*ÓBUDAI EGYETEM*

*Neumann János Informatikai Kar*

*Nappali Tagozat*

**Adatbázis-kezelés féléves feladat**

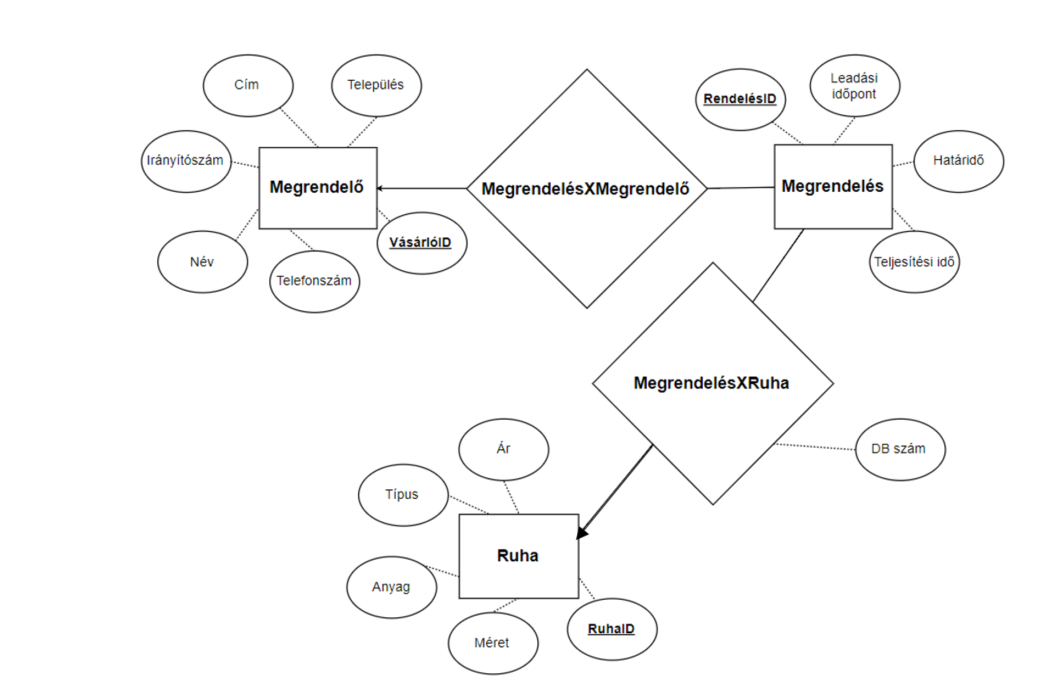
***Név:*** Krujz Gergely

***Neptun kód:*** P71JVI

***A dolgozat címe:*** Egy ruhagyár adatbázisa

**1.Környezet:**

* Egy ruhagyár adatbázisát fogom megtervezni. Lesz egy megrendelő , aki megrendel egy ruhát és a választékból választhat,hogy milyen legyen a mérete, típusa, stb..
* Megrendelő: Minden ami a megrendelővel kapcsolatos információ, azt eltároljuk. Például: név,lakcím,telefonszám.
* Megrendelés: Itt adódik le a rendelés, eltárolódik a leadás időpontja, a határideje, és a körülbelüli teljesítési idő.
* Ruha: A ruhának az adatai ( méret, anyag , típus, ár) itt fog eltárolódni.

**2. E/K modell:**

**3. E/K modell átalakítása relációkká:**

* Megrendelő:(VásárlóID,Név,Cím,Város,Irányítószám,Telefonszám)
* Megrendelés(RendelésID,Leadási időpont,Határidő,Teljesítési idő , VásárlóID, RuhaID, DB szám)
* Ruha(RuhaID,Méret,Anyag,Ár,Típus)
* VásárlóID: idegen kulcs a Megrendelés táblában levő RendelésID-re
* RuhaID: idegen kulcs a Megrendelés táblában levő RendelésID-re

