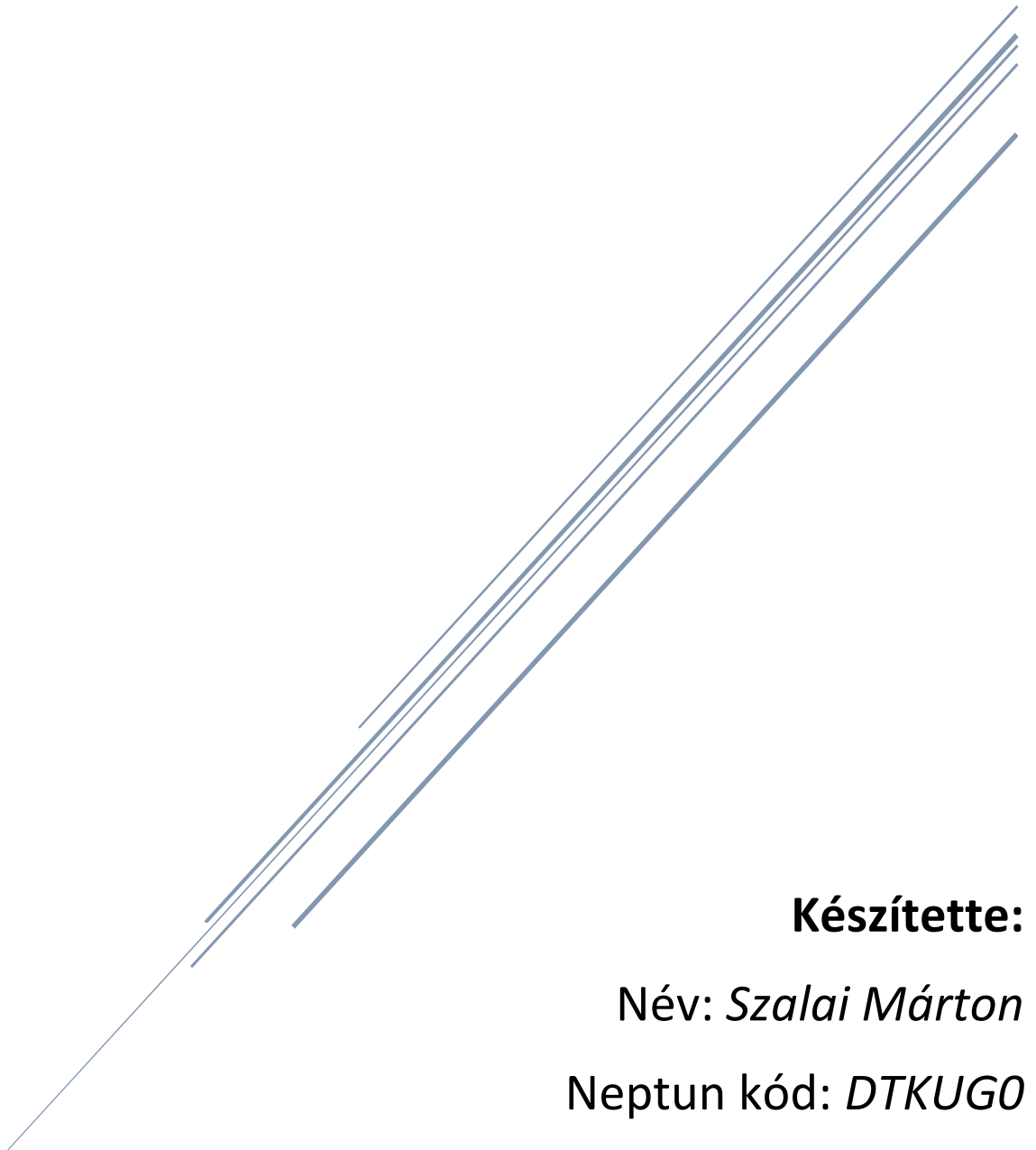


ADATBÁZIS RENDSZEREK 2

FÉLÉVES EGYÉNI FELADAT

PL/SQL feladatrész



Készítette:

Név: *Szalai Márton*

Neptun kód: *DTKUG0*

Gyakorlatvezető neve: *Szűcs Miklós*

Gyakorlat időpontja: *Csütörtök 12-14*

Témakor ismertetése

Beadandómban egy *receptek* és az azt elkészítő *éttermek* adatbázis szintű rendszerezését valósítottam meg.

A *receptek* táblában egy recept, vagyis egy étel nevét, annak jellegét (előétel, főétel, desszert), az elkészítési idejét, illetve a recept utolsó módosításának a dátumát tároltam le. Az *éttermek* táblában az étterem nevét, telefonszámát, első nyitásának dátumát valamint a specialitásokat tároltam le, vagyis, hogy egy étterem inkább előételeket, főételeket vagy desszerteket szokott készíteni.

Illetve az események naplózásához létrehoztam egy *Naplo* nevű táblát.

A tábla szerkezetének ismertetése

1. Receptek tábla létrehozása

```
CREATE TABLE Receptek (
  sorszam int primary key not null,
  nev varchar2(30),
  jelleg varchar2(30),
  elkeszitesi_ido int,
  utolso_modositas DATE
);
```

18 desc receptek

19

Results

Explain

Describe

Saved SQL

History

Object Type

TABLE ?

Object

RECEPTEK ?

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
RECEPTEK	SORSZAM	NUMBER	22	-	0	1	-	-	-
	NEV	VARCHAR2	30	-	-	-	✓	-	-
	JELLEG	VARCHAR2	30	-	-	-	✓	-	-
	ELKESZITESI_IDO	NUMBER	22	-	0	-	✓	-	-
	UTOLSO_MODOSITAS	DATE	7	-	-	-	✓	-	-

2. Étterem tábla létrehozása

```
CREATE TABLE Etterem (
  azonosito int primary key not null,
  nev varchar2(30),
  telefonszam varchar2(30),
  elso_nyitas DATE,
  specialitas varchar2(50),
  FOREIGN KEY(azonosito) REFERENCES Receptek(sorszam)
);
```

119
120 desc Etterem
121
122

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ETTEREM	AZONOSITO	NUMBER	22	-	0	1	-	-	-
	NEV	VARCHAR2	30	-	-	-	✓	-	-
	TELEFONSZAM	VARCHAR2	30	-	-	-	✓	-	-
	ELSO_NYITAS	DATE	7	-	-	-	✓	-	-
	SPECIALITAS	VARCHAR2	50	-	-	-	✓	-	-

3. Napló létrehozása

```
create table Naplo (
  id number primary key,
  muvelet varchar2(30),
  tabla varchar2(20),
  operation_date timestamp default current_timestamp,
  data varchar2(100)
);
```

23 desc Naplo;

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
NAPLO	ID	NUMBER	22	-	-	1	-	-	-
	MUVELET	VARCHAR2	30	-	-	-	✓	-	-
	TABLA	VARCHAR2	20	-	-	-	✓	-	-
	OPERATION_DATE	TIMESTAMP(6)	11	-	6	-	✓	current_timestamp	-
	DATA	VARCHAR2	100	-	-	-	✓	-	-

A táblát feltöltő SQL parancsok

1. Receptek tábla feltöltése

```
begin
  INSERT INTO Receptek VALUES (1, 'Tökfőzelék', 'főétel', 65,
                                '05/13/2019');
  INSERT INTO Receptek VALUES (2, 'Ropogós pizstrángok', 'előétel', 32,
                                '09/09/2001');
  INSERT INTO Receptek VALUES (3, 'Kagyló', 'előétel', 12, '03/18/2021');
  INSERT INTO Receptek VALUES (4, 'Frankfurti leves', 'főétel', 50,
                                '12/31/2018');
  INSERT INTO Receptek VALUES (5, 'Túrógombóc', 'desszert', 30,
                                '04/01/2013');
  INSERT INTO Receptek VALUES (6, 'Vegyes gyümölcsleves', 'előétel', 45,
                                '08/13/2015');
  INSERT INTO Receptek VALUES (7, 'Virslisaláta', 'előétel', 30,
                                '09/23/2011');
  INSERT INTO Receptek VALUES (8, 'Cigánypecsenye', 'főétel', 30,
                                '05/13/2010');
  INSERT INTO Receptek VALUES (9, 'Lekváros palacsinta', 'desszert', 17,
                                '04/28/2003');
  INSERT INTO Receptek VALUES (10, 'Rizskoch', 'desszert', 60,
                                '11/11/2007');
end;
```

