**Practical-9**

**Aim:** To study and execute various JOIN commands to perform data retrieval and

manipulation from Salespeople, Customer, and Order tables based on specific

requirements.

**Theoratical Description:-**

Constraints:

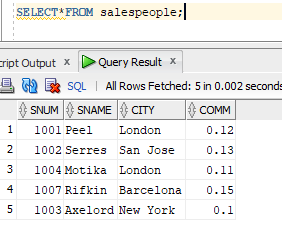
1. Primary Key - Ensures unique identification of records.
2. Foreign Key - Maintains referential integrity.
3. Not Null - Prevents essential fields from being empty.

Key Queries & Expected Results:

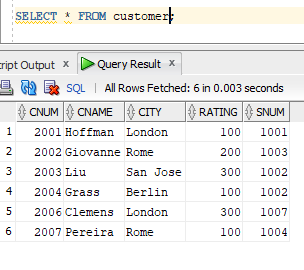
* Retrieve customers serviced by specific salespeople.
* Filter orders based on amount constraints.
* Identify the largest orders for salespeople with high order values.
* Match salespeople and customers based on city.
* List orders with corresponding customer details.
* Retrieve customers served by salespeople with high commission rates.
* Compare customer ratings and find counts above the average.
* Compute total sales per salesperson exceeding the highest single order.
* Categorize customers into "High" or "Low" rating groups using UNION.

Different Tables are Given below

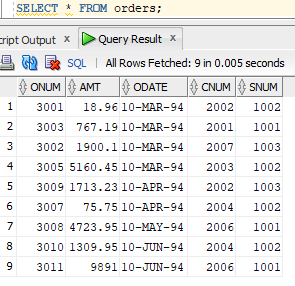
Salespeople:-



customer



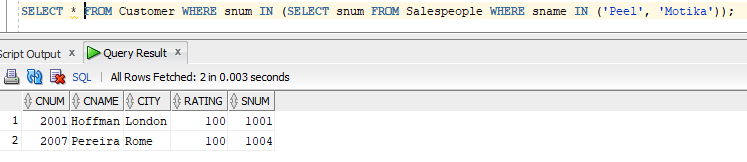
orders



**Query-1:** All customers serviced by Peel or Motika

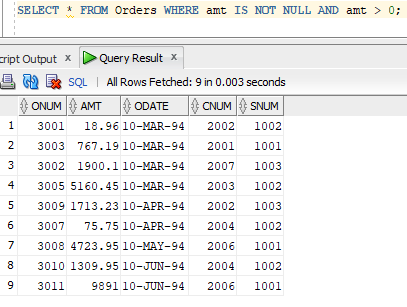
**SQL Statement:**SELECT \* FROM Customer WHERE snum IN (SELECT snum FROM Salespeople WHERE sname IN ('Peel', 'Motika'));

**Output:**



**Query-2:** All orders except those with 0 or null value in the amt field

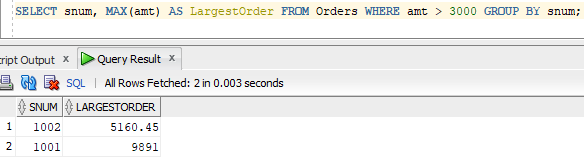
**SQL Statement:**SELECT \* FROM Orders WHERE amt IS NOT NULL AND amt > 0;

