

```

SQL> @Team_PKG;
SQL> -- Chet Michals
SQL> create or replace PACKAGE TEAM_pkg AS -- Package Definition problem 1
  2     name_in_use EXCEPTION;
  3     team_not_found EXCEPTION;
  4     TYPE TeamRecord IS RECORD(Team_ID integer, Tname varchar(15), Rank
integer, City varchar(20), Championships integer);
  5     CURSOR TeamCursor RETURN TeamRecord;
  6     FUNCTION add(TeamName varchar, TeamCity varchar) RETURN integer;
  7     FUNCTION remove(TeamID integer) RETURN integer;
  8     PROCEDURE displayinfo(TeamID integer);
  9     PROCEDURE displayinfo(Teamname varchar);
 10     FUNCTION getTeam(TeamID integer) RETURN TeamRecord;
 11     PROCEDURE displayTeams;
 12     FUNCTION updateCity(TeamID integer, NewCity varchar) RETURN integer;
 13     FUNCTION updateCity(TeamName varchar, NewCity varchar) RETURN integer;
 14     FUNCTION updateName(TeamID integer, NewName varchar) RETURN integer;
 15     FUNCTION updateName(TeamName varchar, NewName varchar) RETURN integer;
 16     FUNCTION updateChampionships(TeamID integer) RETURN integer;
 17     FUNCTION updateChampionships(TeamName varchar) RETURN integer;
 18 END TEAM_pkg;
 19 /

```

Package created.

```

SQL> @Team_PKG_Body;
SQL> create or replace PACKAGE BODY TEAM_pkg AS
  2
  3 CURSOR TeamCursor RETURN TeamRecord is
  4 select TEAM_ID , TNAME , RANK , CITY , CHAMPIONSHIPS  from TEAM order by TEAM_ID
ASC;
  5
  6 FUNCTION add(TeamName varchar, TeamCity varchar) RETURN integer
  7 IS i integer;
  8 BEGIN
  9 select count(*) into i from team where TNAME = TeamName;
 10 if i = 1 then
 11   raise name_in_use;
 12 else
 13   i := TEAM_seq.NEXTVAL;
 14 insert into team (Team_ID, TName, City, Championships) values (i, TeamName,
TeamCity, 0);
 15 RETURN (i);
 16 end if;
 17 END add;
 18
 19     FUNCTION remove(TeamID integer) RETURN integer IS i integer;
 20 BEGIN
 21 delete from TEAM_PLAYER_arena where Team_ID = TeamID;
 22 i := SQL%ROWCOUNT;
 23 delete from TEAM where Team_ID = TeamID;
 24   i := i + SQL%ROWCOUNT;
 25   if i = 0 THEN
 26     return (i);
 27   else

```

```

28     return (TeamID);
29     END IF;
30 END remove;
31
32     PROCEDURE displayinfo(TeamID integer) is i INTEGER; aRecord TeamRecord;
33 BEGIN
34     select count(*) into i from team where Team_ID = TeamID;
35     if i = 0 then
36         raise team_not_found;
37     else
38         select TEAM_ID , TNAME , RANK , CITY ,CHAMPIONSHIPS into aRecord from team where
Team_ID = TeamID;
39         DBMS_OUTPUT.PUT_LINE('Team ID: ' ||aRecord.TEAM_ID|| ' | Team Name: ' ||
aRecord.TNAME || ' | Rank: ' || aRecord.RANK || ' | City: ' ||aRecord.CITY||'|
Championships: ' ||aRecord.CHAMPIONSHIPS);
40     end if;
41     END displayinfo;
42
43
44     PROCEDURE displayinfo(TeamName varchar) is i INTEGER; aRecord TeamRecord;
45 BEGIN
46     select count(*) into i from team where Tname = TeamName;
47     if i = 0 then
48         raise team_not_found;
49     else
50         select TEAM_ID , TNAME , RANK , CITY ,CHAMPIONSHIPS into aRecord from team where
Tname = TeamName;
51         DBMS_OUTPUT.PUT_LINE('Team ID: ' ||aRecord.TEAM_ID|| ' | Team Name: ' ||
aRecord.TNAME || ' | Rank: ' || aRecord.RANK || ' | City: ' ||aRecord.CITY||'|
Championships: ' ||aRecord.CHAMPIONSHIPS);
52     end if;
53     END displayinfo;
54
55
56     FUNCTION getTeam(TeamID integer) RETURN TeamRecord IS aRecord TeamRecord;
57     i integer;
58 BEGIN
59     select count(*) into i from team where Team_ID = TeamID;
60     if i = 0 then
61         raise team_not_found;
62     else
63         select TEAM_ID , TNAME , RANK , CITY , CHAMPIONSHIPS  into aRecord from team
where Team_ID = TeamID;
64         RETURN (aRecord);
65     end if;
66     END getTeam;
67
68     PROCEDURE displayTeams is i INTEGER; aRecord TeamRecord;
69 BEGIN
70     OPEN TeamCursor;
71     LOOP
72         FETCH TeamCursor INTO aRecord;
73         EXIT WHEN TeamCursor%NOTFOUND;

```

```

74     DBMS_OUTPUT.PUT_LINE('Team ID: ' || aRecord.TEAM_ID || ' | Team Name: ' ||
aRecord.TNAME || ' | Rank: ' || aRecord.RANK || ' | City: ' || aRecord.CITY || ' |
Championships: ' || aRecord.CHAMPIONSHIPS);
75     END LOOP;
76     CLOSE TeamCursor;
77     END displayTeams;
78
79
80     FUNCTION updateCity(TeamID integer, NewCity varchar) RETURN integer IS
81 i integer;
82 BEGIN
83 select count(*) into i from team where Team_ID = TeamID;
84 if i = 0 then
85     raise team_not_found;
86 else
87 update team set city = NewCity where Team_ID = TeamID;
88 i := SQL%ROWCOUNT;
89 RETURN (i);
90 end if;
91 END updateCity;
92
93     FUNCTION updateCity(TeamName varchar, NewCity varchar) RETURN integer IS
i integer;
94 BEGIN
95 select count(*) into i from team where TName = TeamName;
96 if i = 0 then
97     raise team_not_found;
98 else
99 update team set city = NewCity where TName = TeamName;
100 i := SQL%ROWCOUNT;
101 RETURN (i);
102 end if;
103 END updateCity;
104
105     FUNCTION updateName(TeamID integer, NewName varchar) RETURN integer IS i
integer;
106 BEGIN
107 select count(*) into i from team where Team_ID = TeamID;
108 if i = 0 then
109     raise team_not_found;
110 else
111 update team set TName = NewName where Team_ID = TeamID;
112 i := SQL%ROWCOUNT;
113 RETURN (i);
114 end if;
115 END updateName;
116
117     FUNCTION updateName(TeamName varchar, NewName varchar) RETURN integer IS
i integer;
118 BEGIN
119 select count(*) into i from team where TName = TeamName;
120 if i = 0 then
121     raise team_not_found;
122 else
123 update team set TName = NewName where TName = TeamName;

