-- stored procedure to give number of medals (gold and silver) by a country in any given commonwealth games -------------------------------------------------------------------------------------------------

create or replace function medals(y integer, cntry varchar(25)) returns table(year integer, country varchar(25), gold integer, silver integer) as $BODY$

DECLARE

event integer;

count integer := 0;

count1 integer := 0;

BEGIN

for event in select r3.event\_id from (select r1.year,r1.event\_id,r1.team\_id,r1.record\_id,r2.result from (select r.year,r.event\_id,r.round\_no,r.sub\_round\_no,r.team\_id,e.number\_of\_rounds,r.record\_id from registered\_in as r join event as e on(e.event\_id = r.event\_id)

where r.round\_no = e.number\_of\_rounds) as r1 join result as r2 on (r1.record\_id = r2.record\_id) where r2.result = 1) as r3

join team as t on (t.team\_id = r3.team\_id) where r3.year = y and t.country = cntry

loop

count := count + 1;

end loop;

for event in select r3.event\_id from (select r1.year,r1.event\_id,r1.team\_id,r1.record\_id,r2.result from (select r.year,r.event\_id,r.round\_no,r.sub\_round\_no,r.team\_id,e.number\_of\_rounds,r.record\_id from registered\_in as r join event as e on(e.event\_id = r.event\_id)

where r.round\_no = e.number\_of\_rounds) as r1 join result as r2 on (r1.record\_id = r2.record\_id) where r2.result = 2) as r3

join team as t on (t.team\_id = r3.team\_id) where r3.year = y and t.country = cntry

loop

count1 := count1 + 1;

end loop;

return query values(y,cntry,count,count1);

end;

$BODY$ language plpgsql;

select \* from medals(2012,'England');

-- stored procedure to create final medal tally of commonwealth games in given year -------------------------------------------------------------------------

create or replace function medals\_1(y integer) returns table(year integer, country varchar(25), Gold integer, Silver integer) as $BODY$

DECLARE

event integer;

count integer := 0;

count1 integer := 0;

cntry varchar(25);

BEGIN

for cntry in select distinct t.country from registered\_in as r join team as t on(t.team\_id = r.team\_id)

loop

count:=0;

count1:=0;

for event in select r3.event\_id from (select r1.year,r1.event\_id,r1.team\_id,r1.record\_id,r2.result from (select r.year,r.event\_id,r.round\_no,r.sub\_round\_no,r.team\_id,e.number\_of\_rounds,r.record\_id from registered\_in as r join event as e on(e.event\_id = r.event\_id)

where r.round\_no = e.number\_of\_rounds) as r1 join result as r2 on (r1.record\_id = r2.record\_id) where r2.result = 1) as r3

join team as t on (t.team\_id = r3.team\_id) where r3.year = y and t.country = cntry

loop

count := count + 1;

end loop;

for event in select r3.event\_id from (select r1.year,r1.event\_id,r1.team\_id,r1.record\_id,r2.result from (select r.year,r.event\_id,r.round\_no,r.sub\_round\_no,r.team\_id,e.number\_of\_rounds,r.record\_id from registered\_in as r join event as e on(e.event\_id = r.event\_id)

where r.round\_no = e.number\_of\_rounds) as r1 join result as r2 on (r1.record\_id = r2.record\_id) where r2.result = 2) as r3

join team as t on (t.team\_id = r3.team\_id) where r3.year = y and t.country = cntry

loop

count1 := count1 + 1;

end loop;

return query values(y,cntry,count,count1);

end loop;

return;

end;

$BODY$ language plpgsql;

select \* from medals\_1(2012) order by gold desc;

-------stored procedure to show number of medals won by a country in all the commonwealth games it participated in-------------

create or replace function medals\_2(cntry varchar(25)) returns table(year integer, country varchar(25), Gold integer, Silver integer) as $BODY$

DECLARE

event integer;

count integer := 0;

count1 integer := 0;

y integer;

