. Сичқонча ўнг тугмаси босилса, **контекст меню** мулоқот ойнаси экранда пайдо бўлади. Унинг буйруқлари билан жадвал устида иш юритилади.

5. Майдон сарлавҳасида сичқонча тугмасини боссак, у ҳолда майдон ажратилиб кўринади.

Запрос (Сўров)лар ташкил қилиш

МО га кириш учун «Сўров» дан фойдаланилади. Бу жараён **МБ** ойнасининг **Запрос** (Сўров) бўлимида яратиш тугмасини босиш билан бошланади ва экранда қуйидаги мулоқот ойнаси пайдо бўлади. **МО** га кириш учун **Запрос** тузишнинг бир қатор усулларини таклиф қилинади:

Конструктор - мустақил равишда янги сўровлар тузиш.

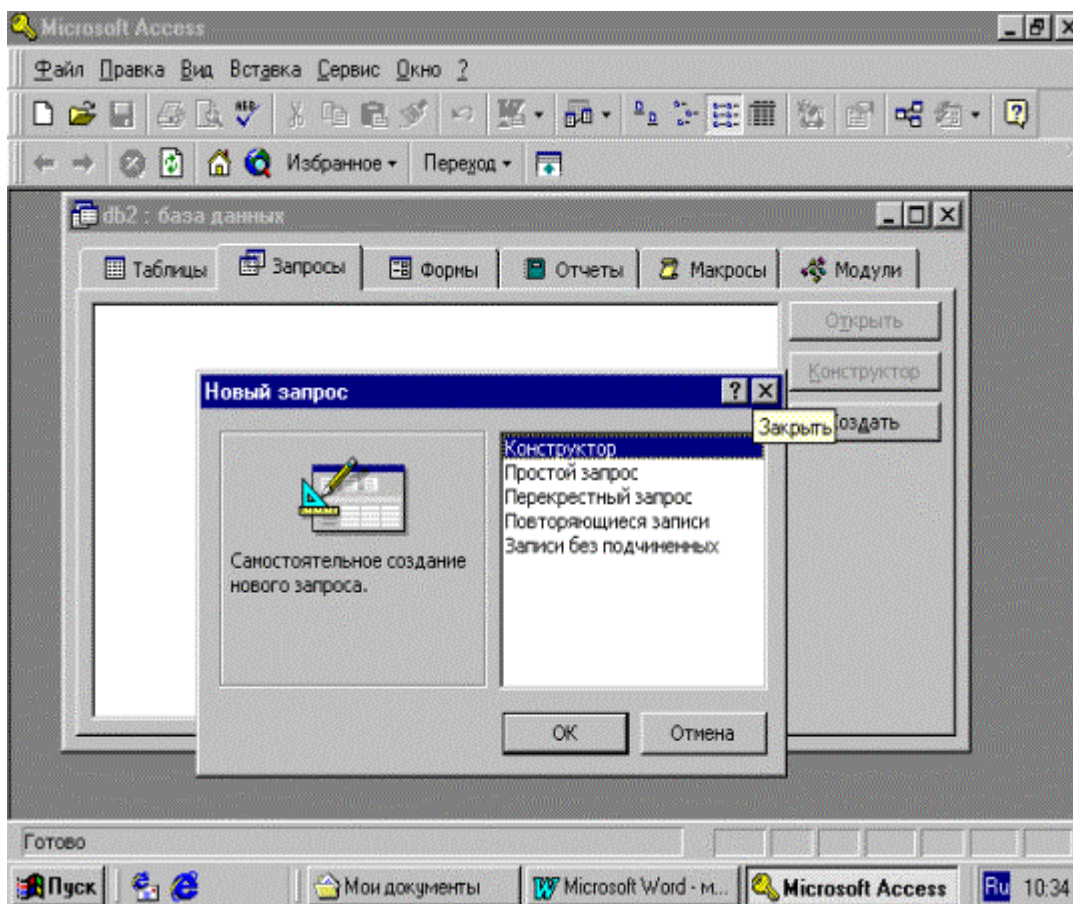
Простой запрос (оддий сўров) - мавжуд аниқ майдонларни танлаб олиш йўли билан сўровлар тузиш.

