

Міністерство освіти і науки України  
НТУУ «Київський політехнічний інститут»  
Фізико-технічний інститут

**Проектування високонавантажених систем**

Лабораторна робота №5

Робота з базовими функціями БД типу column family на прикладі Cassandra

**Виконав:**

Студент 4-го курсу

групи ФІ-21

Климентьев Максим

**Перевірив:**

---

# **Зміст**

<b>1</b>	<b>Код реалізації</b>	<b>3</b>
<b>2</b>	<b>Результати</b>	<b>4</b>
2.1	Частина 1. Робота зі структурами даних у Cassandra . . . . .	4
2.2	Частина 2. Налаштування реплікації у Cassandra . . . . .	5
2.3	Частина 3. Аналіз продуктивності та перевірка цілісності . . . . .	7

# **1 Код реалізації**

## 2 Результати

### 2.1 Частина 1. Робота зі структурами даних у Cassandra

Створіть keyspace з найпростішої стратегією реплікації

В цьому keyspace необхідно буде створити дві таблиці: items та orders Таблиця items містить різноманітні товари (тобто у яких різний набір властивостей).

Для набору властивостей товару виберіть базові характеристики однакові для всіх товарів (name, category, price, producer, ...), а для властивостей які відрізняються використовуйте тип map (створивши індекс для можливості пошуку по її вмісту)

Увага: Необхідно, щоб пошук швидко працював для категорії (category) товарів. Ця вимога має бути врахована при створенні ключа для таблиці (тобто, category має бути partition key).

Наповніть таблиці тестовими даними

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

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

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

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

- назва,
- ціна (в проміжку),
- ціна та виробник

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. Ця вимога має бути врахована при створенні ключа для таблиці (аналогічно як для items).

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

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

3. Для кожного замовників підрахуйте загальну суму на яку були зроблені усі його замовлення

4. Для кожного замовлення виведіть час коли його ціна була занесена в базу (SELECT WRITETIME)

## 2.2 Частина 2. Налаштування реплікації у Cassandra

1. Сконфігурувати кластер з 3-х нод:

2. Перевірити правильність конфігурації за допомогою nodetool status

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

4. В кожному з кейспейсів створити прості таблиці

1. Спробуйте писати і читати в ці таблиці підключаюсь до різних нод через cqlsh.

2. Вставте дані в створені таблиці і подивітесь на їх розподіл по вузлах кластера окремо для кожного з кейспесов (команда nodetool status) - має бути видно відсоток даних який зберігається на ноді

3. Для якогось запису з кожного з кейспейсу виведіть ноди на яких зберігаються дані - має бути видно ір-адреси вузлів на яких зберігається даний рядок

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

- для Keyspace з replication factor 1 - CONSISTENCY ONE

- для Keyspace з replication factor 2 - CONSISTENCY ONE/TWO

- для Keyspace з replication factor 3 - CONSISTENCY ONE/TWO/THREE

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

2. 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)

3. 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

## 2.3 Частина 3. Аналіз продуктивності та перевірка цілісності

Аналогічно попереднім завданням, необхідно, для кластеру налаштованому у попередній частині, створити таблицю з каунтером лайків. Далі з 10 окремих клієнтів одночасно запустити інкрементацію каунтеру лайків по 10\_000 на кожного клієнта з різними опціями взаємодії з Cassandra.

Таблиця має бути створена у Keyspace з replication factor 3.

Для створення каунтеру використовуйте спеціальний тип колонки - counter (цей тип буде підтримувати операції increment/decrement in-place):

1. Вказавши у параметрах запиту Consistency Level One (це буде означати, що запис відбувається синхронно тільки на одну ноду), запустіть 10 клієнтів з інкрементом по 10\_000 на кожному з них. Виміряйте час виконання та перевірте чи кінцеве значення буде дорівнювати очікуваному - 100K

2. Вказавши у параметрах запиту Consistency Level QUORUM (це буде означати, що запис відбувається синхронно на більшість нод), запустіть 10 клієнтів з інкрементом по 10\_000 на кожному з них. Виміряйте час виконання та перевірте чи кінцеве значення буде дорівнювати очікуваному - 100K