

Adatbázis rendszerek II.

Gyakorlati feladatsor

Készítette:
Oravecz Áron
D3U3EE

Dátum:
2022 04.27

Feladatok

1. Készítse el a Vasarlas táblát.(vasarlas.sql)

SORSZAM	NUMBER (38,0)	No	(null)	1 (null)
IDOPONT	TIMESTAMP (6)	Yes	Cur...	2 (null)
TKOD	CHAR (3 BYTE)	No	(null)	3 (null)
DARAB	NUMBER (38,0)	Yes	(null)	4 (null)
VID	CHAR (3 BYTE)	No	(null)	5 (null)

```
CREATE TABLE Vasarlas(
```

```
sorszam number,
```

```
idopont timestamp,
```

```
tkod char(3),
```

```
darab number;
```

```
vid char(3));
```

```
1 CREATE TABLE Vasarlas(
2   sorszam number,
3   idopont timestamp,
4   tkod char(3),
5   darab number,
6   vid char(3));
7
```

Results Explain Describe Saved SQL

Table created.

0.02 seconds

2. Töltse fel a Vasarlas táblát (vasarlas_feltolt.sql)

SORSZAM	IDOPONT	TKOD	DARAB	VID
1	22-MÁRC. -10 09.27.27,2170000000	t14	1	v06
2	22-MÁRC. -10 09.27.27,2480000000	t15	1	v06
4	22-MÁRC. -10 09.27.27,4040000000	t03	4	v01
5	22-MÁRC. -10 09.27.27,4200000000	t04	1	v01
6	22-MÁRC. -10 09.27.27,4350000000	t02	2	v02
7	22-MÁRC. -10 09.27.27,4510000000	t06	2	v02
8	22-MÁRC. -10 09.27.27,4670000000	t07	4	v02
11	22-MÁRC. -10 09.27.27,5140000000	t11	3	v04
13	22-MÁRC. -10 09.27.27,5450000000	t13	100	v05

```

1 begin
2 insert into Vasarlas values(1,TO_DATE('2022-05-27','YYYY-MM-DD'),'t14',1,'v06');
3 insert into Vasarlas values(2,TO_DATE('2022-05-27','YYYY-MM-DD'),'t15',1,'v06');
4 insert into Vasarlas values(4,TO_DATE('2022-05-27','YYYY-MM-DD'),'t03',1,'v01');
5 insert into Vasarlas values(5,TO_DATE('2022-05-27','YYYY-MM-DD'),'t04',1,'v01');
6 insert into Vasarlas values(6,TO_DATE('2022-05-27','YYYY-MM-DD'),'t02',1,'v02');
7 insert into Vasarlas values(7,TO_DATE('2022-05-27','YYYY-MM-DD'),'t06',1,'v02');
8 insert into Vasarlas values(8,TO_DATE('2022-05-27','YYYY-MM-DD'),'t07',1,'v04');
9 end;

```

Results Explain Describe Saved SQL History

1 row(s) inserted.

0.01 seconds

begin

insert into Vasarlas values(1,TO_DATE('2022-05-27','YYYY-MM-DD'),'t14',1,'v06');

insert into Vasarlas values(2,TO_DATE('2022-05-27','YYYY-MM-DD'),'t15',1,'v06');

insert into Vasarlas values(4,TO_DATE('2022-05-27','YYYY-MM-DD'),'t03',1,'v01');

insert into Vasarlas values(5,TO_DATE('2022-05-27','YYYY-MM-DD'),'t04',1,'v01');

insert into Vasarlas values(6,TO_DATE('2022-05-27','YYYY-MM-DD'),'t02',1,'v02');

insert into Vasarlas values(7,TO_DATE('2022-05-27','YYYY-MM-DD'),'t06',1,'v02');

insert into Vasarlas values(8,TO_DATE('2022-05-27','YYYY-MM-DD'),'t07',1,'v04');

end;

3. Készítsen egy tárolt eljárást (VDbKiir), mely kiírja a Vasarlas tábla rekordjainak számát. (vasarlas_Vdb_kiir.sql)

Futtassa a tárolt eljárást.

create or replace procedure VDbKiir as db Vasarlas.sorszam%type;

begin

SELECT COUNT(sorszam) into db from vasarlas;

dbms_output.put_line('A vásárlás tábla rekordjainak a száma:' || db);

end;

```
1 create or replace procedure VDbKiir as db Vasarlas.sorszam%type;
2 begin
3   SELECT COUNT(sorszam) into db from vasarlas;
4   dbms_output.put_line('A vásárlás tábla rekordjainak a száma:'||db);
5 end;
```

Results Explain Describe Saved SQL History

Procedure created.

0.03 seconds

```
1 begin
2   VDbKiir;
3 end;
```

Results Explain Describe Saved SQL History

A vásárlás tábla rekordjainak a száma:7

Statement processed.

