Московский государственный технический университет им.Н.Э.Баумана
кафедра "Системы обработки информации и управления"

Постреляционные базы данных

к лабораторной работе №4

Лабораторная работа «Работа с колоночной NOSQL БД на примереCassandraDb»
по дисциплине «Постреляционные базы данных»

Инструктор：Мария Валерьевна

Email:2623859464@qq.com

Студент: Ван Чаочао

группа ИУ5И-22М

2022/04/01

Цель работы:

1. Изучить модель представления данных и способы работы с колоночными БД NoSql.

2. Освоить методы создания колоночной БД и языки запросов к ним.

3. Получить навыки работы с колоночной БД CassandraDb.

Пункты задания для выполнения:

Задание 1. Создание БД (базовая часть)

1.1 Создать в среде CassandraDb свое пространство ключей.

на языке CQL:

```
cqlsh> CREATE KEYSPACE carsystem WITH replication={'class':'SimpleStrategy','replication_factor':3};
cqlsh> describe carsystem;

CREATE KEYSPACE carsystem WITH replication = {'class': 'SimpleStrategy', 'replication_factor': '3'}  AND durable_writes = true;

cqlsh>
```

1.2. Использовать новое пространство ключей:

```
cqlsh> use carsystem;
cqlsh:carsystem>
```

1.3. Определить семейство столбцов по теме своего ДЗ. Добавить в семейство столбцов строки с данными. Продемонстрировать (вывести на экран) содержимое БД.

Создать таблицу:

```
cqlsh:carsystem> CREATE TABLE Car(
             ... CarsID int PRIMARY KEY,
             ... CarName text,
             ... CarDescription text,
             ... Price int,
             ... Stock int,
             ... Sold int
             ... );
```

```
cqlsh:carsystem> create table users(
            ... userid int PRIMARY KEY,
            ... username text,
            ... password varchar,
            ... realname text,
            ... gender tinyint,
            ... address text,
            ... phone varchar,
            ... dateofbirth date
            ... );
```

вставить данные

```
cqlsh:carsystem> insert into car(carsid,cardescription,carname,price,sold,stock) VALUES (2,'black','Geely',1800000,10,80);
cqlsh:carsystem> insert into car(carsid,cardescription,carname,price,sold,stock) VALUES (3,'silver','Bently',10000000,8,50);
cqlsh:carsystem> insert into car(carsid,cardescription,carname,price,sold,stock) VALUES (4,'Wrangler','JEEP',42900000,9,60);
```

```
cqlsh:carsystem> select * from car;

 carsid | cardescription | carname | price    | sold | stock
--------+----------------+---------+----------+------+-------
      1 |            red |     BMW |  2000000 |   20 |   100
      2 |          black |   Geely |  1800000 |   10 |    80
      4 |        Wrangler |    JEEP | 42900000 |    9 |    60
      3 |         silver |  Bently | 10000000 |    8 |    50

(4 rows)
```

```
cqlsh:carsystem> select * from users;

 userid | address | dateofbirth | gender | password  | phone       | realname     | username
--------+---------+-------------+--------+-----------+-------------+--------------+----------
      1 |  bamstu |  1980-08-27 |      1 | 123456789 | 18801035010 | wangchaochao |  chaochao
      2 |  bamstu |  1988-09-29 |      1 | 265314789 | 79269565693 |     zhangwei |       wei
      4 |  bamstu |  1995-08-13 |      0 | 321654879 | 79298535695 |       huahua |    flower
      3 |  bamstu |  1998-09-29 |      0 | 987654321 | 79298764465 |        li li |       cat
```

Задание 2. CRUD и работа с индексами (базовая часть)

2.1. Продемонстрировать добавление, изменение и удаление данных в БД, используя команды API и/или язык Cassandra Query Language. Примеры (см [3]):

Добавление данных в БД

```
cqlsh:carsystem> insert into car(carsid,cardescription,carname,price,sold,stock) VALUES (4,'Wrangler','JEEP',42900000,9,60);
cqlsh:carsystem>
```

