Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology
(Deemed to be University Estd. u/s 3 of UGC Act, 1956) School of Computing

B.Tech. – Information Technology

Academic Year: 2025– VTR UGE2021- (CBCS) 2026

SUMMER SEMESTER - SS2526

Course Code : 10211IT201

Course Name : Database System Concept

Slot No : S12L5

DBMS TASK - 5 REPORT

TASK:-Implementation of different types of Joins

Submitted by:

VTUNO	REGISTER NUMBER	STUDENT NAME
VTU29010	24UEIT0040	P.Manikanta

ABSTRACT

The Implementation of Different Types of Joins project focuses on understanding and applying various SQL join operations to retrieve meaningful data from multiple related tables. Join operations enable combining data from different entities based on logical relationships between their attributes, thereby improving query optimization and data analysis.

In this task, different types of joins such as Simple Join, Equi Join, NonEqui Join, Self Join, and Outer Joins (Inner, Left, Right, and Full) are implemented. Each join demonstrates how data from multiple tables can be combined to produce comprehensive results—whether to retrieve matching records, unmatched records, or hierarchical relationships.

By performing and analyzing these join operations, this experiment enhances understanding of query optimization techniques and demonstrates how SQL joins play a crucial role in building efficient and relationally sound database systems.

Aim: To Perform the advanced query processing and test its heuristics using the designing of optimizing complex queries and their equivalence queries.

Procedures:

SQL JOIN

A JOIN clause is used to combine rows from two or more tables, based on a related column between them.

Types of JOIN

1. Simple Join
   1. Equi-join :
   2. NonEqui-join :
2. Self-Join Query
3. Outer Join
   1. Inner Join
   2. Left Outer Join
   3. Right Outer Join
   4. Full Outer Join

The join is actually performed by the 'where' clause which combines specified rows of tables.

Simple Join:

It is the most common type of join. It retrieves the rows from 2 tables having a common column and is further classified into 2 joins.

1. Equi-join :

A join, which is based on equalities, is called equi-join.

2. Non Equi-join:

It specifies the relationship between columns belonging to different tables by making use of relational operators other than'='.

Self join:

Joining of a table to itself is known as self-join. It joins one row in a table to another. It can compare each row of the table to itself and also with other rows of the same table.Outer Join:

▪ It extends the result of a simple join. An outer join returns all the rows returned by simplejoin as well as those rows from one table that do not match any row from the table.

▪ The symbol(+) represents outer join.

Outer Join

- Inner Join
- Left Join
- Right Join
- Full Outer Join
- (INNER) JOIN: Returns records that have matching values in both tables
- LEFT (OUTER) JOIN: Returns all records from the left table, and the matched records from the right table
- RIGHT (OUTER) JOIN: Returns all records from the right table, and the matched records from the left table
- FULL (OUTER) JOIN: Returns all records when there is a match in either left or right table

Table Creation Queries

1. CricketBoard Table

```
CREATE TABLE CricketBoard (
    BoardID NUMBER PRIMARY KEY,
    BoardName VARCHAR2(100) NOT NULL,
    Headquarters VARCHAR2(100)
);
```

SQL> desc cricketboard;

Name	Null?	Type
BOARDID	NOT NULL	NUMBER
BOARDNAME	NOT NULL	VARCHAR2(100)
HEADQUARTERS		VARCHAR2(100)

2. Team Table

```
CREATE TABLE Team (
    TeamID NUMBER PRIMARY KEY,
    TeamName VARCHAR2(100) NOT NULL,
    BoardID NUMBER REFERENCES CricketBoard(BoardID),
    CaptainID NUMBER, -- Will reference Player table after creation
    CoachName VARCHAR2(100)
);
```

SQL> desc team;

Name	Null?	Type
TEAMID	NOT NULL	NUMBER
TEAMNAME	NOT NULL	VARCHAR2(100)
BOARDID		NUMBER
CAPTAINID		NUMBER
COACHNAME		VARCHAR2(100)

3. Player Table

```
CREATE TABLE Player (
    PlayerID NUMBER PRIMARY KEY,
    PlayerName VARCHAR2(100) NOT NULL,
    Role VARCHAR2(50),
    TeamID NUMBER REFERENCES Team(TeamID)
```

