Ira A. Fulton Schools of Engineering
Arizona State University

Term: Fall 2023 **Subject:** Computer Science & Engineering (CSE) **Number:** 512

Course Title: Distributed **D**atabase **S**ystems (CSE 512)

GROUP PROJECT TOPIC PROPOSAL

Team Name	E-Com DDS by Team Bro Code	
Team Members	Jaagruti Reddy Duggireddy Surakkagari	1225469193
	Priyadarshini Ramakrishnan	1225407339
	Santhosh Kumar Jorka	1225462862
	Sunil Chowdary Vejendla	1229832203
Project Topic	We have chosen to design a distributed database system for an **E-commerce** platform because it's the need of the hour due to several critical factors that have become increasingly important in this day's digital economy. E-commerce platforms experience varying traffic levels, especially during peak shopping seasons. A distributed database allows it to scale horizontally and remain responsive by adding more nodes to handle increased loads. E-commerce businesses must be available 24/7 to cater to customers across different time zones. We can configure the DB with failover mechanisms to ensure that even if one server or data center goes down, the system remains accessible, minimizing downtime and revenue loss. Maintaining data consistency is crucial to prevent issues like overselling products or displaying incorrect prices. We can employ techniques such as distributed transactions to ensure consistent data updates across all nodes. E-commerce platforms require real-time inventory management and pricing updates. A distributed database system can support these requirements by allowing multiple concurrent updates. E-commerce businesses often operate in multiple regions or countries. We can store data closer to customers, reducing data transfer times and enhancing the user experience. • **Products Table:** This table can store information about the products sold, such as product ID, name, description, price, and category ID. The category ID can link to a Categories Table.	

<table>
<tr><td></td><td colspan="2">

Categories Table: This table can store category ID and name fields. Products can be associated with their categories through the category ID.
Customers Table: It can contain customer information, like ID, name, email, and address details. This customer ID can be linked to orders placed by the customer in the Orders Table.
Orders Table: It can record customer orders and includes order ID, customer ID, order date, and order status. Each order can consist of multiple order Items, which can be linked through order ID.
Order Items Table: To track individual products within orders, this table can include order item ID, order ID, product ID, quantity ordered, and item price. Thereby linking to the product table and orders table.
Inventory Table: Can maintain product stock levels with fields like product ID from the product table, available amount, and other stock-related details.
Transactions Table: Can record payment transactions related to orders, including transaction ID, order ID (linked to the Orders Table), payment method, timestamp, and amount.
Log and Audit Table: Keeps a log of system activities and changes, including user actions and timestamps, to aid auditing and troubleshooting.
Reviews and Rating Table: Can store customer feedback about products, including review ID, product ID (linked to the Products Table), customer ID (linked to the Customers Table), review text, and rating.

</td></tr>
<tr><td>Plan of Action</td><td>Part 1: Design and Implementation of a Distributed Database System</td><td>
<u>Intended tasks to complete:</u>

Designing a schema for a distributed database system tailored to an E-commerce platform
Creating the necessary tables in the schema to store data efficiently and distributing the data across multiple nodes or data centers
Inserting data in real-time as new products and transactions occur, and implementing mechanisms for retrieving the data promptly

<u>Tentative tools usage:</u>

PostgreSQL, Python

<u>Expected deliverables:</u>
</td></tr>
</table>

		• Creating a schema with clear entity-relationship diagrams and table definitions • Implemented code/script to create essential tables with attributes, keys, and constraints • Document the chosen data distribution strategy • Implemented code/script for efficient initial data insertion • Implemented code/script to demonstrate successful data retrieval
	Part 2: Fragmentation and Replication Techniques	<u>Intended tasks to complete:</u> • Implementing horizontal data fragmentation (Split tables into subsets based on specific criteria) • Implementing vertical data fragmentation (Divide tables into smaller subsets based on columns to optimize data retrieval) <u>Tentative tools usage:</u> • PostgreSQL, Python <u>Expected deliverables:</u> • Implemented code/script for the fragmentation process • Result snapshots • Documentation elucidating the assumptions made and the approach employed for implementation
	Part 3: Query Processing and Optimization Techniques	<u>Intended tasks to complete:</u> • Analyzing and optimizing the queries that retrieve the data <u>Tentative tools usage:</u> • PostgreSQL, Python <u>Expected deliverables:</u> • Implemented code/script for the fragmentation process • Result snapshots • Documentation describing our implementation process