**4.Normalizálás:**

* **Megrendelő: (**VásárlóID,Név,Cím,Város,Irányítószám,Telefonszám)
* 1NF az megfelel, mert minden sorának minden mező értéke egy elemi érték.
* 2NF megfelel , mivel a VásárlóID –hez kapcsolható az összes további jellemző.
* 3NF megfelel, mivel nincs egyik(másodlagos) jellemző között funkcionális függőség.
* **Megrendelés:** (RendelésID,Leadási időpont,Határidő,Teljesítési idő , VásárlóID, RuhaID, DB szám)
* 1NF az megfelel, mert minden sorának minden mező értéke egy elemi érték.
* 2NF megfelel, mivel a RendelésID-hez kapcsolható az összes további jellemző.
* 3NF megfelel, mivel nincs egyik(másodlagos) jellemző között funkcionális függőség.
* **Ruha:** (RuhaID,Méret,Anyag,Ár,Típus)
* 1NF az megfelel, mert minden sorának minden mező értéke egy elemi érték.
* 2NF megfelel, mivel a RuhaID-hez kapcsolható az összes további jellemző.
* 3NF megfelel, mivel nincs egyik(másodlagos) jellemző között funkcionális függőség.

**5.Táblák létrehozása és megszorítása**

CREATE TABLE Megrendelo(

VasarloID NUMBER NOT NULL,

Nev VARCHAR2(32),

Cim VARCHAR2(32),

Varos VARCHAR2(32),

Iranyitoszam VARCHAR2(9),

Telefonszam VARCHAR2(18),

CONSTRAINT PK\_VasarloID PRIMARY KEY(VasarloID)

);

CREATE TABLE Ruha(

RuhaID NUMBER NOT NULL,

Ar number,

Tipus VARCHAR2(32),

Anyag VARCHAR2(16),

Meret number,

CONSTRAINT PK\_RuhaID PRIMARY KEY(RuhaID)

);

CREATE TABLE Megrendeles(

RendelesID NUMBER NOT NULL,

Leadasi\_idopont DATE,

Hatarido DATE,

Teljesitesi\_ido number,

DB\_szam number,

VasarloID NUMBER NOT NULL,

RuhaID NUMBER NOT NULL,

CONSTRAINT PK\_RendelesID PRIMARY KEY(RendelesID),

CONSTRAINT FK\_MegrendelesXMegrendelo FOREIGN KEY(VasarloID) REFERENCES Megrendelo(VasarloID),

CONSTRAINT FK\_MegrendelesXRuha FOREIGN KEY(RuhaID) REFERENCES Ruha(RuhaID)

);

**Táblák feltöltése:**

INSERT INTO Megrendelo VALUES(1,'Fehér Jenő','Béke utca 63','Budapest','1000','06308213445');

INSERT INTO Megrendelo VALUES(2,'Piros Péter','Zámbó utca 6','Hévíz','2000','06306661214');

INSERT INTO Megrendelo VALUES(3,'Zöld Zoltán','Kisréti út 10','Badacsony','1100','06304561234');

INSERT INTO Megrendelo VALUES(4,'Kék Károly','Négyzet utca 44','Salgótarján','2300','06305643281');

INSERT INTO Megrendelo VALUES(5,'Fekete László','Kerek utca 1','Dunaharaszti','2400','063034567891');

INSERT INTO Megrendelo VALUES(6,'Barna Béla','Orvos utca 2','Kiskunhalas','2500','06305372822');

INSERT INTO Ruha VALUES(1,100000,'Öltöny(fekete)','vegyes',180);

INSERT INTO Ruha VALUES(2,12000,'Téli kabát(fehér)','műbőr',170);

INSERT INTO Ruha VALUES(3,5000,'Ing(barna)','pamut',170);

INSERT INTO Ruha VALUES(4,9700,'öv(fekete)','műbőr',32);

INSERT INTO Ruha VALUES(5,1000,'Zokni\_pakk(fekete)','pamut',40);

INSERT INTO Ruha VALUES(6,900,'Kesztyű\_pakk(fehér)','pamut',19);

INSERT INTO Megrendeles VALUES(1,TO\_DATE('2017-10-1 10:44:22','YYYY-MM-DD HH24:MI:SS'),TO\_DATE('2017-10-11 10:44:22','YYYY-MM-DD HH24:MI:SS'),10,1,1,1);

INSERT INTO Megrendeles VALUES(2,TO\_DATE('2017-10-2 11:33:54','YYYY-MM-DD HH24:MI:SS'),TO\_DATE('2017-10-22 11:33:54','YYYY-MM-DD HH24:MI:SS'),20,2,2,2);

INSERT INTO Megrendeles VALUES(3,TO\_DATE('2017-10-3 12:56:12','YYYY-MM-DD HH24:MI:SS'),TO\_DATE('2017-10-13 12:56:12','YYYY-MM-DD HH24:MI:SS'),10,3,3,3);

