

## Работа на лекции 2

**Цель работы:** 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 работы:

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

### Ход работы

Предполагаем, что платформа [devops\\_dba\\_25.ova](https://devops_dba_25.ova) [https://disk.yandex.ru/d/gagWU\\_zn1erR8g](https://disk.yandex.ru/d/gagWU_zn1erR8g) запущена и доступна.

### Работа в Cassandra

#### Подключение к среде Cassandra

##### Использование утилиты командной строки Cassandra

Вы можете найти утилиту командной строки `cqlsh` внутри контейнера Cassandra Docker, работающего как часть платформы. Подключитесь через SSH к Docker Host и выполните следующую команду `docker exec`

```
sudo docker exec -ti cassandra-1 cqlsh -u cassandra -p cassandra
```

Это подключит вас к контейнеру `cassandra-1` и запустит `cqlsh` внутри `id`. Вы должны увидеть вывод, похожий на этот.

```
bigdata@bigdata:~$ docker exec -ti cassandra-1 cqlsh-u cassandra -p cassandra
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.
cqlsh>
```

Теперь вы находитесь в командной строке Cassandra CQL и готовы выполнять операторы CQL.

Мы также можем увидеть актуальную версию Cassandra, CQL и `cqlsh`, доступную нам. Попробуйте выполнить следующий оператор CQL

```
SELECT * FROM system_schema.keyspaces;
```

и вы должны увидеть существующие в настоящее время пространства ключей в качестве результатов

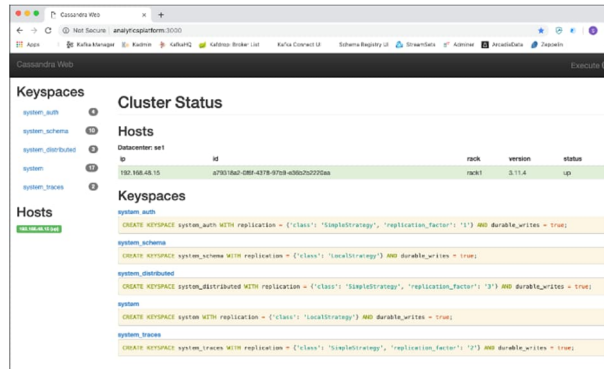
```
cqlsh> SELECT * FROM system_schema.keyspaces;
```

##### Использование браузерного графического интерфейса

Вместо работы через командную строку и, следовательно, необходимости подключаться к Docker Host через SSH, мы также можем использовать браузерный графический интерфейс для доступа к Cassandra. В качестве части платформы доступны две браузерные утилиты.

### Cassandra Web

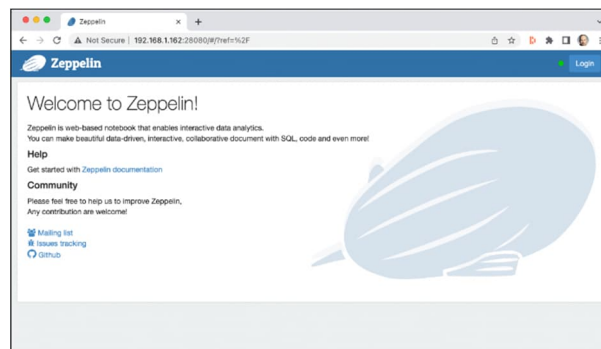
Первый — Cassandra Web. В окне браузера перейдите по адресу <http://localhost:28120/>, и вы сразу попадете на главный экран, как показано ниже.



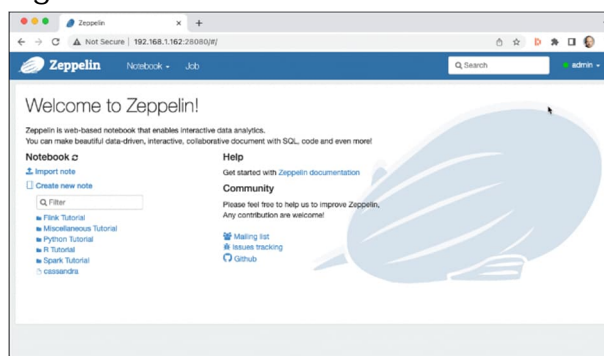
Если нажать кнопку Execute CQL в правом верхнем углу, появится всплывающее окно, в котором можно ввести операторы CQL и выполнить их.

