Assignment 2
Name: Danish Sharma, Umesh Dharmendra Chaudhari, Yash Yadav
IDs: 148201205,114811201 ,116431206
Group: 5
DBS311 NCC

User Guide

To find how to use an object made by group 5: **CTRL + f** and then type the object's name.

spPlayersInsert

Parameters are of the same type as Players table :

- pId: NUMBER, 38 sig digits no decimal, playerId type
- rNum: VARCHAR(15), 15 character, regNumber type
- lName: VARCHAR(25), 25 character, lastName type
- fName: VARCHAR(25), 25 character, firstName type
- isAct: NUMBER, 38 sig digits (Range 0 to 1) no decimal, isActive type
- pErr: OUT, NUMBER,38 sig digits no decimal, playerId type

Expected Output:

You will get pErr for an output that would describe whether the insertion was successful or not. A positive number for pErr means success and a negative number means some error occurred.

Potential Error:

- When a playerId trying to be inserted when the playerId already exists.
- isAct is not in the range of 0 or 1.
- Inserting wrong values, for example trying to insert characters in a number parameter.

Purpose:

It is made for software to contact Oracle to insert a new Player's information in the database.

Non-saved procedural code:

```
DECLARE
      errP players.playerId%type;
BEGIN
      spPlayersInsert(123, ‘123456’, ‘Prac’, ‘Test’, 1, errP);
      IF errP > 0 THEN
            DBMS_OUTPUT.PUT_LINE(‘Insert Success!’);
      ELSIF errP = -1 THEN
            DBMS_OUTPUT.PUT_LINE(‘ERROR: Wrong values!’);
      ELSIF errP = -2 THEN
            DBMS_OUTPUT.PUT_LINE(‘ERROR: Others!’);
      ELSIF errP = -3 THEN
            DBMS_OUTPUT.PUT_LINE(‘ERROR: No rows inserted!’);
      END IF;
END;
```

spPlayersUpdate

Parameters are of same type as Players table :

- pId: NUMBER, 38 sig digits no decimal, playerId type
- rNum: VARCHAR(15), 15 character, regNumber type
- lName: VARCHAR(25), 25 character, lastName type
- fName: VARCHAR(25), 25 character, firstName type
- isAct: NUMBER, 38 sig digits (Range 0 to 1) no decimal, isActive type
- pErr: OUT, NUMBER,38 sig digits no decimal, playerId type

Expected Output:

You will get pErr for an output that would describe whether the update was successful or not. A positive number for pErr means success and a negative number means some error.

Potential Error:

- When playerId trying to be updated when the playerId does not exist.
- isAct is not in the range of 0 or 1.
- Updating wrong values, for example trying to insert characters in a number parameter.

Purpose:

It is made for software to contact Oracle to update a Player's information in the database.

Non-saved procedural code:

```
DECLARE
     errP players.playerId%type;
BEGIN
     spPlayersUpdate(123, ‘123456’, ‘Prac’, ‘Test’, 1, errP);
     IF errP > 0 THEN
          DBMS_OUTPUT.PUT_LINE(‘Update Success!’);
     ELSIF errP = -1 THEN
          DBMS_OUTPUT.PUT_LINE(‘ERROR: Wrong values!’);
     ELSIF errP = -2 THEN
          DBMS_OUTPUT.PUT_LINE(‘ERROR: Others!’);
     ELSIF errP = -3 THEN
          DBMS_OUTPUT.PUT_LINE(‘ERROR: No rows updated!’);
     END IF;
END;
```

spPlayersDelete

Parameters are of the same type as the Players table :

- pId: NUMBER, 38 sig digits no decimal, playerId type
- pErr: OUT, NUMBER, 38 sig digits, playerId type

Expected Output:

You will get pErr for an output that would describe whether the removal was successful or not. A positive number for pErr means success and a negative number means some error.

Potential Error:

- When the same playerId trying to be deleted when the playerId does not exist.
- Inserting wrong values, for example trying to insert characters in a number parameter.

Purpose:

It is made for software to contact Oracle to delete new Player' information in the database.

Non-saved procedural code:

```
DECLARE
     errP players.playerId%type;
BEGIN
     spPlayersUpdate(123, errP);
     IF errP > 0 THEN
          DBMS_OUTPUT.PUT_LINE('Delete Success!');
     ELSIF errP = -1 THEN
          DBMS_OUTPUT.PUT_LINE('ERROR: Wrong values!');
     ELSIF errP = -2 THEN
          DBMS_OUTPUT.PUT_LINE('ERROR: Others!');
     ELSIF errP = -3 THEN
          DBMS_OUTPUT.PUT_LINE('ERROR: No rows deleted!');
     END IF;
END;
```

spPlayersSelect

Parameters are of same type as Players table :

- pId: NUMBER, 38 sig digits no decimal, playerId type
- rNum: OUT, VARCHAR(15), 15 character, regNumber type
- lName: OUT, VARCHAR(25), 25 character, lastName type
- fName: OUT, VARCHAR(25), 25 character, firstName type
- isAct: OUT, NUMBER, 38 sig digits (Range 0 to 1) no decimal, isActive type
- pErr: OUT, NUMBER,38 sig digits no decimal, playerId type

Expected Output:

You will get pErr for an output that would describe whether the SELECT INTO was successful or not. A positive number for pErr means success and a negative number means some error.

Potential Error:

- When playerId trying to be selected for output when a player does not exist or
- Too many players exists.
- Insert wrong values, for example trying to insert characters in a number parameter.

Purpose:

It is made for software to contact Oracle to get a Player's information in the database.

Non-saved procedural code:

```
DECLARE
     errP players.playerId%type;
     pId players.playerId%type := 123;
     rNum players.regNumber%type;
     lName players.lastName%type;
     fName players.firstName%type;
     isAct players.isActive%type;
BEGIN
     BEGIN
     spPlayersSelect(pId, rNum, lName, fName, isAct, errP);
     IF errP > 0 THEN
          DBMS_OUTPUT.PUT_LINE('Player ID: ' || pId);
          DBMS_OUTPUT.PUT_LINE('Player Reg Num: ' || rNum);
          DBMS_OUTPUT.PUT_LINE('Player Last Name: ' ||
          lName);
          DBMS_OUTPUT.PUT_LINE('Player First Name: ' ||
          fName);
          DBMS_OUTPUT.PUT_LINE('Player Is Active: ' ||
          isAct);
     ELSIF errP = -1 THEN
          DBMS_OUTPUT.PUT_LINE('ERROR: No data found!');
     ELSIF errP = -2 THEN
          DBMS_OUTPUT.PUT_LINE('ERROR: Too many rows!');
     ELSIF errP = -3 THEN
          DBMS_OUTPUT.PUT_LINE('ERROR: Others!');
     END IF;
```

```
END;
```

spSlLocationsInsert

Parameters are of same type as SlLocations table :

- lId: NUMBER, 38 sig digits no decimal, locationId type
- locName: VARCHAR(50), 50 character, locationName type
- lField: NUMBER, 38 sig digits no decimal, fieldLength type
- isAct: NUMBER, 38 sig digits (Range 0 to 1) no decimal, isActive type
- lErr: OUT, NUMBER,38 sig digits no decimal, locationId type

Expected Output:

You will get lErr for an output that would describe whether the insertion was successful or not. A positive number for lErr means success and a negative number means some error occurred.