0.01 seconds

4. Készítsen egy JOB-ot, mely azonnal elindul, 1 percenként ismétlődik, és végrehajtja a DbKiir eljárást (vasarlas_job.sql)

```
BEGIN
DBMS_SCHEDULER.CREATE_JOB (
  job_name => 'vasarlas_job',
  job_type => 'STORED_PROCEDURE',
  job_action => 'VDbKiir', --a tárolt eljárás neve
  start_date => SYSTIMESTAMP,
  repeat_interval => 'FREQ=SECONDLY; INTERVAL=1000',
  end_date => SYSTIMESTAMP + INTERVAL '30' day,
  comments => 'My new job');
```

END;

```
1 BEGIN
2 DBMS_SCHEDULER.CREATE_JOB (
3   job_name => 'vasarlas_job',
4   job_type => 'STORED_PROCEDURE',
5   job_action => 'VDbKiir', --a tárolt eljárás neve
6   start_date => SYSTIMESTAMP,
7   repeat_interval => 'FREQ=SECONDLY; INTERVAL=1000',
8   end_date => SYSTIMESTAMP + INTERVAL '30' day,
9   comments => 'My new job');
10 END;
```

Results Explain Describe Saved SQL History

Statement processed.

0.12 seconds

5. Írassa ki a futó JOB-ok listáját (vasarlas_fut.sql)

JOB	LAST_DATE	LAST_SEC	NEXT_DATE	INTERVAL	WHAT
1	(null)	(null)	22-APR.	-24 17:42:05	VDbKiir;

```
1 BEGIN
2   DBMS_SCHEDULER.RUN_JOB(
3     JOB_NAME      => 'vasarlas_job',
4     USE_CURRENT_SESSION => FALSE);
5 END;
```

Results Explain Describe Saved SQL History

Statement processed.

0.02 seconds

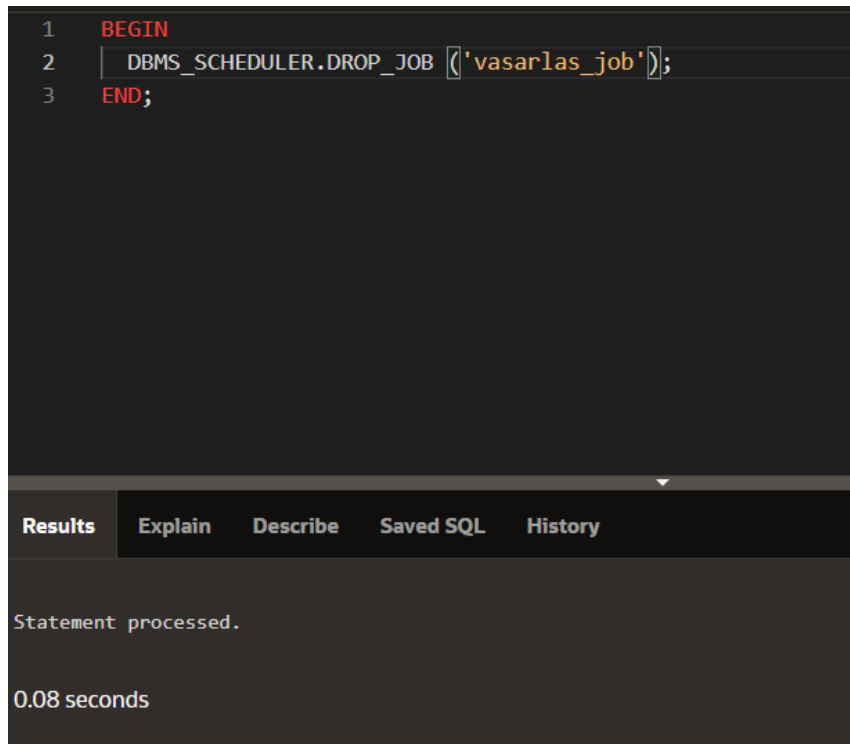
futtatás

6. Törölje le a JOB-ot.

BEGIN

DBMS_SCHEDULER.DROP_JOB ('vasarlas_job');

END;



The screenshot shows a SQL IDE interface with a dark theme. The top section contains a PL/SQL block with three lines of code: `1 BEGIN`, `2 DBMS_SCHEDULER.DROP_JOB ('vasarlas_job');`, and `3 END;`. The code is syntax-highlighted. Below the code editor is a horizontal tab bar with five tabs: **Results**, **Explain**, **Describe**, **Saved SQL**, and **History**. The **Results** tab is selected. The main area below the tabs displays the text `Statement processed.` and the execution time `0.08 seconds`.

```
1 BEGIN
2 DBMS_SCHEDULER.DROP_JOB ('vasarlas_job');
3 END;
```

Results Explain Describe Saved SQL History

Statement processed.

0.08 seconds