INSERT INTO Megrendeles VALUES(4,TO\_DATE('2017-10-4 13:15:05','YYYY-MM-DD HH24:MI:SS'),TO\_DATE('2017-10-24 13:15:05','YYYY-MM-DD HH24:MI:SS'),20,4,4,4);

INSERT INTO Megrendeles VALUES(5,TO\_DATE('2017-10-5 14:23:20','YYYY-MM-DD HH24:MI:SS'),TO\_DATE('2017-10-25 14:23:20','YYYY-MM-DD HH24:MI:SS'),20,5,5,5);

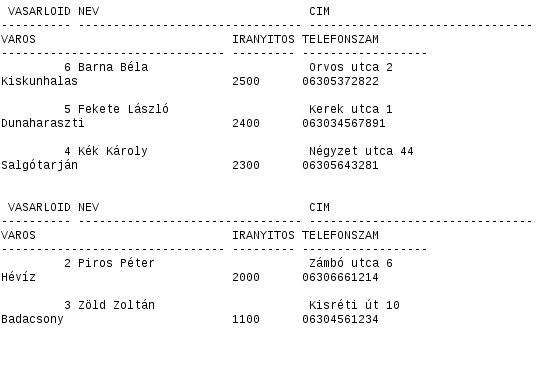
INSERT INTO Megrendeles VALUES(6,TO\_DATE('2017-10-6 15:11:40','YYYY-MM-DD HH24:MI:SS'),TO\_DATE('2017-10-16 15:11:40','YYYY-MM-DD HH24:MI:SS'),10,6,6,6);

**6.Lekérdezések:**

**Egyszerű lekérdezések:**

**Válasszuk ki minden adatot azokról akik nem budapestre rendeltek, név szerint növekvő sorrendbe.**

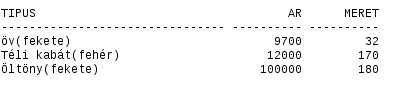
select \* from Megrendelo where Varos not like 'Budapest' order by nev;



**Válasszuk ki azokat a Ruhák tipusát árát és méretét ahol az ár nagyobb mint 6000 ft**

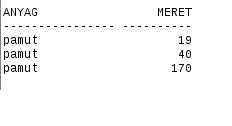
**és rendszerezzük méret szerint.**

Select Tipus,Ar,Meret from Ruha where Ar>6000 order by Meret;



**Válasszuk a Ruhák közül az anyagot és a méretet ahol az anyag az pamut és rendszerezzük méret szerint**

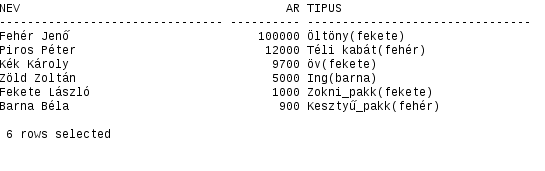
Select Anyag, Meret from Ruha where Anyag='pamut' order by Meret;



**Több táblás lekérdezések:**

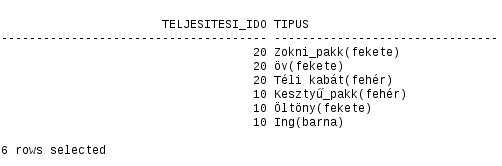
**Válasszuk ki a Megrendelő nevét a ruha árát és tipusát és az ár szerint növekvő sorrendbe helyezzük el.**

Select Nev,Ar,Tipus from Megrendelo natural join Megrendeles natural join Ruha order by Ar desc;



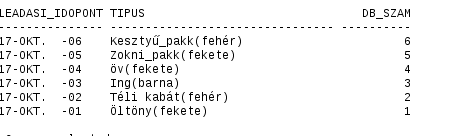
**Válasszuk ki a teljesitesi időt és a ruha tipusát teljesitési idő szerint növekvő sorrendben**

Select Teljesitesi\_ido , Tipus from Megrendeles natural join Ruha order by Teljesitesi\_ido desc;



**Válasszuk ki a leadás időpontját a ruha tipusét és db számát a db szám szerint növekvő sorrendben.**

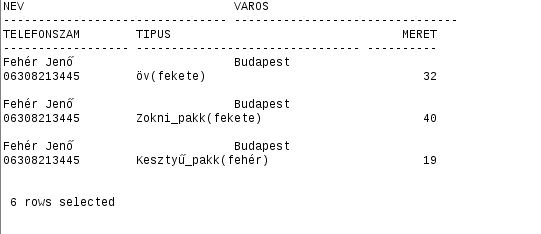
Select Leadasi\_idopont,Tipus,DB\_szam from Megrendeles natural join Ruha order by DB\_szam desc;



