



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

BUXORO MUHANDISLIK-TEXNOLOGIYA INSTITUTI

**«ELEKTROTEXNIKA VA ISHLAB CHIQRISHDA AXBOROT
KOMMUNIKASIYA TEXNOLOGIYALARI» FAKULTETI**

**”AXBOROT TEXNOLOGIYALARI”
KAFEDRASI**

“WEB – DASTURLASH”

fanidan

REFERAT

Mavzu: Mysql - ma`lumotlar omborini boshqarish tizimi

Bajardi:

**12-12 MIAT guruhi talabasi
Tursunov Otabek**

Qabul qildi:

Sariyev R.B.

BUXORO – 2016

MYSQL - MA`LUMOTLAR OMBORINI BOSHQARISH TIZIMI

- 1. MySql ga kirish.**
- 2. MOBT**
- 3. Ma`lumotlar ombori kontceptciyalari**
- 4. Tarmoqli ma`lumotlar ombori**
- 5. MySql asosiy xarakteristikolari**
- 6. Ma`lumotlar omborini yaratish, jadvallar bilan ishlash asoslari**
- 7. Foydalanilgan adabiyotlar**

MySql ga kirish.

Hozirgi kunda ma`lumotlar ombori haqida tez-tez gapirilmoqda. Kompyuterlar zamonaviy jamiyatning ajralmas qismini tashkil qiladi, shu sababli quyidagiga o`xshash iboralarni ko`p eshitish mumkin «Men yozuvlaringni ma`lumotlar omboridan qidirib ko`raman». Bu erda hujjatlar saqlanadigan katta qutilar emas, balki ma`lumotlarni tezkor qidirishga mo`ljallangan kompyuter tizimlari nazarda tutiladi.

Kompyuterlar hayotimizga chuqurroq kirib bormoqda, chunki ularni tez takrorlanadigan operatsiyalarni bajarish yoki kompyuterning hisoblash tezligi va ma`lumot tashish sig`imisiz echib bo`lmaydigan masalalarning echimini topishga dasturlash mumkin. Ma`lumotni qog`ozga joylashtirish va qog`ozlarni jild va kartotekalarda saqlash sxemasini ishlab chiqish – aniq ishlangan jarayon bo`lsada, qattiq diskdagi elektron hujjatlar jildlarining ko`chirilishi mumkinligi ko`pchilikka ma`qul keldi.

Ma`lumotlar omborining funktsiyalaridan biri ma`lumotni tartiblash va indeksatsiya qilish hisoblanadi. Bu kutubxona kartotekasi singari, faqat kerakli yozuvni topish uchun arxivning yarmini ko`rib chiqish shart emas. hammasi ancha tez bajariladi.

Barcha ma`lumotlar ombori ham bir xil printcip asosida yaratilmaydi, lekin an`anaviy tarzda ularda yozuvlar ko`rinishida ma`lumotlarni tashkil qilish g`oyasi qo`llaniladi. Har bir yozuv belgilangan maydonlar to`plamiga ega. Yozuvlar jadvalga joylashtiriladi, jadvallar yig`indisi esa ma`lumotlar omborini tashkil qiladi.

Ma`lumotlar ombori bilan ishlaganda MOBT (ma`lumotlar omborini boshqarish tizimlari), ya`ni ma`lumotlarga kirish bilan bog`liq barcha ishlarni o`z zimmasiga oluvchi dastur zarur. Unda jadvallar yaratishga imkon beruvchi buyruqlar mavjud, jadvallarga yozuvlar kiritish, qidirish va hatto yozuvlarni o`chirishi xam mumkin.

MySql – bu tezkor, ishonchli, ochiq tarqatiladigan MOBT hisoblanadi. **MySql** boshqa ko'pchilik MOBT lar singari, «mijoz/server» modeli asosida ishlaydi. Bunda kompyuterlar mijoz yoki server rolini o'ynaydigan tarmoqli arxitektura tushuniladi. 1.1 rasmda mijoz kompyuteri va serverning qattiq diski o'rtasida ma'lumot uzatish sxemasi ko'rsatilgan.

1.1 rasm. "Mijoz/server" arxitekturasida ma'lumotlarni uzatish sxemasi.

MOBT bir yoki bir nechta ma'lumotlar omborini boshqaradi. Ma'lumotlar ombori ko'plik shaklida tashkil qilingan axborot yig'indisidir. Har bir ko'plik o'zida unificirlangan yozuvni saqlaydi. Yozuvlar esa maydonlardan tashkil topgan. Odatda ko'pliklar jadvallar deb yuritiladi, yozuvlar esa – jadval satrlari.

Ma'lumotlarning mantiqiy modeli shunday tashkili qilingan. Barcha ma'lumotlar ombori qattiq diskda bitta faylda saqlanishi mumkin. **MySql** da har bir ma'lumotlar ombori uchun alohida katalog yaratiladi, har jadvalga esa uchta fayl to'g'ri keladi. Boshqa MOBT larda ma'lumotlarni jismoniy saqlashning boshqacha printsiplari qo'llanilishi mumkin.

Jadval satrlari o'zaro quyidagi uch usuldan biri bilan bog'lanishi mumkin. Eng oddiy munosabat – “birga bir”. Bunday holda birinchi jadval satri ikkinchi jadvalning bitta yagona satriga mos keladi. Diagrammalarda bunday munosabat 1:1 yozuvi bilan ifodalanadi.

“Ko'pga bir” munosabati bir jadvalning satri boshqa jadvalning bir nechta satriga mos keluvchi vaziyatni anglatadi. Bu munosabatlarning eng ko'p tarqalgan turidir. Diagrammalarda 1:N yozuvi bilan ifodalanadi.