);

SQL> desc player;

Name	Null?	Type
PLAYERID	NOT NULL	NUMBER
PLAYERNAME	NOT NULL	VARCHAR2(100)
ROLE		VARCHAR2(50)
TEAMID		NUMBER

After creating Player, we can now add a foreign key for CaptainID in Team to refer to Player.

```
ALTER TABLE Team
ADD CONSTRAINT fk_captain FOREIGN KEY (CaptainID) REFERENCES
Player(PlayerID);
```

SQL> desc team;

Name	Null?	Type
TEAMID	NOT NULL	NUMBER
TEAMNAME	NOT NULL	VARCHAR2(100)
BOARDID		NUMBER
CAPTAINID		NUMBER
COACHNAME		VARCHAR2(100)

4. Match Table

```
CREATE TABLE Match (
   MatchID NUMBER PRIMARY KEY,
   Team1ID NUMBER REFERENCES Team(TeamID), Team2ID
   NUMBER REFERENCES Team(TeamID),
   MatchDate DATE,
   Location VARCHAR2(100)
);
```

SQL> desc match;

Name	Null?	Type
MATCHID	NOT NULL	NUMBER
TEAM1ID		NUMBER
TEAM2ID		NUMBER
MATCHDATE		DATE
LOCATION		VARCHAR2(100)

5. Match_Player Table

```
CREATE TABLE Match_Player (
   MatchID NUMBER REFERENCES Match(MatchID),
   PlayerID NUMBER REFERENCES Player(PlayerID),
   PRIMARY KEY (MatchID, PlayerID)
);
```

SQL> desc match_player;

Name	Null?	Type
MATCHID	NOT NULL	NUMBER
PLAYERID	NOT NULL	NUMBER

INSERT QUERIES:-

1. Insert into CricketBoard

```
INSERT INTO CricketBoard VALUES (1, 'Board of Control for Cricket in India',
'Mumbai');
INSERT INTO CricketBoard VALUES (2, 'Australian Cricket Board', 'Sydney'); INSERT
INTO CricketBoard VALUES (3, 'England and Wales Cricket Board',
'London');
```

SQL> SELECT*FROM CRICKETBOARD;

BOARDID

BOARDNAME

HEADQUARTERS

1
Board of Control for Cricket in India
Mumbai

2
Australian Cricket Board
Sydney

3
England and Wales Cricket Board
London

2. Insert into Team
INSERT INTO Team VALUES (101, 'Chennai Super Kings', 1, NULL, 'Stephen Fleming');
INSERT INTO Team VALUES (102, 'Mumbai Indians', 1, NULL, 'Mark Boucher');
INSERT INTO Team VALUES (103, 'Nellai Royal Kings', 1, NULL, 'S. Badrinath');
INSERT INTO Team VALUES (201, 'Sydney Sixers', 2, NULL, 'Greg Shipperd');
INSERT INTO Team VALUES (301, 'London Lions', 3, NULL, 'Trevor Bayliss');

SQL> SELECT*FROM CRICKETBOARD;

BOARDID

BOARDNAME

HEADQUARTERS

1

Board of Control for Cricket in India

Mumbai

2

Australian Cricket Board

Sydney

BOARDID

BOARDNAME

HEADQUARTERS

3

England and Wales Cricket Board

London

SQL> SELECT*FROM TEAM;

```
    TEAMID
----------
TEAMNAME
--------------------------------------------------------------------------------
  BOARDID CAPTAINID
---------- ----------
COACHNAME
--------------------------------------------------------------------------------
       101
Chennai Super Kings
         1       1001
Stephen Fleming

       102
Mumbai Indians
         1       1004
Mark Boucher


       103
Nellai Royal Kings
         1       1006
S. Badrinath


       201
Sydney Sixers
         2       1008
Greg Shipperd

       301
```

