

Adatbázis Rendszerek 2

Jegyzőkönyv

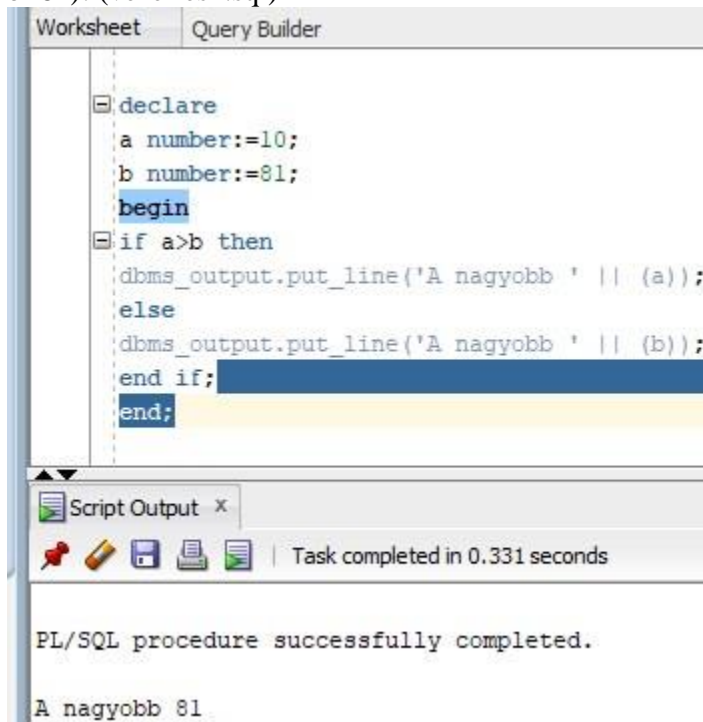
2022.03.23 gyakorlat

Szerző: Kovács Sándor Gergő

Neptunkód: C00KSI

Tárgyfelelős: Dr. Bednarik László

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



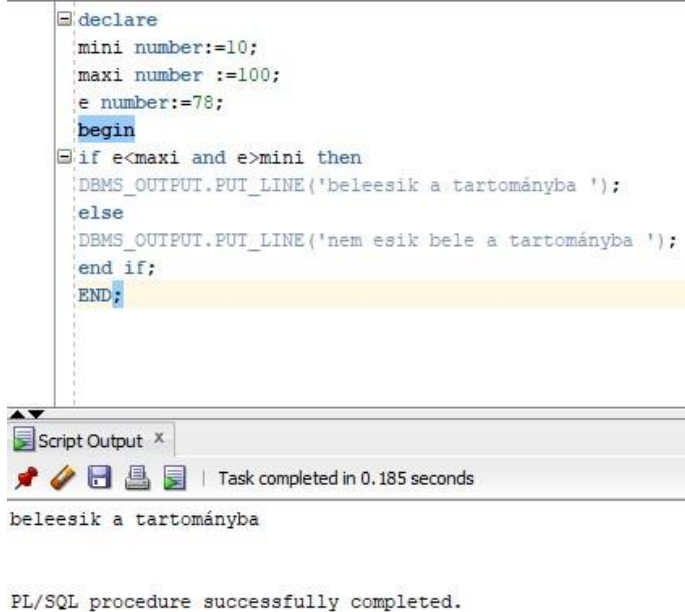
The screenshot shows the Oracle SQL Developer interface. The top pane is titled 'Worksheet' and contains a PL/SQL script. The script declares two variables, 'a' and 'b', with values 10 and 81 respectively. It then uses an 'if' statement to compare them and prints the result using 'dbms_output.put_line'. The bottom pane is titled 'Script Output' and shows the message 'PL/SQL procedure successfully completed.' followed by the output 'A nagyobb 81'. The status bar indicates 'Task completed in 0.331 seconds'.

```
declare
a number:=10;
b number:=81;
begin
if a>b then
dbms_output.put_line('A nagyobb ' || (a));
else
dbms_output.put_line('A nagyobb ' || (b));
end if;
end;
```

PL/SQL procedure successfully completed.

