

Francis Aguilar #22243

Angela Garcia #22869

Proyecto No. 1
Desarrollo y consulta de Base de Datos
(Parejas)

Modalidad y fecha de entrega

- El proyecto se hará en parejas y debe de ser enviado antes de la fecha límite de entrega: lunes 26 de febrero a las 11:55 a.m.
- No se permitirá la entrega o envío de proyecto más allá de la fecha límite

Descripción general del proyecto

El proyecto contempla el uso de tecnologías de bases de datos para la creación y carga de modelos de datos, con el objetivo de utilizar lenguaje SQL para investigación, desarrollo y presentación de resultados sobre preguntas de negocio para apoyo de toma de decisiones.

El conjunto de datos a utilizar serán los archivos en formato CSV que pueden descargarse en Canvas. Este conjunto de datos incluye todos los juegos de fútbol de las cinco principales ligas europeas entre las temporadas de 2,014 a 2,020 (es decir, 7 años) así como información de los jugadores y sus características.

El objetivo general del proyecto es investigar los datos presentados para responder a la siguiente pregunta: **Basado en el desempeño de los equipos y jugadores según este modelo, ¿a qué equipo le apostaría usted? (debe de dar fundamentos basados en los datos)**

Para responder fundamentalmente a esta pregunta usted deberá analizar y entender el modelo de datos presentado, definir métricas que puedan servir de base para justificar una decisión como esta y desarrollar los queries necesarios para calcular dichas métricas.

Etapa 1

En la etapa 1 se encargará de procesar los archivos CSV proporcionados y levantarlos en una base de datos PostgreSQL donde pueda ejecutar sus queries para análisis. Para esto deberá:

1. Descargar los archivos CSV y verificar qué información se incluye en cada uno
2. Crear la base de datos y las tablas en donde almacenará los datos
3. Desarrollar un script de Python (o lenguaje a su elección) que pueda leer la información de los archivos CSV, conectarse a su base de datos y alimentar la información en las tablas creadas

