

Adatbázis Rendszerek II.

PL/SQL beadandó feladat

Firtkó Bence
TPOM8Y

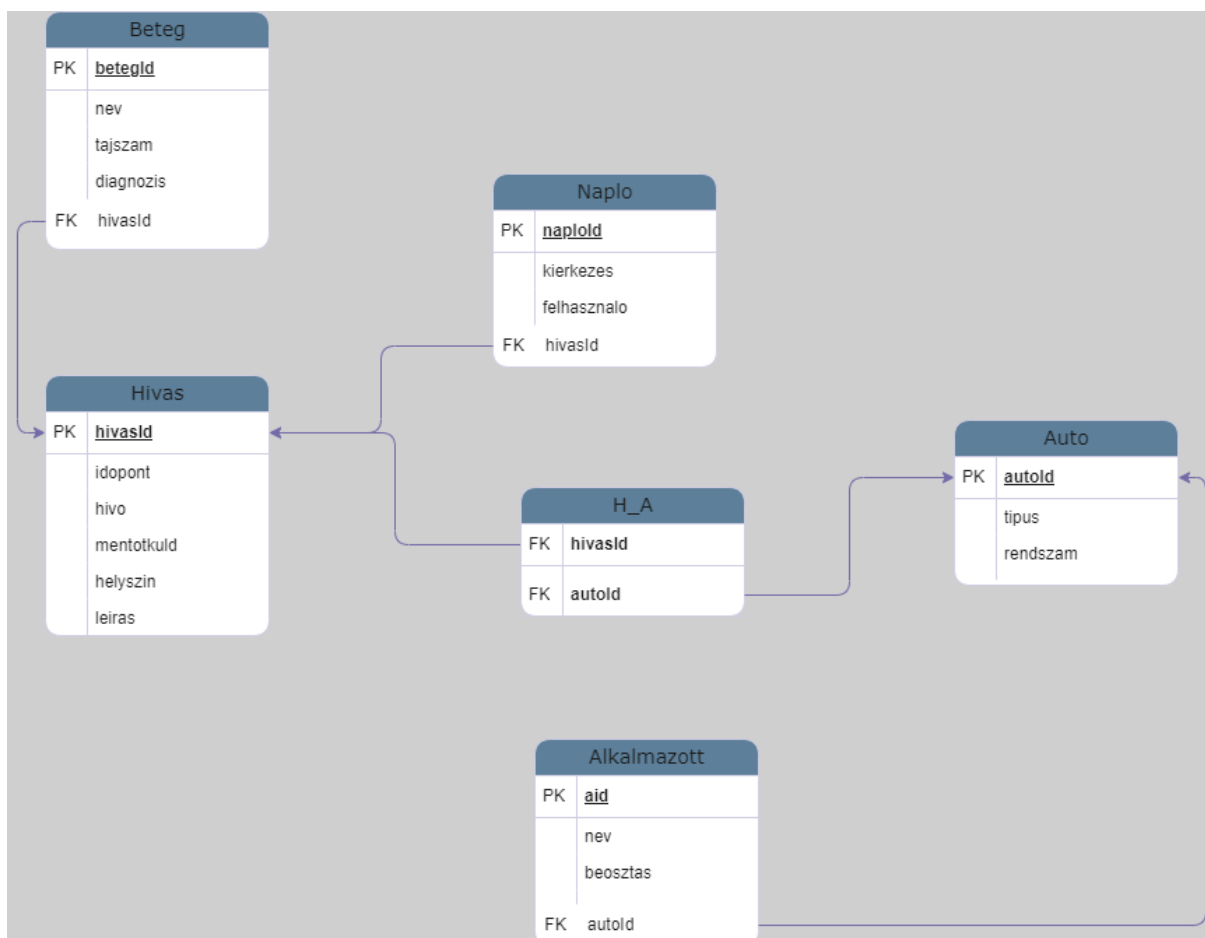
Gyakorlatvezető: dr. Kovács László
Gyakorlat időpontja: Hétfő 10:00
Képzés: BSc nappali

