Unit 1: Introduction to Database Management Systems (DBMS)

The first unit explores the fundamental need for Database Management Systems (DBMS) by emphasizing the limitations of traditional file-based systems, such as data redundancy, inconsistency, and security issues. It highlights how DBMS addresses these challenges through features like data independence, integrity, and improved data sharing. The evolution of DBMS is discussed, tracing the journey from hierarchical and network models to modern relational and NoSQL databases. Additionally, this unit delves into database system architecture, covering the internal, conceptual, and external levels of data abstraction and their relevance to data independence. Students will also learn about the database design process, which involves requirement analysis, conceptual design, logical design, and physical design, underscoring the importance of systematic approaches to ensure scalability, performance, and security in database applications.

Unit 2: Data Modeling and Database Design

This unit introduces the Entity-Relationship (ER) model as a tool for conceptual data modeling, covering entities, attributes, and relationships. It discusses different types of entities and relationships, as well as converting ER diagrams into relational schemas. The Extended ER model builds on this by incorporating advanced concepts such as generalization, specialization, and aggregation, allowing for the representation of complex data scenarios. The relational model, which forms the backbone of modern databases, is introduced along with its structure and principles. The unit also explains Codd's 12 rules that define relational database systems. A detailed discussion on normalization follows, emphasizing its necessity in reducing data redundancy and improving database efficiency. Topics include functional dependencies, inference rules, closure, minimal cover, decomposition properties, and various normal forms (1NF, 2NF, 3NF, BCNF, and 4NF). The unit concludes with an examination of relational synthesis algorithms, which ensure optimal database design.

Unit 3: Relational Algebra, SQL, and PL/SQL

The third unit covers relational algebra as the theoretical foundation of relational databases, focusing on operations such as selection, projection, union, intersection, and joins. SQL is introduced as the standard query language for relational databases, with a focus on Data Definition Language (DDL) and Data Manipulation Language (DML) commands. Students learn to write queries involving set, string, date, and numerical functions, as well as aggregate functions and clauses like GROUP BY and HAVING. Advanced query techniques such as joins, nested queries, and subqueries are also discussed. The unit extends into Data Control Language (DCL) and Transaction Control Language (TCL), emphasizing their role in managing database security and consistency. Finally, the unit introduces PL/SQL for procedural programming within databases, including stored procedures, functions, and triggers. Mapping relational algebra operations to SQL queries is also covered to solidify the theoretical and practical connection.

Unit 4: Storage, Query Optimization, and Transaction Management

This unit focuses on the internal mechanics of databases, starting with storage and file structures, and exploring indexed files, including single-level and multi-level indexing for efficient data retrieval. Query processing and optimization techniques are discussed, highlighting how databases execute and optimize queries to improve performance. The unit also introduces the Parquet file format, which is widely used for efficient storage and querying in big data applications. Transaction management is a critical topic, encompassing the concept of transactions, their states, and ACID properties (Atomicity, Consistency, Isolation, Durability). The transaction state diagram is introduced, along with scheduling concepts and types of serializability, including conflict and view serializability. Various concurrency control protocols and recovery techniques are also discussed to ensure database consistency and fault tolerance.

Unit 5: Advanced Database Concepts: Parallel and Distributed Databases

This unit delves into advanced database architectures, starting with parallel databases. It discusses the architecture of parallel databases and explores various types of parallelism, including I/O parallelism, interquery, intraquery, intraoperation, and interoperation parallelism, emphasizing their role in enhancing database performance. Distributed databases are then introduced, covering types of distributed database systems and their architectures. Topics include distributed data storage strategies, query processing in distributed environments, and the challenges of maintaining consistency across distributed systems. An introduction to ElasticSearch indexes is provided, highlighting their use in distributed search and analytics applications.

Unit 6: NoSQL Databases

The final unit introduces NoSQL databases, discussing their emergence as an alternative to relational databases for handling unstructured and semi-structured data. Various types of NoSQL databases are explored, including key-value stores, document-oriented databases, column-family stores, and graph databases. The unit emphasizes the suitability of NoSQL systems for modern applications requiring scalability, flexibility, and high-performance data handling. Concepts such as eventual consistency, CAP theorem, and the trade-offs between consistency, availability, and partition tolerance are discussed. The unit concludes with a comparison between relational and NoSQL databases, helping students understand when and how to use each effectively