Potential Error:

- When a locationId trying to be inserted when the locationId already exists.
- isAct is not in the range of 0 or 1.
- Inserting wrong values, for example trying to insert characters in a number parameter.

Purpose:

It is made for software to contact Oracle to insert a new Location’s information in the database.

Non-saved procedural code:

```
DECLARE
     errL slLocations.locationId%type;
BEGIN
     spSlLocationsInsert(123, ‘Stadium’, 35, 1, errL);
     IF errL > 0 THEN
          DBMS_OUTPUT.PUT_LINE(‘Insert Success!’);
     ELSIF errL = -1 THEN
          DBMS_OUTPUT.PUT_LINE(‘ERROR: Wrong values!’);
     ELSIF errL = -2 THEN
          DBMS_OUTPUT.PUT_LINE(‘ERROR: Others!’);
     ELSIF errL = -3 THEN
          DBMS_OUTPUT.PUT_LINE(‘ERROR: No rows inserted!’);
     END IF;
END;
```

spSlLocationsUpdate

Parameters are of the same type as SlLocations table :

- lId: NUMBER, 38 sig digits no decimal, locationId type
- locName: VARCHAR(50), 50 character, locationName type
- lField: NUMBER, 38 sig digits no decimal, fieldLength type
- isAct: NUMBER, 38 sig digits (Range 0 to 1) no decimal, isActive type
- lErr: OUT, NUMBER,38 sig digits no decimal, locationId type

Expected Output:

You will get lErr for an output that would describe whether the update was successful or not. A positive number for lErr means success and a negative number means some error.

Potential Error:

- When locationId trying to be updated when the locationId does not exist.
- isAct is not in the range of 0 or 1.
- Updating wrong values, for example trying to insert characters in a number parameter.

Purpose:

It is made for software to contact Oracle to update a Location's information in the database.

Non-saved procedural code:

```
DECLARE
     errL slLocations.locationId%type;
BEGIN
     spSlLocationsUpdate(123, 'Stadium', 35, 1, errL);
     IF errL > 0 THEN
          DBMS_OUTPUT.PUT_LINE('Update Success!');
     ELSIF errL = -1 THEN
          DBMS_OUTPUT.PUT_LINE('ERROR: Wrong values!');
     ELSIF errL = -2 THEN
          DBMS_OUTPUT.PUT_LINE('ERROR: Others!');
     ELSIF errL = -3 THEN
          DBMS_OUTPUT.PUT_LINE('ERROR: No rows updated!');
     END IF;
END;
```

spSlLocationsDelete

Parameters are of the same type as SlLocations table :

- lId: NUMBER, 38 sig digits no decimal, locationId type
- lErr: OUT, NUMBER,38 sig digits no decimal, locationId type

Expected Output:

You will get lErr for an output that would describe whether the delete was successful or not. A positive number for lErr means success and a negative number means some error.

Potential Error:

- When the same locationId trying to be deleted when the locationId does not exist.
- Inserting wrong values, for example trying to insert characters in a number parameter.

Purpose:

It is made for software to contact Oracle to delete a Location's information in the database.

Non-saved procedural code:

```
DECLARE
     errL slLocations.locationId%type;
BEGIN
     spSlLocationsDelete(123, errL);
     IF errL > 0 THEN
          DBMS_OUTPUT.PUT_LINE('Delete Success!');
     ELSIF errL = -1 THEN
          DBMS_OUTPUT.PUT_LINE('ERROR: Wrong values!');
     ELSIF errL = -2 THEN
          DBMS_OUTPUT.PUT_LINE('ERROR: Others!');
     ELSIF errL = -3 THEN
          DBMS_OUTPUT.PUT_LINE('ERROR: No rows delete!');
     END IF;
END;
```

spSlLocationsSelect

Parameters are of the same type as the SlLocations table :

- lId: NUMBER, 38 sig digits no decimal, locationId type
- locName: OUT, VARCHAR(50), 50 character, locationName type
- lField: OUT, NUMBER, 38 sig digits no decimal, fieldLength type
- isAct: OUT, NUMBER, 38 sig digits (Range 0 to 1) no decimal, isActive type
- lErr: OUT, NUMBER,38 sig digits no decimal, locationId type

Expected Output:

You will get lErr for an output that would describe whether the SELECT INTO was successful or not. A positive number for lErr means success and a negative number means some error.

Potential Error:

- When locationId trying to be selected for output when a location does not exist or
- Too many locations exist.
- Insert wrong values, for example trying to insert characters in a number parameter.

Purpose:

It is made for software to contact Oracle to get a Location's information in the database.

Non-saved procedural code:

```
DECLARE
     errL slLocation.locationId%type;
     lId slLocations.locationId%type := 123;
     locName slLocations.locationName%type;
     lField slLocations.fieldLength%type;
     isAct slLocations.isActive%type;
BEGIN
     BEGIN
     spPlayersSelect(lId, locName, lField,isAct, errL);
     IF errL > 0 THEN
          DBMS_OUTPUT.PUT_LINE('SlLocation ID: ' || LId);
          DBMS_OUTPUT.PUT_LINE('SlLocation Name: ' ||
          locName);
          DBMS_OUTPUT.PUT_LINE('SlLocation Field Length: '
          || lField);
          DBMS_OUTPUT.PUT_LINE('SlLocation Is Active: ' ||
          isAct);
     ELSIF errL = -1 THEN
          DBMS_OUTPUT.PUT_LINE('ERROR: No data found!');
     ELSIF errL = -2 THEN
          DBMS_OUTPUT.PUT_LINE('ERROR: Too many rows!');
     ELSIF errL = -3 THEN
          DBMS_OUTPUT.PUT_LINE('ERROR: Others!');
     END IF;
END;
```

spGoalScorersInsert

Parameters would be:

