Practical-6

Aim: To implement Joins and Views.

As the name shows, JOIN means to combine something. In case of SQL, JOIN means "to combine two or more tables". In SQL, JOIN clause is used to combine the records from two or more tables in a database.

Cross JOIN or Cartesian Product:

This type of JOIN returns the cartesian product of rows from the tables in Join. It will return a table which consists of records which combines each row from the first table with each row of the second table.

Cross JOIN Syntax is:

SELECT column-name-list FROM table-name1 CROSS JOIN table-name2; **INNER Join or**

EQUI Join:

This is a simple JOIN in which the result is based on matched data as per the equality condition specified in the SQL query.

Inner Join Syntax is:

SELECT column-name-list FROM table-name1 INNER JOIN table-name2 WHERE tablename1.column-name = table-name2.column-name; **Natural JOIN:**

Natural Join is a type of Inner join which is based on column having same name and same datatype present in both the tables to be joined.

The syntax for Natural Join is:

SELECT * FROM table-name1 NATURAL JOIN table-name2; **OUTER JOIN:**

Outer Join is based on both matched and unmatched data. Outer Joins subdivide further into,

1. Left Outer Join
2. Right Outer Join
3. Full Outer Join

1. Left Outer Join:

The left outer join returns a resultset table with the matched data from the two tables and then the remaining rows of the left table and null from the right table's columns.

Syntax for Left Outer Join is:

SELECT column-name-list FROM table-name1 LEFT OUTER JOIN table-name2 ON tablename1.column-name = table-name2.column-name;

2. Right Outer Join:

The right outer join returns a resultset table with the matched data from the two tables being joined, then the remaining rows of the right table and null for the remaining left table's columns.

Syntax for Right Outer Join is:

SELECT column-name-list FROM table-name1 RIGHT OUTER JOIN table-name2 ON tablename1.column-name = table-name2.column-name;

3. Full Outer Join:

The full outer join returns a resultset table with the matched data of two table then remaining rows of both left table and then the right table.

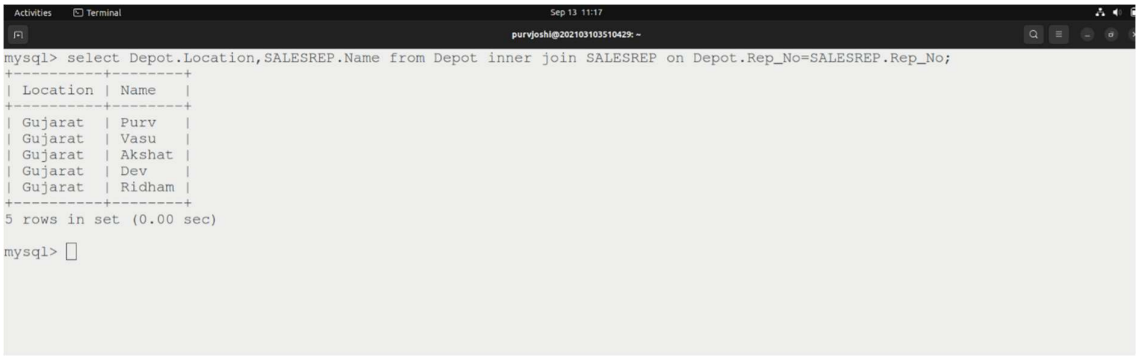
Syntax of Full Outer Join is:

SELECT column-name-list FROM table-name1 FULL OUTER JOIN table-name2 ON tablename1.column-name = table-name2.column-name;

Queries:

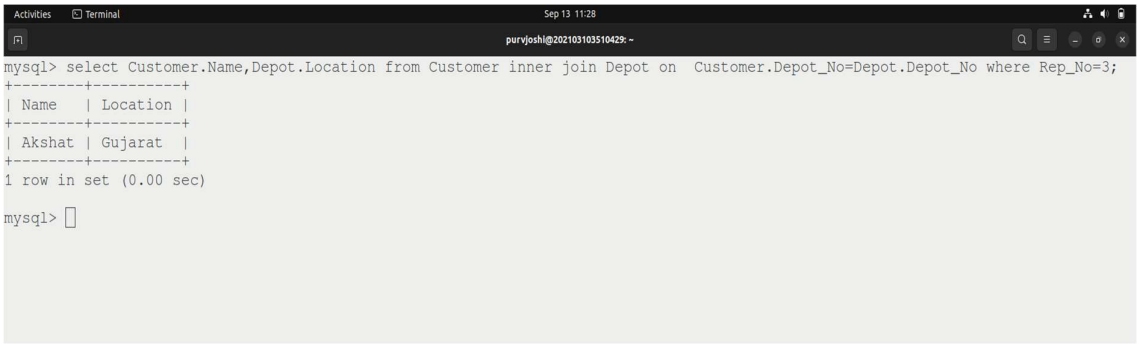
(Q-1) Give a list of depot locations paired with the name of the sales rep who covers that depot.

