FAQ’s: SQL: short answers

1. Types of commands and their examples.

DDL – Data Definition Language – create, alter (add/modify/drop), drop

DML – Data Manipulation Language – insert, update, delete

DCL – Data Control Language – grant, revoke

TCL – Transaction Control Language – commit, rollback

2. What is Normalization and Denormalization?

Normalization is the process of organizing data in a database to minimize redundancy and dependency. It ensures data integrity and prevents anomalies.

Denormalization is the opposite process, where redundancy is intentionally added to improve query performance. It sacrifices some normalization benefits for faster reads.

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

1NF: Ensures each cell in a table contains atomic values, eliminating repeating groups.

2NF: Extends 1NF by ensuring all attributes depend on the entire primary key, avoiding partial dependencies.

3NF: Builds upon 2NF by removing transitive dependencies, ensuring each attribute depends only on the primary key.

These Normal Forms help organize data efficiently, minimize redundancy, and prevent anomalies in database design.

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

In an e-commerce website with a large product catalog, denormalization is applied to improve query performance. Redundant category information is stored directly within the "Products" table, reducing the need for joins when users browse products within a category. This optimization enhances response times and user experience, especially during high traffic periods.

5.What is primary key and foreign key?

A primary key uniquely identifies each record in a table, while a foreign key establishes relationships between tables by referencing the primary key of another table, thus enforcing referential integrity in a relational database.

6. What is alternate and candidate key?

Candidate keys are potential choices for primary keys within a table, and alternate keys are the candidate keys that are not selected as the primary key. Both candidate keys and alternate keys are essential for maintaining data integrity and ensuring that each row within a table can be uniquely identified.

7.What are window functions?

Window functions are SQL functions that operate on a set of rows related to the current row. They can perform calculations, aggregations, rankings, and more within defined partitions and orderings, without grouping rows together. Examples include ROW\_NUMBER(), SUM(), LEAD(), and RANK(). They offer powerful analytical capabilities for tasks like ranking, aggregation, and trend analysis within SQL queries.

8.Explain Ranking Functions? GIven a small table , write the output.

Ranking functions in SQL assign a rank to rows based on specified criteria within a partition of the result set.

ROW\_NUMBER(): Assigns a unique number to each row within the partition.

RANK(): Assigns a unique rank to each distinct row, allowing ties with gaps in ranks.

DENSE\_RANK(): Similar to RANK(), but without gaps in ranks for ties.

These functions are useful for analyzing data based on order or other criteria, providing insights into the relative positions of rows within a dataset.

9. Types of Joins? With example and usecase. All the number of records return and exact records.

INNER JOIN: Returns rows with matching values in both tables.

LEFT JOIN: Returns all rows from the left table and matched rows from the right table.

RIGHT JOIN: Returns all rows from the right table and matched rows from the left table.

FULL JOIN: Returns all rows when there's a match in either table.

CROSS JOIN: Returns the Cartesian product of both tables, i.e., all possible combinations of rows.

10.Use case when self join is required.

Hierarchical Data:

Used to represent parent-child relationships within the same table, such as organizational structures.

Comparing Records:

Enables comparison of data within the same table, useful for analysing trends or identifying anomalies, such as comparing an employee's performance with their peers or past performance.

11.What is subquery?

A subquery is a nested query within another SQL statement, used to retrieve data based on specific conditions. It's enclosed in parentheses and can be used in various parts of a SQL statement like WHERE, FROM, HAVING, or SELECT. Subqueries help filter, join, or manipulate data in the outer query.

12.What is corelated subquery?

A correlated subquery is a type of subquery where the inner query references columns from the outer query. It relies on the outer query for its execution and is used for dynamic filtering or comparison based on the current row being processed.

13.What is CTE?

A CTE (Common Table Expression) is a temporary result set in SQL that you can reference within a query. It's defined using the WITH keyword and offers a way to write more readable and modular SQL queries.

14.What is derived table?

A derived table is a temporary table created within a SQL query using a SELECT statement in the FROM clause. It's useful for performing intermediate calculations or transformations on data before using it in the main query.

15.Find third highest employee based on salary?