London Lions

3 1010

Trevor Bayliss

3. Insert into Player

```
INSERT INTO Player VALUES (1001, 'MS Dhoni', 'Captain/WK', 101);
INSERT INTO Player VALUES (1002, 'Ruturaj Gaikwad', 'Batsman', 101);
INSERT INTO Player VALUES (1003, 'Ravindra Jadeja', 'All-rounder', 101);

INSERT INTO Player VALUES (1004, 'Rohit Sharma', 'Captain/Batsman', 102);
INSERT INTO Player VALUES (1005, 'Jasprit Bumrah', 'Bowler', 102);

INSERT INTO Player VALUES (1006, 'Baba Aparajith', 'Captain/Batsman', 103);
INSERT INTO Player VALUES (1007, 'Sanjay Yadav', 'All-rounder', 103);

INSERT INTO Player VALUES (1008, 'Moises Henriques', 'Captain/All-rounder', 201);
INSERT INTO Player VALUES (1009, 'Josh Philippe', 'Wicketkeeper', 201);

INSERT INTO Player VALUES (1010, 'Joe Root', 'Captain/Batsman', 301);
INSERT INTO Player VALUES (1011, 'Ben Stokes', 'All-rounder', 301);

SQL> SELECT *FROM PLAYER;

  PLAYERID
----------
PLAYERNAME
--------------------------------------------------------------------------------
ROLE                                               TEAMID
-------------------------------------------------- ----------
     1001
MS Dhoni
Captain/WK                                            101
```

1002

Ruturaj Gaikwad

Batsman 101

1003

Ravindra Jadeja

All-rounder

101

1004

Rohit Sharma

Captain/Batsman

102

1005

Jasprit Bumrah

Bowler

102

1006

Baba Aparajith

Captain/Batsman

103

1007

Sanjay Yadav

All-rounder

103

1008

Moises Henriques

Captain/All-rounder

201

1009

Josh Philippe

Wicketkeeper

201

1010
Joe Root
Captain/Batsman 301

1011
Ben Stokes

All-rounder 301

4. Insert into Match

```
INSERT INTO Match VALUES (501, 101, 102, TO_DATE('2024-05-01', 'YYYY-MM-DD'), 'Chennai');
INSERT INTO Match VALUES (502, 103, 101, TO_DATE('2024-05-05', 'YYYY-MM-DD'), 'Coimbatore');
INSERT INTO Match VALUES (503, 201, 301, TO_DATE('2024-05-10', 'YYYY-MM-DD'), 'Sydney');
INSERT INTO Match VALUES (504, 102, 201, TO_DATE('2024-05-15', 'YYYY-MM-DD'), 'Mumbai');
```

```
SQL> SELECT*FROM MATCH;

   MATCHID    TEAM1ID    TEAM2ID MATCHDATE
---------- ---------- ---------- ---------
LOCATION
--------------------------------------------------------------------------------
       501        101        102 01-MAY-24
Chennai

       502        103        101 05-MAY-24
```

Coimbatore

503 201 301 10-MAY-24
Sydney

504 102 201 15-MAY-24
Mumbai

5. Insert into Match_Player

-- Match 501: CSK vs MI

INSERT INTO Match_Player VALUES (501, 1001);

INSERT INTO Match_Player VALUES (501, 1004);

INSERT INTO Match_Player VALUES (501, 1002); INSERT INTO Match_Player VALUES (501, 1005);

-- Match 502: NRK vs CSK

INSERT INTO Match_Player VALUES (502, 1001);

INSERT INTO Match_Player VALUES (502, 1002);

INSERT INTO Match_Player VALUES (502, 1006);

INSERT INTO Match_Player VALUES (502, 1007);

-- Match 503: Sydney Sixers vs London Lions

INSERT INTO Match_Player VALUES (503, 1008);

INSERT INTO Match_Player VALUES (503, 1010);

INSERT INTO Match_Player VALUES (503, 1009); INSERT INTO Match_Player VALUES (503, 1011);

-- Match 504: MI vs Sydney Sixers

INSERT INTO Match_Player VALUES (504, 1004);

INSERT INTO Match_Player VALUES (504, 1008);

INSERT INTO Match_Player VALUES (504, 1005);

INSERT INTO Match_Player VALUES (504, 1009);

SQL> SELECT*FROM MATCH_PLAYER;

MATCHID	PLAYERID
501	1001
501	1004
501	1002
501	1005
502	1001
502	1002
502	1006
502	1007
503	1008
503	1010
503	1009

MATCHID	PLAYERID
503	1011
504	1004
504	1008
504	1005
504	1009

GIVEN QUESTIONS;

