Задание 1

От разработчиков поступила задача: нужно выяснить, какие запросы шли на IP-адреса. IP-адрес состоит из четырёх чисел, они разделены точками. Тебе нужны адреса, которые начинаются с "233.201.".

Логи лежат на удалённом сервере по адресу logs/2019/12. День, когда случилась ошибка, неизвестен.

Задача — узнать, какие запросы были отправлены.

***Команда, которой удалось получить нужные логи:***

grep -R ^233.201.\* ~/logs/2019/12

***Логи:***

**233.201.188.154 - - [18/12/2019:21:46:01 +0000] "DELETE** /events HTTP/1.1" 403 3971

**233.201.182.9 - - [21/12/2019:21:56:20 +0000] "PATCH** /users HTTP/1.1" 400 4118

Задание 2

В системе обнаружен баг. Он проявлялся 30.12.2019 и 31.12.2019 с 21:30:00 до 21:39:59. При этом появлялись ошибки с номерами 400 и 500. Твоя задача — сохранить в отдельный файл логи, которые были записаны в этот период. Затем эти логи надо разложить по отдельным файлам: логи с одинаковой ошибкой положи в один файл. Как это сделать:

1. В домашней директории на удалённом сервере создай директорию bug1.
2. Все запросы, которые произошли в указанный период, положи в файл main.txt.
3. Внутри директории bug1 создай директорию events.
4. Внутри директории events создай файлы для ошибок с номерами 400 и 500. Назови эти файлы 400.txt и 500.txt соответственно. В них выдели логи с соответствующей ошибкой из файла main.txt.

***Команды, которые создают директории bug1 и events.***  
mkdir bug1

cd bug1

mkdir events

***Команда, которой выбираешь запросы за указанный период. Это те запросы, которыми ты отбираешь логи в файл main.txt.***

grep -R 3[01]/12/2019:21:3[1-9] ~/logs/2019/12 > main.txt

***Команды, которыми ты кладёшь логи в файлы 400.txt и 500.txt из main.txt.***

grep '" 400 ' ~/bug1/main.txt > ~/bug1/events/400.txt

grep '" 500 ' ~/bug1/main.txt > ~/bug1/events/500.txt

***400.txt***

**/bug1/events**$ cat 400.txt

***500.txt.***

**/bug1/events**$ cat 500.txt

Задание 3

У тебя есть база данных с поездками на такси. По плану на линию обслуживания должно было выйти 10550 автомобилей — эта цифра покрывает спрос пользователей. Команде поступило много жалоб — свободных автомобилей оказалось недостаточно. Сколько такси вышло на линии на самом деле? Информация о всех машинах на линии есть в таблице cabs.

1. Зайди на удалённый сервер.
2. Подключись к базе данных chicago\_taxi
3. Посчитай, сколько всего автомобилей в таблице cabs.

***Число автомобилей***

5529

***Запрос, которым тебе удалось решить задачу.***

SELECT COUNT(\*)

FROM cabs;

Задание 4

Посчитай количество автомобилей в каждой компании из таблицы cabs. Отсортируй значения по убыванию. Команда предполагает, что некоторые компании не вывели достаточно автомобилей на линию.

Выведи те компании, в которых меньше 100 автомобилей. Поле с числом автомобилей назови cnt, поле с названием компании — company\_name.

Чтобы решить задачу, примени оператор HAVING — аналог WHERE для агрегирующих функций.

***Список компаний с числом автомобилей меньше 100.***

51 компания, прикрепила скриншот списка

***Запрос, которым тебе удалось решить задачу.***

SELECT company\_name, COUNT(cabs) AS cnt

FROM cabs

GROUP BY company\_name

HAVING COUNT(cabs) < 100

ORDER BY cnt DESC;

Задание 5

В приложении такси рассчитывается коэффициент стоимости поездки. Если погода хорошая, значение коэффициента равно 1. Если на улице дождь или шторм, коэффициент повышается до 2. У команды есть гипотеза, что в расчётах коэффициента ошибка. Чтобы проверить расчёт коэффициента, команде нужна выборка данных: разработчик может сверить коэффицент с данными в логах и исправить баг. Твоя задача — получить выборку.

Чтобы это сделать:

1. Получи описание погодных условий из таблицы weather\_records для каждого часа.
2. Раздели все часы на две группы оператором CASE: 'Bad', если поле description содержит слова rain или storm; 'Good' для всех остальных.
3. Полученное поле назови weather\_conditions.

В результирующей таблице должно быть два поля — дата и час (ts) и weather\_conditions.

Сделай выборку за период с 2017-11-05 00:00 по 2017-11-06 00:00.

***Полученная таблица с данными за указанный период.***

Скриншот таблицы прикрепила

***Запрос, которым удалось решить задачу.***

SELECT ts,

CASE WHEN description LIKE '%rain%' OR description LIKE '%storm%' THEN 'Bad' ELSE 'Good'

END AS weather\_conditions

FROM weather\_records

WHERE ts BETWEEN '2017-11-05' AND '2017-11-06';

Задание 6

После обновления ПО таксопарки стали сообщать, что прибыль, которую они получают, не сходится с данными, которые отдаёт приложение. Разработка предполагает, что проблема может быть в данных о количестве поездок.

Чтобы определить, есть ли баг, нужно получить выборку с количеством поездок каждого таксопарка за 15 и 16 ноября 2017 года.

1. Выведи поле company\_name. Поле с числом поездок назови trips\_amount и выведи его.
2. Результаты, полученные в поле trips\_amount отсортируй по убыванию.

Подсказка: чтобы решить задачу, соедини таблицы cabs и trips. Примени агрегирующие функции с группировкой. Не забудь написать конструкцию с условием.

***Полученная таблица с данными за указанный период.***

Скриншот таблицы прикрепила

***Запрос, которым удалось решить задачу.***

SELECT cabs.company\_name, COUNT (trips.trip\_id) AS trips\_amount

FROM cabs

INNER JOIN trips ON trips.cab\_id = cabs.cab\_id

WHERE CAST (trips.start.ts as date) >= ‘2017-11-15’ AND CAST (trips.start.ts as date) < ‘2017-11-17’

GROUP BY cabs.company\_name

ORDER BY trips\_amount DESC;