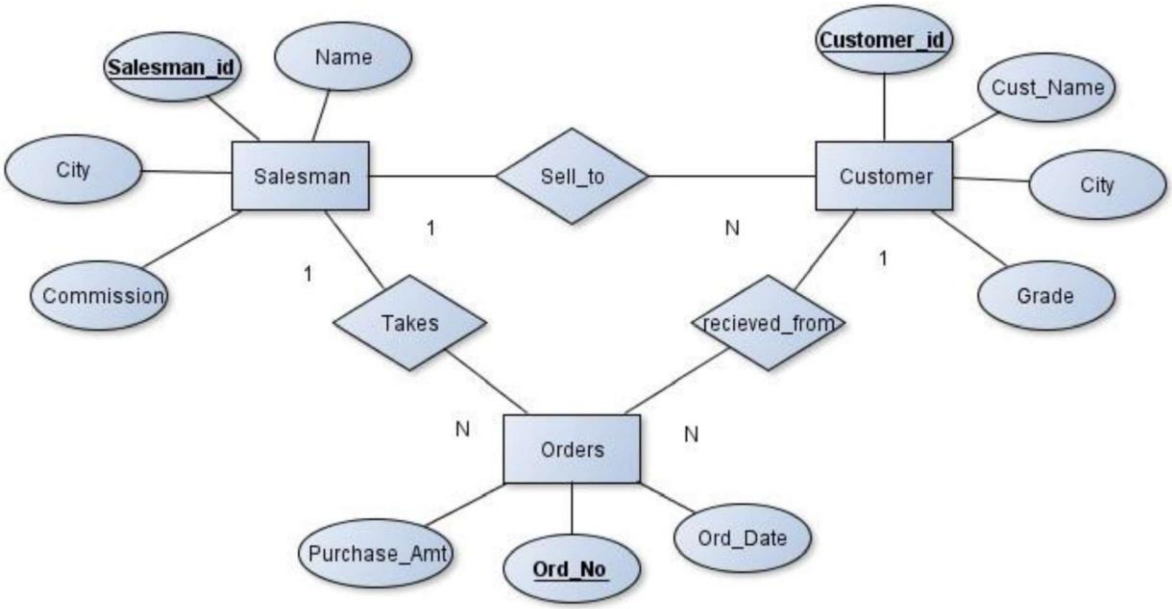
Consider the following schema for Order Database:

SALESMAN (Salesman_id, Name, City, Commission)
CUSTOMER (Customer_id, Cust_Name, City, Grade, Salesman_id)
ORDERS (Ord_No, Purchase_Amt, Ord_Date, Customer_id, Salesman_id)

Write SQL queries to

1. **Count the customers with grades above Bangalore's average.**
2. **Find the name and numbers of all salesmen who had more than one customer.**
3. **List all salesmen and indicate those who have and don't have customers in their cities (Use UNION operation.)**
4. **Create a view that finds the salesman who has the customer with the highest order of a day.**
5. **Demonstrate the DELETE operation by removing salesman with id 1000. All his orders must also be deleted.**

