

Zadania do realizacji

1. Zaimplementuj funkcję o nazwie `odleglosc(x1,y1,x2,y2)` obliczającą odległość na płaszczyźnie pomiędzy punktami $(x1,y1)$ oraz $(x2,y2)$. Przetestuj działanie zaimplementowanej funkcji.
2. Utwórz tabelę `produkt(id,nazwa,cena)` i dodaj przykładowe rekordy.
Napisz procedurę o nazwie `aktualizacja_ceny(p_id_produkt,p_procent)`, która ma zaktualizować cenę zadanego produktu o podany przez parametr procent. Przetestuj działanie procedury.
3. Napisz funkcję o nazwie `wiek(p_data_ur)`, która dla podanej przez parametr wartości daty urodzenia obliczy wiek osoby. Przetestuj działanie napisanej funkcji.
4. Dana jest tabela: `CREATE TABLE l_pierwsze(liczba INT);`
Napisz procedurę o nazwie `wpisz_liczby_pierwsze(p_od,p_do)`, która wpisze do tabeli `l_pierwsze` wszystkie liczby pierwsze z przedziału $<p_od,p_do>$.
Napisz instrukcję testującą działanie utworzonej procedury.
5. a) Napisz funkcję o nazwie `czy_piątek_trzynastego(p_data)` zwracającą prawdę, gdy zadana przez parametr `p_data` jest piątkiem trzynastego lub fałsz, gdy nie jest to piątek trzynastego.
b) Utwórz tabelę `urodziny(id_osoba,imie,nazwisko,data_ur)` i dodaj do niej kilka rekordów.
c) Użyj funkcji `czy_piątek_trzynastego` i z tabeli `urodziny` wypisz wszystkie osoby, które urodziły się w piątek trzynastego.
d) Napisz procedurę o nazwie `wypisz_piątki_trzynastego(p_data_od,p_data_do)`, która wpisze na ekranie wszystkie piątki trzynastego z zakresu dat $<data_od,data_do>$. Użyj funkcji `czy_piątek_trzynastego`. Przetestuj działanie utworzonej procedury.

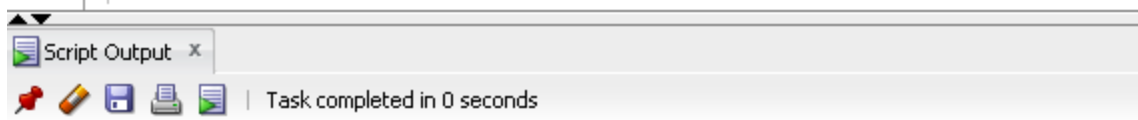
Rozwiązania

1. Zaimplementuj funkcję o nazwie `odleglosc(x1,y1,x2,y2)` obliczającą odległość na płaszczyźnie pomiędzy punktami $(x1,y1)$ oraz $(x2,y2)$. Przetestuj działanie zaimplementowanej funkcji.

```
1 CREATE OR REPLACE FUNCTION odleglosc(  
2     x1 NUMBER,  
3     y1 NUMBER,  
4     x2 NUMBER,  
5     y2 NUMBER )  
6 RETURN NUMBER IS  
7 BEGIN  
8     RETURN SQRT (POWER(x1-x2,2)+POWER(y1-y2,2)) ;  
9 END;  
10 /
```

Przykładowe wywołanie funkcji dla $x1=0$, $y1=0$, $x2=1$, $y2=1$:

```
11 SET SERVEROUTPUT ON  
12 /  
13 BEGIN  
14     DBMS_OUTPUT.PUT_LINE(odleglosc(0,0,1,1));  
15 END;  
16 /
```



```
anonymous block completed  
1,41421356237309504880168872420969807857
```

2. Utwórz tabelę produkt(id,nazwa,cena) i dodaj przykładowe rekordy.

Napisz procedurę o nazwie aktualizacja_ceny(p_id_produkt,p_procent), która ma zaktualizować cenę zadanego produktu o podany przez parametr procent. Przetestuj działanie procedury.

```

1 CREATE TABLE produkt (
2     id INT,
3     nazwa VARCHAR2(30),
4     cena NUMBER(10,2)
5 );
6 /
7 INSERT INTO produkt VALUES (1, 'Telewizor', 4000);
8 INSERT INTO produkt VALUES (2, 'Pralka', 1500);
9 INSERT INTO produkt VALUES (3, 'Komórka', 2000);
10 /
11 CREATE OR REPLACE PROCEDURE aktualizacja_ceny(
12     p_id_produkt produkt.id%TYPE,
13     p_procent INT
14 ) IS
15 BEGIN
16     UPDATE produkt
17     SET cena=cena*(1+p_procent/100)
18     WHERE id=p_id_produkt;
19 END;
20 /
21 SET SERVEROUTPUT ON
22 /
23 SELECT * FROM produkt;
24 /
25 EXECUTE aktualizacja_ceny(1, 10);
26 /
27 SELECT * FROM produkt;
28 /

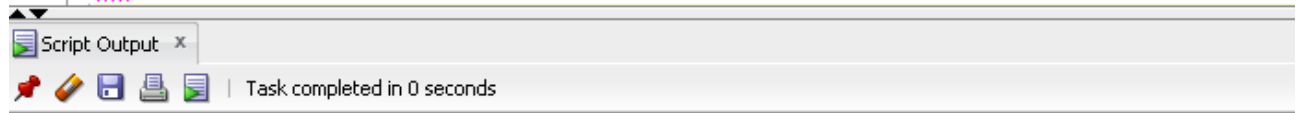
```

