DBMS:

1. What is DBMS ? Mention advantages..
2. What is Database?
3. What is a database system?
4. What is RDBMS ? Properties..
5. Types of database languages
6. ACID properties (VVVVV IMP)
7. Difference between vertical and horizontal scaling
8. What is sharding
9. Keys in DBMS
10. Types of relationship
11. Data abstraction in DBMS, three levels of it
12. Indexing in DBMS
13. What is DDL (Data Definition Language)
14. What is DML (Data Manipulation Language)
15. What is normalization ? Types of them ..
16. What is denormalization ?
17. What is functional dependency ?
18. E-R Model ?
19. Conflict Serializability in DBMS ..
20. Explain Normal forms in DBMS
21. What is CCP ? (Concurrency Control Protocols)
22. Entity, Entity Type, Entity Set, Weak Entity Set..
23. What are SQL commands ? Types of them..
24. Nested Queries in SQL ?
25. What is JOIN .. Explain types of JOINs
26. Inner and Outer Join
27. Practice sql queries from leetcode
28. Diff between 2 tier and 3 tier architecture
29. Diff between TRUNCATE and DELETE command ..
30. Difference between Intension and Extension in a DataBase
31. Difference between share lock and exclusive lock, definition of lock

Basic questions:

1. DBMS and EX of dbms ? nosql,mongo db
2. RDBMS? Mysql,oracle
3. MYSQL – used for web development
4. Who owns oracle and mysql?
5. Dbms supported for mobile application? Sqlite
6. What is Sql and its uses?
7. What are sublanguages of sql? DDL,DML,TCL,DCL , select queries (IMP)
8. DDL – create , alter, drop ,truncate
9. DML – insert,delete,update
10. DCL- grant , revoke
11. TCL – savepoint , commit,rollback
12. Integrity Constraints
13. Domain integrity constraints : not null , unique, check , default
14. Entity integrity constraints : primary key
15. Referrencial integrity constraints: foreign key
16. Participation integrity constraints : total participation , partial participation
17. SQL JOINS
18. ORDER BY (in cases for asc/des in sorting) , GROUP BY(in cases of aggregate functions like average,min,max,sum,count,each city,each dept) , HAVING (for condition) , LIMIT [offset,] (used for finding the second maxi,third max,nth max,nth min )
19. Postgresql
20. ACID PROPERTIES – Atomicity, Consistency, Isolation and Durability (IMP)

DATABASE DESIGN:

1. ER MODEL DIAGRAM, ER to RELATIONAL SCHEMA conversion (6 steps)
2. KEYS,FUCTIONAL KEYS – ( primary key/super key/candidate key)
3. FUNCTIONAL DEPENDENCIES (determinant -> dependent)
4. NORMALISATION – 1NF/2NF/3NF/BCNF/4NF/5NF (get to know till bcnf) (IMP)

Purpose of normalization ( to achieve a good db by optimising it )

**1. What is a DBMS?**  
Answer: A DBMS is software that manages databases, providing an interface to store, retrieve, and manipulate data efficiently and securely.

**2. What are the different types of DBMS models?**  
Answer: The main types are:  
Relational DBMS (RDBMS)  
NoSQL DBMS (Document, Key-Value, Columnar, Graph)

**3. Explain ACID properties in DBMS.**  
Answer: ACID stands for Atomicity, Consistency, Isolation, and Durability, which ensure database transactions are reliable and maintain data integrity.

**4. What is normalization in DBMS?**  
Answer: Normalization is the process of organizing data to eliminate redundancy and improve data integrity in a relational database.

**5. What is the difference between a primary key and a foreign key?**  
Answer: A primary key uniquely identifies a record in a table, while a foreign key refers to the primary key of another table, establishing a relationship between the two.

**6. Explain the difference between DELETE and TRUNCATE in SQL.**  
Answer: DELETE removes rows from a table one by one with transaction logging, while TRUNCATE removes all rows at once without transaction logging.

**7. What is the difference between INNER JOIN and OUTER JOIN?**  
Answer: INNER JOIN returns only matching rows from both tables, while OUTER JOIN (LEFT, RIGHT, FULL) returns matching rows as well as non-matching rows from one or both tables.

**8. What is a stored procedure?**  
Answer: A stored procedure is a precompiled set of SQL statements that can be stored and executed on the database server to perform specific tasks.

**9. Explain the difference between a database and a DBMS.**  
Answer: A database is a collection of related data, while a DBMS is software used to manage, store, and retrieve data efficiently from the database.

**10. What is the purpose of an index in a database?**  
Answer: An index improves the speed of data retrieval operations by creating a data structure that allows the database to locate and access data quickly.

