TASK 5:- WRITING JOINS QUERIES, EQUIVALENT [illegible]
VE QUERIES

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

Procedure:-
the SQL Joins clause is used to combine records from two or more tables in a database. A join is a means for combining fields from two tables by using values common to each. The join is actually performed by the 'where' clause which combines specified rows of tables,

* Create the database and tables (students, department, courses, enrollments).
* write SQL queries using different types of joins
* ingest sample data,
* Display results and verify correctness,

Syntax:
SELECT column 1, column 2, column 3,.. FROM table-name1, table name 2 WHERE table-name1, column = table-name2 column Name;

types of joins:
1. simple joins
2. self join
3. outer join

* (INNER) Join: Returns records that have matching values in both tables SELECT column-name(s) from table1 INNER Join table2 ON table1.column-name = table2.column-name;
* LEFT (OUTER) JOIN:- Return all records from the left table, and the matched records from the right table,
SELECT column-name(s) FROM table1 [illegible] table2. column-name

* Right (outer) Join: Return all records from the right table, and the matched records from the left table.
SELECT column-name(s) from table1
Right join table2 ON table1.column-name = table2.column-name;
* full (outer) Join:- Return all records when there is a match in either left or right table SELECT column-name = table2.column-name,
full outer join table2 ON table1.column-name = table2.column-name;

1. Join queries (All Types)

CREATE TABLE ~~Departments~~,
Create tables customers(
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 (10,8) CHECK(Price, 39000);
);
CREATE TABLE Payment(
Payment ID INT, Primary key,
Purchase ID INT UNIQUE,
Amount Decimal (10,2) Not Null,
Payment Date Default,
current-Date,

Phone-id	brand	model	Price	ram	Storage	tax
1	Realme	14Pro	30,000	16 4g	256GB	3000
2	Readmi	10Pro	15,000	8GB	128GB	4 5000
				12GB	256GB	8 7000

Right (outer) join: Returnall records from the right table and the matched records from left table.

```
SELECT m.Phone_id, m.brand, m.model, s.ram, s.storage, s.battery
from mobile Phones m
Right Join Phone Specifications on m.Phone_id = s.Phone_id.
```

Phone id	brand	model	Price	ram	storage	battery
1	realme	14Pro	30000	16GB	256GB	5000mah
2	redmi	10Pro	15000	8GB	128GB	4500mah
3	vivo	T3Pro	25000	12gb	256GB	5000mah

Full outer join: Return all records from the match in either left or right table;

```
SELECT m.Phone_id, m.brand, m.model, s.ram, s.storage, s.battery
from mobile Phone's m
full outer join Phone Specification on m.Phone_id, s.Phone_id.
```

Phone_id	brand	model	Price	ram	storage	battery
1	realme	14Pro	30,000	16GB	256GB	5000MAH
2.	redmi	10Pro	15,000	8GB	128GB	9,500MAH
3	vivo	T3Pro	25,000	12GB	256GB	5500MAX

VELTECH	
EX No.	5
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (3)	5
RECORD (4)	—
TOTAL (15)	15
	16/9

Result:-

the implemention of SQL commands using joins and recursive Curies are executed successfully