

# SQL Homework

First.fm

Вам представляется возможность поработать с базой данных вымышленного музыкального сервиса First.fm. внутри неё будет несколько таблиц: Users, Artists, Albums, Songs, Listened.

Таблица Users содержит в себе информацию о пользователях данного сервиса:

```
create table Users (  
    id integer primary key autoincrement, -- user id  
    username text unique not null, -- username,  
    registered datetime not null -- when user  
                                -- registered  
);
```

Таблица Artists содержит информацию об исполнителях:

```
create table Artists (  
    id integer primary key autoincrement, -- artist id  
    name text not null -- name  
);
```

Таблица Albums содержит информацию об альбомах:

```
create table Albums (  
    id integer primary key autoincrement, -- album id  
    artist_id integer not null, -- artist who recorded  
    name text not null, -- albums title  
    release_year int -- year of release  
);
```

```
sqlite> select * from Albums limit 3;  
1|1|Forgotten Gods|1992  
2|1|The Flayed God|1994  
3|1|Rituales Prehispanicos|1997
```

Колонка release\_year может оказаться пустой.

Таблица Songs описывает, какие песни входят в альбомы:

```
create table Songs (  
    id integer primary key autoincrement, -- song id  
    album_id integer not null, -- album id  
    name text not null, -- songs title  
    position text, -- position inside album  
    duration integer -- duration in seconds  
);
```

```
sqlite> select * from songs order by random() limit 3;  
4682|316|Kockar|B1|  
5019|344|Sunrise|1|353  
826|39|Scene 14: Booking Hall, Piccadilly|5.35|14
```

В нашей базе может быть не указана длительность песни. **Поэтому вместо функции sum лучше использовать функцию *total* – она посчитает пустые поля за 0.**

Таблица Listened описывает прослушанные пользователями треки:

```
create table Listened (  
    user_id integer not null, -- user id  
    song_id integer not null, -- song id  
    start_time datetime not null -- when user listened  
                                -- to it  
);
```

```
sqlite> select * from listened order by random() limit 1;  
8|4334|2014-02-05 06:53:45
```

Это значит, что пользователь с *id=8* прослушал песню с *id=4334* утром пятого февраля.

От вас в задании потребуется составить sql-запросы, чтобы найти информацию из этой базы данных.

1. Перечислить пользователей в алфавитном порядке username (**все колонки в любом порядке**)
2. 5 пользователей, которые зарегистрировались последними (**все колонки в любом порядке**)
3. Топ 5 пользователей по числу прослушиваний (**все колонки в любом порядке**)
4. Информацию об исполнителях и количестве их альбомов (**имя исполнителя|число**)
5. Информацию об исполнителях и количестве их песен (**имя исполнителя|число**)
6. Самый длинный альбом по числу песен. (**имя исполнителя|название альбома|число**)
7. Самый длинный альбом по суммарной продолжительности. (**имя исполнителя|название альбома|число**)
8. Альбом самой большой средней продолжительностью трека. (**имя исполнителя|название альбома|число**)
9. Топ 5 самых прослушиваемых треков. (**имя исполнителя|название альбома|название песни|число**)
10. Найдите год, что песни, выпущенные в этом году, слушают больше всего (**год|количество прослушиваний**)
11. Для пользователя с id=47 найдите топ 20 последних прослушанных треков. (**имя исполнителя|название альбома|название песни|когда**)
12. (Больше join Богу join) Для каждого пользователя и каждой песни, которую пользователь прослушал, найдите количество прослушиваний. (**имя пользователя|имя исполнителя|название альбома|название песни|число**) . Подсказка – можно использовать пару колонок в group by.