A nagyobb 81

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



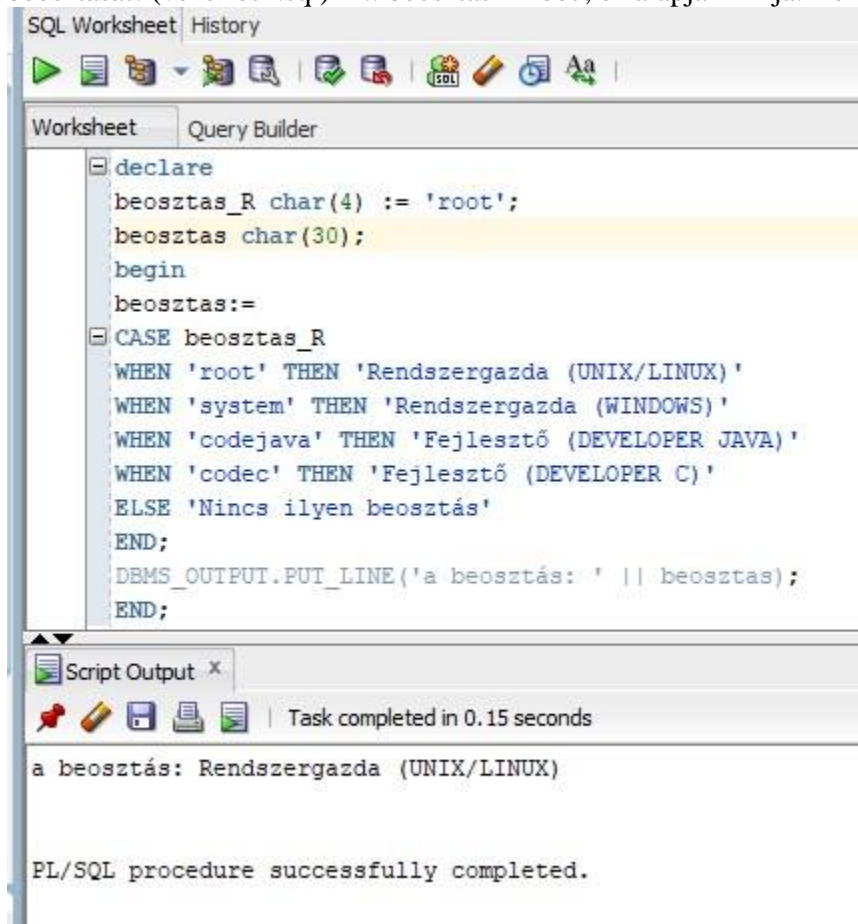
The screenshot shows the Oracle SQL Developer interface. The top pane is titled 'Worksheet' and contains a PL/SQL script. The script declares three variables: 'mini' (10), 'maxi' (100), and 'e' (78). It then uses an 'if' statement to check if 'e' is between 'mini' and 'maxi' and prints the result using 'DBMS_OUTPUT.PUT_LINE'. The bottom pane is titled 'Script Output' and shows the message 'PL/SQL procedure successfully completed.' followed by the output 'beleesik a tartományba'. The status bar indicates 'Task completed in 0.185 seconds'.

```
declare
mini number:=10;
maxi number :=100;
e number:=78;
begin
if e<maxi and e>mini then
DBMS_OUTPUT.PUT_LINE('beleesik a tartományba ');
else
DBMS_OUTPUT.PUT_LINE('nem esik bele a tartományba ');
end if;
END;
```

beleesik a tartományba

PL/SQL procedure successfully completed.