**11. What is a deadlock, and how can it be avoided?**  
Answer: A deadlock occurs when two or more transactions are unable to proceed because each is waiting for the other to release a lock. Deadlocks can be avoided by using proper transaction management and lock acquisition strategies.

**12. What is a trigger in DBMS?**  
Answer: A trigger is a set of actions that automatically execute in response to specific database events, such as insert, update, or delete operations on a table.

**13. Explain the concept of data integrity in a DBMS.**  
Answer: Data integrity ensures that data is accurate, consistent, and valid throughout the database, preventing unauthorized or erroneous changes.

**14. What is a self-join in SQL?**  
Answer: A self-join is a SQL query where a table is joined with itself to combine related rows based on a common column.

**15. What is the difference between a heap table and a clustered table?**  
Answer: A heap table does not have a clustered index, while a clustered table stores data physically sorted based on the indexed column.

**16. What are the advantages and disadvantages of denormalization?**  
Answer: Denormalization can improve read performance but may lead to data redundancy and increased complexity in maintaining data consistency.

**17. Explain the concept of a transaction in DBMS.**  
Answer: A transaction is a sequence of one or more database operations that are treated as a single unit, ensuring data consistency and allowing rollback in case of failures.

**18. What is the purpose of the COMMIT statement in SQL?**  
Answer: The COMMIT statement is used to permanently save all changes made during a transaction.

**19. What is the difference between a clustered and non-clustered index?**  
Answer: A clustered index determines the physical order of data rows in a table, whereas a non-clustered index creates a separate structure for quick data access.

**20. What are the advantages of using a NoSQL database over an RDBMS?**  
Answer: NoSQL databases offer flexibility, scalability, and better performance for unstructured or semi-structured data compared to traditional RDBMS.

**21. Explain the CAP theorem in the context of distributed databases.**  
Answer: The CAP theorem states that it is impossible for a distributed database to simultaneously achieve Consistency, Availability, and Partition tolerance. It can satisfy at most two out of the three.

**22. What are triggers in the context of database events?**  
Answer: Triggers are special stored procedures that are automatically executed in response to specific data manipulation events, like INSERT, UPDATE, or DELETE operations.

**23. Explain the difference between optimistic and pessimistic locking.**  
Answer: Optimistic locking assumes no conflicts between transactions until the end, checking for conflicts only during the commit. Pessimistic locking acquires locks immediately, assuming conflicts might occur.

**24. What is a materialized view in DBMS?**  
Answer: A materialized view is a database object that stores the result of a query, allowing faster data retrieval since the results are precomputed and stored.

**25. What is a B-tree index?**  
Answer: A B-tree index is a self-balancing tree data structure used to speed up data retrieval by maintaining a sorted hierarchy of values.

**26. Explain the concept of sharding in database systems.**  
Answer: Sharding involves splitting a large database into smaller, more manageable pieces (shards) to distribute the data across multiple servers or nodes for improved scalability and performance.

**27. What is an Entity-Relationship Diagram (ERD)?**  
Answer: An ERD is a visual representation of the relationships among entities in a database, showing how different tables are connected.

**28. How can you optimize the performance of a SQL query?**  
Answer: Query performance can be improved by adding indexes, optimizing table structures, and rewriting complex queries.

**29. What is a transaction log in a database, and why is it essential?**  
Answer: The transaction log records all changes made to a database and is crucial for recovery in case of system failures or crashes.

**30. Explain the concept of referential integrity in a relational database.**  
Answer: Referential integrity ensures that relationships between tables are maintained by enforcing valid foreign key constraints, preventing the creation of orphaned records.

Bonus Questions:

1. Explain the purpose of the SQL JOIN clause in DBMS.  
   Answer: The JOIN clause in SQL is used to combine rows from two or more tables based on a related column between them. It allows data from different tables to be retrieved together to create more comprehensive result sets.
2. What are the advantages and disadvantages of using stored procedures in a DBMS?  
   Answer: Advantages of stored procedures include improved performance (as they are precompiled), reduced network traffic (since the code resides on the server), and enhanced security (as users only need execute permissions). However, disadvantages include increased complexity and maintenance overhead, limited portability across different database systems, and a potential lack of direct debugging capabilities.
3. What are the different types of locks in a DBMS, and how do they affect data concurrency?  
   Answer: The main types of locks are:

* Shared Lock (Read Lock): Allows multiple transactions to read data simultaneously but not modify it.
* Exclusive Lock (Write Lock): Only one transaction can acquire this lock, preventing other transactions from reading or writing the locked data.
* Update Lock: A hybrid lock that allows multiple transactions to read the data but prevents concurrent updates.  
  Intent Lock: Indicates that a transaction intends to acquire a higher-level lock to protect a range of data.

