Taskno: 5
Date: 9|9|25

Writing Join Queries, Equivalent AND|OR Recursive Queries.

Aim:

To implement and excute JOIN queries, equivalent queries and recursive queries.

Type of Joins in SQL:

* inner join: Return records Matching Values in both tables.

SELECT Column_name(s)
from table1
Inner join table2 on table1.Column.name = table2.Column_name

* left (Outer) Join: Return all records from the left table and the matched records from the right table

SELECT Column_name(s) FROM table1 LEFT JOIN table2.
ON table 1. Column-name = table2. Column-name.

* Right (outer) join: Return all records from the right table and the matched records from the right table and the matched record from the left table Select.

SELECT Column_name FROM table1 RIGHT JOIN
table2 ON table1. column_name = table2. Column_name

* Full (outer) Join: Returns all records when there is a match in either left or right table. Syntax:

SELECT Column_name From table1 Full Outer
JOIN table1. Column_name = table 2. Column_n.

1. JOIN Queries (All Types..

Create tables

Create table User (
    Userid int primary key , Phone Varchar(15),
    Name Varchar (15) Not null , E-mail Varchar(100)
    Unique Not null , Password Varchar (100) Not null,
    Address Varchar(100));

Create table products :
    product ID int primary key, Name Varchar (50)
    Not Null , Description Varchar (50), Image
    URL Varchar (50), Price int not null , Stock
    int not null , Qty int , Discount int, Category
    ID int, foreign key (category ID) References Categories
    ); (Category ID)

Create table Category (
    Category ID int primary key
    Category Name Varchar (50) Not null);

Insert into sample data:

Insert into User Values ((111,'9574459494', 'Sanju','Sanju
@gmail.com,'password#7', 'Chennai');
(112, '7534001234', 'Jax', jax@gmail.com,'jack#7', pudukkottai),
(113, '8612345632', 'Adithya', Adi@gmail.com, 'adi#8', trandy),
(114, '7532465193', 'Anas', Anas@gmail.com, 'Liki#9', Padi),
(115, '43257977', 'Vino', Queenie@gmail.com, 'Q#7', Chennai');

Insert into products values ((711, 'Cricket Bat,' Branded Decathlon
bat', 'bat.jpg' ,2500, 10, 1, 3, 1), (712, 'Ball', 'Nivi', 'ball.jpg'
, 50, 100, 3, 1) , (713,'shirts', 'van', 'shirt.jpg', 2000, 15, 2, 3, 2),(714, 'Television',
'LG', TV.jpg', 5000, 10, 3, 3)(715, 'AC', 'LG', 'AC.jpg', '30000', 25, 3, 3));

Invest into categories values ((1, 'sports items'), (2, fashion'), (3, Home Appliances') (4, hadgets'));

3. JOIN Queries:

Inner Join:

SELECT Product ID, Name, Categories. Category Name, FROM Products Innerjoin Categories on Products-Category ID = Categories Category ID;

Output:

Product ID	Name	Category Name
711	Cricket Bat	Sports items
712	Ball	Sports items
713	Shirts	fashion
714	Television	Home Application
715	Ac	Home Application

Left Outer Join:

SELECT Product ID, Name, Categories, Category Namefrom Products Left outer join Categories on Product on Product Category ID Categories - Category ID;

Output:

Product ID	Name	Category Name
711	Cricket Bat	Sports items
712	Ball	Sports items
713	shirts	fashion
714	Television	Home Appliances
715	Ac	Home Appliances.

Right Outer Join:
SELECT Product ID, Name, Catgories. Category Name FROM Products Right outer join categories on Products. Category ID = Category. Category ID);

output:

Product ID	Name	Category Name
711	Cricket Bat	Sports item.
712	Ball	Sports item
713	Shirts	Fashion
714	Television	Home Appliance
715	Ac	Home Applience
NULL		

full Outer join: SELECT Product ID, Name, Catgories, Category name From Product Full outer join Categories on product Category ID = Category. Ctegry ID);
output:

Product ID	Name	Category Name
711	Cricket Bat	Sports items
712	Ball	Sports item
713	Shirts	fashion
714	Television	Home Appliance
715	Ac	Home
NULL	Null	Gadgets

VEL TECH

EX NO.	5
PERFORMANCE (5)	5
RESULT AND ANALYSIS (5)	3
VIVA VOCE (5)	1
RECORD (5)	
TOTAL (20)	14
SIGN WITH DATE	9/8/25

Result:
Thus the implementation JOIN queries and recursive have was executed Successfully.