**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 Системы обработки информации и управления

**Лабораторная работа №3**

**«Apache Cassandra»**

ИСПОЛНИТЕЛИ: Гаврилов Л.Я.

ФИО

группа ИУ5-24М

подпись

"" март 2024 г.

ПРЕПОДАВАТЕЛЬ: Виноградова М.В.

ФИО

подпись

" " 2024 г.

Москва – 2024

1. **Цель работы**

* Изучить модель представления данных и способы работы с колоночными БД.
* Освоить методы создания колоночной БД и языки запросов к ним.
* Получить навыки работы с колоночной СУБД Apache Cassandra.

1. **Задание**
2. Создать в среде CassandraDb свое пространство ключей. Определить семейство столбцов по теме «Подбор фильмов на основе глубокого обучения».
3. Продемонстрировать добавление, изменение и удаление данных в БД, используя команды язык Cassandra Query Language.
4. Продемонстрировать (вывести на экран) содержимое БД.
5. Создать второе семейство столбцов по той же теме, определить для них

распределительный и кластерный индексы.

1. Определить для семейства столбцов индекс(ы). Выполнить запросы с фильтрацией по ключам и индексам. Продемонстрировать работу allow filtering.
2. Выполнить запросы к базе данных с селекцией и проекцией. Выполнить запрос с использованием скалярных и агрегатных функций.
3. Добавить строку с указанием TTL, продемонстрировать действие TTL.

-- Создаем пространство ключей с репликацией в одном датацентре

**CREATE** **KEYSPACE** movie\_recommendation **WITH** REPLICATION = {

**'class'**: **'SimpleStrategy'**,

**'replication\_factor'**: 1

};

-- Используем созданное пространство ключей

**USE** movie\_recommendation;

-- Создаем семейство столбцов для хранения информации о фильмах

**CREATE** **TABLE** movies (

movie\_id **UUID** **PRIMARY** **KEY**, -- уникальный идентификатор фильма

title **TEXT**, -- название фильма

genre **TEXT**, -- жанр фильма

release\_year **INT**, -- год выпуска

rating **FLOAT** -- рейтинг фильма

);

