

Департамент образования и науки города Москвы
Государственное автономное образовательное учреждение
высшего образования города Москвы
«Московский городской педагогический университет»
Институт цифрового образования
Департамент информатики управления и технологий

Кузьмина Дарья Юрьевна БД-241м

Инструменты хранения и анализа больших данных

Лабораторная работа 2.1. Часть 2.
Cassandra
Вариант 11

Направление подготовки/специальность
38.04.05 - Бизнес-информатика
Бизнес-аналитика и большие данные
(очная форма обучения)

Руководитель дисциплины:
Босенко Т.М., доцент департамента
информатики, управления и технологий,
доктор экономических наук

Москва
2025

Содержание

Введение.....	2
Основная часть	2
Заключение	21

Введение

Цель:

Получить практические навыки работы с базой данных Cassandra, изучив основные операции по управлению данными, включая создание и использование ключспейсов, таблиц, выполнение запросов CQL, а также работу с различными инструментами подключения и администрирования.

Задачи:

1. Подключиться к Cassandra через cqlsh или браузерные интерфейсы (Cassandra Web, Apache Zeppelin).
2. Создать ключспейс с заданной стратегией репликации (SimpleStrategy).
3. Создать таблицы для хранения данных (например, movies, actors) с использованием примитивных типов и коллекций.
4. Выполнить операции CRUD (добавление, выборка, обновление, удаление данных) в созданных таблицах.
5. Изучить метаданные ключспейсов и таблиц с помощью команд DESCRIBE и запросов к системным таблицам.

Основная часть

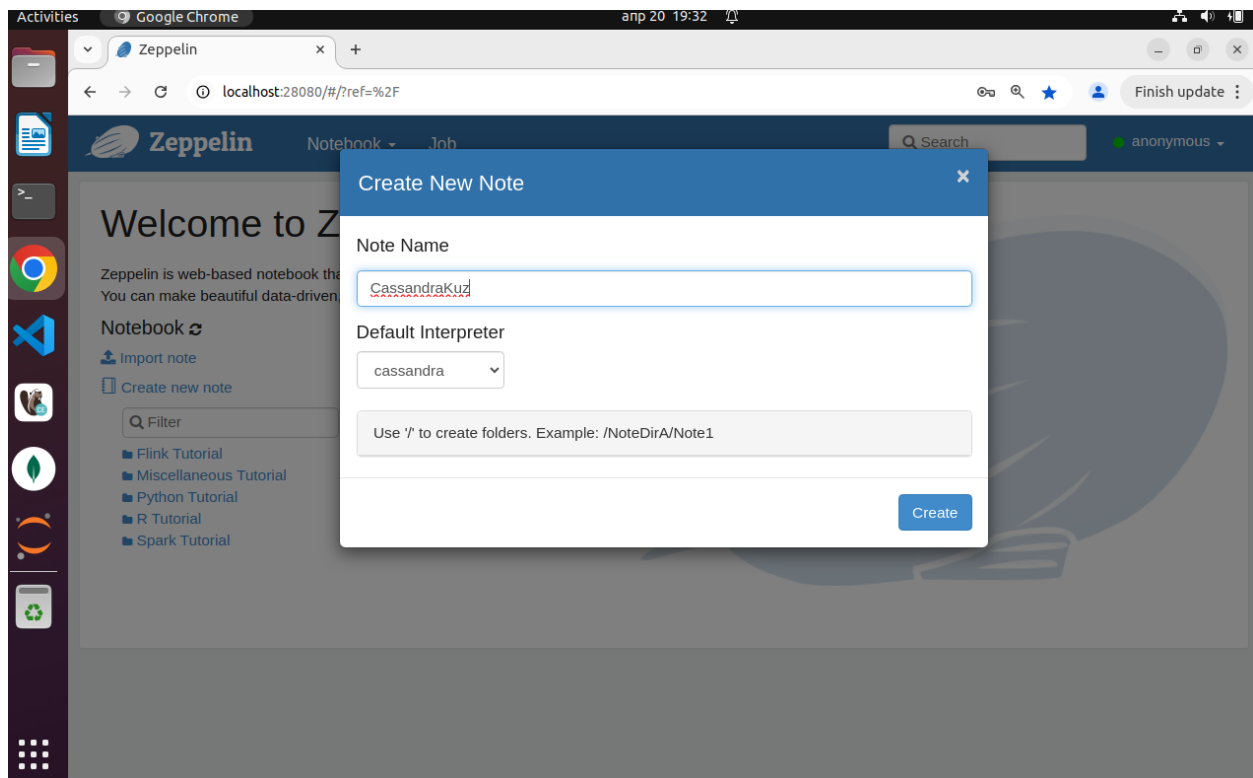
Задача 1.