### Apache Zeppelin

Другой универсальный инструмент «данных» — Apache Zeppelin. В окне браузера перейдите на <http://localhost:28080/>, и вы сразу же попадете на главный экран, как показано ниже.



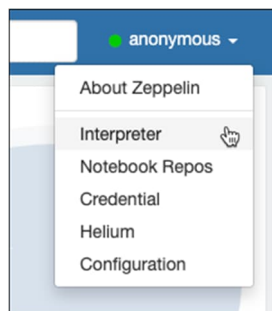
Нажмите Login в правом верхнем углу и введите admin в поле User Name и abc123! в поле Password. Нажмите Login.



Apache Zeppelin использует так называемую модель на основе «Notebook», которая позволяет выполнять интерактивную аналитику данных и совместную работу с документами на основе SQL, Scala и т. д.

Zeppelin использует концепцию интерпретаторов. Каждый интерпретатор может «общаться» с заданными системами данных. При создании Notebook вы можете указать интерпретатор «по умолчанию», все остальные интерпретаторы также могут использоваться, но тогда в каждой ячейке должна использоваться директива `%<interpreter-name>`.

Zeppelin подключил интерпретатор Cassandra, который мы будем здесь использовать. Но прежде, чем мы сможем его использовать, его нужно настроить. Щелкните раскрывающийся список anonymous или User Name вверху справа и выберите Interpreter, как показано на изображении ниже.



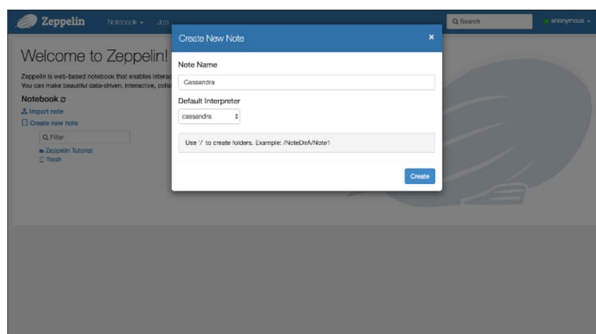
Выбрать Cassandra Interpreter, выполнив поиск или прокрутив страницу вниз, чтобы добраться до него. Нажмите edit и измените `cassandra.credentials.username` на `cassandra`, `cassandra.credentials.password` на `cassandra` и свойство `cassandra.host` на `cassandra-1` (имя службы нашего узла Cassandra в конфигурации docker-compose).

name	value	action
cassandra.cluster	Test Cluster	✕
cassandra.compression.protocol	NONE	✕
cassandra.credentials.password	---	✕
cassandra.credentials.username	---	✕
cassandra.hosts	cassandra-1	✕
cassandra.interpreter.parallelism	11	✕
cassandra.keyspace	system	✕
cassandra.load_balancing_policy	DEFAULT	✕
cassandra.max_schema_agreement_wait_second	10	✕
cassandra.native.port	9042	✕
cassandra.pooling.connection_per_host.local	2	✕
cassandra.pooling.connection_per_host.remote	1	✕
cassandra.pooling.hearbeat_interval.seconds	30	✕
cassandra.pooling.idle_timeout.seconds	100	✕
cassandra.pooling.max_connections_per_host.local	8	✕

Прокрутите вниз до конца настроек интерпретатора и нажмите Save. Подтвердите, чтобы Zeppelin перезапустил интерпретатор с новыми настройками.

Нажмите Zeppelin в верхнем левом углу, чтобы вернуться на главный экран.

Создадим новый блокнот, нажав на ссылку Create new note. Во всплывающем окне введите `Cassandra` в поле Note Name и выберите `cassandra` для Default Interpreter и нажмите Create.