Nihoyat, “ko'plikka ko'plik” munosabatida birinchi jadval satrlari ikkinchi jadvaldagi ixtiyoriy sonli satrlari bilan bog'lanishi mumkin. Bunday munosabat N:M ko'rinishida yoziladi.

MOBT

Ma'lumotlar ombori bilan ishlaydigan dasturchi ushbu ma'lumotlar qanday saqlanishi haqida bosh qotirmaydi, hamda MOBT bilan o'zaro bog'liq holda ishlovchi ilovalar, ma'lumotlarni diskka yozish usuli haqida bilishmaydi.

“Tashqaridan” faqatgina ma`lumotlarning mantiqiy obrazi ko`rinadi, bu esa ilovalar kodiga tegmasdan MOBT kodini o`zgartirishi imkonini beradi.

Ma`lumotlarni bunday qayta ishlash to`rtinchi avlod tili (4GL) yordamida amalga oshiriladi, bu til darhol yoziladigan va bajariladigan so`rovlarni qo`llaydi. Ma`lumotlar dolzarbligini tez yo`qotadi, shuning uchun ularga kirish tezligi muhimdir. Bundan tashqari, dasturchi yangi so`rovlarni ishlash imkoniga ega bo`lishi zarur. Ular reglamentirlanmagan (ad hoc) deyiladi, chunki ma`lumotlar omborining o`zida saqlanmaydi va tor mutaxassislik maqsadlariga xizmat qiladi.

To`rtinchi avlod tili sxemalar – ma`lumotlar va ular orasidagi munosabatlarni aniqlashlarni yaratishga imkon beradi. Sxema ma`lumotlar omborining bir qismi sifatida saqlanishi va ma`lumotlarga zarar etkazmagan holda o`zgartirilishi mumkin.

Sxema ma`lumotlar butunligini nazorat qilish uchun mo`ljallangan. Agar, misol uchun, maydon butun sonli qiymatlarni saqlaydi deyilgan bo`lsa, u holda MOBT unga vergulli sonlar yoki satrlarni yozmaydi. Yozuvlar orasidagi munosabatlar ham qattiq nazorat qilinadi va kelishilmagan ma`lumotlar kiritilmaydi. Operatciyalarni “hammasi yoki hech narsa” printcipi bo`yicha bajariladigan tranzaktciyalarga guruhlash mumkin.

MOBT ma`lumotlar xavfsizligini ta`minlaydi. Foydalanuvchilarga axborotga kirish uchun ma`lum huquqlar beriladi. Ayrim foydalanuvchilarga axborotlarni faqat ko`rishga ruxsat beriladi, boshqalari esa jadval tarkibini o`zgartirishlari mumkin.

MOBT ma`lumotlar omboriga parallel kirishni qo`llaydi. Ilovalar ma`lumotlar omboriga bir vaqtda murojaat qilishi mumkin, bu esa tizimning umumiy samaradorligini oshiradi. Bundan tashqari, samaradorlikni yanada oshirish uchun alohida operatciyalar “parallellashi” mumkin.

Nihoyat, MOBT ko`zda tutilmagan uzilishlar yuz berganda axborotni qayta tiklash maqsadida foydalanuvchilarga sezdirmay ma`lumotlarning zahira nusxalarini olib qo`yadi.

Ma`lumotlar ombori kontseptciyalari

Fayllarni boshqarish tizimlari

Eng oddiy ma`lumotlar ombori odatiy fayllar to'plami ko'rinishida tashkil qilingan. Ushbu model qutilarda saqlanadigan jildlar hamda bu jildlarga tikib qo'yilgan bir nechta sahifali hujjatlarning kartotekali tashkil qilinishini eslatadi.

Fayllarni boshqarish tizimlarini MOBT singari toifalash mumkin emas, chunki odatda ular operatsion tizimning bir qismi hisoblanadi va fayllarning ichki tarkibi haqida hech narsa bilishmaydi. Bu narsa fayllar bilan ishlovchi amaliy dasturlarga kiritilgan. Misol sifatida /etc/passwd faylida saqlanuvchi UNIX foydalanuvchilar jadvalini keltirish mumkin. Ushbu faylga murojaat qiluvchi dasturlar uning birinchi maydonida ikki nuqta bilan tugallanadigan foydalanuvchi nomi mavjudligini bilishadi. Agar ilova ushbu axborotni tahrirlashi kerak bo'lsa, u to'g'ridan to'g'ri faylni ochishi va maydonlarni to'g'ri formatlash haqida qayg'urishi kerak.

Ma`lumotlar omborining bunday modeli juda noqulay, chunki u uchinchi avlod tili (3GL) ni qo'llashni talab qiladi. Natijada so'rovlarni dasturlash vaqti oshadi, dasturchi esa yanada yuqori kvalifikatsiyaga ega bo'lishi kerak, chunki u ma`lumotlarni saqlashning nafaqat mantiqiy, balki jismoniy saqlanish strukturasi ham o'ylab chiqishi kerak. Bu shunga olib keladiki, ilova va fayl orasida mustahkam aloqa yuzaga keladi. Jadvallar maydonlari haqidagi barcha informatsiya ilovada kodlashtirilgan. Shu faylga murojaat qiladigan boshqa ilova mavjud kodni ikkilashi kerak.

Ilovalar soni oshishiga qarab ma`lumotlar omborini boshqarish murakkabligi ham ortib boradi. Ma`lumotlar sxemasi o'zgarishi har bir dastur komponentiga, agar uning uchun bu zarur bo'lsa, o'zgartirish kiritishga olib keladi. Yangi so'rovlarni shakllantirish shunchalik ko'p vaqtni oladiki, ba`zan bu ma`noga ega bo'lmaydi.

