

Sql Interview Queries Part 4

Retrieve all employees and their respective managers.

```
SELECT e1.name AS Employee, e2.name AS Manager
FROM employees e1
LEFT JOIN employees e2 ON e1.manager_id = e2.employee_id;
```

Find customers who haven't made any purchases in the last year.

```
SELECT * FROM customers
LEFT JOIN orders ON customers.customer_id = orders.customer_id
WHERE orders.order_date < DATE_SUB(CURDATE(), INTERVAL 1 YEAR) OR orders.order_date IS NULL;
```

Get the top product sold in each category.

```
SELECT categories.category_name, products.product_name, SUM(order_details.quantity) as total_sold
FROM products
JOIN categories ON products.category_id = categories.category_id
JOIN order_details ON products.product_id = order_details.product_id
GROUP BY categories.category_name
ORDER BY total_sold DESC;
```

List all suppliers and the total products supplied by them.

```
SELECT suppliers.supplier_name, COUNT(products.product_id) as total_products
FROM suppliers
LEFT JOIN products ON suppliers.supplier_id = products.supplier_id
GROUP BY suppliers.supplier_name;
```

Find employees who have sold products of a total value greater than \$10,000.

```
SELECT employees.name, SUM(order_details.quantity * products.price) as total_sales
FROM employees
JOIN orders ON employees.employee_id = orders.employee_id
JOIN order_details ON orders.order_id = order_details.order_id
JOIN products ON order_details.product_id = products.product_id
GROUP BY employees.name
HAVING total_sales > 10000;
```

Retrieve customers who have bought the same products as customer with ID '5'.

```
SELECT DISTINCT c.customer_id, c.customer_name
FROM customers c
JOIN orders o ON c.customer_id = o.customer_id
JOIN order_details od ON o.order_id = od.order_id
```

**WHERE od.product_id IN (SELECT od2.product_id FROM order_details od2 JOIN orders o2 ON
od2.order_id = o2.order_id WHERE o2.customer_id = 5)
AND c.customer_id != 5;**

List products that have been bought together more than 3 times.

**SELECT od1.product_id AS product1, od2.product_id AS product2, COUNT(*) as frequency
FROM order_details od1
JOIN order_details od2 ON od1.order_id = od2.order_id AND od1.product_id < od2.product_id
GROUP BY od1.product_id, od2.product_id
HAVING frequency > 3;**

Find the departments where the average salary is higher than the company's overall average salary.

**SELECT department, AVG(salary) as avg_dept_salary
FROM employees
GROUP BY department
HAVING avg_dept_salary > (SELECT AVG(salary) FROM employees);**

Retrieve products that are supplied by more than one supplier.

**SELECT product_name, COUNT(DISTINCT supplier_id) as suppliers_count
FROM products
GROUP BY product_name
HAVING suppliers_count > 1;**

List all employees who have more sales than their manager.

**SELECT e1.name AS employee_name, e2.name AS manager_name
FROM employees e1
JOIN employees e2 ON e1.manager_id = e2.employee_id
JOIN orders o1 ON e1.employee_id = o1.employee_id
JOIN orders o2 ON e2.employee_id = o2.employee_id
WHERE SUM(o1.total_amount) > SUM(o2.total_amount);**

Retrieve all products and their supplier, including products without suppliers.

**SELECT products.product_name, suppliers.supplier_name
FROM products
LEFT JOIN suppliers ON products.supplier_id = suppliers.supplier_id;**

List employees and their latest order, including employees who haven't made any orders.

**SELECT employees.name, MAX(orders.order_date) as latest_order_date
FROM employees
LEFT JOIN orders ON employees.employee_id = orders.employee_id
GROUP BY employees.name;**

Find customers who haven't made a purchase in a specific product category (e.g., "Electronics").

```
SELECT customers.customer_name
FROM customers
LEFT JOIN orders ON customers.customer_id = orders.customer_id
LEFT JOIN order_details ON orders.order_id = order_details.order_id
LEFT JOIN products ON order_details.product_id = products.product_id
WHERE products.category = 'Electronics' AND orders.order_id IS NULL;
```

Get the total sales of each product, including products that haven't been sold.

```
SELECT products.product_name, COALESCE(SUM(order_details.quantity), 0) as total_sales
FROM products
LEFT JOIN order_details ON products.product_id = order_details.product_id
GROUP BY products.product_name;
```

List employees who don't have a manager.

```
SELECT employees.name
FROM employees
LEFT JOIN employees_managers ON employees.manager_id = managers.employee_id
WHERE managers.employee_id IS NULL;
```

Retrieve categories and the number of products in each, including categories with no products.

```
SELECT categories.category_name, COALESCE(COUNT(products.product_id), 0) as product_count
FROM categories
LEFT JOIN products ON categories.category_id = products.category_id
GROUP BY categories.category_name;
```

Identify products that have never been in an order placed in the month of December.

```
SELECT products.product_name
FROM products
LEFT JOIN order_details ON products.product_id = order_details.product_id
LEFT JOIN orders ON order_details.order_id = orders.order_id AND MONTH(orders.order_date) = 12
WHERE orders.order_id IS NULL;
```

List suppliers and the average price of their supplied products, including suppliers who haven't supplied any products.

```
SELECT suppliers.supplier_name, COALESCE(AVG(products.price), 0) as avg_product_price
FROM suppliers
LEFT JOIN products ON suppliers.supplier_id = products.supplier_id
GROUP BY suppliers.supplier_name;
```

Retrieve departments and their total employees, including departments without any employees.

```
SELECT departments.department_name, COALESCE(COUNT(employees.employee_id), 0) as  
employee_count  
FROM departments  
LEFT JOIN employees ON departments.department_id = employees.department_id  
GROUP BY departments.department_name;
```

Find all customers who haven't placed any orders in the last six months.

```
SELECT customers.customer_name  
FROM customers  
LEFT JOIN orders ON customers.customer_id = orders.customer_id AND orders.order_date >  
DATE_SUB(CURDATE(), INTERVAL 6 MONTH)  
WHERE orders.order_id IS NULL;
```