--betesz egy alkalmazotat vagy

CREATE OR REPLACE FUNCTION Regisztracio(nev felhasznalok.felhasznalonev%TYPE, pss felhasznalok.jelszo%TYPE)

RETURN NUMBER IS db NUMBER;

max\_id NUMBER;

BEGIN

SELECT felhasznalo\_id INTO max\_id FROM Felhasznalok WHERE felhasznalonev = nev;

RETURN max\_id;

EXCEPTION WHEN NO\_DATA\_FOUND THEN

SELECT MAX(felhasznalo\_id)+1 INTO max\_id FROM Felhasznalok ;

db := elsoID;

INSERT INTO Felhasznalok (felhasznalo\_id,felhasznalonev, jelszo, szerep\_id)

VALUES (db,nev, pss, (SELECT szerep\_id FROM Szerepkorok WHERE szerepnev = 'csakolvas'));

RETURN db;

WHEN OTHERS THEN DBMS\_OUTPUT.PUT\_LINE('Valami hiba!');

RECORD\_ERROR;

END Regisztracio;

/

CREATE OR REPLACE FUNCTION elsoID

RETURN NUMBER IS

v\_min\_id NUMBER;

CURSOR id\_cursor IS SELECT felhasznalo\_id FROM felhasznalok ORDER BY felhasznalo\_id;

BEGIN

v\_min\_id := NULL;

FOR id\_rec IN id\_cursor LOOP

IF id\_rec.felhasznalo\_id IS NULL THEN

v\_min\_id := 1;

ELSIF id\_rec.felhasznalo\_id > v\_min\_id + 1 THEN

v\_min\_id := v\_min\_id + 1;

RETURN v\_min\_id;

ELSE

v\_min\_id := id\_rec.felhasznalo\_id;

END IF;

END LOOP;

RETURN v\_min\_id + 1;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN 1;

END elsoID;

/

CREATE OR REPLACE FUNCTION felhasznaloTorles(nev felhasznalok.felhasznalonev%TYPE, pss felhasznalok.jelszo%TYPE)

RETURN NUMBER IS db NUMBER;

max\_id NUMBER;

BEGIN

SELECT felhasznalo\_id INTO max\_id FROM Felhasznalok WHERE felhasznalonev = nev;

DELETE felhasznalok

WHERE felhasznalo\_id = max\_id;

RETURN max\_id;

EXCEPTION WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('Nincs ilyen felhasznalo');

RETURN -1;

WHEN OTHERS THEN DBMS\_OUTPUT.PUT\_LINE('Valami hiba!');

RECORD\_ERROR;

END felhasznaloTorles;

/

SELECT \* FROM auto\_tipus;

/

CREATE OR REPLACE FUNCTION EvLekeredezes(tipus auto\_tipus.auto\_tipus%TYPE)

RETURN NUMBER IS db NUMBER;

max\_id NUMBER;

BEGIN

SELECT auto\_tipus\_id INTO max\_id FROM auto\_tipus WHERE auto\_tipus = tipus;

SELECT auto\_ev INTO db FROM autok WHERE auto\_tipus\_id = max\_id;

RETURN db;

EXCEPTION WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('Nincs ilyen auto');

RETURN -1;

WHEN OTHERS THEN DBMS\_OUTPUT.PUT\_LINE('Valami hiba!');

RECORD\_ERROR;

END EvLekeredezes;

/

CREATE OR REPLACE FUNCTION TeljesitmenyLekeredezes(tipus auto\_tipus.auto\_tipus%TYPE)

RETURN NUMBER IS db NUMBER;

max\_id NUMBER;

BEGIN

SELECT auto\_tipus\_id INTO max\_id FROM auto\_tipus WHERE auto\_tipus = tipus;

SELECT auto\_teljesitmeny INTO db FROM autok WHERE auto\_tipus\_id = max\_id;

RETURN db;

EXCEPTION WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('Nincs ilyen auto');

RETURN -1;

WHEN OTHERS THEN DBMS\_OUTPUT.PUT\_LINE('Valami hiba!');

RECORD\_ERROR;

END TeljesitmenyLekeredezes;

/

CREATE OR REPLACE FUNCTION ArLekeredezes(tipus auto\_tipus.auto\_tipus%TYPE)

RETURN NUMBER IS db NUMBER;

max\_id NUMBER;

BEGIN

SELECT auto\_tipus\_id INTO max\_id FROM auto\_tipus WHERE auto\_tipus = tipus;

SELECT auto\_ar INTO db FROM autok WHERE auto\_tipus\_id = max\_id;