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



The screenshot shows an SQL Worksheet application with a toolbar at the top containing icons for running, saving, and editing. Below the toolbar are two tabs: 'Worksheet' and 'Query Builder'. The 'Worksheet' tab is active, displaying a PL/SQL script. The script declares a variable 'beosztas_R' of type 'char(4)' and assigns it the value 'root'. It then declares another variable 'beosztas' of type 'char(30)'. The script begins a block where it uses a CASE statement to assign a value to 'beosztas' based on 'beosztas_R'. The CASE statement has four WHEN clauses: 'root' maps to 'Rendszergazda (UNIX/LINUX)', 'system' maps to 'Rendszergazda (WINDOWS)', 'codejava' maps to 'Fejlesztő (DEVELOPER JAVA)', and 'codec' maps to 'Fejlesztő (DEVELOPER C)'. An ELSE clause assigns 'Nincs ilyen beosztás'. After the CASE statement, the script uses 'DBMS_OUTPUT.PUT_LINE' to print the value of 'beosztas' and ends the block. Below the script editor is a 'Script Output' window. It shows a message 'Task completed in 0.15 seconds' and the output 'a beosztás: Rendszergazda (UNIX/LINUX)'. At the bottom of the window, it states 'PL/SQL procedure successfully completed.'

```
SQL Worksheet History

Worksheet Query Builder

declare
beosztas_R char(4) := 'root';
beosztas char(30);
begin
beosztas:=
CASE beosztas_R
WHEN 'root' THEN 'Rendszergazda (UNIX/LINUX)'
WHEN 'system' THEN 'Rendszergazda (WINDOWS)'
WHEN 'codejava' THEN 'Fejlesztő (DEVELOPER JAVA)'
WHEN 'codec' THEN 'Fejlesztő (DEVELOPER C)'
ELSE 'Nincs ilyen beosztás'
END;
DBMS_OUTPUT.PUT_LINE('a beosztás: ' || beosztas);
END;
```

Script Output x

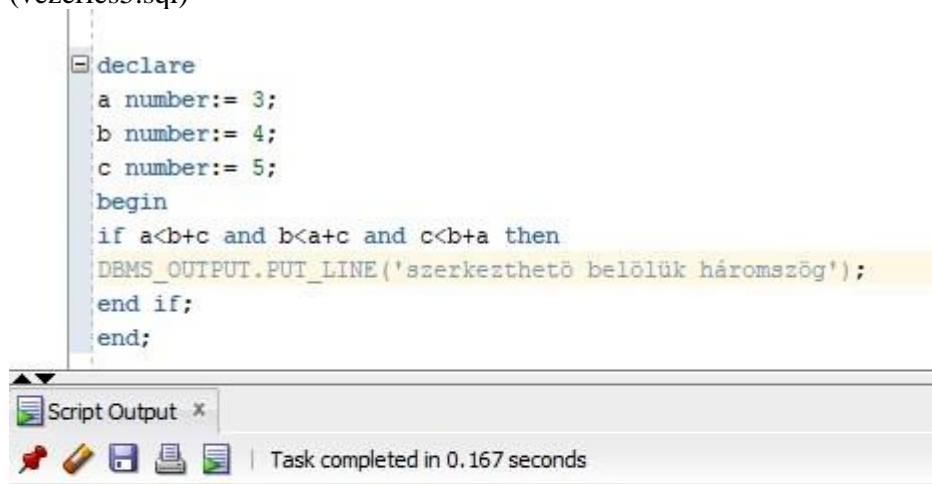
Task completed in 0.15 seconds

a beosztás: Rendszergazda (UNIX/LINUX)

PL/SQL procedure successfully completed.

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

```
declare
a number:= 3;
b number:= 4;
c number:= 5;
begin
if a<b+c and b<a+c and c<b+a then
DBMS_OUTPUT.PUT_LINE('szerkezthető belőlük háromszög');
end if;
end;
```

The image shows a screenshot of a SQL IDE. The top part is a code editor with a PL/SQL script. The script declares three variables: a, b, and c, all of type number, with values 3, 4, and 5 respectively. It then enters a begin block with an if statement. The if statement checks the triangle inequality: a < b+c, b < a+c, and c < b+a. If all three conditions are true, it calls DBMS_OUTPUT.PUT_LINE with the string 'szerkezthető belőlük háromszög'. The script ends with end if; and end;. Below the code editor is a 'Script Output' window. It has a title bar with a close button. Below the title bar is a toolbar with icons for a pin, a pencil, a folder, a document, and a printer. To the right of the toolbar, it says 'Task completed in 0.167 seconds'.

