

Project Definition and Rubric (CSC545 Final Project)

Overview

You will create a project that demonstrates the Command Query Responsibility Segregation (CQRS) pattern with Python, using MySQL for the write side (command) and Cassandra for the read side (query). This project will help you understand the principles, benefits and drawbacks of using a modern architectural pattern (CQRS), demonstrate the differences between SQL and NoSQL databases, and practice integrating multiple data stores within a single application using an event hub or service bus.

Project Definition

Objective: Implement a Python application that adheres to the CQRS architecture pattern. The application should handle write operations using MySQL and read operations using Cassandra. This project aims to demonstrate the separation of concerns, scalability, and the practical application of combining SQL and NoSQL databases within a modern architecture.

Scope:

The project itself does not have to be unnecessarily complicated, and does not require you to create a modern user interface (a text based interface will suffice). At a minimum, the project must contain:

1. A python application that uses a task based interface that can write data to a MySQL database for at least 2 different user interface tasks. This will represent the "Command" part of the CQRS pattern. An example of a user interface task could be something like "post tweet" or "change seat reservation". **Do not create a CRUD (create/read/update/delete) interface.**
2. Custom defined user events. We will discuss this in class.
3. A module that implements an event hub or service bus that will subscribe to your application, and accept events onto the hub/bus.
4. A module that will "pop" events off the bus and add those events to keyspace(s) / table(s) in Cassandra.
5. A module that will allow a user to project data from Cassandra into a user interface. This will represent the "Query" part of the CQRS pattern.

Where to Begin

- We will begin discussing these concepts in Lecture 8. You will be provided:
 1. Instructions on how to setup Cassandra on your local computer, or in a public cloud account (Azure / AWS / etc..)
 2. Instructions for how to setup an event hub or service bus in python
 3. Some boilerplate code to demonstrate some of the required concepts (e.g. querying mysql, creating a text based interface, querying cassandra (reads and writes), using a service bus, etc..)
 4. A discussion on the principles of CQRS
 5. A discussion on the basics of using Cassandra (read/write operations, dynamic schemas, setting up a cassandra cluster, consistency tuning, etc..)
 6. We will assume that you already have MySQL installed on your local laptop, and that you have python (and pip) installed locally
- You will need to choose a domain to implement your application. This could be anything that uses a task based interface, such as a social media site, airplane reservation system, or any domain allowing for clear command (write) and query (read) operations.
- You can work in groups of 1 or 2. If you work in a group of 2, the group will share the grade.
- We will dedicate significant class time to allow you to work on your projects, ask questions, etc..

Project Rubric

1. Project Setup and Proposal (10%)

- Demonstrate that you were able to setup the environment correctly, and were able to implement the required project components. You will also create a 2-3 sentence proposal that documents the domain or problem you want to solve.

2. Database Schema Design (20%)

- **MySQL (Write Side):** Well-structured relational schema with tables, relationships, and indexes optimized for performance.
- **Cassandra (Read Side):** Effective NoSQL schema design optimized for query efficiency, utilizing Cassandra's strengths in horizontal scaling and fast read operations.

3. Application Build (50%)

- Implementation of the CQRS pattern, demonstrating clear separation between command and query responsibilities.
- Use of appropriate Python frameworks (e.g., Flask, Django for the web interface, SQLAlchemy for ORM with MySQL, and a Cassandra driver for Python).
- Demonstrating that you were able to build a functioning application.

4. Demonstration (20%)

- At the end of the semester, perform a 5-10 minute demo that shows your functioning application, and can discuss what you learned (challenges, impediments, etc..)

Deliverables

- The source code you used to create your application. You can submit your source code in a zip file into Microsoft Teams
- All database SQL and CQL statements you used to setup your databases (both schema and queries). Include this in your zip file
- Your 2-3 sentence project statement (the problem you are solving). You will submit this in an upcoming assignment.
- Your demo at the end of the semester.

Assessment Criteria

- **Completeness:** All requirements are implemented, and the application functions as expected.
- **Understanding:** You can demonstrate a clear understanding of the CQRS pattern, database technologies, and application architecture principles.
- **Code Quality:** The code is well-organized, readable, and follows best practices for Python development and database interactions.
- **Innovation and Problem Solving:** You show creativity in solving architectural and data synchronization challenges.