

**Bazy danych,
lista nr 6 (ćwiczenia)**

1. Wykonano poniższe działanie na relacjach, gdzie $\oplus$ oznacza złączenie naturalne:

<u>place</u>	<u>time</u>	<u>tmp</u>	<u>pres</u>		<u>place</u>	<u>time</u>	<u>id</u>
gr	0	15	1015		gr	3	b16
trp	0	-10	613	$\oplus$	trp	0	jk
gr	15	18	1010		trp	0	lk
trp	17	-16	600		gr	15	b16

Jaka relacja powstała w wyniku powyższego działania?

2. Wykonano poniższe działanie na relacjach, gdzie $\oplus$ oznacza złączenie naturalne:

<u>a</u>	<u>b</u>	<u>c</u>		<u>a</u>	<u>b</u>	<u>c</u>		<u>a</u>	<u>b</u>	<u>d</u>		<u>a</u>	<u>b</u>	<u>c</u>
3	2	1	$\oplus$	3	2	1	$\oplus$	3	3	9	$\oplus$	3	2	1
4	6	5		4	6	5	$\oplus$	3	2	9		4	6	5
3	2	9		3	2	9		4	6	8		3	2	9
								4	6	7				

Jaka relacja powstała w wyniku powyższego działania?

3. Wykonano poniższe działanie na relacjach, gdzie $\oplus_L$ oznacza lewe złączenie zewnętrzne, zaś $\oplus_R$ prawe złączenie zewnętrzne:

<u>a</u>	<u>b</u>	<u>c</u>		<u>a</u>	<u>b</u>	<u>d</u>		<u>e</u>	<u>b</u>	<u>d</u>
1	2	3	$\oplus_L$	1	2	8	$\oplus_R$	7	2	9
4	5	6		4	6	5		7	2	7
1	2	7		1	2	9		1	2	8

Jaka relacja powstała w wyniku powyższego działania?

4. Poniższy cykl komend SQLa i proceduralnego SQLa, wykonany na pustej bazie danych, zwróci na końcu krotkę z kilkoma liczbami. Jakie to będą liczby?

```
CREATE TABLE tabela(a INTEGER PRIMARY KEY, b INTEGER);
INSERT INTO tabela(a,b) VALUES (1,2), (3,4), (5,6);

CREATE FUNCTION przyklad(x INTEGER) RETURNS INTEGER AS $$
DECLARE
  y RECORD;
  z INTEGER;
BEGIN
  SELECT * INTO y FROM tabela WHERE a=x;
  IF (FOUND) THEN
    z = y.b;
  ELSE
    z = x;
  END IF;
  RETURN z;
END;
$$ LANGUAGE 'plpgsql';

SELECT przyklad(1), przyklad(2), przyklad(3), przyklad(4);
```

5. W pustej bazie danych założono (z powodzeniem) następującą funkcję proceduralnego SQL-a:

```
CREATE FUNCTION liczba(n INTEGER) RETURNS INTEGER AS $$
DECLARE
  a INTEGER;
  w INTEGER;
BEGIN
  a=1;
  w=a;
  WHILE(a<n) LOOP
    a=a+1;
    w=w*a;
  END LOOP;
  RETURN w;
END;
$$ LANGUAGE 'plpgsql';
```

Opisz, co dokładnie liczy ta funkcja. Jaką liczbę zwróci polecenie `SELECT liczba(6);`?

6. W pustej bazie danych założono (z powodzeniem) następującą funkcję proceduralnego SQL-a:

```
CREATE FUNCTION liczba(x INTEGER) RETURNS INTEGER AS $$
DECLARE
  a INTEGER;
  b INTEGER;
  z INTEGER;
  n INTEGER;
BEGIN
  a=0;
  b=1;
  z=1;
  n=1;
  WHILE(n<x) LOOP
    z=a+b;
    a=b;
    b=z;
    n=n+1;
  END LOOP;
  RETURN z;
END;
$$ LANGUAGE 'plpgsql';
```

Opisz, co dokładnie liczy ta funkcja. Jaką liczbę zwróci polecenie `SELECT liczba(12);`?