**Allekérdezések:**

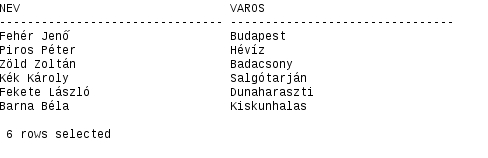
**Válasszuk ki a Megrendelő nevét Városát Telefonszámát És a megrendelt ruha tipusát és méretét ahol a város megegyezik Budapesttel**

Select Nev,Varos,Telefonszam,Tipus,Meret from Megrendelo natural join Ruha where (Varos='Budapest');



**Válasszuk ki a Megrendelő nevét városát ahol az adott megrendelő pontosan 12000 ft-ot hagyott ott 1 db ruháért.**

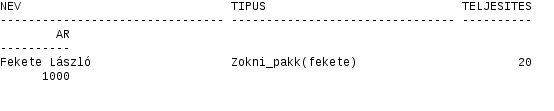
Select Nev,Varos from Megrendelo natural join Ruha where (Ar=12000);



**Válasszuk ki Fekete László Megrendelő nevét a Ruha tipusát , a teljesitési időt és az árat, ahol db onként kifizetett ruháit rangsoroljuk.**

Select Nev,Tipus, Teljesitesi\_ido Teljesites,Ar from Megrendelo Natural join Ruha natural join Megrendeles

where (Nev='Fekete László') order by Ar;



**Nézetek:**

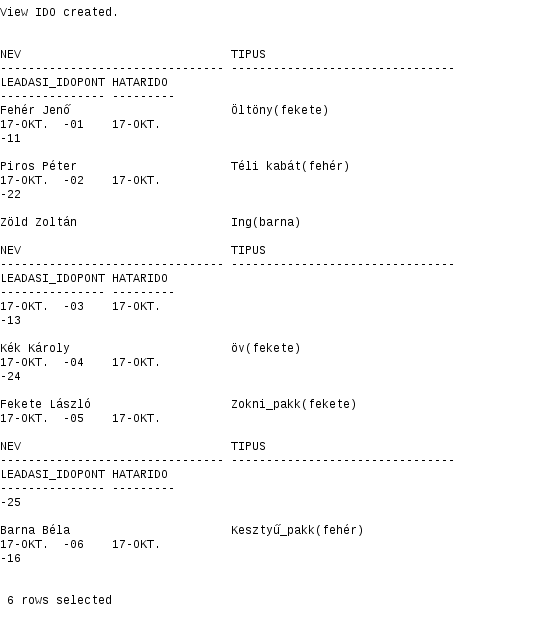
**Készítsünk nézetet ahol megtudjuk vizsgalni hogy ki milyen ruhát rendelt , milyen időpontban és hogy mennyi idő alatt kell kiszállítani**

create or replace view Ido as

select Nev, Tipus , Leadasi\_idopont, Hatarido from Megrendelo natural join Ruha

natural join Megrendeles;

select \* from Ido;

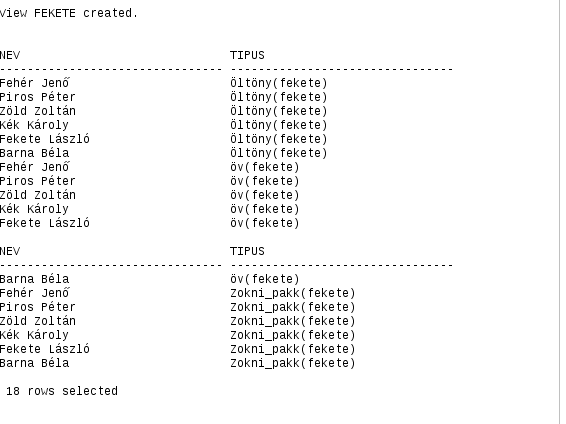


**Készítsünk nézetet ahol csak a fekete ruhákat listázzuk.**

create or replace view Fekete as

select Nev,Tipus from Megrendelo natural join Ruha where (Tipus Like '%fekete%');

Select \* from Fekete;



**Készítsünk nézetet ahol a megrendelő többet fizetett az összes ruhájáért mint 1000 ft.**

create or replace view Ar as

select Nev from Megrendelo natural join Ruha natural join Megrendeles where (DB\_szam>1) And ((Ar\*DB\_szam)>1000);

