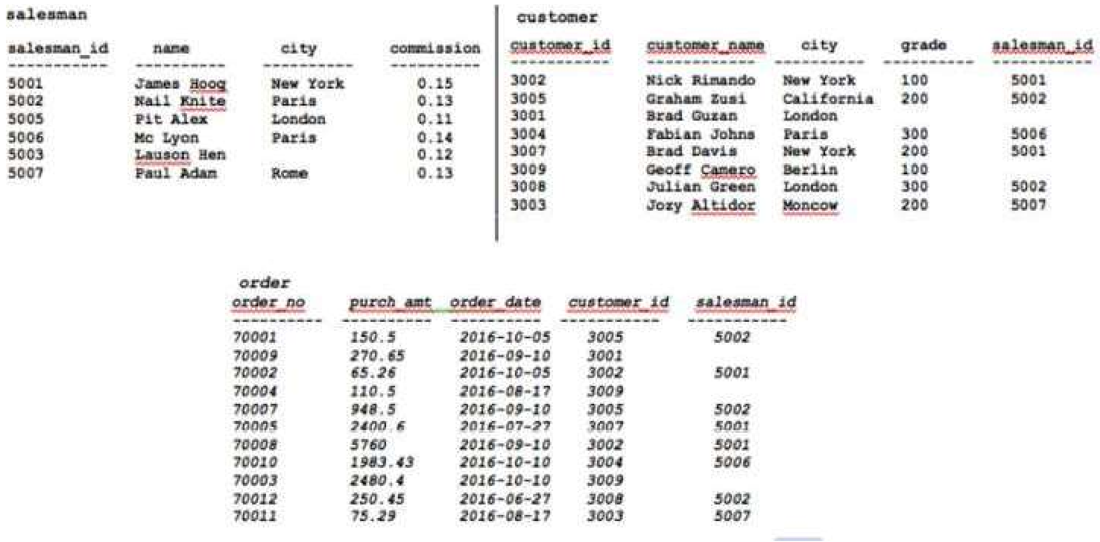
Firdaus Mulla | L007

Mithibai College MSc(Data Sci and AI)

Practical-1 : DDL operations on Relational Schema

Date:-06/12/2024 **Submission Date:- 3/02/2024**

Design the following schema and execute the following queries on it:



salesman

salesman_id	name	city	commission
5001	James Hoog	New York	0.15
5002	Nail Knite	Paris	0.13
5005	Pit Alex	London	0.11
5006	Mc Lyon	Paris	0.14
5003	Lauson Hen		0.12
5007	Paul Adam	Rome	0.13

customer

customer_id	customer_name	city	grade	salesman_id
3002	Nick Rimando	New York	100	5001
3005	Graham Zusi	California	200	5002
3001	Brad Guzan	London		
3004	Fabian Johns	Paris	300	5006
3007	Brad Davis	New York	200	5001
3009	Geoff Camero	Berlin	100	
3008	Julian Green	London	300	5002
3003	Jozy Altidor	Moncow	200	5007

order

order_no	purch_amt	order_date	customer_id	salesman_id
70001	150.5	2016-10-05	3005	5002
70009	270.65	2016-09-10	3001	
70002	65.26	2016-10-05	3002	5001
70004	110.5	2016-08-17	3009	
70007	948.5	2016-09-10	3005	5002
70005	2400.6	2016-07-27	3007	5001
70008	5760	2016-09-10	3002	5001
70010	1983.43	2016-10-10	3004	5006
70003	2480.4	2016-10-10	3009	
70012	250.45	2016-06-27	3008	5002
70011	75.29	2016-08-17	3003	5007

```
mysql> create database mydb;
Query OK, 1 row affected (0.03 sec)

mysql> use mydb;
Database changed

mysql> create table salesman(salesman_id int, name varchar(50), city
varchar(50), commission int, PRIMARY KEY(salesman_id));

mysql> create table salesman(salesman_id int, name varchar(50), city varchar(50),
commission float, PRIMARY KEY(salesman_id));
Query OK, 0 rows affected (0.01 sec)

mysql> insert into salesman values(5001, "James Hoog", "New York", 0.15);
Query OK, 1 row affected (0.00 sec)

mysql> insert into salesman values(5002, "Nail Knite", "Paris", 0.13);
Query OK, 1 row affected (0.00 sec)

mysql> insert into salesman values(5005, "Pit Alex", "London", 0.11);
Query OK, 1 row affected (0.00 sec)
```

