

004
M34

R.V.QOBULOV, B.B.AKBARALIYEV,
J.T.USMONOV, O.B.RO'ZIBAYEV

MA'LUMOTLAR BAZASI



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

**MUHAMMAD AL-XORAZMIY NOMIDAGI TOSHKENT
AXBOROT TEXNOLOGIYALARI UNIVERSITETI**

**R.V.QOBULOV, B.B.AKBARALIYEV,
J.T.USMONOV, O.B.RO'ZIBAYEV**

MA'LUMOTLAR BAZASI

**O'zbekiston Respublikasi Oliy va O'rta maxsus ta'lim vazirligi tomonidan Oliy
o'quv yurtlari talabalari uchun o'quv qo'llanma sifatida tavsiya etilsin.**

TOSHKENT – 2021

UDK: 004.6(075)

BBK: 32.973-018.2

M 34

R.V.Qobulov, B.B.Akbaraliyev, J.T.Usmonov, O.B.Ro'zibayev.
Ma'lumotlar bazasi. (O'quv qo'llanma) – T.: «Aloqachi», 2021. –
140 b.

ISBN 978-9943-6398-1-2

O'quv qo'llanmada ma'lumotlar modellari, relyatsion model, ma'lumotlar bazasini boshqarish tizimlari, relyatsion algebra va hisob, SQL tilida ma'lumotlar bazasini yaratish, so'rovlar yaratish, cheklanishlar va tasvirlar, ma'lumotlar bazasini loyihalashtirish, CASE tizimlar, tranzaksiyalarni qayta ishlash, ma'lumotlar bazasini himoyalash, XML va multimodel ma'lumotlar bazalari yoritilgan.

O'quv qo'llanma universitetning barcha ta'lim yo'nalishlari talabalari va professor- o'qituvchilar, ilmiy tadqiqotchilar uchun mo'ljallangan.

UDK: 004.6(075)

BBK: 32.973-018.2

Taqrizchilar:

Kerimov K.F.

Madraximov Sh.F.

ISBN 978-9943-6398-1-2

© «Aloqachi» nashriyoti, 2021.

Kirish

Hozirgi davrda turli sohalarda avtomatlashtirilgan axborot tizimlarini yaratish va qo'llash katta ahamiyatga ega. Bunday tizimlarga halq xo'jaligini boshqarish, ilmiy izlanishlarni avtomatlashtirish loyihalashtirish va ishlab chiqarishni avtomatlashtirish tizimlari kiradi. Zamonaviy axborot texnologiyalari, hisoblash tizimlari rivojlanishi keng ko'lamli korporativ tizimlarni yaratish uchun keng imkoniyat yaratmoqda. Zamonaviy avtomatlashtirilgan axborot tizimlari asosini maxsus ma'lumotlar bazasini boshqarish tizimlarida yaratilgan ma'lumotlar bazalari tashkil etadi. Ma'lumotlar bazalari tuzilmasi asosida ma'lumotlar modellari yotadi. Oxirgi davrda obyektga yo'naltirilgan, grafli ma'lumotlar bazalari keng rivojlanib bormoqda. Lekin relyatsion ma'lumotlar bazalari yetakchi rolini yo'qotgani yo'q. O'quv qo'llanmada ma'lumotlar bazalari hozirgi holatini aks ettirishga harakat qilindi. SQL tili asoslari va turli modellar uchun kengaytmalariga ham e'tibor berildi.

- Birinchi bobda ma'lumotlar bazasi asosiy tushunchalari berilgan.
- Ikkinchi bobda ma'lumotlar bazasi modellari va normal formalar berilgan.
- Uchinchi bobda SQL tili orqali ma'lumotlar ustida amallar bajarish yoritilgan.
- To'rtinchi bobda Ma'lumotlar bazasiga murojaatni amalga oshirish yoritilgan .

O'quv qo'llanma yo'nalish uchun fan o'quv dasturiga asoslangan. O'quv qo'llanmada bilimlarni mustaxkamlash uchun nazariy savollar hamda amaliy misollar berildi. Misollarda amaliyotda hamda o'quv adabiyotda keng uchraydigan ma'lumotlar bazalaridan foydalanildi. O'quv qo'llanmadan mustaqil ta'lim uchun ham foydalanish mumkin.

1-BOB. Ma`lumotlar bazasining asosiy tushunchalari

1.1. Ma`lumotlar bazasining maqsadi, vazifalari va asosiy tushunchalari

Zamonaviy axborot tizimlari asosini ma`lumotlar bazasi tashkil etadi. Ma`lumotlar bazasi deb kompyuter xotirasida saqlanuvchi ma`lum sohaga tegishli faktografik ma`lumotlar majmuasiga aytiladi. Faktografik ma`lumotlar ma`lum turdagi foydalanuvchilar uchun kerak bo`lgan real predmetlar, hodisalar va jarayonlar xossalarini aks ettiruvchi ma`lumotlardir.

Predmet sohasi bu boshqarishni avtomatlashtirish lozim bo`lgan sohadir. Ma`lumotlar bazasida konkret predmet sohasiga tegishli dinamik ravishda yangilanuvchi ma`lumotlar tizimi saqlanadi. Ma`lumotlar bazasida ma`lumotlar bilan birga ularning tavsiflari saqlanadi. Foydalanuvchi dasturlariga bog`liq bo`lmagan bu ma`lumotlar metama`lumotlar deyiladi. Metama`lumotlar ma`lumotlar bazalari lug`atlarida saqlanadi.

Ma`lumotlar banki deganda ma`lumotlar integrallashgan majmuasini qo`llash uchun mo`ljallangan algoritmik, dasturiy, texnik va tashkiliy vositalarga aytiladi.

Axborot texnologiyasi — axborot tizimlarni yaratish va yuritish usullari hamda vositalari majmuasi;

Axborot tizimi — axborotni to`plash, saqlash, izlash, o`nga ishlov berish hamda undan foydalanish imkonini beruvchi tizim.

Har qanday axborot tizimi strukturasi, qo`llanish sohasidan qat'iy nazar, bir nechta qismlardan iborat bo`ladi. Ta`minlovchi qismlar olti xil bo`ladi: texnik, matematik, dasturiy, tashkiliy, huquqiy ta`minotlar.

Ta`riflar:

Mohiyat - predmetlar, hodisalar, jarayonlar umumiy xossalari asosida birlashtiruvchi tushuncha.

Mohiyat nusxasi – umumiy mohiyatga ega konkret predmetlar, hodisalar, jarayonlar.

Mohiyatlar to`plami – bir xil xossalarga ega mohiyatlar to`plami.

Atribut – mohiyat xossasi.

Domen- atribut aniqlanish sohasi.

Tobe mohiyat – agar bir mohiyat mavjudligi ikkinchi mohiyat mavjudligiga bog‘liq bo‘lsa *tobe mohiyat* deyiladi.

Birlamchi kalit – mohiyat nusxasi unikalligini ta‘minlovchi atribut yoki atributlar ro‘yxati.

Tashqi kalit – bog‘langan mohiyat birlamchi kaliti.

Bog‘lanish - bir nechta mohiyatlar orasidagi assotsiatsiya.

Bog‘lanishda mohiyat roli – aloqada mohiyat bajaradigan funksiya.

Bog‘lanishlar to‘plami – har bir biror mohiyatlar to‘plamiga tegishli n ($n > 1$) mohiyatlar orasida aloqa.

Bog‘lanish darajasi - boshqa mohiyat bilan aloqalar to‘plami orqali assotsiatsiya qilinuvchi mohiyatlar soni.

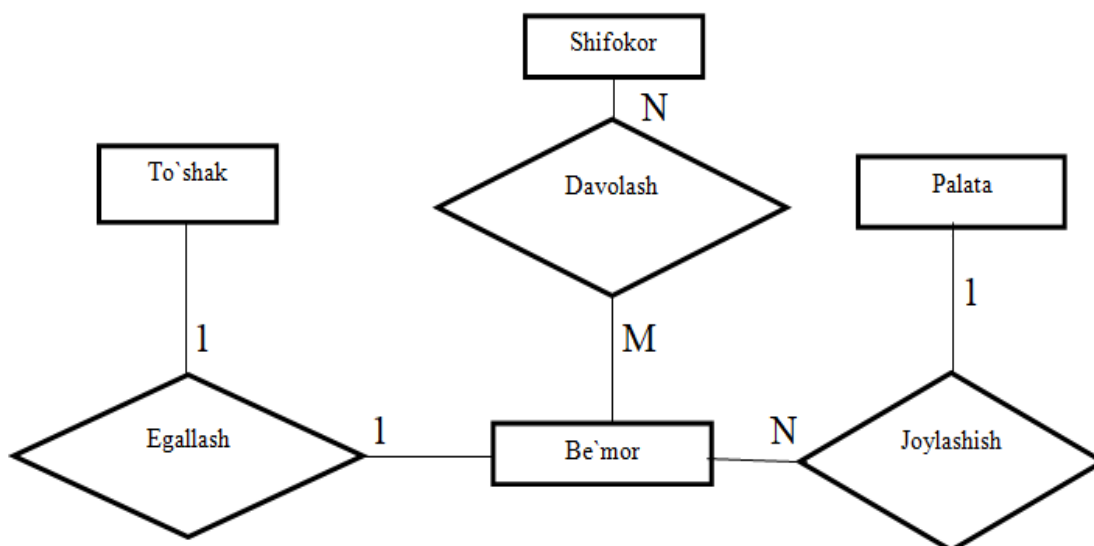
Bog‘lanish tegishlilik sinfi yoki kardinalligi. Mohiyatlar orasida quyidagi munosabatlar aniqlangan:

Birga – bir (1:1) munosabat. Ikki mohiyat orasida birga bir bog‘lanish mavjud deyiladi agar birinchi mohiyatning har bir nusxasiga ikkinchi mohiyatning bitta nusxasi mos kelishi mumkin bo‘lsa, va aksincha.

Birga – ko‘p (1:n) munosabat. Ikki mohiyat orasida birga ko‘p bog‘lanish mavjud deyiladi. Agar birinchi mohiyatning har bir nusxasiga ikkinchi mohiyatning bir nechta nusxasi mos kelishi mumkin bo‘lsa, va ikkinchi mohiyatning har bir nusxasiga birinchi mohiyatning bittadan ko‘p bo‘lmagan nusxasi mos kelishi mumkin bo‘lsa.

Ko‘pga – bir (n:1) munosabat. Ikki mohiyat orasida ko‘pga bir bog‘lanish mavjud deyiladi agar birinchi mohiyatning bir nechta nusxasiga ikkinchi mohiyatning ko‘pi bilan bitta nusxasi mos kelishi mumkin bo‘lsa va ikkinchi mohiyatning har bir nusxasiga birinchi mohiyatning bir nechta nusxasi mos kelishi mumkin bo‘lsa.

Ko‘pga – ko‘p (m:n) munosabat. Ikki mohiyat orasida ko‘pga ko‘p bog‘lanish mavjud deyiladi, agar birinchi mohiyatning har bir nusxasiga ikkinchi mohiyatning bir nechta nusxasi mos kelishi mumkin bo‘lsa va aksincha.



1.1-Rasm. "Mohiyat-aloqa" diagrammasi

Modellar turlari. Datalogik bosqichda xilma – xil ma‘lumotlar bazalari bilan ishlash ta‘minlanadi. Ma‘lumot modeli, bu MB ni ma‘lumot elementlari to‘plami orasidagi bog‘lanish strukturalarini tasvirlovchi umumiy sxemadir. Ma‘lumot modeli tushunchasini aniq ta‘rifini Kodd bergan. U ma‘lumot modelini 3 ta kerakli komponentasini keltirgan:

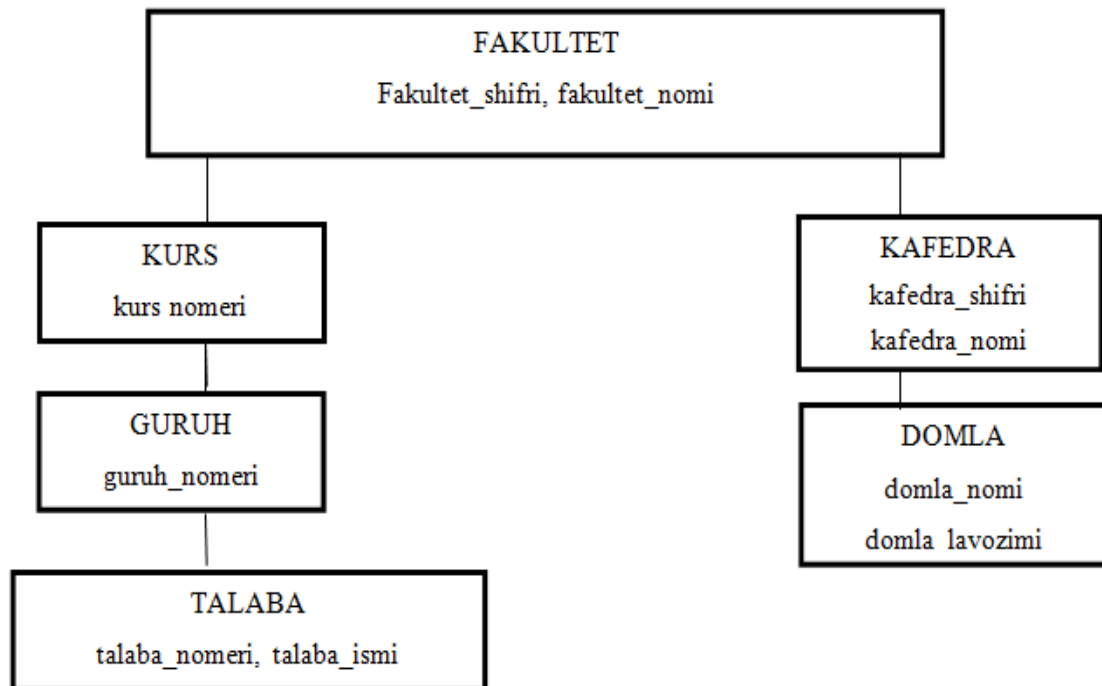
- 1.Mavjud bo‘lgan ma‘lumot strukturalarini aniqlash vositalari majmuasi;
- 2.Ma‘lumotlarni qidirish va takomillashtirish uchun MB holatiga qo‘llaniladigan amallar to‘plami;
- 3.Oshkor yoki oshkormas holda MB holatini aniqlovchi va bir butunligini ta‘minlovchi vositalar to‘plami.

Ierarxik modellar. Ierarxik modellarda yozuvlar daraxtsimon strukturali ko‘rinishda bo‘ladi. Ierarxik tuzilishli ma‘lumotlar sodda yaratiladi. Bu ko‘pincha tadbirlarda qulay, lekin ma‘lumotlarni ko‘plari daraxtsimon strukturali bog‘lanish tabiatiga ega emas.

Ierarxik model «birga bir» va «birga ko‘p» bog‘lanishlarni qo‘llaydi. Yana «ko‘pga ko‘p» bog‘lanishni ma‘lumotlarni ikkalash hisobiga qo‘llashi mumkin. Asosiy yaxlitlik cheklanishi shundaki avlod ajdodsiz bo‘lishi mumkin emas. Shuning uchun ajdod tugun o‘chirilganda butun ostki daraxti ham o‘chiriladi.

Ierarxik model ustunligi xotira samarali foydalanishi, asosiy amallarning tez bajarilishi, ierarxik axborot bilan ishlash qulayligi.

Ierarxik model kamchiligi ajdodga ega bo'lmagan nusxalarni saqlash mumkin emasligi «ko'pga ko'p» bog'lanishni joriy etish murakkabligi.



1.2-Rasm. Daraxtsimon ma'lumotlar modeli

Tarmoqli model. Agar munosabatdagi joriy element bir nechta berilgan elementga ega bo'lsa, bunday bog'lanishlarni ierarxik strukturalar bilan tavsiflab bo'lmaydi.

Bunday tuzilmalar tarmoqli graflar bilan tavsiflanadi. Tarmoqli strukturalarda element ixtiyoriy boshqa element bilan bog'lanishi mumkin.

Ya'ni, tarmoqli bir nechta kichkina obyektlardan topgan yirik obyekt deb qarash mumkin.

Tarmoqli MB turida ma'lumotlar bilan quyidagi ishlarni bajarish mumkin.

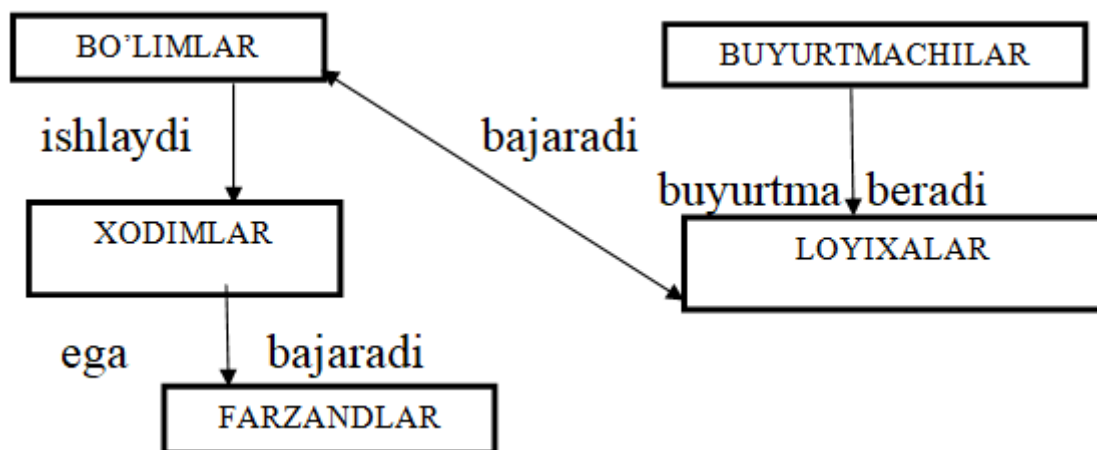
- MB yozuvlarini qidirish
- Yangi yozuvni yaratish
- Joriy yozuvni olib tashlash
- Joriy yozuvni tiklash

- Yozuvni bog‘lanishga qo‘shish
- Yozuvni bog‘lanishdan olib tashlash
- Bog‘lanishlarni o‘zgartirish

Bog‘lanish «ko‘pga ko‘p» tashkil etish uchun bitta yozuvga bog‘langan ikkita «birga ko‘p» bog‘lanish tashkil etiladi. Lekin quyidagi cheklanishlar bajarilishi lozim:

- 1) har bir to‘plamda faqat bitta yozuv ajdod bo‘lishi mumkin, lekin bitta yozuv bir nechta to‘plamda ajdod bo‘lishi mumkin;
- 2) bir yoki bir nechta yozuv bitta to‘plam elementi bo‘lishi mumkin;
- 3) yozuv bir nechta to‘plamga kirishi mumkin;
- 4) yozuv bitta to‘plamda ajdod, boshqasida avlod bo‘lishi mumkin;
- 5) ikki yozuv orasida ixtiyoriy sondagi to‘plam bo‘lishi mumkin;
- 6) yozuvlar siklni tashkil etishi mumkin;
- 8) yozuv biror to‘plam a‘zosi bo‘lishi shart emas.

Tarmoqli model afzalligi yozuvlar orasida ixtiyoriy aloqalarni o‘rnatish mumkinligi.



1.3-Rasm. Tarmoqli ma‘lumotlar modeli

Relyatsion model. Ma‘lumotlarning relyatsion modeli muallifi Edgar Frank hisoblanadi.

Relyatsion model ma‘lumotlarni sodda jadval ko‘rinishda tasvirlashga asoslangan. Jadvallar quyidagi talablarga javob berishi lozim:

- bir ustundagi hamma qiymatlar bir turga tegishli bo‘lishi lozim;

- jadvalda bir xil satr ikki marta uchramaydi;
- jadval ustunlari **atributlar** deb ataladi va har biri alohida nomga ega bo‘ladi. Jadval satrlari **yozuvlar** deb ataladi.
- atribut aniqlanish sohasi **domen** deb ataladi.
- har bir yozuv unikalligini ta‘minlovchi atribut yoki atributlar ro‘yxati **birlamchi kalit** deb ataladi.
- boshqa jadval birlamchi kalitini tashkil qiluvchi atribut yoki atributlar ro‘yxati **tashqi kalit** deb ataladi.

Jadvallar quyidagicha belgilanadi.

Jadval (atribut nomi,..., atribut nomi)

Yaxlitlik sharti. Birlamchi kalit cheklanishlari. Birlamchi kalit atributlari NULL qiymatga ega bo‘lmasligi lozim.

Tashqi kalit cheklanishlari. Biror yozuvda tashqi kalit atributlari NULL qiymatga ega bo‘lmasa mos jadvalda birlamchi kalit xuddi shunday qiymatli yozuvi mavjud bo‘lishi kerak.

Ma‘lumotlar bazasi. Hozirgi kunda inson faoliyatida ma‘lumotlar bazasi (**MB**) kerakli axborotlarni saqlash va undan oqilona foydalanishda juda muhim rol o‘ynamoqda. Sababi: jamiyat taraqqiyotining qaysi jabhasiga nazar solmaylik o‘zimizga kerakli ma‘lumotlarni olish uchun albatta, MBga murojaat qilishga majbur bo‘lamiz. Demak, MBni tashkil qilish axborot almashuv texnologiyasining eng dolzarb hal qilinadigan muammolaridan biriga aylanib borayotgani davr taqozasidir. Informatsion texnologiyalarning rivojlanishi va axborot oqimlarining tobora ortib borishi, ma‘lumotlarning tez o‘zgarishi kabi holatlar insoniyatni bu ma‘lumotlarni o‘z vaqtida qayta ishlash choralarining yangi usullarini qidirib topishga undamoqda. Ma‘lumotlarni saqlash, uzatish va qayta ishlash uchun MBni yaratish, so‘ngra undan keng foydalanish bugo‘ngi kunda dolzarb bo‘lib qolmoqda. Moliya, ishlab chiqarish, savdo-sotiq va boshqa korxonalar ishlarini ma‘lumotlar bazasisiz tasavvur qilib bo‘lmaydi. Ma‘lumki, MB tushunchasi fanga kirib kelgo‘nga qadar, ma‘lumotlardan turli ko‘rinishda foydalanish juda qiyin edi. Dastur tuzuvchilar ma‘lumotlarini shunday tashkil qilar edilarki, u faqat qaralayotgan masala ucho‘ngina o‘rinli bo‘lardi. Har bir yangi masalani hal qilishda ma‘lumotlar qaytadan tashkil qilinadi va

bu hol yaratilgan dasturlardan foydalanishni qiyinlashtirar edi. Har qanday axborot tizimining maqsadi real muhit obyektlari haqidagi ma'lumotlarga ishlov berishdan iborat. Keng ma'noda ma'lumotlar bazasi - bu qandaydir bir predmet sohasidagi real muhitning aniq obyektlari haqidagi ma'lumotlar to'plamidir. Predmet sohasi deganda avtomatlashtirilgan boshqarishni tashkil qilish uchun o'rganilayotgan real muhitning ma'lum bir qismi tushuniladi. Masalan, korxonalar, zavodlar, ilmiy tekshirish instituti, oliy o'quv yurti va boshqalar. Shuni qayd qilish lozimki, **MB**ni yaratishda ikkita muhim shartni hisobga olmoq zarur: **Birinchi**dan, ma'lumotlar turi, ko'rinishi, ularni qo'llaydigan programmalarga bog'liq bo'lmasligi lozim, ya'ni **MB**ga yangi ma'lumotlarni kiritganda yoki ma'lumotlar turini o'zgartirganda, programmalarni o'zgartirish talab etilmasligi lozim.

Ikkinchidan, **MB**dagi kerakli ma'lumotni bilish yoki izlash uchun biror programma tuzishga hojat qolmasin. Shuning uchun ham **MB**ni tashkil etishda ma'lum qonun va qoidalarga amal qilish lozim. Bundan buyon **axborot** so'zini **ma'lumot** so'zidan farqlaymiz, ya'ni **axborot** so'zini umumiy tushuncha sifatida qabul qilib, **ma'lumot** deganda aniq bir belgilangan narsa yoki hodisa sifatlarini nazarda tutamiz. Ma'lumotlar bazasini yaratishda, foydalanuvchi axborotlarni turli belgilar bo'yicha tartiblashga va ixtiyoriy belgilar birikmasi bilan tanlanmani tez olishga intiladi. Buni faqat ma'lumotlar tizimlashtirilgan holda bajarish mumkin.

Tizimli almashtirish – bu ma'lumotlarni tasvirlash usullari haqidagi kelishuvni kiritishdir. Agar ma'lumotlarni tasvirlash usuli haqida kelishuv bo'lmasa, u holda ular tizimlashtirilmagan deyiladi. Tizimlashtirilmagan ma'lumotlarga misol sifatida matn fayliga yozilgan ma'lumotlarni ko'rsatish mumkin.

Ma'lumotlar bazasidan foydalanuvchilar turli amaliy dasturlar, dasturiy vositalari, predmet sohasidagi mutaxassislar bo'lishi mumkin.

Ma'lumotlar bazasining zamonaviy texnologiyasida ma'lumotlar bazasini yaratish, uni dolzarb holatda yuritishni va foydalanuvchilarga undan axborot olishini ta'minlovchi maxsus dasturiy vosita, ya'ni ma'lumotlar bazasini boshqarish tizimi yordami bilan markazlashtirilgan holda amalga oshirishni nazarda tutadi. Ma'lumotlar bazasi – EHM xotirasiga yozilgan ma'lum bir strukturaga ega, o'zaro bog'langan va tartiblangan ma'lumotlar

majmuasi bo'lib, u biror bir obyektning xususiyatini, holatini yoki obyektlar o'rtasidagi munosabatni ma'lum ma'noda ifodalaydi. MB foydalanuvchiga strukturalashtirilgan ma'lumotlarni saqlash va ishlatishda optimal qulaylikni yaratib beradi.

Ma'lumki ma'lumotlarni kiritish va ularni qayta ishlash jarayoni katta hajmdagi ish bo'lib ko'p mehnat va vaqt talab qiladi. MB bilan ishlashda undagi ma'lumotlarning aniq bir strukturagi ega bo'lishi, birinchidan foydalanuvchiga ma'lumotlarni kiritish va qayta ishlash jarayonida undagi ma'lumotlarni tartiblashtirish, ikkinchidan kerakli ma'lumotlarni izlash va tez ajratib olish kabi qulayliklarni tug'diradi. MB tushunchasi fanga kirib kelgo'nga qadar, ma'lumotlardan turli ko'rinishlarda foydalanish juda qiyin edi. Bugungi kunda turli ko'rinishdagi ma'lumotlardan zamonaviy kompyuterlarda birgalikda foydalanish va ularni qayta ishlash masalasi hal qilindi. Kompyuterlarda saqlanadigan MB maxsus formatga ega bo'lgan muayyan tuzilmali fayl bo'lib, undagi ma'lumotlar o'zaro bog'langan va tartiblangandir.

Demak, ma'lumotlar bazasi deganda ma'lum bir strukturada saqlanadigan ma'lumotlar to'plami tushuniladi. Boshqacha qilib aytganda MB - bu ma'lum berilgan aniq bir strukturaga ega bo'lgan ma'lumotlarni o'z ichiga oluvchi maxsus formatga ega bo'lgan fayldir. Ma'lumotlarni strukturalashtirish - bu shunchaki ma'lumotlarni tasvirlashda qandaydir moslikni kiritish usulidir. Odatda MB ma'lum bir obyekt sohasini ifodalaydi va uning ma'lumotlarni o'z ichiga oladi, ularni saqlaydi va foydalanuvchiga ma'lumotlarni qayta ishlashda undan foydalanish imkonini yaratib beradi.

Ma'lumotlar bazasi – bu ma'lum bir predmet sohasiga oid tizimlashtirilgan(strukturalashtirilgan) ma'lumotlarning nomlangan to'plamidir. Ma'lumotlar bazasi - axborot tizimlarining eng asosiy tarkibiy qismi bo'lib hisoblanadi. Ma'lumotlar bazasidan foydalanish uchun foydalanuvchi ishini yengillashtirish maqsadida ma'lumotlar bazasini boshqarish tizimlari yaratilgan. Bu tizimlar ma'lumotlar bazasini amaliy dasturlardan ajratadi.

Ma'lumotlar bazasini boshqarish tizimi (MBBT) -bu dasturiy va apparat vositalarining murakkab majmuasi bo'lib, ular yordamida foydalanuvchi ma'lumotlar bazasini yaratish va shu bazadagi ma'lumotlar ustida ish yuritishi mumkin.

Juda ko‘p turdagi MBBT mavjud. Ular o‘z maxsus dasturlash tillariga ham ega bo‘lib, bu tillarga SUBD buyruqli dasturlash tillari deyiladi. MBBTga Oracle, Clipper, Paradox, FoxPro, Access va boshqalarni misol keltirish mumkin.

Ma‘lumotlar bazasi tushunchasi maydon, yozuv, fayl (jadval) kabi elementlar bilan chambarchas bog‘liq. **Maydon**-bu ma‘lumotlarni mantiqiy tashkil etishni elementar birligi bo‘lib, u axborotning eng kichik va bo‘linmas birligi bo‘lgan rekvizitga mos keladi. Maydonni tasvirlash uchun quyidagi tavsiflardan foydalaniladi:

Maydon nomi, masalan, familiyasi, ismi, tug‘ilgan sana, lavozimi, ish staji, mutaxassisligi.

1-maydon nomi	2-maydon nomi	3-maydon nomi	...	N-maydon nomi
Yozuv	Yozuv	Yozuv	Yozuv	Yozuv

Ma‘lumotlar bazasi tuzulmasining asosiy elementlari:

Maydon turi, masalan, son (chislovoy), simvol (simvolʼnyy), sana/vaqt (data/vremya), mantiqiy (logicheskiy).

Maydon uzunligi (o‘lchami), masalan, eng ko‘p simvollar sig‘imi;

Maydon aniqligi, (son tipidagi ma‘lumotlar uchun) masalan, sonning o‘nlik kasr qismini aks ettirish uchun o‘nlik raqamdan to‘rtta. **Yozuv** –bu mantiqiy bog‘langan maydonlar to‘plami. Yozuv tuzilishi uchun uning tarkibiga kiruvchi maydonlar tarkibi va joylashish ketma-ketligi bilan aniqlanib, ularni har biri ichida elementar **yozuvlarning nusxasi** deb ataladi. Yozuv obyektning biror bir elementi haqida to‘liq ma‘lumotni ifodalaydi.

Fayl (jadval) -bu bir xil tuzilmaga ega bo‘lgan yozuvning nusxalar to‘plamidir. U o‘zicha har bir maydonda qiymatga ega. *Misol* . Student faylidagi (jadvalidagi) yozuvlarning mantiqiy strukturasi tavsiflashga doir misolda ko‘rsatilgan. Student faylidagi yozuvning tuzilishi chiziqli bo‘lib, u o‘zgarmas uzunlikdagi yozuvlardan iborat. Yozuv maydonlari takrorlanuvchi qiymatlar guruhiga ega emas. Maydon qiymatiga murojaat uning nomeri bo‘yicha amalga oshiriladi.

Student faylidagi yozuvlar mantiqiy tuzilishini tavsifi.

Fayl nomi Student					
Maydon		Kalit belgisi	Maydon formati		
Nomni belgilash	To‘liq nomlanish (rekvizit)		Tip	Uzunligi	Aniqligi
Nomer	Talaba reyting daftarchasi nomeri	*	simvol	10	
Famil	Talaba familiyasi		simvol	10	
Ismi	Talaba ismi		simvol	8	
Ota_ismi	Talaba otasi ismi		simvol	10	
T_kun	Talabaning tug‘ilgan sanasi		sana	8	
O‘rta_baho	Talabaning o‘rtacha bahosi		son	3	2

Har bir MB jadvali o‘zining birlamchi kalitiga ega bo‘lishi mumkin. **Birlamchi kalit** deganda yozuvlar qaytarilmasligini ta‘minlovchi maydon (polya) yoki maydonlar guruhi tushiniladi. Birlamchi kalit sifatida ishlatiladigan maydon yoki maydonlar guruhi, bir xil yozuvga ega bo‘lmaslik shartini bajarishi kerak. Boshqa maydonlarda bir xil yozuvlar takrorlanishi mumkin. Shu sabab ular birlamchi kalit bo‘la olmaydi. Birlamchi kalit qisqa va sonli maydonlardan tashkil topishi maqsadga muvofiqdir. MB jadvaliga birlamchi kalitni kiritishdan maqsad, jadvaldagi ma‘lumotlarni izlash, tartiblashtirish va tanlab olishda qulaylikni beradi. Birlamchi kalit kiritish yoki kiritmaslik foydalanuvchi tamonidan MB jadvali strukturasi tashkil qilishda aniqlanadi.

Bo‘sh jadval yordamida qaram jadvaldagi mos ma‘lumotlarni chaqirishni ta‘minlash uchun qaram jadvalda tashqi kalit tashkil qilinadi. “Bitta-ko‘pga” bog‘lanish holatida tashqi kalit bosh jadvalda tashkil qilinadi. Birinchi va ikkinchi kalitlarni aniqlashda MBBT avtomatik ravishda jadvalda indeksni quradi.

AATning konseptual modeli. Axborot tizimi foydalanuvchilarning talabiga muvofiq axborotlarni yig'ish, qayd etish, uzatish, saqlash, to'plash, qayta ishlash, tayyorlash va taqdim etishga mo'ljallangan. Konseptual nuqtai nazardan qaraganda, axborot tizimi – bu operatsiyani bajaruvchi tizim va boshqaruvchi tizim o'rtasidagi vositachi sanaladi. Axborot-kommunikasiya texnologiyalari axborot tizimi ichidagi texnologiya sanaladi. Axborot tizimi tizimdagi ma'lumotlar, axborotlar bilan operatsiyalarni amalga oshiradi. Axborot tegishli muammoga qaratilgan bo'lib qarorlar qabul qilish uchun asos bo'lib xizmat qiladi. Axborot hal etilishi lozim bo'lgan vazifaga muvofiq va ushbu vazifani hal etuvchi xodimning qobiliyatiga muvofiq qayta ishlanadi.

Axborot tizimining funksional modeli. Axborot tizimining funksional modelini quyidagicha tasavvur etish mumkin. Mazkur modeldan ko'rinib turibdiki, axborot tizimining sohasi axborot obyektlari majmuidan iborat axborot makonini ifodalaydi. Umuman olganda axborot makoni bir xilda emas, chunki unda axborotlarning yuzaga kelishi, tashkil etilishi va joylashtirilishi jihatidan farqlanuvchi axborot obyektlarini o'zida saqlaydi. Tizim orqali barcha axborotlarning yuzaga kelishini quyidagi asosiy proseduralarga ajratish mumkin: saqlash, qidirish, qayta ishlash, kiritish va chiqarish. Birinchi uchtasi ichki bosqich sanaladi, to'rtinchi va beshinchilari esa mazkur tizim bilan axborot manbai va tashqi muhit o'rtasidagi aloqani ta'minlaydi. 1 - axborotni tashkil etish, saqlash va taqdim etish tizimi; 2 - axborotni kiritish, yangilash va tuzatish tizimi.

Axborot muhiti. Axborot muhiti o'zaro bog'langan uchta tarkibiy qismni o'z ichiga oladi. Bular: foydalanuvchining axborot tuzilmasi, axborot-kommunikasiya texnologiyalari, boshqaruvning ishtirok etuvchi obyektlari. Axborot infratuzilmasi axborotlarni o'z maqsadlariga erishish uchun foydalanadi. Axborot-kommunikasiya texnologiyalari foydalanuvchilarni zarur texnologiyalar bilan ta'minlash vositasi sanaladi. Axborot infratuzilmasi doirasida axborot-kommunikasiya texnologiyalari foydalanuvchilari ham o'zaro harakatlanuvchi o'ziga xos muhit sifatida ko'rib chiqiladi. Foydalanuvchi kerakli axborotlarni olish uchun rasmiy (formal) va norasmiy axborot tizimlari yordamida uning manbaiga murojat qilishi

lozim. Tashqi manbaga rasmiy tizim orqali ko‘rib boriladi. Bu tizim axborotlarni raqam va matnli ma‘lumot (statistik hisobotlar, kitob, jurnal, xabar va hakazo) ko‘rinishida taqdim etadi. Ichki manbaga murojat qilish axborot- kommunikasiya texnologiyalari komponentlari - kompyuterlar, tizimli va amaliy dasturiy ta‘minot hamda zarur hollarda kommunikasiya vositalari yordamida amalga oshiriladi. Ichki manbalar norasmiy tizim vositasida ma‘lumotlar bazasidan so‘rovga javob tariqasida foydalanuvchini axborot bilan ta‘minlaydi. Foydalanuvchi rasmiy va norasmiy tizimga suyanib ijtimoiy faoliyat, korxonalar va tashkilot ishini tavsiflovchi axborotlarni oladi. An‘anaviy axborot-kommunikasiya texnologiyalari rivojlanishi ikki an‘anaviy segment ma‘lumot va matndan tashqari, yana qo‘shimcha ikkita segment-tasvir va nutqni qayta ishlashni ta‘minlaydi. Axborot muhiti axborotlarni qayta ishlash, qabul qilish, o‘tkazish va qidirish qobiliyatiga ko‘ra qismlarga bo‘linadi. O‘z navbatida, qayta ishlash qobiliyati insonning axborotlarni qabul qilish imkoniyatiga ko‘ra aniqlanadi. Ayrim hollarda axborot shakl, hajm va hakazo belgilar bo‘yicha tarkiblashtirishni talab qiladi. Qayd etish lozimki, foydalanuvchi axborot manbaiga muhtoj bo‘ladi. Chunki u axborotlarni uyg‘unlashtirishga qancha ko‘p vaqt va kuch sarflasa, samaradorlik ham shuncha kam bo‘ladi. Foydalanuvchining faoliyat samaradorligini oshirish uchun axborotlarni integrallash jarayonini turli yo‘llar bilan amalga oshirish mumkin. Integrallash jarayonining darajasini belgilashni yangi axborot-kommunikasiya texnologiyalari SHK, MBBTlar ta‘minlaydi. Asosiy urg‘u turli xildagi axborotlarni qayta ishlash imkonini beradigan va o‘z ichiga integrallashgan ma‘lumotlarni qayta ishlash vositalari, ma‘lumotlar bazalarini boshqarish tizimi, aloqa vositalari va matnli prosessorlarni oluvchi amaliy dasturiy ta‘minotga qaratiladi. Axborot tizimining namunaviy tarkibi. Avtomatlashtirilgan axborot tizimiga quyidagilar kiradi: odam(xodim), texnik vositalar, axborot va dasturiy ta‘minot. Ular birgalikda boshqaruv usullari uchun ma‘lumotlarni qayta ishlaydi.

