

## Тестовое задание

### 1. Слияние логов

Имеется два файла с логами в формате **JSONL**, пример лога:

```
...
{"timestamp": "2021-02-26 08:59:20", "log_level": "INFO", "message": "Hello"}
{"timestamp": "2021-02-26 09:01:14", "log_level": "INFO", "message": "Crazy"}
{"timestamp": "2021-02-26 09:03:36", "log_level": "INFO", "message": "World!"}
...
```

Сообщения в заданных файлах упорядочены по полю **timestamp** в порядке возрастания.

Требуется написать скрипт, который объединит эти два файла в один.

При этом сообщения в получившемся файле тоже должны быть упорядочены в порядке возрастания по полю **timestamp**.

К заданию прилагается вспомогательный скрипт на python3, который создает два файла **"log\_a.jsonl"** и **"log\_b.jsonl"**.

Командлайн для запуска:

**log\_generator.py** <path/to/dir>

Ваше приложение должно поддерживать следующий командлайн:

**<your\_script>.py** <path/to/log1> <path/to/log2> -o <path/to/merged/log>

### 2. Миграция базы данных

Есть база данных и два типа сервисов **А** и **Б**:

- Сервисы типа **А** добавляют в базу записи в формате: **id, name, status, timestamp**.
- Сервисы типа **Б** читают эти данные для агрегации и прочих нужд.

В какой-то момент выясняется, что строковые имена в этих записях занимают слишком много памяти, и при этом часто повторяются. Поэтому целесообразно вынести их в отдельную табличку.

В результате данные должны будут добавляться в следующем формате: **id, name\_id, status, timestamp**. Где **name\_id** внешний ключ к новой таблице с полями: **id, name**.

Задача заключается в том, что бы составить пошаговый план миграции базы и сервисов на новый формат данных, при этом не разломав работоспособность системы.

#### ВАЖНО:

1. Нельзя остановить и обновить все сервисы разом (т.е. нельзя остановить сразу все сервисы типа А, либо все сервисы типа Б, обновление сервисов происходит по одному за раз).
2. В базе данных атомарно можно делать только следующие запросы:
  - добавить колонку
  - удалить колонку
  - переименовать колонку