Firdaus Mulla | L007

```
mysql> insert into salesman values(5006, "Mc Lyon", "Paris", 0.14);
Query OK, 1 row affected (0.00 sec)

mysql> insert into salesman values(5003, "Lauson Hen", " ", 0.12);
Query OK, 1 row affected (0.00 sec)

mysql> insert into salesman values(5007, "Paul Adam", "Rome", 0.13);
Query OK, 1 row affected (0.00 sec)
Query OK, 0 rows affected (0.01 sec)

mysql> select * from salesman;
+-------------+------------+----------+------------+ |
salesman_id | name       | city     | commission |
+-------------+------------+----------+------------+
|        5001 | James Hoog | New York |       0.15 |
|        5002 | Nail Knite | Paris    |       0.13 |
|        5003 | Lauson Hen |          |       0.12 |
|        5005 | Pit Alex   | London   |       0.11 |
|        5006 | Mc Lyon    | Paris    |       0.14 |
|        5007 | Paul Adam  | Rome     |       0.13 |
+-------------+------------+----------+------------+
6 rows in set (0.00 sec)
```

Customer Table:

```
mysql> create table customer(customer_id int, customer_name varchar(50), city
varchar(50), grade int, salesman_id int, PRIMARY KEY(customer_id), FOREIGN
KEY(salesman_id
) REFERENCES salesman(salesman_id));
Query OK, 0 rows affected (0.01 sec)

mysql> insert into customer values(3002, "Nick Rimando", "New York", 100, 5001);
Query OK, 1 row affected (0.00 sec)

mysql> insert into customer values(3005, "Graham Zusi", "California", 200, 5002);
Query OK, 1 row affected (0.00 sec)

mysql> insert into customer(customer_id, customer_name, city) values(3001, "Brad Guz
an", "London");
Query OK, 1 row affected (0.00 sec)

mysql> insert into customer values(3004, "Fabian Johns", "Paris", 300, 5006);
Query OK, 1 row affected (0.00 sec)
```

Firdaus Mulla | L007

```
mysql> insert into customer values(3007, "Brad Davis", "New York", 200, 5001);
Query OK, 1 row affected (0.00 sec)

mysql> insert into customer(customer_id, customer_name, city, grade) values(3009, "G
eoff Camero", "Berlin", 100);
Query OK, 1 row affected (0.00 sec)

mysql> insert into customer values(3008, "Julian Green", "London", 300, 5002);
Query OK, 1 row affected (0.00 sec)

mysql> insert into customer values(3003, "Jozy Altidor", "Moncow", 200, 5007);
Query OK, 1 row affected (0.00 sec)

mysql> select * from customer
    -> ;
+-------------+---------------+------------+-------+-------------+ |
customer_id | customer_name | city       | grade | salesman_id |
+-------------+---------------+------------+-------+-------------+
|        3001 | Brad Guzan    | London     |  NULL |        NULL |
|        3002 | Nick Rimando  | New York   |   100 |        5001 |
|        3003 | Jozy Altidor  | Moncow     |   200 |        5007 |
|        3004 | Fabian Johns  | Paris      |   300 |        5006 |
|        3005 | Graham Zusi   | California |   200 |        5002 | |
3007 | Brad Davis    | New York   |   200 |        5001 |
|        3008 | Julian Green  | London     |   300 |        5002 |
|        3009 | Geoff Camero  | Berlin     |   100 |        NULL |
+-------------+---------------+------------+-------+-------------+
8 rows in set (0.00 sec)
```

Order:

```
mysql> CREATE TABLE orders(order_no INT, purch_amt FLOAT, order_date
DATE, customer_id INT, salesman_id INT, PRIMARY KEY(order_no), FOREIGN
KEY(customer_id) REFERENCES customer(customer_id), FOREIGN
KEY(salesman_id) REFERENCES salesman(salesman_id));
Query OK, 0 rows affected (0.01 sec)
```

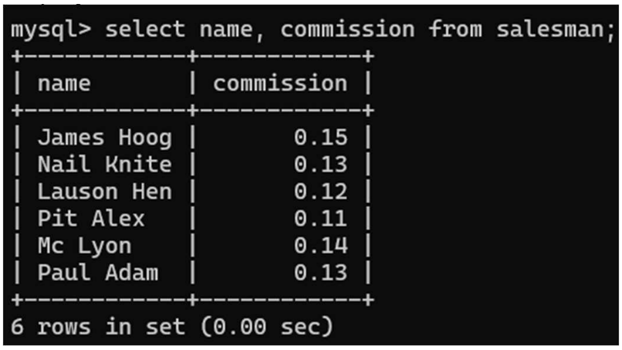
Inserting

```
mysql> INSERT INTO orders values(70001, 150.5, '2016-10-05', 3005,
5002);
Query OK, 1 row affected (0.00 sec)
```