2. Étterem tábla feltöltése

```
begin
  INSERT INTO Etterem VALUES (1, 'EztEddMeg',
                                '30-545-3234', '02/02/2020', 'desszertek');
  INSERT INTO Etterem VALUES (2, 'Itt a helyed',
                                '49-999-999', '01/01/2022', 'főételek, előételek');
  INSERT INTO Etterem VALUES (3, 'Jól laksz',
                                '20-510-4207', '09/01/2001', 'főételek, előételek');
  INSERT INTO Etterem VALUES (4, 'Éhes vagy?',
                                '70-380-9800', '05/30/2006', 'előételek');
  INSERT INTO Etterem VALUES (5, 'Svéd asztal',
                                '30-596-1114', '03/04/2012', 'desszertek, előételek');
  INSERT INTO Etterem VALUES (6, 'All inclusive',
                                '43-052-414', '08/16/2015', 'főétel');
  INSERT INTO Etterem VALUES (7, 'Étel & ital',
                                '53-456-924', '11/01/2015', 'főételek, desszertek');
  INSERT INTO Etterem VALUES (8, 'Elvitelre minden',
                                '30-111-1112', '05/07/2013', 'előételek');
  INSERT INTO Etterem VALUES (9, 'Éhség csillapító',
                                '70-822-9632', '12/31/2017', 'főételek');
  INSERT INTO Etterem VALUES (10, 'Egyél itt',
                                '49-300-580', '03/09/2003', 'desszertek');
end;
```

Paraméterekkel működő tárolt eljárás adatok felvitelére

Ez az eljárás biztosítja, hogy adatokat vihessük fel a táblákba.

```
create or replace procedure InsertReceptek (innev in varchar2,  
      injelleg in varchar2, inel_ido in varchar2, inut_mod in varchar2) as  
begin  
    INSERT INTO Receptek (nev, jelleg, elkeszitesi_ido, utolso_modositas)  
    VALUES (innev, injelleg, inel_ido, inut_mod);  
end;
```

A létrehozás képernyőképe:

```
54 create or replace procedure InsertReceptek (innev in varchar2, injelleg in varchar2, inel_ido in varchar2, inut_mod in varchar2) as  
55 begin  
56     INSERT INTO Receptek (nev, jelleg, elkeszitesi_ido, utolso_modositas) VALUES (innev, injelleg, inel_ido, inut_mod);  
57 end;  
58
```

Results	Explain	Describe	Saved SQL	History
Procedure created.				
0.01 seconds				

A futtatás képernyőképe:

```
--  
63 begin  
64     InsertReceptek('XXXXXXXXX', 'XXXXX', 32, '01/01/2000');  
65 end;  
66
```

Results	Explain	Describe	Saved SQL	History
Statement processed.				
0.07 seconds				

```
create or replace procedure InsertEtterem (innev in varchar2,  
    intel in varchar2, inelso_nyit in varchar2, inspec in varchar2) as  
begin  
    INSERT INTO Etterem (nev, telefonszam, elso_nyitas, specialitas)  
    VALUES (innev, intel, inelso_nyit, inspec);  
end;
```

A létrehozás képernyőképe:

```
--  
56 create or replace procedure InsertReceptek (innev in varchar2, injelleg in varchar2, inel_ido in varchar2, inut_mod in varchar2) as  
57 begin  
58     INSERT INTO Receptek (nev, jelleg, elkeszitesi_ido, utolso_modositas) VALUES (innev, injelleg, inel_ido, inut_mod);  
59 end;  
60  
--
```

Results Explain Describe Saved SQL History

Procedure created.

0.01 seconds

A futtatás képernyőképe:

```
68  
69 begin  
70     InsertEtterem('Üdv itt', '30-999-9999', '01/01/2000', 'Minden is');  
71 end;
```

Results Explain Describe Saved SQL History

Statement processed.

0.01 seconds

Paraméterekkel működő tárolt eljárás egy rekord módosítására

Ez az eljárás arra szolgál, hogy a **receptek** táblában, egy rekord adott sorszámu mezőjét módosíthassuk, implicit kurzorral kiegészítve.

```
create or replace procedure ModReceptek (inszoveg in varchar2,
    inmelyikmezo in number, insorszam in number) as
    found number;
    rownum number;
begin
    /* check hogy van-e ilyen sorszámmal recept */
    select count(*) into found from Receptek WHERE sorszam = insorszam;

    if sql%notfound then
        dbms_output.put_line('A receptek tábla üres');
    end if;

    if found <> 1 then
        dbms_output.put_line('Nincs ilyen sorszámu rekord!');
    end if;

    if inmelyikmezo < 1 OR inmelyikmezo > 4 then
        dbms_output.put_line('Hibás mezősorszám!');
    end if;

    if inmelyikmezo = 1 then /* nev */
        UPDATE Receptek SET nev = inszoveg WHERE sorszam = insorszam;
    elsif inmelyikmezo = 2 then /* jelleg */
        UPDATE Receptek SET jelleg = inszoveg WHERE sorszam = insorszam;
    elsif inmelyikmezo = 3 then /* elkészítési idő */
        UPDATE Receptek SET elkeszitesi_ido = TO_NUMBER(inszoveg)
                                WHERE sorszam = insorszam;
    elsif inmelyikmezo = 4 then /* utolsó módosítás */
        UPDATE Receptek SET utolso_modositas = inszoveg
                                WHERE sorszam = insorszam;
    end if;

    if sql%found then
        rownum := sql%rowcount;
        dbms_output.put_line(rownum || ' receptek módosítva');
    end if;

end;
```

A létrehozás képernyőképe:

```
1 create or replace procedure ModReceptek (inszoveg in varchar2, inmelyikmezo in number, insorszam in number) as
2     found number;
3     rownum number;
4 begin
5
6     /* check hogy van-e ilyen sorszámmal recept */
7     select count(*) into found from Receptek WHERE sorszam = insorszam;
8
9     if sql%notfound then
10         dbms_output.put_line('A receptek tábla üres');
11     end if;
12
13     if found <> 1 then
14         dbms_output.put_line('Nincs ilyen sorszámu rekord!');
15     end if;
16
17     if inmelyikmezo < 1 OR inmelyikmezo > 4 then
18         dbms_output.put_line('Hibás mezősorszám!');
19     end if;
20
21     if inmelyikmezo = 1 then /* nev */
22         UPDATE Receptek SET nev = inszoveg WHERE sorszam = insorszam;
23     elsif inmelyikmezo = 2 then /* jelleg */
24         UPDATE Receptek SET jelleg = inszoveg WHERE sorszam = insorszam;
25     elsif inmelyikmezo = 3 then /* elkészítési idő */
26         UPDATE Receptek SET elkészitesi_ido = TO_NUMBER(inszoveg) WHERE sorszam = insorszam;
27     elsif inmelyikmezo = 4 then /* utolsó módosítás */
28         UPDATE Receptek SET utolso_modositas = inszoveg WHERE sorszam = insorszam;
29     end if;
30
```

Results Explain Describe Saved SQL History

Procedure created.