```

```

124 i := SQL%ROWCOUNT;
125 RETURN (i);
126 end if;
127 END updateName;
128
129     FUNCTION updateChampionships(TeamID integer) RETURN integer IS i integer;
130 BEGIN
131 select count(*) into i from team where Team_ID = TeamID;
132 if i = 0 then
133   raise team_not_found;
134 else
135 update team set Championships = Championships + 1 where Team_ID = TeamID;
136 select Championships into i from team where Team_ID = TeamID;
137 RETURN (i);
138 end if;
139 END updateChampionships;
140
141     FUNCTION updateChampionships(TeamName varchar) RETURN integer IS i
integer;
142 BEGIN
143 select count(*) into i from team where TName = TeamName;
144 if i = 0 then
145   raise team_not_found;
146 else
147 update team set Championships = Championships + 1 where TName = TeamName;
148 select Championships into i from team where TName = TeamName;
149 RETURN (i);
150 end if;
151 END updateChampionships;
152 END TEAM_pkg;
153 /

```

Package body created.

```

SQL> @Player_PKG;
SQL> -- Chet Michals
SQL> create or replace PACKAGE PLAYER_pkg AS -- Package Definition problem 2
2     invalid_value EXCEPTION;
3     player_not_found EXCEPTION;
4     TYPE PlayerRecord IS RECORD(Player_ID integer, Pname varchar(30),
Position varchar(20), draft_year integer, retire_year integer, Education
varchar(30));
5     FUNCTION add(Pname varchar, Position varchar, draft_year integer,
Education varchar) RETURN integer;
6     FUNCTION retire(PlayerID integer, retireYear integer) RETURN integer;
7     FUNCTION retire(PlayerName varchar, retireYear integer) RETURN integer;
8     FUNCTION getPlayer(PlayerID integer) RETURN PlayerRecord;
9 END PLAYER_pkg;
10 /

```

Package created.

```

SQL> @Player_PKG_Body;
SQL> -- Chet Michals
SQL> -- Problem 2 Body

```

```

SQL>
SQL> create or replace PACKAGE BODY PLAYER_pkg AS
  2     FUNCTION add(Pname varchar, Position varchar, draft_year integer, Education
varchar) RETURN integer
  3     IS ID integer;
  4     playerID INT;
  5     BEGIN
  6         if (Pname IS NULL) OR (Position IS NULL) OR (Education IS NULL) THEN
  7             raise invalid_value;
  8         else
  9             select max(Player_ID) into playerID from PLAYER;
 10             ID := playerID + 1;
 11             insert into PLAYER(Player_ID, Pname, Position, draft_year, Education)
values (ID, Pname, Position, draft_year, Education);
 12             return (ID);
 13         end if;
 14     END add;
 15
 16     FUNCTION retire(PlayerID integer, retireYear integer) RETURN integer IS ID
integer;
 17     BEGIN
 18         select count(*) into ID from PLAYER where PLAYER.Player_ID = PlayerID;
 19         if ID = 0 THEN
 20             raise player_not_found;
 21         ELSE
 22             update PLAYER set retire_year = retireYear where player.Player_ID =
PlayerID;
 23             return (PlayerID);
 24         end if;
 25     END retire;
 26
 27     FUNCTION retire(PlayerName varchar, retireYear integer) RETURN integer
 28     IS
 29     ID integer;
 30
 31     BEGIN
 32         select count(*) into ID from PLAYER where PLAYER.Pname = PlayerName;
 33         if ID = 0 THEN
 34             raise player_not_found;
 35         ELSE
 36             select Player_ID into ID from PLAYER where PLAYER.Pname = PlayerName;
 37             ID := retire(ID,retireYear);
 38             return (ID);
 39         END IF;
 40
 41
 42
 43     END retire;
 44
 45     FUNCTION getPlayer(PlayerID integer) RETURN PlayerRecord    IS theRecord
PlayerRecord; aNumber integer;
 46     BEGIN
 47         select count(*) into aNumber from player where Player_ID = PlayerID;
 48         if aNumber = 0 then
 49             raise player_not_found;

```

```

50     else
51     select * into theRecord from player where Player_ID = PlayerID;
52     return(theRecord);
53     end if;
54     END getPlayer;
55 END PLAYER_pkg;
56 /

```

Package body created.

```

SQL> @Arena_PKG;
SQL> -- Chet Michals
SQL> create or replace PACKAGE ARENA_pkg AS -- Package Definition problem 3
2     invalid_value EXCEPTION;
3     TYPE ArenaRecord IS RECORD(ARENA_ID integer, Aname varchar(20), city
varchar(20));
4     FUNCTION add(Aname varchar, city varchar) RETURN INTEGER;
5     FUNCTION remove(ArenaID integer) RETURN INTEGER;
6     FUNCTION remove(ArenaName varchar) RETURN INTEGER;
7     FUNCTION getArenaByID(ArenaID integer) RETURN ArenaRecord;
8     FUNCTION getArenaByName(ArenaName varchar) RETURN ArenaRecord;
9     FUNCTION getArenaByCity(ArenaCity varchar) RETURN ArenaRecord;
10 END ARENA_pkg;
11 /

```

Package created.

```

SQL> @Arena_PKG_Body;
SQL> -- Chet Michals
SQL> -- Problem 3 Body
SQL>
SQL> create or replace PACKAGE BODY ARENA_pkg AS
2     FUNCTION add(Aname varchar, city varchar) RETURN INTEGER
3     IS ID integer;
4     arenaID INT;
5     BEGIN
6     if (Aname IS NULL) OR (city IS NULL) THEN
7     raise invalid_value;
8     else
9     select max(ARENA_ID) into arenaID from ARENA;
10    ID := arenaID + 1;
11    insert into ARENA(ARENA_ID, ANAME, CITY) values (ID, Aname, city);
12    return (ID);
13    end if;
14    END add;
15
16    FUNCTION remove(ArenaID integer) RETURN INTEGER
17    IS i integer;
18    BEGIN
19    delete from TEAM_PLAYER_ARENA where ARENA_ID = ArenaID;
20    i := SQL%ROWCOUNT;
21    delete from ARENA where ARENA_ID = ArenaID;
22    i := i + SQL%ROWCOUNT;
23    if i = 0 THEN
24    return (i);

