PSN COLLEGE OF ENGINEERING AND TECHNOLOGY (Autonomous),
Melathediyoor, Tirunelveli – 627152

END SEMESTER QUESTION BANK / AY (2023-2024) / ODD

Branch: B.E CSE	Department of Computer Science and Engineering	Year / Semester : III / V
Course Code: CS606604	Course Name: **DATABASE DESIGN (NPTEL)**	
Max. Marks: 100 Marks	Session:	Regulation: 2022

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

CO1:	Understand the basic concepts of Web development
CO2:	Design a HTML, CSS and JavaScript frameworks in web front end development.
CO3:	Develop the backend development of an application.
CO4:	Analyze the concept of testing frameworks and web hosting.
CO5:	Evaluating the different techniques of web development and its application

BL – *Bloom's Level (1- Remembering, 2- Understanding, 3 – Applying, 4 – Analysing, 5 – Evaluating, 6 - Creating)*;
CO – *Course Outcome*;

UNIT – 1

Q. No.	PART-A	Marks	CO	BL
1	Define databases in the context of information systems. **Answer:** Databases are organized collections of data, typically stored electronically in a computer system, that are accessible in various ways.	2	1	1
2	Why is understanding database system architecture important?	2	2	1

	Answer: Understanding database system architecture is crucial for comprehending how data is organized, stored, and accessed within a database management system (DBMS).			
3	Provide an example of a usage context for databases and information systems. **Answer:** A usage context could be an online shopping website where databases are used to store product information, customer details, and transaction records.	2	1	2
4	What are some key components of database system architecture? **Answer:** Key components include the database itself, the DBMS software, users, applications, and hardware infrastructure.	2	1	1
5	Define DBMS and its role in database systems. **Answer:** A Database Management System (DBMS) is software that facilitates the creation, maintenance, and manipulation of databases. It acts as an interface between the users/applications and the database.	2	2	1
6	Why are databases and information systems essential in today's digital age? **Answer:** Databases and information systems enable efficient storage, retrieval, and management of vast amounts of data, supporting various applications and decision-making processes in organizations.	2	1	4
7	What distinguishes databases from traditional file systems? **Answer:** Databases offer data independence, support for concurrent access, data integrity through enforced constraints, and enhanced data retrieval capabilities compared to traditional file systems.	2	1	1
8	Write the concept of data independence in the context of databases.	2	1	1

	Answer: Data independence refers to the separation of the logical structure of the database from its physical storage, allowing changes to the physical storage without affecting the application's view of the data.			
9	What is the importance of understanding database system concepts and architecture for aspiring database professionals? **Answer:** Understanding database system concepts and architecture provides the foundational knowledge necessary for designing, implementing, and maintaining efficient database systems, which are essential skills for database professionals.	2	2	1
10	Define an example usage context for database systems in the context of a social media platform. **Answer:** In a social media platform like Facebook, databases are used to store user profiles, posts, comments, likes, and other social interactions, facilitating efficient retrieval and management of vast amounts of user-generated content.	2	1	1
11	How do databases contribute to data consistency and integrity within an organization? **Answer:** Databases enforce data consistency and integrity through the implementation of data constraints, such as primary keys, foreign keys, and check constraints, ensuring that data remains accurate and reliable even with concurrent access and updates from multiple users.	2	1	1
12	Define high-level conceptual modeling in the context of database design. **Answer:** High-level conceptual modeling involves abstracting the essential elements and relationships of a system to create a conceptual representation that serves as the basis for database design.	2	2	1
13	Explain the purpose of ER modeling in database design.	2	1	2

