1. From the following table, create a view for those salespersons belong to the city 'New York'.

Sample table: 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
        5007 | Paul Adam  | Rome     |       0.13
        5003 | Lauson Hen | San Jose |       0.12
```

Sample Output:

mysql> create view salesown1_view as select salesman_id,name,city from salesman;

Query OK, 0 rows affected (0.01 sec)

Mysql>select * from salesown1_view ;

```
salesman_id |    name    |   city   | commission
------------+------------+----------+------------
        5001 | James Hoog | New York |       0.15
```

2. From the following table, create a view for all salespersons. Return salesperson ID, name, and city.

Sample table: 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
        5007 | Paul Adam  | Rome     |       0.13
        5003 | Lauson Hen | San Jose |       0.12
```

Output

mysql> create view salesown1_view as select * from salesman;

Query OK, 0 rows affected (0.01 sec)

```
mysql> select * from salesown1_view;
+-------------+--------------+------------+
| salesman_id | name         | city       |
+-------------+--------------+------------+
|        5001 |  James Hoog  |  New York  |
|        5002 |  Nail Knite  |  Paris     |
|        5005 |  Pit Alex    |  London    |
|        5006 |  Mc Lyon     |  Paris     |
|        5007 |  Paul Adam   |  Rome      |
|        5003 |  Lauson Hen  |  San Jose  |
+-------------+--------------+------------+
```

3. From the following table, create a view to find the salespersons of the city 'New York'.

4. From the following table, create a view to count the number of customers in each grade.

Customer table

```
customer_id |   cust_name    |    city    | grade |
salesman_id
-------------+----------------+------------+-------+----------
---
        3002 | Nick Rimando   | New York   |   100 |
5001
        3007 | Brad Davis     | New York   |   200 |
5001
        3005 | Graham Zusi    | California |   200 |
5002
        3008 | Julian Green   | London     |   300 |
5002
        3004 | Fabian Johnson | Paris      |   300 |
5006
        3009 | Geoff Cameron  | Berlin     |   100 |
5003
```

```
        3003 | Jozy Altidor    | Moscow     |   200 |
5007

        3001 | Brad Guzan      | London     |       |
5005
```

mysql> create view count_cust as select grade,count(*) from customer group by grade;

Query OK, 0 rows affected (0.01 sec)

mysql> select * from count_cust;

+-------+----------+

| grade | count(*) |

+-------+----------+

| 100 | 2 |

| 200 | 3 |

| 300 | 2 |

| NULL | 1 |

+-------+----------+

4 rows in set (0.02 sec)

5. From the following table, create a view to count the number of unique customer, compute average and total purchase amount of customer orders by each date.

Sample table : orders

ord_no	purch_amt	ord_date	customer_id	salesman_id
70001	150.5	2012-10-05	3005	5002
70009	270.65	2012-09-10	3001	5005
70002	65.26	2012-10-05	3002	5001
70004	110.5	2012-08-17	3009	5003
70007	948.5	2012-09-10	3005	5002
70005	2400.6	2012-07-27	3007	5001

```
70008        5760         2012-09-10  3002          5001
70010        1983.43      2012-10-10  3004          5006
70003        2480.4       2012-10-10  3009          5003
70012        250.45       2012-06-27  3008          5002
70011        75.29        2012-08-17  3003          5007
70013        3045.6       2012-04-25  3002          5001
```

mysql> create view customercount2 as select ord_date, count(distinct customer_id),avg(purch_amt),sum(purch_amt) from orders group by ord_date;

Query OK, 0 rows affected (0.01 sec)

mysql> select * from customercount2;

```
+------------+--------------------------+-------------------+-------------------+
| ord_date   | count(distinct customer_id) | avg(purch_amt)    | sum(purch_amt)    |
+------------+--------------------------+-------------------+-------------------+
| 2012-04-25 |                        1 |  3045.60009765625 |  3045.60009765625 |
| 2012-06-27 |                        1 |  250.4499969482422 |  250.4499969482422 |
| 2012-07-27 |                        1 |  2400.60009765625 |  2400.60009765625 |
| 2012-08-17 |                        2 |  92.89500045776367 | 185.79000091552734 |
| 2012-09-10 |                        3 |  2326.383331298828 |  6979.149993896484 |
| 2012-10-05 |                        2 | 107.88000106811523 | 215.76000213623047 |
| 2012-10-10 |                        2 | 2231.9149780273438 | 4463.8299560546875 |
+------------+--------------------------+-------------------+-------------------+
```

7 rows in set (0.01 sec)

6. From the following tables, create a view to get the salesperson and customer by name. Return order name, purchase amount, salesperson ID, name, customer name.

