DBMS Queries

1. Produce the order no, amount, and date of all orders.

SELECT ONUM, AMOUNT, ODATE

FROM Orders;

Give all the information about all the customers with salesman number 1001.

SELECT \*

FROM Customers

WHERE SNUM = 1001;

Display the following information in the order of city, sname, snum, and commission.

SELECT CITY, SNAME, SNUM, COMMISSION

FROM Salesmen

ORDER BY CITY, SNAME, SNUM, COMMISSION;

List the rating followed by the name of each customer in Surat.

SELECT RATING, CNAME

FROM Customers

WHERE CITY = 'Surat'

ORDER BY RATING;

1. List the snum of all salesmen with orders in the order table without any duplicates.

SELECT DISTINCT SNUM

FROM Orders;

1. List all orders for more than Rs. 1000.

SELECT \*

FROM Orders

WHERE AMOUNT > 1000;

1. List the names and cities of all salesmen in London with a commission above 10%.

SELECT SNAME, CITY

FROM Salesmen

WHERE CITY = 'London' AND COMMISSION > 10;

1. List all customers whose names begin with the letter 'C'.

SELECT \*

FROM Customers

WHERE CNAME LIKE 'C%';

1. List all customers whose names begin with the letter 'A' to 'G'.

SELECT \*

FROM Customers

WHERE CNAME BETWEEN 'A' AND 'G%';

1. List all orders with zero or NULL amount.

SELECT \*

FROM Orders

WHERE AMOUNT = 0 OR AMOUNT IS NULL;

1. Find out the largest orders of salesman 1002 and 1007.

SELECT \*

FROM Orders

WHERE SNUM IN (1002, 1007)

ORDER BY AMOUNT DESC

LIMIT 1;

1. Count all orders of October 3, 1997.

SELECT COUNT(\*)

FROM Orders

WHERE ODATE = '10/03/97';

1. Calculate the total amount ordered.

SELECT SUM(AMOUNT) AS TotalAmount

FROM Orders;

1. Calculate the average amount ordered.

SELECT AVG(AMOUNT) AS AverageAmount

FROM Orders;

1. Count the number of salesmen currently having orders.

SELECT COUNT(DISTINCT SNUM)

FROM Orders;

1. List all salesmen with their % of commission.

SELECT SNAME, COMMISSION

FROM Salesmen;

1. Assume each salesperson has a 12% commission. Write a query on the order table that will produce the order number, salesman number, and the amount of commission for that order.

SELECT ONUM, SNUM, AMOUNT \* 0.12 AS Commission

FROM Orders;

1. Find the highest rating in each city in the form: For the city (city), the highest rating is: (rating).

SELECT CITY, MAX(RATING) AS HighestRating

FROM Customers

GROUP BY CITY;

1. List all in descending order of rating.

SELECT \*

FROM Customers

ORDER BY RATING DESC;

1. Calculate the total of orders for each day and place the result in descending order.

SELECT ODATE, SUM(AMOUNT) AS TotalAmount

FROM Orders

GROUP BY ODATE

ORDER BY TotalAmount DESC;

1. Show the name of all customers with their salesman's name.

SELECT C.CNAME, S.SNAME

FROM Customers C

JOIN Salesmen S ON C.SNUM = S.SNUM;

1. List all customers and salesmen who share the same city.

SELECT C.CNAME, C.CITY, S.SNAME, S.CITY

FROM Customers C

JOIN Salesmen S ON C.CITY = S.CITY;

1. List all orders with the names of their customer and salesman.

SELECT O.\*, C.CNAME, S.SNAME

FROM Orders O

JOIN Customers C ON O.CNUM = C.CNUM

JOIN Salesmen S ON O.SNUM = S.SNUM;

1. List all orders by the customers not located in the same city as their salesman.

SELECT O.\*

FROM Orders O

JOIN Customers C ON O.CNUM = C.CNUM

JOIN Salesmen S ON O.SNUM = S.SNUM

WHERE C.CITY <> S.CITY;

1. List all customers serviced by salespeople with a commission above 12%.

SELECT C.\*

FROM Customers C

JOIN Salesmen S ON C.SNUM = S.SNUM

WHERE S.COMMISSION > 12;

26.Calculate the amount of the salesman's commission on each order by a customer with a rating above 100

SELECT O.ONUM, O.AMOUNT \* S.COMMISSION AS CommissionAmount

FROM Orders O

JOIN Customers C ON O.CNUM = C.CNUM

JOIN Salesmen S ON C.SNUM = S.SNUM

WHERE C.RATING > 100;

1. Find all pairs of customers having the same rating without duplication.

SELECT C1.CNAME, C2.CNAME, C1.RATING

FROM Customers C1, Customers C2

WHERE C1.RATING = C2.RATING AND C1.CNUM <> C2.CNUM;

1. List all orders that are greater than the average of October 4, 1997.

SELECT \*

FROM Orders

WHERE AMOUNT > (

SELECT AVG(AMOUNT)