BEGIN

for y in select s.year from Summit as s

loop

count:=0;

count1:=0;

for event in select r3.event\_id from (select r1.year,r1.event\_id,r1.team\_id,r1.record\_id,r2.result from (select r.year,r.event\_id,r.round\_no,r.sub\_round\_no,r.team\_id,e.number\_of\_rounds,r.record\_id from registered\_in as r join event as e on(e.event\_id = r.event\_id)

where r.round\_no = e.number\_of\_rounds) as r1 join result as r2 on (r1.record\_id = r2.record\_id) where r2.result = 1) as r3

join team as t on (t.team\_id = r3.team\_id) where r3.year = y and t.country = cntry

loop

count := count + 1;

end loop;

for event in select r3.event\_id from (select r1.year,r1.event\_id,r1.team\_id,r1.record\_id,r2.result from (select r.year,r.event\_id,r.round\_no,r.sub\_round\_no,r.team\_id,e.number\_of\_rounds,r.record\_id from registered\_in as r join event as e on(e.event\_id = r.event\_id)

where r.round\_no = e.number\_of\_rounds) as r1 join result as r2 on (r1.record\_id = r2.record\_id) where r2.result = 2) as r3

join team as t on (t.team\_id = r3.team\_id) where r3.year = y and t.country = cntry

loop

count1 := count1 + 1;

end loop;

return query values(y,cntry,count,count1);

end loop;

return;

end;

$BODY$ language plpgsql;

select \* from medals\_2('India');

-- stored procedure to show the details of all the sports a country participated in -----------------------------------------------------------------------------------

create or replace function medals\_3(y integer , cntry varchar(25)) returns table(year integer, country varchar(25),event varchar(25) ,Gold integer, Silver integer) as $BODY$

DECLARE

evnt integer;

evnt1 varchar(25);

count integer := 0;

count1 integer := 0;

rcrd integer;

BEGIN

for evnt in select h.event\_id from held\_in as h where h.year = y

loop

count:=0;

count1:=0;

for rcrd in select r3.event\_id from (select r2.year,r2.event\_id,r2.round\_no,r2.sub\_round\_no,r2.team\_id,t.country,r2.result,r2.record\_id from

(select \* from registered\_in as r natural join result as r1 where r.year = y) as r2 join team as t on(t.team\_id = r2.team\_id)) as r3

join event as e on(e.event\_id = r3.event\_id) where r3.round\_no = e.number\_of\_rounds and r3.country = cntry and result = 1 and r3.event\_id = evnt

loop

count := count + 1;

end loop;

for rcrd in select r3.event\_id from (select r2.year,r2.event\_id,r2.round\_no,r2.sub\_round\_no,r2.team\_id,t.country,r2.result,r2.record\_id from

(select \* from registered\_in as r natural join result as r1 where r.year = y) as r2 join team as t on(t.team\_id = r2.team\_id)) as r3

join event as e on(e.event\_id = r3.event\_id) where r3.round\_no = e.number\_of\_rounds and r3.country = cntry and result = 2 and r3.event\_id = evnt

loop

count1 := count1 + 1;

end loop;

select e.name into evnt1 from event as e where e.event\_id = evnt;

return query values(y,cntry,evnt1,count,count1);

end loop;

return;

end;

$BODY$ language plpgsql;

select \* from medals\_3(2012,'India');

-- Gives all the opponents of a given team\_id in given year with timestamp and location of there competiton -----------------------------------------------------------------------------------------------------------

create or replace function competiton(t\_id integer, yr integer) returns table(opponent\_team\_id integer, event\_id integer, location varchar(25), time\_stamp timestamp)

as $BODY$

DECLARE

tmp\_year integer;

tmp\_event\_id integer;

tmp\_round\_no integer;

tmp\_sub\_round\_no integer;

t\_op\_team\_id integer;

t\_location varchar(25);

t\_time\_stamp timestamp;

t\_event\_name varchar(25);

t\_event\_id integer;

BEGIN

for tmp\_year,tmp\_event\_id,tmp\_round\_no,tmp\_sub\_round\_no in

select year,r.event\_id,round\_no,sub\_round\_no from registered\_in natural join round as r where year = yr and team\_id = t\_id

loop

for t\_op\_team\_id,t\_location,t\_time\_stamp,t\_event\_id in select team\_id,r1.location,r1.time\_stamp,r1.event\_id from registered\_in natural join round as r1

where year = tmp\_year and r1.event\_id = tmp\_event\_id and round\_no = tmp\_round\_no and sub\_round\_no = tmp\_sub\_round\_no

loop

if(t\_op\_team\_id != t\_id) then

select name into t\_event\_name from event as e where e.event\_id = t\_event\_id;

return query values(t\_op\_team\_id,t\_event\_id,t\_location,t\_time\_stamp);

end if;

end loop;

end loop;

end;

$BODY$ language plpgsql;

select \* from competiton(19,2012);