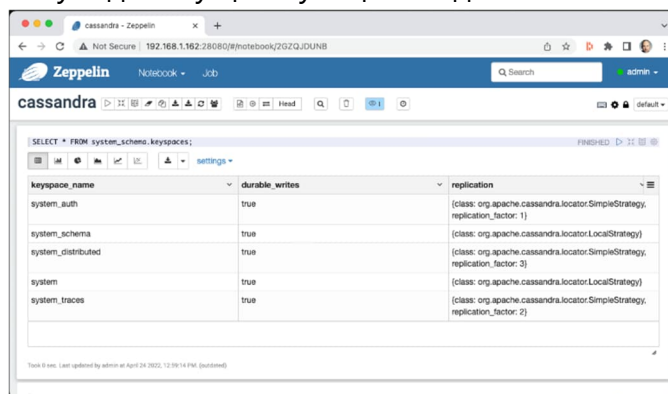


Появится пустой блокнот с одной ячейкой. Теперь эта ячейка готова к использованию и имеет назначенную интерпретацию Cassandra. Введите каждую команду в отдельную ячейку и либо щелкните значок play справа, либо нажмите Ctrl-Enter, чтобы выполнить ячейку. Новая ячейка автоматически появится при выполнении текущей.

Выполните скрипт

```
SELECT * FROM system_schema.keyspaces;
```

выполните скрипт, должны увидеть существующие на данный момент пространства ключей



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}

Для всех команд, которые следуют сейчас в этом семинаре, вы можете использовать один из 3 различных вариантов, показанных выше.

### Создание Keyspace для примера Movie

Keyspace в Cassandra является эквивалентом database/schema в реляционных базах данных. При создании keyspace необходимо указать настройки репликации:

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

Мы будем использовать SimpleStrategy для простоты, поскольку наша настройка Cassandra — это всего лишь один узел.

В производственной среде, где обычно имеется несколько центров обработки данных, обычно используется NetworkTopologyStrategy, поскольку она лучше распределяет данные по центрам обработки данных.

Коэффициент репликации = 1 означает, что на определенном узле будет одна копия строки. Более высокие коэффициенты репликации устанавливаются в реальных системах для создания нескольких реплик, что обеспечивает доступность данных в случае сбоев диска.

Чтобы иметь возможность работать с таблицами, вам необходимо использовать свое пространство ключей, как показано в следующем операторе:

```
USE movies;
```

Другой вариант — добавлять к имени таблицы префикс имени пространства ключей во всех запросах, аналогично тому, что вы можете сделать при ссылке на схему.

В любое время вы можете DESCRIBE пространство ключей, для этого используйте следующую команду:

```
DESCRIBE KEYSPACE movies;
```

получим

```
CREATE KEYSPACE movies WITH replication = {'class': 'SimpleStrategy',  
'replication_factor': '1'} AND durable_writes = true;
```

Если вы хотите перечислить все пространства ключей, имеющиеся в базе данных, резервное пространство ключей Cassandra с именем system будет вам полезно. Оно содержит множество системных таблиц для определения объектов базы данных и конфигурации кластера. Давайте перечислим все записи из таблицы `schema_keyspaces`, которая содержит записи для каждого пространства ключей.

Введите следующую команду:

```
SELECT * FROM system_schema.keyspaces;
```

### Использование "Static" таблиц (skinny row)

#### Создайте таблицы Movie и Actor

Выполните следующую инструкцию для создания таблицы с именем user. Для тех, кто знаком с SQL, следующий синтаксис должен показаться очень знакомым и почти идентичным. Даже некоторые соглашения об именовании, а также правила форматирования можно использовать повторно.

```
DROP TABLE IF EXISTS movies.movie;
CREATE TABLE movies.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)
);
```

Это создает первое статическое семейство столбцов (таблицу) в Cassandra. Теперь сделайте то же самое для таблицы Actor.

```
DROP TABLE IF EXISTS movies.actor;
CREATE TABLE movies.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)
);
```

Вы можете просмотреть метаданные таблицы с помощью команды `DESCRIBE`, как показано в следующем операторе.

```
DESCRIBE TABLE movie;
DESCRIBE TABLE actor;
```

