Task 5 - Робота з базовими функціями БД типу column family на прикладі Cassandra

Виконав: Потужний Богдан

Установка:

- Встановіть Cassandra локально або на Docker / віртуальну машину (для наступного завдання необхідно буде створювати кластер з декількох екземплярів)
- Docker Cassandra

Документація

https://docs.datastax.com/en/dse/6.8/cgl/index.html

Завдання:

Ознайомтеся з особливістю моделювання даних у Cassandra:

- https://cassandra.apache.org/doc/latest/cassandra/data_modeling/index.html
- https://www.datastax.com/dev/blog/the-most-important-thing-to-know-in-cassandra-data-modeling-the-primary-key
- https://www.instaclustr.com/blog/cassandra-data-modeling/

Створіть keyspace з найпростішої стратегією реплікації https://docs.datastax.com/en/dse/6.8/cql/cql/cql/cql/using/useCreateKeyspace.html

```
CREATE KEYSPACE online_store WITH REPLICATION = { 'class' : 'SimpleStrategy',
'replication_factor' : 1 };
USE online store;
```

Аналогічно завданням по MongoDB Вам необхідно змоделювати інтернет-магазин який містить різноманітні товара (тобто у яких різний набір властивостей) - таблиця *items*. **Необхідно, щоб пошук швидко працював для визначеної категорії товарів**

Для набору властивостей товару виберіть базові характеристики однакові для всіх товарів (назва, ціна, виробник, ...), а для властивостей які відрізняються використовуйте тип *map* (з індексом для можливості пошуку по її вмісту) https://docs.datastax.com/en/dse/6.8/cql/cql/cql/cql/using/useIndexColl.html

```
CREATE TABLE items (
    category TEXT,
    price DECIMAL,
    product id UUID,
    name TEXT,
    producer TEXT,
    properties MAP<TEXT, TEXT>,
    PRIMARY KEY (category, price, product id)
) WITH CLUSTERING ORDER BY (price ASC);
```

!!! У запитах заборонено використовувати ALLOW FILTERING !!!

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



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



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





ціна та виробник

```
CREATE MATERIALIZED VIEW items_by_producer AS
   SELECT * FROM items
   WHERE category IS NOT NULL AND producer IS NOT NULL AND price IS NOT NULL AND product_id IS NOT NULL
   PRIMARY KEY (category, producer, price, product_id);
SELECT * FROM items_by_producer WHERE category = 'Phone' AND producer = 'Apple' AND price <= 1000;
   1 Phone
                                 100 4377c0c6-b468-4e7b-9ff9-54e905014955 iPhone 6 {'color': 'Silver',
```

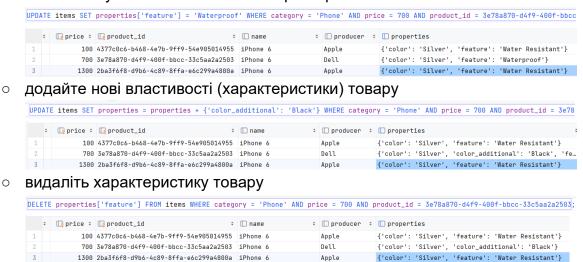
- 4) Напишіть запити, які вибирають товари за:
 - наявність певних характеристик



о певна характеристика та її значення



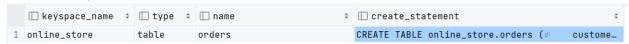
- 5) Оновити опис товару:
 - змінить існуючі значення певної характеристики



Створіть таблицю *orders* в якій міститься ім'я замовника і інформація про замовлення: перелік іd-товарів у замовленні, вартість замовлення, дата замовлення, **Для кожного замовника повинна бути можливість швидко шукати його замовлення і виконувати по них запити.**

```
CREATE TABLE orders (
    customer_name TEXT,
    order_date TIMESTAMP,
    order_id UUID,
    product_ids LIST<UUID>,
    order_value DECIMAL,
    PRIMARY KEY (customer_name, order_date, order_id)
) WITH CLUSTERING ORDER BY (order_date DESC);
```

