

DE Тестовое задание

1. Напишите python скрипт который бы загружал следующую таблицу из google sheet в ClickHouse

<https://docs.google.com/spreadsheets/d/1qexS9l4CRPxu6z407TfGUNkHHztrDwTgHoMm0SQssD3E/edit?usp=sharing>

Описание данных:

```
user_id - идентификатор пользователя,  
store - платформа с которой он играет,  
app_name_dashboard - название приложения,  
installed_at - дата установки,  
created_at - дата события,  
event_name - название события (iap_revenue, ad_revenue, event),  
reporting_revenue - доход которое сгенерировало событие,  
ad_mediation_platform - источник рекламного дохода
```

В качестве решения пришлите архив с python скриптом, а также инструкциями по запуску данного кода

2. Напишите SQL код для подсчета следующих метрик:

```
iap_revenue_d0,  
iap_revenue_d1,  
iap_revenue_d3,  
iap_revenue_d7,  
iap_revenue_total,  
api_ad_revenue_d0,  
api_ad_revenue_d1,  
api_ad_revenue_d3,  
api_ad_revenue_d7,  
api_ad_revenue_total,  
sdk_ad_revenue_d0,  
sdk_ad_revenue_d1,  
sdk_ad_revenue_d3,  
sdk_ad_revenue_d7,  
sdk_ad_revenue_total
```

В следующем разрезе:

```
app_name_dashboard,  
store,  
install_date
```

Подсказка:

- sdk ad revenue это ad_revenue у которого значение ad_mediation_platform содержит sdk, все остальное считаем как api ad revenue
- метрики d0,d1...dX мы считаем следующим образом:

```
dateDiff('second', installed_at, created_at) < (86400 * X+1))
```

В качестве решения пришлите файл с SQL кодом

Дополнительно:

1. Подумайте как можно было бы оптимизировать следующую схему таблицы:

```
create table costs
(
    date                Date                default 0,
    app_token           String              default '',
    app_name            String              default '',
    platform            String              default '',
    network             String              default '',
    campaign            String              default '',
    adgroup_name        String              default '',
    ad_name             String              default '',
    country_code        String              default '',
    ad_revenue          Float32             default 0,
    ad_revenue_d0       Float32             default 0,
    ad_revenue_d1       Float32             default 0,
    ad_revenue_d7       Float32             default 0,
    ad_revenue_d14      Float32             default 0,
    ad_revenue_d30      Float32             default 0,
    iap_revenue         Float32             default 0,
    iap_revenue_d0      Float32             default 0,
    iap_revenue_d1      Float32             default 0,
    iap_revenue_d7      Float32             default 0,
    iap_revenue_d14     Float32             default 0,
    iap_revenue_d30     Float32             default 0
)
engine = MergeTree PARTITION BY toDate(date)
ORDER BY (date
, campaign
, network
, app_token
, adgroup_name
, country_code
, ad_name
, platform
, app_name);
```

В качестве решения пришлите файл с улучшенной схемой и вашими мыслями почему вы так решили оптимизировать

3. На основе данных из google sheet напишите SQL запрос для Clickhouse который проранжирует пользователей по следующим основаниям:

- a. суммарный доход,
- b. суммарный доход с рекламы (event_name = 'ad_revenue'),
- c. суммарный доход с платежей (event_name = 'iap_revenue'),
- d. среднему времени в секундах между любыми событиями
- e. максимальному количеству дней после установки до события

На выходе должна быть таблица с колонками:

user_id,
total_revenue_rank,
ad_revenue_rank,
iap_revenue_rank,
diff_between_events_rank,
max_days_rank