

Assignment 1 Solution :-

1. Creating the Tables

A. Table for Salespeople:

```
CREATE TABLE Salespeople
(
    Snum INT(4),
    Sname VARCHAR(10),
    City VARCHAR(10),
    Comm FLOAT(3,2)
);
```

B. Table for Customers:

```
CREATE TABLE Customers
(
    Cnum INT(4),
    Cname VARCHAR(10),
    City VARCHAR(10),
    Rating INT(4),
    Snum INT(4)
);
```

C. Table for Orders:

```
CREATE TABLE Orders
(
    Onum INT(4),
    Amt FLOAT(7,2),
    Odate DATE,
    Cnum INT(4),
    Snum INT(4)
);
```

2. Inserting the Sample Data

A. Inserting into Salespeople Table:

```
INSERT INTO Salespeople (Snum, Sname, City, Comm) VALUES
(1001, 'Peel', 'London', 0.12),
(1002, 'Serres', 'San Jose', 0.13),
(1004, 'Motika', 'London', 0.11),
(1007, 'Rifkin', 'Barcelona', 0.15),
(1003, 'Axelrod', 'New York', 0.10);
```

B. Inserting into Customers Table:

```
INSERT INTO Customers (Cnum, Cname, City, Rating, Snum) VALUES
(2001, 'Hoffman', 'London', 100, 1001),
(2002, 'Giovanni', 'Rome', 200, 1003),
(2003, 'Liu', 'San Jose', 200, 1002),
(2004, 'Grass', 'Berlin', 300, 1002),
(2006, 'Clemens', 'London', 100, 1001),
(2008, 'Cisneros', 'San Jose', 300, 1007),
(2007, 'Pereira', 'Rome', 100, 1004);
```

C. Inserting into Orders Table:

```
INSERT INTO Orders (Onum, Amt, Odate, Cnum, Snum) VALUES
(3001, 18.69, '1990-10-03', 2008, 1007),
(3003, 767.19, '1990-10-03', 2001, 1001),
(3002, 1900.10, '1990-10-03', 2007, 1004),
(3005, 5160.45, '1990-10-03', 2003, 1002),
(3006, 1098.16, '1990-10-04', 2006, 1001),
(3009, 1713.23, '1990-10-04', 2004, 1004),
(3007, 75.75, '1990-10-04', 2004, 1004),
(3008, 4723.00, '1990-10-05', 2006, 1002),
(3010, 1309.95, '1990-10-06', 2002, 1003),
```

(3011, 9891.88, '1990-10-06', 2006, 1001);

C:\Users\prati>mysql -u pratik -p

Enter password: ***

Welcome to the MySQL monitor. Commands end with ; or \g.

Your MySQL connection id is 17

Server version: 9.0.1-commercial MySQL Enterprise Server - Commercial

Copyright (c) 2000, 2024, Oracle and/or its affiliates.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases

```
-> ;
+-----+
| Database |
+-----+
| assignment1 |
| exercise1 |
| information_schema |
| performance_schema |
+-----+
4 rows in set (0.00 sec)
```

mysql> assignment1;

ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'assignment1' at line 1

mysql> use assignment1;

Database changed

mysql> show tables;

```
+-----+
| Tables_in_assignment1 |
+-----+
| customers |
| orders |
| salespeople |
+-----+
3 rows in set (0.00 sec)
```

mysql> select * from customers;

```
+-----+-----+-----+-----+
| Cnum | Cname | City | Rating | Snum |
+-----+-----+-----+-----+
| 2001 | Hoffman | London | 100 | 1001 |
| 2002 | Giovanni | Rome | 200 | 1003 |
| 2003 | Liu | San Jose | 200 | 1002 |
| 2004 | Grass | Berlin | 300 | 1002 |
| 2006 | Clemens | London | 100 | 1001 |
| 2008 | Cisneros | San Jose | 300 | 1007 |
| 2007 | Pereira | Rome | 100 | 1004 |
+-----+-----+-----+-----+
7 rows in set (0.00 sec)
```

