**O‘ZBEKISTON RESPUBLIKASI**

**RAQAMLI TEXNOLOGIYAR VAZIRLIGI**

**MUHAMMAD AL-XORAZMIY NOMIDAGI**

**TOSHKENT AXBOROT TEXNOLOGIYALARI UNIVERSITETI**



**Ma’lumotlar bazasi fanidan**

# **14 – TOPSHIRIQ**

Bajardi: Abduraimov Javohirmirza

**TOSHKENT 2024**

# Amaliy mashg’ulot

**Mavzu: SQLda protseduralar yaratish**

**Ishdan** **maqsad:** Berilgan predmet soha ma`lumotlar bazasi uchun protsedura yaratish.

**Masalani qo`yilishi:** Predmet soha ma`lumotlar bazasi shakllantirilgandan so`ng ob`yektlarda proseduralarni yaratish va ulardan foydalanish.

**Uslubiy** **ko`rsatmalar**:

Protsedura bu ma'lumotlar bazasida saqlanadigan oddiy skript tilidagi kichik dastur hisoblanadi. (subprogram kabi). MySQL holatida protseduralar MySQL-da yoziladi va MySQL ma'lumotlar bazasida/serverda saqlanadi.

MySQl da protdedura bilan ishlashning umumiy sintaksisi:

**Protsedura yaratish:**

**delimiter //**

**create procedure protsedura\_nomi(protsedura prototipi)**

**begin**

**{protsedura so’rovlar amallar… maydoni}**

**end//**

protsedurani chaqirish:

**call protsedura\_nomi(@prototip)//**

**protsedura natijasini ko’rish:**

**select @prototip**;

Eslatma: delimiterning vazifasi shuki bu kodlarni bitta satrda emas bir necha satrlarda yozib birlashtirish imkonini beradi. Delimiter belgisi orasida proseduraning kodi yoziladi. Uning sintaksisi:

1. Boshlanishi:

Delimiter kalit\_belgi

1. Tugashi:

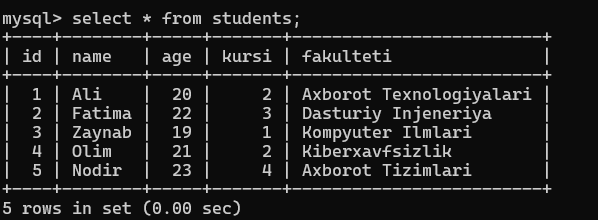
End kalit\_belgi

**Ishni bajarish tartibi**

1. “xodimlar” jadvalini chaqiramiz:

**So`rov: select \* from students;**

**Natija:**



**16.1 – rasm. xodimlar** jadvalidagi ma’lumotlarni ko’rish

1. Qurilma jadvalini chiqaruvchi sodda protsedura tuzamiz:

**So` rov: Delimiter //**

**Create procedure chiqarish()**

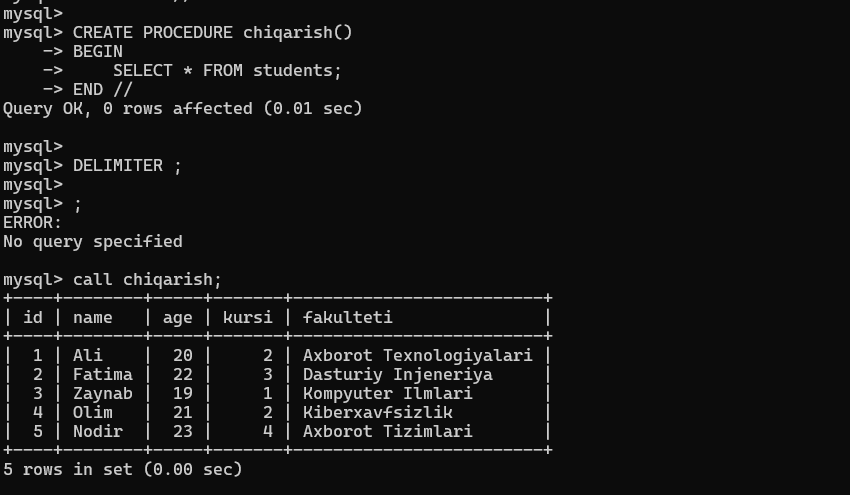
**Begin**

**Select \* from students;**

**End//**

**Call chiqarish(); //**

**Natija:**

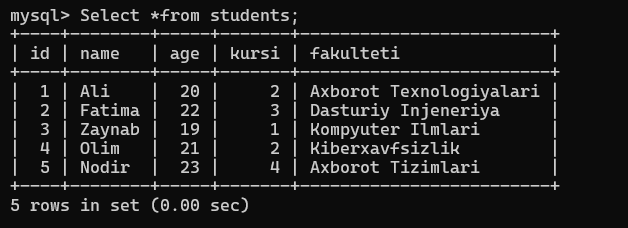


**16.2 – rasm. Chiqarish()** prosedurasini ishga tushirish natijasi

Endi boshqacharoq protsedura hosil qilamiz, yani “shahar” jadvalidan eng katta yoshning qiymatini aniqlovchi protsedura tuzamiz:

