Bazy danych 2022: rozwiązania zadań z listy nr 2

Bartosz Brzostowski

11 kwietnia 2022

Często istnieje istnieje kilka istotnie różnych, równie dobrych rozwiązań tego samego zadania. Poniższe rozwiązania są przykładowe. Jeżeli jednak Twoje rozwiązanie bardzo różni się od przykładowego, a zwłaszcza jeśli daje różne wyniki, zastanów się, na czym polega różnica i ew. skonsultuj się z prowadzącym.

Podane rozwiązania nie wykorzystują agregacji ani podzapytań – dla wielu zadań istnieją alternatywne rozwiązania wykorzystujące podzapytania właśnie. W kilku rozwiązaniach pojawia się grupowanie, ale w dosyć specyficzny sposób, i nie jest ono wymagane.

Zadanie 1

```
SELECT * FROM klienci JOIN produkty;
```

Zadanie 2

```
SELECT zamow.*, sztuk FROM zamow JOIN detal_zamow ON idz = z_id;
```

Zadanie 3

```
SELECT nazwa, sztuk FROM produkty JOIN detal_zamow ON idp = p_id;
```

Wiersze mogą się powtarzać – jeśli jakiś produkt został dwukrotnie zamówiony w tej samej liczbie sztuk, nie ma powodu, żeby takie wiersze „sklejać”. Dlatego użycie `SELECT DISTINCT` nie jest tu uzasadnione.

Zadanie 4

```
SELECT cena * sztuk AS wartosc
  FROM detal_zamow
    JOIN produkty on idp = p_id
  ORDER BY 1 DESC;
```

Zadanie 5

```
SELECT sztuk
  FROM zamow
    JOIN detal_zamow ON idz = z_id
  WHERE TIME(data) > "12:00:00";
```

Zadanie 6

```
SELECT DISTINCT MONTHNAME(data)
  FROM klienci
    JOIN zamow ON idk = k_id
  WHERE miasto != "Wrocław";
```

Tutaj `SELECT DISTINCT` jest właściwe – pytani jesteśmy o miesiące spełniające pewien warunek, bez powtórzeń.

Zadanie 7

```
SELECT DISTINCT cena
  FROM produkty
    JOIN detal_zamow ON idp = p_id
    JOIN zamow ON idz = z_id
  WHERE WEEKDAY(data) = 4;
```

Zadanie 8

```
SELECT nazwa
  FROM klienci
    JOIN zamow ON idk = k_id
    JOIN detal_zamow ON idz = z_id
  WHERE sztuk > 4
  -- GROUP BY idk
  ORDER BY miasto DESC;
```

Tutaj chciałoby się zrobić `SELECT DISTINCT`, ale dostaniemy błąd – jest to niekompatybilne z `ORDER BY`. To dlatego, że teoretycznie mogą istnieć różni klienci (z różnych miast) o tej samej nazwie, która wtedy nie miałaby jednoznacznej pozycji w sortowaniu. Pożądany efekt można uzyskać przez grupowanie (tj. odkomentowując klauzulę `GROUP BY` w ww. przykładzie) – to jest już poprawne i da pożądany efekt. To samo stosuje się do dwóch kolejnych zadań.

Zauważmy też, że nie ma sensu robić złączeń zewnętrznych – wiersze z NULLami w kolumnie `sztuk` i tak nie przetrwają filtrowania `WHERE`.

Zadanie 9

```
SELECT adres
  FROM klienci
    JOIN zamow ON idk = k_id
    JOIN detal_zamow ON idz = z_id
    JOIN produkty ON idp = p_id
  WHERE produkty.nazwa LIKE "%laptop%"
  -- GROUP BY idk
  ORDER BY REVERSE(klienci.nazwa);
```

Zadanie 10

```
SELECT produkty.nazwa
  FROM produkty
    JOIN detal_zamow ON idp = p_id
    JOIN zamow ON idz = z_id
    JOIN klienci ON idk = k_id
  WHERE telefon LIKE "%4%"
  -- GROUP BY idp
  ORDER BY cena;
```

Tutaj opcjonalne grupowanie jest po ID produktu, nie klienta – bo zarówno wypisywane kolumny, jak i kryterium sortowania, to atrybuty tabeli `produkty`.

