Частина 1. Робота зі структурами даних у Cassandra:

```
PS C:\Users\dmytr\Desktop\5_1\pvs_labs\lab5> docker exec -it cassandra1 nodetool status

Datacenter: datacenter1
=======================
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
--  Address     Load       Tokens  Owns  Host ID                               Rack
UN  172.20.0.2  266.27 KiB  16     ?     572bffc9-09fd-43e0-98b2-73745b97d178  rack1

Note: Non-system keyspaces don't have the same replication settings, effective ownership information is meaningless
```

Створіть keyspace з найпростішої стратегією реплікації

```
PS C:\Users\dmytr\Desktop\5_1\pvs_labs\lab5> docker exec -it cassandra1 cqlsh
Connected to MyCluster at 127.0.0.1:9042
[cqlsh 6.2.0 | Cassandra 5.0.2 | CQL spec 3.4.7 | Native protocol v5]
Use HELP for help.
cqlsh> CREATE KEYSPACE shop_keyspace WITH replication = {'class': 'SimpleStrategy', 'replication_factor': 1};
cqlsh> USE shop_keyspace;
```

В цьому keyspace необхідно буде створити дві таблиці: *items*

```
cqlsh:shop_keyspace> CREATE TABLE items (
                ...     category TEXT,
                ...     id UUID,
                ...     price DECIMAL,
                ...     name TEXT,
                ...     manufacturer TEXT,
                ...     attributes MAP<TEXT, TEXT>,
                ...     PRIMARY KEY ((category), price, id, name, manufacturer)
                ... );
cqlsh:shop_keyspace>
```

Напишіть запит, який показує структуру створеної таблиці (команда *DESCRIBE*)

```
cqlsh:shop_keyspace> DESCRIBE TABLE items;

CREATE TABLE shop_keyspace.items (
    category text,
    price decimal,
    id uuid,
    name text,
    manufacturer text,
    attributes map<text, text>,
    PRIMARY KEY (category, price, id, name, manufacturer)
) WITH CLUSTERING ORDER BY (price ASC, id ASC, name ASC, manufacturer ASC)
    AND additional_write_policy = '99p'
    AND allow_auto_snapshot = true
    AND bloom_filter_fp_chance = 0.01
    AND caching = {'keys': 'ALL', 'rows_per_partition': 'NONE'}
    AND cdc = false
    AND comment = ''
    AND compaction = {'class': 'org.apache.cassandra.db.compaction.SizeTieredCompactionStrategy', 'max_threshold': '32', 'min_threshold': '4'}
    AND compression = {'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
    AND memtable = 'default'
    AND crc_check_chance = 1.0
    AND default_time_to_live = 0
    AND extensions = {}
    AND gc_grace_seconds = 864000
    AND incremental_backups = true
    AND max_index_interval = 2048
    AND memtable_flush_period_in_ms = 0
    AND min_index_interval = 128
    AND read_repair = 'BLOCKING'
    AND speculative_retry = '99p';
cqlsh:shop_keyspace>
```

Напишіть запит, який виводить усі товари в певній категорії відсортовані за ціною

```
cqlsh:shop_keyspace> SELECT * FROM items WHERE category = 'Laptop' ORDER BY price ASC;

 category | price   | id                                   | name           | manufacturer | attributes
----------+---------+--------------------------------------+----------------+--------------+-----------------------------------------------------------------------
   Laptop | 1199.99 | b304bcd2-34c4-44df-8600-b2d026076335 | MacBook Air 13" |        Apple |    {'RAM': '8GB', 'Screen': '13-inch', 'Storage': '256GB'}
   Laptop | 1399.99 | 97ed1d71-5ee5-4d3b-9322-fe8b43aa88f3 | HP Spectre x360 |           HP | {'RAM': '16GB', 'Screen': '13.5-inch', 'Storage': '512GB'}
   Laptop | 1599.99 | 2968561a-534b-4787-8bd5-b7a432284e3b |     Dell XPS 15 |         Dell |   {'RAM': '16GB', 'Screen': '15-inch', 'Storage': '512GB'}
   Laptop | 2499.99 | eb795779-061f-47cf-9f2d-c535b916d6ad | MacBook Pro 14" |        Apple |   {'RAM': '16GB', 'Screen': '14-inch', 'Storage': '512GB'}

(4 rows)
```

Напишіть запити, які вибирають товари за різними критеріями в межах певної категорії (тут де треба замість індексу використайте Matirialized view):

