

Laboratorium 3

Selekcja oparta na jednej tabeli: funkcje operujące na datach i napisach, porównywanie wzorców (like, similar to). Łączenie wyników zapytań (union, intersect, except). Obliczenia w zapytaniach.

Zadanie 3.1

baza danych: cukiernia

Napisz zapytanie w języku SQL wyświetlające informacje na temat zamówień (idZamowienia, dataRealizacji), które mają być zrealizowane:

1. między 12 i 20 listopada 2013,

```
SELECT idzamowienia, datarealizacji FROM zamowienia
WHERE datarealizacji BETWEEN '2013-11-12' AND '2013-11-20';
```

2. między 1 i 6 grudnia lub między 15 i 20 grudnia 2013

```
SELECT idzamowienia, datarealizacji FROM zamowienia
WHERE datarealizacji BETWEEN '2013-12-01' AND '2013-12-06'
OR datarealizacji BETWEEN '2013-12-15' AND '2013-12-20';
```

3. w grudniu 2013 (nie używaj funkcji date_part ani extract),

```
SELECT idZamowienia, dataRealizacji FROM public.zamowienia
WHERE datarealizacji BETWEEN '2013-12-01' AND '2013-12-31';
```

4. w listopadzie 2013 (w tym i kolejnych zapytaniach użyj funkcji date_part lub extract),

```
SELECT idZamowienia, dataRealizacji FROM public.zamowienia
WHERE DATE_PART('year', datarealizacji) = 2013
AND DATE_PART('month', datarealizacji) = 11;
```

5. ★ w listopadzie lub grudniu 2013,

```
SELECT idZamowienia, dataRealizacji FROM public.zamowienia
WHERE DATE_PART('year', datarealizacji) = 2013
AND (DATE_PART('month', datarealizacji) = 11
OR DATE_PART('month', datarealizacji) = 12);

SELECT idzamowienia, datarealizacji FROM zamowienia
WHERE EXTRACT(month FROM datarealizacji) BETWEEN 11 AND 12
AND EXTRACT(year FROM datarealizacji) = 2013;
```

6. ★ 17, 18 lub 19 dnia miesiąca,

```
SELECT idZamowienia, dataRealizacji FROM public.zamowienia
WHERE DATE_PART('day', datarealizacji) IN ('17', '18', '19');
```

```
SELECT idzamowienia, datarealizacji FROM zamowienia
WHERE EXTRACT(day FROM datarealizacji) BETWEEN 17 AND 19;
```

7. ★ 46 lub 47 tygodniu roku.

```
SELECT idZamowienia, dataRealizacji FROM public.zamowienia
WHERE DATE_PART('week', datarealizacji) IN ('46', '47');
```

```
SELECT idzamowienia, datarealizacji FROM zamowienia
WHERE EXTRACT(week FROM datarealizacji) BETWEEN 46 AND 47;
```

Zadanie 3.2

baza danych: cukiernia

Napisz zapytanie w języku SQL wyświetlające informacje na temat czekoladek (idCzekoladki, nazwa, czekolada, orzechy, nadzienie), których nazwa:

1. rozpoczyna się na literę 'S',

```
SELECT idczekoladki, nazwa, czekolada, orzechy, nadzienie FROM czekoladki
WHERE nazwa ~ '^S';
```

```
SELECT idCzekoladki, nazwa, czekolada, orzechy, nadzienie FROM czekoladki
WHERE nazwa LIKE 'S%';
```

2. rozpoczyna się na literę 'S' i kończy się na literę 'i',

```
SELECT idCzekoladki, nazwa, czekolada, orzechy, nadzienie FROM czekoladki
WHERE nazwa LIKE 'S%i';
```

```
SELECT idczekoladki, nazwa, czekolada, orzechy, nadzienie FROM czekoladki
WHERE nazwa ~ '^S.*i$';
```

3. rozpoczyna się na literę 'S' i zawiera słowo rozpoczynające się na literę 'm',

```
SELECT idCzekoladki, nazwa, czekolada, orzechy, nadzienie FROM czekoladki
WHERE nazwa LIKE 'S%m%';
```

```
SELECT idczekoladki, nazwa, czekolada, orzechy, nadzienie FROM czekoladki
WHERE nazwa ~ '^S.*m.*$';
```

4. rozpoczyna się na literę 'A', 'B' lub 'C',

```
SELECT idCzekoladki, nazwa, czekolada, orzechy, nadzienie FROM czekoladki
WHERE nazwa LIKE 'A%' OR nazwa LIKE 'B%' OR nazwa LIKE 'C%';
```

```
SELECT idczekoladki, nazwa, czekolada, orzechy, nadzienie FROM czekoladki
WHERE nazwa ~ '^(A|B|C)';
```