Fayllarni boshqarish tizimlari axborotni nusxasini ko'paytirishga to'sqinlik qila olmaydi. Yanada yomon, ma`lumotlar kelishilmaganligining oldini oluvchi mexanizmlarning mavjud emasligidir. Faraz qiling, kompaniyaning barcha xodimlari to'g'risida fayl bor. Har bir satrda boshliq nomi yozilgan maydon

mavjud. Bir boshliq qo'l ostida bir nechta ishchi ishlaydi, shu sababli uning nomi doim takrorlanadi. Agar qaerdadir bu ism noto'g'ri yozilgan bo'lsa, rasman bu xodimning boshqa boshlig'i bo'ladi. Boshliqni almashtirganda uning ismini butun ma'lumotlar ombori bo'ylab qidirish kerak bo'ladi.

Oddiy fayllar xavfsizligi operatsion tizim orqali nazorat qilinadi. Alohida fayl u yoki bu foydalanuvchi tomonidan ko'rish yoki modifikatsiya qilish uchun bloklangan, ammo bu faqat operatsion tizim darajasida bajariladi. Ma'lum bir vaqtda faqat bitta ilova faylga yozuv kiritishi mumkin, bu esa umumiy samaradorlikni pasaytiradi.

Tarmoqli ma'lumotlar ombori

Tarmoqli model ko'plik yozuvlar orasidagi aloqani guruhlashga imkon beruvchi ierarxik modelni kengaytiradi. Mantiqiy jihatdan olib qaralganda, aloqa – bu yozuvning o'zi emas. Aloqalar faqatgina yozuvlar orasidagi munosabatni bildiradi. Ierarxik modeldagi singari, aloqalar ona yozuvdan qiz yozuvga olib boradi, lekin bu gal ko'plik qo'llab quvvatlanadi.

CODASYL spetsifikatsiyasiga mos holda, tarmoqli model DDL (Data Definition Language — ma'lumotlarni aniqlash tili) va DML (Data Manipulation Language — ma'lumotlarni qayta ishlash tili) ni qo'llaydi. Bu maxsus tillar bo'lib, ma'lumotlar ombori tuzilishini aniqlash va so'rovlar tuzish uchun mo'ljallangan. Ularning mavjudligiga qaramasdan, dasturchi avvalgidek ma'lumotlar ombori tuzilishini bilishi lozim.

Tarmoqli modelda “ko'plikka ko'plik” munosabati ruxsat etiladi, yozuvlar esa bir biriga bog'liq emas. O'chirilganda yozuvlar va uning barcha aloqalari o'chiriladi, biroq bog'liq yozuvlarning o'zi o'chirilmaydi.

Tarmoqli modelda butunlikni saqlash va nusxalashga yo'l qo'ymaslik maqsadida aloqalarning mavjud yozuvlar orasida o'rnatilishi talab qilinadi. Ma'lumotlarni mos jadvallarda izolyatsiya qilishva boshqa jadvallardagi yozuvlar bilan bog'lash mumkin.

Dasturchi ma'lumotlarni diskda qanday saqlanishi haqida qayg'urishi shart emas. Bu ilovalar va ma'lumotlar tobeligini susaytiradi. Ammo tarmoqli modelda

dasturchi so'rovlarni shakllantirayotganda ma'lumotlar strukturasi yodda tutishi talab qilinadi.

Ma'lumotlar omborining optimal strukturasi shakllantirish oson emas, tayyor tuzilmani almashtirish esa qiyin. Agar jadval ko'rinishi ko'p o'zgartirilsa, boshqa jadvallar bilan barcha bog'liqliklar ma'lumotlar butunligini buzilmasligi uchun qaytadan o'rnatilishi kerak. Ushbu masalaning murakkabligi shunga olib keladiki, dasturchilar ilovalarni soddalashtirish maqsadida butunlikning ba'zi cheklovlarini inkor etadilar.

MySql asosiy xarakteristikalari

MySql mijoz dasturi o'zida buyruq satrning utilitasini namoyon qiladi. Ushbu dastur serverga tarmoq bo'yicha ulanadi. Server tomonidan bajariladigan buyruqlar odatda qattiq diskdagi ma'lumotlarni o'qish va yozish bilan bog'liq.

Mijoz dasturlar nafaqat buyruq satri rejimida ishlaydi. Grafik klientlar ham mavjud, masalan **MySql** GUI, PhpMyAdmin va b. Ammo ular – boshqa kurs mavzusi.

MySql ma'lumotlar ombori bilan SQL (Structured Query Language – tuzilmaviy so'rovlar tili) deb nomlanadigan tilda harakat qiladi.

SQL relyatcion ma'lumotlar omborini boshqarish tizimlari (RDBMS) da saqlanuvchi ma'lumotlarni manipulyatciya qilish uchun mo'ljallangan. SQL da ma'lumotlarni olish, tartiblash, yangilash, o'chirish va qo'shish imkonini beruvchi buyruqlar mavjud. SQL tili standartlarini ANSI (American National Standards Institute) aniqlaydi. Hozirgi vaqtda 2003 yilda qabul qilingan standart amal qiladi (SQL-3).

SQL ni RDBMS **MySql**, mSQL, PostgreSQL, Oracle, Microsoft SQL Server, Access, Sybase, Ingres kabilar bilan ishlatish mumkin. RDBMS ning bu tizimlarini SQL ning barcha muhim va umumiy qabul qilingan operatorlari qo'llaydi, ammo ularning har biri o'zining xususiy patentlangan operator va kengliklariga ega.

SQL turli tipdagi bir necha ma'lumotlar omborlari uchun so'rovlarning umumiy tili hisoblanadi. Mazkur kurs **MySql** tizimini ko'rib chiqadi, u

MySql.com saytida yuklash mumkin bo'lgan ochiq yakuniy kodli RDBMS hisoblanadi.

