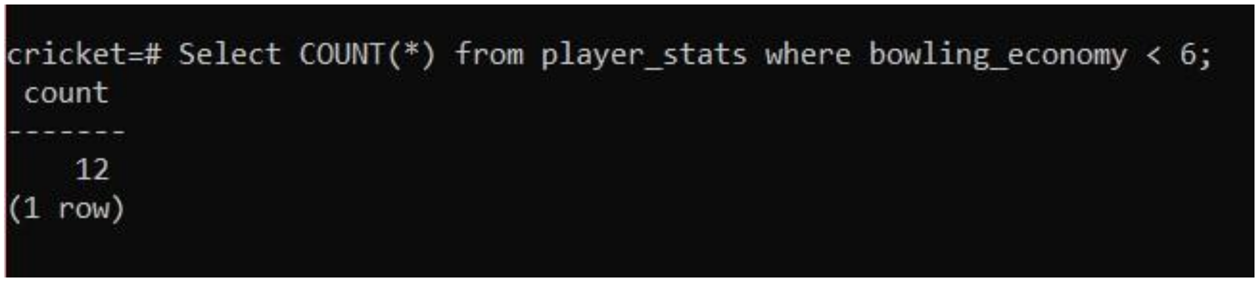
Queries

1. Query to view number of players who have below 6 bowling economy

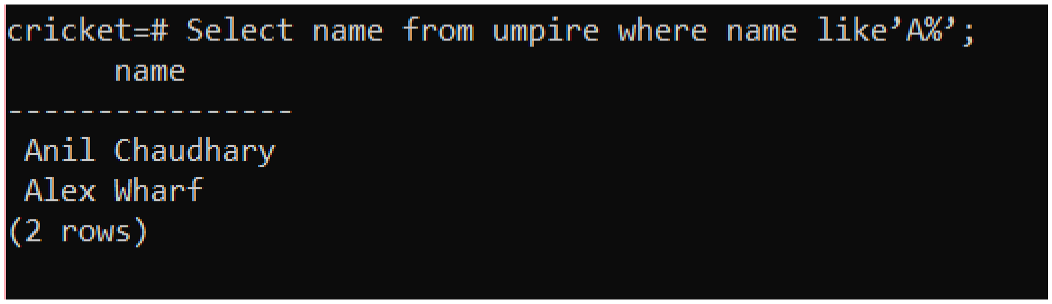
Select COUNT(*) from player_stats where bowling_economy < 6;



```
cricket=# Select COUNT(*) from player_stats where bowling_economy < 6;
 count
-------
    12
(1 row)
```

2. Name all the umpires whose names start with letter ‘A’

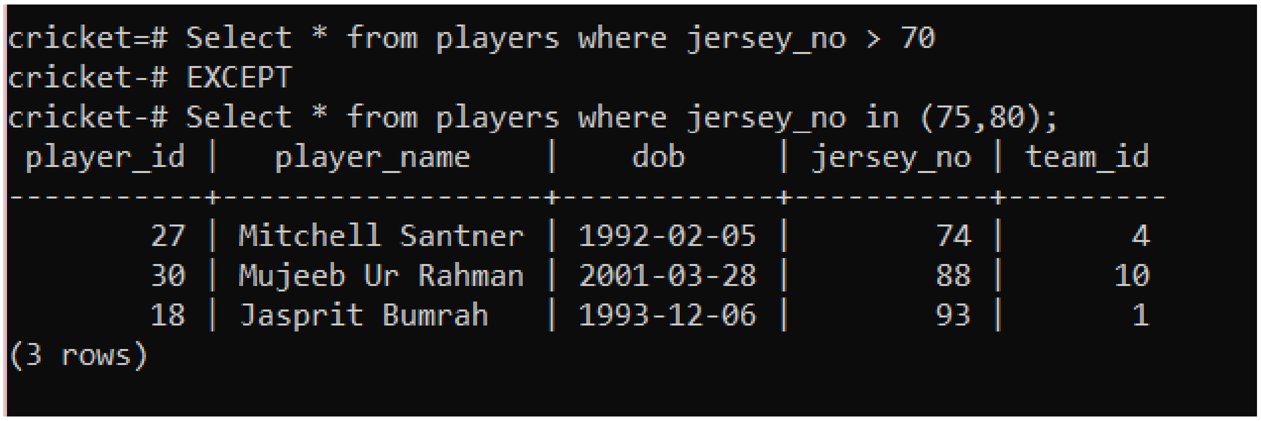
Select name from umpire where name like’A%’;



```
cricket=# Select name from umpire where name like’A%’;
      name
----------------
 Anil Chaudhary
 Alex Wharf
(2 rows)
```

3. Query to view player jersey numbers above 70 but not 75,80

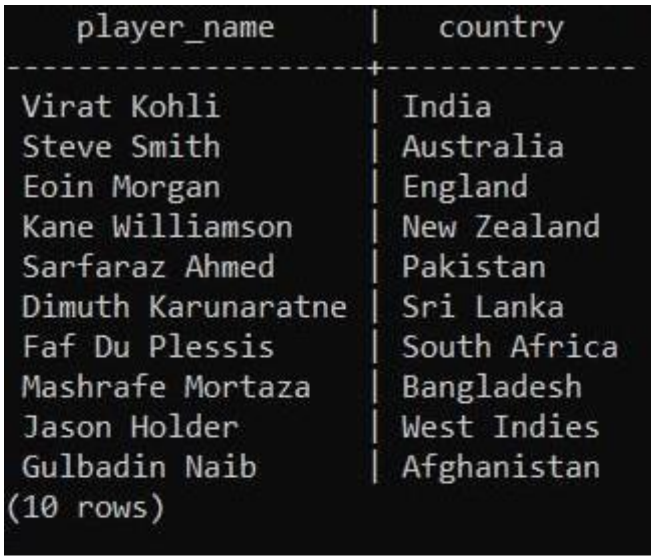
Select * from players where jersey_no > 70
EXCEPT
Select * from players where jersey_no in (75,80);



```
cricket=# Select * from players where jersey_no > 70
cricket-# EXCEPT
cricket-# Select * from players where jersey_no in (75,80);
 player_id |   player_name    |    dob     | jersey_no | team_id
-----------+------------------+------------+-----------+---------
        27 | Mitchell Santner | 1992-02-05 |        74 |       4
        30 | Mujeeb Ur Rahman | 2001-03-28 |        88 |      10
        18 | Jasprit Bumrah   | 1993-12-06 |        93 |       1
(3 rows)
```