Sample table: 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
        5007 | Paul Adam  | Rome      |       0.13
        5003 | Lauson Hen | San Jose  |       0.12
```

Sample table: customer

```
customer_id |   cust_name    |    city    | grade |
salesman_id
-------------+----------------+------------+-------+----------
---
        3002 | Nick Rimando   | New York   |   100 |
5001
        3007 | Brad Davis     | New York   |   200 |
5001
        3005 | Graham Zusi    | California |   200 |
5002
        3008 | Julian Green   | London     |   300 |
5002
        3004 | Fabian Johnson | Paris      |   300 |
5006
        3009 | Geoff Cameron  | Berlin     |   100 |
5003
        3003 | Jozy Altidor   | Moscow     |   200 |
5007
        3001 | Brad Guzan     | London     |       |
5005
```

Sample table: orders

```
ord_no      purch_amt   ord_date    customer_id  salesman_id
----------  ----------  ----------  -----------  -----------
70001       150.5       2012-10-05  3005         5002
70009       270.65      2012-09-10  3001         5005
70002       65.26       2012-10-05  3002         5001
70004       110.5       2012-08-17  3009         5003
70007       948.5       2012-09-10  3005         5002
70005       2400.6      2012-07-27  3007         5001
70008       5760        2012-09-10  3002         5001
70010       1983.43     2012-10-10  3004         5006
70003       2480.4      2012-10-10  3009         5003
70012       250.45      2012-06-27  3008         5002
70011       75.29       2012-08-17  3003         5007

mysql> create view nameorders as select ord_no, purch_amt,
a.salesman_id, name, cust_name FROM orders a, customer b,
salesman c WHERE a.customer_id = b.customer_id AND
a.salesman_id = c.salesman_id;
Query OK, 0 rows affected (0.01 sec)

mysql> select * from nameorders;
+--------+-----------+-------------+--------------+-----------
-----+
| ord_no | purch_amt | salesman_id | name         | cust_name
|
```

```
+--------+----------+------------+-------------+----------
-----+
|  70013 |    3045.6 |        5001 |  James Hoog  | Nick
Rimando   |
|  70008 |      5760 |        5001 |  James Hoog  | Nick
Rimando   |
|  70002 |     65.26 |        5001 |  James Hoog  | Nick
Rimando   |
|  70005 |    2400.6 |        5001 |  James Hoog  | Brad Davis
|
|  70007 |     948.5 |        5002 |  Nail Knite  | Graham
Zusi    |
|  70001 |     150.5 |        5002 |  Nail Knite  | Graham
Zusi    |
|  70012 |    250.45 |        5002 |  Nail Knite  | Julian
Green   |
|  70010 |   1983.43 |        5006 |  Mc Lyon     | Fabian
Johnson |
|  70003 |    2480.4 |        5003 |  Lauson Hen  | Geoff
Cameron  |
|  70004 |     110.5 |        5003 |  Lauson Hen  | Geoff
Cameron  |
|  70011 |     75.29 |        5007 |  Paul Adam   | Jozy
Altidor   |
|  70009 |    270.65 |        5005 |  Pit Alex    | Brad Guzan
|
+--------+----------+------------+-------------+----------
-----+
12 rows in set (0.00 sec
```

7. From the following table, create a view to find all the customers who have the highest grade. Return all the fields of customer.

Refer customer table

mysql> create view highestgrade as select * from customer where grade=(select max(grade) from customer);

Query OK, 0 rows affected (0.02 sec)

mysql> select * from highestgrade;

```
+-------------+---------------+--------+-------+-------------+
| customer_id | cust_name     | city   | grade | salesman_id |
+-------------+---------------+--------+-------+-------------+
|        3008 | Julian Green  | London |   300 |        5002 |
```

```
|       3004 | Fabian Johnson | Paris  |   300 |       5006 |
+------------+---------------+--------+-------+------------+
```

2 rows in set (0.01 sec)

8. From the following table, create a view to count number of the salesperson in each city. Return city, number of salespersons.
Refer salesman table

```
mysql> create view citycount as select city,count(*) from salesman group by city;
Query OK, 0 rows affected (0.01 sec)

mysql> select * from citycount;
+------------+----------+
| city            | count(*) |
+------------+----------+
|  New York  |       1 |
|  Paris         |       2 |
|  London     |       1 |
|  Rome       |       1 |
|  San Jose  |       1 |
+------------+----------+
5 rows in set (0.00 sec)
```

9. From the following table, create a view to compute average purchase amount and total purchase amount for each salesperson. Return name, average purchase and total purchase amount. (Assume all names are unique).
Refer salesman and orders table

```
mysql> create view uniq_name as select  name , avg(purch_amt),sum(purch_amt) from salesman s,orders o where s.salesman_id=o.salesman_id group by name;

Query OK, 0 rows affected (0.01 sec)

mysql> select * from uniq_name;
+--------------+-------------------+-------------------+
| name         | avg(purch_amt)     | sum(purch_amt)     |
+--------------+-------------------+-------------------+
|  Nail Knite  | 449.81666564941406 | 1349.4499969482422 |
|  Pit Alex    |  270.6499938964844 |  270.6499938964844 |
|  James Hoog  | 2817.8650493621826 |  11271.46019744873 |
|  Lauson Hen  |  1295.449951171875 |   2590.89990234375 |
```

| Mc Lyon | 1983.4300537109375 | 1983.4300537109375 |
| Paul Adam | 75.29000091552734 | 75.29000091552734 |
+-------------+-------------------+-------------------+
6 rows in set (0.01 sec)

10. From the following tables, create a view to find those salespeople who handle more than one customer. Return all the fields of salesperson.

Refer customer and salesman table

```
mysql> create view salespeople as select * from salesman s where
1<(select count(*) from customer c where
s.salesman_id=c.salesman_id);

Query OK, 0 rows affected (0.01 sec)



mysql> select * from salespeople;

+-------------+-------------+-----------+------------+

| salesman_id | name        | city      | commission |

+-------------+-------------+-----------+------------+

|        5001 |  James Hoog  |  New York  |       0.15 |

|        5002 |  Nail Knite  |  Paris     |       0.13 |

+-------------+-------------+-----------+------------+

2 rows in set (0.00 sec)
```