MySql ni uning yaratuvchilari quyidagicha izohlashadi.

- **MySql** – bu ma'lumotlar omborini boshqarish tizimi.

Ma'lumotlar ombori ma'lumotlarning tuzilmaviy yig'indisini bildiradi. Bu ma'lumotlar turlicha bo'lishi mumkin – sotib olinishi kerak bo'lgan mollar oddiy ro'yxatidan tortib suratlar galereyasidagi eksponatlar ro'yxati yoki korporativ tarmoqdagi ko'p miqdordagi axborotgacha bo'lishi mumkin. Kompyuterning ma'lumotlar omborida saqlanadigan ma'lumotlarni yozish, tanlash va qayta ishlash uchun **PO MySql** kabi ma'lumotlar omborini boshqarish tizimi zarur. Kompyuterlar katta miqdordagi ma'lumotlarni qayta ishlashni yaxshi uddalashini hisobga olsak, hisoblashlarda ma'lumotlar omborini boshqarish asosiy o'rinni egallaydi. Bunday boshqaruv turlicha amal qilishi mumkin – alohida utilita ko'rinishida yoki boshqa ilovalar tarkibiga kiruvchi kod ko'rinishida bo'lishi mumkin.

- **MySql** – bu relyatsion ma'lumotlar omborlarini boshqarish tizimi.

Relyatsion omborda ma'lumotlar alohida jadvallarda saqlanadi, shuning hisobidan tezlik va qayishqoqlikda yutuqqa erishish mumkin. Jadvallar o'zaro munosabatlar yordamida bog'lanadi, shuning hisobiga so'rovni bajarganda bir nechta jadvaldagi ma'lumotlarni birlashtirish imkoniyati ta'minlanadi. **SQL** ni **MySql** tizimining bir qismi sifatida quyidagicha xarakterlash mumkin: tuzilmaviy so'rovlar tili plus ma'lumotlar omboriga kirish uchun foydalaniladigan keng tarqalgan standart til.

- **MySql** dasturiy ta'minoti – bu ochiq kodli DT.

Ochiq kodli DT uni har bir kishi qo'llashi va modifikatsiya qilishi mumkinligini anglatadi. Bunday DT ni Internet dan olish va bepul ishlatish mumkin. Bunda har bir foydalanuvchi yakuniy kodni o'rganishi va uni o'zining ehtiyojlariga mos ravishda o'zgartirishi mumkin.

- **MySql** MOBT ning texnik imkoniyatlari

MySql DT mijoz – server tizimi hisoblanadi, uning tarkibida ko'p oqimli SQL-server mavjud bo'lib, bu server ma'lumotlar omborining turli hisoblash mashinalarini qo'llashni, shuningdek bir nechta turli xil mijoz dasturlar va kutubxonalar, ma'muriy vositalar va dasturiy interfeyslar (API) ning keng spektri ta'minlaydi.

- Xavfsizlik

Xavfsizlik tizimi chetdagi kompyuter verifikatsiyasi imkoni mavjud parol va imtiyozlarga asoslangan, buning hisobiga qayishqoqlik va xavfsizlik ta'minlanadi. Parollar tarmoq bo'yicha uzatilganda server bilan bog'langanda shifrlanadi. Mijozlar **MySql** ga TCP/IP, Unix soketlaridan yoki nomlangan kanallar (named pipes, pod NT) dan foydalanib bog'lanishlari mumkin.

- Ma'lumotlar sig'imi

Jadvallarning yangi tipini qo'llaydigan **MySql** 3.23 versiyasidan boshlab, jadvalning maksimal qiymati 8 million terabayt (2^{63} bytes) ga etgan. Lekin shuni ta'kidlash zarurki, operatsion tizimlar fayl razmerlariga o'z cheklovlarini qo'yadi. Quyida bir qancha misollar keltirilgan:

- 32-razryali Linux-Intel – jadval razmeri 4 Gb.
- Solaris 2.7 Intel - 4 Gb
- Solaris 2.7 UltraSPARC - 512 Gb
- WindowsXP - 4 Gb

Ko'rinib turibdiki, **MySql** ma'lumotlar omborida jadval razmeri odatda operatsion tizim bilan limitlanadi. Jimlik qoidasiga binoan, **MySql**-jadvallar taxminan 4 Gb ga teng maksimal razmerga ega. Ixtiyoriy jadval uchun uning maksimal razmerini aniqlash/tekshirish SHOW TABLE STATUS yoki myisamchk -dv table_name buyruqlari rdamida amalga oshiriladi. Agar katta jadval faqat o'qish uchun mo'ljallangan bo'lsa, bir nechta jadvalni birlashtirish va kichraytirish uchun myisampack dan foydalanish mumkin. Odatda myisampack jadvalni ko'pi bilan 50 % ga siqishi mumkin, shuning natijasida juda katta jadvallarni olish mumkin.

Ma'lumotlar omborini yaratish, jadvallar bilan ishlash asoslari

Ushbu ma`ruzada ma`lumotlar omborini yaratish usullari, jadvallar yaratish va ularni o`chirish buyruqlari ko`rib chiqiladi.

Ushbu ma`ruzada ma`lumotlar omborini yaratishni o`rganamiz.

Windows va Linux da ma`lumotlar omborini yaratish buyruqlari bir xil. Ammo Linux da oldindan bajariladigan buyruqlar murakkabroq.

BigFoot nomli biror kompaniyaning ishchilari haqidagi ma`lumotlardan tashkil topgan employees nomli ma`lumotlar omborini yaratamiz. Ushbu ma`lumotlar omborida ishchilarning ismi, familiyasi, maoshi, yoshi, manzili, e-mail, tug`ilgan sanasi, qiziqishlari, telefon raqamlari va boshqalarni saqlash ko`zda tutiladi.