RETURN db;

EXCEPTION WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('Nincs ilyen auto');

RETURN -1;

WHEN OTHERS THEN DBMS\_OUTPUT.PUT\_LINE('Valami hiba!');

RECORD\_ERROR;

END ArLekeredezes;

/

CREATE OR REPLACE FUNCTION UzemanyagLekeredezes(tipus auto\_tipus.auto\_tipus%TYPE)

RETURN VARCHAR2 IS db VARCHAR2(250);

max\_id NUMBER;

BEGIN

SELECT auto\_tipus\_id INTO max\_id FROM auto\_tipus WHERE auto\_tipus = tipus;

SELECT auto\_uzemanyag INTO db FROM autok WHERE auto\_tipus\_id = max\_id;

RETURN db;

EXCEPTION WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('Nincs ilyen auto');

RETURN -1;

WHEN OTHERS THEN DBMS\_OUTPUT.PUT\_LINE('Valami hiba!');

RECORD\_ERROR;

END UzemanyagLekeredezes;

/

CREATE OR REPLACE FUNCTION dbLekeredezes(tipus auto\_tipus.auto\_tipus%TYPE)

RETURN NUMBER IS darab NUMBER;

max\_id NUMBER;

BEGIN

SELECT auto\_tipus\_id INTO max\_id FROM auto\_tipus WHERE auto\_tipus = tipus;

SELECT db INTO darab FROM autok WHERE auto\_tipus\_id = max\_id;

RETURN darab;

EXCEPTION WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('Nincs ilyen auto');

RETURN -1;

WHEN OTHERS THEN DBMS\_OUTPUT.PUT\_LINE('Valami hiba!');

RECORD\_ERROR;

END dbLekeredezes;

/

CREATE OR REPLACE FUNCTION ArAlkatresz(nev Alkatreszek.alkatresz\_nev%TYPE)

RETURN NUMBER IS db NUMBER;

max\_id NUMBER;

BEGIN

SELECT alkatresz\_ar INTO db FROM Alkatreszek WHERE alkatresz\_nev = nev;

RETURN db;

EXCEPTION WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('Nincs ilyen auto');

RETURN -1;

WHEN OTHERS THEN DBMS\_OUTPUT.PUT\_LINE('Valami hiba!');

RECORD\_ERROR;

END ArAlkatresz;

/

CREATE OR REPLACE FUNCTION DbAlkatresz(nev Alkatreszek.alkatresz\_nev%TYPE)

RETURN NUMBER IS db NUMBER;

max\_id NUMBER;

BEGIN

SELECT db INTO db FROM Alkatreszek WHERE alkatresz\_nev = nev;

RETURN db;

EXCEPTION WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('Nincs ilyen auto');

RETURN -1;

WHEN OTHERS THEN DBMS\_OUTPUT.PUT\_LINE('Valami hiba!');

RECORD\_ERROR;

END DbAlkatresz;

/

CREATE OR REPLACE FUNCTION MarkaAlkatresz(tipus auto\_tipus.auto\_tipus%TYPE)

RETURN VARCHAR2 IS db VARCHAR2(250);

BEGIN

SELECT alkatresz\_marka INTO db FROM Alkatreszek WHERE alkatresz\_nev = tipus;

RETURN db;

EXCEPTION WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('Nincs ilyen auto');

RETURN -1;

WHEN OTHERS THEN DBMS\_OUTPUT.PUT\_LINE('Valami hiba!');

RECORD\_ERROR;

END MarkaAlkatresz;

/

--az ugyfelek telefonszamat adja vissza a vezeteknev es a keresztnev alapjan, -1 hiba esten

CREATE OR REPLACE FUNCTION ugyfeltelo(vnev Ugyfelek.ugyfel\_vezeteknev%TYPE,knev Ugyfelek.ugyfel\_keresztnev%TYPE)

RETURN VARCHAR2 IS db VARCHAR2(250);

BEGIN

SELECT ugyfel\_telefonszam INTO db FROM Ugyfelek WHERE ugyfel\_vezeteknev = vnev AND ugyfel\_keresztnev = knev;

RETURN db;

EXCEPTION WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('Nincs ilyen ugyfelek');

RETURN -1;

WHEN OTHERS THEN DBMS\_OUTPUT.PUT\_LINE('Valami hiba!');

RECORD\_ERROR;

END ugyfeltelo;

/

--az ugyfelek emailcimet adja vissza a vezeteknev es a keresztnev alapjan, -1 hiba esten