0.01seconds

A futtatás képernyőképe:

```
38 begin
39     ModReceptek('Tejfölös raguleves', 1, 1);
40 end;
```

Results Explain Describe Saved SQL History

1 recept módosítva!

Statement processed.

0.01 seconds

Természetesen ugyanezzel a módszerrel módosítható rekord az **étterem** táblában is egy másik eljárással:

```
create or replace procedure ModEtterem (inszoveg in varchar2,
    inmelyikmezo in number, insorszam in number) as
    found number;
    rownum number;
begin

    if sql%notfound then
        dbms_output.put_line('A receptek tábla üres');
    end if;

    /* check hogy van-e ilyen sorszámmal étterem */
    select count(*) into found from Etterem WHERE azonosito = insorszam;

    if found <> 1 then
        dbms_output.put_line('Nincs ilyen sorszámú rekord!');
    end if;

    if inmelyikmezo < 1 OR inmelyikmezo > 4 then
        dbms_output.put_line('Hibás mezősorszám!');
    end if;

    if inmelyikmezo = 1 then /* nev */
        UPDATE Etterem SET nev = inszoveg WHERE azonosito = insorszam;
    elsif inmelyikmezo = 2 then /* jelleg */
        UPDATE Etterem SET telefonszam = inszoveg WHERE azonosito = insorszam;
    elsif inmelyikmezo = 3 then /* elkészítési idő */
        UPDATE Etterem SET elso_nyitas = inszoveg WHERE azonosito = insorszam;
    elsif inmelyikmezo = 4 then /* utolsó módosítás */
        UPDATE Etterem SET specialitas = inszoveg WHERE azonosito = insorszam;
    end if;

    if sql%found then
        rownum := sql%rowcount;
        dbms_output.put_line(rownum || ' étterem módosítva!');
    end if;

end;
```

A létrehozás képernyőképe:

```

1  create or replace procedure ModEtterem (inszoveg in varchar2, inmelyikmezo in number, insorszam in number) as
2      found number;
3      rownum number;
4  begin
5
6      if sql%notfound then
7          dbms_output.put_line('A receptek tábla üres');
8      end if;
9
10     /* check hogy van-e ilyen sorszámmal étterem */
11     select count(*) into found from Etterem WHERE azonosito = insorszam;
12
13     if found <> 1 then
14         dbms_output.put_line('Nincs ilyen sorszámu rekord!');
15     end if;
16
17     if inmelyikmezo < 1 OR inmelyikmezo > 4 then
18         dbms_output.put_line('Hibás mezősorszám!');
19     end if;
20
21     if inmelyikmezo = 1 then /* nev */
22         UPDATE Etterem SET nev = inszoveg WHERE azonosito = insorszam;
23     elsif inmelyikmezo = 2 then /* jelleg */
24         UPDATE Etterem SET telefonszam = inszoveg WHERE azonosito = insorszam;
25     elsif inmelyikmezo = 3 then /* elkészítési idő */
26         UPDATE Etterem SET elso_nyitas = inszoveg WHERE azonosito = insorszam;
27     elsif inmelyikmezo = 4 then /* utolsó módosítás */
28         UPDATE Etterem SET specialitas = inszoveg WHERE azonosito = insorszam;
29     end if;
30
Results Explain Describe Saved SQL History

```

Procedure created.

0.03 seconds

A futtatás képernyőképe:

```

38  begin
39      ModEtterem('STOP éhség!', 1, 1);
40  end;

```

Results	Explain	Describe	Saved SQL	History
1 étterem módosítva!				
Statement processed.				
0.00 seconds				

Paraméteres tárolt eljárás adatsor törlésére

Ez az eljárás arra szolgál, hogy egy adott sorszám/azonosítószám intervallumon belül rekordokat törölhessünk az adatbázisból. Az eljárás meghívásakor elég a táblák nevének az első két betűjét megadni, valamint az intervallumot.

```
create or replace procedure DeleteRecord(intablanev in varchar2,
    inmin in number, inmax in number) as
    cursor recept_adatok is select * from Receptek where sorszam
        between inmin and inmax;
    cursor etterem_adatok is select * from Etterem where azonosito
        between inmin and inmax;
    data varchar2(70);
    recepteknum number := instr(intablanev, 're',1);
    etteremnum number := instr(intablanev, 'et',1);
begin
    if recepteknum > 0 then
        FOR a IN recept_adatok LOOP
            delete from Receptek where sorszam = a.sorszam;
            data:= a.sorszam||' - '||a.nev||' - '||a.jelleg||' - '||
                a.elkeszitesi_ido||' - '||a.utolso_modositas;
            dbms_output.put_line('Törölt adatsor: '||data);
        END LOOP;
    end if;

    if etteremnum > 0 then
        FOR b IN etterem_adatok LOOP
            delete from Etterem where azonosito = b.azonosito;
            data:=b.azonosito||' - '||b.nev||' - '||b.telefonszam||' - '||
                b.elso_nyitas||' - '||b.specialitas;
            dbms_output.put_line('Törölt adatsor: '||data);
        END LOOP;
    end if;

end;
```

A létrehozás képernyőképe:

```

1  create or replace procedure DeleteRecord(intablanev in varchar2, inmin in number, inmax in number) as
2      cursor recept_adatok is select * from Receptek where sorszam between inmin and inmax;
3      cursor etterem_adatok is select * from Etterem where azonosito between inmin and inmax;
4      data varchar2(70);
5      recepteknum number:= instr(intablanev, 're',1);
6      etteremnum number := instr(intablanev, 'et',1);
7  begin
8      if recepteknum > 0 then
9          FOR a IN recept_adatok LOOP
10             delete from Receptek where sorszam = a.sorszam;
11             data:= a.sorszam||' - '||a.nev||' - '||a.jelleg||' - '||a.elkeszitesi_ido||' - '||a.utolso_modositas;
12             dbms_output.put_line('Törölt adatsor: '||data);
13          END LOOP;
14      end if;
15
16      if etteremnum > 0 then
17          FOR b IN etterem_adatok LOOP
18             delete from Etterem where azonosito = b.azonosito;
19             data:=b.azonosito||' - '||b.nev||' - '||b.telefonszam||' - '||b.elso_nyitas||' - '||b.specialitas;
20             dbms_output.put_line('Törölt adatsor: '||data);
21          END LOOP;
22      end if;
23  end;
24
25

```

Results Explain Describe Saved SQL History

Procedure created.

0.03 seconds

A futtatás képernyőképe:

```

26  begin
27      DeleteRecord('rec', 8, 10);
28  end;

```

Results Explain Describe Saved SQL History

Törölt adatsor: 8 - Cigánypecsenye - főétel - 30 - 05/13/2010
Törölt adatsor: 9 - Lekváros palacsinta - desszert - 17 - 04/28/2003
Törölt adatsor: 10 - Rizskoch - desszert - 60 - 11/11/2007

Statement processed.

0.01 seconds

Paraméteres tárolt függvény adott rekord mezőjének lekérdezésére

Ez az függvény egy adott dátum intervallumon belül visszaadja, hogy hány recept vagy étterem esik bele a tartományba.

```
create or replace function InDateRangeOutCount (intablanev in varchar2,
    inmin in DATE, inmax in DATE) return number as
    amount number;
    recepteknum number:= instr(intablanev, 're',1);
    etteremnum number := instr(intablanev, 'et',1);
begin
    if recepteknum > 0 then
        select count(*) into amount from Receptek where utolso_modositas
            between inmin and inmax;
        end if;

        if etteremnum > 0 then
            select count(*) into amount from Etterem where elso_nyitas
                between inmin and inmax;
            end if;

        return amount;
    end;
```

A létrehozás képernyőképe:

```
26 create or replace function InDateRangeOutCount (intablanev in varchar2, inmin in DATE, inmax in DATE) return number as
27     amount number;
28     recepteknum number:= instr(intablanev, 're',1);
29     etteremnum number := instr(intablanev, 'et',1);
30 begin
31     if recepteknum > 0 then
32         select count(*) into amount from Receptek where utolso_modositas between inmin and inmax;
33     end if;
34
35     if etteremnum > 0 then
36         select count(*) into amount from Etterem where elso_nyitas between inmin and inmax;
37     end if;
38
39     return amount;
40 end;
```

Results Explain Describe Saved SQL History

Function created.

