Q4----

**View**

Name: vwPlayerRosters

Purpose: A view storing the players for each team in the database

Attributes:

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Size | Meaning |
| playerID | Number | 38 | The unique Id of the player |
| pgRegNumber | Varchar2 | 15 | The player’s registration number |
| pFirstName | Varchar2 | 25 | The first name of the player |
| pLastName | Varchar2 | 25 | The last name of the player |
| pIsActive | Number | 38 | If the player is active |
| rPlayerID | Number | 38 | The unique Id of the player |
| rosterID | Number | 38 | The roster identification of the player |
| rTeamID | Number | 38 | The team id that the player belongs to |
| rosterIsActive | Number | 38 | Value of 1 indicates that the roster is active |
| teamName | Varchar2 | 10 | The name of the team the player belongs to |
| tIsActive | Varchar2 | 38 | Value of 1 indicates the team is active |
| tJerseycolour | Varcahr2 | 10 | The jersey color of the player |

Data Sample:

A screenshot of a computer

Description automatically generated

Q5---

**Procedure**

Name: spTeamRosterByID

Purpose: Displays a report of the players in team input

Parameters:

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Size | Meaning |
| pTeamId | Number | Default | Input. the team id to be searched for in the query |
| cPlayer | Sys\_refcursor | NOT APPLICABLE | Output. the player data holding teamname, rosterid and fullname  Query:  SELECT  teamName,  rosterID,  pFirstName || ' ' || pLastName AS fullName  FROM vwPlayerRosters  WHERE rTeamID = pTeamID; |
| Exitcode | number | Default | Output. Indicates successful (0) or unsuccessful procedure run. |

Local Variables: NOT APPLICABLE

Error Codes:

-5 : No Data found related to pTeamId

-1 : generic error, all other exceptions not handled

Execution:

DECLARE

teamID NUMBER := 212;

cTeamRoastPlayer SYS\_REFCURSOR;

teamName VARCHAR2(25);

rosterID NUMBER;

fullName VARCHAR2(50);

exitcode NUMBER;

BEGIN

spTeamRosterByID(teamID, cTeamRoastPlayer, exitcode);

IF exitcode = 0 THEN

DBMS\_OUTPUT.PUT\_LINE('Fetching Data...');

DBMS\_OUTPUT.PUT\_LINE('----------------------------------------');

DBMS\_OUTPUT.PUT\_LINE('Team Name | Roster ID | Full Name');

DBMS\_OUTPUT.PUT\_LINE('----------------------------------------');

LOOP

FETCH cTeamRoastPlayer

INTO

teamName,

rosterID,

fullName;

EXIT WHEN cTeamRoastPlayer%NOTFOUND;

-- display

DBMS\_OUTPUT.PUT\_LINE(

RPAD(teamName, 9) || ' | ' ||

RPAD(rosterID, 9) || ' | ' ||

RPAD(fullName, 25)

);

END LOOP;

CLOSE cTeamRoastPlayer;

DBMS\_OUTPUT.PUT\_LINE('----------------------------------------');

ELSE

DBMS\_OUTPUT.PUT\_LINE('Exit code: ' || exitcode);

END IF;

END;

Output:

A screenshot of a computer

Description automatically generated

Q6—

**Procedure**

Name: spTeamRosterByName

Purpose: Displays a report of the players in team input

Parameters:

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Size | Meaning |
| tName | Varchar2 | 10 | Input. the team name to be searched for in the query |
| cPlayer | Sys\_refcursor | Default | Output. the player data holding teamname, rosterid and fullname  Query:  SELECT  teamName,  rosterID,  pFirstName || ' ' || pLastName AS fullName  FROM vwPlayerRosters  WHERE rTeamID = pTeamID; |
| Exitcode | number | Default | Output. Indicates successful (0) or unsuccessful procedure run. |

Local Variables: NOT APPLICABLE

Error Codes:

-8 : cursor invalid resulting from a failed Query tName input not existing in database

-1 : generic error, all other exceptions not handled

Execution:

DECLARE

vTeamName VARCHAR2(25);

cTeamRoastPlayer SYS\_REFCURSOR;

cTeamRoast VARCHAR2(25);

rosterID NUMBER;

fullName VARCHAR2(50);

exitcode NUMBER;

BEGIN

vTeamName := '&team\_name'; --get team

spTeamRosterByName(vTeamName, cTeamRoastPlayer, exitcode);

IF exitcode = 0 THEN

DBMS\_OUTPUT.PUT\_LINE('----------------------------------------');

DBMS\_OUTPUT.PUT\_LINE('Team Name | Roster ID | Full Name');

DBMS\_OUTPUT.PUT\_LINE('----------------------------------------');

LOOP

FETCH cTeamRoastPlayer INTO cTeamRoast, rosterID, fullName;

EXIT WHEN cTeamRoastPlayer%NOTFOUND;

-- Display

DBMS\_OUTPUT.PUT\_LINE(

RPAD(cTeamRoast, 11) || ' | ' ||

RPAD(rosterID, 11) || ' | ' ||

RPAD(fullName, 25)

);

END LOOP;

CLOSE cTeamRoastPlayer;

DBMS\_OUTPUT.PUT\_LINE('----------------------------------------');

ELSE

DBMS\_OUTPUT.PUT\_LINE('Exit code: ' || exitcode);

END IF;

END;

Output:

A screenshot of a computer

Description automatically generated

Q11—

**Procedure**

Name: spSchedPastGames

Purpose: Displays the games played in the past input days from user’s current date

Parameters:

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Size | Meaning |
| n | Number | Default | Input. The amount of days to be deducted from current date |
| Exitcode | number | Default | Output. Indicates successful (0) or unsuccessful procedure run. |

Local Variables

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Size | Meaning |
| matchCt | Number | Default | Stores the number of games found between (sysdate – n) and sysdate |
| e\_negativeInput | Exception | NOT APPLICABLE | User defined exception, only positive values allowed. Exception raised when input value is below 0 |
| data\_cur | CURSOR | NOT APPLICABLE | Stores the row returned from Query:  SELECT \*  FROM vwSchedule s  WHERE s.gameDateTime BETWEEN TRUNC(SYSDATE) - n AND TRUNC(SYSDATE)  AND s.isplayed = 1; |
| Data\_rec | vwSchedule%ROWTYPE | NOT APPLICABLE | Stores the row fetched from data\_cur; Data type matches vwSchedule attributes |

Error Codes:

-1 : generic error, all other exceptions not handled

-5 : no data found related to games query

-6 : too many rows returned in a query, only 1 row per loop is handled

-7 : invalid input, number of days to deduct from current date less than 0

Execution:

DECLARE exitcode INT;

BEGIN

spSchedPastGames(60, exitcode);

DBMS\_OUTPUT.PUT\_LINE('Exitcode: ' || exitcode);

END;

Output:

A screenshot of a computer

Description automatically generated

Q12—

**Table**

Name: tempStandings

Purpose: A table storing each teams calculated value of wins, loses, ties, points, total goals, goals scored against, and goals difference

Attributes:

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Size | Meaning |
| Theteamid | Number | 38 | The team’s unique identifier |
| Teamname | Varchar2 | 10 | The team’s name |
| Gp | Number | Default | The total games played by team |
| W | Number | Default | The sum of total games won |
| L | Number | Default | The sum of total games lost |
| T | Number | Default | The sum of total games tied |
| Pts | Number | Default | The sum of total points |
| Gf | Number | Default | The sum of total goals scored |
| Ga | Number | Default | The sum of total goals scored against the team |
| Gd | Number | Default | The difference between goals scored and goals scored against |

Data Sample:

A table with numbers and symbols

Description automatically generated

**Procedure**

Name: spRunStandings

Purpose: Displays the data stored within tempStandings table

Parameters: Not applicable

Local Variables:

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Size | Meaning |
| csStandings | Cursor | NOT APPLICABLE | Stores the row returned from the query:  SELECT \* FROM tempStandings |
| sRpt | ROWTYPE | NOT APPLICABLE | Stores the row fetched from csStandings |
| found | Boolean | NOT APPLICABLE | Value is true indicating query successfully returned a row |

Error Codes: NOT APPLICABLE

Execution:

DECLARE exitcode NUMBER;

BEGIN spRunStandings;

END;

Output:

A table of numbers and names

Description automatically generated

Q13—

**Trigger**

Name: trgUpdateTempStandings

Purpose: Automates the update of tempStandings table and executes spRunStandings

Error Codes: NOT APPLICABLE

Execution: UPDATE games SET homescore = 96 WHERE gameid = 1;

Output:

A table of numbers and names

Description automatically generated

Q14—

**Procedure**

Name: spMVPperteam

Purpose: Displays a list of top players in each team. Teams included in the list are teams who have scored at least 1 goal.

Parameters:

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Size | Meaning |
| Exitcode | number | Default | Output. Indicates successful (0) or unsuccessful procedure run. |

Local Variables:

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Size | Meaning |
| mvp\_cur | CURSOR | NOT APPLICABLE | Stores the row returned from the query:  SELECT  p.playerid,  lastname,  firstname,  SUM(numgoals) AS goalCt,  g.teamid,  teamname  FROM players p  JOIN goalscorers g ON p.playerID = g.playerid  JOIN teams t ON g.teamid = t.teamid  HAVING(SUM(numgoals)) || g.teamid IN (  SELECT MAX(goalCt) || teamid  FROM(  SELECT p.playerid,  lastname,  firstname,  SUM(numgoals) AS goalCt,  teamid  FROM players p  JOIN goalscorers g ON p.playerID = g.playerid  GROUP BY  p.playerid,  lastname,  firstname,  teamid  )  GROUP BY teamid  )  GROUP BY  p.playerid,  lastname,  firstname,  g.teamid,  teamname  ORDER BY  goalct DESC,  g.teamID DESC; |
| mRpt | mvp\_cur%ROWTYPE | NOT APPLICABLE | Stores the row fetched from mvp\_cur |
| found | Boolean | NOT APPLICABLE | Value is true indicating query successfully returned a row |

Error Codes:

-8 : cursor invalid resulting from a failed Query stored in mvp\_cur

-6 : too many rows returned in a query, only 1 row per loop is handled

-5 : No Data found related to mvp\_cur

-1 : generic error, all other exceptions not handled

Execution:

DECLARE exitcode NUMBER;

BEGIN spMVPperteam(exitcode);

END;

Output:

A screenshot of a computer

Description automatically generated A white screen with black text

Description automatically generated

**Exit Codes**

Purpose:

Error Codes simplify exception handling of each Procedure and Function

Definition:

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Type | Value | Meaning |
| Success | Number | 0 | Procedure or Function executed successfully |
| Generic Error | Number | -1 | All other exceptions not handled |
| Insert Error | Number | -2 | Table row Insertion failed. 0 row counts returned after Insert operation executed |
| Insert Existing ID | Number | -201 | Table row insertion error related to inserting existing primary id |
| Update Error | Number | -3 | Table row Update failed. 0 row counts returned after Update operation executed. |
| Delete Error | Number | -4 | Table row Delete failed. 0 row counts returned after Delete operation executed. |
| No Data Found Error | Number | -5 | No data returned from query |
| Too Many Rows returned | Number | -6 | More than 1 row returned from query |
| Invalid Input | Number | -7 | Input invalid. Validity of input is function/procedure dependent. |
| Invalid Cursor | Number | -8 | Fetching from a cursor that is not open or non existing |