4. Display name of captains for their respective country

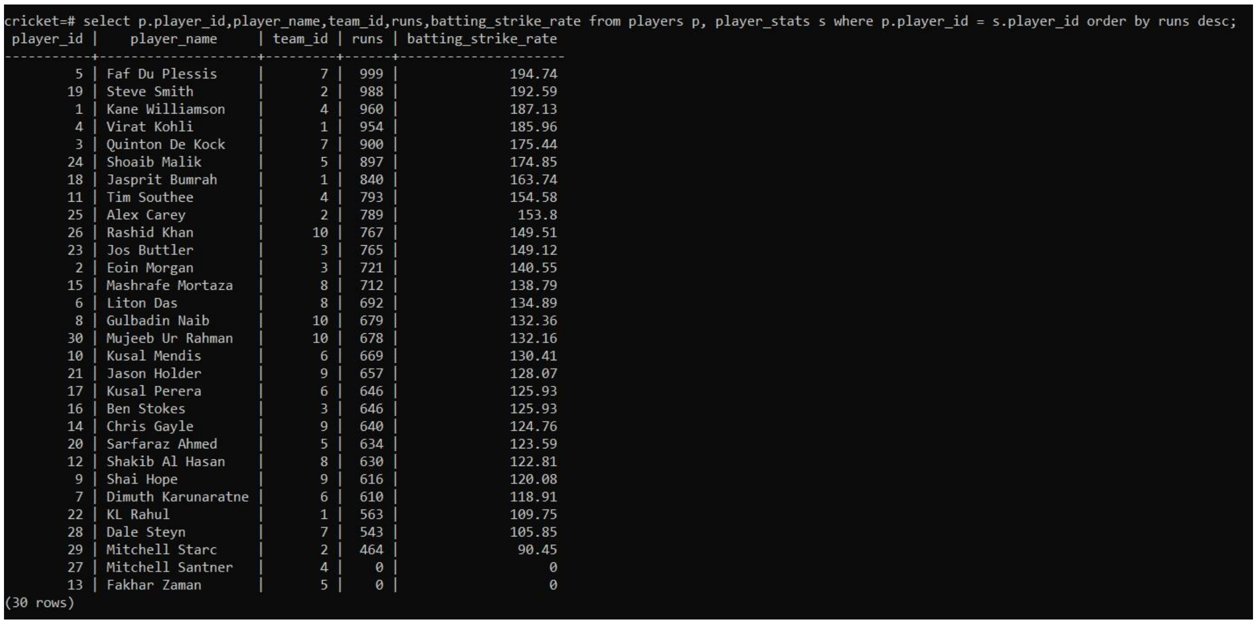
select player_name,country from players p, team_authority t,teams te where p.player_id = t.captain_id and p.team_id = te.team_id;



```
    player_name     |   country
--------------------+--------------
 Virat Kohli        | India
 Steve Smith        | Australia
 Eoin Morgan        | England
 Kane Williamson    | New Zealand
 Sarfaraz Ahmed     | Pakistan
 Dimuth Karunaratne | Sri Lanka
 Faf Du Plessis     | South Africa
 Mashrafe Mortaza   | Bangladesh
 Jason Holder       | West Indies
 Gulbadin Naib      | Afghanistan
(10 rows)
```

5. Display playerid,name,team_id, runs and batting strike rate in order of highest number of runs

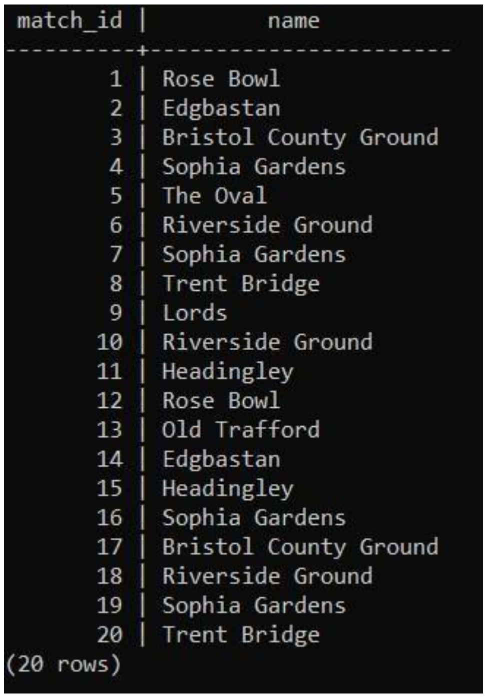
select p.player_id,player_name,team_id,runs,batting_strike_rate from players p, player_stats s where p.player_id = s.player_id order by runs desc;



```
cricket=# select p.player_id,player_name,team_id,runs,batting_strike_rate from players p, player_stats s where p.player_id = s.player_id order by runs desc;
 player_id |    player_name     | team_id | runs | batting_strike_rate
-----------+--------------------+---------+------+---------------------
         5 | Faf Du Plessis     |       7 |  999 |              194.74
        19 | Steve Smith        |       2 |  988 |              192.59
         1 | Kane Williamson    |       4 |  960 |              187.13
         4 | Virat Kohli        |       1 |  954 |              185.96
         3 | Quinton De Kock    |       7 |  900 |              175.44
        24 | Shoaib Malik       |       5 |  897 |              174.85
        18 | Jasprit Bumrah     |       1 |  840 |              163.74
        11 | Tim Southee        |       4 |  793 |              154.58
        25 | Alex Carey         |       2 |  789 |               153.8
        26 | Rashid Khan        |      10 |  767 |              149.51
        23 | Jos Buttler        |       3 |  765 |              149.12
         2 | Eoin Morgan        |       3 |  721 |              140.55
        15 | Mashrafe Mortaza   |       8 |  712 |              138.79
         6 | Liton Das          |       8 |  692 |              134.89
         8 | Gulbadin Naib      |      10 |  679 |              132.36
        30 | Mujeeb Ur Rahman   |      10 |  678 |              132.16
        10 | Kusal Mendis       |       6 |  669 |              130.41
        21 | Jason Holder       |       9 |  657 |              128.07
        17 | Kusal Perera       |       6 |  646 |              125.93
        16 | Ben Stokes         |       3 |  646 |              125.93
        14 | Chris Gayle        |       9 |  640 |              124.76
        20 | Sarfaraz Ahmed     |       5 |  634 |              123.59
        12 | Shakib Al Hasan    |       8 |  630 |              122.81
         9 | Shai Hope          |       9 |  616 |              120.08
         7 | Dimuth Karunaratne |       6 |  610 |              118.91
        22 | KL Rahul           |       1 |  563 |              109.75
        28 | Dale Steyn         |       7 |  543 |              105.85
        29 | Mitchell Starc     |       2 |  464 |               90.45
        27 | Mitchell Santner   |       4 |    0 |                   0
        13 | Fakhar Zaman       |       5 |    0 |                   0
(30 rows)
```

