Task 5 :- WRITING JOIN QUERIES, EQUIVALENT AND/OR RECURSIVE QUERIES

Aim:- To implement and Execute join queries equivalent queries and recursive queries using a university database scenario.

Procedure:
The Sql Join clause is used to combine records from two or more table in a database. A join is a more ~~of~~ for combine fields from two tables by using values common values.

* Create the database and tables students, Department, courses (Enrolment).

* write Sql Queries using different types of joins.

* insert sample data.

* Display results and verify correction.

Syntax:

Select column, column2, column 3, from table name, table same 2 where_name, column = table _name 2 column Name.

Types of joins:-

1. Simple joins
2. self join
3. Outer join.

1. Inner join:- Returns recursive that have matching values in both tables SELECT column_name (s) from table. Inner join table 2 on table, column from the left table.
~~[illegible]~~

left outer join:- Return all reserul from the left and the records from the guide

Phone_id	brand	model	price	ram	storage	battery
1	Realme	14pro	30000	16GB	256GB	3000MAH
2	Redmi	10pro	15000	8GB	128GB	4500MAH
3	vivo	13pro	25000	12GB	256GB	5500MAH

Phone_id	brand	model	price	ram	storage	battery
1	realme	14pro	30000	16GB	256GB	5000mah
2	redmi	10pro	15000	8GB	128GB	4500mah
3	vivo	13pro	25000	12GB	256GB	5500mah

Phone_id	brand	model	price	ram	storage	battery
1.	realme	14pro	30000	16GB	256GB	5000mah
2.	redmi	10pro	15000	8GB	128GB	2500mah
3.	vivo	13pro	25000	12GB	256GB	5500mah

SELECT M.Phone.id ; m.brand , m; model , S ram,
S storage S.battery
FROM Mobile phones m
LEFT JOIN phone specifications on m.phone
id = S.phone_id

INSERT INTO payment values (30000 ~~A~~, 15000, 25000, 2025-08-17);

1) Join Queries

a) INNER JOIN

SELECT m.phone_id, m brand , m .model,
S.ram , S.Storage S. battery.
FROM Mobile phone
Inner join phone specification on m phone _ id = S phone id.

b) LEFT JOIN

SELECT M phone id m brand m.model S. ram
S - Storage , S - battery .
Left join phone specification on m phone id = S phone id

c) Right Join

SELECT m phone id , m brand , m model ;
S.ram , S.storage , S. battery.
from mobile phones m
Right join.

3 JOIN QUERIES

CREATE TABLES

Create tables Customer (
Cust ID INT PRIMARY KEY ;
Cust NAME VARCHAR (50) NOT NULL ;
);
Create table mobile (

```
mobile ID INT PRIMARY KEY,
Brand VARCHAR (50) NOT NULL,
Mobile VARCHAR (50) NOT NULL,
price Decimal (10,2) CHECK (Price 3000),
);
CREATE TABLE Purchase (
purchase ID INT PRIMARY KEY
CUST ID INT PRIMARY KEY
Mobile ID NOT NULL
Quantity IN CHECK (Quantity 20);
purchase Date Date Default current Date.
);

CREATE Table PAYMENT (
PAYMENT INT PRIMARY KEY;
PURCHASE ID INT UNIQUE;
payment Date Default,
current data.
```

((Right outer) Join):-

```
SELECT m.phone_id , m brand _m.model. S.ram.
  S.storage , S. battery
FROM Mobile phones m
FULL Outer Join phone Specification on m phone
id _ S_phone_ id.
```

3, FULL Outer Join

```
SELECT :m.phone m.brand, m.mobile _S. ram S.storage
    S_battery.
FROM mobile phone m.
FULL Outer Join phone Specification on m phone_
id . phone_id.
```

Equivalent Queries

```
SELECT S mobile name  M. Model name
FROM mobile phone
JOIN Brand ON S.phone ID= M.phone ID;
    using Subquery
SELECT mobile Name,
  (SELECT Brand Name from B rand B.
  where M.phone ID= S.phoneID) AS
      model name   from Mobile phone;
```

Recursive Query (Purchase hierarchy)

```
with Recursive purchases.
SELECT  Payment ID  , phone ID
from  Pre requesties
UNION
SELECT  payment ID  cphone ID
FROM  Preques P
```

VEL TECH - CSE	
EX NO.	05
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	4
VIVA VOCE (5)	4
RECORD (5)	–
TOTAL (20)	13

8/9/25

Result: The implementation of SQL commands using Joins and recursive queries are executed successfully.