**Stored Procedure to find Best Batsman of the match and its run

```
create Type my_type AS (name varchar(20) , runs integer);

create or replace function best_score(match_n integer) returns my_type  as $BODY$

declare

        ans my_type;

begin

        select player_name,run into ans.name, ans.runs from ((select sum(runs) as run,player_name
from ball_by_ball_batting where match_no=match_n and ball_event='l' group by player_name) as a join

  (select max(run) as mx from (select sum(runs) as run,player_name from ball_by_ball_batting where
match_no=match_n and ball_event='l' group by player_name) as c) as b on a.run = b.mx) as d;

  return ans;

end

$BODY$ LANGUAGE plpgsql;

select best_score(5);
```

**Stored Procedure to find innings of particular match and team. We have made this as procedure because we were needed to find innings of particular team from its name only in particular match at many places

```
create or replace function find_inning(matchno integer,Team varchar(3))

returns integer as $BODY$

declare

        player varchar(20);

  ans integer;

  i ball_by_ball_batting%rowtype;

  j players%rowtype;
```

```
begin
       for i in select * from ball_by_ball_batting where match_no = matchno
  loop
       for j in select * from players
    loop
      if (i.player_name = j.player_name and j.team_id = Team) then
        ans := i.inning;
      end if;
    end loop;
  end loop;
  return ans;
end
$BODY$ language plpgsql;
select find_inning(1,'IND');
---------------------------------------------------------------------------------------------
```

**Stored Procedure to find highest wicket taker of particular match and of particular team

```
create or replace function Highest_Wickets(matchno integer,Team varchar(3))
returns setof varchar(20) as $BODY$
declare
       innings integer;
  var integer;
  i varchar(20);
begin
       select find_inning(matchno,Team) into innings;
```

```
   if innings = 1 then

              innings := 2;

   else

           innings := 1;

   end if;

   for i in select bowler from((select bowler,count(ball_event) from (select * from((select player_name as bowler,over_no from ball_by_ball_bowling where (match_no = matchno and inning = innings)) as a join

    (select runs,player_name as batsman,over_no,ball_event from ball_by_ball_batting where ( match_no = matchno and inning = innings)) as b on a.over_no = b.over_no) as c) as d where ball_event = 'w' group by bowler) as f

    join

   (select max(count) as mx from(select bowler,count(ball_event) from (select * from((select player_name as bowler,over_no from ball_by_ball_bowling where (match_no = matchno and inning = innings)) as a join

    (select runs,player_name as batsman,over_no,ball_event from ball_by_ball_batting where ( match_no = matchno and inning = innings)) as b on a.over_no = b.over_no) as c) as d where ball_event = 'w' group by bowler) as e) as g on f.count = g.mx) as h

    loop

           return next i;

    end loop;

    return;

end

$BODY$ language plpgsql;

table matches;

select Highest_Wickets(2,'SA');
```

**Stored procedure to find winner of particular match

```
create or replace function winner_of_match(matchno integer) returns varchar(20) as $BODY$

declare

            team1 varchar(3);

   team2 varchar(3);

   inn1 integer;

   inn2 integer;

   run1 integer;

   run2 integer;

   tie varchar(20);

   ans varchar(20);

begin

            select team_1,team_2 into team1,team2 from matches where match_no = matchno;

   select sum(runs) into run1 from ball_by_ball_batting natural join players where match_no=matchno and team_id=team1;

   select sum(runs) into run2 from ball_by_ball_batting natural join players where match_no=matchno and team_id=team2;

   if run1 > run2 then

            ans := team1;

   elsif run1 < run2 then

            ans := team2;

   else

                        ans := 'Match Tied';

            end if;

   return ans;

end

$BODY$ language plpgsql;
```