**So`rov: Select \*from students;**

**Natija:**



**16.3 – rasm. shahar** jadvalidagi ma’lumotlarni ko’rsatuvchi so`rov

1. Protsedura yaratamiz:

**So`rov: DELIMITER //**

**CREATE PROCEDURE kop\_yosh(out k INT)**

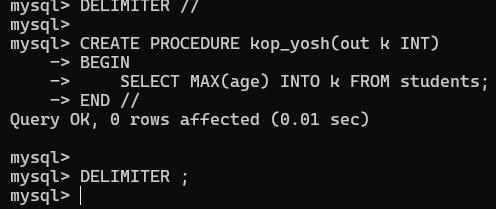
**BEGIN**

**SELECT MAX(age) INTO k FROM students;**

**END //**

**DELIMITER ;**

**Natija:**



* 1. **– rasm. Max()** funksiyasi qatnashgan prosedura yaratish

1. Protsedurani chaqiramiz:

**So`rov: CALL kop\_yosh(@max\_age);**

**Natija:**

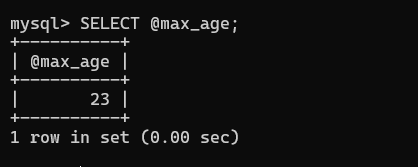


* 1. **– rasm.** Prosedurani chaqirish so`rovi

1. Natijani ko’ramiz:

**So`rov: SELECT @max\_age;**

**Natija:**



**16.6 – rasm. kop\_aholi()** prosedurasini chaqirish orqali olingan natija

Eslatma: bu yerda “//” belgi delimiter hisoblanadi u so’rov tugaganda qo’yiladi.

**Mavzu: SQLda triggеrlar yaratish**

MySQLda, trigger - bu bog'langan jadvalda sodir bo'lgan kiritish, yangilash yoki o'chirish kabi hodisaga javoban avtomatik ravishda chaqiriladigan saqlanadigan dastur. Masalan, jadvalga yangi qator qo'yilishidan oldin avtomatik ravishda ishga tushadigan tetikni belgilashingiz mumkin.

MySQL INSERT, UPDATE yoki DELETE voqealariga javoban ishga tushadigan triggerlarni qo'llab-quvvatlaydi.

SQL standarti triggerlarning ikki turini aniqlaydi: qator-darajali triggerlar va bayonot darajasidagi triggerlar. Qo'shilgan, yangilangan yoki o'chirilgan har bir satr uchun qator darajasidagi trigger faollashadi. Masalan, agar jadvalda 100 satr qo'shilgan, yangilangan yoki o'chirilgan bo'lsa, ta'sirlangan 100 ta satr uchun trigger avtomatik ravishda 100 martta chaqiriladi. Qancha qator qo'shilgan, yangilangan yoki yo'q qilinganidan qat'iy nazar, bayonot darajasidagi trigger har bir operatsiya uchun bir martta bajariladi.

**MySQL** da tiriggerlarning yaratilish umumiy sintaksisi:

**CREATE TRIGGER trigger\_nomi trigger\_joylashishi trigger\_buyrug’i  
ON jadval\_nomi FOR EACH ROW trigger\_tanasi;**

Bu yerda trigger ishlatilishi turlari:

1. Before (oldidan)
2. After (oxiridan) va boshqalar.

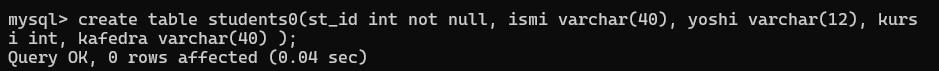
Trigger buyruqlari turlari:

1. Insert (kiritish)
2. Update (yangilash)
3. Delete (o’chirish) va boshqalar

Bizda shahar jadvali bor edi. Triggerlar bilan ishlash uchun shahar0 degan jadval yaratamiz:

**create table students0(st\_id int not null, ismi varchar(40), yoshi varchar(12), kurs**

**i int, kafedra varchar(40) );**

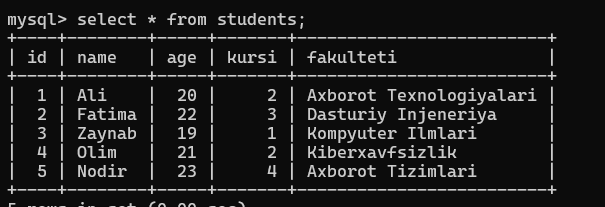


* 1. **– rasm. Shahar0** jadvalining yaratilishi

Endi triggerlarni yaratishni boshlaymiz:

