

Adatbázis rendszerek 2

Féléves egyéni **PL/SQL** feladat
Egyéni oktatási cégek

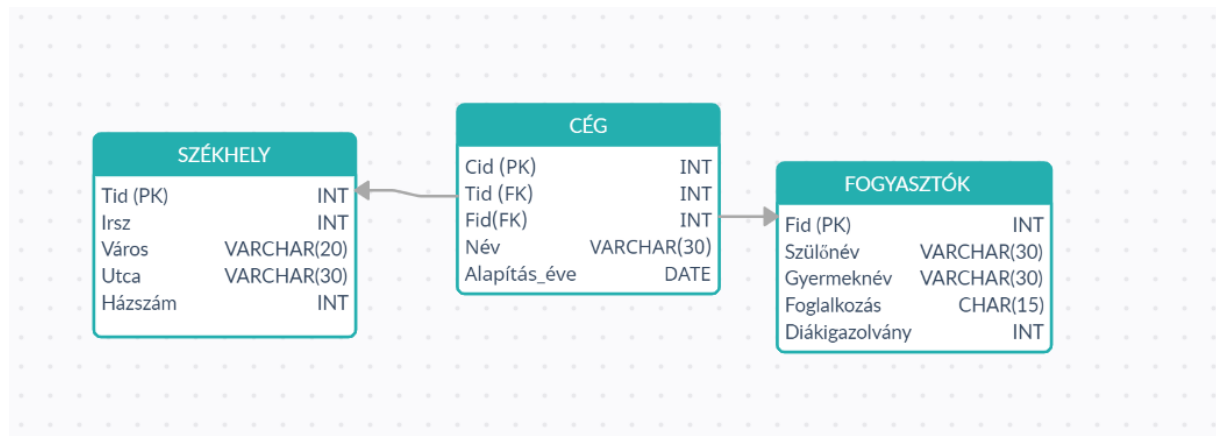
Készítette:

Oravecz Áron D3U3EE

Gyakorlatvezető:

Bednarik László

Adatbázis modell:



Adatbázis jellemzése:

- Székhely jelen esetben 1:1 kapcsolatban áll a Céggel.
- A Fogyasztók a céggel pedig 1:N kapcsolatban állnak egymással.
- A Székhely és a Fogyasztókat kapcsolótáblával köthetjük össze a Cég tábla jelenlétében.
- **Székhely:**
 - A Cég tartózkodási helyét tartalmazza. Elsődleges kulcsként egy Tid-t alkalmazok, ami azonosítóként szolgál.
 - Irányítószám: ahol elhelyezkedik a székhely körzeti szinten, ugyanakkor egész szám típusú rekord.
 - Város és az utca szöveges típusúak.
 - A házszám numerikus számként funkcionál jelen esetben.
- **Fogyasztók:**
 - A Céget igénybe vevő csoportot foglalja magába.
 - Fid: azonosító és egyben elsődleges kulcsként működik. Ő tárolja el az összefüggő rekordok adatait.
 - Gyermeknév és a Szülőnév szöveges adattípusként tárolandó.
 - Diákigazolvány pedig numerikus adattípus.
- **Cég:**
 - A Cég azonosítóval rendelkezik.

- De ebben az esetben leginkább kapcsolótáblaként szolgál, ami összekapcsolja a Fogyasztókat a Székhelyekkel.
- A másik két táblának a kulcsait tartalmazza, így azok idegen kulcsként értelmezhetőek.
- A Cég tartalmaz egy szöveges adattípust, ami a Cég nevét tárolja.
- Tartalmaz egy Dátum típusú adatot is a tábla, ami az alapítás dátumát tartalmazza.

Táblák létrehozása:

Székhely:

```
1 CREATE TABLE SZEKHELY
2 ( szekhely_id number(10) NOT NULL,
3   szekhely_irsz number(10),
4   szekhely_varos varchar2(50) NOT NULL,
5   szekhely_utca varchar2(50) NOT NULL,
6   department_hazsam number(10)
7 );
```

Results Explain Describe Saved SQL History

Table created.

0.10 seconds

Kiegészítés mert a primary key nem lett definiálva.

```
1 ALTER TABLE SZEKHELY
2 ADD CONSTRAINT szekhely_pk PRIMARY KEY (szekhely_id);
```

Results Explain Describe Saved SQL History

Table altered.

0.12 seconds

Column Name	Data Type	Nullable	Default	Primary Key
SZEKHELY_ID	NUMBER(10,0)	No	-	1
SZEKHELY_IRSZ	NUMBER(10,0)	Yes	-	-
SZEKHELY_VAROS	VARCHAR2(50)	No	-	-
SZEKHELY_UTCA	VARCHAR2(50)	No	-	-
DEPARTMENT_HAZSZAM	NUMBER(10,0)	Yes	-	-

Fogyasztók:

1 CREATE TABLE FOGYASZTOK
2 (
3 fogyaszo_id number(10) NOT NULL,
4 fogyaszo_szulonev varchar2(50) NOT NULL,
5 fogyaszo_gyermek varchar2(50) NOT NULL,
6 fogyaszo_foglalkozas varchar2(50) NOT NULL,
7 fogyaszo_diakigazolvany number(10),
8 CONSTRAINT fogyaszo_pk PRIMARY KEY (fogyaszo_id)
9);

Results Explain Describe Saved SQL History

Table created.

0.13 seconds

FOGYASZTO_ID	NUMBER(10,0)	No	-	1
FOGYASZTO_SZULONEV	VARCHAR2(50)	No	-	-
FOGYASZTO_GYERMEK	VARCHAR2(50)	No	-	-
FOGYASZTO_FOGLALKOZAS	VARCHAR2(50)	No	-	-
FOGYASZTO_DIAKIGAZOLVANY	NUMBER(10,0)	Yes	-	-

Cég:

1 CREATE TABLE CEG(
2 cid number PRIMARY KEY,
3 nev varchar2 (50),
4 alapitasiev DATE,
5 sz_id number,
6 f_id number,
7 CONSTRAINT sz_fk
8 FOREIGN KEY (sz_id)
9 REFERENCES SZEKHELY(szekhely_id),
10 CONSTRAINT f_fk
11 FOREIGN KEY (f_id)
12 REFERENCES FOGYASZTOK(fogyaszo_id));

Results Explain Describe Saved SQL History

Table created.

0.07 seconds

Column Name	Data Type	Nullable	Default	Primary Key
CID	NUMBER	No	-	1
NEV	VARCHAR2(50)	Yes	-	-
ALAPITASIEV	DATE	Yes	-	-
SZ_ID	NUMBER	Yes	-	-
F_ID	NUMBER	Yes	-	-
Download Print				

Táblák feltöltése:

Szekhely:

BEGIN

INSERT INTO SZEKHELY VALUES (1,1027,'Budapest','Varga köz',18);

INSERT INTO SZEKHELY VALUES (2,1025,'Budapest','Kossuth utca',7);

INSERT INTO SZEKHELY VALUES (3,3531,'Miskolc','Bársony út',2);

INSERT INTO SZEKHELY VALUES (4,102,'Pest','Pestőfi utca',30);

INSERT INTO SZEKHELY VALUES (5,1027,'Buda','Pákász utca',8);

INSERT INTO SZEKHELY VALUES (6,3000,'Füzesabony','Mihály köz',9);

END;

1 BEGIN

2 INSERT INTO SZEKHELY VALUES (1,1027,'Budapest','Varga köz',18);

3 INSERT INTO SZEKHELY VALUES (2,1025,'Budapest','Kossuth utca',7);

4 INSERT INTO SZEKHELY VALUES (3,3531,'Miskolc','Bársony út',2);

5 INSERT INTO SZEKHELY VALUES (4,102,'Pest','Pestőfi utca',30);

6 INSERT INTO SZEKHELY VALUES (5,1027,'Buda','Pákász utca',8);

7 INSERT INTO SZEKHELY VALUES (6,3000,'Füzesabony','Mihály köz',9);

8 END;

9

Results

Explain

Describe

Saved SQL

History

1 row(s) inserted.

0.08 seconds

Fogyasztó:

BEGIN

INSERT INTO FOGYASZTOK VALUES (1,'Bánfi Orsolya','Kovács Péter','Asztalos',7417175879);

INSERT INTO FOGYASZTOK VALUES (2, Tóth József','Tóth Miklós','Programozó',7417175880);

INSERT INTO FOGYASZTOK VALUES (3, 'Braskó Áron','Braskó Anett','Orvos',7417175130);

INSERT INTO FOGYASZTOK VALUES (4, 'Bánfi Orsolya', 'Kovács Norbert', 'Programozó', 7417175672);

INSERT INTO FOGYASZTOK VALUES (5, 'Kossuth Lajos', 'Kossuth János', 'Kohász', 7417175249);

INSERT INTO FOGYASZTOK VALUES (6, 'Szent István', 'Szent Árpád', 'Bányász', 7417175159);

END;

```
1 BEGIN
2 INSERT INTO FOGYASZTOK VALUES (1, 'Bánfi Orsolya', 'Kovács Péter', 'Asztalos', 7417175879);
3 INSERT INTO FOGYASZTOK VALUES (2, 'Tóth József', 'Tóth Miklós', 'Programozó', 7417175880);
4 INSERT INTO FOGYASZTOK VALUES (3, 'Braskó Áron', 'Braskó Anett', 'Orvos', 7417175130);
5 INSERT INTO FOGYASZTOK VALUES (4, 'Bánfi Orsolya', 'Kovács Norbert', 'Programozó', 7417175672);
6 INSERT INTO FOGYASZTOK VALUES (5, 'Kossuth Lajos', 'Kossuth János', 'Kohász', 7417175249);
7 INSERT INTO FOGYASZTOK VALUES (6, 'Szent István', 'Szent Árpád', 'Bányász', 7417175159);
8 END;
9
```

Results	Explain	Describe	Saved SQL	History
1 row(s) inserted.				
0.09 seconds				

Ceg:

BEGIN

INSERT INTO CEG VALUES (1, 'Logischool', TO_DATE ('20080521', 'YYYYMMDD'), 1, 4);

INSERT INTO CEG VALUES (2, 'WoodenSchool', TO_DATE ('20200118', 'YYYYMMDD'), 2, 1);

INSERT INTO CEG VALUES (3, 'Debreceni Egyetem Private School', TO_DATE ('20121231', 'YYYYMMDD'), 3, 3);

INSERT INTO CEG VALUES (4, 'Codecool', TO_DATE ('20100509', 'YYYYMMDD'), 4, 2);

INSERT INTO CEG VALUES (5, 'MinerLiner', TO_DATE ('19980520', 'YYYYMMDD'), 5, 6);

INSERT INTO CEG VALUES (6, 'Miskolci Egyetem Kohászat', TO_DATE ('19751021', 'YYYYMMDD'), 6, 5);

END;

```

1 BEGIN
2 INSERT INTO CEG VALUES (1,'Logischool', TO_DATE ('20080521', 'YYYYMMDD'),1 ,4);
3 INSERT INTO CEG VALUES (2,'WoodenSchool', TO_DATE ('20200118', 'YYYYMMDD'),2 ,1);
4 INSERT INTO CEG VALUES (3,'Debreceni Egyetem Private School', TO_DATE ('20121231', 'YYYYMMDD'),3 ,3);
5 INSERT INTO CEG VALUES (4,'Codecool', TO_DATE ('20100509', 'YYYYMMDD'),4 ,2);
6 INSERT INTO CEG VALUES (5,'MinerLiner', TO_DATE ('19980520', 'YYYYMMDD'),5 ,6);
7 INSERT INTO CEG VALUES (6,'Miskolci Egyetem Kohászat', TO_DATE ('19751021', 'YYYYMMDD'),6 ,5);
8 END;
9

```

Results Explain Describe Saved SQL History

1 row(s) inserted.

