

## Assignment-21

### Joins on sql

*Kaviya C*

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:**

```
mysql> select salesman.name AS "Salesman",
-> customer.cust_name, customer.city
-> FROM salesman, customer
-> WHERE salesman.city=customer.city;
```

| Salesman   | cust_name      | city     |
|------------|----------------|----------|
| James Hoog | Nick Rimando   | New York |
| James Hoog | Brad Davis     | New York |
| Mc Lyon    | Fabian Johnson | Paris    |
| Nail Knite | Fabian Johnson | Paris    |
| Pit Alex   | Brad Guzan     | London   |

5 rows in set (0.04 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 |
|-------------|--------------|------------|-------|
| 3002        | Nick Rimando | New York   | 100   |
| 3007        | Brad Davis   | New York   | 200   |
| 3005        | Graham Zusi  | California | 200   |
| 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:

```
mysql> SELECT o.ord_no,o.purch_amt,
-> cust.cust_name,cust.city
-> FROM orders o,customer cust
-> where o.customer_id=cust.customer_id
-> AND o.purch_amt BETWEEN 500 AND 2000;
```

| ord_no | purch_amt | cust_name      | city       |
|--------|-----------|----------------|------------|
| 7007   | 948.5     | Graham Zusi    | California |
| 7010   | 1983.43   | Fabian Johnson | Paris      |

2 rows in set (0.00 sec)

3.

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:

```
mysql> SELECT cust.cust_name AS "Customer Name",
-> cust.city, sales.name AS "Salesman",
-> sales.commission FROM customer cust
-> INNER JOIN salesman sales
-> ON cust.salesman_id=sales.salesman_id;
```

| Customer Name  | city       | Salesman   | 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:

```
mysql> SELECT cust.cust_name AS "Customer Name",
-> cust.city,sales.name AS "Salesman",
-> sales.commission FROM customer cust
-> INNER JOIN salesman sales
-> ON cust.salesman_id=sales.salesman_id
-> WHERE sales.commission>.12;
```

| Customer Name  | city       | Salesman   | 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       |
| Jozy Altidor   | Moscow     | Paul Adam  | 0.13       |

6 rows in set (0.00 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:

```
mysql> SELECT cust.cust_name AS "Customer Name",
-> cust.city, sales.name AS "Salesman",sales.city,sales.commission
-> FROM customer cust INNER JOIN salesman sales
-> ON cust.salesman_id=sales.salesman_id WHERE sales.commission>.12
-> AND cust.city<>sales.city;
```

| Customer Name | city       | Salesman   | city  | commission |
|---------------|------------|------------|-------|------------|
| Graham Zusi   | California | Nail Knite | Paris | 0.13       |
| Julian Green  | London     | Nail Knite | Paris | 0.13       |
| Jozy Altidor  | Moscow     | Paul Adam  | Rome  | 0.13       |

3 rows in set (0.00 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:

```
mysql> SELECT orders.ord_no,orders.ord_date,orders.purch_amt,cust.cust_name AS "Customer Name",
-> cust.grade, sales.name AS "Salesman",sales.commission FROM orders orders
-> INNER JOIN customer cust ON orders.customer_id=cust.customer_id
-> INNER JOIN salesman sales ON orders.salesman_id=sales.salesman_id;
```

| ord_no | ord_date   | purch_amt | Customer Name  | grade | Salesman   | commission |
|--------|------------|-----------|----------------|-------|------------|------------|
| 7013   | 2012-04-25 | 3045.6    | Nick Rimando   | 100   | James Hoog | 0.15       |
| 7008   | 2012-09-10 | 5760      | Nick Rimando   | 100   | James Hoog | 0.15       |
| 7002   | 2012-10-05 | 65.26     | Nick Rimando   | 100   | James Hoog | 0.15       |
| 7007   | 2012-09-10 | 948.5     | Graham Zusi    | 200   | Nail Knite | 0.13       |
| 7001   | 2012-10-05 | 150.5     | Graham Zusi    | 200   | Nail Knite | 0.13       |
| 7012   | 2012-06-27 | 250.45    | Julian Green   | 300   | Nail Knite | 0.13       |
| 7010   | 2012-10-10 | 1983.43   | Fabian Johnson | 300   | Mc Lyon    | 0.14       |
| 7003   | 2012-10-10 | 2480.4    | Geoff Cameron  | 100   | Lauson Hen | 0.12       |
| 7004   | 2012-08-17 | 110.5     | Geoff Cameron  | 100   | Lauson Hen | 0.12       |
| 7011   | 2012-08-17 | 75.29     | Jozy Altidor   | 200   | Paul Adam  | 0.13       |
| 7009   | 2012-09-10 | 270.65    | Brad Guzan     | 0     | Pit Alex   | 0.11       |

