

SQL Report Queries



give me the sql for:

Write a SQL statement to make a report with customer name, city, order number, order date, order amount salesman name and commission to find that either any of the existing customers have placed no order or placed one or more orders by their salesman or by own.

Write a SQL statement to make a list in ascending order for the salesmen who works either for one or more customer or not yet join under any of the customers.

From the following tables write a SQL query to list all salespersons along with customer name, city, grade, order number, date, and amount.

Write a SQL statement to make a list for the salesmen who either work for one or more customers or yet to join any of the customer. The customer may have placed, either one or more orders on or above order amount 2000 and must have a grade, or he may not have placed any order to the associated supplier.

Write a SQL statement to make a report with customer name, city, order no. order date, purchase amount for those customers from the existing list who placed one or more orders or which order(s) have been placed by the customer who is not on the list.

Write a SQL statement to make a report with customer name, city, order no. order date, purchase amount for only those customers on the list who must have a grade and placed one or more orders or which order(s) have been placed by the customer who is neither in the list nor have a grade.

Here are the SQL queries for each of the problems you've listed. These queries are based on commonly used table structures like `customer`, `salesman`, and `orders`. If your actual table schema differs, you may need to adjust the table or column names accordingly.

1. Customers with or without orders (with salesman details)

```
sql

SELECT c.cust_name, c.city, o.ord_no, o.ord_date, o.purch_amt,
       s.name AS salesman_name, s.commission
FROM customer c
LEFT JOIN orders o ON c.customer_id = o.customer_id
LEFT JOIN salesman s ON c.salesman_id = s.salesman_id;
```

2. Salesmen with or without customers, in ascending order

```
sql

SELECT DISTINCT s.salesman_id, s.name, s.city, s.commission
FROM salesman s
LEFT JOIN customer c ON s.salesman_id = c.salesman_id
ORDER BY s.name ASC;
```

3. All salespersons with customer, order, and grade details

```
sql

SELECT s.name AS salesman_name, c.cust_name, c.city, c.grade,
       o.ord_no, o.ord_date, o.purch_amt
FROM salesman s
LEFT JOIN customer c ON s.salesman_id = c.salesman_id
LEFT JOIN orders o ON c.customer_id = o.customer_id;
```

4. Salesmen with customers who have either (a) placed orders ≥ 2000 and have a grade, or (b) no orders

```
sql

SELECT DISTINCT s.salesman_id, s.name
FROM salesman s
LEFT JOIN customer c ON s.salesman_id = c.salesman_id
LEFT JOIN orders o ON c.customer_id = o.customer_id
WHERE
    (o.purch_amt >= 2000 AND c.grade IS NOT NULL)
    OR o.ord_no IS NULL;
```

5. Orders placed by listed customers or unlisted customers

```
sql

SELECT c.cust_name, c.city, o.ord_no, o.ord_date, o.purch_amt
FROM orders o
LEFT JOIN customer c ON o.customer_id = c.customer_id;
```

6. Orders by customers who have a grade or who are not listed and have no grade

```
sql

SELECT c.cust_name, c.city, o.ord_no, o.ord_date, o.purch_amt
FROM orders o
LEFT JOIN customer c ON o.customer_id = c.customer_id
WHERE
    (c.customer_id IS NOT NULL AND c.grade IS NOT NULL)
    OR (c.customer_id IS NULL OR c.grade IS NULL);
```

Would you like a diagram or schema reference to better visualize these relationships?



explain the questions

Certainly! Here's an explanation of each question in simple terms so you understand **what is being asked** and **why the SQL behaves the way it does**.

1. Customers with or without orders (with salesman details)

Question Summary:

Make a report showing:

- Customer details (name, city)
- Their order details (order number, date, amount)
- Salesman who handled them (name and commission)

Logic:

Some customers **may have placed orders**, while others **may not have**. We use a **LEFT JOIN** from ``customer`` to ``orders`` to include customers who didn't place any order.

2. Salesmen with or without customers (sorted)

Question Summary:

List all **salesmen** whether:

- They have **at least one customer** working with them, or
- They **have no customers** yet.

Logic:

This is a **LEFT JOIN** from ``salesman`` to ``customer``. It ensures even salesmen with no customers show up. We then sort the list **in ascending order** by name.

3. All salespersons with customer, order, and grade details

Question Summary:

List:

- All salespersons
- The customers they serve (if any)

- Customer details: city, grade
- Order details: number, date, amount

Logic:

You want a **full view** of the entire chain: **salesman** → **customer** → **order**. Use multiple **LEFT JOINS** so people/customers with no orders still show up.

4. Salesmen with customers who either:

- Placed orders **on or above 2000** and have a **grade**, or
- Have **not placed any order at all**

Question Summary:

You are asked to **filter** only those salesmen who have customers:

- With **substantial orders (≥2000)** and a **grade**
- Or with **no orders** placed

Logic:

This is a **conditional check** using ``WHERE``:

- One part checks for large purchases **and** grade
 - The other checks for **absence of orders**
-

5. Orders by customers on the list or from unknown customers

Question Summary:

List all order info for:

- Customers who are in your ``customer`` table and **placed orders**
- Or for **orders placed by someone not listed** in the ``customer`` table

Logic:

This question is saying:

Include both types: those with customer records and those without.

So we do a **LEFT JOIN** of ``orders`` with ``customer`` and just select everything.

6. Orders placed by customers who:

- Are in the list **and have a grade**, or
- Are **not in the list** and **have no grade**

Question Summary:

This is more specific than #5. It asks for **only**:

- Orders by **known customers who have a grade**
- OR
- Orders by **unknown customers** or those **without a grade**

Logic:

We use a **LEFT JOIN** again and then a ``WHERE`` clause that filters for:

- ``(customer exists AND has grade)``
- OR ``(customer doesn't exist OR has no grade)``

It's like saying:

Show me **valid or fallback cases** only.

Would you like me to sketch an ER diagram (Entity-Relationship) to make these relationships even clearer?