Изменение данных в БД

```
cqlsh:carsystem> UPDATE car SET cardescription='blue',
             ... carname='Benz',
             ... price=5000000,
             ... sold=15
             ... where carsid=1;
cqlsh:carsystem> select * from car where carsid=1;

 carsid | cardescription | carname | price   | sold | stock
--------+----------------+---------+---------+------+-------
      1 |           blue |    Benz | 5000000 |   15 |   100

(1 rows)
```

удаление данных в БД

```
cqlsh:carsystem> delete from car where carsid=4;
cqlsh:carsystem> select * from car;

 carsid | cardescription | carname | price    | sold | stock
--------+----------------+---------+----------+------+-------
      1 |           blue |    Benz |  5000000 |   15 |   100
      2 |          black |   Geely |  1800000 |   10 |    80
      3 |         silver |  Bently | 10000000 |    8 |    50

(3 rows)
```

2.2. Определить для семейства столбцов индекс(ы). Выполнить запросы к с фильтрацией по ключам и индексам. Продемонстрировать работу allow filtering. (см. примеры и описание https://habr.com/ru/post/205176/).

Перед добавлением индекса

```
cqlsh:carsystem> select * from car where carname = 'Benz';
InvalidRequest: Error from server: code=2200 [Invalid query] message="Cannot execute this query as it might involve data filter
ing and thus may have unpredictable performance. If you want to execute this query despite the performance unpredictability, us
e ALLOW FILTERING"
```

После добавления индекса

```
cqlsh:carsystem> CREATE INDEX carnameIndex ON car (carname);
cqlsh:carsystem> select * from car where carname='Benz';

 carsid | cardescription | carname | price   | sold | stock
--------+----------------+---------+---------+------+-------
      1 |           blue |    Benz | 5000000 |   15 |   100

(1 rows)
```

```
cqlsh:carsystem> CREATE INDEX soldIndex ON car (sold);
cqlsh:carsystem> select * from car where sold<10;
InvalidRequest: Error from server: code=2200 [Invalid query] m
ing and thus may have unpredictable performance. If you want t
e ALLOW FILTERING"
cqlsh:carsystem> select * from car where sold=10;

 carsid | cardescription | carname | price   | sold | stock
--------+----------------+---------+---------+------+-------
      2 |          black |   Geely | 1800000 |   10 |    80

(1 rows)
```

ALLOW FILTERING

```
cqlsh:carsystem> select * from car where stock=50 ALLOW FILTERING;

 carsid | cardescription | carname | price    | sold | stock
--------+----------------+---------+----------+------+-------
      3 |         silver |  Bently | 10000000 |    8 |    50

(1 rows)
```

Задание 3. Запросы к БД. (базовая часть)

3.1. Выполнить запросы к базе данных с селекцией и проекцией.

```
cqlsh:carsystem> select * from car;

 carsid | cardescription | carname | price    | sold | stock
--------+----------------+---------+----------+------+-------
      5 |            red |     BMW |  3000000 |   20 |   100
      1 |           blue |    Benz |  5000000 |   15 |   100
      2 |          black |   Geely |  1800000 |   10 |    80
      4 |       Wrangler |    JEEP | 42900000 |    9 |    60
      6 |            red |     BMW |  8000000 |   20 |   100
      3 |         silver |  Bently | 10000000 |    8 |    50

(6 rows)
cqlsh:carsystem> select * from car
             ... where price>5000000 allow filtering;

 carsid | cardescription | carname | price    | sold | stock
--------+----------------+---------+----------+------+-------
      4 |       Wrangler |    JEEP | 42900000 |    9 |    60
      6 |            red |     BMW |  8000000 |   20 |   100
      3 |         silver |  Bently | 10000000 |    8 |    50

(3 rows)
```

3.2. Выполнить запрос с использованием агрегатных функций.