mysql> select * from orders;

```
+-----+-----+-----+-----+
| Onum | Amt | Odate | Cnum | Snum |
+-----+-----+-----+-----+
| 3001 | 18.69 | 1990-10-03 | 2008 | 1007 |
| 3003 | 767.19 | 1990-10-03 | 2001 | 1001 |
| 3002 | 1900.10 | 1990-10-03 | 2007 | 1004 |
| 3005 | 5160.45 | 1990-10-03 | 2003 | 1002 |
| 3006 | 1098.16 | 1990-10-03 | 2008 | 1007 |
| 3009 | 1713.23 | 1990-10-04 | 2002 | 1003 |
| 3007 | 75.75 | 1990-10-04 | 2004 | 1002 |
| 3008 | 4723.00 | 1990-10-05 | 2006 | 1001 |
| 3010 | 1309.95 | 1990-10-06 | 2004 | 1002 |
| 3011 | 9891.88 | 1990-10-06 | 2006 | 1001 |
+-----+-----+-----+-----+
10 rows in set (0.00 sec)
```

mysql> select * from salespeople;

```

+-----+-----+-----+
| Snum | Sname | City | Comm |
+-----+-----+-----+
| 1001 | Peel | London | 0.12 |
| 1002 | Serres | San Jose | 0.13 |
| 1004 | Motika | London | 0.11 |
| 1007 | Rifkin | Barcelona | 0.15 |
| 1003 | Axelrod | New York | 0.10 |
+-----+-----+-----+

```

5 rows in set (0.00 sec)

```

+-----+-----+-----+-----+-----+
| Cnum | Cname | City | Rating | Snum |
+-----+-----+-----+-----+-----+
| 2001 | Hoffman | London | 100 | 1001 |
| 2002 | Giovanni | Rome | 200 | 1003 |
| 2003 | Liu | San Jose | 200 | 1002 |
| 2004 | Grass | Berlin | 300 | 1002 |
| 2006 | Clemens | London | 100 | 1001 |
| 2008 | Cisneros | San Jose | 300 | 1007 |
| 2007 | Pereira | Rome | 100 | 1004 |
+-----+-----+-----+-----+-----+

```

7 rows in set (0.00 sec)

mysql> select * from orders;

```

+-----+-----+-----+-----+-----+
| Onum | Amt | Odate | Cnum | Snum |
+-----+-----+-----+-----+-----+
| 3001 | 18.69 | 1990-10-03 | 2008 | 1007 |
| 3003 | 767.19 | 1990-10-03 | 2001 | 1001 |
| 3002 | 1900.10 | 1990-10-03 | 2007 | 1004 |
| 3005 | 5160.45 | 1990-10-03 | 2003 | 1002 |
| 3006 | 1098.16 | 1990-10-03 | 2008 | 1007 |
| 3009 | 1713.23 | 1990-10-04 | 2002 | 1003 |
| 3007 | 75.75 | 1990-10-04 | 2004 | 1002 |
| 3008 | 4723.00 | 1990-10-05 | 2006 | 1001 |
| 3010 | 1309.95 | 1990-10-06 | 2004 | 1002 |
| 3011 | 9891.88 | 1990-10-06 | 2006 | 1001 |
+-----+-----+-----+-----+-----+

```

10 rows in set (0.00 sec)

mysql> select * from salespeople;

```

+-----+-----+-----+-----+
| Snum | Sname | City | Comm |
+-----+-----+-----+-----+
| 1001 | Peel | London | 0.12 |
| 1002 | Serres | San Jose | 0.13 |
| 1004 | Motika | London | 0.11 |
| 1007 | Rifkin | Barcelona | 0.15 |
| 1003 | Axelrod | New York | 0.10 |
+-----+-----+-----+-----+

```

5 rows in set (0.00 sec)

Assignment 2: Introducing Relational Databases

- Which field of the Customers table is the primary key?
 - The primary key of the Customers table is Cnum (Customer Number).
- What is the 4th column of the Customers table?
 - The 4th column in the Customers table is Rating.
- What is another word for row? For column?
 - Another word for row is tuple or record.
 - Another word for column is attribute or field.
- Why isn't it possible to see the first five rows of a table?
 - This question might refer to a technical or software-specific limitation. In relational databases, if you aren't using a query with a LIMIT clause (like LIMIT 5 in SQL), there is no automatic retrieval of the first five rows. You would need to explicitly specify this in your query.

