Тестовое задание

Задание 1. ClickHouse-запрос

Дана таблица с историей изменения свойств для неких объектов. Напишите запрос, который выведет акутальное (последнее) свойство для каждого объекта.

```
CREATE TABLE Item
  (id UInt64,
   property LowCardinality(String),
   created_at DateTime,
   sign Int8)
ENGINE = CollapsingMergeTree(sign)
ORDER BY id;
```

Задание 2. Витрина данных в ClickHouse

Даны таблицы Deposit и Withdrawal, хранящие данные о пользовательских депозитах и выплатах соответственно. Создайте витрину данных для хранения суммы депозитов, суммы выплат и разности этих сумм с агрегацией по дню. Предложите методы обновления витрины.

```
CREATE TABLE Deposit
  (id UInt64,
   user_id UInt64,
   amount Decimal(18, 5),
   created_at DateTime)
ENGINE = ReplacingMergeTree
PARTITION BY toYYYYMM(created_at)
ORDER BY id;

CREATE TABLE Withdrawal
  (id UInt64,
   user_id UInt64,
   amount Decimal(18, 5),
   created_at DateTime)
ENGINE = ReplacingMergeTree
PARTITION BY toYYYYMM(created_at)
ORDER BY id;
```

Задание 3. Bash-скрипт

Для следующей задачи потребуется локально развернуть ClickHouse (гайд).

Требуется написать bash-скрипт, который будет совершать следующие действия:
1. Пересоздавать таблицу в ClickHouse (листинг DDL будет описан ниже), используя clickhouse-client
2. Выгружать csv-файл

3. Выполнять insert содержимого csv-файла в созданную таблицу в ClickHouse (если в Вашем решении для этого потребуются некоторые преобразования исходных данных в файле, то эти преобразования должны быть выполнены тоже с помощью bash-скрипта)

DDL-запрос для создания таблицы:

```
CREATE TABLE EventsTest(
user_id    String,
product_identifier  Nullable(String),
start_time  DateTime,
end_time   Nullable(DateTime),
price_in_usd    Nullable(Float32)
)
ENGINE = ReplacingMergeTree
PARTITION BY toYYYYMM(start_time)
ORDER BY (toDate(start_time), user_id);
```