CREATE OR REPLACE FUNCTION ugyfelemail(vnev Ugyfelek.ugyfel\_vezeteknev%TYPE,knev Ugyfelek.ugyfel\_keresztnev%TYPE)

RETURN VARCHAR2 IS db VARCHAR2(250);

BEGIN

SELECT ugyfel\_email INTO db FROM Ugyfelek WHERE ugyfel\_vezeteknev = vnev AND ugyfel\_keresztnev = knev;

RETURN db;

EXCEPTION WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('Nincs ilyen ugyfelek');

RETURN -1;

WHEN OTHERS THEN DBMS\_OUTPUT.PUT\_LINE('Valami hiba!');

RECORD\_ERROR;

END ugyfelemail;

/

--az ugyfelek orszagat adja vissza a vezeteknev es a keresztnev alapjan, -1 hiba esten

CREATE OR REPLACE FUNCTION ugyfelorszag(vnev Ugyfelek.ugyfel\_vezeteknev%TYPE,knev Ugyfelek.ugyfel\_keresztnev%TYPE)

RETURN VARCHAR2 IS db VARCHAR2(250);

BEGIN

SELECT ugyfel\_orszag INTO db FROM Ugyfelek WHERE ugyfel\_vezeteknev = vnev AND ugyfel\_keresztnev = knev;

RETURN db;

EXCEPTION WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('Nincs ilyen ugyfelek');

RETURN -1;

WHEN OTHERS THEN DBMS\_OUTPUT.PUT\_LINE('Valami hiba!');

RECORD\_ERROR;

END ugyfelorszag;

/

--az ugyfelek telepuleset adja vissza a vezeteknev es a keresztnev alapjan, -1 hiba esten

CREATE OR REPLACE FUNCTION ugyfeltelepules(vnev Ugyfelek.ugyfel\_vezeteknev%TYPE,knev Ugyfelek.ugyfel\_keresztnev%TYPE)

RETURN VARCHAR2 IS db VARCHAR2(250);

BEGIN

SELECT ugyfel\_telepules INTO db FROM Ugyfelek WHERE ugyfel\_vezeteknev = vnev AND ugyfel\_keresztnev = knev;

RETURN db;

EXCEPTION WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('Nincs ilyen ugyfelek');

RETURN -1;

WHEN OTHERS THEN DBMS\_OUTPUT.PUT\_LINE('Valami hiba!');

RECORD\_ERROR;

END ugyfeltelepules;

/

--az ugyfelek lakcimet adja vissza a vezeteknev es a keresztnev alapjan, -1 hiba esten

CREATE OR REPLACE FUNCTION ugyfellakcim(vnev Ugyfelek.ugyfel\_vezeteknev%TYPE,knev Ugyfelek.ugyfel\_keresztnev%TYPE)

RETURN VARCHAR2 IS db VARCHAR2(250);

BEGIN

SELECT ugyfel\_lakcim INTO db FROM Ugyfelek WHERE ugyfel\_vezeteknev = vnev AND ugyfel\_keresztnev = knev;

RETURN db;

EXCEPTION WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('Nincs ilyen ugyfelek');

RETURN -1;

WHEN OTHERS THEN DBMS\_OUTPUT.PUT\_LINE('Valami hiba!');

RECORD\_ERROR;

END ugyfellakcim;

/

--a rendeles neve alapjan kiirja az arat, ki rendelte, kinek es mikor

CREATE OR REPLACE FUNCTION rendeles(rndl Rendelesek.rendeles\_nev%TYPE)

RETURN VARCHAR2 IS

db INT;

ar INT;

datum date;

alkid VARCHAR2(250);

alkalmazott VARCHAR2(250);

ugyid VARCHAR2(250);

ugyfel VARCHAR2(250);

res VARCHAR2(250);

BEGIN

SELECT rendeles\_darab INTO db FROM Rendelesek WHERE rndl = rendeles\_nev;

SELECT rendeles\_ar INTO ar FROM Rendelesek WHERE rndl = rendeles\_nev;

SELECT rendeles\_datum INTO datum FROM Rendelesek WHERE rndl = rendeles\_nev;

SELECT alkalmazott\_id INTO alkid FROM Rendelesek WHERE rndl = rendeles\_nev;

SELECT ugyfel\_id INTO ugyid FROM Rendelesek WHERE rndl = rendeles\_nev;

SELECT alkalmazott\_id INTO alkid FROM Rendelesek WHERE rndl = rendeles\_nev;

SELECT ugyfel\_vezeteknev || ugyfel\_keresztnev AS Nev INTO ugyfel FROM Ugyfelek WHERE ugyid = ugyfel\_id;