5. zawiera rdzeń 'orzech' (może on wystąpić na początku i wówczas będzie pisany z wielkiej litery),

```
SELECT idCzekoladki, nazwa, czekolada, orzechy, nadzienie FROM czekoladki
WHERE nazwa ~ '(Orzech|orzech)';
```

```
SELECT idczekoladki, nazwa, czekolada, orzechy, nadzienie FROM czekoladki
WHERE nazwa ~ '(.*) (o|O)rzech(.*)';
```

6. * rozpoczyna się na literę 'S' i zawiera w środku literę 'm',

```
SELECT idCzekoladki, nazwa, czekolada, orzechy, nadzienie FROM czekoladki
WHERE nazwa LIKE 'S%m%';
```

```
SELECT idczekoladki, nazwa, czekolada, orzechy, nadzienie FROM czekoladki
WHERE nazwa ~ '^S.*m.*$';
```

7. * zawiera słowo 'maliny' lub 'truskawki',

```
SELECT idCzekoladki, nazwa, czekolada, orzechy, nadzienie FROM czekoladki
WHERE nazwa ~ '(maliny|truskawki)';
```

8. * nie rozpoczyna się żadną z liter: 'D'-'K', 'S' i 'T',

```
SELECT idczekoladki, nazwa, czekolada, orzechy, nadzienie FROM czekoladki
WHERE nazwa ~ '^[^D-KST]';
```

9. * rozpoczyna się od 'Slod' ('Ślod'),

```
SELECT idCzekoladki, nazwa, czekolada, orzechy, nadzienie FROM czekoladki
WHERE nazwa ~ '(Slod|Ślod)';
```

```
SELECT idczekoladki, nazwa, czekolada, orzechy, nadzienie FROM czekoladki
WHERE nazwa ~ '^S(l|ł)od.*$';
```

10. * składa się dokładnie z jednego słowa.

```
SELECT idczekoladki, nazwa, czekolada, orzechy, nadzienie FROM czekoladki
WHERE nazwa ~ '^\\S*$';
```

Zadanie 3.3

baza danych: cukiernia

Napisz zapytanie w języku SQL oparte na tabeli Klienci, które:

1. wyświetla nazwy miast, z których pochodzą klienci cukierni i które składają się z więcej niż jednego słowa,

```
SELECT miejscowosc FROM public.klienci
WHERE miejscowosc LIKE '%_ %';
```

```
SELECT miejscowosc FROM public.klienci
WHERE miejscowosc ~ '^\\s+$';
```

2. wyświetla nazwy klientów, którzy podali numer stacjonarny telefonu,

```
SELECT nazwa FROM klienci
WHERE telefon ~ '^0';
```

3. * wyświetla nazwy klientów, którzy podali numer komórkowy telefonu,

```
SELECT nazwa FROM klienci
WHERE telefon ~ '^\\[0]';
```

Zadanie 3.4

baza danych: cukiernia

Korzystając z zapytań z zadania 2.4 oraz operatorów UNION, INTERSECT, EXCEPT napisz zapytanie w języku SQL wyświetlające informacje na temat czekoladek (idCzekoladki, nazwa, masa, koszt), których:

1. masa mieści się w przedziale od 15 do 24 g lub koszt produkcji mieści się w przedziale od 15 do 24 gr,

```
(SELECT idczekoladki, nazwa, masa, koszt FROM czekoladki
WHERE masa BETWEEN 15 AND 24)
UNION
(SELECT idczekoladki, nazwa, masa, koszt FROM czekoladki
WHERE koszt BETWEEN 0.15 AND 0.24);
```

2. masa mieści się w przedziale od 25 do 35 g, ale koszt produkcji nie mieści się w przedziale od 25 do 35 gr,

```
(SELECT idczekoladki, nazwa, masa, koszt FROM czekoladki
WHERE masa BETWEEN 25 AND 35)
EXCEPT
(SELECT idczekoladki, nazwa, masa, koszt FROM czekoladki
WHERE koszt BETWEEN 0.25 AND 0.35);
```

