Adatbázis rendszerek II.

Második beadandó feladat

Dobai Attila

DIZ4VX

Gyakorlatvezető: dr. Kovács László

Gyakorlat időpontja: Hétfő 10:00

Képzés: BSc nappali

A feladat leírása:

Egy mentőállomás nyilvántartása, ami magában foglalja a PL/SQL comagokat és a bemutatásához szükséges kódot. A program funkciói a következők:

- tábla feltöltés véletlen elemekkel

- tábla feltöltés állományból

- új hívás felvitele

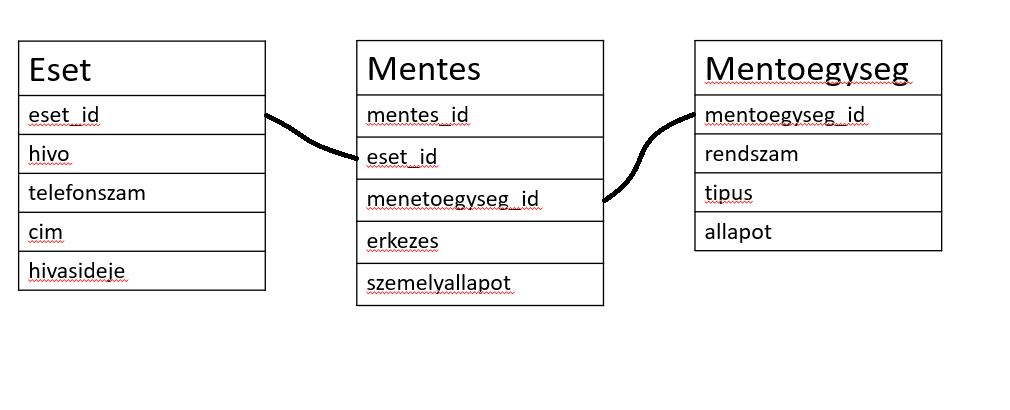
- hívás kezelés adminisztrálása

- kivonulási napló feltöltése

- függvény, mely egy paraméterként adott naphoz megadja a hívások számát

- eddig hívások lekérdezése személyre vagy időszakra, betegre szűrve

Relációs Modell:

Tábla: Eset

Mezők:

eset\_id (egész, elsődleges kulcs)

hivo (szöveg)

telefonszam (szöveg)

cim (szöveg)

hivasideje(date)

Tábla: Mentoegyseg

Mezők:

mentoegyseg\_id (egész, elsődleges kulcs)

rendszam (szöveg)

tipus (szöveg)

allapot (szöveg)

Tábla: Mentes

Mezők:

mentes\_id (egész, elsődleges kulcs)

eset\_id (egész, külső kulcs az eset táblához)

mentoegyseg\_id (egész, külső kulcs a Mentőegység táblához)

erkezes (dátum és idő)

szemelyallapot (szöveg)

Táblák létrehozása:

-- Eset tábla létrehozása

CREATE TABLE Eset (

eset\_id NUMBER PRIMARY KEY,

hivo VARCHAR2(255),

telefonszam VARCHAR2(255),

cim VARCHAR2(255),

hivasideje DATE

);

-- Mentoegyseg tábla létrehozása

CREATE TABLE Mentoegyseg (

mentoegyseg\_id NUMBER PRIMARY KEY,

rendszam VARCHAR2(255),

tipus VARCHAR2(255),

allapot VARCHAR2(255)

);

-- Mentes tábla létrehozása

CREATE TABLE Mentes (

mentes\_id NUMBER PRIMARY KEY,

eset\_id NUMBER,

mentoegyseg\_id NUMBER,

erkezes DATE,

szemelyallapot VARCHAR2(255),

FOREIGN KEY (eset\_id) REFERENCES Eset(eset\_id),

FOREIGN KEY (mentoegyseg\_id) REFERENCES Mentoegyseg(mentoegyseg\_id)

);

PL/SQL kód:

CREATE OR REPLACE PACKAGE TBL\_FELTOLTES\_PKG AS

-- Eset tábla feltöltése véletlen adatokkal

PROCEDURE feltolt\_eset;

-- Mentoegyseg tábla feltöltése véletlen adatokkal

PROCEDURE feltolt\_mentoegyseg;

-- Mentes tábla feltöltése véletlen adatokkal

PROCEDURE feltolt\_mentes;

END TBL\_FELTOLTES\_PKG;