Min:

```
cqlsh:carsystem> SELECT min(sold) AS min_sold FROM car;

 min_sold
----------
        8

(1 rows)
```

Sum:

```
cqlsh:carsystem> SELECT sum(sold) AS sum_sold FROM car;

 sum_sold
----------
       42

(1 rows)
```

3.3. Добавить строку с указанием TTL, продемонстрировать действие TTL

TTL

```
cqlsh:carsystem> insert into car(carsid, cardescription, carname, price, sold, stock) VALUES (5,'red','BMW',
2000000, 20, 100) USING TTL 30;
cqlsh:carsystem> select * from car;

 carsid | cardescription | carname | price    | sold | stock
--------+----------------+---------+----------+------+-------
      5 |            red |     BMW |  2000000 |   20 |   100
      1 |           blue |    Benz |  5000000 |   15 |   100
      2 |          black |   Geely |  1800000 |   10 |    80
      4 |       Wrangler |    JEEP | 42900000 |    9 |    60
      3 |         silver |  Bently | 10000000 |    8 |    50

(5 rows)
cqlsh:carsystem> select * from car;

 carsid | cardescription | carname | price    | sold | stock
--------+----------------+---------+----------+------+-------
      1 |           blue |    Benz |  5000000 |   15 |   100
      2 |          black |   Geely |  1800000 |   10 |    80
      4 |       Wrangler |    JEEP | 42900000 |    9 |    60
      3 |         silver |  Bently | 10000000 |    8 |    50

(4 rows)
```

Задание 4. Группировка и сортировка (Хорошо)

4.1. Выполнить запросы с группировкой и сортировкой данных.

```
cqlsh:carsystem> create table orders(id int, name text, cprice int, primary key(id, cprice, name));
```

```
cqlsh:carsystem> select * from orders;

 id | cprice    | name
----+-----------+------
  1 |   3000000 |  BMW
  1 |   4000000 |  BMW
  1 |   5000000 |  BMW
  2 | 100000000 | Benz

(4 rows)
```

сортировка данных

```
cqlsh:carsystem> SELECT * FROM orders WHERE id=1 ORDER BY cprice;

 id | cprice  | name
----+---------+------
  1 | 3000000 |  BMW
  1 | 4000000 |  BMW
  1 | 5000000 |  BMW

(3 rows)
```

Группировка данных

4.2. Создать еще одно семейство столбцов по теме ДЗ, определить для него кластерный и распределительный ключи. Выполнить запросы к с фильтрацией по ключам.

```
cqlsh:carsystem> create table orders(id int,name text,cprice int,primary key(id,cprice));
```

```
cqlsh:carsystem> select * from orders;

 id | cprice    | name
----+-----------+-------
  1 |   3000000 |   BMW
  1 |   4000000 |   BMW
  1 |   5000000 |   BMW
  2 | 100000000 |  Benz
  2 | 200000000 |  Benz
  3 |   1000000 | Geely
  3 |   2000000 | Geely
```

```
cqlsh:carsystem> select * from orders where id=1 and cprice>3000000;

 id | cprice  | name
----+---------+------
  1 | 4000000 |  BMW
  1 | 5000000 |  BMW
```

4.3. Продемонстрировать усечение таблицы и удаление таблицы/индекса

усечение таблицы

```
cqlsh:carsystem>  TRUNCATE orders;
cqlsh:carsystem> select * from orders;

 id | cprice | name
----+--------+------

(0 rows)
```

удаление таблицы

```
cqlsh:carsystem> drop table orders;
cqlsh:carsystem> describe tables;

car  users  numberofrequests
```

Удаление индекса

```
cqlsh:carsystem> CREATE INDEX sprice ON orders(cprice);
```

```
    AND speculative_retry = '99PERCENTILE';
CREATE INDEX sprice ON carsystem.orders (cprice);

cqlsh:carsystem> drop index sprice;
cqlsh:carsystem> describe table orders;

CREATE TABLE carsystem.orders (
    id int,
    cprice int,
    name text,
    PRIMARY KEY (id, cprice)
) WITH CLUSTERING ORDER BY (cprice ASC)
    AND bloom_filter_fp_chance = 0.01
    AND caching = {'keys': 'ALL', 'rows_per_partition': 'NONE'}
    AND comment = ''
    AND compaction = {'class': 'org.apache.cassandra.db.compaction.SizeTieredCompactionStrategy
', 'max_threshold': '32', 'min_threshold': '4'}
    AND compression = {'chunk_length_in_kb': '64', 'class': 'org.apache.cassandra.io.compress.L
Z4Compressor'}
    AND crc_check_chance = 1.0
    AND dclocal_read_repair_chance = 0.1
    AND default_time_to_live = 0
    AND gc_grace_seconds = 864000
    AND max_index_interval = 2048
    AND memtable_flush_period_in_ms = 0
    AND min_index_interval = 128
    AND read_repair_chance = 0.0
    AND speculative_retry = '99PERCENTILE';
```

