

Date:- 9/9/21

Task: 5

## WRITING JOIN QUERIES, EQUIVALENT, AND/OR RECURSIVE QUERIES

Aim:- To implement and execute join queries, equivalent queries, and recursive queries using mobile database.

### INNER JOIN:-

Returns records that matching values in both tables

SELECT m.phone-id, m.brand, m.model,  
s.ram, s.storage, s.battery.

FROM mobile phones m

INNER JOIN phone specs

Phone-id	brand	model	Price
1	Realme	10 pro	30,000
2	Redmi	10 pro	15,000
3	Vivo	T3 Pro	25,000

INNER JOIN phone specifications

on m.phone-id = s.phone-id;

phone-id	ram	storage	battery
1	16GB	256GB	5000mAh
2	8GB	128GB	4800mAh
3	12GB	256GB	5500mAh

LEFT (Outer) JOIN: Return all records from the table, and the matched records from the right table.

SELECT m.phone-id, m.brand, m.model,  
s.ram, s.storage, s.battery.

From Mobile Phone s.m

LEFT JOIN phone specifications ON m.phone-id  
= s.phone-id;

phone-id	brand	model	Price
1	realme	14 Pro	30,000
2	Redmi	10 Pro	15,000
3	vivo	T3 Pro	25,000

ram	storage	battery
16GB	256GB	5000mAh
8GB	128GB	4500mAh
12GB	256GB	5500mAh

RIGHT (Outer) JOIN:- Return all records from the right table, and the matched records from the 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	10Pro	30,000	16GB	256GB	5000 MAH
2	Redmi	10Pro	15,000	8GB	128GB	4500 MAH
3	vivo	T3 Pro	25,000	12GB	256GB	5500 MAH

FULL OUTER JOIN:- Return all rewards when there is a match in either left or right table

SELECT :- m.phone-id , m.brand , m.model  
s.ram , s-storage , s.battery

FROM Mobile Phones m

FULL OUTER JOIN Phone specifications

ON m.phone-id = s.phone-id;

phone-id	brand	model	Price	ram	storage	batter
1	realme	10Pro	30,000	16GB	256GB	5000 MAH
2	Redmi	10Pro	15,000	8GB	128GB	4500 MAH
3	vivo	T3 Pro	25,000	12GB	256GB	5500 MAH



## JOIN QUERIES:

### (a) INNER JOIN

```
SELECT m.phone-id, m.brand, m.model,  
s.ram, s.storage, s.battery.  
FROM mobile.phone m
```

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.  
FROM mobile.phone m
```

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.phone m

RIGHT JOIN phone specification

on m.phone-id = s.phone-id;

### d) FULL OUTER JOIN:-

```
SELECT :- m.phone-id, m.brand, m.model  
s.ram, s.storage is battery.
```

FROM mobile phone m

FULL OUTER JOIN phone specifications ON

m.phone\_id = s.phone\_id;

4. Equivalent Queries:

SELECT . Mobile name . model name  
FROM mobile phone

JOIN Brand m1 . phone ID . m . phone ID;

- using subquery.

SELECT Mobile Name,

(SELECT Brand name FROM . B Brand D.

WHERE m.phone ID = s.phone ID) As

Model name.

FROM mobile phone;

5) RECURSIVE QUERY (Purchase Hierarchy)

WITH RECURSIVE purchases Afc

SELECT Payment ID , phone ID

FROM prerequisites.

UNION

SELECT Payment ID , c.phone ID

FROM . Preregister P.  
phone

JOIN Payment Hierarchy 'ON P.phone ID =  
Payment ID

SELECT \* FROM . Payment Hierarchy

VEL TECH	
EX NO.	5
PERFORMANCE (5)	7
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	4
RECORD (5)	1
TOTAL (20)	14
SIGN WITH DATE	

9/9/23

Result:- Thus, the implementation of SQL  
commands using Joins and recursive  
queries are executed successfully.