Перекрестный запрос (киёсий сўров) - **МО** да мавжуд бўлган бир нечта жадвал ва сўровларни чатишмасидан янги сўровлар яратиш.

Повторяющиеся записи (такрорланувчи ёзувлар) жадвалда ёки сўровларда такрорланувчи ёзувларни қидириб топиш учун сўровлар тузиш.

Записи без подчиненных (боғланмаган ёзувлар) жорий жадвалга мос келмайдиган ёзувларни қидириб топиш учун сўровлар тузиш.

Хуллас, **Запрос** ёрдамида асосий **МО**дан натижавий (фойдаланувчини қизиқтирган) жадвал ташкил қилиш ва уни қайта ишлаш имконияти пайдо булади. **Запрос** билан ишлаганда маълумотларни **саралаш (фильтрдан ўтказиш), жамлаш, ажратиш, ўзгартириш** мумкин. Аммо бу амал ҳар бажарилганда асосий **МБ** да ҳечқандай ўзгариш содир бўлмайди. Бундан ташқари, **Запрос** ёрдамида «**натижаларни ҳисоблаш**», ўрта арифметик қийматини топиш, йиғинди ҳосил қилиш ёки бирор майдон устида математик амаллар бажариш мумкин.



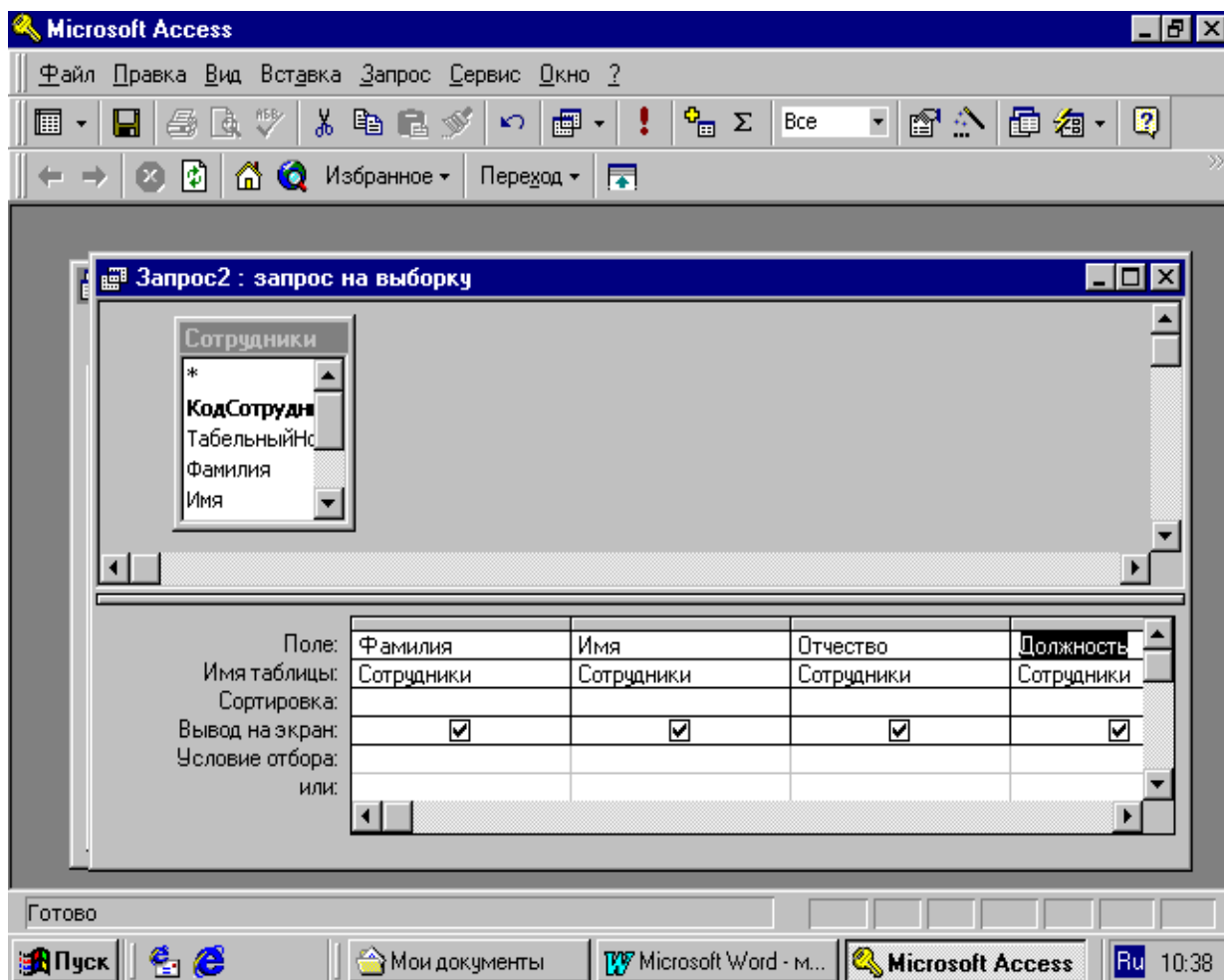
МОда ажратиш учун «Сўров»

