Assignment –7

Summarizing Data with Aggregate Functions.

1) Write a query that counts all orders for October 3.

```
SELECT COUNT(*)
FROM ORDERS
WHERE Odate = '1990-10-03';
```

2) Write a query that counts the number of different non-NULL city values in the Customers table.

```
SELECT COUNT(DISTINCT City)
FROM CUSTOMERS
WHERE City IS NOT NULL;
```

3) Write a query that selects each customer's smallest order.

```
SELECT Cnum, MIN(Amt)
FROM ORDERS
GROUP BY Cnum;
```

4) Write a query that selects the first customer, in alphabetical order, whose name begins with G.

```
SELECT Cname
FROM CUSTOMERS
WHERE Cname LIKE 'G%'
ORDER BY Cname
LIMIT 1;
```

5) Write a query that selects the highest rating in each city.

```
SELECT City, MAX(Rating)
FROM CUSTOMERS
GROUP BY City;
```

6) Write a query that counts the number of salespeople registering orders for each day. (If a salesperson has more than one order on a given day, he or she should be counted only once.).

```
SELECT Odate, COUNT(DISTINCT Snum)
FROM ORDERS
GROUP BY Odate;
```

Assignment – 8 Formatting Query output.

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

```
SELECT Onum, Snum, Amt * 0.12 AS Commission
FROM ORDERS;
```

2) Write a query on the Customers table that will find the highest rating in each city. Put the output in this form: For the city (city), the highest rating is : (rating).

```
SELECT City, CONCAT ('For the city ', City, ', the highest rating is: ', MAX(Rating)) AS Message
FROM CUSTOMERS
GROUP BY City;
```

3) Write a query that lists customers in descending order of rating. Output the rating field first, followed by the customer's name and number.

```
SELECT Rating, Cname, Cnum
FROM CUSTOMERS
ORDER BY Rating DESC;
```

4) Write a query that totals the orders for each day and places the results in descending order.

```
SELECT Odate, SUM(Amt) AS Total
FROM ORDERS
GROUP BY Odate
ORDER BY Total DESC;
```

Assignment – 9 Querying Multiple Tables at Once.

1) Write a query that lists each order number followed by the name of the customer who made the order.

```
SELECT ORDERS.Onum, CUSTOMERS.Cname
FROM ORDERS
JOIN CUSTOMERS ON ORDERS.Cnum = CUSTOMERS.Cnum;
```

2) Write a query that gives the names of both the salesperson and the customer for each order along with the order number.

```
SELECT ORDERS.Onum, SALESPEOPLE.Sname AS Salesperson, CUSTOMERS.Cname AS
Customer
FROM ORDERS
JOIN SALESPEOPLE ON ORDERS.Snum = SALESPEOPLE.Snum
JOIN CUSTOMERS ON ORDERS.Cnum = CUSTOMERS.Cnum;
```

3) Write a query that produces all customers serviced by salespeople with a commission above 12%. Output the customer's name, the salesperson's name, and the salesperson's rate of commission.

```
SELECT CUSTOMERS.Cname, SALESPEOPLE.Sname, SALESPEOPLE.Comm
FROM CUSTOMERS
JOIN SALESPEOPLE ON CUSTOMERS.Snum = SALESPEOPLE.Snum
WHERE SALESPEOPLE.Comm > 0.12;
```

4) Write a query that calculates the amount of the salesperson's commission on each order by a customer with a rating above 100.

```
SELECT ORDERS.Onum, SALESPEOPLE.Sname, CUSTOMERS.Cname, ORDERS.Amt *
SALESPEOPLE.Comm AS Commission
FROM ORDERS
JOIN CUSTOMERS ON ORDERS.Cnum = CUSTOMERS.Cnum
JOIN SALESPEOPLE ON ORDERS.Snum = SALESPEOPLE.Snum
WHERE CUSTOMERS.Rating > 100;
```

Assignment – 10 Joining a Table to Itself

1) Write a query that produces all pairs of salespeople who are living in the same city. Exclude combinations of salespeople with themselves as well as duplicate rows with the order reversed.

```
SELECT A.Sname, B.Sname

FROM SALESPEOPLE A

JOIN SALESPEOPLE B ON A.City = B.City AND A.Snum < B.Snum;
```

2) Write a query that produces the names and cities of all customers with the same rating as Hoffman.

```
SELECT Cname, City

FROM CUSTOMERS

WHERE Rating = (SELECT Rating FROM CUSTOMERS WHERE Cname = 'Hoffman');
```