Axborot tizimining namunaviy tarkibi:

- avtomatlashtirilgan axborot tizimi
- texnik vositalar Dasturiy ta‘minoti
- boshqaruv usullari
- tizimli dasturiy ta‘minoti

- amaliy dasturiy ta‘minoti
- boshqaruv modeli Inson (xodim)
- axborot ta‘minoti

Avtomatlashtirilgan axborot tizimi tuzilmasi.

Avtomatlashtirilgan axborot tizimi ta‘minlovchi va funksional qismlarga ega. Ta‘minlovchi qism axborot, texnik, matematik, dasturiy, tashkiliy, xuquqiy, uslubiy, ergonomik, psixologik va lingvistik ta‘minotdan iborat bo‘ladi.

Axborot ta‘minoti – tashkilotda aylanib yuruvchi axborotlarni tashkil etish shakli, joylashtirilish hajmi (axborotlarni tasniflash va kodlashtirish, hujjatlarni unifikasiyalashtirish tizimi, axborot oqimlarining yagona tizimi) bo‘yicha loyiha qarorlarining shuningdek, ma‘lumotlar bazasi tuzilish uslubining majmuidir. U ko‘rsatkichlarni, ma‘lumotnomalarni axborotni tasniflovchi hujjatlarning unifikasiyalashgan tizimini, tashuvchi vositalardagi axborotlarni o‘z ichiga oladi.

Texnik ta‘minot.

Texnik ta‘minot – axborot tizimlari ishi uchun mo‘ljallangan texnik vosita-lar kompleksi, shuningdek, ushbu vositalar va texnologik jarayonlarga tegishli hujjatlardir. *Matematik ta‘minot.*

Matematik ta‘minot – axborot tizimlarida vazifalarni hal etishda foydalaniladigan axborotlarni qayta ishlash algoritmi va modellari, matematik uslublari majmui.

Dasturiy ta‘minot – bu axborot tizimlarining maqsad va vazifalarini amalga oshirish uchun dasturlar majmui, shuningdek texnik vositalar kompleksining me‘yorida ishlab turishi demakdir. Dasturiy ta‘minot tarkibiga umumiy tizimi va maxsus dasturli mahsulotlar, shuningdek texnik hujjatlar, jumladan: operasion tizimlar, dasturlash tizimi, dasturchining asbob-uskuna vositasi, test va tashhis dasturlari, telekommunikasiyaning dasturiy vositasi, axborotlarni himoyalash, funksional dasturiy ta‘minot (avtomatlashtirilgan ish joylari, ma‘lumotlar bazalarini boshqarish tizimi va hokazo).

Umumiy tizimli dasturiy ta‘minot. Ularga foydalanuvchiga mo‘ljallangan va axborotlarni qayta ishlashning an‘anaviy vazifalarini hal etish uchun belgilangan dasturlar kompleksi kiradi. Ular

kompyuterlarning imkoniyatlarini kengaytirish, ma'lumotlarni qayta ishlash jarayonlarini boshqarish va nazorat qilishga xizmat qiladi.

Maxsus dasturiy ta'minot. Aniq bir dasturiy tizimni yaratishda ishlab chiqilgan dasturlar majmuini ifoda qiladi. Uning tarkibiga turli darajada ishlab chiqilgan:

- texnik ta'minot
- matematik ta'minot
- lingvistik ta'minot
- dasturiy ta'minot
- axborot ta'minoti
- tashkiliy Uslubiy
- huquqiy Ergonomik
- psixologik
- Ishlab chiqarish
- reja
- hisob-kitob
- marketing
- xodimlar
- ombor
- transport
- ta'minlovchi qism
- funktsional qism

modellar, ma'lum bir obyektning ishlashini aks ettiruvchi amaliy dasturlar paketi kiradi. Uslubiy ta'minot va tashkiliy ta'minot – axborot tizimlarini ishga tushirish va ishlatish jarayonida axborot tizimi xodimlarining texnik vositalar bilan va o'zaro ta'sirini belgilovchi metodlar, vositalar va hujjatlar majmuidir.

Ergonomik ta'minot (sharoit) – ish joylariga, axborot modellariga, xodimning ish faoliyatiga nisbatan turli ergonomik talablardan iborat hujjatlar, ularni amalga oshirish usullari to'plamidir. Maqsad – xodim ishining yuqori samaradorligini ta'minlash. **Huquqiy ta'minot** – axborot tizimlarining huquqiy maqomi va uni ishga tushirishni belgilovchi huquqiy me'yorlar majmuidir. Axborotlarni olish qayta o'zgartirish va foydalanish tartibi belgilab qo'yiladi. Huquqiy ta'minotning asosiy maqsadi qonunchilikni mustahkamlash sanaladi. Huquqiy ta'minot tarkibiga qonunlar, farmoyish, hukumat

organlari qarorlari, buyruqlari, yo‘riqnomalari va vazirliklar, idoralar, mahalliy hukumat organlarining boshqa me‘yoriy hujjatlari kiradi. Axborot tizimlarining funksional qismi axborot tizimlarining vazifa va topshiriqlari bajarilishini ta‘minlaydi. Amalda bu yerda tashkilotni boshqarish tizimining modeli saqlanadi. Mazkur tizim doirasida boshqaruv maqsadlarining funksiyalariga funksiyalarning esa axborot tizimi kenja tizimiga o‘zgarishi ro‘y beradi. Kenja tizimlar vazifalarni amalga oshiradi. Ular tizimning biror bir belgisiga ko‘ra ajratib ko‘rsatilgan qismidir. Odatda axborot tizimlarida funksional qism funksional belgilariga ko‘ra kenja tizimlarga bo‘linadi:

- Boshqaruv darajasi(oliy, o‘rta, quyi);
- Boshqariladigan resurs turi(moddiy, mehnat, moliyaviy va hakazo);
- Qo‘llanish sohasi (bank, jamg‘arma bozori va hakazo);
- Boshqaruv ishi va davri.

Shuni qayd etish lozimki, avtomatlashtirilgan axborot tizimlarining funksional qismi tarkibi va mazmuni ma‘lum bir obyektga bog‘liq. Axborot tizimlarining ta‘minlovchi qismi tarkibi va mazmuni turli obyektlar uchun bir xilda bo‘ladi.

Nazorat savollari

1. Maydon uzunligi nima?
2. Obyekt ga tarif bering.
3. Obyektlar atributlariga misollar keltiring.
4. MB administrator deganda nimani tushunasiz?
5. Ma‘lumotlar banki tushunchasiga ta‘rif bering.
6. Ma‘lumotlar Bazasini ishlatish afzalliklari sanang.
- 7.

1.2. Ma‘lumot bazasi tizimining arxitekturasi va uch bosqichli arxitektura

Tarmoqlangan tizimlar asosiy tushunchalari. Markazlashgan ma‘lumotlar bazasi bir kompyuter xotirasida saqlanuvchi ma‘lumotlar bazasidir.

Taqsimlangan ma‘lumotlar bazasi– bu tarmoq orqali o‘zaro bog‘langan bir necha kompyuter xotirasida saqlanuvchi ma‘lumotlar bazasidir. Tarmoq xar bir tuguni o‘z ma‘lumotlar

bazasiga ega bo'lishi mumkin yoki ma'lumotlar bazasi bir qismini saqlashi mumkin lekin foydalanuvchi uchun yagona ma'lumotlar bazasi bo'lib ko'rinishi lozim

Taqsimlangan ma'lumotlar bazasini boshqarish tizimi taqsimlangan dasturiy tizimdir. Lekin foydalanuvchi yagona tizim sifatida foydalanishi lozim.

Teng darajali tarmoqlar - tarmoqqa ulangan barcha kompyuterlar teng xuquqli. Bunday tarmoq *bir xil martabali* (peer-to-peer) tarmoq deb ataladi.

Tarmoq va fayl serverli texnologiya («fayl-server» arxitektura). Qo'yilgan masalalar murakkabligining oshishi, shaxsiy kompyuterlar va lokal xisoblash tarmoqlarining paydo bo'lishi yangi «fayl-server» arxitekturani paydo bo'lishiga zamin yaratdi. Tarmoq orqali murojat qilish mumkin bo'lgan ma'lumotlar bazasining ushbu arxitekturasida tarmoqdagi bitta kompyuter server sifatida ajratilgan va unda ma'lumotlar bazasining fayllari joylashtirilgan. Foydalanuvchilarning so'roviga mos xolda fayllar fayl-serverdan foydalanuvchilarning ishchi stantsiyalariga uzatilgan va u erda ma'lumotlarni qayta ishlash bilan bog'liq asosiy ishlar bajarilgan. Bunda markaziy server ma'lumotlarni qayta ishlashda qatnashmasdan faqat fayllarni saqlovchi vazifasini bajargan

Tarmoqlangan ma'lumotlarni boshqarish tizimlari. «Mijoz – server» arxitekturasi quyidagicha qurilgan:

- Ma'lumotlar bazasi fayllar to'plami shaklida maxsus kompyuterning (tarmoq serveri) qattiq diskida joylashadi.
- MBBT ham tarmoq serverida joylashadi.
- Mijoz kompyuterlaridan iborat bo'lgan lokal tarmoq mavjud bo'lib, xar bir kompyuterda MB bilan ishlovchi mijoz dasturi o'rnatilgan.
- Har bir mijoz kompyuterda foydalanuvchilar dasturni ishga tushirish imkoniga ega. Dasturning foydalanuvchi interfeysi yordamida foydalanuvchilar ma'lumotlarni tanlash/yangilash uchun serverdagi MB ga murojaat qiladi. Murojaat qilish uchun maxsus SQL so'rovlar tili ishlatiladi, ya'ni serverga faqat so'rov matni jo'natiladi.

- MBBT da serverdagi MB ning fizik strukturasi xaqidagi ma'lumotlar saqlanadi.

- MBBT serverdagi ma'lumotlarga murojaatni qayd qiladi va serverda

ma'lumotlarni qayta ishlash amallari bajariladi va natija mijoz kompyuteriga yuboriladi. SHu tarzda MBBT natijalarni mijoz dasturiga yuboradi. Dastur esa o'zidagi foydalanuvchi interfeysi yordamida natijalarni tasvirlab beradi. Bajariladigan funktsiyalar server va mijoz o'rtasida taqsimlanishi:

Server har doim kutish xolatida bo'ladi, u mijozlardan so'rovlar kutgan xolda, tarmoqni domiy ravishda *eshitib* (listen) turadi.

Mijoz server bilan bog'lanadi va unga ko'rsatilishi kerak bo'lgan xizmatlar, masalan, kerakli fayl nomi, bayon etilgan *so'rov* (request) jo'natadi.

Server so'rovni qayta ishlaydi va javob (response) jo'natadi.

Bundan keyin aloqa uzilishi yoki davom etishi mumkin. Bu jarayon *sessiya* (session) deb ataladigan aloqa seansini tashkil etadi.

Dastur-serverlarni bir biridan farqlash uchun dastur-serverga *port nomeri* (port) beriladi. Bu oddiy musbat butun son bo'lib, ixtiyoriy bo'lishi mumkin, ammo keng tarqalgan protokollar uchun standart raqamlar beriladi.

Masalan, elektron pochta portining standart raqami 25, fayllarni uzatish — 21, Web-server — 80. Standart raqamlar 0 dan 1023 gacha bo'lgan oraliqda bo'ladi. 1024 dan boshlab, 65 535 gacha bo'lgan sonlarni xususiy portlarni raqamlash uchun ishlatish mumkin.

Server yukini teng taqsimlash uchun, ko'p xollarda bir necha portlar bir xil turdagi dastur serverlar tomonidan tinglanadi. Web-server, 80 raqamli portdan tashqari, 8080, 8001 va boshqa portlarni ham tinglashi mumkin.

Tarmoqlangan tizimlarda ma'lumotlarni qayta ishlash texnologiyalari. Uch zvenoli (ko'p zvenoli) «mijoz – server» arxitektura. Uch zvenoli (ba'zida ko'p zvenoli) arxitektura (N-tier yoki multi-tier) «mijoz-server» texnologiyasining rivojlangan va takomillashtirilgan shaklini ifodalaydi. «Mijoz – server» arxitekturasi ikkita zvenodan

tashkil topgan: birinchi zveno – mijoz dasturi, ikkinchi zveno – MB serveri + MB ning o‘zi.

Uch zvenoli arxitektura bu butunicha biznes-logika, oldin mijoz dastur bo‘lgan zveno aloxida ilovalar zvenosiga kiritiladi. Bunda mijoz dasturi sifatida faqat foydalanuvchi interfeysi qoladi. Unda mijoz ilovasi sifatida Web-brauzer ishlatiladi. Uch zvenoli arxitekturadan foydalanganda biznes – logika o‘zgarganda mijoz dasturlarinin o‘zgartirish va barcha foydalanuvchilarni uchun yangilash zarur emas. Bundan tashqari foydalanuvchi kompyuterining apparaturasiga bo‘lgan talab keskin pasayadi.

Uch zvenoli arxitekturaning ishlash tartibi quyidagicha qurilgan:

- Ma’lumotlar bazasi fayllar to‘plami shaklida maxsus ajratilgan kompyuterning (tarmoq serveri) qattiq diskida joylashadi.

- MBBT ham tarmoq serverida joylashadi.

- Maxsus ajratilgan ilovalar serveri mavjud bo‘lib, unda ishbiarmon doiraning (biznes – logika) dasturiy ta’minoti joylashadi.

- Ko‘p sonli mijoz kompyuterlari mavjud va ularda "kichik mijoz"

deb ataluvchi mijoz dasturi o‘rnatilgan bo‘lib ular foydalanuvchi interfeysini xosil qiladi.

- Har bir mijoz kompyuterida foydalanuvchilar kichik – mijoz dasturini ishga tushirish imkoniga ega. Bu dastur yaratgan foydalanuvchi interfeysi yordamida foydalanuvchilar ilovalar serveridagi ishbiarmon doiraning dasturiy ta’minotiga murojaat etadi.

- MBBT da serverdagi MB ning fizik strukturasi xaqidagi ma’lumotlar saqlanadi.

- MBBT serverdagi ma’lumotlarga murojaatni bajaradi va uning natijalari ilovalar serveriga jo‘natiladi.

- Ilovalar serveri natijani mijoz dasturiga (foydalanuvchiga) jo‘natadi.

- Ilova foydalanuvchi interfeysi yordamida so‘rov natijalarini tasvirlab beradi.

Tarmoqlangan tizim arxitekturasi. Ko‘p sonli mustaqil serverlarga ega arxitektura. Xar bir server o‘z lokal ma’lumotlar bazasiga ega. Mijozlar serverlar xaqida ma’lumot va dasturiy ta’minotga ega.

O‘zaro aloqa qiluvchi serverlardan tashkil topgan arxitektura. Xar bir server qolgan serverlar xaqida ma’lumotga ega. Xar bir mijoz o‘z serveriga ega.

Bir darajali serverlardan iborat arxitektura. Xar bir kompyuterda ma’lumotlar bazasi va dasturiy ta’minot saqlanadi.

Taqsimlangan ma’lumotlar bazasi tamoyillari.. Taqsimlangan ma’lumotlar bazasi uchun Deyt K.Dj. quyidagi fundamental tamoyilni ilgari surgan:

Taqsimlangan tizim foydalanuvchi uchun taqsimlanmagan tizim sifatida ko‘rinishi lozim. Bu tamoyildan quyidagi qoidalar kelib chiqadi:

1. Lokal avtonomiya. (local autonomy)

Xar bir tugundagi ma’lumotlar lokal boshqariladi. Xar bir tugundagi ma’lumotlardan mustaqil ma’lumotlar bazasi sifatida foydalaniladi.

2. Markaziy tugundan mustaqillik. (no reliance on central site)

Ideal tizimda xamma tugunlar teng va mustaqil. Xar bir tugundagi ma’lumotlar bazasi to‘liq ximoyalangan.

3. Uzluksiz amallar. (continuous operation)

Ma’lumotlarga doim murojaat qilish mumkin. Ma’lumotlar ustida amallar uzluksiz bajariladi.

4. Joylashuvdan mustaqillik. (location independence)

Ma’lumotlarga murojaat ularning joylashuviga bog‘liq emas. Bu xususiyat joylashuv shaffofligi deb xam ataladi.

5. Fragmentatsiyadan musaqillik. (fragmentation independence)

Ma’lumotlarga murojaat ularning taqimlanishiga bog‘liq emas. Bu xususiyat fragmentatsiya shaffofligi deb xam ataladi.

6. Replikatsiyadan mustaqillik. (replication independence)

Ma’lumotlarga murojaat ularning nusxalanishiga bog‘liq emas. Bu xususiyat replikatsiya shaffofligi deb xam ataladi.

7. Taqsimlangan so‘rovlarni qayta ishlash. (distributed query processing)

Taqsimlangani so‘rov lokal so‘rov bilan bir xil ishlanadi. Ixtiyoriy turdagi so‘rovlar uchun yagona til ishlatiladi.

8. Taqsimlangan tranzaktsiyalarni qayta ishlash. (distributed transaction processing)

Taqsimlangan tranzaktsiya lokal tranzaktsiya bilan bir xil ishlanadi. Xar bir tugunda ma'lumotlar yaxlitligi ta'minlanadi.

9. Apparat ta'minotdan mustaqillik. (hardware independence)

Tugunlarda ixtiyoriy turdagi kompyuterlar ishlatilishi mumkin.

Bu xossa tarmoq shaffofligi deb xam ataladi.

10. Operatsion tizimdan mustaqillik. (operating system independence)

Tugunlarda ixtiyoriy turdagi operatsion tizimlar ishlatilishi mumkin.

11. Tarmoqdan mustaqillik. (network independence)

Tugunlar orasida ixtiyoriy turdagi kanallar va protokollar ishlatilishi mumkin.

12. Ma'lumotlar bazasidan mustaqillik. (database independence)

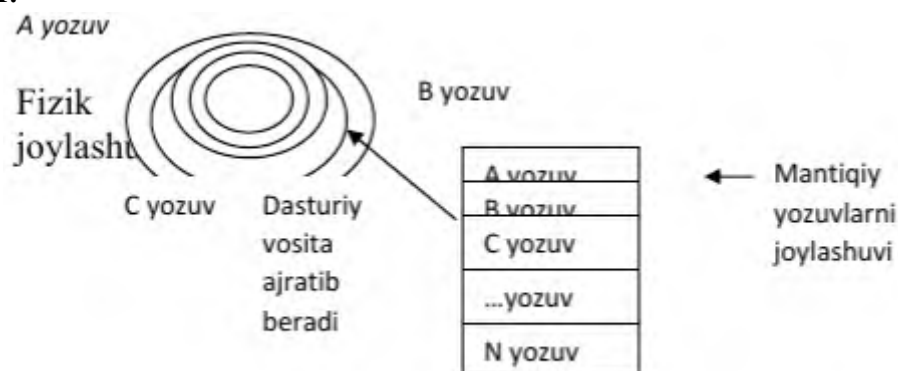
Tugunlarda ixtiyoriy turdagi ma'lumotlarni boshqarish tizimlaridan foydalanish mumkin.

MB ni logik va fizik tasvirlash. Ma'lumotlarni tavsiflash va ular orasidagi munosabat aloqalar o'rnatish 2 xil bo'ladi:

1 Logik yoki mantiqiy;

2 Fizik;

Fizik tasvirlashda ma'lumotlar mashinani tashqi xotirasida saqlashi bilan farqlanadi. Mantiqiy tasvirlashda esa amaliy dasturchi yoki foydalanuvchi tomonidan ma'lumotlarni tasvirlash ko'rinishi tushuniladi.



Fizik bosqich – fayllarda joylashgan ma'lumotlarni tashqi axborot saqlovchilarida joylashishini belgilaydi. Bu arxitektura ma'lumotlar bilan ishlaganda mantiqiy va fizik mustaqillikni ta'minlab beradi. Mantiqiy mustaqillik bitta ilovani o'zgartirishni, shu baza bilan ishlaydigan boshqa ilovani o'zgartirmasdan amalga

o'shinishni bildiradi. Fizik mustaqiliylik, saqlanuvchi ma'lumotlarni bir qattiq diskdan boshqasiga ko'chirganda uni ishlash qobiliyatini saqlab qolgan holda o'tkazishni bildiradi.

Ma'lumotlar bazasini boshqarish tizimi – *bu ma'lumotlar bazasini yaratish, ularni dolzarb holatini ta'minlash va undagi zarur axborotni topish ishlarini tashkil etish uchun mo'ljallangan dasturlar majmui va til vositasidir.* MB tashkil qilish, ularga qo'shimcha ma'lumotlarni kiritish va mavjud MBdan foydalanish uchun maxsus MBlar bilan ishlaydigan **dasturlar** zarur bo'ladi. Bunday dasturlar majmui **ma'lumotlar bazasini boshqarish tizimi (MBBT)** deb yuritiladi. Aniqroq qilib aytganda, **MBBT**–bu ko'plab foydalanuvchilar tomonidan MBni yaratish, o'nga qo'shimcha ma'lumotlarni kiritish va MBni birgalikda ishlatish uchun zarur bo'lgan dasturlar majmuidir. MBBTning tarkibida asosiy komponenti–bu *ma'lumotlar* bo'lsa, boshqa komponenti–*foydalanuvchilar*, **hardware**- texnik va **software**-dasturiy ta'minoti hisoblanadi. **Hardware** tashqi qo'shimcha xotiradan (disk, magnit lentasi) iborat bo'lsa, dastur qismi esa MB bilan foydalanuvchi o'rtasidagi muloqotni tashkil qilishni amalga oshiradi. MBning tuzilishi o'rganilayotgan ob'ektning ma'lumotlari ko'rinishi, ma'nosi, tuzilishi va hajmiga bog'liq bo'ladi. Odatda, foydalanuvchilar quyidagi kategoriyalarga bo'linadilar:

- foydalanuvchi-dastur tuzuvchi,
- tizimli dastur tuzuvchi,
- ma'lumotlar bazasi administratori.

Bunda dastur tuzgan **foydalanuvchi MBBT** uchun yozgan dasturiga javob beradi, **tizimli dastur tuzuvchi** esa butun tizimning ishlashi uchun javobgar hisoblanadi. U holda, **MB administratori** tizimning saqlanish holatiga va ishonchliligiga javob beradi. Insoniyat tomonidan katta miqdordagi bilimning to'planishi, turli xil axborotlarni saqlash masalasini hal qilishni talab etadi. Bunda, axborotlarni saqlash yagona maqsad hisoblanmaydi, balki u kerakli ma'lumotlardan kerakli vaqtda foydalana olish, turli xujjatlarni qayta ishlashga mo'ljallangan. Hozirgi kunda bir kancha **ma'lumotlar bazasini boshqarish tizimi** yaratilgan: REBUS, KARAT, SUBDQ, DBASE, FOXBASE, FOXRRO, ACCESS va boshqalar. Bu tizimlar quyidagi vazifalarni bajaradi:

- Ma'lumotlar bazasida joylashgan ma'lumotlarni ko'rish
- Ma'lumotlar bazasiga yangi yozuvlarni kiritish
- Ma'lumotlar bazasining yozuvlarini taxrirlash
- Ma'lumotlar bazasidan tegishli hisobotlarni olish
- Ma'lumotlar bazasining yozuvlaridan nusxa olish va boshqalar.

Har bir tizim bir-biridan buyruqning bajarilish tezligi va miqdori bilan farqlanadi. Ma'lumotlar bazasini boshqarish tizimlari bir vaqtning o'zida to'qqiz xil turdagi fayllar bilan ish yuritadi. Xotirada saqlanayotgan har bir fayl universal nomga ega bo'lib, fayl nomi va fayl turidan tashkil topadi. **Fayl nomi** foydalanuvchi tomonidan kiritilsa, faylning turi foydalanayotgan buyruqqa ko'ra tizim tomonidan avtomatik ravishda o'rnatilgan. Ma'lumotlar bazasi fayllari quyidagi turlarga ega:

- dbt - ma'lumotlar bazasining xotira fayli;
- dbf - ma'lumotlar bazasining aktiv fayli;
- ndx - ma'lumotlar bazasining tartiblashgan fayli;
- mem - xotira faylining ishchi fayli;
- rrg - ma'lumotlar bazasining buyruqli fayli;
- fmt - ma'lumotlar bazasining formatlashgan fayli;
- ebe - ma'lumotlar bazasining ko'rsatkichli fayli;
- frm - ma'lumotlar bazasining hisobot fayli;
- txt - ma'lumotlar bazasining matnli fayli;

Ma'lumotlar bazasining aktiv fayli foydalanuvchi tomonidan kiritilgan barcha axborotlarni o'zida saqlaydi. Har bir fayllar bir milliarddan ortiq yozuvni saqlashi mumkin. Bir yozuvda 128 ta ustunni ifodalash mumkin. Ma'lumotlar bazasi bilan ish yuritishda xotira qismini 15 ta oblastga ajratish mumkin va maxsus buyruqlar yordamida har bir oblastga alohida faylni chaqirib maxsus ishlarni bajarish mumkin. Ya'ni, yangi ma'lumotlarni kiritish, ortiqcha ma'lumotlarni o'chirish, nusxa olish, hisobot ishlarini amalga oshirish va boshqalar. Bundan tashqari, oblastlarga chaqirilgan fayllarni o'zaro bog'lash, ya'ni biriktirish mumkin. Fayllarni biriktirishda faqat ikkita soha qatnashadi. Yuqoridagi vazifalarni amalga oshirish uchun ma'lumotlar bazasini boshqarish tizimlarini tegishli buyruqlar bilan taxminlanadi. MBBT shunday dastur qobig'iki, uning yordamida jadvallarni strukturasi, jadvallar orasidagi bog'lanish, jadvallarni

ma'lumotlar bilan to'ldirgandan keyin, uning yordamida MB yaratiladigan programma vositasidir. Shu munosabat bilan MBBT bir qancha tarkibiy qismlardan iborat. Dasturiy vositalariga translyatorlar va MB ga ma'lumotlarni kiritadigan, qayta ishlaydigan, saqlaydigan, takomillashtiridigan, testdan o'tkazadigan, ma'lumotlarni kiritish, chiqarishni ta'minlaydigan boshqarish tizimlari kiradi. Asosiy dasturlash tili sifatida Object Pascal, C++ kabi tillarni ishlatiladi.

Nazorat savollari

1. Ma'lumotlar qaysi belgilari bo'yicha sinflarga ajratiladi?
2. Ma'lumotlarni logik (mantiqiy) va fizik tasvirlash nima?
3. Ma'lumot bazasini qanday uch bosqichli arxitekturalari mavjud?
4. Ma'lumotlar bazasi administratorining asosiy vazifalarini aytib bering.
5. Ma'lumotlar bazasida tasvirlar qanday saqlanadi? Misollar keltiring.
6. Ma'lumotlar bazasining uch bosqichli arxitekturasi chizmasini tasvirlang.
7. So'rovlarni qayta ishlashda MBBTning bajaradigan ishlar ketma-ketligini tavsiflab bering.

2-BOB. MA'LUMOTLAR BAZASI MODELLARI VA NORMAL FORMALAR

2.1. Ma'lumotlar bazasi modellari va mohiyat-aloqa modeli

Mashina muhitida ma'lumotlarni tashkil etish ikki pog'onadan iborat bo'lib, **mantiqiy va fizik** pog'onalar bilan harakterlanadi. Ma'lumotlarni bevosita fizik tashkil etishda ularni mashina "tashuvchisi"da joylashtirish usuli aniqlab olinadi. Zamonaviy amaliy dasturlar vositalarida ma'lumotlarni tashkil etishning bu pog'onasi avtomatik ravishda foydalanuvchi aralashuvisiz ta'minlanadi. Odatda, foydalanuvchi amaliy dasturlar vositalarining ma'lumotlarni mantiqiy tashkil etish haqidagi tushunchalar bilan operatsiyalar bajaradi. Mashina "tashuvchisi"da ma'lumotlarni mantiqiy tashkil etish, foydalanilayotgan dasturiy vositalardan va mashina muhitida ma'lumotlar bilan yuritishdan bog'liq. Ma'lumotlarni tashkil etishning mantiqiy usuli **foydalanilayotgan ma'lumotlar tuzilishining turi** va dasturiy vositalar orqali qo'llaniladigan *modelning shakli* aniqlanadi.

Ma'lumotlarning modeli - bu ma'lumotlar o'zaro bog'langan tuzilishlari va ular ustida bajariladigan operatsiyalar to'plamidir. Modelning shakli va unda foydalaniladigan ma'lumotlar tuzilishining turi dasturlash tizimi tilida foydalangan ma'lumotlarni tashkil etish va ishlov berish konsepsiyasini aks ettiradi.

Ma'lumki, aynan bir axborotni mashina ichki muhitida joylashtirish uchun ma'lumotlarning turli xil tuzilishlari va modellaridan foydalanish mumkin. Ulardan qaysi birini tanlash axborotlar bazasini yaratayotgan foydalanuvchining zimmasiga yuklatilgan bo'lib, u ko'plab omillarga bog'liq. Bu omillar qatoriga mavjud texnik va dasturiy ta'minotlar, hamda avtomatlashtirilayotgan masalalarning murakkabligi va axborotning hajmi kabilari kiradi. Ma'lumotlar modeli quyidagi tarkibiy qismdan iborat:

1. Foydalanuvchining ma'lumotlar bazasiga munosabatini namoyish etishga muljallangan ma'lumotlar tuzilmasi.

2. Ma'lumotlar tuzilishida bajarilish mumkin bo'lgan operatsiyalar.

Ular ko'rib chiqilayotgan ma'lumotlar modeli uchun ma'lumotlar tilining asosini tashkil etadi. Yaxshi ma'lumotlar tuzilmasining

o'zigina y yetarli emas. Ma'lumotlarni aniqlash tili (MAT) va ma'lumotlar bilan amallar bajarish tilining (MABT) turli operatsiyalari yordamida bu tuzilma bilan ishlash imkoniga ega bulish zarur. 3.Yaxlitlikni nazorat qilish uchun cheklashlar. Ma'lumotlar modeli uning yaxlitligini saqlash va himoya qilishga imkon beruvchi vositalar bilan ta'minlangan bo'lishi lozim.

Mashina muhitidagi ma'lumotlarning murakkabroq modellari – **tarmoqli va ierarxik** modellar bo'lib hisoblanadi. Bu modellar ularning o'zlariga xos turdagi ma'lumotlar bazasini boshqarish tizimida ishlatiladi. MBBTda ma'lumotlarni mantiqiy tashkil etish usuli ma'lumotlarning tarmoqli yoki ierarxik modeliga mos holda ko'rsatiladi. Bunday model o'zaro bog'liq obyektlarning majmuidir. Ikki obyektning aloqasi ularning bir-biriga tobeligini aks ettiradi. Tarmoqli yoki ierarxik modelida obyekt bo'lib, MBBT kiritilgan ma'lumotlar tuzilmasining asosiy turlari hisoblanadi. Turli MBBTlarda bu turdagi ma'lumotlarning tuzilmasi turlicha aniqlanishi va nomlanishi mumkin.

Modellarda ma'lumotlarning tuzilmalari. Ma'lumotlarning namunaviy tuzilmalariga quyidagilar kiradi: ma'lumotlarning elementi, ma'lumotlarning agregati, yozuv, ma'lumotlar bazasi va h.k. Bu elementlar va agregatlar o'zaro aloqada bo'lgan tuzilma bilan tavsiflanadi. Shuning uchun yozuvning tuzilmasi ierarxik harakterga ega bo'lishi mumkin. Bir xil tuzilmaga ega bo'lgan yozuv nusxalari to'plamining hammasi yozuv turini tashkil etadi.

Tarmoqli ma'lumotlar modeli. Agar munosabatdagi joriy element bir necha berilgan elementga ega bo'lsa, bunday bog'lanishlarni ierarxik strukturalar bilan tavsiflab bo'lmaydi. Bunday tuzilmalar tarmoqli graflar bilan tavsiflanadi.

Tarmoqli strukturalarida element ixtiyoriy boshqa element bilan bog'lanishi mumkin. Ya'ni, tarmoqli bir necha kichkina obyektlardan topgan yirik obyekt deb qarash mumkin.Ushbu chizmada bog'lanishlarni tarmoqli modelda tasvirlanishi keltirilgan.

Shunday qilib, tarmoqli model ma'lumotlar elementlari orasidagi xilma-xil bog'lanishlarni ixtiyoriy ko'rinishdagi grafik yordamida akslantiradi. Tarmoqli model yozuvlar to'plami va mos bog'lanishlar to'plamidan tashkil topadi. Bog'lanishlarni yaratish uchun alohida

cheklanishlar qo`yilmaydi. Misol: oddiy tarmoqli MB sxemasi sifatida quyidagini keltirish mumkin.

Ma'lumotlarning relyasion modeli. Relyasion MB kuchli nazariy fundamentga ega bo'lib, u matematik munosabatlar (otnosheniya) nazariyasiga asoslangan. Ma'lumotlarning relyasion modeli konsepsiyasi 1970 yilda E.F.Kodd tomonidan taklif qilingan bo'lib, u ma'lumotlarni tavsiflash va tasvirlashning amaliy dasturlaridan bog'liq bo'lmasligini ta'minlash masalasini hal qilish uchun xizmat qiladi.

Ma'lumotlarning relyasion modeli asosida «munosabat» tushunchasi yotib, u inglizcha relation so'zidan olingan. Ba'zi bir qoidalarga amal qilgan holda munosabatlarni ikki o'lchovli jadval ko'rinishda tasvirlash mumkin. Jadval har qanday odamga tushunarli va qulaydir. Real dunyo obyektlari haqidagi ma'lumotlarini EHM xotirasida saqlash va ular orasidagi munosabatlarni modellashtirish uchun munosabatlar (jadval) to'plamidan foydalanish mumkinligini E.F.Kodd isbotlab berdi. Masalan, «talaba» mazmunini saqlash uchun TALABA munosabatidan foydalaniladi. Bu mazmunning asosiy xususiyatlarini quyidagi jadvalning ustunlari tasvirlaydi:

2.1.Jadval. Talaba jafvali

Familiyasi I.O.	Tug'ilgan sana	Bosqich	Mutaxassisligi
Karimova M.N.	15/01/1979	4	menejement
Boltayev A.V	03/11/1978	4	Buxgalteriya.

Munosabat ustunlari **atributlar** deb ataladi va ularga nomlar beriladi. Munosabat atributlarining nomlaridan iborat ro'yxatini **munosabatlar sxemasi** deyiladi. Bizning misolimizdagi TALABA munosabatining sxemasi quyidagicha yoziladi: **TALABA (Familiyasi I.O., Tug'ilgan sana, Bosqich, Mutaxassisligi)**

Ma'lumotlarning relyasion bazasi - bu o'zaro bog'langan munosabatlar, ya'ni jadvallar to'plamidir. Har qanday munosabat (jadval) kompyuterlarning xotirasida fayl ko'rinishda joylashtiriladi. Ularning orasida quyidagi moslik mavjud:

Fayl	Jadval	Munosabat	Mazmuni
Yozuv Maydon	Satr Ustun	kortej atribut	mazmunning nusxasi atribut

Jadval hamma uchun juda qulay bo'lishi bilan bir qatorda ma'lumotlarni manipulyasiya qilishning asosiy uch operatsiyasini bajarish uchun noqulaydir, ya'ni tartiblash, indekslarning qiymatlari bo'yicha guruhlash va daraxt ko'rinishidagi parametrlar bilan ishlash. Jadvalda ushbu uch operatsiya bir-biri bilan chambarchas bog'langan. Bu esa ba'zi bir operatsiyalarni bajarishda ma'lum bir qiyinchiliklarga olib keladi. Masalan, ma'lumotlarni bir parametr asosida tartiblash ikkinchi bir parametr bo'yicha tartiblashni buzib yuborishi tufayli zarur ma'lumotlarni izlab topish operatsiyasi bir parametr bo'yicha osonlashsa, boshqalari bo'yicha qiyinlashadi. Kodd taklif qilgan usulining originalligi shundan iboratki, u munosabatlarga (jadvallarga) tadbiiq qilish uchun juda chiroyli qurilgan operatsiyalar tizimini ishlab chiqdi. Ularni amalga oshirish natijasida bir munosabatni boshqa munosabat orqali hisoblab chiqish imkoniyati paydo bo'ldi. Bu axborotlarni saqlanadigan va saqlanmaydigan (hisoblanadigan) qismlarga ajratish, hamda kompyuter xotirasini tejash zarur bo'lgan paytda axborotlarning saqlanmaydigan qismini saqlanadiganlar asosida hisoblab chiqish imkoniyatini beradi. Ma'lumotlarning relyasion bazasidagi munosabatlar ustida bajariladigan asosiy operatsiyalar sakkizta bo'lib, ular quyidagilardan iborat: -to'plamlar ustidagi ananaviy (traditsion) operatsiyalar, ya'ni to'plamlarning birlashmasi (yig'indisi), kesishmasi (ko'paytmasi), to'ldiruvchisi (ayirmasi), dekart ko'paytmasi, bo'lishmasi; -maxsus relyasion operatsiyalar, ya'ni proeksiyalash, bog'lanish (qo'shilish), birlashtirish (ulab qo'yish) va tanlash. Har bir ma'lumotlar bazasini boshqarish tizimining samaradorligi ushbu operatsiyalarning borligi va ularni bajarish vositalarining qanchalik qulayligi bilan aniqlanadi. Relyasion MBBTda munosabatlar ustida operatsiyalar bajarish uchun mo'ljallangan tillarini ikki sinfga ajratish mumkin: *relyasion algebra tili* (RAT) va *relyasion hisob tili* (RHT).

RAT relyasion algebraga (Kodd algebrasiga, a-algebraga) asoslangan. Ma'lum tartib munosabatlar ustida operatsiyalarni ketma-

ket yozish asosida xohlagan natijaga erishish mumkin. Shuning uchun RATni *protsedurali til* deyishadi.

RHT predikatlarni hisoblab chiqishning klassik usuliga asoslangan. Ular foydalanuvchilarga so‘rovlarni yozish uchun ma‘lum qoidalar to‘plamini beradi. Bunday so‘rovlarda faqat xohlagan natija haqidagi axborotlar bo‘ladi holos. Ushbu so‘rov asosida MBBT yangi munosabatlar hosil qilish yo‘li bilan avtomatik tarzda zarur natijani beradi. Shuning uchun RHTni *protseduralimas til* deyishadi.