3. masa mieści się w przedziale od 15 do 24 g i koszt produkcji mieści się w przedziale od 15 do 24 gr lub masa mieści się w przedziale od 25 do 35 g i koszt produkcji mieści się w przedziale od 25 do 35 gr,

```
((SELECT idczekoladki, nazwa, masa, koszt FROM czekoladki
WHERE masa BETWEEN 15 AND 24)
INTERSECT
(SELECT idczekoladki, nazwa, masa, koszt FROM czekoladki
WHERE koszt BETWEEN 0.15 AND 0.24))
UNION
((SELECT idczekoladki, nazwa, masa, koszt FROM czekoladki
WHERE masa BETWEEN 25 AND 35)
INTERSECT
(SELECT idczekoladki, nazwa, masa, koszt FROM czekoladki
WHERE koszt BETWEEN 0.25 AND 0.35));
```

4. * masa mieści się w przedziale od 15 do 24 g i koszt produkcji mieści się w przedziale od 15 do 24 gr,

```
(SELECT idczekoladki, nazwa, masa, koszt FROM czekoladki
WHERE masa BETWEEN 15 AND 24)
INTERSECT
(SELECT idczekoladki, nazwa, masa, koszt FROM czekoladki
WHERE koszt BETWEEN 0.15 AND 0.24);
```

5. * masa mieści się w przedziale od 25 do 35 g, ale koszt produkcji nie mieści się ani w przedziale od 15 do 24 gr, ani w przedziale od 29 do 35 gr.

```
(SELECT idczekoladki, nazwa, masa, koszt FROM czekoladki
WHERE masa BETWEEN 25 AND 35)
EXCEPT
((SELECT idczekoladki, nazwa, masa, koszt FROM czekoladki
WHERE koszt BETWEEN 0.15 AND 0.24)
UNION
(SELECT idczekoladki, nazwa, masa, koszt FROM czekoladki
WHERE koszt BETWEEN 0.29 AND 0.35));
```

Zadanie 3.5

baza danych: cukiernia

Korzystając z operatorów UNION, INTERSECT, EXCEPT napisz zapytanie w języku SQL wyświetlające:

1. identyfikatory klientów, którzy nigdy nie złożyli żadnego zamówienia,

```

SELECT idklienta FROM klienci
EXCEPT
SELECT idklienta FROM zamowienia;

```

2. identyfikatory pudełek, które nigdy nie zostały zamówione,

```

SELECT idpudelka FROM pudelka
EXCEPT
SELECT idpudelka FROM artykuły;

```

3. ★ nazwy klientów, czekoladek i pudełek, które zawierają rz (lub Rz),

```

SELECT nazwa FROM klienci
WHERE nazwa ~ '(Rz|rz)'
UNION
SELECT nazwa FROM czekoladki
WHERE nazwa ~ '(Rz|rz)'
UNION
SELECT nazwa FROM pudelka
WHERE nazwa ~ '(Rz|rz)';

```

4. ★ identyfikatory czekoladek, które nie występują w żadnym pudełku.

```

SELECT idczekoladki FROM czekoladki
EXCEPT
SELECT idczekoladki FROM zawartosc;

```

Zadanie 3.6

baza danych: siatkówka

Napisz zapytanie w języku SQL wyświetlające:

1. identyfikator meczu, sumę punktów zdobytych przez gospodarzy i sumę punktów zdobytych przez gości,

```

SELECT
    idmeczu,
    (SELECT SUM(wyniki) FROM UNNEST(gospodarze) AS wyniki)
    AS gospodarze,
    (SELECT SUM(wyniki) FROM UNNEST(goscie) AS wyniki)
    AS goscie
FROM statystyki;

```

2. identyfikator meczu, sumę punktów zdobytych przez gospodarzy i sumę punktów zdobytych przez gości, dla meczów, które skończyły się po 5 setach i zwycięzca ostatniego seta zdobył ponad 15 punktów,

```

SELECT
    idmeczu,
    (SELECT SUM(wyniki) FROM UNNEST(gospodarze) AS wyniki)
    AS gospodarze,
    (SELECT SUM(wyniki) FROM UNNEST(goscie) AS wyniki)
    AS goscie
FROM siatkowka.statystyki
WHERE gospodarze[5] IS NOT NULL
    AND (gospodarze[5] > 15
        OR goscie[5] > 15);

