Сидоренко Катерина, ФБ-11мп

Task 4 - Робота з базовими функціями БД типу column family на прикладі Cassandra

Створіть keyspace з найпростішої стратегією реплікації

```
Connected to Test Cluster at 127.0.0.1:9042
[cqlsh 6.0.0 | Cassandra 4.0.1 | CQL spec 3.4.5 | Native protocol v5]
Use HELP for help.
cqlsh> CREATE KEYSPACE ks WITH replication = {'class': 'SimpleStrategy', 'replication_factor': 1 };
cqlsh>
cqlsh> USE ks;
cqlsh:ks> CREATE TABLE items ( model text, category text, producer text, price int, info map<text, text>, id int, PRIMARY KEY(category, price,
producer)) ;
cqlsh:ks>
cqlsh:ks>
cqlsh:ks> CREATE INDEX IF NOT EXISTS info_idx_entries ON items (ENTRIES(info));
cqlsh:ks> CREATE INDEX IF NOT EXISTS info_idx_keys ON items (KEYS(info));
cqlsh:ks>
cqlsh:ks> CREATE INDEX IF NOT EXISTS info_idx_model ON items (model);
```

```
cqlsh:ks> INSERT INTO items(model, category, producer, price, id)  VALUES ('Mi Band 3','Smartwatch', 'XiaoMi', 20, 1);
cqlsh:ks> INSERT INTO items(model, category, producer, price, id)  VALUES ('Body Scale 2','Smartweight', 'XiaoMi', 200, 2) ;
cqlsh:ks> INSERT INTO items(model, category, producer, price, id)  VALUES ('Uehk737 Light','Laptop', 'Samsung', 1000, 3) ;
cqlsh:ks>
cqlsh:ks> INSERT INTO items(model, category, producer, price, id, info) VALUES ('Asus VivoBook','Laptop', 'Asus', 2000, 4, {'description' :'Jus
t a normal Laptop'});
cqlsh:ks> INSERT INTO items(model, category, producer, price, id, info) VALUES ('Galaxy A52','Phone', 'Samsung', 600, 5, {'cameras':'3'});
cqlsh:ks> INSERT INTO items(model, category, producer, price, id, info) VALUES ('iPhone 6', 'Phone', 'Apple', 200, 6,  {'cameras':'2'}) ;
cqlsh:ks> INSERT INTO items(model, category, producer, price, id, info) VALUES ('iPhone 11', 'Phone', 'Apple', 600, 7,  {'cameras':'2', 'descri
ption' :'WOW'}) ;
```

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

 category   | price | producer | id | info                                                | model
------------+-------+----------+----+-----------------------------------------------------+---------------
       Phone |   200 |    Apple |  6 |                                {'cameras': '2'} |      iPhone 6
       Phone |   600 |    Apple |  7 |  {'cameras': '2', 'description': 'WOW'} |     iPhone 11
       Phone |   600 |  Samsung |  5 |                                {'cameras': '3'} |    Galaxy A52
 Smartweight |   200 |   XiaoMi |  2 |                                           null |  Body Scale 2
      Laptop |  1000 |  Samsung |  3 |                                           null | Uehk737 Light
      Laptop |  2000 |     Asus |  4 | {'description': 'Just a normal Laptop'} | Asus VivoBook
  Smartwatch |    20 |   XiaoMi |  1 |                                           null |     Mi Band 3
```

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

```
cqlsh:ks> DESCRIBE items;

CREATE TABLE ks.items (
    category text,
    price int,
    producer text,
    id int,
    model text,
    info map<text, text>,
    PRIMARY KEY (category, price, producer)
) WITH CLUSTERING ORDER BY (price ASC, producer ASC)
    AND additional_write_policy = '99p'
    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 crc_check_chance = 1.0
    AND default_time_to_live = 0
    AND extensions = {}
    AND gc_grace_seconds = 864000
    AND max_index_interval = 2048
    AND memtable_flush_period_in_ms = 0
    AND min_index_interval = 128
    AND read_repair = 'BLOCKING'
    AND speculative_retry = '99p';