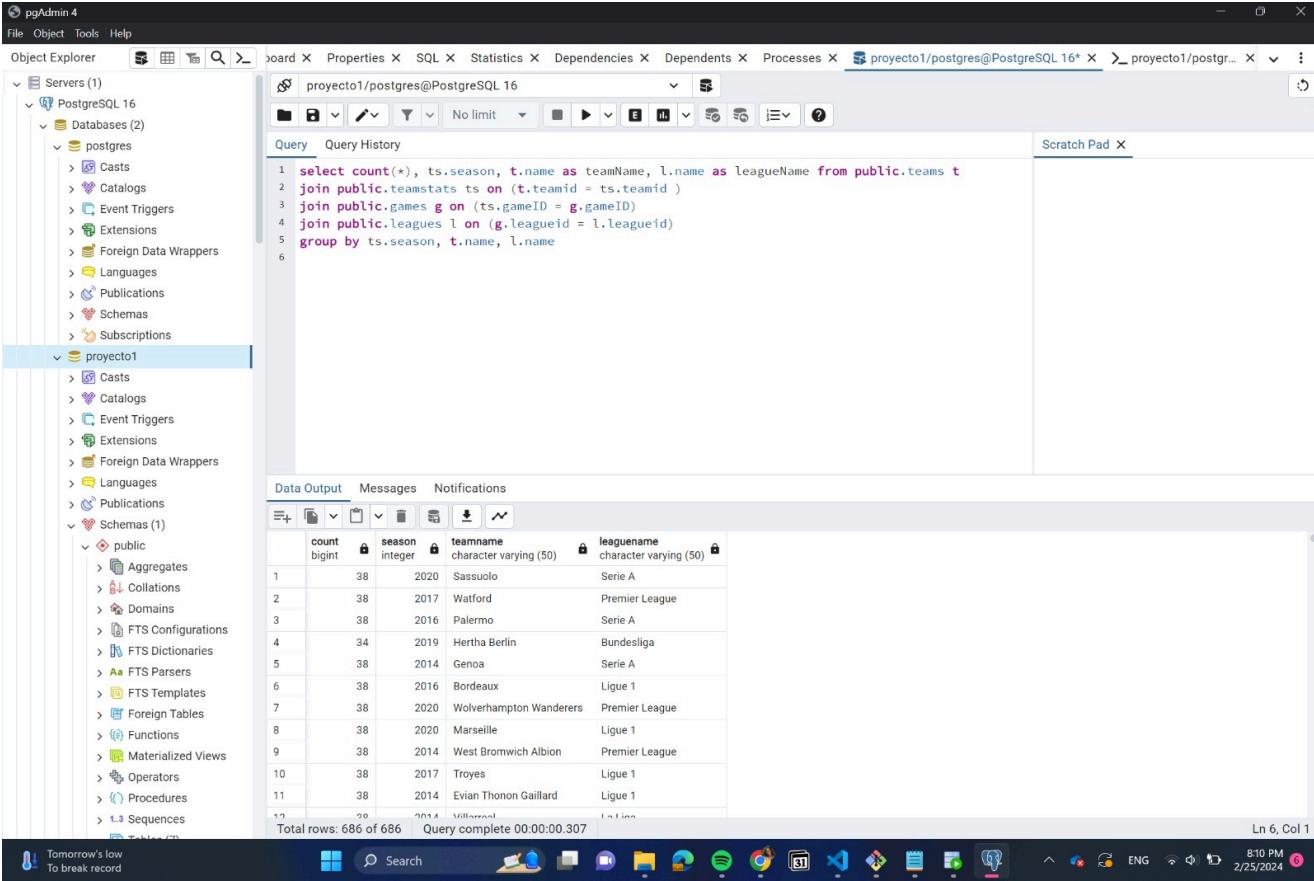
Etapa 2

En la etapa 2 usted ejecutará algunos queries que le permitan familiarizarse con el modelo de datos presentado.

Para esto deberá obtener lo siguiente:

Según estadísticas:

1. La cantidad de juegos jugados en cada temporada por cada equipo, de cada liga (tome en cuenta que cada equipo puede jugar como visitante o como anfitrión).



The screenshot shows the pgAdmin 4 interface. On the left, the Object Explorer pane displays the database structure under 