0.10 seconds

A Táblázatokkal néhány kapcsolatos függvény és lekérdezés

Darab számláló:

```
CREATE OR REPLACE PROCEDURE darab1 AS x FOGYASZTOK.fogyaszto_id%type;
```

```
BEGIN
```

```
SELECT COUNT(fogyaszto_id) INTO x FROM FOGYASZTOK;
```

```
DBMS_OUTPUT.PUT_LINE('A fogyasztók darabszáma: ' || x);
```

```
END;
```

```

1 CREATE OR REPLACE PROCEDURE darab1 AS x FOGYASZTOK.fogyaszto_id%type;
2 BEGIN
3 SELECT COUNT(fogyaszto_id) INTO x FROM FOGYASZTOK;
4 DBMS_OUTPUT.PUT_LINE('A fogyasztók darabszáma: ' || x);
5 END;
6

```

```
BEGIN
```

```
darab1;
```

```
END;
```

A fogyasztók darabszáma: 6

Statement processed.

0.00 seconds

PL/SQL függvény, ami egy megadott gyermek alapján megadja milyen szakterületen tanul.

```

CREATE OR REPLACE FUNCTION SzakmaGyereknek (gyermek IN CHAR) RETURN CHAR AS y
FOGYASZTOK.fogyaszto_foglalkozas%type;

BEGIN

SELECT fogyaszto_foglalkozas INTO y FROM FOGYASZTOK WHERE fogyaszto_gyermek = gyermek;

RETURN y;

END;

```

```

1 CREATE OR REPLACE FUNCTION SzakmaGyereknek (gyermek IN CHAR) RETURN CHAR as y FOGYASZTOK.fogyaszto_foglalkozas%type;
2 BEGIN
3 SELECT fogyaszto_foglalkozas INTO y FROM FOGYASZTOK WHERE fogyaszto_gyermek = gyermek;
4 RETURN y;
5 END;
6

```

Results Explain Describe Saved SQL History

Function created.

0.02 seconds

```

SELECT SzakmaGyereknek('Braskó Anett') FROM FOGYASZTOK;

```

SZAKMAGYEREKNEK('KOVÁCSPÉTER')
Asztalos

1 rows returned in 0.01 seconds Download

A **function** olyan, hogy vissza kell adnia egy visszatérési értéket

A **procedure** olyan, mint a C programozási nyelvben a void vagyis végrehajt valamit, de nem ad vissza értéket.

A feladat kötelező lépései:

- tárolt eljárás adatok felvitelére,

SZÉKHELY:

Megvizsgálom, hogy ha létezik már ilyen id-val egy székhely akkor nem sikeres a beszúrás. Ellenkező esetben felveszem.

```

create or replace procedure BeszurSzekhely(szid in number, irsz in number, szekhely_varos in char,
szekhely_utca in char, hazszam in number) as

```

```

    x number;

begin

select count(*) into x from SZEKHELY where szekhely_id = szid;

if x > 0 then

    dbms_output.put_line('Már létezik ezzel az ID-val egy székhely!');

else

```



```
insert into SZEKHELY values (szid, irsz, szekhely_varos, szekhely_utca, hazszam);
```

```
dbms_output.put_line('Egy sor beszúrva a Székhely táblába!');
```

```
end if;
```

```
end;
```

```
1 create or replace procedure BeszurSzekhely(szid in number, irsz in number, szekhely_varos in char, szekhely_utca in char, hazszam in number) as
2   x number;
3 begin
4   select count(*) into x from SZEKHELY where szekhely_id = szid;
5   if x > 0 then
6     dbms_output.put_line('Már létezik ezzel az ID-val egy székhely!');
7   else
8     insert into SZEKHELY values (szid, irsz, szekhely_varos, szekhely_utca, hazszam);
9     dbms_output.put_line('Egy sor beszúrva a Székhely táblába!');
10    end if;
11  end;
12
13
```

Results Explain Describe Saved SQL History

Procedure created.

0.03 seconds

```
BEGIN
```

```
SELECT BeszurSzekhely(10, 3532, 'Eger', 'Kálmán tér', 46) FROM SZEKHELY;
```

```
END;
```

```
1 BEGIN
2   BeszurSzekhely(10, 3532, 'Eger', 'Kálmán tér', 46);
3 END;
4
```

Results Explain Describe Saved SQL History

Egy sor beszúrva a Székhely táblába!

Statement processed.

0.06 seconds

rossz esetben:

```
Már létezik ezzel az ID-val egy székhely!

Statement processed.

0.01 seconds
```

FOGYASZTÓK:

Megnézem létezik e már ezzel az id-val a fogyasztó, ha nem akkor megnézem, hogy ugyan az e a diákigazolványszáma, ha az sem hasonlít egyikre se akkor felveszem.

create or replace procedure BeszurFogyasztok(fid in number, sznev in char, gynev in char, ffoglalkozas in char, diakig in number) as

 x number;

begin

select count(*) into x from FOGYASZTOK where fogyaszto_id = fid;

if x > 0 then

 dbms_output.put_line('Már létezik ezzel az ID-val egy fogyasztó!');

else

select count(*) into x from FOGYASZTOK where fogyaszto_diakigazolvany = diakig;

if x > 0 then

 dbms_output.put_line('Már létezik ezzel az igazolványszámmal egy fogyasztó!');

else

 insert into FOGYASZTOK values (fid, sznev, gynev, ffoglalkozas, diakig);

 dbms_output.put_line('Egy sor beszúrva a fogyasztók táblába!');

end if;

end if;

end;

```
1 create or replace procedure BeszurFogyasztok(fid in number, sznev in char, gynev in char, ffoglalkozas in char, diakig in number) as
2     x number;
3 begin
4     select count(*) into x from FOGYASZTOK where fogyaszto_id = fid;
5     if x > 0 then
6         dbms_output.put_line('Már létezik ezzel az ID-val egy fogyasztó!');
7     else
8         select count(*) into x from FOGYASZTOK where fogyaszto_diakigazolvany = diakig;
9         if x > 0 then
10             dbms_output.put_line('Már létezik ezzel az igazolványszámmal egy fogyasztó!');
11         else
12             insert into FOGYASZTOK values (fid, sznev, gynev, ffoglalkozas, diakig);
13             dbms_output.put_line('Egy sor beszúrva a fogyasztók táblába!');
14             end if;
15         end if;
16     end if;
17 end;
18
```

Results Explain Describe Saved SQL History

Procedure created.

0.09 seconds

```
1 BEGIN
2 BeszurFogyasztok(10, 'Tóth Péter', 'Tóth Annamária', 'Programozó',7412345678);
3 END;
```

Results Explain Describe Saved SQL History

Egy sor beszúrva a fogyasztók táblába!

Statement processed.

0.08 seconds

rossz esetben:

<pre>Már létezik ezzel az ID-val egy fogyasztó!</pre>	<pre>Már létezik ezzel az igazolványszámmal egy fogyasztó!</pre>
<pre>Statement processed.</pre>	<pre>Statement processed.</pre>
<pre>0.00 seconds</pre>	<pre>0.06 seconds</pre>

CÉG:

Megvizsgáljuk ez a cég létezik e már ezzel az ID, ha nem megnézzük a székhely id-ját jó helyen helyezkedik e el. Illetve megvizsgáljuk, ha az a fogyasztó már létezik e azon a telephelyen.

create or replace procedure BeszurCeg(cegid in number, cnev in char, datum in date, szekid in number, fogyid in number) as

x number;

begin

select count(*) into x from CEG where cid = cegid;

if x > 0 then

dbms_output.put_line('Már létezik ezzel az ID-val egy cég!');

else

select count(*) into x from CEG where sz_id = szekid;

if x > 0 then

dbms_output.put_line('Már létezik a cég ezzel a székhely id-val');

else

select count(*) into x from CEG where f_id = fogyid;

if x < 1 then

dbms_output.put_line('Nem létezik ez a fogyasztó id a cégnél!');

```

else

    insert into CEG values (cegid, cnev, datum, szekid, fogyid);

    dbms_output.put_line('Egy sor beszúrva a cég táblába!');

end if;

end if;

end if;
end;

```

```

1  create or replace procedure BeszurCeg(cegid in number, cnev in char, datum in date, szekid in number, fogyid in number) as
2      x number;
3  begin
4      select count(*) into x from CEG where cid = cegid;
5      if x > 0 then
6          dbms_output.put_line('Már létezik ezzel az ID-val egy cég!');
7      else
8
9          select count(*) into x from CEG where sz_id = szekid;
10         if x > 0 then
11             dbms_output.put_line('Már létezik a cég ezzel a székhely id-val');
12         else
13
14             select count(*) into x from CEG where f_id = fogyid;
15             if x > 0 then
16                 dbms_output.put_line('Már létezik ezzel a fogyasztó id-val ez a cég!');
17             else
18                 insert into CEG values (cegid, cnev, datum, szekid, fogyid);
19
Results Explain Describe Saved SQL History

```

Procedure created.

0.05 seconds

```

1  BEGIN
2  BeszurCeg(7, 'Dumaszinház', TO_DATE('20000612', 'YYYYMMDD'), 10, 3);
3  END;

```

Results Explain Describe Saved SQL History

Egy sor beszúrva a cég táblába!

Statement processed.

0.01 seconds

rossz esetben pl.:

<pre> Már létezik ezzel az ID-val egy cég! Statement processed. 0.00 seconds </pre>	<pre> Már létezik a cég ezzel a székhely id-val Statement processed. 0.00 seconds </pre>
---------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------

- tárolt eljárás adatok módosítására,

FOGYASZTO

Megváltoztatom a megadott fogyasztó id-ja alapján a diákigazolvány számát.

```
CREATE OR REPLACE PROCEDURE FOGYASZTODiakMod (fogyid in number, diakig in number) as x
number;
```

```
BEGIN
```

```
    SELECT COUNT(*) INTO x FROM FOGYASZTOK WHERE fogyasztó_id = fogyid;
```

```
    if x < 1 THEN
```

```
        DBMS_OUTPUT.PUT_LINE('Nem létezik ezzel az ID-val a Fogyasztó!');
```

```
    else
```

```
        UPDATE FOGYASZTOK SET fogyasztó_diakigazolvány = diakig where fogyasztó_id = fogyid;
```

```
        DBMS_OUTPUT.PUT_LINE('Sikeres Update');
```

```
    end if;
```

```
end;
```

```
1  CREATE OR REPLACE PROCEDURE FOGYASZTODiakMod (fogyid in number, diakig in number) as x number;
2  BEGIN
3      SELECT COUNT(*) INTO x FROM FOGYASZTOK WHERE fogyasztó_id = fogyid;
4      if x < 1 THEN
5          DBMS_OUTPUT.PUT_LINE('Nem létezik ezzel az ID-val a Fogyasztó!');
6      else
7          UPDATE FOGYASZTOK SET fogyasztó_diakigazolvány = diakig where fogyasztó_id = fogyid;
8          DBMS_OUTPUT.PUT_LINE('Sikeres Update');
9      end if;
10 end;
11
```

Results Explain Describe Saved SQL History

Procedure created.

