

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 `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 (1 pkt.)**
- Dodaj do tabeli `comments` kolumnę `lasteditdate` typu `timestamp` z więzmem `NOT NULL` i domyślną wartością `now()`. Wypełnij ją obecnymi wartościami `creationdate`.
 - Utwórz tabelę `commenthistory` z kolumnami `id` `SERIAL PRIMARY KEY`, `commentid` `integer`, `creationdate` `timestamp` oraz `text` `text`.

Zadanie 2 (5 pkt.) Napisz wyzwalacz, który zostanie uruchomiony przy każdej próbie wykonania operacji `UPDATE` na tabeli `comments` i sprawi, że:

- jakiegokolwiek zmiany pola `creationdate` zostaną zignorowane, tzn. po operacji ma pozostać dotychczasowa wartość tego pola,
- w przypadku próby zmiany `id`, `postid` lub `lasteditdate` powinien być zgłoszony błąd za pomocą `RAISE EXCEPTION`,
- jeśli operacja zmienia pole `text` to:
 - w wyniku tej operacji `lasteditdate` ma przyjąć wartość `now()`,
 - do tabeli `commenthistory` zostanie dodana krotka z opisem starej wersji: `commentid` powinna przyjąć wartość `id` zmienianego komentarza, `creationdate` – dotychczasową wartość `lasteditdate`, a `text` – dotychczasową wartość `text`.

Zadanie 3 (2 pkt.) Aplikacja obsługująca forum dodaje komentarze za pomocą polecenia

INSERT INTO

```
comments(id, postid, score, text, creationdate, userid, userdisplayname)  
VALUES (...)
```

nie biorąc pod uwagę powyższych zmian, tj. nie ustawia wartości `lasteditdate`. W efekcie pole to przyjmuje obecnie ustawioną wartość domyślną – `now()`. Okazało się, że czasem powoduje to różnicę pomiędzy wartościami `lasteditdate` oraz `creationdate` (ustalaną po stronie aplikacji), która jest kłopotliwa (podobno *data scientist* się zdenerwował).

W jaki sposób sprawić, aby wartość `lasteditdate` dla każdego nowododanego komentarza początkowo była równa `creationdate`? Oczywiście nie masz dostępu do kodu aplikacji i nie możesz go zmieniać. Napisz odpowiedni kod SQL.