ID	NAZWA	CENA
1	1 Telewizor	4000
2	2 Pralka	1500
3	3 Komórka	2000

ID	NAZWA	CENA
1	1 Telewizor	4400
2	2 Pralka	1500
3	3 Komórka	2000

3. Napisz funkcję o nazwie `wiek(p_data_ur)`, która dla podanej przez parametr wartości daty urodzenia obliczy wiek osoby. Przetestuj działanie napisanej funkcji.

```
1 CREATE OR REPLACE FUNCTION wiek(p_data_ur DATE)
2 RETURN INTEGER IS
3 BEGIN
4     RETURN months_between(sysdate,p_data_ur)/12;
5 END;
6 /
7 BEGIN
8     DBMS_OUTPUT.PUT_LINE(wiek(to_date('1975-05-25','yyyy-mm-dd')));
9 END;
10 /
```



The screenshot shows a 'Script Output' window with a toolbar containing icons for redo, undo, save, print, and refresh. Below the toolbar, it displays the message 'Task completed in 0 seconds'.

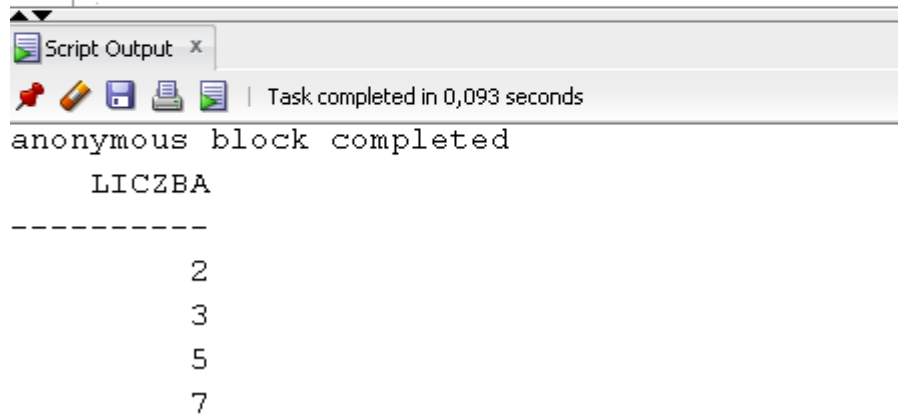
```
anonymous block completed
40
```

4. Dana jest tabela: `CREATE TABLE l_pierwsze(liczba INT);`
Napisać procedurę o nazwie `wpisz_liczby_pierwsze(p_od,p_do)`, która wpisze do tabeli `l_pierwsze` wszystkie liczby pierwsze z przedziału `<p_od,p_do>`.
Napisać instrukcję testującą działanie utworzonej procedury.

```
1 CREATE TABLE l_pierwsze(liczba INT);  
2 /  
3 CREATE OR REPLACE PROCEDURE wpisz_liczby_pierwsze (  
4     p_od INT,  
5     p_do INT  
6 ) IS  
7     v_czy BOOLEAN;  
8 BEGIN  
9     FOR v_i IN p_od..p_do LOOP  
10         v_czy:=true; --zakładamy, że liczba jest pierwsza  
11         FOR v_j IN 2..(v_i/2) LOOP  
12             IF MOD(v_i,v_j)=0 THEN  
13                 v_czy:=false; --jednak liczba nie jest pierwsza  
14                 continue;  
15             END IF;  
16         END LOOP;  
17         IF v_czy=true THEN  
18             INSERT INTO l_pierwsze(liczba) VALUES (v_i);  
19         END IF;  
20     END LOOP;  
21 END;  
22 /
```

Przykładowe wywołanie procedury dla $p_od=2$ i $p_do=10$:

```
23 SET SERVEROUTPUT ON;  
24 /  
25 EXECUTE wpisz_liczby_pierwsze (2, 10);  
26 /  
27 SELECT * FROM l_pierwsze;  
28
```



