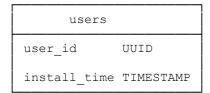
# **Тестовое задание на позицию Marketing Data Quality Engineer**

## Задание 1

Дана следующая структура:



sessions	
user_id	UUID
session_time	TIMESTAMP

payments	
user_id	UUID
payment_time	TIMESTAMP
revenue	NUMERIC

Необходимо написать SQL-запрос, который будет выводить следующую информацию:

- install\_date дата установки когорты
- installs КОЛИЧЕСТВО УСТАНОВОК В install date
- purchases количество платежей за все время
- buyers количество плательщиков в за все время
- revenue сумма платежей за все время
- revenue 1 сумма платежей, совершенных когортой до первого когортного дня (N<=1)
- revenue 3 сумма платежей, совершенных когортой до третьего когортного дня
- revenue 5 сумма платежей, совершенных когортой до пятого когортного дня
- revenue 7 сумма платежей, совершенных когортой до седьмого когортного дня

Примечание: Когортный день считается 24-часовыми интервалам, т.е. [install\_time + N \* 24h; install time + (N + 1) \* 24h), где N=0..inf

## Задание 2

Изменить код ниже так, чтобы данные из таблицы считывались параллельно. Ваше решение должно включать функцию previous date.

```
CREATE OR REPLACE FUNCTION previous_date(date DATE) RETURNS DATE
    LANGUAGE sql
AS
$$
SELECT date - 1;
$$;

SELECT
    previous_date(date)
FROM
    large_table;
```

План запроса выше выглядит так:

```
QUERY PLAN
Seq Scan on large table
```

Ожидаемый план запроса:

```
QUERY PLAN
Gather
  Workers Planned: 3
  -> Parallel Seq Scan on large table
```

### Задание 3

Дана следующая схема данных:

users		
user_id	UUID	
install_time	TIMESTAMP	
platform	TEXT	
is_paid	BOOLEAN	

payments		
user_id	UUID	
payment_time	TIMESTAMP	
revenue	NUMERIC	

#### Ожидаемые поля на выходе:

- install date дата установки когорты
- platform платформа (android/ios)
- is\_paid флаг платного трафика
- cohort\_day когортный день, считается 24-часовыми интервалам, т.е. [install\_time + N \* 24h; install time + (N + 1) \* 24h), где N=0..inf
- acc\_revenue аккумулированная по когортным дням сумма платежей когорты в разрезе install\_date, platform, is\_paid.
- 1. Написать SQL-запрос, используя оконные функции, который выведет данные в формате выше
- 2. Написать SQL-запрос, **не** используя оконные функции, который выведет данные в формате выше

## Задание 4

Вместе с тестовым заданием был приложен файл test.csv. Необходимо провалидировать данные, которые есть в этом датасете.

В качестве ответа ожидается связный текст, в котором будет:

- 1. подробно описаны ваши действия по поиску ненормальных данных, возможно с приложением скриншотов или скриптов
- 2. обособленно написаны все найдены проблемы, чтобы этот список можно было передать разработчикам для исправления

#### Поля в датасете:

- install date дата установки приложения пользователем в формате YYYY-MM-DD
- country двухбуквенный код страны в формате ISO 3166
- campaign id числовой идентификатор рекламной кампании
- campaign\_name название рекламной кампании, которое соответствует этому campaign id
- installs количество установок в разрезе install date, country, campaign id

Если вы не уверены, является ли что-то ошибкой, лучше все равно это указать, такая инициатива покажет ваши способности нестандартно мыслить, что важно для этой вакансии.

## Задание 5

Восстановите SQL-запрос по его плану. План запроса в приложенном файле explain.txt.