В виртуальной машине проверяем, что сервер запущен:

```
● dev@dev-vm: ~/Downloads/dba/nonrel/mongo$ cd ..  
● dev@dev-vm: ~/Downloads/dba/nonrel$ cd cassandra
```

```
● dev@dev-vm: ~/Downloads/dba/nonrel/cassandra$ docker ps  
CONTAINER ID   IMAGE                                COMMAND                  CREATED        STATUS        PORTS  
b7804eb56d    mongo:7.0.17-rc1-jammy             "docker-entrypoint.s..." 56 minutes ago Up 56 minutes 0  
a5bb7206afbd  postgres:16                        "docker-entrypoint.s..." 6 weeks ago   Up About an hour 0  
ec4919b750da  dpage/pgadmin4:latest              "/entrypoint.sh"         6 weeks ago   Up About an hour 0  
0.0.0.0:27017->27017/tcp, [::]:27017->27017/tcp  mongo-1  
0.0.0.0:5432->5432/tcp, [::]:5432->5432/tcp  postgres16  
0.0.0.0:80->80/tcp, [::]:80->80/tcp, 443/tcp  pgadmin
```

Убеждаемся, что запущены 3 контейнера



Открываем ноутбук

The screenshot shows the Cassandra Web interface in a Google Chrome browser. The address bar shows `localhost:28200`. The interface has a sidebar on the left with icons for various tools. The main content area is divided into two columns. The left column, titled 'Keyspaces', lists several system keyspaces: `system_auth` (4), `system_schema` (10), `system_distributed` (3), `system` (17), and `system_traces` (2). The right column, titled 'Cluster Status', shows the 'Hosts' section with a table for 'Datacenter: datacenter1'. The table has columns for 'ip', 'id', 'rack', 'version', and 'status'. One host is listed with IP `172.21.0.4`, ID `8e14859e-b0f0-4c02-995b-6116f3347130`, rack `rack1`, version `3.11.19`, and status `up`. Below the hosts section, the 'Keyspaces' section shows the creation commands for `system_auth`, `system_schema`, `system_distributed`, and `system`.

ip	id	rack	version	status
172.21.0.4	8e14859e-b0f0-4c02-995b-6116f3347130	rack1	3.11.19	up

Смотрим, работает ли кассандра, проверяем доступность кластера (1 нода).

The screenshot shows a terminal window with a `docker-compose.yml` file open in the background. The terminal output shows the execution of the command `sudo docker exec -ti cassandra-1 cqlsh -u cassandra -p cassandra`. The output indicates a successful connection to the `CassandraCluster` at `127.0.0.1:9042`. The terminal shows the prompt `cassandra@cqlsh>`.

```
dev@dev-vm:~/Downloads/dba/nonrel/cassandra$ sudo docker exec -ti cassandra-1 cqlsh -u cassandra -p cassandra
[sudo] password for dev:
Connected to CassandraCluster at 127.0.0.1:9042.
[cqlsh 5.0.1 | Cassandra 3.11.19 | CQL spec 3.4.4 | Native protocol v4]
Use HELP for help.
cassandra@cqlsh>
```

Проверим запуск

```
[cqlsh 5.0.1 | Cassandra 3.11.19 | CQL spec 3.4.4 | Native protocol v4]
Use HELP for help.
cassandra@cqlsh> SELECT * FROM system_schema.keyspaces;

keyspace_name | durable_writes | replication
-----
system_auth | True | {'class': 'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor': '1'}
system_schema | True | {'class': 'org.apache.cassandra.locator.LocalStrategy'}
system_distributed | True | {'class': 'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor': '3'}
system | True | {'class': 'org.apache.cassandra.locator.LocalStrategy'}
system_traces | True | {'class': 'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor': '2'}

(5 rows)
cassandra@cqlsh> █
```

Посмотрим ключи по схемам

cassandra %cassandra ●

edit restart remove

Option

The interpreter will be instantiated Globally in shared process ⓘ

☐ Connect to existing process

☐ Set permission

Properties

Name	Value	Description
cassandra.hosts	cassandra-1	Comma separated Cassandra hosts (DNS name or IP address). Default = localhost. Ex: '192.168.0.12,node2,node3'
cassandra.native.port	9042	Cassandra native port. Default = 9042
cassandra.protocol.version	DEFAULT	Cassandra protocol version. Default = auto-detect

