

1. From the following tables write a SQL query to find the salesperson and customer who belongs to same city. Return Salesman, cust_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

Sample table: customer

customer_id	cust_name	city	grade
salesman_id			
-----+-----+-----+-----+-----			

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

Query :

```
select salesman.name,customer.cust_name,customer.city from salesman inner join
customer on salesman.city=customer.city;
```

Output :

```
mysql> select salesman.name,customer.cust_name,customer.city from salesman inner join customer on
-> salesman.city=customer.city;
```

name	cust_name	city
James Hoog	Nick Rimando	New York
James Hoog	Brad Davis	New York
Pit Alex	Julian Green	London
Mc Lyon	Fabian Johnson	Paris
Nail Knite	Fabian Johnson	Paris
Pit Alex	Brad Guzan	London

```
6 rows in set (0.00 sec)
```

2. From the following tables write a SQL query to find those orders where order amount exists between 500 and 2000. Return ord_no, purch_amt, cust_name, city.

Orders table

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

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

Query :

```
select o.ord_no,o.purch_amt,c.cust_name,c.city from orders o,customer c where
o.customer_id=c.customer_id and o.purch_amt between 500 and 2000;
```

Output :

```
mysql> select o.ord_no,o.purch_amt,c.cust_name,c.city from orders o,customer c where
-> o.customer_id=c.customer_id and o.purch_amt between 500 and 2000;
+-----+-----+-----+-----+
| ord_no | purch_amt | cust_name | city |
+-----+-----+-----+-----+
| 70007 | 948.5 | Graham Zusi | California |
| 70010 | 1983.43 | Fabian Johnson | Paris |
+-----+-----+-----+-----+
2 rows in set (0.02 sec)
```

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

Refer customer and salesman table above

Query :

```
select c.cust_name,c.city,s.name,s.commission from customer c inner join salesman s
on c.salesman_id=s.salesman_id;
```

Output :

```
mysql> select c.cust_name,c.city,s.name,s.commission from customer c inner join salesman
-> s on c.salesman_id=s.salesman_id;
+-----+-----+-----+-----+
| cust_name | city | name | commission |
+-----+-----+-----+-----+
| Nick Rimando | New York | James Hoog | 0.15 |
| Brad Davis | New York | James Hoog | 0.15 |
| Graham Zusi | California | Nail Knite | 0.13 |
| Julian Green | London | Nail Knite | 0.13 |
| Fabian Johnson | Paris | Mc Lyon | 0.14 |
| Geoff Cameron | Berlin | Lauson Hen | 0.12 |
| Jozy Altidor | Moscow | Paul Adam | 0.13 |
| Brad Guzan | London | Pit Alex | 0.11 |
+-----+-----+-----+-----+
8 rows in set (0.00 sec)
```

4. 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.

Refer customer and salesman table above

Query :

```
select c.cust_name,c.city,s.salesman_id,s.commission from customer c inner join  
salesman s on c.salesman_id=s.salesman_id where commission>0.12;
```

Output :

```
mysql> select c.cust_name,c.city,s.salesman_id,s.commission from customer c inner join  
-> salesman s on c.salesman_id=s.salesman_id where commission>0.12;
```

cust_name	city	salesman_id	commission
Nick Rimando	New York	5001	0.15
Brad Davis	New York	5001	0.15
Graham Zusi	California	5002	0.13
Julian Green	London	5002	0.13
Fabian Johnson	Paris	5006	0.14
Jozy Altidor	Moscow	5007	0.13

```
6 rows in set (0.02 sec)
```

5. From the following tables 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.

Refer customer and salesman table above

Query :

```
select c.cust_name,c.city,s.salesman_id,s.commission from customer c inner join  
salesman s on c.salesman_id=s.salesman_id where c.city<>s.city and  
s.commission>0.12;
```

Output :

```
mysql> select c.cust_name,c.city,s.salesman_id,s.commission from customer c inner join  
-> salesman s on c.salesman_id=s.salesman_id where c.city<>s.city and s.commission>0.12;
```

cust_name	city	salesman_id	commission
Graham Zusi	California	5002	0.13
Julian Green	London	5002	0.13
Jozy Altidor	Moscow	5007	0.13

```
3 rows in set (0.02 sec)
```

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

Refer order customer and salesman tables

Query :

```
select o.ord_no,o.ord_date,o.purch_amt,c.cust_name as
'customername',c.grade,s.salesman_id,s.commission from orders o inner join
customer c on o.customer_id=c.customer_id inner join salesman s on
c.salesman_id=s.salesman_id;
```

