Adatbázisok

9. gyakorlat:

PL/SQL: tárolt eljárások, függvények, triggerek

TÁROLT ELJÁRÁSOK, FÜGGVÉNYEK

Tárolt eljárások, függvények

- Már ismert koncepció: "metódus"
  - eljárás: utasítások összessége, névvel ellátva, nincs visszatérési érték, **PROCEDURE** kulcsszóval jelezzük

    (hasonlóan a C#/Java **void**-dal jelzett visszatérési értékű függvényeihez)

  - függvény: utasítások összessége, névvel ellátva, van jelzett típusos visszatérési érték, **FUNCTION** kulcsszóval jelezzük

    (hasonlóan a C#/Java konkrét típussal jelzett visszatérési értékű függvényeihez)

Tárolt eljárások, függvények (2.)

- Segítségükkel a feladatok megoldásához szükséges, de az alaprendszerben eddig még nem szereplő eljárások, illetve függvények elkészíthetők, a feladat elvárásaihoz igazítva,
- A tárolt függvények és eljárások az adatbázis tartalmával együtt kimenthetők, illetve betölthetők (újrafelhasználhatók),

PROCEDURE

```
CREATE [OR REPLACE] PROCEDURE
  eljárásnév (paraméterlista)
IS
  lokális változók
BEGIN
  utasítások
[EXCEPTION …]
END;
```

PROCEDURE

- Futtatása PL/SQL blokkon belül: szokásos, a nevével kell hivatkozni rá,
- Futtatása PL/SQL blokkon kívülről:

  EXECUTE eljárásnév;
- Törlés:
  DROP PROCEDURE eljárásnév;
- <OR REPLACE> szerepe: ha már létezik, akkor lecseréli

Gyakorlás: Futtatás PL/SQL blokkon kívülről

- Készíts eljárást, amely 2 szám paraméterrel rendelkezik. Az eljárás írja ki a két szám összegét. Ezt követően töröld az eljárást.

Feladat

- Készíts egy eljárást, amely paraméterként egy dolgozó vezetéknevét kapja. Ha van ilyen dolgozó, akkor annak a fizetését megemeli 10%-kal.

Feladat

- Készíts eljárást, amely egy szöveges paraméterrel rendelkezik, amely legyen egy részlegnév. Az eljárás írja ki az adott részleg összfizetését, illetve ha nincs ilyen nevű részleg, akkor pedig a 'Nem létező részleg!' szöveget.

Feladat

- Készíts tárolt eljárást, amely megemeli a dolgozók jutalékát, ha egy megadott éve már dolgoznak a cégnél. Az eljárás két paraméterrel rendelkezik. Az egyik paraméter a cégnél minimálisan eltöltendő évek számát adja meg, a másik az emelés mértékét.

Feladat

- Készíts egy másolatot az employees tábláról employees 2 néven!
- Készíts egy tárolt eljárást amelynek egy paramétere van: depid, ami szám. Az eljárás töröljön mindenkit az employees2 táblából, akinél a department_id egyenlő a depid-val.

Feladat

- Készíts eljárást, amely paraméterként kap egy Manager beosztású alkalmazott ID -ját. Töröljük a manager összes beosztottját, majd őt is az employees táblából!

FUNCTION

```
CREATE [OR REPLACE] FUNCTION
      függvénynév (paraméterlista)
RETURN típus
IS
 lokális változók
BEGIN
 utasítások
[EXCEPTION …]
END;
```

Gyakorlás

- Készíts olyan függvényt, amelynek a paraméterei két szám, a visszatérési értéke pedig a két szám minimuma.

Feladat

- Készíts olyan függvényt, amelynek a visszatérési értéke egy adott részlegnél a legmagasabb fizetés.

Feladat

- Készíts egy függvényt, amelynek paramétere a dolgozó vezetékneve, a visszatérési értéke pedig a dolgozó részlegének a neve. Ha nincs ilyen nevű dolgozó, akkor a ‚Nincs ilyen’ szöveget adja vissza.

Feladat

- Készíts függvényt, ami visszaadja az n-edik legtöbb dolgozóval rendelkező részleg nevét.

Feladat

- Készíts egy függvényt, ami paraméterként kap egy részleg nevet, majd eredményként visszaadja, hogyan hányan dolgoznak abban a részlegben.