	Answer: ER modeling is used to visually represent the entities, attributes, and relationships within a database system, providing a clear and concise way to conceptualize the database schema.			
14	What are ER diagrams, and how are they used in database design? **Answer:** ER diagrams are graphical representations of the entities, attributes, and relationships in a database system. They help visualize the database schema and communicate its structure to stakeholders.	2	1	1
15	Define cardinality constraints in ER modeling. **Answer:** Cardinality constraints specify the number of occurrences of one entity that are associated with a single occurrence of another entity in a relationship.	2	2	1
16	What is a weak-entity type in ER modeling? **Answer:** A weak-entity type is an entity that cannot be uniquely identified by its attributes alone and relies on a relationship with another entity, known as its owner entity, for identification.	2	1	1
17	Explain the concept of subclasses and inheritance in ER modeling. **Answer:** Subclasses and inheritance allow entities to inherit attributes and relationships from a parent entity, facilitating the modeling of specialization and generalization relationships.	2	1	1
18	What is the Enhanced ER Model (EER), and how does it extend traditional ER modeling? **Answer:** The Enhanced ER Model (EER) extends traditional ER modeling by incorporating additional constructs such as subclasses, inheritance, and union types to represent more complex data relationships.	2	1	2
19	Define specialization and generalization in the context of ER modeling. **Answer:** Specialization refers to the process of defining subclasses based on shared characteristics of entities, while generalization involves grouping similar entities into a higher-level entity.	2	1	1

20	How is the modeling of UNION types achieved using categories in ER modeling? **Answer:** Union types, which represent entities that may belong to multiple entity types simultaneously, can be modeled using categories to define common attributes and relationships shared by the participating entity types.	2	1	1
PART-B				
1	Discuss the role of databases in information systems, highlighting their significance in modern computing environments	13	1	2
2	Explain the process of high-level conceptual modeling in database design, emphasizing the importance of accurately capturing the semantics of the real-world domain	13	1	2
3	Discuss cardinality constraints in ER modeling and their significance in defining the nature of relationships between entities.	13	2	4
4	Explore the concepts of enhanced Entity-Relationship (EER) modeling, focusing on advanced features such as weak-entity types, subclasses, and inheritance.	13	1	4
5	Define specialization and generalization in the context of database design, highlighting their role in organizing entities into hierarchies based on shared characteristics.	13	1	1
6	Explain the concept of UNION types in database modeling and how they can be represented using categories.	13	2	1
7	Compare and contrast different data modeling approaches, including relational, object-oriented, and NoSQL databases.	13	1	2
8	Explain the concept of normalization in database design, focusing on the principles of minimizing redundancy and dependency among data attributes.	13	1	3

PART -C				
1	Apply the principles of semantic database design to a real-world scenario or case study. Describe the process of conceptual modeling, entity identification, relationship definition, and schema refinement based on the specific requirements of the application domain. Discuss the challenges and considerations involved in translating conceptual models into physical database schemas.	15	1	1
2	Discuss the integration of semantic database design principles with software development methodologies such as agile and iterative development. Explain how the conceptual modeling phase aligns with requirements gathering and analysis in software engineering.	15	2	2
UNIT – II				
Q. No.	PART – A	Marks	CO	BL
1	Define relational algebra and its role in database management. **Answer:** Relational algebra is a theoretical framework used to define operations for manipulating relational data, such as selection, projection, join, and set operations. It serves as the foundation for query languages and database query optimization techniques.	2	2	2
2	What are relational integrity constraints, and why are they important in database systems? **Answer:** Relational integrity constraints enforce rules and relationships between data elements in a relational database, ensuring data consistency and integrity. They include primary key constraints, foreign key constraints, and other domain constraints.	2	2	1
3	Write the basic structure of SQL (Structured Query Language). **Answer:** SQL consists of data definition language (DDL) statements for defining database schema, data manipulation language (DML) statements for querying and modifying data, data control language (DCL) statements for managing access permissions, and transaction control statements.	2	1	1

4	What are views in SQL, and how are they used in database systems? **Answer:** Views are virtual tables derived from one or more base tables or other views. They provide a way to present a subset of data or a transformed view of the underlying tables without modifying the original data.	2	2	2
5	Write the concept of integrity constraints in SQL. **Answer:** Integrity constraints in SQL enforce rules or conditions that data must satisfy to maintain data consistency and integrity, such as primary key constraints, foreign key constraints, unique constraints, and check constraints.	2	1	1
6	What is embedded SQL, and how does it differ from standard SQL? **Answer:** Embedded SQL refers to SQL statements embedded within a host programming language (e.g., C, Java) to interact with a relational database. It allows database operations to be integrated directly into application code for improved performance and flexibility.	2	2	1
7	How does ER (Entity-Relationship) modeling facilitate the mapping of database schemas to relational schemas? **Answer:** ER modeling provides a high-level conceptual representation of data entities, attributes, and relationships, which can be mapped to relational schemas using mapping rules and techniques to convert ER diagrams into relational tables.	2	2	1
8	What are some key features of the IBM DB2 database system? **Answer:** IBM DB2 is a relational database management system (RDBMS) known for its scalability, reliability, and support for various data types and platforms. It offers advanced features such as online transaction processing (OLTP), data warehousing, and analytics capabilities.	2	2	2
9	What is the relational algebra, and how does it relate to the relational model?	2	2	1

