Lab Manual for Introduction to Database Systems

Lab-11

SQL INNER, RIGHT, LEFT AND FULL JOIN

Table of Contents

Lab7: SQL INNER,RIGHT, LEFT AND FULL JOIN

1. Introduction

The purpose of this lab is to familiarize you to SQL Joins. The SQL Joins clause is used to combine records from two or more tables in a database. A JOIN is a means for combining fields from two tables by using values common to each. In this lab discuss the SQL joins.

Relevant Lecture Material

a) Revise Lecture No. 11 and 12
b) Text Book: Java: Text Book: Database Systems, A practical approach to design, implementation and management by Thomas Connolly, Carolyn Begg, Addison Wesley , Fifth Edition,
   1. Read URL:
      i. http://www.dofactory.com/sql/join
   2. Revise the concept of Joins INNER, OUTER, LEFT, RIGHT, SELF

2. Activity Time boxing

Table 1: Activity Time Boxing

Task No.	Activity Name	Activity time	Total Time
6.2	Setting-up and Setting Up XAMPP (MySQL, Apache)	20mins	20mins
6.3	Walkthrough Tasks	30mins	60mins
7	Practice tasks	20 to 30mins for each task	50mins
8	Evaluation Task	40mins for all assigned task	40mins

3. Objective of the experiment

- To understand the use RIGHT and LEFT Joins.
- To be able to optimize queries.
- To understand the concept of joins.

4. Concept Map

4.1 INNER JOIN

An INNER JOIN of A and B gives the result of A intersect B. It returns all the common records between two tables. If there's no related record, it will contain NULL.

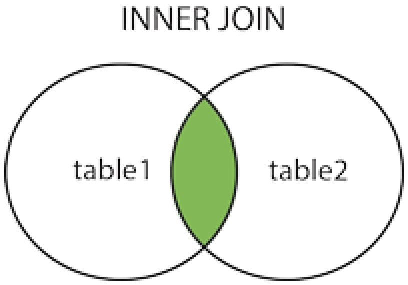


Figure 1 visually illustrated inner join

Syntax:

```
SELECT * FROM a

INNER JOIN b on a.id = b.id;
```

4.2 LEFT JOIN

A LEFT JOIN gives all rows in A, plus any common rows in B. If a record in A doesn't exist in B, it will return NULL for that row.

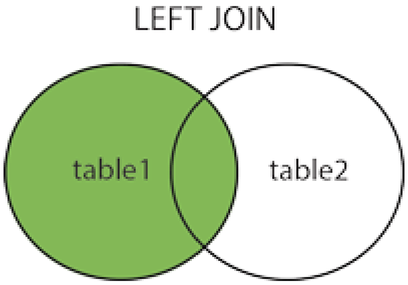


Figure visually illustrated left join

Syntax:

```
SELECT * FROM a

LEFT JOIN b on a.id = b.id;
```

4.3 RIGHT JOIN

A RIGHT JOIN gives all rows in table B, plus any common rows in A. If a record in B doesn't exist in A, it will return NULL for that row.

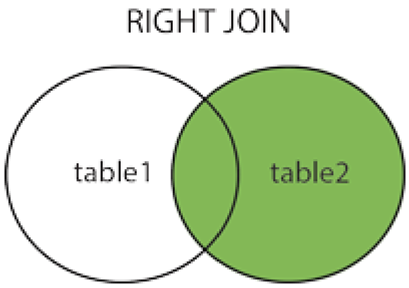


Figure visually illustrated Right join

Syntax:

```
SELECT * FROM a

LEFT JOIN b on a.id = b.id;
```

4.4 FULL JOIN

MySQL does not support FULL JOIN, so you have to combine JOIN, UNION and LEFT JOIN to get an equivalent. It gives the results of A union B. It returns all records from both tables. Those columns which exist in only one table will contain NULL in the opposite table.

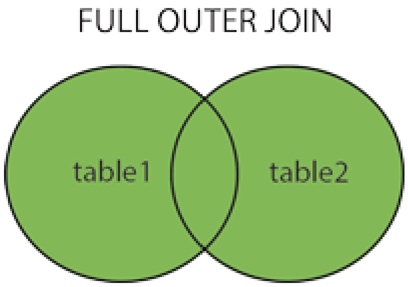


Figure visually illustrated Full join

Syntax:

```
SELECT * FROM a

LEFT JOIN b ON a.id = b.id

UNION

SELECT * FROM a

RIGHT JOIN b ON a.id = b.id
```

5. Homework before Lab

You must solve the following problems at home before the lab.

5.1 Problem Solution Modeling

After reading the reference material mentioned in the introduction, now you are ready to perform homework assigned to you.

Problem 1:

What is the purpose and utilities of Joins?

