PRACTICAL 1
L030

```
 CREATE TABLE salesman (
   ->     salesman_id INT PRIMARY KEY,
   ->     name VARCHAR(50),
   ->     city VARCHAR(50),
   ->     commission DECIMAL(5, 2)
   -> );
Query OK, 0 rows affected (0.00 sec)

mysql>
mysql> INSERT INTO salesman (salesman_id, name, city, commission)
   -> VALUES
   ->     (5001, 'James Hoog', 'New York', 0.15),
   ->     (5002, 'Nail Knite', 'Paris', 0.13),
   ->     (5005, 'Pit Alex', 'London', 0.11),
   ->     (5006, 'Mac Lyon', 'Paris', 0.14),
   ->     (5007, 'Paul Adam', 'Rome', 0.13);

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 | Mac Lyon   | Paris    |       0.14 |
|        5007 | Paul Adam  | Rome     |       0.13 |
+-------------+------------+----------+------------+


mysql> create table customer (custid int Primary key , cust_name varchar(50)
,city varchar(50), grade int ,salesman_id int, foreign key (salesman_id) references
salesman (salesman_id));
Query OK, 0 rows affected (0.01 sec)
mysql> select * from customer;
+--------+----------------+------------+-------+-------------+
| custid | cust_name      | city       | grade | salesman_id |
```

```
+--------+----------------+------------+-------+-------------+
|  3001 | Brad gauz      | London     |  NULL |       NULL |
|  3002 | Nick Rimando   | New York   |  100 |       5001 |
|  3003 | Fabian Johnson | Paris      |  200 |       5006 |
|  3004 | Geoff Cameron  | Berlin     |  100 |       NULL |
|  3005 | Graham Zusi    | California |  200 |       5002 |
|  3007 | Brad Davis     | New York   |  200 |       5001 |
|  3008 | Julian Green   | London     |  300 |       5002 |
|  3009 | Jozy Altidore  | Moscow     |  200 |       5007 |
+--------+----------------+------------+-------+-------------+
```

```
mysql> create table order_table (orderno int Primary key , purch_amt
decimal(10,2), order_date date , custid int , salesman_id int , foreign key(cust
id) references customer (custid),foreign key (salesman_id) references salesman
(salesman_id));

INSERT INTO order_table (orderno, purch_amt, order_date, custid, salesman_id)
VALUES
(70001, 150.5, '2016-10-05', 3005, 5002),
 (70009, 270.65, '2016-09-10', 3001, null),
 (70002, 65.26, '2016-10-05', 3002, 5001),
 (70004, 110.5, '2016-08-17', 3009, null),
 (70007, 948.5, '2016-11-30', 3005, 5002),
 (70005, 2400.6, '2016-07-13', 3007, 5001),
 (70008, 5760, '2016-09-10', 3002, 5001),
 (70010, 1983.43, '2016-10-10', 3004, 5006),
(70003, 2480.4, '2016-10-10', 3009, null),
(70011, 75.29, '2016-06-17', 3003, 5007);
mysql> select * from order_table;
+---------+-----------+------------+--------+-------------+
| orderno | purch_amt | order_date | custid | salesman_id |
+---------+-----------+------------+--------+-------------+
|  70001 |    150.50 | 2016-10-05 |  3005 |       5002 |
|  70002 |     65.26 | 2016-10-05 |  3002 |       5001 |
|  70003 |   2480.40 | 2016-10-10 |  3009 |       NULL |
|  70004 |    110.50 | 2016-08-17 |  3009 |       NULL |
|  70005 |   2400.60 | 2016-07-13 |  3007 |       5001 |
|  70007 |    948.50 | 2016-11-30 |  3005 |       5002 |
```

```
|   70008 |   5760.00 | 2016-09-10 |   3002 |       5001 |
|   70009 |    270.65 | 2016-09-10 |   3001 |       NULL |
|   70010 |   1983.43 | 2016-10-10 |   3004 |       5006 |
|   70011 |     75.29 | 2016-06-17 |   3003 |       5007 |
+---------+-----------+------------+--------+------------+
```

Q1Display 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 |
| Mac Lyon   |       0.14 |
| Paul Adam  |       0.13 |
+------------+------------+
```

Q2.Retrieve salesman id of all salesmen from orders table without any repeats.

```
mysql> SELECT distinct salesman_id from order_table;
+-------------+
| salesman_id |
+-------------+
|        NULL |
|        5001 |
|        5002 |
|        5006 |
|        5007 |
+-------------+
```

Q3.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 |
| Mac Lyon   | Paris |
+------------+-------+
```

Q4.Display all the information for those customers with a grade of 200.

```
mysql> select * from customer where grade=200;
+--------+---------------+------------+-------+-------------+
| custid | cust_name     | city       | grade | salesman_id |
+--------+---------------+------------+-------+-------------+
|   3003 | Fabian Johnson | Paris      |   200 |        5006 |
|   3005 | Graham Zusi    | California |   200 |        5002 |
|   3007 | Brad Davis     | New York   |   200 |        5001 |
|   3009 | Jozy Altidore  | Moscow     |   200 |        5007 |
+--------+---------------+------------+-------+-------------+
```

Q5.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 orderno,purch_amt,order_date from order_table where
salesman_id
=5001;
+---------+-----------+------------+
| orderno | purch_amt | order_date |
+---------+-----------+------------+
|   70002 |     65.26 | 2016-10-05 |
|   70005 |   2400.60 | 2016-07-13 |
|   70008 |   5760.00 | 2016-09-10 |
+---------+-----------+------------+
```

Q6.Display all the customers, who either belong to the city New York or not had a grade above 100.