	Answer: Relational algebra is a mathematical framework for manipulating relational data. It consists of operations like selection, projection, join, and set operations, which are used to query and manipulate data in relational databases. Relational algebra forms the theoretical foundation of the relational model.			
10	Write the process of ER (Entity-Relationship) to relational mapping. **Answer:** ER to relational mapping involves translating the entities, attributes, and relationships represented in an ER diagram into relational schemas (tables). Each entity becomes a table, and each attribute becomes a column, while relationships are represented using foreign keys.	2	1	2
11	What is the significance of specifying indexes in SQL? **Answer:** Indexes in SQL are used to improve the performance of database queries by providing faster access to data. They allow for efficient data retrieval by organizing data in a structured manner and facilitating quick lookup operations.	2	2	2
12	Write the concept of primary indexes in indexing methods. **Answer:** Primary indexes are indexes created on the primary key column(s) of a table. They provide a unique identifier for each row and are used to enforce the primary key constraint. Primary indexes help optimize data retrieval operations and enforce data integrity.	2	2	2
13	What are the different types of indexing methods commonly used in database systems? **Answer:** Common indexing methods include primary indexes, clustering indexes, secondary indexes, multilevel indexes, ISAM (Indexed Sequential Access Method), B-trees, and B+ trees.	2	1	2

14	How do multilevel indexes improve data access efficiency in database systems? **Answer:** Multilevel indexes organize index entries into a hierarchical structure, allowing for faster data access by reducing the number of disk accesses required to locate data. They provide an efficient balance between index size and search performance.	2	2	1
15	Write the role of buffering of blocks in storage structures. **Answer:** Buffering of blocks involves caching frequently accessed data blocks in memory to reduce disk I/O operations. It improves data access performance by minimizing disk latency and optimizing data retrieval operations.	2	2	2
16	Define the term "relational integrity constraints" in the context of SQL. **Answer:** Relational integrity constraints are rules that enforce data consistency and integrity in a relational database. They include primary key constraints, foreign key constraints, unique constraints, and check constraints.	2	1	2
17	How does SQL support data definition and manipulation operations? **Answer:** SQL provides data definition language (DDL) statements for defining database schema objects such as tables, indexes, and views. It also offers data manipulation language (DML) statements for querying and modifying data stored in tables.	2	2	1
18	What is the purpose of specifying indexes in SQL queries? **Answer:** Specifying indexes in SQL queries allows database administrators to optimize query performance by instructing the database management system to utilize specific indexes for data retrieval, sorting, and join operations.	2	2	2

19	What is the architecture of IBM DB2, and how does it contribute to its performance and scalability? **Answer:** IBM DB2 follows a client-server architecture, where the database server manages data storage and processing, while clients interact with the server to perform database operations. This architecture enables distributed data access and supports scalable deployment configurations, contributing to DB2's performance and scalability.	2	2	2
20	Write the concept of embedded SQL and its advantages in database application development. **Answer:** Embedded SQL allows SQL statements to be embedded directly within application code written in programming languages like C or Java. This integration facilitates seamless interaction between the application and the database, reducing network overhead and improving performance compared to executing SQL statements separately. Additionally, embedded SQL enables developers to leverage the full power and flexibility of SQL within their application logic.	2	1	
	PART-B			
1	Explain the core concepts of the relational model and relational algebra.	13	2	1
2	Define relational integrity constraints such as primary key, foreign key, unique, and check constraints. Discuss their significance in maintaining data consistency and integrity within a relational database.	13	2	3
3	Describe the syntax and semantics of SQL (Structured Query Language) for data definition and manipulation.	13	2	3
4	Discuss the concept of database views in SQL and their role in simplifying complex queries and enhancing data security.	13	2	4