Полная версия размещена на портале Github

<https://github.com/BosenkoTM/nosql-workshop/tree/main/03-working-with-cassandra>

## Варианты самостоятельной работы.

### Вариант 1.

1. Создайте ключспейс library с репликацией SimpleStrategy и коэффициентом репликации 1.
2. Создайте таблицу books в ключспейсе library с полями book\_id (int), title (text), author (text), year\_published (int), genre (text), isbn (text) и первичным ключом book\_id.
3. Вставьте три книги в таблицу books.
4. Выберите все книги из таблицы books.
5. Обновите год публикации книги с book\_id = 1 на 1962.

### Вариант 2.

1. Создайте ключспейс music с репликацией SimpleStrategy и коэффициентом репликации 1.
2. Создайте таблицу albums в ключспейсе music с полями album\_id (int), title (text), artist (text), year\_released (int), genre (text) и первичным ключом album\_id.
3. Вставьте три альбома в таблицу albums.
4. Выберите все альбомы из таблицы albums.
5. Удалите альбом с album\_id = 2.

### Вариант 3.

1. Создайте ключспейс movies с репликацией SimpleStrategy и коэффициентом репликации 1.
2. Создайте таблицу movies в ключспейсе movies с полями movie\_id (int), title (text), release\_year (int), running\_time (int), languages (set<text>), genres (set<text>), plot\_outline (text), cover\_url (text), top250\_rank (int) и первичным ключом movie\_id.
3. Вставьте два фильма в таблицу movies.
4. Выберите все фильмы из таблицы movies.
5. Обновите поле plot\_outline фильма с movie\_id = 0133093.

### Вариант 4.

1. Создайте ключспейс actors с репликацией SimpleStrategy и коэффициентом репликации 1.
2. Создайте таблицу actors в ключспейсе actors с полями actor\_id (int), name (text), headshot\_url (text), birth\_date (text), trade\_mark (list<text>) и первичным ключом actor\_id.
3. Вставьте три актера в таблицу actors.
4. Выберите всех актеров из таблицы actors.
5. Обновите поле trade\_mark актера с actor\_id = 0000246.

### Вариант 5.

1. Создайте ключспейс restaurants с репликацией SimpleStrategy и коэффициентом репликации 1.
2. Создайте таблицу restaurants в ключспейсе restaurants с полями restaurant\_id (int), name (text),
3. Вставьте три ресторана в таблицу restaurants.
4. Выберите все рестораны из таблицы restaurants.
5. Обновите поле rating ресторана с restaurant\_id = 1.

**Вариант 6.**

1. Создайте ключспейс schools с репликацией SimpleStrategy и коэффициентом репликации 1.
2. Создайте таблицу students в ключспейсе schools с полями student\_id (int), name (text), age (int), grade (text), email (text) и первичным ключом student\_id.
3. Вставьте четыре студента в таблицу students.
4. Выберите всех студентов из таблицы students.
5. Обновите поле email студента с student\_id = 2.

**Вариант 7.**

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 = 1.

**Вариант 8.**

1. Создайте ключспейс universities с репликацией SimpleStrategy и коэффициентом репликации 1.
2. Создайте таблицу professors в ключспейсе universities с полями professor\_id (int), name (text), department (text), email (text), office (text) и первичным ключом professor\_id.
3. Вставьте три профессора в таблицу professors.
4. Выберите всех профессоров из таблицы professors.
5. Обновите поле office профессора с professor\_id = 2.

**Вариант 9.**

1. Создайте ключспейс restaurants с репликацией SimpleStrategy и коэффициентом репликации 1.
2. Создайте таблицу restaurants в ключспейсе restaurants с полями restaurant\_id (int), name (text), address (text), cuisine (text), rating (float) и первичным ключом restaurant\_id.
3. Вставьте четыре ресторана в таблицу restaurants.
4. Выберите все рестораны из таблицы restaurants.
5. Обновите поле rating ресторана с restaurant\_id = 3.