Feladat leírás: A feladat egy nyilvántartás készítése volt egy mentőszolgálat részére. Ebbe beletartozik az adatbázis megtervezése, a relációs modell elkészítése és tábla generáló parancsok létrehozása is. A beadandó legfontosabb része egy olyan csomag létrehozása volt, amely a következőket valósítja meg:

- tábla feltöltés véletlen elemekkel
- tábla feltöltés állományból
- új hívás felvitele
- hívás kezelés adminisztrálása
- a hívások naplózása
- függvény létrehozás, amely egy megadott naphoz megadja a hívások számát
- hívások lekérdezése hívó, időszakra és betegre szűrve

Ezen csomag használata egy JDBC alkalmazás keretein belül.

Relációs modell:



A táblák generáló SQL parancsai:

```
CREATE TABLE HIVAS(hivasId INT PRIMARY KEY, idopont TIMESTAMP, hivo varchar2(15), mentotkuld int, helyszin varchar2(30), leiras varchar2(50));
```

```
CREATE TABLE NAPLO(naploId INT PRIMARY KEY, kierkezes TIMESTAMP, felhasznalo varchar2(20), hivasId int, FOREIGN KEY (hivasId) REFERENCES HIVAS(hivasId));
```

```
CREATE TABLE BETEG(betegId INT PRIMARY KEY,nev varchar2(15), tajszam varchar2(15), diagnozis varchar2(20), hivasId int, FOREIGN KEY (hivasId) REFERENCES HIVAS(hivasId));
```

```
CREATE TABLE AUTO(autold INT PRIMARY KEY, tipus varchar2(15), rendszam varchar2(8));
```

```
CREATE TABLE ALKALMAZOTT(alId INT PRIMARY KEY, nev varchar2(15), beosztas varchar2(10), autold int, FOREIGN KEY (autold) REFERENCES AUTO(autold));
```

```
CREATE TABLE H_A(hivasId int, FOREIGN KEY (hivasId) REFERENCES HIVAS(hivasId), autold int, FOREIGN KEY (autold) REFERENCES AUTO(autold));
```

A triggerek generáló kódjai:

```
CREATE OR REPLACE TRIGGER feltoltnaplo AFTER insert ON HIVAS
```

```
DECLARE
```

```
sor hivas%ROWTYPE;
```

```
minutes int;
```

```
BEGIN
```

```
SELECT * INTO sor FROM HIVAS WHERE hivasId = (SELECT * FROM (SELECT hivasId FROM HIVAS ORDER BY idopont desc) WHERE ROWNUM <= 1);
```

```
minutes := MOD(ABS(DBMS_RANDOM.RANDOM()), 20) + 5;
```

```
IF (sor.mentotkuld >= 1) then
```

```
INSERT INTO NAPLO VALUES (naplo_sqe.nextval, sysdate + (1/1440)*minutes, (SELECT USER FROM DUAL), sor.hivasId);
```

```
END IF;
```

```
END;
```

```

CREATE OR REPLACE TRIGGER feltolt_h_a AFTER insert ON HIVAS

DECLARE

sor hivas%ROWTYPE;

autoszam int;

BEGIN

SELECT * INTO sor FROM HIVAS WHERE hivasId = (SELECT * FROM (SELECT hivasId FROM HIVAS ORDER BY
idopont desc) WHERE ROWNUM <= 1);

autoszam := MOD(ABS(DBMS_RANDOM.RANDOM()), 3) + 1;

IF sor.mentotkuld >= 1 then

INSERT INTO H_A VALUES (sor.hivasId, autoszam);

END IF;

END;