Ma‘lumotlar bazasini loyihalashtirishda relyasion model bilan ishlash ancha noqulayliklarga olib keladi. Shu sabab ma‘lumotlar bazasini loyihalashda har xil semantik modellar ham ishlatiladi. Ulardan eng ko‘p tarqalganlaridan biriga –**ER** modeli deyiladi. 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 diagrammalarini oluvchi bir necha har xil turdagi komponentalarni birlashtirgan. Piter Chen mohiyatlar to‘plami va ular orasida bog‘lanish sifatida relyasion ma‘lumotlar strukturasi interpritatsiya qilishni taklif qildi. ER - modelining asosiy komponentalari mohiyat, bog‘lanish va atribut (xossa) bo‘lib hisoblanadi.

Mohiyat -bu ma‘lumotlari ma‘lumotlar bazasida saqlanishi kerak bo‘lgan biror real yoki tasavvur qilingan obyektidir. ER modeli diagrammasida mohiyat odatda to‘rtburchak shaklida tasvirlanib, uning ichiga mohiyat nomi qo‘yiladi.

Mohiyat – bog‘lanish. Axborot mantiqiy modellar orasida eng mashhur model "mohiyat-bog‘lanish" (entity - relationship model, ER - model) modelidir. "Mohiyat-bog‘lanish" modeli asosida datalogik modellar yaratilishi mumkin. Shuning uchun eng keng tarqalgan modeldir. Lekin bu model ma‘lumotlar ustida amallar aniqlamaydi. "Mohiyat-bog‘lanish" modeli 1976 yilda Piter Chen tomonidan taklif etilgan.

Nazorat savollari

1. Ma‘lumot baza modeli nima?
2. Ierarxik (shajara) modeli ma‘lumot va uning asosiy harakteristikalarini.
3. Tarmoqli model ma‘lumot va uning asosiy harakteristikalarini.
4. PS mohiyat aloqa usulida tavsiflaganda qanday ishlar bajariladi?

5. Mosliklarni (munosabatlar) qanday turlari bor? Ularni tavsiflang.
6. Mohiyat - aloqa diagrammasi qanday quriladi?
7. Axborot tizimlarini loyihalashga infologik yondoshishni asosiy qoidalarini tushuntiring.

2.2. Relyatsion ma'lumot bazasi va ma'lumotlar bazasida munosabatlar

Ma'lumotlarning relyatsion modeli muallifi Edgar Kodd hisoblanadi.

Relyatsion model ma'lumotlarni sodda jadval ko'rinishda tasvirlashga asoslangan. Jadvallar quyidagi talablarga javob berishi lozim:

1. Bir ustundagi hamma qiymatlar bir turga tegishli bo'lishi lozim;
2. Jadvalda bir xil satr ikki marta uchramaydi;
 - Jadval ustunlari **atributlar** deb ataladi va har biri alohida nomga ega bo'ladi.
 - Jadval satrlari **yozuvlar** deb ataladi.
 - Atribut aniqlanish sohasi **domen** deb ataladi.

Har bir yozuv unikalligini ta'minlovchi atribut yoki atributlar ro'yxati **birlamchi kalit** deb ataladi. Boshqa jadval birlamchi kalitini tashkil qiluvchi atribut yoki atributlar ro'yxati **tashqi kalit** deb ataladi. Jadvallar quyidagicha belgilanadi.

Jadval (atribut nomi,..., atribut nomi)

Yaxlitlik sharti. Birlamchi kalit cheklanishlari. Birlamchi kalit atributlari NULL qiymatga ega bo'lmasligi lozim.

Tashqi kalit cheklanishlari. Biror yozuvda tashqi kalit atributlari NULL qiymatga ega bo'lmasa mos jadvalda birlamchi kalit xuddi shunday qiymatli yozuvi mavjud bo'lishi kerak.

Birlamchi kalit qiymatlari o'zgartirilganda yoki o'chirilganda quyidagi qoidalar ko'rsatilishi mumkin:

CASCADE – tashqi kalit hamma satrlari o'zgartiriladi yoki o'chiriladi;

SET NULL – tashqi kalit qiymati NULL qiymat bilan almashtiriladi;

SET DEFAULT - tashqi kalit qiymati ko‘zda tutilgan qiymat bilan almashtiriladi;

NO ACTION – tashqi kalit qiymatlarini o‘zgartirish mumkin emas. Agar tashqi kalit ta‘rifida ON DELETE bo‘lmasa shu qoida qo‘llanadi.

RESTRICT birlamchi kalit ixtiyoriy satrini o‘chirish man qilinadi.

Bog‘liqlik. Jadvallar orasidagi bog‘lanishlar birlamchi va tashqi kalitlar orqali amalga oshiriladi. Har bir mohiyatga jadval mos qo‘yiladi.

Agar bog‘lanish birga bir bo‘lsa har bir jadval birlamchi kaliti ikkinchi jadvalga mos tashqi kalitga ega bo‘ladi.

Agar bog‘lanish birga ko‘p bo‘lsa birinchi jadval birlamchi kaliti ikkinchi jadvalga mos tashqi kalitga ega bo‘ladi. Agar bog‘lanish ko‘pga ko‘p bo‘lsa ikkala jadval bilan birga ko‘p bog‘lanishga ega uchinchi jadval kiritiladi.

Misol:

1- jadval. Talaba jadvali

Nomer	Familiya	Yosh	Viloyat	Guruh
1	Olimov	21	Xorazm	312-17
2	Olimov	21	Xorazm	312-17
3	Kamolov	22	Buxoro	312-16
4	Kamolov	22	Buxoro	312-17

2- jadval. Guruh jadvali

Guruh	Fakultet	Kurs
311-17	DI	1
312-16	KI	2
311-16	DI	1
312-17	KI	2

Normallashtirish. Normallashtirish deganda jadvallarni ma‘lum shartlarga mos keluvchi normal shaklga keltirishga aytiladi. Normallashtirish dekompozitsiya ya‘ni bir nechta jadvallarga ajratish yo‘li orqali amalga oshiriladi. Normal shaklga mos kelmaslik turli anomaliyalarga olib kelishi mumkin. Quyidagi anomaliyalar bo‘lishi mumkin.

Qaytarishlik anomaliyasi. Bir xil ma'lumot bir nechta satrlarda qaytarilishi mumkin. □

Yangilash anomaliyasi. Bitta satrda ma'lumot o'zgartirish uchun hamma satrlarni o'zgartirishga to'g'ri kelishi mumkin.

O'chirish anomaliyasi. Keraksiz ma'lumot o'chirilganda kerakli ma'lumot ham o'chib ketishi mumkin.

Qo'shish anomaliyasi. Kerakli ma'lumot qo'shilmasligi mumkin.

Birinchi normal forma (1NF): *Jadval birinchi normal formada deyiladi, qachonki undagi hech bir qatorning istalgan maydonda bittadan ortiq qiymat joylashmagan bo'lsa.*

Boshqacha aytganda birinchi normal forma bu har bir satr va ustun kesishmasida bitta bo'linmas atomar qiymat joylashgan jadvaldir.

Misol:

Xodimlar(Nomer, Ismi, Yoshi, Bolalar_nomi, Bolalar_Yoshi)

Bu jadvalda Bolalar_nomi, Bolalar_Yoshi atributlari ko'p qiymatli atributlar. Normallashtirish natijasida ikki jadval hosil qilinadi:

Xodimlar (Nomer, Ismi, Yoshi)

Bolalar (Nomer, Nomi, Yoshi, Xodim_nomeri)

Ikkinchi normal forma (2NF): *Jadval ikkinchi normal formada deyiladi, qachonki agar u birinchi normal formada bo'lsa va har bir birlamchi kalitga kirmagan atribut birlamchi kalit atributlariga to'liq funksional bog'liq bo'lsa.*

Agar ixtiyoriy vaqtda, bitta atribut har bir qiymatiga ikkinchi atributni bittadan ortiq bo'lmagan qiymati mos kelsa, ikkinchi atribut birinchi atributga *funksional bog'liq* deyiladi va quyidagicha belgilanadi: $A \rightarrow V$

Misol:

Buyurtma (Nomer, Maxsulot_nomi, firma, Mijoz_nomi, telefon, adres)

Bu jadvalda maxsulot va mijozga tegishli atributlar buyurtma nomeriga funksional bog'liq emas. Normallashtirish natijasida quyidagi jadvallar kelib chiqadi.

Buyurtma(Nomer, Maxsulot_nomeri, Mijoz_nomeri)

Maxsulot(Nomer, nom, firma)

Mijoz(Nomer, nom, telefon adres)

Uchinchi normal forma (3NF): *Jadval uchinchi normal formada deyiladi, qachonki u ikkinchi normal formada bo'lsa va tranzitiv bog'lanishlar bo'lmasa.*

Agar V atribut A atributga, S atribut esa V atributga bog'langan bo'lsa, ya'ni, $A \rightarrow V$ va $V \rightarrow S$ bo'lsa va $A \not\rightarrow V$, $V \not\rightarrow S$, $V \not\rightarrow A$ shartlar bajarilsa S atribut A atributga tranzitiv bog'langan deyiladi. Agar $C \rightarrow B$, bajarilsa qat'iy tranzitiv bog'lanish deyiladi.

Misol:

Xodim(Nomer, Ismi, Yoshi, Xaydovchi_Guvoxnoma_nomeri, turi, rangi)

Bu jadvalda mashina turi va rangi jadval birlamchi kaliti bo'lgan xodim nomeriga tranzitiv bog'liqdir. Normallashtirish natijasida quyidagi jadvallar kelib chiqadi.

Xodim(Nomer, Ismi, Yoshi, Xaydovchi_Guvoxnoma_nomeri)

Mashina (Xaydovchi_Guvoxnoma_nomeri, turi, rangi)

Normallashtirish algoritmi (ya'ni munosabatlarni 3NF ga keltirish algoritmi) quyidagi ko'rinishda yoritiladi:

1-qadam (1NF ga keltirish). Birinchi qadamda predmet soha tushunchasini tasvirlaydigan bitta yoki bir nechta jadvallar beriladi. Predmet soha modeli bo'yicha topilgan funksional bog'liqliklar yoziladi. Barcha jadvallar avtomatik ravishda 1NF da bo'ladi.

2-qadam (2NF ga keltirish). Agar qandaydir jadvalda atributlarning murakkab kalitga bog'liqligi topilgan bo'lsa, bu jadvalni bir nechta jadvalga quyidagi ko'rinishda ajratamiz: murakkab kalit qismiga bog'liq bo'lgan atributlar shu kalit qismi bilan birgalikda alohida jadvalga joylashtiriladi. Boshlang'ich jadvalda barcha kalitli atributlar qoladi.

Boshlang'ich jadval: $R(K1, K2, A1, \dots, An, B1, \dots, Bm)$.

Kalit: $(K1, K2)$ - murakkab.

Funksional bog'liqliklar:

□ $\{K1, K2\} \rightarrow \{A1, \dots, An, B1, \dots, Bm\}$ $\{K1\} \rightarrow \{A1, \dots, An\}$ - ba'zi atributlarni murakkab kalitning qismiga bog'liqligi.

Ajratilgan munosabatlar:

□ $R1(K1, K2, B1, \dots, Bm)$ - boshlang'ich jadvaldan qolgani.

Kalit $(K1, K2)$.

□ $R2(K1, A1, \dots, An)$ - boshlang'ich jadvaldan murakkab kalit qismi bilan olingan atributlar.

Kalit $K1$.

3-qadam (3NF ga keltirish). Agar ba'zi jadvalda ba'zi kalit bo'lmagan atributlarning boshqa ba'zi kalit bo'lmagan atributlarga bog'liqligi topilgan bo'lsa, u holda bu jadvalni bo'lishni quyidagi ko'rinishda amalga oshiramiz: boshqa kalit bo'lmagan atributlarga bog'liq kalit bo'lmagan atributlar alohida jadvallarga joylashtiriladi. Yangi jadvalda funksional bog'liqlik determinanti kalit hisoblanadi:

Boshlang'ich munosabat: $R(K, A_1, \dots, A_n, B_1, \dots, B_m)$.

Kalit: K .

Funksional bog'liqliklar:

$K \rightarrow \{A_1, \dots, A_n, B_1, \dots, B_m\}$ - barcha atributlarni jadval kalitiga bog'liqligi.

$\{A_1, \dots, A_n\} \rightarrow \{B_1, \dots, B_m\}$ - ba'zi kalit bo'lmagan atributlarning boshqa kalit bo'lmagan atributlarga bog'liqligi.

Ajratilgan jadvallar:

$R_1(K, A_1, \dots, A_n)$ - boshlang'ich jadvaldan qolgani.

Kalit K .

$R_2(A_1, \dots, A_n, B_1, \dots, B_m)$ - boshlang'ich jadvaldan funksional bog'liqlik determinanti bilan olingan atributlar. Kalit $\{A_1, \dots, A_n\}$.

Tarmoqlangan tizimlarda ma'lumotlarni taqsimlash usullari

Biror R munosabat berilgan. Bu munosabatni shunday R_1, \dots, R_n munosabatlarga ajratish lozimki berilgan samaradorlik kriteriyalariga javob bersin. Samaradorlik kriteriyasi bo'lib, murojaat vaqti yoki kompyuterlar yuklanganligi kabi xossalar ishlatilishi mumkin.

Fragmentatsiya korrekt, agar to'liq bo'lsa ya'ni kesmalarni o'z ichiga olmasa va qayta tiklanishi mumkin bo'lsa.

R munosabatni R_1, R_2, \dots, R_n fragmentlarga dekompozitsiyasi to'liq bo'ladi agar R bilan xar bir ma'lumotlar elementi biror R_i munosabatga tegishli bo'lsa. Dekompozitsiya qayta tiklanuvchan bo'ladi agar shunday relyatsion ifoda $\varphi(R_1, R_2, \dots, R_n)$ mavjud bo'lsaki $R = \varphi(R_1, R_2, \dots, R_n)$ bo'lsa.

Dekompozitsiya kesmalarga ega bo'lmaydi agar R bilan xar bir ma'lumotlar elementi faqat bitta fragmentda saqlansa.

Munosabat fragmentatsiyasi uch turi mavjud:

- gorizontal;
- vertikal;
- aralash.

Gorizontaal fragmentatsiya. Gorizontaal fragmentatsiya kortejlarni fragmentlarga taqsimlashdan iborat. Taqsimlash biror predikat asosida amalga oshiriladi

Misol.

LOYIXA (Kod, Nom, Tur, Narx).

Predikatlar

{ Narx < 300000. Narx = 300000. Narx > 300000 }

Kesmalarga ega bo'lmagan to'liq taqsimlashni amalga oshiradi

Predikatlar

{ Narx < 300000. Narx > 200000 }

Kesmalarga ega to'liq taqsimlashni amalga oshiradi

Predikat

{ Narx < 200. Narx > 300 }

Kesmalarga ega bo'lmagan to'liq bo'lmagan taqsimlashni amalga oshiradi

Gorizontaal fragmentatsiya avfzalliklari:

- munosabat fragmentlarini parallel qayta ishlashga imkon beradi;
- kortejlarni eng ko'p murojaat qilinadigan tugunlarga joylashtirishga imkon beradi.

Vertikal fragmentatsiya. Vertikal fragmentatsiyada munosabatlar bir necha proektsiyalarga ajratiladi, ya'ni munosabat sxemasi bir necha ostki sxemalarga ajratiladi.

Tiklash uchun xar bir ostki sxema birlamchi kalitni o'z ichiga olishi lozim. Boshqa usulda xar bir fragmentga avtomatik kortej identifikatori qo'shiladi. Bu identifikator qiymati tizim tomonidan avtomatik o'rnatiladi.

Aralash fragmentatsiya. Aralash fragmentatsiya ketma ket vertikal va gorizontaal fragmentatsiyalarni qo'llashdan iborat.

Ma'lumotlarni tugunlarga taqsimlash.

F fragmentlarni S tarmoq Q tugunlari bo'yicha optimal taqsimlash usulini topish lozim.

Optimallikni aniqlaganda quyidagi parametrlarni xisobga olish lozim:

- ma'lumotlarni uzatish, saqlash va qayta ishlash narxi;
- vaqt xarakteristikalarini;
- samaradorlik;
- cheklanishlar masalan tarmoq tuguni sig'imi.

Fragmentlarni tugunlar bo'yicha taqsimlash uchun quyidagi ma'lumotlar lozim:

- ma'lumotlar bazasi xaqida va fragmentlar kattalikasi;
- ilovalar xaqida (joylashuvi, foydalanish chastotalari);
- tugunlar xaqidai (tugunlarda ma'lumotlar saqlash va qayta ishlash narxi);
- tarmoq xaqida (narxi va ma'lumotlarni uzatish vaqtlari).

Ma'lumotlarni fragmentatsiyasi va replikatsiyasi. Replikatsiya – bitta ma'lumot nusxalarini izlash tezligini oshirish va ishonchlilikni ta'minlash uchun turli tugunlarda saqlash mexanizmiga aytiladi. Munosabat yoki fragment replikatsiya qilingan deyiladi agar uning nusxalari bir yoki bir necha tugunlarda saqlansa. Nusxalarni replika deb ataladi. To'liq replikatsiyada nusxalar xamma tugunlarda saqlanadi. Butun ma'lumotlar bazasi xamma tugunlarda saqlanishi mumkn. Bu xolda ma'lumotlar bazasi to'liq replikatsiya qilingan deyiladi.

Replikatsiya avfzalliklari:

- xavfsizlik (R munosabat saqlanayotgan tugun ishdan chiqsa, boshqa tugunlarda murojaat qilish mumkin);
- parallelizm (R munosabatga nisbatan so'rovlar xamma replikalari bo'yicha parallellanishi mumkin);
- ma'lumotlarni uzatish narxini kamayishi (R munosabatga replikalari mavjud xamma tugunlarda murojaat qilish mumkin).

Replikatsiya kamchiliklari:

- ma'lumotlarni saqlash, yaratish va tiklash narxi oshadi;
- resurslarga talab oshadi;
- ma'lumotlarni yaxlitligini saqlash qiyinlashadi, masalan bir vaqtni o'zida bitta munosabat xamma replikalarini tiklash.

Replikatsiya mexanizmlari.

Replikatsiyani amalga oshirish uchun uchta server ishlatiladi: noshir, distribyutor va o'quvchi.

Noshir deb o'zidagi ma'lumotlarni boshqa serverlarga nusxalash uchun taqdim etuvchi serverga aytiladi. Nusxa tayyorlashdan tashqari noshir ma'lumotlar bazasiga kiritilgan o'zgarishlarni kuzatib boradi va yangi nusxa tayyorlaydi.

Distribyutor noshir tayyorlagan nashrlarni nusxalaydi va o'quvchilarga jo'natadi.

Distribyutor aloxida server yoki noshir yoki o'quvchi uchun ajratilgan server bo'lishi mumkin. Distribyutor funktsiyalari replikasiya usullariga bog'liq.

O'quvchi xamma nusxalarni qabul qiluvchi server. O'quvchi ma'lumotlar o'zgartirish mexanizmi farq qiladi.

Replikatsiya modellari

- momental suratlar replikasiyasi;
- tranzaksiyalar replikasiyasi.

Momental suratlar replikasiyasi eng sodda replikasiya bo'lib, momental surat deb ataluvchi ma'lumotlar to'liq nusxasi o'quvchilarga jo'natiladi. Tranzaksiyalar replikasiyasida ma'lumotlar bazasi tranzaksiyalar jurnalidan foydalaniladi. Tanlangan tranzaksiyalar bajarish ketma ketligi xaqidagi ma'lumot bilan birga distribyutor ma'lumotlar bazasiga nusxalanadi, so'ngra o'quvchi serverlarga jo'natiladi va ko'rsatilgan ketma ketlikda bajariladi.

Bu tarmoq yuklanishini kamaytiradi. O'zgarishlar soni kichik bo'lgan xajmi katta ma'lumotlar bazasida qo'llash tavsiya etiladi.

Replikatsiyalar topologiyasi.

Replikatsiyalar topologiyasi quyidagi turlari ajratiladi:

- replikasiya «bitta -ko'pga» bitta noshir va bir necha o'quvchilar;
- replikasiya «ko'p-bittaga» bir necha noshir va bitta o'quvchi;
- replikasiya «ko'p -ko'pga» bir necha noshir va bir necha o'quvchi.

KODD qoidalari

1983 yil oktyabrida E.F. Kodd ikki maqola chop ettirdi. Birinchi maqolada to'liq relyatsion ma'lumotlar bazasi qoniqtirishi lozim bo'lgan 12 qoida asoslandi. Ikkinchi maqolada dasturiy ta'minotlar shu qoidalarga mosligi taxlil etildi.

Qoida 1: Axborot qoidasi (The Information Rule):

Relyatsion bazada hamma ma'lumot mantiqiy darajada bir usulda taqdim etilishi lozim: jadvallardagi qiymatlar bilan.

Qoida 2: Ma'lumotlarga garantiyalangan murojaat (Guaranteed Access Rule):

Relyatsion ma'lumotlar bazasida har bir ma'lumot elementiga mantiqiy jadval nomi, birlamchi kalit qiymati va ustun nomi orqali murojaat qilish mumkin.

Qoida 3: NULL qiymatni sistematik qo'llash (Systematic Treatment of Null Values):

Har qanday ma'lum qiymatdan farqli NULL qiymatlar, hamma ma'lumotlar turlari uchun hamma amallar bajarilganda qo'llanishi lozim. Masalan, sonli qiymatlar uchun 0 qiymat yoki belgili qiymatlar uchun bo'sh satrdan farq qilishi lozim.

Qoida 4: Relyatsion model asosida dinamik aktiv lug'at (Active On-Line Catalog Based on the Relational Model):

Lug'at relyatsion jadval shaklida saqlanishi lozim va ma'lumotlar bazasini boshqarish tizimlari bu jadvalga foydalanuvchi ma'lumotlari saqlanuvchi jadvalga murojjaat qilish uchun qo'llanadigan standart vositalarni qo'llash imkonini berishi lozim.

Qoida 5: Til ostki to'plami to'liqligi (Comprehensive Data Sublanguage Rule):

Relyatsion ma'lumotlar bazasini boshqarish tizimlari juda bo'lmasa bitta quyidagi xossalarga ega tilni qo'llashi lozim:

(a) chiziqli sintaksisga ega,
(b) ham interaktiv, ham amaliy dasturlarda qo'llanishi mumkin,
(v) ma'lumotlar va tasvirlarni ta'riflash, ma'lumotlarni modifikatsiyalash (interaktiv yoki dasturiy), yaxlitlik cheklanishlarini ta'riflash, murojaatni boshqarish va tranzaksiya amallarini boshqarish (begin, commit va rollback).

Qoida 6: Tasvirlarni molifikatsiyalash qoidasi (View Updating Rule):

Nazariy jixatdan modifikatsiyalanuvchi tasvirlarni tizim ham modifikatsiyalashi lozim. Har bir modifikatsiyalanuvchi tasvirga relyatsion jadvallarga qo'llaniladigan amallar qo'llash imkoniyati mavjud bo'lishi lozim: tanlash, joylash, o'zgartirish va ma'lumotlarni o'chirish.

Qoida 7: Yuqori darajada joylash, yangilash va o'chirish (High-Level Insert, Update, and Delete):

Asosiy yoki hosila munosabat bilan bitta operand sifatida foydalanish faqat o'qish uchun emas joylash, yangilash va o'chirish amallariga ham qo'llanishi lozim. Joylash, yangilash va o'chirish

amallari relyatsion jadvallar faqat bitta emas bir nechta satrlariga qo'llanishi lozim.

Qoida 8: Ma'lumotlar fizik mustaqilligi (Physical Data Independence):

Ilovalar va terminal amallar ma'lumotlarni saqlash va murojaat usullariga bog'liq bo'lmasligi lozim.

Qoida 9: Ma'lumotlar mantiqiy mustaqilligi (Logical Data Independence):

Ilovalar va terminal amallar nazariy jixatdan ma'lumotlarni saqlovchi ixtiyoriy o'zgarishlarga bog'liq bo'lmasligi lozim. Agar normallashtirish jarayonida jadval ajratilsa ilovaga ta'sir etmasligi lozim.

Qoida 10: Yaxlitlik mustaqilligi (Integrity Independence):

Konkret ma'lumotlar bazasi yaxlitligi cheklanishlari relyatsion til ostki to'plamida tasvirlash imkoniyati mavjud bo'lishi va bu tasvir ilovalarda emas ma'lumotlar lug'atida saqlanishi lozim. Ma'lumotlar bilan ishlaydigan til kiritilayotgan ma'lumotlarni tekshirishi va ma'lumotlarning yaxlitligini saqlashi lozim.

Qoida 11: Tarqatish mustaqilligi (Distribution Independence):

Relyatsion ma'lumotlar bazasida taqsimlash mustaqildir. Ma'lumotlar bazasi tarmoqlangan ya'ni bir nechta kompyuterlar xotirasida saqlanishi mumkin, lekin bu ilova ishiga ta'sir qilmasligi lozim.

Qoida 12: Til darajalari muvofiqligi (The Nonsubversion Rule):

Agar ma'lumotlar bilan ishlash uchun quyi darajadagi til ishlatilsa, yuqori darajadagi til qo'llovchi xavfsizlik va yaxlitlik qoidalarini qo'llashi lozim. Quyi darajadagi til yordamida ma'lumotlar yaxlitligini buzish imkoniyati mavjud bo'lmasligi lozim.

3- jadval. Sotuvchilar jadvali

snum	sname	city	comm
1001	Peel	London	.12
1002	Serres	San Jose	.13
1004	Motika	London	.11
1007	Rifkin	Barcelona	.15
1003	Axelrod	New York	.10

4- jadval. Buyurtmalar jadvali

onum	amt	odate	cnum	snum
3001	18.69	10/03/1990	2008	1007
3003	767.19	10/03/1990	2001	1001
3002	1900.10	10/03/1990	2007	1004
3005	5160.45	10/03/1990	2003	1002
3006	1098.16	10/03/1990	2008	1007
3009	1713.23	10/04/1990	2002	1003
3007	75.75	10/04/1990	2004	1002
3008	4723.00	10/05/1990	2006	1001
3010	1309.95	10/06/1990	2004	1002
3011	9891.88	10/06/1990	2006	1001

5- jadval. Buyurtmachilar jadvali

cnum	cname	city	rating	snum
2001	Hoffman	London	100	1001
2002	Giovanni	Rome	200	1003
2003	Liu	San Jose	200	1002
2004	Grass	Berlin	300	1002
2006	Clemens	London	100	1001
2008	Cisneros	San Jose	300	1007
2007	Pereira	Rome	100	1004

Obyektlarni identifikatsiyalash uchun yoki barcha aytganda kompyuter xotirasida yozuvlarning o'rnini aniqlashda ma'lumot elementi ishlatiladi. Bu elementni biz **kalit** deb ataymiz. Agar kalit obyektini bir qiymatli identifikatsiyalasa, u **birlamchi kalit** deyiladi. Aks holda, **ikkilamchi kalit** deyiladi. Agar obyektlarni identifikatsiyalash uchun bir nechta atributlar talab etilsa, bunday kalit **turlangan kalit** deb ataladi. Agar A va V guruhdagi obyektlar berilgan bo'lsa, ular orasidagi quyidagi mosliklar yoki munosabatlarni o'rnatishimiz mumkin:

1. Birga – bir (1:1)munosabat. A va V obyektlar to'plami orasida 1:1munosabat o'rnatilgan deyiladi, agarda A obyektning har bir nusxasiga V obyektning bitta nusxasi mos kelsa, va aksincha, V obyektning har bir nusxasiga A obyektning bitta nusxasi mos kelsa.

2. Birga – ko'p (1:n) munosabat. A va V obyektlar to'plamida A obyektning har bir nusxasiga V obyektning bir nechta nusxasi mos kelsa, shu bilan birga V obyektning har bir nusxasiga A obyektning

bittadan ko'p bo'lmagan nusxasi mos kelsa shunday munosabat hosil bo'ladi.

3. Ko'pga – bir (n:1) munosabat A va V obyektlar to'plami orasida o'rnatilsa, unda A obyektning har bir nusxasiga V obyektning ko'pi bilan bitta nusxasi mos keladi. V obyektning nusxalari orasida shundaylari mavjudki, ularga A obyektning bir nechta nusxasi mos keladi.

4. Ko'pga – ko'p (m: n) munosabat. A va V obyektlar orasida shunday munosabat o'rnatilgan deyiladi, agarda A obyektning har bir nusxasiga V obyektini bir nechta nusxasi mos kelsa va aksincha.

Obyektlarni tahlil qilib bo'lingandan so'ng, shu obyektga qo'yiladigan boshlang'ich so'rovni ishlab chiqamiz. Avtovakzalni faoliyati iqtisodiy va texnik ko'rsatgichlar bilan bog'liq bo'lganligi uchun, yo'lovchilarga informatsion ma'lumot berganligi uchun yaratayotgan informatsion sistemamiz quyidagi so'rovlarga javob berishi kerak:

1. Har bir reys uchun nechta bilet sotilganligi va umumiy sotilgan biletlarni aniqlash;
2. Reysni nomeri bo'yicha reys haqida ma'lumotlar chiqarish;
3. Marshrutlar haqidagi zarur axborotlarni chiqarish;
4. Aniq reyslar uchun qapysi haydovchilar tayinlanganligi va ular haqida ma'lumotlar olish;
5. Avtobusni texnik xarakteristikalari haqidagi ma'lumotni olish.

Ko'rib chiqilayotgan predmet sohani obyektlari orasida quyidagi tipdagi bog'lanishlar mavjud:

- 1 1:1 – biletlar bilan reys obyektlari orasidagi bog'lanish (sotilgan);
 - 2 M:1 – marshrut va reys orasidagi bog'lanish (marshrut munosabati);
 - 3 1:M – marshrut va bekatlar orasidagi bog'lanish (bekatlar);
 - 4 1:N – reys va haydovchi orasidagi bog'lanish (tayinlash);
 - 5 M:N- haydovchi va avtobus modeli orasidagi bog'lanish (haydovchiga ruxsat berish);
 - 6 M:1 – avtobus va avtobus modeli orasidagi bog'lanish (toifa).
- Shunday qilib, ko'rilayotgan masalada asosan obyektlar aniqlanadi va ular orasidagi bog'lanish topiladi, sinflarga ajratiladi.

Nazorat savollari

1. Relyatsion ma'lumotlar modeli qanday farqlanadi?
2. Relyatsion ma'lumotlar bazasining asosiy tushunchalari.
3. Munosabat xossalriga nimalar kiradi?
4. Munosabatlar sxemasiga misollar keltiring.
5. Munosabat turlari nechta?
6. Relyatsion algebra amallarini sanab bering va misol keltiring.

2.3. Relyatsion algebra va relyatsion hisobot elementlari

Munosabat D_1, D_2, \dots, D_n domenlarda (ularning bari har xil bo'lishligi shart emas) sarlavha va tanadan iborat bo'ladi. **Munosabat darajasi** — bu uning atributlar soni.

Munosabat darajasi bitta bo'lsa *unar* deyiladi. Ikkita daraja bo'lsa — *binary* deyiladi. Uch daraja bo'lsa *ternar* va hokazo. Kardinal son yoki munosabat quvvat — bu uning kortejlar soni. Kardinal munosabatlar soni vaqt bo'yicha o'zgaradi.

Munosabat — bu to'plamlardir. To'plam esa ta'rif bo'yicha mos elementlarni o'ziga olmaydi. Aytaylik, R munosabat A_1, A_2, \dots, A_n atributlarga ega. Aytiladiki, R munosabat atributlar to'plami $K=(A_i, A_j, \dots, A_k)$ qachonki ikkita vaqtga bog'liq bo'lmagan shart bajarilsa kalit bo'lishi mumkin. Munosabatlar ustida amallar.

Munosabatlar ustida har xil amallar bajariladi. Relyatsion ma'lumotlar modelini xususiyatlaridan biri ma'lumotlarni qayta ishlashni relyatsion algebra operatorlari (amallari) yordamida amalga oshirishdir. Relyatsion algebra quyidagi 8 ta operator keng ishlatiladi. Ulardan 4tasi an'anaviy to'plamlar ustidagi amallar kiradi.

An'anaviy (qabul qilingan) amallarga quyidagilar kiradi.

1 Birlashtirish

2 Kesishuv

3 Ayirma

4 Dekart ko'paytma

Maxsus amallarga esa quyidagilari kiradi.

1 Tanlash (seleksiya)

2 Proeksiya

3 Qo'shish

4 Bo`lish

Munosabatlar ustida bajariladigan birlashtirish, kesishuv, ayiruv amallari operatorlarning tili yoki turi bo`yicha mosligini talab etadi. 2 ta munosabat tipi bo`yicha mos keladi, agarda ularda ekvivalent munosabat sxemasi bulib:

1) ulardagi har bir darajasi bir xil bo`lsa yoki ular bir xil atribut to`plamiga ega bo`lsa;

2) sxema atributlarini shunday tartiblash mumkinki, bir xil o`rinda turib solishtirilayotgan atributlari bir xil domenda aniqlangan bo`lishi kerak.

Har bir munosabatda kortejlar identifikator kalitiga ega bo`lishi kerak. Kalit quyidagi ikkita xossaga ega bo`lishi kerak:

- Kartej kalit qiymati bilan bir qiymatli ifodalanishi kerak;
- Kalitda ortiqchalik bo`lmasligi kerak.

Bu degani hech qanday atributni kalitdan olib tashlash mumkin emas.

Relyatsion MB da informatsiyalarni ortiqchaligini normallashtirish yo`li bilan kamaytiriladi. Jadvallar ustida har xil amallar bajarish mumkin. Bu amallarni tartiblab ishlab chiqqan odam Kodd. Amallarga quyidagilar kiradi:

- To`plamlar ustida birlashtirish, kesishuv, ayirma, dekart ko`paytma va bo`lish amallari kiradi.

- Maxsus relyatsion amallar, ularga: proeksiya, birlashtirish, ajratish (tanlab olish) amallari kiradi.

Munosabatlar ustida amalni bajarish uchun ishlatiladigan tillarni ikki sinfga ajratishimiz mumkin:

- c) Relyatsion algebra tillari;
- d) Relyatsion hisoblash tillari.

Munosabatlar o`z mazmuniga qarab ikki sinfga ajratiladi:

- c) Obyektli munosabatlar;
- d) Bog`lanuvchi munosabatlar;

Obyektli munosabatlarda obyektlar haqidagi munosabatlar saqlanadi. Masalan, talaba munosabati. Bog`lanish munosabatlarida asosan, obyektli munosabatlarning kalitlari saqlanadi. Kalit atributlari oddiy va murakkab bo`lishi mumkin. Agar kalit ikkita va undan ortiq atributdan tashkil topgan bo`lsa, murakkab hisoblanadi.

Relyatsion MBBT da ma'lumotlar bilan ishlash uchun bir qancha tillar yaratilgan. Ba'zi hollarda bu tillarni ma'lumotlarni qism tillari deb ataladi. MB bilan ishlovchilar bu tillarda avtomatlashtirishni 3 bosqichga bo'lishadi:

1) Eng pastki bosqich – kortej deb ataladi. Bunda dasturchi yozuvlar yoki kartijlar bilan ishlaydi.

2) Relyatsion algebra deyiladi. Bunda foydalanuvchi munosabatlar ustida yuqori bosqichli amallar to'plamini kiritadi.

3) Eng yuqori bosqich – hisoblash bosqichi.

Bunda foydalanuvchi bevosita kompyuterga maxsus tillarda murojaat qiladi va mashina bu murojaatni qabul qiladi. Relyatsion algebra amallarini operandlari sifatida doimiy yoki o'zgarmas va o'zgaruvchan munosabatlar ishlatiladi. Relyatsion algebrada 5ta amal ishlatiladi:

1) Birlashtirish , R va S munosabatlarni birlashtirish $R \cup S$ ko'rinishida berilib, bu amalning natijasi R munosabatga tegishli bo'lgan yoki S munosabatga tegishli bo'lgan yoki ikkalasiga ham tegishli bo'lgan kartejlar to'plamidir. Bu amallarni bajarayotganda bir xil tartibda bo'lishi kerak. Natijani tartibi ham operandlar tartibiga teng bo'ladi.

2) Ayirma R va S munosabatlarni ayirmasi $R - S$ ko'rinishida yoziladi va undagi kortejlar to'plami R munosabatga tegishli, lekin S munosabatga tegishli bo'lmagan kortejlardir. Bu amalning bajarilishida ham operandlarni tartibi bir xil bulishi kerak.

3) Dekart ko'paytma. Bizda R va S munosabat berilgan bo'lsin. R munosabatni tartibi R va S munosabatniki S ga teng bo'lsin. Unda dekart ko'paytma $R * S$ ko'rinishida yozilib, uning natijasi uzunligi $R + q$ ga teng bo'lgan kortejlar to'plamidan iborat bo'lib, bu kortejlarni birinchi R komponentasi R kortejga teng bo'ladi, qolgan q komponentasi S kortejga teng bo'ladi.

4) Proeksiya, R munosabatga bu amal tadbiriq etilganda, R munosabatlardan Relyatsion MBBT da ma'lumotlar bilan ishlashda ishlatiladigan 2ta katta guruh tillari relyatsion hisoblash deyiladi.

Relyatsion hisoblash predikatlarini hisoblashga asoslangan bo'lib ifodalarni yozishga mo'ljallangan to'plamidan iboratdir. Ular yordamida biz mavjud munosabatlardan yangi munosabatlar yaratishni ta'minlaymiz. Bunday ifodalarni yozishda solishtirish

amallari, mantiqiy amallar va mavjudlik kvanteri va umumiylik kvanteri ishlatiladi.

Hozirgi paytda relyatsion MBBT ni taraqqiyotida yangitil QBE tili ishlatilmoqda.

Bu tilda relyatsion algebra va relyatsion hisoblashlarda koʻzda tutilmagan bir qancha imkoniyatlar kirgan. Bu tilni hususiyati shundan iboratki, u terminallarda ishlashga muljallangan. Soʻrovlarni yaratish uchun maxsus ekran redaktoridan, munosabat va redaktorlaridan foydalanamiz. QBE tilida foydalanuvchi oʻzi olishini moʻljallagan natijani soʻrov koʻrinishida tasvirlaydi va MBBT uni kerakli amallar ketma – ketligiga aylantirib beradi. Maʼlumot modelini rivojlanish konsepsiyasi 5 ta bosqichni koʻrsatishi mumkin:

1. 60- yillarning 2 – yarmida, bunda asosan ierarxik modellarga eʼtibor berilgan;
2. 70- yillarni 1 – yarmi, tarmoqli modellar;
3. 70- yillarning 2 – yarmi, relyatsion modellar;
4. 80- yillarning 1 – yarmi, semantik modellar;
5. 80- yillarning 2 – yarmi, obyektga moʻljallangan sistema.