0.13 seconds

1 BEGIN

2 FOGYASZTODiakMod (1, 7417175812);

3 END;

4

5

Results Explain Describe Saved SQL History

Sikeres Update

Statement processed.

0.01 seconds

Nem létezik ezzel az ID-val a Fogyasztó!

Statement processed.

0.00 seconds

SZEKHELY:

A székhely id segítségével megváltoztatom a székhely irányítószámát, ha az az id nem létezik, nem sikeres a módosítás.

```
CREATE OR REPLACE PROCEDURE SZEKHELYirszMod (szekid in number, irszmod in number) as x
number;
```

```
BEGIN
```

```
    SELECT COUNT(*) INTO x FROM SZEKHELY WHERE szekhely_id = szekid;
```

```
    if x < 1 THEN
```

```
        DBMS_OUTPUT.PUT_LINE('Nem létezik ezzel az ID-val a Székhely!');
```

```
    else
```

```
        UPDATE SZEKHELY SET szekhely_irsz = irszmod where szekhely_id = szekid;
```

```
        DBMS_OUTPUT.PUT_LINE('Sikeres update a székhely táblán');
```

```
    end if;
```

```
end;
```

```
1 CREATE OR REPLACE PROCEDURE SZEKHELYirszMod (szekid in number, irszmod in number)
2 as x number;
3 BEGIN
4     SELECT COUNT(*) INTO x FROM SZEKHELY WHERE szekhely_id = szekid;
5     if x < 1 THEN
6         DBMS_OUTPUT.PUT_LINE('Nem létezik ezzel az ID-val a Székhely!');
7     else
8         UPDATE SZEKHELY SET szekhely_irsz = irszmod where szekhely_id = szekid;
9         DBMS_OUTPUT.PUT_LINE('Sikeres update a székhely táblán');
10    end if;
11 end;
12
```

Results Explain Describe Saved SQL History

Procedure created.

0.12 seconds

```
1 BEGIN
2 SZEKHELYirszMod (1,1023);
3 END;
```

Results Explain Describe Saved SQL History

Sikeres update a székhely táblán

Statement processed.

0.02 seconds

rossz esetben pedig:

```
Nem létezik ezzel az ID-val a Székhely!
```

Statement processed.

0.00 seconds

CEG:

a helyes cég id megadásával lehet a fogyasztó idját és a székhely idját is változtatni.

```
CREATE OR REPLACE PROCEDURE CEGszidfidMod (cegid in number, szid in number, fid in number)
```

```
as x number;
```

```
BEGIN
```

```
    SELECT COUNT(*) INTO x FROM CEG WHERE cid = cegid;
```

```
if x < 1 THEN
```

```
    DBMS_OUTPUT.PUT_LINE('Nem létezik ezzel az ID-val a CÉG!');
```

```
else
```

```
    UPDATE CEG SET f_id = fid where cid = cegid ;
```

```
    UPDATE CEG SET sz_id = szid where cid = cegid;
```

```
    DBMS_OUTPUT.PUT_LINE('Sikeres update a cég táblán');
```

```
end if;
```

```
end;
```

```
1  CREATE OR REPLACE PROCEDURE CEGszidfidMod (cegid in number, szid in number, fid in number)
2  as x number;
3  BEGIN
4      SELECT COUNT(*) INTO x FROM CEG WHERE cid = cegid;
5      if x < 1 THEN
6          DBMS_OUTPUT.PUT_LINE('Nem létezik ezzel az ID-val a CÉG!');
7      else
8          UPDATE CEG SET f_id = fid where cid = cegid;
9          UPDATE CEG SET sz_id = szid where cid = cegid;
10         DBMS_OUTPUT.PUT_LINE('Sikeres update a cég táblán');
11     end if;
12 end;
```

Results Explain Describe Saved SQL History

Procedure created.

0.04 seconds

```
1  BEGIN
2  CEGszidfidMod (1,1, 1);
3  END;
```

Results Explain Describe Saved SQL History

Sikeres update a cég táblán

Statement processed.

0.03 seconds

Rossz esetben:

```
Nem létezik ezzel az ID-val a CÉG!
```

Statement processed.

0.01 seconds

- tárolt eljárás adatok törlésére,

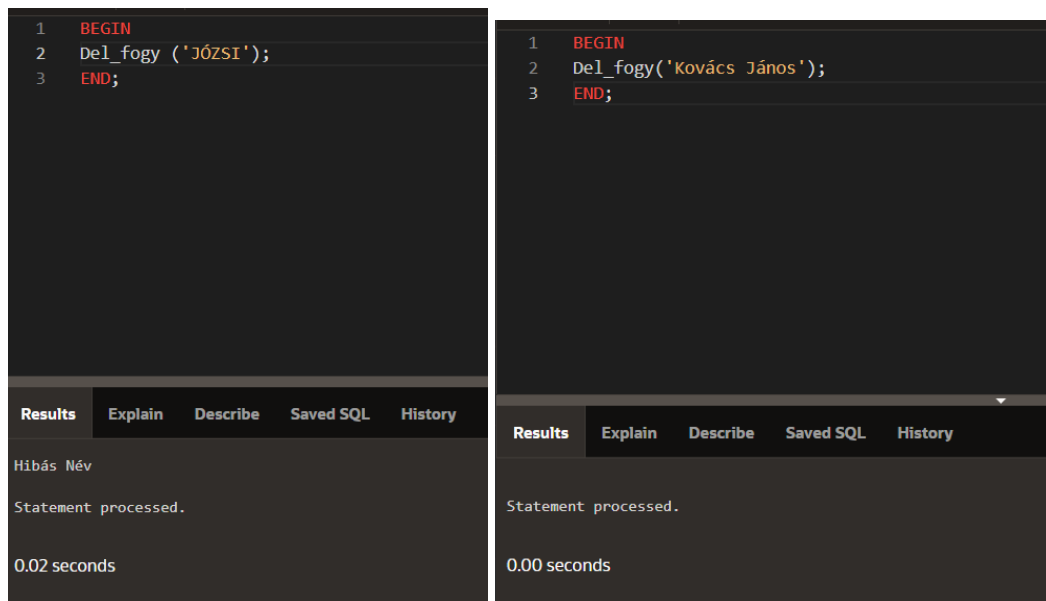
FOGYASZTOK:

töröl egy paraméterként kapott fogyasztói név alapján, ha nincs ilyen akkor hibát ír ki a képernyőre de csak azokat tudja törölni amik nincsenek hivatkozva másik táblában

```
CREATE OR REPLACE PROCEDURE Del_fogy (fogynev in char) as db number;  
  
h_nev exception;  
  
begin  
  
select count(*) into db from FOGYASZTOK where fogyasztó_gyermek = fogynev;  
  
if db=0 then raise h_nev;  
  
else delete from FOGYASZTOK where fogyasztó_gyermek = fogynev;  
  
end if;  
  
exception  
  
when h_nev then dbms_output.put_line('Hibás Név');  
  
end;
```

```
1  CREATE OR REPLACE PROCEDURE Del_fogy (fogynev in char) as db number;  
2  h_nev exception;  
3  begin  
4  select count(*) into db from FOGYASZTOK where fogyasztó_gyermek = fogynev;  
5  if db=0 then raise h_nev;  
6  else delete from FOGYASZTOK where fogyasztó_gyermek = fogynev;  
7  end if;  
8  exception  
9  when h_nev then dbms_output.put_line('Hibás Név');  
10 end;  
11
```

Results	Explain	Describe	Saved SQL	History
Procedure created.				
0.12 seconds				



ezért inkább másik módszert alkalmazok és az id alapján törölök rekordot és ha hivatkozok arra akkor nem lehet törölni és egy speciális hibát adok vissza.

```
CREATE OR REPLACE PROCEDURE Del_fogy (fogyid in number) as db number;
```

```
h_nev exception;
```

```
hiv_sor exception;
```

```
begin
```

```
select count(*) into db from FOGYASZTOK where fogyasztó_id = fogyid;
```

```
if db=0 then raise h_nev;
```

```
else
```

```
    select count(*) into db from CEG where f_id = fogyid;
```

```
    if db > 0 then raise hiv_sor;
```

```
    else delete from FOGYASZTOK where fogyasztó_id = fogyid;
```

```
    end if;
```

```
end if;
```

```
exception
```

```
when h_nev then dbms_output.put_line('Hibás Fogyasztó ID');
```

```
when hiv_sor then dbms_output.put_line('Erre a sorra történt hivatkozás a cég táblában, nem lehet törölni!');
```

```
end;
```

```
1 CREATE OR REPLACE PROCEDURE Del_fogy (fogyid in number) as db number;
2 h_nev exception;
3 hiv_sor exception;
4 begin
5 select count(*) into db from FOGYASZTOK where fogyaszto_id = fogyid;
6 if db=0 then raise h_nev;
7 else
8     select count(*) into db from CEG where f_id = fogyid;
9     if db > 0 then raise hiv_sor;
10    else delete from FOGYASZTOK where fogyaszto_id = fogyid;
11    end if;
12 end if;
13 exception
14 when h_nev then dbms_output.put_line('Hibás Név');
15 when hiv_sor then dbms_output.put_line('Erre a sorra történt hivatkozás a cég táblában, nem lehet törölni!');
16 end;
```

Results Explain Describe Saved SQL History

Procedure created.

0.04 seconds

```
1 BEGIN
2 Del_fogy(3);
3 END;
```

Results Explain Describe Saved SQL History

Erre a sorra történt hivatkozás a cég táblában, nem lehet törölni!

Statement processed.

0.02 seconds

```
1 BEGIN
2 Del_fogy(11);
3 END;
```

Results Explain Describe Saved SQL History

Statement processed.

0.01 seconds

SZEKHELY:

CREATE OR REPLACE PROCEDURE Del_szek (szekid in number) as db number;

h_irsz exception;

hiv_sor exception;

begin

select count(*) into db from SZEKHELY where szekhely_id = szekid;

if db=0 then raise h_irsz;

else

select count(*) into db from CEG where sz_id = szekid;

if db > 0 then raise hiv_sor;

else delete from SZEKHELY where szekhely_id = szekid;

end if;

end if;

exception

when h_irsz then dbms_output.put_line('Hibás Székhely ID');

when hiv_sor then dbms_output.put_line('Erre a sorra történt hivatkozás a cég táblában, nem lehet törölni!');

end;

```
1 CREATE OR REPLACE PROCEDURE Del_szek (szekid in number) as db number;
2 h_irsz exception;
3 hiv_sor exception;
4 begin
5 select count(*) into db from SZEKHELY where szekhely_id = szekid;
6 if db=0 then raise h_irsz;
7 else
8     select count(*) into db from CEG where sz_id = szekid;
9     if db > 0 then raise hiv_sor;
10    else delete from SZEKHELY where szekhely_id = szekid;
11    end if;
12 end if;
13 exception
14 when h_irsz then dbms_output.put_line('Hibás Név');
15 when hiv_sor then dbms_output.put_line('Erre a sorra történt hivatkozás a cég táblában, nem lehet törölni!');
16 end;
```

Results Explain Describe Saved SQL History

Procedure created.