Firdaus Mulla | L007

```
mysql> INSERT INTO orders(order_no, purch_amt, order_date, customer_id)
values(70009 , 270.65, '2016-09-10', 3001);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO orders values(70002, 65.26, '2016-10-05', 3002, 5001);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO orders(order_no, purch_amt, order_date, customer_id)
values(70004, 110.5, '2016-08-17', 3009);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO orders values(70007, 948.5, '2016-09-10', 3005, 5002);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO orders values(70005, 2400.6, '2016-07-27', 3007, 5001);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO orders values(70008, 5760, '2016-09-10', 3002, 5001);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO orders values(70010, 1983.43, '2016-10-10', 3004, 5006);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO orders(order_no, purch_amt, order_date, customer_id)
values(70003, 2480.4, '2016-10-10', 3009);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO orders values(70012, 250.45, '2016-06-27', 3008, 5002);
Query OK, 1 row affected (0.00 sec)

mysql> INSERT INTO orders values(70011, 75.29, '2016-08-17', 3003, 5007);
Query OK, 1 row affected (0.00 sec)

mysql> select * from orders;
+----------+-----------+------------+-------------+-------------+
| order_no | purch_amt | order_date | customer_id | salesman_id |
+----------+-----------+------------+-------------+-------------+
|    70001 |     150.5 | 2016-10-05 |        3005 |        5002 |
|    70002 |     65.26 | 2016-10-05 |        3002 |        5001 |
|    70003 |    2480.4 | 2016-10-10 |        3009 |        NULL |
|    70004 |     110.5 | 2016-08-17 |        3009 |        NULL |
|    70005 |    2400.6 | 2016-07-27 |        3007 |        5001 |
|    70007 |     948.5 | 2016-09-10 |        3005 |        5002 |
|    70008 |      5760 | 2016-09-10 |        3002 |        5001 |
|    70009 |    270.65 | 2016-09-10 |        3001 |        NULL |
|    70010 |   1983.43 | 2016-10-10 |        3004 |        5006 |
|    70011 |     75.29 | 2016-08-17 |        3003 |        5007 |
|    70012 |    250.45 | 2016-06-27 |        3008 |        5002 |
+----------+-----------+------------+-------------+-------------+
11 rows in set (0.00 sec)
```

1. Display name and commission for all the salesmen.



```
mysql> select name, commission from salesman;
+------------+------------+
| name       | commission |
+------------+------------+
| James Hoog |       0.15 |
| Nail Knite |       0.13 |
| Lauson Hen |       0.12 |
| Pit Alex   |       0.11 |
| Mc Lyon    |       0.14 |
| Paul Adam  |       0.13 |
+------------+------------+
6 rows in set (0.00 sec)
```

mysql> select name, commission from salesman;

```
+------------+------------+ | name
| commission |
+------------+------------+
| James Hoog |       0.15 |
| Nail Knite |       0.13 |
| Lauson Hen |       0.12 |
| Pit Alex   |       0.11 |
| Mc Lyon    |       0.14 |
| Paul Adam  |       0.13 |
+------------+------------+
6 rows in set (0.00 sec)
```

2. Retrieve salesman id of all salesmen from orders table without any repeats.

mysql> select DISTINCT salesman_id from orders;

```
+-------------+
| salesman_id |
+-------------+
|        NULL |
|        5001 |
|        5002 |
|        5006 |
|        5007 |
+-------------+
5 rows in set (0.00 sec)
```

3. Display names and city of salesman, who belongs to the city of Paris.
   mysql> select name, city from salesman where city = "Paris";

```
+------------+-------+
| name       | city  |
+------------+-------+
| Nail Knite | Paris |
| Mc Lyon    | Paris |
+------------+-------+
2 rows in set (0.00 sec)
```

4. Display all the information for those customers with a grade of 200.
   mysql> select * from customer where grade = 200;

```
+-------------+---------------+------------+-------+-------------+ |
customer_id | customer_name | city       | grade | salesman_id |
+-------------+---------------+------------+-------+-------------+
|        3003 | Jozy Altidor  | Moncow     |   200 |        5007 |
|        3005 | Graham Zusi   | California |   200 |        5002 | |
3007 | Brad Davis    | New York   |   200 |        5001 |
+-------------+---------------+------------+-------+-------------+
3 rows in set (0.00 sec)
```

