Top Amazon SQL Interview Questions and Answers

Okay, let's get started with some of the most common Amazon SQL interview questions. These will give you an idea of what to expect and help you answer confidently.

1. What is SQL and why is it used in database management?

Answer: SQL, or Structured Query Language, is a language used to interact with relational databases. It allows you to create, modify, query, and manage data within these databases. SQL is essential for efficient data retrieval, manipulation, and administration, making it a crucial tool for keeping data systems running smoothly.

2.How do you handle NULL values in SQL, and how does COALESCE work?

Answer: In SQL, NULL represents missing or unknown values. Functions like IS NULL and IS NOT NULL are used to check for NULL values in a column.

The COALESCE() function returns the first non-null value in the list of expressions passed to it. It's helpful for replacing NULL values with default values.

```
SELECT COALESCE(column_name, 'Default Value') FROM table_name;
```

3. What are the different types of JOINS in SQL?

Answer: In SQL, joins combine rows from two or more tables.

- **An INNER JOIN** returns records when there is a match in both tables.
- **A LEFT (OUTER) JOIN** returns all records from the left table and matched records from the right table. Unmatched records will return NULL values.
- **A RIGHT (OUTER) JOIN** returns all records from the right table and matched records from the left table.
- **A FULL (OUTER) JOIN** returns records when there is a match in one of the tables. If there are unmatched records in either table, they will return as NULL.
- **A CROSS JOIN** returns the Cartesian product of both tables (every row from the first table is paired with every row from the second table).

4. How would you optimize a slow running query?

Answer: Optimizing a slow running query involves:

- Use Indexes: Create indexes on the columns that are used in the WHERE clause or join conditions.
- *Don't SELECT : Specify only the columns you need to retrieve instead of selecting all columns.
- Rewrite Subqueries: Use JOINs instead of subqueries where possible.

- Analyze Execution Plan: Use the EXPLAIN statement to see the query execution plan and identify bottlenecks.
- Limit the Number of Rows Returned: Use the LIMIT clause if you only need a subset of the results.

5. What is normalization and why is it used?

Answer: Normalization is the process of organizing database tables to reduce redundancy and improve data integrity. The main goal is to have each table contain data about a single entity and related data in separate tables. Normalization is used to eliminate data duplication, make the database more efficient, and help in maintaining data consistency.

6. What is a stored procedure and how is it different from a function in SQL?

Answer: A stored procedure is a precompiled set of SQL statements that can be run as a program. It helps to encapsulate business logic in the database. The main differences between stored procedures and functions are:

- Stored Procedures: Can do actions (like modify data) and don't have to return a value.
- Functions: Must return a value and are used for calculations or to retrieve data.

7. What is a window function in SQL? Give an example.

Answer: A window function in SQL performs calculations across a set of rows that are related to the current row. Unlike aggregate functions, window functions don't group rows into a single output row—they keep individual rows.

Example:

ELECT employee_id, department_id, salary, RANK() OVER (PARTITION BY department_id ORDER BY salary DESC) AS rank FROM employees;

This will rank employees within each department by salary.

8. How do you handle NULL values in SQL queries?

Answer: You can handle NULL values using functions like COALESCE or ISNULL to replace NULL with default values. You can also use the IS NULL or IS NOT NULL operators in the WHERE clause to filter or check for NULL values in your query.

9. How do you implement pagination in SQL?

Answer: Pagination is implemented using the LIMIT and OFFSET clauses in SQL. For example:

SELECT * FROM products ORDER BY product_id LIMIT 10 OFFSET 20;

This will return 10 rows starting from the 21st row, for pagination.

10. What are the ACID properties in SQL?

Answer: ACID stands for Atomicity, Consistency, Isolation, and Durability. These are the key properties that ensure reliable transaction processing in a database system:

- **Atomicity ensures** that all operations within a transaction are completed or none at all.
- **A transaction must ensure consistency**, bringing the database from one valid state to another.
- **Transactions must be isolated** from each other.
- **Once committed, a transaction is durable**, remaining permanent even in the event of a system failure.

11. What is a primary key and how is it different from a unique key?

Answer: A primary key is a column (or combination of columns) that identifies each row in a table. It can't have NULL values and there can only be one primary key in a table. A unique key ensures all values in a column are unique but can have one NULL value. You can have multiple unique keys in a table.

12. How to get duplicate records from a table?

Answer: To get duplicate records, you can use the GROUP BY clause along with the HAVING clause to filter rows that have duplicate values. Example:

SELECT column_name, COUNT(*) FROM table_name GROUP BY column_name HAVING COUNT(*) > 1;

This will return the rows that have duplicate values in the specified column.

13. What is the difference between UNION and UNION ALL.

Answer:

UNION: Combines the result set of two or more SELECT statements and removes any duplicate rows.

UNION ALL: Combines the result set of two or more SELECT statements but includes all duplicate rows.