Problem 2:

Find and list down the different types of joins.

5.2 Practices from home

Solve the following subtasks.

Task-1

Difference between inner, outer, left, right, and, self joins?

Task-2

Describe how can use the joins with examples.

6. Procedure & Tools

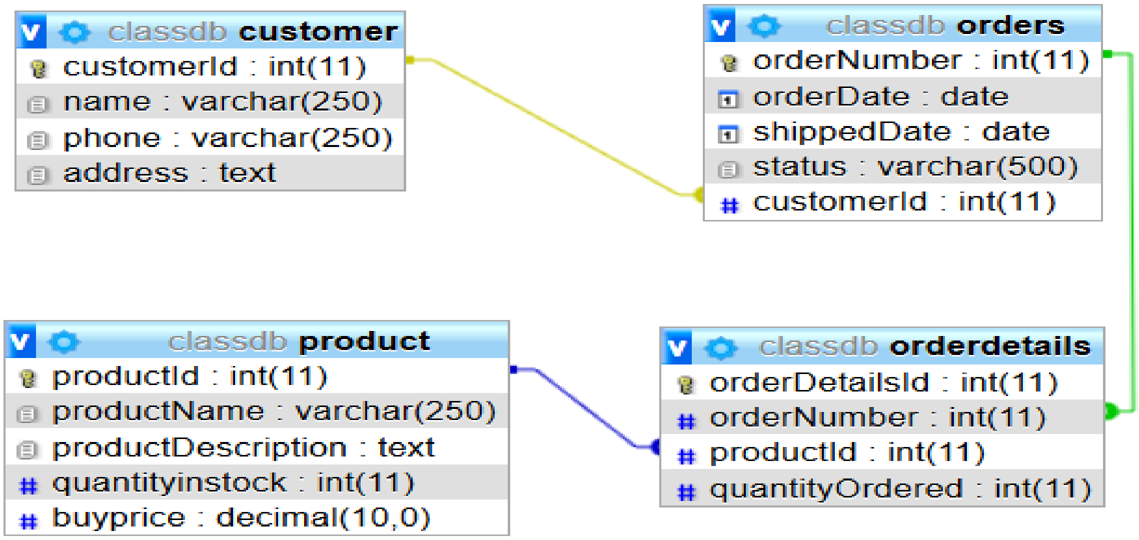
In this section, you will study how to make and run a customized exception.

6.1. Tools

In this section tools installation and setup is defined.

6.1. Walkthrough Task:

Let us consider this Schema which is sub part of Classic Database:



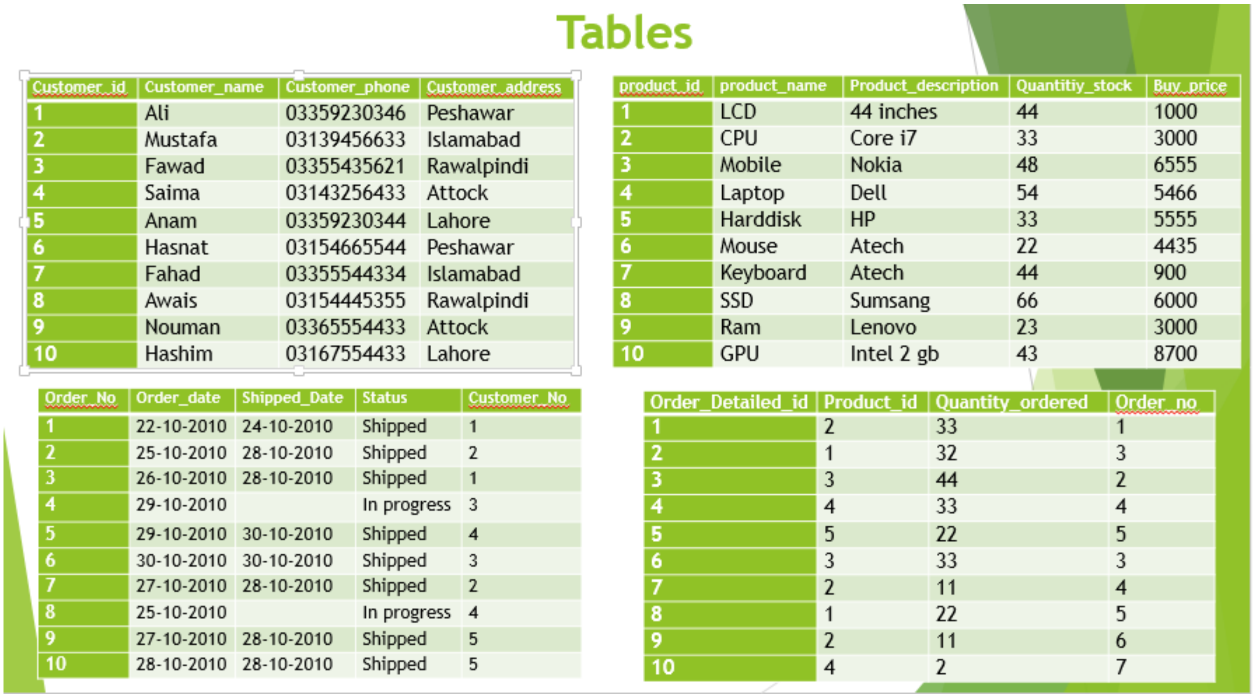
classdb customer
customerId : int(11)
name : varchar(250)
phone : varchar(250)
address : text