0.12 seconds

```
Hibás Név

Statement processed.

0.01 seconds
```

```
1 BEGIN
2 Del_szek (11);
3 end;
```

Results Explain Describe Saved SQL History

Statement processed.

0.01 seconds

```
Erre a sorra történt hivatkozás a cég táblában, nem lehet törölni!

Statement processed.

0.02 seconds
```

CEG:

CREATE OR REPLACE PROCEDURE Del_ceg (cegid in number) as db number;

h_datum exception;

```

begin

select count(*) into db from CEG where cid = cegid;

if db=0 then raise h_datum;

else

    delete from CEG where cid = cegid;

end if;

exception

when h_datum then dbms_output.put_line('Hibás Cég ID');

end;

```

```

1  CREATE OR REPLACE PROCEDURE Del_ceg (cegid in number) as db number;
2  h_datum exception;
3  begin
4  select count(*) into db from CEG where cid = cegid;
5  if db=0 then raise h_datum;
6  else
7      delete from CEG where cid = cegid;
8  end if;
9  exception
10 when h_datum then dbms_output.put_line('Hibás Dátum');
11 end;
12

```

Results Explain Describe Saved SQL History

Procedure created.

0.05 seconds

<pre> 1 begin 2 Del_ceg ([10]); 3 end; 4 </pre>	<pre> 1 begin 2 Del_ceg ([1]); 3 end; 4 </pre>
Results Explain Describe Saved SQL History	Results Explain Describe Saved SQL History
Hibás Dátum	
Statement processed.	Statement processed.
0.00 seconds	0.01 seconds

- **tárolt függvény adott rekord mezőinek lekérdezésére,**

Kurzor attributomok:

%rowcount – értéket ad egy pillanatra beolvasott rekord számára és ennek a száma.

%found – értéke igaz, ha a fetch utasítás után nem értelmezhető. Az első fetch utasítás kiadása előtt mindig az értéke NULL.

%notfound – értéke igaz, ha az adott fetch utasítás már nem talál a táblában rekordot Csak egy már kiadott fetch utasítás után értelmezhető. Az első fetch utasítás kiadása előtt értéke null.

%isopen – ha a kurzor nyitott akkor az értéke igaz.

SZEKHELY:

Adott várossal rendelkező város irányítószámát növelem egy adott százalékkal. for ciklus segítségével oldom meg ezt a feladatot.

```
create or replace function szekhely_irsznov (varosbe in char, ert in int) return number as
```

```
    cursor cur_a is select * from SZEKHELY where szekhely_varos = varosbe
```

```
    for update of szekhely_irsz;
```

```
    db number :=0;
```

```
begin
```

```
    for cv in cur_a loop
```

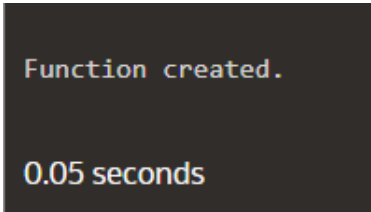
```
        update SZEKHELY set szekhely_irsz = cv.szekhely_irsz*(1+ert/100) where current of  
cur_a;
```

```
        db:= db+1;
```

```
    end loop;
```

```
    return db;
```

```
end;
```



Function created.

0.05 seconds

```
select szekhely_irsznov ('Budapest', 10) from dual;
```



ORA-14551: cannot perform a DML operation inside a query

nem megfelelő ez a futtatás ezért ezt kell használni.

```
1 declare
2 x number(6) := 0;
3 begin
4 x:=szekhely_irsznov ('Budapest', 10);
5 dbms_output.put_line(x);
6 end;
```

Results Explain Describe Saved SQL History

2

Statement processed.

0.02 seconds

10%-al növeltem a budapesti irányítószámokat.

Illetve megszámolom hány ilyen rekord volt amit változtattam.

Kiírom az a Székhely elhelyezkedésének a városát.

implicit kurzor használata: hasonlóképpen csak implicit verzióban.

create or replace procedure szekhely_irsznovim (varosbe in char, ert in int) is db number:=0;

begin

for cv in (select * from SZEKHELY where szekhely_varos = varosbe) loop

dbms_output.put_line('In: ' || cv.szekhely_varos || '-' || cv.szekhely_irsz || '-' || cv.szekhely_id);

update SZEKHELY set szekhely_irsz=cv.szekhely_irsz*(1+ert/100) where SZEKHELY.szekhely_id = cv.szekhely_id;

select * into cv from SZEKHELY where szekhely_id = cv.szekhely_id;

dbms_output.put_line('Out: ' || cv.szekhely_varos || '-' || cv.szekhely_irsz || '-' || cv.szekhely_id);

db:=db+1;

end loop;

dbms_output.put_line('Árváltozás: ' || db);

end;

Procedure created.

0.05 seconds

```
1 begin
2 szekhely_irsznovim ('Miskold', 10);
3 end;
```

Results Explain Describe Saved SQL History

In: Miskolc-3531-3
Out: Miskolc-3884-3
Árváltozás: 1

Statement processed.

FOGYASZTOK:

Kurzor segítségével a FOGYASZTÓK táblában lekérdezem azokat az adatokat, amik, egy bizonyos típusú szakma paraméter alapján kilistázom azokat.

explicit verzióban:

```
create procedure FogyInSzakmaLista3 (szakma in char) is cursor cur_a is select * from FOGYASZTOK
where fogyasztto_foglalkozas = szakma;
```

```
    a cur_a%rowtype;
```

```
begin
```

```
    open cur_a;
```

```
    loop
```

```
        fetch cur_a into a;
```

```
        exit when cur_a%notfound;
```

```
        dbms_output.put_line('Szakma amit a gyermek tanul: ' || a.fogyasztto_foglalkozas ||
'Gyermek neve: ' || a.fogyasztto_gyermek);
```

```
    end loop;
```

```
    close cur_a;
```

```
end;
```


Language SQL Rows 10 Clear Command Find Tables Save Run

A::

```
1 create procedure FogyInSzakmaLista3 (szakma in char) is cursor cur_a is select * from FOGY/
2   a cur_a%rowtype;
3 begin
4   open cur_a;
5   loop
6     fetch cur_a into a;
7     exit when cur_a%notfound;
8     dbms_output.put_line('Szakma amit a gyermek tanul: '|| a.fogyaszto_foglalkozas ||
9   end loop;
10  close cur_a;
11 end;
12
```

Results Explain Describe Saved SQL History

Procedure created.

0.12 seconds

```
1 begin
2 FogyInSzakmaLista3('Programozó');
3 end;
```

Results Explain Describe Saved SQL History

Szakma amit a gyermek tanul: ProgramozóGyermek neve: Tóth Annamária
Szakma amit a gyermek tanul: ProgramozóGyermek neve: Norbert
Szakma amit a gyermek tanul: ProgramozóGyermek neve: Tóth Miklós
Szakma amit a gyermek tanul: ProgramozóGyermek neve: Kovács Norbert
Szakma amit a gyermek tanul: ProgramozóGyermek neve: Kovács Norbert

Statement processed.

CEG:

Kurzor segítségével lekérdezem a CÉG táblában az egy bizonyos index határok között minden adatot.

```
create or replace procedure IndexAlapjanKiir (index1 in int, index2 in int)as
```

```
    cursor cur is select * from CEG where cid >= index1 and cid <= index2;
```

```
    cv cur%rowtype;
```

```
begin
```

```
    open cur;
```

```
    fetch cur into cv;
```

```
    if cur%notfound then
```

```
        dbms_output.put_line('Nincs találat!');
```

```
    else
```

```
        dbms_output.put_line('Cégneve:' || cv.nev || ' Alapítás dátuma: ' || cv.alapitasiev || ' index :'  
|| cv.cid);
```

```
    loop
```

```
        fetch cur into cv;
```

```
        exit when cur%notfound;
```

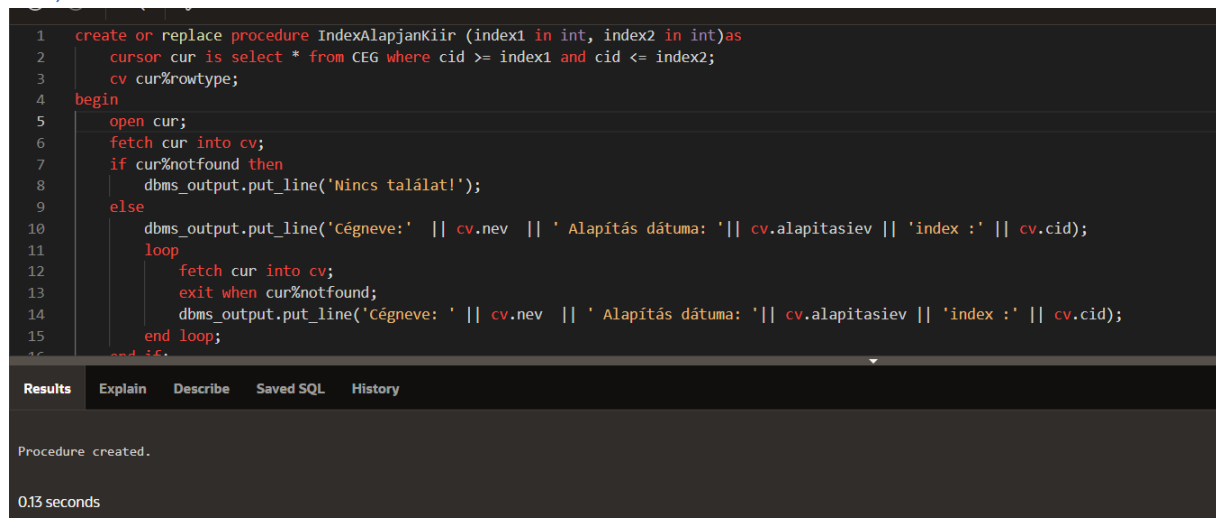
```
        dbms_output.put_line('Cégneve: ' || cv.nev || ' Alapítás dátuma: ' || cv.alapitasiev || ' index  
:' || cv.cid);
```

```
    end loop;
```

```
end if;
```

```
close cur;
```

```
end;
```



```
1 create or replace procedure IndexAlapjanKiir (index1 in int, index2 in int)as
2     cursor cur is select * from CEG where cid >= index1 and cid <= index2;
3     cv cur%rowtype;
4 begin
5     open cur;
6     fetch cur into cv;
7     if cur%notfound then
8         dbms_output.put_line('Nincs találat!');
9     else
10        dbms_output.put_line('Cégneve:' || cv.nev || ' Alapítás dátuma: ' || cv.alapitasiev || 'index :' || cv.cid);
11        loop
12            fetch cur into cv;
13            exit when cur%notfound;
14            dbms_output.put_line('Cégneve: ' || cv.nev || ' Alapítás dátuma: ' || cv.alapitasiev || 'index :' || cv.cid);
15        end loop;
16    end if;
```

Results Explain Describe Saved SQL History

Procedure created.