'Servers' (PostgreSQL 16) and 'Databases' (postgres and proyecto1). The 'Query' tab in the center contains the following SQL code:

```
1 select count(*), ts.season, t.name as teamName, l.name as leagueName from public.teams t
2 join public.teamstats ts on (t.teamid = ts.teamid )
3 join public.games g on (ts.gameID = g.gameID)
4 join public.leagues l on (g.leagueid = l.leagueid)
5 group by ts.season, t.name, l.name
6
```

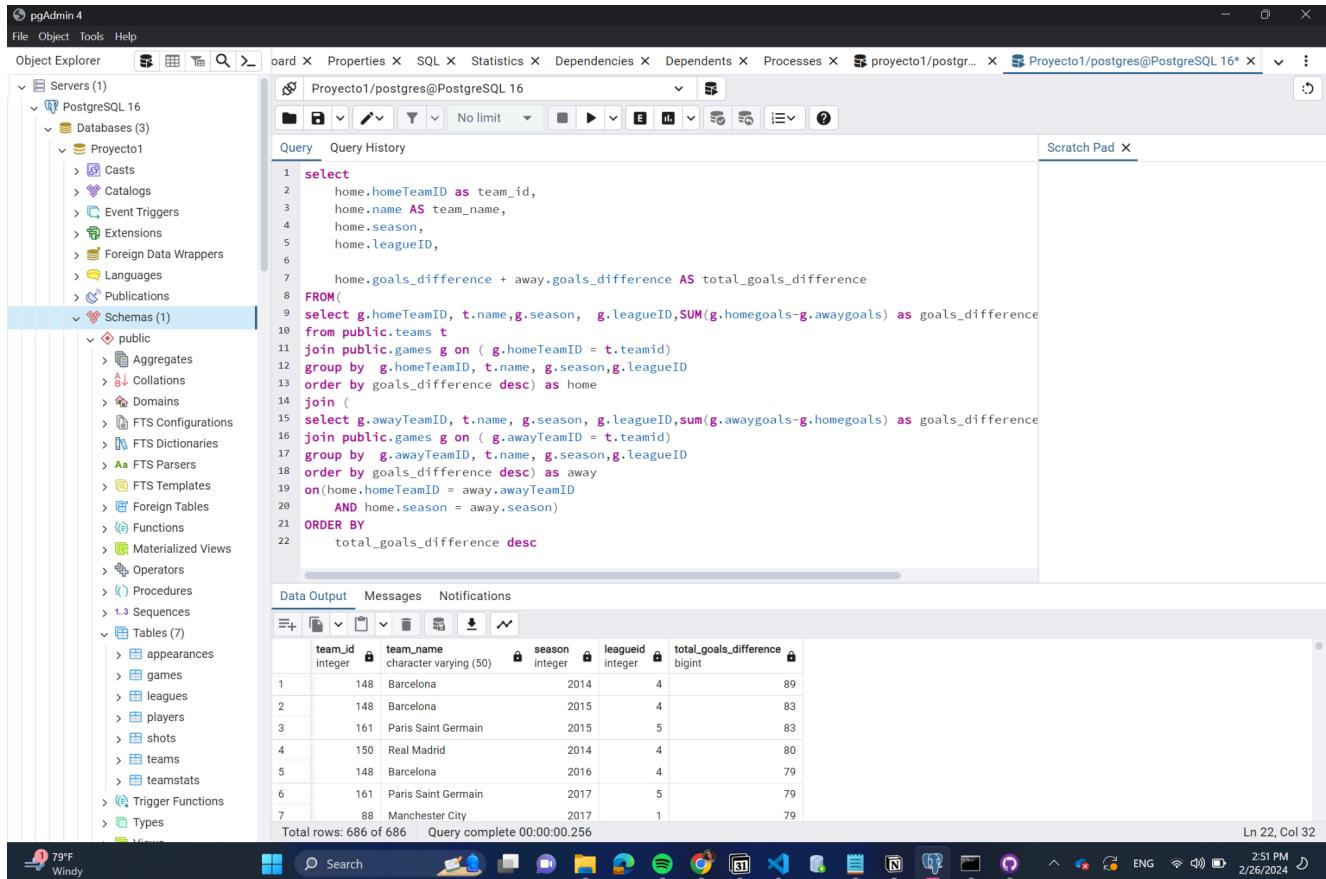
The 'Data Output' tab below shows the results of the query:

| | count | season | teamname | leagueName |
|----|-------|--------|-------------------------|----------------|
| 1 | 38 | 2020 | Sassuolo | Serie A |
| 2 | 38 | 2017 | Watford | Premier League |
| 3 | 38 | 2016 | Palermo | Serie A |
| 4 | 34 | 2019 | Hertha Berlin | Bundesliga |
| 5 | 38 | 2014 | Genoa | Serie A |
| 6 | 38 | 2016 | Bordeaux | Ligue 1 |
| 7 | 38 | 2020 | Wolverhampton Wanderers | Premier League |
| 8 | 38 | 2020 | Marseille | Ligue 1 |
| 9 | 38 | 2014 | West Bromwich Albion | Premier League |
| 10 | 38 | 2017 | Troyes | Ligue 1 |
| 11 | 38 | 2014 | Evian Thonon Gaillard | Ligue 1 |
| 12 | 38 | 2014 | Villarreal | Ligue 1 |

Total rows: 686 of 686 Query complete 00:00:00.307 Ln 6, Col 1

2. ¿Quién es el mejor equipo de todas las ligas y de todas las temporadas según las estadísticas de diferencia de goles?

Hint: Obtenga la cantidad de goles a favor, goles en contra y la diferencia entre las dos anteriores, esto por cada temporada y por cada equipo de cada liga.



```

1 select
2     home.homeTeamID as team_id,
3     home.name AS team_name,
4     home.season,
5     home.leagueID,
6
7     home.goals_difference + away.goals_difference AS total_goals_difference
8 FROM(
9     select g.homeTeamID, t.name,g.season, g.leagueID,SUM(g.homegoals-g.awaygoals) as goals_difference
10    from public.teams t
11   join public.games g on ( g.homeTeamID = t.teamid)
12  group by g.homeTeamID, t.name, g.season,g.leagueID
13  order by goals_difference desc as home
14  join (
15      select g.awayTeamID, t.name, g.season, g.leagueID,sum(g.awaygoals-g.homegoals) as goals_difference
16     from public.games g on ( g.awayTeamID = t.teamid)
17    group by g.awayTeamID, t.name, g.season,g.leagueID
18  order by goals_difference desc as away
19  on(home.homeTeamID = away.awayTeamID
20      AND home.season = away.season)
21 ORDER BY
22     total_goals_difference desc

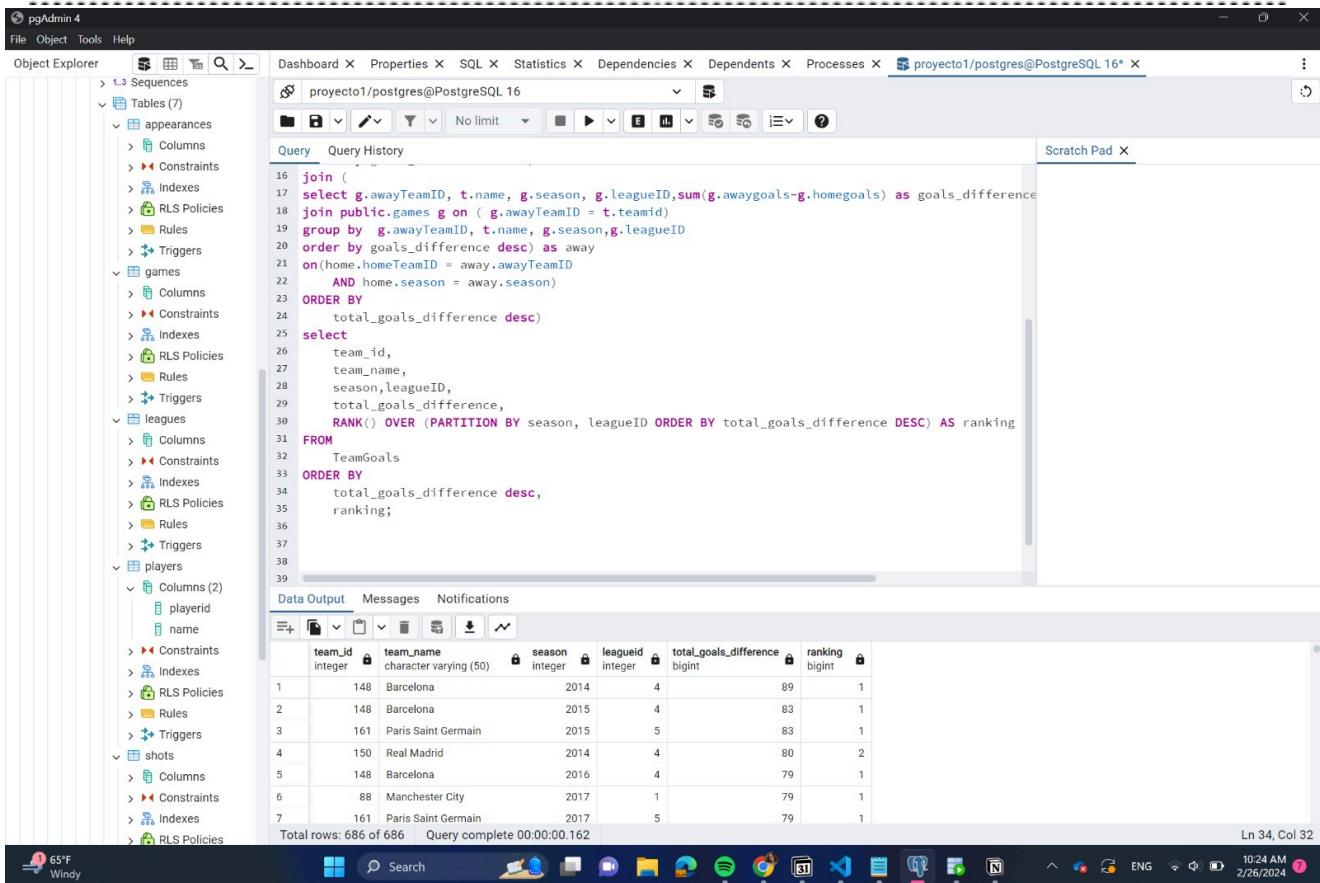
```

| team_id | team_name | season | leagueID | total_goals_difference |
|---------|-----------|---------------------|----------|------------------------|
| 1 | 148 | Barcelona | 2014 | 89 |
| 2 | 148 | Barcelona | 2015 | 83 |
| 3 | 161 | Paris Saint Germain | 2015 | 83 |
| 4 | 150 | Real Madrid | 2014 | 80 |
| 5 | 148 | Barcelona | 2016 | 79 |
| 6 | 161 | Paris Saint Germain | 2017 | 79 |
| 7 | 88 | Manchester City | 2017 | 79 |

Total rows: 686 of 686 Query complete 00:00:00.256 Ln 22, Col 32

R// el mejor equipo de todas las ligas y todas las temporadas es Barcelona con una diferencia de 89 goles en la liga 4 en el 2014