6. Display match number and ground it has been played on

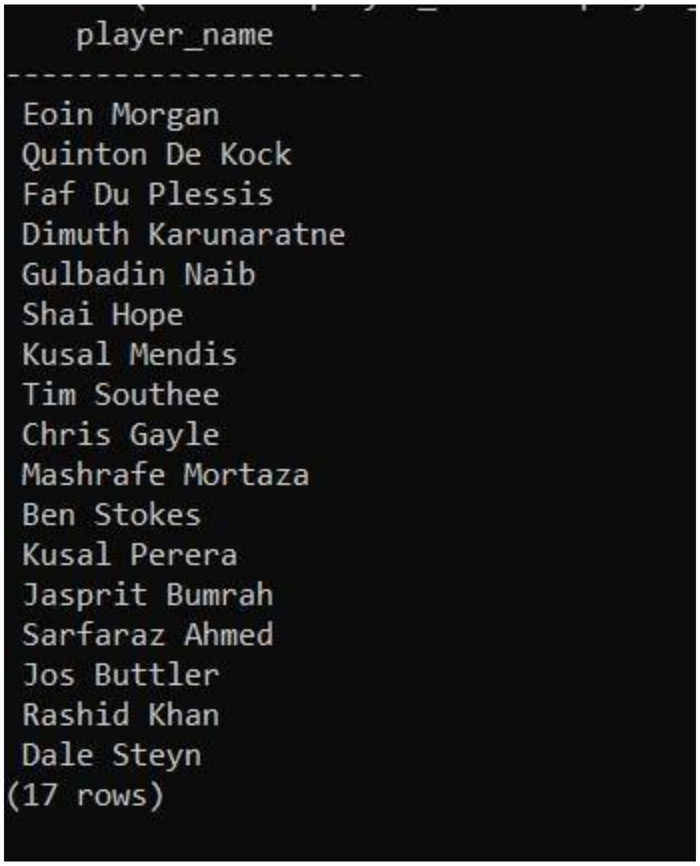
select match_id,name from match inner join ground on ground.ground_id = match.ground_id;



```
 match_id |         name
----------+-----------------------
        1 | Rose Bowl
        2 | Edgbastan
        3 | Bristol County Ground
        4 | Sophia Gardens
        5 | The Oval
        6 | Riverside Ground
        7 | Sophia Gardens
        8 | Trent Bridge
        9 | Lords
       10 | Riverside Ground
       11 | Headingley
       12 | Rose Bowl
       13 | Old Trafford
       14 | Edgbastan
       15 | Headingley
       16 | Sophia Gardens
       17 | Bristol County Ground
       18 | Riverside Ground
       19 | Sophia Gardens
       20 | Trent Bridge
(20 rows)
```

7. Retrieve all players names whose bowling economy is greater than total average

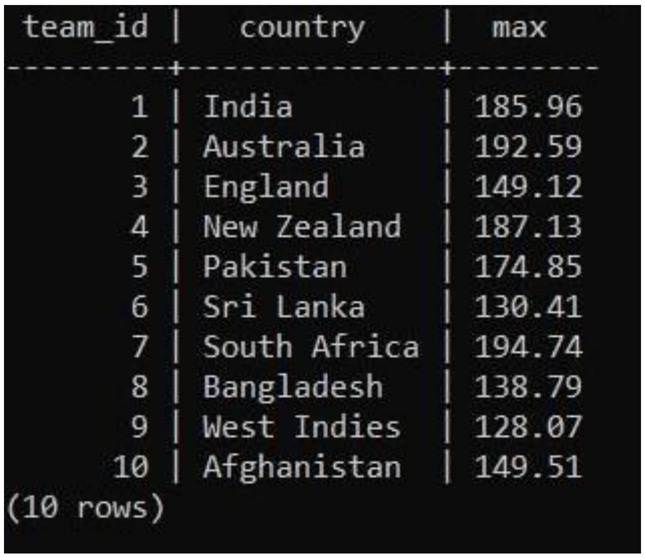
select player_name from players where player_id in (select player_id from player_stats where bowling_economy >= (Select AVG(bowling_economy) from player_stats));



```
    player_name
--------------------
 Eoin Morgan
 Quinton De Kock
 Faf Du Plessis
 Dimuth Karunaratne
 Gulbadin Naib
 Shai Hope
 Kusal Mendis
 Tim Southee
 Chris Gayle
 Mashrafe Mortaza
 Ben Stokes
 Kusal Perera
 Jasprit Bumrah
 Sarfaraz Ahmed
 Jos Buttler
 Rashid Khan
 Dale Steyn
(17 rows)
```

8. Retrieve the best batting strike rate of all countries in ascending order

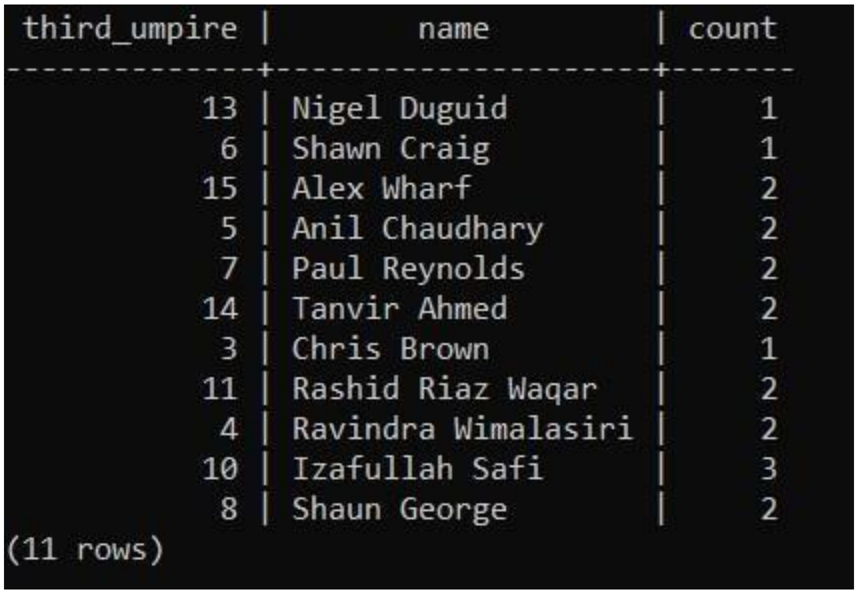
select t.team_id,country,MAX(batting_strike_rate) from players p, player_stats ps,teams t where p.player_id = ps.player_id and t.team_id = p.team_id group by t.team_id order by team_id;



```
 team_id |    country    |  max
---------+--------------+--------
       1 | India        | 185.96
       2 | Australia    | 192.59
       3 | England      | 149.12
       4 | New Zealand  | 187.13
       5 | Pakistan     | 174.85
       6 | Sri Lanka    | 130.41
       7 | South Africa | 194.74
       8 | Bangladesh   | 138.79
       9 | West Indies  | 128.07
      10 | Afghanistan  | 149.51
(10 rows)
```

9. Retrieve the name, number of matches each umpire umpires for as a third umpire.

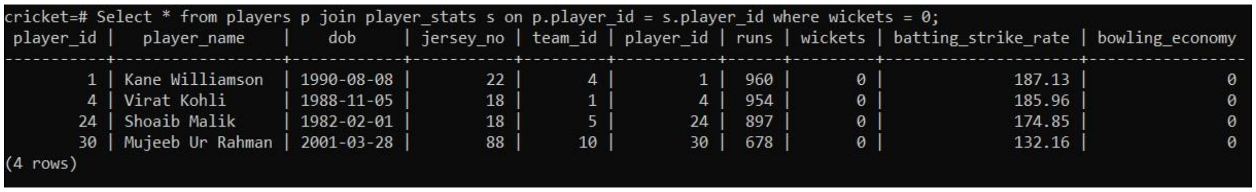
select third_umpire,name,COUNT(third_umpire) from match_umpire inner join umpire on ump_id = third_umpire group by third_umpire,name;



```
 third_umpire |        name         | count
--------------+---------------------+-------
           13 | Nigel Duguid        |     1
            6 | Shawn Craig         |     1
           15 | Alex Wharf          |     2
            5 | Anil Chaudhary      |     2
            7 | Paul Reynolds       |     2
           14 | Tanvir Ahmed        |     2
            3 | Chris Brown         |     1
           11 | Rashid Riaz Waqar   |     2
            4 | Ravindra Wimalasiri |     2
           10 | Izafullah Safi      |     3
            8 | Shaun George        |     2
(11 rows)
```