5. Display the order number, order date and the purchase amount for order(s) which will be delivered by the salesman with ID 5001
   mysql> select order_no, purch_amt, order_date from orders where salesman_id = 5001;

```
+----------+-----------+------------+
| order_no | purch_amt | order_date |
+----------+-----------+------------+
|    70002 |     65.26 | 2016-10-05 |
|    70005 |    2400.6 | 2016-07-27 |
|    70008 |      5760 | 2016-09-10 |
+----------+-----------+------------+
3 rows in set (0.00 sec)
```

6. Show the winner of the 1971 prize for Literature.
7. Show all the details of the winners with first name Louis.
8. Show all the winners in Physics for 1970 together with the winner of Economics for 1971.

9. Show all the winners of Nobel prize in the year 1970 except the subject Physiology and Economics.
10. Find all the details of the Nobel winners for the subject not started with the letter 'P' and arranged the list as the most recent comes first, then by name in order.

11. Find the name and price of the cheapest item(s).

12. Display all the customers, who are either belongs to the city New York or not had a grade above 100.

```
mysql> select * from customer where city = "New York" OR Not grade > 100;
+-------------+---------------+----------+-------+-------------+ |
customer_id | customer_name | city     | grade | salesman_id |
+-------------+---------------+----------+-------+-------------+
|        3002 | Nick Rimando  | New York |   100 |        5001 |
|        3007 | Brad Davis    | New York |   200 |        5001 |
|        3009 | Geoff Camero  | Berlin   |   100 |        NULL |
+-------------+---------------+----------+-------+-------------+ 3 rows in
set (0.00 sec)
```

13. Find those salesmen with all information who gets the commission within a range of 0.12 and 0.14.

```
mysql> select * from salesman where commission between 0.12 and 0.14;
+-------------+------------+-------+------------+ |
salesman_id | name       | city  | commission |
+-------------+------------+-------+------------+
|        5002 | Nail Knite | Paris |       0.13 |
|        5007 | Paul Adam  | Rome  |       0.13 |
+-------------+------------+-------+------------+
2 rows in set (0.00 sec)
```

14. Find all those customers with all information whose names are ending with the letter 'n'.

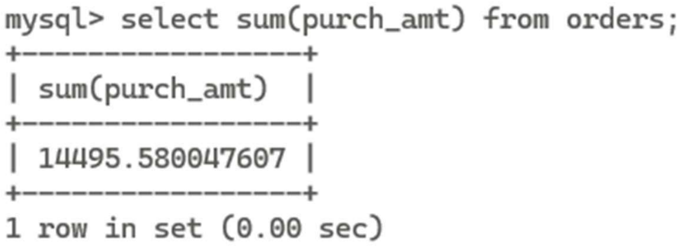
```
mysql> select * from customer where customer_name LIKE "%n";
+-------------+---------------+--------+-------+-------------+ |
customer_id | customer_name | city   | grade | salesman_id |
+-------------+---------------+--------+-------+-------------+
|        3001 | Brad Guzan    | London |  NULL |        NULL |
|        3008 | Julian Green  | London |   300 |        5002 |
+-------------+---------------+--------+-------+-------------+
2 rows in set (0.00 sec)
```

Firdaus Mulla | L007

15. Find those salesmen with all information whose name containing the 1st character is 'N' and the 4th character is 'l' and rests may be any character.
    mysql> select * from salesman where name LIKE "N__l%";
    ```
    +-------------+------------+-------+------------+ |
    salesman_id | name       | city  | commission |
    +-------------+------------+-------+------------+ |        5002
    | Nail Knite | Paris |       0.13 |
    +-------------+------------+-------+------------+
    1 row in set (0.00 sec)
    ```

16. Find that customer with all information who does not get any grade except NULL.
    mysql> select * from customer where grade is NULL;
    ```
    +-------------+---------------+--------+-------+-------------+ |
    customer_id | customer_name | city   | grade | salesman_id |
    +-------------+---------------+--------+-------+-------------+ |       3001
    | Brad Guzan    | London |  NULL |        NULL |
    +-------------+---------------+--------+-------+-------------+ 1 row in set
    (0.00 sec)
    ```

17. Find the total purchase amount of all orders. mysql> select sum(purch_amt) from orders;
    ```
    +-----------------+
    | sum(purch_amt)  |
    +-----------------+
    | 14495.580047607 |
    +-----------------+
    1 row in set (0.00 sec)
    ```



```
mysql> select sum(purch_amt) from orders;
+-----------------+
| sum(purch_amt)  |
+-----------------+
| 14495.580047607 |
+-----------------+
1 row in set (0.00 sec)
```