Script Output x

Task completed in 0.167 seconds

PL/SQL procedure successfully completed.

szerkezthető belőlük háromszög

PL/SQL procedure successfully completed.

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

```
declare
a number:=3;
b number:=4;
c number:=5;
s number;
T number;
begin
s := (a+b+c)/2;
T := sqrt(s*(s-a)*(s-b)*(s-c));
DBMS_OUTPUT.PUT_LINE('a háromszög területe: ' || T);
END;
```

Script Output x

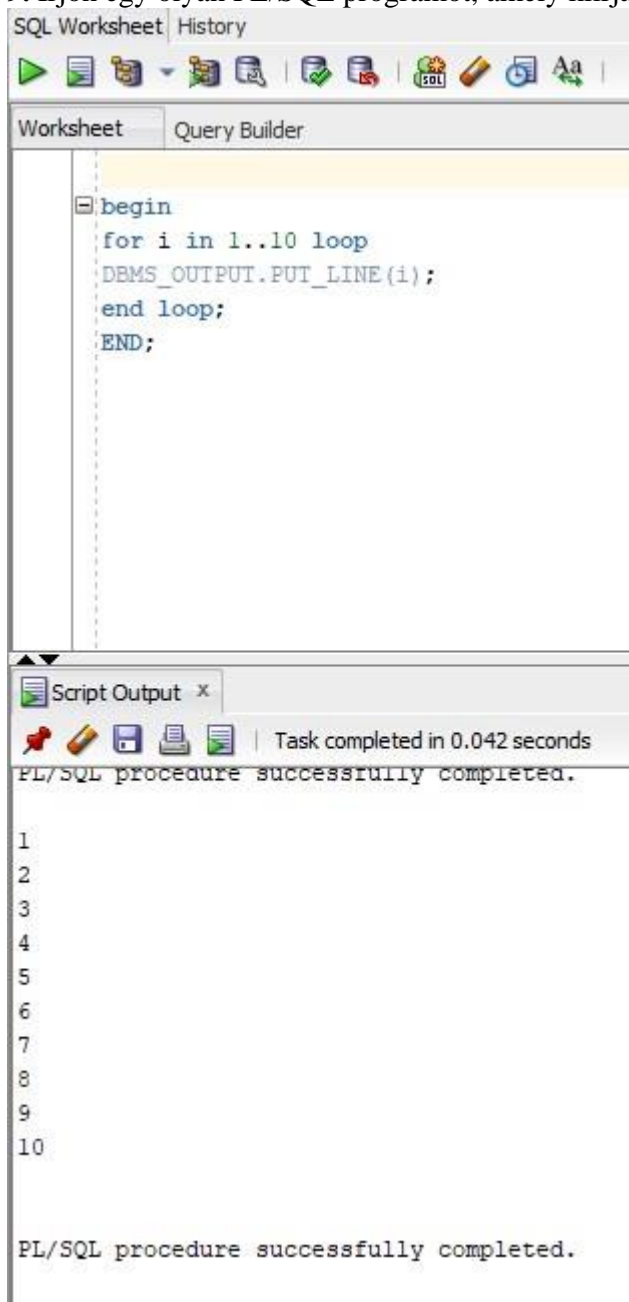
Task completed in 0.178 seconds

PL/SQL procedure successfully completed.

a háromszög területe: 6

PL/SQL procedure successfully completed.

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



The screenshot shows an SQL Worksheet interface. The top toolbar contains icons for running, saving, and other database operations. The main window is divided into two tabs: 'Worksheet' and 'Query Builder'. The 'Worksheet' tab is active, displaying a PL/SQL program. The program is a loop that prints numbers from 1 to 10. Below the code editor, there is a 'Script Output' window. It shows the message 'Task completed in 0.042 seconds' and 'PL/SQL procedure successfully completed.' followed by the numbers 1 through 10, each on a new line. At the bottom of the output window, the message 'PL/SQL procedure successfully completed.' is repeated.

```
begin
  for i in 1..10 loop
    DBMS_OUTPUT.PUT_LINE(i);
  end loop;
END;
```