5	Define embedded SQL and explain how it is used to incorporate SQL statements into host programming languages such as C, C++, Java, and COBOL	13	1	1
6	Explore the architecture of IBM DB2, focusing on its components and layers.	13	2	3
7	Explain the process of mapping an Enhanced Entity-Relationship (EER) model to a relational database schema.	13	2	3
8	Discuss the relationship between relational algebra and SQL query optimization. Explain how relational algebra expressions are translated into optimized query execution plans in a relational database management system (RDBMS).	13	1	3
	PART-C			
1	Provide guidelines and best practices for effective EER to relational mapping.	15	1	3
2	Explore advanced SQL features such as window functions, common table expressions (CTEs), recursive queries, and pivot/unpivot operations. Discuss how these features are used in data warehousing scenarios to analyze large datasets and generate meaningful insights.	15	2	4
	UNIT – III			
Q. No.	PART-A	Marks	CO	BL
1	Define functional dependencies in the context of database design. **Answer:** Functional dependencies represent the relationships between attributes in a relation, where the value of one attribute uniquely determines the value of another attribute.	2	3	4
2	How are keys defined in a relational model? **Answer:** Keys are attributes or combinations of attributes that uniquely identify each tuple (row) in a relation. Primary keys are the minimal set of attributes that uniquely identify tuples, while candidate keys are alternative keys that also satisfy uniqueness constraints.	2	3	1

3	What are the key objectives of normalization in database design? **Answer:** The key objectives of normalization are to eliminate data redundancy, reduce update anomalies, and ensure data integrity by organizing data into well-structured relations with minimal redundancy and dependencies.	2	1	4
4	Write the concept of Boyce-Codd Normal Form (BCNF). **Answer:** Boyce-Codd Normal Form (BCNF) is a higher normal form that ensures that every non-trivial functional dependency in a relation is a dependency on a superkey. It eliminates anomalies related to non-trivial functional dependencies.	2	2	1
5	Define multi-valued dependencies and their relevance in database design. **Answer:** Multi-valued dependencies occur when the values of one set of attributes determine the values of another set of attributes, independently of the primary key. They are relevant for identifying certain types of data redundancy and anomalies in relations.	2	3	2
6	What are join dependencies, and how do they impact database normalization? **Answer:** Join dependencies occur when a non-trivial dependency exists between attributes in different relations. They can complicate database design and may require additional normalization steps to eliminate redundancy and maintain data integrity.	2	3	1
7	Write the role of inclusion dependencies in database design. **Answer:** Inclusion dependencies specify that the values appearing in one set of attributes must also appear in another set of attributes. They are essential for ensuring data consistency and integrity across related relations.	2	3	1

8	How does normalization help in reducing data redundancy in database design? **Answer:** Normalization decomposes relations into smaller, well-structured relations to eliminate data redundancy by storing data in a non-repetitive manner and minimizing the storage of duplicate information.	2	3	1
9	Write the process of achieving Boyce-Codd Normal Form (BCNF) in database normalization. **Answer:** To achieve BCNF, you identify and eliminate partial dependencies by decomposing relations into smaller relations until each relation is in BCNF, ensuring that every non-trivial functional dependency is a dependency on a superkey.	2	3	4
10	What is the significance of further dependencies in database normalization? **Answer:** Further dependencies, such as join dependencies and multi-valued dependencies, highlight potential data anomalies and redundancies that may exist even after achieving higher normal forms. Addressing further dependencies is crucial for ensuring data integrity and minimizing anomalies.	2	3	4
11	Write the concept of third normal form (3NF) and its importance in database design. **Answer:** Third normal form (3NF) eliminates transitive dependencies by ensuring that every non-prime attribute is non-transitively dependent on every candidate key. It helps in reducing data redundancy and update anomalies, thereby improving data integrity.	2	3	1
12	What is the role of functional dependencies in determining the normal form of a relation? **Answer:** Functional dependencies provide the basis for determining the normal form of a relation by identifying dependencies between attributes. Ensuring that relations satisfy certain dependency	2	3	1