Запрос ҳосил қилишнинг турлари кўп. Аммо энг кўп қўлланиладигани **Запрос на «вўборку»** (*Ажратишни ташкил қилувчи сўров*) Accessда “Сўров” ташкил қилишнинг 3 та усули мавжуд: **автоматик равишда, қўлда** ва **мастер** (уста) **ёрдамида**.

Запрос ташкил қилиш учун махсус **SQL(Structured Query Language)** тили мавжуд, аммо бу тилда ишлаш анча мураккаб, шунинг учун ҳам **Access** да махсус «**Намунавий сўров бланки**» ташкил қилинган. Бунда **Запрос** элементларини ойналараро ташиш орқали амалга ошириш мумкин. **МБ** га **Запрос** билан кириш «**Создать**» тугмасини босиш билан амалга оширилади. Унинг мулоқат ойнаси «**Новый запрос**» деб аталади. Унда «**Конструктор**» режимида иш юритилади. Шунда **МБ** тузилмасидан керакли жадвал ва унинг майдонлари **Запрос** бўйича танланади. Жадвал танлаш «**Добавление таблиц**» (Жадвал қўшиш) мулоқот ойнасида содир бўлади. Бунда

МБдаги барча жадваллар рўйхати бор. Ажратилган жадваллар бланкнинг юқори қисмига *Добавить* (Тўлдириш) тугмасини босиш билан амалга оширилади.

