Solutions with marks

1- SQL Statement :

SELECT COLUMN_NAME, DATA_TYPE FROM USER_TAB_COLUMNS

WHERE TABLE_NAME= ' Team_Staff '; 0.5 point

2- GRANT CREATE TABLE to USERNAME; 0.5 point

3- REVOKE privilege ON [table/view] FROM username; 0.5 point

4- This is a PL/SQL code with the structure of error handling using the EXCEPTION block. 0.5 point

5-

SELECT COUNT(*) AS MatchCount 0.25 point

FROM Match 0.25 point

WHERE Team1_ID = 2 OR Team2_ID = 2; 0.25 point

Result = 4 0.25 point

---------- if they use the match_organization table

SELECT COUNT(DISTINCT TEAM_ID) AS Matches_Played 0.25 point

FROM Match_Organization0.25 point

WHERE Team_ID = 2; 0.25 point

Result = 4 0.25 point

6- CREATE OR REPLACE FUNCTION GetAthleteRole (AName IN VARCHAR)

RETURN VARCHAR2 IS

CURSOR crsr IS SELECT Athlete_ID FROM Athlete

WHERE Athlete_Name = AName;

crsr_rec crsr%ROWTYPE;

Role_Name VARCHAR2(100);

BEGIN

FOR crsr_rec IN crsr LOOP

select Role INTO Role_Name

FROM Team_Staff WHERE Athlete_ID = crsr_rec. Athlete_ID;

END LOOP;

RETURN Role_Name ;

END;

/

BEGIN --displaying

dbms_output.put_line ('The role of athlete John Doe is:' || GetAthleteRole ('John Doe'));

END;

/

1.5 points ==(General correct syntax 0.25; variables 0.5== (cursor ;crsr_rec crsr and roleName); loop and return value 0.5, displaying :0.25)

----------Solution without the cursor

```
CREATE OR REPLACE FUNCTION GetAthleteRole (AName IN VARCHAR2)
RETURN VARCHAR2 IS
  Role_Name VARCHAR2(100);
BEGIN
  SELECT ts.Role INTO Role_Name
  FROM Athlete a
  JOIN Team_Staff ts ON a.Athlete_ID = ts.Athlete_ID
  WHERE a.Athlete_Name = AName

END;
/
BEGIN
  dbms_output.put_line (‘The role of athlete John Doe is:' || GetAthleteRole ('John Doe'));
END;
/
```

1.5 points ==(General correct syntax 0.5 (It suppose to be correct); variables 0.25; JOIN 0.5, displaying :0.25)

7- SET SERVEROUTPUT ON; 0.5 point

8-

```
CREATE OR REPLACE TRIGGER check_membership_update
BEFORE UPDATE OF membership_level ON Athlete
FOR EACH ROW
BEGIN
 IF :OLD.membership_level = 'Gold' AND :NEW.membership_level = 'Silver' THEN
  RAISE_APPLICATION_ERROR(-20001, 'Cannot downgrade from Gold to Silver membership.');
 END IF;
```

```
END;

/
```

2 point == (General correct syntax 0. 5; BEFORE UPDATE 0. 5; The condition 0. 5; Raising Error 0. 5)

```
9-      SELECT Athlete.Athlete_Name, Team.Team_Name, Team_Staff.Role,
Team_Staff.Join_Date

FROM Team_Staff

JOIN Athlete ON Team_Staff.Athlete_ID = Athlete.Athlete_ID

JOIN Team ON Team_Staff.Team_ID = Team.Team_ID;
```

2 point == (General correct syntax 0.5; All the needed attributes 0.75; The Join 0.5 ;Result 0.25)

10- One of the two users should select the table for update 1 point