0.10 seconds

A futtatás képernyőképe:

```
42 select InDateRangeOutCount('recept', '01/01/2010', sysdate) from dual;
```

Results Explain Describe Saved SQL History

INDATERANGEOUTCOUNT('RECEPT','01/01/2010',SYSDATE)
8

1 rows returned in 0.01 seconds Download

Paraméteres tárolt függvény adott feltételű rekordok aggregált értékének lekérdezésére

Ez a függvény a receptek elkészítési idejének átlagát adja meg.

```
create or replace function AvgElIdo return number as
    average number;
begin
    select avg(elkeszitesi_ido) into average from Receptek;
    return average;
end;
```

A létrehozás képernyőképe:

```
14 create or replace function AvgElIdo return number as
15     average number;
16 begin
17     select avg(elkeszitesi_ido) into average from Receptek;
18     return average;
19 end;
```

Results Explain Describe Saved SQL History

Function created.

0.01 seconds

A futtatás képernyőképe:

```
21 select AvgElIdo from dual;
```

Results Explain Describe Saved SQL History

AVGELIDO

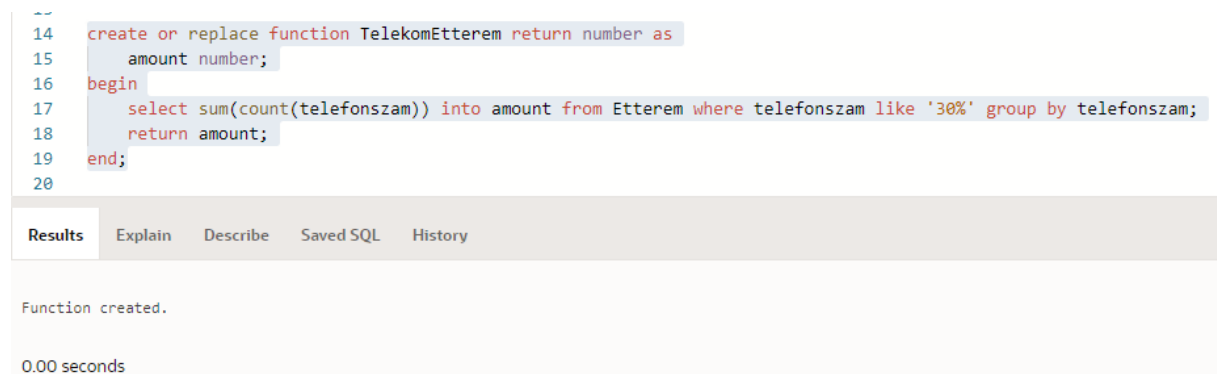
46.44444444444444444444444444444444444

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

Ez a függvény pedig a Telekomos telefonszámmal rendelkező éttermek darabszámát adja vissza.

```
create or replace function TelekomEtterem return number as
    amount number;
begin
    select sum(count(telefonszam)) into amount from Etterem where telefonszam
like '30%' group by telefonszam;
    return amount;
end;
```

A létrehozás képernyőképe:



The screenshot shows the SQL Developer interface. The top pane contains the PL/SQL code for creating the function. The bottom pane shows the results of the execution.

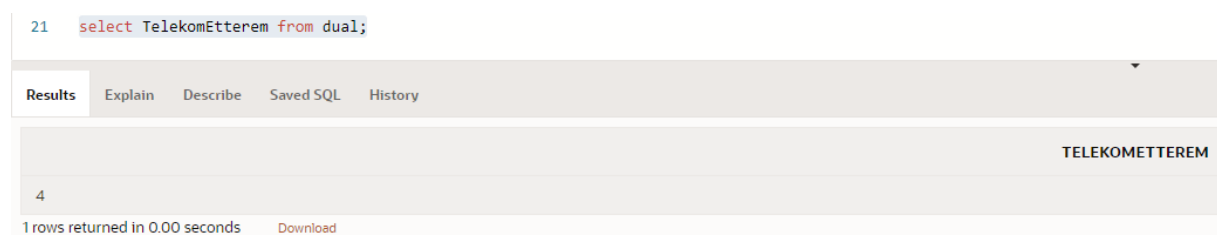
```
14 create or replace function TelekomEtterem return number as
15     amount number;
16 begin
17     select sum(count(telefonszam)) into amount from Etterem where telefonszam like '30%' group by telefonszam;
18     return amount;
19 end;
20
```

Results | Explain | Describe | Saved SQL | History

Function created.

0.00 seconds

A futtatás képernyőképe:



The screenshot shows the SQL Developer interface. The top pane contains the SQL query to call the function. The bottom pane shows the results of the execution.

```
21 select TelekomEtterem from dual;
```

Results | Explain | Describe | Saved SQL | History

TELEKOMETTEREM
4

1 rows returned in 0.00 seconds Download

Tárolt csomag a tábla funkcióinak összefoglalására

A csomagba a teljesség igénye nélkül csak a beszűrő, módosító, és törlő parancsokat illesztettem be.

A létrehozás képernyőképei:

```

169 create or replace package csomag as
170     procedure InsertReceptek (innev in varchar2, injelleg in varchar2, inel_ido in varchar2, inut_mod in varchar2);
171     procedure InsertEtterem (innev in varchar2, intel in varchar2, inelso_nyit in varchar2, inspec in varchar2);
172     procedure ModReceptek (inszoveg in varchar2, inmelyikmezo in number, insorszam in number);
173     procedure ModEtterem (inszoveg in varchar2, inmelyikmezo in number, insorszam in number);
174     procedure DeleteRecord(intablanev in varchar2, inmin in number, inmax in number);
175 end;
176
Results Explain Describe Saved SQL History

Package created.

0.00 seconds

177 create or replace package body csomag as
178     procedure InsertReceptek (innev in varchar2, injelleg in varchar2, inel_ido in varchar2, inut_mod in varchar2) as
179     begin
180         INSERT INTO Receptek (nev, jelleg, elkeszitesi_ido, utolso_modositas) VALUES (innev, injelleg, inel_ido, inut_mod);
181     end;
182     procedure InsertEtterem ( innev in varchar2, intel in varchar2, inelso_nyit in varchar2, inspec in varchar2) as
183     begin
184         INSERT INTO Etterem (nev, telefonszam, elso_nyitas, specialitas) VALUES (innev, intel, inelso_nyit, inspec);
185     end;
186     procedure ModReceptek (inszoveg in varchar2, inmelyikmezo in number, insorszam in number) as
187     found number;
188     rownum number;
189     begin
190         /* check hogy van-e ilyen sorszámmal recept */
191         select count(*) into found from Receptek WHERE sorszam = insorszam;
192
193         if sql%notfound then
194             dbms_output.put_line('A receptek tábla üres');
195         end if;
196
197         if found < 1 then

```

A futtatás eredménye:

```

287 begin
288     csomag.InsertReceptek('Valami finomság', 'főétel', 32, '01/01/2000');
289     csomag.InsertEtterem('Üdv itt', '30-999-9999', '01/01/2000', 'Minden is');
290     csomag.ModReceptek('Hamburger', 1, 1);
291     csomag.ModEtterem('Eszel és eszel', 1, 15);
292     csomag.DeleteRecord('ett', 19, 19);
293 end;

Results Explain Describe Saved SQL History

1 receptek módosítva
1 étterem módosítva!
Törölt adatsor: 19 - Üdv itt - 30-999-9999 - 01/01/2000 - Minden is

Statement processed.

0.01 seconds

