

Lab5 сербіненко олексій

Частина 1. Робота зі структурами даних у Cassandra

Встановив Cassandra

```
alekseyserbinenko@MacBook-Air-Aleksej lab5pvs % docker exec -it cassandra nodetool status
Datacenter: datacenter1
=====
Status=Up/Down
|/ State=Normal/Leaving/Joining/Moving
-- Address      Load          Tokens   Owns (effective)  Host ID                               Rack
UN 172.17.0.2    114.67 KiB    16       100.0%            53489072-7abc-415b-be0d-8b88fd21d7f4  rack1
```

Створить keyspace з найпростішої стратегією реплікації

```
[cqlsh> CREATE KEYSPACE lab5Shop_keyspace WITH replication = {'class': 'SimpleStrategy', 'replication_factor': 1};
[cqlsh> USE lab5Shop keyspace
```

Таблиця Items:

```
cqlsh:lab5shop_keyspace> CREATE TABLE items (
... category TEXT,
... id UUID,
... price DECIMAL,
... name TEXT,
... manufacturer TEXT,
... PRIMARY KEY((category), price, id, name, manufacturer),
... attributes MAP<TEXT, TEXT>
... );
cqlsh:lab5shop_keyspace>
```

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

```
cqlsh:lab5shop_keyspace> DESCRIBE TABLE items;

CREATE TABLE lab5shop_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';
```

Заповнив таблицю

```
cqlsh:lab5shop_keyspace> SELECT * FROM items;
```

category	price	id	name	manufacturer	attributes
t-shirts	19.99	6331569f-2f81-4323-b0c9-3157228d88a1	Uniqlo Airism	Uniqlo	{'color': 'Gray', 'material': 'Nylon', 'size': 'S'}
t-shirts	29.99	166891db-0e12-40a7-b1f0-c995640a2b50	Adidas Climalite	Adidas	{'color': 'White', 'material': 'Cotton', 'size': 'L'}
t-shirts	49.99	fb522c87-ac2a-4bbe-9e63-3d0b82593bf3	Nike Dri-FIT	Nike	{'color': 'Black', 'material': 'Polyester', 'size': 'M'}
jeans	79.99	45c60ba6-70dc-44c6-baf9-af7045befdf1	Wrangler Regular Fit	Wrangler	{'color': 'Black', 'fit': 'Regular', 'size': '34'}
jeans	89.99	0611df4f-e423-43cc-869e-4cb7845a7986	Levis 511	Levis	{'color': 'Blue', 'fit': 'Slim', 'size': '32'}
jeans	99.99	a0472472-2b04-48d6-befd-33cc86ee2b6e	Diesel Thommer	Diesel	{'color': 'Dark Blue', 'fit': 'Skinny', 'size': '30'}
sneakers	99.99	d4f8caf2-f38c-435d-81c5-fd60ead16dd0	Adidas Ultraboost 21	Adidas	{'color': 'Black/Gray', 'size': '43', 'type': 'Running'}
sneakers	119.99	cefbe452-ac27-4825-b9a7-b287cf76119c	Puma RS-X	Puma	{'color': 'Blue/Green', 'size': '41', 'type': 'Casual'}
sneakers	129.99	fc35921f-6b3b-4705-ba36-9c53167c95eb	Nike Air Max 270	Nike	{'color': 'White/Red', 'size': '42', 'type': 'Running'}

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

```
qqlsh:lab5shop_keyspace> SELECT * FROM items WHERE category = 't-shirts' ORDER BY price ASC;
```

category	price	id	name	manufacturer	attributes
t-shirts	19.99	6331569f-2f81-4323-b0c9-3157228d88a1	Uniqlo Airism	Uniqlo	{'color': 'Gray', 'material': 'Nylon', 'size': 'S'}
t-shirts	29.99	166891db-0e12-40a7-b1f0-c995640a2b50	Adidas Climalite	Adidas	{'color': 'White', 'material': 'Cotton', 'size': 'L'}
t-shirts	49.99	fb522c87-ac2a-4bbe-9e63-3d0b82593bf3	Nike Dri-FIT	Nike	{'color': 'Black', 'material': 'Polyester', 'size': 'M'}

(3 rows)