Locks play a vital role in managing data concurrency. Shared locks promote data read consistency, allowing multiple transactions to read the same data simultaneously. Exclusive locks ensure data integrity by preventing multiple transactions from modifying the same data simultaneously. However, excessive or poorly managed locks can lead to contention and performance issues, impacting the overall database performance. Proper lock management and lock escalation techniques are essential for efficient data concurrency control.

**Q1. Difference between groupby and orderby?**  
Difference between GROUP BY and ORDER BY:

GROUP BY and ORDER BY are two distinct clauses used in SQL for different purposes:

GROUP BY:

* GROUP BY is used to group rows based on specific columns.
* It is typically used with aggregate functions like SUM, COUNT, AVG, etc., to perform operations on groups of rows rather than individual rows.
* When you use GROUP BY, the result set is divided into groups, and the aggregate functions are applied to each group, returning one row per group.
* The columns listed in the GROUP BY clause determine the grouping criteria, and all rows with the same values in those columns are grouped together.

Example:

SELECT department, COUNT(\*) AS num\_employees

FROM employees

GROUP BY department;

ORDER BY:

* ORDER BY is used to sort the result set based on one or more columns.
* It does not group rows but rather rearranges the rows in the output based on the specified sort criteria.
* By default, ORDER BY sorts the result in ascending order. You can use the DESC keyword to sort in descending order.

Example:

SELECT name, age

FROM students

ORDER BY age DESC;

In summary, GROUP BY is used for aggregation and dividing rows into groups, while ORDER BY is used for sorting the result set based on specified columns.

**Q2: What are Views in DBMS?**

A view is a virtual table that is derived from one or more base tables or other views. It does not store any data itself but represents a tailored, pre-defined query that simplifies data retrieval. Views act as a layer of abstraction over the underlying tables, providing a more user-friendly and secure way to interact with the data.

Benefits of using views:

* Data Abstraction: Views allow users to work with a simplified representation of the data, hiding unnecessary details and complexity.
* Security: Views can be used to restrict access to certain columns or rows, providing a level of security by only showing specific data to specific users.
* Simplified Querying: Complex queries can be encapsulated within views, making it easier for users to retrieve the desired information without writing complex SQL statements.
* Data Independence: If the underlying schema changes, views can remain the same, and applications using the views will not be affected.

Q1 - How Database is accessed through Application Programs?  
Ans - Apps (written in C++/JAVA) interact with DB using Interface/API. API is provided to send DML/DDL statements to DB and retrieve the results.

Q2 - What is the role of Order by and Group by in SQL?  
Ans - Order by clause is used to sort the result set of a query based on one or more columns in ascending or descending order.  
Group by clause is used to collect data from multiple records and group the result based on one or more columns. It is used with aggregation functions like count(), avg(), etc.  
otherwise, it works like DISTINCT.

Q3 - What is Alias in MySQL?  
Ans - In MySQL, an alias is a temporary name assigned to a column, table, or expression used in a SQL query. It works as a Nickname for expressing the tables or column names.

Q4 - What are Views in MYSQL?  
Ans - A view is a database object that has no values. It contains rows and columns similar to a real table but does not contain any data of its own.

Q5 - What is a Functional Dependency (FD)?  
Ans - It is a relationship between the primary key attribute (usually) of the relation to that of the other attribute of the relation.

Q6 - What are Rules of FD(Armstrong's Axioms)  
Ans - 1. Reflexive, 2. Augmentation, 3. Transitivity

Q7 - What are the different types of anomalies introduced by data redundancy?  
Ans - There are 3 types of anomalies introduced by data redundancy - 1. Insertion Anomaly, 2. Deletion Anomaly, 3. Updation Anomaly.

Q8 - How to implement Atomicity and Durability in Transactions?  
Ans - Method 1 - Shadow Copy Scheme - Based on making copies of DB(aka shadow copies).  
Method 2 - Log Based Recovery Methods - Maintaining Logs of each transaction in some stable storage for easy recovery in case of failures.

Q9 - Are NoSQL Databases always relational? Do they never follow ACID Properties?  
Ans - False. NoSql databases can store relational Data. MongoDB supports ACID Transactions, unlike other NoSQL Databases.

Q-10 - What are the different types of NoSQL Data Models?  
Ans - 1. Key-Value Stores, 2. Column-Oriented, 3. Graph-based stores, 4. Document-based stores.