0.13 seconds

futtatás:

```
begin
```

```
IndexAlapjanKiir(2,8);
```

```
end;
```

```
1  begin
2  IndexAlapjanKiir(2,8);
3  end;
4
5
```

Results	Explain	Describe	Saved SQL	History
Cégneve:WoodenSchool Alapítás dátuma: 01/18/2020 index :2 Cégneve: Debreceni Egyetem Private School Alapítás dátuma: 12/31/2012 index :3 Cégneve: Codecool Alapítás dátuma: 05/09/2010 index :4 Cégneve: Minerliner Alapítás dátuma: 05/20/1998 index :5 Cégneve: Miskolci Egyetem Kohászat Alapítás dátuma: 10/21/1975 index :6 Cégneve: Dumaszház Alapítás dátuma: 06/12/2000 index :7 Statement processed.				

- tárolt függvény adott feltételű rekordok aggregált értékének lekérdezésére,

Cégre:

Kiválasztok egy adott azonosítót és megkeresem rá a megfelelő cég nevét:

```
CREATE OR REPLACE FUNCTION CegIDKeres (cegazonosito IN NUMBER) RETURN CHAR AS x number;
```

```
vi char(100) := 'Nem létező azonosító!';
```

```
y char(40);
```

```
BEGIN
```

```
    SELECT COUNT(*) INTO x FROM CEG WHERE cid = cegazonosito;
```

```
    IF x >= 1 THEN
```

```
        SELECT nev INTO y FROM CEG WHERE cid = cegazonosito;
```

```
        vi := 'A talált cég neve: ' || y;
```

```
    END IF;
```

```
RETURN vi;
```

```
END;
```

Az x változó a bekért azonosítóval rendelkező sorok számát tárolja és az y a cég neve tárolására alkalmazandó.

```
1 CREATE OR REPLACE FUNCTION CegIDKeres (cegazonosito IN NUMBER) RETURN CHAR AS x number;
2 vi char(100) := 'Nem létező azonosító!';
3 y char(40);
4 BEGIN
5     SELECT COUNT(*) INTO x FROM CEG WHERE cid = cegazonosito;
6     IF x >= 1 THEN
7         SELECT nev INTO y FROM CEG WHERE cid = cegazonosito;
8         vi := 'A talált cég neve: ' || y;
9     END IF;
10    RETURN vi;
11 END;
12
```

Results Explain Describe Saved SQL History

Function created.

0.03 seconds

```
1 SELECT CegIDKeres (1) FROM dual;
```

Results Explain Describe Saved SQL History

CEGIDKERES(1)
A talált cég neve: Logiscool

1 rows returned in 0.02 seconds Download

Székhely:

Tárolt program, ami töröl egy paraméterként megadott id alapján az adott székhely rekordot.

Ha nincs ilyen hiba üzenetet kapunk.

```
create or replace procedure del_SZid (idbe in number) as db number;
```

```
h_roszid exception;
```

```
begin
```

```
select count(*) into db from SZEKHELY where szekhely_id = idbe;
```

```
if db=0 then raise h_roszid;
```

```
else
```

```
delete from SZEKHELY where szekhely_id = idbe;
```

```
end if;
```

```
exception
```

```
when h_roszid then dbms_output.put_line('Hibás rendszám');
```

```
end;
```

```
1  create or replace procedure del_SZid (idbe in char) as db number;
2  | h_roszid exception;
3  begin
4  select count(*) into db from SZEKHELY where szekhely_id = idbe;
5  if db=0 then raise h_roszid;
6  else
7  delete from SZEKHELY where szekhely_id = idbe;
8  end if; |
9  exception
10 when h_roszid then dbms_output.put_line('Hibás rendszám');
11 end;
12
13
```

Results Explain Describe Saved SQL History

Procedure created.

0.05 seconds

```
1  begin
2  del_SZid (1);
3  end;|
4
5
```

Results Explain Describe Saved SQL History

Statement processed.

0.02 seconds

másodjára ->

```
1  begin
2  del_SZid (1);
3  end;
4
5
```

Results Explain Describe Saved SQL History

Hibás rendszám

Statement processed.

0.00 seconds

Fogyasztók:

Kiválasztok egy adott azonosítót és megkeresem rá a megfelelő fogyasztó diákigazolványszámát:

```
CREATE OR REPLACE FUNCTION FogyDiakKeres (fogyid IN NUMBER) RETURN CHAR AS x number;
```

```
idvalasz char(100) := 'Nem létező azonosító!';
```

```
ig number(10);
```

BEGIN

SELECT COUNT(*) INTO x FROM FOGYASZTOK WHERE fogyasztó_id = fogyid;

IF x >= 1 THEN

SELECT fogyasztó_diakigazolvány INTO ig FROM FOGYASZTOK WHERE fogyasztó_id =
fogyid;

idvalasz := 'A talált cég neve: ' || ig;

END IF;

RETURN idvalasz;

END;

```
1 CREATE OR REPLACE FUNCTION FogyDiakKeres (fogyid IN NUMBER) RETURN CHAR AS x number;  
2 idvalasz char(100) := 'Nem létező azonosító!';  
3 ig number(10);  
4 BEGIN  
5     SELECT COUNT(*) INTO x FROM FOGYASZTOK WHERE fogyasztó_id = fogyid;  
6     IF x >= 1 THEN  
7         SELECT fogyasztó_diakigazolvány INTO ig FROM FOGYASZTOK WHERE fogyasztó_id = fogyid;  
8         idvalasz := 'A talált cég neve: ' || ig;  
9     END IF;  
10    RETURN idvalasz;  
11 END;  
12
```

Results Explain Describe Saved SQL History

Function created.

0.05 seconds

```
1 select FogyDiakKeres (4) FROM dual;  
2
```

Results Explain Describe Saved SQL History

FOGYDIAKKERES(4)
A talált cég neve: 7417175672

1 rows returned in 0.01 seconds [Download](#)

- tárolt csomag készítése egy tábla funkcióinak összefogására (legyen benne a táblához készített összes tárolt alprogram, de legalább két darab),

Székhely táblán csomag:

Készíték egy **ModSZ** tárolt eljárást, ami módosítja egy paraméterként megadott ID alapján az irányítószámot egy paraméterként megadott értékre.

Majd egy tárolt függvényt hozok létre **KerV** mely visszaadja paraméterként adott székhelyek városát.

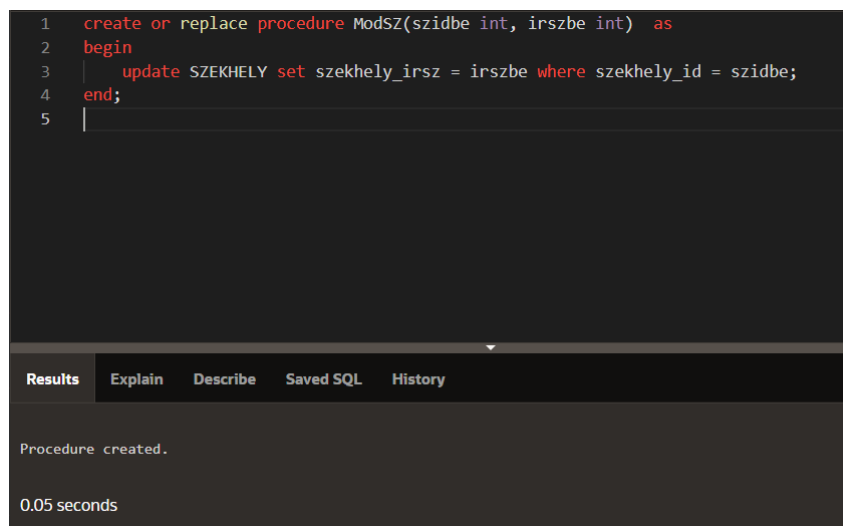
A két alprogramot elhelyezem egy csomagban, aminek a neve **SZEKHELY_cs**.

```
create or replace procedure ModSZ(szidbe int, irszbe int) as
```

```
begin
```

```
    update SZEKHELY set szekhely_irsz = irszbe where szekhely_id = szidbe;
```

```
end;
```



```
1 create or replace procedure ModSZ(szidbe int, irszbe int) as
2 begin
3     update SZEKHELY set szekhely_irsz = irszbe where szekhely_id = szidbe;
4 end;
5
```

The screenshot shows a SQL IDE interface. The top part is a code editor with a dark background and light-colored text. It contains a PL/SQL procedure definition for ModSZ. Below the code editor is a results pane with a dark background and light-colored text. It shows the message "Procedure created." and the execution time "0.05 seconds". The results pane has tabs for "Results", "Explain", "Describe", "Saved SQL", and "History".

```
create or replace function KerV(szidbe in int) return char as szvaros char (25);
```

```
begin
```

```
    select szekhely_varos into szvaros from SZEKHELY where szekhely_id = szidbe;
```

```
    return szvaros;
```

```
end;
```

```

1  create or replace function KerV(szidbe in int) return char as szvaros char (25);
2  begin
3      select szekhely_varos into szvaros from SZEKHELY where szekhely_id = szidbe;
4      return szvaros;
5  end;
6

```

Results Explain Describe Saved SQL History

Function created.

0.04 seconds

create or replace package SZEKHELY_cs as

 procedure ModSZ(szidbe int, irszbe int);

 function KerV(szidbe in int) return char;

end SZEKHELY_cs;

create or replace package body SZEKHELY_cs as

 procedure ModSZ(szidbe int, irszbe int) as

 begin

 update SZEKHELY set szekhely_irsz = irszbe where szekhely_id = szidbe;

 end;

 function KerV(szidbe in int) return char as szvaros char(25);

 begin

 select szekhely_varos into szvaros from SZEKHELY where szekhely_id = szidbe;

 return szvaros;

 end;

end SZEKHELY_cs;


```
1 create or replace package SZEKHELY_cs as
2     procedure ModSZ(szidbe int, irszbe int);
3     function KerV(szidbe in int) return char;
4 end SZEKHELY_cs;
5
```

Results Explain Describe Saved SQL History

Package created.

0.03 seconds

```
1 create or replace package body SZEKHELY_cs as
2     procedure ModSZ(szidbe int, irszbe int) as
3     begin
4         update SZEKHELY set szekhely_irsz = irszbe where szekhely_id = szidbe;
5     end;
6     function KerV(szidbe in int) return char as szvaros char(25);
7     begin
8         select szekhely_varos into szvaros from SZEKHELY where szekhely_id = szidbe;
9         return szvaros;
10    end;
11 end SZEKHELY_cs;
12
```

Results Explain Describe Saved SQL History

Package Body created.

0.04 seconds

futtatás:

begin

SZEKHELY.ModSZ(10,3531);

end;

select SZEKHELY.KerV(10) from dual;

```
1 begin
2 SZEKHELY_cs.ModSZ(10,3531);
3 end;
4 select SZEKHELY_cs.KerV(10) from dual;
5
```

Results Explain Describe Saved SQL History

Statement processed.

0.01 seconds

```
1 begin
2 SZEKHELY_cs.ModSZ(10,3531);
3 end;
4 select SZEKHELY_cs.KerV(10) from dual;
5
```

Results Explain Describe Saved SQL History

SZEKHELY_CS.KERV(10)
Eger

1 rows returned in 0.02 seconds [Download](#)

Fogyasztó táblán csomag:

Készíték egy **ModF** tárolt eljárást, ami módosítja egy paraméterként megadott ID alapján az diákigazolvány számát egy paraméterként megadott értékre.

