

Adatbázis rendszerek II.

Gyakorlati feladatsor

Készítette:
Oravecz Áron
D3U3EE

Dátum:
2022 03.23

1. Írjon egy olyan PL/SQL programot, ami kiírja a kimenetre, hogy "Vezetéknév Keresztnév"! (VezKer.sql)

```
1 DECLARE
2     v_name varchar(40):='Vezetéknév Keresztnév';
3 BEGIN
4     dbms_output.put_line(v_name);
5 END;
```

Results Explain Describe Saved SQL History

Vezetéknév Keresztnév

2. Írjon egy olyan PL/SQL programot, amely összead két számot és kiírja a kimenetre! (osszead.sql)

```
1 DECLARE
2     a int(10) :=1;
3     b int(10) :=2;
4 BEGIN
5     dbms_output.put_line(a+b);
6 END;
```

Results Explain Describe Saved SQL History

3

Statement processed.

0.03 seconds

3. Írjon egy olyan PL/SQL programot, amely összeszoroz két számot és kiírja a kimenetre! (szorzas.sql)

```
1 DECLARE
2   a int(10) :=2;
3   b int(10) :=64;
4 BEGIN
5   dbms_output.put_line(a*b);
6 END;
```

Results Explain Describe Saved SQL History

128

Statement processed.

4. Írjon egy olyan PL/SQL programot, amely kiírja a "Vezetéknév Keresztnév" szöveget *csupa nagy ill. kis betűkkel* a kimenetére! (nagybetu.sql)

```
1 DECLARE
2   k_name varchar2(20):='áron';
3 BEGIN
4   dbms_output.put_line(Upper(k_name));
5 END;
```

Results Explain Describe Saved SQL History

ÁRON

Statement processed.

0.00 seconds

```
1 DECLARE
2   k_name varchar2(20):='ISTVÁN';
3 BEGIN
4   dbms_output.put_line(LOWER(k_name));
5 END;
```

Results Explain Describe Saved SQL History

istván

Statement processed.

0.01 seconds

5. Írjon egy olyan PL/SQL programot, amely összefűzve kiírja a kimenetére a "Vezetéknév " és a "Keresztnév" string-eket! (összefuz.sql)

```
1 DECLARE
2   v_name varchar2(20):='SZENT';
3   k_name varchar2(20):='ISTVÁN';
4 BEGIN
5   dbms_output.put_line(v_name || ' ' || k_name);
6 END;
```

Results Explain Describe Saved SQL History

SZENT ISTVÁN

Statement processed.

0.01 seconds

6. Írjon egy olyan PL/SQL programot, amely kiírja az aktuális rendszeridőt! (sysdate.sql)

```
1 DECLARE
2
3 BEGIN
4     dbms_output.put_line(sysdate);
5 END;
```

Results Explain Describe Saved SQL History

03/16/2022

Statement processed.

0.01 seconds

7. Írjon egy olyan PL/SQL programot, amely kiírja a rendszeridőt 'YYYY-MM-DD' formátumban! (sysdate1.sql)

```
1 DECLARE
2     v_name varchar2(20):='SZENT';
3     k_name varchar2(20):='ISTVÁN';
4 BEGIN
5     dbms_output.put_line(CURRENT_DATE);
6 END;
```

Results Explain Describe Saved SQL History

03/16/2022

Statement processed.

0.00 seconds

8. Számítsa ki a kör területét, ha az $r=12$. (kor.sql)

$2 \cdot r \cdot \pi$

```
1 DECLARE
2     r int(3) := 15;
3     pi float(10) := 3.14;
4 BEGIN
5     dbms_output.put_line(2*r*pi);
6 END;
```

Results Explain Describe Saved SQL History

94.2

Statement processed.

0.03 seconds

Vezérlési szerkezet

2. Írjon egy olyan PL/SQL programot, amely eldönti két szám közül, hogy melyik a nagyobb (a=10; b=81)! (vezerles2.sql)

```
1 DECLARE
2     a int(3) := 10;
3     b int(3) := 81;
4 BEGIN
5     IF a < b THEN
6         dbms_output.put_line(b);
7
8
9     ELSIF a > b THEN
10        dbms_output.put_line(a);
11
12 END IF;
13
14 END;
```

Results Explain Describe Saved SQL History

100

Statement processed.

0.00 seconds

3. Írjon egy olyan PL/SQL programot, amely eldönti egy számról, hogy bele esik e két másik szám által megadott intervallumba! (vezerles3.sql)

Az intervallum: min =10; max=100;

ertek 78;

1

```
1 DECLARE
2     min int(3) := 10;
3     max int(3) := 100;
4     ertek int(3) := 78;
5 BEGIN
6     IF ertek < max THEN
7         IF ertek > min THEN
8             dbms_output.put_line('beleesik az intervallumba');
9         END IF;
10
11        ELSE
12            dbms_output.put_line('Nem megfelelő intervallum');
13        END IF;
14
15    END;
```