Меняем имя хоста

CassandraKuz - Zeppelin x Cassandra Web x nosql-workshop/03-workshop x +

localhost:28080/#/notebook/2KV6DWRCV

Zeppelin Notebook Job Search anonymous

CassandraKuz

SELECT * FROM system_schema.keyspaces; FINISHED

keyspace_name	durable_writes	replication
system_auth	true	{class: org.apache.cassandra.locator.SimpleStrategy, replication_factor: 1}
system_schema	true	{class: org.apache.cassandra.locator.LocalStrategy}
system_distributed	true	{class: org.apache.cassandra.locator.SimpleStrategy, replication_factor: 3}
system	true	{class: org.apache.cassandra.locator.LocalStrategy}
system_traces	true	{class: org.apache.cassandra.locator.SimpleStrategy, replication_factor: 1}

Took 55 sec. Last updated by anonymous at April 21 2025, 10:30:18 PM. (outdated)

READY

Указываем настройки репликации

CREATE KEYSPACE movieskuz WITH replication = {'class': 'SimpleStrategy', 'replication_factor': 1}; FINISHED

No Result [Last query execution info](#)

Took 6 sec. Last updated by anonymous at April 21 2025, 10:39:03 PM.

Появился объект

Keyspaces

system_auth 4

system_schema 10

system_distributed 3

system 17

system_traces 2

movieskuz 0

Hosts

172.21.0.4 (up)

Keyspace: movieskuz

Tables Definition

Name

Total: 0 tables

movieskuz ▼

DESCRIBE KEYSPACE movieskuz;

Legend ▼

movieskuz

Replication

Durable Writes

{ 'class' : 'org.apache.cassandra.locator.SimpleStrategy', 'replication_factor' : '1' }

true

As CQL statement ▼

Took 24 sec. Last updated by anonymous at April 21 2025, 10:41:23 PM.

Создаем таблицы movie и actor

```
DROP TABLE IF EXISTS movieskuz.movie;
CREATE TABLE movieskuz.movie (movie_id int,
    title text,           // title
    release_year int,     // year
    running_time int,     // runtimes
    languages set<text>,  // language codes
    genres set<text>,     // genres
    plot_outline text,    // plot outline
    cover_url text,       // cover url
    top250_rank int,      // top 250 rank
    PRIMARY KEY (movie_id)
);
```

No Result

[Last query execution info](#)

Took 3 sec. Last updated by anonymous at April 21 2025, 10:43:22 PM.

```
DROP TABLE IF EXISTS movieskuz.actor;
CREATE TABLE movieskuz.actor (actor_id int,
    name text,           // name
    headshot_url text,   // headshot
    mini_biography text, // mini biography
    birth_date text,     // birth date
    trade_mark list<text>, // trade mark
    PRIMARY KEY (actor_id)
);
```

No Result

[Last query execution info](#)

Took 2 sec. Last updated by anonymous at April 21 2025, 10:43:53 PM.

DESCRIBE TABLE movie;

FINISHED

DESCRIBE TABLE movie;

Legend

movie		
Column Type	Column Name	Data Type
	movie_id	int
	cover_url	text
	genres	set<text>
	languages	set<text>
	plot_outline	text
	release_year	int
	running_time	int
	title	text
	top250_rank	int
As CQL statement		