```

```

25     else
26     return (ArenaID);
27     END IF;
28     END remove;
29
30     FUNCTION remove(ArenaName varchar) RETURN INTEGER
31     IS i integer;
32     ArenaID integer;
33     BEGIN
34     select ARENA_ID into ArenaID from ARENA where ANAME = ArenaName;
35     delete from TEAM_PLAYER_ARENA where ARENA_ID = ArenaID;
36     i := SQL%ROWCOUNT;
37     delete from ARENA where ARENA_ID = ArenaID;
38     i := i + SQL%ROWCOUNT;
39     if i = 0 THEN
40     return (i);
41     else
42     return (ArenaID);
43     END IF;
44     EXCEPTION
45     when NO_DATA_FOUND then
46     return (0);
47     END remove;
48
49     FUNCTION getArenaByID(ArenaID integer) RETURN ArenaRecord
50     IS theRecord ArenaRecord;
51     i integer;
52     BEGIN
53     select count(*) into i from ARENA where ARENA_ID = ArenaID;
54     if i = 0 then
55     raise invalid_value;
56     else
57     select * into theRecord from ARENA where ARENA_ID = ArenaID;
58     return (theRecord);
59     end if;
60     END getArenaByID;
61
62     FUNCTION getArenaByName(ArenaName varchar) RETURN ArenaRecord
63     IS theRecord ArenaRecord;
64     i integer;
65     BEGIN
66     select count(*) into i from ARENA where ArenaName = AName;
67     if i = 0 then
68     raise invalid_value;
69     else
70     select * into theRecord from ARENA where ArenaName = AName;
71     return (theRecord);
72     end if;
73     return (theRecord);
74     END getArenaByName;
75
76     FUNCTION getArenaByCity(ArenaCity varchar) RETURN ArenaRecord
77     IS theRecord ArenaRecord;
78     i integer;
79     BEGIN

```

```

80  select count(*) into i from ARENA where ArenaCity = city;
81  if i = 0 then
82  raise invalid_value;
83  else
84  select * into theRecord from ARENA where ArenaCity = city;
85  return (theRecord);
86  end if;
87
88  return (theRecord);
89  END getArenaByCity;
90 END ARENA_pkg;
91 /

```

Package body created.

```

SQL> Set serveroutput on;
SQL> SET LIN[ESIZE] 400;
SP2-0268: linesize option not a valid number
SQL> select * from arena;

```

ARENA_ID	ANAME	CITY
1	American Airlines	Miami
2	Moda Center	Portland
3	Staples Center	LA
4	United Center	Chicago
5	TD Garden	Boston
6	ATT Center	San Antonio
7	Philips Arena	Atlanta

7 rows selected.

```
SQL> select * from team;
```

TEAM_ID	TNAME	RANK	CITY	CHAMPIONSHIPS
1	Clippers	6	LA	0
2	Bulls	5	Chicago	6
3	Hornets	9	Charlotte	0
4	Blazers	2	Portland	0
5	Spurs	7	San Antonio	5
6	Mavericks	4	Dallas	1

6 rows selected.

```
SQL> select * from player;
```

PLAYER_ID	PNAME	POSITION	DRAFT_YEAR
1	Chris Paul	Small Forward	2005
	Wake Forest		
2	Tony Parker	Point guard	2001

INSEP

3	Marco Belinelli San Giovanni	Shooting guard	2007
---	---------------------------------	----------------	------

PLAYER_ID	PNAME	POSITION	DRAFT_YEAR
RETIRE_YEAR	EDUCATION		

4	Gary Neal La Salle University	Power Forward	2007
---	----------------------------------	---------------	------

5	Kawhi Leonard San Diego State	Forward	2011
---	----------------------------------	---------	------

6	Patty Mills Marist College	Point guard	2009
---	-------------------------------	-------------	------

PLAYER_ID	PNAME	POSITION	DRAFT_YEAR
RETIRE_YEAR	EDUCATION		

7	Tyson Chandler Dominguez	Center	2001
---	-----------------------------	--------	------

8	Derek Fisher 2014 Arkansas	Shooting guard	1996
---	-------------------------------	----------------	------

8 rows selected.

SQL> select * from team_player_arena;

TEAM_ID	PLAYER_ID	ARENA_ID	POINTS
1	1	1	20
1	1	4	17
2	3	1	32
2	3	2	22
2	3	3	13
2	3	4	25
2	3	5	6
2	3	6	14
2	3	7	18
2	5	2	11
3	3	1	20

TEAM_ID	PLAYER_ID	ARENA_ID	POINTS
3	4	2	5
4	6	3	23
4	6	7	18
5	2	2	24

5	2	4	10
5	5	5	25
5	5	7	16
5	6	2	24
5	1	4	33
5	3	4	31
5	4	4	26

TEAM_ID	PLAYER_ID	ARENA_ID	POINTS
-----	-----	-----	-----
5	5	4	14
5	6	4	5

24 rows selected.

```

SQL> @test;
SQL> -- ARENA PACKAGE TEST
SQL> declare
  2   AName varchar(20) := 'Best Arena';
  3   City varchar(20) := 'Miami';
  4   ArenaID integer(20);
  5   AnArenaRecord ARENA_PKG.ArenaRecord;
  6   i integer;
  7
  8   begin
  9       -- Call the functions in Arena_PKG properly
10       AName := 'Best Arena';
11       City := 'Miami';
12       ArenaID := ARENA_PKG.add('Best Arena', 'Miami');
13       DBMS_OUTPUT.PUT_LINE(ArenaID);
14       i := ARENA_PKG.remove(7);
15       DBMS_OUTPUT.PUT_LINE(i);
16       i := ARENA_PKG.remove('Staples Center');
17       DBMS_OUTPUT.PUT_LINE(i);
18       AnArenaRecord := ARENA_PKG.getArenaByID(ArenaID);
19       DBMS_OUTPUT.PUT_LINE(AnArenaRecord.ARENA_ID || ' ' || AnArenaRecord.Aname || '
'|AnArenaRecord.City);
20       AnArenaRecord := ARENA_PKG.getArenaByName('United Center');
21       DBMS_OUTPUT.PUT_LINE(AnArenaRecord.ARENA_ID || ' ' || AnArenaRecord.Aname || '
'|AnArenaRecord.City);
22       AnArenaRecord := ARENA_PKG.getArenaByCity('Boston');
23       DBMS_OUTPUT.PUT_LINE(AnArenaRecord.ARENA_ID || ' ' || AnArenaRecord.Aname || '
'|AnArenaRecord.City);
24       EXCEPTION
25           WHEN ARENA_PKG.invalid_value THEN
26               raise_application_error(-20004, 'Invalid Value inputed');
27       commit;
28
29   end;
30   /
8
7
3
8 Best Arena Miami
4 United Center Chicago

```

5 TD Garden Boston

PL/SQL procedure successfully completed.

```
SQL>
SQL> -- Call the functions in Arena_PKG improperly
SQL> declare
  2   AName varchar(20) := 'Best Arena';
  3   City varchar(20) := 'Miami';
  4   ArenaID integer(20);
  5   AnArenaRecord ARENA_PKG.ArenaRecord;
  6   i integer;
  7   begin
  8   ArenaID := ARENA_PKG.add('Best Arena', '');
  9   EXCEPTION
 10   WHEN ARENA_PKG.invalid_value THEN
 11       raise_application_error(-20004, 'Invalid Value inputed');
 12   end;
 13   /
declare
*
ERROR at line 1:
ORA-20004: Invalid Value inputed
ORA-06512: at line 11
```

```
SQL> declare
  2   AName varchar(20) := 'Best Arena';
  3   City varchar(20) := 'Miami';
  4   ArenaID integer(20);
  5   AnArenaRecord ARENA_PKG.ArenaRecord;
  6   i integer;
  7   begin
  8   i := ARENA_PKG.remove(-1);
  9   DBMS_OUTPUT.PUT_LINE(i);
 10   i := ARENA_PKG.remove('No Arena');
 11   DBMS_OUTPUT.PUT_LINE(i);
 12   end;
 13   /
0
0
```