CREATE INDEX info_idx_entries ON ks.items (entries(info));

CREATE INDEX info_idx_keys ON ks.items (keys(info));

CREATE INDEX info_idx_model ON ks.items (model);
```

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

```
cqlsh:ks> SELECT * FROM items where category = 'Phone' ORDER BY price;

 category | price | producer | id | info                                                | model
----------+-------+----------+----+-----------------------------------------------------+-----------
    Phone |   200 |    Apple |  6 |                                {'cameras': '2'} |   iPhone 6
    Phone |   600 |    Apple |  7 | {'cameras': '2', 'description': 'WOW'} |  iPhone 11
    Phone |   600 |  Samsung |  5 |                                {'cameras': '3'} | Galaxy A52

(3 rows)
```

3) Напишіть запити, які вибирають товари за різними критеріями в межах певної категорії:
    - назва,

```
cqlsh:ks> SELECT * FROM items where category = 'Phone' AND model = 'iPhone 6';

 category | price | producer | id | info             | model
----------+-------+----------+----+------------------+----------
    Phone |   200 |    Apple |  6 | {'cameras': '2'} | iPhone 6

(1 rows)
```

    - ціна (в проміжку),

```
cqlsh:ks> SELECT * FROM items where category = 'Phone' AND price > 200 AND price <= 700;

 category | price | producer | id | info                                                | model
----------+-------+----------+----+-----------------------------------------------------+-----------
    Phone |   600 |    Apple |  7 | {'cameras': '2', 'description': 'WOW'} |  iPhone 11
    Phone |   600 |  Samsung |  5 |                                {'cameras': '3'} | Galaxy A52

(2 rows)
```

    - ціна та виробник

```
cqlsh:ks> SELECT * FROM items where category = 'Phone' AND price = 600 AND producer = 'Apple';

 category | price | producer | id | info                                                | model
----------+-------+----------+----+-----------------------------------------------------+----------
    Phone |   600 |    Apple |  7 | {'cameras': '2', 'description': 'WOW'} | iPhone 11

(1 rows)
```

4) Напишіть запити, які вибирають товари за:
    - наявність певних характеристик

```
cqlsh:ks> SELECT * FROM items WHERE info CONTAINS KEY 'cameras';

 category | price | producer | id | info                                                | model
----------+-------+----------+----+-----------------------------------------------------+------------
    Phone |   200 |    Apple |  6 |                                {'cameras': '2'} |   iPhone 6
    Phone |   600 |    Apple |  7 | {'cameras': '2', 'description': 'WOW'} |  iPhone 11
    Phone |   600 |  Samsung |  5 |                                {'cameras': '3'} | Galaxy A52

(3 rows)
```

- певна характеристика та її значення

```
cqlsh:ks> SELECT * FROM items WHERE info['cameras'] = '2';

 category | price | producer | id | info                                                | model
----------+-------+----------+----+-----------------------------------------------------+-----------
    Phone |   200 |    Apple |  6 |                                {'cameras': '2'} |  iPhone 6
    Phone |   600 |    Apple |  7 | {'cameras': '2', 'description': 'WOW'} | iPhone 11

(2 rows)
```

5) Оновити опис товару:
   - змінить існуючі значення певної характеристики

```
cqlsh:ks> UPDATE items SET info['cameras' ] = '25' WHERE category = 'Phone' AND price = 600 AND producer = 'Apple';
cqlsh:ks> SELECT * FROM items where model = 'iPhone 11'
      ... ;

 category | price | producer | id | info                                                | model
----------+-------+----------+----+-----------------------------------------------------+----------
    Phone |   600 |    Apple |  7 | {'cameras': '25', 'description': 'WOW'} | iPhone 11

(1 rows)
```

   - додайте нові властивості (характеристики) товару

```
cqlsh:ks> UPDATE items SET info = info+{'review':'10/10'} WHERE category = 'Phone' AND price = 600 AND producer = 'Apple';
cqlsh:ks> SELECT * FROM items where model = 'iPhone 11';

 category | price | producer | id | info                                                                                | model
----------+-------+----------+----+-------------------------------------------------------------------------------------+----------
    Phone |   600 |    Apple |  7 | {'cameras': '25', 'description': 'WOW', 'review': '10/10'} | iPhone 11

