SALESMAN TABLE

mysql> select* from 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 Aam	Rome	0.13
5003	Lauson Hen	San jose	0.12

6 rows in set (0.00 sec)

#CUSTOMER TABLE

mysql> select* from customers;

customer_id	cust_name	city	grade	salesman_id
3002	Nick Rimando	New york	100	5001
3005	graham davis	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	NULL	5005
3007	Brad Davis	New york	200	5002

8 rows in set (0.00 sec)

ORDERS TABLE

mysql> select*from 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 |
+--------+---------+------------+-----------+-----------+
12 rows in set (0.02 sec)
```

1.write a SQL query to find the salesperson and customer who belongs to same city. Return Salesman, cust_name and city

```
mysql> select salesman.name,customers.cust_name,customers.city from salesman inner join
customers on  salesman.salesman_id=customers.salesman_id where
salesman.city=customers.city;
+------------+---------------+----------+
| name       | cust_name     | city     |
+------------+---------------+----------+
| James hoog | Nick Rimando  | New york |
| Mc lyon    | fabian johnson | paris   |
| Pit Alex   | Brad Guzan    | London   |
+------------+---------------+----------+
3 rows in set (0.00 sec)
```

2.write a SQL query to find the salesperson(s) and the customer(s) he handle. Return Customer Name, city, Salesman, commission

```
mysql> select salesman.name as
salesman_name,customers.cust_name,customers.city,salesman.commission from salesman  join
customers on  salesman.salesman_id=customers.salesman_id where
salesman.salesman_id=customers.salesman_id;
+---------------+---------------+------------+------------+
| salesman_name | cust_name     | city       | commission |
+---------------+---------------+------------+------------+
| James hoog    | Nick Rimando  | New york   |       0.15 |
| Nail Knite    | graham davis  | california |       0.13 |
| Nail Knite    | julian green  | london     |       0.13 |
| Mc lyon       | fabian johnson | paris     |       0.14 |
| Lauson Hen    | Geoff Cameron | Berlin     |       0.12 |
| Paul Aam      | Jozy altidor  | moscow     |       0.13 |
| Pit Alex      | Brad Guzan    | London     |       0.11 |
| Nail Knite    | Brad Davis    | New york   |       0.13 |
+---------------+---------------+------------+------------+
8 rows in set (0.00 sec)
```

3.From the following tables write a SQL query to find those salespersons who received a commission from the company more than 12%. Return Customer Name, customer city, Salesman, commission.

mysql> select salesman.name as salesman_name,customers.cust_name,customers.city,salesman.commission from salesman join customers on salesman.salesman_id=customers.salesman_id where salesman.commission >0.12;

```
+---------------+----------------+------------+------------+
| salesman_name | cust_name      | city       | commission |
+---------------+----------------+------------+------------+
| James hoog    | Nick Rimando   | New york   |       0.15 |
| Nail Knite    | graham davis   | california |       0.13 |
| Nail Knite    | julian green   | london     |       0.13 |
| Mc lyon       | fabian johnson | paris      |       0.14 |
| Paul Aam      | Jozy altidor   | moscow     |       0.13 |
| Nail Knite    | Brad Davis     | New york   |       0.13 |
+---------------+----------------+------------+------------+
6 rows in set (0.01 sec)
```

4. write a SQL query to find those salespersons do not live in the same city where their customers live and received a commission from the company more than 12%.
Return Customer Name, customer city, Salesman, salesman city, commission

mysql> select customers.cust_name as customer_name,customers.city as customer_city,salesman.name as salesman_name,salesman.city as salesman_city ,salesman.commission from salesman join customers on salesman.salesman_id=customers.salesman_id where not(customers.city=salesman.city) and salesman.commission > 0.12;

