

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
им. Н.Э. Баумана

Факультет «Информатика и системы управления»
Кафедра «Систем обработки информации и управления»

ОТЧЕТ

Лабораторная работа № 3
по дисциплине «Постреляционные базы данных»

Тема: «Создание колоночной базы данных и работа с ней на примере СУБД
Cassandra»

ИСПОЛНИТЕЛЬ:
группа ИУ5-24М

Журавлев Н.В.
ФИО

подпись

"10" марта 2024 г.

ПРЕПОДАВАТЕЛЬ:

Виноградова М.В.
ФИО

подпись

" " _____ 202_ г.

Москва - 2024

Цель работы

- Изучить постреляционные возможности языка SQL.
- Освоить языки и технологии SQL\PSM на примере PostgreSQL.
- Получить навыки программирования на стороне сервера.

Задание

1. Создать в среде CassandraDb свое пространство ключей.

Определить семейство столбцов по теме, выданной преподавателем.

2. Продемонстрировать добавление, изменение и удаление данных в БД, используя команды API и/или язык Cassandra Query Language.

3. Продемонстрировать (вывести на экран) содержимое БД.

4. Создать второе семейство столбцов по той же теме, определить для них распределительный и кластерный индексы.

5. Определить для семейства столбцов индекс(ы). Выполнить запросы с фильтрацией по ключам и индексам. Продемонстрировать работу allow filtering.

6. Выполнить запросы к базе данных с селекцией и проекцией. Выполнить запрос с использованием скалярных и агрегатных функций.

7. Добавить строку с указанием TTL, продемонстрировать действие TTL.

8. Выполнить запросы с группировкой и сортировкой данных.

9. Продемонстрировать усечение таблицы и удаление таблицы/индекса.

10. Продемонстрировать создание пакета запросов.

Ход работы

Создадим пространство ключей test с стратегией репликации 'SimpleStrategy' и коэффициентом репликации 1:

```
CREATE KEYSPACE test WITH REPLICATION = { 'class' : 'SimpleStrategy', 'replication_factor' : 1 };
```

Перейдем в созданное пространство используя use test.

И создадим таблицу:

```
CREATE TABLE account
```

```
(  
    id uuid,  
    age int,  
    height float,  
    weight float,  
    login text,  
    password text,  
    special text,  
    PRIMARY KEY (id)  
);
```

Для добавления данных:

```
INSERT INTO account (id, age, height, login, password, special, weight)  
VALUES (NOW(), 18, 170, 'login', 'password', 'Здоровый', 70);
```

Выполнив select – запрос отобразим все значения БД:

```
SELECT * FROM system_schema.tables WHERE keyspace_name = 'test';
```

Выполнение запроса для нашей таблицы:

```
SELECT * FROM account;
```

Для обновления используется UPDATE:

```
UPDATE account SET password = 'pass' WHERE id=a55c2d10-e37e-11ee-  
9a45-c56b723d9e3c;
```

Для удаления используется DELETE:

```
DELETE FROM account WHERE id=42f55470-e37f-11ee-9a45-  
c56b723d9e3c;
```

СУБД не дает возможности по умолчанию искать по неключевым колонкам, для которых не создан индекс. Решением этой проблемы является создание вторичного индекса или использование конструкции ALLOW FILTERING:

```
SELECT age FROM account WHERE login = 'login' ALLOW FILTERING;
```

Ключевое слово TTL (Time-to-Live (англ.) – Время жизни) позволяет указать, сколько будет существовать запись (созданная, или измененная). INSERT и UPDATE поддерживают TTL. Время задается в секундах:

```
INSERT INTO account (id, age, height, login, password, special, weight)
VALUES (NOW(), 18, 180, 'delete', 'password', 'Здоровый', 80) USING TTL 15;
```

Для удаления таблиц и индексов, сделаем таблицу, заполним её, затем создадим индекс, после усечём её и удалим индекс, а затем удалим таблицу

```
CREATE TABLE student (
```

```
id uuid,
```

```
citizenship text,
```

```
first_name text,
```

```
last_name text,
```

```
age int,
```

```
PRIMARY KEY (id)
```

```
);
```

```
INSERT INTO student (id, citizenship, first_name, last_name, age) VALUES
(now(), 'Russia', 'Ivan', 'Ivanov', 25);
```

```
CREATE INDEX citizenshipIndex ON student (citizenship);
```

```
TRUNCATE student;
```

```
DROP INDEX citizenshipIndex;
```

```
DROP TABLE student;
```

Создадим таблицу с кластерным и распределительным ключом:

```
CREATE TABLE IF NOT EXISTS public.diet
```

```
(
```

```
id uuid,
```

```
name text,
```

```
description text,
```

```
age int,
```

```
height float,
```

```
weight float,
```