- gId: NUMBER, 38 sig digits no decimal, goalId type
- gmId: NUMBER, 38 sig digits no decimal, gameId type
- pId: NUMBER, 38 sig digits no decimal, playerId type
- tId: NUMBER, 38 sig digits no decimal, teamId type
- numG: NUMBER, 38 sig digits no decimal, numGoals type
- numAss: NUMBER, 38 sig digits no decimal, numAssists type
- gsErr: OUT, NUMBER,38 sig digits no decimal, goalId type

Expected Output:

You will get gsErr for an output that would describe whether the insertion was successful or not. A positive number for gsErr means that goalId was returned as NULL was inserted for gId,

if gsErr is 0 that means that a row was inserted and a negative number means some error occurred, for instance for -3, no rows inserted, and for -1, no data found, and for -2,

too many rows, and for -4, it is a value error and, for -5, it is other error.

Potential Error:

- When a goalId trying to be inserted when the goalId already exists.
- Inserting wrong values, for example trying to insert characters in a number parameter.
- no rows inserted.
- no data found.
- too many rows.
- value error.

Purpose:

It is made for software to contact Oracle to insert new goalScorers information in the database.

Non-saved procedural code:

```
     DECLARE
           errGs goalScorers.goalId%type;
     BEGIN
           spGoalScorersInsert(NULL, 30, 111222, 200, 2, 0, errGs);
           IF errGs > 0 THEN
                DBMS_OUTPUT.PUT_LINE(‘Insert Success, ID: ’ ||
errGs);
           ELSIF errGs = 0 THEN
                DBMS_OUTPUT.PUT_LINE(‘Insert Success!’);
           ELSIF errGs = -1 THEN
                DBMS_OUTPUT.PUT_LINE(‘ERROR: No Data Found!’);
           ELSIF errGs = -2 THEN
                DBMS_OUTPUT.PUT_LINE(‘ERROR: Too Many Rows!’);
           ELSIF errGs = -3 THEN
                DBMS_OUTPUT.PUT_LINE(‘ERROR: No rows inserted!’);
```

```
        ELSIF errGs = -4 THEN
              DBMS_OUTPUT.PUT_LINE('ERROR: Wrong Value!');
        ELSIF errGs = -5 THEN
              DBMS_OUTPUT.PUT_LINE('ERROR: Others!');
        END IF;
  END;
```

spGoalScorersUpdate

Parameters would be:

- gId: NUMBER, 38 sig digits no decimal, goalId type
- gmId: NUMBER, 38 sig digits no decimal, gameId type
- pId: NUMBER, 38 sig digits no decimal, playerId type
- tId: NUMBER, 38 sig digits no decimal, teamId type
- numG: NUMBER, 38 sig digits no decimal, numGoals type
- numAss: NUMBER, 38 sig digits no decimal, numAssists type
- gsErr: OUT, NUMBER,38 sig digits no decimal, goalId type

Expected Output:

You will get gsErr for an output that would describe whether the update was successful or not. A positive number for gsErr means success and a negative number means some error.

Potential Error:

- When goalId trying to be updated when the goalId does not exist.
- Updating wrong values, for example trying to insert characters in a number parameter.

Purpose:

It is made for software to contact Oracle to update a goalScorer's information in the database.

Non-saved procedural code:

```
DECLARE
     errGs goalScorers.goalId%type;
BEGIN
     spGoalScorersUpdate(123, 30, 111222, 200, 2, 0, errGs);
     IF errGs > 0 THEN
          DBMS_OUTPUT.PUT_LINE('Update Success!');
     ELSIF errGs = -1 THEN
          DBMS_OUTPUT.PUT_LINE('ERROR: Wrong values!');
     ELSIF errGs = -2 THEN
          DBMS_OUTPUT.PUT_LINE('ERROR: Others!');
     ELSIF errGs = -3 THEN
          DBMS_OUTPUT.PUT_LINE('ERROR: No rows updated!');
     END IF;
END;
```

spGoalScorersDelete

Parameters are of the same type as the goalScorers table :

- gId: NUMBER, 38 sig digits no decimal, goalId type
- gsErr: OUT, NUMBER, 38 sig digits, goalId type

Expected Output:

You will get gsErr for an output that would describe whether the removal was successful or not. A positive number for gsErr means success and a negative number means some error.

Potential Error:

- When the same goalId trying to be deleted when the goalId does not exist.
- Inserting wrong values, for example trying to insert characters in a number parameter.

Purpose:

It is made for software to contact Oracle to delete new goalScorer’s information in the database.

Non-saved procedural code:

```
DECLARE
     errGs goalScorers.goalId%type;
BEGIN
     spGoalScorersDelete(123, errGs);
     IF errGs> 0 THEN
          DBMS_OUTPUT.PUT_LINE(‘Delete Success!’);
     ELSIF errGs= -1 THEN
          DBMS_OUTPUT.PUT_LINE(‘ERROR: Wrong values!’);
     ELSIF errGs= -2 THEN
          DBMS_OUTPUT.PUT_LINE(‘ERROR: Others!’);
     ELSIF errGs= -3 THEN
          DBMS_OUTPUT.PUT_LINE(‘ERROR: No rows delete!’);
     END IF;
END;
```

spGoalScorersSelect

Parameters are of same type as Players table :

- gId IN goalScorers.goalId%type,
- gmId OUT goalScorers.gameId%type,
- pId OUT goalScorers.playerId%type,
- tId OUT goalScorers.teamId%type,
- numG OUT goalScorers.numgoals%type,
- numAss OUT goalScorers.numassists%type,
- gsErr OUT goalScorers.goalId%type

Expected Output:

You will get gsErr for an output that would describe whether the SELECT INTO was successful or not. A positive number for gsErr means success and a negative number means some error.

Potential Error:

- When goalId trying to be selected for output when a goal does not exist or
- Too many goalscorers exists.
- Insert wrong values, for example trying to insert characters in a number parameter.

Purpose:

It is made for software to contact Oracle to get a goalScorers’s information in the database.