	Part 4: Distributed Transaction Management	Intended tasks to complete: ● Implementing ACID-Compliant Distributed Transactions ● Proposing Concurrency Control Mechanisms by evaluating distributed locking mechanisms. ● Utilizing Apache Ignite to provide high performance and scalability for distributed computing and real-time data processing Tentative tools usage: ● PostgreSQL, Apache Ignite Expected deliverables: ● Working implementation of distributed transactions that ensures ACID properties. ● Documentation describing the setup and configuration of the distributed database environment and proposed concurrency control mechanisms. ● Application code demonstrating how distributed transactions are initiated and managed.
	Part 5: Distributed NoSQL Database Systems Implementation	Intended tasks to complete: ● Defining and documenting the data schema and data model for NoSQL database system ● Implementing basic CRUD (Create, Read, Update, Delete) operations for the domain-specific data ● Creating sample queries and data retrieval operations Tentative tools usage: ● MongoDB, Docker Expected deliverables: ● Implemented code/script for queries ● Result snapshots ● Documentation elucidating the assumptions made and the approach employed for implementation

	Part 6: 3-Minute Video Demo	Tentative tools to be used: ● OBS Studio, OpenShot Video Editor Video structure: ● Significance of the project in the context of our project topic ● Highlighting the main components of the project ● Demonstration of the distributed database system ● Design, implementation, and functionality of the distributed database system ● Project's objectives, challenges, and solutions ● Snapshots/diagrams to illustrate key concepts and results Other details: ● Narration (voice-over) will be included in the video
Team Member Contribution/ Responsibility	Jaagruti Reddy Duggireddy Surakkagari	Roles/Responsibilities: ● Conduct in-depth research on the chosen topic and technologies. Develop a detailed plan and timeline for the project. ● Collaborate with the team to design a robust and scalable database schema. Create entity-relationship diagrams and define tables, attributes, keys, and constraints. ● Conduct thorough testing of each part of the project, ensuring that every function performs correctly and efficiently. ● Document the entire process, including design, implementation, optimization strategies, and testing results. ● Create a concise and informative video presentation showcasing the project, its objectives, and achievements.

		• Ensure effective communication within the team, coordinate tasks, and update the team on progress and challenges. • Provide constructive feedback on teammates' work and incorporate feedback received.
	Sunil Chowdary Vejendla	<u>Roles/Responsibilities:</u> • Conduct in-depth research on the chosen topic and technologies. Develop a detailed plan and timeline for the project. • Implement and manage the NoSQL database system, define the data schema, and implement CRUD operations. • Conduct thorough testing of each part of the project, ensuring that every function performs correctly and efficiently. • Document the entire process, including design, implementation, optimization strategies, and testing results. • Create a concise and informative video presentation showcasing the project, its objectives, and achievements. • Address any issues or challenges that arise during the project and work on solutions. • Provide constructive feedback on teammates' work and incorporate feedback received. • Continuously learn and adapt to new technologies and methods as required by different parts of the project.

	Santhosh Kumar Jorka	<u>Roles/Responsibilities:</u> ● Conduct in-depth research on the chosen topic and technologies. Develop a detailed plan and timeline for the project. ● Collaborate with the team to design a robust and scalable database schema. Create entity-relationship diagrams and define tables, attributes, keys, and constraints. ● Implement fragmentation and replication techniques to optimize system performance. ● Analyze, optimize, and test queries for efficient data retrieval. Implement distributed indexing strategies. ● Conduct thorough testing of each part of the project, ensuring that every function performs correctly and efficiently. ● Document the entire process, including design, implementation, optimization strategies, and testing results. ● Create a concise and informative video presentation showcasing the project, its objectives, and achievements. ● Provide constructive feedback on teammates' work and incorporate feedback received.
	Priyadarshini Ramakrishnan	<u>Roles/Responsibilities:</u> ● Conduct in-depth research on the chosen topic and technologies. Develop a detailed plan and timeline for the project. ● Collaborate with the team to design a robust and scalable database schema.

		Create entity-relationship diagrams and define tables, attributes, keys, and constraints. ● Implement the designed schema, create tables, and ensure data consistency and integrity. ● Implement ACID-compliant distributed transactions and propose concurrency control mechanisms. ● Conduct thorough testing of each part of the project, ensuring that every function performs correctly and efficiently. ● Document the entire process, including design, implementation, optimization strategies, and testing results. ● Create a concise and informative video presentation showcasing the project, its objectives, and achievements. ● Monitor the progress of the project against the timeline and ensure that all tasks are completed on time. ● Develop and implement strategies for data backup and recovery. ● Ensure all components are correctly organized, named, and submitted according to the project requirements. ● Provide constructive feedback on teammates' work and incorporate feedback received.

Signature:

Jaagruti Reddy DS Priyadarshini R Sunil V Santhosh J