```

3. identyfikator i wynik meczu w formacie x:y, np. 3:1 (wynik jest pojedynczą kolumną – napisem),

```
SELECT
    idmecz,
    CONCAT (
        CASE WHEN gospodarze[1] > goscie[1] THEN 1 ELSE 0 END
        + CASE WHEN gospodarze[2] > goscie[2] THEN 1 ELSE 0 END
        + CASE WHEN gospodarze[3] > goscie[3] THEN 1 ELSE 0 END
        + CASE WHEN COALESCE(gospodarze[4], 0) > COALESCE(goscie[4], 0)
            THEN 1 ELSE 0 END
        + CASE WHEN COALESCE(gospodarze[5], 0) > COALESCE(goscie[5], 0)
            THEN 1 ELSE 0 END
        , ': ',
        CASE WHEN goscie[1] > gospodarze[1] THEN 1 ELSE 0 END
        + CASE WHEN goscie[2] > gospodarze[2] THEN 1 ELSE 0 END
        + CASE WHEN goscie[3] > gospodarze[3] THEN 1 ELSE 0 END
        + CASE WHEN COALESCE(goscie[4], 0) > COALESCE(gospodarze[4], 0)
            THEN 1 ELSE 0 END
        + CASE WHEN COALESCE(goscie[5], 0) > COALESCE(gospodarze[5], 0)
            THEN 1 ELSE 0 END
    ) AS wynik
FROM siatkowka.statystyki;
```

4. * identyfikator meczu, sumę punktów zdobytych przez gospodarzy i sumę punktów zdobytych przez gości, dla meczów, w których gospodarze zdobyli ponad 100 punktów,

```
SELECT
    idmecz,
    (SELECT SUM(wynik) FROM UNNEST(gospodarze) AS wynik)
    AS gospodarze,
    (SELECT SUM(wynik) FROM UNNEST(goscie) AS wynik)
    AS goscie
FROM statystyki
WHERE (SELECT SUM(wynik) FROM UNNEST(gospodarze)
        AS wynik) > 100;

SELECT * FROM (
    SELECT
        idmecz,
        (SELECT SUM(wyniki) FROM UNNEST(gospodarze) AS wyniki)
        AS gospodarze,
        (SELECT SUM(wyniki) FROM UNNEST(goscie) AS wyniki)
        AS goscie
        FROM siatkowka.statystyki
    ) AS zap
    WHERE gospodarze > 100;
```

5. * identyfikator meczu, liczbę punktów zdobytych przez gospodarzy w pierwszym secie, sumę punktów zdobytych w meczu przez gospodarzy, dla meczów, w których pierwiastek kwadratowy z liczby punktów zdobytych w pierwszym secie jest mniejszy niż logarytm o podstawie 2 z sumy punktów zdobytych w meczu. ;)

```
SELECT gospodarze1, gospodarze FROM (
    SELECT
        idmecz,
        gospodarze[1] AS gospodarze1,
```

```

        (SELECT sum(wyniki) FROM UNNEST(gospodarze) AS wyniki)
        AS gospodarze,
        (SELECT sum(wyniki) FROM UNNEST(goscie) AS wyniki)
        AS goscie
FROM siatkowka.statystyki
) AS zap
WHERE SQRT(gospodarzel) < LOG(2, gospodarze + goscie);

```

Zadanie 3.7

★ Napisz skrypt składający się z poleceń psql, który wykona zapytanie (użyj dowolnego zapytania z zadania 3.6) oraz zwróci jego wynik jako dokument HTML (nie zapomnij o znacznikach html, body itd.), gdzie odpowiedź serwera będzie tabelą HTML.

UWAGA: Przydatne polecenia: \H \echo \T \pset

psql ... < zapytanie.sql > wynik.html

```

\H \echo \T
SELECT * FROM (
    SELECT
        idmeczu,
        gospodarze[1] AS gospodarzel,
        (SELECT sum(wyniki) FROM UNNEST(gospodarze) AS wyniki)
        AS gospodarze,
        (SELECT sum(wyniki) FROM UNNEST(goscie) AS wyniki)
        AS goscie
    FROM siatkowka.statystyki
) AS zap
WHERE SQRT(gospodarzel) < LOG(2, gospodarze + goscie);

```

Zadanie 3.8

★ Napisz skrypt składający się z poleceń psql, który wykona zapytanie (użyj dowolnego zapytania z zadania 3.6, ale innego niż w zadaniu 3.7) oraz zwróci jego wynik jako dokument tekstowy z polami oddzielonymi przecinkami (Comma Separated Values), z jednym rekordem w jednej linii.

UWAGA: Przydatne polecenia: \a \pset fieldsep \t

psql ... < zapytanie.sql > wynik.txt

```

\a \pset fieldsep ,
SELECT
    idmeczu,
    (SELECT SUM(wyniki) FROM UNNEST(gospodarze) AS wyniki)
    AS gospodarze,
    (SELECT SUM(wyniki) FROM UNNEST(goscie) AS wyniki)
    AS goscie
FROM statystyki;

```

©Marcin Sawczuk