

Adatbázisrendszerek II. – 9. Practice

Topic: Oracle PL/SQL programozás, PLSQL 2

Repository: NEPTUNKOD_DB2Gyak

Folder: NEPTUNKOD_0410

Forrás file-k: forrás fájlok .SQL

Töltse fel a GitHub rendszer aktuális mappába a jegyzőkönyvet és forrás fájlokat!

Határidő: aktuális gyakorlat időpontja, ill. módosítás esetén 2025.04.15.

Feladatok

Projekt név: NEPTUNKOD_PLSQL2

Csomagnév: neptunkoddb2

Feladat

Először a minta feladatsort készítse el, majd a Zoo táblához készítse a feladatokat.

Zoo						
ID	Nev	Fajta	Született	Suly	Ertek	Gondozo

1. Készítsen egy **NewPet(id,n,f,s1,s2,e,g)** nevű tárolt eljárást a **Zoo** táblához egy új állatka felviteléhez!
2. Készítsen egy **DelPet(id)** nevű tárolt eljárást egy állatka törléséhez! Ha nem létezik a megadott id, írja ki ennek tényét a kód. Ha megtörtént a törlés, írja ki ennek tényét a kód.
3. Készítse el a **ModPet(td,fn,d)** nevű tárolt eljárást, mellyel egy állatka tetszőleges adatát módosíthatja!
4. Készítsen egy **PetData(id)** nevű tárolt eljárást egy állatka nevének, fajtájának, gondozójának kiírására. Ha nem létezik a megadott id, írja ki ennek tényét a kód.
5. Készítsen egy **PetData2(name)** nevű tárolt eljárást, mely a megadott nevű állatka fajtáját, és értékét írja ki. Ha nem létezik a megadott név, írja ki ennek tényét a kód.

6. Készítsen egy `ThereIsPetCalled(name)` nevű függvényt, mely visszaadja, hogy van-e (yes, no) megadott nevű állatka. (+1 pont)
7. Készítse el a `CallThereIsPetCalled(name)` nevű tárolt eljárást, melyben meghívja a `ThereIsPetCalled(name)` nevű függvényt, és a visszaadott érték függvényében kiírja, hogy "Van `name` nevű kisállatunk" vagy "Nincs `name` nevű kisállatunk". (+1 pont)
8. Készítsen egy `Insurance()` nevű függvényt, mely visszaadja az állatok biztosításának havi összegét. (+2 pont) A havi díj:
 - 1 millió Ft érték alatt az érték 1,3%-a,
 - 3 millió Ft érték alatt az érték 1,2%-a,
 - 3 millió Ft értéktől az érték 1,1%-a.

HAVI_BIZTOSÍTÁSI_ÖSSZEG

331344

Tárolt eljárások

Rekord felvitele

Auto						
ID	Rendszam	Tipus	Szin	Kor	Ar	Muszaki_erv

Tárolt eljárás rekord beszúrására:

■ Tárolt eljárás rekord beszúrására:

```
CREATE OR REPLACE Procedure UjAuto (idbe in number, rszbe in char,
                                     tipbe in char, szinbe in char, korbe in number,
                                     arbe in number, muszbe in char) is
BEGIN
  INSERT INTO Auto VALUES (idbe, rszbe, tipbe, szinbe, korbe, arbe, muszbe);
END;
```

```
BEGIN
  UjAuto(4, 'RSV616', 'Ford KA', 'Zöld', 16, 900000, '07.18.2023');
END;
```

Rekord módosítása

Tárolt eljárás paraméterként megadott rekord egy adatának módosítására:

Auto						
ID	Rendszam	Tipus	Szin	Kor	Ar	Muszaki_erv

```
CREATE OR REPLACE Procedure AutoModosit (idbe in number, szinbe in char) is
BEGIN
    UPDATE Auto SET Szin = szinbe WHERE ID = idbe;
END;
```

```
BEGIN
    AutoModosit(2, 'Narancs');
END;
```