Non-saved procedural code:

```
DECLARE
           gId goalScorers.goalId%type,
          gmId goalScorers.gameId%type,
          pId goalScorers.playerId%type,
         tId goalScorers.teamId%type,
         numG goalScorers.numgoals%type,
         numAss goalScorers.numassists%type,
         errGs goalScorers.goalId%type

     BEGIN
          BEGIN
          spGoalScorersSelect(gId, gmId, pId, tId, numG, numAss,
errGs);
          IF errGs> 0 THEN
               DBMS_OUTPUT.PUT_LINE(‘Row selection Success!’);
          ELSIF errGs= -1 THEN
               DBMS_OUTPUT.PUT_LINE(‘ERROR: Wrong values!’);
          ELSIF errGs= -2 THEN
               DBMS_OUTPUT.PUT_LINE(‘ERROR: too many Rows!’);
          ELSIF errGs= -3 THEN
               DBMS_OUTPUT.PUT_LINE(‘ERROR: Others!’);
          END IF;
     END;
```

spRostersInsert

Parameters would be:

- rId: NUMBER, 38 sig digits no decimal, rosterId type
- pId: NUMBER, 38 sig digits no decimal, playerId type
- tId: NUMBER, 38 sig digits no decimal, teamId type
- isAct: NUMBER, 38 sig digits no decimal, isActive type
- jrNum: NUMBER, 38 sig digits no decimal, jerseyNumber type
- rsErr: OUT, NUMBER,38 sig digits no decimal, rosterId type

Expected Output:

You will get rsErr for an output that would describe whether the insertion was successful or not. A positive number for rsErr means that rosterId was returned as NULL was inserted for rsId,

if rsErr is 0 that means that a row was inserted and a negative number means some error occurred, for instance for -3, no rows inserted, and for -1, no data found, and for -2, too many rows, and for -4, it is a value error and, for -5, it is other error.

Potential Error:

- When a rosterId is trying to be inserted when the rosterId already exists.
- Inserting wrong values, for example trying to insert characters in a number parameter.
- no rows inserted.
- no data found.
- too many rows.
- value error.

Purpose:

It is made for software to contact Oracle to insert a new roster’s information in the database.

Non-saved procedural code:

```
DECLARE
      errRs rosters.rosterId%type;
BEGIN
      spRostersInsert(NULL, 111222, 200, 1, 55, errRs);
      IF errRs> 0 THEN
            DBMS_OUTPUT.PUT_LINE(‘Insert Success!’);
      ELSIF errRs= -1 THEN
            DBMS_OUTPUT.PUT_LINE(‘ERROR: No Data Found!’);
      ELSIF errRs= -2 THEN
            DBMS_OUTPUT.PUT_LINE(‘ERROR: Too Many Rows!’);
      ELSIF errRs= -3 THEN
            DBMS_OUTPUT.PUT_LINE(‘ERROR: No rows inserted!’);
      ELSIF errRs= -4 THEN
            DBMS_OUTPUT.PUT_LINE(‘ERROR: Wrong Value!’);
      ELSIF errRs= -5 THEN
            DBMS_OUTPUT.PUT_LINE(‘ERROR: Others!’);
      END IF;
END;
```

spRostersUpdate

Parameters would be:

- rId: NUMBER, 38 sig digits no decimal, rosterId type
- pId: NUMBER, 38 sig digits no decimal, playerId type
- tId: NUMBER, 38 sig digits no decimal, teamId type
- isAct: NUMBER, 38 sig digits no decimal, isActive type
- jrNum: NUMBER, 38 sig digits no decimal, jerseyNumber type
- rsErr: OUT, NUMBER,38 sig digits no decimal, rosterId type

Expected Output:

You will get rsErr for an output that would describe whether the update was successful or not. A positive number for rsErr means success and a negative number means some error.

Potential Error:

- When rosterId trying to be updated when the rosterId does not exist.
- isAct is not in the range of 0 or 1.
- Updating wrong values, for example trying to insert characters in a number parameter.

Purpose:

It is made for software to contact Oracle to update a roster's information in the database.

Non-saved procedural code:

```
DECLARE
     errRs spRostersUpdate.rosterId%type;
BEGIN
     spRostersUpdate(123, 30, 111222, 1, 2, errGs);
     IF errRs > 0 THEN
          DBMS_OUTPUT.PUT_LINE('Update Success!');
     ELSIF errRs = -1 THEN
          DBMS_OUTPUT.PUT_LINE('ERROR: Wrong values!');
     ELSIF errRs = -2 THEN
          DBMS_OUTPUT.PUT_LINE('ERROR: Others!');
     ELSIF errRs = -3 THEN
          DBMS_OUTPUT.PUT_LINE('ERROR: No rows updated!');
     END IF;
END;
```

spRostersDelete

Parameters are of the same type as the rosters table :

- rId: NUMBER, 38 sig digits no decimal, rosterId type
- rsErr: OUT, NUMBER, 38 sig digits, rosterId type

Expected Output:

You will get rsErr for an output that would describe whether the removal was successful or not. A positive number for rsErr means success and a negative number means some error.

Potential Error:

- When the same rosterId trying to be deleted when the rosterId does not exist.
- Inserting wrong values, for example trying to insert characters in a number parameter.

Purpose:

It is made for software to contact Oracle to delete new roster's information in the database.

Non-saved procedural code:

```
DECLARE
     errRs spRostersDelete.rosterId%type;
BEGIN
     spRostersDelete(123, errRs);
     IF errRs> 0 THEN
          DBMS_OUTPUT.PUT_LINE(‘Delete Success!’);
     ELSIF errRs= -1 THEN
          DBMS_OUTPUT.PUT_LINE(‘ERROR: Wrong values!’);
     ELSIF errRs= -2 THEN
          DBMS_OUTPUT.PUT_LINE(‘ERROR: Others!’);
     ELSIF errRs= -3 THEN
          DBMS_OUTPUT.PUT_LINE(‘ERROR: No rows delete!’);
     END IF;
END;
```

spRostersSelect

Parameters are of same type as Players table :

- rId IN rosters.rosterId%type,
- pId OUT rosters.playerId%type,
- tId OUT rosters.teamId%type,
- isAct OUT rosters.isActive%type,
- jrNum OUT rosters.jerseyNumber%type,
- rsErr OUT rosters.rosterId%type

Expected Output:

You will get rsErr for an output that would describe whether the SELECT INTO was successful or not. A positive number for rsErr means success and a negative number means some error.

Potential Error:

- When rosterId trying to be selected for output when a roster does not exist or
- Too many roster exists.
- Insert wrong values, for example trying to insert characters in a number parameter.

Purpose:

It is made for software to contact Oracle to get a roster's information in the database.