 **Output:**

**Query-3:** Largest order taken by each sales order value of more than 3000

**SQL Statement:**SELECT snum, MAX(amt) AS LargestOrder FROM Orders WHERE amt > 3000 GROUP BY snum;

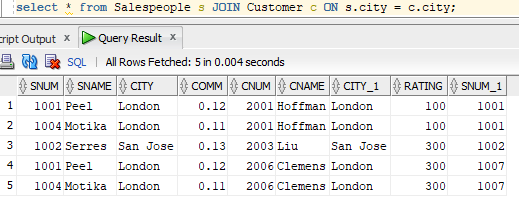
**Output:**



**Query-4:** All combinations of salespeople and customers who belong to the same city

**SQL Statement:**SELECT s.sname, c.cname, s.city FROM Salespeople s JOIN Customer c ON s.city = c.city;

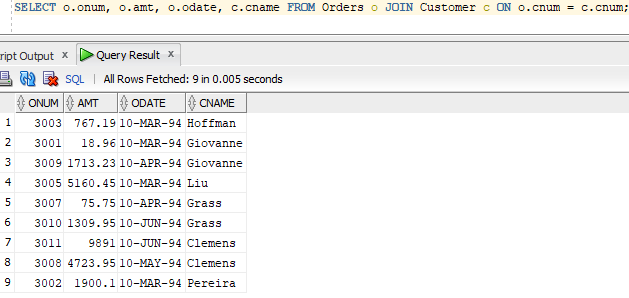
**Output:**



**Query-5:** List each order with the name of the customer who placed the order

**SQL Statement:**SELECT o.onum, o.amt, o.odate, c.cname FROM Orders o JOIN Customer c ON o.cnum = c.cnum;

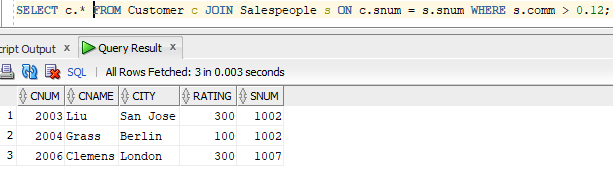
**Output:**



**Query-6:** Produce a listing of all the customers serviced by salespeople having a commission of more than 12%

**SQL Statement:**SELECT c.\* FROM Customer c JOIN Salespeople s ON c.snum = s.snum WHERE s.comm > 0.12;

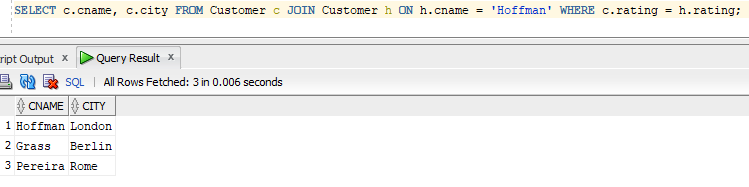
**Output:**



**Query-7:** Produce names and cities of all customers with the same rating as Hoffman

**SQL Statement:**SELECT c.cname, c.city FROM Customer c JOIN Customer h ON h.cname = 'Hoffman' WHERE c.rating = h.rating;

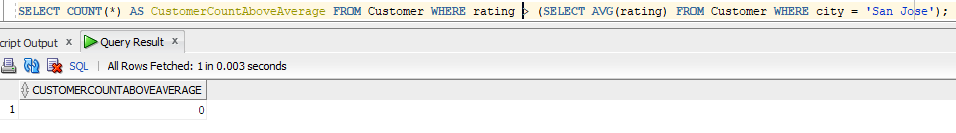
**Output:**



**Query-8:** Count the customers with ratings above San Jose’s average

**SQL Statement:**SELECT COUNT(\*) AS CustomerCountAboveAverage FROM Customer WHERE rating > (SELECT AVG(rating) FROM Customer WHERE city = 'San Jose');

**Output:**

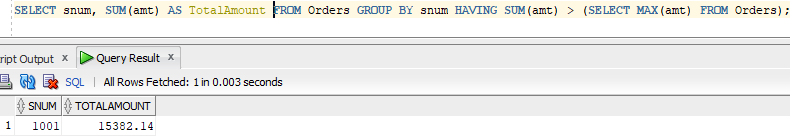


**Query-9:** Find the total amount in orders for each salesperson for whom this total is greater than the

amount of the largest order in the order table

**SQL Statement:**SELECT snum, SUM(amt) AS TotalAmount FROM Orders GROUP BY snum HAVING SUM(amt) > (SELECT MAX(amt) FROM Orders);

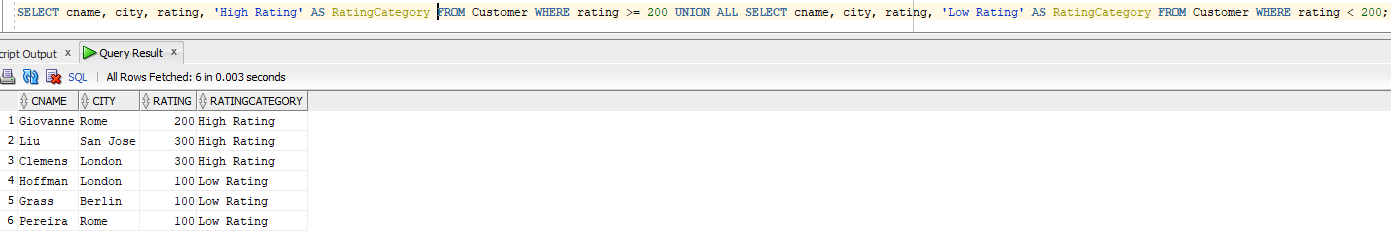
**Output:**



**Query-10:** Create a union of two queries that shows all customers' names, cities, and ratings. Those with a rating of 200 or greater will have the words ‘High Rating’ while others will have ‘Low Rating’.

**SQL Statement:**SELECT cname, city, rating, 'High Rating' AS RatingCategory FROM Customer WHERE rating >= 200 UNION ALL SELECT cname, city, rating, 'Low Rating' AS RatingCategory FROM Customer WHERE rating < 200;

**Output:**



Questions:-

**1)What is the purpose of using the JOIN clause in SQL queries?  
Answer:** The JOIN clause is used to combine rows from two or more tables based on a related column. For example, in Query 5, JOIN is used to retrieve customer names for each order by linking the NewOrder and NewCustomer tables through the cnum column.

**2)What is the difference between HAVING and WHERE in SQL?  
Answer:** WHERE filters rows before grouping, while HAVING filters groups after aggregation. For instance, in Query 9, HAVING is used to filter salespeople whose total order amount exceeds the largest single order.

**3)What is the role of the UNION operator in SQL?  
Answer:** The UNION operator is used to combine the results of two or more queries while ensuring unique records. In Query 10, UNION is used to classify customers into 'High Rating' and 'Low Rating' categories based on their rating.