	constraints helps in achieving higher normal forms and optimizing database design.			
13	Write the concept of a superkey in the context of the relational model. **Answer:** A superkey is a set of one or more attributes that uniquely identifies each tuple (row) in a relation. It may contain more attributes than necessary to uniquely identify tuples, and a minimal subset of attributes within a superkey is called a candidate key.	2	3	2
14	How does normalization contribute to database design optimization? **Answer:** Normalization optimizes database design by reducing data redundancy, eliminating update anomalies, and ensuring data integrity. It helps in organizing data efficiently and maintaining consistency across relations.	2	3	2
15	Define the concept of dependency preservation in database normalization. **Answer:** Dependency preservation ensures that functional dependencies present in the original relation are preserved after decomposition into smaller relations during the normalization process. It ensures that the resulting relations maintain the same functional dependencies as the original relation.	2	3	2
16	Write the concept of partial dependency and its implications in database normalization. **Answer:** Partial dependency occurs when an attribute is functionally dependent on only a part of the candidate key. It can lead to data redundancy and update anomalies, which are addressed by decomposing the relation into smaller relations during normalization.	2	2	2
17	What is the difference between a candidate key and a primary key in the context of the relational model?	2	1	2

	Answer: A candidate key is a set of attributes that uniquely identifies each tuple in a relation, while a primary key is a chosen candidate key that serves as the main identifier for the relation. There can be multiple candidate keys in a relation, but only one primary key.			
18	Write the role of normalization in ensuring data consistency in database design. **Answer:** Normalization helps in organizing data into well-structured relations with minimal redundancy and dependencies, thereby ensuring data consistency by eliminating anomalies such as update, insertion, and deletion anomalies.	2	1	1
19	How does achieving higher normal forms, such as Boyce-Codd Normal Form (BCNF) and Fourth Normal Form (4NF), contribute to database design optimization? **Answer:** Achieving higher normal forms reduces data redundancy and minimizes the occurrence of update anomalies by ensuring that relations are well-structured and free from certain types of dependencies. It optimizes database design by improving data integrity and consistency.	2	2	2
20	What is the primary objective of normalization in database design? **Answer:** The primary objective of normalization is to eliminate data redundancy and minimize update anomalies by organizing data into well-structured relations with minimal redundancy and dependencies.	2	1	1
	PART – B			
1	Define keys in the relational model and discuss their importance in database design. Explain the concept of functional dependencies and how they capture relationships between attributes in a relation.	13	3	4

2	Describe the various normal forms based on functional dependencies, including First Normal Form (1NF), Second Normal Form (2NF), and Third Normal Form (3NF).	13	3	2
3	Define Boyce-Codd Normal Form (BCNF) and discuss its significance in relational database design	13	3	4
4	Explain multi-valued dependencies (MVDs) and their relevance in database design. Discuss Fourth Normal Form (4NF) as a normalization criterion that addresses anomalies related to multi-valued dependencies.	13	3	2
5	Define join dependencies and their role in database normalization. Discuss Fifth Normal Form (5NF) as a normalization criterion that addresses anomalies related to join dependencies.	13	3	2
6	Explain inclusion dependencies and how they represent relationships between attributes across different relations.	13	3	2
7	Explore advanced dependency concepts beyond Fifth Normal Form (5NF), such as domain-key normal form (DK/NF) and sixth normal form (6NF).	13	3	3
8	Provide examples of inclusion dependencies in real-world scenarios and discuss strategies for handling them.	13	3	3
PART-C				
1	Describe the database requirements, including data sources, relationships, and constraints. Walk through the process of normalizing the initial database schema into higher normal forms, addressing functional and multi-valued dependencies.	15	1	2
2	Explore the challenges and considerations involved in dependency analysis and normalization for large-scale database systems. Discuss strategies for identifying and managing dependencies across distributed and heterogeneous data sources.	15	2	3

UNIT – IV

Q. No.	PART-A	Marks	CO	BL
1	**What are secondary storage devices?** **Answer**: Secondary storage devices are non-volatile memory devices used for long-term data storage, such as hard disk drives (HDDs), solid-state drives (SSDs), optical discs, and magnetic tapes.	2	2	1
2	**Define buffering of blocks in the context of storage structures.**	2	2	1