SELECT alkalmazot\_vezeteknev || alkalmazott\_keresztnev AS Nev INTO alkalmazott FROM Alkalmazottak WHERE alkid = alkalmazott\_id;

res := 'A rendelest ' || alkalmazott || ' tette le '||ugyfel||'-nek, '||db||' darabot '||ar|| 'ertekben '||datum||' datumon.';

RETURN res;

EXCEPTION WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('Nincs ilyen ugyfelek');

RETURN -1;

WHEN OTHERS THEN DBMS\_OUTPUT.PUT\_LINE('Valami hiba!');

RECORD\_ERROR;

END rendeles;

/

--a rendeles neve alapjan kiirja az arat, ki rendelte, kinek es mikor

CREATE OR REPLACE FUNCTION alkalmazottlekerdezes(vnev Alkalmazottak.ALKALMAZOT\_VEZETEKNEV%TYPE, knev Alkalmazottak.ALKALMAZOTT\_KERESZTNEV%TYPE)

RETURN VARCHAR2 IS

fizetes Alkalmazottak.ALKALMAZOTT\_FIZETES%TYPE;

teloszam Alkalmazottak.ALKALMAZOTT\_TELEFONSZAM%TYPE;

oszid Alkalmazottak.OSZTALY\_ID%TYPE;

onev Osztalyok.OSZTALY\_NEV%TYPE;

alkalmazott Osztalyok.OSZTALY\_EMAIL%TYPE;

oleiras Osztalyok.OSZTALY\_ROVIDITES%TYPE;

res VARCHAR(3000);

oemail Osztalyok.OSZTALY\_EMAIL%TYPE;

BEGIN

SELECT ALKALMAZOTT\_FIZETES INTO fizetes FROM Alkalmazottak WHERE vnev = ALKALMAZOT\_VEZETEKNEV AND knev = ALKALMAZOTT\_KERESZTNEV;

SELECT ALKALMAZOTT\_TELEFONSZAM INTO teloszam FROM Alkalmazottak WHERE vnev = ALKALMAZOT\_VEZETEKNEV AND knev = ALKALMAZOTT\_KERESZTNEV;

SELECT OSZTALY\_ID INTO oszid FROM Alkalmazottak WHERE vnev = ALKALMAZOT\_VEZETEKNEV AND knev = ALKALMAZOTT\_KERESZTNEV;

SELECT OSZTALY\_NEV INTO onev FROM Osztalyok WHERE oszid = OSZTALY\_ID;

SELECT OSZTALY\_ROVIDITES INTO oleiras FROM Osztalyok WHERE oszid = OSZTALY\_ID;

SELECT OSZTALY\_EMAIL INTO oemail FROM Osztalyok WHERE oszid = OSZTALY\_ID;

SELECT alkalmazot\_vezeteknev || alkalmazott\_keresztnev INTO alkalmazott FROM Alkalmazottak WHERE vnev = ALKALMAZOT\_VEZETEKNEV AND knev = ALKALMAZOTT\_KERESZTNEV;

res := 'Az '||onev|| ' osztalyban dolgozik, fizetese: '||fizetes||', telefonszama: '||teloszam||'. Az osztaly emailje: '||oemail|| ', leirasa: '||oleiras||'.';

RETURN res;

EXCEPTION WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('Nincs ilyen alkalmazott');

RETURN -1;

WHEN OTHERS THEN DBMS\_OUTPUT.PUT\_LINE('Valami hiba!');

RECORD\_ERROR;

END alkalmazottlekerdezes;

/

CREATE OR REPLACE FUNCTION ugyfellekerdezes(vnev Ugyfelek.UGYFEL\_VEZETEKNEV%TYPE, knev Ugyfelek.UGYFEL\_KERESZTNEV%TYPE)

RETURN VARCHAR2 IS

teloszam Ugyfelek.UGYFEL\_TELEFONSZAM%TYPE;

orszag Ugyfelek.UGYFEL\_ORSZAG%TYPE;

telepules Ugyfelek.UGYFEL\_TELEPULES%TYPE;

lakcim Ugyfelek.UGYFEL\_LAKCIM%TYPE;

res VARCHAR(3000);

email Ugyfelek.UGYFEL\_EMAIL%TYPE;

BEGIN

SELECT UGYFEL\_TELEFONSZAM INTO teloszam FROM Ugyfelek WHERE vnev = UGYFEL\_VEZETEKNEV AND knev = UGYFEL\_KERESZTNEV;

SELECT UGYFEL\_TELEPULES INTO telepules FROM Ugyfelek WHERE vnev = UGYFEL\_VEZETEKNEV AND knev = UGYFEL\_KERESZTNEV;