Select \* from Ar ;



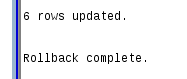
**DML műveletek:**

**Készítsünk updatet ,ahol 20% os leárazás folyik.**

savepoint s0;

update Ruha set Ar=Ar-(0.2\*Ar);

rollback to s0;

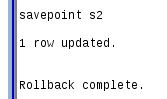


**Készítsünk updatet ahol a Zöld Zoltánnak megváltozott a telefonszáma 0620399213-ra**

savepoint s2;

update Megrendelo set Telefonszam = '0620399213' where upper(nev) like upper('Zöld Zoltán');

rollback to s2;

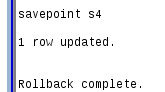


**Készítsünk updatet ahol átírjuk Budapestet Fővárossá mindenhol ahol szerepel a Budapest mint város**

savepoint s5;

update vendeg set varos='Főváros' where varos='Budapest';

rollback to s5;



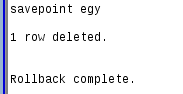
**Adattörlés:**

**Töröljük ki kék károly rendelését.**

savepoint egy;

delete from Megrendeles where VasarloID=(Select VasarloID from Megrendelo where Nev='Kék Károly');

rollback to egy;

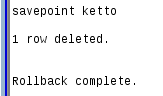


**Töröljük ki azokat a rendeléseket ahol barna színű ruhát rendeltek.**

savepoint ketto;

delete from Megrendeles where RuhaID=(Select RuhaID from Ruha where Tipus like '%barna%');

rollback to ketto;

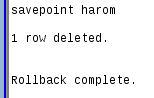


**Töröljük ki azokat a rendeleseket ahol 19 es méretben rendeltek ruhát.**

savepoint harom;

delete from Megrendeles where RuhaID=(Select RuhaID from Ruha where Meret=19);

rollback to harom;



**AdatBeszúrás:**

INSERT INTO Ruha VALUES(7,90,'Cipőfűző(Fehér)','pamut',10);

C:\Users\K.Gergo\Desktop\adatb képek\insert.png

**Alprogramok, triggerek:**

**Eljárások, függvények:**

**Listázzuk ki azokat akik Badacsonyból rendeltek**

create or replace procedure Badacsony is

begin

for i in (select Nev from Magyar.Megrendelo where Varos='Badacsony')

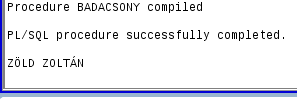
loop

dbms\_output.put\_line(upper(i.Nev));

end loop;

end;

execute Badacsony;



**Készítsünk egy function-t, ami kiszámolja ,hogy hány embernek van 30-as telefonszámuk**

create or replace function telefonszamok return number is

db number:=0;

begin

for i in(select \* from Magyar.Megrendelo where (Telefonszam like '%0630%'))

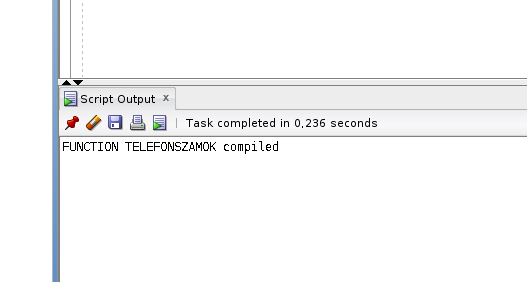
loop

db:=db+1;

end loop;

return db;

end;



**Töröljünk ki egy megrendelést és nézzük meg hogy van e jogosultságunk hozzá.**

create or replace procedure Megrendelestorlese(VasarloID in Megrendelo.VasarloID%type) is

lehetoseg number;

begin

select count(\*) into lehetoseg from Megrendeles where VasarloID=VasarloID;

if lehetoseg=0 then

dbms\_output.put\_line('A törlés engedélyezett');

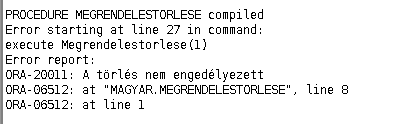
else

raise\_application\_error(-20011,'A törlés nem engedélyezett');

end if;

end;

execute Megrendelestorlese(1);



