

Oczekujemy rozwiązania w postaci pliku zawierającego TREŚCI poleceń SQL, a nie znalezionej odpowiedzi. Nie będą sprawdzane jakiekolwiek 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 `hsm.dump`. Jest to dump bazy `hsm.stackexchange.org` poświęconej dyskusjom na tematy związane ze historią matematyki i nauki.

Zachęcam do korzystania z dokumentacji PostgreSQL.

Format nazwy pliku z rozwiązaniem: `grupa-imie-nazwisko.sql`, gdzie `grupa` to inicjały prowadzącego Twoją grupę: (`pwi/plg/mpy/rfe/pga`), np. `pwi-Jan-Kowalski.sql`. Wymagany format pliku z rozwiązaniem (tu też podaj swoje imię, nazwisko i grupę):

```
-- Imię Nazwisko, grupa np. Jan Kowalski, pwi
-- Zadanie 1
<zapytanie>                -- zastąp napis '<zapytanie>' swoim zapytaniem :)

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

Zadanie 1 (2 pkt) Na podstawie tabeli `badges` oblicz liczbę postów napisanych przez posiadaczy różnych odznak. Dla każdej nazwy odznaki dany post powinien być policzony tyle razy, ile tego typu odznak posiada jego właściciel (grupowanie domyślnie działa w ten właśnie sposób). Wynikiem zapytania ma być jedna tabela z dwiema kolumnami o nazwach odpowiednio `odznaka` i `liczba`. Wyniki posortuj malejąco względem kolumny `liczba`.

Zadanie 2 (2 pkt) „Słowo” to maksymalny, spójny podciąg znaków alfanumerycznych (`\w`) w tekście komentarza. Znajdź w tekstach komentarzy wszystkie słowa, które składają się z trzech lub więcej znaków i powtarzają się w jakimś komentarzu oddzielone niepustym ciągiem białych znaków (`\s`). Usuń powtórzenia, wynik posortuj alfabetycznie. (Uwaga: ciąg `{the}` nie występuje w żadnym komentarzu w bazie, nie jest nawet słowem zgodnie z powyższą definicją.) *Wskazówka: slajdy do wykładu o triggerach.*

Zadanie 3 (4 pkt)

- (2 pkt) Chcielibyśmy dodać więź klucza obcego tak, aby pole `owneruserid` tabeli `posts` wskazywało na `id` tabeli `users`. Stwórz odpowiedni ciąg poleceń SQL (w tym polecenie utworzenia tego klucza obcego). Polecenia uruchomione w podanej przez Ciebie kolejności na niezmodyfikowanej bazie sprawdzianowej nie powinny zwracać żadnych błędów.
- (2 pkt) Zauważ, że w tabeli `posts` jest wiele więzów z wartością `NULL` na polu `owneruserid`. Rozważ, czy ma sens oznaczenie tego pola jako **NOT NULL**.

Wymień 2 rozwiązania pozwalające na usunięcie wartości **NULL** z tej kolumny w taki sposób, aby nie utracić informacji, które są w bazie. Wskaż krótko wady i zalety Twoich rozwiązań w porównaniu do pozostawienia wartości **NULL**. Która z tych 3 możliwości Twoim zdaniem pasuje tutaj najlepiej?