Получим описание таблицы

```
{'Action', 'Sci-Fi'},
$$Thomas A. Anderson is a man living two lives. By day he is an average computer programmer and by night a hacker known as Neo. Neo has always questioned his
reality, but the truth is far beyond his imagination. Neo finds himself targeted by the police when he is contacted by Morpheus, a legendary computer hacker
branded a terrorist by the government. Morpheus awakens Neo to the real world, a ravaged wasteland where most of humanity have been captured by a race of
machines that live off of the humans' body heat and electrochemical energy and who imprison their minds within an artificial reality known as the Matrix. As a
rebel against the machines, Neo must return to the Matrix and confront the agents: super-powerful computer programs devoted to snuffing out Neo and the entire
human rebellion$.
'https://m.media-amazon.com/images/M/MV5BNzQzOTk3OTAtNDQ0Zi00ZTVkLWl0MTEmD1lZjNkYzNjNTc4L2ltYmclXkEyXkFqcGdeQXVyNjU0OTQ0OTY@._V1_SX101_CR0,0,101,150_.jpg',
19);

// insert "Pulp Fiction" - 0110912
INSERT INTO movieskuz.movie (movie_id, title, release_year, running_time, languages,
genres, plot_outline, cover_url, top250_rank)
VALUES (0110912,
'Pulp Fiction',
1994,
154,
{'en', 'es', 'fr'},
{'Crime', 'Drama'},
$$Jules Winnfield (Samuel L. Jackson) and Vincent Vega (John Travolta) are two hit men who are out to retrieve a suitcase stolen from their employer, mob boss
Marsellus Wallace ( Ving Rhames). Wallace has also asked Vincent to take his wife Mia (Uma Thurman) out a few days later when Wallace himself will be out of
town. Butch Coolidge (Bruce Willis) is an aging boxer who is paid by Wallace to lose his fight. The lives of these seemingly unrelated people are woven
together comprising of a series of funny, bizarre and uncalled-for incidents.$$
'https://m.media-amazon.com/images/M/MV5BNGNlMDIzZTU0NTBlZi00MTk1LWFjM2ItYzViMjE3YzI5IjMjXkEyXkFqcGdeQXVyNzkwMjQ5Nz@._V1_SY150_CR1,0,101,150_.jpg',
8);

// insert "Speed" - 0111257
INSERT INTO movieskuz.movie (movie_id, title, release_year, running_time, languages,
genres, plot_outline, cover_url, top250_rank)
VALUES (0111257,
'Speed',
1994,
116,
{'en'},
{'Action', 'Adventure', 'Crime', 'Thriller'},
$$Bomber extortionist's elevator plan backfires, so he rigs a bomb to a LA city bus. The stipulation is: once armed, the bus must stay above 50 mph to keep
from exploding. Also if LAPD Officer tries to unload any passengers off, Payne will detonate it. Joe Morton co-stars as Jack's superior, and Jeff Daniels
supports Jack helping him try to defuse the bomb.$$
'https://m.media-amazon.com/images/M/MV5BYjc0MjYyN2EtZGRhMy00NzJlLWI2Y2Q0YzhiYTU3NzAxNzg4XkEyXkFqcGdeQXVyMTQxNzMzNDI@._V1_SY150_CR0,0,101,150_.jpg',
null);
```

No Result Last query execution info

Took 0 sec. Last updated by anonymous at April 21 2025, 10:45:47 PM.

Загоняем данные о фильмах

No Result Last query execution info

```
-- insert Quentin Tarantino' - 0000233
INSERT INTO movieskuz.actor (actor_id, name, headshot_url, birth_date, trade_mark)
VALUES (0000233,
        'Quentin Tarantino',
        'https://m.media-amazon.com/images/M/MV5BMjYtMGMxOTQzODAzNlSbM1SbanBnXkFtZTcwNmYyZWYwOXB0e..._Vi_Ux67_CR0_0,67,98_AL_.jpg',
        '1963-03-27',
        '[Leads characters usually drive General Motors vehicles, particularly Chevrolet and Cadillac, such as Jules 1974 Nova and Vincents 1960s Malibu.',
        'Briefcases and suitcases play an important role in Pulp Fiction (1994), Reservoir Dogs (1992), Jackie Brown (1997), True Romance (1993) and Kill Bill: Vol. 1 & 2 (2004).',
        'Makes references to cult movies and television'.
        'Frequently works with Harvey Keitel, Tim Roth, Michael Madsen, Uma Thurman, Michael Bowen, Samuel L. Jackson, Michael Parks and Christoph Waltz.'.
        'His films usually have a shot from inside an automobile trunk'.
        'He always has a Dutch element in his films: the opening tune, Little Green Bag, in Reservoir Dogs (1992) was performed by George Baker Selection and written by Jan Gerbrand Visser and Benjamin Bouwens who are all Dutch. The character Freddy Newandyke, played by Tim Roth is a direct translation to a typical Dutch last name, Nieuwendijk. The code name of Tim Roth is Mr. Orange, the royal color of Holland and the last name of the royal family. The Amsterdam conversation in Pulp Fiction (1994), Vincent Vega smokes from a Dutch tobacco shag (Drum), the mentioning of Rutger Hauer in Jackie Brown (1997), the brides name is Beatrix, the name of the Royal Dutch Queen.'.
        '[The Mexican Standoff] All his movies (including True Romance (1993), which he only wrote and did not direct) feature a scene in which three or more characters are pointing guns at each other at the same time.'.
        'Often uses an unconventional storytelling device in his films, such as retrospect (Reservoir Dogs (1992)), non-linear (Pulp Fiction (1994)), or "chapter" format (Kill Bill: Vol. 1 (2003)).',
        'His films will often include one long, unbroken take where a character is followed around somewhere.'));
```

No Result Last query execution info

Проверим результаты загрузки

Поскольку кассандра не позволяет ограничивать поиск по столбцу, кроме первичного ключа, пропусти поиск по имени актера и выполним запрос на поле actor id