**Вариант 10.**

1. Создайте ключспейс schools с репликацией SimpleStrategy и коэффициентом репликации 1.
2. Создайте таблицу students в ключспейсе schools с полями student\_id (int), name (text), age (int), grade (text), email (text) и первичным ключом student\_id.
3. Вставьте четыре студента в таблицу students.
4. Выберите всех студентов из таблицы students.
5. Обновите поле email студента с student\_id = 3.

**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.

**Вариант 12.**

1. Создайте ключспейс universities с репликацией SimpleStrategy и коэффициентом репликации 1.
2. Создайте таблицу professors в ключспейсе universities с полями professor\_id (int), name (text), department (text), email (text), office (text) и первичным ключом professor\_id.
3. Вставьте три профессора в таблицу professors.
4. Выберите всех профессоров из таблицы professors.
5. Обновите поле office профессора с professor\_id = 3.

**Вариант 13.**

1. Создайте ключспейс library с репликацией SimpleStrategy и коэффициентом репликации 1.
2. Создайте таблицу books в ключспейсе library с полями book\_id (int), title (text), author (text), year\_published (int), genre (text), isbn (text) и первичным ключом book\_id.
3. Вставьте четыре книги в таблицу books.
4. Выберите все книги из таблицы books.
5. Обновите поле year\_published книги с book\_id = 2 на 1950.

**Вариант 14.**

1. Создайте ключспейс music с репликацией SimpleStrategy и коэффициентом репликации 1.
2. Создайте таблицу albums в ключспейсе music с полями album\_id (int), title (text), artist (text), year\_released (int), genre (text) и первичным ключом album\_id.
3. Вставьте четыре альбома в таблицу albums.
4. Выберите все альбомы из таблицы albums.
5. Удалите альбом с album\_id = 3.

**Вариант 15.**

1. Создайте ключспейс movies с репликацией SimpleStrategy и коэффициентом репликации 1.
2. Создайте таблицу movies в ключспейсе movies с полями movie\_id (int), title (text), release\_year (int), running\_time (int), languages (set<text>), genres (set<text>), plot\_outline (text), cover\_url (text), top250\_rank (int) и первичным ключом movie\_id.
3. Вставьте три фильма в таблицу movies.
4. Выберите все фильмы из таблицы movies.
5. Обновите поле plot\_outline фильма с movie\_id = 0110912.



**Вариант 16.**

1. Создайте ключейс school с репликацией SimpleStrategy и коэффициентом репликации 1.
2. Создайте таблицу students в ключейсе school с полями student\_id (int), name (text), age (int), grade (text), subjects (set<text>) и первичным ключом student\_id.
3. Вставьте три записи студентов в таблицу students.
4. Выберите всех студентов, у которых возраст больше 15 лет.
5. Обновите поле grade для студента с student\_id = 1 на "A".

**Вариант 17.**

1. Создайте ключейс university с репликацией SimpleStrategy и коэффициентом репликации 1.
2. Создайте таблицу professors в ключейсе university с полями professor\_id (int), name (text), department (text), courses (list<text>), experience\_years (int) и первичным ключом professor\_id.
3. Вставьте два преподавателя в таблицу professors.
4. Выберите всех преподавателей из отдела "Computer Science".
5. Удалите преподавателя с professor\_id = 2.

**Вариант 18.**

1. Создайте ключейс ecommerce с репликацией SimpleStrategy и коэффициентом репликации 1.
2. Создайте таблицу products в ключейсе ecommerce с полями product\_id (int), name (text), price (decimal), category (text), stock (int) и первичным ключом product\_id.
3. Вставьте три товара в таблицу products.
4. Выберите все товары, цена которых меньше 100.
5. Обновите поле stock для товара с product\_id = 1, увеличив его на 50.

**Вариант 19.**

1. Создайте ключейс hospital с репликацией SimpleStrategy и коэффициентом репликации 1.
2. Создайте таблицу patients в ключейсе hospital с полями patient\_id (int), name (text), age (int), diagnosis (text), doctor (text) и первичным ключом patient\_id.
3. Вставьте два пациента в таблицу patients.
4. Выберите всех пациентов, чей диагноз содержит слово "вирус".
5. Обновите поле doctor для пациента с patient\_id = 1.