11 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:

```
mysql> SELECT * FROM orders NATURAL JOIN customer NATURAL JOIN salesman;
```

| salesman_id | city     | customer_id | ord_no | purch_amt | ord_date   | cust_name      | grade | name       | commission |
|-------------|----------|-------------|--------|-----------|------------|----------------|-------|------------|------------|
| 5005        | London   | 3001        | 7009   | 270.65    | 2012-09-10 | Brad Guzan     | 0     | Pit Alex   | 0.11       |
| 5001        | New York | 3002        | 7002   | 65.26     | 2012-10-05 | Nick Rimando   | 100   | James Hoog | 0.15       |
| 5001        | New York | 3002        | 7008   | 5760      | 2012-09-10 | Nick Rimando   | 100   | James Hoog | 0.15       |
| 5006        | Paris    | 3004        | 7010   | 1983.43   | 2012-10-10 | Fabian Johnson | 300   | Mc Lyon    | 0.14       |
| 5001        | New York | 3002        | 7013   | 3045.6    | 2012-04-25 | Nick Rimando   | 100   | James Hoog | 0.15       |

5 rows in set (0.00 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:

```
mysql> SELECT cust.cust_name,cust.city,cust.grade,sales.name AS "Salesman",sales.city FROM customer cust
-> LEFT JOIN salesman sales ON cust.salesman_id=sales.salesman_id
-> order by cust.customer_id;
```

| cust_name      | city       | grade | Salesman   | city     |
|----------------|------------|-------|------------|----------|
| Brad Guzan     | London     | 0     | Pit Alex   | London   |
| Nick Rimando   | New York   | 100   | James Hoog | New York |
| Jozy Altidor   | Moscow     | 200   | Paul Adam  | Rome     |
| Fabian Johnson | Paris      | 300   | Mc Lyon    | Paris    |
| Graham Zusi    | California | 200   | Nail Knite | Paris    |
| Brad Davis     | New York   | 200   | James Hoog | New York |
| Julian Green   | London     | 300   | Nail Knite | Paris    |
| Geoff Cameron  | Berlin     | 100   | Lauson Hen | San Jose |

6 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:

```
mysql> SELECT cust.cust_name,cust.city ,cust.grade,
-> sales.name AS "Salesman",sales.city
-> FROM customer cust LEFT OUTER JOIN salesman sales
-> ON cust.salesman_id=sales.salesman_id WHERE cust.grade<300
-> ORDER BY cust.customer_id;
```

| cust_name     | city       | grade | Salesman   | city     |
|---------------|------------|-------|------------|----------|
| Brad Guzan    | London     | 0     | Pit Alex   | London   |
| Nick Rimando  | New York   | 100   | James Hoog | New York |
| Jozy Altidor  | Moscow     | 200   | Paul Adam  | Rome     |
| Graham Zusi   | California | 200   | Nail Knite | Paris    |
| Brad Davis    | New York   | 200   | James Hoog | New York |
| Geoff Cameron | Berlin     | 100   | Lauson Hen | San Jose |

6 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:

```
mysql> SELECT cust.cust_name,cust.city,orders.ord_no,
-> orders.ord_date,orders.purch_amt AS "Order Amount"
-> FROM customer cust LEFT OUTER JOIN orders orders
-> ON cust.customer_id=orders.customer_id ORDER BY orders.ord_date;
```

| cust_name      | city       | ord_no | ord_date   | Order Amount |
|----------------|------------|--------|------------|--------------|
| Brad Davis     | New York   | NULL   | NULL       | NULL         |
| Nick Rimando   | New York   | 7013   | 2012-04-25 | 3045.6       |
| Julian Green   | London     | 7012   | 2012-06-27 | 250.45       |
| Geoff Cameron  | Berlin     | 7004   | 2012-08-17 | 110.5        |
| Jozy Altidor   | Moscow     | 7011   | 2012-08-17 | 75.29        |
| Nick Rimando   | New York   | 7008   | 2012-09-10 | 5760         |
| Graham Zusi    | California | 7007   | 2012-09-10 | 948.5        |
| Brad Guzan     | London     | 7009   | 2012-09-10 | 270.65       |
| Nick Rimando   | New York   | 7002   | 2012-10-05 | 65.26        |
| Graham Zusi    | California | 7001   | 2012-10-05 | 150.5        |
| Fabian Johnson | Paris      | 7010   | 2012-10-10 | 1983.43      |
| Geoff Cameron  | Berlin     | 7003   | 2012-10-10 | 2480.4       |

12 rows in set (0.00 sec)