

Laboratorium 4

Zapytania oparte na wielu tabelach.

Zadanie 4.1

baza danych: cukiernia

Porównaj wyniki poniższych zapytań:

```
SELECT k.nazwa FROM klienci k;
```

```
SELECT k.nazwa, z.idzamowienia FROM klienci k, zamowienia z;
```

```
SELECT k.nazwa, z.idzamowienia FROM klienci k, zamowienia z  
WHERE z.idklienta = k.idklienta;
```

```
SELECT k.nazwa, z.idzamowienia FROM klienci k NATURAL JOIN zamowienia z;
```

```
SELECT k.nazwa, z.idzamowienia FROM klienci k JOIN zamowienia z  
ON z.idklienta=k.idklienta;
```

```
SELECT k.nazwa, z.idzamowienia FROM klienci k JOIN zamowienia z  
USING (idklienta);
```

1. W którym zapytaniu występuje iloczyn kartezjański?
2. Które zapytanie dostarcza bezwartościowych danych?

Zadanie 4.2

baza danych: cukiernia

Napisz zapytanie w języku SQL wyświetlające informacje na temat zamówień (data realizacji, idzamowienia), które:

1. zostały złożone przez klienta, który ma na imię Antoni,

```
SELECT k.nazwa, z.datarealizacji, z.idzamowienia  
FROM klienci k INNER JOIN zamowienia z USING(idklienta)  
WHERE k.nazwa ~'.*Antoni';
```

2. zostały złożone przez klientów z mieszkań (zwróć uwagę na pole ulica),

```
SELECT k.ulica, z.datarealizacji, z.idzamowienia  
FROM klienci k INNER JOIN zamowienia z USING(idklienta)  
WHERE k.ulica ~ '.*/*.*';
```

3. ★ zostały złożone przez klienta z Krakowa do realizacji w listopadzie 2013 roku.

```
SELECT k.miejscowosc, z.datarealizacji, z.idzamowienia  
FROM klienci k INNER JOIN zamowienia z USING(idklienta)  
WHERE k.miejscowosc = 'Krakow'  
AND DATE_PART('year', z.datarealizacji) = 2013  
AND DATE_PART('month', z.datarealizacji) = 11;
```

W każdym zapytaniu należy dodać dodatkowe pola (poza informacjami o zamówieniu), które pozwolą sprawdzić, czy wynik jest poprawny np. nazwa klienta, ulica etc.

Zadanie 4.3

baza danych: cukiernia

Napisz zapytanie w języku SQL wyświetlające informacje na temat klientów (nazwa, ulica, miejscowość), którzy:

1. złożyli zamówienia z datą realizacji nie starszą niż sprzed pięciu lat,

```
SELECT k.nazwa, k.ulica, k.miejscowosc
FROM klienci k INNER JOIN zamowienia z USING(idklienta)
WHERE DATE_PART('year', CURRENT_DATE) -
      DATE_PART('year', z.datarealizacji) < 5;
```

2. zamówili pudełko Kremowa fantazja lub Kolekcja jesienna,

```
SELECT k.nazwa, k.ulica, k.miejscowosc
FROM klienci k INNER JOIN zamowienia z USING(idklienta)
INNER JOIN artykuly a USING(idzamowienia)
INNER JOIN pudelka p USING(idpudelka)
WHERE p.nazwa IN ('Kremowa_fantazja',
                  'Kolekcja_jesienna');
```

3. złożyli przynajmniej jedno zamówienie,

```
SELECT DISTINCT k.nazwa, k.ulica, k.miejscowosc
FROM klienci k LEFT JOIN zamowienia z USING(idklienta)
WHERE z.idzamowienia IS NOT NULL;

SELECT nazwa, ulica, miejscowosc FROM klienci
WHERE idklienta IN
      (SELECT idklienta FROM zamowienia);
```

4. nie złożyli żadnych zamówień,

```
SELECT nazwa, ulica, miejscowosc FROM klienci
WHERE idklienta NOT IN
      (SELECT idklienta FROM zamowienia);

SELECT DISTINCT k.nazwa, k.ulica, k.miejscowosc
FROM klienci k LEFT JOIN zamowienia z USING(idklienta)
WHERE z.idzamowienia IS NULL;
```

5. ★ złożyli zamówienia z datą realizacji w listopadzie 2013,

```
SELECT k.nazwa, k.ulica, k.miejscowosc
FROM klienci k INNER JOIN zamowienia z USING(idklienta)
WHERE DATE_PART('year', z.datarealizacji) = 2013
      AND DATE_PART('month', z.datarealizacji) = 11;
```

6. ★ zamówili co najmniej 2 sztuki pudełek Kremowa fantazja lub Kolekcja jesienna w ramach jednego zamówienia,

```
SELECT k.nazwa, k.ulica, k.miejscowosc
FROM klienci k INNER JOIN zamowienia z USING(idklienta)
INNER JOIN artykuly a USING(idzamowienia)
INNER JOIN pudelka p USING(idpudelka)
WHERE a.sztuk >= 2
```

```

AND p.nazwa IN
    ('Kremowa_fantazja',
     'Kolekcja_jesienna');