Assignment 3: Overview of SQL

- Does ANSI recognize the data type DATE?
 - Yes, ANSI SQL recognizes the DATE data type. It represents a date in the format YYYY-MM-DD.
- Which subdivision of SQL is used to insert values in tables?

- The subdivision of SQL used to insert values into tables is the **Data Manipulation Language (DML)**. The INSERT statement is part of DML.

Assignment 4: Retrieving Information from Tables

```
mysql> SELECT Onum, Amt, Odate
-> FROM Orders;
```

```
+-----+-----+
| Onum | Amt   | Odate   |
+-----+-----+
| 3001 | 18.69 | 1990-10-03 |
| 3003 | 767.19 | 1990-10-03 |
| 3002 | 1900.10 | 1990-10-03 |
| 3005 | 5160.45 | 1990-10-03 |
| 3006 | 1098.16 | 1990-10-03 |
| 3009 | 1713.23 | 1990-10-04 |
| 3007 | 75.75 | 1990-10-04 |
| 3008 | 4723.00 | 1990-10-05 |
| 3010 | 1309.95 | 1990-10-06 |
| 3011 | 9891.88 | 1990-10-06 |
+-----+-----+
10 rows in set (0.00 sec)
```

```
mysql> SELECT *
-> FROM Customers
-> WHERE Snum = 1001;
```

```
+-----+-----+-----+-----+
| Cnum | Cname | City | Rating | Snum |
+-----+-----+-----+-----+
| 2001 | Hoffman | London | 100 | 1001 |
| 2006 | Clemens | London | 100 | 1001 |
+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

```
mysql> SELECT City, Sname, Snum, Comm
-> FROM Salespeople;
```

```
+-----+-----+-----+-----+
| City | Sname | Snum | Comm |
+-----+-----+-----+-----+
| London | Peel | 1001 | 0.12 |
| San Jose | Serres | 1002 | 0.13 |
| London | Motika | 1004 | 0.11 |
| Barcelona | Rifkin | 1007 | 0.15 |
| New York | Axelrod | 1003 | 0.10 |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

```
mysql> SELECT Rating, Cname
-> FROM Customers
-> WHERE City = 'San Jose';
```

```
+-----+-----+
| Rating | Cname |
+-----+-----+
| 200 | Liu |
| 300 | Cisneros |
+-----+-----+
2 rows in set (0.00 sec)
```

```
mysql> SELECT DISTINCT Snum
-> FROM Orders;
```

```
+-----+
| Snum |
+-----+
| 1007 |
| 1001 |
| 1004 |
| 1002 |
```

| 1003 |

+-----+

5 rows in set (0.00 sec)

mysql>

```
mysql> SELECT Onum, Amt, Odate
-> FROM Orders;
```

Onum	Amt	Odate
3001	18.69	1990-10-03
3003	767.19	1990-10-03
3002	1900.10	1990-10-03
3005	5160.45	1990-10-03
3006	1098.16	1990-10-03
3009	1713.23	1990-10-04
3007	75.75	1990-10-04
3008	4723.00	1990-10-05
3010	1309.95	1990-10-06
3011	9891.88	1990-10-06

10 rows in set (0.00 sec)

```
mysql> SELECT *
-> FROM Customers
-> WHERE Snum = 1001;
```

Cnum	Cname	City	Rating	Snum
2001	Hoffman	London	100	1001
2006	Clemens	London	100	1001

2 rows in set (0.00 sec)

```
mysql> SELECT City, Sname, Snum, Comm
-> FROM Salespeople;
```

City	Sname	Snum	Comm
London	Peel	1001	0.12
San Jose	Serres	1002	0.13
London	Motika	1004	0.11
Barcelona	Rifkin	1007	0.15
New York	Axelrod	1003	0.10

```
mysql> SELECT Rating, Cname
-> FROM Customers
-> WHERE City = 'San Jose';
```

Rating	Cname
200	Liu
300	Cisneros

2 rows in set (0.00 sec)

```
mysql> SELECT DISTINCT Snum
-> FROM Orders;
```

Snum
1007
1001
1004
1002
1003

5 rows in set (0.00 sec)