1. **Before update trigger**

**shahar** jadvaliga e’tibor qilaylik.

****

* 1. **– rasm. shahar** jadvalidagi ma’lumotlarni ko`rish so`rovi

**Shahar\_yangilash** triggerini yaratamiz.

**DELIMITER //**

**CREATE TRIGGER students\_yangilash**

**BEFORE UPDATE ON students**

**FOR EACH ROW**

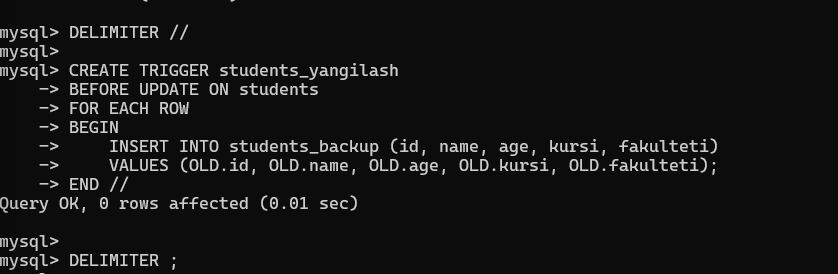
**BEGIN**

**INSERT INTO students\_backup (id, name, age, kursi, fakulteti)**

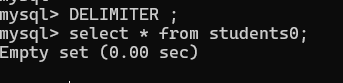
**VALUES (OLD.id, OLD.name, OLD.age, OLD.kursi, OLD.fakulteti);**

**END //**

**DELIMITER ;**

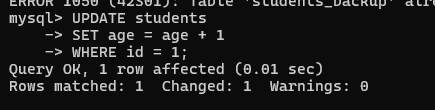
****

* 1. **– rasm. Shahar\_yangilash** triggerini yaratish
  2. **Students0** jadvalini ko’ramiz.

****

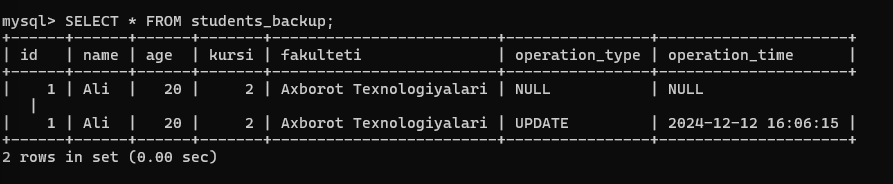
* 1. **– rasm. shahar0** jadvalidagi ma’lumotlarni ko`rish so`rovi

Yaratgan triggerimizni ishlatamiz, yani ID 1ga teng xodim yoshiga 2 ni qo’shib yangilaymiz;

****

**17.5– rasm. shahar** jadvali

* 1. shahar jadvaliga yana bir nazar solsak o’zgarishni ko’ramiz;

****

**17.6 –rasm.** Update operatorini ishlatgandan keyin shahar jadvalidagi ma’lumotlar

**Students0** jadvalini ko`ramiz.

****

**17.7 – rasm.** Update opertaorini ishlatgandan keying master0 jadvali

**3. Before insert trigger:**

* 1. **shahar\_kiritish\_oldin** triggerini yaratamiz.

**DELIMITER //**

**CREATE TRIGGER student\_kiritish\_olddin**

**BEFORE INSERT ON students**

**FOR EACH ROW**

**BEGIN**

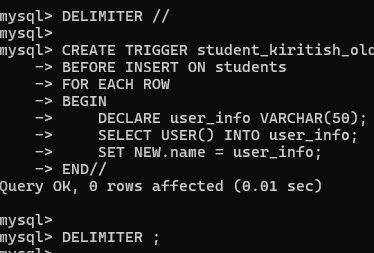
**DECLARE user\_info VARCHAR(50);**

**SELECT USER() INTO user\_info;**

**SET NEW.name = user\_info;**

**END//**

**DELIMITER ;**



* 1. **– rasm. shahar\_kiritish\_oldin** triggerini yaratish so`rovi

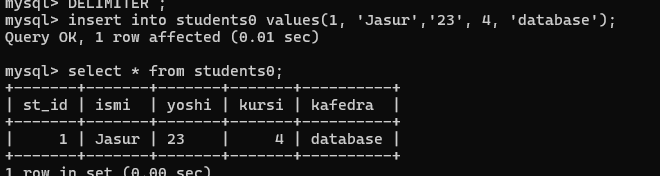
Yaratgan triggerimizni ishlatamiz va natijasini shahar0 jadvalida ko’ramiz; ya’ni biz trigger yordamida yangi satr kiritdik.

**mysql> DELIMITER ;**

**mysql> insert into students0 values(1, 'Jasur','23', 4, 'database');**

**Query OK, 1 row affected (0.01 sec)**

**mysql> select \* from students0;**



**17.13 – rasm. shahar\_kiritish\_oldin** triggerni ishlashi natijasi

**Xulosa**

Berilgan predmet soha ma`lumotlar bazasi uchun protsedura yaratdik.

Predmet soha ma`lumotlar bazasi shakllantirilgandan so`ng ob`yektlarda proseduralarni yaratdik va ulardan foydalandik. SQLda triggеrlar yaratishni o’rgandik.

**Foydalanilgan 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.