Solution:
Entity-Relationship Diagram



Schema Diagram

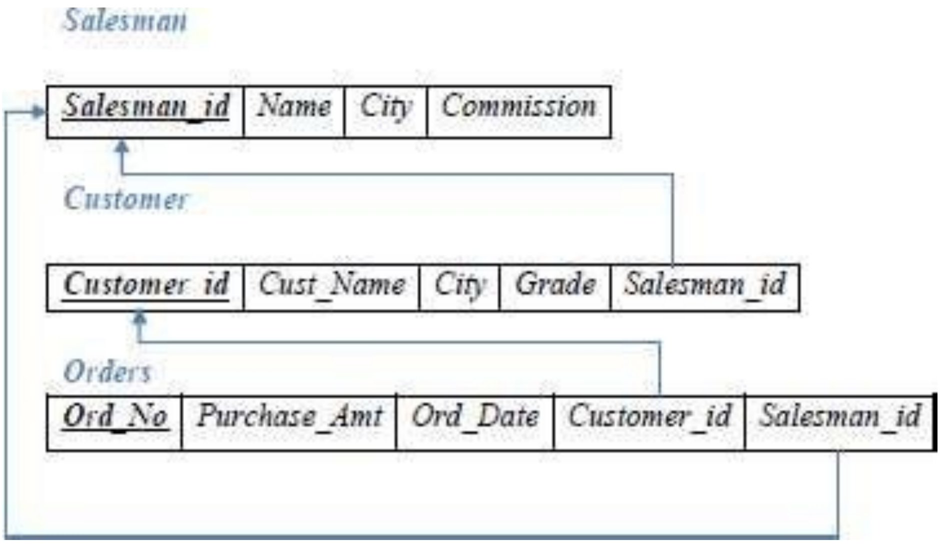
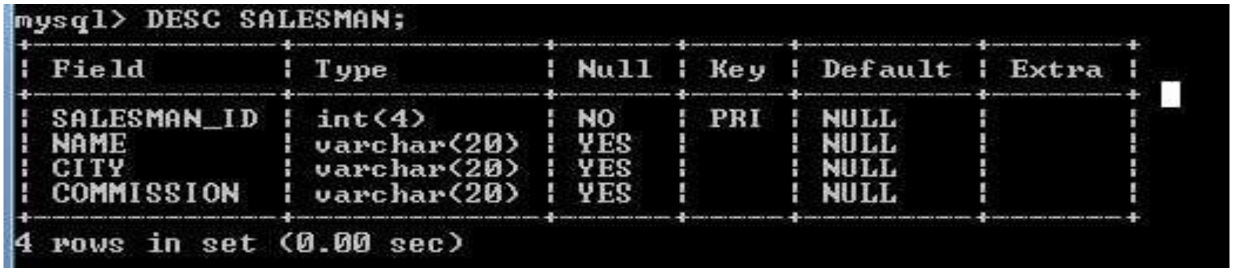


Table Creation

```
CREATE TABLE SALESMAN (
SALESMAN_ID INT (4) PRIMARY KEY,
NAME VARCHAR (20),
CITY VARCHAR (20),
COMMISSION VARCHAR (20));

CREATE TABLE CUSTOMER (
CUSTOMER_ID INT (5) PRIMARY KEY,
CUST_NAME VARCHAR (20),
CITY VARCHAR (20), GRADE INT (4),
SALESMAN_ID INT (6),
FOREIGN KEY (SALESMAN_ID) REFERENCES SALESMAN (SALESMAN_ID) ON DELETE
SET NULL);

CREATE TABLE ORDERS (
ORD_NO INT (5) PRIMARY KEY,
PURCHASE_AMT DECIMAL (10, 2),
ORD_DATE DATE,
CUSTOMER_ID INT (4),
SALESMAN_ID INT (4),
FOREIGN KEY (CUSTOMER_ID) REFERENCES CUSTOMER (CUSTOMER_ID) ON DELETE
CASCADE,
FOREIGN KEY (SALESMAN_ID) REFERENCES SALESMAN (SALESMAN_ID) ON DELETE
CASCADE);
```

