

09/09/2025

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 Phone m

INNER JOIN Phone Spec R.

phone-id	brand	model	price
1	Realme	14Pro	30,000
2	Redmi	10Pro	15,000
3	vivo	73Pro	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	4500mAh
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_phones m
phone.specifications ON m.phone_id
= s.phone_id;

phone-id	brand	model	price
1	realme	10Pro	30,000
2	Redmi	10Pro	15,000
3	vivo	73Pro	25,000
ram	storage	battery.	
16GB	256GB	5000 mAh	
8GB	128GB	4500 mAh	
12GB	256GB	5500 mAh.	

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.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	TUPRO	30,000	12GB	256GB	5000 mah
2	Redmi	10PRO	15,000	8GB	128GB	4500 mah
3	vivo	T3PRO	25,000	12GB	256GB	5000mah

FULL OUTER JOIN : Return all records 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	battery
1	Realme	TUPRO	30,000	16GB	256GB	5000
2	Redmi	10PRO	15,000	8GB	128GB	4500
3	vivo	T3PRO	25,000	12GB	256GB	5500

3. 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 PhoneSpecification 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 phones.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 phones.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, s.battery.
FROM mobile phones M
FULL OUTER JOIN phone specifications

ON m.phone_id = s.phone_id;

4. Equivalent Queries:

SELECT s.mobile_name, model_name
FROM mobile phone
JOIN Brand ON s.phone_id = m.phone_id;
- using subquery.

SELECT 'mobile name'
 (SELECT brand Name FROM .Brand B,
 WHERE M.Phone ID = S.Phone ID) AS
 model Name FROM mobile phone ;

5) Recursive QUERY (Purchase Hierarchy)

WITH RECURSIVE purchase AS
 SELECT payment ID, phone ID
 FROM Phone
 UNION
 SELECT , payment ID, c.phone ID
 FROM . prerequisites p.
 JOIN Payment hierarchy ON P.Phone ID =
 payment ID

) SELECT * FROM , payment hierarchy.

VEL TECH	
EX NO.	5
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	5
RECORD (5)	-
TOTAL (20)	15

Result: Thus, the implementation of
 SQL commands using points and recursive
 Queries are executed successfully.