SELECT UGYFEL\_ORSZAG INTO orszag FROM Ugyfelek WHERE vnev = UGYFEL\_VEZETEKNEV AND knev = UGYFEL\_KERESZTNEV;

SELECT UGYFEL\_LAKCIM INTO lakcim FROM Ugyfelek WHERE vnev = UGYFEL\_VEZETEKNEV AND knev = UGYFEL\_KERESZTNEV;

SELECT UGYFEL\_EMAIL INTO email FROM Ugyfelek WHERE vnev = UGYFEL\_VEZETEKNEV AND knev = UGYFEL\_KERESZTNEV;

res := 'Az telefonszama: '||teloszam|| ', az emailcime: '||email||', lakhely: '||orszag||', '||telepules|| ', '||lakcim||'.';

RETURN res;

EXCEPTION WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('Nincs ilyen alkalmazott');

RETURN -1;

WHEN OTHERS THEN DBMS\_OUTPUT.PUT\_LINE('Valami hiba!');

RECORD\_ERROR;

END ugyfellekerdezes;

/

CREATE OR REPLACE FUNCTION ujRendeles(

nev Rendelesek.RENDELES\_NEV%TYPE,

darab Alkatreszek.DB%TYPE,

avnev Alkalmazottak.ALKALMAZOTT\_KERESZTNEV%TYPE,

aknev Alkalmazottak.ALKALMAZOT\_VEZETEKNEV%TYPE,

uvnev Ugyfelek.UGYFEL\_KERESZTNEV%TYPE,

uknev Ugyfelek.UGYFEL\_VEZETEKNEV%TYPE,

termek Rendelesek.RENDELES\_NEV%TYPE

) RETURN VARCHAR2 IS

alkid Alkalmazottak.ALKALMAZOTT\_ID%TYPE;

ugyid Ugyfelek.UGYFEL\_ID%TYPE;

ar NUMBER;

resar NUMBER;

t NUMBER;

pn NUMBER;

szam NUMBER;

rendid NUMBER;

elsoid Ugyfelek.UGYFEL\_ID%TYPE;

res Alkalmazottak.ALKALMAZOTT\_KERESZTNEV%TYPE;

BEGIN

SAVEPOINT kezdet;

BEGIN

SELECT RENDELES\_AR INTO elsoid FROM Rendelesek WHERE RENDELES\_NEV = nev;

RETURN 'A rendeles ara: ' || elsoid;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

NULL;

WHEN OTHERS THEN

RETURN 'Hiba történt a rendelés ID lekérdezése közben.';

END;

BEGIN

IF (nev IS NOT NULL AND darab IS NOT NULL AND avnev IS NOT NULL AND aknev IS NOT NULL AND uvnev IS NOT NULL AND uknev IS NOT NULL AND termek IS NOT NULL) THEN

SELECT ALKATRESZ\_AR INTO ar FROM Alkatreszek WHERE ALKATRESZ\_NEV = termek;

SELECT ALKATRESZ\_ID INTO t FROM Alkatreszek WHERE termek = ALKATRESZ\_NEV;

alkid := alkalmazottid(avnev, aknev);

ugyid := ugyfelid(uvnev, uknev);

SELECT DB INTO szam FROM Alkatreszek WHERE termek = ALKATRESZ\_NEV;

SELECT PENZ INTO pn FROM Ugyfelek WHERE ugyid = UGYFEL\_ID;

IF szam < darab THEN

RETURN 'Nincs eleg: ' || termek || ' a raktarban';

ELSE

resar := darab \* ar;

IF pn < resar THEN

RETURN 'Nincs eleg osszeg a szamlan ';

END IF;

elsoid := elsoID2;

INSERT INTO Rendelesek (RENDELES\_ID,RENDELES\_NEV, RENDELES\_DARAB, RENDELES\_AR, RENDELES\_DATUM, ALKALMAZOTT\_ID, UGYFEL\_ID, TERMEK)

VALUES (elsoid,nev, darab, resar, SYSDATE, alkid, ugyid, termek);

COMMIT;

RETURN 'Sikeres rendelés: ' || nev||' Az ara: ' || resar;

END IF;

ELSE

RETURN 'Hiányzó adatok, nem sikerült a rendelés.';

END IF;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK TO kezdet;

RETURN 'Hiba történt a rendelés beszúrása közben.';

END;

END ujRendeles;

/

SELECT \* FROM alkatreszek;