1. Query for orders more than Rs. 1,000:

```
SELECT * FROM Orders
WHERE amt > 1000;
```

```
mysql> select * from orders
-> where Amt > 1000;
+-----+-----+-----+-----+-----+
| Onum | Amt   | Odate   | Cnum | Snum |
+-----+-----+-----+-----+-----+
| 3002 | 1900.10 | 1990-10-03 | 2007 | 1004 |
| 3005 | 5160.45 | 1990-10-03 | 2003 | 1002 |
| 3006 | 1098.16 | 1990-10-03 | 2008 | 1007 |
| 3009 | 1713.23 | 1990-10-04 | 2002 | 1003 |
| 3008 | 4723.00 | 1990-10-05 | 2006 | 1001 |
| 3010 | 1309.95 | 1990-10-06 | 2004 | 1002 |
| 3011 | 9891.88 | 1990-10-06 | 2006 | 1001 |
+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)
```

1. Query to get names and cities of salespeople in London with a commission above .10:

```
SELECT sname, city FROM Salespeople
WHERE city = 'London' AND comm > 0.10;
```

```
mysql> select Sname, City from salespeople
-> where City = 'London' and Comm > 0.10;
+-----+-----+
| Sname | City |
+-----+-----+
| Peel  | London |
| Motika | London |
+-----+-----+
2 rows in set (0.00 sec)
```

1. Query to exclude customers with a rating <= 100, unless they are located in Rome:

```
SELECT * FROM Customers
WHERE (rating > 100 OR city = 'Rome');
```

```
mysql> select * from customers
-> where (rating > 100 or city = 'Rome');
+-----+-----+-----+-----+-----+
| Cnum | Cname   | City    | Rating | Snum |
+-----+-----+-----+-----+-----+
| 2002 | Giovanni | Rome    | 200    | 1003 |
| 2003 | Liu      | San Jose | 200    | 1002 |
| 2004 | Grass    | Berlin  | 300    | 1002 |
| 2008 | Cisneros | San Jose | 300    | 1007 |
| 2007 | Pereira  | Rome    | 100    | 1004 |
+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

1. Output of the given query:

```
SELECT * FROM Orders
WHERE (amt < 1000 OR NOT (odate = '1990-10-03' AND cnum > 2003));
```

- This query will fetch orders where:
 - The amount is less than 1000 or
 - The order date is not '1990-10-03' and customer number is greater than 2003.

```
mysql> select * from orders
-> where (amt < 1000 or not (odate = '1990-10-03' and cnum > 2003));
+-----+-----+-----+-----+-----+
| Onum | Amt   | Odate   | Cnum | Snum |
+-----+-----+-----+-----+-----+
| 3001 | 18.69 | 1990-10-03 | 2008 | 1007 |
| 3003 | 767.19 | 1990-10-03 | 2001 | 1001 |
| 3005 | 5160.45 | 1990-10-03 | 2003 | 1002 |
| 3009 | 1713.23 | 1990-10-04 | 2002 | 1003 |
| 3007 | 75.75 | 1990-10-04 | 2004 | 1002 |
| 3008 | 4723.00 | 1990-10-05 | 2006 | 1001 |
| 3010 | 1309.95 | 1990-10-06 | 2004 | 1002 |
| 3011 | 9891.88 | 1990-10-06 | 2006 | 1001 |
+-----+-----+-----+-----+-----+
8 rows in set (0.01 sec)
```

2. Output of the second given query:

```
SELECT * FROM Orders
```

WHERE NOT ((odate = '1990-10-03' OR snum > 1006) AND amt >= 1500);

- This query will fetch orders that don't meet the following conditions:
 - Either the order date is '1990-10-03' or the salesperson number is greater than 1006 and the amount is greater than or equal to 1500.

```
mysql> select * from Orders
-> where NOT ((Odate = '1990-10-03' OR Snum > 1006) AND Amt >= 1500);
+-----+-----+-----+-----+-----+
| Onum | Amt   | Odate   | Cnum | Snum |
+-----+-----+-----+-----+-----+
| 3001 | 18.69 | 1990-10-03 | 2008 | 1007 |
| 3003 | 767.19 | 1990-10-03 | 2001 | 1001 |
| 3006 | 1098.16 | 1990-10-03 | 2008 | 1007 |
| 3009 | 1713.23 | 1990-10-04 | 2002 | 1003 |
| 3007 | 75.75 | 1990-10-04 | 2004 | 1002 |
| 3008 | 4723.00 | 1990-10-05 | 2006 | 1001 |
| 3010 | 1309.95 | 1990-10-06 | 2004 | 1002 |
| 3011 | 9891.88 | 1990-10-06 | 2006 | 1001 |
+-----+-----+-----+-----+-----+
8 rows in set (0.00 sec)
```

3. Simpler way to write the query:

- The original query:

```
SELECT snum, sname, city, comm FROM Salespeople
WHERE (comm > 0.12 OR comm < 0.14);
```

```
mysql> SELECT snum, sname, city, comm FROM Salespeople
-> WHERE (comm > 0.12 OR comm < 0.14);
+-----+-----+-----+-----+
| snum | sname | city   | comm |
+-----+-----+-----+-----+
| 1001 | Peel  | London | 0.12 |
| 1002 | Serres | San Jose | 0.13 |
| 1004 | Motika | London | 0.11 |
| 1007 | Rifkin | Barcelona | 0.15 |
| 1003 | Axelrod | New York | 0.10 |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

- Can be simplified as using BETWEEN Operator:

```
SELECT snum, sname, city, comm FROM Salespeople
WHERE comm BETWEEN 0.12 AND 0.14;
```

```
mysql> SELECT snum, sname, city, comm FROM Salespeople
-> WHERE comm BETWEEN 0.12 AND 0.14;
+-----+-----+-----+-----+
| snum | sname | city   | comm |
+-----+-----+-----+-----+
| 1002 | Serres | San Jose | 0.13 |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Assignment 6: Using Special Operators in Conditions

1. Two queries for orders taken on October 3rd or 4th, 1990:

- Query 1:

```
SELECT * FROM Orders
WHERE odate = '1990-10-03' OR odate = '1990-10-04';
```

```
mysql> SELECT * FROM Orders
-> WHERE odate = '1990-10-03' OR odate = '1990-10-04';
+-----+-----+-----+-----+-----+
| Onum | Amt   | Odate   | Cnum | Snum |
+-----+-----+-----+-----+-----+
| 3001 | 18.69 | 1990-10-03 | 2008 | 1007 |
| 3003 | 767.19 | 1990-10-03 | 2001 | 1001 |
| 3002 | 1900.10 | 1990-10-03 | 2007 | 1004 |
| 3005 | 5160.45 | 1990-10-03 | 2003 | 1002 |
| 3006 | 1098.16 | 1990-10-03 | 2008 | 1007 |
| 3009 | 1713.23 | 1990-10-04 | 2002 | 1003 |
| 3007 | 75.75 | 1990-10-04 | 2004 | 1002 |
+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)
```

- Query 2 using IN:


```
SELECT * FROM Orders
WHERE odate IN ('1990-10-03', '1990-10-04');
```

```
mysql> SELECT * FROM Orders
-> WHERE odate IN ('1990-10-03', '1990-10-04');
```

Onum	Amt	Odate	Cnum	Snum
3001	18.69	1990-10-03	2008	1007
3003	767.19	1990-10-03	2001	1001
3002	1900.10	1990-10-03	2007	1004
3005	5160.45	1990-10-03	2003	1002
3006	1098.16	1990-10-03	2008	1007
3009	1713.23	1990-10-04	2002	1003
3007	75.75	1990-10-04	2004	1002

```
7 rows in set (0.00 sec)
```

1. Query to select customers serviced by Peel or Motika:

- Assuming snum in Orders relates to the salesperson:

```
SELECT * FROM Customers
WHERE snum IN (SELECT snum FROM Salespeople WHERE sname = 'Peel' OR sname = 'Motika');
```

```
mysql> SELECT * FROM Customers
-> WHERE snum IN (SELECT snum FROM Salespeople WHERE sname = 'Peel' OR sname = 'Motika');
```

Cnum	Cname	City	Rating	Snum
2001	Hoffman	London	100	1001
2006	Clemens	London	100	1001
2007	Pereira	Rome	100	1004

```
3 rows in set (0.01 sec)
```

1. Query to select customers whose names begin with a letter from 'A' to 'G':

```
SELECT * FROM Customers
WHERE cname BETWEEN 'A%' AND 'G%';
```

```
mysql> SELECT * FROM Customers
-> WHERE cname BETWEEN 'A%' AND 'G%';
```

Cnum	Cname	City	Rating	Snum
2006	Clemens	London	100	1001
2008	Cisneros	San Jose	300	1007