(1 rows)
```

   - видалить характеристику товару

```
cqlsh:ks> DELETE info['description'] FROM items where category = 'Laptop' AND price =2000 AND producer = 'Asus';
cqlsh:ks> SELECT * FROM items where model = 'Asus VivoBook'
      ... ;

 category | price | producer | id | info | model
----------+-------+----------+----+------+---------------
   Laptop |  2000 |     Asus |  4 | null | Asus VivoBook

(1 rows)
```

Створіть таблицю *orders*

```
cqlsh:ks> CREATE TABLE orders (name text, items_id list<int>, sum int, date date, id int, PRIMARY KEY(name, date, id)) ;
cqlsh:ks> CREATE INDEX IF NOT EXISTS items_id_idx ON orders(items_id);
cqlsh:ks>
cqlsh:ks>
cqlsh:ks> INSERT INTO orders(name, items_id, sum, date, id) VALUES ('Alpha', [1], 900, '2021-03-15', 1) ;
cqlsh:ks> INSERT INTO orders(name, items_id, sum, date, id) VALUES ('Alpha', [1, 2], 300, '2021-02-15', 2) ;
cqlsh:ks> INSERT INTO orders(name, items_id, sum, date, id) VALUES ('Alpha', [1, 3], 100, '2021-04-15', 3) ;
cqlsh:ks> INSERT INTO orders(name, items_id, sum, date, id) VALUES ('Alpha', [1, 4], 400, '2021-04-15', 4) ;
cqlsh:ks> INSERT INTO orders(name, items_id, sum, date, id) VALUES ('Beta', [1, 4], 200, '2021-05-15', 5) ;
cqlsh:ks> INSERT INTO orders(name, items_id, sum, date, id) VALUES ('Gamma', [1, 7], 300, '2021-03-15', 6) ;
```

```
cqlsh:ks> SELECT * FROM orders;

 name  | date       | id | items_id | sum
-------+------------+----+----------+-----
 Gamma | 2021-03-15 |  6 |   [1, 7] | 300
  Beta | 2021-05-15 |  5 |   [1, 4] | 200
 Alpha | 2021-02-15 |  2 |   [1, 2] | 300
 Alpha | 2021-03-15 |  1 |      [1] | 900
 Alpha | 2021-04-15 |  3 |   [1, 3] | 100
 Alpha | 2021-04-15 |  4 |   [1, 4] | 400

(6 rows)
```

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

```
cqlsh:ks> DESCRIBE orders;

CREATE TABLE ks.orders (
    name text,
    date date,
    id int,
    sum int,
    items_id list<int>,
    PRIMARY KEY (name, date, id)
) WITH CLUSTERING ORDER BY (date ASC, id ASC)
    AND additional_write_policy = '99p'
    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 crc_check_chance = 1.0
    AND default_time_to_live = 0
    AND extensions = {}
    AND gc_grace_seconds = 864000
    AND max_index_interval = 2048
    AND memtable_flush_period_in_ms = 0
    AND min_index_interval = 128
    AND read_repair = 'BLOCKING'
    AND speculative_retry = '99p';

CREATE INDEX items_id_idx ON ks.orders (values(items_id));
```

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

```
cqlsh:ks> SELECT  * FROM orders WHERE name='Alpha' ORDER BY date;

 name  | date       | id | items_id | sum
-------+------------+----+----------+-----
 Alpha | 2021-02-15 |  2 |   [1, 2] | 300
 Alpha | 2021-03-15 |  1 |      [1] | 900
 Alpha | 2021-04-15 |  3 |   [1, 3] | 100
 Alpha | 2021-04-15 |  4 |   [1, 4] | 400
```

3) Для замовника знайдіть замовлення з певним товаром