SELECT employee\_id, employee\_name, salary

FROM employees

ORDER BY salary DESC

LIMIT 1 OFFSET 2;

-- This query orders the employees by their salary in descending order

-- Then skips the first two rows using OFFSET 2.

-- Finally, it limits the result to only one row, which represents the third-highest-paid employee.

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

SELECT employee\_id, employee\_name, salary, department\_id

FROM (

SELECT employee\_id, employee\_name, salary, department\_id,

ROW\_NUMBER() OVER (PARTITION BY department\_id ORDER BY salary DESC) AS rank

FROM employees

) AS ranked\_employees

WHERE rank = 3;

This query selects the employee details, ranks them within each department based on salary using the ROW\_NUMBER() window function, and filters for those with a rank of 3, indicating the third-highest-paid employee per department.

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

SELECT column\_name, COUNT(\*) AS duplicate\_count

FROM table\_name

GROUP BY column\_name

HAVING COUNT(\*) > 1;

This query groups the rows by the values in the specified column, counts the occurrences of each value, and then filters for groups with a count greater than 1, indicating duplicates. It returns the duplicate values along with their counts.

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

SELECT column1, column2, ..., COUNT(\*) AS duplicate\_count

FROM table\_name

GROUP BY column1, column2, ...

HAVING COUNT(\*) > 1;

This query will return rows where the specified columns have duplicate combinations of values, along with the count of occurrences.

19.What are ACID properties? give example for each property

ACID stands for Atomicity, Consistency, Isolation, and Durability. It ensures that database transactions are reliable, consistent, independent, and durable.

Atomicity: Ensures all operations in a transaction are completed successfully, or none at all. For instance, in a bank transfer, if funds are debited from one account, they must also be credited to the other account.

Consistency: Guarantees that the database remains in a valid state before and after a transaction. For example, if a database enforces an age constraint, attempting to set a customer's age below the minimum allowed value would be rejected to maintain consistency.

Isolation: Ensures transactions are executed independently without interference from other transactions. In a scenario where two users update the same bank account simultaneously, isolation prevents one transaction from affecting the other.

Durability: Once a transaction is committed, its changes are permanent even in the face of system failures. For instance, if a customer's address is updated, the change will be retained and recoverable even after a system crash.

20.Diff between union and union all

If you want to include only distinct rows in the result set and remove duplicates, use UNION. If you want to include all rows from both result sets, including duplicates, use UNION ALL.

21.Diff between primary key and unique key

Both primary key and unique key constraints enforce uniqueness, but the primary key uniquely identifies each record in a table and does not allow null values, while a unique key allows null values but enforces uniqueness within a column or set of columns.

22.Diff between truncate and delete

The DELETE statement is used when we want to remove some or all of the records from the table, while the TRUNCATE statement will delete entire rows from a table. DELETE is a DML command as it only modifies the table data, whereas the TRUNCATE is a DDL command.

23.Diff between having and where

WHERE: Filters individual rows based on specified conditions.

HAVING: Filters grouped results based on aggregate values after grouping.

In simple terms, WHERE is for filtering rows, while HAVING is for filtering groups.

24.SQL query execution order.

SQL query execution order: FROM-JOIN-ON-WHERE-GROUP BY-HAVING-ORDERBY-LIMIT

25.What are indexes? Types of Indexes and their differences.

Indexes in databases are structures that speed up data retrieval by providing quick access paths to rows based on column values.

Common types include:

B-tree Index: Balanced tree structure, suitable for most queries.

Hash Index: Uses hash table, efficient for equality comparisons.

Bitmap Index: Represents distinct values as bitmaps, useful for low cardinality columns.

Full-Text Index: Optimized for text searches, enabling keyword searches and relevance ranking.

Spatial Index: Specialized for spatial data types, facilitating efficient spatial operations.

Composite Index: Created on multiple columns, useful for queries involving combinations of column values.

26.What is surrogate key? Give example where you used it and how.

A surrogate key is a system-generated unique identifier used as the primary key in a database table. It has no inherent meaning and is commonly used to ensure uniqueness and improve performance in database operations.

Example: In a "products" table for an online retail store, a surrogate key "product\_id" is used as the primary key instead of a natural key like the product name or SKU. It's automatically generated for each new product added to the table, ensuring uniqueness and simplifying database operations.