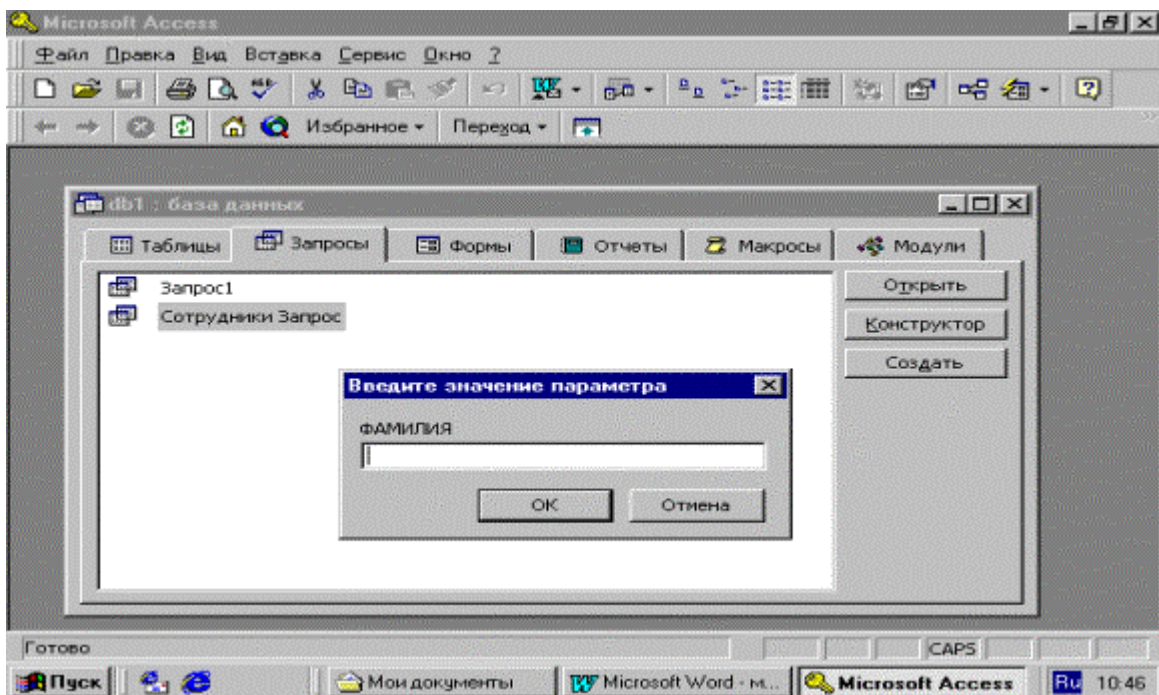
Намунавий Запрос бланкасини тўлдириш



Намунавий бланк 2та панелдан иборат. Юқори қисмида **Запрос** га асосланадиган жадваллар рўйхати тузилган. +уйи қисмида эса **Запрос** тузилмаси буйича тузиладиган натижавий жадвал ўз аксини топган. Бланкнинг майдон ёзиладиган сатҳида жадвалдан керакли майдон номлари ажратиб ўтказилади. Жадвал номи керакли сатрга майдонларни кўчириш жараёнида автоматик тарзда ёзилади.

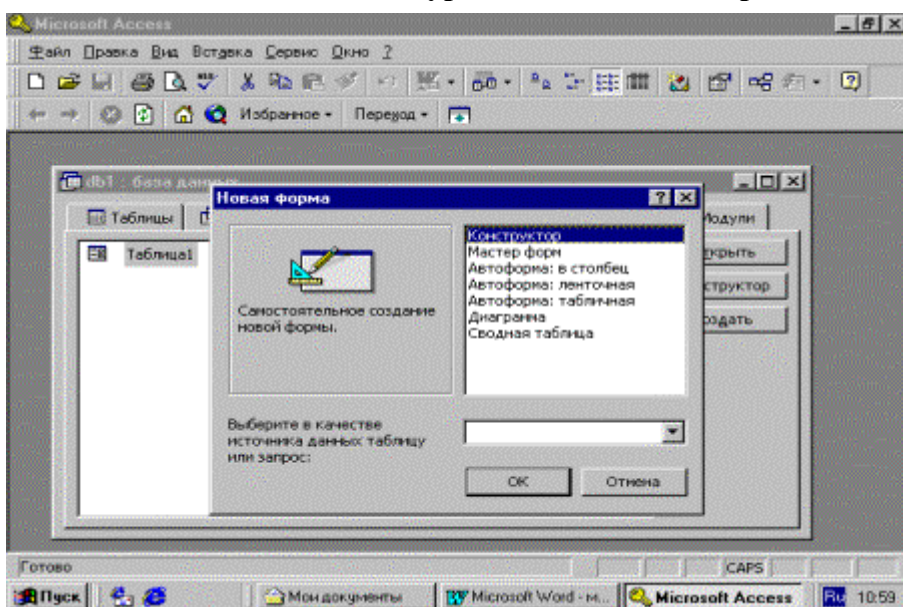
«Саралаш» деган сатрда «сичқонча» тугмаси босилса, бирор `майдондаги маълумотлар сараланади. **Запрос** бланкида *Условия отбора* (танлаш шарти) сатри мавжуд бўлиб, унда натижавий жадвални қониктирадиган *шарт критерийси* жойлашган бўлади. **Запрос Вид** тугмасини босиш билан натижавий жадвал ҳосил бўлади. Натижавий жадвалдан чиқиш учун «Вид» тугмасига яна бир бор босиш лозим.