classdb orders
orderNumber : int(11)
orderDate : date
shippedDate : date
status : varchar(500)
customerId : int(11)

classdb product
productId : int(11)
productName : varchar(250)
productDescription : text
quantityinstock : int(11)
buyprice : decimal(10,0)

classdb orderdetails
orderDetailsId : int(11)
orderNumber : int(11)
productId : int(11)
quantityOrdered : int(11)

Tables Data: (Database file for walkthrough task have placed in lectures folder of Ghulam Mustafa)



Tables

Customer_id	Customer_name	Customer_phone	Customer_address
1	Ali	03359230346	Peshawar
2	Mustafa	03139456633	Islamabad
3	Fawad	03355435621	Rawalpindi
4	Saima	03143256433	Attock
5	Anam	03359230344	Lahore
6	Hasnat	03154665544	Peshawar
7	Fahad	03355544334	Islamabad
8	Awais	03154445355	Rawalpindi
9	Nouman	03365554433	Attock
10	Hashim	03167554433	Lahore

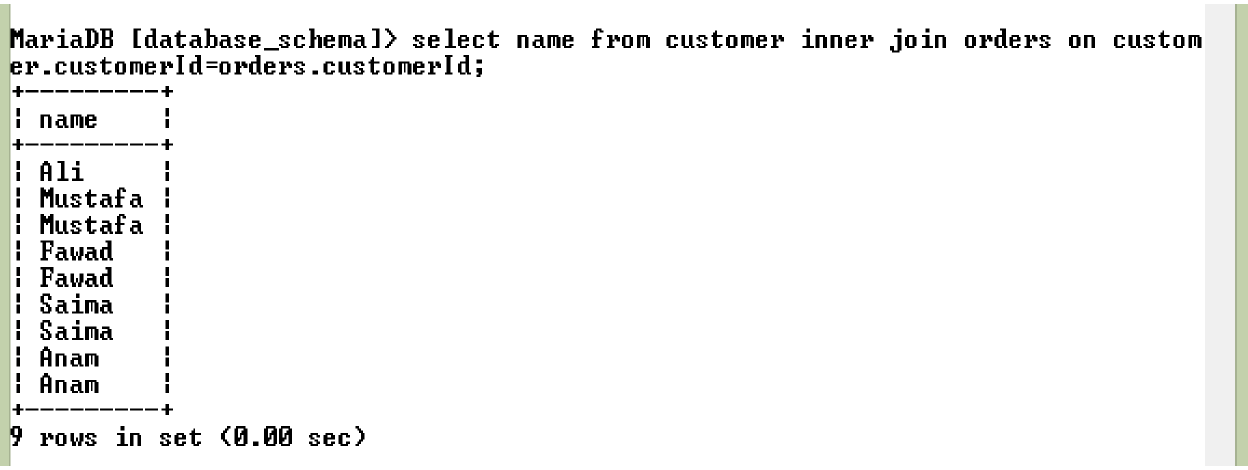
product_id	product_name	Product_description	Quantitiy_stock	Buy_price
1	LCD	44 inches	44	1000
2	CPU	Core i7	33	3000
3	Mobile	Nokia	48	6555
4	Laptop	Dell	54	5466
5	Harddisk	HP	33	5555
6	Mouse	Atech	22	4435
7	Keyboard	Atech	44	900
8	SSD	Sumsang	66	6000
9	Ram	Lenovo	23	3000
10	GPU	Intel 2 gb	43	8700

Order_No	Order_date	Shipped_Date	Status	Customer_No
1	22-10-2010	24-10-2010	Shipped	1
2	25-10-2010	28-10-2010	Shipped	2
3	26-10-2010	28-10-2010	Shipped	1
4	29-10-2010		In progress	3
5	29-10-2010	30-10-2010	Shipped	4
6	30-10-2010	30-10-2010	Shipped	3
7	27-10-2010	28-10-2010	Shipped	2
8	25-10-2010		In progress	4
9	27-10-2010	28-10-2010	Shipped	5
10	28-10-2010	28-10-2010	Shipped	5