/

CREATE OR REPLACE FUNCTION alkalmazottid(vnev Alkalmazottak.ALKALMAZOT\_VEZETEKNEV%TYPE,knev Alkalmazottak.ALKALMAZOTT\_KERESZTNEV%TYPE)

RETURN NUMBER IS

alkid Alkalmazottak.ALKALMAZOTT\_ID%TYPE;

BEGIN

SELECT ALKALMAZOTT\_ID INTO alkid FROM Alkalmazottak WHERE ALKALMAZOT\_VEZETEKNEV LIKE vnev AND ALKALMAZOTT\_KERESZTNEV LIKE knev;

--DBMS\_OUTPUT.PUT\_LINE('Az alkalmazott ID: ' || alkid);

RETURN alkid;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('Helytelen adat');

RETURN -1;

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('Valami hiba!');

RETURN -1;

END alkalmazottid;

/

CREATE OR REPLACE FUNCTION ugyfelid(vnev Alkalmazottak.ALKALMAZOT\_VEZETEKNEV%TYPE,knev Alkalmazottak.ALKALMAZOTT\_KERESZTNEV%TYPE)

RETURN NUMBER IS

alkid Alkalmazottak.ALKALMAZOTT\_ID%TYPE;

BEGIN

SELECT UGYFEL\_ID INTO alkid FROM Ugyfelek WHERE UGYFEL\_VEZETEKNEV LIKE vnev AND UGYFEL\_KERESZTNEV LIKE knev;

--DBMS\_OUTPUT.PUT\_LINE('Az alkalmazott ID: ' || alkid);

RETURN alkid;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('Helytelen adat');

RETURN -1;

WHEN OTHERS THEN

DBMS\_OUTPUT.PUT\_LINE('Valami hiba!');

RETURN -1;

END ugyfelid;

/

CREATE OR REPLACE FUNCTION elsoID2

RETURN NUMBER IS

v\_min\_id NUMBER;

CURSOR id\_cursor IS SELECT RENDELES\_ID FROM Rendelesek ORDER BY RENDELES\_ID;

BEGIN

v\_min\_id := NULL;

FOR id\_rec IN id\_cursor LOOP

IF id\_rec.RENDELES\_ID IS NULL THEN

v\_min\_id := 1;

ELSIF id\_rec.RENDELES\_ID > v\_min\_id + 1 THEN

v\_min\_id := v\_min\_id + 1;

RETURN v\_min\_id;

ELSE

v\_min\_id := id\_rec.RENDELES\_ID;

END IF;

END LOOP;

RETURN v\_min\_id + 1;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN 1;

END elsoID2;

/

CREATE OR REPLACE FUNCTION ujAlkalmazott(

vnev Alkalmazottak.ALKALMAZOT\_VEZETEKNEV%TYPE,

knev Alkalmazottak.ALKALMAZOTT\_KERESZTNEV%TYPE,

fizetes Alkalmazottak.ALKALMAZOTT\_FIZETES%TYPE,

telefonszam Alkalmazottak.ALKALMAZOTT\_TELEFONSZAM%TYPE,

osztaly Osztalyok.OSZTALY\_NEV%TYPE

) RETURN VARCHAR2 IS

osztalyid Osztalyok.OSZTALY\_ID%TYPE;

alkid Alkalmazottak.ALKALMAZOTT\_ID%TYPE;

elsoid NUMBER;

BEGIN

SAVEPOINT kezdet;

BEGIN

SELECT OSZTALY\_ID INTO osztalyid FROM Osztalyok WHERE OSZTALY\_NEV = osztaly;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN 'Hiba: A megadott osztály nem létezik.';

WHEN OTHERS THEN

RETURN 'Hiba történt az osztály ID lekérdezése közben.';

END;

BEGIN

SELECT ALKALMAZOTT\_ID INTO alkid FROM Alkalmazottak WHERE ALKALMAZOT\_VEZETEKNEV = vnev AND ALKALMAZOTT\_KERESZTNEV = knev;

RETURN 'Az alkalmazott már létezik: ' || vnev || ' ' || knev;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

NULL; -- Ha nincs ilyen alkalmazott, folytatjuk a beszúrással

WHEN OTHERS THEN

RETURN 'Hiba történt az alkalmazott ID lekérdezése közben.';

END;

-- Alkalmazott beszúrása

BEGIN

IF (vnev IS NOT NULL AND knev IS NOT NULL AND fizetes IS NOT NULL AND telefonszam IS NOT NULL AND osztalyid IS NOT NULL) THEN

-- ElsoID meghívása (feltételezve, hogy elsoID3 egy értéket ad vissza)

elsoid := elsoID3;

