Indexing in Database

Index Creation:

Create a sample database schema with tables and representative data. Ensure that the schema contains at least two tables with a significant number of records.

Identify appropriate columns in each table that should be indexed to improve query performance.

Write SQL statements to create indexes on the chosen columns.

Execute the SQL statements and verify that the indexes are successfully created.

Index Analysis:

a. Select three sample queries that involve different tables and columns from your database schema.

b. Execute each query multiple times, measuring the execution time without any indexes.

c. Apply appropriate indexes on the required columns involved in the queries.

d. Re-execute the queries multiple times, measuring the execution time with indexes.

e. Record the execution times for each query, both before and after index creation.

f. Analyze the results and describe the impact of indexing on query performance. Discuss any significant improvements or drawbacks observed.

Theoretical Questions:

Answer the following questions in a concise and clear manner:

a. What is indexing in a database, and why is it important?

b. Explain the difference between clustered and non-clustered indexes.

c. When should you consider creating an index on a column? Are there any scenarios where indexing can have a negative impact?

d. How does indexing affect data modification operations, such as insert, update, and delete?

e. What are some alternative data structures or indexing techniques used in modern databases?

Understanding Isolation Levels in Databases

Research:

a) Start by researching the concept of isolation levels in databases. Familiarize yourself with the different isolation levels commonly used, such as Read Uncommitted, Read Committed, Repeatable Read, and Serializable.

b) Explore the key features and characteristics of each isolation level. Understand how they control concurrent access to data, handle dirty reads, non-repeatable reads, and phantom reads.

c) Investigate the advantages and disadvantages of each isolation level in terms of data consistency, concurrency, and performance.

Assignment Components:

a) Write a brief summary describing the concept of isolation levels and their significance in database systems. Explain how different isolation levels help balance data consistency, concurrency, and performance.

b) Create a comparative table or chart that presents the different isolation levels, their characteristics.

c) Discuss the advantages and disadvantages of each isolation level, highlighting their impact on data consistency, concurrency control, and overall database performance.

d) Explain how the choice of isolation level can affect the behavior of database transactions and the potential challenges faced by developers in different scenarios.

e) Provide recommendations for selecting the appropriate isolation level based on specific requirements, such as transactional consistency, concurrency, and performance optimization.