Nazorat savollari

1. Relyatsion maʼlumotlar bazasini asosiy tushunchalari.
2. Munosabat xossalari qanday?
3. Munosabatlar sxemasiga misollar keltiring.
4. Relyatsion algebra amallarini aytib oʻting.
5. Relyatsion hisoblash amallarini ayting va misol keltiring.

2.4. Maʼlumotlar bazasini rejalashtirish, loyihalash va administratorlash

Maʼlumotlar bazasini rejalashtirish. Maʼlumotlar bazasi tizimini yaratish hayot siklini samarali tashkil etishga imkon beradigan boshqaruv faoliyati. Tashkillashtirishning axborot tizimi umumiyashtirilib integratsiyalanishi lozim. **Maʼlumotlar bazasini rejalashtirish quyidagi boshqaruv standartlarini tashkil etishni oʻz ichiga oladi:**

- maʼlumotlarni qay darajada toʻplanganligi;
- qay darajada alohida formatga ega ekanligi;
- qandaydir kerakli hujjatlarni toʻplanganligi;
- qay darajada loyihalash va amalga oshirishganligi.

Ma`lumotlarni loyihalash. Zarur ma`lumotlar bazasi tizimi uchun maqsadli vazifalar va korxonada topshiriqlarini qo`llab - quvvatlashda ma`lumotlar bazasini loyihalash yaratish jarayonidir.

Ma`lumotlar bazasini loyihalash quyidagi asosiy yondashuvlarni o`z ichiga oladi:

Yuqoridan pastga ostin-ustun ichkaridan tashqariga aralash

Ma`lumotlarni loyihalashning uch bosqichi:

Konseptual ma`lumotlar bazasini loyihalash
Logik ma`lumotlar bazasini loyihalash
Fizik ma`lumotlar bazasini loyihalash

Konseptual ma`lumotlar bazasini loyihalash.- Korxonadagi barcha fizik mulohazalarni ma`lumotlar bazasi modelini qurish jarayoni.

- Ma`lumot modeli bu foydalanuvchining alohida talablariga ma`lumotdan foydalanishni yaratish.

- Konseptual model logik loyihalash bosqichi uchun axborotlarning asosi hisoblanadi.

Logik ma`lumotlar bazasini loyihalash.Korxonada ma`lum bir ma`lumot modeli asosida ma`lumot ishlatilgan modelini qurish jarayoni, lekin muayyan MBBT va fizik bo`limlar bundan mustasno.

Konseptual model mantiqiy ma`lumotlar bazasi modeli uchun tartibga keltirilgan va tuzilgan model hisoblanadi.

Fizik ma`lumotlar bazasini loyihalash.Ikkilamchi saqlash bo`yicha ma`lumotlar bazasi amalga oshirishning tavsifini ishlab chiqarish jarayoni.Ma`lumotga samarali murojaatni tashkil qilishda bazalar o`zaro aloqasi fayl tuzilmalaridan foydalanishni tasvirlaydi. Bundan tashqari barcha o`zaro aloqadorlik chegaralari va xavfsizlik choralarini tasvirlaydi.

Maxsus MBBTga mo`ljallangan.

Ma`lumotlarni boshqarish(administratorlash)

- Rivojlantirish va shakllantirish standartlari rivojlantirish siyosati va prosedura
- Ma`lumotlar bazasini konseptual va mantiqiy loyihalash.
- Ma`lumotlar bazasini fizik loyihalash.

- Xavfsizlik va ma`lumotlar butunligini nazorat qilish.
- Jarayonni tekshirib borish.
- Ma`lumotlar bazasini sozlash.

Ma`lumotlar bazasini administratorlash.Ma`lumot bazasi administartori asosiy vazifasi ma`lumotlar bazasini ishchi holatini ta`minlash va uni unumdorligini oshirish. Ishchi holatini ta`minlashda asosiy ishlaridan biri bu ma`lumot bazasini installyatsiya qilish. Bu masalani yechishda parametrlarni to`g`ri tanlash, axborotlarni saqlash vositalarini konfiguratsiyalash, ma`lumotlar bazasini strukturasi aniqlash va ma`lumotlarni saqlash uchun soha ajratish.

Administrator shuningdek bazaga qo`shimchalar va yangilashlarni (tiklashlarni) o`z vaqtida o`rnatishga javob beradi. Administratorni muhim funksiyalaridan biriga ma`lumot bazasini himoyasini ta`minlash ham kiradi. U har doim ma`lumotlarni rezerv nusxalarini yaratish va uni ishonchli yerda saqlashi lozim.

Xavfsizlik.Tizimni administratorlashni muhim omillaridan biri ma`lumotlarni xavfsizligini ta`minlash mexanizmidir. Buning uchun administrator ma`lumotlar bazasi foydalanuvchilarini yaratadi va ularni aniq obyektlar ustida aniq amallar bajarish imtiyozlarini boshqaradi. Foydalanuvchilarni aniqlash va ularga murojatni cheklash ORACLE tizimida operatsion tizim darajasida va MBBT darajasida hal qilinadi. Ikkala bosqich uchun ham harakterli standart yondoshish, administrotor tomonidan foydalanuvchilarni ro`yxatdan o`tkazish va imtiyozlar berish bilan belgilanadi.

Imtiyoz tizim tomonidan ta`minlanuvchi qandaydir belgi bo`lib, u foydalanuvchiga aniq amallar bajara olish huquqini belgilaydi. Imtiyozlar (privilege) tizimda har xil amallar bajarishga ruxsatdir. MB serveri har bir foydalanuvchiga imtiyozlar majmuasini beradi.

Foydalanuvchilarni identifikatsiyalash.RACLE har bir foydalanuvchisi maxsus identifikatorga ega bo`lishi kerak: nomga yoki kirish nuqtasiga. Yangi identifikatorni yaratish administrator yoki shunday huquqqa ega bo`lgan foydalanuvchi tomonidan amalga oshirilishi mumkin. Bu CREATE USER komandasi bilan amalga oshiriladi. Yangi yaratilgan foydalanuvchi tizim obyektlari bilan amal

bajarishga hech qanday huquqqa ega emas. Tizimda muvaffaqiyatli roy`xatdan o`tish uchun, administrator foydalanuvchiga seansni tashkil qilish uchun imtiyozlar berishi kerak. Buning uchun GRANT CREATE SESSION TO foydalanuvchi_nomi, komandasi ishlatiladi. Oracle'da foydalanuvchini ro`yxatdan o`tkazish va imtiyozlar berish

```
CREATE USER Baza IDENTIFIED BY 123456;
```

Foydalanuvchi yaratish.

Foydalanuvchi nomi: Baza. Foydalanuvchi paroli: 123456.

Foydalanuvchi nomi bilan tizimga kirish: Connect Baza/123456;

Foydalanuvchiga imtiyozlar berish GRANT CREATE SESSION TO Baza;

Grant succeeded. Imtiyozlar. Imtiyozlar ikkita sinfga bo`linadi:

-tizimli imtiyozlar (system privilege);

-obyektlarga murojat imtiyozlari(object privilege);

Imtiyozlar MBda foydalanuvchiga biron bir amalni bajarish huquqini beradigan imtiyozlardir.

Obyektlarga murojat imtiyozlari – alohida aniq obyekt ustida amalalrni bajarish uchun foydalanuvchiga beriladigan imtiyozlar. Masalan birorta jadvaldan tanlashlarni bajarish. Foydalanuvchiga tizimli imtiyozlar berish uchun GRANT komandasi ishlatiladi.

GRANT komandasi. ORACLE da tizimli imtiyozlar beradigan operator quyidagi sistaksisga ega: GRANT <tizimli imtiyozlar> TO <foydalanuvchi nomi> Masalan: Faraz qilaylik, baza nomli foydalanuvchiga CREATE TABLE tizimli imtiyoz berilgan bo`lsin Tab1 jadvalni Baza1 nomli foydalanuvchi tizimiga yaratishga uning tizim tomonidan inkor qilinadi. CREATE ANY TABLE imtiyozi berilsa, bu muvaffaqiyatli bajariladi. Bu holatni quyidagi so`rovda ko`rishimiz mumkin.

Nazorat savollari

1. Relyatsion ma`lumotlar bazasini rejalashtirish tushunchalari.
2. Ma`lumotlar bazasini hayot siklini keltiring.
3. Ma`lumotga samarali murojaatni tashkil qilishga misollar keltiring.
4. Ma`lumotlar bazasini administratorlash usullari.

5. Ma'lumotlar bazasida aloqadorlik chegaralari tarifini keltiring.

2.5. Ma'lumotlar bazasini normallashtirish: 1NF, 2NF, 3NF va Kodd normal formalari

Normallashtirish deganda berilgan munosabatni bir necha marta oddiy va kichik munosabatlarga ajratish tushuniladi. Bu jarayonda mumkin bo'lgan barcha funksional bog'lanishlar aniqlanadi.

Misol. A va V atributlar berilgan bo'lsin. Agar ixtiyoriy vaqtda A atributni bittadan ortiq bo'lmagan qiymatimos kelsa, unda V atributda funksional bog'langan deyiladi va quyidagicha belgilanadi:

$A \rightarrow V$ Shaxsiy nomer \rightarrow FamiliyaMasabi \rightarrow Maosh
bog'lanishlar MB uzluksiz o'zgarib turadi. Unda yangi ma'lumot elementlari qo'shiladi. Ular orasida yangi aloqalar yoki bog'lanishlar o'rnatiladi va ularni qayta ishlashni yangi usullari qo'llaniladi. Bu jarayonda imkoni boricha foydalanuvchi yaratgan MB bilan ishlash uchun yaratilgan dastur ilovasini kam o'zgartirishga harakat qiladi.

Bu muammoni hal qilish uchun ma'lumot elementlarini asosli ravishda guruhlariga birlashtirish va ular uchun kalitlarni aniqlash yo'li bilan hal qilinishi mumkin. Hozirgi kunda axborot tizimlari ishlab chiqaruvchilar ma'lumotlarni 3 – normal formada tasvirlab ishlatishni taklif etadilar.

Funksional bog'lanish tushunchasi. Relyasion MB da ma'lumotlarni strukturasi tashqari ularni sxematik informatsiyasiga ham etibor beriladi. MB ni strukturasi haqidagi informatsiya munosabat sxemasi yordamida beriladi. Sxematik informatsiyalar esa atributlar orasidagi funksional bog'lanishlar orqali ifodalanadi. MB munosabatlarida atributlarni tarkibini quyidagi talablarga javobberadigan qilib guruhlash kerak:

- Atributlar orasidagi zaruriy bo'lmagan takrorlanishlar bo'lmasligi kerak.
- Atributlarni guruhlaganda ma'lumotlar takrorlanishi minimal darajada qilib ta'minlanishi kerak.

Bu bevosita ma'lumotlarni tez qayta ishlash imkonini beradi. Bo'nga normallashtirish jarayoni yordamida erishiladi.

Birinchi normal forma (1NF): *Munosabat yoki Jadval birinchi normal formada deyiladi, qachonki undagi hech bir qatorning istalgan maydonda bittadan ortiq qiymat joylashmagan bo'lsa.*

Boshqacha aytganda birinchi normal forma bu har bir satr va ustun kesishmasida bitta bo'linmas atomar qiymat joylashgan munosabat yoki jadvaldir.

Ikkinchi normal forma (2NF): *Munosabat yoki Jadval ikkinchi normal formada deyiladi, qachonki agar u birinchi normal formada bo'lsa va har bir birlamchi kalitga kirmagan atribut birlamchi kalit atributlariga to'liq funksional bog'liq bo'lsa.*

Agar ixtiyoriy vaqtda, bitta atribut har bir qiymatiga ikkinchi atributni bittadan ortiq bo'lmagan qiymati mos kelsa, ikkinchi atribut birinchi atributga funksional bog'liq deyiladi va quyidagicha belgilanadi:

$$A \rightarrow V$$

Uchinchi normal forma (3NF): *Munosabat yoki Jadval uchinchi normal formada deyiladi, qachonki u ikkinchi normal shaklda bo'lsa va tranzitiv bog'lanishlar bo'lmasa.*

Agar V atribut A atributga, S atribut esa V atributga bog'langan bo'lsa, ya'ni, $A \rightarrow V$ va $V \rightarrow S$ bo'lsa va $A \not\rightarrow V$, $V \not\rightarrow S$, $V \not\rightarrow A$ shartlar bajarilsa S atribut A atributga tranzitiv bog'langan deyiladi. Agar $C \rightarrow B$, bajarilsa qat'iy tranzitiv bog'lanish deyiladi.

Uchinchi normal shakl dekompozitsiya deyiladi. Agar hamma ostki sxemalar uchinchi normal shaklda bo'lsa va yo'qotishsiz ulash xossasiga ega bo'lsa.

Optimal uchinchi normal shakldagi dekompozitsiya agar ostki sxemalar soni minimal bo'lsa va $A \rightarrow V \rightarrow S$ tranzitiv bog'lanish dekompozitsiyasida ostki sxemalar A va S komponentalarini o'z ichiga olmasa.

Uchinchi normal shaklni Xit algoritmi modifikatsiyasi yordamida hosil qilish mumkin: Munosabat sxemasi $R = (U, F)$, $A, V, C \subset U$, berilgan bo'lsin va $A \rightarrow V \rightarrow S$ – tranzitiv bog'lanish (S birlamchi kalitga kirmagan atributlardan iborat), u holda dekompozitsiya $\rho = (R_1 = (U_1, F_1), R_2 = (U_2, F_2))$, bu yerda $U_1 = U \setminus S$, $U_2 = B \cup C$, F_1 iva F_2 – funksiyalar to'plami F mos holda U_1 va U_2 ga proektsiyalari.

Agar kerak bo'lsa dekompozitsiya $R1$ va $R2$ dekompozitsiyasi amalga oshiriladi va hokazo.

Munosabat sxemasi uchinchi normal shaklga tegishli ekanligini tekshirish NP -murakkab masala chunki birlamchi bo'lmagan kalitlarni ajratish bilan bog'liq. Umuman olganda $A \rightarrow V \rightarrow S$ munosabat lekompozitsiyasini birlamchi kalitlar S iga tegishlimi yoki yo'qmi tekshirmasdan amalga oshirish mumkin, lekin ortiqcha ostki sxemalar ko'payishiga olib kelishi mumkin, bu esa dekompozitsiya sifatini kamaytiradi.

Quyidagi algoritmlar tranzitiv bog'lanishlarni oshkor shaklda olishni talab qilmaydi lekin optimal bo'lmagan dekompozitsiyalarni beradi.

Boys-Kodd normal formasi: *Munosabat yoki Jadval Boysa-Kodd normal (BKNF) formasida bo'ladi, agar maydonlar orasidagi har qanday funksional bog'lanish to'liq funksional bog'lanishga ega bo'lsa.*

Munosabat sxemasi $R == (U, F)$ Boys – Kodd normal formada deyiladi. Agar ixtiyoriy atributlar to'plami $A \subset U$ uchun agar biror atribut

$x \in U \setminus A$ funksional A dan bog'liq bo'lsa hamma munosabat atributlari ham funksional A dan bog'liq bo'ladi.

Shunday qilib, R Boys – Kodd normal formasida faqat va faqat shu holda bo'ladi agar ixtiyoriy $(X \rightarrow Y)$ va $X \not\subseteq Y$ bog'lanish uchun $X_+ = U$ bajarilsa.

Boys Kodd normal formasi **kuchaytirilgan uchinchi normal forma** deb ham ataladi.

Agar munosabat Boys – Kodd formasida bo'lsa uchinchi normal formada ham bo'ladi, aksi har doim ham to'g'ri emas.

Quyidagi algoritm Xit teoremasiga asoslangan.

Munosabat sxemasi $R = (U, F)$.berilgan bo'lsin.

1. F elementar funksional bazis F^* , ga o'tish, ya'ni to'liq bog'lanishlardan iborat minimal yopilmaga o'tish. Har bir bog'lanish o'ng qismida faqat bitta atribut bo'lishi lozim.

2. Dekompozitsiya ρ munosabat R uchun iterativ usulda quriladi. Bunda ρ har doim yo'qotishsiz ulanish xossasiga ega bo'ladi.

Avval ρ faqat R dan iborat. Agar S – sxema ρ dan va S da $X \rightarrow Y, X \supseteq Y$ bog'lanish mavjud bo'lsa va X o'z ichiga S kalitini olmasa, S

ni dekompozitsiya $S1 = (U1, F1)$, $S2 = (U2, F2)$, bilan almashtiriladi, bunda $U1 = X \cup Y$, $U2 = U \setminus Y$ (bu yerda $S = (U, F)$). Toki ρ hamma ostki sxemalari Boys – Kodd formasiga kelmaguncha jarayon davom etadi. Punkt 1 bajarish shart emas lekin loyihalash qiyinligi oshadi.

Nazorat savollari

1. Munosabatlar nima maqsadda normallashtiriladi?
2. Munosabotlar atributlariga qanday talablar qo'yiladi?
3. Funktsional bog'lanish turlari ayting.
4. 1 NF va undagi shartlar qanday?
5. Qanday qilib 1 NFdan 2 NFga o'tiladi?
6. Qanday qilib 2 NFdan 3 NFga o'tiladi?

3-BOB. SQL TILI ORQALI MA'LUMOTLAR USTIDA AMALLAR BAJARISH

3.1. SQL tili va SQL operatorlarini yozish

SQL tili foydalanuvchi relyatsion ma'lumotlar bazasi bilan muloqat

qilishi uchun mo'ljallangan bo'lib, quyidagi 3 ta qismdan iborat:

DDL (Data Definition Language) – ma'lumotlarni aniqlash tili.

Ma'lumotlar bazasini (jadvallarini, indekslarini va x.k.) yaratish va

uning sxemasini taxrirlash uchun mo'ljallangan.

DCL (Data Control Language) – ma'lumotlarni boshqarish tili.

Foydalanuvchilarning ma'lumotlar bazasi obyektlariga murojatini

chegaralash operatorlaridan iborat.

DML (Data Manipulation Language) – ma'lumotlarni qayta ishlash tili.

Ma'lumotlar bazasi jadvallariga o'zgartirishlar kiritish uchun mo'ljallangan.

Bu tillar ma'lumotlarni yuqori darajali dasturlash qism tillari deyiladi, chunki ularni tarkibida barcha hisoblarni bajarish uchun zarur bo'ladigan til konstruksiyalari bo'lmaydi (shartli o'tish amallari, yoki sikl operatorlari). Ma'lumotlar bazasi bilan ishlovchi ixtiyoriy til foydalanuvchiga quyidagi imkoniyatlarni yaratishi lozim:

strukturasini to'la tavsiflagan holda ma'lumotlar bazasini va jadvallarini yaratish;

ma'lumotlar ustida manipulyatsiya amallarini bajarish, masalan,

jadvallardan ma'lumotlarni kiritish, taxrirlash va o'chirish;

oddiy va murakkab so'rovlarni bajarish.

SQL turlari yoki rejimlari. Yuqorida ta'kidlab o'tilganidek, SQL3 tili tarkibida hisoblash jarayonini boshqarish imkonini beruvchi IF ... THEN ...ELSE, GO TO, DO ... WHILE kabi buyruqlar mavjud emas. Bunday masalalar dasturiy yo'l bilan (dasturlash tili yoki masalalarni boshqarish tili) yoki interaktiv holda (foydalanuvchining so'rovlari asosida) amalga oshiriladi. Imkoniyati cheklanganligi

sababli (hisoblash jarayonini boshqarish imkoniyati) SQL tili 2 ta usulda qoʻllanilish mumkin. Birinchi usulda *interaktiv* ishlash nazarda tutiladi. Bunda foydalanuvchi SQL operatorlarini terminaldan beradi. Ikkinchi usulda protsedurali tildagi dasturga SQL tili operatorlari kiritiladi. Bu usullar baʼzi adabiyotlarda maʼlumotlar bazasi bilan ishlash texnologiyasi yoki rejimi yoki SQLturlari deb ataladi.

Interaktiv rejimda maʼlumotlar bazasi bilan ishlashda foydalanuvchi muloqot rejimida ishlaydi, yaʼni SQL tilidagi soʻrovni kiritadi va natijani oladi, yangi soʻrovni kiritadi va natijaga ega boʻladi va x.k. Interaktiv SQLda foydalanuvchi SQL- soʻrovlar va natija interaktiv rejimda olinadi.

Kiritilgan SQL rejimida boshqa dasturlash tillarida yaratigan dasturlardan foydalaniladi. SQL komandalari boshqa birorta tilga (S++S, Delphi) yozilgan dastur ichiga joylashtiriladi. Bu shunday tillarni ishlatadigan dasturlarni samarador, quvvatli qiladi. Ular relyatsion maʼlumotlar bazasi bilan ishlash imkonini beradi.

Maʼlumotlar tiplari. SQL tilida quyidagi asosiy maʼlumotlar 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)	- oʻnli son, p raqam va ishoradan iborat ($0 < p < 16$). Oʻnli nuqtadan soʻng raqamlar soni q orqali beriladi ($q < p$, agar $q = 0$ boʻlsa, tashlab yuborilishi mumkin).
FLOAT	- xaqiqiy son 15 ta qiymatli raqam va butun darajadan iborat. Daraja MBBT tipi bilan aniqlanadi (masalan, 75 yoki 307).
CHAR(n)	- uzunligi oʻzgarmas, n ga teng boʻlgan simvulli qator ($0 < n < 256$).
VARCHAR(n)	- uzunligi oʻzgaruvchi, 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 xaqiqiy 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.

CREATE

Bu buyruq SQL obyektining bir necha turlarini, shu bilan birga sxemalarni, domenlarni, jadvallar va tasavvurlarni yaratishi mumkin. CREATE SCHEMA (sxemani tuzish) operatori bilan sxemani tuzish, uning egasini aniqlash mumkin. Misol qilib bir operator ko‘rinishini ko‘rsatib o‘tamiz:

```
CREATE SCHEMA SALES  
AUTHORIZATION SALES_MGR  
DEFAULT CHARACTER SET ASCII_FULL ;
```

CREATE DOMAIN (domen tuzish) operatori yordamida ustunda mavjud bo‘lishi mumkin bo‘lgan ma‘lumotlarga cheklashlar qo‘yiladi. Bu cheklashlar orqali domenda mavjud bo‘lishi mumkin va mumkin bo‘lmagan obyektlar belgilanadi. Sxema o‘rnatilgandan keyin domenni tuzish mumkin.

Misol:

```
CREATE DOMAIN AGE AS  
INTEGER CHECK (AGE > 20);
```

Jadvallar CREATE TABLE operatori yordamida, tasavvur esa CREATE VIEW yordamida tuziladi. CREATE TABLE yordamida tuzilayotgan yangi jadval o'sha operatorning ustunida kerakli bo'lgan cheklashlar ham qo'yish mumkin. Ba'zi paytlarda cheklashlarni butun sxemaga ham qo'yish to'g'ri keladi. Bunday vaziyatlarda CREATE ASSERTION operator yordamida amalga oshiriladi. Bundan tshqari yana CREATE CHARACTER SET, CREATE COLLATION va CREATE TRANSLATION operatorlari mavjud.

SQL buyruqlari Relatsyon ma'lumotlar bazasi bilan o'zaro aloqada bo'lish uchun standart SQL buyruqlari

1. CREATE — yaratish,
2. SELECT — tanlash,
3. INSERT — kiritish,
4. UPDATE — yangilash,
5. DELETE – o'chirish va
6. DROP – butunlay o'chirib tashlash.

Ushbu buyruqlarni tabiatiga qarab quyidagi guruhlariga bo'lish mumkin

— *DDL — ma'lumotlarni aniqlash tili Nomer Buyruq va tavsif*

1 CREATE — YARATMOQ Ma'lumotlar bazasida yangi jadval, jadval ko'rinishi yoki boshqa obyekt yaratadi.

2 ALTER – o'zgartirish kiritish Jadval kabi mavjud ma'lumotlar bazasi obyektini o'zgartiradi.

3 DROP Butun jadvalni, jadval ko'rinishini yoki ma'lumotlar bazasidagi boshqa obyektlarni yo'q qiladi.

DML — Ma'lumotlar manipulyatsiyasi tili Nomer Buyruq va tavsif

1 SELECT — TANLASH. Bir yoki bir nechta jadvaldan ma'lum yozuvlarni olish.

2 INSERT — KIRITMOQ Yozuvni yaratadi.

3 UPDATE — YANGILASH Yozuvlarni o'zgartiradi.

4 DELETE — O'chirish Yozuvlarni yo'q qiladi.

DCL — Ma'lumotlarni boshqarish tili Nomer Buyruq va tavsif

1 GRANT Foydalanuvchi uchun imtiyoz beradi.

2 REVOKE — Qayta ko‘rib chiqing Foydalanuvchiga berilgan imtiyozlarni qaytarib oladi. To‘liq matnni bu yerdan o‘qing: <http://uzinfobiz.ru/sql-haqida-malumat.html>

SQL so‘rov tilini o‘rganishni boshlaymiz. Har bir maqolada ma'lum bir sql komandalarini yozib borishga harakat qilaman. Bu so‘rov tili bevosita ma'lumotlar bazasi(jadvallar)ga bog‘liq bo‘lgani uchun, maqola boshida biror jadvalni olib, shu jadvalga har xil so‘rovlar berib boraman.

Misollar, **oracle** ning **isqlplus** muhitida sinab boriladi va natijalar chiqariladi. Isqlplus dan boshqa muhitlarda ham bu komandalar ishlaydi, lekin ba'zi o‘zgarishlar bo‘lishi mumkin(sqlplus, mysql). Komandalarni, o‘rganishga qulay va soddadan qiyinga ko‘tarilib borish tartibida ko‘rsatib boraman. Iloji boricha to‘liq ma'lumot berishga harakat qilaman. Bu maqolada quyidagi jadvaldan foydalanamiz.

NAME	SURNAME	YEAR	NUM
David	Beckham	7	1
Eric	Contana	17	2
Wayne	Rooney	10	3
Rio	Ferdinand	4	4
Nemane	Vidic	6	5

Bu jadval "**misol**" deb nomlanib, 4 ta ustun va 6 ta qatordan iborat. YUqoridagi qatorda faqat qator nomlari aks ettirilgan, ular ma'lumot vazifasini bajarmaydi. 5 ta qatordan iborat ma'lumot, bizning jadvalimizda joylashgan va biz bu ma'lumotlarni sql so‘rov tili orqali har xil ko‘rinishda chiqarib olishimiz mumkin. Albatta haqiqiy ma'lumotlar bazasida bunday kam ma'lumotlar saqlanmaydi, biz misollarni shu kichik ma'lumotlarda bajaramiz.

Demak birinchi komanda bu – **SELECT**.

SELECT * FROM Misol;

Bu so‘rovning ma'nosi, "Misol" jadvalidagi barcha ma'lumotlarni chiqar deganidir. "*" barcha ustunlardagi ma'lumotlarni degani. "**SELECT**" – esa ekranga chiqar degani.

Natijani ko‘ramiz:

NAME	SURNAME	YEAR	NUM
David	Beckham	7	1
Eric	Contana	17	2
Wayne	Rooney	10	3
Rio	Ferdinand	4	4
Nemane	Vidic	6	5

Agar bizga barcha ustun ma'lumotlar emas, faqatgina ba'zi birlari kerak bo'lsa, kerakli ustun nomlarini vergul orqali nomma nom yozamiz.

SELECT name, surname, year FROM misol;

Natija:

NAME	SURNAME	YEAR
David	Beckham	7
Eric	Contana	17
Wayne	Rooney	10
Rio	Ferdinand	4
Nemane	Vidic	6

Ma'lumotlar bazasidan, ma'lumotlarni chiqarishda mantiqiy amallar orqali(+ , — , * , / ,()), ma'lumotlarga o'zgartirish kiritib, ekranga chiqarishimiz mumkin bo'ladi.

SELECT name, year+15, num*10 FROM misol;

Natijani ko'rsangiz, so'rov tili orqali, ma'lumotlarni o'zgartirganimizni ko'rishingiz mumkin bo'ladi.

NAME	YEAR+15	NUM*10
David	22	10
Eric	32	20
Wayne	25	30
Rio	19	40
Nemane	21	50

yoki bo'lmasa quyidagicha

SELECT year+10*num, name FROM misol;

YEAR+10*NUM	NAME
17	David
37	Eric
40	Wayne
44	Rio
56	Nemane

Agar bazadagi ustun nomlari sizga yoqmasa, siz ularni o‘zingiz hohlaganday nom bilan ekranga chiqarishingiz mumkin bo‘ladi, bunda ma'lumotlar bazasidagi ustun nomlari o‘zgarmaydi, faqat ekranda o‘zgargan ko‘rinishi chiqadi. Bunda bizga " as" kalit so‘zi yordam beradi.

SELECT name as ism, surname as familiya FROM misol;

ISM	FAMILIYA
David	Beckham
Eric	Contana
Wayne	Rooney
Rio	Ferdinand
Nemane	Vidic

Agar nomlarni qo‘shirnoq ichiga olsangiz, qo‘shirnoqdagi katta kichik harflar farqli bo‘lib qoladi.

**SELECT name as "IsM",
surname as "FamiliyA" FROM misol;**

IsM	FamiliyA
David	Beckham
Eric	Contana
Wayne	Rooney
Rio	Ferdinand
Nemane	Vidic

Agar 2 ta ustun natijalarini birlashtirib, ekranga chiqarmoqchi bo‘lsangiz "||" belgisidan foydalanishingiz mumkin bo‘ladi.

SELECT year||num, name FROM misol;

YEAR NUM	NAME
71	David
172	Eric
103	Wayne
44	Rio
65	Nemane

yoki

SELECT year||'OK'||num, name FROM misol;

YEAR 'OK' NUM	NAME
7OK1	David
17OK2	Eric
10OK3	Wayne
4OK4	Rio
6OK5	Nemane

Agar chiqarayotgan natijaning davomiga qandaydir belgilar(qo'shtirnoq, vergul, nuqta) chiqarmoqchi bo'lsangiz, maxsus "q" va qo'shtirnoq belgilari orasiga kerakli belgilarni yozishingiz mumkin bo'ladi.

SELECT name||q'[, so? it's]' FROM misol;

NAME Q',SO?IT'S'
David , so? it's
Eric , so? it's
Wayne , so? it's
Rio , so? it's
Nemane , so? it's

Nazorat savollari

1. DDL komandalarini ayting.
2. DML vazifasi va uning ishlatilishini ko'rsating.
3. SQLda toifa tushunchasi nima?
4. SQLda toifalar nima maqsadda qo'llaniladi?
5. SQL tilining ikkita asosiy komponentasi va ularning funksiyalarini ko'rsating.
6. **SELECT** operatori asosiy konstruksiyalari va ularga qo'yiladigan cheklanishlarni ayting.

3.2. Ma'lumotlar manipulyatsiya qilishda oddiy so'rovlar yaratish.

Ko'pgina amaliy masalarni yechishda ma'lum shartlar asosida axborotlarni ajratib olish talab etiladi.

Masalan: Student jadvalidan Petrov familiyali talabalarni chiqarish kerak.

SELECT surname, Name, from Student

WHERE surname='Petrov

WHERE shartida solishtirish amallari jumladan, =>, <, <=, >=, shuningdek, mantiqiy amallar «end» “ot”, “not” amallari ishlashi mumkin. Ular yordamida murakkab shartlar tuzunlikiladi.

Masalan: 3-kurs stipendiya oladigan talabalarni ismi familiyasini chiqaring.

**SELECT name, surname FROM Student
WHERE kurs=3 and stipend>0;**

Mantiqiy shartlarni berishda where parametri tarkibida IN, BETWEEN, LIKE, IS amallari ham ishlatiladi.

IN, NOT parametli ishlatilganda tekshirilayotgan maydon qiymati berilgan ro'yxat bilan solishtiriladi. Bu ro'yxat ni operator o'ng tomonidani () ichida yoziladi.

1 Exam baholari jadvaldan «4» va «5» baholi talabalar ro'yxatini chiqaring.

**SELECT *
FROM Exam_marks
WHERE mark IN (4,5);**

Birorta ham 4,5 olmagan talaba haqidagi ma'lumotlar olish uchun not in yoziladi. **Between** amali maydon qiymatini berilgan intervalga kirganligini tekshirish uchun ishlatiladi. 2: 30 va 40 soat doirasida o'qitiladigan fanlarni chiqarish so'rovini tuzing.

**SELECT*; from Subject
WHERE hour BETWEEN 30 AND 40;**

BETWEEN amali maydonla sonli va simvolli bluganda ham ishlatiladi.

LIKE amali f-t simvolli toifadagi maydonlar uchun ishlatiladi. Bu amal maydonni satrli matnlarni like dan so'ng ko'rsatilgan qism satr bilan solishtiradi.

3 studentlar jadvalida familiyasi «M» harfi bilan boshlanadigan talaba haqida ma'lumot chiqaring.

**SELECT* FROM Student
WHERE surname LIKE P%;**

%- foiz belgisi shu pozitsiyada ixtiyoriy simvollar ketma-ketligi kelishini anglatadi. Bundan tashqari «__» ma'nosi, shu pozitsiyada ixtiyoriy 1 ta simvol kelishini anglatadi.

Bu amallarni maydonda o`tkazib yuborilgan qiymatlarni yoki noaniq qiymatlarni topishda ishlatib bo`lmaydi. SQL tilida ma'lumotlarni almashtiruchi va kiritilgan familiyalar ishlatilishi mumkin. Ular ustun qiymatlari bilan ishlashi uchun yoki const sifati ifodalarda keladi. Const sifatida simvolli const, sonli constlarni ishlatish mumkin. Ular ustunlar ro`yxatiga kiritiladi va huddi virtual ustun kabi aniqlanadi. Agar so`rovda ustun o`rnida son kelsa, bu sonli const hisoblanadi. Simvolli const() ichida yoziladi.

Misol: quyidagi so`rov ushbu jadvalni chiqaradi.

```
SELECT 'familiya', surname, 'imya', 'name', 100  
FROM Student
```

Sonli ma'lumotlarni uzunlik o`zgartirish uchun atribut amallardan foydalanamiz. Bunda quyidagi amallar ishlatiladi: «-», «+», «*», «/»

Misol:

```
SELECT surname, name, stipend kurs, (stipend*kurs)/2;  
FROM Student  
WHERE kurs = 4 AND stipend >0
```

Joylashtirilgan so`rovlar

Misol. Joylashtirilgan so`rov:

```
SELECT name  
FROM users  
WHERE id_user IN  
(SELECT id_author  
FROM topics  
WHERE topic_name='велосипедлар');
```

Misol. Xuddi shu so`rov:

```
SELECT name  
FROM users  
WHERE id_user = (SELECT id_author  
FROM topics  
WHERE topic_name='велосипедлар');
```

Moslashtirilgan so`rovda tashqi so`rov xar bir satri uchun ostki so`rov bajariladi

Misol:

```

SELECT *
FROM Customers outer
WHERE 10/03/1990 IN ( SELECT odate
FROM Orders inner
WHERE outer.cnum = inner.cnum );

```

Shu misol:

```

SELECT *
FROM Customers first, Orders second
WHERE first.cnum = second.cnum
AND second.odate = 10/03/1990;

```

Misol. Tashqi so'rovdan foydalanish.

```

SELECT *
FROM Orders outer
WHERE amt > ( SELECT AVG amt
FROM Orders inner
WHERE inner.cnum = outer.cnum );

```

SQL da almashtirish funksiyalari bilan ishlash.

1. LOWER (<satr>)- berilgan satrni kichik harflarga almashtirib beradi.

2. UPPER (<satr>) – kichik harflarni kata harflarga almashtirib beradi.

3. INIT CAP (<satr>)- satrdagi har bir so'z unlikni 1-harfini bosh harf qilib beradi.

Maslan, ularga quyidagi misolni ko'ramiz.

```

SELECT LOWER (surname), UPPER (name)
FROM Student
WHERE kurs=4 AND stipend 20;

```

4. LPAD(<satr>,uzunlikunlik,[<qism satr>);

Berilgan uzunlikunlikdagi qism satrni chapdan, o'ngga joylashtiriladi. Agar qism satr ko'rsatilmagan bo'lsa, satr sukut bilan, probellar bilan to'ldiriladi. Agar uzunlik o'nlik satr unlikdan kerak bo'lsa berilgan satr ko'rsatilgan uzunlikunlikgacha qirqiladi.

Misol:

```

SELECT LPAD (surname, 10, D), LPAD (name, 10,8)
FROM Student
WHERE kurs=3 and stipend>0;

```

5 .LTRIM (<satr>), [<qism satr>]);

Bu funksiyalarni vazifasi mos ravishda chapdagi (o'ngday) chegaraviy simvol olib tashlashdan iborat. Olib tashlangan simollar qism satrda ko'rsatiladi.

Agar qism satr ishlamasa, probellari olib tashlanadi.

6.SUBSTR (<satr>, <boshlanish>, [<soni>]

Quyidagi bu funksiyalari satrdan berilgan pozitsiyadan boshlab berilgan sondagi simvollar ajratib olinadi. Agar soni ko'rsatilmagan bo'lsa satrni boshidan oxirigacha ajratib olinadi.

Misol:

SUBSTR (hurmatli do'stim: 10,6)=> do'stim

Misol:

SELECT SUBSTR (name, 1,1)

FROM Student

WHERE kurs IN (2,3,4) AND stipend>0;

7. **LENGTH (<satr>)** vazifasi satrni uzunlikunligini aniqlab borishdan iborat.

SELECT city, LENGTH (city)

FROM Student

WHERE kurs IN (2,3,4) AND stipend>0;

Guruhli (agregat) funksiyalar. Guruhli funksiyalar jadvaldan yig'ilgan axborotlarni olish uchun xizmat qiladi. Bu funksiyalari jadvaldagi satrlar guruhi bilan amal bajarib, 1 ta natija chiqaradi. Guruhli funksiyalar uchun quyidagi amallarni ishlatamiz.

1. **COUNT-** jadvaldagi satrlar sonini aniqlab beradi.

2. **SUM-** ko'rsatilgan maydon qiymatlarini yigindisini hisoblaydi.

3. **AVG-**tanlab olingan maydon qiymatlarini o'rta arifmetigini hisoblaydi.

4. **MAX (min)-**tanlab olingan maydon qiymatlarini eng kattasini (kichigini) topib beradi. **SELECT** so'rovida guruhli funksiyalar maydon nomlari kabi ishlatiladi.

Maydon nomlari funksiyalar argumentlari sifatida keladi.

Misol:Exam_marks jadvaldagi o'rta ball.

```
SELECT AVG (mark)  
FROM Exam_marks
```

Jadvaldagi satrlar (yoki yozuvlar) sonini hisoblash uchun quyidagi komandadan foydalanamiz.

```
SELECT COUNT (*)  
FROM Exam_marks
```

Misol:

```
SELECT COUNT (DISTINCT subj_ID)  
FROM Subject;
```

SELECT komandasida **GROUP BY** parametr ham ishlashi.