```
+---------------+---------------+---------------+---------------+------------+
| customer_name | customer_city | salesman_name | salesman_city | commission |
+---------------+---------------+---------------+---------------+------------+
| graham davis  | california    | Nail Knite    | Paris         |       0.13 |
| julian green  | london        | Nail Knite    | Paris         |       0.13 |
| Jozy altidor  | moscow        | Paul Aam      | Rome          |       0.13 |
| Brad Davis    | New york      | Nail Knite    | Paris         |       0.13 |
+---------------+---------------+---------------+---------------+------------+
4 rows in set (0.00 sec)
```

5. write a SQL query to find those customers whose grade less than 300. Return cust_name, customer city, grade, Salesman, saleman city. The result should be ordered by ascending customer_id

mysql> select customers.cust_name as customer_name,customers.city as

customer_city,customers.grade,salesman.name as salesman_name,salesman.city as salesman_city from salesman join customers on salesman.salesman_id=customers.salesman_id order by customers.customer_id ASC;

```
+----------------+--------------+-------+--------------+--------------+
| customer_name  | customer_city | grade | salesman_name | salesman_city |
+----------------+--------------+-------+--------------+--------------+
| Brad Guzan     | London       | NULL  | Pit Alex     | London       |
| Nick Rimando   | New york     |  100  | James hoog   | New York     |
| Jozy altidor   | moscow       |  200  | Paul Aam     | Rome         |
| fabian johnson | paris        |  300  | Mc lyon      | paris        |
| graham davis   | california   |  200  | Nail Knite   | Paris        |
| Brad Davis     | New york     |  200  | Nail Knite   | Paris        |
| julian green   | london       |  300  | Nail Knite   | Paris        |
| Geoff Cameron  | Berlin       |  100  | Lauson Hen   | San jose     |
+----------------+--------------+-------+--------------+--------------+
```
8 rows in set (0.00 sec)

6.write a SQL query to find those customers whose grade less than 300. Return cust_name, customer city, grade, Salesman, saleman city.
The result should be ordered by ascending customer_id

mysql> select customers.cust_name as customer_name,customers.city as customer_city,customers.grade,salesman.name as salesman_name,salesman.city as salesman_city from salesman join customers on salesman.salesman_id=customers.salesman_id where customers.grade < 300 order by customers.customer_id ASC;

```
+---------------+--------------+-------+--------------+--------------+
| customer_name | customer_city | grade | salesman_name | salesman_city |
+---------------+--------------+-------+--------------+--------------+
| Nick Rimando  | New york     |  100  | James hoog   | New York     |
| Jozy altidor  | moscow       |  200  | Paul Aam     | Rome         |
| graham davis  | california   |  200  | Nail Knite   | Paris        |
| Brad Davis    | New york     |  200  | Nail Knite   | Paris        |
| Geoff Cameron | Berlin       |  100  | Lauson Hen   | San jose     |
+---------------+--------------+-------+--------------+--------------+
```
5 rows in set (0.00 sec)

7.write a SQL query to find those orders where order amount exists between 500 and 2000. Return ord_no, purch_amt, cust_name, city.

mysql> select orders.ord_date,orders.purch_amt,customers.cust_name,customers.city from customers join orders on orders.customer_id=customers.customer_id where purch_amt between 500 and 2000;

```
+------------+-----------+---------------+------------+
| ord_date   | purch_amt | cust_name     | city       |
```

