

Task 5: writing Join Queries, equivalent AND/OR recursive queries.

Aim: To implement and execute Join queries equivalent queries and recursive queries using Hospital management Data Base

Inner Join:

=
returns records that matching values in Both tables select Patient-id, Patient name, Patient-Bill.

From Patient SS.

Inner Join Patient specifications

Patient id	Patient Name	Patient Bill	Patient address
1	Vinod	30,000	XATAM
2	Vishal	40,000	Kadapa
3	Vikram	20,000	Nellore
4	Sriram	80,000	Vizag.

INNER JOIN Patient specifications

ON. m. Patient - id = s. Patient - id

Left outer Join: Return all records from the left table & the matched records from the right table

Select m. Patient - id, m. Patient - name, m. Patient - bill, Patient - address, Patient - phone no.
From Patients

Left Join Patients application ON m. Patient - id
= s. Patient - id

Patient - id.	Patient name	Patient Bill	Patient address	Pat
1	Vinod	30,000	Satam	12
2	Virat	40,000	Kadapa	76
3	Vikram	20,000	Nellore	46
4	Sriram	80,000	Nizag	46

Right JOIN return all records from the right table, and the matched record from the left table select m. patient - id, m. patient - name, patient - bill, patient address

From patients

Right Join patient information

Patient - id	Patient name	Patient Bill	Patient address
1	Vinod	30,000	Satam
2	Virat	40,000	Kadapa
3	Vikram	20,000	Nellore
4	Sriram	80,000	Nizag

Full outer Join: Return all records when there is a match in either left or right table

Select m.patient-id, Patient name, Patient Bill, Patient address, Patient phone no, Patient admitted Date,

from Patients

Full outer Join. Patient Information m.patient-id
g. patient-id;

Patient-id	Patient name	Patient Bill	Patient address	Phone	Date
1	Vinod	30,000	Satam	1234567	15-7-25
2	Viral	40,000	Wadala	7654321	16-7-25
3	Vikram	20,000	Nellore	9666792	20-2-25
4	Sitam	80,000	Vizag	9818867	21-2-25

1. Join Queries

Create tables

Create Table Patient C.

Patient-id INT (Primary Key)

Patient name varchar (50) not null

3 Create table Doctor C.

Patient Id int Primary Key;

Patient name varchar (50) not null;

Phone varchar(50) not null;

Specialization varchar[50] NOT NULL;

salary INT [50] NOT NULL;

);

Create table medicine

medicine-ID INT Primary Key;

medicine name varchar[50] not null;

Quantity INT check (Quantity > 0);

Purchase date Date Default current Date;

Foreign key (Patient ID);

Reference medicines (medicine ID)

);

Create Table Payment

Payment ID int Primary Key;

Purchase ID INT unique;

Amount Decimal (10,2) NOT null;

Payment date Default

current - Date;

Payment method Varchar(20)

Check (Payment method INT 'ID' net banking 100);

Foreign key (Purchase ID);

Reference Purchase (Purchase ID)

);

3 Insert sample Data:

Insert into patient values('Diseases name');

(101, 'Diabetes');

(102, 'B.P');

(103, 'cancer');

Insert into patient value payment values

(1, 'Diabetes', 101);

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

(3, 'cancer', 101);

(4, 'sugars', 103);

(5, 'malaria', 101);

involved patient ID for join example

Insert into review values

('c₁', 'Database system: 101');

'c₂', 'good product & working 101');

('c₃', 'product its good; -102');

('c₄', 'afford to buy it -103');

Insert into value (30,000, 15,000, 25,000, 2025-08-19)

180W (each completed)

Result: Reward invested successfully.

3) Join Queries:

a) Inner Join:

select: Patient-id, Patient-name, Patient-bill, address
from Patient

Inner Join Patient specifications 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

d) Full Outer Join:

select: Patient-id, Patient-name, Patient-address,
Patient-bill from Patient P

Full Outer Join Patient specification "ON"
 $P.Patient-id = S.Patient-id;$

ii) equivalent queries:

Select: Patient name, medicine model name from
medicine brand-ID; patient ID, m.Patient ID
using subquery

Select medicine name@

5) Recursive query (Purchase)

with recursive purchase ID.

Select payment ID, Patient ID

from

Select, Payment ID, Patient ID

from PSC requests.

Join Payment ability on patient ID = Patient - ID

1) Select * from Payment history.

Select * from Payment history.

VEL TECH	
EX NO.	6
PERFORMANCE (5)	5
RESULT AND ANALYSE'S (5)	5
VIVA VOCE (5)	5
RECORD (5)	-
TOTAL (20)	14

SIGN WITH DATE

R
9/5/2023

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