Bu paramet bir maydon o'xshash parametrlari (aniqlanayotgan qiymati) bo'yicha guruhlaydi va agregat funksiyalar ishlatilshi, ular shu guruhga bo'ladi.

Misol:

```
SELECT student_ID,MAX (mark)  
FROM Exam_marks  
GROUP BY student_ID
```

Guruhlashni bir nechta maydon bo'yicha ham bajarish mumkin.

```
SELECT student_ID, subject_ID, MAX (mark)  
FROM Exam_marks  
GROUP BY student_ID, subject_ID
```

Guruhlar ichidan kerakli yozuvlarni ajratib olish uchun *having* ishlatiladi.

```
SELECT subj_name, MAX (hour)  
FROM Subject  
GROUP BY subj_name HAVING MAX (hour)>= 34;
```

Ba'zi hollarda natija jadvalidagi ma'lumotlarni tartiblash talab etiladi. Buning uchun **ORDER BY** parametri ishlashi. Bu parametr ko'rsatilgan maydon barcha yozuvlarni o'sib borishi tartibida tartiblab beradi. **ORDER BY DESC** yozilsa, kamayishi tartibida yoziladi. **ORDER BY (ASC)** bo'lsa o'sish tartibida yoziladi.

Misol:

```
SELECT*
```

```
FROM Subject  
ORDER BY subj_name
```

Misol:

```
SELECT*  
FROM Subject  
ORDER BY subj_name DESC
```

Tartiblash bir nechta maydon barcha bajarilishi ham mumkin. Bunda avval tartiblash 1-maydon bo'yicha keyin 2-maydon bo'yicha bajariladi. Shuningdek, order by parametri group by parametri bilan birga ishlatilishi mumkin. Bunda order by so'rovda oxirida keladi va unda guruhni ichidagi yozuvlar tartiblaydi.

Misol:

```
SELECT *  
FROM Subject  
ORDER BY semester, subj_name;
```

ORDER BY semester SQL tili 1 ta so'rov ichiga 2-so'rovni joylashtirib ishlatish imkonini beradi.

Misol: birorta talabani familiyasi bo'yicha uning ID sini topish talab etilsa, va bu talabani barcha baholari haqidagi ma'lumotni ko'rmoqchi bo'lsak, quyidagi so'rovni yozish mumkin.

```
SELECT*  
FROM Exam_marks  
WHERE student_ID =(  
SELECT student_ID  
FROM student  
WHERE surname= 'Petrov'  
)
```

Tasvirlar

Tasvir. Tasvir bir yoki bir nechta doimiy jadvallar asosida yaratilgan virtual jadvaldir.

Tasvirlar quyidagi komanda orqali yaratiladi:

```
CREATE VIEW <tasvir_nomi>  
AS <so'rov>
```

Misol:

```
CREATE VIEW info_order  
AS SELECT onum, amt, cname  
FROM orders, customers  
WHERE orders.cnum = customers.cnum;
```

Tasvirlarga murojaat. Tasvirga SQL komandalarini masalan SELECT komandasini qo'llash mumkin.

Masalan:

```
SELECT * FROM info_order;
```

Tasvirga so'rov asosida yangi tasvir yaratilishi mumkin. Lekin bu imkoniyatdan amaliyotda kam foydalaniladi.

Joylashtirilgan so'rovli tasvir. Misol. Eng katta reytingga ega oluvchilarni ko'rsatuvchi Cust_Rating_Max nomli tasvir yarating:

```
CREATE VIEW Cust_Rating_Max  
AS SELECT cname, city, rating  
FROM Customers  
WHERE rating = (SELECT Max(rating)  
FROM Customers);
```

Guruhlanuvchi tasvir. Misol. Har bir sotuvchi uchun buyurtmalar o'rta narxini ko'rsatuvchi Sale_Amt_Avg nomli tasvir yarating:

```
CREATE VIEW Sale_Amt_Avg  
AS SELECT sname, AVG(amt)  
FROM Salespeople, Orders  
WHERE Salespeople.snum = Orders.snum  
GROUP BY sname;
```

Tartiblanuvchi tasvir. Hamma MBBT lar ham tartiblashli tasvirlarni qo'llamaydi.

Buyurtmalar haqida ma'lumotni buyurtma summasi bo'yicha tartiblab chiqaruvchi tasvir yarating.

```
CREATE VIEW Orders_All  
AS SELECT amt, odate, Customers.cname,  
Salespeople.sname  
FROM Orders, Customers, Salespeople  
WHERE Orders.cnum = Customers.cnum AND
```

**Orders.snum = Salespeople.snum
ORDER BY amt;**

Parametr WITH CHECK OPTION. CREATE VIEW parametri WITH CHECK OPTION vazifasi hamma UPDATE va INSERT komandalari tasvir ta'rifiga mos kelishini garantiya qilish. Agar shartga javob bermasa UPDATE yoki INSERT xato qaytaradi.

Misol:

**CREATE VIEW Customers_View AS
SELECT name, age
FROM Customers
WHERE age IS NOT NULL
WITH CHECK OPTION;**

Tasvirlarni o'zgartirish. Tasvirlarni o'zgartirish komandalari asosiy jadvallarni o'zgartiradi.

Misol. Mijoz uchun yoshini o'zgartirish.

**UPDATE Customers_View
SET AGE = 35
WHERE name = 'Ramesh';**

Bu CUSTOMERS jadvalini ham o'zgartiradi.

Tasvirdan satrlarni o'chirish

Misol. AGE = 22 yozuvni o'chirish.

**DELETE FROM Customers_View
WHERE age = 22;**

Natijada satr CUSTOMERS jadvalidan ham o'chiriladi.

Modifikatsiya qilinuvchi tasvirlar. Modifikatsiya qilinuvchi tasvir deb shunday tasvirga aytiladiki, agar tasvirdagi ma'lumotlar o'zgartirilsa, asosiy jadvallar ham o'zgaradi. Ya'ni tasvirga UPDATE/INSERT/DELETE operatorlar qo'llansa jadvallar ham o'zgaradi.

Hamma tasvirlar ham modifikatsiya qilinuvchi bo'la olmaydi. Modifikatsiya qilinmaydigan tasvirlar faqat *o'qish uchun tasvirlar* deyiladi.

Tasvir modifikatsiya qilinuvchi bo'lishi uchun quyidagi shartlar bajarilishi lozim:

1. SELECT shartida DISTINCT ishlatilmasligi lozim.
2. SELECT shartida agregat funksiyalar bo‘lmasligi lozim.
3. SELECT shartida berilgan funksiyalar bo‘lmasligi lozim.
4. SELECT shartida berilgan operatorlar bo‘lmasligi lozim.
5. SELECT shartida ORDER BY bo‘lmasligi lozim.
6. FROM shartida bir nechta jadval qatnashmasligi lozim.
7. WHERE shartida ostki so‘rovlar qatnashmasligi lozim.
8. So‘rovda GROUP BY va HAVING qatnashmasligi lozim.
9. Qayta ishlangan ustunlar yangilanmasligi lozim.
10. Hamma asosiy jadvaldagi NOT NULL ustunlar tasvirda qatashishi lozim INSERT ishlashi uchun.

Misollar. Faqat o‘qish uchun va modifikatsiyalanuvchi tasvirlarga misollar:

```
CREATE VIEW Dateorders (odate, ocount)
AS SELECT odate, COUNT (*)
FROM Orders
GROUP BY odate;
```

Bu faqat o‘qish uchun tasvir chunki agregat funksiya va GROUP BY mavjud.

```
CREATE VIEW Londoncust
AS SELECT *
FROM Customers
WHERE city = 'London';
```

Bu tasvir modifikatsiya qilinuvchi.

```
CREATE VIEW Sales (name, number, percentage)
AS SELECT sname, snum, comm*100
FROM Salespeople
WHERE city = 'SanJose';
```

Bu faqat o‘qish uchun, chunki "comm * 100" ifoda ishlatiladi.

```
CREATE VIEW Salesonthird
AS SELECT *
FROM Salespeople
WHERE snum IN (SELECT snum
FROM Orders
WHERE odate = 10/03/1990);
```


Bu ANSI bo'yicha faqat o'qish uchun so'rov chunki ostki so'rov mavjud. Lekin ba'zi dasturlarda modifikatsiya qilinuvchi bo'lishi mumkin.

```
CREATE VIEW Someorders  
AS SELECT snum, onum, cnum  
FROM Orders  
WHERE odate IN (10/03/1990,10/05/1990);
```

Bu modifikatsiya qilinuvchi tasvir.

Tasvirlarni o'chirish. Tasvirlarni o'chirish quyidagicha amalga oshiriladi:

```
DROP VIEW view_name;
```

Misol. CUSTOMERS jadvali asosida yaratilgan CUSTOMERS_VIEW tasvirni o'chirish.

```
DROP VIEW CUSTOMERS_VIEW;
```

Tasvirni o'zgartirish komandasi *ALTER VIEW*.

```
ALTER VIEW tasvir_nomi [(ustun_1, ustun_2, ....)]  
AS ifoda_SELECT
```

Masalan OrdersProductsCustomers tasvirni o'zgartirish:

```
ALTER VIEW OrdersProductsCustomers  
AS SELECT Orders.CreatedAt AS OrderDate,  
Customers.FirstName AS Customer,  
Products.ProductName AS Product,  
Products.Manufacturer AS Manufacturer  
FROM Orders  
INNER JOIN Products ON Orders.ProductId = Products.Id  
INNER JOIN Customers ON Orders.CustomerId =  
Customers.Id;
```

Nazorat savollari

1. SQLda almashtirish funksiyalari nima maqsadda ishlatiladi?
2. Guruhli funksiyalarga qanday funksiyalar kiradi?
3. Tasavvurlar (View)ning afzalliklari nimada?
4. Shartli so'rovlar qanday tashkil qilinadi?
5. Guruhli funksiya vazifalari?
6. Guruhli funksiya ko`rinshlari?

7. Murakkab so`rovlar yaratish qanday yaratiladi?

3.3. SQLtili yordamida ma'lumotlarni tavsiflash

Butunlik (inglizcha – teginmaslik, saqlanish, bir butun) – deganda har qanday vaqtda ma'lumotlarning to'g'riligi tushuniladi. Bu maqsad ma'lum chegarada bo'lishi kerak. MBBT ma'lumotlar bazasiga kiritilayotgan har bir bo'lak qiymatning to'g'riligini tekshirish imkonini bermaydi. Masalan, kiritilayotgan 5 (hafta kun nomerini ko'rsatuvchi) qiymati haqiqatda 3 ga teng bo'ishini tekshirmaydi. Boshqa tomondan 9 qiymati aniq xato bo'ladi va MBBT bo'nga javob qaytaradi. Chunki bu nomer (1,2,3,4,5,6,7) sonlar to'plami ichida yo'q. MBning butunligini ta'minlashni ma'lumotlarni har xil to'g'ri bolmagan o'zgarishlar yoki buzulishdan himoyalash deb qarash kerak. Zamonaviy MBBT butunlikni ta'minlash uchun bir qancha vositalarga ega: Butunlikni aniqlash usulini uchta guruhga ajratadi:

- Mohiyat bo'yicha butunlik;
- Murojaat bo'yicha butunlik;
- Foydalanuvchi aniqlaydigan butunlik.

Butunlikni aniqlash usullari:

1. Birinchi kalitda qatnashuvchi atributlarga aniqlanmagan qiymatlar qabul qilinishiga ruxsat etilmaydi.

2. Tashqi kalit qiymati quyidagilar biri bolishi kerak:

- Birinchi kalit qiymatiga teng;
- To'liq aniqlanmagan, ya'ni tashqi kalitda qatnashadigan har bir atribut qiymati aniqlanmagan bo'lishi kerak.

3. Har qanday aniq bir ma'lumotlar bazasi uchun qo'shimcha qoidalar spetsifikatsiyalari mavjud. Ular ishlab chiquvchilar yordamida aniqlanadi. Ko'p hollarda tekshiriladi:

- U yoki bu atributning unikalligi;
- Qiymatlar diapazoni;
- Qiymatlar to'plamining aloqadorligi.

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
);

Masalan: Student1 jadvalini quyidagi komanda bilan yaratish mumkin:

CREATE TABLE Student1
(Student_ID INTEGER,
SURNAME VARCHAR(60),
NAME VARCHAR(60),
STIPEND DOUBLE,
KURS INTEGER,
CITY VARCHAR(60),
BIRTHDAY DATE,
UNIV_ID INTEGER
);

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);

Jadvaldagi ma‘lumotlarni maydonlar bo‘yicha qidirish- tanlash amali yetarli darajada tezlatish uchun ma‘lumotlarni berilgan maydon bo‘yicha indeksatsiya qilish ishlatiladi. Indeksni bitta yoki bir nechta maydon bo‘yicha bajarish mumkin.

Indeks komandasini ko‘rinishi:

**CREATE INDEX < indeks nomi>
ON <jadval nomi>(<ustun nomi>[,<ustun nomi>]);**

Bu komanda bajarilishi uchun jadval yaratilgan bo‘lishi kerak va indeksda ko‘rsatilgan ustunlar unda bo‘lishi kerak. Masalan, Agar EHAM_MARKS jadvalidan talabani Student_ID maydoni qiymati bo‘yicha baxosini qidirish tez-tez talab etilsa, unda shu maydon bo‘yicha indeks bajariladi.

**CREATE INDEX Student_ID_1
ON EHAM_MARKS (Student_ID) ;**

Indeksni olib tashlash uchun (bunda uni nomini albatta bilish kerak) quyidagi komanda ishlatiladi.

DROP INDEX < INDES NOMI>;

Masalan:

DROP INDEX < Student_ID_1 >;

Mavjud jadval strukturasi va parametrlar uchun ALTER TABLE komandasi ishlatiladi. Masalan jadvalga ustunlar qo'shish ALTER TABLE komandasi quyidagicha bo'ladi.

ALTER TABLE < jadval nomi > ADD(<ustun nomi > <ma'lumot tipi> <o'lchami>);

Bu komanda bo'yicha mavjud jadval satrlariga yangi ustun qo'shiladi va o'nga qiymati yoziladi. Jadvalga bir nechta ustun ham qo'shsa bo'ladi. Ular bir-biridan vergul bilan ajratiladi.

ALTER TABLE < jadval nomi > MODIFY (<ustun nomi > <ma'lumot tipi> <o'lcham/aniqlik >);

Ustun harakteristikalarini modifikatsiyalashda quyidagi cheklanishlarni hisobga olish kerak :

- Ma'lumot toifasini o'zgartirishni, faqat ustun bo'sh bo'lsa bajarish mumkin.
- To'ldirilmagan ustun uchun o'lcham/ aniqlik uzunlikga kiritirish mumkin.
- To'ldirilgan ustun uchun o'lcham/ aniqlik faqat kattalashtirish mumkin.
- NOTNULL urnatilishi uchun ustunda birorta ham qiymat bo'lmasligi kerak.

Sukut bilan o'rnatilgan qiymatni har doim uzunlikgartirish mumkin. Jadvallarni olib tashlash quyidagi komanda bilan bajariladi:

DROP TABLE < jadval nomi >;

Mumkin bo'lgan ma'lumot qiymatlar cheklanishlar bo'lishi mumkin. Unda CREATE TABLE komandasi quyidagicha bo'ladi.

**CREATE TABLE < jadval nomi > (
<ustun nomi > ma'lumot toifasi > <usto'nga cheklanishlar>,
<ustun nomi> <ma'lumot toifasi > <usto'nga cheklanishlar>,
< jadvalga cheklanishlar> (<ustun nomi>[,< ustun nomi >])
);**

Masalan , qiymatni Student jadvalini aniqlashda jadvalidagi Student_ID , SURNAME ,NAME , maydonlarida ishlatishni taqiqlash uchun komanda quyidagicha bo‘ladi:

```
CREATE TABLE Student  
( Student_ID INTEGER NOT ,  
SURNAME CHAR (25) NOT NULL,  
NAME CHAR(10 ) NOT ,  
STIPEND INTEGER,  
KURS INTEGER,  
CITY CHAR(15),  
BITHDAY DATE,  
UNIV_ID INTEGER);
```

Ba‘zi hollarda birir maydonga kiritilayotgan barcha qiymatlar bir- biridan farq qilishi kerak. Bunda shu maydon uchun UNIQUE (yagona) so‘z ishlatiladi.

Masalan: Student jadvalida Student_ID qiymatlari farqli bo‘lishi uchun komanda quyidagicha bo‘ladi.

```
CREATE TABLE Student  
( Student_ID INTEGER NOT UNIQUE,  
SURNAME CHAR (25) NOT NULL,  
NAME CHAR(10 ) NOT ,  
STIPEND INTEGER,  
KURS INTEGER,  
CITY CHAR(15),  
BITHDAY DATE);
```

Jadvalda kalit maydonlarni ishlatish komandasi quyidagicha bo‘ladi

```
CREATE TABLE Student  
( Student_ID INTEGER PRIMER KEY ,  
SURNAME CHAR (25) NOT NULL,  
NAME CHAR(10 ) NOT ,  
STIPEND INTEGER,  
KURS INTEGER,  
CITY CHAR(15),  
BITHDAY DATE,  
UNIV_ID INTEGER);
```

SQL tilida jadvalga ma'lumotlar kiritish uchun, ular o'zgartirish va olib tashlash uchun ma'lumotlar bilan manipulyasiya qilish (DML) tilining uchta komandasi mo'ljallangan. Bular

- INSERT (qo'shish),
- UPDATE (tiklash yangilash),
- DELETE (olib tashlash, yo'qotish) komandalaridir.

INSERT komandasi (buyruqi) jadvalga yangi satr qo'shishni amalga oshiradi. Sodda holda uning ko'rinishi quyidagicha:

```
INSERT INTO <jadval nomi>  
VALUES (<qiymat>,<qiymat>);
```

Bunday yozuvda **VALUES** kalit so'zidan keyin qavs ichida ko'rsatilgan qiymatlar jadvaldagi yangi qo'shilgan satrning maydonlariga kiritiladi. Kiritish jadvalini **CREATE TABLE** operatori bilan yaratilish paytidagi ustunlarni ko'rsatish tartibida amalga oshiriladi. Masalan, Student jadvalida yangi satrni qo'shish quyidagicha amalga oshirish mumkin:

```
INSERT INTO Student  
VALUES (101, 'Ivanov', 'Aleksandr', 200,3,  
'Moskva', '6/10/1979', 15);
```

Agar birorta maydonga qiymati qo'shish zarur bo'lsa u oddiy qiymat kabi kiritiladi.

```
INSERT INTO Student  
VALUES (101, 'Ivanov', NULL, 200,3,  
'Moskva', '6/10/1979', 15);
```

Ba'zi hollarda maydonlarning qiymatini **CREATE TABLE** komandasida berilgan tartibdan boshqa tartibda kiritish zaruriyati paydo bo'lsa yoki qiymatlarni ba'zi bir ustunlarga kiritish talab etilmasa, **INSERT** komandasining quyidagi ko'rinishi ishlatiladi.

```
INSERT INTO Student (student_ID, city, surname,  
name) VALUES (101, 'Moskva', 'Ivanov', 'Casha');
```

Qavs ichidagi ro'yxatda nomi keltirilmagan ustunlarga avtomatik ravishda sukut bilan jadval tavsiflashda (**CREATE TABLE** komandasida) tayinlangan qiymat yoki qiymat tayinlanadi. **INSERT**

komandasi yordamida, bir jadvaldan qiymat tanlab olib uni boshqajadvalga joylashtirish mumkin. Masalan: quyidagi so‘rov

```
INSERT INTO Student1  
SELECT *  
FROM Student  
WHERE CITY='Moskva';
```

Bunda Student1 jadvali CREATE TABLE komandasi yordamida yaratilgan bo‘lishi kerak va Student jadvali strukturaga o‘xshash bo‘lishi kerak.

Jadvaldagi satrlarni olib tashlash **uchun DELETE** komandasi ishlatiladi. Quyidagi ifoda **Exam_marks1** jadvalidan barcha satrlarni olib tashlaydi.

```
DELETE FROM Exam_marks1;
```

Buning natijasida jadval bo‘sh bo‘lib qoladi(bundan so‘ng, uni **DROP TABLE** komandasi bilan olib tashlash mumkin). Jadval bir yo‘la birorta shartni qanoatlantiradigan bir nechta satrni olib tashlash uchun WHERE parametridan foydalanish mumkin.

Masalan:

```
DELETE FROM Exam_marks  
WHERE student_ID =103;
```

Satrlar guruhini olib tashlash ham mumkin :

```
DELETE FROM Student1  
WHERE CITY= 'Moskva';
```

UPDATE komandasi jadval satrlari yoki mavjud satrni ba‘zi bir yoki barcha maydonlarini qiymatini tiklash, ya‘ni o‘zgartirish imkonini beradi. Masalan: **University** jadvalidagi, barcha universitetlarini reytingini 200 qiymatga o‘zgartirish uchun quyidagi konstruksiyani ishlatish mumkin:

```
UPDATE University  
SET RATING=200;
```

Jadvaldagi maydon qiymatlarini o‘zgartirish kerak bo‘lgan aniq satrlarni ko‘rsatish uchun, UPDATE komandasi WHERE parametrida predikat ishlatish mumkin.

```
UPDATE University
```

```
SET RATING=200  
WHERE CITY= 'Moskva';
```

Bu so'rov bajarilganda, faqat Moskvada joylashgan universitetlarning reytingi o'zgartiriladi. UPDATE komandasi faqat 1ta ustun emas balki ustunlar to'plamini o'zgartirish imkonini beradi. Qiymatlari, modifikatsiya (takomillashtirilgan) qilinishi zarur bo'lgan aniq ustunlarni ko'rsatish uchun, SET parametri ishlatidi.

Masalan: o'qitilayotgan fan nomi "MATEMATIKA" (uning uchun **subj_ID =43**) "Oliy matematika" nomiga o'zgartirish talab etilsa va bunday indetifikatsion nomeri saqlab o'zgarish qoldirish kerak bo'lib, lekin shu bilan birga jadvaldagi mos satr maydonlariga o'qitiladigan fan haqida yangi ma'lumotlar kiritish uchun so'rov quyidagi ko'rinishda bo'ladi.

```
UPDATE Subject1  
SET subj_name= 'Visshaya matematika',  
hour=36, semestr=1  
WHERE subj_ID=43;
```

UPDATE komandasini SET parametrada skalyar ifodalarni ishlatish mumkin. Skalyar ifodada maydon sifatida o'zgartirilayotgan va boshqa maydonlar kiritilib, u maydon qiymatini o'zgarish usulini ko'rsatadi.

```
UPDATE University1  
SET rating=rating*2;
```

Masalan: Student1 jadvaldagi **stipend** maydon qiymatini Moskva shahri talablari uchun 2 marta oshirish uchun quyidagi so'rov ishlatiladi.

```
UPDATE Student1  
SET stipend = stipend *2  
WHERE city= 'Moskva';
```

SET predikat hisoblanmaydi, shuning uchun unda qiymatniko'rsatish mumkin.

```
UPDATE Student1  
SET stipend =NULL  
WHERE city= 'Moskva';
```

INSERTda qism so‘rovlarini ishlatish. INSERT operatorini qism so‘rovi bilan ishlatish 1 ta jadvalga birdaniga bir nechta satr yuklash imkonini beradi. VALUES ishlatuvchi INSERT operatori 1ta satr qo‘shsa INSERTli qism so‘rov jadvalga, qism so‘rov boshqa jadvaldan qancha satr ajratsa shuncha satr jadvalga qo‘shadi. Bu holda qism so‘rov bilan olinayotgan ustunlar soni va toifasi bo‘yicha, ma‘lumotlari qo‘shilayotgan jadvaldagi ustun soni va toifasiga mos kelishi kerak.

Masalan: faraz qilaylik Student1 jadvalini strukturasi(tuzunlikilmasi), Student jadval strukturasi to‘la mos bo‘lsin. Student jadvalidan Moskva shahri talabalari barchasi haqida yozuvlari bilan Student1 jadvalni to‘ldirish imkonini beradigan so‘rov ko‘rinishi quyidagicha bo‘ladi.

```
INSERT INTO Student1  
SELECT *  
FROM Student  
WHERE city= ‘Moskva’;
```

Student1 jadvaliga Moskvada o‘qiyotgan barcha talabalar haqidagi ma‘lumotlarni qo‘shish uchun WHERE parametrda mos qism so‘rov ishlatish mumkin.

Masalan:

```
INSERT INTO Student1  
SELECT *  
FROM Student  
WHERE univ_ID IN  
( SELECT univ_ID  
FROM University  
WHERE city = ‘Moskva’);
```

Nazorat savollari

1. SQL tilining jadval yaratish operatori sintaksisi qanday?
2. Indeks operatorini ko‘rinishi va uning vazifasi?
3. SQL tilining jadvalga yozish va takomillashtirish komandalarini tavsiflang.
4. Insert opretori haqoda ma`lumot bering.
5. IN predikati nima vazifa bajaradi?
6. Drop va Delete operatorlarining farqi nimada?

7. Update operatori sintaksisi qanday yoziladi?

3.4. SQLda jarayonlar va standart 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: sizni kompaniyangizda reja bajarilishi o'rtacha protsenti qancha?

```
SELECT AVG(100 * (SALES/QUOTA))  
FROM SALESREPS
```

Yana bir shakl: Sizni kompaniyangizda reja bajarilishi o'rtacha protsenti qancha?

```
SELECT AVG(100 * (SALES/QUOTA)) PERCENT  
FROM SALESREPS
```

Bu holda ustun nomi ma'noliroq, lekin bu asosiysi emas. Ustunlar summasini hisoblab ko'ramiz. **SUM()** funksiyasini qo'llaymiz, ustun sonli bo'lishi kerak.

Masalan, quyidagicha: kompaniya xizmatchilari sotuvlar hajmi rejadagi va haqiqiy o'rta qiymati qanchaga teng?

```
SELECT SUM(QUOTA), SUM(SALES)  
FROM SALESREPS
```

AVG() agregatlash funksiyasiga yana bir necha sodda misollarni ko‘ramiz.

Masalan: "ACI" ishlab chiqaruvchi mollari o‘rtacha narxini hisoblang.

```
SELECT AVG(PRICE)  
FROM PRODUCTS  
WHERE MFR_ID = 'ACI'
```

Ekstremumlarni topish funksiyalari yani MIN(), MAX() funksiyalarini ko‘ramiz. Bu funksiyalar sonli ustunlar, sanalar va satrli o‘zgaruvchilar bilan ishlaydi. Eng sodd qo‘llanishi sonlar bilan ishlash.

Masalan quyidagi so‘rov beramiz: eng ko‘p va kam sotuvlar rejadagi hajmi?

```
SELECT MIN(QUOTA), MAX(QUOTA)  
FROM SALESREPS
```

Bu sonlarni o‘z ichiga olgan ustunlardir. Yana bir so‘rov beramiz: bazadagi buyurtmalarning ichida eng oldin berilgan so‘rov sanasi?

```
SELECT MIN(ORDER_DATE)  
FROM ORDERS
```

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: Kompaniyamiz mijozlari soni nechta?

```
SELECT COUNT(CUST_NUM)  
FROM CUSTOMERS
```

Yana bir so‘rov: Qancha xizmatchi rejani ortig‘i bilan bajardi?

```
SELECT COUNT(NAME)  
FROM SALESREPS  
WHERE SALES > QUOTA
```

COUNT(*) funksiyasi qiymatlar sonini emas, satrlar sonini hisoblaydi. Quyidagicha yozish mumkin:

```
SELECT COUNT(*)  
FROM ORDERS  
WHERE AMOUNT > 250
```

NULL qiymat va agregat funksiyalar. Ustun qiymati NULL bo'lsa AVG(), MIN(), MAX(), SUM(), COUNT() funksiyalari qanday qiymat qaytaradi. ANSI/ISO qoidalariga ko'ra "agregat funksiyalar NULL qiymatni e'tiborga olmaydi".

Quyidagi so'rovni ko'ramiz:

```
SELECT COUNT(*), COUNT(SALES), COUNT(QUOTA)  
FROM SALESREPS
```

Jadval bitta lekin so'rovdagi qiymatlar har xil. Chunki QUOTA maydoni- NULL qiymatni o'z ichiga oladi. COUNT funksiyasi COUNT(maydon) ko'rinishda bo'lsa NULL qiymatni e'tiborga olmaydi, COUNT(*) bo'lsa satrlar umumiy sonini hisoblaydi. MIN(), MAX() funksiyalari ham NULL qiymatni e'tiborga olmaydi, lekin AVG(), SUM() - NULL qiymat mavjud bo'lsa chalkashtiradi.

Masalan, quyidagi so'rov:

```
SELECT SUM(SALES), SUM(QUOTA),  
(SUM(SALES) - SUM(QUOTA)), (SUM(SALES - QUOTA))  
FROM SALESREPS
```

(SUM(SALES)-SUM(QUOTA)) va (SUM(SALES-QUOTA)) ifodalari agar QUOTA, maydoni NULL qiymatga ega bo'lsa har xil qiymat qaytaradi. Ya'ni ifoda SUM(ustun qiymati - NULL). Yana NULL qaytaradi. Shunday qilib:

1. Agar ustundagi qiymatlardan biri NULL ga teng bo'lsa, funksiya natijasini hisoblashda ular tashlab yuboriladi.

2. Agar ustundagi hamma qiymatlar NULL ga teng bo'lsa, AVG(), SUM(), MIN(), MAX() funksiyalari NULL qaytaradi. Funksiya COUNT() nol qaytaradi.

3. Agar ustunda qiymatlar bo'lmasa (ya'ni ustun bo'sh), AVG(), SUM(), MIN(), MAX() funksiyalari NULL qaytaradi. Funksiya COUNT() nol qaytaradi.

4. Funksiya COUNT(*) satrlar sonini hisoblaydi va ustunda NULL qiymat bor yo'qligiga bog'liq emas. Agar ustunda satrlar bo'lmasa, bu funksiya nol qaytaradi.

5. DISTINCT funksiyasini agregat funksiyalar bilan birga ishlatish mumkin.

Masalan quyidagi so'rovlarda: kompaniyamizda qancha har xil raportlar nomlari mavjud?

```
SELECT COUNT(DISTINCT TITLE)  
FROM SALESREPS
```

DISTINCT va agregatlar ishlashda quyidagi qoidalar mavjud. Agar siz DISTINCT va agregat funksiyani ishlatsangiz uning argumenti faqat ustun nomi bo'lishi mumkin, ifoda argument bo'lolmaydi. MIN(), MAX() funksiyalarida DISTINCT ishlatish ma'nosi yo'q. COUNT() funksiyasida DISTINCT ishlatiladi, lekin kam hollarda COUNT(*) funksiyasiga umuman DISTINCT qo'llab bo'lmaydi, chunki u satrlar sonini hisoblaydi. Bita so'rovda DISTINCT faqat bir marta qo'llanishi mumkin. Agarda u agregat funksiya argumenti sifatida qo'llanilsa, boshqa argument bilan qo'llash mumkin emas.

Agregatlar va ma'lumotlarni guruhlash. Agregat funksiyalar jadval uchun natijaviy satr hosil qiladi.

Masalan: Buyurtma o'rtacha narxi qancha?

```
SELECT AVG(AMOUNT)  
FROM ORDERS
```

Masalan, oraliq natijani topish lozim bo'lsin. Bu holda guruhlanishli so'rov yordam beradi. Ya'ni SELECT operatorining GROUP BY ifodasi.

Avval GROUP BY ifodasi qatnashgan quyidagi so'rovni ko'ramiz: Har bir xizmatchi uchun buyurtma o'rtacha narxi qancha?

```
SELECT REP, AVG(AMOUNT)  
FROM ORDERS  
GROUP BY REP
```

REP maydoni bu holda guruhlash maydonidir, ya'ni REP maydonning hamma qiymatlari guruhlariga ajratiladi va har bir guruh

uchun AVG(AMOUNT) ifodasi hisoblanadi. Ya'ni quyidagilar bajariladi: so'rovlar har bir xizmatchaga bittadan guruhga ajratiladi. Har bir guruhda REP maydoni bir xil qiymatga ega. Har bir guruh uchun guruhga kiruvchi hamma satrlar bo'yicha AMOUNT ustuni o'rta qiymati hisoblanadi va bita natijaviy satr hosil qilinadi. Bu qator guruh uchun REP ustuni qiymati vash u guruh uchun so'rov o'rta qiymatini o'z ichiga oladi.

Shunday qilib, GROUP BY ifodasi qo'llanilgan so'rov, "GURUHLANISHLI SO'ROV " deb ataladi. Shu ifodadan keyin kelgan ustun "**guruhlash ustuni** " deyiladi. Yana bir necha guruhlanishli so'rovlarni ko'rib chiqamiz.

Har bir ofis uchun sotuvlarning rejalashtirilgan hajmi diapazoni qancha?

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

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

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

Yana bir guruhlanishli qiziqarli so'rov: Har bir xizmatchi nechta mijozga xizmat ko'rsatadi?

```
SELECT COUNT(DISTINCT CUST_NUM),
CUSTOMERS FOR SALESREPS', CUST_REP
FROM CUSTOMERS
GROUP BY CUST_REP
```

Bu yerda 'CUSTOMERS FOR SALESREPS' psevdomaydonning ishlatilishiga e'tibor bering. So'rov natijalarini bir nechta ustun bo'yicha guruhlash mumkin.

Masalan, quyidagicha:

Har bir xizmatchi uchun har bir klient bo'yicha buyurtmalar umumiy sonini hisoblash.

```
SELECT REP, CUST, SUM(AMOUNT)
FROM ORDERS
GROUP BY REP, CUST
```


Lekin ikki ustun bo'yicha guruhlashda natijalar ikki darajasiga ega guruhlar va ostki guruhlar yaratish mumkin emas. Lekin tartiblashni qo'llash mumkin. Shu bilan birga GROUP BY ishlatilganda so'rov natijalari avtomatik tartiblanadi. Quyidagi so'rovni ko'ramiz:

Har bir xizmatchi uchun har bir klient bo'yicha buyurtmalar umumiy sonini hisoblash; so'rov natijalarini klientlar va xizmatchilar bo'yicha tartiblash.

```
SELECT REP, CUST, SUM(AMOUNT)  
FROM ORDERS  
GROUP BY REP, CUST  
ORDER BY REP, CUST
```

Shunday qilib GROUP BY ifodasi SELECT ni guruhlarni qayta ishlashga majbur qiladi.

MS SQL serverida COMPUTE ifodasi mavjud bo'lib relyasion so'rovlar asoslariga zid keladi. Lekin uning yordamida saqlanuvchi protseduralardan foydalanmasdan sho'nga o'xshash natijalarni olish mumkin. Guruhlanishli so'rovlar uchun chegaralar mavjud. Satrlarni hisoblanuvchi ifoda asosida guruhlash mumkin emas. Qaytarilayotgan qiymatlar elementlariga ham chegaralar mavjud. Qaytariluvchi ustun bo'lishi umkin:

1. Konstantalar.
2. Guruhga kirgan hamma satrlar uchun bitta qiymat qaytaruvchi agregat funksiya.
3. Guruh hamma satrlarida bir xil qiymatga ega guruhlash ustuni.
4. Ko'rsatilgan elementlarni o'z ichiga oluvchi ifoda.

Odatda guruhlanishli so'rovlar qaytaruvchi ustunlarga guruhlash ustuni va agregat funksiya kiradi. Agar agregat ko'rsatilmasa GROUP BY dan foydalanmasdan DISTINCT ifodasidan foydalanish yetarli. Agar so'rovga guruhlash ustuni qo'shilmasa, u yoki bu satr qaysi guruhga tegishligini aniqlash mumkin emas. Shu kabi SQL92 guruhlanishli so'rovlarni tahlil qilishda birlamchi va ikkilamchi kalitlar haqidagi ma'lumot ishlatilmaydi.

Har bir xizmatchi uchun buyurtmalar umumiy sonini hisoblash::

```
SELECT EMPL_NUM, NAME, SUM(AMOUNT)
FROM ORDERS, SALESREPS
WHERE REP = EMPL_NUM
GROUP BY EMPL_NUM, NAME
```

Yana soddaroq shakl: har bir xizmatchi uchun buyurtmalar umumiy sonini hisoblash:

```
SELECT NAME, SUM(AMOUNT)
FROM ORDERS, SALESREPS
WHERE REP = EMPL_NUM
GROUP BY NAME
```

Agar guruhlash maydonlaridan birida NULL qiymat mavjud bo'lsa qaysi guruhga tegishli bo'ladi? WHERE ifodasida NULL va NULL tenglikka solishtirish natijasi yana NULL beradi. Shuning uchun ANSI/ISO standartida GROUP BY ifodasida NULL qiymatlar teng deb qabul qilingan.

Guruhlash va HAVING yordamida ajratish. Shart bo'yicha satrlarni ajratish uchun WHERE ifodasidan foydalangan edik. Shart bo'yicha guruhlarini 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 guruhlash sharti sifatida kelmoqda. Agar SUM(AMOUNT) > 300 sharti yolg'on bo'lsa, bu guruh natijaviy to'plamdan chiqariladi. Agar rost bo'lsa guruh natijaviy to'plamga kiradi.

Yana bir misol ko'raylik: ikki va undan ortiq xizmatchiga ega har bir ofisning hamma xizmatchilari uchun rejadagi va xaqiqiy sotuvlar umumiy hajmini 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. Yana shunga e’tibor berish kerakki HAVING ifodasida agregat funksiyalardan foydalaniladi. **OFFICES** va **SALESREPS** jadvallari xizmatchi yashaydigan gshaharni topish uchun qo‘shiladilar. Qo‘shilgan jadval satrlarlari ofislar bo‘yicha guruhlanadilar. Ikkidan kam satrga ega guruhlar tashlab yuboriladi. Ular HAVING ifodasi talabiga javob bermaydilar. Har bir guruh uchun xaqiqiy va rejadagi sotuvlar hajmlari hisoblanadi.

