9/9/25 Task-5

writing Join queries, Equivalent. AVOIDR RelwCes. queries

Aim:- To implement & Execute Join queries Equvalent queries.
& receraices queries using Hospital rangement database.

Inner Join:-
Returns records that matching values in both tables Select patient_id, patient name, patient bill, From patients.

Inner Join patient Specifications

Patient_id	patient name	patient_Bill	patient_address
1	Vinod	30,000	rajam
2	Virat	40,000	kadapa
3	Vikram	20,000	vellore.
4	Sriram	50,000	Vizag

INNER JOINT patient Specifications.
ON. m. patient_id = s. patient_id.

left Coota Join: Return all records from the table, & the. matched records from the right table.

Select m. patient_id, m. patient_name, m. patient_bill, patient. address, patient. phone. no.
Left Join patient Specification. ON. m. patient_id = s. patient_id.

patient id	patient name.	patient_Bill	patient address
1	vinod	30,000	rajam
2	virat	40,000	kadapa
3	vikram	20,000	vellore

Right (outer) Join: Return all records from the right table, & the matched records, from the left table

Select m. patient_id, m.patient_name; m. patient_name, patient_bill, patient address:

From patients

Right join patients Specifications.

on m.patient_id = spatient_id;"

Patient Id	Patient_name	patient_bill	patient_address	patient_phno
1	vinod	30,000	Rajam	7992892.
2	virat	40,000	kadapa	6634321
3	vikram	20,000	nellore	99897694.

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 From;

S. stage; S battery.

From patients.

Full outer Join patient doctor Specialize ators

m. patient_id = s. patient_id;

patient_id	patient	bill	patient_address	patient ph.no
1	vinod	30,000	rajan	77992892
2	vivek	40,000	kadapa	6634321
3	Vikram	20,000	nellore	99697694.

JOIN Queries;

Create Table.;

Create table patient (

Patient ID INT. PRIMARY KEY;

patient name, VARCHAR (50) NOT NULL);

Specialization VARCHAR [50] Not NULL;
Salary INT [50] NOT NULL;
);

Create table medicine (
medicen ID INT primary key;
medicine brand VARCHAR [50] NOT NULL;
medine name VARCHAR [50] NOT NULL)
Quantity INT Check (Qeauntity 20);
Purchase date; Date defat Current date;
FoReign key (patient ID);
Reference medicines (medicine ID)
);

Create Table payment.
Payment ID int Primary key;
Purchase ID Int Unique;
Amount Decimal (10-1) NOT NULL;
Payment date default;
Current - Date;
Payment method VARCHAR (20)
CHECK (payment method IN 'ID' Net banking 100);
Foreign key (perchase ID);
Reference purchase (purchase ID)
);

2) INSERT SAMPLE DATA

Insert into patient values (Doctor names)

(101, DiaBatis);

(102, 'B.P');

(103, Cancer');

Insert into patient value payment values

(1, 'Diabatics', 01);

(2, 'B.P', 102);

(3, 'Cancer', 103);

(4, 'Sugar', 103);

(5, malaria, 104);

- involed patient ID, for valter Join En.

Insert INTO Review value.

(`C1: 'Database system: 101');

(`C2: Good product & worth it; 101);

(C3: product is 'good; 102');

(C31: afford to by it '103');

Insert into value (30,000, 15,600, 25000, 2025-08-19)

1 ROW (readed Completed)

Result:- Reward inserted Sucessfully.

3) JOIN queries :-

a) Inner Join :-

select · patient id ; patient name, patient bill, address.

From patient.

Inner Join patient Specification on patient -id, patient-name,

b) left Join :-

select patient -id, patient -name, patient address, patient -bill, patient -age.

From patient.

left Join patient. Specification on patient -id, patient -name.

c) Right Join :-

Select patient -id, patient -name, patient -bill, patient address.

From patient.

d) Full outer Join :-

Select : patient-id, patient-name, patient-address, patient-bill

From patient.p

Full outer Join patient Specification 'ON'.

P. Patient -id = s. Patient -id ;

4) Equivalent queries :-

Select :- patient name, medicen model name from medicine.

drand -ID ; patient =ID ; m. edicent ID.

using subsequary

select medicine name.

5) Recursive query (Purchase)

```
With recursive purchase ID
Select payement ID, patient ID.
    From prerequisites.
UNION
Select , payement ID, patient ID
From pre required p.
Join payment. abolidy on patient ID = patient ID.
)
Select * From. payment
```

VEL TECH	
EX NO.	5
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	5
VIVA VOCE (5)	0
RECORD (5)	10
TOTAL (20)	20
SIGN WITH DATE	

23/9

Result:- Thus, the implementation of Sql Command using Joins & recursive queries are executed successfully.