Query: select Depot. Location, SALESREP. Name from Depot inner join SALESREP on Depot. Rep_No=SALESREP. Rep_No;



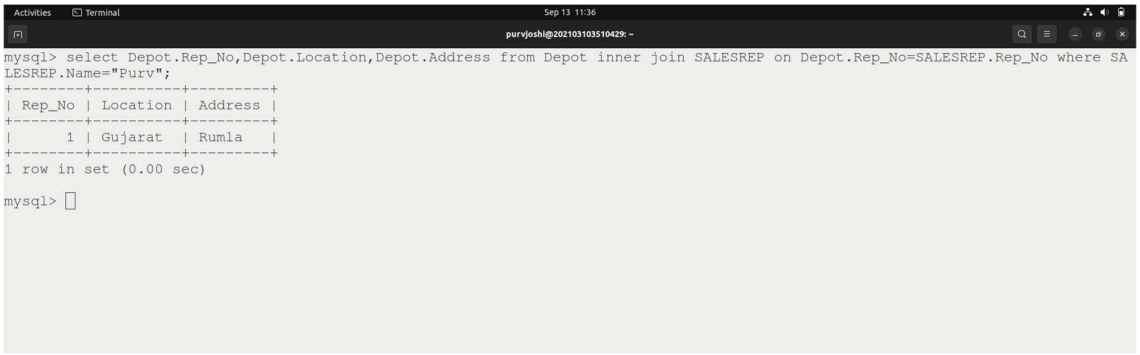
(Q-2) List the customer's name and the depot location for the depot delivering to that customer for all customers who receive deliveries from depots looked after by sales rep number (rep_no) 3.

Query: select Customer.Name, Depot. Location from Customer inner join Depot on Customer.Depot_No-Depot.Depot_No where Rep_No=3;



(Q-3) List the sales rep number (rep_no) and depot location and address for depots looked after by the sales rep whose name is Purv.

Query: select Depot. Rep_No, Depot. Location, Depot. Address from Depot inner join SALESREP on Depot. Rep_No-SALESREP. Rep_No where SA LESREP.Name="Purv";



(Q-4) For all order lines (online) for all orders (Order_No) for customers whose name is Purv, list the customer address, the date_placed, the Product_no and the quantity.

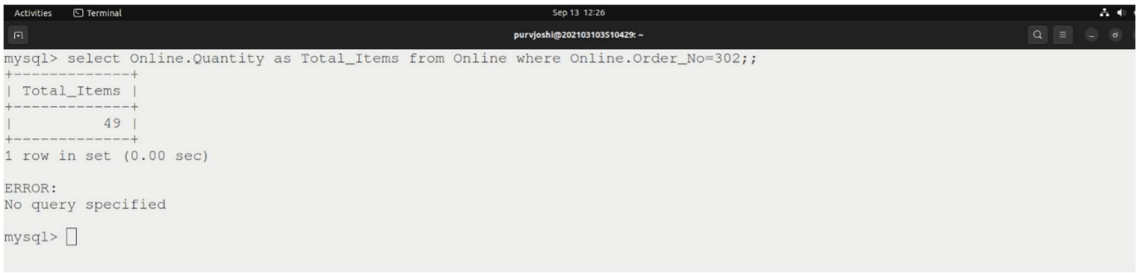
Query: select Customer.Address, Orders.Date_Placed, Online. Product_No, Online. Quantity from Customer inner join Orders on Customer .Customer_No=Orders.Customer_No inner join Online on Online. Order_No=Orders.Order_No where Customer.Name="Purv";

(Q-5) Give the total number of items (quantity) in stock in all depots.