Majd egy tárolt függvényt hozok létre **KerSZ** mely visszaadja paraméterként adott fogyasztónak a szülő nevét.

A két alprogramot elhelyezem egy csomagban, aminek a neve **FOGYASZTOK_cs**.

create or replace procedure ModF (fidbe int, diakbe int) as

begin

```
update FOGYASZTOK set fogyaszto_diakigazolvany = diakbe where fogyaszto_id = fidbe;

end;
```

```
1 create or replace procedure ModF (fidbe int, diakbe int) as
2 begin
3     update FOGYASZTOK set fogyaszto_diakigazolvany = diakbe where fogyaszto_id = fidbe;
4 end;
5 s|
```

Results Explain Describe Saved SQL History

Procedure created.

0.03 seconds

```
create or replace function KerSZ(fidbe in int) return char as sznev char(30);

begin

    select fogyaszto_szulonev into sznev from FOGYASZTOK where fogyaszto_id = fidbe;

    return sznev;

end;
```

```
1 create or replace function KerSZ(fidbe in int) return char as sznev char(30);|
2 begin
3     select fogyaszto_szulonev into sznev from FOGYASZTOK where fogyaszto_id = fidbe;
4     return sznev;
5 end;
6
```

Results Explain Describe Saved SQL History

Function created.

0.04 seconds

```
create or replace package FOGYASZTOK_cs as

    procedure ModF(fidbe int, diakbe int);

    function KerSZ(fidbe int) return char;

end FOGYASZTOK_cs;
```

create or replace package body FOGYASZTOK_cs as

 procedure ModF(fidbe int, diakbe int) as

 begin

 update FOGYASZTOK set fogyaszto_diakigazolvany = diakbe where fogyaszto_id =
fidbe;

 end;

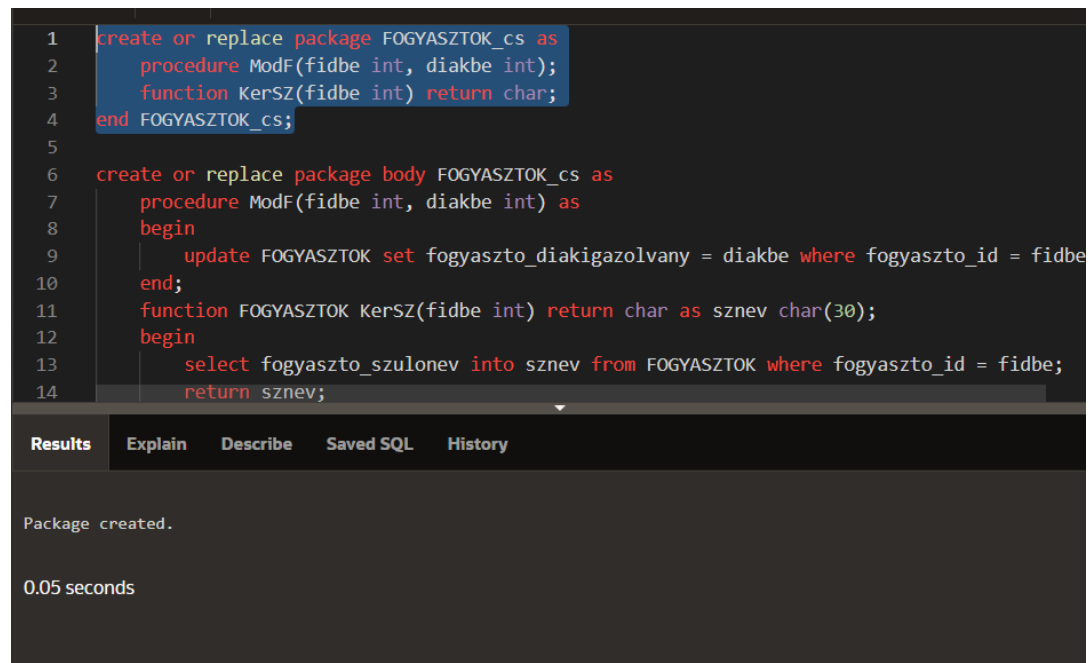
 function KerSZ(fidbe int) return char as sznev char(30);

 begin

 select fogyaszto_szulonev into sznev from FOGYASZTOK where fogyaszto_id = fidbe;

 return sznev;

 end;



The screenshot shows a SQL IDE interface. The top pane displays the SQL code for creating and replacing the package body FOGYASZTOK_cs. The code is as follows:

```
1 create or replace package FOGYASZTOK_cs as
2     procedure ModF(fidbe int, diakbe int);
3     function KerSZ(fidbe int) return char;
4 end FOGYASZTOK_cs;
5
6 create or replace package body FOGYASZTOK_cs as
7     procedure ModF(fidbe int, diakbe int) as
8     begin
9         update FOGYASZTOK set fogyaszto_diakigazolvany = diakbe where fogyaszto_id = fidbe;
10    end;
11    function FOGYASZTOK KerSZ(fidbe int) return char as sznev char(30);
12    begin
13        select fogyaszto_szulonev into sznev from FOGYASZTOK where fogyaszto_id = fidbe;
14        return sznev;
```

The bottom pane shows the execution results. The 'Results' tab is selected, displaying the message 'Package created.' and the execution time '0.05 seconds'.

```
5
6 create or replace package body FOGYASZTOK_cs as
7   procedure ModF(fidbe int, diakbe int) as
8   begin
9     update FOGYASZTOK set fogyaszo_diakigazolvany = diakbe where fogyaszo_id = fidbe;
10  end;
11  function KerSZ(fidbe int) return char as sznev char(30);
12  begin
13    select fogyaszo_szulonev into sznev from FOGYASZTOK where fogyaszo_id = fidbe;
14    return sznev;
15  end;
16 end;
17
```

Results Explain Describe Saved SQL History

Package Body created.

0.04 seconds

```
1 begin
2   FOGYASZTOK_cs.ModF(10,3531);
3 end;
4 select FOGYASZTOK_cs.KerSZ(10) from dual;
5
```

Results Explain Describe Saved SQL History

Statement processed.

0.02 seconds

```
1 begin
2   FOGYASZTOK_cs.ModF(10,3531);
3 end;
4 select FOGYASZTOK_cs.KerSZ(10) from dual;
5
```

Results Explain Describe Saved SQL History

FOGYASZTOK_CS.KERSZ(10)
Tóth Péter

1 rows returned in 0.02 seconds Download

Cég táblán csomag:

Készíték egy **ModD** tárolt eljárást, ami módosítja egy paraméterként megadott ID alapján a dátumot egy paraméterként megadott értékre.

Majd egy tárolt függvényt hozok létre **KerN**, mely visszaadja paraméterként adott cégnek nevét.

A két alprogramot elhelyezem egy csomagban, aminek a neve `CEG_cs`.

```
create or replace procedure ModD(cidbe int, datumbe char) as
```

```
begin
```

```
    update CEG set alapitasiev = To_Date(datumbe,'YYYYMMDD') where cid = cidbe;
```

```
end;
```

```
1  create or replace procedure ModD(cidbe int, datumbe char) as
2  begin
3  |  update CEG set alapitasiev = To_Date(datumbe,'YYYYMMDD') where cid = cidbe;
4  end;
5
```

Results	Explain	Describe	Saved SQL	History
Procedure created.				
0.13 seconds				

```
create or replace function KerN(cidbe in int) return char as cnev char(35);
```

```
begin
```

```
    select nev into cnev from CEG where cid = cidbe;
```

```
    return cnev;
```

```
end;
```

```
1  create or replace function KerN(cidbe in int) return char as cnev char(35);
2  begin
3  |  select nev into cnev from CEG where cid = cidbe;
4  |  return cnev;
5  end;
6
```

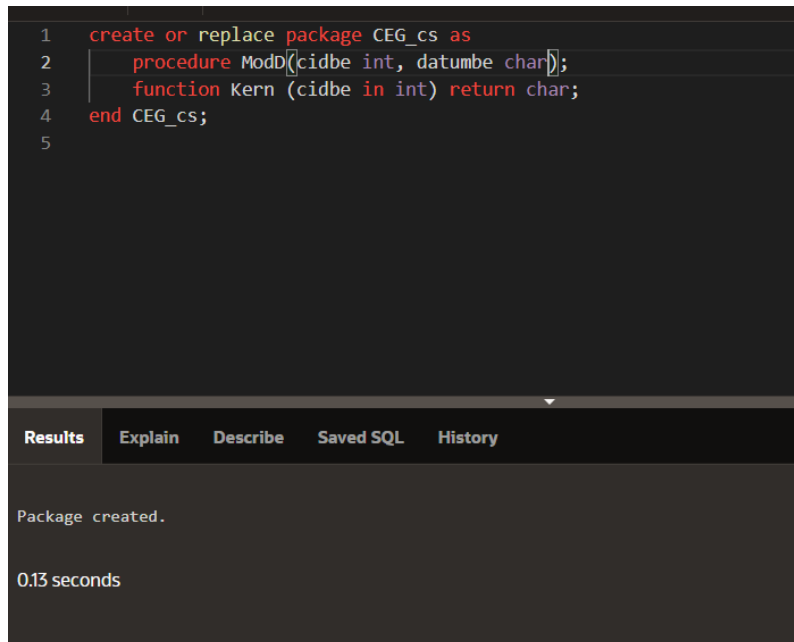
Results	Explain	Describe	Saved SQL	History
Function created.				
0.13 seconds				

create or replace package CEG_cs as

 procedure ModD(cidbe int, datumbe char);

 function Kern (cidbe in int) return char;

end CEG_cs;



The screenshot shows a SQL IDE with a dark theme. The top pane contains the following SQL code:

```
1 create or replace package CEG_cs as
2     procedure ModD(cidbe int, datumbe char);
3     function Kern (cidbe in int) return char;
4 end CEG_cs;
5
```

The bottom pane has a tabbed interface with 'Results' selected. It displays the message 'Package created.' and the execution time '0.13 seconds'.

create or replace package body CEG_cs as

 procedure ModD(cidbe int, datumbe char) as

 begin

 update CEG set alapitasiev = TO_DATE(datumbe,'YYYYMMDD') where cid = cidbe;

 end;

 function KerN(cidbe in int) return char as cnev char (35);

 begin

 select nev into cnev from CEG where cid = cidbe;

 return cnev;

 end;

end CEG_cs;

```
1 create or replace package body CEG_cs as
2   procedure ModD(cidbe int, datumb char) as
3   begin
4     update CEG set alapitasiev = TO_DATE(datumb,'YYYYMMDD') where cid = cidbe;
5   end;
6   function Kern(cidbe in int) return char as cnev char(35);
7   begin
8     select nev into cnev from CEG where cid = cidbe;
9     return cnev;
10  end;
11 end CEG_cs;
12
```

Results Explain Describe Saved SQL History

Package Body created.

0.05 seconds

```
1 begin
2   CEG_cs.ModD(2,'20020205');
3 end;
4 select CEG_cs.KerN(2) from dual;
5
6
```

Results Explain Describe Saved SQL History

Statement processed.

