

10 advanced
SQL interview
practical
query
questions
along with
their solutions

1. Question: Retrieve the top 5 highest-paid employees for each department, sorted by salary in descending order.

Solution

```
1  SELECT
2      department,
3      employee_name,
4      salary
5  FROM
6      (
7          SELECT
8              department,
9              employee_name,
10             salary,
11             ROW_NUMBER() OVER (
12                 PARTITION BY department
13                 ORDER BY
14                     salary DESC
15             ) AS rank
16         FROM
17             employees
18     ) ranked
19 WHERE
20     rank <= 5;
```

2. Question: Calculate the total sales for each month of the current year, including months with zero sales.

```
1  SELECT
2      to_char(sale_date, 'YYYY-MM') AS month,
3      COALESCE(
4          SUM(sales_amount),
5          0
6      ) AS total_sales
7  FROM
8      generate_series(
9          DATE_TRUNC('YEAR', CURRENT_DATE),
10         DATE_TRUNC('YEAR', CURRENT_DATE) + INTERVAL '1 year' - INTERVAL '1 day',
11         INTERVAL '1 month'
12     ) AS months(sale_date)
13  LEFT JOIN sales ON to_char(sale_date, 'YYYY-MM') = to_char(sales_date, 'YYYY-MM')
14  GROUP BY
15      month;
```

3. Question: Find customers who have made a purchase every month for the last six months.

```
1 SELECT
2     customer_id
3 FROM
4     customers
5 WHERE
6     date_trunc('month', CURRENT_DATE) - INTERVAL '6 months' <= ALL (
7     SELECT
8         date_trunc('month', purchase_date)
9     FROM
10        purchases
11    WHERE
12        customer_id = customers.customer_id
13 );
```

4. Question: Calculate the running total of sales for each day within the past month.

```
SELECT
    date,
    SUM(sales_amount) OVER (
        ORDER BY
            date
    ) AS running_total
FROM
    generate_series(
        DATE_TRUNC('MONTH', CURRENT_DATE) - INTERVAL '1 month',
        DATE_TRUNC('MONTH', CURRENT_DATE) - INTERVAL '1 day',
        INTERVAL '1 day'
    ) AS dates(date)
LEFT JOIN sales ON dates.date = sales.sales_date;
```

5. Question: List the products that have been sold in all cities where the company operates.

```
1 SELECT
2   product_id,
3   product_name
4 FROM
5   products
6 WHERE
7   product_id NOT IN (
8     SELECT
9       DISTINCT product_id
10    FROM
11      sales
12   WHERE
13     city NOT IN (
14       SELECT
15         DISTINCT city
16      FROM
17        locations
18    )
19 );
```

6. Question: Retrieve the top 10 customers who have spent the most on their single purchase.

```
1 SELECT
2   customer_id,
3   MAX(purchase_amount) AS max_purchase_amount
4 FROM
5   purchases
6 GROUP BY
7   customer_id
8 ORDER BY
9   max_purchase_amount DESC
10 LIMIT
11   10;
```

7. Question: Find the employees who manage the same number of employees as their manager.

```
1 SELECT
2     e1.employee_name AS employee,
3     e1.managed_count AS direct_reports
4 FROM
5     employees e1
6     JOIN employees e2 ON e1.manager_id = e2.employee_id
7 WHERE
8     e1.managed_count = e2.managed_count;
```

8. Question: Calculate the 30-day moving average of sales for each product.

```
1 SELECT
2     product_id,
3     sales_date,
4     sales_amount,
5     AVG(sales_amount) OVER (
6         PARTITION BY product_id
7         ORDER BY
8             sales_date RANGE BETWEEN INTERVAL '30 days' PRECEDING
9             AND CURRENT ROW
10    ) AS moving_avg
11 FROM
12     sales;
```

9. Question: List the departments where the average salary is higher than the company's overall average salary.

```
1 SELECT
2   department
3 FROM
4   employees
5 GROUP BY
6   department
7 HAVING
8   AVG(salary) > (
9     SELECT
10      AVG(salary)
11     FROM
12      employees
13   );
```

10. Question: Retrieve the top 3 most recent orders for each customer.

```
1 SELECT
2   customer_id,
3   order_id,
4   order_date
5 FROM
6   (
7     SELECT
8       customer_id,
9       order_id,
10      order_date,
11      ROW_NUMBER() OVER (
12        PARTITION BY customer_id
13        ORDER BY
14          order_date DESC
15      ) AS rank
16    FROM
17      orders
18  ) ranked
19 WHERE
20   rank <= 3;
```