JVM Institute

SQL notes (theory)

1. SQL-It stands for Structured Query Language. A programming language used for interaction with relational database management systems (RDBMS). This includes fetching, updating, inserting, and removing data from tables.

2. SQL statement-Also known as an SQL command. It's a string of characters interpreted by the SQL engine as a legal command and executed accordingly. Some examples of SQL statements are SELECT, CREATE, DELETE, DROP, REVOKE, and so on.

3. What is a database-A structured storage space where the data is kept in many tables and organized so that the necessary information can be easily fetched, manipulated, and summarized.

4. What is DBMS-It stands for Database Management System, a software package used to perform various operations on the data stored in a database, such as accessing, updating, wrangling, inserting, and removing data. There are various types of DBMS, such as relational, hierarchical, network, graph, or object-oriented

5. Types of SQL commands (or SQL subsets)

- Data Definition Language (DDL) – to define and modify the structure of a database-
- commands in ddl-CREATE, ALTER TABLE, DROP, TRUNCATE, and ADD COLUMN.

- Data Manipulation Language (DML) – to access, manipulate, and modify data in a database
- commands in dml-DML: UPDATE, DELETE, and INSERT

- Data Control Language (DCL) – to control user access to the data in the database and give or revoke privileges to a specific user or a group of users.
- commands in dcl-DCL: GRANT and REVOKE

- Transaction Control Language (TCL) – to control transactions in a database.
- commands in tcl-: COMMIT, SET TRANSACTION, ROLLBACK, and SAVEPOINT

- Data Query Language (DQL) – to perform queries on the data in a database to retrieve the necessary information from it.
- commands in DQL: – SELECT.

6. What are tables and fields in SQL?
   A table is an organized set of related data stored in a tabular form, i.e., in rows and columns. A field is another term for a column of a table.

7. What is a subquery-Also called an inner query; a query placed inside another query, or an outer query. A subquery may occur in the clauses such as SELECT, FROM, WHERE, UPDATE, etc. It's also possible to have a subquery inside another subquery. The innermost subquery is run first, and its result is passed to the containing query (or subquery).

8. What is a constraint, and why use constraints-A set of conditions defining the type of data that can be input into each column of a table. Constraints ensure data integrity in a table and block undesired actions.

9. What SQL constraints-
- DEFAULT – provides a default value for a column.
- UNIQUE – allows only unique values.
- NOT NULL – allows only non-null values.
- PRIMARY KEY – allows only unique and strictly non-null values (NOT NULL and UNIQUE).
- FOREIGN KEY – provides shared keys between two and more tables.

10. PRIMARY KEY-a primary key is a combination of the NOT NULL and UNIQUE constraints. The primary key uniquely identifies each record of the table. Each table should contain a primary key and can't contain more than one primary key.

11. unique key-A column (or multiple columns) of a table to which the UNIQUE constraint was imposed to ensure unique values in that column, including a possible NULL value (the only one).

12. foreign key-A column (or multiple columns) of a table to which the FOREIGN KEY constraint was imposed to link this column to the primary key in another table (or several tables). The purpose of foreign keys is to keep connected various tables of a database.

13. what is index-A special data structure related to a database table and used for storing its important parts and enabling faster data search and retrieval. Indexes are especially efficient for large databases, where they significantly enhance query performance..

14. types of index-
- Unique index – doesn't allow duplicates in a table column and hence helps maintain data integrity.
- Clustered index – defines the physical order of records of a database table and performs data searching based on the key values. A table can have only one clustered index.
- Non-clustered index – keeps the order of the table records that doesn't match the physical order of the actual data on the disk. It means that the data is stored in one place and a non-clustered index – in another one.

15. What is a schema?
    A collection of database structural elements such as tables, stored procedures, indexes, functions, and triggers. It shows the overall database architecture, specifies the relationships between various objects of a database, and defines different access permissions for them.

16. Difference between delete drop truncate- the DROP command removes a table from the database, DELETE removes records from a table, and TRUNCATE removes all rows from a table.

17. Difference between where and having clause-The main difference between WHERE and HAVING clause is that the WHERE clause allows you to filter data from specific rows (individual rows) from a table based on certain conditions. In contrast, the HAVING clause allows you to filter data from a group of rows in a query based on conditions involving aggregate.

18. 14.SQL Aggregate Functions-

- Aggregate functions are often used with the GROUP BY clause of the SELECT statement.
- MIN()- returns the smallest value within the selected column
- MAX()- returns the largest value within the selected column
- COUNT()- returns the number of rows in a set
- SUM()- returns the total sum of a numerical column
- AVG()- returns the average value of a numerical column
- Aggregate functions ignore null values (except for COUNT()).

19. Difference between group by orderby-
20. A GROUP BY statement sorts data by grouping it based on column(s) you specify in the query and is used with aggregate functions. An ORDER BY allows you to organize result sets alphabetically or numerically and in ascending or descending order.

21. Difference between union and unionall-
22. The main difference between Union and Union All is that Union has the ability to remove duplicate rows from the table while Union All cannot

23. Stored Procedure-
    A stored procedure is a prepared SQL code that you can save, so the code can be reused over and over again. So if you have an SQL query that you write over and over again, save it as a stored procedure, and then just call it to execute it.
    You can also pass parameters to a stored procedure, so that the stored procedure can act based on the parameter value(s) that is passed.