	Answer: Buffering of blocks involves temporarily storing data blocks from secondary storage into main memory (buffer) to reduce the number of disk accesses and improve performance.			
3	**What is file organization?** **Answer**: File organization refers to the arrangement of records in a file on secondary storage, which impacts how efficiently data can be accessed and manipulated.	2	1	2
4	**Write the concept of heaps in storage structures.** **Answer**: Heaps are a type of file organization where records are stored in a heap without any specific order. Insertion is performed by appending the record at the end of the file, making it suitable for applications where sequential access is common.	2	2	1
5	**What are sorted files?** **Answer**: Sorted files are organized in ascending or descending order based on a key field. This organization facilitates efficient searching and retrieval operations, especially binary search.	2	1	4
6	**What is hashing in the context of storage structures?** **Answer**: Hashing is a technique used to map data items to locations in a hash table based on their keys. It enables fast retrieval of records by directly accessing the location calculated using a hash function.	2	2	1
7	**How are overflow handling techniques used in hashing?** **Answer:** Overflow handling techniques in hashing involve strategies for dealing with collisions, where multiple keys map to the same location in the hash table. Techniques include chaining, open addressing, and rehashing.	2	2	2

8	**Define dynamic hashing.** **Answer**: Dynamic hashing is a hashing technique where the size of the hash table dynamically grows or shrinks as needed to accommodate the number of keys being stored, helping to maintain a balanced distribution of keys and efficient access.	2	2	1
9	**What is extensible hashing?** **Answer**: Extensible hashing is a dynamic hashing technique where the hash table size remains fixed, but buckets are dynamically split or merged to accommodate additional keys, improving efficiency and reducing collision rates.	2	2	1
10	**Name some other file organizations besides heaps and sorted files.** **Answer**: Some other file organizations include sequential files, indexed sequential access method (ISAM), and tree-based structures like B-trees and B+ trees.	2	1	2
11	**Define primary indexes.** **Answer**: Primary indexes are indexes created on the primary key of a file, enabling direct access to records based on their primary key values. They are often implemented using ordered arrays or hash tables.	2	2	4
12	**What is a clustering index?** **Answer**: A clustering index is an index organized on a non-primary key field that arranges records physically on disk in the same order as the index, thereby improving the efficiency of range queries.	2	2	1
13	**Define secondary index.**	2	1	2

	Answer: A secondary index is an index created on a non-primary key field, allowing efficient access to records based on values of that field. It provides alternate access paths to the data.			
14	**What are multilevel indexes?** **Answer**: Multilevel indexes are hierarchical indexes that use multiple levels of indexing to facilitate efficient access to large datasets, reducing the number of disk accesses required for retrieval.	2	2	1
15	**Define ISAM (Indexed Sequential Access Method).** **Answer**: ISAM is a file organization method that combines sequential and indexed access. It uses a primary index to access data sequentially within index ranges, enhancing both sequential and random access efficiency.	2	2	2
16	**What are B-trees?** **Answer**: B-trees are balanced tree data structures used for indexing in databases. They maintain sorted data and allow for efficient insertion, deletion, and search operations by keeping the tree height balanced.	2	1	2
17	**Write short notes about B+ trees.** **Answer**: B+ trees are a variant of B-trees where only data pointers are stored in the leaf nodes, while internal nodes store only keys. This design optimizes for sequential access and range queries.	2	2	4
18	**How are inserting algorithms for B+ trees implemented?** **Answer**: Insertion in B+ trees involves traversing the tree from the root to the leaf node where the new key should be inserted. If the leaf node is full, it may split, and the split may propagate upwards if necessary to maintain balance.	2	2	1