```

7. ★ zamówili pudełka, które zawierają czekoladki z migdałami.

```

SELECT k.nazwa, k.ulica, k.miejscowosc
FROM klienci k INNER JOIN zamowienia z USING(idklienta)
    INNER JOIN artykuly a USING(idzamowienia)
    INNER JOIN pudełka p USING(idpudełka)
    INNER JOIN zawartosc zw USING(idpudełka)
    INNER JOIN czekoladki c
        USING(idczekoladki)
    WHERE orzechy = 'migdaly';

```

W każdym zapytaniu należy dodać dodatkowe pola (poza danymi klientów), które pozwolą sprawdzić, czy wynik jest poprawny np. data realizacji, nazwa pudełka itd.

Wskazówka: W punkcie 4 można wykorzystać złączenie zewnętrzne.

Zadanie 4.4

baza danych: cukiernia

Napisz zapytanie w języku SQL wyświetlające informacje na temat pudełek i ich zawartości (nazwa, opis, nazwa czekoladki, opis czekoladki):

1. wszystkich pudełek,

```

SELECT p.nazwa, p.opis, c.opis, c.nazwa
FROM pudełka p INNER JOIN zawartosc z USING(idpudełka)
    INNER JOIN czekoladki c USING(idczekoladki);

```

2. pudełka o wartości klucza głównego heav,

```

SELECT p.nazwa, p.opis, c.opis, c.nazwa
FROM pudełka p INNER JOIN zawartosc z USING(idpudełka)
    INNER JOIN czekoladki c USING(idczekoladki)
    WHERE p.idpudełka='heav';

```

3. ★ pudełek, których nazwa zawiera słowo Kolekcja.

```

SELECT p.nazwa, p.opis, c.opis, c.nazwa
FROM pudełka p INNER JOIN zawartosc z USING(idpudełka)
    INNER JOIN czekoladki c USING(idczekoladki)
    WHERE p.nazwa ~ '(K|k)olekcja';

```

W każdym zapytaniu należy dodać dodatkowe pola (poza danymi o pudełkach), które pozwolą sprawdzić, czy wynik jest poprawny.

Zadanie 4.5

baza danych: cukiernia

Napisz zapytanie w języku SQL wyświetlające informacje na temat pudełek z czekoladkami (nazwa, opis, cena), które (uwaga: może być konieczne użycie konstrukcji z poprzednich laboratoriów):

1. zawierają czekoladki o wartości klucza głównego d09

```
SELECT p.nazwa, p.opis, p.cena
FROM pudelka p INNER JOIN zawartosc z USING(idpudelka)
WHERE z.idczekoladki = 'd09';
```

2. zawierają przynajmniej jedną czekoladkę, której nazwa zaczyna się na S,

```
SELECT DISTINCT p.nazwa, p.opis, p.cena, c.nazwa
FROM pudelka p INNER JOIN zawartosc z USING(idpudelka)
INNER JOIN czekoladki c USING(idczekoladki)
WHERE c.nazwa ~ '^S';
```

3. zawierają przynajmniej 4 sztuki czekoladek jednego gatunku (o takim samym kluczu głównym),

```
SELECT DISTINCT p.nazwa, p.opis, p.cena, c.nazwa
FROM pudelka p INNER JOIN zawartosc z USING(idpudelka)
INNER JOIN czekoladki c USING(idczekoladki)
WHERE z.sztuk>4;
```

4. zawierają czekoladki z nadzieniem truskawkowym,

```
SELECT DISTINCT p.nazwa, p.opis, p.cena, c.nazwa
FROM pudelka p INNER JOIN zawartosc z USING(idpudelka)
INNER JOIN czekoladki c USING(idczekoladki)
WHERE c.nadzienie = 'truskawki';
```

5. nie zawierają czekoladek w gorzkiej czekoladzie,

```
SELECT DISTINCT p.nazwa, p.opis, p.cena, c.nazwa
FROM pudelka p INNER JOIN zawartosc z USING(idpudelka)
INNER JOIN czekoladki c USING(idczekoladki)
WHERE c.czekolada != 'gorzka';

SELECT DISTINCT p.nazwa, p.opis, p.cena, c.nazwa FROM pudelka p
INNER JOIN zawartosc z ON z.idpudelka = p.idpudelka
INNER JOIN czekoladki c ON c.idczekoladki = z.idczekoladki
EXCEPT
(SELECT DISTINCT p.nazwa, p.opis, p.cena, c.nazwa FROM pudelka p
INNER JOIN zawartosc z ON z.idpudelka = p.idpudelka
INNER JOIN czekoladki c ON c.idczekoladki = z.idczekoladki
WHERE c.czekolada = 'gorzka');
```

