1.SQL commands

Thursday, September 26, 2024 12:21 PM

Types of SQL Commands and Examples
Short: SQL commands include DDL(data definition language) (e.g., CREATE), DML(data manipulation language) (e.g., INSERT), DCL(Data control language) (e.g., GRANT), TCL(Transactional control language) (e.g., COMMIT), and DQL (data query language)(e.g., SELECT).
Long: SQL commands are divided into categories based on their functions. DDL commands like CREATE and ALTER define or modify structures. DML commands like INSERT and UPDATE manipulate data. DCL commands like GRANT control access to data. TCL commands like COMMIT manage transactions. DQL commands like SELECT retrieve data.

2.NORM and DENORM

Friday, September 27, 2024 10:29 AM

2. What is Normalization and Denormalization?
Short: Normalization reduces redundancy, while denormalization improves performance by adding redundancy.
Long: Normalization is a database design technique to minimize redundancy and dependencies, organizing data into smaller tables to eliminate anomalies. Denormalization is the process of adding redundant data to improve query performance in specific use cases, such as read-heavy applications like reporting dashboards.

3.1NF,2NF and 3NF

Friday, September 27, 2024 1:02 PM

Explain 1NF, 2NF, and 3NF
Short: 1NF ensures atomic values, 2NF eliminates partial dependencies, and 3NF removes transitive dependencies.
Long: 1NF ensures that each table column holds atomic values and has a primary key. 2NF requires that non-key attributes depend on the entire primary key, eliminating partial dependencies. 3NF ensures no transitive dependencies, meaning non-key attributes depend only on the primary key.

4.Use case of Denorm

Friday, September 27, 2024 1:04 PM

Use Case for Denormalization in a Database
Short: Denormalization is useful when building a reporting dashboard for fast, aggregated data retrieval.
Long: In real-time reporting systems, denormalization can improve performance. For instance, a denormalized summary table can pre-aggregate sales metrics, reducing the need for complex joins and allowing fast report generation on large datasets.

5.Primary and Foreign Key

Friday, September 27, 2024 1:05 PM

What is a Primary Key and Foreign Key?
Short: A primary key uniquely identifies rows, while a foreign key links two tables.
Long: A primary key is a column or set of columns that uniquely identifies each row in a table. A foreign key is a column in one table that references a primary key in another table, creating a relationship between the two tables, enforcing referential integrity.

6.Alternate and Candidate key

Friday, September 27, 2024 1:07 PM

What is an Alternate Key and Candidate Key?
Short: A candidate key uniquely identifies rows, and an alternate key is a candidate key that isn’t chosen as the primary key.
Long: A candidate key is any column or combination of columns that can uniquely identify a row. If a table has multiple candidate keys, one is selected as the primary key, and the rest are alternate keys. Alternate keys provide additional uniqueness constraints but are not used as the main identifier.

7.Window Function

Friday, September 27, 2024 1:09 PM

What are Window Functions?
Short: Window functions perform calculations across a set of table rows without collapsing them.
Long: Window functions like RANK, ROW_NUMBER, and SUM operate on a range of rows related to the current row, without grouping them into a single result. These functions are useful for tasks like calculating running totals, rankings, or averages over a set of related rows in a query.

8.Ranking Function

Friday, September 27, 2024 2:16 PM

What are Ranking Functions?
Short: Ranking functions assign ranks to rows based on column values.
Long: Ranking functions like ROW_NUMBER, RANK, and DENSE_RANK assign sequential ranks to rows within a partition. These are useful in scenarios like ranking products by sales or employees by performance.

9.Joins Types

Friday, September 27, 2024 2:18 PM

Types of Joins and Their Uses
Short: Common join types include INNER JOIN (matches in both tables), LEFT JOIN (all from left table),RIGHT JOIN(all from right table), and FULL JOIN (all from both tables).
Long: INNER JOIN returns rows present in both tables based on a matching condition. LEFT JOIN returns all rows from the left table, even if no match is found in the right table. RIGHT JOIN FULL returns all rows from right table, even if no match is found in left table. JOIN returns all rows from both tables, matching where possible and filling in NULLs where no match exists.

10.Self Join Use Case

Friday, September 27, 2024 2:22 PM

10. Use Case for Self-Join?
Short: Self joins are used to compare rows within the same table, like finding employees who report to the same manager.
Long: A self-join compares rows within the same table by joining it to itself. For example, in an employee table, you might use a self-join to identify employees and their managers by linking the manager's ID to the employee's manager column.

11.Subquery

Friday, September 27, 2024 2:24 PM

What is a Subquery?
Short: A subquery is a query inside another query.
Long: A subquery is a nested query within the main SQL query. It is used for performing operations that depend on the results of another query. Subqueries are often used in WHERE or FROM clauses to filter or aggregate data.

12.Correlated Subquery

Friday, September 27, 2024 2:26 PM

What is a Correlated Subquery?
Short: A correlated subquery depends on the outer query for its values.
Long: A correlated subquery references columns from the outer query and is executed once for each row processed by the outer query. It's typically used for row-by-row comparisons, such as finding employees earning more than the average salary in their department.

13.CTE

Friday, September 27, 2024 2:28 PM

13. What is CTE?
Short: A CTE is a temporary result set used to simplify complex queries.
Long: A Common Table Expression (CTE) is a temporary result set defined using the WITH clause and referenced within a query. CTEs are useful for breaking down complex queries, making them more readable, and for recursive operations like traversing hierarchical data.