Utilizando este mismo query, obtenga el ranking de los equipos por temporada y por liga, ordenados por ese ranking de manera descendente por diferencia (utilice la función Rank () over partition), para obtener el equipo ganador.



```

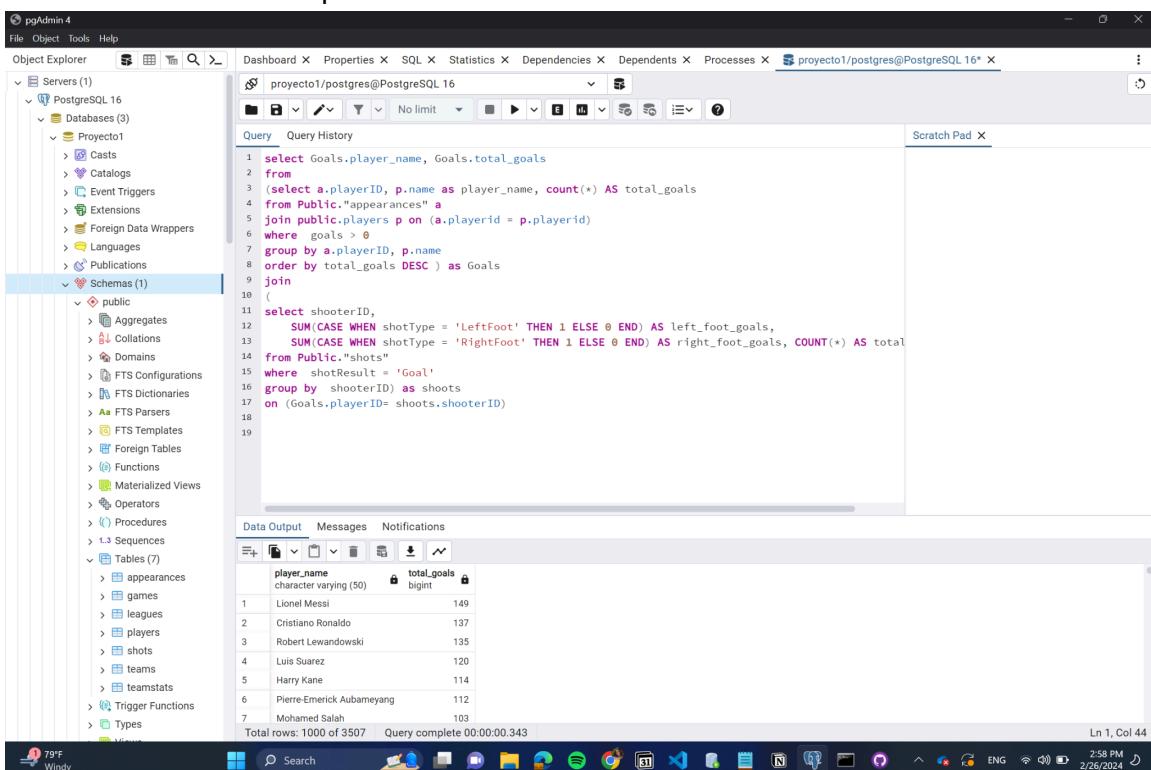
16 join (
17 select g.awayTeamID, t.name, g.season, g.leagueID,sum(g.awaygoals-g.homegoals) as goals_difference
18 join public.games g on ( g.awayTeamID = t.teamid)
19 group by g.awayTeamID, t.name, g.season,g.leagueID
20 order by goals_difference desc) as away
21 on(home.homeTeamID = away.awayTeamID
22 AND home.season = away.season)
23 ORDER BY
24 total_goals_difference desc
25 select
26 team_id,
27 team_name,
28 season,leagueID,
29 total_goals_difference,
30 RANK() OVER (PARTITION BY season, leagueID ORDER BY total_goals_difference DESC) AS ranking
31 FROM
32 TeamGoals
33 ORDER BY
34 total_goals_difference desc,
35 ranking;
36
37
38
39

```

| team_id | team_name | season | leagueID | total_goals_difference | ranking |
|---------|-----------|---------------------|----------|------------------------|---------|
| 1 | 148 | Barcelona | 2014 | 89 | 1 |
| 2 | 148 | Barcelona | 2015 | 83 | 1 |
| 3 | 161 | Paris Saint Germain | 2015 | 83 | 1 |
| 4 | 150 | Real Madrid | 2014 | 80 | 2 |
| 5 | 148 | Barcelona | 2016 | 79 | 1 |
| 6 | 88 | Manchester City | 2017 | 79 | 1 |
| 7 | 161 | Paris Saint Germain | 2017 | 79 | 1 |