10. Retrieve the details of all players who do not have any wickets

Select * from players p join player_stats s on p.player_id = s.player_id where wickets = 0;



```
cricket=# Select * from players p join player_stats s on p.player_id = s.player_id where wickets = 0;
 player_id |   player_name    |    dob     | jersey_no | team_id | player_id | runs | wickets | batting_strike_rate | bowling_economy
-----------+------------------+------------+-----------+---------+-----------+------+---------+---------------------+-----------------
         1 | Kane Williamson  | 1990-08-08 |        22 |       4 |         1 |  960 |       0 |              187.13 |               0
         4 | Virat Kohli      | 1988-11-05 |        18 |       1 |         4 |  954 |       0 |              185.96 |               0
        24 | Shoaib Malik     | 1982-02-01 |        18 |       5 |        24 |  897 |       0 |              174.85 |               0
        30 | Mujeeb Ur Rahman | 2001-03-28 |        88 |      10 |        30 |  678 |       0 |              132.16 |               0
(4 rows)
```

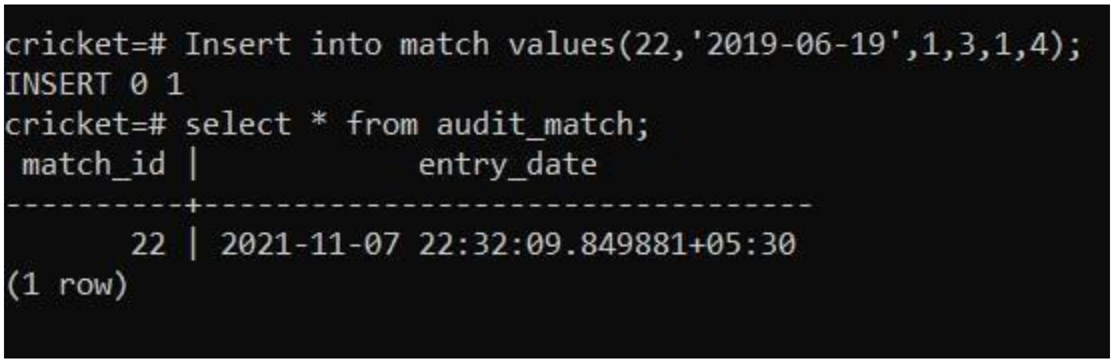
Triggers

1. **Create a trigger such that whenever a new match is inserted, the current timestamp is audited into a table along with the match id to verify the details later on.**

Commands

```
create table audit_match(
  match_id int not null,
  entry_date text not null
);

create or replace function auditmatchfunc() returns trigger as $match$
begin
insert into audit_match(match_id, entry_date) values (new.match_id, current_timestamp);
return new;
end;
$match $language plpgsql;

create trigger match_audit after insert on match
for each row execute procedure auditmatchfunc();
```

Output



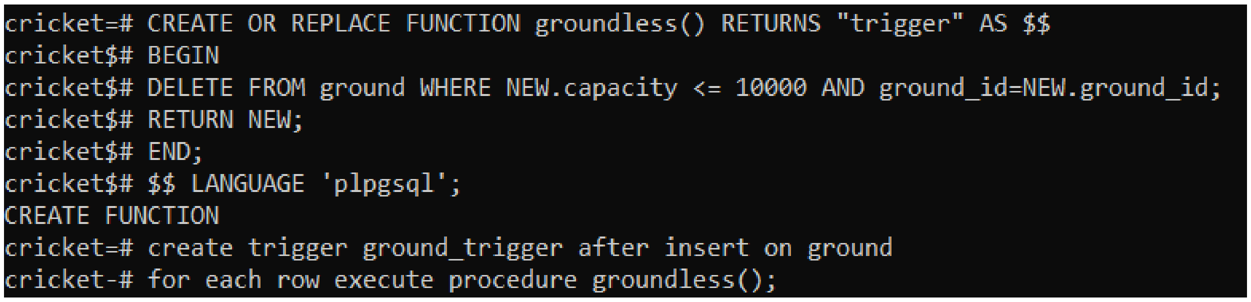
```
cricket=# Insert into match values(22,'2019-06-19',1,3,1,4);
INSERT 0 1
cricket=# select * from audit_match;
 match_id |            entry_date
----------+----------------------------------
       22 | 2021-11-07 22:32:09.849881+05:30
(1 row)
```

2. **Create a trigger to delete a record when ground enters into the ground table where there is less than 10,000 seats capacity.**

Commands

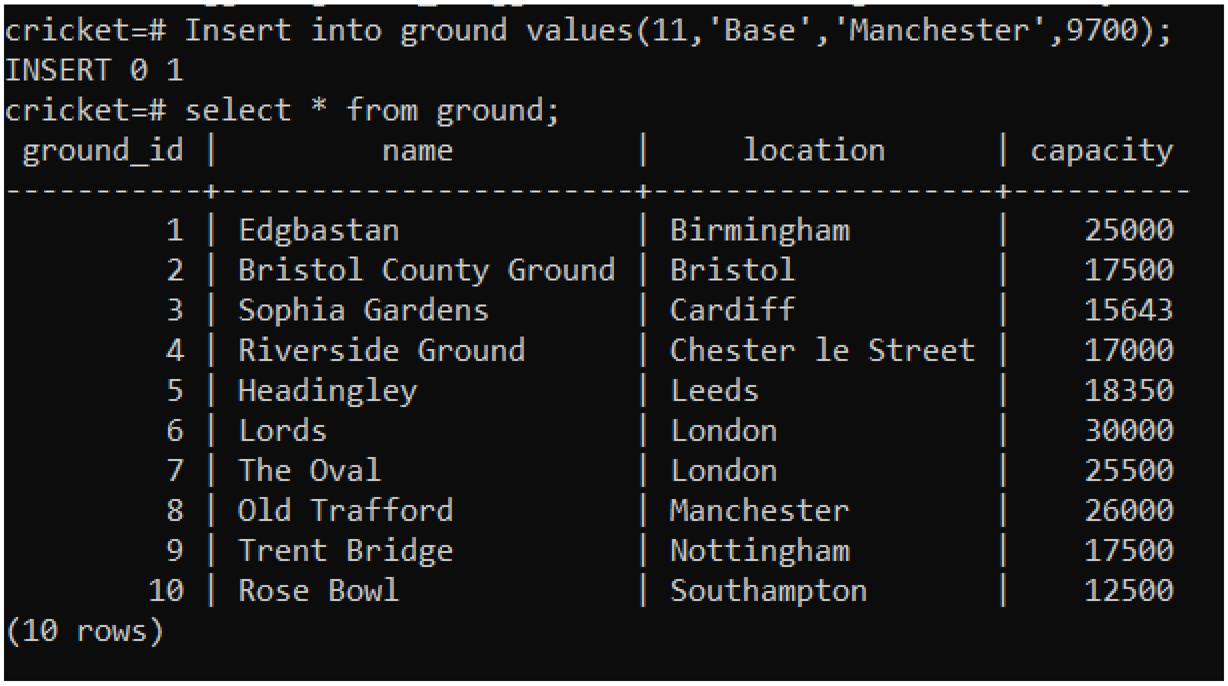
```
CREATE OR REPLACE FUNCTION groundless() RETURNS "trigger" AS $$
BEGIN
DELETE FROM ground WHERE NEW.capacity <= 10000;
RETURN NEW;
END;
$$ LANGUAGE 'plpgsql';
```

```
create trigger ground_trigger after insert on ground
for each row execute procedure groundless();
```



```
cricket=# CREATE OR REPLACE FUNCTION groundless() RETURNS "trigger" AS $$
cricket$# BEGIN
cricket$# DELETE FROM ground WHERE NEW.capacity <= 10000 AND ground_id=NEW.ground_id;
cricket$# RETURN NEW;
cricket$# END;
cricket$# $$ LANGUAGE 'plpgsql';
CREATE FUNCTION
cricket=# create trigger ground_trigger after insert on ground
cricket-# for each row execute procedure groundless();
```