INSERT INTO Alkalmazottak (ALKALMAZOTT\_ID, ALKALMAZOT\_VEZETEKNEV, ALKALMAZOTT\_KERESZTNEV, ALKALMAZOTT\_FIZETES, ALKALMAZOTT\_TELEFONSZAM, OSZTALY\_ID)

VALUES (elsoid, vnev, knev, fizetes, telefonszam, osztalyid);

COMMIT;

RETURN 'Sikeres beszúrás: ' || vnev || ' ' || knev;

ELSE

RETURN 'Hiányzó adatok, nem sikerült a beszúrás.';

END IF;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK TO kezdet;

RETURN 'Hiba történt az alkalmazott beszúrása közben.';

END;

END ujAlkalmazott;

/

CREATE OR REPLACE FUNCTION elsoID3

RETURN NUMBER IS

v\_min\_id NUMBER;

CURSOR id\_cursor IS SELECT ALKALMAZOTT\_ID FROM Alkalmazottak ORDER BY ALKALMAZOTT\_ID;

BEGIN

v\_min\_id := NULL;

FOR id\_rec IN id\_cursor LOOP

IF id\_rec.ALKALMAZOTT\_ID IS NULL THEN

v\_min\_id := 1;

ELSIF id\_rec.ALKALMAZOTT\_ID > v\_min\_id + 1 THEN

v\_min\_id := v\_min\_id + 1;

RETURN v\_min\_id;

ELSE

v\_min\_id := id\_rec.ALKALMAZOTT\_ID;

END IF;

END LOOP;

RETURN v\_min\_id + 1;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN 1;

END elsoID3;

/

CREATE OR REPLACE FUNCTION ujUgyfel(

vnev Ugyfelek.UGYFEL\_VEZETEKNEV%TYPE,

knev Ugyfelek.UGYFEL\_KERESZTNEV%TYPE,

teloszam Ugyfelek.UGYFEL\_TELEFONSZAM%TYPE,

email Ugyfelek.UGYFEL\_EMAIL%TYPE,

orszag Ugyfelek.UGYFEL\_ORSZAG%TYPE,

telepules Ugyfelek.UGYFEL\_TELEPULES%TYPE,

lakcim Ugyfelek.UGYFEL\_LAKCIM%TYPE

) RETURN VARCHAR2 IS

ugyid Ugyfelek.UGYFEL\_ID%TYPE;

elsoid NUMBER;

BEGIN

SAVEPOINT kezdet;

BEGIN

SELECT UGYFEL\_ID INTO ugyid FROM Ugyfelek WHERE UGYFEL\_VEZETEKNEV = vnev AND UGYFEL\_KERESZTNEV = knev;

RETURN 'Ügyfél már létezik: ' || vnev || ' ' || knev;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

NULL;

WHEN OTHERS THEN

RETURN 'Hiba történt az ügyfél ID lekérdezése közben.';

END;

BEGIN

IF (vnev IS NOT NULL AND knev IS NOT NULL AND teloszam IS NOT NULL AND email IS NOT NULL AND orszag IS NOT NULL AND telepules IS NOT NULL AND lakcim IS NOT NULL) THEN

elsoid := elsoID4;

INSERT INTO Ugyfelek (UGYFEL\_ID, UGYFEL\_VEZETEKNEV, UGYFEL\_KERESZTNEV, UGYFEL\_TELEFONSZAM, UGYFEL\_EMAIL, UGYFEL\_ORSZAG, UGYFEL\_TELEPULES, UGYFEL\_LAKCIM)

VALUES (elsoid, vnev, knev, teloszam, email, orszag, telepules, lakcim);

COMMIT;

RETURN 'Sikeres beszúrás: ' || vnev || ' ' || knev;

ELSE

RETURN 'Hiányzó adatok, nem sikerült a beszúrás.';

END IF;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK TO kezdet;

RETURN 'Hiba történt az ügyfél beszúrása közben.';

END;

END ujUgyfel;

/

CREATE OR REPLACE FUNCTION elsoID4

RETURN NUMBER IS

v\_min\_id NUMBER;

CURSOR id\_cursor IS SELECT UGYFEL\_ID FROM Ugyfelek ORDER BY UGYFEL\_ID;

BEGIN

v\_min\_id := NULL;

FOR id\_rec IN id\_cursor LOOP

IF id\_rec.UGYFEL\_ID IS NULL THEN

v\_min\_id := 1;

ELSIF id\_rec.UGYFEL\_ID > v\_min\_id + 1 THEN

v\_min\_id := v\_min\_id + 1;