Windows da ma`lumotlar omborini yaratish

1. **MySql** serverini ishga tushiring, buning uchun c:**MySql**\bin katalogida taklif satridagi **MySql**d-shareware –standalone buyrug`ini tanlang.

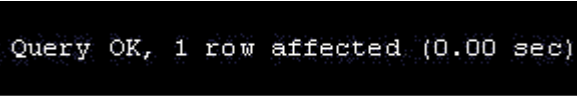
2. So`ngra taklif satrida **MySql** ni kiritish orqali **MySql** mijoz dasturini chaqiring.

3. Taklif **MySql**> ga o`zgaradi. Buyruqni kiriting:

create database employees;

(Izoh: Buyruq nuqtali vergul bilan tugaydi).

4. **MySql** serveri taxminan 3.1 rasmdagidek javob berishi kerak



```
Query OK, 1 row affected (0.00 sec)
```

3.1 Rasm. Jadval yaratish buyrug`i ishining natijasi

[So`rov qayta ishlandi, 1 satr o`zgardi (0.00 sek)]

5. Bu ma`lumotlar ombori muvaffaqiyatli yaratilganligini anglatadi. Endi tizimda nechta ma`lumotlar ombori mavjudligini ko`ramiz. Quyidagi buyruqni bajaring:

show databases;

Server 3.2 rasmda ko`rsatilgandek, ma`lumotlar omborlari ro`yxati bilan javob beradi.

```
+-----+
| Data base |
+-----+
| employees |
| mysql     |
| test      |
+-----+
2 rows in set (0.00 sec)
```

3.2 Rasm. Ma`lumotlar omborlarini ko`rish.

Bu erda uchta ma`lumotlar ombori ko`rsatilgan, ulardan ikkitasi **MySql** tomonidan o`rnatish vaqtida yaratilgan hamda yangi yaratilgan employees ma`lumotlar ombori.

6. Yana DOS taklifiga qaytish uchun **MySql** taklifidan quit buyrug`ini tanlang.

Linux da ma`lumotlar ombori yaratish

1. Foydalanuvchi o`zining hisob yozuvi ustidan o`z ishlagani ma`qul, superfoydalanuvchi root kabi emas. Terminal seansni ishga tushirish zarur va superfoydalanuvchi bo`lishingiz mumkin (Buning uchun su buyrug`ini bajaring va superfoydalanuvchi root parolini kiriting).

2. **MySql** serverini ishga tushiramiz. Kiritamiz:

MySql -u root -p

Tizim **MySql** ni Linux ga o`rnatganda berilgan root **MySql** foydalanuvchi parolini kiritishni taklif qiladi. (Izoh: Bu Linux tizimining root foydalanuvchining emas, **MySql** tizimining root foydalanuvchi paroli). Ekranda xavfsiz parolni kiriting.

Muvaffaqiyatli registratsiyadan so`ng, tizim 3.3 rasmda ko`rsatilganidek salomlashish va **MySql** ga taklif matnini chiqaradi.

```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1 to server version: 5.01.01

Type 'help' for help.

mysql>
```

3.3 Rasm. Tizim bilan salomlashish.

(Siz **MySql** monitoriga xush kelibsiz. Buyruqlar ; yoki \g. id simvollar bilan tugallanadi, **MySql** bilan bog'lanish 5.01.01 versiyadagi server uchun 1 ga teng. Ma'lumotnoma olish uchun 'help' ni kiriting).

3. Endi employees ma'lumotlar omborini yaratsak bo'ladi. Quyidagi buyruqni bajarib:

```
create database employees;
```

(Izoh: buyruqlar nuqtali vergul bilan tugallanadi)

4. Shuni ta'kidlash muhimki, ushbu ma'lumotlar ombori root foydalanuvchisi tomonidan yaratiladi va shuning uchun unga kirish faqat root ruxsat bergan foydalanuvchilargagina mumkin bo'ladi. Ushbu ma'lumotlar omboridan boshqa, deylik, misha nomli hisob yozuvi bilan foydalanish uchun quyidagi buyruqlarni bajarish lozim:

```
GRANT ALL ON employees.* TO freak@localhost IDENTIFIED BY "pass"
```

Ushbu buyruq freak@localhost hisob yozuviga employees ma'lumotlar ombori uchun barcha vakolatlarni taqdim qiladi va pass parolini beradi. Ixtiyoriy boshqa foydalanuvchi uchun freak ni foydalanuvchining biror boshqa nomiga o'zgartirish va mos parolni tanlash kerak.

5. **MySql** seansini quit buyrug'ini taklifida GRANT ALL ON employees.* TO freak@localhost IDENTIFIED BY "pass" ni kiritish orqali yoping. Superfoydalanuvchi rejimidan chiqing va o'zingizning hisob yozuvingizga o'ting (exit ni kiriting).

6. **MySql** bilan oddiy hisob yozuvi yordamida bog'lanish uchun quyidagilarni kiriting:

```
MySql -va foydalanuvchi nomi -p
```

Keyin taklifdan so'ng parolni kiriting (Ushbu parol yuqorida GRANTS ALL... buyrug'i bilan berilgan edi). **MySql** da muvaffaqiyatli registratsiyadan so'ng, tizim salomlashish xabarini chiqaradi. Foydalanuvchi seansi 3.4 rasmda ko'rsatilganidek ko'rinishi kerak.

```
[misha@localhost misha]$ mysql -u misha -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 3 to server version: 3.22.32

Type 'help' for help.

mysql>
```

3.4 Rasm. MySql tizimida salomlashish oynasi.

7. SHOW DATABASES buyrug'ini kiritish tizimdagi barcha kirish mumkin bo'lgan ma'lumotlar omborlarining ro'yxatini chiqaradi.