назва,

```
cqlsh:shop_keyspace> CREATE MATERIALIZED VIEW items_by_name AS
                ... SELECT * FROM items
                ... WHERE category IS NOT NULL AND name IS NOT NULL AND price IS NOT NULL AND id IS NOT NULL AND manufacturer IS NOT NULL
                ... PRIMARY KEY (category, name, manufacturer, price, id);

Warnings :
Materialized views are experimental and are not recommended for production use.

cqlsh:shop_keyspace> SELECT * FROM items_by_name WHERE category = 'Smartphone' AND name = 'iPhone 14';

 category   | name      | manufacturer | price  | id                                   | attributes
------------+-----------+--------------+--------+--------------------------------------+-----------------------------------------------------------
 Smartphone | iPhone 14 |        Apple | 999.99 | 3443a5ef-780d-4544-955c-2dc53395cee8 | {'Camera': '12MP', 'Color': 'Black', 'Storage': '128GB'}

(1 rows)
cqlsh:shop_keyspace>
```

ціна (в проміжку),

```
cqlsh:shop_keyspace> CREATE MATERIALIZED VIEW items_by_price AS
                ... SELECT * FROM items
                ... WHERE category IS NOT NULL AND price IS NOT NULL AND id IS NOT NULL AND manufacturer IS NOT NULL AND name IS NOT NULL
                ... PRIMARY KEY (category, price, manufacturer, name, id);

Warnings :
Materialized views are experimental and are not recommended for production use.

cqlsh:shop_keyspace> SELECT * FROM items_by_price WHERE category = 'Laptop' AND price >= 1000;

 category | price   | manufacturer | name           | id                                   | attributes
----------+---------+--------------+----------------+--------------------------------------+-----------------------------------------------------------------------
   Laptop | 1199.99 |        Apple | MacBook Air 13" | b304bcd2-34c4-44df-8600-b2d026076335 |    {'RAM': '8GB', 'Screen': '13-inch', 'Storage': '256GB'}
   Laptop | 1399.99 |           HP | HP Spectre x360 | 97ed1d71-5ee5-4d3b-9322-fe8b43aa88f3 | {'RAM': '16GB', 'Screen': '13.5-inch', 'Storage': '512GB'}
   Laptop | 1599.99 |         Dell |     Dell XPS 15 | 2968561a-534b-4787-8bd5-b7a432284e3b |   {'RAM': '16GB', 'Screen': '15-inch', 'Storage': '512GB'}
   Laptop | 2499.99 |        Apple | MacBook Pro 14" | eb795779-061f-47cf-9f2d-c535b916d6ad |   {'RAM': '16GB', 'Screen': '14-inch', 'Storage': '512GB'}

(4 rows)
cqlsh:shop_keyspace> SELECT * FROM items_by_price WHERE category = 'Laptop' AND price >= 1400;

 category | price   | manufacturer | name           | id                                   | attributes
----------+---------+--------------+----------------+--------------------------------------+-----------------------------------------------------------
   Laptop | 1599.99 |         Dell |     Dell XPS 15 | 2968561a-534b-4787-8bd5-b7a432284e3b | {'RAM': '16GB', 'Screen': '15-inch', 'Storage': '512GB'}
   Laptop | 2499.99 |        Apple | MacBook Pro 14" | eb795779-061f-47cf-9f2d-c535b916d6ad | {'RAM': '16GB', 'Screen': '14-inch', 'Storage': '512GB'}

(2 rows)
cqlsh:shop_keyspace>
```

ціна та виробник

```
cqlsh:shop_keyspace> CREATE MATERIALIZED VIEW items_by_manufacturer AS
                ... SELECT * FROM items
                ... WHERE category IS NOT NULL AND manufacturer IS NOT NULL AND price IS NOT NULL AND id IS NOT NULL AND name IS NOT NULL
                ... PRIMARY KEY (category, manufacturer, name, price, id);

Warnings :
Materialized views are experimental and are not recommended for production use.

cqlsh:shop_keyspace> SELECT * FROM items_by_manufacturer WHERE category = 'Smartwatch' AND manufacturer = 'Apple';

 category   | manufacturer | name               | price  | id                                   | attributes
------------+--------------+--------------------+--------+--------------------------------------+-----------------------------------------------------------------------
 Smartwatch |        Apple | Apple Watch Series 8 | 399.99 | 9314891e-3fc8-4410-9f4e-0021f1febd54 | {'Band': 'Sport Band', 'Connectivity': 'GPS', 'Size': '45mm'}

(1 rows)
cqlsh:shop_keyspace>
```

