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

VTR UGE2021- (CBCS)

Academic Year: 2025–2026

SUMMER SEMESTER - SS2526

Course Code : 10211IT201

Course Name : Database System concepts

Slot No : S12L5

DBMS TASK - 7 REPORT

Submitted by:

VTUNO	REGISTER NUMBER	STUDENT NAME
VTU28684	24UEIT0058	YEKAMBARAM NANDHA KISHORE

ABSTRACT

The *Trigger and View Implementation in PL/SQL* project demonstrates the automation of data integrity and dynamic updates within a relational database system. Triggers in PL/SQL are powerful mechanisms that automatically execute predefined actions in response to specific table events such as *INSERT*, *UPDATE*, or *DELETE*. This experiment focuses on designing a trigger that updates a player's total match count whenever a new match record is inserted, ensuring real-time consistency without manual intervention.

In addition to triggers, a *database view* is created to simplify data access by displaying consolidated information about each player along with the total number of matches played. A *non-recursive PL/SQL procedure* is also implemented to retrieve all even-numbered player IDs using cursor-based iteration and conditional logic.

By integrating these components, the task highlights how PL/SQL enhances database efficiency through automation, abstraction, and procedural logic. The project reinforces core database management concepts such as event-driven execution, cursor handling, and view creation—key techniques for maintaining accuracy and reliability in enterprise data systems.

Aim:

To create a trigger that automatically updates the total number of matches played by each player, a view to display players and their match count, and a procedure to list even-numbered Player IDs.

Step 1: Create Tables

```
-- Player table
CREATE TABLE Player (
    PlayerID VARCHAR2(10) PRIMARY KEY,
    PlayerName VARCHAR2(50),
    TotalMatchesPlayed NUMBER DEFAULT 0
);
```

```
SQL> desc player;
 Name                                      Null?    Type
 PLAYERID                                  NOT NULL VARCHAR2(10)
 PLAYERNAME                                         VARCHAR2(50)
 TOTALMATCHESPLAYED                                 NUMBER
```

```
-- MatchTeam table
CREATE TABLE MatchTeam (
    MatchID VARCHAR2(10),
    PlayerID VARCHAR2(10),
```

```
    CONSTRAINT fk_player FOREIGN KEY (PlayerID)
        REFERENCES Player(PlayerID)
);
SQL> desc matchteam;
 Name                                      Null?    Type
 MATCHID                                            VARCHAR2(10)
 PLAYERID                                           VARCHAR2(10)
```

Step 2: Create Trigger – Auto Update Matches

```
CREATE OR REPLACE TRIGGER trg_update_total_matches
AFTER INSERT ON MatchTeam
FOR EACH ROW
BEGIN
    UPDATE Player
    SET TotalMatchesPlayed = NVL(TotalMatchesPlayed, 0) + 1
    WHERE PlayerID = :NEW.PlayerID;
END;
/
```

Step 3: Create View – Players with Match Count

```
CREATE OR REPLACE VIEW PlayerMatchView AS

SELECT PlayerID, PlayerName, TotalMatchesPlayed

FROM Player;

Step 4: Procedure  -  Retrieve Even-Numbered Player IDs

CREATE OR REPLACE PROCEDURE GetEvenPlayerIDs IS

    CURSOR c1 IS

        SELECT PlayerID

        FROM Player

        WHERE MOD(TO_NUMBER(REGEXP_SUBSTR(PlayerID, '\d+$')), 2) = 0;

BEGIN

    FOR rec IN c1 LOOP

        DBMS_OUTPUT.PUT_LINE('PlayerID: ' || rec.PlayerID);

    END LOOP;

EXCEPTION

    WHEN NO_DATA_FOUND THEN

        DBMS_OUTPUT.PUT_LINE('No even-numbered PlayerIDs found.');

    WHEN OTHERS THEN

        DBMS_OUTPUT.PUT_LINE('Error: ' || SQLERRM);
```

```
END;
/
-- Insert Players
INSERT INTO Player (PlayerID, PlayerName) VALUES ('PID012', 'Rohit');
INSERT INTO Player (PlayerID, PlayerName) VALUES ('PID311', 'Virat');
INSERT INTO Player (PlayerID, PlayerName) VALUES ('PID008', 'Dhoni');
INSERT INTO Player (PlayerID, PlayerName) VALUES ('PID313', 'Gill');

SQL> select*from player;

PLAYERID                                                    PLAYERNAME
TOTALMATCHESPLAYED

PID012                                                          Rohit
0

PID311                                                          Virat
0

PID008                                                          Dhoni
0

PID313                                                           Gill
0
```

```
-- Insert Match Records (Trigger will auto-update TotalMatchesPlayed)

INSERT INTO MatchTeam (MatchID, PlayerID) VALUES ('M001', 'PID012');

INSERT INTO MatchTeam (MatchID, PlayerID) VALUES ('M002', 'PID012');

INSERT INTO MatchTeam (MatchID, PlayerID) VALUES ('M003', 'PID008');

SQL> select*from matchteam;

MATCHID    PLAYERID

M001       PID012

M002       PID012

M003       PID008

COMMIT;
```

RESULT: THE QUERIES ARE EXECUTED SUCCESFULLY