Отчет по Делаем НЕ больно с Apache Kafka

Описание/Пошаговая инструкция выполнения домашнего задания:

1. Установите Apache Kafka удобным способом. Установите ClickHouse и выполните необходимые настройки для работы с Kafka.
2. Запишите данные в Kafka и настройте пайплайн чтения через Kafka Engine, затем переместите их в таблицу MergeTree с помощью Materialized View.
3. Проверьте корректность чтения данных из Kafka в ClickHouse.

1. Apache Kafka и ClickHouse были установлены в docker-контейнер с помощью docker-compose.yml:

```
root@NB-R911LJCT:~/kafka_clickhouse# docker compose up -d
WARN[0000] /root/kafka_clickhouse/docker-compose.yml: the attribute `version` is obsolete, it will be ignored, please remove it to avoid potential confusion
WARN[0000] Found orphan containers ([postgres]) for this project. If you removed or renamed this service in your compose file, you can run this command with the --remove-orphans flag to clean it up.
[+] Running 3/3
 ✓ Container zookeeper   Running
 ✓ Container clickhouse  Running
 ✓ Container kafka       Running
root@NB-R911LJCT:~/kafka_clickhouse#
root@NB-R911LJCT:~/kafka_clickhouse# docker ps
CONTAINER ID   IMAGE                              COMMAND                  CREATED             STATUS              PORTS                                                                                            NAMES
bdd9e0c66bf4   bitnami/kafka:2.8.1                "/opt/bitnami/script…"   About an hour ago   Up About an hour    0.0.0.0:9092->9092/tcp, [::]:9092->9092/tcp                                                      kafka
f93576d7fed5   bitnami/zookeeper:3.8              "/opt/bitnami/script…"   About an hour ago   Up About an hour    2888/tcp, 3888/tcp, 0.0.0.0:2181->2181/tcp, [::]:2181->2181/tcp, 8080/tcp                        zookeeper
aea62d90a7dd   yandex/clickhouse-server:latest    "/entrypoint.sh"         20 hours ago        Up About an hour    0.0.0.0:8123->8123/tcp, [::]:8123->8123/tcp, 0.0.0.0:9000->9000/tcp, [::]:9000->9000/tcp, 9009/tcp   clickhouse
root@NB-R911LJCT:~/kafka_clickhouse#
```

2. Далее через консоль кафка я отправил несколько json-сообщений и прочитал их Clickhouse с помощью kafka-engine:

```
I have no name!@bdd9e0c66bf4:/$ kafka-topics.sh --create --topic topic-otus --bootstrap-server kafka:9092 --partitions 1 --replication-factor 1
Created topic topic-otus.
I have no name!@bdd9e0c66bf4:/$ kafka-console-producer.sh --topic topic-otus --bootstrap-server kafka:9092
>{"id": 1, "name": "Nick", "email": nick@mail.ru"}
>{"id": 1, "name": "Nick", "email": "nick@mail.ru"}
>{"id": 1, "name": "Nick", "email": "nick@mail.ru"}
>{"id": 2, "name": "Nick", "email": "nick@mail.ru"}
>{"id": 1, "name": "Nick", "email": "nick@mail.ru"}
>{"id": 2, "name": "Nick", "email": "nick@mail.ru"}
>{"id": 2, "name": "Nick", "email": "nick@mail.ru"}
>{"id": 3, "name": "Bred", "email": "Bred@mail.ru"}
>
```

2.1. В клике создал таблицу на движке kafka под чтение этого топика, а так же матвью и целевую таблицу. Данные залетели в целевую таблицу:

```sql
create table kafka_users
(
    id    Int64,
    name  String,
    email String
)
    engine = Kafka SETTINGS kafka_broker_list = 'kafka:9092', kafka_topic_list = 'topic-otus', kafka_group_name = 'otus_group', kafka_format = 'JSONEachRow';
```

```sql
CREATE MATERIALIZED VIEW default.kafka_users_mv
            TO default.kafka_users_target
            (
             `id` Int64,
             `name` String,
             `email` String
                )
AS
SELECT  id, name, email *
FROM default.kafka_users;
```

```sql
-- auto-generated definition
create table kafka_users_target
(
    id    Int64,
    name  String,
    email String
)
    engine = MergeTree ORDER BY id
        SETTINGS index_granularity = 8192;
```

	id	name	email
1	1	Nick	nick@mail.ru
2	2	Nick	nick@mail.ru
3	2	Nick	nick@mail.ru
4	3	Bred	Bred@mail.ru

2.2. Далее я создал еще одну тпблицу с движеом kafka нацеленную на тот же топик, но только с другой коесьюмер-группой. Как и ожидалось в клик залетела вся имеющаяся история из кафка:

```
CREATE TABLE kafka_users_new_consumer (
    `id` Int64,
`name` String,
`email` String
) ENGINE = Kafka
SETTINGS kafka_broker_list = 'kafka:9092',
         kafka_topic_list = 'topic-otus',
         kafka_group_name = 'new_consumer',
         kafka_format = 'JSONEachRow',
    kafka_num_consumers = 1;

CREATE TABLE kafka_users_new_consumer_target
(
             `id` Int64,
             `name` String,
             `email` String
               )
ENGINE MergeTree
order by id;

CREATE MATERIALIZED VIEW kafka_users_new_consumer_mv
    to kafka_users_new_consumer_target
as
    SELECT*
    FROM kafka_users_new_consumer;
```

```
339
340 ✓  SELECT * FROM kafka_users_new_consumer_target
```

Output | default.kafka_users_new_consumer_target | DDL

	id	name	email
1	1	Nick	nick@mail.ru
2	2	Nick	nick@mail.ru
3	2	Nick	nick@mail.ru
4	3	Bred	Bred@mail.ru

3. Проверка показала, что данные корректны.

В качестве задания со звездочкой, я сделал вставку в первую «кафка-таблицу» с помощью INSERT INTO, данные залетели в кафку, а уже от туда и во вторую таблицу