```


Trigger módosítási események naplózására

Ez a trigger a receptek táblában lévő valamelyik rekord módosítása és törlése esetén, illetve új sor beszúrásánál is egy bejegyzést hoz létre a Naplo táblában, amely tartalmazza a művelet nevét és idejét, valamint magát a rekordot is az esetleges visszaállítás érdekében. Ahhoz, hogy ez működjön, szükség volt egy EventLogger eljárásra, amelyet a trigger meghívhat, és rögzítheti az eseményt.

```
create or replace procedure EventLogger (inmuvelet in varchar2,  
    intabla in varchar2, indata in varchar2) as  
    createid number;  
begin  
    select max(id) into createid from naplo;  
    if createid > 0 then  
        createid := createid + 1;  
    else  
        createid := 1;  
    end if;  
  
    insert into naplo (id, muvelet, tabla, data)  
        values (createid, inmuvelet, intabla, indata);  
end;
```

A létrehozás képernyőképe:

```
33 create or replace procedure EventLogger (inmuvelet in varchar2, intabla in varchar2, indata in varchar2) as  
34     createid number;  
35 begin  
36     select max(id) into createid from naplo;  
37     if createid > 0 then  
38         createid := createid + 1;  
39     else  
40         createid := 1;  
41     end if;  
42  
43     insert into naplo (id, muvelet, tabla, data) values (createid, inmuvelet, intabla, indata);  
44 end;
```

Results	Explain	Describe	Saved SQL	History
Procedure created.				
0.01 seconds				

Maga a trigger:

```
create or replace trigger ReceptekTrigger
  after insert or update or delete on Receptek for each row
declare
  data varchar2(100);
begin
  if inserting then
    data:=new.sorszam||'#'||:new.nev||'#'||:new.jelleg||'#'||:new.elkeszi
tes_i_ido||'#'||:new.utolso_modositas;
    EventLogger('Insert', 'Receptek', data);
  elsif deleting then
    data:=old.sorszam||'#'||:old.nev||'#'||:old.jelleg||'#'||:old.elkeszi
tes_i_ido||'#'||:old.utolso_modositas;
    EventLogger('Delete', 'Receptek', data);
  else
    data:=old.sorszam||'#'||:old.nev||'#'||:old.jelleg||'#'||:old.elkeszi
tes_i_ido||'#'||:old.utolso_modositas;
    EventLogger('Update', 'Receptek', data);
  end if;
end;
```

A létrehozás képernyőképe:

```
48 create or replace trigger ReceptekTrigger
49   after insert or update or delete on Receptek for each row
50 declare
51   data varchar2(100);
52 begin
53   if inserting then
54     data:=new.sorszam||'#'||:new.nev||'#'||:new.jelleg||'#'||:new.elkeszitesi_ido||'#'||:new.utolso_modositas;
55     EventLogger('Insert', 'Receptek', data);
56   elsif deleting then
57     data:=old.sorszam||'#'||:old.nev||'#'||:old.jelleg||'#'||:old.elkeszitesi_ido||'#'||:old.utolso_modositas;
58     EventLogger('Delete', 'Receptek', data);
59   else
60     data:=old.sorszam||'#'||:old.nev||'#'||:old.jelleg||'#'||:old.elkeszitesi_ido||'#'||:old.utolso_modositas;
61     EventLogger('Update', 'Receptek', data);
62   end if;
63 end;
```

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

Trigger created.

0.02 seconds

Ugyanennek a mintájára létrehoztam egy triggeret az Étterem táblához is.

A létrehozás képernyőképe:

```

65 create or replace trigger EtteremTrigger
66   after insert or update or delete on Etterem for each row
67 declare
68   data varchar2(100);
69 begin
70   if inserting then
71     data:=new.azonosito||'#'||:new.nev||'#'||:new.telefonszam||'#'||:new.else_nyitas||'#'||:new.specialitas;
72     EventLogger('Insert', 'Étterem', data);
73   elsif deleting then
74     data:=old.azonosito||'#'||:old.nev||'#'||:old.telefonszam||'#'||:old.else_nyitas||'#'||:old.specialitas;
75     EventLogger('Delete', 'Étterem', data);
76   else
77     data:=old.azonosito||'#'||:old.nev||'#'||:old.telefonszam||'#'||:old.else_nyitas||'#'||:old.specialitas;
78     EventLogger('Update', 'Étterem', data);
79   end if;
80 end;

```

Results Explain Describe Saved SQL History

Trigger created.

0.03 seconds

```

create or replace trigger EtteremTrigger
  after insert or update or delete on Etterem for each row
declare
  data varchar2(100);
begin
  if inserting then
    data:=:new.azonosito||'#'||:new.nev||'#'||:new.telefonszam||'#'||:new.
else_nyitas||'#'||:new.specialitas;
    EventLogger('Insert', 'Étterem', data);
  elsif deleting then
    data:=:old.azonosito||'#'||:old.nev||'#'||:old.telefonszam||'#'||:old.
else_nyitas||'#'||:old.specialitas;
    EventLogger('Delete', 'Étterem', data);
  else
    data:=:old.azonosito||'#'||:old.nev||'#'||:old.telefonszam||'#'||:old.
else_nyitas||'#'||:old.specialitas;
    EventLogger('Update', 'Étterem', data);
  end if;
end;

```

A futtatások eredménye:

```

129 select * from naplo
130
131

```

ID	MUVELET	TABLA	OPERATION_DATE	DATA
2	Insert	Receptek	16-APR-21 10.12.27.887348 PM	6#Vegyes gyümölcsleves#előétel#45#08/13/2015
1	Insert	Receptek	16-APR-21 10.12.14.712213 PM	2#Ropogós pizstrángok#előétel#32#09/09/2001
4	Insert	Receptek	16-APR-21 10.12.37.729611 PM	9#Lekváros palacsinta#desszert#17#04/28/2003
3	Insert	Receptek	16-APR-21 10.12.33.961784 PM	8#Cigánypecsenye#főétel#30#05/13/2010
8	Update	Étterem	16-APR-21 10.14.04.858893 PM	12#Eat here almost free#20-530-6541#09/03/2011#főételek, desszertek
5	Insert	Receptek	16-APR-21 10.12.40.606630 PM	10#Ritzkoch#desszert#60#11/11/2007
6	Delete	Receptek	16-APR-21 10.13.04.742279 PM	3#Kagyló#előétel#12#03/18/2021
7	Insert	Étterem	16-APR-21 10.13.36.464849 PM	12#Eat here almost free#20-530-6541#09/03/2011#főételek, desszertek

8 rows returned in 0.01 seconds Download

szalaimarci99@gmail.com me_ab2 en Copyright © 1999, 2020, Oracle and/or its affiliates. All rights reserved.