```
cqlsh:ks> SELECT  * FROM orders WHERE name='Alpha' AND items_id CONTAINS 4;

 name  | date       | id | items_id | sum
-------+------------+----+----------+-----
 Alpha | 2021-04-15 |  4 |   [1, 4] | 400

(1 rows)
```

4) Для замовника знайдіть замовлення за певний період часу і їх кількість

```
cqlsh:ks> SELECT  COUNT(*) FROM orders WHERE name='Alpha' AND date > '2021-02-15' ;

 count
-------
     3

(1 rows)
```

5) Для кожного замовників визначте середню вартість замовлення

```
cqlsh:ks> SELECT name, AVG(sum) FROM orders GROUP BY name;

 name  | system.avg(sum)
-------+-----------------
 Gamma |             300
  Beta |             200
 Alpha |             425

(3 rows)
```

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

```
cqlsh:ks> SELECT name, SUM(sum) FROM orders GROUP BY name;

 name  | system.sum(sum)
-------+-----------------
 Gamma |             300
  Beta |             200
 Alpha |            1700

(3 rows)
```

7) Для кожного замовників визначте замовлення з максимальною вартістю

```
cqlsh:ks> SELECT name, MAX(sum) FROM orders GROUP BY name;

 name  | system.max(sum)
-------+-----------------
 Gamma |             300
  Beta |             200
 Alpha |             900

(3 rows)
```

8) Модифікуйте певне замовлення додавши / видаливши один або кілька товарів при цьому також змінюючи вартість замовлення

```
cqlsh:ks> UPDATE orders SET items_id = items_id + [3], sum = 1400 WHERE name = 'Alpha' AND date = '2021-03-15' AND id = 1;
cqlsh:ks> SELECT * FROM orders WHERE name = 'Alpha';

 name  | date       | id | items_id | sum
-------+------------+----+----------+------
 Alpha | 2021-02-15 |  2 |   [1, 2] |  300
 Alpha | 2021-03-15 |  1 |   [1, 3] | 1400
 Alpha | 2021-04-15 |  3 |   [1, 3] |  100
 Alpha | 2021-04-15 |  4 |   [1, 4] |  400

(4 rows)
```

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

```
cqlsh:ks> SELECT WRITETIME(sum), id FROM orders;

 writetime(sum)   | id
------------------+----
 1639693071873917 |  6
 1639693070869053 |  5
 1639693070845717 |  2
 1639693480731612 |  1
 1639693070854787 |  3
 1639693070861902 |  4

(6 rows)
```

10)Створіть замовлення з певним часом життя (TTL), після якого воно видалиться

```
cqlsh:ks> INSERT INTO orders(name, items_id, sum, date, id) VALUES ('Zeta', [], 300, '2021-03-15', 7) USING TTL 30;
cqlsh:ks> SELECT * FROM orders WHERE name = 'Zeta';

 name | date       | id | items_id | sum
------+------------+----+----------+-----
 Zeta | 2021-03-15 |  7 |     null | 300

(1 rows)
cqlsh:ks> SELECT * FROM orders WHERE name = 'Zeta';

 name | date | id | items_id | sum
------+------+----+----------+-----

(0 rows)
```

11)Поверніть замовлення у форматі JSON

```
cqlsh:ks> SELECT json * FROM orders LIMIT 1;

 [json]
------------------------------------------------------------------------------------------
 {"name": "Gamma", "date": "2021-03-15", "id": 6, "items_id": [1, 7], "sum": 300}

(1 rows)
```

12)Додайте замовлення у форматі JSON

```
cqlsh:ks> INSERT INTO orders JSON '{"name": "Yota", "date": "2021-03-16", "id": "22", "items_id": [1], "sum": 1111100}' ;
cqlsh:ks> SELECT * FROM oders WHERE name = 'Yota';
InvalidRequest: Error from server: code=2200 [Invalid query] message="table oders does not exist"
cqlsh:ks> SELECT * FROM orders WHERE name = 'Yota';

 name | date       | id | items_id | sum
------+------------+----+----------+---------
 Yota | 2021-03-16 | 22 |      [1] | 1111100

(1 rows)
```