Задания на БД vk:

1. Проанализировать какие запросы могут выполняться наиболее  
   часто в процессе работы приложения и добавить необходимые индексы.

/\*

\* Делаем обязательно индекс на логический ключ **ЭТО** **НЕ ВНЕШНИЙ КЛЮЧ**

\*/

**create** **index** i\_likes\_target\_id **on** likes(target\_id);

/\*

\* Делаем обязательно индекс по дате рождения, чтобы посылать сообщения с поздравлениями

\*/

**create** **index** i\_profiles\_birthday **on** profiles(birthday);

/\*

\* Делаем обязательно индексы на поля таблицы users, чтобы обеспечить поиск пользователя по вводимым на форме данным

\*/

**create** **index** i\_users\_last\_name\_first\_name **on** users(last\_name, first\_name); -- таков порядок полей более избирателен

**create** **index** i\_users\_username **on** users(username);

/\* На следующих таблицах создаем индекс по полю created\_at для целей статистического анализа и/или администрирования

\* (например, перенос копирование строк в другую БД)

\*/

**create** **index** i\_media\_created\_at om media(created\_at);

**create** **index** i\_friendship\_created\_at **on** friendship(created\_at);

**create** **index** i\_messages\_created\_at **on** messages(created\_at);

**create** **index** i\_posts\_created\_at **on** posts(created\_at);

/\* Возможно иногда может потребоваться функциональный индекс на поле posts.head,

\* чтобы, например, искать по названию поста

\*/

**create** **index** i\_posts\_lower\_head **on** posts((**LOWER**(head)));

/\* Возможно потребуется индекс по какому-либо атрибуту json-поля

\*/

**create** **index** i\_media\_metadata\_owner **on** media((**CAST**(metadata->>'$.owner' **as** **CHAR**(30))));

1. Задание на оконные функции  
   Построить запрос, который будет выводить следующие столбцы:  
   имя группы;  
   среднее количество пользователей в группах;  
   самый молодой пользователь в группе;  
   самый старший пользователь в группе;  
   общее количество пользователей в группе;  
   всего пользователей в системе;  
   отношение в процентах (общее количество пользователей в группе / всего пользователей в системе) \* 100.

**select**

group\_name, -- Имя группы

total / **sum**(1) **over**(),-- среднее количество пользователей в группах. ВСЕГО ПОЛЬЗОВАТЕЛЕЙ ДЕЛИМ НА КОЛИЧЕСТВО ГРУПП

**first**, -- самый молодой пользователь в группе

**last**, -- самый старший пользователь в группе;

total\_in\_group, -- Общее количество пользователей в группе

total, -- всего пользователей в системе

total\_in\_group/total \* 100 relation -- отношение в процентах (общее количество пользователей в группе / всего пользователей в системе) \* 100.

**from** (

**select**

**distinct**

c.name **as** group\_name,

**FIRST\_VALUE**(**concat**(u.first\_name, ' ', u.last\_name)) **OVER** (**PARTITION** **BY** cu.community\_id **ORDER** **BY** p.birthday **DESC**) **AS** 'first',

**FIRST\_VALUE**(**concat**(u.first\_name, ' ', u.last\_name)) **OVER** (**PARTITION** **BY** cu.community\_id **ORDER** **BY** p.birthday **ASC**) **AS** 'last',

**COUNT**(u.id) **OVER** w **as** total\_in\_group,

**COUNT**(u.id) **OVER**() total

**from**

communities c

**left** **join** communities\_users cu **on** cu.community\_id = c.id

**left** **join** profiles p **on** p.user\_id = cu.user\_id

**left** **join** users u **on** u.id = p.user\_id

**WINDOW** w **AS** (**PARTITION** **BY** c.id)

) t

**order** **by** group\_name;