Параметрлар бўйича «Сўров» тузиш



Баъзан фойдаланувчи маълумотлар базасидан муайян параметрлар бўйича маълумотларга мухтож бўлиб қолади. Ана шундай вазиятларда **Запрос** ни параметрлар бўйича ташкил қилиш лозим бўлиб қолади. Шундай мақсад қўйилганда **SQL** тилининг махсус буйруғи **LIKE[...]** орқали **Запрос** ни ташкил қилиш мумкин. Квадрат қавс ичида фойдаланувчи учун ихтиёрый матн кири-тиш мўлжалланган. Масалан, **LIKE[мамлакат номини киритинг]**. Ушбу буйруқни *условие отбора* (танлаш шарти) ёзилган сатрга жойлаштириш лозим. **Запрос** ишга туширилгач, мулоқот ойнаси очилиб фойдаланувчи учун параметр киритиш имкони пайдо бўлади.

Сўров да ҳисоблаш жараёни



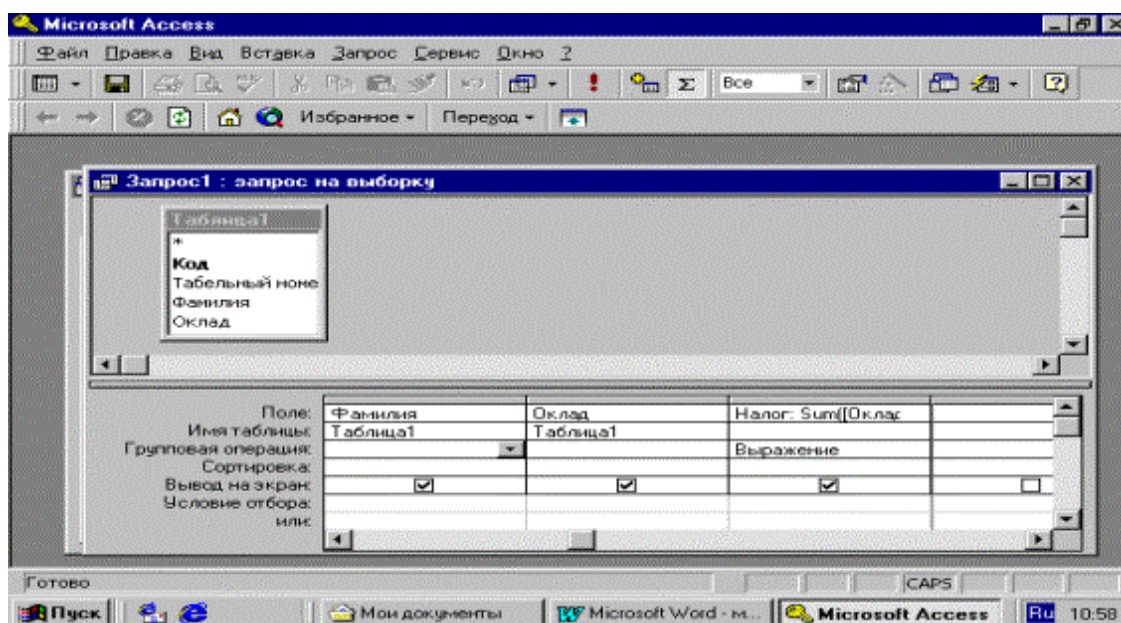
Натижавий жадвалда бошқа майдонлар бўйича ҳисоблашни ташкил этиш натижалари ёзиладиган майдон **ҳисоб майдони** дейилади. Бунда майдон номи ўрнига ҳисоблаш формуласи ва квадрат қавс ёзилади. Ушбу жараёни клавиатуринг **Shift+F2** тугмасини босиш билан ҳам бажариш мумкин.