Murakkabroq misolni ko‘ramiz:

Har bir tovar nomi uchun narxi, ombordagi soni va buyurtma berilganlar umumiy sonini ko‘rsating, agar uning uchun buyurtma berilganlar umumiy soni ombordagi umumiy soni 75 foizidan ko‘p bo‘lsa.

```

SELECT DESCRIPTION, PRICE, QTY_ON_HAND,
SUM(QTY)
FROM PRODUCTS, ORDERS
WHERE MFR = MFR_ID
GROUP BY MFR_ID, PRODUCT_ID, DESCRIPTION,
PRICE, QTY_ON_HAND
HAVING SUM(QTY) > (0.75 * QTY_ON_HAND)
ORDER BY QTY_ON_HAND DESC

```

HAVING uchun o‘ng qo‘shimcha chegaralar mavjuddir. Bu ifoda juda bo‘lmasa bitta agregat funksiyani o‘z ichiga olishi kerak. Chunki WHERE alohida satrlarga HAVING satrlar guruhlariga qo‘llanadi. NULL qiymat uchun WHERE ifodasiga o‘xshab quyidagi qoida o‘rinli. Agar izlash sharti NULL qiymatga ega bo‘lsa satrlar guruhi tashlab yuboriladi. HAVING ifodasini GROUP BY siz qo‘llash mumkin. Bu holda natija hamma satrlardan iborat guruh deb qaraladi, lekin amalda bu kam qo‘llanadi.

Nazorat savollari

1. Agregat funksiyalar qo‘llanishiga misollar keltiring?
2. Guruhlash komandasi uchun so‘rov yozing.
3. Having bilan WHERE ni farqlarini keltiring.
4. Null qanday qiymat hisoblanadi?
5. Standart funksiyaga misollar keltiring?

3.5. Tranzaksiyalarni boshqarishda so‘rovlar yaratish va qayta ishlash.

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

Har bir tranzaksiyaning “faqat o‘qish” yoki “o‘qish va yozish” tartiblari mavjud. Tranzaksiya tartiblari SET TRANSACTION operatori yordamida o‘rnatiladi. Jimlik qoidasiga nisbatan "o‘qish va yozish” tartibi o‘rnatiladi. “Faqat o‘qish" tartibi doimo saqlanadigan bazaviy ma‘lumotlarga qo‘llaniladi. Har bir SQL tranzaksiyasi himoyalani darajasiga ega: READUNCOMMITTED, READCOMMITTED, REPEATABLE READ yoki SERIALIZABLE. SQL tranzaksiyasi himoyalani darajalari bajarilayotgan tranzaksiyaning boshqa parallel bajarilayotgan tranzaksiyalarga ta‘sir etish darajasini aniqlaydi. Tranzaksiyaning aniq darajasini o‘rnatish uchun SET TRANSACTION operatoridan foydalaniladi. Jimlik qoidasiga nisbatan SERIALIZABLE tartibi o‘rnatiladi. Himoyalani darajalari tranzaksiyalarning parallel bajarilishida yuz berishi mumkin bo‘lgan hodisalarni aniqlaydi.

Quyidagi ko‘rinishdagi hodisalar bo‘lishi mumkin:

1. PI (“Dirtyread” - "Yomon o‘qish"): T1 tranzaksiya qatori yaratadi. Keyin T2 tranzaksiya T1 COMMIT amalini bajarmasdan bu qatorni o‘qiydi. Shundan so‘ng T1 ROLLBACK amalini bajarsa, T2 tranzaksiya umuman mavjud bo‘lmagan qatorni o‘qigan bo‘lib chiqadi.

2. P2 ("Non-repeaiableread" - "Takrorlanmaydigan o'qish"): T1 tranzaksiya qatomi o'qiydi. Shundan so'ng T2 tranzaksiya bu buyruqlar qatorini o'zgartiradi yoki olib tashlaydi va COMMITni bajaradi. Shundan so'ng T1 shu qatomi yana o'qishga harakat qiladi, ammo bu qator birinchi holatdagi qator emas yoki olib tashlangani uchun topolmaydi.

3. P3 ("Phantom" - "Fantom"): T1 tranzaksiya biror-bir shartni qanoatlantiradigan N qatomi o'qiydi. Shundan so'ng T2 tranzaksiya bu qatorlar ichidan bir yoki bir nechta qator shartlarini generatsiya qiladi. Agar shu ishlardan keyin T1 o'qishni qaytarsa, u butunlay boshqa qatorlarga ega bo'ladi, MB bilan ish jarayonida ma'lumotlar butligi muhim o'rin tutadi. Ma'lumotlar butligi deganimizda, ma'lumotlarning to'g'riligi va mazmunan qarma-qarshi ma'noga ega emasligi tushuniladi. Masalan, "O'qituvchi" jadvalidagi har bir o'zgarish "Yuklama" jadvalida ham qayd etilishi kerak. O'qituvchining "Yuklama" jadvalida qayd etilmasligi ma'lumotlar butligining buzilishiga olib keladi. Ko'pchilik hollarada MBnin ma'lumotlari butligini saqlashni tashkillashtirish uchun tranzaksiyalardan foydalanishadi. Umuman olganda tranzaksiya - bu mantiqan bo'linmaydigan ish birligi. Bu jarayonda:

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

Tranzaksiyaning bu xususiyati butlik shartining buzilmasligini ta'minlaydi. Ko'pgina MBBTda tranzaksiyalarning ikkita modeli ishlatiladi:

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

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

- COMMIT yoriqnomasi bilan, bunda MBdagi o'zgarishlar doimiy bajariladigan bo'ladi va yangi tranzaksiya COMMIT buyryg'idan so'ng boshlanadi.

- ROLLBACK yoriqnomasi bilan, bunda tranzaksiyada bajarilgan barcha o'zgarishlar bekor bo'ladi va yangi tranzaksiya ROLLBACK buyryg'idan so'ng boshlanadi. Tranzaksiyalarning bajarilishini boshqarish modeli SUBD Sybase dasturida qo'llanlib, quyidagi yoriqnomalardan foydalaniladi:

1. BEGIN TRANSACTION yoriqnomasi, tranzaksiyaning boshlanishini bildiradi.

2. COMMIT TRANSACTION yo`riqnomasi, tranzaksiyaning muvaffaqiyatli tuganini bildiradi. Bunda yangi tranzaksiya avtomatik ravishda ishga tushmaydi.

3. SAVE TRANSACTION yo`riqnomasi, bunda tranzaksiya ichida saqlash nuqtasi tashkillashtiradi va saqlangan saqlash nuqtasiga nom berish imkoniyati yaratiladi. ROLLBACK yoriqnomasi, bunda barcha tranzaksiyadagi amallar bekor qilinadi va MB holati tranzaksiyadan oldingi holatga qaytariladi. Shunday qilib, **tranzaksiya** - bu MBga tugallangan murojaat bo`lib, quyidagi to`rtta shartning bajarilishini kafolatlaydi:

- Bo`linmaslik (atomarlik) - tranzaksiya boshi va oxiriga ega bo`lgan bolinmas blok. Bu blok yoki to`liqligicha bajariladi, yoki umuman bajarilmaydi;

- Kelishuvchanlik - tranzaksiya tugaganidan so`ng, hamma ob`yektlar kelishganlik holatini saqlab qoladi;

- Himoyalanganlik - har bir tranzaksiya jarayoni boshqa tranzaksiya ishiga ta`sir ko`rsatmaydi.

- Doimiylik - tranzaksiya jarayonida bajarilgan barcha o`zgarishlar doimiylik harakterga ega. SQL tilida tranzaksiyalarni maxsus operatorlar yordamida boshqarish imkoniyati mavjud. Shulardan biri tranzaksiya parametrlarini o`rnatish operatori bo`lib, uni yozilish formati quyidagicha:

```
<set transaction statement> ::=
SET TRANSACTION transaction mode>
[ { <comma> transaction mode> }... ]
transaction mode> ::=
<isolation level>
| <transaction access mode>
| «diagnostics size>
«isolation level> ::=
ISOLATION LEVEL «level of isolation>
<level of isolation> ::=
READ UNCOMMITTED | READ COMMITTED |
REPEATABLE
READ | SERIALIZABLE
transaction access mode>
```

READ ONLY | READ WRITE

diagnostics size> ::=

DIAGNOSTICS SIZE <number of conditions>

<number of conditions> ::= <simple value specification>

Bu yerda:

1. Agar himoya darajalari ko'rsatilmasa, himoya darajasi SERIALIZABLE deb tushuniladi.

2. Agar ruxsat tartibi READWRITE kalit so'zi bilan belgilansa, unda himoyalaniş darajasi READUNCOMMITTED bo'lmasligi kerak.

3 Agar ruxsat tartibi va himoyalaniş darajasi READUNCOMMITTED deb ko'rsatilsa, unda beriladigan ruxsat tartibi READONLY deb tushuniladi, aks hollarda ruxsat tartibi READWRITE bo'ladi. Ko'pchilik hollarda tranzaksiyalarning bajarilish jarayonida MB jadvallari strkturasi buzilishining oldini olish uchun tranzaksiyalarga faqat o'qish tartibini o'natish mumkin. Buning uchun quyidagi operator ishlatiladi:

SET TRANSACTION READ ONLY;

Bu operator tranzaksiya jarayoni boshlanishidan oldin ko'rsatiladi.

Masalan,

EXEC SQL SET TRANSACTION READ ONLY;

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

SELECT a.snum, sname, onum, 'Highest on', odate

FROM Salespeople a.

Orders b

WHERE a.snum = b.snum

AND b.amt = (SELECT MAX (amt)

FROM Orders c

WHERE c.odate = b,odate)

UNION

SELECT a.snum, sname, onum, 'Lowest on', odate

FROM Salespeople a,

Orders b WHERE a.snum = b.snum

AND b.amt = (SELECT MIN (amt)

FROM Orders c

WHERE c.odate ; b.odate);

Shu paytgacha UNION kalit so‘zi yordamida birlashtirilgan so‘rov natijalari qanday tartibda matnga chiqarilishi to‘g‘risida gapirmadik. Birlashtirilgan natijalarni ORDER BY kalit so‘zi yordamida tartiblashtirish mumkin.

Yoqoridagi misolni tartub raqamlariga nisbatan tartiblashni ko‘rib o‘tamiz.

SELECT a.snum, sname, onum, ‘Highest on’, odate

FROM Salespeople a.

Orders by WHERE a.snum = b.snum AND

b.amt = (SELECT MAX (amt)

FROM Orders c WHERE c.odate = b.odate)

UNION

SELECT a.snum, sname, onum. ‘Lowest on’, odate

FROM Salespeople a.

Orders by

WHERE a.snum = b.snu AND b.amt = (SELECT MIN (amt)

FROM Orders c WHERE c.odate = b.odate)

ORDER BY 3;

Arifmetik operatorlarning ustuvorligi matematikada bo‘lgani kabi bir xil. Agar kerak bo‘lsa, qavslar yordamida operatorlarni qo‘llash tartibi o‘zgartirilishi mumkin - $(a + b) * (x / (y-z))$. Va yana bir bor takrorlayman, NULL bilan har qanday operatsiya NULL beradi, masalan: $10 + \text{NULL}$, $\text{NULL} * 15/3$, $100 / \text{NULL}$ - bularning barchasi NULL ga olib keladi. I.e. shunchaki noma'lum qiymatni aytish aniq natijani berolmaydi. So‘rovni tuzishda buni hisobga oling va agar kerak bo‘lsa, NULL qiymatlarni ISNULL, COALESCE funksiyalari bilan ishleng:

SELECT

ID,Name,

Salary/100*BonusPercent AS Result1, NULL

Salary/100*ISNULL(BonusPercent,0) AS Result2, ISNULL

Salary/100*COALESCE(BonusPercent,0) AS

FROM Employees

ID	Name	Result1	Result2	Result3
1000	Иванов И.И.	2500	2500	2500
1001	Петров П.П.	225	225	225
1002	Сидоров С.С.	NULL	0	0
1003	Андреев А.А.	600	600	600
1004	Николаев Н.Н.	NULL	0	0
1005	Александров А.А.	NULL	0	0

COALESCE funktsiyasi haqida bir oz gaplashamin:

COALESCE (expr1, expr2, ..., exprn) - qiymatlar ro'yxatidan birinchi NULL bo'lmagan qiymatni qaytaradi.

Misol:

SELECT COALESCE (f1, f1 * f2, f2 * f3) val - bu holda uchinchi qiymat qaytariladi

FROM (SELECT null f1, 2 f2, 3 f3) q

Ko'pincha ma'lumotlar bilan ishlash ma'lum bir ketma-ketlikda bajarilishi kerak bo'lgan ko'rsatmalar to'plamini anglatadi. Masalan, mahsulot sotib olishni qo'shganda, ma'lumotlar jadvaliga ma'lumotlarni kiritish kerak. Biroq, bundan oldin, stokda sotib olingan mahsulot bor yoki yo'qligini tekshirish kerak. Bir qator qo'shimcha shartlarni tekshirishingiz kerak bo'lishi mumkin. Ya'ni, aslida, tovarlarni sotib olish jarayoni muayyan ketma-ketlikda bajarilishi kerak bo'lgan bir nechta harakatlarni o'z ichiga oladi. Va bu holda, ushbu harakatlarning barchasini bitta obyektga - saqlangan protsedura ichiga kiritish yanada maqbul bo'ladi. Ya'ni, mohiyatiga ko'ra, saqlanadigan protseduralar bir butun sifatida bajariladigan ko'rsatmalar to'plamini anglatadi. Shunday qilib, saqlanadigan protseduralar murakkab operatsiyalarni soddalashtiradi va ularni bitta obyektga o'tkazadi. Tovarlarni sotib olish jarayoni o'zgaradi, shuning uchun protsedura kodini o'zgartirish kifoya qiladi. Ya'ni, protsedura kodlarni boshqarishni ham soddalashtiradi. Shuningdek, saqlanadigan protseduralar jadvallarga ma'lumotlarga kirishni cheklash va shu bilan bog'liq ravishda ataylab qilingan yoki ongsiz istalmagan harakatlar ehtimolini kamaytirishga imkon beradi. Va yana bir muhim jihat - bu ishlash. Saqlanadigan protseduralar odatdagi SQL ko'rsatmalariga qaraganda tezroq bajariladi. Buning sababi, protsedura kodi birinchi marta ishga tushirilganda kompilyatsiya qilinadi va keyin kompilyatsiya qilingan shaklda saqlanadi. Saqlanadigan protsedurani yaratish uchun **CREATE PROCEDURE** yoki **CREATE PROC** buyrug'idan foydalaning.

Shunday qilib, saqlanadigan protsedura uchta asosiy xususiyatga ega: kodni soddalashtirish, xavfsizlik va ishlash.

Masalan, ma'lumotlar bazasida mahsulotlar to'g'risidagi ma'lumotlarni saqlaydigan jadval mavjud deylik.

```
CREATE TABLE Products
```

```
(  
  Id INT IDENTITY PRIMARY KEY,  
  ProductName NVARCHAR(30) NOT NULL,  
  Manufacturer NVARCHAR(20) NOT NULL,  
  ProductCount INT DEFAULT 0,  
  Price MONEY NOT NULL  
);
```

Ushbu jadvaldan ma'lumotlarni olish uchun saqlanadigan protsedurani yaratish:

```
USE productsdb;  
GO  
CREATE PROCEDURE ProductSummary AS  
SELECT ProductName AS Product, Manufacturer, Price  
FROM Products
```

CREATE PROCEDURE buyrug'i alohida paketda chaqirilishi kerakligi sababli, joriy ma'lumotlar bazasini o'rnatadigan USE buyrug'idan keyin yangi paketni aniqlash uchun GO buyrug'i ishlatiladi.

Protsedura nomini AS kalit so'zidan keyin olish kerak.

Jarayonning asosiy qismini skriptning qolgan qismidan ajratish uchun protsedura kodi ko'pincha BEGIN ... END blokiga joylashtiriladi:

```
USE productsdb;  
GO  
CREATE PROCEDURE ProductSummary AS  
BEGIN  
SELECT ProductName AS Product, Manufacturer, Price  
FROM Products  
END;
```

"Ilovalar tobora murakkablashayotganiga qaramay, ularni boshqarish va ishlab chiqish jarayonining qulayligini oshirish uchun biz ilovalar darajasini mavhumlashtira olamiz.

Bilmaydiganlar uchun trigger - siz jadvalga kiritgan qoidadir va DELETE, UPDATE yoki INSERT-ni bajarishda qo'shimcha harakatlar amalga oshiriladi. Masalan, biz o'zgartirishlar jurnalini yozib olamiz. Ammo ikkita alohida so'rovni (bitta ma'lumotni o'zgartirish uchun, ikkinchisi jurnalga yozish uchun) yozishning o'rniga, siz qoidani o'z ichiga olgan triggerni yozishingiz mumkin: "Qachonki satr o'zgartirilsa, boshqa jadvalda yangi satr yarating. o'zgarishlar » Ushbu yondashuv asosiy so'rovda biroz ko'proqlikni keltirib chiqaradi, ammo hozirda ikkita turli xil amallarni bajarish uchun ikkita turli xil paketlarni ma'lumotlar bazangiz serveriga o'tkazib bo'lmaydi, bu umuman ishlashni yaxshilaydi.

Triggerlar MySQL-da 5.0.2 versiyasidan beri joriy qilingan. Trigger sintaksisi biroz begona. MySQL protseduralar va boshqa funktsiyalar uchun ANSI SQL: 2003 standartidan foydalanadi. Agar siz dasturlash tillari bilan ishlasangiz, uni tushunish qiyin bo'lmaydi. Spetsifikatsiya bepul mavjud emas, shuning uchun biz oddiy tuzilmalardan foydalanishga va triggerda nima bo'lishini tushuntirishga harakat qilamiz. Xuddi shu tuzilmalar har qanday dasturlash tilidagi kabi qo'llaniladi.

Yuqorida aytib o'tilganidek, triggerlar UPDATE, DELETE va INSERT tadbirlari uchun protseduralar sifatida bajariladi. Ular voqea aniqlangandan oldin yoki keyin bajarilishi mumkin. Shu bilan siz DELETE yoki DELETE-dan keyin bajarilishi kerak bo'lgan triggerni aniqlay olasiz va hokazo. Bu shuni anglatadiki, sizda INSERT-dan oldin bajariladigan bitta trigger va juda kuchli vosita bo'lgan INSERT-dan keyin bajariladigan mutlaqo boshqa trigger bo'lishi mumkin.

POSITION funktsiyasi satr ichida pastki qatorni qidiradi, birinchi belgi o'rnini qaytaradi.

**SELECT POSITION(что_ищем IN поле) FROM fayl nomi
WHERE shart II**

Ushbu misolda, matn "long" so'zining o'rnini qidiradi:

SELECT *, POSITION('uzun' IN text) as pos FROM texts

id айди	text текст	pos позиция подстроки
1	Очень длинный текст!	7
2	Не очень длинный текст!	10
3	Просто текст!	0

Belgilarni raqamlash 1 dan boshlanadi.

Ushbu operator ma'lum bir shartning bajarilishiga, u yoki boshqa natijaga qarab, shartlarni tekshirish va qaytishga imkon beradi.

CASE bayonoti 2 shaklga ega:

Birinchi shakl: Ikkinchi shakl:

ISH

```
WHEN holat_1
THEN return_value_1
```

...

```
WHEN holat_N
THEN qaytish_val__N
[ELSE return_value]
```

END CASE tekshirildi_val

```
WHEN taqqoslash_qiymati_1
THEN return_value_1
```

...

```
WHEN solishtirish_val__N
THEN qaytish_val__N
[ELSE return_value]
```

Tugatish

Misol:

```
SELECT
```

```
ID,Name,Salary,
```

```
CASE
```

```
WHEN Salary>=3000 THEN '3П >= 3000'
```

```
WHEN Salary>=2000 THEN '2000 <= 3П < 3000'
```

```
ELSE '3П < 2000'
```

```
END SalaryTypeWithELSE,
```

CASE

```
WHEN Salary >= 3000 THEN 'ЗП >= 3000'  
WHEN Salary >= 2000 THEN '2000 <= ЗП < 3000'  
END SalaryTypeWithoutELSE  
FROM Employees
```

ID	Name	Salary	SalaryTypeWithELSE	SalaryTypeWithoutELSE
1000	Иванов И.И.	5000	ЗП >= 3000	ЗП >= 3000
1001	Петров П.П.	1500	ЗП < 2000	NULL
1002	Сидоров С.С.	2500	2000 <= ЗП < 3000	2000 <= ЗП < 3000
1003	Андреев А.А.	2000	2000 <= ЗП < 3000	2000 <= ЗП < 3000
1004	Николаев Н.Н.	1500	ЗП < 2000	NULL
1005	Александров А.А.	2000	2000 <= ЗП < 3000	2000 <= ЗП < 3000

Nazorat savollari

1. SQL muhitida tranzaksiyaning vazifasi nimadan iborat?
2. TCL uchun muhim jarayon qaysi?
3. Commit nima vazifani bajaradi? Misol keltiring.
4. Tranzaksiyalarni boshqarishni tushuntiring?
5. Rollback uchun misol keltiring?

4-BOB. MA'LUMOTLAR BAZASIGA MUROJAATNI AMALGA OSHIRISH

4.1. Ma'lumotlar bazasini administratorlash va xavfsizligini ta'minlash

Ishtirokchilar SQL Server resurslaridan foydalanish huquqiga ega bo'lgan individual foydalanuvchilar, guruhlar va jarayonlardir. Himoyalangan obyektlar server, ma'lumotlar bazasi va ma'lumotlar bazasida joylashgan obyektlardir. Ularning har birida SQL Server bilan aloqa qilish maydonini kamaytiradigan bir qator ruxsatnomalar mavjud. Quyidagi jadvalda ishtirokchilar va himoyalangan obyektlar haqida ma'lumotlar mavjud.

Shifrlash va sertifikatlar. Shifrlash kirishni boshqarish bilan bog'liq muammolarni hal qilmaydi. Shu bilan birga, u kirishni boshqarish nazorati chetlab o'tilgan juda kam holatlarda ham ma'lumotlarning yo'qolishini cheklash orqali xavfsizlikni yaxshilaydi. Masalan, agar ma'lumotlar bazasi o'rnatilgan asosiy kompyuter noto'g'ri tuzilgan bo'lsa va zararli foydalanuvchi maxfiy ma'lumotlarni (masalan, kredit karta raqamlari) olish imkoniga ega bo'lsa, o'g'irlangan ma'lumotlar, agar u ilgari shifrlangan bo'lsa foydasiz bo'ladi. Quyidagi jadvalda SQL Serverda shifrlash haqida qo'shimcha ma'lumotlar mavjud.

SQL serverida hisobni boshqarish. Access 2002 yordam dasturi sizga "Asboblar" menyusidagi "Ma'lumotlar bazasi xavfsizligi" buyrug'i yordamida "Access" loyihasida xavfsizlik ma'muriyatchiligi vazifalarini bajarishingiz mumkinligini aytadi. Ushbu buyruq faqat SQL Server Access loyihasi bilan bir xil kompyuterga o'rnatilgan bo'lsa mavjud bo'ladi. Ushbu vositadan foydalanib, SQL Serverida ro'yxatdan o'tish uchun foydalanuvchi qayd yozuvlarini, ma'lumotlar bazasi foydalanuvchi hisoblarini va rollarni qo'shishingiz, o'chirishingiz va o'zgartirishingiz mumkin.

Eslatma: MSDE2000 ma'lumotlar bazasi dvigatelining yangi versiyasi uchun Database Security menyusi va Replikatsiya menyusi mavjud emas. Access 2002 yordam dasturi shunday deydi, ammo biz SQL Server 2000-ga ulangan Access 2002 loyihasida ham xuddi shu kompyuterga o'rnatilgan nomlarni ololmadik. Ma'lumotlarni himoya qilish va SQL Serverga nusxalash uchun Transact-SQL yoki

Enterprise Manager ko'rsatmalaridan foydalaning. (Qo'shimcha ma'lumot olish uchun SQL Server yordamiga qarang.)

SQL Serverda ro'yxatdan o'tish uchun ikkita xavfsizlik tizimi mavjud:

O'rnatilgan SQL Server xavfsizligi. Serverda ro'yxatdan o'tish uchun siz foydalanuvchi nomi va parolni ko'rsatishingiz kerak.

NT foydalanuvchi hisoblaridan foydalangan holda Windows NT bilan o'rnatilgan xavfsizlik. Bunday holda, foydalanuvchi autentifikatsiyasi Windows NT asosida tarmoqqa ro'yxatdan o'tish paytida amalga oshiriladi. SQL Server Windows NT hisob qaydnomasini qabul qiladi va qo'shimcha foydalanuvchini ro'yxatdan o'tkazishni talab qilmaydi.

O'rnatilgan SQL Server xavfsizlik tizimidan foydalanish uchun:

Ushbu xususiyat sukut bo'yicha o'chirilganligi sababli, avval Ko'rish, Server xususiyatlari menyusining buyrug'ini va paydo bo'lgan dialog oynasida, tizim boshqaruvchisini (SA) yoqish katagiga belgi qo'ying. OK ni bosing.

File, Connection menyusidagi Data Link Properties menyusi buyrug'i bilan SQL Server Connection Properties oynasini oching.

Ulanish yorlig'ida ma'lum bir foydalanuvchi nomi va paroldan foydalanish katagiga belgi qo'ying.

Agar siz Access 2002 loyihasini har safar ochganingizda foydalanuvchi ro'yxatdan o'tish parolini kiritishingiz kerak bo'lsa, parolni saqlashga ruxsat berish katagidan tozalang. Agar serverda qayta ro'yxatdan o'tishda parolni kiritish shart bo'lmasa, ushbu katakchani belgilang.

Bunday xavfsizlik tizimi yordamida siz Access 2002 loyihasida ro'yxatdan o'tish uchun ishlatiladigan SQL Server qayd yozuvining parolini Access Tools, Security, Kirish parolini sozlash (Tools, Security, Login parolini o'rnatish) menyu buyrug'i yordamida o'zgartirishingiz mumkin. Loyihaga kirishni yo'qotmaslik uchun o'rnatilgan parolni unutmang.

SQL Serverni Windows NT bilan birlashtirilgan xavfsizlikdan foydalanish uchun:

SQL Server ulanish xususiyatlari oynasini oching Ma'lumotni bog'lash xususiyatlari.

Ulanish yorlig'ida Windows NT bilan o'rnatilgan xavfsizlikdan foydalanish (Windows NT Integrated Security-dan foydalaning) radio tugmachasini tanlang.

Bunday holda, SQL Serverda ro'yxatdan o'tishda parolni kiritish talab qilinmaydi, chunki Windows NT yoki Windows 2000-da **foydalanuvchi autentifikatsiyasi haqiqiy** bo'ladi. **MB administratori deyilganda** birorta shaxs yoki bir nechta shaxslardan iborat bo'lgan va MBni loyihalash, uzatish va samarador ishlashini ta'minlovchidir. Ma'lumotlar bazasi tushunchasi bilan ma'lumotlar banki tushunchasi ham mavjud (ishlatiladi). Ma'lumotlar banki (MBn) tushunchasi ikki xil talqin etiladi.

1. Hozirgi kunda ma'lumotlar markazlashmagan holda (ishchi o'rinlarda) Shu yordamida qayta ishlanadi. Ilgari ular alohida xonalarda joylashgan EHM larda (hisoblash markazlarida (HM)) markazlashgan holda qayta ishlangan. XM lariga axborotlar tashqi qurilmalar orqali kelib to'plangan. Ma'lumotlar bazasi markazlashgani hisobiga ularni **ma'lumotlar banki** deb atashgan. Bunda ma'lumotlarga murojat etish ishchi stansiyalardan markazlashgan holda tashkil etilgan va shuning uchun ma'lumotlar banki bilan ma'lumotlar bazasi tushunchalari o'rtasida farq qilingan. Hozirgi kunda ko'p hollarda ma'lumotlar bazasi markazlashmagan holda tashkil qilinmoqda. Shuning uchun ma'lumotlar banki va ma'lumotlar bazasi sinonim suzlar sifatida ham ishlatiladi.

2. Boshqacha talqinda, ma'lumotlar banki deyilganda ma'lumotlar bazasi uni boshqarish tizimi(MBBT) tushuniladi. Ma'lumot bazasi bilan ishlaydigan dasturni **ilova (prilojenie)** deb ataladi. Bitta ma'lumot bazasi bilan juda ko'p ishlashi mumkin.

MB ni ishlatish afzalliklari:

- Ixchamligi;
- Axborotlarni qayta ishlash tezligini oshishi;
- Kam mexnat sarfi;
- Har doim yangi axborot olish imkoniyati;
- Ma'lumotlarni ortiqchaligini kamayishi.

Ma'lumotlar bazasini loyihalashtirish. Ma'lumotlar bazasini (MB) ishlab chiqish (loyihalash)ning asosiy maqsadi uning mantiqiy tuzilishini aniqlashdan iboratdir. MBni ishlab chiqish predmet sohasini tavsiflash asosida amalga oshiriladi. Bu tavsiflash MBga kiruvchi hamma ma'lumotlarni o'z ichiga oluvchi hujjatlar majmuini

va predmet sohasini ifodalovchi obyekt va jarayonlar haqidagi boshqa ma'lumotlarni o'z ichiga oladi.

MBni yaratishni uni loyihalashdan boshlash lozim. Loyihalash natijasida relyasion bazaning tuzilishi, ya'ni relyasion jadvallar tarkibi, ularning tuzilishi va mantiqiy aloqadorligi aniqlanadi. Relyasion jadvalning tuzilishi esa uning ustunlari tarkibi, ularning ketma-ketligi, ustun ma'lumotlarining turi va o'lchami, shuningdek jadval kaliti bilan aniqlanadi.

Predmet sohasini tahlil qilish. Ixtiyoriy tipdagi MBni loyihalashtirishning birinchi bosqichi predmet sohasini aniqlash bo'lib, u axborot tuzilmasini (konseptual sxemalar) tuzish bilan yakunlanadi. Bu bosqichda foydalanuvchining so'rovlari tahlil qilinadi, axborot obyektlari va uning karakteristikalarini tanlanadi, hamda o'tkazilgan tahlil asosida predmet sohasi tuzilmalashtiriladi. Predmet sohasini tahlil qilish umumiy bosqich bo'lib, MB ishlashini amalga oshiradigan dasturiy va texnik vositalardan bog'liq emas.

Predmet sohasini tahlil qilishni uch pog'onaga bo'lish maqsadga muvofiq:

1. Konseptual talablar va axborot ehtiyojlarini tahlil qilish;

2. Axborot obyektlari va ular orasidagi aloqalarni aniqlash;

3. Predmet sohasining konseptual modelini qurish va MBni konseptual sxemasini loyihalashtirish. Konseptual talablar va axborot ehtiyojlarini tahlil qilishda quyidagi masalalarni hal qilish kerak:

· foydalanuvchilarning MBga bo'lgan talablarini tahlil qilish (konseptual talablar);

· MBdan o'rin olishi lozim bo'lgan axborotlarga ishlov berish bo'yicha mavjud masalalarini aniqlash (tadbikni tahlil qilish);

· kelajakda hal qilinishi lozim bo'lgan masalalarni aniqlash (perespektiv tadbiqu tahlil).

· tahlil natijalarini hujjatlashtirish.

Ishlab chiqarilayotgan MBga foydalanuvchilarning talablari so'rovlar bilan ularning intensivligi ko'rsatilgan ro'yxat va ma'lumotlarning hajmidan iborat. MBni ishlab chiqaruvchilar bu ma'lumotlarni uning bo'lajak foydalanuvchilari bilan suhbat o'tkazish natijasida aniqlaydilar. SHu erda axborotlarni kiritishga, yangilashga va o'zgartirishga bo'lgan talablar ham aniqlanadi. Mavjud va perespektiv tadbiqu tahlil qilish natijasida foydalanuvchilar talablari aniqlashtiriladi va to'ldiriladi.

Turli predmet sohaslarini tahlil qilishda so'rovnomaning taxminiy tarkibini quramiz:

1-misol. Institut talabalarini hisobga olish uchun MBni ishlab chiqish taklif qilinayapti. Predmet sohasining tahlil qilish:

1. Institutda qancha talaba ta'lim oladi?
2. Institutda qancha fakultet va kafedralar bor?
3. Fakultet bo'limlari va kurslari bo'yicha talabalar qanday taqsimlanagan?

4. O'rtacha bir kursda har bir mutaxassislik bo'yicha qancha fanlar o'qitiladi?

5. Institutda qancha o'qituvchi bor?

6. Boshqa shahardan kelgan qancha talaba yotoqxonada, xususiy xonadonlarda (ijarachi sifatida) yashaydi?

7. Ma'ruza va amaliy mashQulotlar o'tkazish uchun qancha auditoriyalar, hamda qancha laboratoriyalar bor?

8. Va boshqa.

Predmet sohasini tahlil qilishning ikkinchi poQonasi axborot obyektlarini tanlash, har bir obyekt uchun zarur xossalarini berish, obyektlar orasidagi aloqalarni aniqlash, axborot obyektlariga qo'yiladigan cheklashlarni aniqlash, axborot obyektlari orasidagi aloqalarning turlarini va axborot obyektlarining tasifnomalarini aniqlashdan iborat.

Masalan, axborot obyektlarini tanlayotganda quyidagi savollarga javob berish kerak bo'ladi.:

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

2. O'ar 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 obyektlarini aniqlash itaratsion jarayon. U axborotlar oqimining tahlili va iste'molchilar bilan suhbat o'tkazish asosida amalga oshiriladi. Axborot obyektlarning tavsifnomalari ham xuddi shu usullar bilan aniqlanadi.

Predmet sohasining axborot ehtiyojini va tizimdan foydalanuvchilarning axborot bilan ishlashdagi manfaatini hisobga olgan

holda predmet sohasini tuzilmalashtirish uchun konseptual model qo'llaniladi.

Har bir MB miqyosida konseptual talablar konseptual modelda umumlashtiriladi. Konseptual model abstrak vositalar yordamida quriladi va predmet sohasidagi hamma axborot obyektlarini qurish imkoniyatini beradi. Bunda predmet sohasini qanchalik keng, aniq va chuqurroq qurishimiz biz tanlagan modelga bog'liq bo'ladi. Minimal imkoniyatlarga ega bo'lgan model ma'lumotlar va ular orasidagi o'zaro aloqalarni berish imkoniyatini ta'minalashi zarur. Konseptual modelning semantik quvvati uning yordamida aniqlanishi mumkin bo'lgan qo'shimcha karakteristikalarining sonini ortishiga mos ravishda ortadi.

Ma'lumki, har bir model ma'lum bir cheklanishlarga ega bo'lib, o'zida faqat ma'lum bir xossalarni aks ettiradi. SHu sababli konseptual loyihalashtirish uchun model tanlashda, real dunyoni o'zida to'liq aks ettiradigan ideal modelni topish juda katta muammo ekanligini hisobga olish zarur. Avvalambor predmet sohasining xususiyatlari va MBga bo'lgan foydalanuvchining talablari modelni tanlash uchun asos bo'ladi. Boshqa muhim talab sifatida konseptual modelning MBBTdan bo'qliq emasligi, chunki u konseptual sxema qurilgandan so'ng tanlanishi kerak. Predmet sohasini tahlil qilishda qo'llaniladigan model xilma-xil bo'lishi mumkin.

Mantiqiy loyihalash. MB yaratishning eng zaruriy va mas'ulyatli bosqichlaridan biri - bu mantiqiy loyihalashtirishdir. Uning asosiy masalasi tanlangan MBBT uchun mo'ljallangan holda MB mantiqiy sxemasini ishlab chiqishdan iborat. Mantiqiy loyihalashtirish bosqichi konseptual loyihalashtirishdan farqli ravishda, u komp'yuterning dasturiy vositasini to'liq hisobga olgan holda amalga oshiriladi. Ish mazmuni bo'yicha mantiqiy loyihalashtirish axborot tizimini va uni tashkil etuvchi qismlarni real MBBTga mos shaklda modellashtirishdan iborat. Mantiqiy loyihalashtirish jarayoni quyidagi bosqichlardan iborat:

1. Aniq bir MBBTni tanlash
2. Konseptual sxemani mantiqiy sxemaga o'tkazish
3. Zarur kalitlarni tanlash
4. So'rov tilini tavsiflash

Aniq bir MBBTni tanlash protsedurasini batafsil qaraymiz. MB loyihagini amalga oshirish uchun MBBTni tanlash juda katta

mas'uliyat talab qiladi. Bu bir tomondan MBBTlarning juda ko'pligi bo'lsa, ikkinchi tomondan ko'p sonli karakteristikalar bo'yicha MBBTni baholash va ularning orasidan aynan shunday tizimni tanlash kerakki, u foydalaniuvchi va ishlab chiqaruvchilar talablarini to'liq qanoatlantirishi mumkin bo'lsin. Chunki MBda axborotdan foydalanish va ishlov berishi samaradorligi MBBTning qanchalik to'qri tanlashdan bo'liq bo'ladi.

MBBTni tanlashning asosiy me'yorlaridan biri-bu ma'lumotlarni ishlatadigan ichki modelining konseptual sxemasini tavsiflash uchun qanchalik samarador ekanligini baholashdan iborat. SHaxsiy komp'yuterlar uchun mo'ljallangan MBBTlarning ko'pchiligi, odatda ma'lumotlarning relyasion yoki tarmoq modeliga tayangan holda ishlaydi. Zamonaviy MBBTlarning juda katta qismi relyasion model asosida yaratilgan. Agar relyasion tizim tanlangan bo'lsa, u holda MBning konseptual sxemasini relyasionga akslantirish (o'tkazish) oldinda turibdi.

Ishning mazmuni bo'yicha ma'lumotning tanlangan modeli (relyasion, tarmoq va ierarxik) ma'lumotlar tizilmasini tavsiflash uchun vosita beradi. Protseduralar MBBT yadrosiga kiradigan ma'lumotlarni tavsiflash tilida bajariladi.

MBBTning ikkinchi tarkibiy qismi ma'lumotlarni manipulyasiya qilish tildan iborat. Undan MBni turli tadbirlar uchun ishlatishda foydalaniladi. Ko'p hollarda ma'lumotlarni manipulyasiya qilish tili (MMT) dasturlashtirish tilga o'rnatilgan (kiritilgan) bo'ladi. MMT turli imkoniyatlarga ega bo'lishi mumkin: quyi pog'onadagi til va yuqori pog'onadagi til. Odatda quyi pog'onadagi til protsedurali, yuqori pog'onadagisi esa deklarativ til bo'ladi. Protsedurali tillardan foydalanish ma'lum tayyorgarlikni talab qiladi, deklarativ til bo'lsa ko'prok professional bo'lmagan foydalanuvchilar uchun yarakli. Shuning uchun ma'lum MMTga ega MBBTni tanlash maxsus tayyorgarligi bo'lmagan foydalanuvchi uchun juda muhimdir. Bundan tashqari MBBTga servis dasturlar va amaliy masalalarni echish uchun vositalar kiradi.

