

Attempt 1



In Progress

NEXT UP: Submit Assignment



Add Comment

Unlimited Attempts Allowed

▼ Details

System Requirements

Create a web application that has both transactional and analytical operations. The entire system must be launched using a Docker Compose file.

Transactional Operations

- Choose a domain that will have **batch updates and race conditions**. Examples:
 - University enrollment (limited slots per class)
 - Flight seat booking
 - Online ordering (limited inventory quantity per product)
 - Inventory
 - Movie seat reservation

You are not limited to the above domains, but your system must have **batch updates and race conditions**.

- Avoid deadlocks. Use deadlock avoidance strategies.
- Design your database schema to be optimized for these transactional operations.
- Do not do in your application what should be done in the database, and vice-versa.

Analytical Operations

- Prepare appropriate reports and visualizations.
 - At least 3.
- Design your database schema to be optimized for these analytical operations.
- Do not do in your application what should be done in the database, and vice-versa.

Primary Database

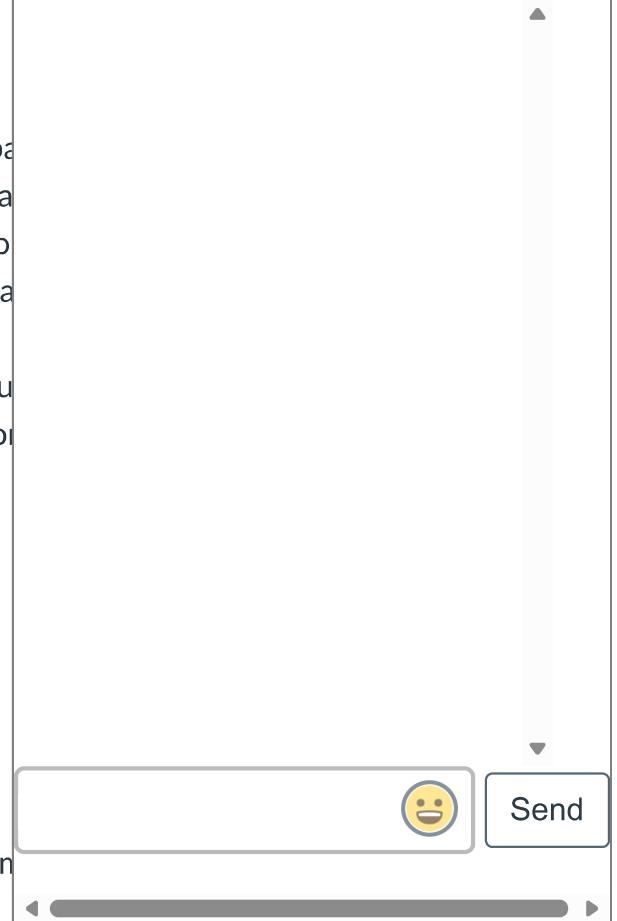
- All transactional operations should hit the primary database server.
- Analytical operations should only be done on the primary database if the data needs to be real-time, otherwise the data should be read from the reports & visualizations server.
- Normalize and/or denormalize appropriately.

Replication & Backup

- **Hot Backup Server**
 - Use a **physical replication** to replicate the main database.
 - Bonus points if you can implement and demonstrate a hot backup server.
- **Reports & Visualizations Database Server ("Data Warehouse")**
 - All reports and visualizations that don't need to be real-time can be stored in a separate reports server, denormalized appropriately.
 - Use **logical replication** and appropriate triggers and functions to keep the reports reasonably up-to-date, as appropriate for the repository.
 - Replicate only the data needed for the reports.
- Implement WAL archiving on an hourly basis.

Load Testing

- Populate your tables with large amounts of data
- Use an automated load testing tool (e.g. JMeter)
- Create load testing test scripts.
- Execute the test scripts and generate a report.
- If there are performance issues or deadlocks, resolve them.



What to Submit

- Video recording of presentation
 - 5 - 15 minutes
- Link to Git Repository
 - Including **docker compose file**. System should be able to launch via a docker compose file, including load testing
- Latest load test report

⌄ View Rubric

MCO2

Criteria	Ratings	Pts
Transactional Operations view longer description	20 to >0 pts Full Marks	0 pts No Marks / 20 pts
Analytical Operations view longer description	20 to >0 pts Full Marks	0 pts No Marks / 20 pts
Primary Database view longer description	20 to >0 pts Full Marks	0 pts No Marks / 20 pts

MCO2

Criteria	Ratings	Pts
Replication & Backup view longer description	20 to >0 pts Full Marks	0 pts No Marks / 20 pts
Load Testing view longer description	20 to >0 pts Full Marks	0 pts No Marks / 20 pts
Total Points: 0		

Keep in mind, this submission will count for everyone in your MCO2 Teams group.

Choose a submission type



Submit Assignment