Lab with Capstone Project-DEA

Code: CS32242CS	Course Type: Prct.	Semester: II
Evolution Scheme (TA-MSE-ESE): 25-25-50	Total Marks: 100	L-T-P (Credits): 0-0-4 (2)

Pre-Requisites: DBMS, Python/SQL, Data Structure, Probability & Statistics

Course Objective: To provide hands-on experience with relational database internals, indexing and query optimization, distributed and NoSQL systems, ETL pipelines, big-data processing frameworks, and scalable batch and streaming analytics workflows.

Course Outcomes: At the end of the course, the student will be able to

CO-1	Implement schema design and advanced SQL optimization techniques
CO-2	Apply transactions, concurrency control and recovery mechanisms
CO-3	Develop distributed and NoSQL database solutions
CO-4	Build scalable batch and streaming analytics pipelines

Course Articulation Matrix:

PO	PEO-1	PEO-2	PEO-3	PEO-4	PEO-5
CO-1	3	2	1	-	-
CO-2	3	2	1	-	1
CO-3	3	2	-	-	3
CO-4	3	2	1	1	-

1 - Slightly; 2 - Moderately; 3 – Substantially

1. Design an ER model for a real-world application and convert it into relational schema.
2. Perform normalization of given relations up to 3NF and BCNF and verify removal of anomalies.
3. Write advanced SQL queries using joins, nested queries and views for complex data retrieval.
4. Implement stored procedures, triggers and user-defined functions for enforcing database logic.
5. Apply primary key, foreign key and check constraints to maintain database integrity.
6. Generate and analyze query execution plans using EXPLAIN PLAN for different SQL queries.
7. Create B+ Tree and Hash indexes and compare their effect on query performance.
8. Compare execution time of indexed and non-indexed queries using large datasets.
9. Implement and compare Nested Loop Join, Sort-Merge Join and Hash Join algorithms.
10. Demonstrate ACID properties using transaction control commands such as COMMIT and ROLLBACK.
11. Perform transactions using different isolation levels and observe concurrency anomalies.
12. Simulate deadlock situations and apply suitable detection and recovery techniques.
13. Implement log-based recovery and checkpoint mechanisms for database failure handling.
14. Perform horizontal and vertical fragmentation in distributed database environment.
15. Implement primary–replica database replication and evaluate availability improvements.
16. Perform CRUD operations and aggregation queries using MongoDB document database.
17. Implement key-value storage operations using Redis and analyze retrieval performance.
18. Execute graph traversal and shortest-path queries using Neo4j graph database.
19. Design and implement an ETL pipeline using Python/Pandas for structured dataset processing.
20. Develop a real-time streaming analytics pipeline using Kafka and Spark Streaming/Flink.

Learning Resources:

Text Books:

1) Raghu Ramakrishnan and Johannes Gehrke, Database Management Systems, McGraw Hill, 3rd Edition.

2) Abraham Silberschatz, Henry F. Korth, S. Sudarshan, Database System Concepts, McGraw Hill, 7th Edition.
3) Martin Kleppmann, Designing Data-Intensive Applications, O'Reilly Publications, 2017.