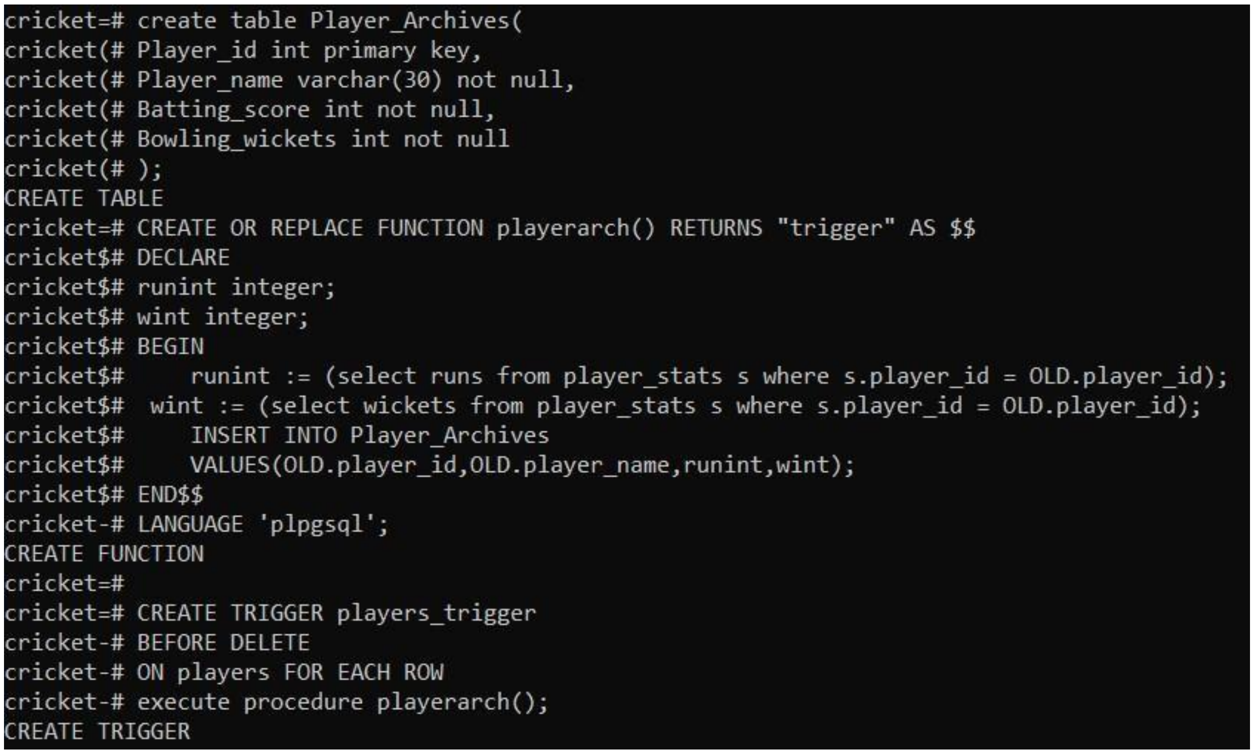
Output
//Values below 10,000 capacity not added



```
cricket=# Insert into ground values(11,'Base','Manchester',9700);
INSERT 0 1
cricket=# select * from ground;
 ground_id |          name          |      location      | capacity
-----------+------------------------+--------------------+----------
         1 | Edgbastan              | Birmingham         |    25000
         2 | Bristol County Ground  | Bristol            |    17500
         3 | Sophia Gardens         | Cardiff            |    15643
         4 | Riverside Ground       | Chester le Street  |    17000
         5 | Headingley             | Leeds              |    18350
         6 | Lords                  | London             |    30000
         7 | The Oval               | London             |    25500
         8 | Old Trafford           | Manchester         |    26000
         9 | Trent Bridge           | Nottingham         |    17500
        10 | Rose Bowl              | Southampton        |    12500
(10 rows)
```

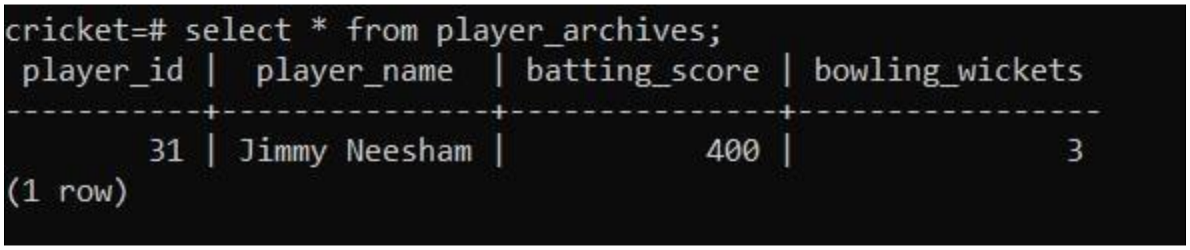
3. **Create a trigger to record all the retired players or injured from the tournament, i.e whenever a player is removed insert his stats to player archives.**

```
create table Player_Archives(
Player_id int not null,
Player_name varchar(30) not null,
Batting_score int,
Bowling_wickets int);

CREATE OR REPLACE FUNCTION playerarch() RETURNS "trigger" AS $$
DECLARE
runint integer;
wint integer;
BEGIN
Runint := 0;
wint:=0;
runint := (select runs from player_stats s where s.player_id = OLD.player_id);
wint := (select wickets from player_stats s where s.player_id = OLD.player_id);
INSERT INTO Player_Archives VALUES(OLD.player_id,OLD.player_name,runint,wint);
Delete from player_stats where player_id = old.player_id;
Return null;
END$$
LANGUAGE 'plpgsql';

CREATE TRIGGER players_trigger
BEFORE DELETE
ON players FOR EACH ROW
execute procedure playerarch();
```



```
cricket=# create table Player_Archives(
cricket(# Player_id int primary key,
cricket(# Player_name varchar(30) not null,
cricket(# Batting_score int not null,
cricket(# Bowling_wickets int not null
cricket(# );
CREATE TABLE
cricket=# CREATE OR REPLACE FUNCTION playerarch() RETURNS "trigger" AS $$
cricket$# DECLARE
cricket$# runint integer;
cricket$# wint integer;
cricket$# BEGIN
cricket$#     runint := (select runs from player_stats s where s.player_id = OLD.player_id);
cricket$#  wint := (select wickets from player_stats s where s.player_id = OLD.player_id);
cricket$#     INSERT INTO Player_Archives
cricket$#     VALUES(OLD.player_id,OLD.player_name,runint,wint);
cricket$# END$$
cricket-# LANGUAGE 'plpgsql';
CREATE FUNCTION
cricket=#
cricket=# CREATE TRIGGER players_trigger
cricket-# BEFORE DELETE
cricket-# ON players FOR EACH ROW
cricket-# execute procedure playerarch();
CREATE TRIGGER
```