**Вариант 20.**

1. Создайте ключейс travel с репликацией SimpleStrategy и коэффициентом репликации 1.
2. Создайте таблицу flights в ключейсе travel с полями flight\_id (int), airline (text), destination (text), departure\_time (timestamp), price (decimal) и первичным ключом flight\_id.
3. Вставьте три рейса в таблицу flights.
4. Выберите все рейсы, направляющиеся в "Paris".
5. Удалите рейс с flight\_id = 2.

#### **Вариант 21.**

1. Создайте ключспейс sports с репликацией SimpleStrategy и коэффициентом репликации 1.
2. Создайте таблицу athletes в ключспейсе sports с полями athlete\_id (int), name (text), sport (text), country (text), achievements (list<text>) и первичным ключом athlete\_id.
3. Вставьте два спортсмена в таблицу athletes.
4. Выберите всех спортсменов из страны "USA".
5. Обновите поле achievements для спортсмена с athlete\_id = 1, добавив новое достижение.

#### **Вариант 22.**

1. Создайте ключспейс recipes с репликацией SimpleStrategy и коэффициентом репликации 1.
2. Создайте таблицу dishes в ключспейсе recipes с полями dish\_id (int), name (text), ingredients (set<text>), cooking\_time (int), difficulty (text) и первичным ключом dish\_id.
3. Вставьте три блюда в таблицу dishes.
4. Выберите все блюда, время приготовления которых менее 30 минут.
5. Удалите блюдо с dish\_id = 2.

#### **Вариант 23.**

1. Создайте ключспейс banking с репликацией SimpleStrategy и коэффициентом репликации 1.
2. Создайте таблицу accounts в ключспейсе banking с полями account\_id (int), owner\_name (text), balance (decimal), currency (text), status (text) и первичным ключом account\_id.
3. Вставьте два счета в таблицу accounts.
4. Выберите все счета с положительным балансом.
5. Обновите поле status для счета с account\_id = 1 на "inactive".

#### **Вариант 24.**

1. Создайте ключспейс social\_media с репликацией SimpleStrategy и коэффициентом репликации 1.
2. Создайте таблицу posts в ключспейсе social\_media с полями post\_id (int), user\_id (int), content (text), likes (int), comments (list<text>) и первичным ключом post\_id.
3. Вставьте три поста в таблицу posts.
4. Выберите все посты с количеством лайков более 10.
5. Удалите пост с post\_id = 2.

#### **Вариант 25.**

1. Создайте ключспейс cinema с репликацией SimpleStrategy и коэффициентом репликации 1.
2. Создайте таблицу screenings в ключспейсе cinema с полями screening\_id (int), movie\_title (text), show\_time (timestamp), hall (text), available\_seats (int) и первичным ключом screening\_id.
3. Вставьте два сеанса в таблицу screenings.
4. Выберите все сеансы, доступных мест в которых более 50.
5. Обновите поле available\_seats для сеанса с screening\_id = 1, уменьшив его на 10.

**Вариант 26.**

1. Создайте ключспейс fitness с репликацией SimpleStrategy и коэффициентом репликации 1.
2. Создайте таблицу workouts в ключспейсе fitness с полями workout\_id (int), name (text), duration (int), calories\_burned (int), exercises (set<text>) и первичным ключом workout\_id.
3. Вставьте три тренировки в таблицу workouts.
4. Выберите все тренировки, продолжительность которых более 60 минут.
5. Удалите тренировку с workout\_id = 2.

**Вариант 27.**

1. Создайте ключспейс education с репликацией SimpleStrategy и коэффициентом репликации 1.
2. Создайте таблицу courses в ключспейсе education с полями course\_id (int), title (text), instructor (text), duration\_weeks (int), enrolled\_students (int) и первичным ключом course\_id.
3. Вставьте два курса в таблицу courses.
4. Выберите все курсы, длительность которых менее 8 недель.
5. Обновите поле enrolled\_students для курса с course\_id = 1, увеличив его на 5.

