

## SELECT version(); command output

```
postgres=# SELECT version();
               version
-----
PostgreSQL 8.3.23 on x86_64-unknown-linux-gnu, compiled by GCC Ubuntu clang ver
sion 14.0.0-1ubuntu1
(1 row)
```

## SQL

### Exercise - 1

```
postgres=# SELECT COUNT(*) FROM matches JOIN tournaments ON matches.tournament_id = tournaments.tournament_id AND tournament_name = '2014 FIFA World Cup'
postgres-# ;
 count
-----
      64
(1 row)
```

1.

This command basically joins 2 tables. If it was just \* query then all the records from matches having tournament\_id = '2014 FIFA World Cup' would have been displayed along with the rows of tournaments having same tournament id, basically the result will have some extra column for tournament table.

Since it is a count query, it just displays the number of records in the matches table having specific tournament\_id because the number of rows having a specific tournament\_id is 1 in tournaments table.

```
postgres=# SELECT COUNT(*) FROM (SELECT DISTINCT goals.match_id FROM players JOIN goals ON players.player_id = goals.player_id AND players.family_name = 'Mbappé' AND given_name='Kylian') AS t;
 count
-----
      3
(1 row)
```

2.

This command selects the number of matches (distinct match id) in which a player with family\_name and given name as 'Mbappe' and 'Kylian' respectively has scored. It matches(JOINS) the player\_id attribute from two tables namely players and goals. and filters the name (Mbappe) and counts distinct match\_id

```

postgres=# SELECT DISTINCT team_name FROM teams JOIN matches ON (teams.team_id =
matches.home_team_id OR teams.team_id = matches.away_team_id) AND matches.stage
_name = 'final';
 team_name
-----
Argentina
Brazil
Croatia
Czechoslovakia
England
France
Germany
Hungary
Italy
Netherlands
Spain
Sweden
Uruguay
West Germany
(14 rows)

```

3.

This command displays all the teams with stage name = 'final'. It joins two tables namely teams and matches. The matches are filtered as matches.stage\_name = final and the team\_id of both home and away teams is matched in the table named teams.

```

postgres=# SELECT COUNT(*) FROM teams JOIN (SELECT * FROM matches JOIN teams ON (
matches.home_team_id = teams.team_id OR matches.away_team_id = teams.team_id) A
ND team_name = 'Germany') AS t ON ((teams.team_id = t.home_team_id OR teams.tea
m_id = t.away_team_id) AND teams.team_name = 'France' AND t.stage_name != 'group
stage');
 count
-----
      1
(1 row)

```

4.

It joins the table matches and teams on team id = home\_team or away\_team and filters team\_name as germany and this is named as t. now t is joined with teams via team id and teams.team\_name = 'france' and t.stage\_name != 'group stage' is filtered out

```

postgres=# SELECT DISTINCT player_id FROM goals JOIN (SELECT * FROM matches JOIN
  tournaments ON matches.tournament_id = tournaments.tournament_id AND tournament
_name = '1930 FIFA World Cup') AS t1 ON goals.match_id = t1.match_id AND goals.o
wn_goal = FALSE;
 player_id
-----
P-00275
P-00290
P-00296
P-00319
P-00455
P-00521
P-00724
P-00826
P-00978
P-02034
P-02775
P-03789
P-03816
P-03855
P-03952
P-03988
P-03989
P-04102
P-04216
P-04418
P-04578
P-04652
P-04702
P-04746
P-05670
P-05896
P-05989
P-06416
P-06435
P-06672
P-06978
P-07295
P-08997
P-09215
P-09831
P-09935
(36 rows)

```

5.

It first selects all the entries from two tables namely matches and tournaments and joins them with tournament\_id and filters the tournament\_name = '1930 FIFA World Cup' and name this as t1. Now from t1, we join the table goals with it with match\_id and filter out all the own goals and display this result