-- Добавляем данные в таблицу movies

**INSERT** **INTO** ***movies*** (***movie\_id***, ***title***, ***genre***, ***release\_year***, ***rating***)

**VALUES** (**uuid**(), **'Inception'**, **'Sci-Fi'**, 2010, 8.8);

**INSERT** **INTO** ***movies*** (***movie\_id***, ***title***, ***genre***, ***release\_year***, ***rating***)

**VALUES** (**uuid**(), **'The Matrix'**, **'Sci-Fi'**, 1999, 8.7);

**INSERT** **INTO** ***movies*** (***movie\_id***, ***title***, ***genre***, ***release\_year***, ***rating***)

**VALUES** (**uuid**(), **'Interstellar'**, **'Sci-Fi'**, 2014, 8.6);

-- Добавляем больше фильмов в таблицу movies

**INSERT** **INTO** ***movies*** (***movie\_id***, ***title***, ***genre***, ***release\_year***, ***rating***)

**VALUES** (**uuid**(), **'The Dark Knight'**, **'Action'**, 2008, 9.0);

**INSERT** **INTO** ***movies*** (***movie\_id***, ***title***, ***genre***, ***release\_year***, ***rating***)

**VALUES** (**uuid**(), **'Pulp Fiction'**, **'Crime'**, 1994, 8.9);

**INSERT** **INTO** ***movies*** (***movie\_id***, ***title***, ***genre***, ***release\_year***, ***rating***)

**VALUES** (**uuid**(), **'The Godfather'**, **'Crime'**, 1972, 9.2);

**INSERT** **INTO** ***movies*** (***movie\_id***, ***title***, ***genre***, ***release\_year***, ***rating***)

**VALUES** (**uuid**(), **'Forrest Gump'**, **'Drama'**, 1994, 8.8);

**INSERT** **INTO** ***movies*** (***movie\_id***, ***title***, ***genre***, ***release\_year***, ***rating***)

**VALUES** (**uuid**(), **'Fight Club'**, **'Drama'**, 1999, 8.8);

**INSERT** **INTO** ***movies*** (***movie\_id***, ***title***, ***genre***, ***release\_year***, ***rating***)

**VALUES** (**uuid**(), **'The Shawshank Redemption'**, **'Drama'**, 1994, 9.3);

**INSERT** **INTO** ***movies*** (***movie\_id***, ***title***, ***genre***, ***release\_year***, ***rating***)

**VALUES** (**uuid**(), **'The Lord of the Rings: The Return of the King'**, **'Fantasy'**, 2003, 8.9);

**INSERT** **INTO** ***movies*** (***movie\_id***, ***title***, ***genre***, ***release\_year***, ***rating***)

**VALUES** (**uuid**(), **'The Avengers'**, **'Action'**, 2012, 8.0);

**INSERT** **INTO** ***movies*** (***movie\_id***, ***title***, ***genre***, ***release\_year***, ***rating***)

**VALUES** (**uuid**(), **'Gladiator'**, **'Action'**, 2000, 8.5);

**INSERT** **INTO** ***movies*** (***movie\_id***, ***title***, ***genre***, ***release\_year***, ***rating***)

**VALUES** (**uuid**(), **'The Silence of the Lambs'**, **'Thriller'**, 1991, 8.6);

-- Выводим все данные из таблицы movies, чтобы увидеть UUID для фильмов

**SELECT** \* **FROM** ***movies***;

-- Изменяем рейтинг фильма Inception

**UPDATE** ***movies*** **SET** ***rating*** = 9.0 **WHERE** movie\_id = <UUID\_INCEPTION>;

-- Удаляем фильм The Matrix

**DELETE** **FROM** ***movies*** **WHERE** movie\_id = <UUID\_MATRIX>;

-- Создаем таблицу с распределительным и кластерным индексами

**CREATE** **TABLE** movie\_recommendations (

user\_id **UUID**, -- уникальный идентификатор пользователя

movie\_id **UUID**, -- уникальный идентификатор фильма

recommended\_movie\_id **UUID**, -- идентификатор рекомендуемого фильма

confidence **FLOAT**, -- уверенность в рекомендации

**PRIMARY** **KEY** (user\_id, movie\_id) -- распределительный и кластерный ключ

);

-- Индекс по рекомендуемому фильму

**CREATE** **INDEX** **ON** movie\_recommendations (recommended\_movie\_id);

-- Индекс по уверенности в рекомендации

**CREATE** **INDEX** **ON** movie\_recommendations (confidence);

-- Выполняем запрос с фильтрацией по индексу

**SELECT** \* **FROM** ***movie\_recommendations*** **WHERE** recommended\_movie\_id = <SOME\_UUID>;

-- Запрос с использованием allow filtering

**SELECT** \* **FROM** ***movie\_recommendations*** **WHERE** ***confidence*** > 0.8 **ALLOW** **FILTERING**;

-- Добавляем тестовые данные в таблицу movie\_recommendations

**INSERT** **INTO** ***movie\_recommendations*** (***user\_id***, ***movie\_id***, ***recommended\_movie\_id***, ***confidence***)

**VALUES** (**uuid**(), **uuid**(), **uuid**(), 0.9);

**INSERT** **INTO** ***movie\_recommendations*** (***user\_id***, ***movie\_id***, ***recommended\_movie\_id***, ***confidence***)

**VALUES** (**uuid**(), **uuid**(), **uuid**(), 0.85);

**INSERT** **INTO** ***movie\_recommendations*** (***user\_id***, ***movie\_id***, ***recommended\_movie\_id***, ***confidence***)

**VALUES** (**uuid**(), **uuid**(), **uuid**(), 0.75);

**INSERT** **INTO** ***movie\_recommendations*** (***user\_id***, ***movie\_id***, ***recommended\_movie\_id***, ***confidence***)

**VALUES** (**uuid**(), **uuid**(), **uuid**(), 0.6);

-- Выборка определенных полей (проекция)

**SELECT** ***user\_id***, ***movie\_id***, ***confidence*** **FROM** ***movie\_recommendations*** **WHERE** confidence > 0.7 **ALLOW** **FILTERING**;

-- Агрегатные функции: подсчет среднего значения уверенности

**SELECT** **AVG**(***confidence***) **FROM** ***movie\_recommendations***;

-- Подсчет количества рекомендаций для конкретного фильма

**SELECT** **COUNT**(\*) **FROM** ***movie\_recommendations*** **WHERE** recommended\_movie\_id = <SOME\_UUID>;

-- Добавляем строку с TTL

**INSERT** **INTO** ***movie\_recommendations*** (***user\_id***, ***movie\_id***, ***recommended\_movie\_id***, ***confidence***)

**VALUES** (**uuid**(), **uuid**(), **uuid**(), 0.63)

**USING** **TTL** 60; -- строка удалится через 60 секунд

-- Проверяем действие TTL

**SELECT** \* **FROM** ***movie\_recommendations*** **WHERE** ***confidence*** = 0.63;