19	**Write about searching algorithms for B+ trees.** **Answer**: Searching in B+ trees starts at the root and recursively follows the appropriate child pointers based on the search key until reaching a leaf node. In leaf nodes, a binary search or linear search is performed to find the desired key.	2	2	2
20	**Name some other indexing methods used in databases.** **Answer**: Other indexing methods include bitmap indexing, sparse indexing, cover-based indexing, and tree-based structures like R-trees and Quad-trees.	2	1	1
	PART-B			
1	Provide a comprehensive overview of storage structures used in database systems. Discuss the characteristics of secondary storage devices and their impact on database performance.	13	4	2
2	Explain the concept of hashing and its role in data storage and retrieval. Discuss different hashing techniques, including static hashing, dynamic hashing, and extensible hashing.	13	4	2
3	Explore dynamic hashing as a technique for dynamically resizing hash tables to accommodate changes in data volume. Discuss the advantages of dynamic hashing over static hashing and its ability to adapt to varying workloads.	13	4	4
4	Compare and contrast heap and sorted file organizations in database storage	13	4	2
5	Define basic terminology related to indexing in database systems, including keys, indexes, and search keys.	13	4	1
6	Discuss the concept of secondary indexes and their use in providing alternate access paths to data records based on non-primary key attributes	13	4	2
7	Discuss the process of traversing B+ tree nodes and performing key comparisons to locate data records. Describe insertion algorithms for maintaining the balance of B+ trees and ensuring optimal tree structure.	13	4	4

8	Illustrate searching algorithms for efficiently locating data records within B+ tree indexes. Provide examples to demonstrate the performance characteristics of B+ tree insertion and search operations.	13	4	2
	PART-C			
1	Provide an in-depth analysis of B-trees and B+ trees as commonly used index structures in database systems.	15	1	2
2	Discuss how storage and indexing structures are integrated into modern database management systems (DBMS). Explain the role of the storage manager in managing data storage and retrieval operations efficiently.	15	2	4
	UNIT - V			
Q. No.	PART-A	Marks	CO	BL
1	**What are OLTP environments?** **Answer**: OLTP (Online Transaction Processing) environments are systems designed to process a large number of short online transactions. They emphasize quick response times and efficient handling of concurrent transactions.	2	2	2
2	**What are concurrency issues in transaction processing?** **Answer:** Concurrency issues arise when multiple transactions access and modify shared data simultaneously, potentially leading to problems such as data inconsistency, lost updates, and conflicts.	2	2	2
3	**Why are transactions necessary in database systems?** **Answer:** Transactions ensure data integrity and consistency by grouping database operations into logical units. They provide atomicity, consistency, isolation, and durability (ACID properties), which are essential for reliable data management.	2	1	1
4	**List the ACID properties of transactions.** **Answer:** The ACID properties of transactions are: o Atomicity: Transactions are treated as indivisible units of work.	2	2	3

	o Consistency: Transactions transform the database from one consistent state to another. o Isolation: Transactions operate independently of each other, as if they were executed sequentially. o Durability: Once a transaction is committed, its effects are permanent and survive system failures.			
5	**Define transaction states in database systems.** **Answer**: Transaction states include active, partially committed, committed, and aborted. A transaction progresses through these states during its execution, reflecting its progress and outcome.	2	1	1
6	**Write about serializability in the context of transactions.** **Answer**: Serializability ensures that the execution of concurrent transactions produces the same result as if they were executed serially, one after the other. It guarantees consistency and avoids interference among transactions.	2	2	2
7	**What are serial schedules in transaction processing?** **Answer**: Serial schedules are sequences of transactions where each transaction executes in its entirety before the next transaction begins. They represent a serial execution order, ensuring consistency but potentially sacrificing concurrency.	2	2	1
8	**Differentiate between conflict serializability and view serializability.** **Answer**: Conflict serializability ensures that conflicting operations (e.g., read/write conflicts) are executed in a consistent order across all serializable schedules. View serializability ensures that transactions produce the same results when their read and write operations are interleaved.	2	2	2
9	**Define recoverable schedules and non-recoverable schedules.** **Answer**: A recoverable schedule is one in which, if a transaction commits, all its updates are permanent, and if it aborts, none of its updates are	2	2	1