Бунда ёрдамчи **область ввода** (киритиш худуди) мулоқат ойнаси очилиб, унда узун формулаларни ҳам киритиш имконияти очилади. Баъзан **ҳисоблаш майдонини саралаш майдони ҳам қилиш мумкин**. Ҳисоблашни ташкил қиладиган **Запрос** ҳам намунавий сўров бланкида ўз аксини топади. Бунда майдон номи ўрнига формула ёзилади. Формулага квадрат қавс ичида ҳисобланадиган майдон номи ҳам киритилади. Аммо торгина майдонга узун формулаларни киритиб бўлмайди. У ҳолда **Shift+F2** тугмачани боссак, у ҳолда ёрдамчи мулоқот ойнаси пайдо бўлади ва исталган узунликдаги формулаларни киритиш имконияти пайдо бўлади.

Натижавий «Сўров» тузиш технологияси

«Сўров» лар нафақат керакли маълумотни олиш ва уни ишлаш учун, балки натижавий ҳисоблашлар ташкил қилиш имконини ҳам беради. **Масалан**, қандайдир **ёзув** (қатор) лар гуруҳи бўйича ўрта арифметик қийматини ёки йиғиндисини топиш. Бу ҳолда ҳам **намунавий сўров бланки** ёрдамида иш бажарилади, аммо **ёзувларни** бирор белгисига қараб алоҳида гуруҳларга жамлаш талаб қилинади ва бунда **гуруҳлаш** деган ёрдамчи қатор пайдо бўлади. Ушбу қаторни намунавий бланкга киритиш учун асбоблар панелидаги **Σ** га курсорни келтириб **«сичқонча»** чап тугмасини босамиз:

Ўзгартиришлар «сўрови»ни тузиш



Автоматик равишда янги жадвал тузишда ёки ҳисоблаш натижалари асосида жадвал ҳосил қилишда вақтинчалик натижавий жадвал тузилади ва бу жадвалдан янгисини ҳосил қилишда ёки ўзгартиришда фойдаланилади. Бу ҳолатда «Сўров» ни ўзгартиришнинг бирнеча усуллари мавжуд:

- жадвал тузиш сўрови,
- жадвал таркибидаги маълумотларни янгилаш сўрови,
- ёзувларни киритиш сўрови,
- ёзувларни йўқотиш сўрови.

Бунинг учун **Запрос** менюсидаги **Создать** буйруғи билан **Конструктор** тартибида иш юритилади.

9– AMALIY MASHG'ULOT

Mavzu: SQL tilidagi boshqaruvchi so'rovlar yordamida ma'lumotlar bazasini modifikatsiyalash va jadvallarni yaratish.

Ishdan maqsad: Talabalarda SQL tilidagi boshqaruvchi so'rovlar yordamida ma'lumotlar bazasini modifikatsiyalash va jadvallarni yaratish ko'nikmasini xosil qilish.

Qisqacha nazariy ma'lumot.

Maydonlar qiymatlarini kiritish, o'chirish va o'zgartirish

Qiymatlarni kiritish.

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] ...)]**
ostki so'rov;

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

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

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

```
INSERT INTO Salepeople (Sname, Comm, SNum) VALUES ('Peel', .12, 11);
```

E'tibor bering City ustuni tashlab yuborilgan, chunki unga ko'zda tutilgan qiymat kiritiladi.

Siz INSERT komandasidan bir jadvaldan qiymat tanlab, so'rov bilan ishlatish uchun, ikkinchisiga joylashishda foydalanishingiz mumkin. Buning uchun siz VALUES ifordasini (oldingi misoldagi) mos so'rovga almashiringiz kerak:

```
INSERT INTO Londonstaff  
SELECT * FROM Salespeople WHERE City = 'London';
```

Satrlarni o'chirish.

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

```
DELETE FROM <table name | view name>  
[WHERE search-condition];
```

Masalan, Sotuvchilar jadvalidagi hamma satrlarni o'chirish uchun, quyidagi shartni kiritish mumkin: **DELETE FROM Salepeople;**

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

```
DELETE FROM Salepeople WHERE SNum = 13;
```

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.