```

A három használt szekvencia kódja:

```
CREATE SEQUENCE beteg_sqe INCREMENT BY 1 START WITH 1;
```

```
CREATE SEQUENCE naplo_sqe INCREMENT BY 1 START WITH 1;
```

```
CREATE SEQUENCE hivas_sqe INCREMENT BY 1 START WITH 1;
```

A csomag generáló kódja:

```

CREATE OR REPLACE PACKAGE mentoszolgalat IS

TYPE hivas_cursor IS REF CURSOR;

PROCEDURE FELTOLTLALK;

PROCEDURE FELTOLTAUTO;

PROCEDURE UJHIVAS(p_hivo varchar2, p_helyszin varchar2, p_leiras varchar2, p_mentotkuld int);

PROCEDURE BETEG_FELTOLT;

PROCEDURE hivasok_datum(p_idopont1 varchar2, p_idopont2 varchar2, p_result OUT hivas_cursor);

PROCEDURE hivasok(p_hivo varchar2, p_result out hivas_cursor);

PROCEDURE hivasok_beteg(p_beteg varchar2,p_result OUT hivas_cursor);

PROCEDURE DELETETABLAk;

FUNCTION RAND_B return VARCHAR2;

```

```
FUNCTION RAND_TIP return VARCHAR2;

FUNCTION hivasokszama(p_datum varchar2) RETURN NUMBER;

end;
```

CREATE OR REPLACE PACKAGE BODY mentoszolgalat IS

```
FUNCTION RAND_B return varchar2 IS

TYPE B IS VARRAY(4) OF VARCHAR2(15);

BEOSZTASOK B;

RAND INT;

BEGIN

BEOSZTASOK:= B('ORVOS','APOLÓ','SOFŐR','MENTŐS');

SELECT MOD(ABS(DBMS_RANDOM.RANDOM()), 4) + 1 INTO RAND FROM DUAL;

RETURN( BEOSZTASOK(RAND) );

EXCEPTION

WHEN OTHERS THEN

DBMS_OUTPUT.PUT_LINE('Nem sikerült beosztást választani!' );

END;
```

```
FUNCTION RAND_TIP return VARCHAR2 is

TYPE T IS VARRAY(3) OF VARCHAR2(20);

TIPUSOK T;

RAND INT;

BEGIN

TIPUSOK:= T('MERCEDES','VOLKSWAGEN','SKODA');

SELECT MOD(ABS(DBMS_RANDOM.RANDOM()), 3) + 1 INTO RAND FROM DUAL;

RETURN( TIPUSOK(RAND) );

EXCEPTION

WHEN OTHERS THEN

DBMS_OUTPUT.PUT_LINE('Nem sikerült típust választani!' );

END;
```

PROCEDURE FELTOLTAUTO is

```
RAND_ST VARCHAR2(3);

ONE int;

TWO int;

THREE int;

BEGIN

DELETE FROM ALKALMAZOTT;

DELETE FROM AUTO;

FOR I IN 1..5 LOOP

    RAND_ST := DBMS_RANDOM.string('u',3);

    ONE := MOD(ABS(DBMS_RANDOM.RANDOM()), 10);

    TWO:= MOD(ABS(DBMS_RANDOM.RANDOM()), 10);

    THREE := MOD(ABS(DBMS_RANDOM.RANDOM()), 10);

    INSERT INTO AUTO VALUES (I, RAND_TIP , RAND_ST || TO_CHAR(ONE) || TO_CHAR(TWO) ||
TO_CHAR(THREE));

END LOOP;

END;
```