Total rows: 686 of 686 Query complete 00:00:00.162 Ln 34, Col 32

3. ¿Quiénes son los jugadores que han realizado mayor cantidad de goles a través de todas las temporadas?



```

1 select Goals.player_name, Goals.total_goals
2 from
3 (select a.playerID, p.name as player_name, count(*) AS total_goals
4 from Public."appearances" a
5 join Public.players p on (a.playerid = p.playerid)
6 where goals > 0
7 group by a.playerID, p.name
8 order by total_goals DESC ) as Goals
9 join
10 (
11 select shooterID,
12 SUM(CASE WHEN shotType = 'LeftFoot' THEN 1 ELSE 0 END) AS left_foot_goals,
13 SUM(CASE WHEN shotType = 'RightFoot' THEN 1 ELSE 0 END) AS right_foot_goals, COUNT(*) AS total
14 from Public."shots"
15 where shotResult = 'Goal'
16 group by shooterID) as shots
17 on (Goals.playerID= shots.shooterID)
18
19

```

| player_name | total_goals |
|---------------------------|-------------|
| Lionel Messi | 149 |
| Cristiano Ronaldo | 137 |
| Robert Lewandowski | 135 |
| Luis Suarez | 120 |
| Harry Kane | 114 |
| Pierre-Emerick Aubameyang | 112 |
| Mohamed Salah | 103 |

Total rows: 1000 of 3507 Query complete 00:00:00.343 Ln 1, Col 44

¿Cuáles son los jugadores con mayor cantidad de pases izquierdos y pases derechos que han hecho goles? (Compare contra los resultados del inciso 2 y determine de manera textual si dichos jugadores pertenecen a los equipos del inciso anterior).

The screenshot shows the pgAdmin 4 interface. On the left is the Object Explorer pane, which lists various database objects like servers, databases, schemas, and tables. The main area contains a SQL editor window with the following query:

```
1 select Goals.player_name, Goals.total_goals, shoots.left_foot_goals, shoots.right_foot_goals
2 , shoots.total_passes
3 from
4 (select a.playerID, p.name as player_name, count(*) AS total_goals
5 from Public."appearances" a
6 join public.players p on (a.playerid = p.playerid)
7 where goals > 0
8 group by a.playerID, p.name
9 order by total_goals DESC ) as Goals
10 join
11 (
12 select shooterID,
13 SUM(CASE WHEN shotType = 'LeftFoot' THEN 1 ELSE 0 END) AS left_foot_goals,
14 SUM(CASE WHEN shotType = 'RightFoot' THEN 1 ELSE 0 END) AS right_foot_goals, COUNT(*) AS total
15 from Public."shots"
16 where shotResult = 'Goal'
17 group by shooterID) as shoots
18 on (Goals.playerID= shoots.shooterID)
19 order by total_passes desc
20
```

Below the query is a Data Output pane displaying the results:

| | player_name | total_goals | left_foot_goals | right_foot_goals | total_passes |
|---|---------------------------|-------------|-----------------|------------------|--------------|
| 1 | Lionel Messi | 149 | 224 | 38 | 272 |
| 2 | Cristiano Ronaldo | 137 | 47 | 138 | 241 |
| 3 | Robert Lewandowski | 135 | 31 | 141 | 207 |
| 4 | Luis Suarez | 120 | 29 | 126 | 171 |
| 5 | Harry Kane | 114 | 35 | 103 | 163 |
| 6 | Pierre-Emerick Aubameyang | 112 | 29 | 111 | 156 |
| 7 | Sergio Agüero | 89 | 28 | 90 | 140 |

Total rows: 1000 of 3507 Query complete 00:00:00.485 Ln 19, Col 27

R// El jugador Leonel Messi si corresponde al equipo del Barcelona ya que ambos se encuentran en el primer lugar, es lógico que el mejor jugador esté en el mejor equipo