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



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



3) Для замовника знайдіть замовлення з певним товаром

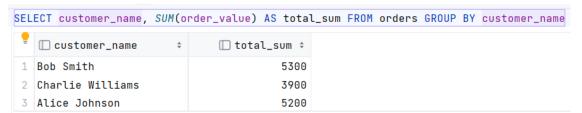
```
CREATE INDEX idx_orders_product_ids ON orders (product_ids);

SELECT * FROM orders WHERE customer_name = 'Alice Johnson' AND product_ids CONTAINS 4377c0c6-b468-4e7b-9ff9-54e905014955;
```

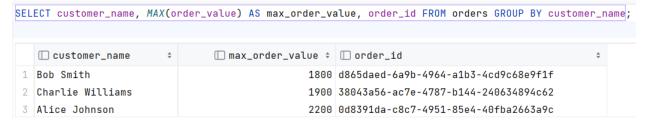
4) Для замовника знайдіть замовлення за певний період часу і їх кількість



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



6) Для кожного з замовників визначте замовлення з максимальною вартістю



7) Модифікуйте певне замовлення додавши / видаливши один або кілька товарів при цьому також змінюючи вартість замовлення "customer_name": "Bob Smith", "order_date": "2023-09-14 16:55:11.225", "order id": "d865daed-6a9b-4964-a1b3-4cd9c68e9f1f", "order value": 900, "product_ids": ["9382b9f0-7f32-4aa9-9076-95f1d220a2b4", "0c380d42-c917-49dc-8821-1bf660fcc825" UPDATE orders SET product ids = product ids + [844a439a-1c14-47fc-9d5c-4fdf427b9797], order value = 1000 WHERE customer name = 'Bob Smith' AND order date = '2023-09-14 16:55:11.225' AND order id = d865daed-6a9b-4964a1b3-4cd9c68e9f1f; "customer name": "Bob Smith", "order date": "2023-09-14 16:55:11.225", "order id": "d865daed-6a9b-4964-a1b3-4cd9c68e9f1f", "order_value": 1000, "product_ids": ["9382b9f0-7f32-4aa9-9076-95f1d220a2b4", "0c380d42-c917-49dc-8821-1bf660fcc825", "844a439a-1c14-47fc-9d5c-4fdf427b9797"

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

```
SELECT WRITETIME(order_value), order_id FROM orders
     1
               1694713858931000 d865daed-6a9b-4964-a1b3-4cd9c68e9f1f
 2
               1694710510866000 13e54c37-3486-4376-8c36-27bba9601178
 3
               1694710510772000 761c8a17-7313-49e8-9008-ebbdbbd5cc66
 4
               1694710510735000 9368b631-7d76-4597-8ff5-55a28bdcf15c
 5
               1694710511162000 38043a56-ac7e-4787-b144-240634894c62
               1694710511094000 483f34ba-7d9e-4e5f-bd80-89bdde9c8f4c
 6
 7
               1694710510965000 9c949299-8731-46f7-96f6-d67ee3e73da0
 8
               1694710511127000 0d8391da-c8c7-4951-85e4-40fba2663a9c
```

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

```
INSERT INTO orders (customer_name, order_date, order_id, product_ids, order_value)
VALUES ('Alice Johnson', '2023-09-15', uvid(), [844a439a-1c14-47fc-9d5c-4fdf427b9797], 800)
USING TTL 3600;
```

10) Поверніть замовлення у форматі JSON

11)Додайте замовлення у форматі JSON

```
INSERT INTO orders

JSON '{"customer_name": "Alice Johnson", "order_date": "2023-09-15", "order_id": "bf92f847-aac2-4130-ac34-a71a3819bc8e", "product_ids":
```

Вимогу до оформлення протоколу:

Завдання здається особисто без протоколу, або надсилається протокол який має містити:

команди та результати їх виконання