

SQL.

Задание 1.

Есть таблица с сессиями, где есть начало сессии, конец и id игрока. Если между сессиями меньше 5 минут мы считаем их как одну, например

Session_id - 1, player_id - 1, start_time - 2021-01-01 15:21:12, end_time - 2021-01-01 15:31:12

Session_id - 2, player_id - 1, start_time - 2021-01-01 15:34:35, end_time - 2021-01-01 15:55:18

Эти 2 сессии будут считаться как одна с продолжительностью 34 мин 6 сек.

Что необходимо получить:

- Продолжительность всех сессий для игрока в рамках дня в минутах.
- Продолжительность самой короткой и длинной сессии для игрока в рамках дня в минутах.
- Также необходимо вывести ранг игрока в рамках страны. Игроки с самой большой продолжительностью всех сессий будут иметь наивысший ранг.

Пример что должно получиться для US

player_id	country	duration_of_all_sessions	shortest_session	longest_session	rank_of_the_user
4	US	530	10	520	1
5	US	99	40	59	2

Code:

```
CREATE TABLE game_sessions (  
    session_id INT PRIMARY KEY,  
    player_id int not null,  
    country varchar(10) not null,  
    start_time datetime2(0) NOT NULL,  
    end_time datetime2(0) NOT NULL  
);  
  
INSERT INTO game_sessions VALUES  
(1, 1, 'UK', '2021-01-01 00:01:00', '2021-01-01 00:10:00'),  
(2, 1, 'UK', '2021-01-01 00:12:00', '2021-01-01 01:24:00'),  
(3, 1, 'UK', '2021-01-01 10:01:00', '2021-01-01 15:10:00'),  
(4, 2, 'UK', '2021-01-01 05:01:00', '2021-01-01 06:10:00'),  
(5, 2, 'UK', '2021-01-01 06:14:00', '2021-01-01 08:00:00'),  
(6, 3, 'UK', '2021-01-01 15:01:00', '2021-01-01 15:10:00'),  
(7, 3, 'UK', '2021-01-01 15:20:00', '2021-01-01 16:00:00'),  
(8, 3, 'UK', '2021-01-01 16:45:00', '2021-01-01 23:40:00'),  
(9, 4, 'US', '2021-01-01 00:30:00', '2021-01-01 01:10:00'),  
(10, 4, 'US', '2021-01-01 01:12:00', '2021-01-01 06:25:00'),  
(11, 5, 'US', '2021-01-01 02:10:00', '2021-01-01 02:50:00'),  
(12, 5, 'US', '2021-01-01 06:01:00', '2021-01-01 07:00:00'),  
(13, 1, 'UK', '2021-01-01 01:27:00', '2021-01-01 01:50:00'),  
(14, 4, 'US', '2021-01-01 06:27:00', '2021-01-01 09:10:00'),  
(15, 4, 'US', '2021-01-01 10:10:00', '2021-01-01 10:20:00'),  
(16, 1, 'UK', '2021-01-01 17:10:00', '2021-01-01 17:20:00'),  
(17, 1, 'UK', '2021-01-01 17:22:00', '2021-01-01 17:55:00');
```

Задание 2.

Используя набор выше показать нарастающий итог продолжительности сессий для игрока в минутах. В данном случае правило 5 минут не учитывается.

К примеру строка с id = 1 будет иметь значение 9, строка с id = 2 значение 81 (72 + 9) и т.д.

Python.

Задание 1.

Есть сайт с курсами валют, <https://fixer.io>, на сайте можно зарегистрироваться и получить бесплатный токен.

Необходимо написать скрипт на Python который делает запрос к API, получает курс валют для пары USD \ EUR за вчера, записывает результат в БД (можно установить postgres локально, можно представить что она уже есть и написать код для вставки).

Если API возвращает ошибку то текст ошибки пишем в файл логов.

Т.е. на выходе у нас должен получится фактически рабочий ETL. Мы берем данные с API, перед вставкой в БД проверяем есть ли уже курс для этой пары на этот день, если есть (возможно по ошибке запустили скрипт 2й раз) то ничего не делаем, если нет то делаем вставку, в случае ошибок пишем текст в файл логов, это может быть обычный txt файл.

Задание 2.

Его необходимо решить также используя Python, но не устанавливая и не используя никаких библиотек.

Дан список в котором хранятся списки id пользователей и их друзей, `[[2,3],[3,4],[5],[2,6]]` к примеру пользователь с id = 2 дружит с пользователем с id = 3, а пользователь id = 5 не имеет друзей совсем. Также один пользователь может иметь несколько друзей. Необходимо написать скрипт который считает кол=во друзей для каждого пользователя.

Также необходимо помнить что мы считаем связи не только «слева направо», т.е. `[[2,5],[3,2]]` будет означать что у пользователя id = 2 - 2 друга, у пользователя с id = 5 и id = 3 – 1 друг.

```
a = [[2,3],[3,4],[5],[2,6],[2,4],[6,1]]
```

```
b = [[1,2],[3],[8],[6,2]]
```