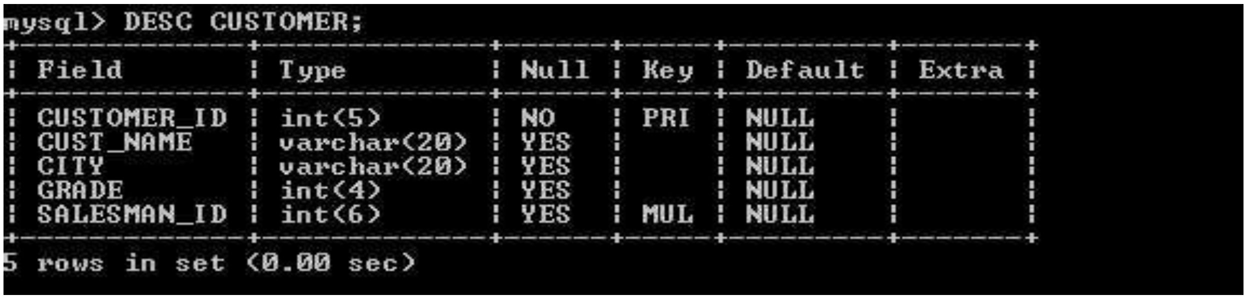
Table Descriptions

DESC SALESMAN;



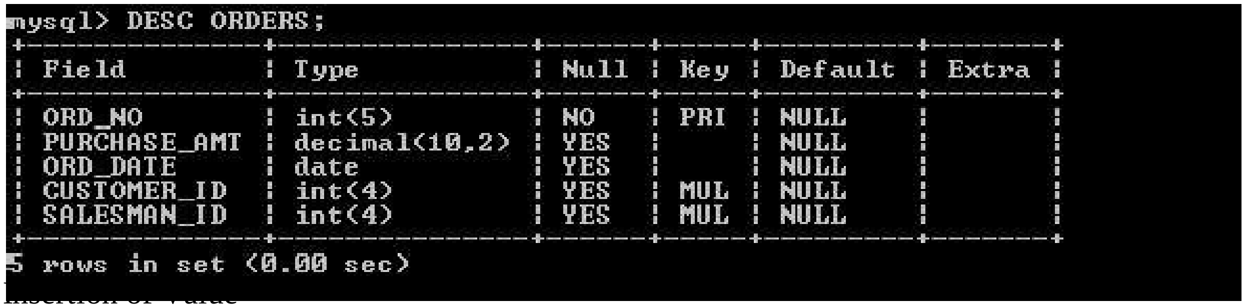
```
mysql> DESC SALESMAN;
+---------------+-------------+------+-----+---------+-------+
| Field         | Type        | Null | Key | Default | Extra |
+---------------+-------------+------+-----+---------+-------+
| SALESMAN_ID   | int(4)      | NO   | PRI | NULL    |       |
| NAME          | varchar(20) | YES  |     | NULL    |       |
| CITY          | varchar(20) | YES  |     | NULL    |       |
| COMMISSION    | varchar(20) | YES  |     | NULL    |       |
+---------------+-------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

DESC CUSTOMER;



```
mysql> DESC CUSTOMER;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| CUSTOMER_ID | int(5)      | NO   | PRI | NULL    |       |
| CUST_NAME   | varchar(20) | YES  |     | NULL    |       |
| CITY        | varchar(20) | YES  |     | NULL    |       |
| GRADE       | int(4)      | YES  |     | NULL    |       |
| SALESMAN_ID | int(6)      | YES  | MUL | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
5 rows in set (0.00 sec)
```

DESC ORDERS;

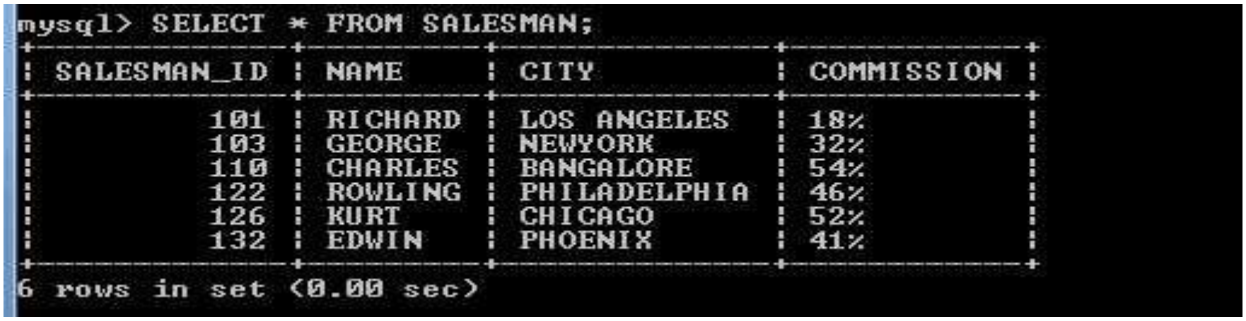