PROCEDURE FELTOLTLALK is

```
BEGIN

DELETE FROM ALKALMAZOTT;

FOR I IN 1..10 LOOP

    INSERT INTO ALKALMAZOTT VALUES (I,'ALKALMAZOTT_' || TO_CHAR(I) , RAND_B,
MOD(ABS(DBMS_RANDOM.RANDOM()), 5) + 1);

END LOOP;

END;
```

```

PROCEDURE ujhivas(p_hivo varchar2, p_helyszin varchar2, p_leiras varchar2, p_mentotkuld int) is
begin
    insert into hivas values(hivas_sqe.nextval, TO_TIMESTAMP(TO_CHAR(SYSDATE, 'MM-DD-YYYY
HH24:MI:SS'),'MM-DD-YYYY HH24:MI:SS','NLS_DATE_LANGUAGE=AMERICAN'), p_hivo, p_helyszin, p_leiras,
        p_mentotkuld);
exception
when others then
    dbms_output.put_line('Valami hiba történt!');
end;

```

```

procedure beteg_feltolt as
F UTL_FILE.FILE_TYPE;
sor varchar2(200);
c1 varchar(30);
c2 varchar(30);
c3 varchar(30);
c4 varchar(30);
begin
F := UTL_FILE.FOPEN('DOP','betegek.txt','R');
loop
    UTL_FILE.GET_LINE(F,sor,100);

    c1 := SUBSTR(sor, 1, INSTR(sor, ',') - 1);
    sor := SUBSTR(sor, INSTR(sor, ',') + 1);

    c2 := SUBSTR(sor, 1, INSTR(sor, ',') - 1);
    sor := SUBSTR(sor, INSTR(sor, ',') + 1);

    c3 := SUBSTR(sor, 1, INSTR(sor, ',') - 1);
    c4 := SUBSTR(sor, INSTR(sor, ',') + 1);

    INSERT INTO beteg VALUES(beteg_sqe.nextval,c1,c2,c3,c4);
end loop;

```

```
UTL_FILE.FCLOSE(F);
```

```
exception
```

```
when others then
```

```
    DBMS_OUTPUT.put_line('Nem sikerült a feltöltés!');
```

```
end;
```

```
PROCEDURE hivasok(p_hivo VARCHAR2, p_result OUT hivas_cursor) IS
```

```
BEGIN
```

```
    OPEN p_result FOR
```

```
        SELECT *
```

```
        FROM hivas
```

```
        WHERE hivo = p_hivo;
```

```
END;
```

```
PROCEDURE hivasok_datum(p_idopont1 varchar2, p_idopont2 varchar2, p_result OUT hivas_cursor) is
```

```
BEGIN
```

```
    OPEN p_result FOR
```

```
        SELECT * FROM hivas WHERE idopont > TO_TIMESTAMP(p_idopont1,'YYYY/MM/DD') AND idopont <  
TO_TIMESTAMP(p_idopont2, 'YYYY/MM/DD');
```

```
END;
```

```
PROCEDURE hivasok_beteg(p_beteg varchar2,p_result OUT hivas_cursor) IS
```

```
BEGIN
```

```
    OPEN p_result FOR
```

```
        SELECT h.hivasId, h.idopont, b.nev, b.tajszam, b.diagnozis
```

```
        FROM hivas h
```

```
        JOIN beteg b ON h.hivasId = b.hivasId
```

```
        WHERE b.nev = p_beteg;
```

```
END;
```



```
FUNCTION hivasokszama(p_datum varchar2) RETURN NUMBER is
    hivasok NUMBER;
BEGIN
    SELECT COUNT(*) INTO hivasok FROM HIVAS WHERE idopont > TO_TIMESTAMP(p_datum,'YYYY/MM/DD')
    AND idopont < (TO_TIMESTAMP(p_datum, 'YYYY/MM/DD') + interval '1' day);
    RETURN hivasok;
END;
```

```
PROCEDURE deletetablak is
BEGIN
    DELETE FROM ALKALMAZOTT;
    DELETE FROM H_A;
    DELETE FROM AUTO;
    DELETE FROM BETEG;
    DELETE FROM NAPLO;
    DELETE FROM HIVAS;
END;
END;
```

A beadandóhoz tartozik még egy JDBC alkalmazás, amely a következő GitHub linken érhető el:

https://github.com/Fbencee/PLSQL_beadando