Lehetőségünk van arra, hogy a napló adott sorszámú rekordja alapján, törölt rekordokat állítsunk vissza, amelyet a RestoreReceptek és RestoreEtterem eljárásokkal tehetünk meg

```

create or replace procedure RestoreReceptek (inid in number) as
    getsorszam number;
    getnev varchar2(30);
    getjelleg varchar2(30);
    getelido number;
    getutmod date;
    getdata varchar2(100);
begin
    select data into getdata from naplo where id = inid;

    getsorszam:= substr(getdata, 1, instr(getdata, '#', 1, 1)-1);
    getnev:= substr(getdata, instr(getdata, '#', 1, 1)+1,
        instr(getdata, '#', 1, 2) - instr(getdata, '#', 1, 1) -1);
    getjelleg:= substr(getdata, instr(getdata, '#', 1, 2)+1,
        instr(getdata, '#', 1, 3) - instr(getdata, '#', 1, 2) -1);
    getelido:= substr(getdata, instr(getdata, '#', 1, 3)+1,
        instr(getdata, '#', 1, 4) - instr(getdata, '#', 1, 3) -1);
    getutmod := substr(getdata, instr(getdata, '#', 1, 4)+1);

    insert into receptek values (getsorszam, getnev, getjelleg, getelido, getu
tmod);
    EventLogger('Restore', 'Receptek', getdata);
end;

```

A létrehozás képernyőképe:

```

141 create or replace procedure RestoreReceptek (inid in number) as
142     getsorszam number;
143     getnev varchar2(30);
144     getjelleg varchar2(30);
145     getelido number;
146     getutmod date;
147     getdata varchar2(100);
148 begin
149     select data into getdata from naplo where id = inid;
150
151     getsorszam:= substr(getdata, 1, instr(getdata, '#', 1, 1)-1);
152     getnev:= substr(getdata, instr(getdata, '#', 1, 1)+1, instr(getdata, '#', 1, 2) - instr(getdata, '#', 1, 1) -1);
153     getjelleg:= substr(getdata, instr(getdata, '#', 1, 2)+1, instr(getdata, '#', 1, 3) - instr(getdata, '#', 1, 2) -1);
154     getelido:= substr(getdata, instr(getdata, '#', 1, 3)+1, instr(getdata, '#', 1, 4) - instr(getdata, '#', 1, 3) -1);
155     getutmod := substr(getdata, instr(getdata, '#', 1, 4)+1);
156
157     insert into receptek values (getsorszam, getnev, getjelleg, getelido, getutmod);
158     EventLogger('Restore', 'Receptek', getdata);
159 end;
160

```

Results Explain Describe Saved SQL History

Procedure created.

0.01 seconds

Ha listázzuk a napló tábla tartalmát, akkor láthatjuk, hogy mely rekordok kerültek törlésre. Az eljárás meghívásakor elég a napló egyedi azonosítóját megadnunk, és a törölt adatok visszakerülnek a helyükre.

A napló tábla tartalma:

```

163 select * from naplo
164
165 begin
166     RestoreReceptek(6);
167 end;
168

```

Results Explain Describe Saved SQL History

ID	MUVELET	TABLA	OPERATION_DATE	DATA
2	Insert	Receptek	16-APR-21 10.12.27.887348 PM	6#Vegyes gyümölcsleves#előétel#45#08/13/2015
1	Insert	Receptek	16-APR-21 10.12.14.712213 PM	2#Ropogós pizstrángok#előétel#32#09/09/2001
4	Insert	Receptek	16-APR-21 10.12.37.729611 PM	9#Lekváros palacsinta#desszert#17#04/28/2003
3	Insert	Receptek	16-APR-21 10.12.33.961784 PM	8#Cigánypecsenye#főétel#30#05/13/2010
8	Update	Étterem	16-APR-21 10.14.04.858893 PM	12#Eat here almost free#20-530-6541#09/03/2011#főételek, desszertek
5	Insert	Receptek	16-APR-21 10.12.40.606630 PM	10#Rizskoch#desszert#60#11/11/2007
6	Delete	Receptek	16-APR-21 10.13.04.742279 PM	3#Kagyló#előétel#12#03/18/2021
7	Insert	Étterem	16-APR-21 10.13.36.464849 PM	12#Eat here almost free#20-530-6541#09/03/2011#főételek, desszertek

8 rows returned in 0.00 seconds Download

szalaimarc199@gmail.com me_eb2 en Copyright © 1999, 2020, Oracle and/or its affiliates. All rights reserved.

A futtatás eredménye:

The screenshot shows the SQL Developer interface. The top pane displays the following PL/SQL code:

```

165 begin
166     RestoreReceptek(6);
167 end;
168

```

The bottom pane shows the execution results:

Statement processed.

0.01 seconds

Below this, a table titled "select * from receptek;" is displayed. The table has the following columns: SORSZAM, NEV, JELLEG, ELKESZITESI_IDO, and UTOLSO_MODOSITAS. The data row shows:

SORSZAM	NEV	JELLEG	ELKESZITESI_IDO	UTOLSO_MODOSITAS
6	Vegyes gyümölcsleves	előétel	45	08/13/2015

Ugyanennek a mintájára elkészítettem az étterem visszaállító eljárást is

```

create or replace procedure RestoreEtterem (inid in number) as
    getazon number;
    getnev varchar2(30);
    gettel varchar2(30);
    getenyit date;
    getspec varchar2(50);
    getdata varchar2(100);
begin
    select data into getdata from naplo where id = inid;

    getazon:= substr(getdata, 1, instr(getdata, '#', 1, 1)-1);
    getnev:= substr(getdata, instr(getdata, '#', 1, 1)+1,
instr(getdata, '#', 1, 2) - instr(getdata, '#', 1, 1) -1);
    gettel:= substr(getdata, instr(getdata, '#', 1, 2)+1,
instr(getdata, '#', 1, 3) - instr(getdata, '#', 1, 2) -1);
    getenyit:= substr(getdata, instr(getdata, '#', 1, 3)+1,
instr(getdata, '#', 1, 4) - instr(getdata, '#', 1, 3) -1);
    getspec := substr(getdata, instr(getdata, '#', 1, 4)+1);

    insert into etterem values (getazon, getnev, gettel, getenyit, getspec);
    EventLogger('Restore', 'Étterem', getdata);
end;

```

A létrehozás képernyőképe:

```

161 create or replace procedure RestoreEtterem (inid in number) as
162     getazon number;
163     getnev varchar2(30);
164     gettel varchar2(30);
165     getenyit date;
166     getspec varchar2(50);
167     getdata varchar2(100);
168 begin
169     select data into getdata from naplo where id = inid;
170
171     getazon:= substr(getdata, 1, instr(getdata, '#', 1, 1)-1);
172     getnev:= substr(getdata, instr(getdata, '#', 1, 1)+1, instr(getdata, '#', 1, 2) - instr(getdata, '#', 1, 1) -1);
173     gettel:= substr(getdata, instr(getdata, '#', 1, 2)+1, instr(getdata, '#', 1, 3) - instr(getdata, '#', 1, 2) -1);
174     getenyit:= substr(getdata, instr(getdata, '#', 1, 3)+1, instr(getdata, '#', 1, 4) - instr(getdata, '#', 1, 3) -1);
175     getspec := substr(getdata, instr(getdata, '#', 1, 4)+1);
176
177     insert into etterem values (getazon, getnev, gettel, getenyit, getspec);
178     EventLogger('Restore', 'Étterem', getdata);
179 end;
180

```

Results Explain Describe Saved SQL History

Procedure created.

0.03 seconds

A napló tábla tartalma:

```

194 select * from naplo;
195

```

Results Explain Describe Saved SQL History

ID	MUVELET	TABLA	OPERATION_DATE	DATA
2	Insert	Receptek	16-APR-21 10:12:27.887348 PM	6#Vegyes gyümölcsleves#előétel#45#08/13/2015
10	Delete	Receptek	16-APR-21 10:23:27.178592 PM	6#Vegyes gyümölcsleves#előétel#45#08/13/2015
1	Insert	Receptek	16-APR-21 10:12:14.712213 PM	2#Ropogós pizstrángok#előétel#32#09/09/2001
4	Insert	Receptek	16-APR-21 10:12:37.729611 PM	9#Lekváros palacsinta#desszert#17#04/28/2003
11	Delete	Étterem	16-APR-21 10:29:43.317321 PM	7#Étel & ital#53-456-924#11/01/2015#főételek, desszertek

A futtatás eredménye:

```

196 begin
197     RestoreEtterem(11);
198 end;
199
200

```

Results Explain Describe Saved SQL History

Statement processed.