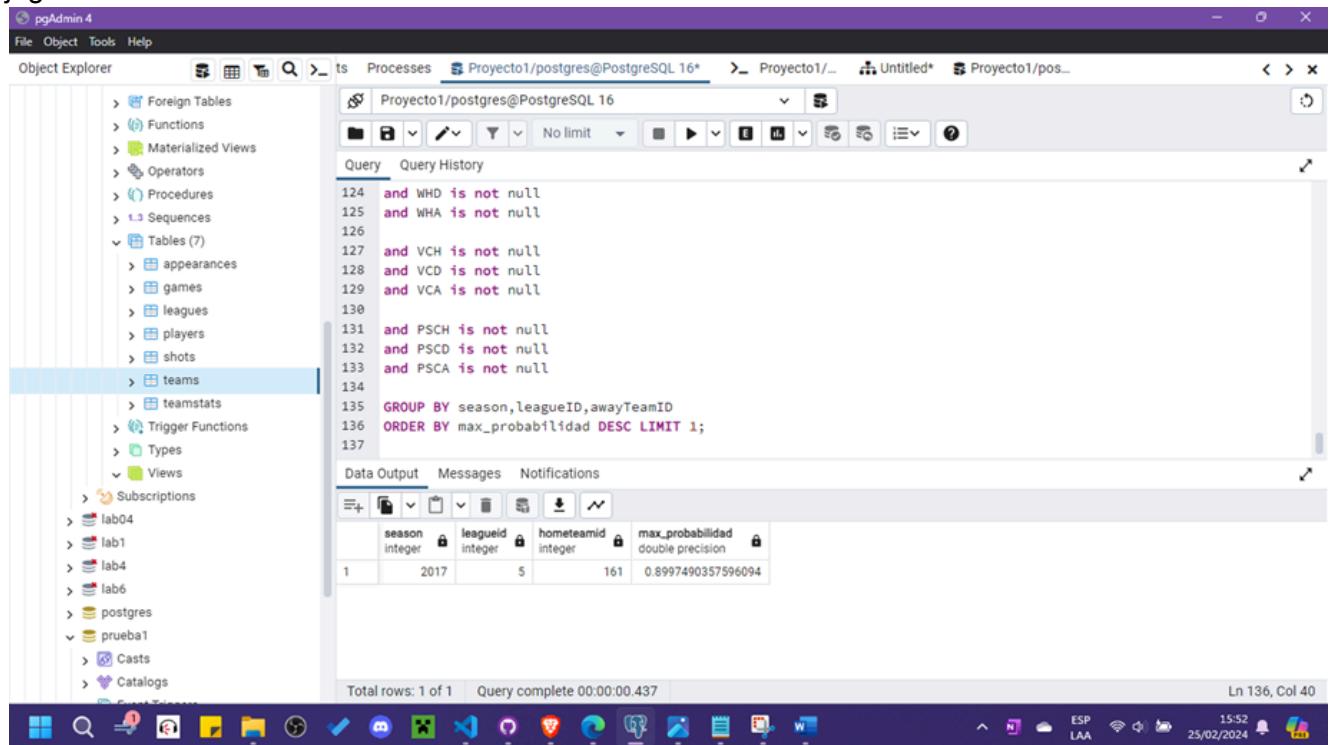
Según apuestas:

4. Realice un comparativo de las probabilidades de todas las casas de apuesta por temporada, liga y equipo, eliminando aquellos equipos que no tienen estadísticas en ninguna casa de apuesta.

Tome en cuenta de que en la tabla de GAMES se representan los datos de probabilidades de que se gane el local, que gane el extranjero o que empate, según diferentes casas de apuestas como Bet365 (B365), Bet&Win (BW), Interwetten (IW), Ladbrokes (LB), William Hill (WH), VC Bet (VC), etc. Por tanto, escoja el valor más alto de estas columnas.

Luego de obtener las probabilidades correctas, escoja la que mejor le convenga para determinar qué equipo tiene la mayor probabilidad de ganar en qué liga de qué temporada.

Tome en cuenta que los valores que aparecen en las columnas (por ejemplo de B365H, B365D y B365A) no son en sí probabilidades porque no se encuentran en el rango entre 0 y 1. Por tanto, para obtener las probabilidades debe de realizar la división de 1 / b365h, por ejemplo. Este es un mecanismo que usan las casas de apuesta para poder confundir al jugador.



