

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/cql/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_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 товари (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_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 !!!

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

```
DESCRIBE TABLE items;
```

	keyspace_name	type	name	create_statement
1	online_store	table	items	CREATE TABLE online_store.items (category...

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

```
SELECT * FROM items WHERE category = 'Phone';
```

	category	price	product_id	name	producer	properties
1	Phone	100	4377c0c6-b468-4e7b-9ff9-54e905014955	iPhone 6	Apple	{'color': 'Silver',
2	Phone	700	3e78a870-d4f9-400f-bbcc-33c5aa2a2503	iPhone 6	Dell	{'color': 'Silver',
3	Phone	1300	2ba3f6f8-d9b6-4c89-8ffa-e6c299a4800a	iPhone 6	Apple	{'color': 'Silver',

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

- назва,

```
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 product_id IS NOT NULL
PRIMARY KEY (category, name, price, product_id);
```

```
SELECT * FROM items_by_name WHERE category = 'Phone' AND name = 'iPhone 6';
```

	category	name	price	product_id	producer	properties
1	Phone	iPhone 6	100	4377c0c6-b468-4e7b-9ff9-54e905014955	Apple	{'color': 'Silver',
2	Phone	iPhone 6	700	3e78a870-d4f9-400f-bbcc-33c5aa2a2503	Dell	{'color': 'Silver',
3	Phone	iPhone 6	1300	2ba3f6f8-d9b6-4c89-8ffa-e6c299a4800a	Apple	{'color': 'Silver',

- ціна (в проміжку),

```
SELECT * FROM items WHERE category = 'Phone' AND price >= 500 AND price <= 1500;
```

	category	price	product_id	name	producer	properties
1	Phone	700	3e78a870-d4f9-400f-bbcc-33c5aa2a2503	iPhone 6	Dell	{'color': 'Silver',
2	Phone	1300	2ba3f6f8-d9b6-4c89-8ffa-e6c299a4800a	iPhone 6	Apple	{'color': 'Silver',

- 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;
```

	category	producer	price	product_id	name	properties
1	Phone	Apple	100	4377c0c6-b468-4e7b-9ff9-54e905014955	iPhone 6	{'color': 'Silver',

4) Напишіть запити, які вибирають товари за:

- о наявність певних характеристик

```
CREATE INDEX idx_items_properties ON items(KEYS(properties));
```

```
SELECT * FROM items WHERE category = 'Phone' AND properties CONTAINS KEY 'feature';
```

	category	price	product_id	name	producer	properties
1	Phone	100	4377c0c6-b468-4e7b-9ff9-54e905014955	iPhone 6	Apple	{'color': 'Silver',
2	Phone	700	3e78a870-d4f9-400f-bbcc-33c5aa2a2503	iPhone 6	Dell	{'color': 'Silver',
3	Phone	1300	2ba3f6f8-d9b6-4c89-8ffa-e6c299a4800a	iPhone 6	Apple	{'color': 'Silver',

- о певна характеристика та її значення

```
SELECT * from items WHERE category = 'Phone' AND properties['feature'] = 'Water Resistant' ALLOW FILTERING;
```

	category	price	product_id	name	producer	properties
1	Phone	100	4377c0c6-b468-4e7...	iPhone 6	Apple	{'color': 'Silver...
2	Phone	1300	2ba3f6f8-d9b6-4c8...	iPhone 6	Apple	{'color': 'Silver...

5) Оновити опис товару:

- о змінити існуючі значення певної характеристики

```
UPDATE items SET properties['feature'] = 'Waterproof' WHERE category = 'Phone' AND price = 700 AND product_id = 3e78a870-d4f9-400f-bbcc-33c5aa2a2503;
```

	price	product_id	name	producer	properties
1	100	4377c0c6-b468-4e7b-9ff9-54e905014955	iPhone 6	Apple	{'color': 'Silver', 'feature': 'Water Resistant'}
2	700	3e78a870-d4f9-400f-bbcc-33c5aa2a2503	iPhone 6	Dell	{'color': 'Silver', 'feature': 'Waterproof'}
3	1300	2ba3f6f8-d9b6-4c89-8ffa-e6c299a4800a	iPhone 6	Apple	{'color': 'Silver', 'feature': 'Water Resistant'}

- о додайте нові властивості (характеристики) товару

```
UPDATE items SET properties = properties + {'color_additional': 'Black'} WHERE category = 'Phone' AND price = 700 AND product_id = 3e78a870-d4f9-400f-bbcc-33c5aa2a2503;
```

	price	product_id	name	producer	properties
1	100	4377c0c6-b468-4e7b-9ff9-54e905014955	iPhone 6	Apple	{'color': 'Silver', 'feature': 'Water Resistant'}
2	700	3e78a870-d4f9-400f-bbcc-33c5aa2a2503	iPhone 6	Dell	{'color': 'Silver', 'color_additional': 'Black', 'fe...
3	1300	2ba3f6f8-d9b6-4c89-8ffa-e6c299a4800a	iPhone 6	Apple	{'color': 'Silver', 'feature': 'Water Resistant'}

- о видаліть характеристику товару