RETURN v\_min\_id;

ELSE

v\_min\_id := id\_rec.UGYFEL\_ID;

END IF;

END LOOP;

RETURN v\_min\_id + 1;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN 1;

END elsoID4;

/

CREATE OR REPLACE FUNCTION rendelesTorlese(nev Rendelesek.RENDELES\_NEV%TYPE)

RETURN VARCHAR2 IS

BEGIN

DELETE FROM Rendelesek WHERE RENDELES\_NEV = nev;

IF SQL%ROWCOUNT = 0 THEN

RETURN 'Nincs ilyen rendeles: ' || nev;

ELSE

RETURN 'Rendeles sikeresen torolve: ' || nev;

END IF;

EXCEPTION

WHEN OTHERS THEN

RETURN 'Hiba történt a rendelés törlése közben';

END rendelesTorlese;

/

CREATE OR REPLACE TRIGGER rendeles\_torlese\_trigger

AFTER DELETE ON Rendelesek

FOR EACH ROW

BEGIN

UPDATE Alkatreszek

SET DB = DB + :OLD.RENDELES\_DARAB

WHERE ALKATRESZ\_NEV = :OLD.TERMEK;

DBMS\_OUTPUT.PUT\_LINE('Rendeles torolve, a termek darabszama visszaallitva: ' || :OLD.TERMEK);

END;

/

CREATE OR REPLACE FUNCTION ugyfelTorlese(vnev Ugyfelek.UGYFEL\_VEZETEKNEV%TYPE, knev Ugyfelek.UGYFEL\_KERESZTNEV%TYPE)

RETURN VARCHAR2 IS

ugyid Ugyfelek.UGYFEL\_ID%TYPE;

BEGIN

SAVEPOINT kezdet;

BEGIN

SELECT UGYFEL\_ID INTO ugyid FROM Ugyfelek WHERE UGYFEL\_VEZETEKNEV = vnev AND UGYFEL\_KERESZTNEV = knev;

DELETE FROM Ugyfelek WHERE UGYFEL\_ID = ugyid;

COMMIT;

RETURN 'Sikeres törlés: ' || vnev || ' ' || knev;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN 'Nincs ilyen ügyfél: ' || vnev || ' ' || knev;

WHEN OTHERS THEN

ROLLBACK TO kezdet;

RETURN 'Hiba történt az ügyfél törlése közben.';

END;

END ugyfelTorlese;

/

CREATE OR REPLACE TRIGGER ugyfel\_torlese\_trigger

AFTER DELETE ON Ugyfelek

FOR EACH ROW

BEGIN

DBMS\_OUTPUT.PUT\_LINE('Ügyfél törölve: ' || :OLD.UGYFEL\_VEZETEKNEV || ' ' || :OLD.UGYFEL\_KERESZTNEV);

END;

/

CREATE OR REPLACE FUNCTION torlesAlkalmazott(

vnev Alkalmazottak.ALKALMAZOT\_VEZETEKNEV%TYPE,

knev Alkalmazottak.ALKALMAZOTT\_KERESZTNEV%TYPE

) RETURN VARCHAR2 IS

alkid Alkalmazottak.ALKALMAZOTT\_ID%TYPE;

BEGIN

SAVEPOINT kezdet;

BEGIN

SELECT ALKALMAZOTT\_ID INTO alkid FROM Alkalmazottak WHERE ALKALMAZOT\_VEZETEKNEV = vnev AND ALKALMAZOTT\_KERESZTNEV = knev;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

RETURN 'Az alkalmazott nem létezik: ' || vnev || ' ' || knev;

WHEN OTHERS THEN

RETURN 'Hiba történt az alkalmazott ID lekérdezése közben.';

END;

BEGIN

DELETE FROM Alkalmazottak WHERE ALKALMAZOTT\_ID = alkid;

IF SQL%ROWCOUNT = 0 THEN

RETURN 'Az alkalmazott törlése sikertelen.';

ELSE

COMMIT;

RETURN 'Sikeres törlés: ' || vnev || ' ' || knev;

END IF;

EXCEPTION

WHEN OTHERS THEN

ROLLBACK TO kezdet;

RETURN 'Hiba történt az alkalmazott törlése közben.';

END;

END torlesAlkalmazott;

/

CREATE OR REPLACE TRIGGER alkalmazott\_torlese\_trigger

AFTER DELETE ON Alkalmazottak

FOR EACH ROW

BEGIN

DELETE FROM Rendelesek WHERE ALKALMAZOTT\_ID = :OLD.ALKALMAZOTT\_ID;

END;

/