Data Analyst

Most Asked Interview Questions

SQL

Here are Top 10 Advanced SQL Interview Questions with real business scenarios that interviewers frequently ask, especially for data analyst and data engineering roles. Each question focuses on writing SQL queries, not just theory.

---

1. Find the Second Highest Salary

Scenario: Your company wants to know who earns the second-highest salary in the employees table.

Table: employees(emp_id, name, salary)

Question:

```
SELECT MAX(salary) AS second_highest_salary
FROM employees
WHERE salary < (SELECT MAX(salary) FROM employees);
```

---

2. Get the Top 3 Products by Sales in Each Category

Scenario: Your manager asks for the top 3 best-selling products in each category.

Table: sales(product_id, category, sales_amount)

Question: Use ROW_NUMBER() window function.

```
SELECT *
FROM (
    SELECT product_id, category, sales_amount,
           ROW_NUMBER() OVER (PARTITION BY category ORDER BY
sales_amount DESC) AS rn
    FROM sales
) ranked
WHERE rn <= 3;
```

---

3. Identify Customers Who Purchased in All Months of 2024

Scenario: Marketing wants a list of loyal customers.

Table: orders(customer_id, order_date)

Question:

```
SELECT customer_id
FROM (
    SELECT customer_id, COUNT(DISTINCT MONTH(order_date)) AS
month_count
    FROM orders
    WHERE YEAR(order_date) = 2024
    GROUP BY customer_id
) monthly_orders
WHERE month_count = 12;
```

---

4. Detect Duplicate Records in a Table

Scenario: Data quality check is needed.

Table: users(email, name)

Question:

```
SELECT email, COUNT(*)
FROM users
GROUP BY email
HAVING COUNT(*) > 1;
```

---

5. Calculate the Running Total of Sales per Month

Scenario: Your finance team needs a cumulative revenue report.

Table: sales(order_date, amount)

Question:

```
SELECT order_date,
       SUM(amount) OVER (ORDER BY order_date ROWS BETWEEN
UNBOUNDED PRECEDING AND CURRENT ROW) AS running_total
```

```
FROM sales;
```

---

6. Find Products That Have Never Been Ordered

Scenario: Inventory team wants to clean up old listings.

Tables:

- products(product_id, product_name)
- orders(order_id, product_id)

  Question:

```
SELECT p.product_id, p.product_name
FROM products p
LEFT JOIN orders o ON p.product_id = o.product_id
WHERE o.product_id IS NULL;
```

---

7. Find the Longest Sequence of Consecutive Login Days per User

Scenario: Engagement team wants to reward consistent users.

Table: logins(user_id, login_date)

Challenge: Use window functions and date arithmetic.

(Advanced query – ask if you want a full solution.)

---

8. Calculate Year-over-Year Growth in Sales

Scenario: Executives want to compare performance with last year.

Table: sales(order_date, revenue)

Question:

```
SELECT YEAR(order_date) AS year,
       SUM(revenue) AS total_revenue,
       LAG(SUM(revenue)) OVER (ORDER BY YEAR(order_date)) AS
prev_year_revenue,
       ROUND((SUM(revenue) - LAG(SUM(revenue)) OVER (ORDER BY
YEAR(order_date))) /
             LAG(SUM(revenue)) OVER (ORDER BY YEAR(order_date)) * 100, 2)
AS YoY_growth
```

```
FROM sales
GROUP BY YEAR(order_date);
```

---

9. Identify Employees Who Earn More Than the Average Salary in Their Department

Scenario: HR wants to review salary structure.

Table: employees(emp_id, name, dept_id, salary)

Question:

```
SELECT *
FROM employees e
WHERE salary > (
    SELECT AVG(salary)
    FROM employees
    WHERE dept_id = e.dept_id
);
```

---

10. Rank Customers Based on Their Total Spend

Scenario: Loyalty program is being redesigned.

Table: orders(customer_id, order_amount)

Question:

```
SELECT customer_id,
       SUM(order_amount) AS total_spent,
       RANK() OVER (ORDER BY SUM(order_amount) DESC) AS
spending_rank
FROM orders
GROUP BY customer_id;
```

(Dipankar pal)
(www.linkedin.com/in/dipankar-data-analyst)

```sql
SELECT MAX(salary) AS second_highest_salary
FROM employees
WHERE salary < (SELECT MAX(salary) FROM employees);
```

```sql
SELECT *
FROM (
    SELECT product_id, category, sales_amount,
           ROW_NUMBER() OVER (PARTITION BY category ORDER BY sales_amount DESC) AS rn
    FROM sales
) ranked
WHERE rn <= 3;
```

```sql
SELECT customer_id
FROM (
    SELECT customer_id, COUNT(DISTINCT MONTH(order_date)) AS month_count
    FROM orders
    WHERE YEAR(order_date) = 2024
    GROUP BY customer_id
) monthly_orders
WHERE month_count = 12;
```

```sql
SELECT email, COUNT(*)
FROM users
GROUP BY email
HAVING COUNT(*) > 1;
```

```sql
SELECT order_date,
       SUM(amount) OVER (ORDER BY order_date ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) AS running_total
FROM sales;
```

```sql
SELECT p.product_id, p.product_name
FROM products p
LEFT JOIN orders o ON p.product_id = o.product_id
WHERE o.product_id IS NULL;
```

```sql
SELECT YEAR(order_date) AS year,
       SUM(revenue) AS total_revenue,
       LAG(SUM(revenue)) OVER (ORDER BY YEAR(order_date)) AS prev_year_revenue,
       ROUND((SUM(revenue) - LAG(SUM(revenue)) OVER (ORDER BY YEAR(order_date))) /
             LAG(SUM(revenue)) OVER (ORDER BY YEAR(order_date)) * 100, 2) AS YoY_growth
FROM sales
GROUP BY YEAR(order_date);
```

```sql
SELECT *
FROM employees e
WHERE salary > (
    SELECT AVG(salary)
    FROM employees
    WHERE dept_id = e.dept_id
);
```

```sql
SELECT customer_id,
       SUM(order_amount) AS total_spent,
       RANK() OVER (ORDER BY SUM(order_amount) DESC) AS spending_rank
FROM orders
GROUP BY customer_id;
```

(Dipankar pal)
(www.linkedin.com/in/dipankar-data-analyst)