

# JEGYZŐKÖNYV

Adatbázisrendszerek II.

II. Féléves feladat

Étterem adatbázis PL/SQL

Készítette: **Fekete Máté**

Neptunkód: **JR9KY7**

Gyakorlatvezető: **Polonkai Dávid**

## Feladat ismertetése:

Egy PL/SQL programot kellett létrehozni, ahol tárolt eljárások segítségével tudunk adatokat felvinni, módosítani vagy törölni az adatbázisból, valamint tárolt függvényekkel lekérdezéseket végrehajtani cursor használattal is. Ezek mellett szükség volt egy tárolt csomagra ahová több procedure és function került. Triggerek szempontjából készítettem egy naplózó trigger a 2 táblára egyet-egyet, amik a táblákon végrehajtott insert, update és delete parancsokat naplózza egy külön táblába, aminek DATA\_LOG a neve és van egy külön trigger is az egyik táblához, ami automatikusan kulcsot generál, illetve van még egy, ami azt kontrollálja, hogy a pincér táblában a pincérek fizetését ne lehessen 20%-nál többel emelni.

DATA\_LOG tábla létrehozása:

```
CREATE TABLE DATA_LOG (DATUM TIMESTAMP, MUVELET VARCHAR2(25), TABLANEV VARCHAR2(25));
```

## PL/SQL:

### Tárolt eljárások adatok felvételére a táblákba:

#### Addpincer eljárás

```
--Pincer tabla rekord felvitele
CREATE OR REPLACE PROCEDURE ADDPINCER (PINCID INT, NEV VARCHAR2, FIZETES INT, KOR INT, KEZDES VARCHAR2) IS
EXSISTERROR EXCEPTION;
rows_found number;

BEGIN
    SELECT COUNT(*) INTO ROWS_FOUND FROM PINCER WHERE PINCERID = PINCID;

    IF ROWS_FOUND != 0 THEN RAISE EXSISTERROR;
    ELSE
        INSERT INTO PINCER VALUES(PINCID, NEV, FIZETES, KOR, KEZDES);
    END IF;
EXCEPTION
    WHEN exsisterror THEN
        dbms_output.put_line('Ez az azonosito mar letezik!');
END ADDPINCER;
```

Meg kell adnunk: egy pincér ID-t, nevet, fizetést, kort és egy kezdést (pl.: 2020-11-17).

Ezt követően ellenőrzi az eljárás, hogy van-e már ilyen pincér ID, ha nincs akkor felviszi az adatot, ha létezik ilyen azonosító akkor nem és hibaüzenetet ad a felhasználónak.

Meghívás:

```
EXECUTE ADDPINCER(4, 'Alma Karcsi', 299000, 21, '2022-11-11');
```

## Addasztal eljárás

```
--Asztal tabla rekord felvitele
CREATE OR REPLACE PROCEDURE ADDASZTAL(ASZTALSZ IN asztal.asztalszam%TYPE, FEROHELY IN asztal.ferohe
EXSISTERROR EXCEPTION;
rows_found number;

BEGIN
    SELECT COUNT(*) INTO ROWS_FOUND FROM ASZTAL WHERE ASZTALSZ = ASZTAL.ASZTALSZAM;

    IF ROWS_FOUND != 0 THEN RAISE EXSISTERROR;
    ELSE
        INSERT INTO ASZTAL VALUES(ASZTALSZ, FEROHELY, EMELET, FOGLALT, PINCERAZ);
    END IF;
EXCEPTION
    WHEN exsisterror THEN
        dbms_output.put_line('Ez az azonosito mar letezik!');
END ADDASZTAL;
```

Meg kell adnunk: egy asztal ID-t, férőhelyet, emeletet, foglalt-e (0 vagy 1), illetve egy létező pincér ID-t.

Ezt követően ellenőrzi az eljárás, hogy van-e már ilyen asztal ID, ha nincs akkor felviszi az adatot, ha létezik ilyen azonosító akkor nem és hibaüzenetet ad a felhasználónak.

Meghívás:

```
EXECUTE ADDASZTAL(12, 6, 1, 0, 4);
```

## Tárolt eljárások adatok módosítására a táblákban:

### Modositpincer eljárás

```
--Pincer tabla modositasa
CREATE OR REPLACE PROCEDURE MODOSITPINCER (ID INT, NAME VARCHAR2) IS
    notexsisterror exception;
    nameequalerror exception;
    rows_found number;
    current_name char(30);

BEGIN
    SELECT COUNT(*) INTO ROWS_FOUND FROM PINCER WHERE PINCERID = ID;

    IF ROWS_FOUND !=0 THEN
        SELECT NEV INTO CURRENT_NAME FROM PINCER WHERE PINCERID = ID;
    END IF;

    IF ROWS_FOUND = 0 THEN RAISE NOTEXSISTERROR;
    ELSIF
        NAME = CURRENT_NAME THEN RAISE NAMEEQUALERROR;
    ELSE
        UPDATE PINCER SET NEV = NAME WHERE PINCERID = ID;
    END IF;
EXCEPTION
    WHEN NOTEXSISTERROR THEN
        DBMS_OUTPUT.PUT_LINE('Ilyen azonositoval nincs pincer!');
    WHEN NAMEEQUALERROR THEN
        DBMS_OUTPUT.PUT_LINE('Ugyanaz a nev van megadva!');
END MODOSITPINCER;
```

