19/9

Task-7

Triggers, views and Exception

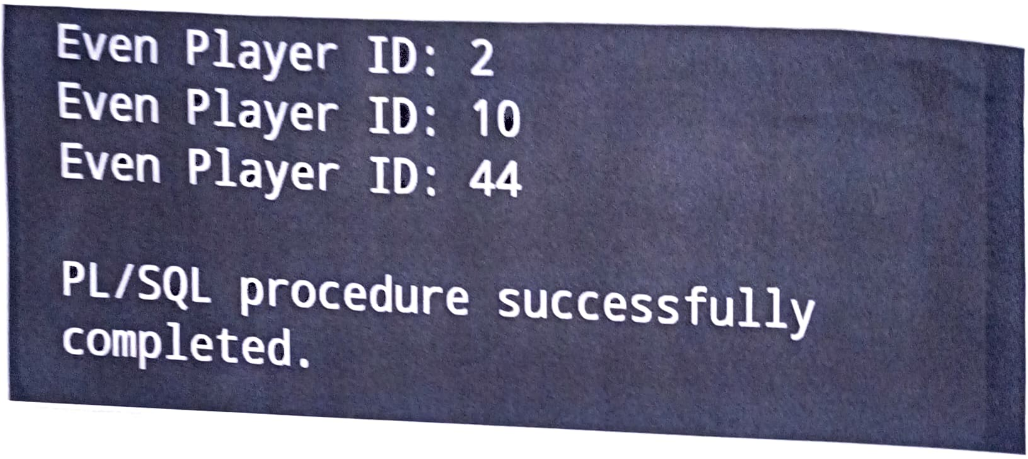
Aim: To conduct events, views & Exception on CRUD operations for restricting Phenomenon

a) To create a trigger in PL/SQL the automatically inserts a new record in the match_result table when a new record is inserted.

```
create OR Replace Trigger insert_match_result
After insert on match
for each Row
Begin
Insert into match_result(match_id, Team ID, result)
values (:new match ID, :new Team ID, 'Pending');
End;
```

b) To create a View that display the details of Players along with their details

```
SQL> create view player Team Details As select Player]
P fname as player Name, P Team ID As player coach,
P. captain As player captain, (select t name from team
t where t. tem ID = P. team ID) As team name,
(select t. coach from t where t. name ID =
team t where t. team ID = P. team ID) As team
captain from player P;
SQL> Select * from player_team details;
```



```
Even Player ID: 2
Even Player ID: 10
Even Player ID: 44

PL/SQL procedure successfully
completed.
```

```
View created.
```

c) To write a non-recursive PL/SQL procedure to retrie[ve] even-numbered player IDs registered for any tournament.

```
Create or Replace procedure
Get even player IDs for tournament (in_tourn_id)

As
Begin
  out_even_player_id = sys.ODCINUMBerlist (); ....

 for player_rec In (SeleCt playerID from player
where tournament ID = in_tournament_id AND MOD
(Player 10,2) loop.
  out_even_player_ids (out_even_player_ids, count
:= player_rec.player ID;
  End loop;
  End;
 DeClare
   tournament_Id number: = 123; ... Replace with the
destined tournament ID.
  even_player_ids sys.OD< Number List;
 Begin
  Get even_player IDs for Tournament (tournament,
 t__d, even_player_ids);
 for i in 1 .. even_player_ids.count loop
 DBMS_output.PUT_Line ('even player ID: ');
  end loop;
  End;
```

Result: Thus, the triggers and views is
   Executed successfully