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

Назва:

```
qqlsh:lab5shop_keyspace> CREATE MATERIALIZED VIEW items_by_name AS
... SELECT category, id, price, name, manufacturer, attributes
... FROM items
... WHERE category IS NOT NULL AND name IS NOT NULL AND manufacturer IS NOT NULL AND price IS NOT NULL AND id IS NOT NULL
... PRIMARY KEY (category, name, manufacturer, price, id);

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

qqlsh:lab5shop_keyspace> SELECT * FROM items_by_name
... WHERE category = 't-shirts' AND name = 'Nike Dri-FIT';
```

category	name	manufacturer	price	id	attributes
t-shirts	Nike Dri-FIT	Nike	49.99	fb522c87-ac2a-4bbe-9e63-3d0b82593bf3	{'color': 'Black', 'material': 'Polyester', 'size': 'M'}

(1 rows)

Ціна(в проміжку):

```
qqlsh:lab5shop_keyspace> CREATE MATERIALIZED VIEW items_by_price AS
... SELECT category, id, price, name, manufacturer, attributes
... FROM items
... WHERE category IS NOT NULL AND price IS NOT NULL AND id IS NOT NULL AND name IS NOT NULL AND manufacturer IS NOT NULL
... PRIMARY KEY (category, price, id, name, manufacturer);

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

qqlsh:lab5shop_keyspace> SELECT * FROM items_by_price
... WHERE category = 't-shirts' AND price >= 20.00 AND price <= 50.00;
```

category	price	id	name	manufacturer	attributes
t-shirts	29.99	166891db-0e12-40a7-b1f0-c995640a2b50	Adidas Climalite	Adidas	{'color': 'White', 'material': 'Cotton', 'size': 'L'}
t-shirts	49.99	fb522c87-ac2a-4bbe-9e63-3d0b82593bf3	Nike Dri-FIT	Nike	{'color': 'Black', 'material': 'Polyester', 'size': 'M'}

(2 rows)

Ціна та виробник:

```
qqlsh:lab5shop_keyspace> CREATE MATERIALIZED VIEW items_by_price_and_manufacturer AS
... SELECT category, id, price, name, manufacturer, attributes
... FROM items
... WHERE category IS NOT NULL AND price IS NOT NULL AND manufacturer IS NOT NULL AND id IS NOT NULL AND name IS NOT NULL
... PRIMARY KEY (category, manufacturer, price, id, name);

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

qqlsh:lab5shop_keyspace> SELECT * FROM items_by_price_and_manufacturer
... WHERE category = 't-shirts' AND manufacturer = 'Nike' AND price >= 20.00 AND price <= 50.00;
```

category	manufacturer	price	id	name	attributes
t-shirts	Nike	49.99	fb522c87-ac2a-4bbe-9e63-3d0b82593bf3	Nike Dri-FIT	{'color': 'Black', 'material': 'Polyester', 'size': 'M'}

(1 rows)

Таблиця Orders:

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

Напишіть запит, який показує структуру створеної таблиці

```
cqlsh:lab5shop_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:lab5shop_keyspace> DESCRIBE TABLE orders;

CREATE TABLE lab5shop_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';
```

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

```
cqlsh:lab5shop_keyspace> SELECT * FROM orders
... WHERE customer_name = 'Michael Johnson'
... ORDER BY order_date ASC;
```

customer_name	order_date	order_id	item_ids	total_price
Michael Johnson	2024-01-01 14:00:00.000000+0000	f026a12f-a8d2-4173-9f2d-ea64ca24ad33	[b786cd84-1446-43e1-973f-9dd26fbed17b, 0524216d-5388-4973-ae02-b3523983d0b5]	129.99
Michael Johnson	2024-01-02 12:45:00.000000+0000	e8e94201-f2d0-4e99-94dd-378013d249fc	[7ef485a5-8dc2-45a6-81b5-29ac5137432d, cbfaa5b8-77af-4b8a-9cdc-e453fa808183, c36ae4c1-9f84-46c8-90e8-23b60e131ee8]	189.99

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

```
cqlsh:lab5shop_keyspace> SELECT customer_name, SUM(total_price) AS total_spent
... FROM orders
... GROUP BY customer_name;
```

customer_name	total_spent
Emily Davis	189.98
Chris Lee	139.98
Michael Johnson	319.98

(3 rows)