Non-saved procedural code:

```
DECLARE
     rId rosters.rosterId%type,
     pId rosters.playerId%type,
     tId rosters.teamId%type,
     isAct rosters.isActive%type,
     jrNum rosters.jerseyNumber%type,
     rsErr rosters.rosterId%type


BEGIN
     BEGIN
     spRostersSelect(rId, pId, tId, isAct, jrNum, errRs);
     IF errRs> 0 THEN
          DBMS_OUTPUT.PUT_LINE('Row selection Success!');
     ELSIF errRs= -1 THEN
          DBMS_OUTPUT.PUT_LINE('ERROR: Wrong values!');
     ELSIF errRs= -2 THEN
          DBMS_OUTPUT.PUT_LINE('ERROR: too many Rows!');
     ELSIF errRs= -3 THEN
          DBMS_OUTPUT.PUT_LINE('ERROR: Others!');
     END IF;
END;
```

spTeamsInsert

Parameters would be:

- tId: NUMBER, 38 sig digits no decimal, teamId typed type
- tName: VARCHAR(10), 10 character, tname type
- isAct: NUMBER, 38 sig digits(range 0 to 1) no decimal, isActive type
- jerseyColour: VARCHAR2(10), 10 character, jerseyColour type
- tErr: OUT, NUMBER,38 sig digits no decimal, teamId type

Expected Output:

You will get tErr for an output that would describe whether the insertion was successful or not. A positive number for tErr means that teamId was returned as NULL was inserted for tId, if tErr is 0 that means that a row was inserted and a negative number means some error occurred, for instance for -3, no rows inserted, and for -1, no data found, and for -2,too many rows, and for -4, it is value error and, for -5, it is other errors.

Potential Error:

- When a teamId is trying to be inserted when the teamId already exists.
- Inserting wrong values, for example trying to insert characters in a number parameter.

Purpose:

It is made for software to contact Oracle to insert a new team's information in the database.

Non-saved procedural code:

```
DECLARE
      tErr teams.teamId%type;
BEGIN
      spTeamsInsert(123, ‘Prac’, 1, ‘Test’, tErr);
      IF tErr > 0 THEN
            DBMS_OUTPUT.PUT_LINE(‘Insert Success!’);
      ELSIF tErr = -1 THEN
            DBMS_OUTPUT.PUT_LINE(‘ERROR: no data found!’);
      ELSIF tErr = -2 THEN
            DBMS_OUTPUT.PUT_LINE(‘ERROR: too many rows!’);
      ELSIF tErr = -3 THEN
            DBMS_OUTPUT.PUT_LINE(‘ERROR: No rows inserted!’);
      ELSIF tErr = -4 THEN
            DBMS_OUTPUT.PUT_LINE(‘ERROR: wrong values!’);
      ELSIF tErr = -5 THEN
            DBMS_OUTPUT.PUT_LINE(‘ERROR: others!’);
      END IF;
END;
```

spTeamsUpdate

Parameters would be:

- tId: NUMBER, 38 sig digits no decimal, teamId typed type
- tName: VARCHAR(10), 10 character, tname type
- isAct: NUMBER, 38 sig digits(range 0 to 1) no decimal, isActive type
- jerseyColour: VARCHAR2(10), 10 character, jerseyColour type
- tErr: OUT, NUMBER,38 sig digits no decimal, teamId type

Expected Output:

You will get tErr for an output that would describe whether the update was successful or not. A positive number for tErr means success and a negative number means some error.

Potential Error:

- When teamId trying to be updated when the teamId does not exist.
- isAct is not in the range of 0 or 1.
- Updating wrong values, for example trying to insert characters in a number parameter.

Purpose:

It is made for software to contact Oracle to update a Player’s information in the database.

Non-saved procedural code:

```
DECLARE
     errP teams.teamId%type;
BEGIN
     spteamsUpdate(123, ‘123456’, ‘Prac’, ‘Test’, 1, errP);
     IF errP > 0 THEN
          DBMS_OUTPUT.PUT_LINE(‘Update Success!’);
     ELSIF errP = -1 THEN
          DBMS_OUTPUT.PUT_LINE(‘ERROR: Wrong values!’);
     ELSIF errP = -2 THEN
          DBMS_OUTPUT.PUT_LINE(‘ERROR: Others!’);
     ELSIF errP = -3 THEN
          DBMS_OUTPUT.PUT_LINE(‘ERROR: No rows updated!’);
     END IF;
END;
```

spTeamsDelete

Parameters would be:

- tId: NUMBER, 38 sig digits no decimal, teamId typed type
- tErr: OUT, NUMBER,38 sig digits no decimal, teamId type

Expected Output:

You will get tErr fowar an output that would describe whether the delete was successful or not. A positive number for tErr means success and a negative number means some error.

Potential Error:

- When the same teamId trying to be deleted when the teamId does not exist.
- Inserting wrong values, for example trying to insert characters in a number parameter.

Purpose:

It is made for software to contact Oracle to delete a Location’s information in the database.

Non-saved procedural code:

```
DECLARE
     tErr teams.teamId%type;
BEGIN
     spteamsDelete(123, tErr);
     IF errL > 0 THEN
          DBMS_OUTPUT.PUT_LINE(‘Delete Success!’);
     ELSIF errL = -1 THEN
          DBMS_OUTPUT.PUT_LINE(‘ERROR: Wrong values!’);
     ELSIF errL = -2 THEN
          DBMS_OUTPUT.PUT_LINE(‘ERROR: Others!’);
     ELSIF errL = -3 THEN
          DBMS_OUTPUT.PUT_LINE(‘ERROR: No rows delete!’);
     END IF;
END;
```

spTeamsSelect

Parameters would be:

- tId: NUMBER, 38 sig digits no decimal, teamId typed type
- tName: VARCHAR(10), 10 character, tname type
- isAct: NUMBER, 38 sig digits(range 0 to 1) no decimal, isActive type
- jerseyColour: VARCHAR2(10), 10 character, jerseyColour type
- tErr: OUT, NUMBER,38 sig digits no decimal, teamId type

Expected Output:

You will get pErr for an output that would describe whether the SELECT INTO was successful or not. A positive number for tErr means success and a negative number means some error.

Potential Error:

- When teamId trying to be selected for output when a player does not exist or
- Too many players exists.
- Insert wrong values, for example trying to insert characters in a number parameter.