	permanent. A non-recoverable schedule allows committed updates to be overwritten by later aborted transactions.			
10	**What is a cascading rollback in transaction processing?** **Answer**: A cascading rollback occurs when a transaction aborts, causing all subsequent transactions that have read values written by the aborted transaction to also abort, leading to a cascade effect of rollbacks.	2	1	4
11	**Define cascadeless schedules.** **Answer**: Cascadeless schedules prevent cascading rollbacks by ensuring that a transaction cannot read values written by another transaction that has not yet committed. This property helps maintain data consistency and prevents ripple effects of aborts.	2	2	1
12	**Write the concept of concurrency control in database systems.** **Answer**: Concurrency control mechanisms manage simultaneous access to shared data by multiple transactions to prevent conflicts, maintain data consistency, and ensure the correctness of transaction execution in a concurrent environment.	2	2	4
13	**What is locking in concurrency control?** **Answer**: Locking is a concurrency control technique where transactions acquire locks on data items to control access and ensure serializability. Locks prevent other transactions from accessing or modifying data until the lock is released.	2	1	1
14	**Define deadlock in the context of concurrency control.** **Answer:** Deadlock occurs when two or more transactions are waiting indefinitely for resources held by each other, preventing progress. Deadlocks can arise in systems using locking-based concurrency control mechanisms.	2	2	4
15	**Write two-phase locking (2PL) in concurrency control.**	2	2	3

	Answer: Two-phase locking is a protocol where transactions acquire all the locks they need before starting execution (growing phase) and release all locks only after completing execution (shrinking phase). This protocol ensures conflict serializability and prevents deadlocks.			
16	**What is timestamp-based concurrency control?** **Answer**: Timestamp-based concurrency control assigns a unique timestamp to each transaction, allowing the system to determine the relative order of transactions and resolve conflicts based on their timestamps. It ensures serializability and prevents deadlock.	2	1	1
17	**Define optimistic concurrency control.** **Answer**: Optimistic concurrency control allows transactions to execute without acquiring locks initially. Conflicts are detected at the time of transaction commit, and if conflicts occur, appropriate actions such as aborting and retrying are taken to ensure consistency.	2	2	4
18	**What is multiversion concurrency control (MVCC)?** **Answer**: Multiversion concurrency control maintains multiple versions of data items to support concurrent transactions without blocking or waiting. Each transaction sees a consistent snapshot of the database at its start time, improving concurrency and isolation.	2	2	1
19	**Write the concept of deadlock detection and resolution.** **Answer**: Deadlock detection involves periodically checking for the presence of deadlocks in the system. If a deadlock is detected, it can be resolved through techniques such as deadlock prevention, deadlock avoidance, or deadlock recovery (e.g., aborting one of the transactions involved).	2	2	4
20	**Name some common techniques for handling concurrency control in distributed database systems.** **Answer**: Common techniques for concurrency control in distributed database systems include distributed locking, distributed timestamp	2	1	3

	ordering, two-phase commit protocol, and optimistic concurrency control with distributed conflict detection.			
	PART – B			
1	Discuss the fundamental concepts of transaction processing in OLTP (Online Transaction Processing) environments.	13	5	3
2	Describe the necessary properties of transactions, including atomicity, consistency, isolation, and durability (ACID properties).	13	5	2
3	Explore the concurrency issues that arise in multi-user database environments. Discuss common problems such as lost updates, uncommitted data, and inconsistent retrievals	13	5	2
4	Discuss the implications of serializability for concurrency control and transaction management in database systems.	13	5	4
5	Compare and contrast conflict serializability and view serializability as two approaches to ensuring transaction serializability.	13	5	2
6	Describe recoverable and non-recoverable schedules and their significance in transaction management.	13	5	2
7	Discuss the concept of cascading rollbacks and its impact on transaction execution and data consistency	13	5	2
8	Compare and contrast locking-based concurrency control mechanisms with timestamp-based concurrency control mechanisms	13	5	3
	PART – C			
1	Explain the principles of two-phase locking (2PL) and timestamp ordering as concurrency control protocols in database systems. Discuss how two-phase locking ensures conflict serializability by acquiring and releasing locks in two phases: growing phase and shrinking phase.	15	4	2
2	Define isolation levels in the context of transaction processing and concurrency control. Discuss the different levels of isolation, including Read Uncommitted, Read Committed, Repeatable Read, and Serializable, and their corresponding guarantees regarding data visibility and consistency.	15	2	1