Order_Detailed_id	Product_id	Quantity_ordered	Order_no
1	2	33	1
2	1	32	3
3	3	44	2
4	4	33	4
5	5	22	5
6	3	33	3
7	2	11	4
8	1	22	5
9	2	11	6
10	4	2	7

Inner Join:

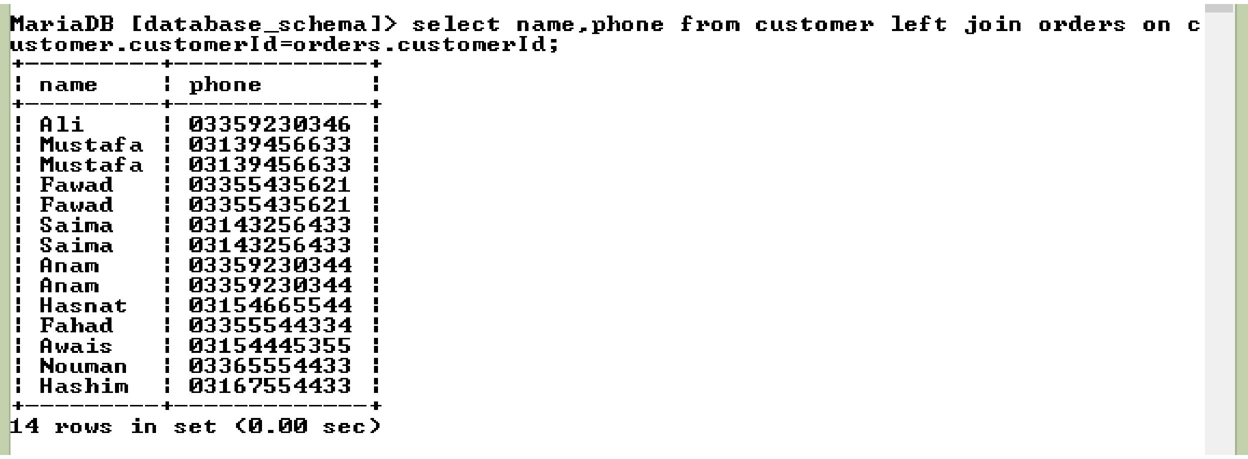
1) Show customer name who placed an order.



```
MariaDB [database_schema]> select name from customer inner join orders on custom
er.customerId=orders.customerId;
+---------+
| name    |
+---------+
| Ali     |
| Mustafa |
| Mustafa |
| Fawad   |
| Fawad   |
| Saima   |
| Saima   |
| Anam    |
| Anam    |
+---------+
9 rows in set (0.00 sec)
```

Left Join:

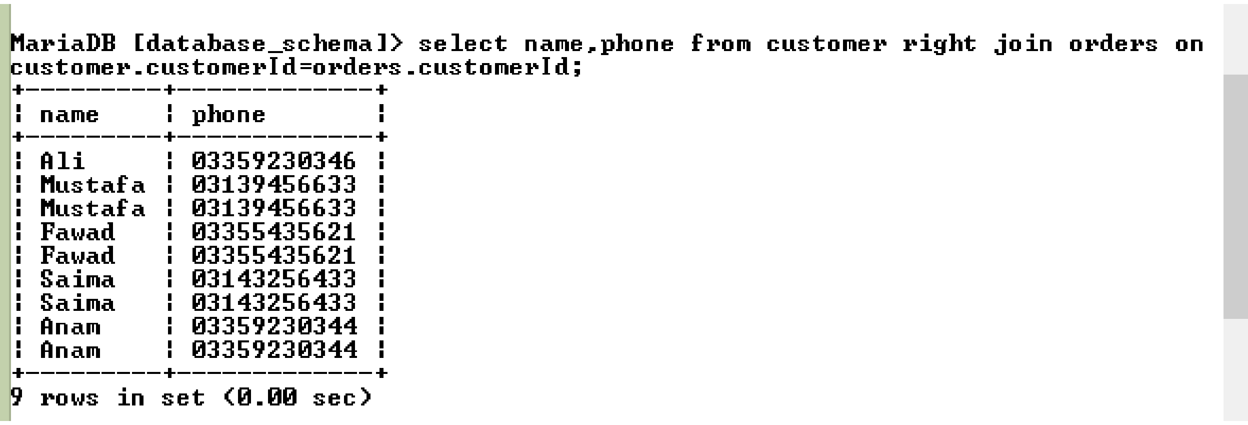
2) Show all the customer name and phone and their order details regardless of whether a customer placed an order or not.



```
MariaDB [database_schema]> select name,phone from customer left join orders on c
ustomer.customerId=orders.customerId;
+---------+-------------+
| name    | phone       |
+---------+-------------+
| Ali     | 03359230346 |
| Mustafa | 03139456633 |
| Mustafa | 03139456633 |
| Fawad   | 03355435621 |
| Fawad   | 03355435621 |
| Saima   | 03143256433 |
| Saima   | 03143256433 |
| Anam    | 03359230344 |
| Anam    | 03359230344 |
| Hasnat  | 03154665544 |
| Fahad   | 03355544334 |
| Awais   | 03154445355 |
| Nouman  | 03365554433 |
| Hashim  | 03167554433 |
+---------+-------------+
14 rows in set (0.00 sec)
```