Ikkinchi variant:

```
UPDATE <table name> SET column = expression, ... [ FROM table-list ]  
[ WHERE search-condition ]
```

Masalan, hamma buyurtmachilar baxosini 200 ga o'zgartirish uchun quyidagini kiritishingiz mumkin: **UPDATE Customers SET Rating = 200;**

Ma'lum satrlarni o'zgartirish uchun DELETE dagi kabi predikatdan foydalanish kerak. Masalan Peel (imeyущego SNum=11) sotuvchining hamma buyurtmachilari uchun bir xil o'zgartirishni quyidagicha kiritish mumkin выполnit:

```
UPDATE Customers SET Rating = 200 WHERE SNum = 11;
```

SET vergul bilan ajratilgan ixtiyoriy sondagi ustunlarga qiymat tayinlashi mumkin. Ixtiyoriy jadval satrlari uchun qiymat tayinlanishi mumkin, lekin bir vaqtning o'zida faqat bitta satrga qiymat tayinlanadi. Masalan:

```
UPDATE Salepeople SET SName = 'Gibson', City = 'Boston', Comm = .10  
WHERE SNum = 14;
```

Siz UPDATE komandasining SET jumlasida skalyar ifodalardan o'zgartirilayotgan maydon ifodasiga qo'shgan xolda foydalanishingiz mumkin. Masalan:

```
UPDATE Salepeople SET Comm = Comm * 2;
```

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 -----
```

Birinchi qoida, SELECT ifodasi o'z ichiga albatta FROM ifodasini olishi kerak. Qolgan ifodalar kerak bo'lsa ishlatiladi.

SELECT ifodasidan so'ng so'rovda qaytariluvchi ustunlar ro'yxati yoziladi.

FROM ifodasidan so'ng so'rovni bajarish uchun jadvallar nomi yoziladi.

WHERE ifodasidan so'ng agar ma'lum satrlarni qaytarish lozim bo'lsa, izlash sharti yoziladi.

GROUP BY ifodasi guruxlarga ajratilgan natijaviy so'rov yaratishga imkon beradi.

HAVING ifodasidan guruxlarni qaytarish sharti yoziladi va GROUP BY bilan birga ishlatiladi.

ORDER BY ifodasi ma'lumotlar natijaviy to'plamini tartiblash yo'nalishini aniqlaydi.

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

SELECT *(vse!) FROM OFFICES

SELECT yordamida ma'lumotlarni tanlash

SELECT Operatori albatta "qaytariluvchi ustunlar ro'yxati " ni o'zichiga olishi kerak, Ya'ni:

SELECT FILED1, FIELD2, FIELD3 ... FROM ...

FILED1, FIELD2, FIELD3 qaytariluvchi ustunlar ro'yxati bo'lib, ma'lumotlar ketma ketligi shu tartibda qaytariladi!

Ya'ni "qaytariluvchi ustunlar ro'yxati" hisoblanuvchi ustunlar va konstantalarni o'zichiga olishi mumkin. **SELECT FILED1, (FIELD2 - FIELD3) "CONST" ... FROM ...**

FROM jumlasini "jadval spetsifikatorlari ", Ya'ni so'rovni tashkil qiluvchi jadvallar nomini o'z ichiga oladi. Bu jadvallar so'rov asoini 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:

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

7.Qiyamatlar diapazoniga tegishlilikni tekshirish. Masalan berilgan qiymat diapazonga kiradimi yo'qmi.

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

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

10. NULL qiymatga tenglikka tekshirish.

Solishtirish amallari maydon va konstantalarnisolishtirish 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'rovlardan foydalaniladi, masalan shahar telefon tarmog'i abonenti, axir ikkita bir xil nomerlar mavjud emas!

Mantiqiy operatorlar

BETWEEN va IN Operatorlari

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

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

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.

```
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 -----
```

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'))
```

Sanalar bilan shu tarzda ishlanadi.

To'rtta konkret xizmatchilar tomonidan olingan hamma buyurtmalarni aniqlash.

```
SELECT ORDER_NUM, REP, AMOUNT FROM ORDERS  
WHERE REP IN (107, 109, 101, 103)
```

NOT IN yordamida diapazonga "tegishli emaslikni " tekshirish mumkin.

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:

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

Ba'zida '%' shablona o'rniga '*' belgisi qo'llanadi znak, masalan v MS SQL uchun, c:\>dir *.exe!