PL/SQL procedure successfully completed.

```
SQL> declare
  2   AName varchar(20) := 'Best Arena';
  3   City varchar(20) := 'Miami';
  4   ArenaID integer(20);
  5   AnArenaRecord ARENA_PKG.ArenaRecord;
  6   i integer;
  7   begin
  8   AnArenaRecord := ARENA_PKG.getArenaByID(-1);
  9   EXCEPTION
 10   WHEN ARENA_PKG.invalid_value THEN
 11       raise_application_error(-20004, 'Invalid Value inputed');
```

```

12 end;
13 /
declare
*
ERROR at line 1:
ORA-20004: Invalid Value inputed
ORA-06512: at line 11

```

```

SQL> declare
2   AName varchar(20) := 'Best Arena';
3   City varchar(20) := 'Miami';
4   ArenaID integer(20);
5   AnArenaRecord ARENA_PKG.ArenaRecord;
6   i integer;
7   begin
8   AnArenaRecord := ARENA_PKG.getArenaByName('UnitedCenter');
9   EXCEPTION
10  WHEN ARENA_PKG.invalid_value THEN
11      raise_application_error(-20004, 'Invalid Value inputed');
12  end;
13  /
declare
*
ERROR at line 1:
ORA-20004: Invalid Value inputed
ORA-06512: at line 11

```

```

SQL> declare
2   AName varchar(20) := 'Best Arena';
3   City varchar(20) := 'Miami';
4   ArenaID integer(20);
5   AnArenaRecord ARENA_PKG.ArenaRecord;
6   i integer;
7   begin
8   AnArenaRecord := ARENA_PKG.getArenaByCity('Baston');
9   EXCEPTION
10  WHEN ARENA_PKG.invalid_value THEN
11      raise_application_error(-20004, 'Invalid Value inputed');
12  end;
13  /
declare
*
ERROR at line 1:
ORA-20004: Invalid Value inputed
ORA-06512: at line 11

```

```
SQL> commit;
```

Commit complete.

```

SQL> @test2;
SQL> -- PLAYER PACKAGE TEST

```

```

SQL> declare
  2   AName varchar(20) := 'Smithy';
  3   Posistion varchar(20) := 'Yes';
  4   DraftYear integer(20) := 2015;
  5   Education varchar(20) := 'Some';
  6   APlayerRecord PLAYER_pkg.PlayerRecord;
  7   i integer;
  8   begin
  9       i := PLAYER_pkg.add(AName, Posistion, DraftYear, Education);
 10       DBMS_OUTPUT.PUT_LINE(i);
 11       i := PLAYER_pkg.retire(1,2015);
 12       DBMS_OUTPUT.PUT_LINE(i);
 13       i := PLAYER_pkg.retire('Gary Neal',2015);
 14       DBMS_OUTPUT.PUT_LINE(i);
 15       APlayerRecord := PLAYER_pkg.getPlayer(2);
 16       DBMS_OUTPUT.PUT_LINE(APlayerRecord.Player_ID||'|'|
'|APlayerRecord.Pname||'|'|'|APlayerRecord.Position||'|'|
'|APlayerRecord.draft_year||'|'|'|APlayerRecord.retire_year||'|'|
'|APlayerRecord.Education);
 17       EXCEPTION
 18           WHEN PLAYER_pkg.player_not_found THEN
 19               raise_application_error(-20003, 'Player Not Found');
 20           WHEN PLAYER_pkg.invalid_value THEN
 21               raise_application_error(-20004, 'Invalid Value inputed');
 22
 23   end;
 24   /
9
1
4
2| Tony Parker| Point guard| 2001| | INSEP

```

PL/SQL procedure successfully completed.

```

SQL>
SQL> -- Testing fail cases to Player Package
SQL> declare
  2   AName varchar(20) := 'Smithy';
  3   Posistion varchar(20) := 'Yes';
  4   DraftYear integer(20) := 2015;
  5   Education varchar(20) := 'Some';
  6   APlayerRecord PLAYER_pkg.PlayerRecord;
  7   i integer;
  8   begin
  9       i := PLAYER_pkg.add('', Posistion, DraftYear, Education);
 10       DBMS_OUTPUT.PUT_LINE(i);
 11       EXCEPTION
 12           WHEN PLAYER_pkg.player_not_found THEN
 13               raise_application_error(-20003, 'Player Not Found');
 14           WHEN PLAYER_pkg.invalid_value THEN
 15               raise_application_error(-20004, 'Invalid Value inputed');
 16   end;
 17   /
declare
*
```

ERROR at line 1:
ORA-20004: Invalid Value inputed
ORA-06512: at line 15

```
SQL>
SQL> declare
  2   AName varchar(20) := 'Smithy';
  3   Posistion varchar(20) := 'Yes';
  4   DraftYear integer(20) := 2015;
  5   Education varchar(20) := 'Some';
  6   APlayerRecord PLAYER_pkg.PlayerRecord;
  7   i integer;
  8   begin
  9       i := PLAYER_pkg.retire(-1,2015);
 10       DBMS_OUTPUT.PUT_LINE(i);
 11       EXCEPTION
 12           WHEN PLAYER_pkg.player_not_found THEN
 13               raise_application_error(-20003, 'Player Not Found');
 14           WHEN PLAYER_pkg.invalid_value THEN
 15               raise_application_error(-20004, 'Invalid Value inputed');
 16   end;
 17   /
declare
*
```

ERROR at line 1:
ORA-20003: Player Not Found
ORA-06512: at line 13

```
SQL>
SQL> declare
  2   AName varchar(20) := 'Smithy';
  3   Posistion varchar(20) := 'Yes';
  4   DraftYear integer(20) := 2015;
  5   Education varchar(20) := 'Some';
  6   APlayerRecord PLAYER_pkg.PlayerRecord;
  7   i integer;
  8   begin
  9       i := PLAYER_pkg.retire('No one',2015);
 10       DBMS_OUTPUT.PUT_LINE(i);
 11       EXCEPTION
 12           WHEN PLAYER_pkg.player_not_found THEN
 13               raise_application_error(-20003, 'Player Not Found');
 14           WHEN PLAYER_pkg.invalid_value THEN
 15               raise_application_error(-20004, 'Invalid Value inputed');
 16   end;
 17   /
declare
*
```