Створіть таблицю *orders* в якій міститься ім'я замовника і інформація про замовлення: перелік id-товарів у замовленні, вартість замовлення, дата замовлення

```
cqlsh:shop_keyspace> CREATE TABLE orders (
                 ...     customer_name TEXT,
                 ...     order_id UUID,
                 ...     item_ids LIST<UUID>,
                 ...     total_price DECIMAL,
                 ...     order_date TIMESTAMP,
                 ...     PRIMARY KEY (customer_name, order_date, order_id)
                 ... );
cqlsh:shop_keyspace> CREATE INDEX ON orders(order_id);
```

Напишіть запит, який показує структуру створеної таблиці

```
cqlsh:shop_keyspace>
cqlsh:shop_keyspace> DESCRIBE TABLE orders;

CREATE TABLE shop_keyspace.orders (
    customer_name text,
    order_date timestamp,
    order_id uuid,
    total_price decimal,
    item_ids list<uuid>,
    PRIMARY KEY (customer_name, order_date, order_id)
) WITH CLUSTERING ORDER BY (order_date ASC, order_id ASC)
    AND additional_write_policy = '99p'
    AND allow_auto_snapshot = true
    AND bloom_filter_fp_chance = 0.01
    AND caching = {'keys': 'ALL', 'rows_per_partition': 'NONE'}
    AND cdc = false
    AND comment = ''
    AND compaction = {'class': 'org.apache.cassandra.db.compaction.SizeTieredCompactionStrategy', 'max_threshold': '32', 'min_threshold': '4'}
    AND compression = {'chunk_length_in_kb': '16', 'class': 'org.apache.cassandra.io.compress.LZ4Compressor'}
    AND memtable = 'default'
    AND crc_check_chance = 1.0
    AND default_time_to_live = 0
    AND extensions = {}
    AND gc_grace_seconds = 864000
    AND incremental_backups = true
    AND max_index_interval = 2048
    AND memtable_flush_period_in_ms = 0
    AND min_index_interval = 128
    AND read_repair = 'BLOCKING'
    AND speculative_retry = '99p';

CREATE INDEX orders_order_id_idx ON shop_keyspace.orders (order_id);
cqlsh:shop_keyspace>
```

Для замовника виведіть всі його замовлення відсортовані за часом коли вони були зроблені

```
CREATE INDEX orders_order_id_idx ON shop_keyspace.orders (order_id);
cqlsh:shop_keyspace> SELECT * FROM orders WHERE customer_name = 'John Doe' ORDER BY order_date DESC;

 customer_name | order_date                      | order_id                             | item_ids                                                                   | total_price
---------------+---------------------------------+--------------------------------------+----------------------------------------------------------------------------+-------------
      John Doe | 2024-12-25 19:10:46.506000+0000 | cda54acb-0ae6-4e35-97ee-f182e267d6bb |                                     [11111111-1111-1111-1111-444444444444] |     1199.99
      John Doe | 2024-12-25 19:10:46.418000+0000 | 1db26677-d264-4b56-8d34-29e1183f146b | [11111111-1111-1111-1111-111111111111, 11111111-1111-1111-1111-222222222222] |     3499.98

(2 rows)
cqlsh:shop_keyspace>
```

Для кожного замовників визначте суму на яку були зроблені усі його замовлення

```
cqlsh:shop_keyspace> SELECT customer_name, SUM(total_price) AS total_spent FROM orders GROUP BY customer_name;

 customer_name | total_spent
---------------+-------------
    Jane Smith |      999.99
   Alice Brown |     2499.99
      John Doe |     4699.97

(3 rows)

Warnings :
Aggregation query used without partition key

cqlsh:shop_keyspace>
```

Для кожного замовлення виведіть час коли його ціна були занесена в базу (SELECT WRITETIME)

```
cqlsh:shop_keyspace> SELECT order_id, WRITETIME(total_price) FROM orders WHERE customer_name = 'John Doe';

 order_id                             | writetime(total_price)
--------------------------------------+------------------------
 1db26677-d264-4b56-8d34-29e1183f146b |       1735153846412784
 cda54acb-0ae6-4e35-97ee-f182e267d6bb |       1735153846505406

(2 rows)
cqlsh:shop_keyspace>
```