SELECT *
FROM movieskuz.actor
WHERE actor_id = 0000246;

FINISHED

settings

actor_id	birth_date	headshot_url	mini_biography	name	trade_mark
246	1955-03-19	https://m.media-amazon.com/images/M/MV5BMjA0MjMzMTE5OF5BMl5BanBnXkFtZTcwMzQ2ODE3Mw@@_V1_UY98_CR8,0,67,98_AL_.jpg	null	Bruce Willis	[Frequently plays a man who suffered a tragedy, had lost something or had a crisis of confidence or conscience., Frequently plays likeable wisecracking heroes with a moral centre, Headlines action-adventures, often playing a policeman, hitman or

Took 0 sec. Last updated by anonymous at April 21 2025, 10:54:51 PM. (outdated)

READY

Включим фильтрацию

SELECT *
FROM movieskuz.actor
WHERE name = 'Bruce Willis'
ALLOW FILTERING;

FINISHED

settings

actor_id	birth_date	headshot_url	mini_biography	name	trade_mark
246	1955-03-19	https://m.media-amazon.com/images/M/MV5BMjA0MjMzMTE5OF5BMl5BanBnXkFtZTcwMzQ2ODE3Mw@@_V1_UY98_CR8,0,67,98_AL_.jpg	null	Bruce Willis	[Frequently plays a man who suffered a tragedy, had lost something or had a crisis of confidence or conscience., Frequently plays likeable wisecracking heroes with a moral centre, Headlines action-adventures, often playing a policeman, hitman or

Однако это неэффективно

UPDATE movieskuz.actor
SET name = 'Bruce Walter Willis'
WHERE actor_id = 0000246;

FINISHED

No Result [Last query execution info](#)

Took 0 sec. Last updated by anonymous at April 21 2025, 10:57:18 PM.

SELECT *
FROM movieskuz.actor
WHERE actor_id = 0000246;

FINISHED

settings

actor_id	birth_date	headshot_url	mini_biography	name	trade_mark
246	1955-03-19	https://m.media-amazon.com/images/M/MV5BMjA0MjMzMTES0F5BMl5BanBnXkFtZTcwMzQ2ODE3Mw@@._V1_UY98_CR8,0,67,98_AL_.jpg	null	Bruce Walter Willis	[Frequently plays a man who suffered a tragedy, had lost something or had a crisis of confidence or conscience., Frequently plays likeable wisecracking heroes with a moral centre, Headlines action-adventures, often playing

Обновили данные о Брюсе Уиллисе

UPDATE movieskuz.actor
SET name = 'Cool Actor'
WHERE actor_id = 99999999;

FINISHED

No Result [Last query execution info](#)

Took 0 sec. Last updated by anonymous at April 21 2025, 10:58:42 PM.

DELETE FROM movieskuz.actor
WHERE actor_id = 99999999;

FINISHED

No Result [Last query execution info](#)

Took 0 sec. Last updated by anonymous at April 21 2025, 10:59:11 PM.

INSERT INTO movieskuz.actor (actor_id, name)
VALUES (99999998, 'An other actor')
USING TTL 120;

FINISHED

No Result [Last query execution info](#)

Took 0 sec. Last updated by anonymous at April 21 2025, 10:59:46 PM.

Добавили и удалили пользователя, создали тестового пользователя на 120 секунд

SELECT TTL(name), actor_id, name
FROM movieskuz.actor;

FINISHED




settings

"ttl(name)"	actor_id	name
null	237	John Travolta
null	113	Sandra Bullock
null	233	Quentin Tarantino
null	168	Samuel L. Jackson
null	235	Uma Thurman
null	206	Keanu Reeves
117	99999998	An other actor
null	246	Bruce Walter Willis

Took 0 sec. Last updated by anonymous at April 21 2025, 11:02:53 PM.

Создадим динамические таблицы Фильмы на акетрам и актеры по фильмам




```
DROP TABLE IF EXISTS movieskuz.movies_by_actor;
CREATE TABLE movieskuz.movies_by_actor (actor_id int,
movie_id int,
title text,
PRIMARY KEY (actor_id, movie_id)
);
```

FINISHED   

No Result [Last query execution info](#)

Took 3 sec. Last updated by anonymous at April 21 2025, 11:05:48 PM.

```
DROP TABLE IF EXISTS movieskuz.actors_by_movie;
CREATE TABLE movieskuz.actors_by_movie (movie_id int,
title text STATIC,
actor_id int,
name text,
PRIMARY KEY (movie_id, actor_id)
);
```

FINISHED   

No Result [Last query execution info](#)

Took 2 sec. Last updated by anonymous at April 21 2025, 11:06:38 PM.