```
mysql> DESC ORDERS;
+--------------+---------------+------+-----+---------+-------+
| Field        | Type          | Null | Key | Default | Extra |
+--------------+---------------+------+-----+---------+-------+
| ORD_NO       | int(5)        | NO   | PRI | NULL    |       |
| PURCHASE_AMT | decimal(10,2) | YES  |     | NULL    |       |
| ORD_DATE     | date          | YES  |     | NULL    |       |
| CUSTOMER_ID  | int(4)        | YES  | MUL | NULL    |       |
| SALESMAN_ID  | int(4)        | YES  | MUL | NULL    |       |
+--------------+---------------+------+-----+---------+-------+
5 rows in set (0.00 sec)
```

Insertion of Value

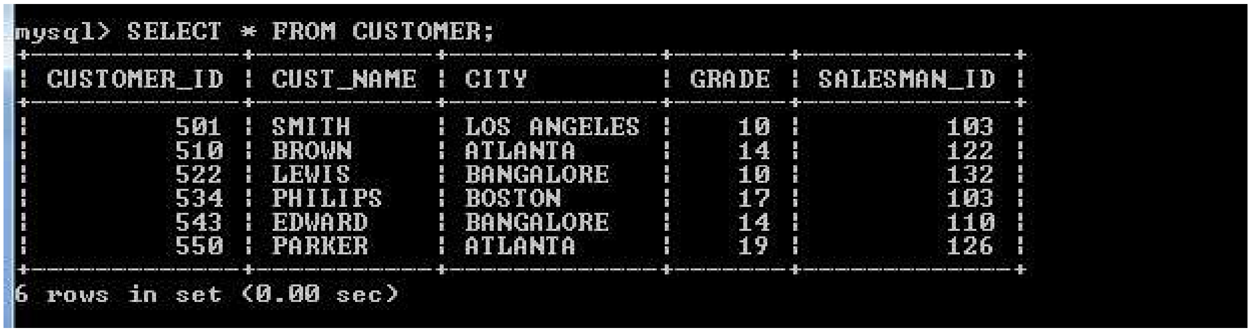
```
INSERT INTO  SALESMAN VALUES(101,'RICHARD','LOS  ANGELES','18%');
INSERT INTO  SALESMAN VALUES(103,'GEORGE','NEWYORK','32%');
INSERT INTO  SALESMAN VALUES(110,'CHARLES','BANGALORE','54%');
INSERT INTO  SALESMAN VALUES(122,'ROWLING','PHILADELPHIA','46%');
INSERT INTO  SALESMAN VALUES(126,'KURT','CHICAGO','52%');
INSERT INTO  SALESMAN VALUES(132,'EDWIN','PHOENIX','41%');

INSERT INTO  CUSTOMER VALUES(501,'SMITH','LOS ANGELES',10,103);
INSERT INTO  CUSTOMER VALUES(510,'BROWN','ATLANTA',14,122);
INSERT INTO  CUSTOMER VALUES(522,'LEWIS','BANGALORE',10,132);
INSERT INTO  CUSTOMER VALUES(534,'PHILIPS','BOSTON',17,103);
INSERT INTO  CUSTOMER VALUES(543,'EDWARD','BANGALORE',14,110);
INSERT INTO  CUSTOMER VALUES(550,'PARKER','ATLANTA',19,126);
INSERT INTO  ORDERS  VALUES(1,1000, '2017-05-04',501,103);
INSERT  INTO  ORDERS  VALUES(2,4000,'2017-01-20',522,132);
INSERT INTO  ORDERS  VALUES(3,2500, '2017-02-24',550,126);
INSERT  INTO  ORDERS  VALUES(5,6000,'2017-04-13',522,103);
INSERT INTO  ORDERS  VALUES(6,7000, '2017-03-09',550,126);
INSERT INTO  ORDERS  VALUES (7,3400,'2017-01-20',501,122);
```