Частина 2. Налаштування реплікації у Cassandra

Сконфігурувати кластер з 3-х нод, перевірити правильність конфігурації

```
PS C:\Users\dmytr\Desktop\5_1\pvs_labs\lab5> docker exec -it cassandra1 nodetool status
Datacenter: dc1
===============
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
--  Address     Load        Tokens  Owns (effective)  Host ID                               Rack
UN  172.22.0.2  124.21 KiB  256     66.0%             db03f678-e06b-4254-b9cf-74afedcbb787  rack1
UN  172.22.0.3  89.61 KiB   256     60.0%             ef0b10d5-bcdd-45e7-9231-aad8c9011d97  rack1
UN  172.22.0.4  84.53 KiB   256     74.0%             60d7b35e-b83a-4c41-a769-9afd3c9f5bdf  rack1
```

Використовуючи *cqlsh*, створити три *Keyspace* з replication factor 1, 2, 3 з SimpleStrategy

```
Connected to MyCluster at 127.0.0.1:9042
[cqlsh 6.2.0 | Cassandra 5.0.2 | CQL spec 3.4.7 | Native protocol v5]
Use HELP for help.
cqlsh>
cqlsh> CREATE KEYSPACE keyspace_rf1
   ... WITH replication = {'class': 'SimpleStrategy', 'replication_factor': 1};
cqlsh>
cqlsh> CREATE KEYSPACE keyspace_rf2
   ... WITH replication = {'class': 'SimpleStrategy', 'replication_factor': 2};
cqlsh> CREATE KEYSPACE keyspace_rf3
   ... WITH replication = {'class': 'SimpleStrategy', 'replication_factor': 3};
cqlsh>
```

В кожному з кейспейсів створити прості таблиці

```
cqlsh> USE keyspace_rf1;
cqlsh:keyspace_rf1> CREATE TABLE example_table (
                ...     id UUID PRIMARY KEY,
                ...     value TEXT
                ... );
```

```
cqlsh:keyspace_rf1> USE keyspace_rf2;
cqlsh:keyspace_rf2> CREATE TABLE example_table (
                ...     id UUID PRIMARY KEY,
                ...     value TEXT
                ... );
cqlsh:keyspace_rf2>
```

```
cqlsh:keyspace_rf2> USE keyspace_rf3;
cqlsh:keyspace_rf3> CREATE TABLE example_table (
                ...     id UUID PRIMARY KEY,
                ...     value TEXT
                ... );
cqlsh:keyspace_rf3>
```

Вставте дані в створені таблиці і подивіться на їх розподіл по вузлах кластера для кожного з кейспесов (команда *nodetool status*)

```
cqlsh:keyspace_rf3> INSERT INTO keyspace_rf1.example_table (id, value) VALUES (uuid(), 'value_rf1');
cqlsh:keyspace_rf3> INSERT INTO keyspace_rf2.example_table (id, value) VALUES (uuid(), 'value_rf2');
cqlsh:keyspace_rf3> INSERT INTO keyspace_rf3.example_table (id, value) VALUES (uuid(), 'value_rf3');
cqlsh:keyspace_rf3>
```

```
PS C:\Users\dmytr\Desktop\5_1\pvs_labs\lab5> docker exec -it cassandra1 nodetool status
Datacenter: dc1
===============
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
--  Address     Load        Tokens  Owns  Host ID                               Rack
UN  172.22.0.2  125.5 KiB   256     ?     db03f678-e06b-4254-b9cf-74afedcbb787  rack1
UN  172.22.0.3  116.79 KiB  256     ?     ef0b10d5-bcdd-45e7-9231-aad8c9011d97  rack1
UN  172.22.0.4  85.4 KiB    256     ?     60d7b35e-b83a-4c41-a769-9afd3c9f5bdf  rack1

Note: Non-system keyspaces don't have the same replication settings, effective ownership information is meaningless
```

Для якогось запису з кожного з кейспейсу виведіть ноди на яких зберігаються дані

```
PS C:\Users\dmytr\Desktop\5_1\pvs_labs\lab5> docker exec -it cassandra1 nodetool getendpoints keyspace_rf1 example_table 331978e2-70aa-4e5b-a747-0ad203d0fb1d
172.22.0.4
PS C:\Users\dmytr\Desktop\5_1\pvs_labs\lab5>
```

```
PS C:\Users\dmytr\Desktop\5_1\pvs_labs> docker exec -it cassandra1 nodetool getendpoints keyspace_rf2 example_table 98e5f6a7-5a17-4c20-9903-07956d88b777
172.22.0.2
172.22.0.4
PS C:\Users\dmytr\Desktop\5_1\pvs_labs> docker exec -it cassandra1 nodetool getendpoints keyspace_rf3 example_table c122b06c-76c1-46f3-b46e-0207c7d49ced
172.22.0.2
172.22.0.3
172.22.0.4
```