ERROR at line 1:
ORA-20003: Player Not Found
ORA-06512: at line 13

```

SQL> declare
  2   AName varchar(20) := 'Smithy';
  3   Posistion varchar(20) := 'Yes';
  4   DraftYear integer(20) := 2015;
  5   Education varchar(20) := 'Some';
  6   APlayerRecord PLAYER_pkg.PlayerRecord;
  7   i integer;
  8   begin
  9   APlayerRecord := PLAYER_pkg.getPlayer(-1);
 10       DBMS_OUTPUT.PUT_LINE(APlayerRecord.Player_ID||'|'
'|APlayerRecord.Pname||'|'|APlayerRecord.Position||'|'
'|APlayerRecord.draft_year||'|'|APlayerRecord.retire_year||'|'
'|APlayerRecord.Education);
 11   EXCEPTION
 12       WHEN PLAYER_pkg.player_not_found THEN
 13           raise_application_error(-20003, 'Player Not Found');
 14       WHEN PLAYER_pkg.invalid_value THEN
 15           raise_application_error(-20004, 'Invalid Value inputed');
 16   end;
 17   /
declare
*
ERROR at line 1:
ORA-20003: Player Not Found
ORA-06512: at line 13

```

```
SQL> commit;
```

Commit complete.

```

SQL> @test3;
SQL> -- Testing Team Package
SQL> declare
  2   TeamID integer(20);
  3   AName varchar(20) := 'Socks';
  4   City varchar(20) := 'Maybe';
  5   ATeamRecord TEAM_pkg.TeamRecord;
  6   i integer;
  7   begin
  8   i:= TEAM_pkg.add(AName, City);
  9   DBMS_OUTPUT.PUT_LINE(i);
 10   i:= TEAM_pkg.remove(1);
 11   DBMS_OUTPUT.PUT_LINE(i);
 12   ATeamRecord := TEAM_pkg.getTeam(2);
 13   DBMS_OUTPUT.PUT_LINE('Team ID: ' ||ATeamRecord.TEAM_ID|| '| Team Name: ' ||
ATeamRecord.TNAME || '| Rank: ' || ATeamRecord.RANK || '| City:
'|ATeamRecord.CITY||'| Championships: ' ||ATeamRecord.CHAMPIONSHIPS);
 14   i:= TEAM_pkg.updateCity(3, 'Nowhere');
 15   DBMS_OUTPUT.PUT_LINE(i);
 16   i:= TEAM_pkg.updateCity('Spurs', 'Dreamland');
 17   DBMS_OUTPUT.PUT_LINE(i);
 18   i:= TEAM_pkg.updateName(5, 'Gloves');
 19   DBMS_OUTPUT.PUT_LINE(i);
 20   i:= TEAM_pkg.updateName('Gloves', 'Spurs');

```

```

21 DBMS_OUTPUT.PUT_LINE(i);
22 i:= TEAM_pkg.updateChampionships(5);
23 DBMS_OUTPUT.PUT_LINE(i);
24 i:= TEAM_pkg.updateChampionships('Spurs');
25 DBMS_OUTPUT.PUT_LINE(i);
26 TEAM_pkg.displayinfo(2);
27 TEAM_pkg.displayinfo('Spurs');
28 TEAM_pkg.displayTeams;
29 EXCEPTION
30     WHEN TEAM_pkg.team_not_found THEN
31         raise_application_error(-20002, 'Team Not Found');
32
33     WHEN TEAM_pkg.name_in_use THEN
34         raise_application_error(-20001, 'Name in Use');
35 end;
36 /
29
1
Team ID: 2| Team Name: Bulls| Rank: 5 | City: Chicago| Championships: 6
1
1
1
1
6
7
Team ID: 2| Team Name: Bulls| Rank: 5 | City: Chicago| Championships: 6
Team ID: 5| Team Name: Spurs| Rank: 7 | City: Dreamland| Championships: 7
Team ID: 2| Team Name: Bulls| Rank: 5 | City: Chicago| Championships: 6
Team ID: 3| Team Name: Hornets| Rank: 9 | City: Nowhere| Championships: 0
Team ID: 4| Team Name: Blazers| Rank: 2 | City: Portland| Championships: 0
Team ID: 5| Team Name: Spurs| Rank: 7 | City: Dreamland| Championships: 7
Team ID: 6| Team Name: Mavericks| Rank: 4 | City: Dallas| Championships: 1
Team ID: 29| Team Name: Socks| Rank:  | City: Maybe| Championships: 0

```

PL/SQL procedure successfully completed.

```

SQL>
SQL> -- Testing fail cases of team Package
SQL>
SQL> declare
2   TeamID integer(20);
3   AName varchar(20) := 'Socks';
4   City varchar(20) := 'Maybe';
5   ATeamRecord TEAM_pkg.TeamRecord;
6   i integer;
7   begin
8   i:= TEAM_pkg.add('Socks',City);
9   EXCEPTION
10      WHEN TEAM_pkg.team_not_found THEN
11          raise_application_error(-20002, 'Team Not Found');
12
13      WHEN TEAM_pkg.name_in_use THEN
14          raise_application_error(-20001, 'Name in Use');
15 end;
16 /

```



```
declare
*
ERROR at line 1:
ORA-20001: Name in Use
ORA-06512: at line 14
```

```
SQL>
SQL> declare
  2   TeamID integer(20);
  3   AName varchar(20) := 'Socks';
  4   City varchar(20) := 'Maybe';
  5   ATeamRecord TEAM_pkg.TeamRecord;
  6   i integer;
  7   begin
  8   i:= TEAM_pkg.remove(-1);
  9   EXCEPTION
 10     WHEN TEAM_pkg.team_not_found THEN
 11       raise_application_error(-20002, 'Team Not Found');
 12
 13     WHEN TEAM_pkg.name_in_use THEN
 14       raise_application_error(-20001, 'Name in Use');
 15   end;
 16   /
```

PL/SQL procedure successfully completed.

```
SQL>
SQL> declare
  2   TeamID integer(20);
  3   AName varchar(20) := 'Socks';
  4   City varchar(20) := 'Maybe';
  5   ATeamRecord TEAM_pkg.TeamRecord;
  6   i integer;
  7   begin
  8   ATeamRecord := TEAM_pkg.getTeam(-2);
  9   EXCEPTION
 10     WHEN TEAM_pkg.team_not_found THEN
 11       raise_application_error(-20002, 'Team Not Found');
 12
 13     WHEN TEAM_pkg.name_in_use THEN
 14       raise_application_error(-20001, 'Name in Use');
 15   end;
 16   /
declare
*
ERROR at line 1:
ORA-20002: Team Not Found
ORA-06512: at line 11
```

```
SQL>
SQL> declare
  2   TeamID integer(20);
  3   AName varchar(20) := 'Socks';
```

```

4   City varchar(20) := 'Maybe';
5   ATeamRecord TEAM_pkg.TeamRecord;
6   i integer;
7   begin
8   i:= TEAM_pkg.updateCity(-3,'Nowhere');
9   EXCEPTION
10      WHEN TEAM_pkg.team_not_found THEN
11          raise_application_error(-20002, 'Team Not Found');
12
13      WHEN TEAM_pkg.name_in_use THEN
14          raise_application_error(-20001, 'Name in Use');
15   end;
16   /
declare
*
ERROR at line 1:
ORA-20002: Team Not Found
ORA-06512: at line 11