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

```
cqlsh:lab5shop_keyspace> SELECT customer_name, order_id, WRITETIME(total_price) AS write_time
... FROM orders;
```

customer_name	order_id	write_time
Emily Davis	67c49dfe-0ca8-4f2c-91f3-98359072be8d	1735476507484606
Emily Davis	142a97b3-0fc3-4287-841b-f7fee6589d47	1735476507943584
Chris Lee	3eb9ebe3-fbeb-4d55-90ef-4fe9b85117bd	1735476514511997
Chris Lee	83fe74d0-ae51-4bb2-be59-8de6558da1d4	1735476515059869
Michael Johnson	f026a12f-a8d2-4173-9f2d-ea64ca24ad33	1735476497887370
Michael Johnson	e8e94201-f2d0-4e99-94dd-378013d249fc	1735476499390313

Частина 2. Налаштування реплікації у Cassandra

Сконфігурувати кластер з 3-х нод, перевірити правильність конфігурації

```
alekseyserbinenko@MacBook-Air-Aleksej lab5pvs % docker exec -it cassandra-1 nodetool status
Datacenter: datacenter1
=====
Status=Up/Down
// State=Normal/Leaving/Joining/Moving
-- Address      Load          Tokens   Owns (effective)  Host ID                               Rack
UN 172.18.0.3    96.75 KiB     16      59.3%             bc4027ed-8328-46f6-8ff4-6770d294df82 rack1
UN 172.18.0.2    136.29 KiB    16      64.7%             2914d677-6f07-422e-9f41-e05d3d178b62 rack1
UN 172.18.0.4    80.06 KiB     16      76.0%             ba42a639-7ff5-412e-924e-b4232cf31225 rack1
```

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

```
alekseyserbinenko@MacBook-Air-Aleksej lab5pvs % docker exec -it cassandra-1 cqlsh
Connected to my-cluster 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 keyspace_rf1
... WITH replication = {'class': 'SimpleStrategy', 'replication_factor': 1};
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> DESCRIBE keyspaces;
```

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

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> █

alekseyserbinenko@MacBook-Air-Aleksej ~ % docker exec -it cassandra-1 nodetool status
Datacenter: datacenter1
=====
Status=Up/Down
// State=Normal/Leaving/Joining/Moving
-- Address      Load          Tokens   Owns  Host ID                               Rack
UN 172.18.0.3    151.19 KiB     16      ?     bc4027ed-8328-46f6-8ff4-6770d294df82 rack1
UN 172.18.0.2    154.25 KiB     16      ?     2914d677-6f07-422e-9f41-e05d3d178b62 rack1
UN 172.18.0.4    80.96 KiB      16      ?     ba42a639-7ff5-412e-924e-b4232cf31225 rack1

Note: Non-system keyspaces don't have the same replication settings, effective ownership information is meaningless
alekseyserbinenko@MacBook-Air-Aleksej ~ % █
```


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

```
alekseyserbinenko@MacBook-Air-Aleksej ~ % docker exec -it cassandra-1 nodetool getendpoints keyspace_rf1 example_table da3874de-6b85-48ef-8184-cf00770ae41a
172.18.0.3
alekseyserbinenko@MacBook-Air-Aleksej ~ % docker exec -it cassandra-1 nodetool getendpoints keyspace_rf2 example_table d3b631c0-788f-4793-91df-544b25765320
172.18.0.3
172.18.0.2
alekseyserbinenko@MacBook-Air-Aleksej ~ % docker exec -it cassandra-1 nodetool getendpoints keyspace_rf3 example_table 480541ba-a61d-43bd-963c-f57b6862f9bb
172.18.0.3
172.18.0.2
172.18.0.4
```

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

```
alekseyserbinenko@MacBook-Air-Aleksej ~ % docker stop cassandra-3
cassandra-3
alekseyserbinenko@MacBook-Air-Aleksej ~ % docker exec -it cassandra-1 nodetool status
Datacenter: datacenter1
=====
Status=Up/Down
-- State=Normal/Leaving/Joining/Moving
-- Address      Load       Tokens     Owns    Host ID                               Rack
UN 172.18.0.3    151.19 KiB  16        ?       bc4027ed-8328-46f6-8ff4-6770d294df82  rack1
UN 172.18.0.2    154.25 KiB  16        ?       2914d677-6f07-422e-9f41-e05d3d178b62  rack1
DN 172.18.0.4    80.96 KiB   16        ?       ba42a639-7ff5-412e-924e-b4232cf31225  rack1
```