Output :

```
mysql> select o.ord_no,o.ord_date,o.purch_amt,c.cust_name as 'customername',c.grade,s.salesman_id,s.commission
-> from orders o inner join customer c on o.customer_id=c.customer_id inner join salesman s on c.salesman_id=s.salesman_id;
```

ord_no	ord_date	purch_amt	customername	grade	salesman_id	commission
70001	2012-10-05	150.5	Graham Zusi	200	5002	0.13
70009	2012-09-10	270.65	Brad Guzan	NULL	5005	0.11
70002	2012-10-05	65.26	Nick Rimando	100	5001	0.15
70004	2012-08-17	110.5	Geoff Cameron	100	5003	0.12
70007	2012-09-10	948.5	Graham Zusi	200	5002	0.13
70005	2012-07-27	2400.6	Brad Davis	200	5001	0.15
70008	2012-09-10	5760	Nick Rimando	100	5001	0.15
70010	2012-10-10	1983.43	Fabian Johnson	300	5006	0.14
70003	2012-10-10	2480.4	Geoff Cameron	100	5003	0.12
70012	2012-06-27	250.45	Julian Green	300	5002	0.13
70011	2012-08-17	75.29	Jozy Altidor	200	5007	0.13
70013	2012-04-25	3045.6	Nick Rimando	100	5001	0.15

12 rows in set (0.00 sec)

7. 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

order customer and salesman tables

Query :

```
select * from orders natural join salesman natural join customer;
```

Output :

```
mysql> select * from orders natural join salesman natural join customer;
ERROR 1146 (42S02): Table 'assignments.orders' doesn't exist
mysql> select * from odrders natural join salesman natural join customer;
ERROR 1146 (42S02): Table 'assignments.odrders' doesn't exist
mysql> select * from orders natural join salesman natural join customer;
```

salesman_id	customer_id	city	ord_no	purch_amt	ord_date	name	commission	cust_name	grade
5005	3001	London	70009	270.65	2012-09-10	Pit Alex	0.11	Brad Guzan	NULL
5001	3002	New York	70002	65.26	2012-10-05	James Hoog	0.15	Nick Rimando	100
5001	3007	New York	70005	2400.6	2012-07-27	James Hoog	0.15	Brad Davis	200
5001	3002	New York	70008	5760	2012-09-10	James Hoog	0.15	Nick Rimando	100
5006	3004	Paris	70010	1983.43	2012-10-10	Mc Lyon	0.14	Fabian Johnson	300
5001	3002	New York	70013	3045.6	2012-04-25	James Hoog	0.15	Nick Rimando	100

6 rows in set (0.03 sec)

8. From the following tables write a SQL query to display the cust_name, customer city, grade, Salesman, salesman city. The result should be ordered by ascending on customer_id.

Refer customer and salesman table

Query :

```
select c.cust_name,c.city,c.grade,s.salesman_id,s.city from customer c inner join
salesman s on c.salesman_id=s.salesman_id order by c.customer_id;
```

Output :

```
mysql> select c.cust_name,c.city,c.grade,s.salesman_id,s.city from customer c inner join
-> salesman s on c.salesman_id=s.salesman_id order by c.customer_id;
```

cust_name	city	grade	salesman_id	city
Brad Guzan	London	NULL	5005	London
Nick Rimando	New York	100	5001	New York
Jozy Altidor	Moscow	200	5007	Rome
Fabian Johnson	Paris	300	5006	Paris
Graham Zusi	California	200	5002	Paris
Brad Davis	New York	200	5001	New York
Julian Green	London	300	5002	Paris
Geoff Cameron	Berlin	100	5003	San Jose

8 rows in set (0.00 sec)

9. From the following tables write a SQL query to find those customers whose grade less than 300. Return cust_name, customer city, grade, Salesman, salesman city. The result should be ordered by ascending customer_id

Refer customer and salesman table

Query :

```
select c.cust_name,c.city,c.grade,s.salesman_id,s.city from customer c inner join
salesman s on c.salesman_id=s.salesman_id where grade<300 order by customer_id;
```

Output :

```
mysql> select c.cust_name,c.city,c.grade,s.salesman_id,s.city from customer c inner join
-> salesman s on c.salesman_id=s.salesman_id where grade<300 order by customer_id;
```

cust_name	city	grade	salesman_id	city
Nick Rimando	New York	100	5001	New York
Jozy Altidor	Moscow	200	5007	Rome
Graham Zusi	California	200	5002	Paris
Brad Davis	New York	200	5001	New York
Geoff Cameron	Berlin	100	5003	San Jose

5 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.

Refer order and customer

Query :

```
select c.cust_name,c.city,o.ord_no,o.ord_date,o.purch_amt from customer c left  
outer join orders o on c.customer_id=o.customer_id order by o.ord_date;
```

Output :

```
mysql> select c.cust_name,c.city,o.ord_no,o.ord_date,o.purch_amt from customer c left  
-> outer join orders o on c.customer_id=o.customer_id order by o.ord_date;
```

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
Nick Rimando	New York	70008	2012-09-10	5760
Graham Zusi	California	70007	2012-09-10	948.5
Brad Guzan	London	70009	2012-09-10	270.65
Nick Rimando	New York	70002	2012-10-05	65.26
Graham Zusi	California	70001	2012-10-05	150.5
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)