Вставляем данные в две таблицы

```
// Фильмы для актера "Bruce Willis"

INSERT INTO movieskuz.movies_by_actor (actor_id, movie_id, title)
VALUES (0000246, 0110912, 'Pulp Fiction');

INSERT INTO movieskuz.movies_by_actor (actor_id, movie_id, title)
VALUES (0000246, 1606378, 'A Good Day to Die Hard');

INSERT INTO movieskuz.movies_by_actor (actor_id, movie_id, title)
VALUES (0000246, 0217869, 'Unbreakable');

INSERT INTO movieskuz.movies_by_actor (actor_id, movie_id, title)
VALUES (0000246, 0377917, 'The Fifth Element');

INSERT INTO movieskuz.movies_by_actor (actor_id, movie_id, title)
VALUES (0000246, 0112864, 'Die Hard: With a Vengeance');

// Фильмы для актера "Keanu Reeves"

INSERT INTO movieskuz.movies_by_actor (actor_id, movie_id, title)
VALUES (0000206, 0133093, 'The Matrix');

INSERT INTO movieskuz.movies_by_actor (actor_id, movie_id, title)
VALUES (0000206, 0234215, 'The Matrix Reloaded');

INSERT INTO movieskuz.movies_by_actor (actor_id, movie_id, title)
VALUES (0000206, 0111257, 'Speed');

// Фильмы для актера "Sandra Bullock"

INSERT INTO movieskuz.movies_by_actor (actor_id, movie_id, title)
VALUES (0000113, 0111257, 'Speed');
```

No Result [Last query execution info](#)

Took 0 sec. Last updated by anonymous at April 21 2025, 11:08:05 PM.

```
// Актеры для фильма "The Matrix"

INSERT INTO movieskuz.actors_by_movie (movie_id, title, actor_id, name)
VALUES (0133093, 'The Matrix', 0000206, 'Keanu Reeves');

INSERT INTO movieskuz.actors_by_movie (movie_id, title, actor_id, name)
VALUES (0133093, 'The Matrix', 0000401, 'Laurence Fishburne');

INSERT INTO movieskuz.actors_by_movie (movie_id, title, actor_id, name)
VALUES (0133093, 'The Matrix', 0005251, 'Carrie-Anne Moss');

INSERT INTO movieskuz.actors_by_movie (movie_id, title, actor_id, name)
VALUES (0133093, 'The Matrix', 0915989, 'Hugo Weaving');

// Актеры для фильма "Pulp Fiction"

INSERT INTO movieskuz.actors_by_movie (movie_id, title, actor_id, name)
VALUES (0110912, 'Pulp Fiction', 0000237, 'John Travolta');

INSERT INTO movieskuz.actors_by_movie (movie_id, title, actor_id, name)
VALUES (0110912, 'Pulp Fiction', 0000168, 'Samuel L. Jackson');

INSERT INTO movieskuz.actors_by_movie (movie_id, title, actor_id, name)
VALUES (0110912, 'Pulp Fiction', 0000246, 'Bruce Willis');
```

No Result **Last query execution info**

Took 0 sec. Last updated by anonymous at April 21 2025, 11:09:10 PM.

Выведем все фильмы, в которых снимался Брюс Уиллис и всех актеров
Криминального чтива

settings

FINISHED

title
Pulp Fiction
Die Hard: With a Vengeance
Unbreakable
The Fifth Element
A Good Day to Die Hard

Took 0 sec. Last updated by anonymous at April 21 2025, 11:09:37 PM. (outdated)

settings

FINISHED

name	title
Samuel L. Jackson	Pulp Fiction
John Travolta	Pulp Fiction
Bruce Willis	Pulp Fiction

Использование столбцов-счетчиков

Создаем статическую таблицу рейтинга со столбцами типа counter, введем данные рейтинга

```
DROP TABLE IF EXISTS movieskuz.rating_by_movie;
CREATE TABLE movieskuz.rating_by_movie (movie_id int,
one_star counter,
two_star counter,
three_star counter,
four_star counter,
five_star counter,
PRIMARY KEY (movie_id)
);
```

FINISHED

No Result [Last query execution info](#)

Took 2 sec. Last updated by anonymous at April 21 2025, 11:14:28 PM.

```
UPDATE movieskuz.rating_by_movie
SET five_star = five_star + 1
WHERE movie_id = 0110912;

UPDATE movieskuz.rating_by_movie
SET four_star = four_star + 1
WHERE movie_id = 0110912;

UPDATE movieskuz.rating_by_movie
SET five_star = five_star + 1
WHERE movie_id = 0110912;

UPDATE movieskuz.rating_by_movie
SET five_star = five_star + 1
WHERE movie_id = 0110912;

UPDATE movieskuz.rating_by_movie
SET two_star = two_star + 1
WHERE movie_id = 0110912;
```