Purpose:

It is made for software to contact Oracle to get a Player’s information in the database

Non-saved procedural code:

```
      DECLARE
            tErr teams.teamId%type;
            tId teams.teamId%type := 123;
            tName teams.teamName%type;
            isAct players.isActive%type;
BEGIN
            BEGIN
            spPlayersSelect(tId, tName, isAct, tErr);
            IF tErr > 0 THEN
                  DBMS_OUTPUT.PUT_LINE(‘Team ID: ’ || tId);
                  DBMS_OUTPUT.PUT_LINE(‘Team Name: ’ || tName);
                  DBMS_OUTPUT.PUT_LINE(‘Player Last Name: ’ ||
                  lName);
                  DBMS_OUTPUT.PUT_LINE(‘Player Is Active: ’ ||
                  isAct);
            ELSIF errP = -1 THEN
                  DBMS_OUTPUT.PUT_LINE(‘ERROR: No data found!’);
            ELSIF tErr = -2 THEN
                  DBMS_OUTPUT.PUT_LINE(‘ERROR: Too many rows!’);
            ELSIF tErr = -3 THEN
                  DBMS_OUTPUT.PUT_LINE(‘ERROR: Others!’);
            END IF;
      END;
```

spGamesInsert

Parameters would be:

- gId: NUMBER, 38 sig digits no decimal, gameId typed type
- dID: NUMBER, 38 sig digits no decimal, divId typed type
- gNum:NUMBER, 38 sig digits no decimal, gameNum type
- gDateTime: DATE,
- gHTeam:NUMBER, 38 sig digits no decimal, gHTeam type
- gVTeam:NUMBER, 38 sig digits no decimal, gVTeam type
- gHScore:NUMBER, 38 sig digits no decimal, gHScore type
- gVScore:NUMBER, 38 sig digits no decimal, gVScore type
- gLocationId:NUMBER, 38 sig digits no decimal, gLocationId type
- gisPlayed:NUMBER, 38 sig digits no decimal, gisPlayed type
- gNotes: VARCHAR (20), 20 character, gNotes type
- gErr: OUT, NUMBER,38 sig digits no decimal, gameId type

Expected Output:

You will get gErr for an output that would describe whether the insertion was successful or not. A positive number for gErr means that teamId was returned as NULL was inserted for gId, if gErr is 0 that means that a row was inserted and a negative number means some error occurred, for instance for -3, no rows inserted, and for -1, no data found, and for -2,too many rows, and for -4, it is value error and, for -5, it is other errors.

Potential Error:

- When a gameId is trying to be inserted when the gameId already exists.
- Inserting wrong values, for example trying to insert characters in a number parameter.

Purpose:

It is made for software to contact Oracle to insert a new team's information in the database.

Non-saved procedural code:

```
DECLARE
     gErr games.gameId%type;
BEGIN
     spGamesInsert(123,2,3,To_date('12-01-2022',
          'dd-mm-yyyy'),212, 0, 300,
          0, 11, 0, NULL, gErr);
     IF gErr > 0 THEN
          DBMS_OUTPUT.PUT_LINE('Insert Success!');
     ELSIF gErr = -1 THEN
          DBMS_OUTPUT.PUT_LINE('ERROR: no data found!');
     ELSIF gErr = -2 THEN
          DBMS_OUTPUT.PUT_LINE('ERROR: too many rows!');
     ELSIF gErr = -3 THEN
          DBMS_OUTPUT.PUT_LINE('ERROR: No rows inserted!');
     ELSIF gErr = -4 THEN
          DBMS_OUTPUT.PUT_LINE('ERROR: wrong values!');
     ELSIF gErr = -5 THEN
          DBMS_OUTPUT.PUT_LINE('ERROR: others!');
```

```
      END IF;
END;
```

spGamesUpdate

Parameters would be:

- gId: NUMBER, 38 sig digits no decimal, gameId typed type
- dID: NUMBER, 38 sig digits no decimal, divId typed type
- gNum:NUMBER, 38 sig digits no decimal, gameNum type
- gDateTime: DATE,
- gHTeam:NUMBER, 38 sig digits no decimal, teamId type
- gHScore:NUMBER, 38 sig digits no decimal, homeScore type
- gVTeam:NUMBER, 38 sig digits no decimal, teamId type
- gVScore:NUMBER, 38 sig digits no decimal, visitScore type
- gLocationId:NUMBER, 38 sig digits no decimal, locationId type
- gisPlayed:NUMBER, 38 sig digits no decimal, isPlayed type
- gNotes: VARCHAR (20), 20 character, notes type
- gErr: OUT, NUMBER,38 sig digits no decimal, gameId type

Expected Output:

You will get gErr for an output that would describe whether the update was successful or not. A positive number for gErr means success and a negative number means some error.

Potential Error:

- When gamesId trying to be updated when the gamesId does not exist.
- isAct is not in the range of 0 or 1.
- Updating wrong values, for example trying to insert characters in a number parameter.

Purpose:

It is made for software to contact Oracle to update a Player’s information in the database.

Non-saved procedural code:

```
DECLARE
     gErr games.gamesId%type;
BEGIN
     spGamesUpdate(123,2,3,To_date('12-01-2022',
          'dd-mm-yyyy'),212, 0, 300,
          0, 11, 0, NULL, gErr);
     IF gErr > 0 THEN
          DBMS_OUTPUT.PUT_LINE('Update Success!');
     ELSIF gErr = -1 THEN
          DBMS_OUTPUT.PUT_LINE('ERROR: Wrong values!');
     ELSIF gErr = -2 THEN
          DBMS_OUTPUT.PUT_LINE('ERROR: Others!');
     ELSIF gErr = -3 THEN
          DBMS_OUTPUT.PUT_LINE('ERROR: No rows updated!');
     END IF;
END;
```

spGamesDelete

Parameters would be:

- gId: NUMBER, 38 sig digits no decimal, gamesId typed type
- gErr: OUT, NUMBER,38 sig digits no decimal, gamesId type

Expected Output:

You will get gErr for an output that would describe whether the delete was successful or not. A positive number for gErr means success and a negative number means some error.

Potential Error:

- When the same gameId trying to be deleted when the gameId does not exist.
- Inserting wrong values, for example trying to insert characters in a number parameter.

Purpose:

It is made for software to contact Oracle to delete a Game's information in the database.