7. W pustej bazie danych założono (z powodzeniem) następujące dwie funkcje (niekoniecznie optymalnie napisane):

```
CREATE OR REPLACE FUNCTION cusum(n INTEGER) RETURNS INTEGER AS $$
DECLARE
    k INTEGER;
    w INTEGER;
BEGIN
    k = 1;
    w = k;
    WHILE(k<n) LOOP
        k = k+1;
        w = w+k*k;
    END LOOP;
    RETURN w;
END;
$$ LANGUAGE 'plpgsql';
```

```
CREATE OR REPLACE FUNCTION policz(a INTEGER, b INTEGER) RETURNS INTEGER AS $$
BEGIN
    IF (a != b) THEN
        RETURN cusum(a) + cusum(b);
    ELSE
        RETURN 2*cusum(a);
    END IF;
END;
$$ LANGUAGE 'plpgsql';
```

Opisz dokładnie działanie tych funkcji. Odpowiedz też na poniższe dwa pytania:

- (a) Jakie liczby (i dlaczego) otrzymamy w wyniku poniższego polecenia?
SELECT cusum(5), cusum(1), cusum(2)*cusum(3);
- (b) Jakie liczby (i dlaczego) otrzymamy w wyniku poniższego polecenia?
SELECT policz(4,4), policz(2,5);

8. W bazie danych założona jest jedna tabela o nazwie **tab** z jedną kolumną **x** typu INTEGER. Tabela jest pusta. Na tabeli nie ma póki co założonych żadnych więzów. Zakładamy na niej (z powodzeniem) wyzwalacz następującym skryptem SQL-owym:

```
CREATE OR REPLACE FUNCTION func() RETURNS TRIGGER AS $$
BEGIN
    IF (NEW.x > 10 AND NEW.x <=100) THEN
        NEW.x = 10;
        RETURN NEW;
    ELSIF (NEW.x > 100) THEN
        RETURN NEW;
    ELSIF (NEW.x <= 10) THEN
        RAISE EXCEPTION 'bad value';
    END IF;
    RETURN NULL;
END;
$$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER tr BEFORE INSERT ON tab FOR EACH ROW EXECUTE PROCEDURE func();
```

Następnie dodajemy do tej tabeli krotki poleceniami:

```
INSERT INTO tab(x) VALUES (1);
INSERT INTO tab(x) VALUES (20);
INSERT INTO tab(x) VALUES (300);
```

Opisz działanie tego wyzwalacza. Jaka dokładnie będzie zawartość tej tabeli po wykonaniu tych trzech INSERTów?

9. W pustej bazie danych wykonano (z powodzeniem) następujący skrypt SQL-owy:

```
CREATE TABLE numbers(id INTEGER PRIMARY KEY, n INTEGER NOT NULL);
CREATE OR REPLACE FUNCTION podwoj() RETURNS TRIGGER AS $$
BEGIN
    IF (NEW.n < OLD.n) THEN
        RAISE EXCEPTION 'Nie pomniejszamy!';
    ELSE
        NEW.n = 2*NEW.n;
    END IF;
    RETURN NEW;
END;
$$ LANGUAGE 'plpgsql';
CREATE TRIGGER podwoj_trigger BEFORE UPDATE ON numbers
FOR EACH ROW EXECUTE PROCEDURE podwoj();
```

Opisz działanie wyzwalacza zawartego w tym skrypcie. Następnie opisz, co się stanie po wykonaniu poniższych poleceń:

```
INSERT INTO numbers(id,n) VALUES (1,2),(2,5),(3,10);
UPDATE numbers SET n = 4 WHERE id = 2;
UPDATE numbers SET n = 7 WHERE n < 3;
SELECT sum(n) FROM numbers;
```