**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. – Computer Science and Engineering**

**VTR UGE2021- (CBCS)**

Academic Year: 2025–2026

SUMMER SEMESTER - SS2526

Course Code : 10211CS207

Course Name : Database Management Systems

Slot No : S2L3

DBMS TASK - 7 REPORT

Title**:** Triggers, Views and Exceptions

Submitted by:

|  |  |  |
| --- | --- | --- |
| **VTUNO** | **REGISTER NUMBER** | **STUDENT NAME** |
| VTU27916 | 24UECS1462 | J.SAI VINEETH KUMAR |

**Triggers, Views and Exceptions**

**Aim:**

To Conduct events, views, exceptions on CRUD operations for restricting phenomenon.

a) To create a trigger in PL/SQL that automatically inserts a new record in the match\_result table when a new record is inserted into the match table.

b) To create a view that displays the details of players along with their team details.

c) To write a non-recursive PL/SQL procedure to retrieve even-numbered PlayerIDs registered for any tournament.

**To create a trigger in PL/SQL that automatically inserts a new record in the match\_result table when a new record is inserted into the match table.**

CREATE OR REPLACE TRIGGER insert\_match\_result

AFTER INSERT ON match

FOR EACH ROW

BEGIN

INSERT INTO match\_result (MatchID, TeamID, Result)

VALUES (:new.MatchID, :new.TeamID, 'Pending'); -- Assuming a default value of 'Pending' for Result

END;

/

**To create a view that displays the details of players along with their team details.**

SQL> CREATE VIEW PlayerTeamDetails AS SELECT p.playerID, p.fname AS PlayerName, p.teamID, p.coach AS PlayerCoach, p.captain AS PlayerCaptain, (SELECT t.name FROM team t WHERE t.teamID = p.teamID) AS TeamName, (SELECT t.coach FROM team t WHERE t.teamID = p.teamID) AS TeamCoach, (SELECT t.captain FROM team t WHERE t.teamID = p.teamID) AS TeamCaptain FROM player p;

SQL> Select \* from PlayerTeamDetails;

**To write a non-recursive PL/SQL procedure to retrieve even-numbered PlayerIDs registered for any tournament.**

CREATE OR REPLACE PROCEDURE GetEvenPlayerIDsForTournament(in\_tournament\_id NUMBER, out\_even\_player\_ids SYS.ODCINUMBERLIST) AS

BEGIN

out\_even\_player\_ids := SYS.ODCINUMBERLIST(); -- Initialize the collection

-- Populate the collection with even-numbered PlayerIDs for the specified tournament

FOR player\_rec IN (SELECT PlayerID FROM Player WHERE TournamentID = in\_tournament\_id AND MOD(PlayerID, 2) = 0) LOOP

out\_even\_player\_ids.EXTEND;

out\_even\_player\_ids(out\_even\_player\_ids.COUNT) := player\_rec.PlayerID;

END LOOP;

END;

/

DECLARE

tournament\_id NUMBER := 123; -- Replace with the desired tournament ID

even\_player\_ids SYS.ODCINUMBERLIST;

BEGIN

GetEvenPlayerIDsForTournament(tournament\_id, even\_player\_ids);

-- You can now use the even\_player\_ids collection as needed.

-- For example, to print the even PlayerIDs:

FOR i IN 1..even\_player\_ids.COUNT LOOP

DBMS\_OUTPUT.PUT\_LINE('Even PlayerID: ' || even\_player\_ids(i));

END LOOP;

END;

/

**Result:**

Thus the Triggers, Views and Exceptions experiment was successfully completed results are verified.