6. ★ zawierają co najmniej 3 sztuki czekoladki Gorzka truskawkowa,

```
SELECT DISTINCT p.nazwa, p.opis, p.cena, c.nazwa
FROM pudelka p INNER JOIN zawartosc z USING(idpudelka)
INNER JOIN czekoladki c USING(idczekoladki)
WHERE c.nazwa = 'Gorzka_truskawkowa'
AND z.sztuk>3;
```

7. ★ nie zawierają czekoladek z orzechami,

```
SELECT DISTINCT p.nazwa, p.opis, p.cena
FROM pudelka p INNER JOIN zawartosc z USING(idpudelka)
INNER JOIN czekoladki c USING(idczekoladki)
WHERE c.orzechy IS NULL;
```

8. ★ zawierają czekoladki Gorzka truskawkowa,

```
SELECT DISTINCT p.nazwa, p.opis, p.cena, c.nazwa
FROM pudelka p INNER JOIN zawartosc z USING(idpudelka)
INNER JOIN czekoladki c USING(idczekoladki)
WHERE c.nazwa = 'Gorzka_truskawkowa';
```

9. ★ zawierają przynajmniej jedną czekoladkę bez nadzienia.

```
SELECT DISTINCT p.nazwa, p.opis, p.cena, c.nazwa
FROM pudelka p INNER JOIN zawartosc z USING(idpudelka)
INNER JOIN czekoladki c USING(idczekoladki)
WHERE c.nadzienie IS NULL;
```

W każdym zapytaniu należy dodać dodatkowe pola (poza danymi o pudełkach), które pozwolą sprawdzić, czy wynik jest poprawny.

Uwaga: Zdania: “Pudełko nie zawiera czekoladek w gorzkiej czekoladzie.” i “Pudełko zawiera jakieś czekoladki, które nie są w gorzkiej czekoladzie” nie oznaczają tego samego ;)

Wskazówka: W punktach 5 i 7 można wykorzystać operator EXCEPT.

Zadanie 4.6

baza danych: cukiernia

Napisz poniższe zapytania w języku SQL:

1. Wyświetl wartości kluczy głównych oraz nazwy czekoladek, których koszt jest większy od kosztu czekoladki o wartości klucza głównego równej d08.

```
SELECT c.idczekoladki, c.nazwa FROM czekoladki c
WHERE c.koszt > (SELECT koszt FROM czekoladki
WHERE idczekoladki = 'd08');
```

2. Kto (nazwa klienta) złożył zamówienia na takie same czekoladki (pudełka) jak zamawiała Górka Alicja.

```
SELECT kk.nazwa
FROM
    klienci kk
    INNER JOIN zamowienia zz ON kk.idklienta = zz.idklienta
    INNER JOIN artykuly aa ON zz.idzamowienia = aa.idzamowienia
    INNER JOIN (
        SELECT a.idpudelka
        FROM
            klienci k
            INNER JOIN zamowienia z ON k.idklienta = z.idklienta
            INNER JOIN artykuly a ON z.idzamowienia = a.idzamowienia
        WHERE k.nazwa = 'Gorka_Alicja'
    ) gorka ON aa.idpudelka = gorka.idpudelka
WHERE kk.nazwa <> 'Gorka_Alicja'
GROUP BY kk.nazwa
ORDER BY kk.nazwa ASC;
```

3. ★ Kto (nazwa klienta, adres) złożył zamówienia na takie same czekoladki (pudełka) jak zamawiali klienci z Katowic.

```

SELECT kk.nazwa, kk.ulica, kk.miejscowosc
FROM
    klienci kk
    INNER JOIN zamowienia zz ON kk.idklienta = zz.idklienta
    INNER JOIN artykuly aa ON zz.idzamowienia = aa.idzamowienia
    INNER JOIN (
        SELECT a.idpudelka
        FROM
            klienci k
            INNER JOIN zamowienia z ON k.idklienta = z.idklienta
            INNER JOIN artykuly a ON z.idzamowienia = a.idzamowienia
        WHERE k.miejscowosc = 'Katowice'
    ) keto ON aa.idpudelka = keto.idpudelka
WHERE kk.miejscowosc <> 'Katowice'
GROUP BY kk.nazwa, kk.ulica, kk.miejscowosc
ORDER BY kk.nazwa ASC;

```

Uwaga: w powyższych zapytaniach należy użyć **samołączeń (złączeń własnych)**.

Wskazówka: Zapytanie w punkcie 2 można znacznie uprościć stosując konstrukcję z WITH.

Uwagi ogólne

Błąd złączenia Wykorzystując USING można doświadczyć błędu:

ERROR: common column name "idczekoladki" appears more than once in left table Jest spowodowany wcześniejszym złym złączeniem, nie uwzględniającym klucza złożonego (na wielu kolumnach). Rezultatem jest pojawienie się kolumny o tej samej nazwie więcej niż raz po lewej stronie nowego złączenia z czym USING nie ma jak sobie poradzić. Przydatne w znalezieniu błędu jest użycie SELECT * ... i weryfikacja jakie kolumny są zwracane przez zapytanie.

©Marcin Sawczuk