```
SELECT  *  FROM   SALESMAN;
```



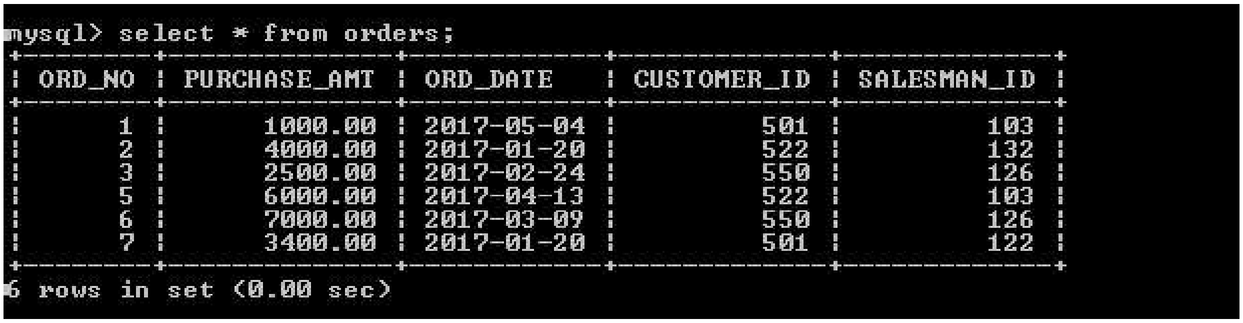
```
mysql> SELECT * FROM SALESMAN;
+-------------+---------+--------------+------------+
| SALESMAN_ID | NAME    | CITY         | COMMISSION |
+-------------+---------+--------------+------------+
|         101 | RICHARD | LOS ANGELES  | 18%        |
|         103 | GEORGE  | NEWYORK      | 32%        |
|         110 | CHARLES | BANGALORE    | 54%        |
|         122 | ROWLING | PHILADELPHIA | 46%        |
|         126 | KURT    | CHICAGO      | 52%        |
|         132 | EDWIN   | PHOENIX      | 41%        |
+-------------+---------+--------------+------------+
6 rows in set (0.00 sec)
```

```
SELECT  *  FROM  CUSTOMER;
```



```
mysql> SELECT * FROM CUSTOMER;
+-------------+-----------+-------------+-------+-------------+
| CUSTOMER_ID | CUST_NAME | CITY        | GRADE | SALESMAN_ID |
+-------------+-----------+-------------+-------+-------------+
|         501 | SMITH     | LOS ANGELES |    10 |         103 |
|         510 | BROWN     | ATLANTA     |    14 |         122 |
|         522 | LEWIS     | BANGALORE   |    10 |         132 |
|         534 | PHILIPS   | BOSTON      |    17 |         103 |
|         543 | EDWARD    | BANGALORE   |    14 |         110 |
|         550 | PARKER    | ATLANTA     |    19 |         126 |
+-------------+-----------+-------------+-------+-------------+
6 rows in set (0.00 sec)
```

