

Тестовое задание на позицию Marketing Data Quality Engineer

Задание 1

Дана следующая структура:

users		sessions		payments	
user_id	UUID	user_id	UUID	user_id	UUID
install_time	TIMESTAMP	session_time	TIMESTAMP	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 ($N \leq 1$)
- 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. $[\text{install_time} + N * 24\text{h}; \text{install_time} + (N + 1) * 24\text{h})$, 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		payments	
user_id	UUID	user_id	UUID
install_time	TIMESTAMP	payment_time	TIMESTAMP
platform	TEXT	revenue	NUMERIC
is_paid	BOOLEAN		

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. $[\text{install_time} + N * 24\text{h}; \text{install_time} + (N + 1) * 24\text{h})$, где $N=0..\text{inf}$
 - `acc_revenue` - аккумулярованная 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 найденные 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`.