```

```

SQL>
SQL> declare
2   TeamID integer(20);
3   AName varchar(20) := 'Socks';
4   City varchar(20) := 'Maybe';
5   ATeamRecord TEAM_pkg.TeamRecord;
6   i integer;
7   begin
8   i:= TEAM_pkg.updateCity('spuds','Dreamland');
9   EXCEPTION
10      WHEN TEAM_pkg.team_not_found THEN
11          raise_application_error(-20002, 'Team Not Found');
12
13      WHEN TEAM_pkg.name_in_use THEN
14          raise_application_error(-20001, 'Name in Use');
15   end;
16   /
declare
*
ERROR at line 1:
ORA-20002: Team Not Found
ORA-06512: at line 11

```

```

SQL>
SQL> declare
2   TeamID integer(20);
3   AName varchar(20) := 'Socks';
4   City varchar(20) := 'Maybe';
5   ATeamRecord TEAM_pkg.TeamRecord;
6   i integer;
7   begin
8   i:= TEAM_pkg.updateName(-5,'Gloves');
9   EXCEPTION
10      WHEN TEAM_pkg.team_not_found THEN

```

```

11         raise_application_error(-20002, 'Team Not Found');
12
13     WHEN TEAM_pkg.name_in_use THEN
14         raise_application_error(-20001, 'Name in Use');
15 end;
16 /
declare
*
ERROR at line 1:
ORA-20002: Team Not Found
ORA-06512: at line 11

```

```

SQL>
SQL> declare
2     TeamID integer(20);
3     AName varchar(20) := 'Socks';
4     City varchar(20) := 'Maybe';
5     ATeamRecord TEAM_pkg.TeamRecord;
6     i integer;
7 begin
8     i:= TEAM_pkg.updateName('Gives','Spurs');
9 EXCEPTION
10     WHEN TEAM_pkg.team_not_found THEN
11         raise_application_error(-20002, 'Team Not Found');
12
13     WHEN TEAM_pkg.name_in_use THEN
14         raise_application_error(-20001, 'Name in Use');
15 end;
16 /
declare
*
ERROR at line 1:
ORA-20002: Team Not Found
ORA-06512: at line 11

```

```

SQL>
SQL> declare
2     TeamID integer(20);
3     AName varchar(20) := 'Socks';
4     City varchar(20) := 'Maybe';
5     ATeamRecord TEAM_pkg.TeamRecord;
6     i integer;
7 begin
8     i:= TEAM_pkg.updateChampionships(-5);
9 EXCEPTION
10     WHEN TEAM_pkg.team_not_found THEN
11         raise_application_error(-20002, 'Team Not Found');
12
13     WHEN TEAM_pkg.name_in_use THEN
14         raise_application_error(-20001, 'Name in Use');
15 end;
16 /
declare

```

```

*
ERROR at line 1:
ORA-20002: Team Not Found
ORA-06512: at line 11

```

```

SQL>
SQL> declare
  2   TeamID integer(20);
  3   AName varchar(20) := 'Socks';
  4   City varchar(20) := 'Maybe';
  5   ATeamRecord TEAM_pkg.TeamRecord;
  6   i integer;
  7   begin
  8   i:= TEAM_pkg.updateChampionships('spins');
  9   EXCEPTION
 10     WHEN TEAM_pkg.team_not_found THEN
 11       raise_application_error(-20002, 'Team Not Found');
 12
 13     WHEN TEAM_pkg.name_in_use THEN
 14       raise_application_error(-20001, 'Name in Use');
 15   end;
 16   /
declare
*

```

```

ERROR at line 1:
ORA-20002: Team Not Found
ORA-06512: at line 11

```

```

SQL>
SQL> commit;

```

Commit complete.

```

SQL> select * from arena;

```

ARENA_ID	ANAME	CITY
1	American Airlines	Miami
2	Moda Center	Portland
4	United Center	Chicago
5	TD Garden	Boston
6	ATT Center	San Antonio
8	Best Arena	Miami

6 rows selected.

```

SQL> select * from team;

```

TEAM_ID	TNAME	RANK	CITY	CHAMPIONSHIPS
2	Bulls	5	Chicago	6
3	Hornets	9	Nowhere	0
4	Blazers	2	Portland	0

5 Spurs	7 Dreamland	7
6 Mavericks	4 Dallas	1
29 Socks	Maybe	0

6 rows selected.

SQL> select * from player;

PLAYER_ID	PNAME	POSITION	DRAFT_YEAR
1	Chris Paul	Small Forward	2005
2	Tony Parker	Point guard	2001
3	Marco Belinelli	Shooting guard	2007
4	Gary Neal	Power Forward	2007
5	Kawhi Leonard	Forward	2011
6	Patty Mills	Point guard	2009
7	Tyson Chandler	Center	2001
8	Derek Fisher	Shooting guard	1996
9	Smithy	Yes	2015

2015 Wake Forest		
------------------	--	--

INSEP		
-------	--	--

San Giovanni		
--------------	--	--

PLAYER_ID	PNAME	POSITION	DRAFT_YEAR
4	Gary Neal	Power Forward	2007
5	Kawhi Leonard	Forward	2011
6	Patty Mills	Point guard	2009
7	Tyson Chandler	Center	2001
8	Derek Fisher	Shooting guard	1996
9	Smithy	Yes	2015

2015 La Salle University		
--------------------------	--	--

San Diego State		
-----------------	--	--

Marist College		
----------------	--	--

PLAYER_ID	PNAME	POSITION	DRAFT_YEAR
7	Tyson Chandler	Center	2001
8	Derek Fisher	Shooting guard	1996
9	Smithy	Yes	2015

Dominguez		
-----------	--	--

2014 Arkansas		
---------------	--	--

Some		
------	--	--

9 rows selected.

SQL> select * from team_player_arena;

TEAM_ID	PLAYER_ID	ARENA_ID	POINTS
---------	-----------	----------	--------

2	3	1	32
2	3	2	22
2	3	4	25
2	3	5	6
2	3	6	14
2	5	2	11
3	3	1	20
3	4	2	5
5	2	2	24
5	2	4	10
5	5	5	25

TEAM_ID	PLAYER_ID	ARENA_ID	POINTS
5	6	2	24
5	1	4	33
5	3	4	31
5	4	4	26
5	5	4	14
5	6	4	5

17 rows selected.