MySql> SHOW DATABASES;

Ekranida 3.2 rasmdagiga o'xshash oyna paydo bo'lishi kerak.

MySql mijoz dasturidan chiqish uchun **MySql**> taklif satrida quit ni kiring.

CREATE DATABASE buyrug'i

CREATE DATABASE buyrug'i sintaksisi quyidagi ko'rinishga ega:

CREATE DATABASE [IF NOT EXISTS] ma'lumotlar ombori nomi

[create_spetsifikatsiyasi [,create_spetsifikatsiyasi]...]

CREATE DATABASE buyrug'i ma'lumotlar omborini ko'rsatilgan nom bilan yaratadi. Buyruqdan foydalanish uchun CREATE imtiyoziga ma'lumotlar ombori uchun ega bo'lish kerak. Agar shu nomli ma'lumotlar ombori mavjud bo'lsa, xato generatsiyalanadi.

create_spetsifikatsiyasi:

[DEFAULT] CHARACTER SET simvollar_to'plami_nomi

[DEFAULT] COLLATE taqqoslash_tartibi_nomi

create_spetsifikatsiyasi optsiyasi ma'lumotlar ombori xarakteristikalarini aniqlash maqsadida ko'rsatilishi mumkin. Ma'lumotlar ombori xarakteristikalari ma'lumotlar katalogida joylashgan db.opt faylida saqlanadi. CHARACTER SET konstruktciyasi jimlik qoidasiga asosan, ma'lumotlar ombori uchun simvollar to'plamini aniqlaydi. COLLATION konstruktciyasi jimlik qoidasiga asosan, taqqoslash tartibini beradi.

MySql da ma'lumotlar omborlari katalog shaklida amal qiladi, kataloglar ma'lumotlar ombori jadvallariga mos keluvchi fayllardan tashkil topgan. Omborda

boshidan hech qanday jadval bo'lmaganligi bois, CREATE DATABASE operatori faqat **MySQL** ma'lumotlar katalogida kichik katalog yaratadi.

Jadvallar bilan ishlash

Endi **MySQL** ning ma'lumotlar omborini tanlash va jadvallar yaratish uchun mo'ljallangan buyruqlarini ko'rib chiqamiz.

Ma'lumotlar ombori ma'lumotlarni jadvallarda saqlaydi. Bu jadvallarning o'zi nima?

Hammasidan ko'ra jadvallarni satr va ustunlardan tashkil topgan deb tasavvur qilish osonroq. Har bir ustun ma'lumotlarni ma'lum bir tipga ajratadi. Satrlar alohida yozuvlarni o'z ichiga oladi.

3.1 Jadvalda ayrim kishilarning shaxsiy ma'lumotlari keltirilgan:

3.1 jadval. Shaxsiy ma'lumotlar			
Ism	oshi	Da vlat	e-mail
Mixail Petrov	8	Ros siya	misha@yandex.ru
Djon Douson	2	Avs traliya	j.dow@australia.com
Moris Dryuon	8	Fra ntciya	md@france.fr
Snejan a	9	Bol gariya	sneg@bulgaria.com

Yuqorida keltirilgan jadval to'rt ustundan iborat, har bir ustunda ism, yosh, mamlakat va e-mail manzili saqlanadi. Har bir satr bir kishining ma'lumotlarini saqlaydi. Ushbu satr yozuv deyiladi. Snejananing e-mail manzili va mamlakatini topish uchun avval birinchi ustundan ismni tanlash hamda ushbu satrdagi uchinchi va to'rtinchi ustunlar tarkibini ko'rish zarur.

Ma`lumotlar ombori tarkibida ko'p jadvallarni saqlashi mumkin, aynan jadvallar haqiqiy ma`lumotlarni saqlaydi.

O'z navbatida o'zaro bog'liq (yoki bog'liq bo'lmagan) ma`lumotlarni turli jadvallarga ajratish mumkin. Employees ma`lumotlar ombori uchun kompaniyaning ishchilari haqidagi ma`lumotlarni bir jadvalda saqlaydi, boshqa jadvalda esa shaxsiy ma`lumotlar saqlanadi. Birinchi jadvalni yaratamiz.

Bunday jadvalni yaratish uchun SQL buyrug'i quyidagi ko'rinishda bo'ladi:

```
CREATE TABLE employee_data
(
emp_id int unsigned not null auto_increment primary key,
f_name varchar(20),
l_name varchar(20),
title varchar(30),
age int,
yos int,
salary int,
perks int,
email varchar(60)
);
```

Izoh: **MySQL** da buyruqlar va ustun nomlari simvol registrklarini farqlay olmaydi, lekin jadvallar va ma`lumotlar ombori nomlari, ular qo'llaniladigan platformaga qarab registrdan bog'liq bo'lishi mumkin (Linux dagi kabi). Shuning uchun CREATE TABLE o'rniga create table dan foydalanish mumkin.

CREATE TABLE kalit so'zlari ortidan yaratilayotgan jadval nomi employee_data keladi. Qavs ichidagi har bir satr bitta ustunni anglatadi. Bu ustunlar har bir xodimning identifikatsion raqami (emp_id), familiyasi (f_name), ismi (l_name), lavozimi (title), yoshi (age), kompaniyada ish staji (yos), maoshi (salary), qo'shimcha (perks) va e-mail manzili (email) ni saqlaydi.

