# Táblák létrehozása:

CREATE TABLE vevok( id INT, nev varchar(30), adoszam int, irsz int, varos varchar (30), utca varchar (50), hazszam int (3), PRIMARY KEY (id) );

CREATE TABLE termekek( id INT, nev varchar(30), gyartasi\_sz int, gyartasi\_i date, egysegar int, szarmazasi\_h varchar(30), garancia int, minosegi\_o varchar(30), PRIMARY KEY (id) );

CREATE TABLE eladasok( id INT, sorszam int, termekek\_id int, db int, vevok\_id int, PRIMARY KEY (id), constraint vevok\_id foreign key (vevok\_id) references vevok(id), constraint termekek\_id foreign key (termekek\_id) references termekek(id));

create table Naplo(Esemeny varchar(20), Adat varchar(100), Datum timestamp);

# Eljárások az adatok feltöltéséhez:

## Vevők feltöltése:

create or replace procedure feltoltVevo(id INT, nev varchar, adoszam int, irsz int, varos varchar, utca varchar, hazszam int) is

perror exception;

begin

if id<=0 or adoszam<=0 or irsz<=0 or hazszam <=0 then

raise perror;

else

insert into vevok values(id, nev, adoszam, irsz, varos, utca, hazszam);

commit;

end if;

exception

when perror then

dbms\_output.put\_line('Az ID, adószám, irányítószám, házszám mezőknek nagyobbnak kell lenniük, mint nulla!');

end;

## Termékek feltöltése:

create or replace procedure feltoltTermek(id INT, nev varchar, gyartasi\_sz int, gyartasi\_i date, egysegar int, szarmazasi\_h varchar, garancia int, minosegi\_o varchar) is

perror exception;

gerror exception;

begin

if id<=0 or gyartasi\_sz<=0 or egysegar<=0 then

raise perror;

elsif garancia<0 or garancia>3 then

raise gerror;

else

insert into termek values(id, nev, gyartasi\_sz, gyartasi\_i, egysegar, szarmazasi\_h, garancia, minosegi\_o);

commit;

end if;

exception

when perror then

dbms\_output.put\_line('Az ID, gyártási szám, egységár mezőknek nagyobbnak kell lenniük, mint nulla!');

when gerror then

dbms\_output.put\_line('A garancia 0 és 3 év közé eshet!');

end;

## Eladások tábla feltöltése:

create or replace procedure feltoltEladasok(id INT, sorszam int, termekek\_id int, db int, vevok\_id int) is

perror exception;

begin

if id<=0 or sorszam<=0 or termekek\_id<=0 or db<=0 or vevok\_id<=0 then

raise perror;

else

insert into eladasok values(id, sorszam, termekek\_id, db, vevok\_id);

commit;

end if;

exception

when perror then

dbms\_output.put\_line('Az ID, sorszám, termék id, darabszám, vevő id mezőknek nagyobbnak kell lenniük, mint nulla!');

end;

# Feltöltés metódusok kipróbálása:

## Vevők:

begin

feltoltvevo(6, 'Nagy Tibor', 111111111, 4000, 'Győr', 'Hadvezér utca', 2 );

end;

A képen szöveg látható

Automatikusan generált leírás

## Hiba esetén:



## Termékek:

begin

feltolttermek(1001,'Akkumulátor 80Ah',145242, TO\_DATE('1989-12-09','YYYY-MM-DD'),64817,'Európai Unió',2,'Professional');

end;



## Hiba esetén:

A képen szöveg látható

Automatikusan generált leírás



## Eladások:

begin

feltolteladasok(1, 213132, 1001, 500, 4);

end;

A képen szöveg látható

Automatikusan generált leírás

## Hiba esetén:



# Módosító eljárások:

## Vevő táblára vonatkozó módosítások:

Név módosítása:

create or replace procedure nevModVevo(uid int, ujnev char) is

begin

update vevok set nev=ujnev where uid=id;

end;

Házszám módosítása:

create or replace procedure hszModVevo(uid int, ujszam int) is

perror exception;

begin

if ujszam<=0 then

raise perror;

else

update vevok set hazszam=ujszam where uid=id;

end if;

exception

when perror then dbms\_output.put\_line('A házszámnak nagyobbnak kell lennie, mint nulla!');

end;