14.Derived Table

Friday, September 27, 2024 2:28 PM

14. What is a Derived Table?
Short: A derived table is a subquery used in the FROM clause.
Long: A derived table is a subquery that is treated as a temporary table in the FROM clause of a SQL query. It simplifies complex queries by allowing you to encapsulate query logic and reuse the result set as if it were a table.

15.Find Third Highest Salary

Friday, September 27, 2024 2:30 PM

How to Find the Third Highest Salary?
Short: Use a subquery or ranking function to find the third highest salary.
Long: To find the third highest salary, you can use a subquery that orders salaries in descending order and skips the first two, or a ranking function like DENSE_RANK() to assign ranks and select the row with a rank of 3.

16.Highest Salary Per Department

Friday, September 27, 2024 2:32 PM

How to Find Third Highest Salary Per Department?
Short: Use a window function like DENSE_RANK() partitioned by department.
Long: To find the third highest salary within each department, use a window function like DENSE_RANK() partitioned by department. This allows you to rank employees by salary within each department and select those ranked third.

17.Find Duplicate single col

Friday, September 27, 2024 2:34 PM

How to Find Duplicate Values in a Single Column?
Short: Use GROUP BY and HAVING COUNT(*) > 1 to find duplicates.
Long: To find duplicate values, use the GROUP BY clause on the column in question and apply the HAVING COUNT(*) > 1 condition. This will return all values that appear more than once in the column.

18.Duplicate multiple cols

Friday, September 27, 2024 2:35 PM

18. How to Find Duplicate Values in Multiple Columns?
Short: Group by the combination of columns and use HAVING COUNT(*) > 1.
Long: To find duplicates across multiple columns, use GROUP BY on the combination of columns and apply HAVING COUNT(*) > 1. This identifies rows where the combination of values is duplicated.

19. Acid Properties

Friday, September 27, 2024 2:37 PM

What are ACID Properties?
Short: ACID ensures Atomicity, Consistency, Isolation, and Durability in transactions.
Long: ACID properties maintain data integrity in databases. Atomicity ensures all operations in a transaction are completed or rolled back. Consistency ensures data remains valid before and after a transaction. Isolation prevents transactions from interfering with each other, and Durability ensures that once a transaction is committed, it remains persistent, even in the case of system failures.

20.Union and Union ALL

Friday, September 27, 2024 2:38 PM

Difference Between UNION and UNION ALL
Short: UNION removes duplicates; UNION ALL includes all rows.
Long: UNION combines the result sets of two queries, removing duplicates. UNION ALL combines the result sets but includes all duplicates, making it faster as no duplicate check is performed.

21.Primary and Unique Key

Friday, September 27, 2024 2:39 PM

Difference Between Primary Key and Unique Key
Short: Primary key enforces uniqueness and disallows NULLs; unique key allows one NULL.
Long: A primary key uniquely identifies each row in a table and does not allow NULL values. A unique key also enforces uniqueness but allows one NULL value. There can be only one primary key, but a table can have multiple unique keys.

22.Truncate and Delete

Friday, September 27, 2024 2:40 PM

Difference Between TRUNCATE and DELETE
Short: TRUNCATE removes all rows and is faster; DELETE removes selected rows and can be rolled back.
Long: TRUNCATE is a DDL command that removes all rows from a table quickly, resetting any auto-increment values but cannot be rolled back. DELETE is a DML command that removes rows based on a condition and can be rolled back if within a transaction.

23.Having and Where

Friday, September 27, 2024 2:42 PM

Difference Between HAVING and WHERE
Short: WHERE filters rows before grouping; HAVING filters after.
Long: WHERE filters rows before any grouping is done and applies to individual rows. HAVING filters the results after aggregation and applies to groups of rows.

24.SQL execution order

Friday, September 27, 2024 2:43 PM

SQL Query Execution Order

Short: SQL executes queries in this logical order: FROM, WHERE, GROUP BY, HAVING, SELECT, ORDER BY, LIMIT.

Long: SQL queries are logically executed in this order: first, tables are identified in the FROM clause, then filtering with WHERE, grouping with GROUP BY, filtering of groups with HAVING, selecting columns, ordering results with ORDER BY, and finally limiting the result set with LIMIT or OFFSET.

25.Indexes

Friday, September 27, 2024 2:44 PM

What are Indexes?
Short: Indexes speed up data retrieval by creating fast lookup paths.
Long: Indexes are special data structures that improve query performance by providing quick access to rows. Types include clustered, non-clustered, and unique indexes, which differ in how they store data and enforce uniqueness.

26.Surrogate Key

Friday, September 27, 2024 2:45 PM

What is a Surrogate Key?
Short: A surrogate key is an artificial unique identifier for rows.
Long: A surrogate key is a system-generated identifier, usually an auto-incremented number, that uniquely identifies each row. It is used when no natural key is available or practical for use as a primary key, ensuring uniqueness across rows.

27.Optimize SQL Queries

Friday, September 27, 2024 2:46 PM

Ways to Optimize SQL Queries
Short: Use indexes, avoid SELECT *, optimize joins, and use appropriate filtering conditions.
Long: SQL queries can be optimized by indexing frequently queried columns, avoiding unnecessary data retrieval (SELECT *), optimizing join operations with proper indexing, and using filtering conditions that allow for index utilization (e.g., avoiding functions on indexed columns in the WHERE clause). Additionally, query execution plans should be analyzed for bottlenecks.