**ASSIGMNENT 2** karthikavula76@gmail.com

**RDBMS Fundamentals**

1. Write a SELECT query to retrieve all columns from a 'customers' table, and modify it to return only the customer name and email address for customers in a specific city.

Ans:

Here is the create the customers tables

Create table customers (empid, name, email, city, state);

Insert into customers vaules (101, ‘santhosh’, ‘santhosh@2410.com’, ‘Hyd’, ‘TG’);

Insert into customers vaules (102, ‘karthik’, ‘karthik@220.com’, ‘Delhi’, ‘Delhi’);

Insert into customers vaules (103, ‘chandu’, ‘ch@230.com’, ‘wrgl’, ‘TG’);

Insert into customers vaules (104, ‘vikki’, ‘vikki@240.com’, ‘mumbai’, ‘MH’);

Insert into customers vaules (105, ‘sri’, ‘sri@2410.com’, ‘vij’, ‘AP’);

Select \* from customers;

Select name, email from customers where city = ‘ HYD’;

Craft a query using an INNER JOIN to combine 'orders' and 'customers' tables for customers in a specified region, and a LEFT JOIN to display all customers including those without orders.

Ans:

**INNER JOIN** to combine ‘orders’ and ‘customers’ tables for customers in a specified region:

Create table customers (cust\_id int primary key, name varchar(20), email varchar(20), city varchar(20));

Insert into customers (cust\_id, name, email, city) vaules (1 ‘karthik’, ‘karthik@24.com’, ‘Hyd’);

Insert into customers vaules (2, ‘karthik’, ‘sri@2410.com’, ‘karimnagar’);

Insert into customers vaules (3, ‘viddu’, ‘viddu@2204.com’, ‘Hyd’);

Insert into customers vaules (4, ‘hari’, ‘hari@1401.com’, ‘ngl’);

Insert into customers vaules (5, ‘krishan’, ‘kri@2367.com’, ‘Wrgl’);

Select \* from customers;

Create table orders (order\_id int primary key, cust\_id int, order\_date DATE);

Insert into orders values (101, 1, ’04-may-2024’);

Insert into orders values (102, 2, ’24-oct-2000’);

Insert into orders values (103, 3, ’22-april-20o1’);

Select \* from orders;

**INNER JOIN (Combining Orders and Customers):** To retrieve customers who placed orders in a specific region, we’ll use an INNER JOIN. For instance, if we want to find orders from the ‘Hyd’ city:

Select c.name, o.order\_id from customers c inner join o ON c.cust\_id where city = ‘Hyd’;

**LEFT JOIN (Including All Customers):** To display all customers (including those without orders), we’ll use a LEFT JOIN:

Select c.name, o.order\_id from customers c left join o ON c.cust\_id;

Utilize a subquery to find customers who have placed orders above the average order value, and write a UNION query to combine two SELECT statements with the same number of columns.