Zadanie 11

To zadanie było po to, żeby mieli Państwo jakąś styczność z nieco rzadziej używaną składnią `NATURAL JOIN` lub `USING`. Dwa rozwiązania:

```
SELECT * FROM klienci NATURAL JOIN produkty
SELECT * FROM klienci JOIN produkty USING (nazwa);
```

To drugie jest „poprawniejsze” w tym sensie, że spełniałoby założenia także w sytuacji, gdyby w schemacie tych tabel były jeszcze jakieś inne niż `nazwa` kolumny o powtarzających się identyfikatorach.

Zadanie 12

W tym i kolejnych zadaniach kluczowe jest użycie złączeń zewnętrznych: tutaj mamy wprost w treści zadania napisane, że w wyniku mają pojawić się wszyscy klienci, tj. nawet ci, którzy nie przetrwaliby złączenia wewnętrznego.

```
SELECT DISTINCT nazwa, DATE(data)
  FROM klienci
    LEFT JOIN zamow ON idk = k_id
  ORDER BY 2;
```

Zadanie 13

```
SELECT DISTINCT produkty.nazwa, miasto
  FROM klienci
    JOIN zamow ON idk = k_id
    JOIN detal_zamow ON idz = z_id
    RIGHT JOIN produkty ON idp = p_id;
```

Zauważmy, że tylko jedno złączenie musi być zewnętrzne. Ale uwaga: gdyby wypisywać tabele w odwrotnym porządku, to zewnętrzne (i oczywiście lewe) musiałyby być wszystkie – chyba, że użyć nawiasowania...

Zadanie 14

W tym i dwóch kolejnych zadaniach złączenie zewnętrzne służy do tego, żeby móc wyfiltrować pożądane wiersze patrząc na NULLe w dołączanych kolumnach.

```
SELECT nazwa
  FROM produkty
    LEFT JOIN detal_zamow ON idp = p_id
  WHERE idd IS NULL;
```

Zadanie 15

Gdyby założyć, że nie ma „pustych" zamówień, tj. takich, dla których nie istnieją żadne detale, wyglądałoby to tak:

```
SELECT nazwa
  FROM klienci
    LEFT JOIN zamow ON idk = k_id
  WHERE idz IS NULL;
```

Natomiast ktoś, kto ma zamówienia, ale wyłącznie puste, nie pojawi się w wyniku. W tym celu należy dołączyć jeszcze tabelę detali zamówień i sprawdzać, czy ID detalu (a nie ID zamówienia) jest NULLem.

```
SELECT nazwa
  FROM klienci
    LEFT JOIN zamow ON idk = k_id
    LEFT JOIN detal_zamow ON idz = z_id
  WHERE idd IS NULL;
```

Zadanie 16

```
SELECT * FROM zamow LEFT JOIN detal_zamow ON idz = z_id WHERE idd IS NULL;
```

Zadanie 17

Po wykładzie nr 3 najbardziej intuicyjnie byłoby użyć agregacji (`MAX` i `COUNT`) oraz podzapytań. Tym niemniej, zastanówmy się, co można osiągnąć korzystając tylko ze złączeń. Popatrzmy na takie zapytanie:

```
SELECT p1.*
  FROM produkty AS p1
    LEFT JOIN produkty AS p2 ON p1.ilosc < p2.ilosc
  WHERE p2.idp IS NULL;
```

Jest ono o tyle nietypowe, że zawiera inny niż równościowy warunek złączenia. Oznacza on tyle, że produkty z `p1` połączą się z droższymi produktami z `p2`. Ponieważ złączenie jest zewnętrzne, to te, które nie mają się z czym połączyć (a więc te najdroższe) przeżyją, i w „nowych" kolumnach będą miały NULLe. Dokładnie te produkty przeżywają też filtrowanie, a więc wynikiem zapytania są wszystkie informacje (`p1.*`) o najdroższych produktach.

Możemy więc dokonać złączenia z tabelą detali zamówień (dalej bez podzapytań!) i dostać detale zamówień dotyczących wyłącznie najdroższych produktów:

```
SELECT DISTINCT z_id
  FROM produkty AS p1
    LEFT JOIN produkty AS p2 ON p1.ilosc < p2.ilosc
    LEFT JOIN detal_zamow ON p1.idp = p_id
  WHERE p2.idp IS NULL;
```

W stanie bazy, w którym jest tylko jeden najdroższy produkt, jedyne, czego jeszcze (tj. spoza wykładów 0–1) potrzebujemy do uzyskania wyników, to `COUNT()`; jeśli jest ich więcej, musimy jeszcze użyć grupowania (a poprawną odpowiedzią będzie zbiór liczb wystąpień).