24. SQL joins:-

- A JOIN clause is used to combine rows from two or more tables, based on a related column between them.
- (INNER) JOIN: Returns records that have matching values in both tables
- LEFT (OUTER) JOIN: Returns all records from the left table, and the matched records from the right table
- RIGHT (OUTER) JOIN: Returns all records from the right table, and the matched records from the left table
- FULL (OUTER) JOIN: Returns all records when there is a match in either left or right table

25. Character manipulation functions:_

- CONCAT() – joins two or more string values appending the second string to the end of the first one
- SUBSTR() – returns a part of a string satisfying the provided start and end points
- LENGTH() (in other SQL flavors – LEN()) – returns the length of a string, including the blank spaces
- REPLACE() – replaces all occurrences of a defined substring in a provided string with another substring
- INSTR() – returns the numeric position of a defined substring in a provided string
- LPAD() and RPAD() – return the padding of the left-side/right-side character for right-justified/left-justified value
- TRIM() – removes all the defined characters, as well as white spaces, from the left, right, or both ends of a provided string

26. 20.set operators :_

- UNION – returns the records obtained by at least one of two queries (excluding duplicates)
- UNION ALL – returns the records obtained by at least one of two queries (including duplicates)
- INTERSECT – returns the records obtained by both queries
- EXCEPT (called MINUS in MySQL and Oracle) – returns only the records obtained by the first query but not the second one.

27. View:-
    A virtual table containing a subset of data retrieved from one or more database tables (or other views). Views take very little space, simplify complex queries, limit access to the data for security reasons, enable data independence, and summarize data from multiple tables.

28. Simple View
    A simple view is based on a single table and does not contain any complex joins, groups, or subqueries. It allows for basic data abstraction and security.

29. Complex View
    A complex view is based on multiple tables and may contain joins, groups, or subqueries. It provides a more advanced abstraction of data.

30. Materialized View (Indexed View)
    A materialized view, also known as an indexed view, stores the result set of the view physically. This can improve query performance, especially for complex and frequently accessed queries. However, it requires more storage and needs to be refreshed to stay up-to-date

31. Can we still use a view if the original table is deleted?
    No. Any views based on that table will become invalid after deleting the base table. If we try to use such a view anyway, we'll receive an error message.

32. Normalization in SQL
    Normalization is a process of database design that includes organizing and restructuring data in a way to reduce data redundancy, dependency, duplication, and inconsistency

33. Denormalization in SQL
    Denormalization is the process opposite of normalization: it introduces data redundancy and combines data from multiple tables

34. DIFFERENCE BETWEEN RANK,DENSERANK,ROWNUMBER

- RANK() assigns the same rank with gaps for equal values. DENSE_RANK() assigns the same rank without gaps for equal values. ROW_NUMBER() assigns a unique rank to each row.
- LEAD AND LAG
- The LAG function is used to access data from a previous row, while the LEAD function is used to return data from rows further down the result set.

35. How would you optimize a slow-performing query in SQL Server?
    - Optimize a slow-performing query in SQL Server, follow these steps:
    - Analyze the query execution plan for potential issues.
    - Use appropriate indexing to improve query performance.
    - Retrieve only necessary columns and rows to minimize data retrieval.
    - Optimize join conditions and types for efficient data retrieval.

- Simplify complex queries by breaking them into smaller parts.
- Update table and index statistics for accurate query optimization.
- Tune server configuration settings based on hardware and workload.
- Consider query rewriting or caching to improve performance.
- Monitor and optimize I/O performance.
- Keep SQL Server updated and maintain the database regularly.

36. What factors do you consider when designing and implementing database schemas for large-scale applications?
    - Functional requirements: Gather and understand the application's requirements to determine the entities, relationships, and data attributes needed.
    - Data integrity and consistency: Enforce constraints, define key relationships, and ensure data consistency.
    - Performance optimization: Optimize indexing, normalize or denormalize data structures, and optimize queries for efficient data retrieval.
    - Scalability and growth: Design the schema to accommodate future data growth and scalability requirements.
    - Security and access control: Implement proper access controls, authentication mechanisms, and data encryption to ensure security.
    - Data migration and integration: Plan for data migration, handle schema changes, and integrate with other systems or data sources.
    - Data normalization: Strive for an optimal level of data normalization to minimize redundancy and update anomalies.
    - Flexibility and extensibility: Design the schema to allow for future modifications and enhancements.

37. What is the ACID property in a database?
    - Atomicity: It means either the whole transaction succeeds or none of it will do. It is like going to a departmental store and buying a complete range of items if one item is missing, you do not buy anything at all.
    - Consistency: It is about how well the data sticks to the rules. Just imagine ensuring your recipe menu does all the steps correctly; in case any step is omitted or done wrongly then you will not serve the dish.
    - Isolation: Picture it as managing multiple transactions without interference. For instance, in a busy kitchen where every chef has their own dish; no one will interfere with someone else's recipe.
    - Durability: Look at it as how well the transaction can stand against interruptions. You can think of this as having your message passed through despite harsh weather, once sent always delivered regardless of external factors.