Feladat

- Készíts olyan függvényt, melynek paramétere egy manager ID -ja, visszatérési értéke pedig a beosztottainak száma. Ha nincsenek beosztottai, akkor 0 -val térjen vissza!

TRIGGEREK

Triggerek

- Adott esemény bekövetkezésére reagálunk (tárolt eljárást futtat)
- Egyfajta eseménykezelést valósít meg
- Milyen műveletek hatására aktivizálódik:
  INSERT, UPDATE, DELETE, CREATE, ALTER, DROP
- Mikor aktivizálódik: BEFORE, AFTER, INSTEAD OF

Triggerek

```
CREATE OR REPLACE TRIGGER
  {név}
  [BEFORE | AFTER | INSTEAD
  OF]
  {esemény} [OR {esemény} …]
  ON {tábla}
[FOR EACH ROW [WHEN
  {feltétel}]]
[DECLARE {változók}]
BEGIN
  {utasítások}
```

Trigger példa

```
create table departments2 as select * from departments;

create or replace trigger Jelzo
after insert on departments2
begin
  dbms_output.put_line('Új részleg beszúrva!');
end;

insert into departments2 (department_id, department_name)
values (22, 'asd');
```

Gyakorlás

- Készíts egy triggert, amely az employees2 táblán végzett beszúrás és törlés előtt fut le minden érintett sorra. Kiírja az érintett dolgozó vezetéknevét és a rajta végzett műveleteket (beszúrás/törlés).

Feladat

- Készíts triggert, amely UPDATE művelet után kiírja az érintett dolgozók vezetéknevét, régi és új fizetésüket.
- Továbbfejlesztés: csak akkor írjon ki bármit, ha a fizetés értéke megváltozott.
- Továbbfejlesztés: UPDATE előtt fusson le, illetve amennyiben december van, akkor írja ki azt is, hogy ‚Kellemes ünnepeket!’.

Feladat

- Készíts triggert amelyik megakadályozza, hogy csökkentsük a dolgozók fizetését!

Feladat

- Készíts egy triggert, amely nem enged egyetlen alkalmazottnak sem department –
  et váltani anélkül, hogy az új főnöke annak
  a departmentnek a managere len ne a továbbiakban!

Feladat

- Készíts egy triggert, ami ellenőrzi, hogy a countries táblába csak olyan country_name-et lehessen beszúrni (vagy módosítani), ami az angol ABC kis- és nagybetűit tartalmazza!
- Segítség: használd a REGEXP_LIKE függvényt az ellenőrzéshez!
- A trigger működését ezzel tesztelheted:

```
insert into countries values('HU', '1234', 2);
```

TCL UTASÍTÁSOK

Tranzakciók

- `SAVEPOINT ittmentettünk;`
  - mentési pont létrehozása
- `ROLLBACK TO ittmentettünk;`
  - visszagörgetés a megadott mentési pontig
- `ROLLBACK;`
  - visszagörgeti az aktuális tranzakciót
- `COMMIT;`
  - véglegesítés

Feladat

1. Készítsünk másolatot a **departments** tábláról **departments2** néven.
2. Készítsünk mentési pontot **sp** néven.
3. Szúrjunk be egy tetszőleges sort a **departments2** táblába.
4. Listázzuk a táblát.
5. Álljunk vissza az **sp** mentési pontra.
6. Listázzuk ismét a táblát.

Megoldás

```
Create table departments2 as select
* from departments;

SAVEPOINT sp;

INSERT INTO departments2
VALUES
(300,'Cheesmakers',200,1500);

SELECT * FROM departments2;

ROLLBACK TO sp;

SELECT * FROM departments2;
```

Feladat 2

1. Készítsünk mentési pontot **sp2** néven.
2. Készítsünk másolatot a **departments** tábláról **departments2** néven.
3. Listázzuk a táblát.
4. Álljunk vissza az **sp2** mentési pontra.
5. Listázzuk a táblát.

Megoldás

```
SAVEPOINT sp2;

CREATE TABLE departments2 AS
SELECT * FROM departments;

SELECT * FROM departments2;

ROLLBACK TO sp2;
```

- ... eltűnt az sp2 mentési pont!!

DDL utasításokat nem tudunk visszavonni, automatikusan véglegesíti őket a rendszer.

- **Emlékeztető:**
  TRUNCATE TABLE vs. DELETE FROM