**1. Primary Key**

* A primary key is a unique identifier for each record in a table. It must contain unique values, and it cannot contain NULL values.

**2. Foreign Key**

* A foreign key is a field in one table that uniquely identifies a row of another table. It establishes a relationship between the two tables.

**3. Different Vendors Supporting SQL**

* SQL is supported by multiple vendors like Oracle, MySQL, Microsoft SQL Server, PostgreSQL, and SQLite, each having some unique features but adhering to standard SQL.

**4. Constraints**

* Constraints are rules enforced on data in a table to maintain accuracy and integrity. Types include NOT NULL, UNIQUE, PRIMARY KEY, FOREIGN KEY, CHECK, and DEFAULT.

**5. Referential Integrity**

* It ensures that foreign key values in one table correspond to primary key values in another, preventing orphan records and ensuring data consistency.

**6. Joins**

* Joins are used to combine rows from two or more tables. Types include INNER JOIN, LEFT JOIN, RIGHT JOIN, and FULL JOIN.

**7. Subqueries**

* A subquery is a query nested inside another SQL query. It can be used in SELECT, INSERT, UPDATE, and DELETE statements for complex filtering.

**8. Aggregate Functions**

* Functions like SUM, AVG, COUNT, MAX, and MIN are used to perform calculations on a set of values and return a single value.

**9. DELETE and TRUNCATE**

* DELETE removes rows from a table based on conditions and logs each row deletion. TRUNCATE removes all rows without logging individual row deletions and cannot be rolled back.

**10. DDL, DML, DCL, and TCL**

* **DDL** (Data Definition Language) - Defines data structures (CREATE, ALTER, DROP).
* **DML** (Data Manipulation Language) - Modifies data (INSERT, UPDATE, DELETE).
* **DCL** (Data Control Language) - Manages permissions (GRANT, REVOKE).
* **TCL** (Transaction Control Language) - Manages transactions (COMMIT, ROLLBACK).

**11. Normalization**

* The process of organizing data to reduce redundancy. Forms include 1NF, 2NF, 3NF, BCNF, etc., each with its own rules to eliminate redundancy.

**12. GROUP BY with HAVING**

* GROUP BY is used to group rows with the same values. HAVING filters groups based on aggregate function results (similar to WHERE but for groups).

**13. ER Diagram**

* An Entity-Relationship Diagram is a visual representation of the database structure, showing entities, attributes, and relationships.

**14. Cursor with Types**

* Cursors in SQL are used to retrieve rows one at a time. Types include implicit and explicit cursors, as well as FOR and SCROLL cursors in PL/SQL.

**15. Difference between Named and Unnamed Block**

* A named block has a defined name and can be reused. An unnamed block is an anonymous, one-time code block, often executed immediately.

**16. Exception Handling in PL/SQL**

* Handling runtime errors using EXCEPTION blocks. WHEN OTHERS captures all exceptions; specific exceptions can be handled individually.

**17. Triggers with Types**

* Triggers are automated actions executed on specified events. Types include BEFORE and AFTER triggers for INSERT, UPDATE, and DELETE.

**18. ON UPDATE CASCADE and ON DELETE CASCADE**

* ON UPDATE CASCADE automatically updates foreign keys if the referenced primary key changes.
* ON DELETE CASCADE automatically deletes rows with a foreign key that references a deleted primary key.

**19. How to Select a Primary Key**

* Choose a unique, non-null field, often an integer like an ID, that can uniquely identify each record and is unlikely to change.

**20. 2-Tier and 3-Tier Architecture**

* **2-Tier**: Client and server interact directly (e.g., client applications directly access the database).
* **3-Tier**: Adds a middle layer (application server), enhancing scalability and security.

**21. Difference between SQL and NoSQL**

* **SQL**: Relational, structured schema, and supports complex queries (e.g., MySQL).
* **NoSQL**: Non-relational, flexible schema, and optimized for unstructured data (e.g., MongoDB).

**22. CAP Theorem and BASE Property**

* **CAP Theorem**: Consistency, Availability, and Partition Tolerance — a distributed system can only guarantee two of the three.
* **BASE Property**: Stands for Basically Available, Soft state, and Eventually consistent, often used in NoSQL databases.

**23. Autosharding**

* A process in NoSQL databases to automatically distribute data across multiple servers to improve scalability.

**24. MapReduce and Aggregation Comparison**

* **MapReduce**: Used in NoSQL for processing large datasets by dividing them into smaller tasks.
* **Aggregation**: In SQL, similar to GROUP BY, used to summarize data.

**25. Object ID**

* A unique identifier in NoSQL databases like MongoDB, usually automatically generated for each document.

**26. JSON**

* JavaScript Object Notation is a lightweight, text-based format for data interchange. Used in NoSQL for storing unstructured data.

**27. NoSQL Data Models**

* Common models include document-based (e.g., MongoDB), key-value (e.g., Redis), column-based (e.g., Cassandra), and graph-based (e.g., Neo4j).

**28. INSERT and SAVE Difference**

* INSERT is used to add a new record. SAVE can both add a new record and update an existing one if a primary key is found.

**29. getIndex and ensureIndex**

* **getIndex**: Retrieves current indexes on a collection.
* **ensureIndex**: Creates an index if it doesn’t already exist, optimizing query performance.

**30. Proof of Applying Index**

* After creating an index, query performance is measured using EXPLAIN in SQL or getIndexes in MongoDB to verify improvements in efficiency.