```
SELECT  *  FROM  ORDERS;
```

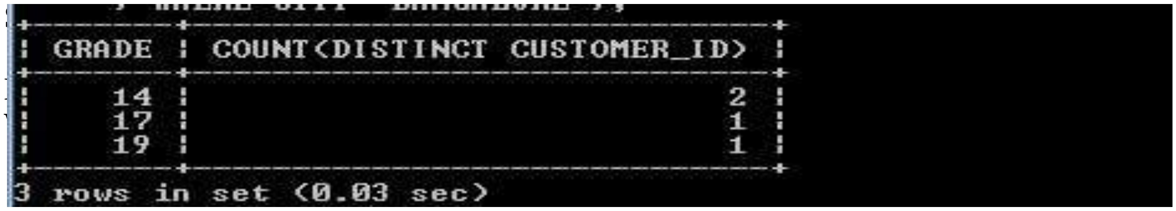


```
mysql> select * from orders;
+--------+--------------+------------+-------------+-------------+
| ORD_NO | PURCHASE_AMT | ORD_DATE   | CUSTOMER_ID | SALESMAN_ID |
+--------+--------------+------------+-------------+-------------+
|      1 |      1000.00 | 2017-05-04 |         501 |         103 |
|      2 |      4000.00 | 2017-01-20 |         522 |         132 |
|      3 |      2500.00 | 2017-02-24 |         550 |         126 |
|      5 |      6000.00 | 2017-04-13 |         522 |         103 |
|      6 |      7000.00 | 2017-03-09 |         550 |         126 |
|      7 |      3400.00 | 2017-01-20 |         501 |         122 |
+--------+--------------+------------+-------------+-------------+
6 rows in set (0.00 sec)
```

Queries

1.Count the customers with grades above Bangalore's average.

```
SELECT GRADE, COUNT (CUSTOMER_ID) FROM
CUSTOMER GROUP BY GRADE
HAVING GRADE > (SELECT AVG (GRADE) FROM
CUSTOMER WHERE CITY='BANGALORE');
```



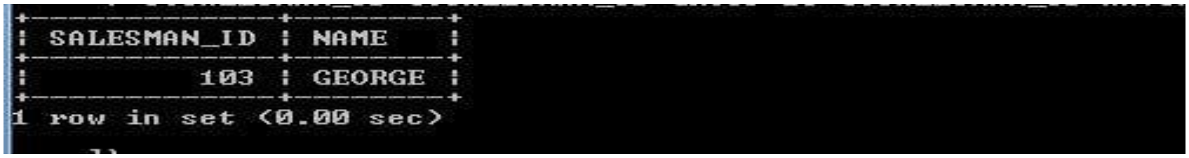
```
+-------+-------------------------------+
| GRADE | COUNT(DISTINCT CUSTOMER_ID) |
+-------+-------------------------------+
|    14 |                             2 |
|    17 |                             1 |
|    19 |                             1 |
+-------+-------------------------------+
3 rows in set (0.03 sec)
```

2. Find the name and numbers of all salesmen who had more than one customer.

```
SELECT SALESMAN_ID,NAME
FROM SALESMAN A
WHERE 1 <(SELECT COUNT(*) FROM CUSTOMER
WHERE SALESMAN_ID=A.SALESMAN_ID)
```

OR

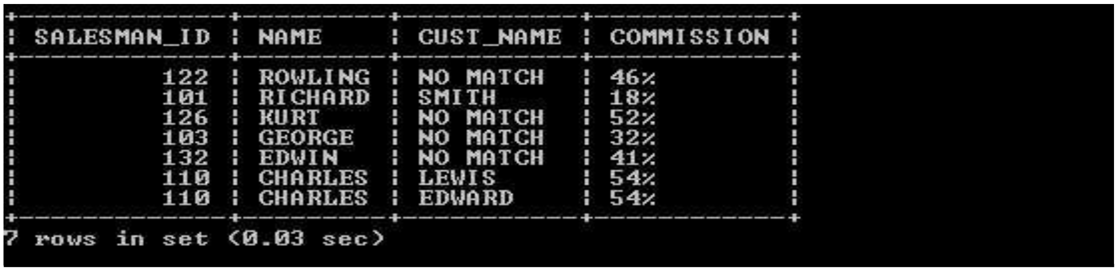
```
SELECT S.SALESMAN_ID,NAME, FROM CUSTOMER
C,SALESMAN S WHERE
S.SALESMAN_ID=C.SALESMAN_ID GROUP BY
C.SALESMAN_ID HAVING COUNT(*)>1
```



```
+-------------+--------+
| SALESMAN_ID | NAME   |
+-------------+--------+
|         103 | GEORGE |
+-------------+--------+
1 row in set (0.00 sec)
```

3. List all salesmen and indicate those who have and don't have customers in their cities (Use UNION operation.)