18. Find the number of salesman currently listing for all of their customers.
    mysql> select count(Distinct salesman_id) as no_salesman from customer;
    ```
    +-------------+
    | no_salesman |
    +-------------+
    |           4 |
    +-------------+
    1 row in set (0.00 sec)
    ```

19. Find the highest grade for each of the cities of the customers.

```
mysql> SELECT city, MAX(grade) AS highest_grade
    -> FROM customer
    -> GROUP BY city;
+------------+---------------+ | city       |
highest_grade |
+------------+---------------+
| Berlin     |           100 |
| California |           200 |
| London     |           300 |
| Moncow     |           200 |
| New York   |           200 |
| Paris      |           300 |
+------------+---------------+
6 rows in set (0.00 sec)
```

20. Find the highest purchase amount ordered by each customer with their ID and highest purchase amount.

```
mysql> select customer_id, max(purch_amt) as highest_purchase_amt from orders
group by customer_id;
+-------------+----------------------+
| customer_id | highest_purchase_amt |
+-------------+----------------------+
|        3001 |      270.64999389648 |
|        3002 |                 5760 |
|        3003 |      75.290000915527 | |
3004 |      1983.4300537109 | |        3005
|                  948.5 |
|        3007 |      2400.6000976563 | |
3008 |      250.44999694824 | |        3009 |
2480.3999023438 |
+-------------+----------------------+
8 rows in set (0.00 sec)
```

21. Find the highest purchase amount ordered by each customer on a particular date with their ID, order date and highest purchase amount.

```
mysql> select customer_id, order_date, max(purch_amt) as highest_purchase from
orders group by customer_id, order_date;
+-------------+------------+------------------+
| customer_id | order_date | highest_purchase |
+-------------+------------+-----------------+
|        3001 | 2016-09-10 |          270.65 |
|        3002 | 2016-09-10 |            5760 |
|        3002 | 2016-10-05 |           65.26 |
|        3003 | 2016-08-17 |           75.29 |
|        3004 | 2016-10-10 |         1983.43 |
|        3005 | 2016-09-10 |           948.5 |
|        3005 | 2016-10-05 |           150.5 |
|        3007 | 2016-07-27 |          2400.6 |
|        3008 | 2016-06-27 |          250.45 | |
3009 | 2016-08-17 |           110.5 |
|        3009 | 2016-10-10 |          2480.4 |
+-------------+------------+-----------------+
11 rows in set (0.00 sec)
```

22. Find the highest purchase amount on a date '2012-08-17' for each salesman with their ID.
mysql> select salesman_id, max(purch_amt) as highest_purchase from orders where order_date = "2016-08-17" group by salesman_id;

```
+-------------+-----------------+
| salesman_id | highest_purchase |
+-------------+-----------------+
|        NULL |           110.5 |
|        5007 |  75.290000915527 |
+-------------+-----------------+
2 rows in set (0.00 sec)
```

23. Find the highest purchase amount with their customer ID and order date, for only those customers who have the highest purchase amount in a day is more than 2000.
mysql> select customer_id, max(purch_amt) as highest_purchase, order_date from orders where customer_id in (select customer_id from orders group by customer_id, order_date having max(purch_amt) > 2000) group by customer_id, order_date order by highest_purchase desc;

```
+-------------+-----------------+------------+
| customer_id | highest_purchase | order_date |
+-------------+-----------------+------------+
|        3002 |            5760 | 2016-09-10 |
|        3009 |          2480.4 | 2016-10-10 |
|        3007 |          2400.6 | 2016-07-27 |
|        3009 |           110.5 | 2016-08-17 |
|        3002 |           65.26 | 2016-10-05 |
+-------------+-----------------+------------+
```

Firdaus Mulla | L007

24. Write a SQL statement that counts all orders for a date August 17th, 2016.

```
mysql> select count(*) as total_orders from orders where order_date = "201608-17";
+--------------+ |
total_orders |
+--------------+
|            2 |
+--------------+
1 row in set (0.00 sec)
```

25. Count the customers with grades above Bangalore’s average.

```
mysql> select count(*) as customers_above_avg from customer where grade >
(select coalesce(avg(grade), 0) from customer where city = "BANGLORE");
+---------------------+
| customers_above_avg |
+---------------------+
|                   7 |
+---------------------+
1 row in set (0.00 sec)
```