Right Join:

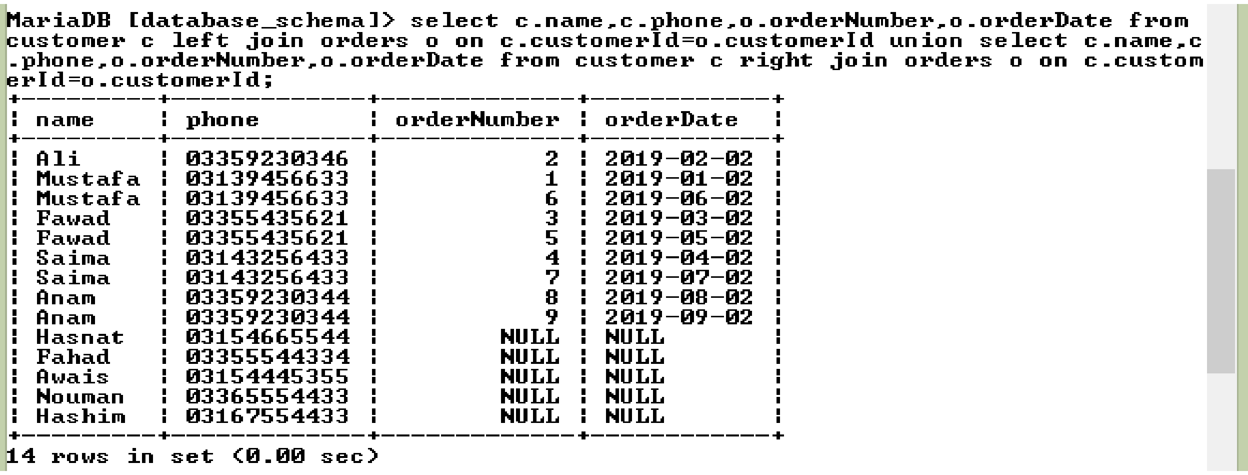
3) Show list of all orders information regardless of whether a customer information is present against or not.



```
MariaDB [database_schema]> select name,phone from customer right join orders on
customer.customerId=orders.customerId;
+---------+-------------+
| name    | phone       |
+---------+-------------+
| Ali     | 03359230346 |
| Mustafa | 03139456633 |
| Mustafa | 03139456633 |
| Fawad   | 03355435621 |
| Fawad   | 03355435621 |
| Saima   | 03143256433 |
| Saima   | 03143256433 |
| Anam    | 03359230344 |
| Anam    | 03359230344 |
+---------+-------------+
9 rows in set (0.00 sec)
```

Full join:

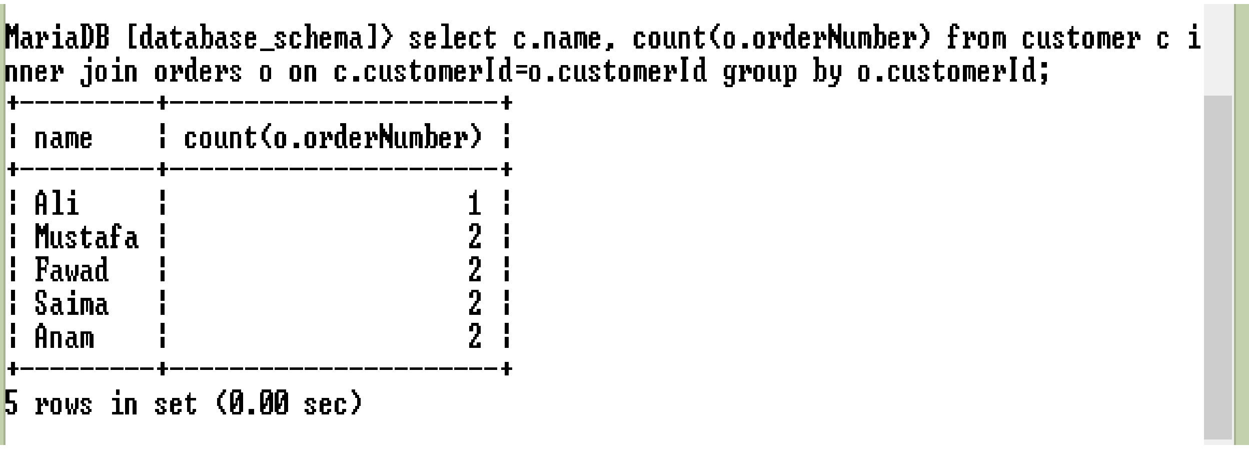
Show all the customer records and orders information.



```
MariaDB [database_schema]> select c.name,c.phone,o.orderNumber,o.orderDate from
customer c left join orders o on c.customerId=o.customerId union select c.name,c
.phone,o.orderNumber,o.orderDate from customer c right join orders o on c.custom
erId=o.customerId;
+---------+-------------+-------------+------------+
| name    | phone       | orderNumber | orderDate  |
+---------+-------------+-------------+------------+
| Ali     | 03359230346 |           2 | 2019-02-02 |
| Mustafa | 03139456633 |           1 | 2019-01-02 |
| Mustafa | 03139456633 |           6 | 2019-06-02 |
| Fawad   | 03355435621 |           3 | 2019-03-02 |
| Fawad   | 03355435621 |           5 | 2019-05-02 |
| Saima   | 03143256433 |           4 | 2019-04-02 |
| Saima   | 03143256433 |           7 | 2019-07-02 |
| Anam    | 03359230344 |           8 | 2019-08-02 |
| Anam    | 03359230344 |           9 | 2019-09-02 |
| Hasnat  | 03154665544 |        NULL | NULL       |
| Fahad   | 03355544334 |        NULL | NULL       |
| Awais   | 03154445355 |        NULL | NULL       |
| Nouman  | 03365554433 |        NULL | NULL       |
| Hashim  | 03167554433 |        NULL | NULL       |
+---------+-------------+-------------+------------+
14 rows in set (0.00 sec)
```

Inner join with group by:

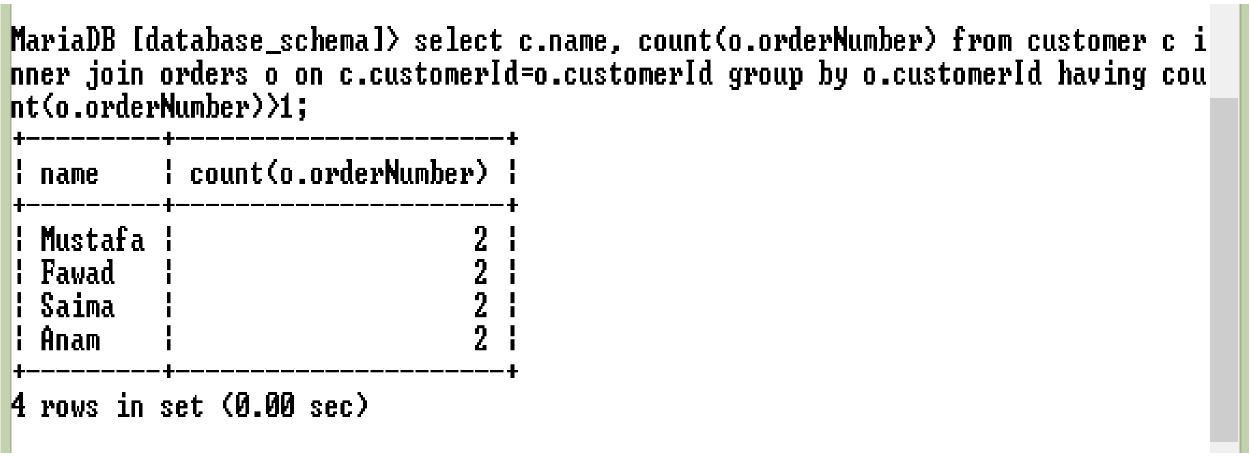
Show all the customer name and number of order they have placed.



```
MariaDB [database_schema]> select c.name, count(o.orderNumber) from customer c i
nner join orders o on c.customerId=o.customerId group by o.customerId;
+---------+----------------------+
| name    | count(o.orderNumber) |
+---------+----------------------+
| Ali     |                    1 |
| Mustafa |                    2 |
| Fawad   |                    2 |
| Saima   |                    2 |
| Anam    |                    2 |
+---------+----------------------+
5 rows in set (0.00 sec)
```

Inner join with group by and having:

