Posk 5: Writing Join Queries, Equivalent, AND/OR 109-25 Recursive Queries.

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

Types of Joins in SQL:

in both table.

SYNTAX: Select column = name(s) From table 1 INNER

JORN table 2 ON table 1. Column = name = table 2. colu
mn_Mame;

2. Left outer JoIN: - Returns all records from the lest toble, and the matched records from the right table syntax: Select column_name(s) FROM table 1 LEFT JoIN table 2 ON table 1. column_mame = table 2. column_name.

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

Syntax: Column-name(S) FROM table 1 RIGHT JOIN

table 2 ON table 1. column-name = table 2. column-name;

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

Syntax: Select column_name (s) From table 1
Full Outer Join table 2 on table 1. Column_name =
table 2. column_name;

1. JOIN QUERCES :

Create Tables

Create table customer(

customer ID int primary key, name varichar (50), address varichar (150)

```
Create Table bank account (
     account-number int primary key,
     Customer ID int,
      balance int,
      Category varchar (so)
      Foreign key (customer (customer (customer (customer PD)
 );
Create table branch (
     branch2D int primary key,
      branch Name vonchor (50),
);
2. Insert sample data
 insert into customer (customerID, name, address) values
  (101, 1 Ram kuman', 'chennai');
 insert into customer (customerID, name, address) values
  (102 y 'vijay RAO', 'Hyderabad');
 insent into customer(customerID, name, address) va-
 -lues (103, 'vasu', Vizag');
  insert into Customer (customer DD, name, address)
   Values (104, 'vinay', 'chennai');
  insert into customer (customer ID, name, address)
  values (105, 'Rohit', Delhi)
          into bank-account (account-number, custome)!
   bolance, category) values (1000) 101, 15000, souings)
   insert into bank-account (account-number, customerse
   balance, category) values (1002, 102, D, Current);
          into bank-account (account - rumber, contomer ID)
   balance, categily) values/1003, 103, 5000, savings
```

insert into bank account (a crount_number, Customer20, balance, category) values (1004, 105, 2000, 1 courrent);

insert into branch (branch ID, branch Name) values
(1, 1 Chenna; * Branch');

insert into Branch (branch ID, branchware) values (2)
'Hyderabod Branch');

insert into branch (branch ID, branch dame) values (-3, vizag Branch);

& Join Queries

(a) Inner Join:

Query: - Select Conome, baccount number from customers C inner John bank account 6 and count b. customers D;

Output :-

Name	account_Number
Ram kumon	1001
vijay	(002
wiray vasu	1003
vinay	1004

6) Left Join :-

Query: Select c. name, b.acrount_number from customer C left join bank-acrount & onl c. Custo-merID = b.CustomerID;

Output:

Name	account - number
Rankuman	1001
vijay	1002
was & a	£00)
vinay	(004-
Rohit	NOWLL

= ? Right Join !-

Query: Select C.name, b. account_number from Customers C Right Join bank-account b on c.customers = b. CustomersD;

Output:

Name	account_number
Ram kumas	1001
uijag	(00)=
valu	£ 00)
Vinag	1004

d) Full Outer Join:

Query: Select coname, baccount-humber from cultoner c Full outer join bank-account book constoners

= b. customen 20;

M-4011/01/22	
Name	account number
Ram Kumay	1001
uijay	1000
valu	$\varepsilon \omega s$
viray	4004
Pohit'	WULL NULL

Equivalent Query

Query: Select c. name As customerslame, b. accountnumber As Account number From customer C John bank-account b on c. customers = b. customers = 0;

output:

Customer Name	-Account Number
Ramtumon	1001
uijay	1002
Vasu	1003
vinay	1004

bousing Subacery

Query: Using c. nome As customer Name, (select 6. account number From bank-account to where 6. customer ID = c. customer ID limit 1 As Account Number From customer

c;

뭐하다 그 아니라인 모양하다 일보다 !		200
customerdame	stomerdame -account Number	
Kom Kuman	loot	g0
vijay	(00)	
rasu	(003	16
vinay	100≯	
Rohith	NULL	4,15

5) Recursive Query:With Recursive teefer tool Hierarchy As (select
Customer ID, referred By ID From Customer where
reffer Bed By ID is NOT NULL UNION
Select c. customer ID, c. referred By ID From Customer
C Join Referral Hierarchy vh Dr c. referred By ID:
The customer ID) select from Referral Hierarchy;

autput:

customeIID	referredbyID
102	(0)
103	102
104	103

and the second	
VEL TECH	End.
EX NO.	1
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	6
VIVA VOCE (5)	5
RECORD (5)	5
TOTAL (20)	13
S ANWITH DATE	12
The state of the second	as
181011	D
(_

Result: The implementation of SQL commands erecuted and recurgive orwings one using Joins Successfully.