Query: select sum (Quantity) from Stock;

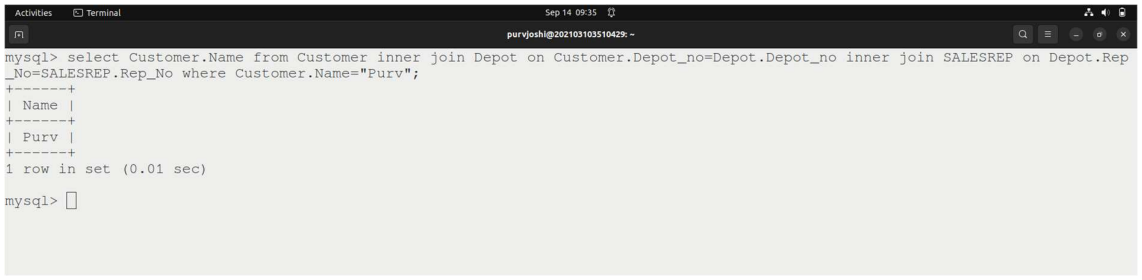
(Q-6) Give the total number of items (order line quantity) which have been ordered on the order with Order_no=302.

Query: select Online. Quantity as Total_Items from Online where Online. Order_No=302;



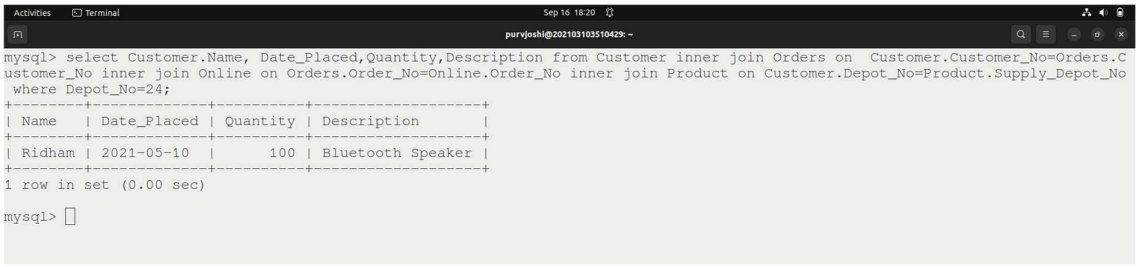
(Q-7) List the names of all customers who receive deliveries from depots which are looked after by the sales rep whose name is Purv.

Query: select Customer.Name from Customer inner join Depot on Customer.Depot_no=Depot. Depot_no inner join SALESREP on Depot.Rep _NO=SALESREP.Rep_No where Customer.Name="Purv";



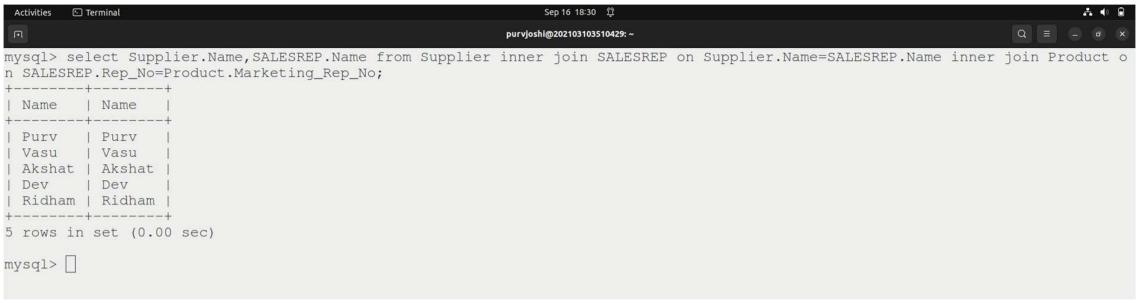
(Q-8) List the customer name, order date_placed, order line quantity and product description for each order line (with its linked, order, customer and product rows) for customers who receive deliveries from depot number 24.

Query: select Customer.Name, Date_Placed, Quantity, Description from Customer inner join Orders on Customer.customer_No-Orders.Customer_No inner join online on Orders.Order_No=Online.Order_No inner join Product on Customer.Depot No=Product.Supply_Depot_No where Depot_No=24;



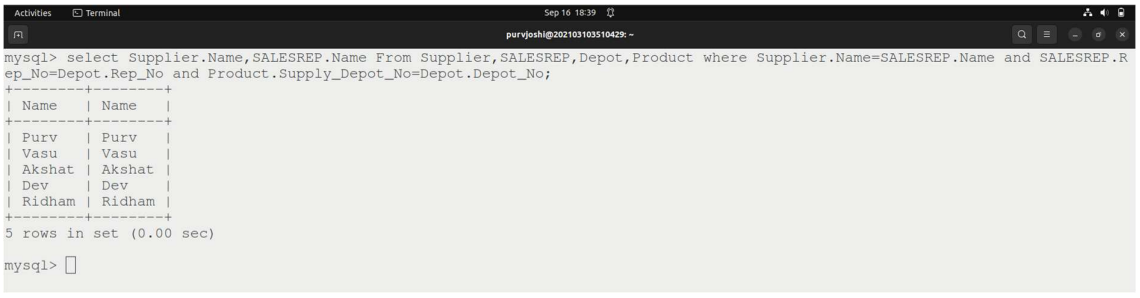
(Q-9) List supplier names paired with the names of the sales reps who market products supplied by that supplier.