Meg kell adnunk: a módosítani kívánt pincér azonosítóját és hogy mire szeretnénk cserélni a nevét.

Ezt követően ellenőrzi az eljárás, hogy van e már ilyen pincér ID, ha van akkor az ahhoz tartozó nevet módosítja a megadott értékkel, ha nincs ilyen ID, akkor hibaüzenetet dob a felhasználónak.

Meghívás:

```
EXECUTE MODOSITPINCER(4, 'Alma Karoly');
```

### Modositasztal eljárás

```
--Asztal tabla rekord modositasa
CREATE OR REPLACE PROCEDURE MODOSITASZTAL(ID INT ,RESERVED INT) IS
NOTEXSISTERROR EXCEPTION;
RESERVEDISNOTVALID EXCEPTION;
rows_found number;

BEGIN

    SELECT COUNT(*) INTO ROWS_FOUND FROM ASZTAL WHERE ASZTALSZAM = ID;
    IF ROWS_FOUND = 0 THEN RAISE NOTEXSISTERROR;
    ELSIF
        RESERVED != 1 AND RESERVED != 0 THEN RAISE RESERVEDISNOTVALID;
    ELSE
        UPDATE ASZTAL SET FOGLALT = RESERVED WHERE ASZTALSZAM = ID;
    END IF;
EXCEPTION
    WHEN NOTEXSISTERROR THEN
        DBMS_OUTPUT.PUT_LINE('Ilyen azonosítóval nincs asztal!');
    WHEN RESERVEDISNOTVALID THEN
        DBMS_OUTPUT.PUT_LINE('A megadott adat nem megfelelo');
END MODOSITASZTAL;
```

Meg kell adnunk: a módosítani kívánt asztalhoz tartozó ID-t és azt, hogy foglalt e (0 vagy 1).

Ezt követően ellenőrzi az eljárás, hogy létezik e ilyen ID az asztal táblában és hogy érvényes e foglalt paraméter, azaz vagy 1-et vagy 0-t adtunk e meg. Ha ezek teljesülnek, akkor módosítja a megadott adatot, ha valamelyik nem teljesül dob egy megfelelő kivételt, hibaüzenetet a felhasználónak.

Meghívás:

```
EXECUTE MODOSITASZTAL(12, 1);
```

## Tárolt eljárások rekordok törlésére a táblákból:

### Deletepincer eljárás:

```
--Pincer tabla rekordok torlese
CREATE OR REPLACE PROCEDURE DELETEPINCER(ID INT) IS
NOTEXSISTERRR EXCEPTION;
rows_found number;

BEGIN
    SELECT COUNT(*) INTO ROWS_FOUND FROM PINCER WHERE PINCERID = ID;

    IF ROWS_FOUND = 0 THEN RAISE NOTEXSISTERRR;
    ELSE
        DELETE FROM PINCER WHERE PINCERID = ID;
    END IF;
    EXCEPTION
        WHEN NOTEXSISTERRR THEN
            DBMS_OUTPUT.PUT_LINE('Ilyen azonosítóval nincs pincer!');
END DELETEPINCER;
```

Meg kell adnunk: a törölni kívánt pincer azonosítóját. Fontos: csak akkor tudunk pincer rekordot törölni, ha nem szolgál fel egy asztalnál sem, ha épp igen, akkor ki kell törölni vagy módosítani az adott asztalt.

Az eljárás ellenőzi, hogy létezik e ilyen pincer ID, ha igen akkor kitörli a táblából a hozzá tartozó rekordot, ha nincs ilyen ID hibaüzenetet dob.

### Meghívás:

```
EXECUTE DELETEPINCER(4);
```

### Deleteasztal eljárás

```
--Asztal tabla rekordok torlese
CREATE OR REPLACE PROCEDURE DELETEASZTAL(ID INT) IS
NOTEXSISTERRR EXCEPTION;
rows_found number;

BEGIN
    SELECT COUNT(*) INTO ROWS_FOUND FROM ASZTAL WHERE ASZTALSZAM = ID;

    IF ROWS_FOUND = 0 THEN RAISE NOTEXSISTERRR;
    ELSE
        DELETE FROM ASZTAL WHERE ASZTALSZAM = ID;
    END IF;
    EXCEPTION
        WHEN NOTEXSISTERRR THEN
            DBMS_OUTPUT.PUT_LINE('Ilyen azonosítóval nincs asztal!');
END DELETEASZTAL;
```

Meg kell adnunk: a törölni kívánt asztal azonosítóját.