Univerzális módosító

Auto						
ID	Rendszam	Tipus	Szin	Kor	Ar	Muszaki_erv

```
CREATE OR REPLACE Procedure UniAutoModosito (idbe in number,
                                              mezo in char, adat in varchar2) is
    plssql_p VARCHAR2(500);
BEGIN
    plssql_p:='update auto set '||mezo||' = '''||adat||''''
                                              where id = '||idbe;
    EXECUTE IMMEDIATE plssql_p;
END;
```

```
BEGIN
    UniAutoModosito(2, 'szin', 'Barna');
END;
```

```
BEGIN
    UniAutoModosito(2, 'ar', '1000000');
END;
```

```
BEGIN
    UniAutoModosito(2, 'ar', 1000000);
END;
```

A PL/SQL lehetővé teszi azt is, hogy egy SQL utasítás pontos szövege csak futási időben alakuljon ki, tehát futásról futásra változhasson. Az ilyen típusú utasítások a *dinamikus SQL* utasítások.

ID	RENSZAM	TIPUS	SZIN	KOR	AR	MUSZAKI_ERV
2	QWJ612	Suziki Alto	Barna	15	1000000	03/28/2024

Magyarázat

```
plsql_p:='update auto set '||mezo||' = '''||adat||'''  
                                where id = '||idbe;
```

Ha a szövegben szükséges egy aposztróf ('), hármat kell belőle beírni!

```
BEGIN  
    UniAutoModosito(2, 'ar', '1000000');  
END;
```

A dinamikusan összeállított parancs jelen esetben:

```
update auto set ar = '1000000' where id = 2
```

Az SQL szabvány lehetőséget ad arra, hogy a numerikus adatokat is aposztrófok között adjuk meg!

Rekord törlése

Auto						
ID	Rendszam	Tipus	Szin	Kor	Ar	Muszaki_erv

■ Tárolt eljárás paraméterként megadott rekord törlésére:

```
CREATE OR REPLACE Procedure AutoTorol (idbe in number) is  
BEGIN  
    DELETE FROM Auto WHERE ID = idbe;  
END;
```

```
BEGIN  
    AutoTorol(2);  
END;
```

Rekord módosítása 2.

Auto						
ID	Rendszam	Tipus	Szin	Kor	Ar	Muszaki_erv

```
CREATE OR REPLACE Procedure AutoMod2 (idbe in number, szinbe in char) is
    pc number;
BEGIN
    SELECT COUNT(*) INTO pc FROM Auto WHERE ID= idbe;
    IF pc < 1 THEN
        dbms_output.put_line('Nem létező ID!');
    ELSE
        UPDATE AUTO SET Szin = szinbe WHERE ID= idbe;
        dbms_output.put_line('Módosítás: OK!');
    END IF;
END;
```

```
BEGIN
    AutoMod2(3, 'Fekete');
END;
```

Csak akkor írja ki, hogy sikeres a módosítás, ha létezik az adott ID-jű rekord.

Példa az **IF THEN ELSE END IF** szerkezet használatára.

PLSQL IF utasítás

```
IF feltétel1 THEN
    --utasítások1
ELSIF feltétel2 THEN
    --utasítások2
ELSIF feltétel3 THEN
    --utasítások3
...
ELSE
    --utasításokN
END IF;
```

Rekord törlése 2.

Auto						
ID	Rendszam	Tipus	Szin	Kor	Ar	Muszaki_erv

```
CREATE OR REPLACE Procedure AutoTorol2 (idbe in number) is
BEGIN
    ???
END;
```

Feladat ugyanaz, mint az előbbi kódnál: ha nincs a paraméterben megadott ID, írja ki ezt a kód, ha van, írja ki, hogy: "A **xxx** ID-jű rekord törlődött az adatbázisból!".

```
BEGIN
    AutoTorol2(1);
END;
```

Átlagár

```
CREATE OR REPLACE Procedure AtlagAr is
    avgAr number;
BEGIN
    SELECT avg(ar) INTO avgAr FROM Auto;
    dbms_output.put_line('Az autók átlagára: ' || avgAr);
END;
```

```
BEGIN
    AtlagAr;
END;
```

