

Bazy danych 2023: lista zadań nr 3

27 luty 2023

We wszystkich zadaniach zastosuj jakieś „rozsądne” uporządkowanie wyników. Oczywiście nie dotyczy to tych zapytań, których wynikiem zawsze będzie jeden wiersz.

Na tej liście nie używaj podzapytań.

Zad. 1. Jaka jest łączna wartość towaru na stanie sklepu?

Zad. 2. Wypisz nazwy wszystkich klientów w porządku alfabetycznym, oddzielone przecinkiem i spacją (tj. jako pojedynczy napis).

Zad. 3. Ile dni upłynęło pomiędzy pierwszym a ostatnim zamówieniem, w którego skład wchodziły produkty firmy Samsung (tj. zawierające słowo „Samsung” w nazwie)? Ile różnych takich produktów się w nich pojawiało? Ile było takich zamówień? Wszystkie odpowiedzi powinny stanowić osobne kolumny jednego zapytania.

W treści trzech poniższych zadań nie może być mowy o „wszystkich” dniach tygodnia, latach, miesiącach czy „kategoriach”. Przy danym schemacie bazy, jeśli nie istnieje zamówienie z datą w roku np. 2000, to baza w ogóle „nie wie”, że taki rok istnieje. Toteż dopuszczalna jest możliwość, że np. w liście lat i miesięcy będą luki.

Zad. 4. Wypisz dni tygodnia i liczby klientów składających zamówienia w te dni.

Zad. 5. Wypisz lata, miesiące, i łączne wartości zamówień złożonych w tych miesiącach tych lat.

Zad. 6. Niech *kategorią produktu* będzie jego cena zaokrąglona do pełnych tysięcy złotych w górę. Dla każdej (reprezentowanej) kategorii wypisz ją oraz tablicę (*array*) JSON zawierającą nazwy produktów do niej należących.

Zad. 7. Wypisz nazwy produktów, których łączna wartość zamówień przekracza 7000 zł.

Zad. 8. Wypisz nazwy wszystkich klientów, którzy zamówili łącznie więcej niż jedną sztukę produktów o cenie jednostkowej przekraczającej 1800 zł.

W poniższym zadaniu zakładamy dla uproszczenia, że w jednym zamówieniu jest najwyżej jeden detal zamówienia dotyczący danego produktu, czyli tak, jakby w tabeli `detal_zamow` był więz `UNIQUE (z_id, p_id)`.

Zad. 9. Dla każdego produktu, który był zamawiany w piątek, wypisz jego nazwę i średnią liczbę zamawianych jednorazowo egzemplarzy (bez ograniczania się do zamówień z piątku).

Podpowiedź: Oczywiście zamówienia spoza piątku nie mogą zostać odfiltrowane w klauzuli **WHERE**, skoro mają wejść do agregacji. Trzeba w takim razie zagregować informację, czy wśród zamówień w grupie było jakieś z piątku, i użyć jej w klauzuli **HAVING**. Przypomnij sobie boolowskie funkcje agregujące (wspomniane na wykładzie 2).

W kolejnych zadaniach zadbaj o to, by *każdy* byt (produkt, klient, miejscowość) pojawiał się w wyniku. Czy powinny w nich występować **NULL**e? Czym je zastąpić?

Zad. 10. Dla każdego klienta wypisz jego nazwę i liczbę złożonych przez niego zamówień.

Zad. 11. Dla każdego produktu wypisz jego nazwę i łączną liczbę zamówionych egzemplarzy.

Zad. 12. Dla każdej miejscowości wypisz jej nazwę i łączną liczbę sztuk produktów zamówionych z niej.

Zad. 13. Dla każdego klienta wypisz jego nazwę, miasto i łączną wartość złożonych przez niego zamówień.

Zad. 14. Dla każdego zamówienia wypisz jego datę i całkowitą wartość.

Dwa kolejne zadania może być trudno zrobić bez podzapytań – ale się da. Zastanów się, co by się stało, gdyby dla złączenia *zewnętrznego* do jego warunku (**ON**) „dorzucić” (po spójniku **AND**) warunek filtrujący którąś z łączonych tabel.

Zad. 15. Dla każdego klienta wypisz jego nazwę i liczbę różnych zamówionych przez niego produktów o cenie jednostkowej powyżej 1500 zł.

Zad. 16. Dla każdego produktu wypisz jego nazwę i liczbę klientów o siedzibach w miejscowościach na literę „W”, którzy go zamawiali.

Zad. 17. Wypisz komplet informacji o trzech ostatnich złożonych zamówieniach (zakładając, że daty i godziny składania zamówień są unikatowe). Nie używaj klauzuli **LIMIT**.

Podpowiedź: Dla każdego zamówienia możesz zagregować liczbę zamówień złożonych później od niego z odpowiedniego złączenia dwóch „kopii” tabeli **zamow**.

Zad. 18. Rozwiąż zad. 19 z listy 1, pisząc tylko jedno zapytanie. (To jest rozwinięcie poprzedniego zadania, oczywiście dla innych tabel.)