1. Retrieve all cricket boards and their teams

```
SELECT cb.BoardName, t.TeamName
FROM CricketBoard cb
JOIN Team t ON cb.BoardID = t.BoardID;
```

BOARDNAME

TEAMNAME

Board of Control for Cricket in India

Chennai Super Kings

Board of Control for Cricket in India

Mumbai Indians

Board of Control for Cricket in India

Nellai Royal Kings

BOARDNAME

TEAMNAME

Australian Cricket Board

Sydney Sixers

England and Wales Cricket Board

London Lions

2. List all matches along with the teams and their captains

SELECT

```
m.MatchID, t1.TeamName
AS Team1, p1.PlayerName
AS Captain1, t2.TeamName
AS Team2, p2.PlayerName
AS Captain2
```

```
FROM Match m
JOIN Team t1 ON m.Team1ID = t1.TeamID
JOIN Player p1 ON t1.CaptainID = p1.PlayerID
JOIN Team t2 ON m.Team2ID = t2.TeamID
JOIN Player p2 ON t2.CaptainID = p2.PlayerID;
```

```
   MATCHID
----------
TEAM1
--------------------------------------------------------------------------------
CAPTAIN1
--------------------------------------------------------------------------------
TEAM2
--------------------------------------------------------------------------------
CAPTAIN2
--------------------------------------------------------------------------------
       502
Nellai Royal Kings
Baba Aparajith

   MATCHID
----------
TEAM1
--------------------------------------------------------------------------------
CAPTAIN1
--------------------------------------------------------------------------------
TEAM2
--------------------------------------------------------------------------------
CAPTAIN2

Chennai Super Kings
```

MS Dhoni

MATCHID

TEAM1

CAPTAIN1

TEAM2

CAPTAIN2

501
Chennai Super Kings
MS Dhoni

MATCHID

TEAM1

CAPTAIN1

TEAM2

CAPTAIN2

Mumbai Indians
Rohit Sharma

MATCHID

TEAM1

CAPTAIN1

TEAM2

CAPTAIN2

504

Mumbai Indians

Rohit Sharma

MATCHID

TEAM1

CAPTAIN1

TEAM2

CAPTAIN2

Sydney Sixers

Moises Henriques

MATCHID

TEAM1

CAPTAIN1

TEAM2

CAPTAIN2

503

Sydney Sixers

Moises Henriques

MATCHID

TEAM1

CAPTAIN1

TEAM2

CAPTAIN2

London Lions

Joe Root

3. Count the number of matches played by each team

```
SELECT
   t.TeamName,
   COUNT(m.MatchID) AS MatchesPlayed
FROM Team t
JOIN Match m ON t.TeamID = m.Team1ID OR t.TeamID = m.Team2ID
GROUP BY t.TeamName;
```

TEAMNAME

MATCHESPLAYED

Chennai Super Kings

2

Mumbai Indians

2

Nellai Royal Kings

1

TEAMNAME

MATCHESPLAYED

Sydney Sixers

2

London Lions

1

4. Find all the players who are part of the team named ‘Nellai Royal Kings’

```
SELECT p.PlayerName
FROM Player p
JOIN Team t ON p.TeamID = t.TeamID
WHERE t.TeamName = 'Nellai Royal Kings';
```

PLAYERNAME

Baba Aparajith

Sanjay Yadav

4. Find all the matches played by a specific player (example: ‘MS Dhoni’)

If using a Match_Player (junction) table:

```
SELECT DISTINCT m.MatchID, m.MatchDate, m.Location
FROM Match m
JOIN Match_Player mp ON m.MatchID = mp.MatchID
JOIN Player p ON mp.PlayerID = p.PlayerID
WHERE p.PlayerName = 'MS Dhoni';
```

```
   MATCHID MATCHDATE
---------- ---------
LOCATION
--------------------------------------------------------------------------------
      501 01-MAY-24
Chennai

      502 05-MAY-24
Coimbatore
```

RESULT:- THE QUERIES ARE IMPLEMENTED SUCCESSFULLY