```
DELETE properties['feature'] FROM items WHERE category = 'Phone' AND price = 700 AND product_id = 3e78a870-d4f9-400f-bbcc-33c5aa2a2503;
```

	price	product_id	name	producer	properties
1	100	4377c0c6-b468-4e7b-9ff9-54e905014955	iPhone 6	Apple	{'color': 'Silver', 'feature': 'Water Resistant'}
2	700	3e78a870-d4f9-400f-bbcc-33c5aa2a2503	iPhone 6	Dell	{'color': 'Silver', 'color_additional': 'Black'}
3	1300	2ba3f6f8-d9b6-4c89-8ffa-e6c299a4800a	iPhone 6	Apple	{'color': 'Silver', 'feature': 'Water Resistant'}

Створіть таблицю *orders* в якій міститься ім'я замовника і інформація про замовлення: перелік id-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*)

	keyspace_name	type	name	create_statement
1	online_store	table	orders	CREATE TABLE online_store.orders (customer_name VARCHAR(255) NOT NULL, order_date DATETIME NOT NULL, order_id VARCHAR(255) NOT NULL, order_value DECIMAL(10, 2) NOT NULL, product_ids VARCHAR(255) NOT NULL, PRIMARY KEY (customer_name, order_id))

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

```
SELECT * FROM orders WHERE customer_name = 'Alice Johnson';
```

	customer_name	order_date	order_id	order_value	product_ids
1	Alice Johnson	2023-09-14 16:55:11.132	0d8391da-c8c7-4951-85e4-40fba2663a9c	2200	[3c77f043-c8c7-4951-85e4-40fba2663a9c]
2	Alice Johnson	2023-09-14 16:55:10.670	057f695f-2a71-42c6-b31d-f01177ddd19e	1200	[db2d2215-2a71-42c6-b31d-f01177ddd19e]
3	Alice Johnson	2023-09-14 16:55:10.599	8a21de91-8ee7-4924-a617-6e8c2d3eb76f	1800	[4377c0c6-b468-4e7b-9ff9-54e905014955]

- 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

```
SELECT * FROM orders WHERE customer_name = 'Alice Johnson' AND order_date >= '2023-01-01' AND order_date <= '2023-12-31';
```

	customer_name	order_date	order_id	order_value	product_ids
1	Alice Johnson	2023-09-14 16:55:11.132	0d8391da-c8c7-4951-85e4-40fba2663a9c	2200	[3c77f043-c8c7-4951-85e4-40fba2663a9c]
2	Alice Johnson	2023-09-14 16:55:10.670	057f695f-2a71-42c6-b31d-f01177ddd19e	1200	[db2d2215-2a71-42c6-b31d-f01177ddd19e]
3	Alice Johnson	2023-09-14 16:55:10.599	8a21de91-8ee7-4924-a617-6e8c2d3eb76f	1800	[4377c0c6-b468-4e7b-9ff9-54e905014955]

```
SELECT COUNT(*) FROM orders WHERE customer_name = 'Alice Johnson' AND order_date >= '2023-01-01' AND order_date <= '2023-12-31';
```

	count
1	3

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

```
SELECT customer_name, SUM(order_value) AS total_sum FROM orders GROUP BY customer_name
```

	customer_name	total_sum
1	Bob Smith	5300
2	Charlie Williams	3900
3	Alice Johnson	5200

- 6) Для кожного з замовників визначте замовлення з максимальною вартістю

```
SELECT customer_name, MAX(order_value) AS max_order_value, order_id FROM orders GROUP BY customer_name;
```

	customer_name	max_order_value	order_id
1	Bob Smith	1800	d865daed-6a9b-4964-a1b3-4cd9c68e9f1f
2	Charlie Williams	1900	38043a56-ac7e-4787-b144-240634894c62
3	Alice Johnson	2200	0d8391da-c8c7-4951-85e4-40fba2663a9c

- 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-4964-a1b3-4cd9c68e9f1f;

[
  {
    "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		
	⏏ writetime(order_value) ⚡	⏏ order_id ⚡
1	1694713858931000	d865daed-6a9b-4964-a1b3-4cd9c68e9f1f
2	1694710510866000	13e54c37-3486-4376-8c36-27bba9601178
3	1694710510772000	761c8a17-7313-49e8-9008-ebdbb5cc66
4	1694710510735000	9368b631-7d76-4597-8ff5-55a28bdcf15c
5	1694710511162000	38043a56-ac7e-4787-b144-240634894c62
6	1694710511094000	483f34ba-7d9e-4e5f-bd80-89bdde9c8f4c
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', uuid(), [844a439a-1c14-47fc-9d5c-4fdf427b9797], 800)
USING TTL 3600;
```

- 10) Поверніть замовлення у форматі JSON

```
CREATE MATERIALIZED VIEW orders_by_date AS
SELECT * FROM orders
WHERE order_date IS NOT NULL AND customer_name IS NOT NULL AND order_id IS NOT NULL
PRIMARY KEY (order_date, customer_name, order_id);
```

```
SELECT JSON * FROM orders_by_date WHERE order_date = '2023-09-15'
```

[json]

1	{"order_date": "2023-09-15 00:00:00.000Z", "customer_name": "Alice Johnson", "order_id": "bf92f847-aac2-4130..."}
---	---

- 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