## Termékek táblára vonatkozó módosítások:

Gyártási idő módosítása:

create or replace procedure gyidModTermek(uid int, ujdatum date) is

begin

update termekek set gyartasi\_i=ujdatum where id=uid;

end;

Egységár módosítása:

create or replace procedure earModTermek(uid int, ear int) is

perror exception;

begin

if ear <=0 then

raise perror;

else

update termekek set egysegar=ear where id=uid;

end if;

exception

when perror then

dbms\_output.put\_line('Az egységárnak nagyobbnak kell lennie, mint nulla!');

end;

## Eladások táblára vonatkozó módosítások:

Termék\_id módosítása:

create or replace procedure t\_iModEladasok(uid int, termek\_idu int) is

perror exception;

terror exception;

t int;

begin

select count(id) into t from termekek where id = uid;

if termek\_idu <=0 then

raise perror;

elsif t<1 then

raise terror;

else

update eladasok set termekek\_id=termek\_idu where id= termek\_idu;

end if;

exception

when perror then

dbms\_output.put\_line('Az id-nek nagyobbnak kell lennie, mint nulla!');

when terror then

dbms\_output.put\_line('A megadott termék id nem található meg az adatbázisban!');

end;

Vevő\_id módosítása:

create or replace procedure v\_iModEladasok(uid int, vevo\_idu int) is

perror exception;

terror exception;

t int;

begin

select count(id) into t from vevok where id = vevo\_idu;

if vevo\_idu <=0 then

raise perror;

elsif t<1 then

raise terror;

else

update eladasok set vevok\_id=vevo\_idu where id=uid;

end if;

exception

when perror then

dbms\_output.put\_line('Az id-nek nagyobbnak kell lennie, mint nulla!');

when terror then

dbms\_output.put\_line('A megadott termék id nem található meg az adatbázisban!');

end;

# Módosító eljárások kipróbálása:

## Vevő nevének módosítása:

begin

nevModVevo(1, 'Kiss István');

end;





## Vevő házszámának módosítása:

begin

hszModVevo(1, 25);

end;



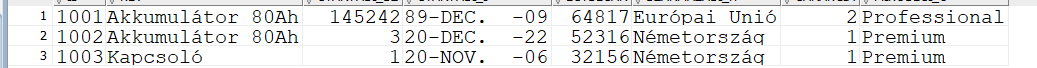


## Termék egységárának módosítása:

begin

earmodtermek(1001, 60000);

end;





## Termék gyártási idejének módosítása:

begin

gyidmodtermek(1001, TO\_DATE('2000-03-09','YYYY-MM-DD'));

end;





## Termék\_id módosítása az eladási táblában:

begin

t\_imodeladasok(1, 1003);

end;

A képen szöveg látható

Automatikusan generált leírás

A képen szöveg látható

Automatikusan generált leírás

## Vevő\_id módosítása az eladási táblában:

begin

v\_imodeladasok(1, 1);

end;

A képen szöveg látható

Automatikusan generált leírás



A módosításoknál is van hibakezelés, például a negatív, illetve nullás értékekre is, azonban az utolsó kettőnél, akár az alábbi hibaüzenettel is találkozhatunk:



# Eljárások rekordok törlésére:

## Vevő törlése:

create or replace procedure torolVevo(idu int) is

perror exception;

t int;

v eladasok.vevok\_id%type;

begin

select id into v from vevok where id=idu;

select count(id) into t from eladasok where vevok\_id = v;

if t > 0 then

raise perror;

else

delete from vevok where id=idu;

end if;

exception

when perror then

dbms\_output.put\_line('A referenciákat a megadott vevőre ki kell törölni a többi táblából!');

end;

## Termék törlése:

create or replace procedure torolTermek(idu int, termek\_idu int) is

perror exception;

t int;

begin

select count(id) into t from eladasok where termekek\_id = termek\_idu;

if t >0 then

raise perror;

else

delete from termekek where id=idu;

end if;

exception

when perror then

dbms\_output.put\_line('A referenciákat a megadott termékre ki kell törölni a többi táblából!');

end;