```
2 rows in set (0.00 sec)
```

1. Query to select all customers whose names begin with the letter 'C':

```
SELECT * FROM Customers
WHERE cname LIKE 'C%';
```

```
mysql> SELECT * FROM Customers
-> WHERE cname LIKE 'C%';
```

Cnum	Cname	City	Rating	Snum
2006	Clemens	London	100	1001
2008	Cisneros	San Jose	300	1007

```
2 rows in set (0.00 sec)
```

1. Query to select all orders except those with zeroes or NULLs in the amt field:

```
SELECT * FROM Orders
WHERE amt IS NOT NULL AND amt != 0;
```



```
mysql> SELECT * FROM Orders
-> WHERE amt IS NOT NULL AND amt != 0;
+-----+-----+-----+-----+-----+
| Onum | Amt   | Odate   | Cnum | Snum |
+-----+-----+-----+-----+-----+
| 3001 | 18.69 | 1990-10-03 | 2008 | 1007 |
| 3003 | 767.19 | 1990-10-03 | 2001 | 1001 |
| 3002 | 1900.10 | 1990-10-03 | 2007 | 1004 |
| 3005 | 5160.45 | 1990-10-03 | 2003 | 1002 |
| 3006 | 1098.16 | 1990-10-03 | 2008 | 1007 |
| 3009 | 1713.23 | 1990-10-04 | 2002 | 1003 |
| 3007 | 75.75 | 1990-10-04 | 2004 | 1002 |
| 3008 | 4723.00 | 1990-10-05 | 2006 | 1001 |
| 3010 | 1309.95 | 1990-10-06 | 2004 | 1002 |
| 3011 | 9891.88 | 1990-10-06 | 2006 | 1001 |
+-----+-----+-----+-----+-----+
10 rows in set (0.00 sec)
```

Assignment –7

Summarizing Data with Aggregate Functions.

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

```
mysql> select count(*) from orders
-> where Odate = '1990-10-03';
+-----+
| count(*) |
+-----+
|          5 |
+-----+
1 row in set (0.00 sec)
```

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

```
mysql> select count(distinct city) from customers
-> where city is not null;
+-----+
| count(distinct city) |
+-----+
|          4 |
+-----+
1 row in set (0.00 sec)
```

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

```
mysql> select cnum, min(amt) as SmallestOrder from orders
-> group by Cnum;
+-----+-----+
| cnum | SmallestOrder |
+-----+-----+
| 2008 | 18.69 |
| 2001 | 767.19 |
| 2007 | 1900.10 |
| 2003 | 5160.45 |
| 2002 | 1713.23 |
| 2004 | 75.75 |
| 2006 | 4723.00 |
+-----+-----+
7 rows in set (0.00 sec)
```

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

```
mysql> select * from customers
-> where cname like 'G%'
-> order by cname
-> limit 1;
+-----+-----+-----+-----+-----+
| Cnum | Cname   | City | Rating | Snum |
+-----+-----+-----+-----+-----+
| 2002 | Giovanni | Rome | 200 | 1003 |
+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

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

```
mysql> select city, max(rating) as HighestRating from customers
-> group by City;
```

city	HighestRating
London	100
Rome	200
San Jose	300
Berlin	300

4 rows in set (0.04 sec)

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.).

```
mysql> select odate, count(distinct snum) as SalesPeopleCount from orders
-> group by odate;
```

odate	SalesPeopleCount
1990-10-03	4
1990-10-04	2
1990-10-05	1
1990-10-06	2

4 rows in set (0.04 sec)

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.

```
mysql> select onum, snum, amt * 0.12 as Commission from orders;
```

onum	snum	Commission
3001	1007	2.24
3003	1001	92.06
3002	1004	228.01
3005	1002	619.25
3006	1007	131.78
3009	1003	205.59
3007	1002	9.09
3008	1001	566.76
3010	1002	157.19
3011	1001	1187.03

10 rows in set (0.00 sec)

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).

```
mysql> select
-> concat('For the city ', City, ', the highest rating is : ', max(Rating))
as Output
-> from customers
-> group by city;
```

Output
For the city London, the highest rating is : 100
For the city Rome, the highest rating is : 200
For the city San Jose, the highest rating is : 300
For the city Berlin, the highest rating is : 300