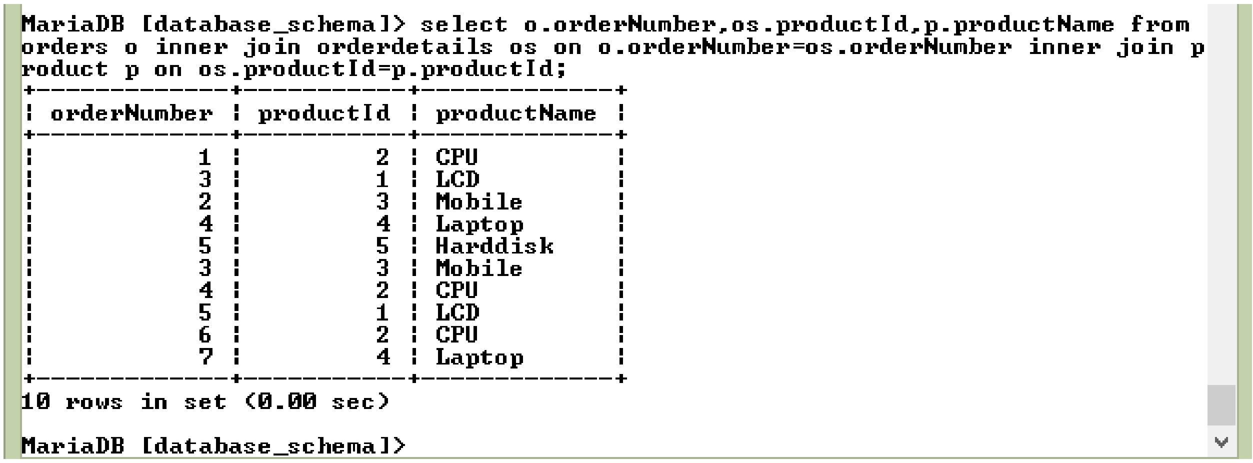
Show number of order of all customers where number of ordered greater than 1



```
MariaDB [database_schema]> select c.name, count(o.orderNumber) from customer c i
nner join orders o on c.customerId=o.customerId group by o.customerId having cou
nt(o.orderNumber)>1;
+---------+----------------------+
¦ name    ¦ count(o.orderNumber) ¦
+---------+----------------------+
¦ Mustafa ¦                    2 ¦
¦ Fawad   ¦                    2 ¦
¦ Saima   ¦                    2 ¦
¦ Anam    ¦                    2 ¦
+---------+----------------------+
4 rows in set (0.00 sec)
```

Joining three tables using inner join:

List all orders with product names, quantities, and prices.

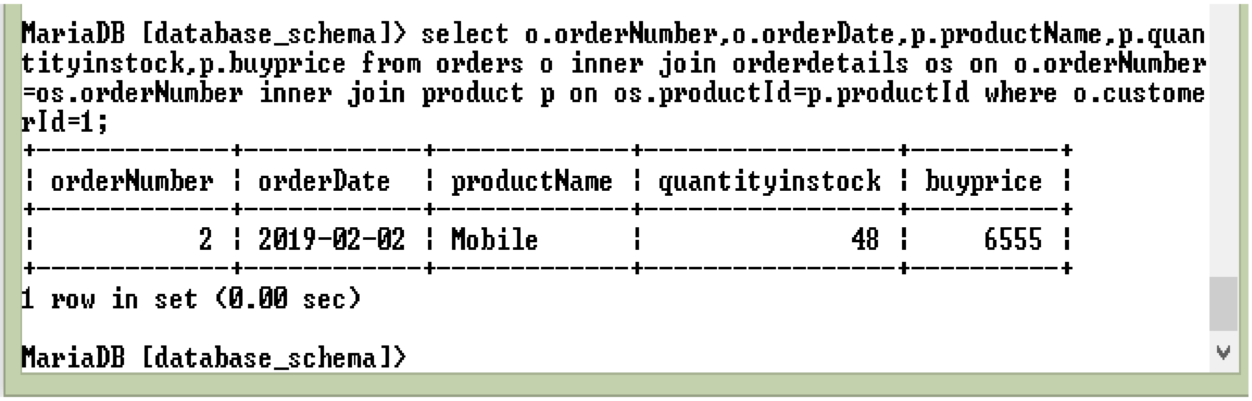


```
MariaDB [database_schema]> select o.orderNumber,os.productId,p.productName from
orders o inner join orderdetails os on o.orderNumber=os.orderNumber inner join p
roduct p on os.productId=p.productId;
+-------------+-----------+-------------+
¦ orderNumber ¦ productId ¦ productName ¦
+-------------+-----------+-------------+
¦           1 ¦         2 ¦ CPU         ¦
¦           3 ¦         1 ¦ LCD         ¦
¦           2 ¦         3 ¦ Mobile      ¦
¦           4 ¦         4 ¦ Laptop      ¦
¦           5 ¦         5 ¦ Harddisk    ¦
¦           3 ¦         3 ¦ Mobile      ¦
¦           4 ¦         2 ¦ CPU         ¦
¦           5 ¦         1 ¦ LCD         ¦
¦           6 ¦         2 ¦ CPU         ¦
¦           7 ¦         4 ¦ Laptop      ¦
+-------------+-----------+-------------+
10 rows in set (0.00 sec)

MariaDB [database_schema]>
```