Output



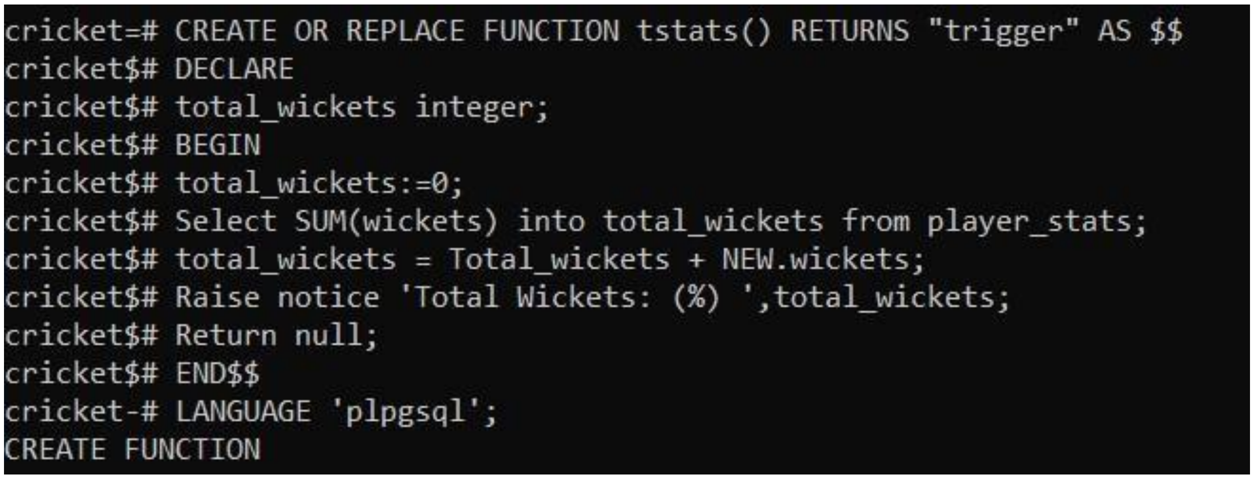
```
cricket=# select * from player_archives;
 player_id |  player_name   | batting_score | bowling_wickets
-----------+---------------+---------------+-----------------
        31 | Jimmy Neesham |           400 |               3
(1 row)
```

4.Create a trigger to raise tournament stats of total wickets taken every time a new player stat is inserted

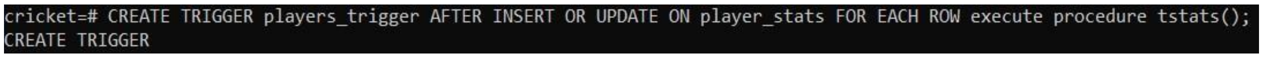
```
CREATE OR REPLACE FUNCTION tstats() RETURNS "trigger" AS $$
DECLARE
total_wickets integer;
BEGIN
total_wickets:=0;
Select SUM(wickets) into total_wickets from player_stats;
total_wickets = Total_wickets + NEW.wickets;
Raise notice 'Total Wickets: (%) ',total_wickets;
Return null;
END$$
```

LANGUAGE 'plpgsql';

CREATE TRIGGER players_trigger AFTER INSERT OR UPDATE ON player_stats FOR EACH ROW execute procedure tstats();

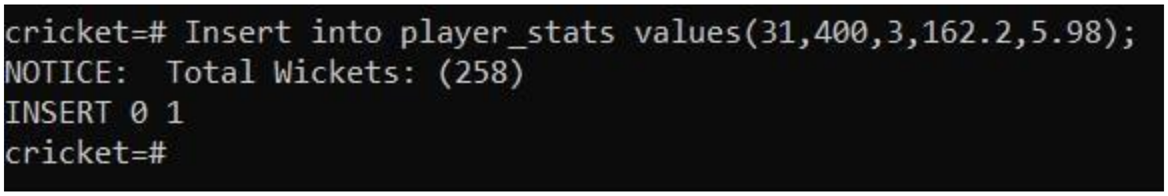


```
cricket=# CREATE OR REPLACE FUNCTION tstats() RETURNS "trigger" AS $$
cricket$# DECLARE
cricket$# total_wickets integer;
cricket$# BEGIN
cricket$# total_wickets:=0;
cricket$# Select SUM(wickets) into total_wickets from player_stats;
cricket$# total_wickets = Total_wickets + NEW.wickets;
cricket$# Raise notice 'Total Wickets: (%) ',total_wickets;
cricket$# Return null;
cricket$# END$$
cricket-# LANGUAGE 'plpgsql';
CREATE FUNCTION
```



```
cricket=# CREATE TRIGGER players_trigger AFTER INSERT OR UPDATE ON player_stats FOR EACH ROW execute procedure tstats();
CREATE TRIGGER
```

Output

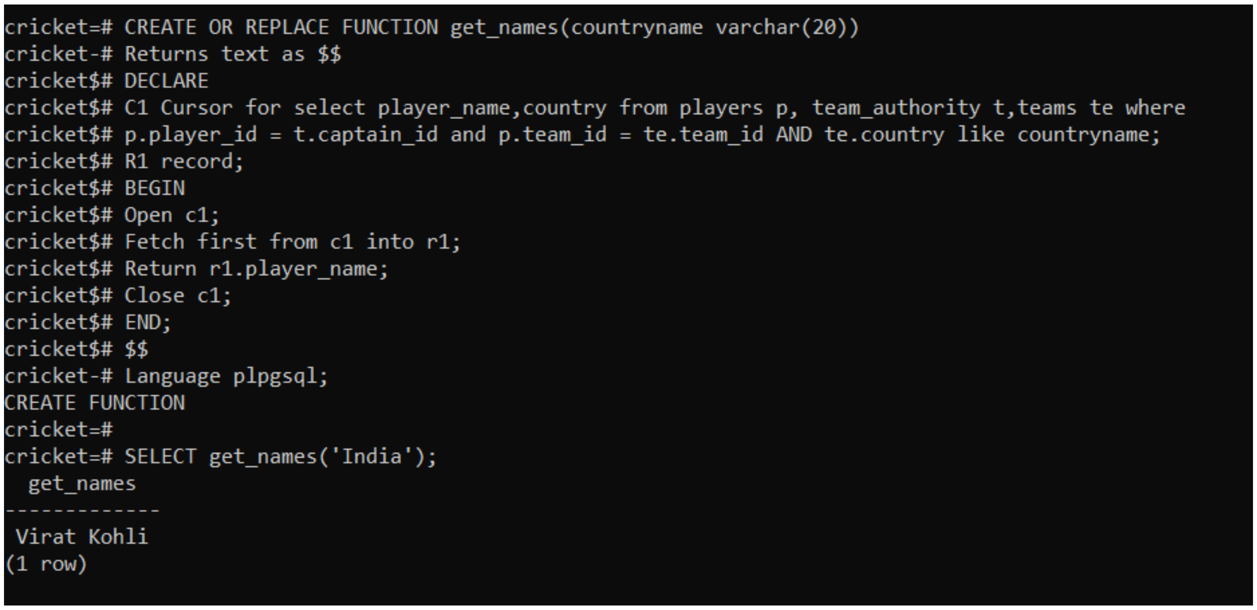


```
cricket=# Insert into player_stats values(31,400,3,162.2,5.98);
NOTICE:  Total Wickets: (258)
INSERT 0 1
cricket=#
```