Har bir ustun nomi ortidan ustun tipi keladi. Ustunlar tipi ulardagi ma`lumotlar tipini aniqlaydi. Ushbu misolda f_name, l_name, title va email

ustunlari matnli satrlardan tashkil topgan bo'ladi, shuning uchun ustun tipi varchar, ya'ni simvollarning o'zgaruvchi miqdori kabi berilgan. Varchar ustunlari uchun simvollarning maksimal soni ustun nomidan so'ng qavs ichida yoziladigan son bilan aniqlanadi. Age, yos, salary i perks ustunlari sonlar (butun) dan tashkil topganligi uchun ustun tipi int kabi beriladi. Birinchi ustun (emp_id) xodimning identifikatsion raqami (id) ni saqlaydi. Uning ustuni tipi birmuncha murakkab ko'ringanligi uchun uni qismlarga bo'lib ko'rib chiqamiz:

- int: ustun tipini butun son sifatida aniqlaydi.
- unsigned: son belgisiz bo'lishini belgilaydi (musbat son).
- not null: qiymat null (bo'sh) bo'la olmasligini, ya'ni ushbu ustundagi har bir satr qiymatga ega bo'lishini belgilaydi.
- auto_increment: **MySql** auto_increment atributli ustunga duch kelganda, ustundagi eng katta qiymatdan bir miqdorga katta yangi qiymat paydo bo'ladi. Shuning uchun biz ushbu ustun uchun qiymat berishimiz kerak, **MySql** uni mustaqil generatsiyalaydi. Bundan kelib chiqadiki, ushbu ustundagi har bir qiymat unikal bo'ladi.
- primary key: ustunni indeksatsiyalashda yordam beradi, bu esa qiymatlarni qidirishni tezlashtiradi. Har bir qiymat unikal bo'lishi kerak. Kalit ustun ma'lumotlar takrorlanishini oldini olish uchun zarur. Misol uchun, ikki xodim bir xil ismga ega bo'lishi mumkin, bunda agar ularga unikal identifikatsion raqam berilmasa, ularni farqlash muammosi yuzaga keladi. Agar unikal qiymatli ustun mavjud bo'lsa, ikki yozuvni oson ajratish mumkin. Yaxshisi, unikal qiymatlarni o'zlashtirishni **MySql** tizimining o'ziga topshirish kerak.

Ma'lumotlar omboridan foydalanish

Employees ma'lumotlar ombori yaratildi. U bilan ishlash uchun uni "aktivlashtirish" yoki "tanlash" zarur. **MySql** taklifida quyidagi buyruqni bajaring:

```
SELECT DATABASE();
```

Ekranida 3.5 rasmda ko'rsatilganidek, tizim javobini ko'ramiz

```
mysql> SELECT DATABASE();
+-----+
| DATABASE |
+-----+
|          |
+-----+
1 rows in set (0.00 sec)
```

3.5 Rasm. Ma`lumotlar omborini tanlash

Bu shuni anglatadiki, birorta ma`lumotlar ombori tanlanmadi. Aslida **MySql** mijosi bilan har marotaba ishlaganda, qaysi ma`lumotlar omboridan foydalanilishni aniqlash zarur.

Joriy ma`lumotlar omborini bir necha usul bilan aniqlash mumkin:

- ishga tushirgan vaqtda ma`lumotlar ombori nomini aniqlash

Tizim taklifida quyidagilarni kiriting:

MySql employees (Windows da)

MySql employees -u manish -p (Linux da)

- Ma`lumotlar omborini **MySql** taklifidagi **USE** operatori yordamida aniqlash

MySql>USE employees;

- Ma`lumotlar omborini **MySql** taklifidagi \u yordamida aniqlash

MySql>\u employees;

MySql bilan ishlaganda foydalaniladigan ma`lumotlar omborini aniqlab olish muhim, aks holda **MySql** xatoni keltirib chiqaradi.

Jadval yaratish

Employees ma`lumotlar ombori tanlangandan so`ng, **MySql** taklifida **CREATE TABLE** buyrug`ini tanlang.

```
CREATE TABLE employee_data
```

```
(
```

```
emp_id int unsigned not null auto_increment primary key,
```

```
f_name varchar(20),
```

```
l_name varchar(20),
```

```
title varchar(30),
```

```
age int,  
yos int,  
salary int,  
perks int,  
email varchar(60)  
);
```

Izoh: Birinchi satrdan so'ng, Enter tugmasini bosish **MySql** taklifini -> ga o'zgartiradi. Bu shuni anglatadiki, **MySql** buyruq tugamaganligini tushunadi va qo'shimcha operatorlarni kiritishni taklif etadi. Unutmang, har bir **MySql** buyrug'i nuqtali vergul bilan tugaydi, ustundagi har bir e`lon vergul bilan ajratiladi. Agar istasangiz, hamma buyruqni bitta satrda kiritish mumkin.

Ekran 3.6 rasmdagi ko'rinishga ega bo'lishi kerak.

```
mysql> CREATE TABLE employee data  
-> (  
-> emp_id int unsigned not null auto_increment primary key,  
-> f_name varchar(20),  
-> l_name varchar(20),  
-> title varchar(30),  
-> age int,  
-> yos int,  
-> salary int,  
-> perks int,  
-> email varchar(60)  
-> );  
  
Query OK, 0 rows affected (0.01 sec)
```

3.6 Rasm. Jadval yaratish

CREATE TABLE buyrug'ining sintaksisi

CREATE TABLE instrukciyasining umumiy formati quyidagicha:

CREATE [TEMPORARY] TABLE [IF NOT EXISTS] nom

[(spetsifikatsiya, ...)]

[optsiya, ...]