/

CREATE OR REPLACE PACKAGE BODY TBL\_FELTOLTES\_PKG AS

-- Eset tábla feltöltése véletlen adatokkal

PROCEDURE feltolt\_eset IS

v\_hivo VARCHAR2(100);

v\_telefonszam VARCHAR2(100);

v\_cim VARCHAR2(100);

v\_hivasideje DATE;

BEGIN

FOR i IN 1..10 LOOP -- Feltöltés 10 véletlen rekorddal

v\_hivo := 'Hivo' || i;

v\_telefonszam := 'Telefonszam' || i;

v\_cim := 'Cim' || i;

v\_hivasideje := SYSDATE - ROUND(DBMS\_RANDOM.VALUE(1, 30)); -- Véletlen hivásidő az elmúlt 30 napban

INSERT INTO Eset (eset\_id, hivo, telefonszam, cim, hivasideje)

VALUES (i, v\_hivo, v\_telefonszam, v\_cim, v\_hivasideje);

END LOOP;

COMMIT;

END feltolt\_eset;

-- Mentoegyseg tábla feltöltése véletlen adatokkal

PROCEDURE feltolt\_mentoegyseg IS

v\_rendszam VARCHAR2(100);

v\_tipus VARCHAR2(100);

v\_allapot VARCHAR2(100);

BEGIN

FOR i IN 1..10 LOOP -- Feltöltés 10 véletlen rekorddal

v\_rendszam := 'Rendszam' || i;

v\_tipus := 'Tipus' || i;

v\_allapot := 'Allapot' || i;

INSERT INTO Mentoegyseg (mentoegyseg\_id, rendszam, tipus, allapot)

VALUES (i, v\_rendszam, v\_tipus, v\_allapot);

END LOOP;

COMMIT;

END feltolt\_mentoegyseg;

-- Mentes tábla feltöltése véletlen adatokkal

PROCEDURE feltolt\_mentes IS

v\_eset\_id NUMBER;

v\_mentoegyseg\_id NUMBER;

v\_erkezes DATE;

v\_szemelyallapot VARCHAR2(100);

BEGIN

FOR i IN 1..10 LOOP -- Feltöltés 10 véletlen rekorddal

v\_eset\_id := ROUND(DBMS\_RANDOM.VALUE(1, 10)); -- Véletlen eset\_id 1 és 10 között

v\_mentoegyseg\_id := ROUND(DBMS\_RANDOM.VALUE(1, 10)); -- Véletlen mentoegyseg\_id 1 és 10 között

v\_erkezes := SYSDATE - ROUND(DBMS\_RANDOM.VALUE(1, 30)); -- Véletlen érkezési idő az elmúlt 30 napban

v\_szemelyallapot := 'Szemelyallapot' || i;

INSERT INTO Mentes (mentes\_id, eset\_id, mentoegyseg\_id, erkezes, szemelyallapot)

VALUES (i, v\_eset\_id, v\_mentoegyseg\_id, v\_erkezes, v\_szemelyallapot);

END LOOP;

COMMIT;

END feltolt\_mentes;

-- Eset tábla feltöltése állományból

PROCEDURE all\_feltolt\_eset(p\_file\_path\_in IN VARCHAR2) IS

v\_hivo VARCHAR2(100);

v\_telefonszam VARCHAR2(100);

v\_cim VARCHAR2(100);

v\_hivasideje DATE;

v\_file UTL\_FILE.FILE\_TYPE;

v\_line VARCHAR2(1000);

BEGIN

v\_file := UTL\_FILE.FOPEN('DIRECTORY\_NAME', p\_file\_path\_in, 'R'); -- Az állomány elérési útvonala és a könyvtár neve

LOOP

BEGIN

UTL\_FILE.GET\_LINE(v\_file, v\_line);

-- Feldolgozás és adatok kinyerése a sorból

v\_hivo := SUBSTR(v\_line, 1, 100);

v\_telefonszam := SUBSTR(v\_line, 101, 100);

v\_cim := SUBSTR(v\_line, 201, 100);

v\_hivasideje := TO\_DATE(SUBSTR(v\_line, 301, 10), 'YYYY-MM-DD');

-- Adatok beszúrása az Eset táblába

INSERT INTO Eset (eset\_id, hivo, telefonszam, cim, hivasideje)

VALUES (Eset\_seq.NEXTVAL, v\_hivo, v\_telefonszam, v\_cim, v\_hivasideje);

EXCEPTION

WHEN UTL\_FILE.READ\_ERROR THEN

EXIT; -- Kilépés, ha nincs több sor az állományban

END;