Stored Procedure With Cursor

1. Print Captain of team specified in a stored procedure

CREATE OR REPLACE FUNCTION get_names(countryname varchar(20))
Returns text as $$
DECLARE
C1 Cursor for select player_name,country from players p, team_authority t,teams te where p.player_id = t.captain_id and p.team_id = te.team_id AND te.country like countryname;
R1 record;
BEGIN
Open c1;
Fetch first from c1 into r1;
Return r1.player_name;
Close c1;
END;
$$
Language plpgsql;

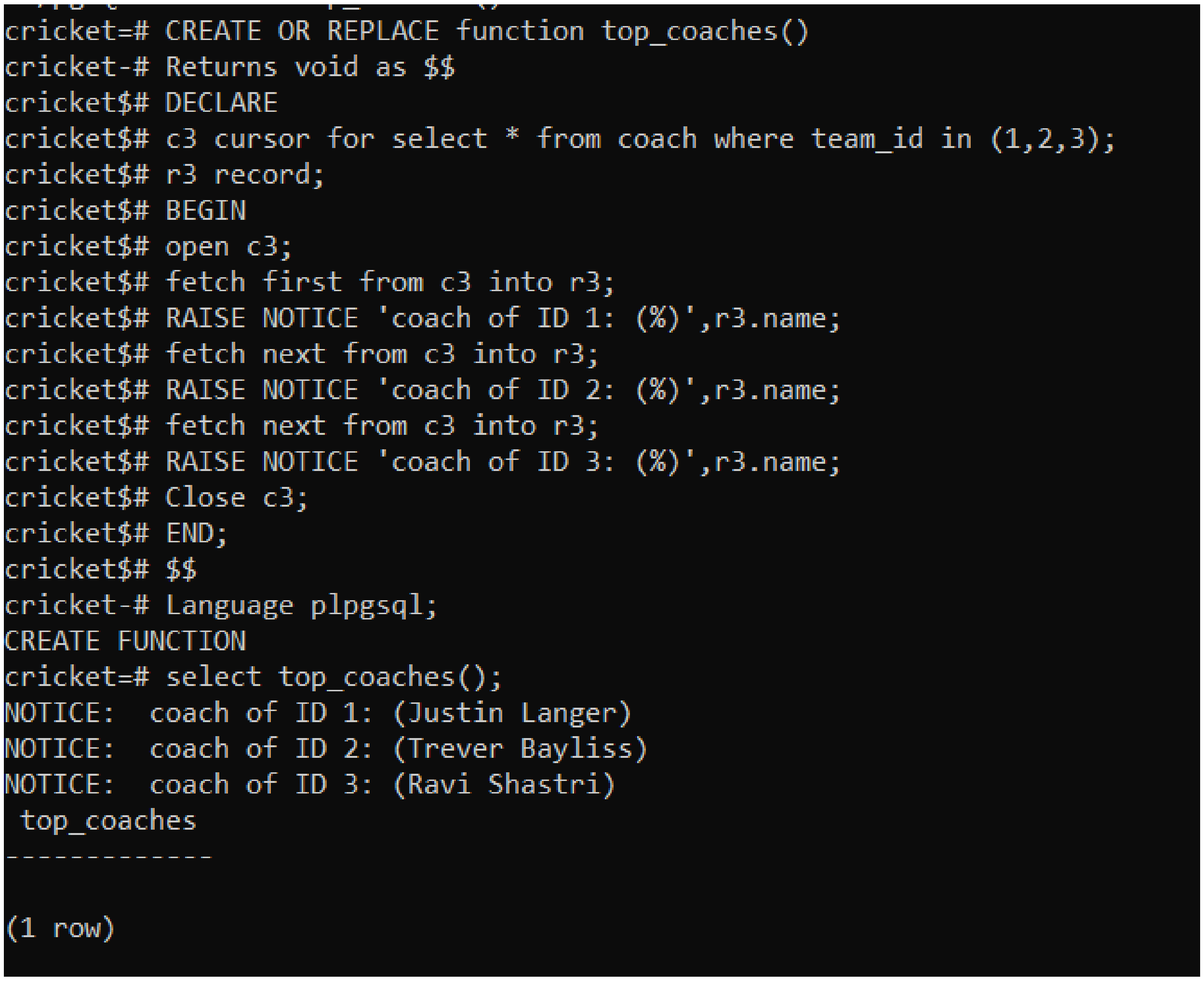


```
cricket=# CREATE OR REPLACE FUNCTION get_names(countryname varchar(20))
cricket-# Returns text as $$
cricket$# DECLARE
cricket$# C1 Cursor for select player_name,country from players p, team_authority t,teams te where
cricket$# p.player_id = t.captain_id and p.team_id = te.team_id AND te.country like countryname;
cricket$# R1 record;
cricket$# BEGIN
cricket$# Open c1;
cricket$# Fetch first from c1 into r1;
cricket$# Return r1.player_name;
cricket$# Close c1;
cricket$# END;
cricket$# $$
cricket-# Language plpgsql;
CREATE FUNCTION
cricket=#
cricket=# SELECT get_names('India');
  get_names
-------------
 Virat Kohli
(1 row)
```

2. Write a function using cursors to view coaches from team_id 1,2 &3

CREATE OR REPLACE function top_coaches()
Returns void as $$
DECLARE
c3 cursor for select * from coach where team_id in (1,2,3);
r3 record;
BEGIN

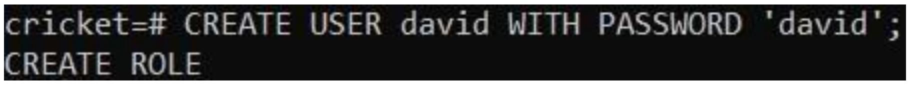
open c3;
fetch first from c3 into r3;
RAISE NOTICE 'coach of ID 1: (%)',r3.name;
fetch next from c3 into r3;
RAISE NOTICE 'coach of ID 2: (%)',r3.name;
fetch next from c3 into r3;
RAISE NOTICE 'coach of ID 3: (%)',r3.name;
Close c3;
END;
$$
Language plpgsql;



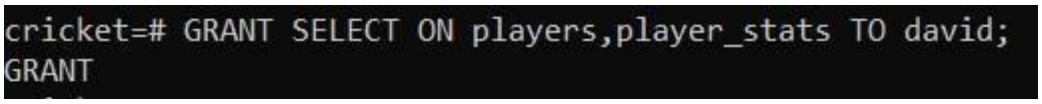
```
cricket=# CREATE OR REPLACE function top_coaches()
cricket-# Returns void as $$
cricket$# DECLARE
cricket$# c3 cursor for select * from coach where team_id in (1,2,3);
cricket$# r3 record;
cricket$# BEGIN
cricket$# open c3;
cricket$# fetch first from c3 into r3;
cricket$# RAISE NOTICE 'coach of ID 1: (%)',r3.name;
cricket$# fetch next from c3 into r3;
cricket$# RAISE NOTICE 'coach of ID 2: (%)',r3.name;
cricket$# fetch next from c3 into r3;
cricket$# RAISE NOTICE 'coach of ID 3: (%)',r3.name;
cricket$# Close c3;
cricket$# END;
cricket$# $$
cricket-# Language plpgsql;
CREATE FUNCTION
cricket=# select top_coaches();
NOTICE:  coach of ID 1: (Justin Langer)
NOTICE:  coach of ID 2: (Trever Bayliss)
NOTICE:  coach of ID 3: (Ravi Shastri)
 top_coaches
-------------

(1 row)
```

