Practical-1 : DDL operations on Relational Schema

```
mysql> create database mydb;
Query OK, 1 row affected (0.01 sec)

mysql> use mydb;
Database changed
mysql>
mysql> create table salesman(salesman_id int PRIMARY KEY,name varchar(20),city varchar(10),commission float);
Query OK, 0 rows affected (0.05 sec)
```

```
mysql> insert into salesman values(5001,'James Hoog','New York',0.15);
Query OK, 1 row affected (0.01 sec)

mysql> insert into salesman values(5002,'Nail Knite','Paris',0.13);
Query OK, 1 row affected (0.01 sec)

mysql> insert into salesman values(5005,'Pit Alex','London',0.11);
Query OK, 1 row affected (0.01 sec)

mysql> insert into salesman values(5006,'Mc Lyon','Paris',0.14);
Query OK, 1 row affected (0.01 sec)

mysql> insert into salesman values(5003,'Lauson Hen','',0.12);
Query OK, 1 row affected (0.01 sec)

mysql> insert into salesman values(5007,'Paul Adam','Rome',0.13);
Query OK, 1 row 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)
```

```
mysql> create table customer(customer_id int PRIMARY KEY,customer_name varchar(20),city varcha
r(15),grade int,salesman_id int, FOREIGN KEY(salesman_id) REFERENCES salesman(salesman_id));
Query OK, 0 rows affected (0.04 sec)

mysql> insert into customer values(3002,'Nick Rimando','New York',100,5001);
Query OK, 1 row affected (0.01 sec)

mysql> insert into customer values(3005,'Graham Zusi','California',200,5002);
Query OK, 1 row affected (0.01 sec)
```

```
mysql> insert into customer(customer_id,customer_name,city) values(3001,'Bra
d Guran','London');
Query OK, 1 row affected (0.01 sec)

mysql> insert into customer values(3004,'Fabian Johna','Paris',300,5006);
Query OK, 1 row affected (0.01 sec)

mysql> insert into customer values(3007,'Brad Davis','New York',200,5001);
Query OK, 1 row affected (0.01 sec)

mysql> insert into customer(customer_id,customer_name,city,grade) values(300
9,'Geoff Camero','Berlin',100);
Query OK, 1 row affected (0.01 sec)

mysql> insert into customer values(3008,'Julian Green','London',300,5002);
Query OK, 1 row affected (0.01 sec)

mysql> insert into customer values(3003,'Jozy Altidor','Moncow',200,5007);
Query OK, 1 row affected (0.01 sec)
```

```
mysql> select * from customer;
+-------------+---------------+------------+-------+-------------+
| customer_id | customer_name | city       | grade | salesman_id |
+-------------+---------------+------------+-------+-------------+
|        3001 | Brad Guran    | London     |  NULL |        NULL |
|        3002 | Nick Rimando  | New York   |   100 |        5001 |
|        3003 | Jozy Altidor  | Moncow     |   200 |        5007 |
|        3004 | Fabian Johna  | 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)
```

```
mysql> CREATE TABLE orders (
    ->     order_no INT PRIMARY KEY,
    ->     purch_amt FLOAT,
    ->     order_date DATE,
    ->     customer_id INT,
    ->     salesman_id INT,
    ->     FOREIGN KEY (customer_id) REFERENCES customer(customer_id),
    ->     FOREIGN KEY (salesman_id) REFERENCES salesman(salesman_id)
    -> );
Query OK, 0 rows affected (0.04 sec)

mysql> insert into orders values(70001,150.5,'2016-10-05',3005,5002);
Query OK, 1 row affected (0.01 sec)

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 (0.01 sec)

mysql> insert into orders values(70002,65.26,'2016-10-05',3002,5001);
Query OK, 1 row affected (0.01 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 (0.01 sec)

mysql> insert into orders values(70007,948.5,'2016-09-10',3005,5002);
Query OK, 1 row affected (0.01 sec)

mysql> insert into orders values(70005,2400.6,'2016-07-27',3007,5001);
Query OK, 1 row affected (0.01 sec)

mysql> insert into orders values(70008,5760,'2016-09-10',3002,5001);
Query OK, 1 row affected (0.01 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 (0.01 sec)

mysql> insert into orders values(70012,250.45,'2016-06-27',3008,5002);
Query OK, 1 row affected (0.01 sec)

mysql> insert into orders values(70011,75.29,'2016-08-17',3003,5007);
Query OK, 1 row affected (0.01 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)
```