```
SELECT S.SALESMAN_ID,NAME,CUST_NAME,COMMISSION FROM SALESMAN
S,CUSTOMER C
WHERE S.CITY = C.CITY
UNION
SELECT SALESMAN_ID, NAME, 'NO MATCH',COMMISSION FROM SALESMAN
WHERE NOT CITY = ANY (SELECT CITY
FROM CUSTOMER) ORDER BY 2 DESC;
```

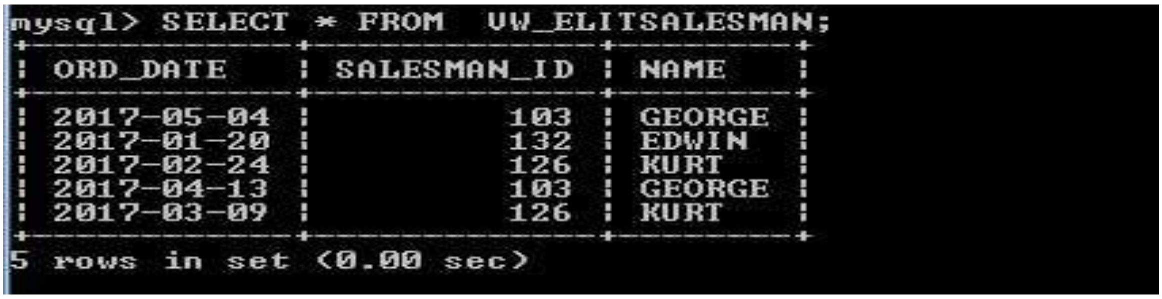


```
+-------------+----------+-----------+------------+
| SALESMAN_ID | NAME     | CUST_NAME | COMMISSION |
+-------------+----------+-----------+------------+
|         122 | ROWLING  | NO MATCH  | 46%        |
|         101 | RICHARD  | SMITH     | 18%        |
|         126 | KURT     | NO MATCH  | 52%        |
|         103 | GEORGE   | NO MATCH  | 32%        |
|         132 | EDWIN    | NO MATCH  | 41%        |
|         110 | CHARLES  | LEWIS     | 54%        |
|         110 | CHARLES  | EDWARD    | 54%        |
+-------------+----------+-----------+------------+
7 rows in set (0.03 sec)
```

4. **Create a view that finds the salesman who has the customer with the highest order of a day.**

CREATE VIEW VW_ELITSALESMAN AS
SELECT B.ORD_DATE,A.SALESMAN_ID,A.NAME FROM
SALESMAN A, ORDERS B WHERE A.SALESMAN_ID =
B.SALESMAN_ID AND B.PURCHASE_AMT=(SELECT
MAX(PURCHASE_AMT) FROM ORDERS C
WHERE C.ORD_DATE =

B.ORD_DATE); SELECT *