Az eljárás ellenőzi, hogy létezik e ilyen asztal ID, ha igen akkor kitörli a táblából a hozzá tartozó rekordot, ha nincs ilyen ID hibaüzenetet dob.

Meghívás:

```
EXECUTE DELETEASZTAL(12);
```

## A Package (PKG\_ASZTAL)

```
CREATE OR REPLACE PACKAGE PKG_ASZTAL IS
    PROCEDURE ADDASZTAL(ASZTALSZ IN asztal.asztalszam%TYPE, FEROHELY IN asztal.ferohely%TYPE, EMELET IN asztal.emelet%TYPE,
    PROCEDURE MODOSITASZTAL(ID INT ,RESERVED INT);
    PROCEDURE DELETEASZTAL(ID INT);
    FUNCTION GET_AVG_FEROHELY RETURN NUMBER;
END PKG_ASZTAL;
```

Itt az előző részben látott eljárásokat hozom létre és kiegészítem egy függvénnyel is.

--Package body kezdete

```
CREATE OR REPLACE PACKAGE BODY PKG_ASZTAL IS
```

--Asztal tabla rekord felvittele

```
PROCEDURE ADDASZTAL(ASZTALSZ IN asztal.asztalszam%TYPE, FEROHELY IN asztal.ferohely%TYPE, EMELET IN asztal.emelet%TYPE,
IS EXSISTERROR EXCEPTION;
rows_found number;
```

BEGIN

```
SELECT COUNT(*) INTO ROWS_FOUND FROM ASZTAL WHERE ASZTALSZ = ASZTAL.ASZTALSZAM;
```

```
IF ROWS_FOUND != 0 THEN RAISE EXSISTERROR;
```

```
ELSE
```

```
INSERT INTO ASZTAL VALUES(ASZTALSZ, FEROHELY, EMELET, FOGLALT, PINCERAZ);
```

```
END IF;
```

EXCEPTION

```
WHEN exsisterror THEN
```

```
dbms_output.put_line('Ez az azonosito mar letezik!');
```

END;

--Asztal tabla rekord modositasa

```
PROCEDURE MODOSITASZTAL(ID INT ,RESERVED INT) IS
```

```
NOTEXSISTERROR EXCEPTION;
```

```
RESERVEDISNOTVALID EXCEPTION;
```

```
rows_found number;
```

BEGIN

```
SELECT COUNT(*) INTO ROWS_FOUND FROM ASZTAL WHERE ASZTALSZAM = ID;
```

```
IF ROWS_FOUND = 0 THEN RAISE NOTEXSISTERROR;
```

```
ELSIF
```

```
RESERVED != 1 AND RESERVED != 0 THEN RAISE RESERVEDISNOTVALID;
```

```
ELSE
```

```
UPDATE ASZTAL SET FOGLALT = RESERVED WHERE ASZTALSZAM = ID;
```

```
END IF;
```

EXCEPTION

```
WHEN NOTEXSISTERROR THEN
```

```
DBMS_OUTPUT.PUT_LINE('Ilyen azonositoval nincs asztal!');
```

```
WHEN RESERVEDISNOTVALID THEN
```

```
DBMS_OUTPUT.PUT_LINE('A megadott adat nem megfelelo');
```

END;

```

--Asztal tabla rekordok torlese
PROCEDURE DELETEASZTAL(ID INT) IS
NOTEXSISTERROR EXCEPTION;
rows_found number;

BEGIN
    SELECT COUNT(*) INTO ROWS_FOUND FROM ASZTAL WHERE ASZTALSZAM = ID;

    IF ROWS_FOUND = 0 THEN RAISE NOTEXSISTERROR;
    ELSE
        DELETE FROM ASZTAL WHERE ASZTALSZAM = ID;
    END IF;
    EXCEPTION
        WHEN NOTEXSISTERROR THEN
            DBMS_OUTPUT.PUT_LINE('Ilyen azonosítóval nincs asztal!');
END;

--Asztal tabla ferohely mezoinak atlaga, mennyi az atlag feohely
FUNCTION GET_AVG_FEROHELY RETURN NUMBER IS AVG_FEROHELY NUMBER;

BEGIN
    SELECT AVG(FEROHELY) INTO AVG_FEROHELY FROM ASZTAL;
    RETURN AVG_FEROHELY;
END;

END; --end PACKAGE BODY

```

A get\_avg\_ferohely függvény visszaadja, hogy a táblában szereplő asztalok közül mennyi az átlagos férőhely.