Axborot obyektlari va ular orasidagi aloqalarni aniqlash. Predmet sohasini tahlil qilishning ikkinchi pog'onasi axborot obyektlarini tanlash, har bir obyekt uchun zarur xossalarini berish, obyektlar orasidagi aloqalarni aniqlash, axborot obyektlariga qo'yiladigan cheklashlarni aniqlash, axborot obyektlari orasidagi

aloqalarning turlarini va axborot obyektlarining tasifnomalarini aniqlashdan iborat.

Axborot obyektlarini tanlayotganda quyidagi savollarga javob berishga harakat qilamiz:

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

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

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

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

Axborot obyektlarini aniqlash -itaratsion jarayon. U axborotlar oqimining tahlili va iste'molchilar bilan suhbat o'tkazish asosida amalga oshiriladi. Axborot obyektlarning tasifnomalari ham xuddi shu usullar bilan aniqlanadi.

Keyin axborot obyektlari orasidagi aloqalarni aniqlaymiz. Bu jarayon borishida quyidagi savollarga javob berishga harakat qilamiz:

1.Axborot obyektlari 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? Obyektlarga, ularning tasiflariga va aloqalariga cheklanishlar berishga harakat qilamiz. Buning uchun quyidagi savollarga javob berish zarur:

1.Sonli tavsifnomalar uchun qiymatlarning o'zgarish sohasi qanday?

2.Bir axborot obyektning tavsiflari orasidagi qanday funksional bo'lanishlar bor?

3.Har bir tur aloqalarga qanday turdagi akslantirishlar mos keladi?

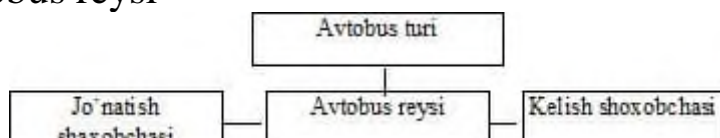
Axborot obyektlarining o'zaro aloqasiga misol sifatida TALABA, SINOV_Daftari, FAKULTET, O'QITUVCHI axborot obyektlarini qarash mumkin.



TALABA, SINOV_DAFTARI, FAKULTET,
O'QITUVCHI axborot obyektlari o'rtasidagi o'zaro aloqa

Axborot tuzilmalarini qurish. Premet sohasini tahlil qilishning oxirgi bosqichi uning axborot tuzilmalarini (yoki konseptual sxemalarni) loyihalashtirishdan iboratdir. Predmet sohasini tavsiflash yoki konseptual sxemalarni loyihalashtirishni, bu maqsadlar uchun maxsus yaratilgan modellardan foydalangan holda amalga oshirish mumkin.

Ko'p hollarda, konseptual sxemalarni qurish uchun ananaviy agregatsiya va umumlashtirish usullaridan foydalaniladi. Agregatsiya qilishda axborot obyektlari (ma'lumotlar elementlari) ularning orasidagi semantik aloqalarga mos ravishda bir obyektga birlashtiriladi. Masalan. "DEAWOO" turdagi avtobus yo'lovchilarni jo'nash punktidan kelish punktiga tashiydi. Agregatsiya usuli bilan quyidagi atributlarga ega bo'lgan REYS axborot obyektini hosil qilamiz. "avtobus turi; "jo'natish shaxobchasi", "kelish shaxobchasi", "avtobus reysi"



AVTOBUS REYSI axborot obyekti

Axborot obyektlarini (ma'lumotlar elementlarini) birlash-tirishda ularni sinfdosh obyektlarga birlashtiriladi.



Axborot obyektlarini sinfdosh obyektlarga birlashtirish.

Predmet sohasining axborot ehtiyojini va tizimdan foydalanuvchilarning axborot bilan ishlashdagi manfaatini hisobga olgan holda predmet sohasini tuzilmalashtirish uchun konseptual model qo'llaniladi. Har bir MB miqyosida konseptual talablar konseptual modelda umumlashtiriladi. Konseptual model abstrak vositalar

yordamida quriladi va predmet sohasidagi hamma axborot obyektlarini qurish imkoniyatini beradi. Obrazli aytganda konseptual model xuddi predmet sohasining “yuqorisiga ko‘tarilib” va uning har bir elementini qurish imkoniyatini beradi. Bunda predmet sohasini qanchalik keng, aniq va chuqurroq qurishimiz biz tanlagan modeldan bog‘liq bo‘ladi. Minimal imkoniyatlarga ega bo‘lgan model ma‘lumotlar va ular orasidagi o‘zaro aloqalarni berish imkoniyatini ta‘minlashi zarur. Konseptual modelning semantik quvvati uning yordamida aniqlanishi mumkin bo‘lgan qo‘shimcha karakteristikalarining sonini ortishiga mos ravishda ortadi. Ma‘lumki, har bir model ma‘lum bir cheklanishlarga ega bo‘lib, o‘zida faqat ma‘lum bir xossalarni aks ettiradi. SHu sababli konseptual loyihalashtirish uchun model tanlashda, real dunyoni o‘zida to‘liq aks ettiradigan ideal modelni topish juda katta muammo ekanligini hisobga olish zarur. Avvalambor predmet sohasining xususiyatlari va MBga bo‘lgan foydalanuvchining talablari modelni tanlash uchun asos bo‘ladi. Boshqa muhim talab sifatida konseptual modelning MBBTdan bo‘liqlik emasligi, chunki u konseptual sxema qurilgandan so‘ng tanlanishi kerak. MB tizimini qurish masalasi quyidagi hollarda paydo bo‘ladi:

1. Agar turli tadbirlar qandaydir miqdorda umumiy axborot obyektlarini talb qilsak, ammo hamma axborot obyektlari va ular orasidagi hamma aloqalarni bitta MBda amalga oshirib bo‘lmasa.

2. Agar axborot obyektlari umumiy aloqaga, munosabatga va atributlarning umumiy turiga ega bo‘lsa. MB tizimining axborot tuzilmasini ko‘rishga misol sifatida avtomobilsozlikda ilmiy-tadqiqot va tajriba konstruktorlik ishlarini axborot ta‘minoti tizimining MB uchun axborot tuzilmasini ko‘rish jarayonini qaraymiz.

Ushbu predmet sohasini tahlil qilish jarayonida quyidagi axborot obyektlari ajratildi:

1. Fizik effektlar. Ushbu predmet sohasidagi fizik effektlar texnik obyektlarning ishlash (harakatlanishi) tamoyili sifatida ko‘rsatiladi.

2. Texnik qarorlar. Bu qurilma, ishlash usuli yoki modda.

3. Mahsulot. Masalan, dvigatel.

4. Mahsulotni oluvchi obyekt. Masalan, dvigatel avtomashinaga o‘rnatiladi.

5. Usullar va uslubiyat.
6. Asboblar va stendlar.
8. Texnologik jarayonlar va uskunalalar.
9. Tashkil qilish va mutaxassislar.
10. Me'yorlar.

Bu obyektlar orasidagi aloqalar qarab chiqilib ma'lumotlarning aloqalarini aniqlovchi va tasvirlovchi jarayonlar yoki hodisalar ajratiladi.

Nazorat savollari

1. Ma'lumotlar bazalarining ichki himoya vazifasini keltiring?
2. SQL Server himoya vositasiga deganda nimani tushunasiz?
3. GRANT privilege nima vazifani bajaradi? Misol keltiring.
4. GRANT UPDATE uchun misollar yozing?
5. GRANT REFERENCES uchun misol keltiring?

4.2. Ma'lumotlar bazasiga murojaatni tashkil etishda ODBC va obyektga yo'naltirilgan dasturlar foydalanish

ODBC yordamida MySQL ma'lumotlar bazalari bilan ishlash

MySQL ma'lumotlar bazasi serveri bilan ishlash uchun WEB-ga asoslangan grafik foydalanuvchi interfeysini (GUI - Grafik foydalanuvchi interfeysi) yaratish DBMS ma'muriyatchilik mexanizmlarini tashkil qilishda keng tarqalgan usul bo'lishiga qaramay, ba'zi kamchiliklar mavjud emas. Xususan, ular syokin Internet aloqasini yoki turli xil brauzerlar yordamida ma'lumotlar bazasiga kirish imkoniyatini hisobga olish zarurligini o'z ichiga oladi. Tabiiyki, bunday muammolar ko'p jihatdan sizga axborot tizimining ishlab chiqaruvchisi sifatida bog'liq emas, ammo mijoz bunday nozik narsalarni ko'rib chiqishi dargumon. Shunday qilib, Internet-tizim ma'lumotlar bazasi asosida ishlab chiqish bilan bog'liq muammolarga qo'shimcha ravishda, siz ma'lumotlarni kiritishni tashkil qilishda bosh og'rig'iga duch kelasiz.

MyODBC-ni o'rnatishda birinchi qadam haqiqiy drayverni olish bo'lishi kerak. Agar sizda Internetga kirish imkoni bo'lsa, buni <http://www.mysql.com/downloads/api-myodbc.html> havolasini bosish orqali amalga oshirishingiz mumkin. Saytda Windows 95/98 /

NT, UNIX va Solaris operatsion tizimlarini boshqarish uchun drayverlar mavjud. MS Access-dan foydalanib MySQL bilan ishlashni istaganimiz uchun, albatta, Windows operatsion tizimi oilasi uchun drayverga murojaat qilishimiz kerak. Fayl hajmi taxminan 1.53 MB. O'rnatishning butun jarayoni sakkiz bosqichdan iborat:

1. Internetdan tegishli faylni yuklab oling (yoki uni boshqa usul bilan oling).

2. Arxivni ochish (Internet fayllari uchun eng keng tarqalgan arxivchi - WinZip - [http // www.winzip.com](http://www.winzip.com)).

3. O'rnatuvchilarni ishga tushiring (setup.exe).

4. Xush kelibsiz ekran paydo bo'lganda, Davom etish-ni bosing.

5. Endi drayverni o'rnatish dialogi paydo bo'ladi. Mavjud drayverlar orasida MySQL drayverini tanlang va "Ok" ni bosing.

6. Shundan so'ng, ma'lum miqdordagi fayllar o'rnatiladi.

7. Ma'lumot manbalarini tanlash uchun dialog oynasi paydo bo'ladi. "Yopish" ni bosing.

8. Haydovchini o'rnatish tugallandi. Tabriklaymiz, endi siz ODBC drayveri orqali MySQL bilan ishlash qobiliyatining g'urur egasisiz.

Men sizning e'tiboringizni quyidagilarga qaratmoqchiman: drayverni o'rnatishda MFC30.dll kutubxonasi bilan bog'liq muammolar paydo bo'lishi mumkin. O'rnatish jarayonida xato yuzaga kelsa, quyidagilardan birini bajaring:

1. Windows himoyalangan rejimda bo'lganida drayverni o'rnatish (Xavfsiz rejim). Buning uchun, qayta yuklashda, F8 tugmasini bosing.

2. Yana bir variant: odbccp32.inf faylini tahrirlash (faqat Windows 98 / NT operatsion tizimlarida ishlaganda) fayldan chiziqni o'chirish:

"MFC" = 1, mfc30.dll,, 1995-07-11,,,,,, 322832,,, 3.2.0.0,

Shunday qilib, drayverni o'rnatish muvaffaqiyatli yakunlandi. Men quyida MySQL va MS Access o'rtasidagi o'zaro aloqani tashkil qilish uchun MyODBC drayveri tomonidan taqdim etilgan imkoniyatlarni namoyish etadigan misol keltiraman.

Demo-loyiha. Aytaylik, biz Internet orqali ma'lum bir kompaniyaning mijozlari to'g'risidagi ma'lumotlarni ko'rish qobiliyatini tashkil qilishimiz kerak. Ishlab chiquvchi sifatida mijoz

bizdan Internet orqali ishlash tezligiga bog'liq bo'lmagan ilova yordamida ma'lumotlarni kiritish imkoniyatini berishimizni talab qiladi. Hech narsa oson emas. Biz shunchaki MS Access dasturidan foydalanib ma'lumotlarni kiritishni ta'minlaymiz. Shunday qilib, boshlaylik.

Birinchi qadam MySQL ma'lumotlar bazasini va uning jadvallarini yaratishdir. Buning uchun musqladmin dasturini ishga tushiring va quyidagi buyruqni bajaring:

```
%> musqladmin testdb yaratish
```

Endi biz yangi foydalanuvchi yaratamiz va o'nga yangi yaratilgan ma'lumotlar bazasiga to'liq kirish huquqini beramiz (ma'lumotlarni o'qish, qo'shish, yangilash va o'chirish uchun):

```
%> mysql -u ildiz -p mysql
```

Parolni kiriting:

MySQL monitoriga xush kelibsiz. Buyruqlar bilan tugaydi; yoki \\ g.

MySQL ulanish raqamingiz 17-server versiyasi: 3.22.34-dastur-disk raskadrovka

Yordam uchun "help" so'zini kiriting.

mysql> GRANT ni tanlang, qo'shing, yangilang, testdb-ga admin-ga o'chiring;

Maqolada biz GRANT buyrug'ining sintaksisini batafsil ko'rib chiqmaymiz. Men shuni aytishim mumkinki, bu holatda biz administratorga testdb ma'lumotlar bazasiga kirish huquqini berdik.

Endi "administrator" sifatida MySQL-ga kiring va mijozlar jadvalini yarating:

```
%> mysql -u admin
```

```
mysql> testdb-dan foydalaning;
```

```
mysql> jadvallar mijozlarini yaratish (
```

```
-> clientId o'rtacha bo'lmagan avtomatik_rektsiya,
```

```
-> name char (30) null emas,
```

```
-> city char (50) null emas,
```

```
-> telefon char (10) null emas,
```

```
-> elektron pochta uchun char (55) null emas,
```

```
-> birlamchi kalit (clientId));
```

Eslatib o'taman, jadval namoyish qilish uchun mo'ljallangan va siz ushbu maqolada ba'zi joylarni kiritish maqsadga muvofiqligi haqida savol tug'dirmasligingiz kerak.

Shunday qilib, barcha tayyorgarlik ishlari yakunlandi. Keyingi qadam, testdb ma'lumotlar bazasi bilan bog'liq ODBC manbasini mavjud ma'lumotlar manbalari ro'yxatiga qo'shish. Buning uchun biz quyidagi amallarni bajaramiz.

Mijoz kompyuterida ODBC Data Source Administrator (ODBC DSA) dasturini ishlatish. Buning uchun Boshqarish panelida "ODBC ma'lumotlar manbalari (32bit)" yorlig'ini tanlang. Tanlangan belgini bosgandan so'ng, sizning oldingizda ODBC DSA dastur oynasi paydo bo'ladi. "Qo'shish ..." tugmasini bosing. Siz yangi ma'lumotlar manbasini qo'shish uchun dialog oynasini ko'rasiz. Ushbu dialogda MySQL drayverini tanlang va "Finish" tugmachasini bosing. Shundan so'ng, biz yaratadigan ma'lumotlar manbasini sozlash uchun zarur bo'lgan ma'lumotlarni kiritish uchun shaklni ko'rasiz. Barcha maydonlar haydovchini konfiguratsiyasiga bir darajaga yoki boshqasiga ta'sir qilsa ham, eng muhimlaridan bir nechtasini ajratib ko'rsatish kerak.

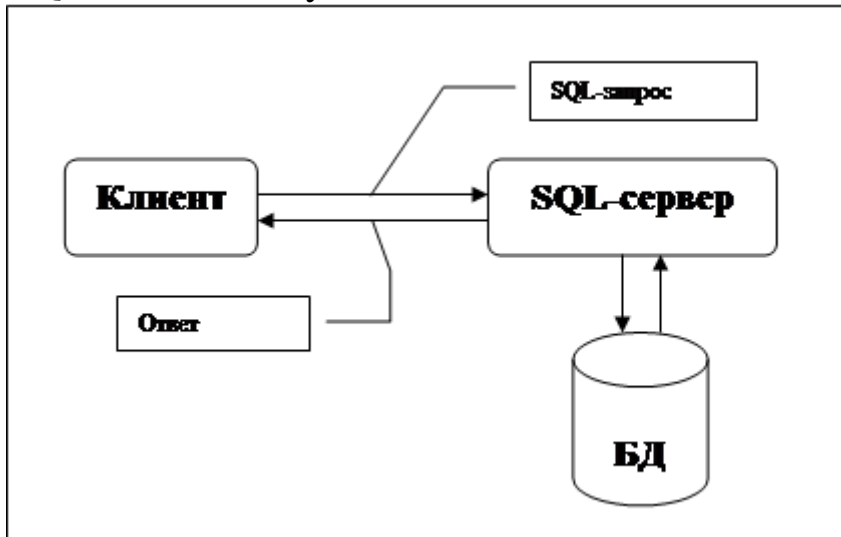
Windows DSN Name - Ma'lumotlarga kirish uchun mijoz dasturida keyinchalik foydalaniladigan ma'lumotlar manbasining nomi. Xususan, ushbu maydon uchun men tegishli ma'lumotlar bazasining nomini kiritaman. Bunday holda, bu testdb bo'ladi.

Shunday qilib, yuqoridagi barcha joylarni to'g'ri to'ldirgandan so'ng, "Ok" tugmasini bosing. Endi bizda ODBC standartiga mos keladigan har qanday dastur yordamida testdb ma'lumotlar bazasi bilan ishlash imkoniyati mavjud. Maqolaning keyingi qismida biz MS Access bilan ishlash haqida gaplashamiz.

4// Shunday qilib, bizda Internetda tarqatiladigan gipermedia hujjatlarini ishlab chiqish uchun qulay vositalar, WWW ma'lumotlarini olish uchun oddiy, qulay, ishlab chiqilgan va birlashtirilgan interfeyslar mavjud. Bundan tashqari, bizda turli xil MBBTlar tomonidan boshqariladigan juda ko'p miqdordagi qimmatli ma'lumotlar bazasi, shuningdek, ushbu ma'lumotlar bazalarini barcha odamlar uchun (ommaviy ma'lumotlar bazasi holatida) yoki geografik jihatdan taqsimlangan korporatsiya a'zolariga (korporativ ma'lumotlar bazasi holatida) kirish imkoniyatiga ega bo'lish istagi mavjud. Ushbu ikkita texnologiyani birlashtirish va veb-interfeysda ma'lumotlar

bazalariga kirishni ta'minlash tabiiy istagi bor. Ikki yil oldin, bunday xoch uchun faqat g'oyalar mavjud edi va amalga oshirishga juda puxta ishlab chiqilmagan yondashuvlar mavjud edi. Bugo'ngi kunda bunday mexanizmlar allaqachon mavjud va ishlatiladi.

SQL Server tamoyillari



SQL bu kompyuter ma'lumotlar bazasidagi ma'lumotlarni qayta ishlash va o'qish uchun mo'ljallangan vositadir. SQL (tuzilgan so'rovlar tili) nomidan ko'rinib turibdiki, foydalanuvchilarning ma'lumotlar bazasi bilan o'zaro munosabatlarini tashkil qilish uchun foydalaniladigan dasturlash tili. Aslida, SQL faqat ma'lum bir turdagi ma'lumotlar bazalari bilan ishlaydi, ular relyatsion deb ataladi.

Rasmda SQL ishlash sxemasi ko'rsatilgan. Ushbu sxema bo'yicha kompyuter tizimida muhim ma'lumotlar saqlanadigan ma'lumotlar bazasi mavjud. Agar ma'lumotlar bazasi biznes sohasiga tegishli bo'lsa, unda moddiy boyliklar, mahsulotlar, sotish hajmi va ish haqi to'g'risidagi ma'lumotlar saqlanishi mumkin. Shaxsiy kompyuterdagi ma'lumotlar bazasi chiqarilgan cheklar, telefonlar va manzillar yoki katta kompyuter tizimidan olingan ma'lumotlarni saqlashi mumkin. Ma'lumotlar bazasini boshqaradigan kompyuter dasturi ma'lumotlar bazasini boshqarish tizimi yoki DBMS deb nomlanadi.

Agar foydalanuvchi ma'lumotlar bazasidan ma'lumotlarni o'qishi kerak bo'lsa, u ularni SQL-dan DBMS-dan foydalanib so'raydi. SQL so'rovni qayta ishlaydi, kerakli ma'lumotlarni topadi va foydalanuvchiga yuboradi. Ma'lumotni so'rash va natijani olish

jarayoni ma'lumotlar bazasi so'rovi deb nomlanadi: shuning uchun nom - tuzilgan so'rovlar tili.

Biroq, bu nom umuman to'g'ri emas. Bugo'ngi kunda SQL bu oddiy so'rov vositasidan ko'proq narsa, garchi aynan u aynan aynan aynan shunday yaratilgan bo'lsa. Ma'lumotni o'qish hali ham SQL-ning eng muhim funktsiyalaridan biri bo'lishiga qaramay, hozirda ushbu til ma'lumotlar bazasi foydalanuvchiga beradigan barcha funktsiyalarni amalga oshirish uchun ishlatiladi, xususan:

- ma'lumotlarni tashkil qilish. SQL foydalanuvchiga ma'lumotlarni taqdim etish tuzilmasini o'zgartirish, shuningdek ma'lumotlar bazasi elementlari o'rtasidagi munosabatlarni o'rnatish imkoniyatini beradi.

Ma'lumotlarni o'qish. SQL foydalanuvchi yoki dasturga ma'lumotlar bazasida mavjud bo'lgan ma'lumotlarni o'qish va ulardan foydalanish qobiliyatini beradi.

Ma'lumotlarga ishlov berish. SQL foydalanuvchi yoki dasturga ma'lumotlar bazasini o'zgartirish imkoniyatini beradi, ya'ni. o'nga yangi ma'lumotlar qo'shing, shuningdek undagi ma'lumotlarni o'chiring yoki yangilang.

Kirish huquqini boshqarish. SQL-dan foydalanib, siz ma'lumotni o'qish va o'zgartirish imkoniyatini cheklashingiz, ruxsatsiz kirishdan himoya qilishingiz mumkin.

- Ma'lumot almashish. SQL bir-biriga halaqit bermasligi uchun parallel ravishda ishlaydigan foydalanuvchilar o'rtasida ma'lumotlarni almashishni muvofiqlashtiradi.

Ma'lumotlarning yaxlitligi. SQL ma'lumotlar bazasining yaxlitligini ta'minlashga, uni tizimdagi uzilishlar yoki buzilishlar tufayli buzilishlardan himoya qiladi.

Shunday qilib, ma'lumotlar bazasi SQL bilan o'zaro ishlash uchun yetarlicha kuchli vositadir.

Relyatsion ma'lumotlar bazasining asosiy obyektlari:

(TABLE) jadvali

ROWS va COLUMNSlardan iborat to'rtburchaklar stol. Jadvalni aniqlash uning qaysi ustunlardan iboratligini ko'rsatishni anglatadi.

(ROW) qator

Maydon - ustunlardan iborat yozuv. Har bir maydon o'z qiymatini o'z ichiga oladi yoki NULL - "bo'sh". Jadvaldagi satrlar har

qanday raqam bo'lishi mumkin. Ularning bir-biriga nisbatan joylashishining jismoniy tartibi aniq emas.

(COLUMN) ustun Jadvaldagi har bir ustun o'z nomi va turiga ega.

SQL jadvallari.Bog'lanish ma'lumotlar bazasida ma'lumotlar jadvallar ko'rinishida va satrlar va ustunlarga bo'linib, ularning kesishmasida ma'lumotlar qiymatlari joylashtirilgan. SQL tilida ishlatiladigan kalit birikmalar (CREATE TABLE my_table - jadvalni yaratish my_table) "jumlar" deb nomlanadi. Jadvallar SLQ-da CREATE TABLE gapidan foydalanib tuziladi. CREAT TABLE yozuvi yaratilishi kerak bo'lgan asosiy jadvalning nomi, uning ustun nomlari va ushbu ustunlar uchun ma'lumotlar turlarini belgilaydi. CREAT TABLE - doimiy taklif. Agar SQL serveriga CREATE TABLE so'rovi berilsa, tizim dastlab bo'sh bo'lgan jadvalni yaratadi: unda faqat ustun sarlavhalari qatori bo'ladi, ammo ma'lumotlar bilan bir qator mavjud bo'lmaydi. Ma'lumotlar jadvalga INSERT bayonnomasi yordamida kiritiladi.

SQL so'rovining tuzilishi. Bir yoki bir nechta jadvallardan deyarli har qanday ma'lumotlarni olish uchun barcha so'rovlar bitta SELECT gapidan foydalanib bajariladi.

Ma'lumotlar bazasiga kirish uchun sintaktik tuzilmalarda quyidagi belgi qo'llaniladi:

"Hamma narsa" degan ma'noni anglatuvchi yulduzcha (*) odatdagi ma'noda dasturlash uchun ishlatiladi, ya'ni. "Ta'rifga javob beradigan barcha holatlar";

· Kvadrat qavslar ([]) - bu qavslarga o'ralgan konstruktsiyalar ixtiyoriy ekanligini anglatadi (ya'ni ular o'tkazib yuborilishi mumkin);

· Jinalak qavslar ({}) - bu qavslar ichiga o'rnatilgan inshootlar butun sonli sintaktik birliklar sifatida ko'rib chiqilishini anglatadi, ya'ni. ular SQL sintaksisida ishlatiladigan odatiy qavslarni almashtirib, sintaktik tuzilishlarni tahlil qilish tartibini aniqlashtirishga imkon beradi;

· Ellipsis (...) - bu oldingi sintaktik birlik ixtiyoriy ravishda bir yoki bir necha marta takrorlanishi mumkinligini anglatadi;

To'g'ri chiziq (|) - ikki yoki undan ko'p imkoniyatlarni tanlashni anglatadi. Masalan, ASC | DESC belgilashi ASC yoki DESC atamalaridan birini tanlash mumkinligini bildiradi; tanlov

elementlaridan biri kvadrat qavs ichiga o‘ralgan bo‘lsa, bu uning sukut bo‘yicha tanlanganligini anglatadi (demak, [ASC] | DESC bu butun qurilishning yo‘qligi ASC tanlovi sifatida qabul qilinishini anglatadi);

- Nuqta-vergul (;) - SQL iboralarining oxirgi elementi;
- Vergul (,) - ro‘yxat elementlarini ajratish uchun ishlatiladi;
- Bo‘sh joylar () - har qanday SQL iboralarining sintaktik tuzilishlari orasidagi ko‘rinishni oshirish uchun kiritilishi mumkin;
- Katta katta lotin harflari va belgilari - SQL tuzilishini yozish uchun ishlatiladi va aniq ko‘rsatilgan holda yozilishi kerak -;
- Kichik harflardan foydalanishda konstruktsiyalarni yozish uchun foydalaniladi, ularni foydalanuvchi tanlagan aniq qiymatlar bilan almashtirish kerak va aniqlik uchun ushbu inshootlarning individual so‘zlari pastki chiziq () bilan bog‘langan;

"Jadval" va "ustun" atamaları (sintaktik tuzilmalar matnini qisqartirish uchun) mos ravishda "table_name", "column_name", ... so‘zlarini almashtiradi;

"Jadval" atamasi - jadval_table, view yoki tahallus kabi jadval turlarini umumlashtirish uchun ishlatiladi; bu erda tahallus vaqtincha (so‘rov vaqtida) nomini o‘zgartirish va (yoki) base_table (ko‘rinish) ning ishchi nusxasini yaratish uchun ishlatiladi.

SELECT (tanlash) buyrug‘i quyidagi formatga ega:

```
subquery [UNION [ALL] subquery] ...  
[BUYURTMA by {jadval.] Ustun |  
SELECT_element_number} [[ASC] | DESC]  
[, {jadval.] ustun | SELECT_element_number} [[ASC] |  
DESC]] ...;
```

va birlashtirish (UNION) va keyin bir nechta "quyi so‘rovlar" yordamida olingan ma'lumotlarni tanlash natijalarini (ORDER BY) buyurtma qilish imkonini beradi. Bunday holda, buyurtma ko‘tarilish tartibida amalga oshirilishi mumkin - ASC (ASCending) yoki DESC (DESCending) kamayadi, asl qiymati - ASC.

Ushbu jumlada pastki so‘rov sizga kerakli ma'lumotlarni tanlash shartlarini va (agar kerak bo‘lsa) ularni qayta ishlashni ko‘rsatishga imkon beradi

TANLASH

Belgilangan ustunlardan ma'lumotni tanlang (tanlang) va (agar kerak bo‘lsa) belgilangan iboralar va (yoki) funktsiyalarga mos ravishda chiqishdan oldin ularning konversiyasini amalga oshiring.

FROM

(bu) ustunlar joylashgan jadvallar qayerda(bu erda) ko'rsatilgan jadvaldagi qatorlar belgilangan qatorni tanlash shartlariga mos kelishi kerak:

GROUP BY

SQL SUM (yig'indisi), COUNT (miqdori), MIN (minimal qiymati), MAX (maksimal qiymati) yoki AVG funktsiyalaridan foydalanib, har bir guruh uchun yagona jamlangan qiymatni olish uchun ko'rsatilgan ustunlar ro'yxatini (guruhlarga ajratish) o'rtacha qiymati)

TANLASH [[HAMMA] | DISTINCT] { * | element_SELECT [, element_SELECT] ... }

Nazorat savollari

1. Ma'lumotlar bazalarining ichki himoya vazifasini keltiring?
2. SQL Server himoya vositasiga deganda nimani tushunasiz?
3. GRANT privilege nima vazifani bajaradi? Misol keltiring.
4. GRANT UPDATE uchun misollar yozing?
5. GRANT REFERENCES uchun misol keltiring?

4.3. XML va ma'lumotlar bazalari

Web-texnologiyaning (Internet-texnologiya) Web-dizayn qismini o'rganishni razmetkali til tasnifi bilan boshlaymiz. Maxsus til mavjud bo'lib, bu til yordamida matnlar, grafik ma'lumotlar Web-sahifa xujjatga joylashtiriladi va bu xujjatni barcha kompyuterda ko'rish imkoniyati mavjuddir. Bunday maxsus tillar razmetkali tillar deb ataladi. Ularning asosiy vazifasi – Web-sahifaga “ma'lumotlarni joylashtirish” va ular orasidagi aloqani (giperssilokalar) ta'minlashdan iborat.

XML (eXtensible Markup Language).XML tili ham HTML tiliga o'xshash til hisoblanadi. HTML dan farqli tomoni shundaki, XML da dasturchi o'zining shaxsiy teglarini yaratadi va ular orasigama'lumotlar joylashtiradi. XML-teglar harflar katta kichikligini farqlaydi.

XML tili

XML va HTML orasidagi oltin qoidani ifodalovchi til, XML (extensible Markup Language) — kengaytirilgan belgilashlar tilidir. Bu til SGML dan ajratib olingan to‘plam bo‘lib, web- programmistga o‘z teglarini yaratish imkonini beradi va ortiqcha murakkablikdan holi.

XML ni ishlab chiqish 1996 boshlandi. Bu ish bilan jamoat tashkiloti W3C(World Wide Web Consortium), shug‘ullanadi. Uning asosiy sayti : <http://www.w3c.org/>. XML

haqidagi ma‘lumotlar <http://www.w3c.org/XML/> sahifaga joylashtirilgan 1998 bu tashkilot XML 1.0. versiyasini chiqardi. XML tobora takomillashib bormoqda. Uning oxirgi versiyasi quyidagi adresda bo‘ladi: <http://www.w3c.org/TR/rec-xml>.

- XML taraqqiyotining kelguvsidagi yo‘nalishlari:
- XFragment
- XML hujjatning alohida qismlariga ma‘no berishga urinish.
- XML Schemas
- XML alifbosi bo‘yicha, XML hujjatning mazmunini ifodalash urinish.
- XML Query Language (XML zaproslar tili)

Hujjatda biror shartni qanoatlantiruvchi elementlarni izlashga urinish

- Canonical XML
- XML hujjatdagi elementning argumentlari oladigan qiymatlarni ko‘rsatishni standart algoritmi.
- XML Signatures
- XML hujjatda raqamli podpisni ifodalashni standart vositalari
- XML o‘zini turli texnologiyalar uchun ishonchli, qat’iy asos ekanini ko‘rsatdi.
- XML hujjatning strukturasi
- XML da tayyor teglar yo‘q, balki o‘z teglarini yozish qoidalari mavjud.
- Aniq teglar haqida gapirganda XML ni amalda qo‘llash nazarda tutiladi
- XML qoidalarini amalda qo‘llab, ma‘lum teglar guruhi tuzilgan, ular Aniq sohada ishlatiladi:

- XHTML — XML qoidalariga mos ravishda tuzilgan giperMatnni belgilash tili;
- Math ML — matematik formulalarni yozish tili;
- CML — ximik formulalarni yozish tili;
- Vox ML — ovozlarni yozish tili;
- WML —simsiz texnologiyada ishlatiladigan til.

XML hujjatni bezatishni va teglar yaratishni faqatgina umumiy qoidalarini beradi. Bu qoidalar XMLning hususiyatida ko‘rsatiladi. Hujjat prolog keltirishdan boshlanadi.

XML hujjatning prologi

Prolog ikki qismdan iborat bo‘ladi. Prologning birinchi qismida XML e‘lon yoziladi. (XML Declaration). U `<?...?>` simvollar orasiga yoziladi, bu e‘londa xml belgi va uning versiyasi keltiriladi. U quyidagi ko‘rinishga ega bo‘ladi:

```
<?xml version="1.0"?>
```

Bundan tashqari e‘londa hujjat yozilgan tilning kodi ham keltirilishi mumkin

```
<?xml version="1.0" encoding="utf-8"?>
```

Kiril alfaviti ishlatilgan hujjatlar uchun, e‘lonning ko‘rinishi quyidagicha bo‘ladi: `<?xml version="1.0" encoding="windows-1251"?>`

E‘londa yana standalone parametrini ham keltirilishi mumkin. Bu parametr ikki

qiymatdan birini oladi: "yes" yoki "no". "no" qiymati shuni ko‘rsatadiki, bu hujjarda ishlatilgan elementlarni ta‘riflash uchun tashqi hujjatdan foydalanilgan. Odatda standalone "yes" qiymatini oladi. Demal XML hujjatni e‘lon qilish quyidagi ko‘rinishda bo‘ladi:

```
<?xml version="1.0" encoding="windows-1251" standalone="yes"?>
```

Prologni ikkinchi qismida — hujjatning tipi e‘lon qilinadi, DTD (Document Type Declaration). Bu qismda hujjatda foydalanilgan teglar yoki alohida faylga ssilka bo‘ladi. Hujjatni tipini ko‘rsatish `<DOCTYPE`, so‘zi bilan boshlanadi va uchburchakli qavs belgisi `>` bilan tugaydi.

Quyidagi web-saytda xml tilining vazifasi va amalda qo‘llash maqsadi tushuntiriladi.

<http://www.w3.org/TR/REC-xml>.

1. XML Internetda bevosita foydalaniladigan til bo'lishi kerak.

XML asosan web- muhitida axborotni saqlash va tarqatish uchun ishlab chiqilgan til.

2. XML ko'p miqdordagi programmalar aro axborot almashishda qo'llaniladi.

XML ni asosiy vazifasi serverlar va brauzerlar orqali axborot tarqatish bolsa ham, undan boshqa programmalarda ham foydalaniladi. Masalan finans programmolari aro axborot almashinuvi, programma maxsulotlarini yangilash va tarqatish.

3. XML, SGML bilan mos keladi.

XML aslida SGMLning maxsus shoxchasi hisoblanadi.

4. XML-hujjatlarni qayta ishlovchi programmalarini yozish oson.

5. XMLning qo'shimcha funktsiyalarini miqdori minimal, idealda nolga teng bo'lsin. Qo'shimcha funktsiyalarni minimal bo'lishi xml-hujjatni qayta ishlovchi programmani yozishni soddalashtiradi

6. XML-hujjatlar foydalanuvchi uchun tushunarli va ochiq-ravshan bo'lishi kerak. XML butun dunyodagi programmalar va foydalanuvchilar orasida axborot almashinuvi uchun universal til (lingua franca) boladi. Foydalanuvchilar XML- hujjatni osonlik bilan o'qiydilar, chunki u oddiy Matnsifatida ifodalanadi va daraxt ko'rinishidagi iyerarxik strukturaga ega.

7. XML ni ishlab chiqish yyetarli darajada tez tugallanishi kerak.

8. XML tili formal va qisqa bo'lishi kerak.

9. XML-hujjatlarni yaratish oson ish bo'lsin.

10. XML-hujjatlar siqilgan formatda bo'lsa , hajmi minimal qiymatni oladi.

Ma'lumotga yo'naltirilgan hujjatlarda ma'lumotlarni saqlash va olish uchun sizga qanday dastur kerakligi sizning ma'lumotlaringiz qanchalik yaxshi tuzilganligiga bog'liq bo'ladi. Telefon kitobidagi oq sahifalar kabi yuqori darajada tuzilgan ma'lumotlar uchun sizga ma'lumotni saqlash uchun konfiguratsiya qilingan XML bilan bog'langan ma'lumotlar bazasi, masalan, relatsion yoki obyektga yo'naltirilgan ma'lumotlar bazasi va ma'lumot uzatish dasturining bir turi kerak bo'ladi. Buni ma'lumotlar bazasiga kiritish mumkin (bu holda ma'lumotlar bazasi XML-quvvatlanadi) yoki u o'rta dasturlar yoki XML-server kabi uchinchi tomon dasturlari bo'lishi mumkin. Agar sizning ma'lumotlaringiz qisman tuzilgan bo'lsa, masalan, telefon daftaridagi sariq sahifalar yoki sog'liq to'g'risidagi

ma'lumotlar, sizda ikkita variant mavjud. O'zingizning ma'lumotlaringizni aloqador ma'lumotlar bazasi kabi yaxshi tuzilgan ma'lumotlar bazasiga joylashtirishga harakat qilishingiz mumkin yoki yarim tuzilgan ma'lumotlarni qayta ishlash uchun mo'ljallangan o'z XML bazangizda saqlashingiz mumkin.

Ko'pgina protokollar, fayl tizimlari va Butunjahon Internetining texnologiyalari singari kengaytiriladigan markalash tili (XML) - kengaytiriladigan belgilash tili - kam vaqt ichida keng foydalanishga o'tdi.

Dastlab veb-nashrlar tomonidan hujjatlar almashish texnologiyasi sifatida tanilgan XML sanoat tomonidan tanilgan saqlash va uzatish muhitiga aylandi.

Bir vaqtlar Internetda chop etilgan hujjatlar deyarli faqat matnlar va rasmlardan iborat bo'lgan bo'lsa, endi ushbu hujjatlar (allaqachon XML asosida) ichki (back end) tizimlarning ma'lumotlar bazasidan tanlab olingan ma'lumotlarni, shuningdek ilovalarni va ma'lumotlarni etkazib berish uchun eng maqbul vositaga aylanmoqda. o'rta (o'rta daraja) va tashqi (oldingi) tizim hujjatlari.