READY

Результат

|| SELECT * FROM movieskuz.rating_by_movie WHERE movie_id = 0110912; FINISHED

settings

movie_id	five_star	four_star	one_star	three_star	two_star
110912	3	1	null	null	1

Создадим таблицу просмотров с сортировкой по половому признаку году и месяцу просмотра, дополним ее данными.

```
DROP TABLE IF EXISTS movieskuz.movie_viewed_by_time;
CREATE TABLE movieskuz.movie_viewed_by_time (movie_id int,
      year int,
      month int,
      male counter,
      female counter,
      PRIMARY KEY (movie_id, year, month)
) WITH CLUSTERING ORDER BY (year DESC, month DESC);
```

No Result [Last query execution info](#)

Took 2 sec. Last updated by anonymous at April 21 2025, 11:17:38 PM.

```
// Pulp Fiction Views 2019/03
UPDATE movieskuz.movie_viewed_by_time
SET male = male + 1
WHERE movie_id = 0110912 AND year = 2019 and month = 03;

UPDATE movieskuz.movie_viewed_by_time
SET male = male + 1
WHERE movie_id = 0110912 AND year = 2019 and month = 03;

UPDATE movieskuz.movie_viewed_by_time
SET female = female + 1
WHERE movie_id = 0110912 AND year = 2019 and month = 03;

// Pulp Fiction Views 2019/04
UPDATE movieskuz.movie_viewed_by_time
SET female = female + 1
WHERE movie_id = 0110912 AND year = 2019 and month = 04;
```

Выведем данные

```
SELECT *
FROM movieskuz.movie_viewed_by_time
WHERE movie_id = 0110912;
```

FINISHED ▶ ⌕ 📖 ⚙️






       settings ▼

movie_id	year	month	female	male
110912	2019	5	2	2
110912	2019	4	3	1
110912	2019	3	1	2

Криминальное чтиво за все время

```
SELECT *
FROM movieskuz.movie_viewed_by_time
WHERE movie_id = 0110912 AND year = 2019 AND month = 05;
```

FINISHED ▶ ⌕ 📖 ⚙️

       settings ▼

movie_id	year	month	female	male
110912	2019	5	2	2

За месяцц

SELECT *
FROM movieskuz.movie_viewed_by_time
WHERE movie_id = 0110912 AND year = 2019 AND month >= 01 AND month <= 5;

FINISHED

movie_id	year	month	female	male
110912	2019	5	2	2
110912	2019	4	3	1
110912	2019	3	1	2

С января по май

Индивидуальное задание **Вариант 11**

1. Создайте ключспейс shops с репликацией SimpleStrategy и коэффициентом репликации 1.
2. Создайте таблицу products в ключспейсе shops с полями product_id (int), name (text), price (float), category (text), stock (int) и первичным ключом product_id.
3. Вставьте четыре продукта в таблицу products.
4. Выберите все продукты из таблицы products.
5. Обновите поле stock продукта с product_id = 2.

Добавляем свои данные с гитхаба.

Судя по лекции, концептуально, это задание выглядело бы так:

1. Создание ключспейса shops

Для начала работы необходимо создать ключспейс (пространство ключей) с именем "shops", используя стратегию репликации SimpleStrategy и коэффициент репликации 1 (так как мы работаем с одним узлом Cassandra):

```
CREATE KEYSPACE shops WITH replication = {
    'class': 'SimpleStrategy',
    'replication_factor': 1
};
```


Эта команда создает новое пространство ключей, где:

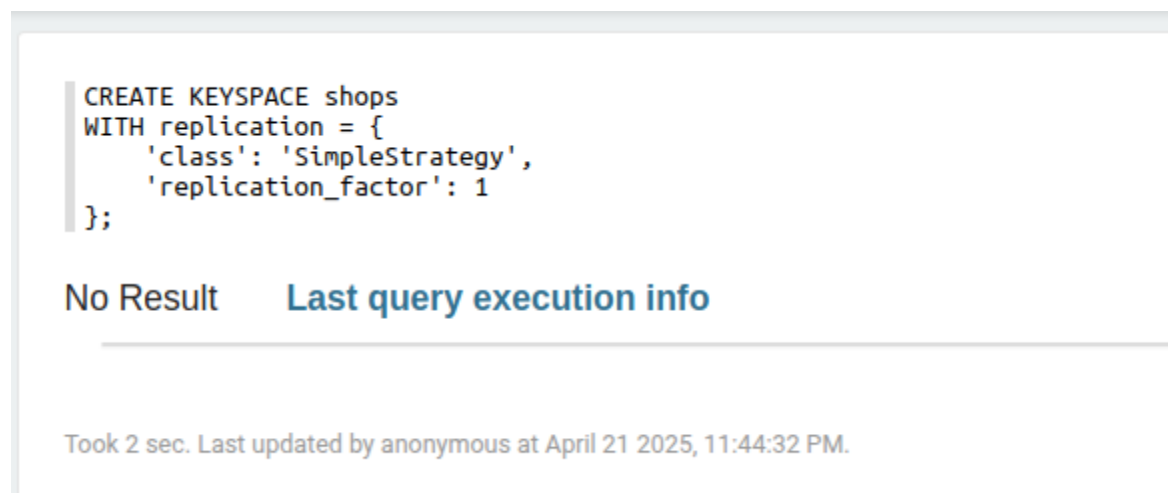
- `SimpleStrategy` - простая стратегия репликации, подходящая для однодатецентрированных кластеров
- `replication_factor: 1` - указывает, что каждая запись будет храниться только на одном узле

Индивидуальное задание

" data-bbox="103 244 664 494"/>

Создаем новый блокнот

Создаем новый кейспейс по заданию



```
CREATE KEYSPACE shops
WITH replication = {
  'class': 'SimpleStrategy',
  'replication_factor': 1
};
```

No Result [Last query execution info](#)

Took 2 sec. Last updated by anonymous at April 21 2025, 11:44:32 PM.

Keyspaces

system_auth	4
system_schema	10
system_distributed	3
system	17
system_traces	2
movieskuz	6
shops	0

Keyspace: shops

Tables		Definition	
Name		Actions	
Total: 0 tables		+ Create	

2. Создание таблицы shops

После создания ключспейса переходим в него и создаем таблицу products:

```
USE shops;
```

```
CREATE TABLE products (  
    product_id int,  
    name text,  
    price float,  
    category text,  
    stock int,  
    PRIMARY KEY (product_id)  
);
```

Таблица содержит следующие поля:

- `product_id` - уникальный идентификатор продукта (первичный ключ)
- `name` - название продукта
- `price` - цена продукта

- `category` - категория продукта
- `stock` - количество товара на складе

```
USE shops;

CREATE TABLE products (
  product_id int PRIMARY KEY,
  name text,
  price float,
  category text,
  stock int
);
```

No Result **Last query execution info**

Took 2 sec. Last updated by anonymous at April 22 2025, 12:02:02 AM.

Создали таблицу шопс

DESCRIBE TABLE products;

FINISHED ▶ ⌵ 📖 ⚙

DESCRIBE TABLE products;

Legend ▾

products		
Column Type	Column Name	Data Type
PK	product_id	int
	category	text
	name	text
	price	float
	stock	int
As CQL statement ▾		

3. Вставка данных в таблицу products

Добавим четыре продукта в таблицу:

FINISHED

Took 0 sec. Last updated by anonymous at April 22 2025, 12:07:17 AM.

FINISHED

4. Выборка всех продуктов

```
SELECT * FROM products;
```

FINISHED    

5. Обновление количества товара на складе

```
UPDATE products SET stock = 30 WHERE product_id = 2;
```

После выполнения этой команды количество товара "Smartphone" изменится с 25 до 30.

Убедимся, что обновление прошло успешно:

```
SELECT * FROM products WHERE product_id = 2;
```

```
UPDATE products
SET stock = 20
WHERE product_id = 2;
```

FINISHED ▶ ⌵ ⌵ ⚙

No Result [Last query execution info](#)

Took 0 sec. Last updated by anonymous at April 22 2025, 12:07:54 AM.

```
SELECT *
FROM shops.products;
```

FINISHED ▶ ⌵ ⌵ ⚙

📊

📈

📉

📊

📈

📉

👤

⌵

settings ⌵

product_id	category	name	price	stock	⌵
1	Electronics	Laptop	357000	15	
2	Electronics	Smartphone	109699	20	
4	Kitchen	Coffee cup	199	50	
3	Furniture	Desk Chair	12570	8	

Заключение

В теории удалось получить практические навыки работы с базой данных Cassandra, изучив основные операции по управлению данными, включая создание и использование ключспейсов, таблиц, выполнение запросов CQL, а также работу с различными инструментами подключения и администрирования.