```
mysql> select * from customer where city="New York" OR grade<100;
+--------+--------------+----------+-------+-------------+
| custid | cust_name    | city     | grade | salesman_id |
+--------+--------------+----------+-------+-------------+
|   3002 | Nick Rimando | New York |   100 |        5001 |
|   3007 | Brad Davis   | New York |   200 |        5001 |
+--------+--------------+----------+-------+-------------+
```

Q7.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 >= 0.12 and commission <=
0.14
;
+-------------+------------+-------+------------+
```

```
| salesman_id | name       | city  | commission |
+-------------+------------+-------+------------+
|       5002 | Nail Knite | Paris |       0.13 |
|       5003 | Lauson hen |       |       0.12 |
|       5006 | Mac Lyon   | Paris |       0.14 |
|       5007 | Paul Adam  | Rome  |       0.13 |
+-------------+------------+-------+------------+
```

q8.Find all those customers with all information whose names are ending with the letter 'n'.

```
mysql> select * from customer where cust_name Like "%n";
+--------+---------------+--------+-------+-------------+
| custid | cust_name     | city   | grade | salesman_id |
+--------+---------------+--------+-------+-------------+
|   3003 | Fabian Johnson | Paris  |   200 |        5006 |
|   3004 | Geoff Cameron  | Berlin |   100 |        NULL |
|   3008 | Julian Green   | London |   300 |        5002 |
+--------+---------------+--------+-------+-------------+
```

Q9.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__i%";
Empty set (0.00 sec)
```

Q10.Find that customer with all information who does not get any grade except NULL.

```
mysql> select * from customer where grade="NULL";
Empty set, 7 warnings (0.00 sec)
```

Find the total purchase amount of all orders.

```
mysql> SELECT SUM(purch_amt) AS total_purchase_amount
    -> FROM order_table;
+----------------------+
| total_purchase_amount |
+----------------------+
|              14245.13 |
+----------------------+
```

Q12.Find the number of salesman currently listing for all of their customers.

```
mysql> SELECT COUNT(DISTINCT salesman_id) AS number_of_salesmen
```

```
   -> FROM customer
   -> WHERE salesman_id IS NOT NULL;
+--------------------+
| number_of_salesmen |
+--------------------+
|                  4 |
+--------------------+
```

Q13.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 |
| Moscow     |           200 |
| New York   |           200 |
| Paris      |           200 |
+------------+---------------+
```

Q14.Find the highest purchase amount ordered by each customer with their ID and highest purchase amount.

```
mysql> SELECT custid, MAX(purch_amt) AS highest_purchase_amount
   -> FROM order_table
   -> GROUP BY custid;
+--------+------------------------+
| custid | highest_purchase_amount |
+--------+------------------------+
|   3001 |                 270.65 |
|   3002 |                5760.00 |
|   3003 |                  75.29 |
|   3004 |                1983.43 |
|   3005 |                 948.50 |
|   3007 |                2400.60 |
|   3009 |                2480.40 |
+--------+------------------------+
```

Q15.Find the highest purchase amount ordered by each customer on a particular date with their ID, order date and highest purchase amount.

```
mysql> SELECT custid, order_date, MAX(purch_amt) AS
highest_purchase_amount
   -> FROM order_table
   -> GROUP BY custid, order_date;
+--------+------------+------------------------+
| custid | order_date | highest_purchase_amount |
+--------+------------+------------------------+
|   3001 | 2016-09-10 |                 270.65 |
|   3002 | 2016-09-10 |                5760.00 |
|   3002 | 2016-10-05 |                  65.26 |
|   3003 | 2016-06-17 |                  75.29 |
|   3004 | 2016-10-10 |                1983.43 |
|   3005 | 2016-10-05 |                 150.50 |
|   3005 | 2016-11-30 |                 948.50 |
|   3007 | 2016-07-13 |                2400.60 |
|   3009 | 2016-08-17 |                 110.50 |
|   3009 | 2016-10-10 |                2480.40 |
+--------+------------+------------------------+
```

Q16.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_amount
   -> FROM order_table
   -> WHERE order_date = '2012-08-17'
   -> GROUP BY salesman_id;
Empty set (0.00 sec)
```

Q17.Write a SQL statement that counts all orders for a date August 17th, 2012.

```
mysql> SELECT custid, order_date, MAX(purch_amt) AS
highest_purchase_amount
   -> FROM order_table
   -> GROUP BY custid, order_date
   -> HAVING MAX(purch_amt) > 2000;
+--------+------------+------------------------+
| custid | order_date | highest_purchase_amount |
+--------+------------+------------------------+
```

```
|   3002 | 2016-09-10 |              5760.00 |
|   3007 | 2016-07-13 |              2400.60 |
|   3009 | 2016-10-10 |              2480.40 |
+--------+------------+-----------------------+
```

Q18.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 custid, order_date, MAX(purch_amt) AS
highest_purchase_amount
    -> FROM order_table
    -> GROUP BY custid, order_date
    -> HAVING MAX(purch_amt) > 2000;
+--------+------------+-----------------------+
| custid | order_date | highest_purchase_amount |
+--------+------------+-----------------------+
|   3002 | 2016-09-10 |              5760.00 |
|   3007 | 2016-07-13 |              2400.60 |
|   3009 | 2016-10-10 |              2480.40 |
+--------+------------+-----------------------+
```