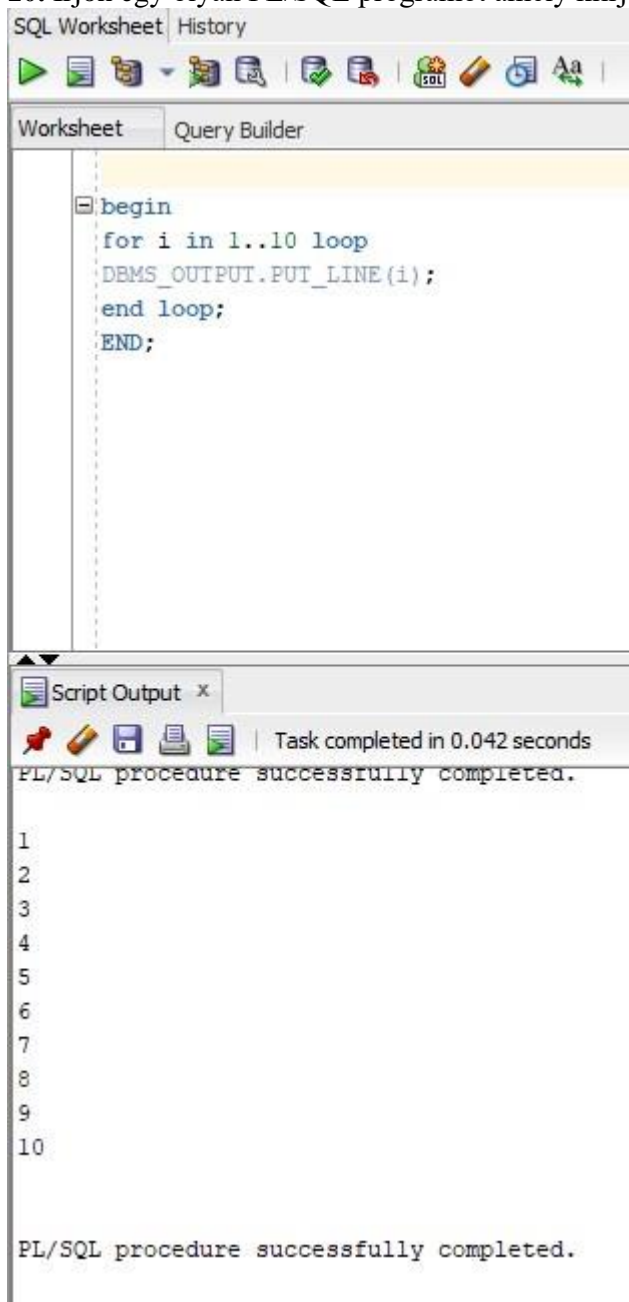
Task completed in 0.042 seconds

PL/SQL procedure successfully completed.

1
2
3
4
5
6
7
8
9
10

PL/SQL procedure successfully completed.

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



The screenshot shows an SQL Worksheet interface. The top toolbar includes icons for running, saving, and editing. The main window is divided into a 'Worksheet' tab and a 'Query Builder' tab. The 'Worksheet' tab contains a PL/SQL program. Below the worksheet, a 'Script Output' window is open, displaying the execution results.

```
begin
  for i in 1..10 loop
    DBMS_OUTPUT.PUT_LINE(i);
  end loop;
END;
```

Task completed in 0.042 seconds

PL/SQL procedure successfully completed.

1
2
3
4
5
6
7
8
9
10

PL/SQL procedure successfully completed.

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

```
declare
a number:=7;
osztó integer := 2;
prim boolean := true;
begin
loop
if mod(a,osztó)=0 then
prim := false;
end if;
osztó := osztó+1;
exit when prim=false or osztó>a/2;
end loop;
dbms_output.put_line(' primszám: ' || case when prim = true then 'igen' else 'nem' end);
end;
```

Script Output x

Task completed in 0.06 seconds

PL/SQL procedure successfully completed.

primszám: igen

PL/SQL procedure successfully completed.