**Triggerek:**

**Készítsünk egy triggert ami törlés előtt kiírja h milyen adatokkal rendelkezett a bizonyos rendelés**

create or replace trigger Megrendeltorles

before delete on Megrendeles

for each row

begin

dbms\_output.put\_line('Leadása: ' || :old.Leadasi\_idopont || ' Határideje: ' || :old.Hatarido || ' A vásárló id je ' ||

:old.VasarloID || ' DB szám ' || :old.DB\_szam);

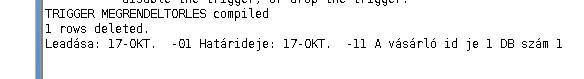
exception

when no\_data\_found then

dbms\_output.put\_line('Hiba! Nem jó adatbevitel');

end;

delete from Magyar.Megrendeles where RendelesID=1;

****

**Készítsünk triggert amikor nőnek a ruhaméretek 1 –el**

create or replace trigger Novekedes

before update on Ruha

begin

dbms\_output.put\_line('A Ruhaméretek nőttek 1 el');

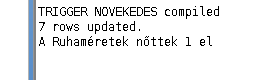
exception

when no\_data\_found then

dbms\_output.put\_line('Hiba lépett fel , nem található ruha méret');

end;

update Ruha set Meret=Meret+1;

****

**PL/SQL:**

**Új Ruha típus hozzáadása**

declare

egysor Ruha%rowtype;

cursor kurzor is select \* from Ruha;

van boolean:=false;

kivetel exception;

begin

for egysor in kurzor

loop

if egysor.Tipus='öv(fehér)' then

van:=true;

exit;

end if;

end loop;

if van=false then

dbms\_output.put\_line('Ruha típus hozzáadva');

INSERT INTO Ruha VALUES(7,9700,'öv(fehér)','műbőr',34);

else

raise kivetel;

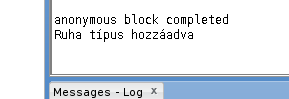
end if;

exception

when kivetel then

dbms\_output.put\_line('Van már ilyen Ruha');

end;

****

**Árak módosítása:**

declare

sor Ruha%rowtype;

cursor kurzor is select \* from Ruha

for update of Ar nowait;

begin

open kurzor;

loop

fetch kurzor into sor;

exit when kurzor%notfound;

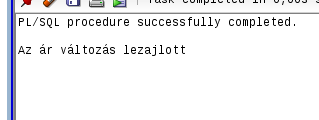
update Ruha set Ar=Ar-100 where current of kurzor;

end loop;

close kurzor;

dbms\_output.put\_line('Az ár változás lezajlott');

end;

****

**Létezik-e a megrendelőnk?**

declare

sor Megrendelo%rowtype;

cursor kurzor is select \* from Megrendelo;

begin

for sor in kurzor

loop

if sor.Nev='Fekete László' then

dbms\_output.put\_line('Létezik ilyen megrendelői név');

dbms\_output.put\_line('Adatai: ' || sor.VasarloID|| '//' || sor.Nev || '//' || sor.Varos || '//' || sor.Cim || '//' ||sor.Iranyitoszam || '//' || sor.Telefonszam);

exit;

end if;

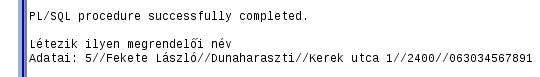
end loop;

exception

when no\_data\_found then

dbms\_output.put\_line('Nincs nevű megrendelőnk');

end;



**Jogosultság kezelés:**

**Szerepkörök létrehozása:**

create role megrendelok;

create role adminisztratorok;

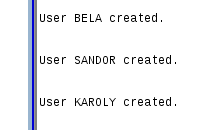
C:\Users\K.Gergo\Desktop\felh.png

**Felhasználók létrehozása:**

create user Bela identified by Belaa;

create user Sandor identified by Sandorr;

create user Karoly identified by Karolyy;

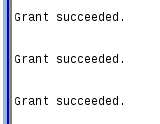


**Szerepkörök jogosultságainak beállítása:**

grant select,insert,update on Megrendeles to megrendelok;

grant select,insert,update on Megrendelo to megrendelok;

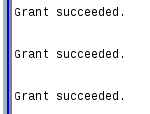
grant select,insert,update on Ruha to megrendelok;



grant all on Megrendeles to adminisztratorok;

grant all on Megrendelo to adminisztratorok;

grant all on Ruha to adminisztratorok;



**Felhasználók szerepkörökhöz rendelése:**

grant megrendelok to Bela;

grant megrendelok to Sandor;

grant adminisztratorok to Karoly;

