

Sprawdzian z SQL nr 2

28.04.2022

Dla każdego z poniższych zadań napisz odpowiednie polecenia SQL. Oczekujemy rozwiązania w postaci pliku zawierającego **treści** poleceń SQL, a nie znalezionej odpowiedzi. Nie będą sprawdzane jakiegokolwiek zapytania niepoprawne składniowo — sprawdź swoje rozwiązanie używając `\i plik.sql` ! Plik możesz wysyłać wielokrotnie, sprawdzana będzie wyłącznie najnowsza wersja.

Wczytaj do swojej bazy danych plik `tpch-keys.sql`. Rozwiązania wysyłaj przez formularz pod adresem <https://hera.stud.ii:2015>. Rób to jak najczęściej! Wszystkie dane na Waszych komputerach są kasowane po restarcie.

Format pierwszej linijki rozwiązania: `-- grupa-imie-nazwisko`, gdzie grupa to inicjały prowadzącego Twoją grupę (`pwi/jotop/plg/mabi/akr/rfe`), np. `pwi-Jan-Kowalski`. Wymagany format całego pliku z rozwiązaniem:

```
-- grupa-imie-nazwisko
-- Zadanie 1
<zapytanie>

-- Zadanie 2
<zapytanie>
...
```

Zadanie 1 (2 pkt.) Wypisz nazwy klientów z Europy wraz z informacją o sumarycznej wartości ich wszystkich zamówień złożonych w roku 1997 (wg `totalprice`). Uwzględnij jedynie klientów z wartością zamówień powyżej 500 000 a wyniki uporządkuj malejąco według nazw klientów.

Rozwiązanie

```
SELECT c_name, SUM(o_totalprice)
FROM customer
  JOIN orders ON o_custkey = c_custkey
  JOIN nation ON n_nationkey = c_nationkey
  JOIN region ON n_regionkey = r_regionkey
WHERE EXTRACT('YEAR' FROM o_orderdate) = 1997
  AND r_name = 'EUROPE'
GROUP BY c_custkey
HAVING SUM(o_totalprice) > 500000
ORDER BY 1 DESC;
```

Zapytanie wzorcowe zwraca 456 krotek.

Zadanie 2 (3 pkt.) Dla każdego kraju wypisz jego nazwę, nazwy klientów z tego kraju o najwyższej sumie wartości zamówień złożonych w roku 1997 oraz tę sumę. Gdyby nie było w kraju takich klientów, krotka powinna zawierać `NULL`.

Rozwiązanie

```
WITH ordertotals AS (  
    SELECT c_nationkey, c_name, SUM(o_totalprice) AS total  
    FROM customer JOIN orders ON c_custkey = o_custkey  
    WHERE EXTRACT('YEAR' FROM o_orderdate) = 1997  
    GROUP BY c_custkey  
)  
SELECT n_name, c_name, total  
FROM (SELECT c_nationkey, MAX(total) AS total  
    FROM ordertotals  
    GROUP BY c_nationkey  
    ) AS foo  
NATURAL JOIN ordertotals  
RIGHT JOIN nation ON c_nationkey = n_nationkey
```

Zapytanie wzorcowe zwraca 25 krotek i używa konstrukcji **WITH**.

Zadanie 3 (3 pkt.) Dla *wszystkich* części sprzedawanych w opakowaniu typu 'JUMBO PKG' wypisz ich nazwy wraz z łączną liczbą sztuk (availqty) dostępnych u dostawców ze Środkowego Wschodu (tabela region). Wynik nie powinien zawierać NULLi – użyj COALESCE, żeby je zastąpić (czym?).

Rozwiązanie

```
SELECT p_name, COALESCE(SUM(ps_availqty), 0)  
FROM (  
    SELECT ps_partkey, ps_availqty  
    FROM region  
    JOIN nation ON r_regionkey = n_regionkey  
    JOIN supplier ON n_nationkey = s_nationkey  
    JOIN partsupp ON s_suppkey = ps_suppkey  
    WHERE r_name = 'MIDDLE EAST'  
    ) AS foo  
RIGHT JOIN part ON ps_partkey = p_partkey  
WHERE p_container = 'JUMBO PKG'  
GROUP BY p_partkey
```

a dla obłąkanych szczególnie zainteresowanych

```
SELECT p_name, COALESCE(SUM(ps_availqty), 0)  
FROM region  
JOIN nation ON r_regionkey = n_regionkey  
JOIN supplier ON n_nationkey = s_nationkey  
JOIN partsupp ON s_suppkey = ps_suppkey
```

```
RIGHT JOIN part ON (ps_partkey = p_partkey
AND r_name = 'MIDDLE EAST') -- sic!
WHERE p_container = 'JUMBO PKG'
GROUP BY p_partkey
```

Zapytanie wzorcowe zwraca 492 krotki.

Zadanie 4 (2 pkt.) Utwórz tabelę `customer_no_orders` i przepisuj tam wszystkich klientów (`custkey`) z tabeli `customer`, którzy nie złożyli żadnego zamówienia (`orders`).

Rozwiązanie

```
CREATE TABLE customer_no_orders (LIKE customer);
INSERT INTO customer_no_orders
SELECT customer.*
FROM customer
LEFT JOIN orders ON c_custkey = o_custkey
WHERE o_orderkey IS NULL;
```

Zapytanie wzorcowe przepisuje 5000 klientów, do utworzenia nowej tabeli używa składni `CREATE TABLE newtable (LIKE oldtable)`.