Відключити одну з нод. Для кожного з кейспейсів перевірити з якими рівнями *consistency* можемо читати та писати

```
PS C:\Users\dmytr\Desktop\5_1\pvs_labs> docker stop cassandra3
cassandra3
PS C:\Users\dmytr\Desktop\5_1\pvs_labs>
```

```
PS C:\Users\dmytr\Desktop\5_1\pvs_labs\lab5> docker exec -it cassandra1 nodetool status
Datacenter: dc1
===============
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
--  Address     Load        Tokens  Owns  Host ID                               Rack
UN  172.22.0.2  1.18 MiB    256     ?     db03f678-e06b-4254-b9cf-74afedcbb787  rack1
UN  172.22.0.3  1.15 MiB    256     ?     ef0b10d5-bcdd-45e7-9231-aad8c9011d97  rack1
DN  172.22.0.4  181.09 KiB  256     ?     60d7b35e-b83a-4c41-a769-9afd3c9f5bdf  rack1
```

```
cqlsh:keyspace_rf3> CONSISTENCY ONE;
Consistency level set to ONE.
cqlsh:keyspace_rf3> USE keyspace_rf1;
cqlsh:keyspace_rf1> SELECT * FROM example_table;
NoHostAvailable: ('Unable to complete the operation against any hosts', {<Host: 127.0.0.1:9042 dc1>: Unavailable('Error from server: code=1000 [Unavailable exception] message="Cannot achieve
consistency level ONE" info={\'consistency\': \'ONE\', \'required_replicas\': 1, \'alive_replicas\': 0}')})
```

```
cqlsh:keyspace_rf1> USE keyspace_rf2;
cqlsh:keyspace_rf2> CONSISTENCY ONE;
Consistency level set to ONE.
cqlsh:keyspace_rf2> SELECT * FROM example_table;

 id                                   | value
--------------------------------------+-----------
 98e5f6a7-5a17-4c20-9903-07956d88b777 | value_rf2

(1 rows)
cqlsh:keyspace_rf2> CONSISTENCY TWO;
Consistency level set to TWO.
cqlsh:keyspace_rf2> SELECT * FROM example_table;
NoHostAvailable: ('Unable to complete the operation against any hosts', {<Host: 127.0.0.1:9042 dc1>: Unavailable('Error from server: code=1000 [Unavailable exception] message="Cannot achieve
consistency level TWO" info={\'consistency\': \'TWO\', \'required_replicas\': 2, \'alive_replicas\': 1}')})
```

```
cqlsh:keyspace_rf2> USE keyspace_rf3;
cqlsh:keyspace_rf3> CONSISTENCY ONE;
Consistency level set to ONE.
cqlsh:keyspace_rf3> SELECT * FROM example_table;

 id                                   | value
--------------------------------------+-----------
 c122b06c-76c1-46f3-b46e-0207c7d49ced | value_rf3

(1 rows)
cqlsh:keyspace_rf3> CONSISTENCY TWO;
Consistency level set to TWO.
cqlsh:keyspace_rf3> SELECT * FROM example_table;

 id                                   | value
--------------------------------------+-----------
 c122b06c-76c1-46f3-b46e-0207c7d49ced | value_rf3

(1 rows)
cqlsh:keyspace_rf3> CONSISTENCY THREE;
Consistency level set to THREE.
cqlsh:keyspace_rf3> SELECT * FROM example_table;
NoHostAvailable: ('Unable to complete the operation against any hosts', {<Host: 127.0.0.1:9042 dc1>: Unavailable('Error from server: code=1000 [Unavailable exception] message="Cannot achieve
consistency level THREE" info={\'consistency\': \'THREE\', \'required_replicas\': 3, \'alive_replicas\': 2}')})
cqlsh:keyspace_rf3>
```

