Введение

- ❏ Репозиторий для сдачи: http://gitlab.atp-fivt.org/hobod2025/XXX-spanner
- ❏ Ветки: **spannertask1, spannertask2. Тестов в задачах нет.** Для сдачи нужно сделать merge request из веток в мастер.

Входные данные

Входные данные лежат в файловой системе **HDFS**: **/data/movielens**. Нам интересны 2 таблицы:

Movies

/data/movielens/movies.csv

Название поля	Тип данных
movieId	INT64
title	STRING
genres	ARRAY<STRING>

Ratings

/data/movielens/ratings_20m.csv

Название поля	Тип данных
userId	INT64
movieId	INT64
rating	FLOAT64
timestamp	TIMESTAMP

Для решения задачи используйте Google Spanner API. Можете выбрать любой удобный вам язык программирования.

Задание 1 [0,8 баллов]. Неблокирующее чтение в Spanner

Цель данного задания - убедиться в том, что read-only транзакция в Spanner не блокирует read-write транзакцию. Т.е. если во время транзакции чтения происходит обновление данных, мы продолжаем читать то же состояние данных, которое было в начале транзакции. Это происходит за счет версионирования данных.

Создайте Spanner instance, создайте в нем схему данных и заполните данными из Movielens.

1. Создайте транзакцию, которая состоит из N запросов на чтение 10 строк таблицы ratings. Например, 1й запрос читает строки 1-10, 2й 11-20 и т.д. Выполнение такой транзакции очевидно займет некоторое время.
2. Напишите запрос, который поднимает оценки всем фильмам в жанре Fantasy на 2. (это приведет к тому, что почти у всех таких фильмов оценка станет 5). Запустите этот запрос на той же базе после начала транзакции из пункта 1.
3. Убедитесь, что обновление оценок не повлияло на чтение данных и транзакция осталась изолированной.
4. После того, как транзакция завершится, запустите её ещё раз и убедитесь что на этот раз читаются обновленные оценки.

Код запросов залейте в ветку **spannertask1**. TOP-100 строк из вывода запросов (1) и (4) также положите в эту ветку.

Задание 2 [0,7 баллов]. Чтение и запись в Spanner

Цель данного задания - проверить 2 факта:

1. чтение данных происходит гораздо быстрее записи;
2. эта разница более заметна в мульти региональных instances.

Действительно, в большинстве хранилищ экосистемы BigData чтение работает быстрее (это продиктовано подходом “write once, read many”). Но почему в мультирегиональном instance это более заметно?

При создании Spanner instance можно выбрать 2 типа Instance:

- ➢ Regional
- ➢ Multi-regional

1. Создайте 2 instance, один из которых regional, другой multi-regional. В каждом instance должно быть не менее 3 нод.
2. В обеих instance создайте таблицы для данных movielens.
3. В обеих instance выполните следующее:
   a. Запишите в таблицу movies 1 фильм, а в таблицу ratings 1 оценку для этого фильма.
   b. Сделайте `SELECT` таблицы ratings (с 1 записью).
   c. Замеряйте время записи и чтения на обоих instances.
   d. В результате прогона должна получиться табличка:

	Regional	Multi-regional
Чтение, мс.		
Запись, мс.		

   e. Проведите данный эксперимент с 10, 100, затем с 1000 записями.
   f. Ответьте на вопросы:
      i. Как отличается время чтения в зависимости от типа instance? А время записи?
      ii. Чем вы это можете объяснить? (см. лекцию Р. Г. Липовского).

Код залейте в ветку **spannertask2**. Таблицу и ответы на вопросы также положите в эту ветку[1].

Полезные ссылки

1. Работа со Spanner с помощью Spark.

[1] Желательно в формате .md, но можно и в другом удобном вам.