**Вариант 28.**

1. Создайте ключспейс inventory с репликацией SimpleStrategy и коэффициентом репликации 1.
2. Создайте таблицу items в ключспейсе inventory с полями item\_id (int), name (text), quantity (int), location (text), last\_updated (timestamp) и первичным ключом item\_id.
3. Вставьте три предмета в таблицу items.
4. Выберите все предметы, количество которых менее 10.
5. Обновите поле quantity для предмета с item\_id = 1, увеличив его на 3.

**Вариант 29.**

1. Создайте ключспейс real\_estate с репликацией SimpleStrategy и коэффициентом репликации 1.
2. Создайте таблицу properties в ключспейсе real\_estate с полями property\_id (int), address (text), price (decimal), area (int), rooms (int) и первичным ключом property\_id.
3. Вставьте два объекта недвижимости в таблицу properties.
4. Выберите все объекты, площадь которых превышает 100 квадратных метров.
5. Удалите объект с property\_id = 2.

**Вариант 30.**

1. Создайте ключспейс transport с репликацией SimpleStrategy и коэффициентом репликации 1.
2. Создайте таблицу vehicles в ключспейсе transport с полями vehicle\_id (int), model (text), year (int), mileage (int), owner (text) и первичным ключом vehicle\_id.
3. Вставьте три транспортных средства в таблицу vehicles.
4. Выберите все автомобили, пробег которых менее 50000 км.
5. Обновите поле owner для автомобиля с vehicle\_id = 1.

## **Критерии оценки и требования к отчету по практической работе с Cassandra**

Что необходимо предоставить в отчете:

### **1. Титульный лист:**

- Название учебного заведения.
- Название работы.
- Номер варианта.
- ФИО студента.
- Группа.
- ФИО преподавателя.
- Год выполнения.

### **2. Содержание отчета:**

- Цель работы.
- Задачи варианта.
- Описание используемого программного обеспечения и версий.
- Листинг команд с пояснениями для каждой задачи.
- Скриншоты результатов выполнения команд.
- Описание возникших проблем и способов их решения.
- Выводы по работе.

### **3. Приложения (если есть):**

- Дополнительные скрипты.
- Исходные данные.
- Дополнительные материалы.

## **Критерии оценки**

Оценка "Отлично" (90-100 баллов):

- Все 5 заданий выполнены правильно и полностью.
- Использованы оптимальные решения.
- Код хорошо структурирован и документирован.
- Отчет оформлен аккуратно и содержит все необходимые элементы.
- Продемонстрировано глубокое понимание материала.
- Работа сдана в срок.

Оценка "Хорошо" (70-89 баллов):

- Выполнено 4 задания полностью.
- Решения рабочие, но не оптимальные.
- Код в целом структурирован.
- Отчет содержит незначительные недочеты.
- Продемонстрировано понимание основных концепций.
- Работа сдана в срок.

Оценка "Удовлетворительно" (50-69 баллов):

- Выполнено 3 задания.
- Решения содержат ошибки или неоптимальны.
- Код недостаточно структурирован.
- Отчет содержит существенные недочеты.

- Базовое понимание материала
- Работа сдана с небольшим опозданием

Оценка "Неудовлетворительно" (0-49 баллов):

- Выполнено менее 3 заданий.
- Решения содержат критические ошибки.
- Код не структурирован.
- Отчет оформлен небрежно или отсутствует.
- Непонимание базовых концепций.
- Значительное опоздание сдачи работы.

#### **Сроки сдачи**

- Стандартный срок: 1 неделя с момента получения задания.
- Крайний срок: 2 недели (с понижением максимальной оценки).
- После крайнего срока работы принимаются только с разрешения преподавателя.

#### **Защита работы**

Студент должен:

1. Продемонстрировать работоспособность решений.
2. Объяснить использованные подходы.
3. Ответить на вопросы по теоретической части.
4. Показать понимание возможных альтернативных решений.