09/09/85 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

AInner Joins

Returns records that matching values in both tables SELECT m-Phone-id m.brand, m-moded, s. ram, S-storage, s-buttery, FROM mobiles m

DIN, Phone	specify	1
brand	model	Price
Realma	14Pro	30,000
Redmi	10 Pro	15,000
vivo	T3 Pro	25,000
	brand Realma Red mi	Realma 14Pro Red mi 10Pro

INNER JOIN Phone specifications on milhone-id

Phone-id	ram	storage	battery
,	166B	2566B	5000 MAH
2	8 GB	128GB	4500MAH
3	12GB	25668	5500 MAH

LEFT louter Joins: Return all records from the table and the matched records from the right table select m. phone-id, m. brand, m-model, 3-rum s. storage, s. battery

FROM mobile Phones m

LEFT JOIN Phone specifications on m-Phone-id = siPhone-id

Phone-	īd	bra	nd	maa	lel	Pric	e	raw		store		bothey	
1		Real +	ne	141	ro !	300a	0	160	6	256	48	30000A	
2		Redn	nī	101	ro	15,00	00	86	B	1286	В	4500mA	h
3		vivo	,	Tal	Pro	2500	00	12	GB	2569	В	\$500 m	4))
RIGHT (	RIGHT (outer) 50m) = Returnal records from the												
right table and the matched records from left table													
SELECT m-Phone-id, m. brand m. model s. 89m,													
	sistorage, sibattery							1					
Door	Many mal la show so may												
RIGHT	RIGHT JOIN Phone specifications on m- phone id =												
signone ia;													
Phoneid	60	and	mode	1 1	Price	· Y	um		stor	_	_	itten	
1	rea	Ime	H Pro		30000		1691	3		4 B		000ma	
2		lmi	10800		15000	,	86	В	128	GB	4	500 mA	+)
3		ivo	T3Pro	,	.2500	00	120	B	25	6GB	5	SSOOMA	H
am SEL	atc	ER John	eithe Phone e_s	-id -ba	left ,m- ttery	or bra	Rig	ht	tab	ler		ere is	
from mobile thones m													
FULL OUTER JOIN Phone specification on m. Phone.id.													
s. Phone-id;													
Phone-	id	brand	mo	del	frie	ce	ray	n s	to	rage	16	alte	ry
-1		realme	2 14	Pro	30	000	76	aB	25	64B	1	5000MA	H
2		redm	ė is	Pro	15	000	8	GB	1	18GB	1	4500m	АН
3		VIVO	T	3/0	25	000	1	268	2	56 GB		5500n	TAH

```
JOIN Queriel
 CREATE TABLES
   create tables customer (
  cust ID INT PRIMARY KEY:
  CUSTNAME VARCHE (SD) NOT NULL;
 create table mobile (
 mobile ID INT PRIMARY KEY:
 Brand NAREHR (50) NOT NUll;
 Mobile VARCHAR (50) NOT. NULL;
 Price · Decimal (10,0) · CHECK · [Price : 30,000);
 ):
 CREATE TABLE Purchase (
 Purchase ID INT PRIMARY REY
  cust 10 NOT NUll ;
 Mobile 10 NOT NULL
 Quantity INT CHECK (Quantity 20);
  Purchase Date Date Defualt · Current DATE;
  FORIEGN KEY (cust 10);
  REFERENCES · Mobiles (Mobile 10)
  CREATE TABLE Payment(
    Payment ID INT PRIMARY KEY,
    Purchase 10 INT UNIQUE
     Amount Decimal (1952) Not Null;
    Payment Date Default.
     current - Date,
```

```
Payment method VARCHAR. (20)
  CHECK (layment method INT . NEthanking; (100));
   fortegen key (purchase 10)
  Reference Purchases · (purchase 10) ·
INSERT SAMPLE DATA
 insert into mobile values ( Andred item);
   (101, 'Real-me'.).
   (102, 'Redmi);
   (103, 'vivo');
insert into Mobile value payment values
     (1, 'Realme', 101).
     (2, 'Redmi', 102);
     (3, Vivo', 101).
     (4, 'POCO', 103);
     15, (19 00', 104); -- invalid Phone ID for outersoin
       Example
Insert into Review values;
     ('ci'; Database system; (01);
     ( (2', Good Product worthit toths
     (cg', Product it's good 102);
      (c4, afford to buy it' 103);
```

Insert INTO Payment values (30000, 15000, 25000, 2025-08-79) I Row ( completed); Result: Record inserted successfully 3. JOIN QUERIES a) INNER JOIN SELECT m. Phone -id, m-brand, m-model + s-ram, s-Storage ·S-battery FROM Mobile Phone to inner soin Phone sperification on m. Phone-id = s. Phoneid. 6) LEFTJOIN SELECT mphone-id in brand, m-model, s-ram, s-storage 5-battery From Mobile Phones m reft Join Phone specification on m-phone-id = s-Phone id. C) RIGHT JOIN SELECT m. Phople id , m. brand ; m-model ; s. ram s. storage, s. battery From mobile phones m

From mobile phones m

RIGHT JOIN phone specification

on m phone-id = 5-phone-id;

4) Full QUTER JOIN :-SELECT ·m phone-id ? m brand, m-model, s-ram, Shtorage , sbattery From Mobile Phones M full outer join . Phone specifications on mphone-id = sphone-id; SELECT S. Mobile Name H. Model Name From Mobile Phone Equivalent queries From Mobile Phone JOIN Brand ON SPHONE ID = M. Phone ID; using subquery SELECT Mobile Name; (SELECT Brand Name From Brand B. where M. Phone 10 = S. Phone. 10) . As model wome from Mobile Phone; Recursive Query (purchase hierarchy) with Recursive Purchases SELECT Payment IP, Phone 10 From Prerequestres UNION SELECT 1. Payment 1D. cphone 10 FROM Prereques P

Join Payment Hierarchy on P. Phone ID = Payment ID

SELECT \* FROM Payment Hierarchy

VEL TECH	\$7
EX NO.	)
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	2
RECORD (5)	177
TOTAL (20)	15
SIGN WITH DATE	14

Result: The implementation of SQL commands 1019 using joins and recursive queres are executed successfully