**DATABASE MANAGEMENT SYSTEMS (U20ITT306)**

**UNIT I: INTRODUCTION**

1. **Explain Database System Architecture.**

Database System Architecture refers to the arrangement of components that make up a database system, including hardware, software, data, procedures, and users.

1. **Explain the Entity-Relationship Model.**

The Entity-Relationship Model is a visual representation used to describe the entities, relationships, and attributes in a database. Entities are represented by rectangles, relationships by diamonds, and attributes by ovals.

1. **Discuss the process of converting an ER Diagram into a Relational Model.**

The process involves mapping entities to tables, relationships to foreign keys, and attributes to columns. Each entity becomes a table, and relationships are represented through keys.

1. **Describe the structure of Relational Databases.**

Relational Databases consist of tables with rows and columns. Each table represents an entity, and relationships are established through keys.

1. **Compare and contrast the Extended ER Model with the basic Entity-Relationship Model.**

The Extended ER Model extends the basic model by adding concepts like subclasses and inheritance. It provides more expressive power compared to the basic model.

1. **Explain the concept of keys in the Relational Model.**

Keys in the Relational Model are used to uniquely identify records in a table. Primary keys are used to uniquely identify each record, and foreign keys establish relationships between tables.

**UNIT II: DATABASE LANGUAGES**

1. **What is the purpose of DDL in SQL?**

DDL (Data Definition Language) in SQL is used to define and manage the structure of the database, including creating, altering, and deleting tables and defining constraints.

1. **Explain different types of Database languages in detail.**

Database languages include DDL for schema definition, DML for data manipulation, and SQL for querying. Procedural languages like PL/SQL and T-SQL are used for creating stored procedures and triggers.

1. **Define Set Operations in the context of SQL.**

Set operations in SQL include UNION, INTERSECT, and EXCEPT, used to combine or compare sets of rows from different tables.

1. **Discuss the importance of Triggers in a database.**

Triggers in a database are special procedures that automatically execute in response to specific events, ensuring data integrity and consistency.

1. **Explain the concept of Nested Queries in SQL.**

Nested Queries involve embedding one query within another. They are used to retrieve information based on the results of another query.

1. **Compare and contrast Relational Algebra with Extended-Relational Algebra Operations.**

Relational Algebra deals with basic operations like selection, projection, and join. Extended-Relational Algebra includes additional operations like aggregation and grouping.

1. **Describe the role of Stored Procedures in database languages.**

Stored Procedures are precompiled SQL statements stored in the database. They improve performance and allow for modular programming.

1. **Explain different types of Joins.**

Joins combine rows from two or more tables based on a related column. Types include INNER JOIN, LEFT JOIN, RIGHT JOIN, and FULL OUTER JOIN.

**UNIT III: RELATIONAL-DATABASE DESIGN AND DATA STORAGE**

1. **What is Schema Refinement in the context of Relational Database Design?**

Schema Refinement involves improving the design of a database schema to eliminate redundancy and improve efficiency.

1. **Define Functional Dependencies.**

Functional Dependencies describe the relationship between attributes in a database, indicating how changes in one attribute affect another.

1. **Discuss the significance of Normal Forms (1NF, 2NF, 3NF, BCNF, 4NF) in database design.**

Normal Forms ensure the elimination of redundancy and dependency issues in a database by organizing data into well-structured tables.

1. **Explain the concept of RAID in data storage.**

RAID (Redundant Array of Independent Disks) is a storage technology that combines multiple physical drives into a single logical unit to enhance performance, reliability, or both.

1. **Compare and contrast Static and Dynamic Hashing.**

Static Hashing uses a fixed number of buckets, while Dynamic Hashing dynamically adjusts the number of buckets to accommodate data growth.

**UNIT IV: TRANSACTIONS**

1. **Define Concurrent Execution in transactions.**

Concurrent Execution refers to multiple transactions being executed simultaneously. It is essential for improving system efficiency but requires careful management to ensure data consistency.

1. **Explain the concept of Serializability.**

Serializability ensures that the execution of multiple transactions produces a result equivalent to some sequential order, maintaining data integrity.

1. **Discuss the Lock-based Protocol for Concurrency Control.**

Lock-based protocols use locks to control access to data, preventing conflicts and ensuring that transactions are executed in a coordinated manner.

1. **Describe the Timestamp-based Protocol.**

The Timestamp-based Protocol uses timestamps to order transactions, ensuring consistency and preventing conflicts in a concurrent environment.

1. **Explain the process of Log-Based Recovery in a database system.**

Log-Based Recovery involves maintaining a transaction log that records changes to the database. In case of a failure, the log is used to restore the database to a consistent state.

1. **Discuss the advantages and disadvantages of Shadow Paging.**

Shadow Paging is a recovery technique that involves creating a shadow or duplicate of the entire database. While simple, it can be resource-intensive.

**UNIT V: CASE STUDY**

1. **What is a NoSQL database?**

A NoSQL database is a type of database that provides a mechanism for storage and retrieval of data that is modeled in ways other than the tabular relations used in relational databases.

1. **Explain the concept of a Document Database using MongoDB.**

A Document Database stores and retrieves data in a semi-structured format, typically using JSON-like documents. MongoDB is an example of a Document Database.

1. **Discuss the characteristics of multi-dimensional databases, citing Cassandra as an example.**

Multi-dimensional databases organize data into cubes, allowing for efficient analysis across multiple dimensions. Cassandra is an example of a NoSQL database that supports multi-dimensional data models.

1. **Explain the importance of indexing in database systems.**

Indexing enhances database performance by providing a quick and efficient way to locate specific rows or columns within a table.

1. **Compare and contrast the features of MongoDB and Cassandra.**

MongoDB is a Document Database, while Cassandra is a wide-column store NoSQL database. MongoDB uses a flexible schema, whereas Cassandra offers high scalability and fault tolerance.

1. **Discuss the relevance of NoSQL databases in modern data management.**

NoSQL databases are relevant in modern data management due to their ability to handle large volumes of unstructured or semi-structured data, provide high scalability, and support flexible data models.