

2-9-25 TASK 7: Triggers, Views and Exception

Aim: TO conduct events, views, exceptions on CRUD operations for restricting phenomenon.

- 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 then 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), TeamID Varchar(6),
Result 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
```

```
INSTR(UPPER(:NEW.RESULT), 'WIN');
```

```
VALUES ('R' || :NEW.MatchID || '1', :NEW.MatchID,
:NEW.TeamID, 'WIN');
```

```
INSERT INTO match Result (ResultID, MatchID, TeamID,
Result)
```

```
VALUES ('R' || :NEW.MatchID || '2', :NEW.MatchID,
Result)
```



```
INSERT INTO Match-Result (Result ID, Match ID, Match ID, Result)
```

```
VALUES ('R':NEW.MatchID||'1', :NEW.MatchID, Team ID, Result 'FAIL');
```

```
INSERT INTO Match-Result (Result ID, Match ID, Team ID, Result)
```

```
VALUES ('R'||:NEW.MatchID||'2', :NEW.MatchID, :NEW, TeamID2, 'WIN');
```

```
END IF;
```

```
END;
```

```
SQL>insert into match values ('MO1', 'CCBO1', 'TCBO1', '22-JUN-2022', 1.3, 'TCBO1 - WIN');
```

```
SQL>insert into match values ('MO2', 'CCBO2', 'TCBO2', '22-JUN-2022', 8.3, 'CCBO1 - WIN');
```

```
SQL>insert into match values ('MO3', 'TRIBO1', '24-JUN-2022', 8.3, 'TCBO1 - WIN');
```

```
SQL>insert into match values ('MO4', 'TRIBO1', 'TCBO2', '25-JUN-2022', 8.3, 'TRICBO1 - WIN');
```

```
SQL>insert into match values ('MO5', 'TCBO1', 'TCBO2', DATE '2025-09-20', 16.00, 'TCBO2 - WIN');
```

```
SQL>Select * from Match-Result;
```

Write the output

b.

b. to create view that display 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.playing Role AS
playing Role, P.Batting AS Batting style, P.Bowling AS Bowling style,
t.TeamID AS TeamID, t.Name AS Team Name, t.Coach
AS Coach, t.Captain AS Captain FROM Player P JOIN Team
t ON P.team ID = t.team ID;
```

```
SQL > select * from player-Team-View;
```

write the output.

C. to write non recursive PL/SQL Procedure to retrieve even-numbered player ID's registered for any match with exception handling

```
CREATE OR REPLACE PROCEDURE
Get-Even-Player IDs for Matches IS
CURSOR even-players IS
SELECT DISTINCT P.TeamID = t.team ID
FROM Player P
JOIN Team t ON t.team ID = t.team ID
JOIN Match ON (m.team ID 1 = t.team ID OR m.team ID 2
= t.team ID)
WHERE MOD(TO-NUMBER(REGEXP-SUBSTR(P.PlayerID
'id+$'), 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 = V COUNT + 1
DBMS-OUTPUT.PUT LINE('Match ID' || rec Match ID
'| Player ID' || rec Player ID ||
'| Name: ' || rec FName || ' ' || rec L Name);
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-players-IDs-FOR-matches;

Write the output.

VEL TECH	
EX NO.	
PERFORMANCE (5)	
RESULT AND ANALYSIS (5)	
VIVA VOCE (5)	
RECORD (5)	
TOTAL (20)	
WITH DATE	

Result: Thus the triggers, views and exceptions

experiment was successfully completed.

Results are verified