```
(1 rows)
[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 datacenter1>: Unavailable('Error from server: 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
d3b631c0-788f-4793-91df-544b25765320	value_rf2

(1 rows)

```
[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
480541ba-a61d-43bd-963c-f57b6862f9bb	value_rf3

(1 rows)

```
[cqlsh:keyspace_rf3> CONSISTENCY TWO;
Consistency level set to TWO.
[cqlsh:keyspace_rf3> SELECT * FROM example_table;
```

id	value
480541ba-a61d-43bd-963c-f57b6862f9bb	value_rf3

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

```
alekseyserbinenko@MacBook-Air-Aleksej ~ % docker network disconnect lab5pvs_cassandra-net cassandra-2
alekseyserbinenko@MacBook-Air-Aleksej ~ % docker network disconnect lab5pvs_cassandra-net cassandra-3
alekseyserbinenko@MacBook-Air-Aleksej ~ % docker network inspect lab5pvs_cassandra-net
[
  {
    {
      "Name": "lab5pvs_cassandra-net",
      "Id": "93cd40d637074399946fbd985567c0496e7d98b6f0bf7e7560be75acc379f160",
      "Created": "2024-12-29T13:38:36.511962587Z",
      "Scope": "local",
      "Driver": "bridge",
      "EnableIPv6": false,
      "IPAM": {
        "Driver": "default",
        "Options": null,
        "Config": [
          {
            "Subnet": "172.18.0.0/16",
            "Gateway": "172.18.0.1"
          }
        ]
      },
      "Internal": false,
      "Attachable": false,
      "Ingress": false,
      "ConfigFrom": {
        "Network": ""
      },
      "ConfigOnly": false,
      "Containers": {
        "1b60b3a96ee88e9789f1e108e13b0dcf790bbfa149996a95ee446c2d9ade2305": {
          "Name": "cassandra-1",
          "EndpointID": "2d8446ebd9e38c1900a2eef4865f3df3a9a9e3706bf0509d148b65354610f985",
          "MacAddress": "02:42:ac:12:00:02",
          "IPv4Address": "172.18.0.2/16",
          "IPv6Address": ""
        }
      },
      "Options": {},
      "Labels": {
        "com.docker.compose.network": "cassandra-net",
        "com.docker.compose.project": "lab5pvs",
        "com.docker.compose.version": "2.30.3"
      }
    }
  ]
}
```

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> INSERT INTO example_table (id,value) VALUES (1111-1111-111-1111-11111, 'value_from_node1');
SyntaxException: line 1:49 no viable alternative at input '-1111' (...(id,value) VALUES (1111[-1111]...))
cqlsh:keyspace_rf3> INSERT INTO example_table (id,value) VALUES (1, 'value_from_node1');
InvalidRequest: Error from server: code=2200 [Invalid query] message="Invalid INTEGER constant (1) for "id" of type uuid"
cqlsh:keyspace_rf3> INSERT INTO example_table (id,value) VALUES (550e8400-e29b-41d4-a716-446655440000, 'value_from_node1');
cqlsh:keyspace_rf3>
```

```
cqlsh> exit
alekseyserbinenko@MacBook-Air-Aleksej lab5pvs % docker exec -it cassandra-2 cqlsh
Connected to my-cluster 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;
Improper USE command.
[cqlsh> DESCRIBE keyspaces;

keyspace_rf1 system system_schema system_virtual_schema
keyspace_rf2 system_auth system_traces
keyspace_rf3 system_distributed system_views

[cqlsh> USE keyspace_rf3;
[cqlsh:keyspace_rf3> INSERT INTO example_table (id,value) VALUES (550e8400-e29b-41d4-a716-446655440000, 'value_from_node2');
```

```
alekseyserbinenko@MacBook-Air-Aleksej lab5pvs % docker exec -it cassandra-3 cqlsh
Connected to my-cluster 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 (550e8400-e29b-41d4-a716-446655440000, 'value_from_node3');
```

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

```
cqlsh> USE keyspace_rf3;  
cqlsh:keyspace_rf3> SELECT * FROM example_table WHERE id=550e8400-e29b-41d4-a716-446655440000  
... ;
```

id	value
550e8400-e29b-41d4-a716-446655440000	value_from_node1