1) Most valuable player(Stored Procedure)

```
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
        here 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;
```

Running This Query:-

```
Select* from Most_Valuable_Player() AS foo(player_id int,totalassist
bigint,totalgoals int,points bigint);
```

2) Total goals or assist of all players

```
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)
```

3) Percentage of winning away from home

```
select away_team, count(*)*10 as winpercentage from match
where score_a>score_h group by away_team order by winpercentage DESC
```

4) Most substituted player

```
select player_id_out, count (*)*1 as no_of_time_substituted from substitution
group by player_id_out order by no_of_time_substituted DESC LIMIT 1;
```

5) Squad Rotation By Managers

```
select name, count(*) as number_of_substitution
from((select name,team_name from manager*team) as r1
natural join
(select team_name,player_id from substitution join player on
 substitution.player_id_in = player.player_id)as r2)
group by name order by number_of_substitution desc
```

6) Referee Punishing the most number of players

```
select r.name, r.ref_licence_number,count(*) as no_of_bookings from has_referee
as hs
natural join referee as r
natural join booking as b where hs.ref_type = 'MAIN'
group by r.name,r.ref_licence_number order by no_of_bookings desc
```

7) Best Attacker(Stored Procedure)

```
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
                                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;
        end loop;
        Return z;
end $$ LANGUAGE plpgsql;

Running This Query:-
select* from BEST_Attacker();


8) Most goals scored from any position

select pid, count(goal_id)as total_score from goals join
(select pid, player_id from position natural join plays_on) as t1
on (scored_by = player_id) where(goal_type!='O')
group by pid order by total_score DESC;

9) Most Goals by a substituted Player(Super Sub)

select player_id_in, count(goal_id) as scored from substitution join goals on
(scored_by = player_id_in)
group by player_id_in order by scored DESC LIMIT 1;

10) Young Player Of the Year

select player_id, player_name, age, count(goal_id)as scored from player join
goals on (player_id = scored_by)
 where (age <22)
group by (player_id, Player_name, age) order by scored DESC;

11) Favourite Venue of every Player

select count(scored_by) as number_of_goals,venue_name from (goals natural join
matches)
join (team natural join venue)
on home_team=team_name group by venue_name;

12) which country's performance is best

select nationality, count(goal_id) as scored from goals join player on
(scored_by = player_id)
group by nationality order by scored DESC;

13) Points Table(Stored Procedure)

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;

Running This Query:-

Select* from POINT_T();
```