Query: select Supplier. Name, SALESREP.Name from Supplier inner join SALESREP on Supplier.Name=SALESREP.Name inner join Product on SALESREP.Rep_No=Product.Marketing_Rep_No;



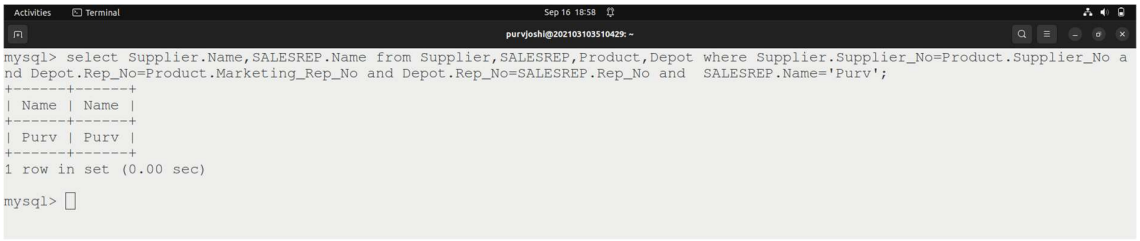
(Q-10) List supplier names paired with the names of the sales reps who look after the depots where products from that supplier are delivered.

Query: select Supplier.Name, SALESREP.Name From Supplier, SALESREP, Depot, Product where Supplier.Name SALESREP.Name and SALESREP.Rep_No=Depot. Rep_No and Product.Supply_Depot_No=Depot. Depot_No;



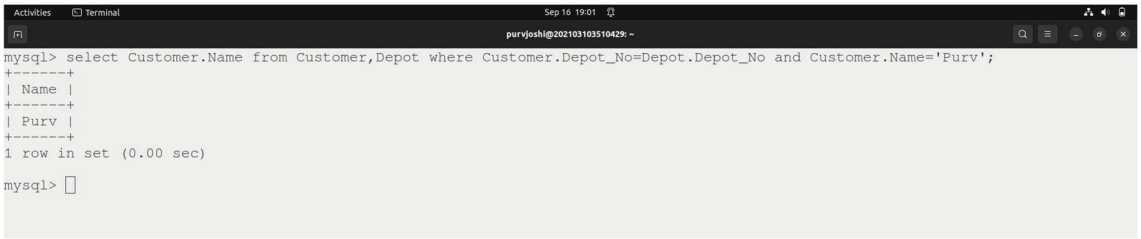
(Q-11) List the names of all customers who have ordered products which are marketed by the sales rep whose name is Purv.

Query: select Supplier. Name, SALESREP.Name from Supplier, SALESREP, Product, Depot where Supplier.Supplier_No-Product.Supplier_No a nd Depot.Rep_No-Product.Marketing_Rep_No and Depot. Rep_No-SALESREP. Rep_No and SALESREP.Name='Purv';



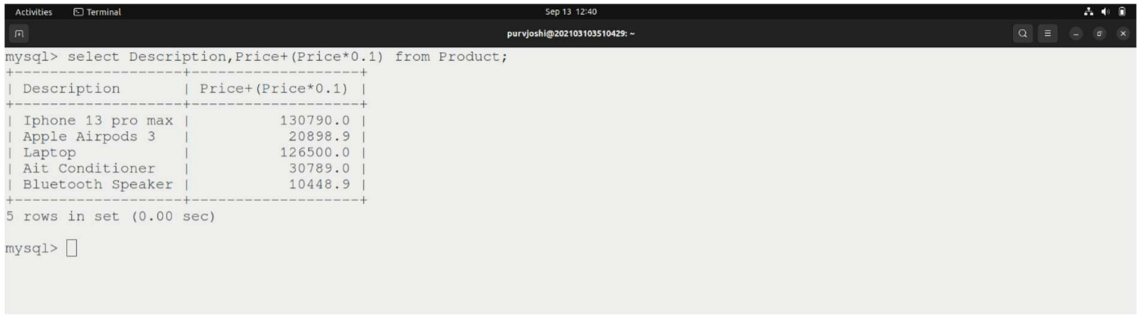
(Q-12) List the names of all customers who are delivered to by the depot which delivers to the customer whose name is Purv.