PRIMARY KEY (name, id)
) WITH CLUSTERING ORDER BY (id DESC, age DESC);

Создание индекса:

CREATE INDEX heightIndex ON diet (height);

Запросы с фильтрацией по ключам и индексам:

SELECT * FROM diet WHERE height = 150;

SELECT height FROM diet WHERE name = 'name 1';

Выполнить запросы к базе данных с селекцией и проекцией:

SELECT name FROM diet WHERE age > 18 ALLOW FILTERING;

Выполнить запрос с использованием скалярных и агрегатных функций:

SELECT avg(age) FROM diet WHERE age > 18 ALLOW FILTERING;

SELECT now() FROM diet;

Выполнить запросы с группировкой и сортировкой данных:

SELECT name, MAX(age) FROM diet GROUP BY name;

SELECT * FROM diet WHERE name = 'name 1' ORDER BY id, age;

Создание пакета запросов:

BEGIN BATCH

INSERT INTO diet (id, name, age, description, height, weight) VALUES
(9709f7c0-e391-11ee-acdb-7d1b0b77aa61, 'name diet', 18, 'Description', 180, 70);

UPDATE diet SET description = 'desc_diet' WHERE age=18 AND name = 'name
diet' AND id=9709f7c0-e391-11ee-acdb-7d1b0b77aa61;

APPLY BATCH;

Вывод

В результате выполнения работы были

Список используемой литературы

1. Фаулер, Мартин, Садаладж, Прамодкумар Дж. NoSQL: новая методология разработки нереляционных баз данных. : Пер. с англ. - М.: ООО "И.Д. Вильямс", 2013г.
2. Habr. Как устроена apache Cassandra. – Текст. Изображение. : электронные // Habr : [сайт]. – URL: <https://habrahabr.ru/post/155115/>

(дата обращения 01.05.2022)

3. Apache Cassandra – Текст. Изображение. : электронные // Apache Cassandra. : [сайт]. – URL: https://cassandra.apache.org/_/index.html (дата обращения 01.05.2022)
4. Apache Cassandra: Cassandra documentation: The Cassandra Query Language (CQL) – Текст. Изображение. : электронные // Apache Cassandra. : [сайт]. – URL: <http://cassandra.apache.org/doc/latest/cql/index.html> (дата обращения 01.05.2022)
5. Datastax. Documentation. CQL for Apache Cassandra 3.0: Create Keyspace. – Текст. : электронные // Datastax: Documentation : [сайт]. – URL: https://docs.datastax.com/en/cql-oss/3.3/cql/cql_reference/cqlCreateKeyspace.html (дата обращения 01.06.2020)
6. Datastax. Documentation. CQL for Apache Cassandra 3.0: A deep look at the CQL where clause. – Текст. : электронные // Datastax: Documentation : [сайт]. – URL: <https://www.datastax.com/blog/2015/06/deep-look-cql-where-clause> (дата обращения 01.06.2020)
7. Apache Cassandra: Cassandra documentation. – Текст. Изображение. : электронные // Apache Cassandra. : [сайт]. – URL: <http://cassandra.apache.org/doc/latest/> (дата обращения 01.05.2022)
8. Habr. Моделирование данных в Cassandra 2.0 на CQL3. – Текст. Изображение. : электронные // Habr : [сайт]. – URL: <https://habr.com/ru/post/203200/> (дата обращения 01.05.2022)
9. Habr. SELECT...WHERE запросы в Cassandra 2.0 на CQL3. – Текст. Изображение. : электронные // Habr : [сайт]. – URL: <https://habr.com/ru/post/205176/> (дата обращения 01.05.2022)
10. How to Program Blog: Using Group By in Apache Cassandra. – Текст. Изображение. : электронные // How to Program Blog: [сайт]. – URL: <https://howtoprogram.xyz/2017/02/18/using-group-apache-cassandra/> (дата

обращения 01.05.2022)

11.Datastax. Documentation. CQL for Apache Cassandra 3.0: Time-to-live. –

Текст. : электронные // Datastax: Documentation : [сайт]. – URL:

https://docs.datastax.com/en/cql-oss/3.3/cql/cql_using/useTTL.html (дата

обращения 01.06.2020)

12.Datastax. Documentation. CQL for Apache Cassandra 3.0: Materialized

View. – Текст. :электронные // Datastax: Documentation : [сайт]. – URL:

[https://docs.datastax.com/en/cql-oss/3.3/cql/cql_reference/cqlCreateMateriali](https://docs.datastax.com/en/cql-oss/3.3/cql/cql_reference/cqlCreateMaterializedView.html)

[zedView.html](https://docs.datastax.com/en/cql-oss/3.3/cql/cql_reference/cqlCreateMaterializedView.html) (дата обращения 01.06.2020)