```
mysql> SELECT * FROM  VW_ELITSALESMAN;
+------------+-------------+--------+
| ORD_DATE   | SALESMAN_ID | NAME   |
+------------+-------------+--------+
| 2017-05-04 |         103 | GEORGE |
| 2017-01-20 |         132 | EDWIN  |
| 2017-02-24 |         126 | KURT   |
| 2017-04-13 |         103 | GEORGE |
| 2017-03-09 |         126 | KURT   |
+------------+-------------+--------+
5 rows in set (0.00 sec)
```

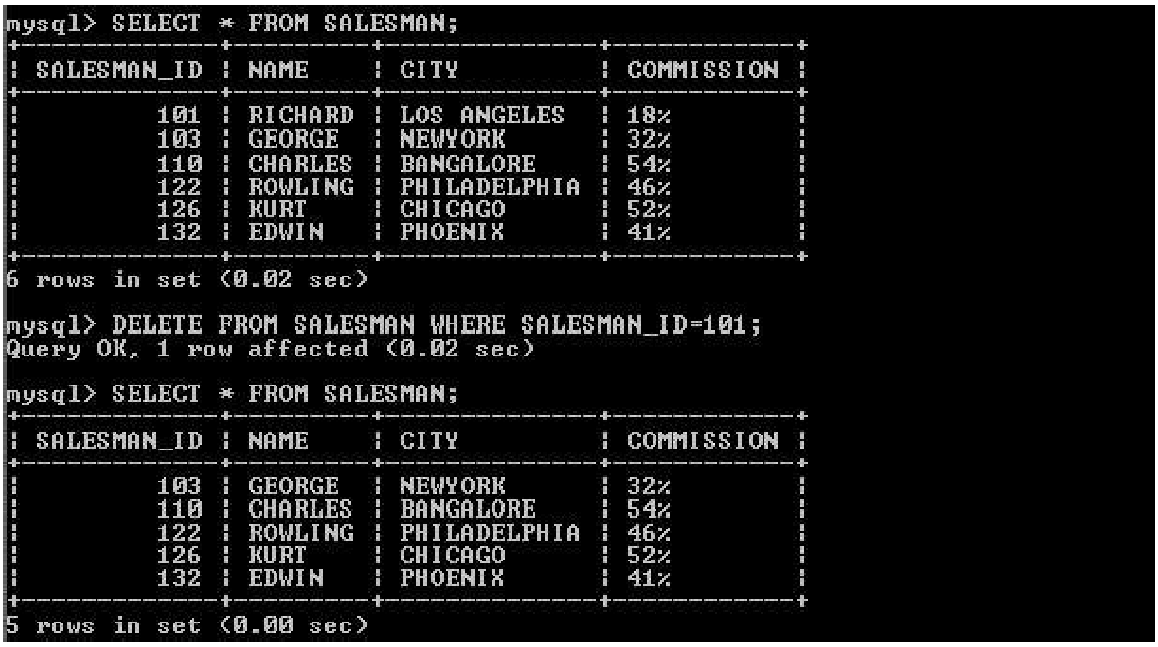
FROM

VW_ELITSALESMAN

5. **Demonstrate the DELETE operation by removing salesman with id 1000. All his orders must also be deleted.**

Use ON DELETE CASCADE at the end of foreign key definitions while creating child table orders and then execute the following:

DELETE FROM SALESMAN WHERE SALESMAN_ID=101;



```
mysql> SELECT * FROM SALESMAN;
+-------------+---------+--------------+------------+
| SALESMAN_ID | NAME    | CITY         | COMMISSION |
+-------------+---------+--------------+------------+
|         101 | RICHARD | LOS ANGELES  | 18%        |
|         103 | GEORGE  | NEWYORK      | 32%        |
|         110 | CHARLES | BANGALORE    | 54%        |
|         122 | ROWLING | PHILADELPHIA | 46%        |
|         126 | KURT    | CHICAGO      | 52%        |
|         132 | EDWIN   | PHOENIX      | 41%        |
+-------------+---------+--------------+------------+
6 rows in set (0.02 sec)

mysql> DELETE FROM SALESMAN WHERE SALESMAN_ID=101;
Query OK, 1 row affected (0.02 sec)

mysql> SELECT * FROM SALESMAN;
+-------------+---------+--------------+------------+
| SALESMAN_ID | NAME    | CITY         | COMMISSION |
+-------------+---------+--------------+------------+
|         103 | GEORGE  | NEWYORK      | 32%        |
|         110 | CHARLES | BANGALORE    | 54%        |
|         122 | ROWLING | PHILADELPHIA | 46%        |
|         126 | KURT    | CHICAGO      | 52%        |
|         132 | EDWIN   | PHOENIX      | 41%        |
+-------------+---------+--------------+------------+
5 rows in set (0.00 sec)
```