Зробить так щоб три ноди працювали, але не бачили одна одну по мережі (заблокуйте чи відключити зв'язок між ними)

```
PS C:\Users\dmytr\Desktop\5_1\pvs_labs\lab5> docker network disconnect lab5_cassandra_network cassandra2
PS C:\Users\dmytr\Desktop\5_1\pvs_labs\lab5> docker network disconnect lab5_cassandra_network cassandra3
PS C:\Users\dmytr\Desktop\5_1\pvs_labs\lab5>
```

```
PS C:\Users\dmytr\Desktop\5_1\pvs_labs\lab5> docker network inspect lab5_cassandra_network
[
    {
        "Name": "lab5_cassandra_network",
        "Id": "6ae4f55554cad30c023c68cd249ef3d0808911e8cc377ceaa32751c9437909d6",
        "Created": "2024-12-25T16:10:43.659300317Z",
        "Scope": "local",
        "Driver": "bridge",
        "EnableIPv6": false,
        "IPAM": {
            "Driver": "default",
            "Options": null,
            "Config": [
                {
                    "Subnet": "172.22.0.0/16",
                    "Gateway": "172.22.0.1"
                }
            ]
        },
        "Internal": false,
        "Attachable": false,
        "Ingress": false,
        "ConfigFrom": {
            "Network": ""
        },
        "ConfigOnly": false,
        "Containers": {
            "fd3b6b0d7c457afaea1ff4c387f0919f1d1718acb29ae09b4ca33dfacddceff9": {
                "Name": "cassandra1",
                "EndpointID": "9f95ee55f29efb4d6236df39e7001945ec2be99032a8f96788849e3086850b4b",
                "MacAddress": "02:42:ac:16:00:02",
                "IPv4Address": "172.22.0.2/16",
                "IPv6Address": ""
            }
        },
        "Options": {},
        "Labels": {
            "com.docker.compose.network": "cassandra_network",
            "com.docker.compose.project": "lab5",
            "com.docker.compose.version": "2.23.0"
        }
    }
]
PS C:\Users\dmytr\Desktop\5_1\pvs_labs\lab5>
```

Для кейспейсу з *replication factor* 3 задайте рівень consistency рівним 1. Виконайте по черзі запис значення з однаковим primary key, але різними іншими значенням окремо на кожну з нод (тобто створіть конфлікт)

```
cqlsh:keyspace_rf3> CONSISTENCY ONE;
Consistency level set to ONE.
cqlsh:keyspace_rf3> USE keyspace_rf3;
cqlsh:keyspace_rf3> INSERT INTO example_table (id, value) VALUES (11111111-1111-1111-1111-111111111111, 'value_from_node1');
cqlsh:keyspace_rf3>
```

```
PS C:\Users\dmytr\Desktop\5_1\pvs_labs> docker exec -it cassandra2 cqlsh
Connected to MyCluster at 127.0.0.1:9042
[cqlsh 6.2.0 | Cassandra 5.0.2 | CQL spec 3.4.7 | Native protocol v5]
Use HELP for help.
cqlsh> CONSISTENCY ONE;
Consistency level set to ONE.
cqlsh> USE keyspace_rf3;
cqlsh:keyspace_rf3> INSERT INTO example_table (id, value) VALUES (11111111-1111-1111-1111-111111111111, 'value_from_node2');
```

```
PS C:\Users\dmytr\Desktop\5_1\pvs_labs> docker exec -it cassandra3 cqlsh
Connected to MyCluster at 127.0.0.1:9042
[cqlsh 6.2.0 | Cassandra 5.0.2 | CQL spec 3.4.7 | Native protocol v5]
Use HELP for help.
cqlsh> CONSISTENCY ONE;
Consistency level set to ONE.
cqlsh> USE keyspace_rf3;
cqlsh:keyspace_rf3> INSERT INTO example_table (id, value) VALUES (11111111-1111-1111-1111-111111111111, 'value_from_node3');
cqlsh:keyspace_rf3>
```

Відновіть зв'язок між нодами, і перевірте що вони знову об'єдналися у кластер. Визначте яким чином була вирішений конфлікт даних та яке значення було прийнято кластером та за яким принципом

```
PS C:\Users\dmytr\Desktop\5_1\pvs_labs\lab5> docker network connect lab5_cassandra_network cassandra2
PS C:\Users\dmytr\Desktop\5_1\pvs_labs\lab5> docker network connect lab5_cassandra_network cassandra3
PS C:\Users\dmytr\Desktop\5_1\pvs_labs\lab5>
```

```
cqlsh:keyspace_rf3> CONSISTENCY ONE;
Consistency level set to ONE.
cqlsh:keyspace_rf3> USE keyspace_rf3;
cqlsh:keyspace_rf3> SELECT * FROM example_table WHERE id=11111111-1111-1111-1111-111111111111;

 id                                   | value
--------------------------------------+-----------------
 11111111-1111-1111-1111-111111111111 | value_from_node1

(1 rows)
cqlsh:keyspace_rf3>
```