Assignment – 19 **Views.**

1) Create a view that shows all of the customers who have the highest ratings.

```
CREATE VIEW HighRatingCustomers AS
SELECT *
 FROM Customers
 WHERE Rating = (SELECT MAX(Rating) FROM Customers);

SELECT * FROM HighRatingCustomers;
```

```
D3_92819_Shubham>CREATE VIEW HighRatingCustomers AS
    -> SELECT *
    -> FROM Customers
    -> WHERE Rating = (SELECT MAX(Rating) FROM Customers);
Query OK, 0 rows affected (0.02 sec)

D3_92819_Shubham>SELECT * FROM HighRatingCustomers;
+------+----------+----------+--------+------+
| Cnum | Cname    | City     | Rating | Snum |
+------+----------+----------+--------+------+
| 2004 | Grass    | Berlin   |    300 | 1002 |
| 2008 | Cisneros | San Jose |    300 | 1007 |
+------+----------+----------+--------+------+
2 rows in set (0.01 sec)

D3_92819_Shubham>
```

2) Create a view that shows the number of salespeople in each city.

```
CREATE VIEW SalespeopleCountByCity AS
 SELECT City, COUNT(*) AS NumSalespeople
 FROM Salespeople
 GROUP BY City;

SELECT * FROM SalespeopleCountByCity;
```

```
D3_92819_Shubham>CREATE VIEW SalespeopleCountByCity AS
    -> SELECT City, COUNT(*) AS NumSalespeople
    -> FROM Salespeople
    -> GROUP BY City;
Query OK, 0 rows affected (0.01 sec)

D3_92819_Shubham>SELECT * FROM SalespeopleCountByCity;
+-----------+----------------+
| City      | NumSalespeople |
+-----------+----------------+
| London    |              2 |
| San Jose  |              1 |
| Barcelona |              1 |
| New York  |              1 |
+-----------+----------------+
4 rows in set (0.01 sec)

D3_92819_Shubham>
```

3) Create a view that shows the average and total orders for each salesperson after his or her name. Assume all names are unique.

```
CREATE VIEW SalespersonOrderStats AS
SELECT s.Sname,
AVG(o.Amt) AS AvgOrderAmount,
SUM(o.Amt) AS TotalOrderAmount
FROM Salespeople s
JOIN Orders o ON s.Snum = o.Snum
GROUP BY s.Sname;
```

```
SELECT * FROM SalespersonOrderStats;
```

```
D3_92819_Shubham>CREATE VIEW SalespersonOrderStats AS
    -> SELECT s.Sname,
    ->        AVG(o.Amt) AS AvgOrderAmount,
    ->        SUM(o.Amt) AS TotalOrderAmount
    -> FROM Salespeople s
    -> JOIN Orders o ON s.Snum = o.Snum
    -> GROUP BY s.Sname;
Query OK, 0 rows affected (0.02 sec)

D3_92819_Shubham>SELECT * FROM SalespersonOrderStats;
+---------+----------------+------------------+
| Sname   | AvgOrderAmount | TotalOrderAmount |
+---------+----------------+------------------+
| Rifkin  |     558.425017 |          1116.85 |
| Peel    |    5127.356628 |         15382.07 |
| Motika  |    1900.099976 |          1900.10 |
| Serres  |    2182.050049 |          6546.15 |
| Axelrod |    1713.229980 |          1713.23 |
+---------+----------------+------------------+
5 rows in set (0.00 sec)

D3_92819_Shubham>
```

4) Create a view that shows each salesperson with multiple customers

```
CREATE VIEW SalespeopleWithMultipleCustomers AS
SELECT s.Snum, s.Sname, s.City, s.Comm, COUNT(c.Cnum) AS NumCustomers
FROM Salespeople s
JOIN Customers c ON s.Snum = c.Snum
GROUP BY s.Snum, s.Sname, s.City, s.Comm
HAVING COUNT(c.Cnum) > 1;
```

```
SELECT * FROM SalespeopleWithMultipleCustomers;
```

```
D3_92819_Shubham>CREATE VIEW SalespeopleWithMultipleCustomers AS
    -> SELECT s.Snum, s.Sname, s.City, s.Comm, COUNT(c.Cnum) AS NumCustomers
    -> FROM Salespeople s
    -> JOIN Customers c ON s.Snum = c.Snum
    -> GROUP BY s.Snum, s.Sname, s.City, s.Comm
    -> HAVING COUNT(c.Cnum) > 1;
Query OK, 0 rows affected (0.02 sec)

D3_92819_Shubham>SELECT * FROM SalespeopleWithMultipleCustomers;
+------+--------+----------+------+--------------+
| Snum | Sname  | City     | Comm | NumCustomers |
+------+--------+----------+------+--------------+
| 1001 | Peel   | London   | 0.12 |            2 |
| 1002 | Serres | San Jose | 0.13 |            2 |
+------+--------+----------+------+--------------+
2 rows in set (0.00 sec)

D3_92819_Shubham>
```