SQL> SPOOL off;

Tables initially:

```
cmichals@oraclelinux:~  
87  
88     return (theRecord);  
89     END getArenaByCity;  
90 END ARENA_pkg;  
91 /  
  
Package body created.  
  
SQL> Set serveroutput on;  
SQL> SET LIN[ESIZE] 400;  
SP2-0268: linesize option not a valid number  
SQL> select * from arena;  
  
ARENA_ID ANAME CITY  
-----  
1 American Airlines Miami  
2 Moda Center Portland  
3 Staples Center LA  
4 United Center Chicago  
5 TD Garden Boston  
6 ATT Center San Antonio  
7 Philips Arena Atlanta  
  
7 rows selected.  
  
SQL> select * from team;  
  
TEAM_ID TNAME RANK CITY CHAMPIONSHIPS  
-----  
1 Clippers 6 LA 0  
2 Bulls 5 Chicago 6  
3 Hornets 9 Charlotte 0  
4 Blazers 2 Portland 0  
5 Spurs 7 San Antonio 5  
6 Mavericks 4 Dallas 1  
  
6 rows selected.  
  
SQL> select * from player;  
  
PLAYER_ID PNAME POSITION DRAFT_YEAR  
-----  
RETIRE_YEAR EDUCATION  
-----  
1 Chris Paul Small Forward 2005  
Wake Forest  
2 Tony Parker Point guard 2001  
INSEP  
3 Marco Belinelli Shooting guard 2007  
San Giovanni  
  
PLAYER_ID PNAME POSITION DRAFT_YEAR  
-----  
RETIRE_YEAR EDUCATION  
-----  
4 Gary Neal Power Forward 2007  
La Salle University  
5 Kawhi Leonard Forward 2011
```

```
San Diego State
6 Patty Mills          Point guard          2009
  Marist College

PLAYER_ID PNAME          POSITION          DRAFT_YEAR
-----
RETIRE_YEAR EDUCATION
-----
7 Tyson Chandler      Center          2001
  Dominguez
8 Derek Fisher        Shooting guard   1996
  2014 Arkansas

8 rows selected.

SQL> select * from team_player_arena;

  TEAM_ID  PLAYER_ID  ARENA_ID  POINTS
-----
1          1          1          20
1          1          4          17
2          3          1          32
2          3          2          22
2          3          3          13
2          3          4          25
2          3          5           6
2          3          6          14
2          3          7          18
2          5          2          11
3          3          1          20

  TEAM_ID  PLAYER_ID  ARENA_ID  POINTS
-----
3          4          2           5
4          6          3          23
4          6          7          18
5          2          2          24
5          2          4          10
5          5          5          25
5          5          7          16
5          6          2          24
5          1          4          33
5          3          4          31
5          4          4          26

  TEAM_ID  PLAYER_ID  ARENA_ID  POINTS
-----
5          5          4          14
5          6          4           5

24 rows selected.

SQL> @test;
SQL> -- ARENA PACKAGE TEST
SQL> declare
2   AName varchar(20) := 'Best Arena';
3   City varchar(20) := 'Miami';
4   ArenaID integer(20);
```


Tables after scripts run:

```
cmichals@oraclelinux:~  
SQL>  
SQL> commit;  
  
Commit complete.  
  
SQL> select * from arena;  
  
  ARENA_ID ANAME          CITY  
-----  
1 American Airlines    Miami  
2 Moda Center          Portland  
4 United Center         Chicago  
5 TD Garden             Boston  
6 ATT Center            San Antonio  
8 Best Arena            Miami  
  
6 rows selected.  
  
SQL> select * from team;  
  
  TEAM_ID TNAME          RANK CITY          CHAMPIONSHIPS  
-----  
2 Bulls           5 Chicago          6  
3 Hornets          9 Nowhere          0  
4 Blazers          2 Portland          0  
5 Spurs            7 Dreamland        7  
6 Mavericks        4 Dallas            1  
29 Socks           Maybe              0  
  
6 rows selected.  
  
SQL> select * from player;  
  
  PLAYER_ID PNAME          POSITION          DRAFT_YEAR  
-----  
1 Chris Paul      Small Forward    2005  
2015 Wake Forest  
  
2 Tony Parker      Point guard      2001  
INSEP  
  
3 Marco Belinelli  Shooting guard    2007  
San Giovanni  
  
  PLAYER_ID PNAME          POSITION          DRAFT_YEAR  
-----  
4 Gary Neal        Power Forward     2007  
2015 La Salle University  
  
5 Kawhi Leonard    Forward           2011  
San Diego State  
  
6 Patty Mills      Point guard       2009  
Marist College
```

```
San Giovanni

PLAYER_ID PNAME                POSITION                DRAFT_YEAR
-----
RETIRE_YEAR EDUCATION
-----
4 Gary Neal                Power Forward                2007
2015 La Salle University

5 Kawhi Leonard            Forward                2011
San Diego State

6 Patty Mills              Point guard                2009
Marist College

PLAYER_ID PNAME                POSITION                DRAFT_YEAR
-----
RETIRE_YEAR EDUCATION
-----
7 Tyson Chandler           Center                2001
Dominguez

8 Derek Fisher             Shooting guard            1996
2014 Arkansas

9 Smithy                   Yes                2015
Some

9 rows selected.

SQL> select * from team_player_arena;

TEAM_ID  PLAYER_ID  ARENA_ID  POINTS
-----
2         3          1         32
2         3          2         22
2         3          4         25
2         3          5          6
2         3          6         14
2         5          2         11
3         3          1         20
3         4          2          5
5         2          2         24
5         2          4         10
5         5          5         25

TEAM_ID  PLAYER_ID  ARENA_ID  POINTS
-----
5         6          2         24
5         1          4         33
5         3          4         31
5         4          4         26
5         5          4         14
5         6          4          5

17 rows selected.

SQL> SPOOL off;
SQL>
```