0.00 seconds

Trigger kulcs érték automatikus megadására

Ezek a triggerek minden insert előtt lefutnak és egyedi azonosítót biztosítanak a rekordoknak beszúrás előtt.

```
create sequence seq1;  
create sequence seq2;
```

```
create trigger SorszamGen before insert on Receptek for each row  
begin  
    :new.sorszam := seq1.nextval;  
end;
```

```
create trigger AzonGen before insert on Etterem for each row  
declare  
begin  
    :new.azonosito := seq2.nextval;  
end;
```

A létrehozás képernyőképei:

```
16 create trigger SorszamGen before insert on Receptek for each row  
17 begin  
18     :new.sorszam := seq1.nextval;  
19 end;  
20
```

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

Trigger created.

0.05 seconds

```
21 create trigger AzonGen before insert on Etterem for each row  
22 declare  
23 begin  
24     :new.azonosito := seq2.nextval;  
25 end;
```

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

Trigger created.

0.06 seconds

A futtatás képernyőképe:

```
65 begin
66   InsertEtterem('otodik', '30-999-9999', '01/01/2000', 'Minden is');
67 end;
68
69 select * from etterem;
70
71
```

Results	Explain	Describe	Saved SQL	History	
AZONOSITO		NEV	TELEFONSZAM	ELSO_NYITAS	SPECIALITAS
3		XXXXXXXX	30-999-9999	01/01/2000	Minden is
5		otodik	30-999-9999	01/01/2000	Minden is

Trigger a módosítások kontrollálására

Ez a trigger a receptek tábla jelleg mezőjét ellenőrzi, hogy csak előételt, főételt, vagy desszertet lehessen megadni, esetleg ezeknek a valamilyen formában elírt változatát. Egyéb esetben hibaüzenetet kapunk.

```
create or replace trigger JellegCheck before insert on Receptek for each row
begin
    if lower(:new.jelleg) in ('eloetel', 'előétel') THEN
        :new.jelleg := 'Előétel';
    elsif lower(:new.jelleg) in ('foetel', 'főétel') THEN
        :new.jelleg := 'Főétel';
    elsif lower(:new.jelleg) in ('desszert', 'dessz', 'deszert', 'desz') THEN
        :new.jelleg := 'Desszert';
    else
        dbms_output.put_line('A jelleg csak előétel, főétel
                               vagy desszert lehet!');
    end if;
end;
```

A létrehozás képernyőképe:

```
100 create or replace trigger JellegCheck before insert on Receptek for each row
101 begin
102     if lower(:new.jelleg) in ('eloetel', 'előétel') THEN
103         :new.jelleg := 'Előétel';
104     elsif lower(:new.jelleg) in ('foetel', 'főétel') THEN
105         :new.jelleg := 'Főétel';
106     elsif lower(:new.jelleg) in ('desszert', 'dessz', 'deszert', 'desz') THEN
107         :new.jelleg := 'Desszert';
108     else
109         dbms_output.put_line('A jelleg csak előétel, főétel vagy desszert lehet!');
110     end if;
111 end;
```

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

Trigger created.

0.02 seconds

A futtatás képernyőképe:

```
65 begin
66     InsertReceptek('XXXXXXXX', 'nincs ötletem', 32, '01/01/2000');
67 end;
```

Results	Explain	Describe	Saved SQL	History
A jelleg csak előétel, főétel vagy desszert lehet!				
Statement processed.				
0.05 seconds				

Tárolt metódust implicit kurzor kezelés bemutatására

Ez az eljárás arra szolgál, hogy a **receptek** táblában, egy rekord adott sorszámu mezőjét módosíthassuk, implicit kurzorral kiegészítve.

```
create or replace procedure ModReceptek (inszoveg in varchar2, inmelyikmezo in
number, insorszam in number) as
    found number;
    rownum number;
begin
    /* check hogy van-e ilyen sorszámmal recept */
    select count(*) into found from Receptek WHERE sorszam = insorszam;

    if sql%notfound then
        dbms_output.put_line('A receptek tábla üres');
    end if;

    if found <> 1 then
        dbms_output.put_line('Nincs ilyen sorszámu rekord!');
    end if;

    if inmelyikmezo < 1 OR inmelyikmezo > 4 then
        dbms_output.put_line('Hibás mezősorszám!');
    end if;

    if inmelyikmezo = 1 then /* nev */
        UPDATE Receptek SET nev = inszoveg WHERE sorszam = insorszam;
    elsif inmelyikmezo = 2 then /* jelleg */
        UPDATE Receptek SET jelleg = inszoveg WHERE sorszam = insorszam;
    elsif inmelyikmezo = 3 then /* elkészítési idő */
        UPDATE Receptek SET elkeszitesi_ido = TO_NUMBER(inszoveg)
                                WHERE sorszam = insorszam;
    elsif inmelyikmezo = 4 then /* utolsó módosítás */
        UPDATE Receptek SET utolso_modositas = inszoveg
                                WHERE sorszam = insorszam;
    end if;

    if sql%found then
        rownum := sql%rowcount;
        dbms_output.put_line(rownum || ' receptek módosítva');
    end if;

end;
```

A létrehozás képernyőképe:

```
1 create or replace procedure ModReceptek (inszoveg in varchar2, inmelyikmezo in number, insorszam in number) as
2     found number;
3     rownum number;
4 begin
5
6     /* check hogy van-e ilyen sorszámmal recept */
7     select count(*) into found from Receptek WHERE sorszam = insorszam;
8
9     if sql%notfound then
10         dbms_output.put_line('A receptek tábla üres');
11     end if;
12
13     if found <> 1 then
14         dbms_output.put_line('Nincs ilyen sorszámu rekord!');
15     end if;
16
17     if inmelyikmezo < 1 OR inmelyikmezo > 4 then
18         dbms_output.put_line('Hibás mezősorszám!');
19     end if;
20
21     if inmelyikmezo = 1 then /* nev */
22         UPDATE Receptek SET nev = inszoveg WHERE sorszam = insorszam;
23     elsif inmelyikmezo = 2 then /* jelleg */
24         UPDATE Receptek SET jelleg = inszoveg WHERE sorszam = insorszam;
25     elsif inmelyikmezo = 3 then /* elkészítési idő */
26         UPDATE Receptek SET elkészitesi_ido = TO_NUMBER(inszoveg) WHERE sorszam = insorszam;
27     elsif inmelyikmezo = 4 then /* utolsó módosítás */
28         UPDATE Receptek SET utolso_modositas = inszoveg WHERE sorszam = insorszam;
29     end if;
30
```

Results Explain Describe Saved SQL History

Procedure created.

0.01seconds

A futtatás képernyőképe:

```
38 begin
39     ModReceptek('Tejfölös raguleves', 1, 1);
40 end;
```

Results Explain Describe Saved SQL History

1 recept módosítva!

Statement processed.