Joining three tables using inner join and where clause :

Show product names, quantity, and total price of all products ordered by customer id=1



```
MariaDB [database_schema]> select o.orderNumber,o.orderDate,p.productName,p.quan
tityinstock,p.buyprice from orders o inner join orderdetails os on o.orderNumber
=os.orderNumber inner join product p on os.productId=p.productId where o.custome
rId=1;
+-------------+------------+-------------+-----------------+----------+
¦ orderNumber ¦ orderDate  ¦ productName ¦ quantityinstock ¦ buyprice ¦
+-------------+------------+-------------+-----------------+----------+
¦           2 ¦ 2019-02-02 ¦ Mobile      ¦              48 ¦     6555 ¦
+-------------+------------+-------------+-----------------+----------+
1 row in set (0.00 sec)

MariaDB [database_schema]>
```

7. Practice Tasks

This section will provide more practice exercises which you need to finish during the lab. You need to finish the tasks in the required time. When you finish them, put these tasks in the following folder:
\\fs\assignments$\IDBS\Lab11$

7.1 Practice Task [Expected time = 40mins]

Consider the schema given in Lab01 (7.1 Practice Task), write down following SQL queries. You can use data types of your own choice.

1. List products that have not sold by order date.
2. List customers that have not placed orders.
3. List all the orders for the 1940 Ford Pickup Truck.
4. List the names of customers and the corresponding order numbers where a particular order from that customer has a value greater than Rs25,000?
5. Are there any products that appear on all orders?
6. List those orders containing items sold at less than the MSRP.
7. List the products ordered on Monday.
8. What is the quantity on hand for products listed on 'On Hold' orders?

9. Show product name, buy price and text description of those products whose buy price is greater than 77rs.
10. Show number of order of all customers where number of ordered greater than 2.
11. Show customer name, customer number, customer address, total price of an order, order date, status of the order, order number from customer, orders, and orderdetails of those customer who's Sale between 9000 to 10000.
12. Write a SQL statement to know which employee are working for which customer.
13. Write a SQL statement to make a list in ascending order for the employees who works either through an employees or by own.
14. Write a SQL statement to make a report with customer name, city, order number, order date, order amount employee name to find that either any of the existing customer have placed no first order or placed one or more orders.
15. List all orders with customers information.
16. List all orders with product names, quantities, and prices.
17. Show product names, quantity, and total price of all products ordered by customer "Asim Ali"
18. Show product name, quantity, and total price of all products ordered in order number "10103"
19. Show order number, total products ordered, and total price for order number "10103"
20. List the products in each product line.

7.3Out comes

After completing this lab, student will be able to understand the usage right and LEFT joins.

8. Evaluation Task (Unseen) [Expected time = 55mins for two tasks]

The lab instructor will give you unseen task depending upon the progress of the class.

8.1.Evaluation criteria

The evaluation criteria for this lab will be based on the completion of the following tasks. Each task is assigned the marks percentage which will be evaluated by the instructor in the lab whether the student has finished the complete/partial task(s).

Table 3: Evaluation of the Lab

Sr. No.	**Task No**	**Description**	**Marks**
1	6	Procedures and Tools	05
2	7	Practice tasks and Testing	15
3	8	Evaluation Tasks (Unseen)	80

9. Further Reading

This section provides the references to further polish your skills.

9.1.Text Book

Database Systems, A practical approach to design, implementation and management by Thomas Connolly, Carolyn Begg, Addison Wesley , Fifth Edition,

9.2.Slides

The slides and reading material can be accessed from the folder of the class instructor available at \\fs\lectures$\

10. REFERENCES:

10.1. SQL-99 Complete, Really, by Peter Gulutzan & Trudy Pelzer.

- More examples for the SELECT command: http://dev.mysql.com/doc/mysql/en/select.html

 MySQL operators:
http://dev.mysql.com/doc/mysql/en/non-typed_operators.html

- Built-in functions: http://dev.mysql.com/doc/mysql/en/functions.html
- Joining tables: http://www.melonfire.com/community/columns/trog/article.php?id=148
- Using subqeries: http://www.melonfire.com/community/columns/trog/article.php?id=204

 Using subqeries:
http://www.melonfire.com/community/columns/trog/article.php?id=204