Here are some **SQL interview questions** along with their detailed answers:

**1. How do you group data by month and year?**

To group data by **month and year**, you can use **EXTRACT()** or the equivalent in SQL. Here's an example:

**Example (PostgreSQL/Standard SQL):**

SELECT

EXTRACT(YEAR FROM ORDERDATE) AS order\_year,

EXTRACT(MONTH FROM ORDERDATE) AS order\_month,

SUM(REVENUE) AS total\_revenue

FROM sales\_order

GROUP BY

EXTRACT(YEAR FROM ORDERDATE),

EXTRACT(MONTH FROM ORDERDATE)

ORDER BY order\_year, order\_month;

**Example (MySQL):**

SELECT

YEAR(ORDERDATE) AS order\_year,

MONTH(ORDERDATE) AS order\_month,

SUM(REVENUE) AS total\_revenue

FROM sales\_order

GROUP BY

YEAR(ORDERDATE),

MONTH(ORDERDATE)

ORDER BY order\_year, order\_month;

**2. What's the difference between COUNT(\*) and COUNT(DISTINCT col)?**

* **COUNT(\*)**: Counts all rows in a table, including those with NULL values. It simply returns the total number of rows in a query result.
* **COUNT(DISTINCT col)**: Counts only **unique** non-NULL values in the specified column. It ignores duplicate values.

**Example:**

-- COUNT(\*) counts all rows

SELECT COUNT(\*) FROM sales\_order;

-- COUNT(DISTINCT CLIENTNO) counts unique client numbers

SELECT COUNT(DISTINCT CLIENTNO) FROM sales\_order;

**3. How do you calculate monthly revenue?**

To calculate **monthly revenue**, group by **year and month**, and use **SUM()** to total the revenue for each month.

**Example (PostgreSQL/Standard SQL):**

SELECT

EXTRACT(YEAR FROM ORDERDATE) AS order\_year,

EXTRACT(MONTH FROM ORDERDATE) AS order\_month,

SUM(REVENUE) AS monthly\_revenue

FROM sales\_order

GROUP BY

EXTRACT(YEAR FROM ORDERDATE),

EXTRACT(MONTH FROM ORDERDATE)

ORDER BY order\_year, order\_month;

**Example (MySQL):**

SELECT

YEAR(ORDERDATE) AS order\_year,

MONTH(ORDERDATE) AS order\_month,

SUM(REVENUE) AS monthly\_revenue

FROM sales\_order

GROUP BY

YEAR(ORDERDATE),

MONTH(ORDERDATE)

ORDER BY order\_year, order\_month;

**4. What are aggregate functions in SQL?**

**Aggregate functions** are used to perform calculations on a set of values and return a single result. Common aggregate functions include:

* **COUNT()**: Counts rows or distinct values.
* **SUM()**: Sums numeric values.
* **AVG()**: Calculates the average value.
* **MIN()**: Returns the smallest value.
* **MAX()**: Returns the largest value.

**Example:**

SELECT

COUNT(\*) AS total\_orders,

SUM(REVENUE) AS total\_revenue,

AVG(REVENUE) AS average\_revenue

FROM sales\_order;

**5. How to handle NULLs in aggregates?**

In SQL:

* **COUNT()** ignores NULL values, meaning it counts only non-NULL values.
* **SUM(), AVG(), MIN(), MAX()** also ignore NULL values.

However, you can **handle NULLs explicitly** using **COALESCE()** or **IFNULL()** to replace NULL with a default value.

**Example: Replace NULL with 0 in SUM() and AVG():**

SELECT

SUM(COALESCE(REVENUE, 0)) AS total\_revenue,

AVG(COALESCE(REVENUE, 0)) AS average\_revenue

FROM sales\_order;

**Example: Counting NULL values explicitly:**

SELECT

COUNT(CASE WHEN REVENUE IS NULL THEN 1 END) AS null\_count

FROM sales\_order;

**6. What’s the role of ORDER BY and GROUP BY?**

* **GROUP BY**: Used to group rows based on a specific column (or columns). It's often used with aggregate functions like SUM(), COUNT(), AVG() to calculate values for each group.
* **ORDER BY**: Used to sort the result set in ascending (ASC) or descending (DESC) order based on one or more columns. It doesn't affect grouping but is used to arrange the output.

**Example:**

SELECT

CLIENTNO,

SUM(REVENUE) AS total\_revenue

FROM sales\_order

GROUP BY CLIENTNO

ORDER BY total\_revenue DESC;

**7. How do you get the top 3 months by sales?**

To get the **top 3 months by sales**, you can calculate the monthly revenue first, then sort it in descending order and limit the result.

**Example (PostgreSQL / Standard SQL):**

SELECT

EXTRACT(YEAR FROM ORDERDATE) AS order\_year,

EXTRACT(MONTH FROM ORDERDATE) AS order\_month,

SUM(REVENUE) AS monthly\_revenue

FROM sales\_order

GROUP BY

EXTRACT(YEAR FROM ORDERDATE),

EXTRACT(MONTH FROM ORDERDATE)

ORDER BY monthly\_revenue DESC

LIMIT 3;

**Example (MySQL):**

SELECT

YEAR(ORDERDATE) AS order\_year,

MONTH(ORDERDATE) AS order\_month,

SUM(REVENUE) AS monthly\_revenue

FROM sales\_order

GROUP BY

YEAR(ORDERDATE),

MONTH(ORDERDATE)

ORDER BY monthly\_revenue DESC

LIMIT 3;