Adatkiírás

Auto						
ID	Rendszam	Tipus	Szin	Kor	Ar	Muszaki_erv

```
CREATE OR REPLACE Procedure AdatKiir (idbe in number) is
  pc number;
  rsz varchar2(40);
  tip varchar2(40);
  arr int;
BEGIN
  SELECT COUNT(*) INTO pc FROM Auto WHERE ID= idbe;
  if pc < 1 then
    dbms_output.put_line('Nem létező ID: '||idbe);
  else
    SELECT rendszam, tipus, ar INTO rsz, tip, arr FROM Auto WHERE id= idbe;
    dbms_output.put_line('Rendszám: '||rsz||' típus: '||tip||' ár: '||arr);
  end if;
END;
```

Minősítés

Auto						
ID	Rendszam	Tipus	Szin	Kor	Ar	Muszaki_erv

```
CREATE OR REPLACE Procedure Minosit (idbe in number) is
  arr int;
BEGIN
  SELECT ar INTO arr FROM Auto WHERE ID= idbe;
  if arr < 1500000 then
    dbms_output.put_line('Ára: '||arr||' az autó olcsó!');
  elsif arr < 4000000 then
    dbms_output.put_line('Ára: '||arr||' az autó ára átlagos!');
  else
    dbms_output.put_line('Ára: '||arr||' az autó drága!');
  end if;
END;
```

Tárolt függvények

SzínÁtlagAr

Auto						
ID	Rendszam	Tipus	Szin	Kor	Ar	Muszaki_erv

```
CREATE OR REPLACE Function SzinAtlagAr (szinbe in char) RETURN int AS
    atlag int:= 0;
BEGIN
    SELECT avg(Ar) INTO atlag FROM Auto WHERE Szin= szinbe;
    RETURN atlag;
END;
```

Elindítási lehetőségek

(A) `SELECT SzinAtlagAr('Piros') from dual;`

(B)

```
DECLARE
    atl int;
BEGIN
    atl := SzinAtlagAr('Kék');
    DBMS_OUTPUT.PUT_LINE('Az átlag: '||atl);
END;
```

AutóIdRendszám

Auto						
ID	Rendszam	Tipus	Szin	Kor	Ar	Muszaki_erv

```
CREATE OR REPLACE Function AutoIdRendszam (idbe int) RETURN varchar2 AS
    ret varchar2(80):= 'Nem létező azonosító: '||idbe;
    pc number;
    rsz varchar2(20);
BEGIN
    SELECT COUNT(*) INTO pc FROM Auto WHERE Id= idbe;
    IF pc > 0 THEN
        SELECT rendszam INTO rsz FROM Auto WHERE ID= idbe;
        ret:= 'A(z) '||idbe||'ID-jű autó rendszáma: '||rsz;
    END if;
    RETURN ret;
END;
```


ArTartománybanDarab

```
CREATE OR REPLACE Function ArTartomanybanDarab (minArbe int, maxArbe int)
                                                    RETURN int AS
    db int:= 0;
BEGIN
    SELECT COUNT(*) INTO db FROM Auto WHERE Ar > minArbe AND Ar < maxArbe;
    RETURN db;
END;
```

VanSzinesAuto

```
CREATE OR REPLACE Function VanSzinesAuto (szinbe char) RETURN boolean as
    ret boolean:= false;
    db int:= 0;
BEGIN
    SELECT COUNT(*) INTO db FROM Auto WHERE Szin = szinbe;
    IF db > 0 THEN
        ret:= true;
    END IF;
    RETURN ret;
END;
```

A **logikai** visszatérő értékű függvények nem hívhatók a SELECT paranccsal!

(A) SELECT VanSzinesAuto('Piros') from dual;



Error at line 1/8: ORA-00902: invalid datatype

(B)

```
DECLARE
    ret boolean:= false;
    szn varchar2(30):='Piros';
BEGIN
    ret:= vanszinesauto(szn);
    if ret then dbms_output.put_line('Van '||szn||' autó');
    else dbms_output.put_line('Nincs '||szn||' autó');
    end if;
END;
```

Van Piros autó

Nincs Lila autó