Báo cáo tiến độ

Track 1: Java + Spring Boot

Tạo Spring Boot application bằng Intellij

Không gặp khó khăn gì

Register bean using @Bean or @Component

@Component : Tạo class instance , register vào bean container

@Bean : Tạo bean từ method

Custom Port, Prefix Path using Java Configuration

Không gặp khó khăn gì

Create non-web application

Implement CommandLineRunner

Config file application.properties

spring.main.web-environment=false  
spring.main.web-application-type=NONE

Dependency Injection with specific Profile

Tạo các file application-x.properties với x là profile tương ứng

Annotation @Profile(x) tạo bean tương ứng với từng profile

Inject all beans instance of an interface

Không gặp khó khăn gì

Cài đặt Lombok

Không gặp khó khăn gì

Multithreading

Tạo , chạy các thread

Xử lý race condition trên các critical section với synchronized block , Lock interface ( Reentrant lock )

Track 2: Mysql

CRUD sử dụng JDBC

Viết query , xử lý kết quả từ ResultSet

Parameterized query

CRUD sử dụng Spring JPA

Sử dụng các JPA template từ CrudRepository

Sử dụng custom query với @Query , JPQL

Multiple datasource Spring JPA

Override bean DataSource , EntityManagerFactory cho từng datasource tương ứng

Pagination using Mysql

Limit , offset

Tìm hiểu Partition, Indexing trong Mysql

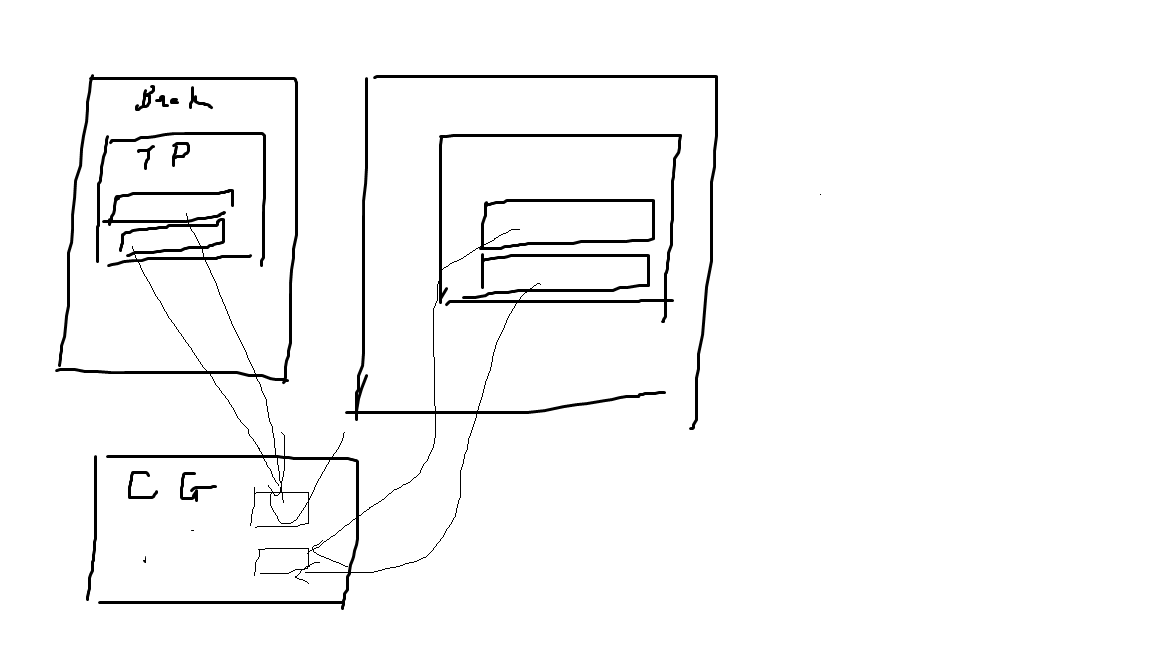
Indexing:

Hiểu được cách tổ chức , lưu trữ với B-tree

Các vấn đề và câu hỏi:

1. Cách B-tree so sánh key với composite key ( 1 index trên nhiều cột ) .
   * Có thể mỗi lead node của B tree với cột A là 1 B-tree với trường B ?
2. Cách tìm kiếm trên index với query trên 2 cột ( 2 cột đều có index )
   * Tìm theo index trên từng cột rồi lấy intersect ?

Track 3: Kafka

Kiến trúc kafka 

Thuật ngữ

Log : one way queue (append only )

Current offset : Message mới nhất chưa được gửi

Committed offset : Message mới nhất được xử lý , đánh dấu là committed. Sử dụng khi rebalancing group , hoặc consumer nào đó bị sập . Consumer mới sẽ nhận message bắt đầu từ committed message mới nhất

Cài đặt Kafka

Không gặp vấn đề gì

Produce message

Sử dụng java kafka client , gửi message vào topic theo phương thức mặc định ( round-robin )

Consume message

Tạo nhiều service chạy trên các port khác nhau , cùng subscribe vào 1 topic

Set group id giống nhau

Quan sát consumer group rebalance khi bật/tắt các consumer

Consumer group

Một tập các consumer

Không quản lý offset trên các partition

Mỗi consumer group có offset riêng đối với topic đang được consumed ( quản lý bởi kafka )

Tìm hiểu về partition và replicate trong Kafka

Vai trò partition :

* 1 partition chỉ có thể được consumed bởi 1 consumer
* Chia nhỏ topic để được consumed song song

Vai trò replica :

* Đảm bảo tính sẵn sàng của dữ liệu
* Trên mỗi node/brokder có các replica của các partition
* Thực hiện trên partition level

Tìm hiểu vai trò của ZooKeeper trong Kafka

* Quản lý thông tin các topic , broker , vị trí các partition
* Chọn leader cho các partition
* Kiếm tra trạng thái của các broker
* Quản lý thông tin các offset trên partition