END LOOP;

UTL\_FILE.FCLOSE(v\_file);

COMMIT;

END all\_feltolt\_eset;

-- Mentoegyseg tábla feltöltése állományból

PROCEDURE all\_feltolt\_mentoegyseg(p\_file\_path\_in IN VARCHAR2) IS

v\_rendszam VARCHAR2(100);

v\_tipus VARCHAR2(100);

v\_allapot VARCHAR2(100);

v\_file UTL\_FILE.FILE\_TYPE;

v\_line VARCHAR2(1000);

BEGIN

v\_file := UTL\_FILE.FOPEN('DIRECTORY\_NAME', p\_file\_path\_in, 'R'); -- Az állomány elérési útvonala és a könyvtár neve

LOOP

BEGIN

UTL\_FILE.GET\_LINE(v\_file, v\_line);

-- Feldolgozás és adatok kinyerése a sorból

v\_rendszam := SUBSTR(v\_line, 1, 100);

v\_tipus := SUBSTR(v\_line, 101, 100);

v\_allapot := SUBSTR(v\_line, 201, 100);

-- Adatok beszúrása a Mentoegyseg táblába

INSERT INTO Mentoegyseg (mentoegyseg\_id, rendszam, tipus, allapot)

VALUES (Mentoegyseg\_seq.NEXTVAL, v\_rendszam, v\_tipus, v\_allapot);

EXCEPTION

WHEN UTL\_FILE.READ\_ERROR THEN

EXIT; -- Kilépés, ha nincs több sor az állományban

END;

END LOOP;

UTL\_FILE.FCLOSE(v\_file);

COMMIT;

END all\_feltolt\_mentoegyseg;

-- Mentes tábla feltöltése állományból

PROCEDURE all\_feltolt\_mentes(p\_file\_path\_in IN VARCHAR2) IS

v\_eset\_id NUMBER;

v\_mentoegyseg\_id NUMBER;

v\_erkezes DATE;

v\_szemelyallapot VARCHAR2(100);

v\_file UTL\_FILE.FILE\_TYPE;

v\_line VARCHAR2(1000);

BEGIN

v\_file := UTL\_FILE.FOPEN('DIRECTORY\_NAME', p\_file\_path\_in, 'R'); -- Az állomány elérési útvonala és a könyvtár neve

LOOP

BEGIN

UTL\_FILE.GET\_LINE(v\_file, v\_line);

-- Feldolgozás és adatok kinyerése a sorból

v\_eset\_id := TO\_NUMBER(SUBSTR(v\_line, 1, 10));

v\_mentoegyseg\_id := TO\_NUMBER(SUBSTR(v\_line, 11, 10));

v\_erkezes := TO\_DATE(SUBSTR(v\_line, 21, 19), 'YYYY-MM-DD HH24:MI:SS');

v\_szemelyallapot := SUBSTR(v\_line, 41, 100);

-- Adatok beszúrása a Mentes táblába

INSERT INTO Mentes (mentes\_id, eset\_id, mentoegyseg\_id, erkezes, szemelyallapot)

VALUES (Mentes\_seq.NEXTVAL, v\_eset\_id, v\_mentoegyseg\_id, v\_erkezes, v\_szemelyallapot);

EXCEPTION

WHEN UTL\_FILE.READ\_ERROR THEN

EXIT; -- Kilépés, ha nincs több sor az állományban

END;

END LOOP;

UTL\_FILE.FCLOSE(v\_file);

COMMIT;

END all\_feltolt\_mentes;

-- Eset tábla manuális felvétele

PROCEDURE man\_felvetel\_eset(p\_hivo IN VARCHAR2, p\_telefonszam IN VARCHAR2, p\_cim IN VARCHAR2, p\_hivasideje IN DATE) IS

BEGIN

INSERT INTO Eset (eset\_id, hivo, telefonszam, cim, hivasideje)

VALUES (Eset\_seq.NEXTVAL, p\_hivo, p\_telefonszam, p\_cim, p\_hivasideje);

COMMIT;

END man\_felvetel\_eset;

-- Mentoegyseg tábla manuális felvétele

PROCEDURE man\_felvetel\_mentoegyseg(p\_rendszam IN VARCHAR2, p\_tipus IN VARCHAR2, p\_allapot IN VARCHAR2) IS

BEGIN

INSERT INTO Mentoegyseg (mentoegyseg\_id, rendszam, tipus, allapot)

VALUES (Mentoegyseg\_seq.NEXTVAL, p\_rendszam, p\_tipus, p\_allapot);