FROM Orders

WHERE ODATE = '10/04/97'

);

1. Find the average commission of salesmen in London.

SELECT AVG(COMMISSION) AS AverageCommission

FROM Salesmen

WHERE CITY = 'London';

1. Find all orders attributed to salesmen in 'London' using both the subquery and join methods.

-- Subquery method

SELECT \*

FROM Orders

WHERE SNUM IN (

SELECT SNUM

FROM Salesmen

WHERE CITY = 'London'

);

-- Join method

SELECT O.\*

FROM Orders O

JOIN Salesmen S ON O.SNUM = S.SNUM

WHERE S.CITY = 'London';

1. List the commission of all salesmen serving customers in 'London'.

SELECT S.COMMISSION

FROM Salesmen S

JOIN Customers C ON S.SNUM = C.SNUM

WHERE C.CITY = 'London';

1. Find all customers whose cnum is 1000 above the snum of Sejal.

SELECT \*

FROM Customers

WHERE CNUM > (

SELECT SNUM + 1000

FROM Salesmen

WHERE SNAME = 'Sejal'

);

1. Count the number of customers with a rating above the average of 'Surat'.

SELECT COUNT(\*)

FROM Customers

WHERE RATING > (

SELECT AVG(RATING)

FROM Customers

WHERE CITY = 'Surat'

);

1. Find all salesmen with customers located in their cities using ANY and IN.

SELECT \*

FROM Salesmen

WHERE SNUM = ANY (

SELECT SNUM

FROM Customers

WHERE CITY = Salesmen.CITY

);

-- Alternative query using IN

SELECT \*

FROM Salesmen

WHERE SNUM IN (

SELECT SNUM

FROM Customers

WHERE CITY = Salesmen.CITY

);

1. Find all salesmen for whom there are customers that follow them in alphabetical order.

SELECT S.\*

FROM Salesmen S

WHERE EXISTS (

SELECT \*

FROM Customers C

WHERE C.SNUM = S.SNUM

ORDER BY C.CNAME

);

1. Find all customers having a rating greater than any customer in 'Rome'.

SELECT \*

FROM Customers

WHERE RATING > ANY (

SELECT RATING

FROM Customers

WHERE CITY = 'Rome'

);

1. List all orders that have an amount greater than at least one of the orders from October 6, 1997.

SELECT \*

FROM Orders

WHERE AMOUNT > ANY (

SELECT AMOUNT

FROM Orders

WHERE ODATE = '10/06/97'

);

1. Find all orders with amounts smaller than any amount for a customer in 'London'.

SELECT \*

FROM Orders

WHERE AMOUNT < ANY (

SELECT AMOUNT

FROM Orders O

JOIN Customers C ON O.CNUM = C.CNUM

WHERE C.CITY = 'London'

);

1. Find all customers who have a greater rating than every customer in 'Rome'.

SELECT \*

FROM Customers

WHERE RATING > ALL (

SELECT RATING

FROM Customers

WHERE CITY = 'Rome'

);

1. Create a union of two queries that shows the names, cities, and ratings of all customers. Those with a rating of >= 200 should display 'HIGH RATING' and those with < 200 should display 'LOW RATING'.

SELECT CNAME, CITY,

CASE

WHEN RATING >= 200 THEN 'HIGH RATING'

ELSE 'LOW RATING'

END AS RatingStatus

FROM Customers

UNION

SELECT SNAME, CITY, ''

FROM Salesmen;

1. Produce the name and number of each salesman and each customer with more than one current order in alphabetical order of names.

SELECT S.SNUM, S.SNAME

FROM Salesmen S

JOIN Orders O ON S.SNUM = O.SNUM

GROUP BY S.SNUM, S.SNAME

HAVING COUNT(\*) > 1

UNION

SELECT C.CNUM, C.CNAME

FROM Customers C

JOIN Orders O ON C.CNUM = O.CNUM

GROUP BY C.CNUM, C.CNAME

HAVING COUNT(\*) > 1

ORDER BY SNAME;

1. Create a union of three queries. First, select snum of all salesmen in Surat, second, the cnum of all customers in 'Surat', and third, the onum of all orders of October 3rd. Retain duplicates between the last two queries but remove the duplicates between either of them and the first.

(SELECT SNUM FROM Salesmen WHERE CITY = 'Surat')

UNION ALL

(SELECT CNUM FROM Customers WHERE CITY = 'Surat')

UNION

(SELECT ONUM FROM Orders WHERE ODATE = '10/03/97');

1. Remove all orders from customer Chirag from the orders table.

DELETE FROM Orders

WHERE CNUM = (

SELECT CNUM

FROM Customers

WHERE CNAME = 'Chirag'

);

44.Set the ratings of all the customers of Piyush to 400.

UPDATE customers

SET rating = 400

WHERE customer\_name = 'Piyush';

45.Increase the rating of all customers in Rome by 100

UPDATE customers

SET rating = rating + 100

WHERE city = 'Rome';