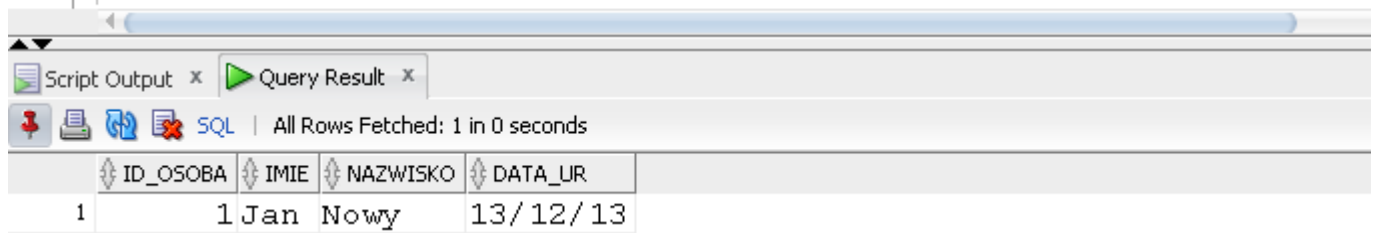
```
Script Output x  
Task completed in 0,093 seconds  
anonymous block completed  
LICZBA  
-----  
2  
3  
5  
7
```

5. a) Napisać funkcję o nazwie czy_piątek_trzynastego(p_data) zwracającą prawdę, gdy zadana przez parametr p_data jest piątkiem trzynastego lub fałsz, gdy nie jest to piątek trzynastego.
b) Utworzyć tabelę urodziny(id_osoba,imie,nazwisko,data_ur) i dodać do niej kilka rekordów.
c) Użyć funkcji czy_piątek_trzynastego i z tabeli urodziny wypisz wszystkie osoby, które urodziły się w piątek trzynastego.
d) Napisać procedurę o nazwie wypisz_piątki_trzynastego(p_data_od,p_data_do), która wpisze na ekranie wszystkie piątki trzynastego z zakresu dat <data_od,data_do>. Użyć funkcji czy_piątek_trzynastego. Przetestuj działanie utworzonej procedury.

```
1  --a
2  CREATE OR REPLACE FUNCTION czy_piątek_trzynastego(p_data DATE)
3  RETURN INTEGER IS
4  BEGIN
5  IF to_char(p_data, 'd')=5 AND to_char(p_data, 'dd')=13 THEN
6      RETURN 1;
7  ELSE
8      RETURN 0;
9  END IF;
10 END;
11 /
```

```
12 --b
13 CREATE TABLE urodziny(
14     id_osoba INT,
15     imie VARCHAR2(20),
16     nazwisko VARCHAR2(30),
17     data_ur DATE
18 );
19 /
20 INSERT INTO urodziny VALUES (1, 'Jan', 'Nowy', '2013-12-13');
21 INSERT INTO urodziny VALUES (1, 'Anna', 'Stara', '2013-12-12');
22 /
```

```
23  --c
24  SELECT * FROM urodziny WHERE czy_piątek_trzynastego(data_ur)=1;
25  /
```



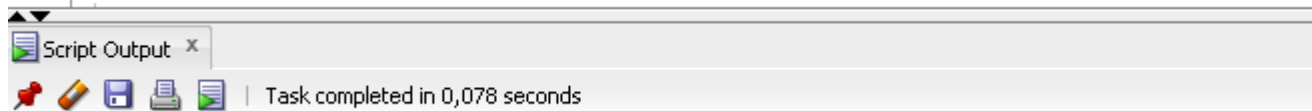
The screenshot shows the SQL Developer interface. At the top, there are tabs for 'Script Output' and 'Query Result'. Below the tabs, a status bar indicates 'All Rows Fetched: 1 in 0 seconds'. The main area displays a table with the following columns: ID_OSOBA, IMIE, NAZWISKO, and DATA_UR. The table contains one row of data.

ID_OSOBA	IMIE	NAZWISKO	DATA_UR
1	Jan	Nowy	13/12/13

```
26  --d
27  CREATE PROCEDURE wypisz_piatki_trzynastego(
28      p_data_od DATE,
29      p_data_do DATE
30  ) IS
31      v_data_tmp DATE;
32  BEGIN
33      --znajdujemy pierwszy dzień 13-go po dacie p_data_od
34      v_data_tmp:=to_date(to_char(p_data_od, 'yyyy') || '-'
35                          || to_char(p_data_od, 'mm') || '-13');
36      IF( p_data_od > v_data_tmp) THEN
37          v_data_tmp:=add_months(v_data_tmp, 1);
38      END IF;
39      --szukamy piątków 13-go
40      WHILE (v_data_tmp<p_data_do) LOOP
41          IF to_char(v_data_tmp, 'd')=5 THEN
42              --wypisujemy piątki 13-go
43              DBMS_OUTPUT.PUT_LINE(to_char(v_data_tmp, 'yyyy-mm-dd'));
44          END IF;
45          v_data_tmp:=add_months(v_data_tmp, 1);
46      END LOOP;
47  END;
48  /
```



```
49 | SET SERVEROUTPUT ON
50 | /
51 | EXECUTE wypisz_piatki_trzynastego('2013-01-01','2014-12-31');
52 | /
```



anonymous block completed

2013-09-13

2013-12-13

2014-06-13