COMMIT;

END man\_felvetel\_mentoegyseg;

-- Mentes tábla manuális felvétele

PROCEDURE man\_felvetel\_mentes(p\_eset\_id IN NUMBER, p\_mentoegyseg\_id IN NUMBER, p\_erkezes IN DATE, p\_szemelyallapot IN VARCHAR2) IS

BEGIN

INSERT INTO Mentes (mentes\_id, eset\_id, mentoegyseg\_id, erkezes, szemelyallapot)

VALUES (Mentes\_seq.NEXTVAL, p\_eset\_id, p\_mentoegyseg\_id, p\_erkezes, p\_szemelyallapot);

COMMIT;

END man\_felvetel\_mentes;

-- Eset tábla adatok törlése

PROCEDURE torles\_eset(p\_eset\_id IN NUMBER) IS

BEGIN

DELETE FROM Eset WHERE eset\_id = p\_eset\_id;

COMMIT;

END torles\_eset;

-- Mentoegyseg tábla adatok törlése

PROCEDURE torles\_mentoegyseg(p\_mentoegyseg\_id IN NUMBER) IS

BEGIN

DELETE FROM Mentoegyseg WHERE mentoegyseg\_id = p\_mentoegyseg\_id;

COMMIT;

END torles\_mentoegyseg;

-- Mentes tábla adatok törlése

PROCEDURE torles\_mentes(p\_mentes\_id IN NUMBER) IS

BEGIN

DELETE FROM Mentes WHERE mentes\_id = p\_mentes\_id;

COMMIT;

END torles\_mentes;

-- Eset tábla adatok módosítása

PROCEDURE modositas\_eset(p\_eset\_id IN NUMBER, p\_hivo IN VARCHAR2, p\_telefonszam IN VARCHAR2, p\_cim IN VARCHAR2, p\_hivasideje IN DATE) IS

BEGIN

UPDATE Eset

SET hivo = p\_hivo,

telefonszam = p\_telefonszam,

cim = p\_cim,

hivasideje = p\_hivasideje

WHERE eset\_id = p\_eset\_id;

COMMIT;

END modositas\_eset;

-- Mentoegyseg tábla adatok módosítása

PROCEDURE modositas\_mentoegyseg(p\_mentoegyseg\_id IN NUMBER, p\_rendszam IN VARCHAR2, p\_tipus IN VARCHAR2, p\_allapot IN VARCHAR2) IS

BEGIN

UPDATE Mentoegyseg

SET rendszam = p\_rendszam,

tipus = p\_tipus,

allapot = p\_allapot

WHERE mentoegyseg\_id = p\_mentoegyseg\_id;

COMMIT;

END modositas\_mentoegyseg;

-- Mentes tábla adatok módosítása

PROCEDURE modositas\_mentes(p\_mentes\_id IN NUMBER, p\_eset\_id IN NUMBER, p\_mentoegyseg\_id IN NUMBER, p\_erkezes IN DATE, p\_szemelyallapot IN VARCHAR2) IS

BEGIN

UPDATE Mentes

SET eset\_id = p\_eset\_id,

mentoegyseg\_id = p\_mentoegyseg\_id,

erkezes = p\_erkezes,

szemelyallapot = p\_szemelyallapot

WHERE mentes\_id = p\_mentes\_id;

COMMIT;

END modositas\_mentes;

FUNCTION esetek\_szama(p\_datum IN DATE) RETURN NUMBER IS

v\_esetek\_szama NUMBER;

BEGIN

SELECT COUNT(\*) INTO v\_esetek\_szama

FROM Eset

WHERE TRUNC(hivasideje) = TRUNC(p\_datum);

RETURN v\_esetek\_szama;

END esetek\_szama;

FUNCTION esetek\_szama\_szures(p\_hivo IN VARCHAR2 DEFAULT NULL, p\_start\_date IN DATE DEFAULT NULL, p\_end\_date IN DATE DEFAULT NULL) RETURN NUMBER IS

v\_esetek\_szama NUMBER;

BEGIN

SELECT COUNT(\*) INTO v\_esetek\_szama

FROM Eset

WHERE (p\_hivo IS NULL OR hivo = p\_hivo)

AND (p\_start\_date IS NULL OR hivasideje >= p\_start\_date)

AND (p\_end\_date IS NULL OR hivasideje <= p\_end\_date);

RETURN v\_esetek\_szama;

END esetek\_szama\_szures;

END TBL\_FELTOLTES\_PKG;

/