Задание 5. Дополнительные возможности (Отлично)

5.1. Создать материализованное представление. Продемонстрировать работу с ним.

```
cqlsh:carsystem> CREATE MATERIALIZED VIEW orders_by_price AS
             ... select * from orders
             ... where cprice>=4000000 and cprice<=20000000
             ... primary key (id, cprice)
             ... with comment='filter price';
cqlsh:carsystem> select * from orders_by_price;

 id | cprice   | name
----+----------+------
  1 |  4000000 |  BMW
  1 |  5000000 |  BMW

(2 rows)
```

5.2. Продемонстрировать создание пакета запросов

```
cqlsh> use carsystem;
cqlsh:carsystem>  begin batch insert into car(carsid,cardescription,carname,price,sold,stock) VALUES (7,'Wrangler','JEEP',5000000,9,60);
insert into car(carsid,cardescription,carname,price,sold,stock) VALUES (8,'Wrangler','BMW',6000000,8,66);apply batch;
cqlsh:carsystem> select * from car;

 carsid | cardescription | carname | price    | sold | stock
--------+----------------+---------+----------+------+-------
      5 |            red |     BMW |  3000000 |   20 |   100
      1 |           blue |    Benz |  5000000 |   15 |   100
      8 |       Wrangler |     BMW |  6000000 |    8 |    66
      2 |          black |   Geely |  1800000 |   10 |    80
      4 |       Wrangler |    JEEP | 42900000 |    9 |    60
      7 |       Wrangler |    JEEP |  5000000 |    9 |    60
      6 |            red |     BMW |  8000000 |   20 |   100
      3 |         silver |  Bently | 10000000 |    8 |    50

(8 rows)
```

5.3. Продемонстрировать изменение и удаление данных в БД, используя условия

```
cqlsh:carsystem> delete from car where carsid=8;
cqlsh:carsystem> select * from car;

 carsid | cardescription | carname | price    | sold | stock
--------+----------------+---------+----------+------+-------
      5 |            red |     BMW |  3000000 |   20 |   100
      1 |           blue |    Benz |  5000000 |   15 |   100
      2 |          black |   Geely |  1800000 |   10 |    80
      4 |       Wrangler |    JEEP | 42900000 |    9 |    60
      7 |       Wrangler |    JEEP |  5000000 |    9 |    60
      6 |            red |     BMW |  8000000 |   20 |   100
      3 |         silver |  Bently | 10000000 |    8 |    50

(7 rows)
```

Update:

```
cqlsh:carsystem> update car set carname ='Benz' where carsid=5;
cqlsh:carsystem> select * from car;

 carsid | cardescription | carname | price    | sold | stock
--------+----------------+---------+----------+------+-------
      5 |            red |    Benz |  3000000 |   20 |   100
      1 |           blue |    Benz |  5000000 |   15 |   100
      2 |          black |   Geely |  1800000 |   10 |    80
      4 |       Wrangler |    JEEP | 42900000 |    9 |    60
      7 |       Wrangler |    JEEP |  5000000 |    9 |    60
      6 |            red |     BMW |  8000000 |   20 |   100
      3 |         silver |  Bently | 10000000 |    8 |    50

(7 rows)
```

Литература:

1. Фаулер, Мартин, Садаладж, Прамодкумар Дж. NoSQL: новая методология разработки нереляционных баз данных. : Пер. с англ. - М.: ООО "И.Д. Вильямс", 2013г.

2. Теория - https://habrahabr.ru/post/155115/

3. Документация - http://cassandra.apache.org/doc/latest/ ; Язык CQL - http://cassandra.apache.org/doc/latest/cql/index.html