Main test of Arena package

```
cmichals@oraclelinux:~  
5      6      4      5  
24 rows selected.  
SQL> @test;  
SQL> -- ARENA PACKAGE TEST  
SQL> declare  
2   AName varchar(20) := 'Best Arena';  
3   City varchar(20) := 'Miami';  
4   ArenaID integer(20);  
5   AnArenaRecord ARENA_PKG.ArenaRecord;  
6   i integer;  
7  
8 begin  
9   -- Call the functions in Arena_PKG properly  
10  AName := 'Best Arena';  
11  City := 'Miami';  
12  ArenaID := ARENA_PKG.add('Best Arena', 'Miami');  
13  DBMS_OUTPUT.PUT_LINE(ArenaID);  
14  i := ARENA_PKG.remove(7);  
15  DBMS_OUTPUT.PUT_LINE(i);  
16  i := ARENA_PKG.remove('Staples Center');  
17  DBMS_OUTPUT.PUT_LINE(i);  
18  AnArenaRecord := ARENA_PKG.getArenaByID(ArenaID);  
19  DBMS_OUTPUT.PUT_LINE(AnArenaRecord.ARENA_ID||' '||AnArenaRecord.AName||' '||AnArenaRecord.City);  
20  AnArenaRecord := ARENA_PKG.getArenaByName('United Center');  
21  DBMS_OUTPUT.PUT_LINE(AnArenaRecord.ARENA_ID||' '||AnArenaRecord.AName||' '||AnArenaRecord.City);  
22  AnArenaRecord := ARENA_PKG.getArenaByCity('Boston');  
23  DBMS_OUTPUT.PUT_LINE(AnArenaRecord.ARENA_ID||' '||AnArenaRecord.AName||' '||AnArenaRecord.City);  
24  EXCEPTION  
25      WHEN ARENA_PKG.invalid_value THEN  
26          raise_application_error(-20004, 'Invalid Value inputed');  
27      commit;  
28  
29 end;  
30 /  
8  
7  
3  
8 Best Arena Miami  
4 United Center Chicago  
5 TD Garden Boston  
  
PL/SQL procedure successfully completed.  
  
SQL>  
SQL> -- Call the functions in Arena_PKG improperly  
SQL> declare  
2   AName varchar(20) := 'Best Arena';  
3   City varchar(20) := 'Miami';  
4   ArenaID integer(20);  
5   AnArenaRecord ARENA_PKG.ArenaRecord;  
6   i integer;  
7 begin  
8   ArenaID := ARENA_PKG.add('Best Arena', '');  
9   EXCEPTION  
10  WHEN ARENA_PKG.invalid_value THEN  
11      raise_application_error(-20004, 'Invalid Value inputed');  
12 end;  
13 /  
declare  
*
```

Main test of Player Package

```
cmichals@oraclelinux:~  
10  WHEN ARENA_PKG.invalid_value THEN  
11      raise_application_error(-20004, 'Invalid Value inputed');  
12  end;  
13  /  
declare  
*  
ERROR at line 1:  
ORA-20004: Invalid Value inputed  
ORA-06512: at line 11  
  
SQL> commit;  
  
Commit complete.  
  
SQL> @test2;  
SQL> -- PLAYER PACKAGE TEST  
SQL> declare  
2      AName varchar(20) := 'Smithy';  
3      Posistion varchar(20) := 'Yes';  
4      DraftYear integer(20) := 2015;  
5      Education varchar(20) := 'Some';  
6      APlayerRecord PLAYER_pkg.PlayerRecord;  
7      i integer;  
8  begin  
9      i := PLAYER_pkg.add(AName, Posistion, DraftYear, Education);  
10     DBMS_OUTPUT.PUT_LINE(i);  
11     i := PLAYER_pkg.retire(1,2015);  
12     DBMS_OUTPUT.PUT_LINE(i);  
13     i := PLAYER_pkg.retire('Gary Neal',2015);  
14     DBMS_OUTPUT.PUT_LINE(i);  
15     APlayerRecord := PLAYER_pkg.getPlayer(2);  
16     DBMS_OUTPUT.PUT_LINE(APlayerRecord.Player_ID||' '||APlayerRecord.Pname||' '||APlayerRecord.Position||' '||APlayerRec  
ord.draft_year||' '||APlayerRecord.retire_year||' '||APlayerRecord.Education);  
17     EXCEPTION  
18         WHEN PLAYER_pkg.player_not_found THEN  
19             raise_application_error(-20003, 'Player Not Found');  
20         WHEN PLAYER_pkg.invalid_value THEN  
21             raise_application_error(-20004, 'Invalid Value inputed');  
22  
23  end;  
24  /  
9  
1  
4  
2| Tony Parker| Point guard| 2001| | INSEP  
  
PL/SQL procedure successfully completed.  
  
SQL>  
SQL> -- Testing fail cases to Player Package  
SQL> declare  
2      AName varchar(20) := 'Smithy';  
3      Posistion varchar(20) := 'Yes';  
4      DraftYear integer(20) := 2015;  
5      Education varchar(20) := 'Some';  
6      APlayerRecord PLAYER_pkg.PlayerRecord;  
7      i integer;  
8  begin  
9      i := PLAYER_pkg.add('', Posistion, DraftYear, Education);  
10     DBMS_OUTPUT.PUT_LINE(i);  
11     EXCEPTION
```

Main test of Team Package:

```
cmichals@oraclelinux:~  
16 end;  
17 /  
declare  
*  
ERROR at line 1:  
ORA-20003: Player Not Found  
ORA-06512: at line 13  
  
SQL> commit;  
  
Commit complete.  
  
SQL> @test3;  
SQL> -- Testing Team Package  
SQL> declare  
2 TeamID integer(20);  
3 AName varchar(20) := 'Socks';  
4 City varchar(20) := 'Maybe';  
5 ATeamRecord TEAM_pkg.TeamRecord;  
6 i integer;  
7 begin  
8 i:= TEAM_pkg.add(AName, City);  
9 DBMS_OUTPUT.PUT_LINE(i);  
10 i:= TEAM_pkg.remove(i);  
11 DBMS_OUTPUT.PUT_LINE(i);  
12 ATeamRecord := TEAM_pkg.getTeam(2);  
13 DBMS_OUTPUT.PUT_LINE('Team ID: ' || ATeamRecord.TEAM_ID || ' Team Name: ' || ATeamRecord.TNAME || ' Rank: ' || ATeamRecord.RANK |  
| ' City: ' || ATeamRecord.CITY || ' Championships: ' || ATeamRecord.CHAMPIONSHIPS);  
14 i:= TEAM_pkg.updateCity(3, 'Nowhere');  
15 DBMS_OUTPUT.PUT_LINE(i);  
16 i:= TEAM_pkg.updateCity('Spurs', 'Dreamland');  
17 DBMS_OUTPUT.PUT_LINE(i);  
18 i:= TEAM_pkg.updateName(5, 'Gloves');  
19 DBMS_OUTPUT.PUT_LINE(i);  
20 i:= TEAM_pkg.updateName('Gloves', 'Spurs');  
21 DBMS_OUTPUT.PUT_LINE(i);  
22 i:= TEAM_pkg.updateChampionships(5);  
23 DBMS_OUTPUT.PUT_LINE(i);  
24 i:= TEAM_pkg.updateChampionships('Spurs');  
25 DBMS_OUTPUT.PUT_LINE(i);  
26 TEAM_pkg.displayinfo(2);  
27 TEAM_pkg.displayinfo('Spurs');  
28 TEAM_pkg.displayTeams;  
29 EXCEPTION  
30 WHEN TEAM_pkg.team_not_found THEN  
31 raise_application_error(-20002, 'Team Not Found');  
32  
33 WHEN TEAM_pkg.name_in_use THEN  
34 raise_application_error(-20001, 'Name in Use');  
35 end;  
36 /  
29  
1  
Team ID: 2| Team Name: Bulls| Rank: 5 | City: Chicago| Championships: 6  
1  
1  
1  
1  
6  
7  
Team ID: 2| Team Name: Bulls| Rank: 5 | City: Chicago| Championships: 6
```