4 rows in set (0.00 sec)

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.

```
mysql> select rating, cname, cnum from customers
-> order by rating desc;
+-----+-----+-----+
| rating | cname   | cnum |
+-----+-----+-----+
| 300    | Grass   | 2004 |
| 300    | Cisneros | 2008 |
| 200    | Giovanni | 2002 |
| 200    | Liu     | 2003 |
| 100    | Hoffman | 2001 |
| 100    | Clemens | 2006 |
| 100    | Pereira | 2007 |
+-----+-----+-----+
7 rows in set (0.00 sec)
```

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

```
mysql> select odate, sum(amt) as TotalAmount from orders
-> group by odate
-> order by TotalAmount desc;
+-----+-----+
| odate      | TotalAmount |
+-----+-----+
| 1990-10-06 | 11201.83    |
| 1990-10-03 | 8944.59     |
| 1990-10-05 | 4723.00     |
| 1990-10-04 | 1788.98     |
+-----+-----+
4 rows in set (0.00 sec)
```

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.

```
mysql> select orders.onum, customers.cname from orders, customers
-> where Orders.Cnum = Customers.Cnum;
+-----+-----+
| onum | cname   |
+-----+-----+
| 3001 | Cisneros |
| 3003 | Hoffman |
| 3002 | Pereira |
| 3005 | Liu     |
| 3006 | Cisneros |
| 3009 | Giovanni |
| 3007 | Grass   |
| 3008 | Clemens |
| 3010 | Grass   |
| 3011 | Clemens |
+-----+-----+
10 rows in set (0.00 sec)
```

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

```
mysql> SELECT Orders.Onum, Customers.Cname, Salespeople.Sname
-> FROM Orders, Customers, Salespeople
-> WHERE Orders.Cnum = Customers.Cnum
-> AND Orders.Snum = Salespeople.Snum;
```

Onum	Cname	Sname
3003	Hoffman	Peel
3009	Giovanni	Axelrod
3005	Liu	Serres
3010	Grass	Serres
3007	Grass	Serres
3011	Clemens	Peel
3008	Clemens	Peel
3006	Cisneros	Rifkin
3001	Cisneros	Rifkin
3002	Pereira	Motika

10 rows in set (0.00 sec)

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.

```
mysql> SELECT Customers.Cname, Salespeople.Sname, Salespeople.Comm
-> FROM Customers, Salespeople
-> WHERE Customers.Snum = Salespeople.Snum
-> AND Salespeople.Comm > 0.12;
```

Cname	Sname	Comm
Liu	Serres	0.13
Grass	Serres	0.13
Cisneros	Rifkin	0.15

3 rows in set (0.01 sec)

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

```
mysql> SELECT Orders.Onum, Customers.Cname, Salespeople.Sname, (Orders.Amt * Salespeople.Comm) AS Commission
-> FROM Orders, Customers, Salespeople
-> WHERE Orders.Cnum = Customers.Cnum
-> AND Orders.Snum = Salespeople.Snum
-> AND Customers.Rating > 100;
```

Onum	Cname	Sname	Commission
3010	Grass	Serres	170.29
3007	Grass	Serres	9.85
3005	Liu	Serres	670.86
3006	Cisneros	Rifkin	164.72
3001	Cisneros	Rifkin	2.80
3009	Giovanni	Axelrod	171.32

6 rows in set (0.05 sec)

mysql> |

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.


```
mysql> SELECT A.Sname AS Salesperson1, B.Sname AS Salesperson2, A.City
-> FROM Salespeople A, Salespeople B
-> WHERE A.City = B.City
-> AND A.Snum < B.Snum;
+-----+-----+-----+
| Salesperson1 | Salesperson2 | City   |
+-----+-----+-----+
| Peel         | Motika       | London |
+-----+-----+-----+
1 row in set (0.00 sec)
```

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

```
mysql> SELECT C1.Cname, C1.City
-> FROM Customers C1, Customers C2
-> WHERE C1.Rating = C2.Rating
-> AND C2.Cname = 'Hoffman';
+-----+-----+
| Cname   | City   |
+-----+-----+
| Hoffman | London |
| Clemens | London |
| Pereira | Rome   |
+-----+-----+
3 rows in set (0.00 sec)
```