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,order_date,purch_amt from orders where salesman_id=5001;
+----------+------------+-----------+
| order_no | order_date | purch_amt |
+----------+------------+-----------+
|    70002 | 2016-10-05 |     65.26 |
|    70005 | 2016-07-27 |    2400.6 |
|    70008 | 2016-09-10 |      5760 |
+----------+------------+-----------+
3 rows in set (0.00 sec)
```

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 (0.02 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 Guran    | London |  NULL |        NULL |
|        3008 | Julian Green  | London |   300 |        5002 |
+-------------+---------------+--------+-------+-------------+
2 rows in set (0.00 sec)
```

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 Guran    | London |  NULL |        NULL |
+-------------+---------------+--------+-------+-------------+
1 row in set (0.00 sec)
```

17. Find the total purchase amount of all orders.

```
mysql> select sum(purch_amt) as total_purchase_amt from orders;
+--------------------+
| total_purchase_amt |
+--------------------+
| 14495.580047607422 |
+--------------------+
1 row in set (0.01 sec)
```

18. Find the number of salesman currently listing for all of their customers.

```
mysql> SELECT COUNT(*) AS salesman_count
    -> FROM Salesman s
    -> WHERE NOT EXISTS (
    ->     SELECT *
    ->     FROM Customer c
    ->     WHERE NOT EXISTS (
    ->         SELECT 1
    ->         FROM Orders o
    ->         WHERE o.salesman_id = s.salesman_id
    ->           AND o.customer_id = c.customer_id
    ->     )
    -> );
+----------------+
| salesman_count |
+----------------+
|              0 |
+----------------+
1 row in set (0.01 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 |
+------------+---------------+
| London     |           300 |
| New York   |           200 |
| Moncow     |           200 |
| Paris      |           300 |
| California |           200 |
| Berlin     |           100 |
+------------+---------------+
6 rows in set (0.01 sec)
```

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

```
mysql> select o.customer_id,max(o.purch_amt) as highest_purchase_amt from or
ders o group by o.customer_id;
+-------------+----------------------+
| customer_id | highest_purchase_amt |
+-------------+----------------------+
|        3001 |               270.65 |
|        3002 |                 5760 |
|        3003 |                75.29 |
|        3004 |              1983.43 |
|        3005 |                948.5 |
|        3007 |               2400.6 |
|        3008 |               250.45 |
|        3009 |               2480.4 |
+-------------+----------------------+
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_amt
from orders  group by customer_id,order_date;
+-------------+------------+----------------------+
| customer_id | order_date | highest_purchase_amt |
+-------------+------------+----------------------+
|        3005 | 2016-10-05 |                150.5 |
|        3002 | 2016-10-05 |                65.26 |
|        3009 | 2016-10-10 |               2480.4 |
|        3009 | 2016-08-17 |                110.5 |
|        3007 | 2016-07-27 |               2400.6 |
|        3005 | 2016-09-10 |                948.5 |
|        3002 | 2016-09-10 |                 5760 |
|        3001 | 2016-09-10 |               270.65 |
|        3004 | 2016-10-10 |              1983.43 |
|        3003 | 2016-08-17 |                75.29 |
|        3008 | 2016-06-27 |               250.45 |
+-------------+------------+----------------------+
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_amt
    -> from orders
    -> where order_date = '2012-08-17'
    -> group by salesman_id;
Empty 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,order_date,max(purch_amt) as highest_purchase_amt
    -> from orders
    -> group by customer_id,order_date
    -> having max(purch_amt) > 2000;
+-------------+------------+---------------------+
| customer_id | order_date | highest_purchase_amt |
+-------------+------------+---------------------+
|        3009 | 2016-10-10 |               2480.4 |
|        3007 | 2016-07-27 |               2400.6 |
|        3002 | 2016-09-10 |                 5760 |
+-------------+------------+---------------------+
3 rows in set (0.01 sec)
```

24.Write a SQL statement that counts all orders for a date August 17th, 2012.

```
mysql> select count(order_no) as order_count
    -> from orders
    -> where order_date='2012-08-17';
+-------------+
| order_count |
+-------------+
|           0 |
+-------------+
1 row in set (0.00 sec)
```