14. How to get the second highest salary from an employee table?

Answer: There are several ways to get the second highest salary. One common way is using the LIMIT clause with a subquery:

SELECT MAX(salary) FROM employee WHERE salary < (SELECT MAX(salary) FROM employee);

Another way is using the ROW_NUMBER() function:

SELECT salary FROM (SELECT salary, ROW_NUMBER() OVER (ORDER BY salary DESC) AS rank FROM employee) temp WHERE rank = 2;

15. What is an index and why in SQL?

Answer: An index is a database object that helps to speed up data retrieval from a table. It creates an internal structure so that SQL can find rows faster without scanning the entire table. Indexes are important for query optimization especially in large tables. But indexes slow down insert, update and delete operations since the index needs to be updated.

16. How to delete duplicate rows in SQL?

Answer: You can use a common table expression (CTE) or subquery with ROW_NUMBER() function to identify and delete duplicates. Example:

WITH CTE AS (SELECT column_name, ROW_NUMBER() OVER (PARTITION BY column_name ORDER BY column_name) AS row_num FROM table_name) DELETE FROM CTE WHERE row_num > 1;

This will delete duplicate rows while retaining the first occurrence.

17. How to create self join in SQL?

Answer: Self join is a join of a table with itself. It's used when you need to compare rows within the same table. Example:

SELECT A.employee_name, B.manager_name FROM employees A JOIN employees B ON A.manager_id = B.employee_id;

This query joins the employees table with itself to find the manager of each employee.

18. What is a cross join?

Answer: A cross join produces the Cartesian product of two tables. It pairs every row from the first table with every row from the second table. The result is a big list of combinations between both tables.

SELECT * FROM table1 CROSS JOIN table2;

If table1 has 3 rows and table2 has 4 rows, the result will have 12 rows.

19. How do you use the COALESCE function in SQL?

Answer: COALESCE returns the first non-NULL value from a list of arguments. It's used to handle NULLs.

Example:

SELECT COALESCE(column1, column2, 'Default') FROM table_name;

This will return the value from column1 if it's not NULL, otherwise column2. If both are NULL, it returns 'Default'.

20. How to get the last inserted ID in a table?

Answer: You can use the LAST_INSERT_ID() function in MySQL to get the last inserted ID in a table. Other databases like PostgreSQL use the RETURNING clause, and SQL Server uses SCOPE_IDENTITY().

INSERT INTO table_name (column1, column2) VALUES (value1, value2); SELECT LAST_INSERT_ID();

21. What is the difference between TRUNCATE and DELETE?

Answer:

- **DELETE** removes rows from a table one by one, logging each deletion. It can be rolled back and uses a WHERE clause to specify which rows to delete.
- **A truncate command** removes all rows from a table without logging individual row deletions and cannot be rolled back. It's faster than DELETE but doesn't fire triggers.

22. How do you use window functions in SQL?

Answer: Window functions perform calculations across a set of table rows related to the current row, but unlike aggregate functions, they do not group the result into a single row.

Example:

SELECT employee_name, department, salary, RANK() OVER (PARTITION BY department ORDER BY salary DESC) AS salary_rank FROM employees;

This query ranks employees based on salary within each department.

23. What's the difference between aggregate and scalar functions?

Answer:

Aggregate Functions: Operate on a set of values and return a single value (e.g., SUM, AVG, MAX, MIN, COUNT).

Scalar Functions: Operate on a single value and return a single value (e.g., UPPER, LOWER, ROUND, LENGTH).

24. How do you update data in one table based on data in another table?

Answer: You can use an UPDATE statement with a JOIN to update one table based on data in another table.

Example:

UPDATE table1 JOIN table2 ON table1.id = table2.id SET table1.column_name = table2.column_name;

This updates table1 with data from table2 where id matches.

25. How do you handle many-to-many relationships in SQL?

Answer: A many-to-many relationship is handled by a junction table (or bridge table). The junction table has foreign keys from both tables to connect them.

Example:

CREATE TABLE student_course (student_id INT, course_id INT, PRIMARY KEY (student_id, course_id), FOREIGN KEY (student_id) REFERENCES students(student_id), FOREIGN KEY (course_id) REFERENCES courses(course_id));

26. How do you get a running total in SQL?

Answer: You can get a running total with the SUM() window function.

Example:

SELECT order_id, order_amount, SUM(order_amount) OVER (ORDER BY order_id) AS running_total FROM orders;

27. What is the difference between DELETE CASCADE and DELETE RESTRICT?

Answer:

- **DELETE CASCADE** deletes the related rows in the child table when you delete a row in the parent table.
- **DELETE RESTRICT** prevents the deletion of a row in the parent table if there are related rows in the child table.

28. How do you get only the first name from a full name column in SQL?

Answer: You can use the SUBSTRING and CHARINDEX functions to get the first name from a full name column.

Example:

SELECT SUBSTRING(full_name, 1, CHARINDEX(' ', full_name) - 1) AS first_name FROM employees;

This assumes the full name is in "First Last" format.

29. How do you get the difference between two dates in SQL?

Answer: You can get the difference with the DATEDIFF() function.

Example:

SELECT DATEDIFF(day, '2024-01-01', '2024-10-07') AS date_diff;

This gets the difference between two dates.

30. What is a foreign key and how does it enforce referential integrity?

Answer: A foreign key is a column or set of columns in one table that references the primary key in another table. It enforces referential integrity by making sure the value in the foreign key column matches a value in the referenced primary key column. This prevents orphaned records in the database.