SQL

1. Types of commands and their examples.

There are three types of commands in SQL: Data Definition Language (DDL), Data Manipulation Language (DML) and Data Control Language (DCL).

DDL : CREATE, ALTER, DROP

DML : SELECT, INSERTM UPDATE, DELETE

DCL: GRANT, REVOKE

1. What is Normalization and denormalization?

Normalisation is the process of organising data within a database to remove redundancy and dependency issues. Normalisation is usually done by breaking tables down into smaller groups.

Denormalization involves combining tables and introducing redundancy to improve performance due to the decreased number of tables.

1. Explain 1NF, 2NF, 3NF.

First Normal Form (1NF) : 1NF requires a table that has a primary key and each column contain values which can be divided further, i.e. atomic. 1NF eliminates repeating groups and ensures each attribute only holds one value.

Second Normal Form (2NF) : 2NF builds upon 1NF by requiring that every non-key attribute in a table is functionally dependent on the entire primary key, not just part of it. It eliminates partial dependencies. In other words, if you have a composite primary key consisting of multiple columns, each non-key attribute should depend on the entire composite key, not just a subset of it.

Third Normal Form (3NF) : 3NF refines the process of normalisation further by eliminating transitive dependencies. This means, if a non-key attribute depends on another non-key attribute, it should be moved to a separate table. This helps avoid data redundancies and anomalies.

1. Share use case where you had to do denormalization in database..

Denormalization is used to improve database performance by introducing redundancy. In an e-commerce application, denormalization can be applied to store product details, categories, and pricing information in a single table, reducing the need for joins and improving query performance. However, denormalization should be used carefully, considering the trade-off between performance and data integrity.

1. What is primary key and foreign key?

Primary Key: A column or set of columns that uniquely identifies each record in a table.

Foreign Key: A column or set of columns that references the primary key of another table, establishing a relationship between the two tables.

1. what is alternate and candidate key?

Candidate Key: A column or set of columns that uniquely identifies each record in a table and can be chosen as the primary key.

Alternate Key: A candidate key that is not selected as the primary key. It provides an alternative means to uniquely identify records in a table.

1. What are window functions?

Window functions are SQL functions that perform calculations and aggregations within a specified window or subset of rows. They allow for advanced analysis, ranking, and cumulative calculations without reducing the number of rows in the result set.

1. Explain Ranking Functions?

Ranking functions in SQL assign a rank or position to each row within a window. They order the rows based on a specified criteria and provide unique rank values. Ranking functions handle ties and can be partitioned. They are used to identify top or bottom performers and data based on rank.

1. Types of Joins?

1. Inner Join: The inner join returns only the matching rows from both tables. It retrieves records where the join condition is satisfied in both tables.

2. Left Join (or Left Outer Join): The left join returns all rows from the left (or "left-hand") table and the matching rows from the right table. If there is no match, null values are returned for the columns of the right table.

3. Right Join (or Right Outer Join): The right join is the opposite of the left join. It returns all rows from the right (or "right-hand") table and the matching rows from the left table. Non-matching rows from the left table contain null values.

4. Full Join (or Full Outer Join): The full join returns all rows from both tables, regardless of whether they have a match or not. Non-matching rows contain null values for columns of the opposite table.

5. Cross Join (or Cartesian Join): The cross join produces the Cartesian product of the two tables, combining each row from the first table with every row from the second table. It results in a large output, as it does not require any join condition.

6. Self Join: A self join occurs when a table is joined with itself. It is useful when you need to compare records within the same table based on a relationship or condition.

1. Use case when self join is required.

A common use case for a self join is when you want to compare or relate records within the same table, such as retrieving employee and manager information from an "Employees" table based on a shared relationship defined by columns like "EmployeeID" and "ManagerID".

1. What is subquery?

A subquery is a nested query enclosed within parentheses that is used within another query. It allows you to use the results of one query as part of another query, providing more specific or filtered results. Subqueries are used for various purposes such as filtering data, performing calculations, retrieving aggregated results, or constructing complex queries.

1. What is corelated subquery?

A correlated subquery is a subquery in which the inner query references a column from the outer query. It is executed for each row of the outer query and allows for comparisons or filtering based on the outer query's values.

1. What is CTE?

A Common Table Expression (CTE) is a temporary named result set that can be used within a SQL query. It allows for the creation of temporary result sets that improve query readability, and performance. CTEs are defined using the WITH clause and are particularly useful for simplifying complex queries or breaking them down into smaller parts.

1. Find third highest employee based on salary?

To find the third-highest employee based on salary, you can use the following SQL query:

SELECT EmployeeName, Salary

FROM Employees

ORDER BY Salary DESC

LIMIT 1 OFFSET 2;

Limit only returns 1 result and Offset of 2 skips the first two highest salaries, resulting in the third-highest salary being returned.

1. Find third highest employee based on salary per department?

Third highest employee by department could be done by:

SELECT Department, EmployeeName, Salary

FROM (

SELECT Department, EmployeeName, Salary,

ROW\_NUMBER() OVER (PARTITION BY Department ORDER BY Salary DESC) AS RowNum

FROM Employees

) AS RankedEmployees

WHERE RowNum = 3;

1. How to find duplicate values in a single column?

To find duplicate values in a single column, you can use the GROUP BY clause along with the HAVING clause in your SQL query:

SELECT ColumnName, COUNT(\*) as DuplicateCount

FROM TableName

GROUP BY ColumnName

HAVING COUNT(\*) > 1;

HAVING filters out the results with a count greater than 1, which indicates there is a duplicate.

1. How to find duplicate values in a multiple column?

Similar to the query previous, we now select the multiple columns we wish to examine:

SELECT Column1, Column2, COUNT(\*) as DuplicateCount

FROM TableName

GROUP BY Column1, Column2

HAVING COUNT(\*) > 1;

1. What are ACID properties?

ACID properties are fundamental principles in database systems:

- Atomicity ensures that a transaction is treated as a single unit of work.

- Consistency ensures that data remains in a valid state throughout a transaction.

- Isolation ensures that concurrent transactions do not interfere with each other.

- Durability guarantees that committed changes are permanent and survive system failures.

1. Diff between union and union all

UNION: Returns distinct rows from multiple SELECT statements, eliminating duplicates.

UNION ALL: Returns all rows from multiple SELECT statements, including duplicates.

1. Diff between primary key and unique key

The main differences between Primary Key and Unique key are:

Primary Key:

- Uniquely identifies each record in a table.

- Mandatory and enforces uniqueness.

- Allows no NULL values.

- Only one primary key per table.

Unique Key:

- Ensures uniqueness in a table.

- Allows NULL values (except for one unique key).

- Optional and can be defined multiple times per table.

1. Diff between truncate and delete

The main differences between TRUNCATE and DELETE are:

TRUNCATE:

- Removes all data from a table.

- Faster and does not generate individual delete statements.

- Cannot be rolled back.

- Resets identity column (if applicable).

DELETE:

- Removes specific rows based on a condition.

- Slower and generates individual delete statements.

- Can be rolled back.

- Triggers and associated triggers are executed.

1. SQL query execution order.