## Eladások törlése:

create or replace procedure torolEladasok(idu int) is

begin

delete from eladasok where id=idu;

end;

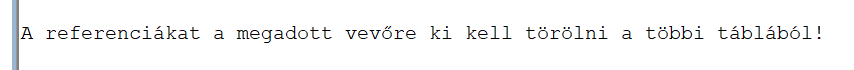
# Törlési eljárások kipróbálása:

## Vevő törlése:

begin

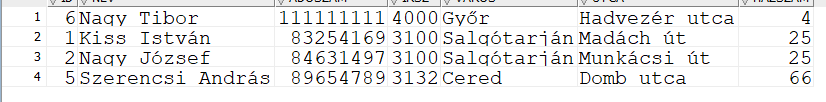
torolvevo(4);

end;



A képen szöveg látható

Automatikusan generált leírás



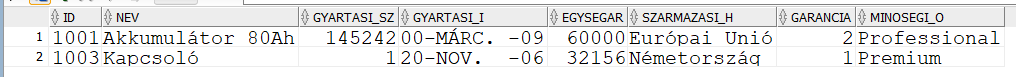
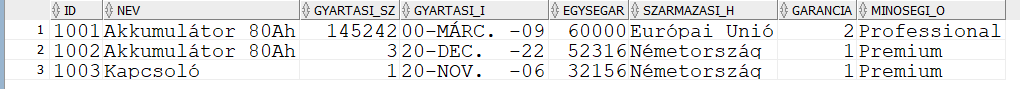
## Termék törlése:

begin

toroltermek(1002);

end;





## Eladás törlése:

begin

toroleladasok(1);

end;

A képen szöveg látható

Automatikusan generált leírás

A képen szöveg látható

Automatikusan generált leírás

# Tárolt függvények:

## Adott városban élő vevők nevei:

create or replace function varosVevo(varosu in char) return varchar is

ember char(30);

cursor cur is select nev from vevok where varos=varosu;

cv cur%rowtype;

begin

open cur;

loop

fetch cur into cv;

exit when cur%notfound;

dbms\_output.put\_line('A ' || varosu || ' városban lakó vevő: ' || cv.nev);

end loop;

close cur;

return ember;

end;

Ellenőrzés:

begin

dbms\_output.put\_line(varosvevo('Salgótarján'));

end;

A képen szöveg látható

Automatikusan generált leírás

## Adott származási helyű termékek átlagára:

create or replace function helyAtlag(szarmazasu in char) return number is

atlag number:=0;

cursor cur is select egysegar from termekek where szarmazasi\_h = szarmazasu;

cv cur%rowtype;

begin

open cur;

loop

fetch cur into cv;

exit when cur%notfound;

atlag := atlag + cv.egysegar;

end loop;

atlag:= atlag/cur%rowcount;

close cur;

return atlag;

end;

Ellenőrzés:

begin

dbms\_output.put\_line('A megadott helyen a termékek átlagára: ' || helyatlag('Európai Unió'));

end;



## Adott darabszám feletti eladások vevőinek neve:

create or replace function dbVevo(darabsz in number) return char is

ember char(30);

cursor cur is select \* from vevok v inner join eladasok e on v.id = e.vevok\_id where e.db>darabsz;

cv cur%rowtype;

begin

open cur;

loop

fetch cur into cv;

exit when cur%notfound;

dbms\_output.put\_line('A ' || darabsz || ' feletti vétellel rendelkező vevő: ' || cv.nev);

end loop;

close cur;

return ember;

end;

Ellenőrzés:

begin

dbms\_output.put\_line(dbvevo(499));

end;

A képen szöveg látható

Automatikusan generált leírás

# Táblák kilistázása függvénnyel:

## Vevő tábla kilistázása:

create or replace function vevoLista return char is

lista char(30);

cursor cur is select \* from vevok;

cv cur%rowtype;

begin

open cur;

loop

fetch cur into cv;

exit when cur%notfound;

dbms\_output.put\_line('Név: ' || cv.nev || ' Város: ' || cv.varos || ' Adószám: ' || cv.adoszam);

end loop;

close cur;

return lista;

end;

Ellenőrzés:

begin

dbms\_output.put\_line(vevoLista());

end;