[[IGNORE | REPLACE] so'rov]

TEMPORARY bayrog'i joriy seans davomida amal qiladigan vaqtinchalik jadvalni yaratadi. Seans tugashi bilan jadval o'chiriladi. Vaqtincha amal qiladigan jadvallarga boshqa jadvallar nomini berish mumkin, bunda shu jadvallarga

vaqtincha kirib bo'lmaydi. IF NOT EXISTS spetsifikatori agar ko'rsatilgan nomli jadval mavjud bo'lsa, xatolar to'g'risida xabar chiqarishni kamaytiradi. Jadvallar nomlari ma'lumotlar omborining nuqta bilan ajratilgan nomidan keyin kelishi mumkin. Agar bunday qilinmagan bo'lsa, jadval jimlik qoidasiga binoan o'rnatilgan ma'lumotlar omborida yaratiladi.

Jadvalga probelli nom berish uchun uni teskari qavsga, masalan, 'courses list' ga joylashtirish lozim. Xuddi shu ishni jadvalga barcha murojaatlar uchun qilish kerak, chunki probellar identifikatorlarni ajratish uchun qo'llaniladi.

Ustunsiz jadvallar yaratishga ruxsat beriladi, ammo ko'p hollarda hech bo'lmaganda bitta ustun spetsifikatsiyasi mavjud bo'ladi. Ustunlar va indekslar spetsifikatsiyasi aylana qavslar va vergullar bilan ajratiladi. Spetsifikatsiyalar formati quyidagicha:

nom tip

[NOT NULL | NULL]

[DEFAULT qiymat]

[AUTO_INCREMENT]

[KEY]

[murojaat]

Tip spetsifikatsiyasi tipning nomlanishi va uning o'lchamlanishini o'z ichiga oladi. Jimlik qoidasiga asosan, ustunlar NULL qiymatini oladi. NOT NULL spetsifikatori bunday holatni man qiladi.

Har bir ustunda jimlik bo'yicha qiymati mavjud. Agar u ko'rsatilmagan bo'lsa, **MySQL** dasturi uni mustaqil tanlaydi. NULL qiymatini qabul qiluvchi ustunlar uchun jimlik bo'yicha qiymat NULL bo'ladi, satrli ustunlar uchun – bo'sh satr, sonli ustunlar uchun – nul. Bunday o'rnatishni o'zgartirish uchun DEFAULT taklifidan foydalanish zarur.

AUTO_INCREMENT bayrog'i yordamida yaratiladigan hisoblagich – maydonlar jimlik bo'yicha berilgan qiymatlarni qabul qilmaydi, chunki ularda tartib raqamlari yoziladi. Hisoblagich tipi belgisiz butun bo'lishi zarur. Jadvalda

faqat bitta hisoblagich – maydon bo'lishi mumkin. U birinchi kalit bo'lishi shart emas.

Jadvalni o'chirish

Jadvalni o'chirishdan oldin uning mavjudligiga ishonch hosil qilamiz. Buni 3.7 rasmda ko'rsatilganidek, SHOW TABLES buyrug'i yordamida tekshirish mumkin.

```
mysql> SHOW TABLES;
+-----+
| Tables in employees |
+-----+
| employee data      |
+-----+
1 rows in set (0.00 sec)
```

3.7 Rasm. Ma`lumotlar omborida jadvalni ko'rish

Jadvalni o'chirish uchun 3.8 rasmda ko'rsatilganidek, DROP TABLE buyrug'idan foydalaniladi.

```
mysql> DROP TABLE employee_data;
Query OK, 0 rows affected (0.01 sec)
```

3.8 Rasm. Jadvalni o'chirish

Endi SHOW TABLES buyrug'i bu jadvalni boshqa ko'rsatmaydi.

DROP TABLE buyrug'i sintaksisi

DROP TABLE instrukciyasi quyidagi sintaksisga ega:

DROP TABLE [IF EXISTS] jadval [RESTRICT | CASCADE]

IF EXISTS spetsifikatsiyasi berilgan jadval mavjud bo'lmagan holatda xato haqidagi xabarlarini chiqarishni kamaytiradi. Bir necha jadval nomlarini, ularni vergul bilan ajratib ko'rsatish mumkin.

RESTRICT i CASCADE bayroqlari boshqa MOBT da yaratilgan stsenariylarni bajarish uchun mo'ljallangan.

FOYDALANILGAN ADABIYOTLAR

1. Ратшиллер Т., Геркен Т. PHP4: разработка Web-приложений. Питер, 2001. - 384 с.
2. Д.Котерев, А.Котерев. PHP5. наиболее полное руководство./Санкт Петербург «БХВ-Петербург» - 2005.- 1120 с.
3. Уилтон П. JAVASCRIPT. Основы. Символ-плюс. 2002. 1056 с.
4. Кингли-Хью Э., Кингли-Хью К. JAVASCRIPT 1.5: Учебный курс. Питер. 1-е издание. 2002.
5. Дронов В. JavaScript в Web дизайне. Питер. 2005.
6. Мазуркевич М.М., Мазуркевич И.М. PHP4. Настольная книга программиста. СПб. 2003.
7. Бранденбау. JAVASCRIPT. Сборник рецептов для профессионалов. СПб. 2001.
8. Томсон Л., Веллинг Л. Разработка Web-приложений на PHP и MySQL. - К.: "ДиаСофт", 2008. - 672 с.
9. Ahmedov A. Tayloqov N. Informatika. –Т.: -2001 у.
10. Hayitov A.G. Informatikadan mashqlarni kompyuterda bajarishning nazariy asoslari. –Т.: A.Qodiriy. -2003у. -211 б.
11. Т.Н. Holmatov, N.I.Tayloqov, U.A.Nazarov. Informatika. –Т.: -2003 у.
12. М.Арипов, В.Бегалов, У.Бегимқолов. Ахборот texnologiyalari. Toshkent 2009 у.