Non-saved procedural code:

```
DECLARE
     gErr games.gamesId%type;
BEGIN
     spGamesDelete(123, gErr);
     IF gErr > 0 THEN
          DBMS_OUTPUT.PUT_LINE('Delete Success!');
     ELSIF gErr = -1 THEN
          DBMS_OUTPUT.PUT_LINE('ERROR: Wrong values!');
     ELSIF gErr = -2 THEN
          DBMS_OUTPUT.PUT_LINE('ERROR: Others!');
     ELSIF gErr = -3 THEN
          DBMS_OUTPUT.PUT_LINE('ERROR: No rows delete!');
     END IF;
END;
```

spGamesSelect

Parameters would be:

- gId: NUMBER, 38 sig digits no decimal, gameId type
- dID: OUT, NUMBER, 38 sig digits no decimal, divId type
- gNum: OUT, NUMBER, 38 sig digits no decimal, gameNum type
- gDateTime: OUT, DATE,
- gHTeam: OUT, NUMBER, 38 sig digits no decimal, teamId type
- gHScore: OUT,NUMBER, 38 sig digits no decimal, homeScore type
- gVTeamOUT,:NUMBER, 38 sig digits no decimal, teamId type
- gVScore: OUT,NUMBER, 38 sig digits no decimal, visitScore type
- gLocationId: OUT,NUMBER, 38 sig digits no decimal, locationId type
- gisPlayed: OUT,NUMBER, 38 sig digits no decimal, isPlayed type
- gNotes: OUT,VARCHAR (20), 20 character, notes type
- gErr: OUT, NUMBER,38 sig digits no decimal, gameId type

Expected Output:

You will get gErr for an output that would describe whether the SELECT INTO was successful or not. A positive number for gErr means success and a negative number means some error.

Potential Error:

- When gameId trying to be selected for output when a gameId does not exist or
- Too many games exist.
- Insert wrong values, for example trying to insert characters in a number parameter.

Purpose:

It is made for software to contact Oracle to get a Game’s information in the database

Non-saved procedural code:

```
DECLARE
     gErr games.gameId%type;
     gId games.gameId%type := 123;
     dId games.divId%type;
     gNum games.gameNum%type;
     gDateTime games.DateTime%type;
     gHTeam games.homeTeam%type;
     gVteam games.visitTeam%type;
     gHscore games.homeScore%type;
     gVscore games.visitScore%type;
     gLocationId games.locationID%type;
     gisPlayed games.isPlayed%type;
     gNotes games.notes%type;

BEGIN
     BEGIN
     spGamesSelect(gId,dId,gNum,gDateTime,gHteam, gHscore,
      gVteam, gVscore, gLocationId, gisPlayed, gNotes,
      gErr);
IF gErr > 0 THEN
      DBMS_OUTPUT.PUT_LINE(‘Game ID: ’ || gId);
      DBMS_OUTPUT.PUT_LINE(‘Div ID: ’ ||dId);
      DBMS_OUTPUT.PUT_LINE(‘Game Number ’ ||gNum);
      DBMS_OUTPUT.PUT_LINE(‘Date Time: ’ || gDateTime);
      DBMS_OUTPUT.PUT_LINE(‘Home Team: ’ || gHteam);
      DBMS_OUTPUT.PUT_LINE(‘Visitor Team: ’ || gVteam);
      DBMS_OUTPUT.PUT_LINE(‘Home Score: ’ || gHScore);
      DBMS_OUTPUT.PUT_LINE(‘Visitor Score: ’ ||
gVScore);
      DBMS_OUTPUT.PUT_LINE(‘Location Id: ’ ||
gLocationId);
      DBMS_OUTPUT.PUT_LINE(‘Notes: ’ || gNotes);
ELSIF gErr = -1 THEN
      DBMS_OUTPUT.PUT_LINE(‘ERROR: No data found!’);
ELSIF gErr = -2 THEN
      DBMS_OUTPUT.PUT_LINE(‘ERROR: Too many rows!’);
ELSIF tErr = -3 THEN
      DBMS_OUTPUT.PUT_LINE(‘ERROR: Others!’);
END IF;
END;
```

spGamesOutput

Parameter:
NA

Expected Output:
the procedure will display all games’ information. If any error occurs, it will display what kind of error happened.

Potential Errors:

- Rowtype mismatch when fetching cursor’s row and inserting in variable.
- If no data was found, display “No data found!”.

Purpose:
It is used to get information on games. It will display every field of every row of games table.

Non-saved procedural code:

```
BEGIN
     spGamesOutput;
END;
```

spPlayersOutput

Parameter:
NA

Expected Output:
the procedure will display all player’ information. If any error occurs, it will display what kind of error happened.

Potential Errors:

- Rowtype mismatch when fetching cursor’s row and inserting in variable.
- If no data was found, display “No data found!”.

Purpose:
It is used to get information on players. It will display every field of every row of player's table.

Non-saved procedural code:

```
BEGIN
     spPlayersOutput;
END;
```

spTeamsOutput

Parameter:
NA

Expected Output:

The procedure will display all teams' information. If any error occurs, it will display what kind of error happened.

Potential Errors:

- Rowtype mismatch when fetching cursor's row and inserting in variable.
- If no data was found, display “No data found!”.

Purpose:

It is used to get information on teams. It will display every field of every row of teams's table.

Non-saved procedural code:

```
BEGIN
     spTeamsOutput;
END;
```

spGoalScorersOutput

Parameter:
NA

Expected Output:

The procedure will display all goalScorer' information. If any error occurs, it will display what kind of error happened.

Potential Errors:

- Rowtype mismatch when fetching cursor's row and inserting in variable.
- If no data was found, display "No data found!".

Purpose:

It is used to get information on goalScorer. It will display every field of every row of goalScorer's table.

Non-saved procedural code:

```
BEGIN
     spGoalScorersOutput;
END;
```

spRosterOutput

Parameter:
NA

Expected Output:

The procedure will display all Roster' information. If any error occurs, it will display what kind of error happened.

Potential Errors:

- Rowtype mismatch when fetching cursor's row and inserting in variable.
- If no data was found, display "No data found!".

Purpose:

It is used to get information on Roster. It will display every field of every row of Roster's table.

Non-saved procedural code:

```
BEGIN
     spRosterOutput;
END;
```

spSlLocationsOutput

Parameter:
NA

Expected Output:
the procedure will display all slLocations' information. If any error occurs, it will display what kind of error happened.

Potential Errors:

- Rowtype mismatch when fetching cursor's row and inserting in variable.
- If no data was found, display "No data found!".

Purpose:
It is used to get information on slLocations. It will display every field of every row of slLocations's table.

Non-saved procedural code:

```
BEGIN
     spSlLocationsOutput;
END;
```

vwPlayerRosters

Parameter: N/A

Expected Output: N/A

Potential Errors: N/A

Purpose:

It is not an object but a view that is used in the database. It uses fields from players, rosters and team to make a bulkier table, which consists of all of this table's information, which is used in objects of our database.