A képen szöveg látható

Automatikusan generált leírás

## Termékek tábla kilistázása:

create or replace function listaTermek return char is

lista char(50);

cursor c is select \* from termekek;

begin

open c;

for c in (select \* from termekek) loop

dbms\_output.put\_line('Termék neve: ' || c.nev || ' Egységár: ' || c.egysegar || ' Származási hely: ' || c.szarmazasi\_h);

end loop;

close c;

return lista;

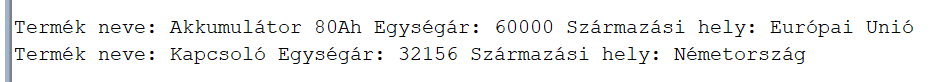
end;

Ellenőrzés:

begin

dbms\_output.put\_line(listatermek());

end;



## Eladások tábla kilistázása:

create or replace function listaEladasok return char is

lista char(50);

cursor c is select \* from eladasok;

begin

open c;

for c in (select \* from eladasok) loop

dbms\_output.put\_line('Termék id: ' || c.termekek\_id || ' Vevő id: ' || c.vevok\_id || ' Darabszám: ' || c.db);

end loop;

close c;

return lista;

end;

Ellenőrzés:

begin

dbms\_output.put\_line(listaeladasok());

end;

A képen szöveg, narancs látható

Automatikusan generált leírás

# Triggerek készítése:

## Insertek esetén:

create or replace trigger NaploBesz after insert on vevok for each row

begin

insert into Naplo values('Beszúrás vevő', :new.id, sysdate);

end;

create or replace trigger NaploBesz2 after insert on termekek for each row

begin

insert into Naplo values('Beszúrás termék', :new.id, sysdate);

end;

create or replace trigger NaploBesz3 after insert on eladasok for each row

begin

insert into Naplo values('Beszúrás eladás', :new.id, sysdate);

end;

## Módosítások esetén:

create or replace trigger Updvev after update on vevok for each row

begin

insert into Naplo values('Modify vevő', :new.id, sysdate);

end;

create or replace trigger Updterm after update on termekek for each row

begin

insert into Naplo values('Modify termék', :new.id, sysdate);

end;

create or replace trigger Updelad after update on eladasok for each row

begin

insert into Naplo values('Modify eladás', :new.id, sysdate);

end;

## Törlések esetében:

create or replace trigger Delvevo after delete on vevok for each row

begin

insert into Naplo values('Delete vevő', :old.id, sysdate);

end;

create or replace trigger Delterm after delete on termekek for each row

begin

insert into Naplo values('Delete termék', :old.id, sysdate);

end;

create or replace trigger Delelad after delete on eladasok for each row

begin

insert into Naplo values('Delete eladás', :old.id, sysdate);

end;

## Trigger készítése, hogy ha a vevő kitörlődik, és volt összeköttetése az eladások táblával, akkor ott is törlődjön ki az eladás:

create or replace trigger vevoEladas after delete on vevok for each row

declare

cursor cur is select \* from eladasok;

cv cur%rowtype;

begin

open cur;

loop

fetch cur into cv;

exit when cur%notfound;

if (cv.vevok\_id = :old.id) then

delete from eladasok where vevok\_id = :old.id;

end if;

end loop;

close cur;

end;

Ellenőrzés:

Vevő tábla eredetileg:

A képen szöveg látható

Automatikusan generált leírás

Eladások tábla eredetileg:

A képen asztal látható

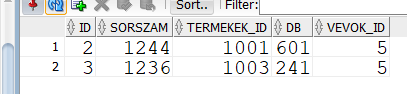
Automatikusan generált leírás

Vevő tábla módosítás után:

A képen szöveg látható

Automatikusan generált leírás

Eladások tábla trigger után:



## Trigger készítése, hogy ha a vevő id-je módosul, és volt összeköttetése az eladások táblával, akkor ott is módosuljon az eladás:

create or replace trigger vevoEladasMod after update on vevok for each row

declare

cursor cur is select \* from eladasok;

cv cur%rowtype;

begin

open cur;

loop

fetch cur into cv;

exit when cur%notfound;

if (cv.vevok\_id = :old.id) then

update eladasok set vevok\_id=:new.id where vevok\_id = :old.id;

