Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ **ИНФОРМАТИКА, ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ И СИСТЕМЫ УПРАВЛЕНИЯ**

КАФЕДРА **КОМПЬЮТЕРНЫЕ СИСТЕМЫ И СЕТИ (ИУ6)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.04.01 Информатика и вычислительная техника**

МАГИСТЕРСКАЯ ПРОГРАММА **09.04.01/07 Интеллектуальные системы анализа, обработки и интерпретации больших данных**

О Т Ч Е Т

по лабораторной работе №10

Название: Spark

Дисциплина: Языки программирования для работы с большими данными

Студент	ИУ6-22М		А.М. Панфилкин
	(Группа)	(Подпись, дата)	(И.О. Фамилия)
Преподаватель			П.В. Степанов
		(Подпись, дата)	(И.О. Фамилия)

Москва, 2022 г.

Весь приведенный ниже код также доступен в следующем репозитории:

https://github.com/SilkSlime/iu6plfbd

Задание: Выбрать любой датасет на kaggle.com. Сделать 10 выборок данных по выбранной предметной области.

Был выбран датасет rotten_tomatoes_movies, который содержит данные о фильмах и их оценках с сайта rotten tomatoes. Файл имеет следующие поля заголовка: id,title,audienceScore,tomatoMeter,rating,ratingContents,releaseDateTheaters,releaseDateStreaming,runtimeMinutes,genre,originalLanguage,director,writer,boxOffice,distributor,soundMix.

Листинг 1 – Задание 1

```
package org.example;

import org.apache.spark.sql.SparkSession;
import org.apache.spark.sql.Dataset;
import org.apache.spark.sql.Row;

public class Main {
    public static void main(String[] args) {
        // Грузим файл из resources/rotten_tomatoes_movies.csv
        SparkSession spark = SparkSession
                .builder()
                .appName("Java Spark SQL basic example")
                .config("spark.master", "local")
                .getOrCreate();
        // Создаем датасет из csv файла
        Dataset<Row> df =
spark.read().option("header","true").csv("src/main/resources/rotten_tomatoes_movies.csv");
        // Создаем временную таблицу rtm
        df.createOrReplaceTempView("rtm");
        // Заголовки в csv файле:
        //
id,title,audienceScore,tomatoMeter,rating,ratingContents,releaseDateTheaters,releaseDateStreaming,runtimeMinutes,genre,originalLanguage,director,writer,boxOffice,distributor,soundMix
        // Используем SQL и .show() для 10 примеров выборки и аггрегации
        // Example 1: Выбрать все фильмы с рейтингом > 4
        spark.sql("SELECT * FROM rtm WHERE rating = 5 AND tomatoMeter = 100").show();
        // Example 2: Выбрать все фильмы с рейтингом > 4 и рейтингом критиков > 90
        spark.sql("SELECT * FROM rtm WHERE rating = 5 AND tomatoMeter = 100 AND audienceScore =
100").show();
        // Example 3: Выбрать все фильмы с жанром "Comedy"
        spark.sql("SELECT * FROM rtm WHERE genre LIKE '%Comedy%'").show();
        // Example 4: Аггрегировать по жанрам и посчитать количество фильмов в каждом жанре.
Сортировать по убыванию
        spark.sql("SELECT genre, COUNT(*) FROM rtm GROUP BY genre ORDER BY COUNT(*)
DESC").show();
        // Example 5: Вывести жанры 10 самых плохих фильмов
        spark.sql("SELECT genre FROM rtm ORDER BY tomatoMeter ASC LIMIT 10").show();
```

```
        // Example 6: Вывести средний рейтинг фильмов по жанрам
        spark.sql("SELECT genre, AVG(tomatoMeter) FROM rtm GROUP BY genre ORDER BY
AVG(tomatoMeter) DESC").show();
        // Example 7: Вывести средний рейтинг фильмов по жанрам, у которых рейтинг критиков >
90
        spark.sql("SELECT genre, AVG(tomatoMeter) FROM rtm WHERE tomatoMeter > 90 GROUP BY
genre ORDER BY AVG(tomatoMeter) DESC").show();
        // Example 8: Вывести количество фильмов по годам (использовать из releaseDateTheaters
первые 4 символа)
        spark.sql("SELECT SUBSTRING(releaseDateTheaters, 1, 4) AS year, COUNT(*) FROM rtm GROUP
BY year ORDER BY year").show();
        // Example 9: Вывести года по убыванию среднего рейтинга фильмов
        spark.sql("SELECT SUBSTRING(releaseDateTheaters, 1, 4) AS year, AVG(tomatoMeter) FROM
rtm GROUP BY year ORDER BY AVG(tomatoMeter) DESC").show();
        // Example 10: Вывести количество фильмов по годам и жанрам
        spark.sql("SELECT SUBSTRING(releaseDateTheaters, 1, 4) AS year, genre, COUNT(*) FROM
rtm GROUP BY year, genre ORDER BY year, genre").show();


    }
}
```

Вывод: в ходе данной лабораторной работы были изучены принципы реализации высокопроизводительных вычислений через среду Spark в Java. Был выбран датасет из открытого источника, после чего загружен в среду Spark как таблица. Далее были выведены 10 различных выборок с условиями и аггрегациями.