

- 2-9-25 TASK 7: Triggers, Views, and Exception
- Aim: To conduct events, views, exceptions on CRUD operations for restricting phenomena.
- To create trigger in PL/SQL that automatically inserts a new record in match-result table when a new record is inserted into match table.
  - To create view that displays detail of players along with their team details.
  - To write non-recursive PL/SQL procedure retrieve user numbered player IDs registered for match with exception handling.

a. To create trigger in PL/SQL that automatically inserts a new record in match-result table when new record is inserted into match table.

SQL> create TABLE match-Result (Result ID varchar(10) PRIMARY KEY, MATCHID varchar(10), TEAM ID varchar(20), FOREIGN KEY(MATCH ID) REFERENCES MATCH (MATCH ID), FOREIGN KEY (TEAM ID) REFERENCES team(Team ID))

SQL> CREATE OR REPLACE TRIGGER trg-insert-match - result  
AFTER INSERT ON MATCH FOR EACH ROW

DECLARE

N NUMBER;

BEGIN

N:=INSTR(UPPER(:NEW.RESULT), 'WIN');  
IF N>0  
VALUES ('R'||:NEW.MATCHID||'1', :NEW.MATCH ID,  
:NEW.TEAM ID, 'WIN');

INSERT INTO Match Result (RESULT ID, MATCH ID, TEAM ID,  
RESULT)  
VALUES ('R'||:NEW.MATCH ID||'2', :NEW.MATCH ID,  
RESULT)

```

    INSERT INTO Match_Result (ResultID, MatchID, MatchIDB,
    Result)
VALUES ('R1': NEW.MatchID || '1': NEW.MatchID, TeamID,
        Result, 'FAIL');

INSERT INTO Match_Result (ResultID, MatchID, MatchIDB,
    TeamID, Result)
VALUES ('R1' || :NEW.MatchID || '2': NEW.MatchID,
        :NEW.TeamID2, 'WIN');

END IF;
END;

```

SQL > insert into match values ('M01', 'CCB01', 'TCB01',
 '22-JUN-2022', 1.3, 'TCB01 - WIN');

SQL > insert into match values ('M02', 'CCB02', 'TCB02',
 '22-JUN-2022', 8.3, 'CCB01 - WIN');

SQL > insert into match values ('M03', 'TRICB01', '24-JUN-
 2022', 8.3, 'TCB01 - WIN');

SQL > insert into match values ('M04', 'TRICB01', 'TCB02',
 '25-JUN-2022', 8.3, 'TRICB01 - WIN');

SQL > insert into match values ('M05', 'TCB01', 'TCB02', DATE
 '2025-09-20', 16.00, 'TCB02 - WIN');

SQL > Select \* from Match\_Result;

Write the output

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

SQL > CREATE OR REPLACE VIEW Player\_Team\_View AS  
SELECT P.PlayerID, P.FName AS FirstName, P.PlayingRole AS PlayingRole, P.BattingStyle AS BattingStyle, P.Bowling AS BowlingStyle, T.TeamID AS TeamID, T.Name AS TeamName, T.Coach AS Coach, T.Captain AS Captain FROM Player P JOIN Team T ON P.TeamID = T.TeamID;

SQL > Select \* from player\_team\_view;

write the output.

(1000, 'Rohit Sharma') and so on

(1001, 'Virat Kohli', 'Batsman', 'Right Handed', 'Left Arm', 1001, 'India', 'Rishabh Pant', 'Captain');

C. To write a non-recursive PL/SQL procedure to retrieve even-numbered playerID's registered for any match with exception handling.

CREATE OR REPLACE PROCEDURE

Get\_Even\_PlayerIDs FOR MATCHES

CURSOR even\_players IS

SELECT DISTINCT P.TeamID = T.TeamID  
FROM Player P

JOIN Team T ON P.TeamID = T.TeamID

JOIN Match M ON (M.TeamID1 = T.TeamID OR M.TeamID2 = T.TeamID)

WHERE MOD(TO\_NUMBER(REGEXP\_SUBSTR(P.PlayerID, '[^0\$]')), 2) = 0;

V COUNT NUMBER := 0;

BEGIN

DBMS\_OUTPUT.PUT\_LINE('Even-numbered player IDs registered for matches.');

FOR rec IN even\_players LOOP

VCOUNT := VCOUNT + 1

DBMS\_OUTPUT.PUT\_LINE('Match ID' || rec.MatchID)

'| Player ID' || rec.PlayerID ||

'| name: ' || rec.FName || ' ' || rec.LName);

END LOOP;

Handle case where no records found

If v\_count = 0 THEN

RAISE NO-DATA-FOUND;

END IF;

EXCEPTION

WHEN NO-DATA-FOUND THEN

DBMS-OUTPUT.PUT-LINE('No even-numbered  
player IDs found for any match');

when others THEN

DBMS-OUTPUT.PUT-LINE('Error: // SQLERRM');

END;

SQL>SET SERVEROUTPUT ON

SQL>/

Procedure created.

SQL>SET SERVEROUTPUT ON;

SQL>EXEC GET-EVEN-PLAYERIDS-FOR-MATCHES;

Writer of the output:

VEL TECH	
EX NO.	1001
PERFORMANCE (5)	5/5
RESULT AND ANALYSIS (5)	5/5
INNOVATION (5)	5/5
RECORD (5)	5/5
TOTAL (20)	20/20
DATE WITH DATE	10/10/2023

Result: Thus the triggers, views and exceptions experiment was successfully completed, results are verified