Meghívások a package-ből:

```

BEGIN
    PKG_ASZTAL.addasztal(14, 4, 2, 0, 2);
    PKG_ASZTAL.modositasztal(14, 1);
    PKG_ASZTAL.deleteasztal(14);
END;

SELECT PKG_ASZTAL.get_avg_ferohely() FROM dual;

```

Sequence létrehozása, ami automatikusan ID-t generál a pincér táblához:

```

CREATE SEQUENCE PINCER_SEQ;

CREATE OR REPLACE TRIGGER PINCER_INSERT BEFORE INSERT ON PINCER FOR EACH ROW

BEGIN
    SELECT PINCER_SEQ.nextval INTO :new.PINCERID FROM dual;
END;

```

## Többi trigger

### Pincer trigger

```
CREATE OR REPLACE TRIGGER Pincer_TRIGGER
BEFORE INSERT OR DELETE OR UPDATE ON Pincer FOR EACH ROW
BEGIN
    IF INSERTING THEN
        INSERT INTO DATA_LOG VALUES (CURRENT_TIMESTAMP, 'BESZURAS', 'Pincer');
    END IF;
    IF DELETING THEN
        INSERT INTO DATA_LOG VALUES (CURRENT_TIMESTAMP, 'TORLES', 'Pincer');
    END IF;
    IF UPDATING THEN
        INSERT INTO DATA_LOG VALUES (CURRENT_TIMESTAMP, 'MODOSITAS', 'Pincer');
    END IF;
END;
```

### Asztal trigger

```
CREATE OR REPLACE TRIGGER ASZTAL_TRIGGER
BEFORE INSERT OR DELETE OR UPDATE ON ASZTAL FOR EACH ROW
BEGIN
    IF INSERTING THEN
        INSERT INTO DATA_LOG VALUES (CURRENT_TIMESTAMP, 'BESZURAS', 'ASZTAL');
    END IF;
    IF DELETING THEN
        INSERT INTO DATA_LOG VALUES (CURRENT_TIMESTAMP, 'TORLES', 'ASZTAL');
    END IF;
    IF UPDATING THEN
        INSERT INTO DATA_LOG VALUES (CURRENT_TIMESTAMP, 'MODOSITAS', 'ASZTAL');
    END IF;
END;
```

A fenti két triggernél azt láthatjuk, hogy egy trigger többféle parancs hatására is aktiválódik (insert, update, delete) és egy napló táblába rögzíti a műveleteket.

### Check trigger

```
CREATE OR REPLACE TRIGGER CHECK_TRIGGER BEFORE UPDATE ON Pincer FOR EACH ROW
BEGIN
    IF NOT ((:NEW.FIZETES / (:OLD.FIZETES / 100)) >= 80 AND (:NEW.FIZETES / (:OLD.FIZETES / 100)) <= 120) THEN
        IF :OLD.FIZETES < :NEW.FIZETES THEN
            :NEW.FIZETES := :OLD.FIZETES * 1.2;
        END IF;
        IF :NEW.FIZETES < :OLD.FIZETES THEN
            :NEW.FIZETES := :OLD.FIZETES * 0.8;
        END IF;
        DBMS_OUTPUT.PUT_LINE('A pincerhez tartozo fizetesemeles meghaladja a 20%-ot, ezert korlatozva lett!');
    END IF;
END;
```

Ez a trigger nem engedi, hogy 20%-nál többel módosítsuk a pincék fizetését.



## Explicit kurzor használata

```
DECLARE
  P_ID Pincer.PincerID%TYPE;
  P_NEV Pincer.NEV%TYPE;
  P_FIZ Pincer.FIZETES%TYPE;
  P_KOR Pincer.KOR%TYPE;
  P_KEZD Pincer.KEZDES%TYPE;
  CURSOR LEKERD IS SELECT PincerID, NEV, FIZETES, KOR, KEZDES FROM Pincer;
BEGIN
  OPEN LEKERD;
  LOOP
    FETCH LEKERD INTO P_ID, P_NEV, P_FIZ, P_KOR, P_KEZD;
    EXIT WHEN LEKERD%NOTFOUND;
    DBMS_OUTPUT.PUT_LINE(P_ID || ' ' || P_NEV || ' ' || P_FIZ || ' ' || P_KOR || ' ' || P_KEZD);
  END LOOP;
  CLOSE LEKERD;
END;
```

## Eredmény

```
1 Szabo Laszlo 320000 36 2006-02-12
2 Kovacs Istvan 235000 27 2010-05-19
3 Lukacs Balazs 260000 21 2014-10-05
```

PL/SQL procedure successfully completed.

## Relációs séma