Query: select Customer. Name from Customer, Depot where Customer.Depot No=Depot.Depot_No and Customer.Name='Purv';



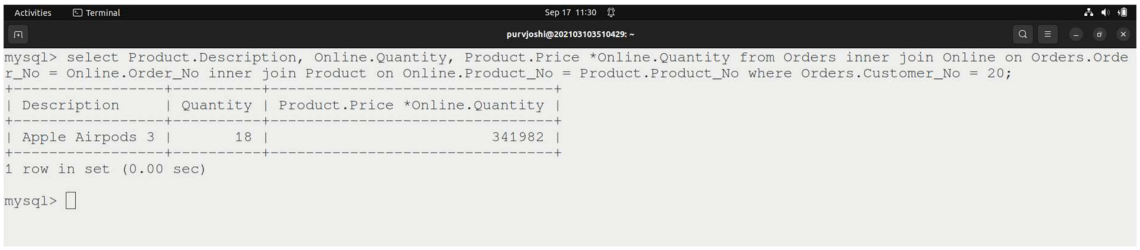
(Q-13) List each product description and its price increased by 10%.

Query: select Description, Price+ (Price*0.1) from Product;



(Q-14) List all order lines for the customer with customer_no 20 giving the product description, the order line quantity and the value of the order line. (i.e. the order line quantity * the price from the linked product row).

Query: select Product. Description, Online.Quantity, Product.Price *Online. Quantity from Orders inner join Online on Orders.Orde r_No Online. Order_No inner join Product on Online. Product No = Product.Product No where Orders. Customer_No = 20;



(Q-15) List the locations and addresses of all depots which do not stock product number 122. (i.e., where there is no stock row for that product for the depot).

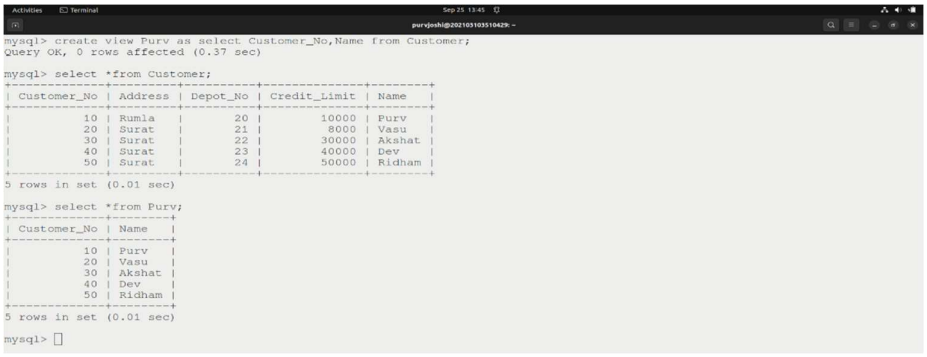
Query: select Depot. Location, Depot.Address from Depot inner join Stock on Depot. Depot No=Stock.Depot_No where Stock.Product_No not in (122);

(Q-16) Set up a query which lists the names of all customers who have placed an order with the order number (Order_no) of the order merged with the names of all customers who have never placed an order (shown once, with the order number attribute null) i.e., an outer join.

Query: select Customer. Name, Orders. Order_No from Customer left join Orders on Customer. Customer_No=Orders. Customer_No;

(Q-17) Creating view for database.

Query: create view Purv as select Customer_No, Name from Customer;



```
mysql> create view Purv as select Customer_No,Name from Customer;
Query OK, 0 rows affected (0.37 sec)

mysql> select *from Customer;
+-------------+---------+----------+--------------+--------+
| Customer_No | Address | Depot_No | Credit_Limit | Name   |
+-------------+---------+----------+--------------+--------+
|          10 | Rumla   |       20 |        10000 | Purv   |
|          20 | Surat   |       21 |         8000 | Vasu   |
|          30 | Surat   |       22 |        30000 | Akshat |
|          40 | Surat   |       23 |        40000 | Dev    |
|          50 | Surat   |       24 |        50000 | Ridham |
+-------------+---------+----------+--------------+--------+
5 rows in set (0.01 sec)

mysql> select *from Purv;
+-------------+--------+
| Customer_No | Name   |
+-------------+--------+
|          10 | Purv   |
|          20 | Vasu   |
|          30 | Akshat |
|          40 | Dev    |
|          50 | Ridham |
+-------------+--------+
5 rows in set (0.01 sec)

mysql>
```