

Stored Procedure

--best attacker

```
Create type out1 as(
player_id integer,
player_position varchar(2),
goal_assist integer
);
Create or Replace Function Best_Attacker()
Returns out1 AS $$
Declare
    r record;
    x out1;
    z out1;
    y plays_on%rowtype;
    sum integer:=0;
    sum1 integer:=0;
Begin
    for r in select* from((Select ASSISTED_BY,Count(*) as TotalAssist from
Goals
where assisted_by is not null group by ASSISTED_BY) as r1
full outer join
(Select SCORED_BY,Count(*) as TotalGoals from Goals
group by SCORED_BY) as r2
on r1.ASSISTED_BY = r2.SCORED_BY)
LOOP
    sum1 :=0;
    if r.scored_by is NULL then
        sum1 := r.totalassist;
        x.player_id = r.assisted_by;
    end if;
    if r.assisted_by is NULL then
        sum1 := r.totalgoals;
        x.player_id = r.scored_by;
    end if;
    if r.scored_by is NOT NULL then
        if r.assisted_by is NOT NULL then
            sum1:= (r.totalassist) + (r.totalgoals);
            x.player_id = r.scored_by;
        end if;
    end if;
    x.goal_assist = sum1;
    for y in Select* from plays_on
    Loop
        if(x.player_id = y.player_id) then
            if(y.pid = 'LW') then
                if(x.goal_assist > sum) then
                    sum = x.goal_assist;
                    x.player_position = y.pid;
                    z = x;
                end if;
            end if;
            if(y.pid = 'RW') then
```

```

Stored Procedure
    if(x.goal_assist > sum) then
        sum = x.goal_assist;
        x.player_position = y.pid;
        z = x;
    end if;
end if;
if(y.pid = 'CF') then
    if(x.goal_assist > sum) then
        sum = x.goal_assist;
        x.player_position = y.pid;
        z = x;
    end if;
end if;
if(y.pid = 'SS') then
    if(x.goal_assist > sum) then
        sum = x.goal_assist;
        x.player_position = y.pid;
        z = x;
    end if;
end if;
end if;
end loop;
Return z;
end $$ LANGUAGE plpgsql;
select* from BEST_Attacker();

```

--golden boot

```

Create type out2 as(
Player_id integer,
Total_Goals integer
);
Create or Replace Function Golden_Boot()
Returns setof out2 AS $$
Declare
    r record;
    v out2;
Begin
    for r in Select* from(Select SCORED_BY,Count(*) as TotalGoals from Goals
    group by SCORED_BY order by TotalGoals Desc) as r1
    Loop
        v.player_id = r.SCORED_BY;

```

```

                                Stored Procedure
                                v.Total_Goals = r.TotalGoals;
                                return next v;
                        end loop;
                Return;
end $$ LANGUAGE plpgsql;
Select* from Golden_Boot();

--most valuable player

Create or Replace Function Most_Valuable_Player()
Returns record AS $$
Declare
    r record;
    x record;
    sum integer:=0;
    sum1 integer:=0;
Begin
    for r in select* from((Select ASSISTED_BY,Count(*) as TotalAssist from
Goals
where assisted_by is not null group by ASSISTED_BY) as r1
full outer join
(Select SCORED_BY,Count(*) as TotalGoals from Goals
group by SCORED_BY) as r2
on r1.ASSISTED_BY = r2.SCORED_BY)
LOOP
        sum1 :=0;
        if r.scored_by is NULL then
            sum1 := r.totalassist*2;
        end if;
        if r.assisted_by is NULL then
            sum1 := r.totalgoals*4;
        end if;
        if r.scored_by is NOT NULL then
            if r.assisted_by is NOT NULL then
                sum1:= (r.totalassist*2) + (r.totalgoals*4);
            end if;
        end if;
        if(sum1>sum) then
            x = r;
            sum = sum1;
        end if;
    end loop;
    x.scored_by = x.totalgoals;
    x.totalgoals = sum;
    Return x;
end $$ LANGUAGE plpgsql;
Select* from Most_Valuable_Player()
AS foo(player_id int,totalassist bigint,totalgoals int,points bigint);

```

Stored Procedure

--point table

Drop Table POINT_TABLE Cascade;

```
CREATE TABLE Point_Table(  
TEAM_NAME VARCHAR(50) PRIMARY KEY,  
MATCHES_PLAYED SMALLINT NOT NULL,  
WON SMALLINT NOT NULL,  
DRAW SMALLINT NOT NULL,  
LOST SMALLINT NOT NULL,  
GOAL_FOR SMALLINT NOT NULL,  
GOAL_AGAINST SMALLINT NOT NULL,  
POINTS INT NOT NULL,  
POSITION INT NOT NULL,  
GOAL_DIFF INT);
```

Create or Replace Function Point_T()

Returns setof Point_Table AS \$\$

Declare

```
t team%rowtype;  
b Point_Table%rowtype;  
c Point_Table%rowtype;  
x integer := 0;
```

Begin

```
for t in Select* from team order by points DESC
```

```
Loop
```

```
x := x+1;
```

```
b.TEAM_NAME = t.TEAM_NAME;
```

```
b.MATCHES_PLAYED = t.MATCHES_PLAYED;
```

```
b.WON = t.WON;
```

```
b.DRAW = t.DRAW;
```

```
b.LOST = t.LOST;
```

```
b.GOAL_FOR = t.GOAL_FOR;
```

```
b.GOAL_AGAINST = t.GOAL_AGAINST;
```

```
b.Points = t.points;
```

```
b.POSITION = x;
```

```
b.GOAL_DIFF := t.GOAL_FOR - t.GOAL_AGAINST;
```

```
INSERT INTO POINT_TABLE
```

```
VALUES(b.TEAM_NAME,b.MATCHES_PLAYED,b.WON,b.DRAW,b.LOST,b.GOAL_FOR,b.GOAL_AGAINS  
T,b.points,b.POSITION,b.GOAL_DIFF);
```

```
end loop;
```

```
for c in Select* from POINT_TABLE order by POINTS DESC,GOAL_DIFF DESC
```

```
Loop
```

```
return next c;
```

```
end LOOP;
```

```
return;
```

```
end $$ LANGUAGE plpgsql;
```