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

```
mysql> Select name, salesman_id From salesman Where salesman_id IN(Select
salesman_id From customer Group By salesman_id Having Count(customer_id) >
1);
+------------+-------------+
| name       | salesman_id |
+------------+-------------+
| James Hoog |        5001 |
| Nail Knite |        5002 |
+------------+-------------+
2 rows in set (0.01 sec)
```

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

```
mysql> Select s.name, s.salesman_id, "Have Customers" As customer_status from
salesman s
    -> Join customer c On s.salesman_id = c.salesman_id
    -> Where s.city = c.city Group by s.salesman_id
    -> UNION
    -> Select s.name, s.salesman_id, "Don not have Customers" As cutomer_status
from salesman s
    -> LEFT Join customer c on s.salesman_id = c.salesman_id
```

```
   -> Where c.city IS NULL or s.city != c.city;
+------------+-------------+-----------------------+
| name       | salesman_id | customer_status       |
+------------+-------------+-----------------------+
| James Hoog |        5001 | Have Customers        |
| Mc Lyon    |        5006 | Have Customers        |
| Nail Knite |        5002 | Don not have Customers |
| Lauson Hen |        5003 | Don not have Customers |
| Pit Alex   |        5005 | Don not have Customers |
| Paul Adam  |        5007 | Don not have Customers |
+------------+-------------+-----------------------+
6 rows in set (0.47 sec)
```

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

```
mysql> create view highest_order_salesman As
   -> Select o.order_date, o.salesman_id, s.name As salesman_name,
c.customer_name, o.purch_amt from orders o
   -> Join salesman s on o.salesman_id = s.salesman_id
   -> Join customer c on o.customer_id = c.customer_id
   -> Where(o.order_date, o.purch_amt) In (Select order_date, Max(purch_amt)
from orders Group by order_date);
Query OK, 0 rows affected (0.98 sec)
mysql> Select * from highest_order_salesman;
+------------+-------------+---------------+---------------+-----------+
| order_date | salesman_id | salesman_name | customer_name | purch_amt |
+------------+-------------+---------------+---------------+-----------+
| 2016-09-10 |        5001 | James Hoog    | Nick Rimando  |      5760 |
| 2016-10-05 |        5002 | Nail Knite    | Graham Zusi   |     150.5 |
| 2016-07-27 |        5001 | James Hoog    | Brad Davis    |    2400.6 |
| 2016-06-27 |        5002 | Nail Knite    | Julian Green  |    250.45 |
+------------+-------------+---------------+---------------+-----------+
4 rows in set (0.05 sec)
```

29. Demonstrate the DELETE operation by removing salesman with id 5002. All his orders must also be deleted

```
mysql> Delete from customer where salesman_id = 5002;
Query OK, 2 rows affected (0.13 sec)

mysql> Delete from orders where salesman_id = 5002;
Query OK, 3 rows affected (0.09 sec)
```

```
mysql> Delete from orders where salesman_id = 5002;
Query OK, 0 rows affected (0.00 sec)

mysql> select * from salesman;
+-------------+------------+----------+------------+
| salesman_id | name       | city     | commission |
+-------------+------------+----------+------------+
|        5001 | James Hoog | New York |       0.15 |
|        5003 | Lauson Hen |          |       0.12 |
|        5005 | Pit Alex   | London   |       0.11 |
|        5006 | Mc Lyon    | Paris    |       0.14 |
|        5007 | Paul Adam  | Rome     |       0.13 |
+-------------+------------+----------+------------+
5 rows in set (0.00 sec)
mysql> select * from orders;
+----------+-----------+------------+-------------+-------------+
| order_no | purch_amt | order_date | customer_id | salesman_id |
+----------+-----------+------------+-------------+-------------+
|    70002 |     65.26 | 2016-10-05 |        3002 |        5001 |
|    70003 |    2480.4 | 2016-10-10 |        3009 |        NULL |
|    70004 |     110.5 | 2016-08-17 |        3009 |        NULL |
|    70005 |    2400.6 | 2016-07-27 |        3007 |        5001 |
|    70008 |      5760 | 2016-09-10 |        3002 |        5001 |
|    70009 |    270.65 | 2016-09-10 |        3001 |        NULL |
|    70010 |   1983.43 | 2016-10-10 |        3004 |        5006 |
|    70011 |     75.29 | 2016-08-17 |        3003 |        5007 |
+----------+-----------+------------+-------------+-------------+
8 rows in set (0.00 sec)
```

****