0.02 seconds

begin

CEG_cs.ModD(2,'20020205');

end;

select CEG_cs.KerN(2) from dual;

```
1 begin
2   CEG_cs.ModD(2,'20020205');
3 end;
4 select CEG_cs.KerN(2) from dual;
5
6
```

Results Explain Describe Saved SQL History

CEG_CS.KERN(2)
WoodenSchool

1 rows returned in 0.01 seconds [Download](#)

- **trigger készítése kulcs érték automatikus megadására,**

Trigger: adott esemény bekövetkezésekor automatikusan végrehajtásra kerülő tárolt művelet sor.
Nem kell rá hivatkozni!

SZEKHELY:

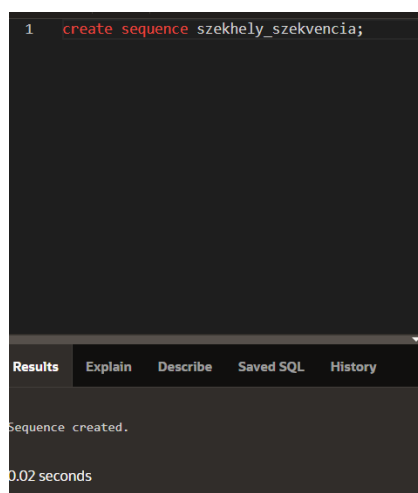
```
create sequence szekhely_szekvencia;
```

```
create or replace trigger szekhely_on_beszur before INSERT ON SZEKHELY FOR EACH ROW
```

```
BEGIN
```

```
    SELECT szekhely_szekvencia.nextval into :new.szekhely_id from dual;
```

```
end;
```

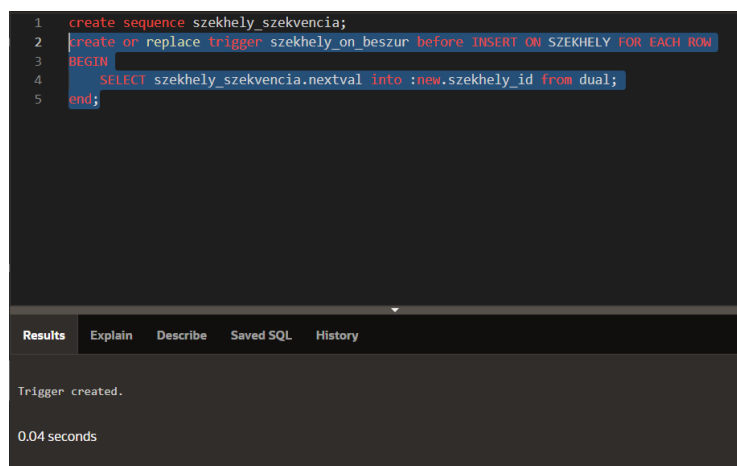


```
1 create sequence szekhely_szekvencia;
```

Results Explain Describe Saved SQL History

Sequence created.

0.02 seconds



```
1 create sequence szekhely_szekvencia;
2 create or replace trigger szekhely_on_beszur before INSERT ON SZEKHELY FOR EACH ROW
3 BEGIN
4     SELECT szekhely_szekvencia.nextval into :new.szekhely_id from dual;
5 end;
```

Results Explain Describe Saved SQL History

Trigger created.

0.04 seconds

FOGYASZTOK:

```
create sequence fogyasztok_szekvencia;
```

```
create or replace trigger fogyasztok_on_beszur before INSERT ON FOGYASZTOK FOR EACH ROW
```

```
BEGIN
```

```
    SELECT fogyasztok_szekvencia.nextval into :new.fogyaszto_id from dual;
```

```
end;
```

```
1 create sequence fogyasztok_szekvencia;
2 create or replace trigger fogyasztok_on_beszur before INSERT ON FOGYASZTOK
3 BEGIN
4     SELECT fogyasztok_szekvencia.nextval into :new.fogyaszto_id from
5 end;
```

Results Explain Describe Saved SQL History

Sequence created.

0.02 seconds

```
1 create sequence fogyasztok_szekvencia;
2 create or replace trigger fogyasztok_on_beszur before INSERT ON FOGYASZTOK FOR EACH ROW
3 BEGIN
4     SELECT fogyasztok_szekvencia.nextval into :new.fogyaszto_id from dual;
5 end;
```

Results Explain Describe Saved SQL History

Trigger created.

0.04 seconds

CEG:

```
create sequence ceg_szekvencia1;
```

```
create or replace trigger ceg_on_beszur before INSERT ON CEG FOR EACH ROW
```

```
BEGIN
```

```
    SELECT ceg_szekvencia1.nextval into :new.cid from dual;
```

```
end;
```

```
1 create sequence ceg_szekvencia1;
2 create or replace trigger ceg_on_beszur before INSERT ON CEG FOR EACH ROW
3 BEGIN
4 |   SELECT ceg_szekvencia1.nextval into :new.cid from dual;
5 end;
6
```

Results Explain Describe Saved SQL History

Sequence created.

0.02 seconds

```
1 create sequence ceg_szekvencia1;
2 create or replace trigger ceg_on_beszur before INSERT ON CEG FOR EACH ROW
3 BEGIN
4 |   SELECT ceg_szekvencia1.nextval into :new.cid from dual;
5 end;
6
```

Results Explain Describe Saved SQL History

Trigger created.

0.03 seconds

- trigger készítése a módosítások kontrollálására.

SZEKHELY:

Ellenőrizzük, hogy megfelelő e a irányítószámnak a nagysága, ha nem akkor nem módosítja.

```
create or replace trigger irszell before UPDATE of szekhely_irsz on SZEKHELY FOR EACH ROW
```

```
declare
```

```
maxirsz number;
```

```
minirsz number;
```

```
BEGIN
```

```
maxirsz :=10000;
```

```
minirsz := 999;
```

```

if (:new.szekhely_irsz not between minirsz and maxirsz ) then

    :new.szekhely_irsz := :old.szekhely_irsz;

end if;

end;

```

```
update SZEKHELY set szekhely_irsz =3531 where szekhely_id =3;
```

```
select * from SZEKHELY where szekhely_id =3;
```

SZEKHELY_ID	SZEKHELY_IRSZ	SZEKHELY_VAROS	SZEKHELY_UTCA	DEPARTMENT_HAZSZAM
3	3531	Miskolc	Bársony út	2

1 rows returned in 0.00 seconds [Download](#)

```

14  update SZEKHELY set szekhely_irsz =353 where szekhely_id =3;
15  select * from SZEKHELY where szekhely_id =3;
16
17

```

Results Explain Describe Saved SQL History

1 row(s) updated.

0.00 seconds

```

8      if (:new.szekhely_irsz not between minirsz and maxirsz ) then
9          :new.szekhely_irsz := :old.szekhely_irsz;
10     end if;
11 end;
12
13
14 update SZEKHELY set szekhely_irsz =353 where szekhely_id =3;
15 select * from SZEKHELY where szekhely_id =3;
16
17

```

Results Explain Describe Saved SQL History

SZEKHELY_ID	SZEKHELY_IRSZ	SZEKHELY_VAROS	SZEKHELY_UTCA	DEPARTMENT_HAZSZAM
3	3531	Miskolc	Bársony út	2

1 rows returned in 0.01 seconds [Download](#)

nem változtatta meg az irányítószámot helytelen adatra.

FOGYASZTOK:

ellenőrizzük, hogy megfelelő e a diákigazolvány nagysága, ha nem akkor nem módosítja.

create or replace trigger diakigell before UPDATE ON FOGYASZTOK FOR EACH ROW

declare

```

maxdiak number;

mindiak number;

BEGIN

maxdiak :=10000000000;

mindiak := 999999999;

    if (:new.fogyaszto_diakigazolvany not between mindiak and maxdiak ) then

        :new.fogyaszto_diakigazolvany := :old.fogyaszto_diakigazolvany;

    end if;

end;

```

```

update FOGYASZTOK set fogyaszto_diakigazolvany =12345678 where fogyaszto_id =2;

select * from fogyasztok where fogyaszto_id =2;

```

```

1  create or replace trigger diakigell before UPDATE ON FOGYASZTOK FOR EACH ROW
2  declare
3  maxdiak number;
4  mindiak number;
5  BEGIN
6  maxdiak :=10000000000;
7  mindiak := 999999999;
8  if (:new.fogyaszto_diakigazolvany not between mindiak and maxdiak ) then
9  :new.fogyaszto_diakigazolvany := :old.fogyaszto_diakigazolvany;
10 end if;
11 end;
12
13
14 update FOGYASZTOK set fogyaszto_diakigazolvany =12345678 where fogyaszto_id =2;

```

Results Explain Describe Saved SQL History

Trigger created.

0.02 seconds

FOGYASZTO_ID	FOGYASZTO_SZULONEV	FOGYASZTO_GYERMEK	FOGYASZTO_FOGLALKOZAS	FOGYASZTO_DIAKIG
2	Tóth József	Tóth Miklós	Programozó	1234567891

```
4  mindiak number;
5  BEGIN
6  maxdiak :=10000000000;
7  mindiak := 999999999;
8  if (:new.fogyaszto_diakigazolvany not between mindiak and maxdiak ) then
9  :new.fogyaszto_diakigazolvany := :old.fogyaszto_diakigazolvany;
10 end if;
11 end;
12
13
14 update FOGYASZTOK set fogyaszto_diakigazolvany =12345678 where fogyaszto_id =2;
15 select * from fogyaszto where fogyaszto_id =2;
16
```

Results Explain Describe Saved SQL History

1 row(s) updated.

0.02 seconds

Language SQL Rows 10 Clear Command Find Tables Save Run

```
4  mindiak number;
5  BEGIN
6  maxdiak :=10000000000;
7  mindiak := 999999999;
8  if (:new.fogyaszto_diakigazolvany not between mindiak and maxdiak ) then
9  :new.fogyaszto_diakigazolvany := :old.fogyaszto_diakigazolvany;
10 end if;
11 end;
12
13
14 update FOGYASZTOK set fogyaszto_diakigazolvany =12345678 where fogyaszto_id =2;
15 select * from fogyaszto where fogyaszto_id =2;
16
```

Results Explain Describe Saved SQL History

FOGYASZTO_ID	FOGYASZTO_SZULONEV	FOGYASZTO_GYERMEK	FOGYASZTO_FOGLALKOZAS	FOGYASZTO_DI
2	Tóth József	Tóth Miklós	Programozó	1234567891

1 rows returned in 0.00 seconds Download

nem változtatta meg a diákigazolványszámát helytelen adatra.

CEG:

Cél: A megváltoztatni kívánt dátum, ha nagyobb, mint a jelenlegi sysdate akkor sajnos nem jön létre a cég.

helyes megoldás:

create or replace trigger CheckEvCeg before update of alapitasiev on CEG for each row

begin

if :new.alapitasiev > sysdate then

:new.alapitasiev := sysdate;

end if;

end;

UPDATE CEG SET alapitasiev = to_date('20330205','YYYYMMDD') where cid =1;

select * from ceg where cid =1;

```
1 create or replace trigger CheckEvCeg before update of alapitasiev on CEG for each row
2 begin
3     if :new.alapitasiev > sysdate then
4         :new.alapitasiev := sysdate;
5     end if;
6 end;
7
8 UPDATE CEG SET alapitasiev = to_date('20330205','YYYYMMDD') where cid =1;
9
10 select * from ceg where cid =1;
11
12 select to_char(sysdate) from dual;
13
14 DROP TRIGGER CheckEvCeg;
```

Results Explain Describe Saved SQL History

Trigger created.