2

```
1 DECLARE
2     min int(3) := 10;
3     max int(3) := 100;
4     ertek int(3) := 78;
5 BEGIN
6     IF ertek < max AND ertek > min THEN
7         dbms_output.put_line('beleesik az intervallumba');
8     ELSE
9         dbms_output.put_line('Nem megfelelő intervallum');
10    END IF;
11
12    END;
```

4. Ismertesse a Swith Case szerkezetet!

```
1 DECLARE
2     nameChar char(1) := 'A';
3 BEGIN
4     CASE nameChar
5         when 'P' then dbms_output.put_line('Petra');
6         when 'L' then dbms_output.put_line('Laci');
7         when 'R' then dbms_output.put_line('Robi');
8         when 'J' then dbms_output.put_line('Juli');
9         when 'Li' then dbms_output.put_line('Levi');
10        when 'B' then dbms_output.put_line('Bence');
11        when 'A' then dbms_output.put_line('Anna');
12        when 'V' then dbms_output.put_line('Viki');
13        when 'Zs' then dbms_output.put_line('Zsolt');
14        else dbms_output.put_line('Nincs ilyen név');
15    END CASE;
16 END;
```

Results Explain Describe Saved SQL History

Anna

Statement processed.

5. Írjon egy olyan PL/SQL programot, amely a beosztás rövidítése alapján megadja, a dolgozó teljes beosztását! (vezerles4.sql) Pl.: beosztas = 'root', ez alapján kiírja: Rendszergazda (UNIX/LINUX).

```
1 DECLARE
2     beosztas char(10) := 'root';
3 BEGIN
4     CASE beosztas
5         when 'root' then dbms_output.put_line('Rendszergazda(UNIX/LINUX)');
6         when 'engine' then dbms_output.put_line('Járműmérnök(INVENTOR)');
7         when 'chef' then dbms_output.put_line('Szakács(PAN)');
8         when 'writer' then dbms_output.put_line('Író(PEN)');
9         when 'pilot' then dbms_output.put_line('Repülőgépezető(Joystick)');
10        when 'police' then dbms_output.put_line('Rendőrk(NIGHTSTICK)');
11        when 'fireman' then dbms_output.put_line('Tűzoltó(AXE)');
12        else dbms_output.put_line('Nincs ilyen dolgozó');
13    END CASE;
14 END;
```

6. Írjon egy olyan PL/SQL programot, amely eldönti három számról, hogy alkot-e háromszöget! (vezerles5.sql)

a=3; b=4; c=5;


```
1 DECLARE
2
3 av int(1) :=3;
4 bv int(1) :=4;
5 ck int(1) :=5;
6
7
8 BEGIN
9     IF av+bv > ck THEN
10         dbms_output.put_line('létrejöhhet a háromszög');
11     ELSIF bv+ck > av THEN
12         dbms_output.put_line('létrejöhhet a háromszög');
13     ELSIF av+ck > bv THEN
14         dbms_output.put_line('létrejöhhet a háromszög');
15     ELSE
16         dbms_output.put_line('NEM jöhhet létre a háromszög');
17     END IF;
18
19
20
```

Results	Explain	Describe	Saved SQL	History
létrejöhhet a háromszög				
Statement processed.				

7. Írjon egy olyan PL/SQL programot, amely a háromszög három oldala ismeretében kiszámolja a területét a Héron képlet segítségével! (vezerles6.sql)

a=3; b=4; c=5;

Heron képlet:

$$T = \sqrt{s(s-a)(s-b)(s-c)}$$
$$s = \frac{a+b+c}{2}$$

ahol a , b és c a háromszög oldalai, s a háromszög kerületének a fele, és T a háromszög területe.

```

1  DECLARE
2  av int(1) :=3;
3  bv int(1) :=7;
4  ck int(1) :=5;
5
6  s float(4):=(av+bv+ck)/2;
7
8  T float(4):=sqrt(s*(s-av)*(s-bv)*(s-ck));
9
10 BEGIN
11
12  dbms_output.put_line('A terület értéke: ');
13  dbms_output.put_line(T);
14
15  dbms_output.put_line('A kerület értéke:');
16  dbms_output.put_line(s);
17
18  END;

```

Results	Explain	Describe	Saved SQL	History
---------	---------	----------	-----------	---------

```

A terület értéke:
6.5
A kerület értéke:
7.5

```

8. Ismertesse a Loop, While és a For ciklust PL/SQL-ben!

Loop:

```

1  DECLARE
2  i number(1);
3  j number(1);
4  BEGIN
5  << outer_loop >>
6  FOR i IN 1..5 LOOP
7  << inner_loop >>
8  FOR j IN 1..5 LOOP
9  dbms_output.put_line('i értéke : ' || i || ' és j értéke: ' || j);
10 END loop inner_loop;
11 END loop outer_loop;
12 END;
13

```

```
i értéke : 1 és j értéke: 1
i értéke : 1 és j értéke: 2
i értéke : 1 és j értéke: 3
i értéke : 1 és j értéke: 4
i értéke : 1 és j értéke: 5
i értéke : 2 és j értéke: 1
i értéke : 2 és j értéke: 2
i értéke : 2 és j értéke: 3
i értéke : 2 és j értéke: 4
i értéke : 2 és j értéke: 5
```