User Privileges

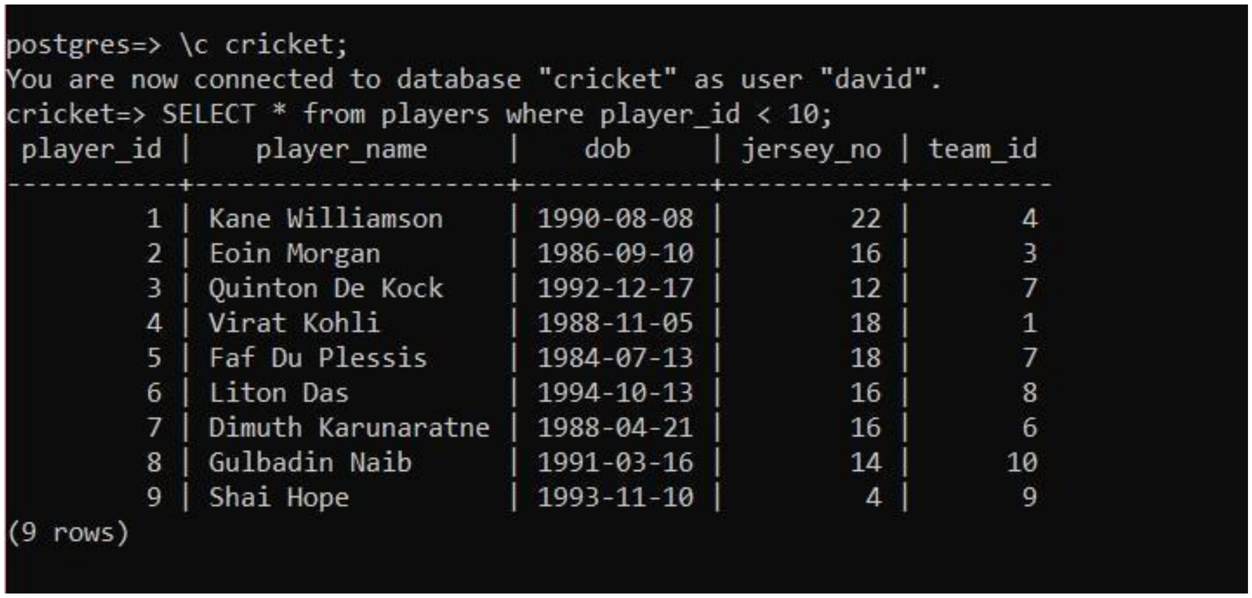
1. Grant select on player and their stats to user david



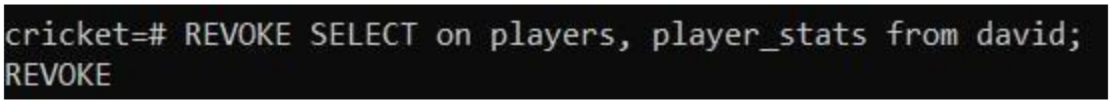
```
cricket=# CREATE USER david WITH PASSWORD 'david';
CREATE ROLE
```



```
cricket=# GRANT SELECT ON players,player_stats TO david;
GRANT
```

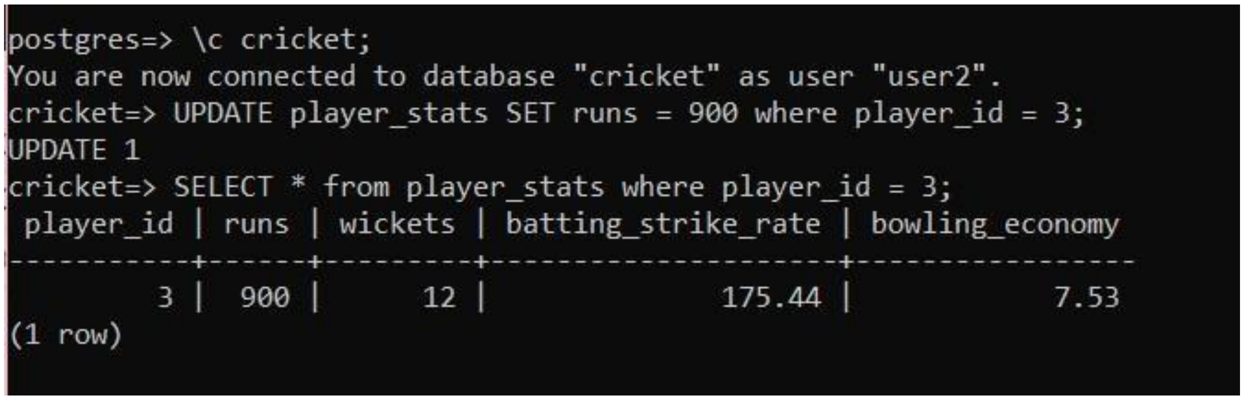


```
postgres=> \c cricket;
You are now connected to database "cricket" as user "david".
cricket=> SELECT * from players where player_id < 10;
 player_id |    player_name     |    dob     | jersey_no | team_id
-----------+--------------------+------------+-----------+---------
         1 | Kane Williamson    | 1990-08-08 |        22 |       4
         2 | Eoin Morgan        | 1986-09-10 |        16 |       3
         3 | Quinton De Kock    | 1992-12-17 |        12 |       7
         4 | Virat Kohli        | 1988-11-05 |        18 |       1
         5 | Faf Du Plessis     | 1984-07-13 |        18 |       7
         6 | Liton Das          | 1994-10-13 |        16 |       8
         7 | Dimuth Karunaratne | 1988-04-21 |        16 |       6
         8 | Gulbadin Naib      | 1991-03-16 |        14 |      10
         9 | Shai Hope          | 1993-11-10 |         4 |       9
(9 rows)
```

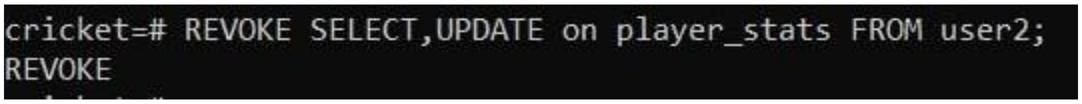


```
cricket=# REVOKE SELECT on players, player_stats from david;
REVOKE
```

2. Grant Select and Update on player stats, update runs to 900 where player-id is 3 from user2



```
postgres=> \c cricket;
You are now connected to database "cricket" as user "user2".
cricket=> UPDATE player_stats SET runs = 900 where player_id = 3;
UPDATE 1
cricket=> SELECT * from player_stats where player_id = 3;
 player_id | runs | wickets | batting_strike_rate | bowling_economy
-----------+------+---------+---------------------+-----------------
         3 |  900 |      12 |              175.44 |            7.53
(1 row)
```



```
cricket=# REVOKE SELECT,UPDATE on player_stats FROM user2;
REVOKE
```