0.03 seconds

eredeti dátum:

CID	NEV	ALAPITASIEV	SZ_ID	F_ID
1	DÁTUMMÓDOSÍTÁSTRIGGERREL	02/05/2013	6	5

1 rows returned in 0.00 seconds Download

majd a rossz dátum használata és kiírás:

```
1 create or replace trigger CheckEvCeg before update of alapitasiev on CEG for each row
2 begin
3     if :new.alapitasiev > sysdate then
4         :new.alapitasiev := sysdate;
5     end if;
6 end;
7
8 UPDATE CEG SET alapitasiev = to_date('20330205','YYYYMMDD') where cid =1;
9
10 select * from ceg where cid =1;
11
12 select to_char(sysdate) from dual;
13
14 DROP TRIGGER CheckEvCeg;
```

Results Explain Describe Saved SQL History

CID	NEV	ALAPITASIEV	SZ_ID	F_ID
1	DÁTUMMÓDOSÍTÁSTRIGGERREL	04/23/2022	6	5

1 rows returned in 0.01 seconds Download

- trigger készítése módosítási események naplózására (többféle esemény naplózása egyetlen triggerrel plusz pontot ér)

napló létrehozása:

create table Naplo (muvelet char(50), datum date, felhasznalo char(20));

szekhely:

create or replace trigger irszellnaplo before UPDATE of szekhely_irsz on SZEKHELY FOR EACH ROW

declare

maxirsz number;

minirsz number;

msg char(50) := concat(:old.szekhely_irsz, ' Módosítva lett a Székhely irányítószáma.');

BEGIN

maxirsz :=10000;

minirsz := 999;

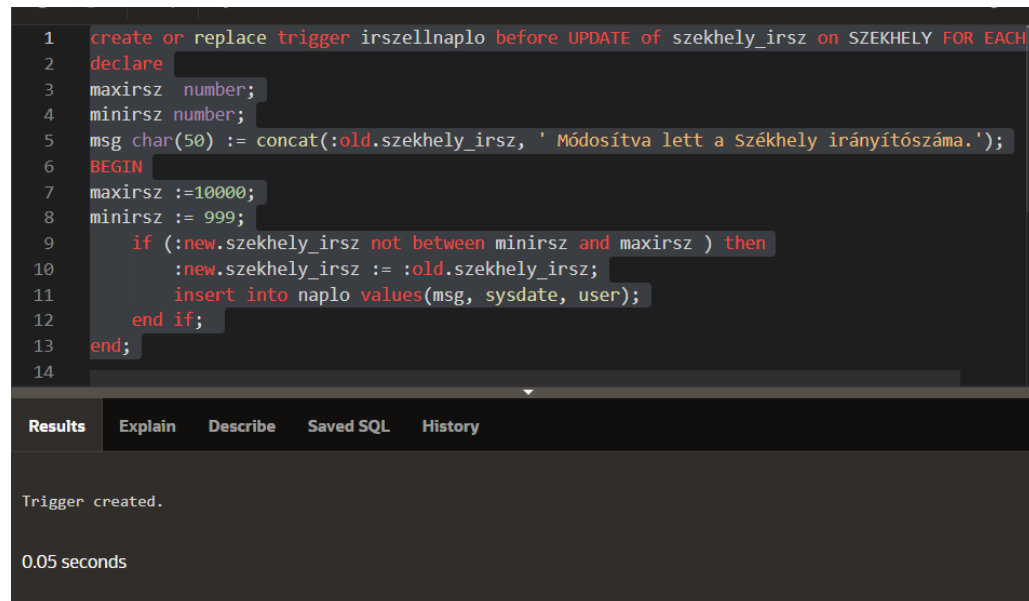
if (:new.szekhely_irsz not between minirsz and maxirsz) then

:new.szekhely_irsz := :old.szekhely_irsz;

insert into naplo values(msg, sysdate, user);

end if;

end;



```
1 create or replace trigger irszellnaplo before UPDATE of szekhely_irsz on SZEKHELY FOR EACH ROW
2 declare
3 maxirsz number;
4 minirsz number;
5 msg char(50) := concat(:old.szekhely_irsz, ' Módosítva lett a Székhely irányítószáma.');
```

The screenshot shows a SQL IDE interface. The top pane displays the SQL code for creating a trigger. The bottom pane shows the results of the execution, indicating that the trigger was created successfully and the execution time was 0.05 seconds.

Results	Explain	Describe	Saved SQL	History
Trigger created.				
0.05 seconds				

naplózás és érték felvétel megpróbálása alacsony szinten:


```

1 update SZEKHELY set szekhely_irsz=128 where szekhely_id=2;
2
3 select * from SZEKHELY WHERE szekhely_id=2;
4
5 select * from naplo;
6
7 create or replace trigger irszellnaplo before UPDATE of szekhely_irsz on SZEKHELY FOR EACH
8 declare
9 maxirsz number;
10 minirsz number;
11 msg char(50) := concat(:old.szekhely_irsz, ' Módosítva lett a Székhely irszáma.');
```

Results Explain Describe Saved SQL History

MUVELET	DATUM	FELHASZNALO
01/01/2022 Módosítva lett a Cég Dátuma.	04/23/2022	APEX_PUBLIC_USER
1128 Módosítva lett a Székhely irszáma.	04/23/2022	APEX_PUBLIC_USER

2 rows returned in 0.01 seconds [Download](#)

fogyasztok:

create or replace trigger diakigellnaplo before UPDATE ON FOGYASZTOK FOR EACH ROW

declare

maxdiak number;

mindiak number;

msg char(50) := concat(:old.fogyaszto_diakigazolvany, ' Módosítva lett a fogyasztó diákja.');

BEGIN

maxdiak :=100000000000;

mindiak := 9999999999;

if (:new.fogyaszto_diakigazolvany not between mindiak and maxdiak) then

:new.fogyaszto_diakigazolvany := :old.fogyaszto_diakigazolvany;

insert into naplo values(msg, sysdate, user);

end if;

end;

```
1 create or replace trigger diakigellnaplo before UPDATE ON FOGYASZTOK FOR EACH ROW
2 declare
3 maxdiak number;
4 mindiak number;
5 msg char(50) := concat(:old.fogyaszto_diakigazolvany, ' Módosítva lett a fogyasztó diákja.')
6 BEGIN
7 maxdiak :=10000000000;
8 mindiak := 999999999;
9 if (:new.fogyaszto_diakigazolvany not between mindiak and maxdiak ) then
10 :new.fogyaszto_diakigazolvany := :old.fogyaszto_diakigazolvany;
11 insert into naplo values(msg, sysdate, user);
12 end if;
13 end;
14
```

Results Explain Describe Saved SQL History

Trigger created.

0.04 seconds

update FOGYASZTOK set fogyaszto_diakigazolvany =12345678 where fogyaszto_id =2;

select * from fogyasztok where fogyaszto_id =2;

frissítés:

```
16 update FOGYASZTOK set fogyaszto_diakigazolvany =12345678 where fogyaszto_id =1;
17 select * from fogyasztok where fogyaszto_id =1;
18
19 select * from naplo;
```

Results Explain Describe Saved SQL History

1 row(s) updated.

0.02 seconds

frissítés megnézése, hogy teljes mértékben nem történt meg:

```
17 select * from fogyasztok where fogyaszto_id =1;
18
19 select * from naplo;
```

Results Explain Describe Saved SQL History

FOGYASZTO_ID	FOGYASZTO_SZULONEV	FOGYASZTO_GYERMEK	FOGYASZTO_FOGLALKOZAS	FOGYASZTO_DIAKIGAZOLVANY
1	Bánfi Orsolya	Kovács Péter	Asztalos	1234567890

Completed in 0.01 seconds Download

napló megtekintése:

```

16 update FOGYASZTOK set fogyasztó_diakigazolvány =12345678 where fogyasztó_id =1;
17 select * from fogyasztok where fogyasztó_id =1;
18
19 select * from naplo;

```

MUVELET	DATUM	FELHASZNALO
01/01/2022 Módosítva lett a Cég Dátuma.	04/23/2022	APEX_PUBLIC_USER
1234567890 Módosítva lett a fogyasztó diákja.	04/23/2022	APEX_PUBLIC_USER
1128 Módosítva lett a Székhely irszáma.	04/23/2022	APEX_PUBLIC_USER

ceg:

Cél: A megváltoztatni kívánt dátum, ha nagyobb, mint a jelenlegi sísdate akkor sajnos nem jön létre a cég és ennek naplózása.

create or replace trigger CheckEvCegNaplo before update of alapitasiev on CEG for each row

declare

msg char(50) := concat(:old.alapitasiev, ' Módosítva lett a Cég Dátuma.');

begin

if :new.alapitasiev > sysdate then

:new.alapitasiev := sysdate;

insert into naplo values(msg, sysdate, user);

end if;

end;

```
13 create or replace trigger CheckEvCegNaplo before update of alapitasiev on CEG for each row
14 declare
15 msg char(50) := concat(:old.alapitasiev, ' Módosítva lett a Cég Dátuma.');
```

```
16 begin
17
18     if :new.alapitasiev > sysdate then
19         :new.alapitasiev := sysdate;
20         insert into naplo values(msg, sysdate, user);
21     end if;
22 end;
23
```

Results Explain Describe Saved SQL History

Trigger created.

0.06 seconds

eredeti:

```
1 create or replace trigger CheckEvCegNaplo before update of alapitasiev on CEG for each row
2 declare
3 msg char(50) := concat(:old.alapitasiev, ' Módosítva lett a Cég Dátuma.');
```

```
4 begin
5
6     if :new.alapitasiev > sysdate then
7         :new.alapitasiev := sysdate;
8         insert into naplo values(msg, sysdate, user);
9     end if;
10 end;
11
12 UPDATE CEG SET alapitasiev = to_date('20220101','YYYYMMDD') where cid=3;
13
14 select * from CEG where cid =3;
15
16 select * from naplo;
```

Results Explain Describe Saved SQL History

CID	NEV	ALAPITASIEV	SZ_ID	F_ID
3	Debreceni Egyetem Private School	01/01/2022	3	3

naplózáshoz megváltoztatjuk a dátumot:

```
11
12 UPDATE CEG SET alapitasiev = to_date('20330101','YYYYMMDD') where cid=3;
13
14 select * from CEG where cid =3;
15
16 select * from naplo;
```

Results Explain Describe Saved SQL History

CID	NEV	ALAPITASIEV	SZ_ID	F_ID
3	Debreceni Egyetem Private School	04/23/2022	3	3

naplóba került adat:

```
9      end if;
10     end;
11
12     UPDATE CEG SET alapitasiev = to_date('20330101','YYYYMMDD') where cid=3;
13
14     select * from CEG where cid =3;
15
16     select * from naplo;
```

Results Explain Describe Saved SQL History

MUVELET	DATUM	FELHASZNALO
01/01/2022 Módosítva lett a Cég Dátuma.	04/23/2022	APEX_PUBLIC_USER

1 rows returned in 0.01 seconds [Download](#)

A PL/SQL-ben megvalósítandó elemek:

- normál parancsok használata,
- implicit kurzor,
- explicit kurzor,
- kivételkezelés (exception használata).