Task 5 Writing join queries, equivalent queries, recursive queries

Date 9/9/25

Aim: To implement and execute join queries, equivalent queries and recursive queries.

Types of joins in SQL:

1. Inner join: Returns records that have matching values in both tables.

Syntax: Select column_name(s) from table1 INNER JOIN table2 on table1.column_name = table2.column_name;

2. left outer join: Returns all records from the left table, and the matched records from the right table.

Syntax: Select column_name(s) from table1 left join table2 on table1.column_name = table2.column_name;

3. Right outer join: Return all records from the right table, and the matched records from the left table.

Syntax: Select column_name(s) from table1 Right join table2 on table1.column_name = table2.column_name;

4. Full outer join: Returns all records when there is a match in either left or right table.

Syntax: Select column_name(s) from table1 full Outer join table2 on table1.column_name = table2.colum_name;

1. Join queries

Create Tables

```
Create table customer(
    customerID int primary key,
    name varchar(50),
    address varchar(100)
    );
Create table bank_account(
    account_number int primary key;
    customerID int,
    balance int,
    category varchar(50),
    foregin key (customerID) references customer
     (customeID)
    );
Create  table branch(
    branchID int primary key,
    branchName varchar(50),
    );
```

2. Insert Sample data.

```
Insert into customer (customerID, name, address) values
(101, 'Ram kumar', 'chennai');
Insert into customer (customerID, name, address) Values
(102, 'Vijay Rao', 'Hyderabad');
Insert into customer(customerID, name, address)Values
(103, 'Vasu Reddy', 'Vizag');
Insert into customer (customerID, name, address) values
(104, 'Vinay kumar', 'chennai');
Insert into customer (customerID, name, address) values
(105, 'Rohit', 'Delhi');
Insert into customer (customerID, name, address) values
(106, 'Varun', 'Kerala');
```

Insert into bank. account (account_number, customerID, balance, category) values (1001, 101, 15000, 'Savings');

Insert into bank. account (account_number, CustomerID, balance, category) values (1002, 102, 0, 'current');

Insert into bank-account (account-number, CustomerID, balance, category) values (1003, 103, 5000, 'Savings');

Insert bank-account (account_number, CustomerID, balance, Category) values (1004, 10 5, 2000, 'current');

Insert into branch (branchID, branchName) values (1, 'Chennai Branch');

Insert into branch (branchID, branchName) values (2, 'Hyderabad Branch');

Insert into branch (branchID, branchName) values (3, 'Vizag Branch');

3. Join Queries:

(a) Inner join:

Query: Select c. name, b. account_number from Customer c inner join bank-account b ON c. customer ID = b. customerID;

Output:

Name	account - Number.
Raj kumar	1001
Vijay Rao	1002
Venu Reddy	1003
Vinay kumar	1004

left join:

query: Select c.name, b.account number from customer c left join bank account b on c.customer id = b.customer id;

output

name	account number
Raj Kumar	1001
Vijay Das	1002
Amar Reddy	1003
Vinay Kumar	1004
Rohit Sharma	null

c) right join:

Query: Select c.name, b.account number from customer c Right join bank account b on c.customer id = b.customer id;

output

Name	account number
Raj Kumar	1001
Vijay Das	1002
Amar Reddy	1003
Vinay Kumar	1004

d) full outer join:

Query Select c.name, b.account number from customer c full outer join bank account b on c.customer id = b.customer id;

(b) Left Join

Query: Select c.name, b.account number from customers c left join bank account b on c.customerID = b.customerID;

Output

Name	Account number
Raj Kumar	1001
Vijay Rao	1002
Ravi Reddy	1003
Vinay Kumar	1004
Rohit Sharma	NULL

(c) Right Join

Query: Select c.name, b.account number from customers c right join bank account b on c.customerID = b.customerID;

Output

Name	Account number
Raj Kumar	1001
Vijay Rao	1002
Ravi Reddy	1003
Vinay Kumar	1004

(d) Full outer join

Query: Select c.name, b.account number from customers c full outer join bank account b on c.customerID = b.customerID;