While:

```
1  DECLARE
2
3  BEGIN
4  WHILE condition
5  | LOOP
6  |   'Utasítások amiket ismételni szeretnénk';
7  |   END LOOP;
8  |
9  END;
```

```
1  DECLARE
2  |   num NUMBER := 1;
3  BEGIN
4  |   WHILE num <= 10
5  |   | LOOP
6  |   |   DBMS_OUTPUT.PUT_LINE(num);
7  |   |   num := num+1;
8  |   |   END LOOP;
9  END;
```

Results	Explain	Describe	Saved SQL	History
---------	---------	----------	-----------	---------

1
2
3
4
5
6
7
8
9
10

Statement processed.

For loop:

```
1 DECLARE
2 BEGIN
3   FOR var IN 1..10
4   LOOP
5     DBMS_OUTPUT.PUT_LINE(var);
6   END LOOP;
7 END;
```

1
2
3
4
5
6
7
8
9
10

9. Írjon egy olyan PL/SQL programot, amely kiírja 1-től n-ig (n=10) a számokat!
(vezerles7.sql)

```
1 DECLARE
2 BEGIN
3   FOR var IN 1..10
4   LOOP
5     DBMS_OUTPUT.PUT_LINE(var);
6   END LOOP;
7 END;
```

Results	Explain	Describe	Saved SQL	History
1 2 3 4 5 6 7 8 9 10				

10. Írjon egy olyan PL/SQL programot amely kiírja az első n darab Fibonacci számot!
(vezerles8.sql)

```

1  DECLARE
2  av int(3):=1;
3  bv int(3):=1;
4  x int(3):= bv;
5
6  BEGIN
7      WHILE x <= 4
8      LOOP
9          x:=av+bv;
10         av:=bv;
11         bv:=x;
12         DBMS_OUTPUT.PUT_LINE(x);
13     END LOOP;
14 END;
```

Results Explain Describe Saved SQL History

```

2
3
5
```

Statement processed.

0.01seconds

11. Írjon egy olyan PL/SQL programot amely eldönti egy n számról hogy prím szám-e! (vezerles9.sql) (n=10)

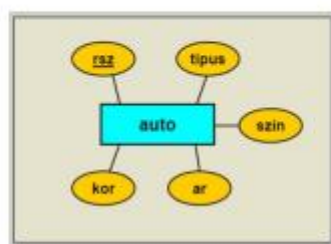
```

1  DECLARE
2  n int(3):= 10;
3  BEGIN
4  FOR i IN 1..10
5  LOOP
6      IF n%i ==0 THEN
7          IF n%n ==0 THEN
8              DBMS_OUTPUT.PUT_LINE('Ez a szám prím');
9          END IF;
10     ELSE
11         DBMS_OUTPUT.PUT_LINE('Ez a szám NEM prím');
12     END IF;
13 END LOOP n;
14 END;
```

(nem 100%-os)

1. Az ER modell alapján készítse el a következő táblát: Mentés: **neptunkod_auto.sql**

Hozzon létre egy *auto* nevű táblát a *PLSQL fejlesztővel*.



```

1 CREATE TABLE auto(
2     rsz INT(6),
3     tipus VARCHAR2(20) NOT NULL,
4     szin VARCHAR2(20) NOT NULL,
5     ar INT(100),
6     kor INT(10),
7     PRIMARY KEY(rsz)
8 );
9
10

```

2. Írjon egy PL/SQL blokkot egy autó rekord felvitelére, ahol a mezőértékeket memóriaváltozókon keresztül adjuk meg, melyek típusa az AUTO tábla mezőinek típusával megegyező. **Mentés:** *autos_feltoltegysor.sql*

auto				
rsz	tipus	szin	kor	ar

3. Töltse fel a táblát a minta alapján. **Mentés:** *autos_feltolt.sql*

RSZ	TIPUS	SZIN	KOR	AR
1 ABC500	Opel Corsa	piros	8	720000
2 MCC325	Opel Insignia	fekete	2	6800000
3 BBM104	Suzuki Swift	piros	5	1500000
4 CHR411	Renault Twingo	piros	12	630000
5 JRE2932	BMW M3	fekete	5	4500000
6 JEG113	Opel Corsa	piros	7	810000
7 DER042	Seat Ibiza	szürke	14	450000
8 BAB422	Lada 1300S	fehér	28	198000
9 UFF666	Audi TT	fekete	4	7500000
10 ABC124	Fiat	Piros	3	18000000

4. Írjon egy PL/SQL blokk-t, mely egy memóriaváltozóban megadott kornál idősebb autók (7 év) árát csökkenti 10%-al. **Mentés:** *auto_ar_csokk_10.sql*

5. Írjon egy PL/SQL blokk-t az autók átlagárának lekérdezésére és kiírására.

6. Írjon egy PL/SQL blokk-t, amely a memóriaváltozóban 10 évnél idősebb auto törlésére. **Mentés:** *autos_kor_torles.sql*