```
+------------+----------+---------------+------------+
| 2012-09-10 |    948.5 | graham davis  | california |
| 2012-10-10 |  1983.43 | fabian johnson | paris     |
+------------+----------+---------------+------------+
2 rows in set (0.00 sec)
```

8.write a SQL query to find the details of an order. Return ord_no, ord_date, purch_amt, Customer Name, grade, Salesman, commission

mysql> select orders.ord_no ,orders.ord_date ,orders.purch_amt,customers.cust_name,customers.grade,salesman.name as salesman_name,salesman.commission from orders join customers on orders.customer_id=customers .customer_id join salesman on orders.salesman_id=salesman.salesman_id;

ord_no	ord_date	purch_amt	cust_name	grade	salesman_name	commission
70013	2012-04-25	3045.6	Nick Rimando	100	James hoog	0.15
70008	2012-09-10	5760	Nick Rimando	100	James hoog	0.15
70002	2012-10-05	65.26	Nick Rimando	100	James hoog	0.15
70007	2012-09-10	948.5	graham davis	200	Nail Knite	0.13
70001	2012-10-05	150.5	graham davis	200	Nail Knite	0.13
70012	2012-06-27	250.45	julian green	300	Nail Knite	0.13
70010	2012-10-10	1983.43	fabian johnson	300	Mc lyon	0.14
70003	2012-10-10	2480.4	Geoff Cameron	100	Lauson Hen	0.12
70004	2012-08-17	110.5	Geoff Cameron	100	Lauson Hen	0.12
70011	2012-08-17	75.29	Jozy altidor	200	Paul Aam	0.13
70009	2012-09-10	270.65	Brad Guzan	NULL	Pit Alex	0.11
70005	2012-07-27	2400.6	Brad Davis	200	James hoog	0.15

12 rows in set (0.00 sec)

9.Write a SQL statement to make a join on the tables salesman, customer and orders in such a form that the same column of each table will appear once and only the relational rows will come

mysql> select salesman.name as salesman_name ,salesman.city,salesman.commission,customers.cust_name,customers.city,customers.grade,orders.ord_no,orders.ord_date,orders.purch_amt from ORDERS join customers on ORDERS.CUSTOMER_ID=customers.customer_id join salesman on orders.salesman_id = salesman.salesman_id;

| salesman_name | city | commission | cust_name | city | grade | ord_no | ord_date |

purch_amt |

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

James hoog	New York	0.15	Nick Rimando	New york	100	70013	2012-04-25	3045.6
James hoog	New York	0.15	Nick Rimando	New york	100	70008	2012-09-10	5760
James hoog	New York	0.15	Nick Rimando	New york	100	70002	2012-10-05	65.26
Nail Knite	Paris	0.13	graham davis	california	200	70007	2012-09-10	948.5
Nail Knite	Paris	0.13	graham davis	california	200	70001	2012-10-05	150.5
Nail Knite	Paris	0.13	julian green	london	300	70012	2012-06-27	250.45
Mc lyon	paris	0.14	fabian johnson	paris	300	70010	2012-10-10	1983.43
Lauson Hen	San jose	0.12	Geoff Cameron	Berlin	100	70003	2012-10-10	2480.4
Lauson Hen	San jose	0.12	Geoff Cameron	Berlin	100	70004	2012-08-17	110.5
Paul Aam	Rome	0.13	Jozy altidor	moscow	200	70011	2012-08-17	75.29
Pit Alex	London	0.11	Brad Guzan	London	NULL	70009	2012-09-10	270.65
James hoog	New York	0.15	Brad Davis	New york	200	70005	2012-07-27	2400.6

+---------------+-----------+-----------+---------------+-----------+-------+--------+-----------+----------+
12 rows in set (0.00 sec)

 10.Write a SQL statement to make a report with customer name, city, order number, order date, and order amount in ascending order according to the order date to find that either any of the existing customers have placed no order or placed one or more orders.

mysql> select customers.cust_name,customers.city,orders.ord_no,orders.ord_date,orders.purch_amt FROM ORDERS LEFT JOIN CUSTOMERS ON ORDERS.CUSTOMER_ID= CUSTOMERS.CUSTOMER_ID order by ord_date ASC;

+----------------+-----------+--------+------------+-----------+
| cust_name | city | ord_no | ord_date | purch_amt |
+----------------+-----------+--------+------------+-----------+
Nick Rimando	New york	70013	2012-04-25	3045.6
julian green	london	70012	2012-06-27	250.45
Brad Davis	New york	70005	2012-07-27	2400.6
Geoff Cameron	Berlin	70004	2012-08-17	110.5
Jozy altidor	moscow	70011	2012-08-17	75.29

```
| Brad Guzan     | London     | 70009 | 2012-09-10 |    270.65 |
| graham davis   | california | 70007 | 2012-09-10 |     948.5 |
| Nick Rimando   | New york   | 70008 | 2012-09-10 |      5760 |
| graham davis   | california | 70001 | 2012-10-05 |     150.5 |
| Nick Rimando   | New york   | 70002 | 2012-10-05 |     65.26 |
| fabian johnson | paris      | 70010 | 2012-10-10 |   1983.43 |
| Geoff Cameron  | Berlin     | 70003 | 2012-10-10 |    2480.4 |
+----------------+------------+-------+------------+-----------+
12 rows in set (0.00 sec)
```