Non-saved procedural code: N/A

spTeamRosterById

Parameter:

- tId: NUMBER, 38 sig digits no decimal, teamId type in the teams table

Expected Output:

With a valid tId input, the procedure will display all players' information that belongs to the teamId (tId). If any error occurs, it will display what kind of error happened.

Potential Errors:

- Rowtype mismatch when fetching cursor's row and inserting in a variable.
- If no data was found, display "No data found!".

Purpose:

It is used to get information on players in a specific Team by using teamId. It will display team rosters, with names for a teamId.

Non-saved procedural code:

```
BEGIN
     spTeamRosterById(123);
END;
```

spTeamRosterByName

Parameter:

- tName: VARCHAR2(10), 10 characters, teamName type in the teams table

Expected Output:

With a valid tName input, the procedure will display all players’ information that belongs to a similar teamName or part of teamName(tName). If any error occurs, it will display what kind of error happened.

Potential Errors:

- Rowtype mismatch when fetching cursor’s row and inserting in variable.
- If no data was found (not an error but it will show no data found).
- Wrong datatype input for the parameter.

Purpose:

It is used to get information on players in a specific Team by using teamName. It will display team rosters, with names for a similar teamName or part of teamName.

Non-saved procedural code:

```
BEGIN
     spTeamRosterByName(‘son’);
END;
```

vwTeamsNumPlayers

Parameter: N/A

Expected Output: N/A

Potential Errors: N/A

Purpose:
It is not an object but a view that is used in the database. It uses fields from players and teams to make a bulkier table, which consists of all of this table's information, which is used in objects of our database.

Non-saved procedural code: N/A

fncNumPlayersByTeamID

Parameter:

- tId: NUMBER, 38 sig digits no decimal, teamId type

Expected Output:

It will return the number of players that exist in a team. The function's parameter is of teamId which then counts the number of players in that teamId. If it returns negative values -1,-2, or -3, that means it has errors and will display what errors have occurred and if it is a positive value, that is the number of players in that team.

Potential Errors:

- If there is no team for the parameter teamId, then there will be no data found.
- If there are too many teams for the parameter teamId, then there will be too many rows

Purpose:

It is used to get the number of players using teamid. This function will accept team id as a parameter and will further search for the same in the vwTeamsNumPlayers view, and will display results if no error is found.

Non-saved procedural code:

```
Declare
     numPlayers INT;
BEGIN
     numPlayers = fncNumPlayersByTeamId(212);
     IF numPlayers > 0 THEN
          DBMS_OUTPUT.PUT_LINE('Number of Players are: ' ||
               numPlayers);
     ELSIF numPlayers = -1 THEN
          DBMS_OUTPUT.PUT_LINE('ERROR: No data found!');
     ELSIF numPlayers = -2 THEN
          DBMS_OUTPUT.PUT_LINE('ERROR: Too many Rows!');
     ELSIF numPlayers = -3 THEN
          DBMS_OUTPUT.PUT_LINE('ERROR: Others!');
     END IF;
END;
```

vwSchedule

Parameter:
NA

Expected Output:
NA

Potential Errors:
NA

Purpose:
It is not an object but a view that is used in the database. It is using three tables, teams, slLocations ,and games to make a bulkier table, which consists of all of these table's information, which is used in objects of our database.

Non-saved procedural code:
NA

spSchedUpcomingGames

Parameter:

- numDay: INT, Integer, -2,147,483,648 to 2,147,483,647, to have a max range of days for scheduled games (upcoming).

Expected Output:

To have a full display of games that will be happening in the future. If numDay input is out of scope or no games will be played during the numDay, then it will show No data found.

Potential Errors:

- If wrong input for numDay was given, for example trying to input character to an INT datatype
- If no data was found (not an error but it will show no data found).
- Rowtype mismatch when fetching cursor's row and inserting in the variable.

Purpose:

The procedure will give information on the games that will be happening during the range of date given in the parameter. This gives a report with all the fields in the vwScheduled view.

Non-saved procedural code:

```
BEGIN
    spSchedUpcomingGames(10);
END;
```

spSchedPastGames

Parameter:

- numDay: INT, Integer, -2,147,483,648 to 2,147,483,647, to have a max range of days for scheduled games (past).

Expected Output:

To have a full display of games that have already happened in the past. If numDay input is out of scope or no games were played during the numDay, then it will show No data found.

Potential Errors:

- If wrong input for numDay was given, for example trying to input character to an INT datatype
- If no data was found (not an error but it will show no data found).
- Rowtype mismatch when fetching cursor's row and inserting in the variable.

Purpose:

The procedure will give information on the games that happened during the range of dates given in the parameter. This gives a report with all the fields in the vwScheduled view.

Non-saved procedural code:

```
BEGIN
    spSchedPastGames(10);
END;
```

fncTopScorer

Parameter: N/A

Expected Output:

This function will output a playerId of the top scoring player in the league.

Potential Errors:

- If no data was found from the SELECT INTO statement
- If too many rows were found from the SELECT INTO statement

Purpose:

This function's purpose is to give playerId when called so if the top scorer's information is required in a SELECT statement, playerId from this function can be used to get all the corresponding data related to the player.

Non-saved procedural code:

```
DECLARE
     errP players.playerId%type;
     pId players.playerId%type := fncTopScorer;
     rNum players.regNumber%type;
     lName players.lastName%type;
     fName players.firstName%type;
     isAct players.isActive%type;
BEGIN
     BEGIN
     spPlayersSelect(pId, rNum, lName, fName, isAct, errP);
     IF errP > 0 THEN
          DBMS_OUTPUT.PUT_LINE('Player ID: ' || pId);
          DBMS_OUTPUT.PUT_LINE('Player Reg Num: ' || rNum);
          DBMS_OUTPUT.PUT_LINE('Player Last Name: ' ||
          lName);
          DBMS_OUTPUT.PUT_LINE('Player First Name: ' ||
          fName);
          DBMS_OUTPUT.PUT_LINE('Player Is Active: ' ||
          isAct);
     ELSIF errP = -1 THEN
          DBMS_OUTPUT.PUT_LINE('ERROR: No data found!');
     ELSIF errP = -2 THEN
          DBMS_OUTPUT.PUT_LINE('ERROR: Too many rows!');
     ELSIF errP = -3 THEN
          DBMS_OUTPUT.PUT_LINE('ERROR: Others!');
     END IF;
END;
```