0.01 seconds

Természetesen ugyanezzel a módszerrel módosítható rekord az **étterem** táblában is egy másik eljárással:

```
create or replace procedure ModEtterem (inszoveg in varchar2, inmelyikmezo in
number, insorszam in number) as
    found number;
    rownum number;
begin

    if sql%notfound then
        dbms_output.put_line('A receptek tábla üres');
    end if;

    /* check hogy van-e ilyen sorszámmal étterem */
    select count(*) into found from Etterem WHERE azonosito = insorszam;

    if found <> 1 then
        dbms_output.put_line('Nincs ilyen sorszámú rekord!');
    end if;

    if inmelyikmezo < 1 OR inmelyikmezo > 4 then
        dbms_output.put_line('Hibás mezősorszám!');
    end if;

    if inmelyikmezo = 1 then /* nev */
        UPDATE Etterem SET nev = inszoveg WHERE azonosito = insorszam;
    elsif inmelyikmezo = 2 then /* jelleg */
        UPDATE Etterem SET telefonszam = inszoveg WHERE azonosito = insorszam;
    elsif inmelyikmezo = 3 then /* elkészítési idő */
        UPDATE Etterem SET elso_nyitas = inszoveg WHERE azonosito = insorszam;
    elsif inmelyikmezo = 4 then /* utolsó módosítás */
        UPDATE Etterem SET specialitas = inszoveg WHERE azonosito = insorszam;
    end if;

    if sql%found then
        rownum := sql%rowcount;
        dbms_output.put_line(rownum || ' étterem módosítva!');
    end if;

end;
```

A létrehozás képernyőképe:

```

1  create or replace procedure ModEtterem (inszoveg in varchar2, inmelyikmezo in number, insorszam in number) as
2      found number;
3      rownum number;
4  begin
5
6      if sql%notfound then
7          dbms_output.put_line('A receptek tábla üres');
8      end if;
9
10     /* check hogy van-e ilyen sorszámmal étterem */
11     select count(*) into found from Etterem WHERE azonosito = insorszam;
12
13     if found <> 1 then
14         dbms_output.put_line('Nincs ilyen sorszámu rekord!');
15     end if;
16
17     if inmelyikmezo < 1 OR inmelyikmezo > 4 then
18         dbms_output.put_line('Hibás mezősorszám!');
19     end if;
20
21     if inmelyikmezo = 1 then /* nev */
22         UPDATE Etterem SET nev = inszoveg WHERE azonosito = insorszam;
23     elsif inmelyikmezo = 2 then /* jelleg */
24         UPDATE Etterem SET telefonszam = inszoveg WHERE azonosito = insorszam;
25     elsif inmelyikmezo = 3 then /* elkészítési idő */
26         UPDATE Etterem SET elso_nyitas = inszoveg WHERE azonosito = insorszam;
27     elsif inmelyikmezo = 4 then /* utolsó módosítás */
28         UPDATE Etterem SET specialitas = inszoveg WHERE azonosito = insorszam;
29     end if;
30
Results Explain Describe Saved SQL History

```

Procedure created.

0.03 seconds

A futtatás képernyőképe:

```

38  begin
39      ModEtterem('STOP éhség!', 1, 1);
40  end;

```

Results	Explain	Describe	Saved SQL	History
1 étterem módosítva!				
Statement processed.				
0.00 seconds				

Tárolt metódus hibakezelés bemutatására

Hibakezelést elsősorban a receptek beszűrő eljárásánál valósítottam meg, miszerint a mai dátumnál ne lehessen megadni az „utolsó módosítás” mezőnek nagyobb értéket, vagyis későbbi dátumot, mert annak nem lenne értelme

```
create or replace procedure InsertReceptek (innev in varchar2, injelleg in var
char2, inel_ido in varchar2, inut_mod in varchar2) as
    invaliddate exception;
begin
    if inut_mod > sysdate then
        raise invaliddate;
    else
        INSERT INTO Receptek (nev, jelleg, elkeszitesi_ido, utolso_modositas)
VALUES (innev, injelleg, inel_ido, inut_mod);
    end if;

exception
    when invaliddate then dbms_output.put_line('Az utolsó módosítás dátuma nem
    lehet későbbi mint a mai dátum!');
end;
```

A létrehozás képernyőképe:

```
149 create or replace procedure InsertReceptek (innev in varchar2, injelleg in varchar2, inel_ido in varchar2, inut_mod in varchar2) as
150     invaliddate exception;
151 begin
152     if inut_mod > sysdate then
153         raise invaliddate;
154     else
155         INSERT INTO Receptek (nev, jelleg, elkeszitesi_ido, utolso_modositas) VALUES (innev, injelleg, inel_ido, inut_mod);
156     end if;
157
158 exception
159     when invaliddate then dbms_output.put_line('Az utolsó módosítás dátuma nem lehet későbbi mint a mai dátum!');
160 end;
```

Results Explain Describe Saved SQL History

Procedure created.

0.00 seconds

szalaimarci99@gmail.com me_ab2 en Copyright © 1999, 2020, Oracle and/or its affiliates. All rights reserved.

A hibás működés képernyőképe:

```
163 begin
164     InsertReceptek('XXXXXXXX', 'előétel', 32, '01/01/2022');
165 end;
```

Results Explain Describe Saved SQL History

Az utolsó módosítás dátuma nem lehet későbbi mint a mai dátum!

Statement processed.

0.00 seconds

A normál működés képernyőképe:

```
164  
165 begin  
166     InsertReceptek('Pörkölt', 'főétel', 32, '01/01/2021');  
167 end;  
168
```

Results Explain Describe Saved SQL History

Statement processed.

0.01 seconds

Tárolt metódus explicit kurzor kezelésének bemutatására

Ez az eljárás arra szolgál, hogy egy adott sorszám/azonosítószám intervallumon belül rekordokat törölhessünk az adatbázisból. Az eljárás meghívásakor elég a táblák nevének az első két betűjét megadni, valamint az intervallumot.

```
create or replace procedure DeleteRecord(intablanev in varchar2, inmin in number, inmax in number) as
    cursor recept_adatok is select * from Receptek where sorszam between inmin and inmax;
    cursor etterem_adatok is select * from Etterem where azonosito between inmin and inmax;
    data varchar2(70);
    recepteknum number:= instr(intablanev, 're',1);
    etteremnum number := instr(intablanev, 'et',1);
begin
    if recepteknum > 0 then
        FOR a IN recept_adatok LOOP
            delete from Receptek where sorszam = a.sorszam;
            data:= a.sorszam||' - '||a.nev||' - '||a.jelleg||' - '||a.elkeszitesi_ido||' - '||a.utolso_modositas;
            dbms_output.put_line('Törölt adatsor: '||data);
        END LOOP;
    end if;

    if etteremnum > 0 then
        FOR b IN etterem_adatok LOOP
            delete from Etterem where azonosito = b.azonosito;
            data:=b.azonosito||' - '||b.nev||' - '||b.telefonszam||' - '||b.elso_nyitas||' - '||b.specialitas;
            dbms_output.put_line('Törölt adatsor: '||data);
        END LOOP;
    end if;
end;
```

A létrehozás képernyőképe:

```
1 create or replace procedure DeleteRecord(intablanev in varchar2, inmin in number, inmax in number) as
2     cursor recept_adatok is select * from Receptek where sorszam between inmin and inmax;
3     cursor etterem_adatok is select * from Etterem where azonosito between inmin and inmax;
4     data varchar2(70);
5     recepteknum number:= instr(intablanev, 're',1);
6     etteremnum number := instr(intablanev, 'et',1);
7 begin
8     if recepteknum > 0 then
9         FOR a IN recept_adatok LOOP
10             delete from Receptek where sorszam = a.sorszam;
11             data:= a.sorszam||' - '||a.nev||' - '||a.jelleg||' - '||a.elkeszitesi_ido||' - '||a.utolso_modositas;
12             dbms_output.put_line('Törölt adatsor: '||data);
13         END LOOP;
14     end if;
15
16     if etteremnum > 0 then
17         FOR b IN etterem_adatok LOOP
18             delete from Etterem where azonosito = b.azonosito;
19             data:=b.azonosito||' - '||b.nev||' - '||b.telefonszam||' - '||b.elso_nyitas||' - '||b.specialitas;
20             dbms_output.put_line('Törölt adatsor: '||data);
21         END LOOP;
22     end if;
23 end;
```

Results Explain Describe Saved SQL History

Procedure created.

0.03 seconds

A futtatás képernyőképe:

```
26 begin
27     DeleteRecord('rec', 8, 10);
28 end;
```

Results Explain Describe Saved SQL History

Törölt adatsor: 8 - Cigánypecsenye - főétel - 30 - 05/13/2010
Törölt adatsor: 9 - Lekváros palacsinta - desszert - 17 - 04/28/2003
Törölt adatsor: 10 - Rizskoch - desszert - 60 - 11/11/2007

Statement processed.

0.01 seconds