Ammo endi XML ma'lumotlar ombori tarkibiga kira boshladi va ma'lumotlarni saqlash tuzilmalarining ajralmas elementiga aylandi.

Ilgari, Oracle XML Developers to'plamlari (XDKs) ma'lumotlar bazasidan tashqarida XML ma'lumotlarida (yoki XML ma'lumotlarida) ishlash uchun ishlatilgan, ammo endi dastur ishlab chiquvchilari ushbu ma'lumotlarga ishlov berishni dasturiy interfeyslardan foydalanadigan barcha darajalar o'rtasida taqsimlashlari mumkin. ushbu to'plamlar XDK API yoki SQL XML interfeyslari.

Biroq, ma'lumotlar bazalarida XML-ni saqlash va boshqarish bilan bog'liq ko'plab muammolar mavjud va bu muammolar sizning barchangizga ularni engib o'tish mumkinligini bilishi mumkin, ammo buni qanday qilishni aniq bilmaydi. Asosiy ma'lumotlar bazalari relyatsion tuzilishga ega va XML ierarxik tuzilishga ega va yaqin vaqtgacha ularni birlashtirishning oddiy va oqlangan usuli yo'q edi. An'anaga ko'ra, ishlab chiquvchilar ikkita tanlov qilishgan:

yoki deklaratsiyalash / dekompozitsiyalash uchun hujjat ma'lumotlarini nisbiy ma'lumotlarga aylantirish yoki ularni ma'lumotlar bazasida saqlash uchun foydalaning, yoki butun hujjatni

matnli fayl sifatida saqlang (ma'lumotlar bazasida yozuv sifatida) va shu bilan uning tuzilishini saqlab qoladi.

Shuni yodda tutish kerakki, XML-ni ma'lumotlar bazasiga "ko'chirish" barcha holatlarda bir xildir. Ma'lumotni saqlashning har bir modeli o'zining ijobiy va salbiy tomonlariga ega. Qaysi XML saqlash modeli sizning dasturingiz ehtiyojlariga eng mos kelishini tushunish sizning muvaffaqiyatingiz uchun juda muhim va sizning ilovangizni ba'zi XML saqlash modeliga moslashtirishdan ko'ra muhimroqdir.

Ushbu maqolaning maqsadlaridan biri o'zingiz uchun eng yaxshi modelni aniqlashga yordam berishdir. Ushbu maqolada men saqlashning ikkala modelini ham tushuntiraman:

Hujjatni "parchalash" (XML faylini aloqador ma'lumotlarga aylantirish va keyinchalik SQL orqali boshqarish) va harakterli katta obyekt (CLOB) saqlash modeli.

Keyin men sizni Oracle-ning XML DB omborxonasi va ushbu (ikkita) XMLType bilan tanishtiraman, ular bir nechta qo'shimcha funktsiyalar bilan ushbu ikki modelning kombinatsiyasidir. Ideal holda, sizning ilovangizga mos keladigan saqlash modeli haqida yaxshi tasavvurga ega bo'lasiz.

Agar HTML - bu gipermatnli belgilash tili bo'lsa, XML - bu har qanday tuzilgan ma'lumotlarni taqdim etishga imkon beradigan til. HTML bizga hujjatning tuzilishini, har qanday elementlarni, chiziqlarni, bloklarni va boshqalarni namoyish qilishimizga imkon beradi. XML bizga ma'lumotlarni saqlash uchun qulay bo'lgan strukturani yaratishga imkon beradi.

Belgilangan tillarning barchasi bitta, ammo juda katta kamchilikka ega. Xuddi shu HTML veb-sahifalarni yoki HTML-hujjatlarni taqdim qilishimizga imkon beradi va biz ushbu go'zallikni brauzerlarimizda ko'rishimiz mumkin, ammo HTML boshqa ba'zi mijozlar yoki dasturlarda yoki boshqa maqsadlarda qulay bo'lishi haqiqat emas. XML-da bunday kamchiliklar yo'q. Uni kengaytiruvchi deb nomlash bejiz emas, X harfi eXtensible bo'lib, kengaytiriladigan deb tarjima qilinadi.

XML bu mutlaqo universal va mutlaqo kengaytiriladigan ma'lumotlarni belgilash tili. XML-ning kengayishi tufayli biz qo'llarimizni kuchli vositaga olamiz, uning asosida siz belgilash tilini o'zingiz ishlab chiqishingiz va undan foydalanishingiz mumkin.

XML-ning yana bir shubhasiz afzalligi - bu platformaning mustaqilligi. Ya'ni biz baraban ustida, qanday operatsion tizim bilan ishlashni, qaysi dastur bilan XML bilan ishlashni, qaysi dasturlash tilini, bularning barchasida XML uchun.

XML hujjati. XML hujjatining tuzilishi. XML hujjatida nima bo'lishi mumkin?

Endi XML hujjatlari haqida suhbatlashamiz, XML hujjatining tuzilishi bilan tanishamiz va shuningdek, XML hujjatiga nima kiritilishi mumkinligini ko'rib chiqamiz. XML-ning maqsadi ma'lumotlarni siz uchun qulay bo'lgan dasturlash tilidan foydalangan holda sizga qulay shaklda va platformada taqdim etishdir. Hammasi yaxshi bo'lgani kabi aqldan ozishingiz mumkin.

Va shuning uchun XML hujjati quyidagilarni o'z ichiga oladi: ma'lumotlar, ushbu ma'lumotlarning tuzilishi, shuningdek ma'lumotlar taqdim etish darajasi. Umuman olganda, biz hujjatlarga tez-tez duch kelamiz, pochta qutilariga spam ko'rinishidagi Excel-da turli xil narxlar ro'yxati keladi, men kecha maktabda geografiya bo'yicha insho topshirdim, chorak tugaganidan so'ng, men o'zimning baholarimni kuchaytirishim kerak, shuning uchun bu abstrakt ham hujjat edi, lekin allaqachon. dok formatida.

Shunday qilib, geografiya bo'yicha abstraktda Yerning dumaloq va aylanuvchi ekanligi haqida ma'lumotlar bor edi, Galiley Galileyning tarjimai holi - bu ma'lumotlar. Mening geografiya haqidagi inshomim tuzilishga ega edi, ya'ni men o'zimning inshomim nima va uni qanday maqsadda yozganim haqida gapirib berdim (tabiiyki, mening maqsadim xudbinlik, uchlik olish edi, shunda chorakda ikkitasi bo'lmasdi, lekin uni kirish qismida yozmadim), xulosa chiqardim, bajarilgan ishimni (Vikipediya dan asabiy nusxa ko'chirma nusxasi va tezislar va dz saytlaridan) xulosa chiqardim, qulaylik sifatida men abstraktni bir nechta mantiqiy qismlarga ajratdim: Galileo Galileyning tarjimai holi, Planet Er, sayyora Yer Quyosh tizimining obyekt sifatida. Bularning barchasi mening mavhumimning tuzilishidir va XML-da biz XML hujjatining tuzilishini o'zimiz yaratamiz. Odatda, XML hujjati daraxt tuzilmasiga ega, uni shartnomalar bilan taqqoslash mumkin, masalan, ipoteka shartnomasi, 1-band, 1.1-band, keyin 1.1.1-band, uni bajargan kishi bu aslida byurokratik tuzilma ekanligini biladi. Bularning barchasi uyasi

yoki daraxt yoki daraxt tuzilishi deb atash mumkin, bu XML hujjatining tuzilishi haqida.

XML taqdimot qatlami yoki hujjatning XML namoyishi. Yaxshi, bularning barchasi juda sodda, taqdimot darajasi - bu hujjatni monitor ekranida, brauzerda, printerda va hokazolarni qanday ko'rayotganingiz, mening inshoimga qaytarganimdan so'ng, men jasur sarlavhalarni ajratib ko'rsatdim, ularni 16 ta shrift o'lchamiga aylantirdim va shuningdek qildim. ular kursiv, men chap tomonda 20 mm, chapda, o'ngda, pastki va tepada 5 mm. Mavzu ichidagi matn uchun men Times New Roman shriftini 14 o'lchamga o'rnatdim, qo'shimcha uslublarsiz, satr oralig'i 1,5 edi, bularning barchasi Wordda va eskirganimdan so'ng ekranda ko'rish mumkin bo'lgan hujjatning taqdimoti deb atash mumkin. geografiya bosma. XML shuningdek hujjatning namoyishiga ega.

Ammo XML boshqa markalash tillaridan juda muhim farq qiladi, yana HTML - bu elementlar qaerda joylashishi, qanday joylashishi, ushbu elementlarning o'lchamlari va boshqalar bo'lgan odatiy sahifa uchun tavsiflash tili. Ammo XML-da taqdimot darajasi yo'q, XML yordamida siz faqat ma'lumotlar va munosabatlarni tasvirlashingiz mumkin. Bizning birinchi stolimizdan bo'lgan nadejda, Nadejda Petrovnaning (matematika o'qituvchisi) sevimlisi, albatta savol berishadi - Buni qanday tushunish mumkin? Yoki XML ma'lumotlarni qaysi shaklda taqdim etadi? Bu savolga javob oddiy - XML siz istamaguningizcha ma'lumotlarni biron bir shaklda taqdim etmaydi va hujjat siz xohlaganingizcha ko'rinishi mumkin, ammo bu XML formatida standart emas. Yoki XML hujjati brauzer ekranida qanday ko'rinishi kabi savol? Javob - XML hujjati brauzer ekranida hech qanday ko'rinishga ega bo'lmaydi. Faqat XML hujjati o'zi brauzer ekranida bosilgan. XML bu juda mavhum til bo'lib, unda hujjatni o'zingiz qilishingiz va yuragingiz xohlagan narsani bajarishingiz mumkin.

XML ma'lumotlari bo'yicha so'rovlar. Turli xil standartlar va so'rovlar tillarining paydo bo'lishining chuqurligi va kelib chiqish sabablarini o'rganishdan oldin, biz aniq bir savolga javob beramiz - nima uchun bularning barchasi, eski SQL-dan foydalanish mumkinmi?

Nega SQL emas?

Ma'lumotlar bazasini boshqarish tizimlari bugo'ngi kunda yirik axborot tizimlari uchun amalda standart hisoblanadi. Biroq, nisbiy tushunchaning o'zi XML ma'lumotlarini saqlash uchun juda mos emas. XML hujjatlarini saqlash oldindan aniqlangan sxemani talab qilmaydi, bu ma'noda XML obyektlari o'z-o'zidan talab qilinadi, ular so'rovni yaratish va individual elementlarga kirish uchun zarur bo'lgan barcha ma'lumotlarni o'z ichiga oladi. Ushbu yondashuv oxir-oqibat biroz ko'proq saqlash joyini talab qiladi, ammo hech qanday qayta qurishsiz narsalarni qo'shish va o'chirishga imkon beradi. Aloqador ma'lumotlar bazasida yangi qiymatlardan foydalanish imkoniyati uchun kamida UPDATE SCHEMA ADD / DROP COLUMN ishlash talab etiladi - mijoz dasturlarini qayta dasturlash.

Klassik relatsion modelda qo'llab-quvvatlanmaydigan XML hujjatlarini saqlash va boshqarish tizimlari uchun bir qator muhim talablar mavjud. Birinchidan, ta'rifiga ko'ra (birinchi oddiy shaklning talabi), ustunlar atom qiymatiga ega bo'lishi kerak, bu XML ma'lumotlari uchun mutlaqo maqbul emas. Shunday qilib, XML obyekti bir necha jihatdan saqlanadi va ma'lumotlarni olish uchun bir nechta ma'lumot ulanish operatsiyalari talab qilinadi, ya'ni. asosiy va tashqi kalitlarni taqqoslash, shu bilan birga XML hujjatlaridagi havolalar to'g'ridan-to'g'ri ko'rsatgichdan boshqa narsa emas. Bundan tashqari, o'zaro bog'liqlik sxemasida butun obyektning yaxlitligini ta'minlash uchun tegishli tetikleyicilarni amalga oshirish talab etiladi.

Ikkinchidan, munosabatlarda chiziqlar buyurtma qilinmaydi, ya'ni. chiziqqa pozitsiyali emas, balki faqat tarkib bo'yicha kirish mumkin - ierarxiya qo'llab-quvvatlanmaydi. Albatta, sanoat tizimlarida ierarxiyalarni bitta jadvalda (munosabatlarda) qo'llab-quvvatlash usullari mavjud, ammo oxir-oqibat bu hali ham qiymat bo'yicha qo'ng'iroq orqali amalga oshiriladi.

Keling, XML ma'lumot modeli va relyatsion model o'rtasidagi yuqoridagi farqlarni umumlashtiramiz.

- Teglar va atributlar nomlari kabi metadata hujjatning o'zida joylashgan (ma'lumotlar tuzilishi to'g'risidagi metadata alohida tizim jadvallarida mavjud).
- Hujjatda daraxt tuzilishi mavjud.

- Hujjatning elementlari joylashtirilishi va atributlardan iborat bo'lishi mumkin (ustunlar atom qiymatlariga ega bo'lishi kerak).
- Bir xil tegli bir nechta elementlarga ruxsat beriladi (bir xil nomli bir nechta ustunlarga ruxsat berilmaydi).
- Elementlar ketma-ketlikni hosil qiladi va ma'lum bir pozitsiyaga ega (bir nechta atributlar to'plami, ya'ni tartibga solinmagan to'plam).

So'rovlar tillariga murojaat qilsak, SQL bir necha normallashtirilgan jadvallardan ma'lumotlarni birlashmalar orqali olishga yo'naltirilgan ma'lumotlarning nisbiy modelini amalga oshiradi. XML holatida, ko'proq bitta hujjatning ierarxik tuzilishidagi elementlarni ta'kidlash haqida. Shunday qilib, SQL-ning ifodali kuchi XML uchun foydasiz va XML hujjatlarining ma'lumotlar modeli hech qanday tarzda ishlatilmaydi. Mavjud so'rovlar tillarini o'rganayotganda SQL-ning semantik cheklovlari aniqroq bo'ladi.

XML ma'lumot to'plami va XML ism maydonlari. XML-da so'rovlar qurilishini muhokama qilishni davom ettirishdan oldin, boshqa barcha standartlar va tavsiyalarga asoslangan W3C XML Faoliyatida ikkita asosiy standartni eslatib o'tish kerak.

XML ma'lumot to'plami yaxshi tashkil etilgan XML hujjatlarida mavjud bo'lgan ma'lumotni tavsiflovchi mavhum ma'lumotlar to'plamini belgilaydi. XML Infoset faqat Ism maydoni spetsifikatsiyasiga mos keladigan hujjatlar uchun belgilanadi. XML Infoset tashqi ma'lumot taqdim etish interfeysini aniqlamaydi va afzal ko'rmaydi, xoh u daraxt tuzilishi, na voqea yoki so'rovlarga asoslangan interfeys. Ma'lumotlar to'plamidagi ma'lumotlar biron-bir tarzda yoki boshqa usulda mavjud bo'lgandan so'ng, XML Infoset spetsifikatsiyasining talablari qondirilgan deb hisoblanadi.

Axborot to'plamida har biri xususiyatlarga ega bo'lgan o'n besh xil turli xil ma'lumotlar elementlari mavjud. Axborot elementlari orasida quyidagilar mavjud: hujjat, element, atribut, ishlov berish bo'yicha ko'rsatma, DTD, CDATA bo'limining boshi va oxiri, shuningdek harflar belgisi elementi. Belgilar to'g'risidagi ma'lumot elementi hujjat elementidagi va CDATA bo'limidagi har bir belgi uchun, shuningdek, normallashtirilgan atribut qiymatlari uchun

belgilanadi. Axborot elementlarining xususiyatlari orasida ushbu elementning bolalari mavjud (ba'zi ma'lumot elementlari, masalan, ishlov berish ko'rsatmalari, bolalari yo'q), ya'ni. Shunday qilib, ma'lumotlar to'plami ta'kidlangan uchi ("hujjat" ma'lumotlar elementi) bilan butun yo'naltirilgan grafik strukturani hosil qiladi. Shunday qilib, XML ma'lumot to'plami XML hujjatidan olinishi mumkin bo'lgan juda ko'p ma'lumotlarni (bo'sh ma'noda) tavsiflaydi va matn maydonlarining har qanday belgilariga murojaat qilish mumkin.

XML Ism maydonlari - bu XML spetsifikatsiyasi bilan uzviy bog'liq bo'lgan juda muhim spetsifikatsiya. XML Ism maydonlari XML hujjatlarda nomlarni birlashtirish va nomlarni turli xil dastur modullari o'rtasida ajratish (juda ko'p bekor qilish o'rniga) juda muhim vazifani hal qiladi. Ismlar bo'shliqlari, shuningdek, nomlarning to'qnashuvi muammosini hal qilish uchun mo'ljallangan, chunki XML hujjatlarida odam o'qiydigan teglardan foydalanish rag'batlantiriladi va XML hujjatlarining turli sohalarida (va hatto bitta maydon ichida) bir xil nomlarning muqarrar ravishda turli xil ma'nolar bilan paydo bo'lishi aniq.

Nom maydonlari buni har bir nomni noyob resurs identifikatori (Unified Resource Identifier) bilan taqqoslash orqali amalga oshiradi. Biroq, nomlarda taqiqlangan belgilar URI-da ishlatilishi mumkinligi sababli, URI-ga haritaga kiritilgan qisqartirilgan ismlar bo'shlig'i prefiksi kiritiladi. Prefiks va URI havolasi o'rtasidagi bog'liqlik atribut sintaksisiga asoslangan hujjatning o'zida tasvirlangan.

XML Ismlar bo'shliqlari W3C tavsiyalarining eng yuqori darajasiga ega, shuning uchun barcha boshqa tavsiflar o'nga asoslanishi kerak.

So'rov standartlari.Aslida, XML hujjatidan ma'lumotlarni olishni tavsiflovchi standart allaqachon mavjud. Bu boshqa W3C standartlaridan (XSL, XPointer, XLink), shuningdek, deyarli barcha mavjud XML so'rovlari tillarini hujjatning alohida elementlariga kirish uchun asosiy texnologiya sifatida ishlatadigan XPath tavsiyalari. W3C a'zolaridan biri aytganidek: "... XPath-ga asoslanmagan XML ma'lumotlarini so'rovni yaratish yo'lini tasavvur qilish qiyin."

Rasmiy ravishda, XPath XML hujjatining alohida qismlariga murojaat qilish (ta'kidlash) tili sifatida aniqlanadi va uni qo'llashning asl sohasi XSLT va XPointer protsessorlari bo'lgan.

XPath ifoda sintaksisi hujjat tuzilishini yog'ochdan aks ettirishni nazarda tutadi. Biroq, bundan tashqari, hujjatning to'liq manzili, ya'ni. Veb-saytning butun hujjat daraxti XML hujjati sifatida harakat qilishi mumkin. Bu XPath-da Internetdagi XML hujjatlar to'plamiga murojaat qilish imkoniyatini yaratadi, chunki XPath veb-sayt darajasida va alohida XML hujjat darajasida hujjatlar va elementlarning yagona izchil namoyishini ifoda etadi.

Shunday qilib, XPath hujjatni tugunlardan tashkil topgan daraxt turidagi struktura sifatida ko'rib chiqadi. XPath ma'lumotlar modeli ettita turli xil tugunlarni (elementar, atribut, matn va hk) o'z ichiga oladi, ularning har biri to'g'ridan-to'g'ri tugun ichida joylashgan yoki avlodlarning qiymati bo'yicha hisoblangan. Ba'zi tugun turlari nomga ega. XPath Ismlar maydonini to'liq qo'llab-quvvatlaydi. Shunday qilib, xost nomi mahalliy nom va ehtimol kengaytirilgan nom deb nomlangan to'liq URIdan iborat juftlik sifatida modellashirilgan. XPath shuningdek, InfoSet ma'lumotlar modelida XPath ma'lumotlar modelini haritasi bor degan ma'noda InfoSet-ni qo'llab-quvvatlaydi.

Kontseptual darajada tugun mustaqil shaxs sifatida hujjat bo'lishi mumkin, har qanday hujjat yoki atribut, ishlov berish bo'yicha ko'rsatma yoki sharh. Hujjatning o'zi ildizning to'g'ridan-to'g'ri avlodlari bo'lgan tugunlarning ro'yxatini o'z ichiga oladi. O'z navbatida, bu tugunlar, shuningdek, tugunlarning ro'yxatlarini va boshqalarni o'z ichiga olishi mumkin. Ushbu universal yondashuv XPath-ni XML hujjati (yoki XML-sayt elementlari) elementlariga kirishni ta'minlash uchun oddiy, ammo kuchli vositaga aylantiradi. E'tibor bering, XPath faqat jismoniy saqlash formatini emas, balki faqat hujjatning mantiqiy tuzilishini tavsiflaydi.

XML hujjatining ierarxik tuzilishi va takrorlanadigan elementlar tufayli, XML ma'lumotlariga kirish mexanizmi ierarxik aloqalarni, ketma-ketlik aloqalarini va elementlarning joylashishini aniqlashni qo'llab-quvvatlashi kerak. XPath ushbu munosabatlarga o'tish uchun mos kvalifikatorlarga ega. Ularning to'liq ro'yxati va barcha funktsiyalar ro'yxati [3] da keltirilgan. XPath turli xil sharoitlarda, taqqoslash va o'zgartirishlarda sinovlarni bajaradigan bir qator funktsiyalarga ega. Xususan, XPath to'rt xil ma'lumotlarni qo'llab-

quvvatlaydi: tugunlar ro'yxati, simli, raqamli va mantiqiy. XML hujjatining tarkibi harakterli ma'lumotlar bo'lganligi sababli, XPath fu bilan ta'minlaydi.

So'rovlar tillari. Tillar turli xil yaratish maqsadlariga ega, har xil ekspressiv kuchlarga ega, ammo tadqiqotchilar fikriga ko'ra [5.16], ularning har biri muayyan turdagi so'rovlarni bajarishda afzalliklarga ega.

Yarim tuzilgan ma'lumotlar va XML uchun so'rovlar tillariga ikkita asosiy yondashuv mavjud, xususan: hujjatlarni qayta ishlash va ma'lumotlar bazasi nuqtai nazaridan. Turli xil qarashlar so'rovlar va turli xil funksiyalarni qurish uchun turli xil printsiplarga olib keladi (shuni ta'kidlash kerakki, tadqiqotchilar jamoasi XML ma'lumotlari bo'yicha so'rovlar tillariga yagona talablar qo'yadilar [5.13] va ishlab chiquvchilar o'zlarining quyi tizimlarining yangi versiyalarida to'liq funksional imkoniyatlarni berishga intilishadi).

Shunday qilib, tadqiqotning ikkita yo'nalishi turli xil sohalarga ega. Ma'lumotlar bazasi hamjamiyati so'rovlarni qayta ishlash nuqtai nazaridan ishlaydi. Tizimlarning ko'p turlari mavjud: relyatsion, ierarxik, tarmoq, deduktiv, obyektga yo'naltirilgan, yarim tuzilgan va boshqalar, ammo ko'pgina printsiplar barcha turlarga xosdir. Ushbu hamjamiyatning vazifalari katta ma'lumotlar omborxonalariga, turli xil manbalardan integratsiyaga, eski tizimlardan ma'lumotlarni eksport qilishga va umumiy qabul qilingan ma'lumotlar almashish formatlariga o'tishga qaratilgan.

Boshqa tomondan, hujjatlarni qayta ishlash sohasidagi tadqiqotchilar hamjamiyati katta hajmdagi hujjatlarni to'liq matnli qidirish, tuzilgan hujjatlar bo'yicha so'rovlar, to'liq matnli va tarkibiy so'rovlarni birlashtirish va chiqish shakllarini turli xil taqdimotlari uchun hujjatlarni o'zgartirish bilan shug'ullanadi

Birinchi avlod tillari

LoREL

LoREL tili [4] dastlab Stanford universitetida LoREL yarim tuzilgan ma'lumotlarni boshqarish tizimi loyihasi doirasida ishlab chiqilgan va keyinchalik XML bilan ishlashga o'zgartirilgan. Bu OQL-ga qaraganda ko'proq xususiyatlarga ega, ammo sodda SQL uslubidagi sintaksis bilan do'stona til. LoRELning asosiy afzalliklari va yangiliklari

- bu turdagi konversiyani keng qo'llash va umumiy yo'lni ifodalash deb nomlangan qattiq bo'lmagan ma'lumotlar strukturasi orqali so'rov yo'lini qurish uchun kuchli mexanizm. Lorel o'ziga xos ma'lumotlar modelini [7] amalga oshiradi, bu obyekt ma'lumotlari modelining kengaytmasi (ODMG) bo'lib, XML ma'lumotlari elementlariga mos keladigan tugunlari bilan grafik namoyishi mavjud. Lorel barcha bir nechta operatsiyalarni qo'llab-quvvatlaydi, tanlangan ma'lumotlarning manba hujjatlarida taqdim etilishida tartiblash yoki saralashni ta'minlaydi. Bu bugo'ngi kunda foydalanuvchi tomonidan belgilangan ma'lumotlar turlariga ruxsat beradigan va ma'lumotlarni o'zgartirish va yo'q qilish uchun tuzilmalarni o'z ichiga olgan bir nechta tillardan biridir.

Lorel tilining kuchli tomonlaridan biri shablonlardan foydalangan holda "noaniq" sxemalar bo'yicha navigatsiya qilishning boy imkoniyatidir, bundan tashqari shablonlarni alohida qiymatlar yoki model grafikasida qirralarning nomlari bo'yicha ham, yo'llar bo'yicha ham aniqlash mumkin (chekka nomlarning variantlari, havolaning majburiy / ixtiyoriy mavjudligi) yo'llar, aniqlanmagan yo'nalishdagi ulanishlar va boshqalar). Sintaksis tafsilotlariga kirmasdan, yo'llarda naqshlardan foydalanishga misol.

Hidoyat.restaurant (.address)?. Zipkod

Qo'llanma.restaurant.#@P.comp%.name

Hidoyat.restaurant (.nearby) * {R} .name

Birinchi ibora zipcode va restoran o'rtasidagi ixtiyoriy qirrali yo'lni belgilaydi. Ikkinchisi restoran va chekka o'rtasida joylashgan, noma'lum sonli chekkalari bilan barcha yo'llarni belgilaydi (belgi #) va substringdan boshlab chekka. Yo'l o'zgaruvchisi? #? Mezoniga javob beradigan barcha ma'lumotlar bilan bog'liq. Oxirgi ibora restoranning chetidan o'tadigan barcha yo'llarni, so'ngra noma'lum tugagan yaqin atrofdagi va cheklanmagan chekkalarni belgilaydi. Obyekt o'zgaruvchisi R obyekt bilan darhol chekka nomidan oldin bog'lanadi.

Tilning asosiy xususiyatlari:

- Yangi XML elementini yaratish uchun konstruktsiyalar.
- Bitta hujjat ichida ham, bir nechta hujjatlar orasidagi ulanishlar (qo'shilish).
- Bir nechta operatsiyalar - birlashma, kesishish, farq.

- Umumiylik, mavjudlik va inkor predikatlarida osilgan hisoblagichlar.
- Barcha asosiy turdagi agregatlarni qo‘llab-quvvatlash.
- Ichki so‘rovlarni qayta ishlash.

Internet rivojlanishi bilan turli xil manbalardan ma'lumotlarni birlashtirish va ularni turli formatlarda taqdim etish vazifasi ma'lumotlar bazasi hamjamiyatining asosiy faoliyatlaridan biriga aylandi. Ushbu muammoni hal qilishning eng keng tarqalgan yondashuvlaridan biri bu vositachilar va qadoqlovchilar yoki o‘rashchilar kontsepsiyasiga asoslangan virtual ma'lumotlar ombori [6,7]. O‘z navbatida, vositachilar va paketlovchilarning arxitekturasini rivojlantirish uchun ma'lumotlar almashish muammosini hal qilish kerak, bu esa ko‘plab ma'lumotlar bazalari sohasidagi tadqiqotchilarning sa'y-harakatlari yo‘naltirilgan.

INRIA vositachilar va paketlovchilar uchun asos bo‘lib xizmat qiladigan, istiqbolli nomi YAT (Yana bir daraxtga asoslangan tizim) bilan prototip tizimini yaratdi va ishlab chiqdi, ya'ni ma'lumotlarni o‘zgartirish muammosini hal qilish uchun [8]. YAT heterojen manbalar orqali transduserlarni aniqlash va joylashtirish uchun vositalar to‘plamini taqdim etadi.

Ko‘pgina boshqa tizimlar singari, YAT ham ma'lumotlar uchun grafik ko‘rinishni qo‘llaydi

Nazorat savollari

1. XML ning vazifasi nima?
2. Ma'lumotlar markazlashtiruvchi hujjatlar
3. XML hujjatining tuzilishi ma'lumotlar bazasining tuzilishidan nimasi bilan farq qiladi?
4. Shablon asosida so‘rovlar tillariga misol keltiring?
5. Tuzilmaviy indekslar nima?

Glossariy

Termin	O'zbek tilidagi sharhi	Ingliz tilidagi sharhi
SQL	Strukturalangan so'rovlar tili	Structured Query Language
Access	Microsoft Office RMBBT dasturi	The Microsoft Office RDBMS application
Append	Jadval oxirigacha ma'lumotlarni qo'shish	Adding data to the end of table
Ascending order	Eng past va eng yuqori uchun sanada asoslangan matn sohasida alifbo tartibi	In order from lowest to highest. Also called alphabetical order, when a sort is based on a text field, and chronological, when a sort is based on a date field
Autonumber field	Yozishga qaraganda katta maydonga qo'shimcha ravishda avtomatik saqlash	A field that automatically stores a numeric value, that is one greater than that in the last record added
Database	Tegishli ma'lumotlarni yig'ish	An organized collection of related data
Database schema	Ma'lumotlar bazasida jadvallar yacheykasiga ma'lumotlarning bayoni va ma'lumotlarni uzatishni tashkil etish	A description of the data, and the organization of the data, into tables in a relational database

Datasheet	Ma'lumotlar uchun satrlar ustunlar sohalarda va yozuvlar bilan tashkil etish	The data for a table organized with fields in columns and records in rows
Datasheet view	Satrlar ustunlar sohalarda va yozuvlar bilan, bir ma'lumot sahifasida bir stol asosiy tuzilishini ko'rish uchun ishlatiladi	Used to display the basic structure of a table in a datasheet, with fields in columns and records in rows
Entry	Jadval uchun ma'lumotlar	The data for a field
Field	Jadvaldagi maydonlarni belgilaydi	A column in a table. Used to store data
OLAP	Хақиқий вақтда маълумотларга аналитик ишлов бериш	On-Line Analytical Processing
OLTP	Хақиқий вақтда транзакцияларга ишлов бериш	On-Line Transaction Processing
Form	So'rovlar yordamida ma'lumotlarni ko'rishda ishlatiladi	A database object used for entering records into a table, and for viewing existing records
Long integer	Uzun butun toifa	A field size that indicates a whole number
Lookup field	Maydondagi ma'lumotlarni saqlaydi	A field that stores data; retrieved from a field in another table
Name	Jadval nomi bo'lib,	Word or words,

	soz orqali ifodalanadi	used to describe the data stored in a field
Primary key	Birlamchi kalit hisoblanadi	A field in a table that is designated to contain unique data.
RDBMS	Relatsion ma`lumotlar bazasini boshqarish tizimi	(Relational Database Management System) A software application that contains tools to manage data, answer queries, create user-friendly forms for data entry, and generate printed reports.
Date/time field	Maydon sana yoki vaqtni saqlaydi	A field that stores a date or time
Decending order	Oliy maqsadidan eng past uchun	In order from highest to lowest
Design view	Jadvallar uchun maydon ta'riflar ko'rsatadi	The table view that shows the field definitions for a table
Yes/No field	ha / yo'q, to'g'ri / noto'g'ri, yoki / off vakillik qilish.	A field that is either selected or not selected to represent yes/no, true/false, or on/off.
ERP	Korxonalar resurslarini rejalashtirish	Enterprise Resource Planning
Record	Jadvaldagi maydonlarni uchun ma'lumotlar majmui	A set of data for fields in a table
Updating	Yozuvni	Modifying a record

	o`zgartirish	
Table	Ma'lumotlar bazasi obyekti. Satr va ustunlar ichiga tashkil etilgan tegishli ma'lumotlarni saqlaydi	A database object that stores related data organized into rows and columns.
Text field	Jadvallarda belgilar (harflar, belgilar, so'zlar, harflar va raqamlar kombinatsiyasini) hisob talab qilmaydigan va sonlar saqlaydi	A field that stores characters (letters, symbols, words, a combination of letters and numbers) and numbers that do not require calculations.
CRM	Mijozlar bilan o'zaro munosabatlarni boshqarish	Customer Relations Management
LAN	Lokal hisoblash tarmog'i	Local Area Network
MAN	Mahalliy hisoblash tarmog'i	Metropolitan Area Network
WAN	Hududiy hisoblash tarmog'i	Wide Area Network
ISO	Xalqaro standartlashtirish tashkiloti	International Organization for Standardization
WWW	Umumjahon o'rgimchak to'ri	World Wide Web
ASCII	Axborot almashishning Amerika standarti	American Standard Code for Information Interchange

Adabiyotlar ro`yxati

1. В.П. Базы данных. Книга 2 распределенные и удаленные базы данных: учебник.// Москва ИД «ФОРУМ» - ИНФРА-М. – 2018. – С 261.
2. Голицына О.Л. Базы данных: учеб. Пособие // – 4-е изд., перераб. И доп. – М.: ФОРУМ: ИНФРА-М, 2018. – 400 с.
3. Мартишин С.А. Базы данных. Практическое применение СУБД SQL –и NoSQL – типа для проектирования информационных систем: учеб. Пособие // - Москва: ИД «ФОРУМ» - ИНФРА-М, 2019, – 368 с.
4. Rahul Batra. SQL Primer An Accelerated introduction to SQL Basics.// Gurgaon, India. 2019. –P 194.
5. Поликов А.М. Безопасность Oracle глазами аудитория: нападение и защита. –Москва. 2017. –336 с.
6. Usmonov J.T., Xujaqulov T.A. Ma'lumotlar bazasini boshqarish tizimi// o`quv qo`llanma. - T. : Aloqachi, 2018. – 96 b.
7. Usmonov J. T., Xo`jaqulov T. A. Ma'lumotlar bazasini boshqarish tizimi fanidan laboratoriya ishlarini bajarish bo'yicha uslubiy ko'rsatma - T. : TATU, 2016. – 55 b.
8. Eric Redmond, Jim R. Wilson. A Guide to Modern Databases and the NoSQL MovementAQSH, 2015. – 347 с.
9. Elmasri, R., S. B. Navathe: Fundamentals of Database Systems (5th Ed.)// Addison Wesley, 2015. – 671 p.

Qo'shimcha adabiyotlar

1. Ўзбекистон Республикаси Президентининг 2017 йил 7 февралдаги ПФ-4947-сон “Ўзбекистон Республикасини янада ривожлантириш бўйича Ҳаракатлар стратегияси тўғрисида”ги Фармони.
2. Мирзиёев Ш.М. Буюк келажакимизни мард ва олижаноб халқимиз билан бирга кураимиз. Тошкент. «Ўзбекистон», НМИУ, 2017. – 488 б.
3. Fundamentals of database systems sixth edition. Ramez Elmasri. Department of Computer Science and Engineering The University of Texas at Arlington. 2011. – 261 с.
4. Введение в Oracle 10g. Перри Джеймс, Пост Джеральд. 697 стр 2013

5. Диго С.М. Базы данных Проектирование и использование. издательство "Финансы и статистика". 2005 г. – 592 с.

6. Конноли Т., Брегк К. Базы данных, проектирование, реализация и сопровождения, теория и практика, Университет Пейсли, Шотландия, изд. М.- СПб.- Киев. 2003. – 264 с.

Internet saytlari

1. www.intiut.ru;
2. www.oracle.com
3. www.library.tuit.uz;
4. www.intuit.ru;
5. www.w3school.com;
6. www.ziyonet.uz;

Mundarija

Kirish	3
1-BOB. Ma'lumotlar bazasining asosiy tushunchalari	4
1.1.Ma'lumotlar bazasining maqsadi, vazifalari va asosiy tushunchalari.....	4
1.2.Ma'lumot bazasi tizimining arxitekturasi va uch bosqichli arxitektura	18
2-BOB. Ma'lumotlar bazasi modellari va normal formalar	27
2.1.Ma'lumotlar bazasi modellari va mohiyat-aloqa modeli	27
2.2. Relyatsion ma'lumot bazasi va ma'lumotlar bazasida munosabatlar.....	32
2.3. Relyatsion algebra va relyatsion hisobot elementlari	44
2.4. Ma'lumotlar bazasini rejalashtirish, loyihalash va administratorlash.....	47
2.5. Ma'lumotlar bazasini normallashtirish: 1NF, 2NF, 3NF va Kodd normal formalari	51
3-BOB. SQL tili orqali ma'lumotlar ustida amallar bajarish	54
3.1. SQL tili va SQL operatorlarini yozish	54
3.2. Ma'lumotlar manipulyatsiya qilishda oddiy so'rovlar yaratish.	61
3.3. SQLtili yordmida ma'lumotlarni tavsiflash	72
3.4. SQLda jarayonlar va standart funksiyalar.....	81
3.5. Tranzaksiyalarni boshqarishda so'rovlar yaratish va qayta ishlash..	90
4-BOB. Ma'lumotlar bazasiga murojaatni amalga oshirish	100
4.1. Ma'lumotlar bazasini administratorlash va xavfsizligini ta'minlash.....	100
4.2. Ma'lumotlar bazasiga murojaatni tashkil etishda ODBC va obyektga yo'naltirilgan dasturlar foydalanish.....	110
4.3. XML va ma'lumotlar bazalari	118
Glossariy	133
Adabiyotlar ro'yxati.....	137

R.V.QOBULOV, B.B.AKBARALIYEV,
J.T.USMONOV, O.B.RO'ZIBAYEV

MA'LUMOTLAR BAZASI

(O'quv qo'llanma)

Toshkent – «Aloqachi» – 2021

Muharrir: A.Tog'ayev
Tex. muharrir: P.Tog'ayeva
Musavvir: B.Esanov
Musahhiha: G.Tog'ayeva
Kompyuterda
sahifalovchi: B.Berdimurodov

Нашр. лиц. ii № 176, 11.06. 2010.
Bosishga ruxsat etildi: 29.01.2021. Bichimi 60x841 /16.
Shartli bosma tabog'i 9,0. Nashr bosma tabog'i 8,75.
Adadi 100. Buyurtma № 1.

«Nihol print» Ok da chop etildi.
Toshkent sh., M. Ashrafiy ko'chasi, 99/101.