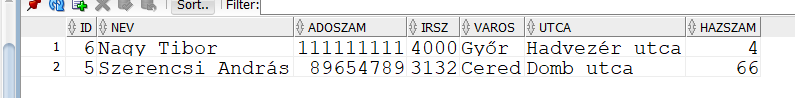
end if;

end loop;

close cur;

end;

Vevő tábla eredetileg:



Eladások tábla eredetileg:

A képen asztal látható

Automatikusan generált leírás

Vevő tábla módosítva:

A képen szöveg látható

Automatikusan generált leírás

Eladások tábla módosítva:

A képen szöveg látható

Automatikusan generált leírás

# Automatikus kulcsgenerálás és inkrementálás:

CREATE SEQUENCE vevok\_sequence;

CREATE SEQUENCE termekek\_sequence;

CREATE SEQUENCE eladasok\_sequence;

create or replace TRIGGER vevok\_on\_insert

BEFORE INSERT ON vevok

FOR EACH ROW

BEGIN

SELECT vevok\_sequence.nextval

INTO :new.id

FROM dual;

END;

create or replace TRIGGER termekek\_on\_insert

BEFORE INSERT ON termekek

FOR EACH ROW

BEGIN

SELECT termekek\_sequence.nextval

INTO :new.id

FROM dual;

END;

create or replace TRIGGER eladasok\_on\_insert

BEFORE INSERT ON eladasok

FOR EACH ROW

BEGIN

SELECT eladasok\_sequence.nextval

INTO :new.id

FROM dual;

END;

# Csomag létrehozása:

## Vevő táblához tartozó csomag létrehozása:

create or replace package VevoPCKG is

function dbVevo(darabsz in number) return char;

function varosVevo(varosu in char) return varchar;

function VevoLista return char;

procedure hszModVevo(uid int, ujszam int);

procedure nevModVevo(uid int, ujnev char);

end VevoPCKG;

create or replace package body VevoPCKG is

function dbVevo(darabsz in number) return char is

ember char(30);

cursor cur is select \* from vevok v inner join eladasok e on v.id = e.vevok\_id where e.db>darabsz;

cv cur%rowtype;

begin

open cur;

loop

fetch cur into cv;

exit when cur%notfound;

dbms\_output.put\_line('A ' || darabsz || ' feletti vétellel rendelkező vevő: ' || cv.nev);

end loop;

close cur;

return ember;

end dbVevo;

function varosVevo(varosu in char) return varchar is

ember char(30);

cursor cur is select nev from vevok where varos=varosu;

cv cur%rowtype;

begin

open cur;

loop

fetch cur into cv;

exit when cur%notfound;

dbms\_output.put\_line('A ' || varosu || ' városban lakó vevő: ' || cv.nev);

end loop;

close cur;

return ember;

end varosVevo;

function vevoLista return char is

lista char(30);

cursor cur is select \* from vevok;

cv cur%rowtype;

begin

open cur;

loop

fetch cur into cv;

exit when cur%notfound;

dbms\_output.put\_line('Név: ' || cv.nev || ' Város: ' || cv.varos || ' Adószám: ' || cv.adoszam);

end loop;

close cur;

return lista;

end vevoLista;

procedure hszModVevo(uid int, ujszam int) is

perror exception;

begin

if ujszam<=0 then

raise perror;

else

update vevok set hazszam=ujszam where uid=id;

end if;

exception

when perror then dbms\_output.put\_line('A házszámnak nagyobbnak kell lennie, mint nulla!');

end hszModVevo;

procedure nevModVevo(uid int, ujnev char) is

begin

update vevok set nev=ujnev where uid=id;

end nevModVevo;

end vevoPCKG;

Ellenőrzés:

begin

dbms\_output.put\_line(vevopckg.dbvevo(400));

vevopckg.hszmodvevo(1,100);

vevopckg.nevmodvevo(1, 'Kiss István');

dbms\_output.put\_line(vevopckg.varosvevo('Cered'));

dbms\_output.put\_line(vevopckg.vevolista());

end;

A képen szöveg, beltéri látható

Automatikusan generált leírás

A képen asztal látható

Automatikusan generált leírás