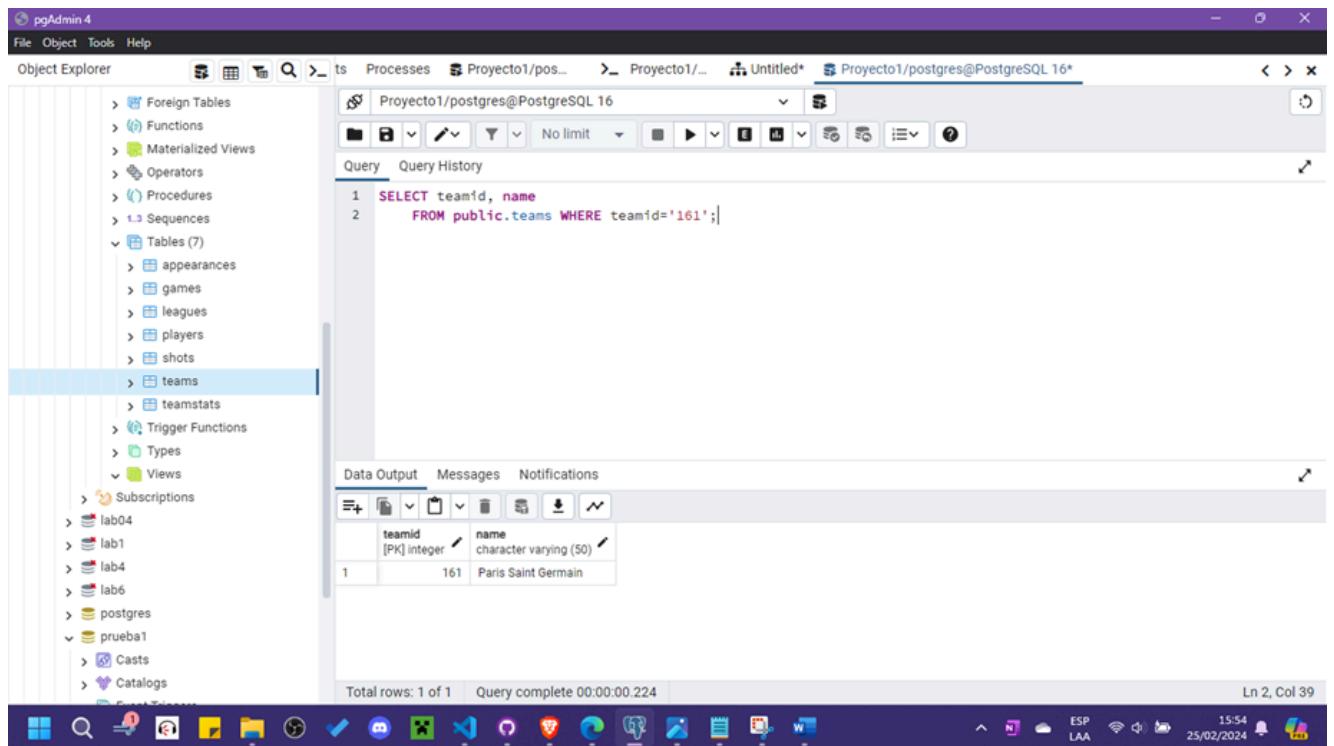
The screenshot shows the pgAdmin 4 interface with a query being run against a PostgreSQL database named 'Proyecto1'. The query retrieves data from the 'games' table where multiple columns (WHD, WHA, VCH, VCD, PSCD, PSCA) are not null. It then groups the results by season, leagueID, and awayTeamID, ordering them by max_probabilidad in descending order and limiting the result to one row. The resulting data is displayed in a table with columns: season, leagueid, hometeamid, and max_probabilidad. The data shows a single row for season 2017, leagueid 5, hometeamid 161, and max_probabilidad 0.8997490357596094.

| season | leagueid | hometeamid | max_probabilidad |
|--------|----------|------------|--------------------|
| 2017 | 5 | 161 | 0.8997490357596094 |

5. ¿Cuál es el mejor equipo de todas las ligas y de todas las temporadas según las apuestas?

Hint: Apóyese o complemente el query del inciso anterior para obtener este.

El mejor equipo es Paris Saint Germain



The screenshot shows the pgAdmin 4 interface. The left pane is the Object Explorer, displaying a tree structure of database objects including Foreign Tables, Functions, Materialized Views, Operators, Procedures, Sequences, Tables (7), Triggers, Types, and Views. The 'teams' table under 'Tables (7)' is selected. The right pane contains a query editor window titled 'Proyecto1/postgres@PostgreSQL 16'. It shows the following SQL query:

```
1 SELECT teamid, name
2   FROM public.teams WHERE teamid='161';
```

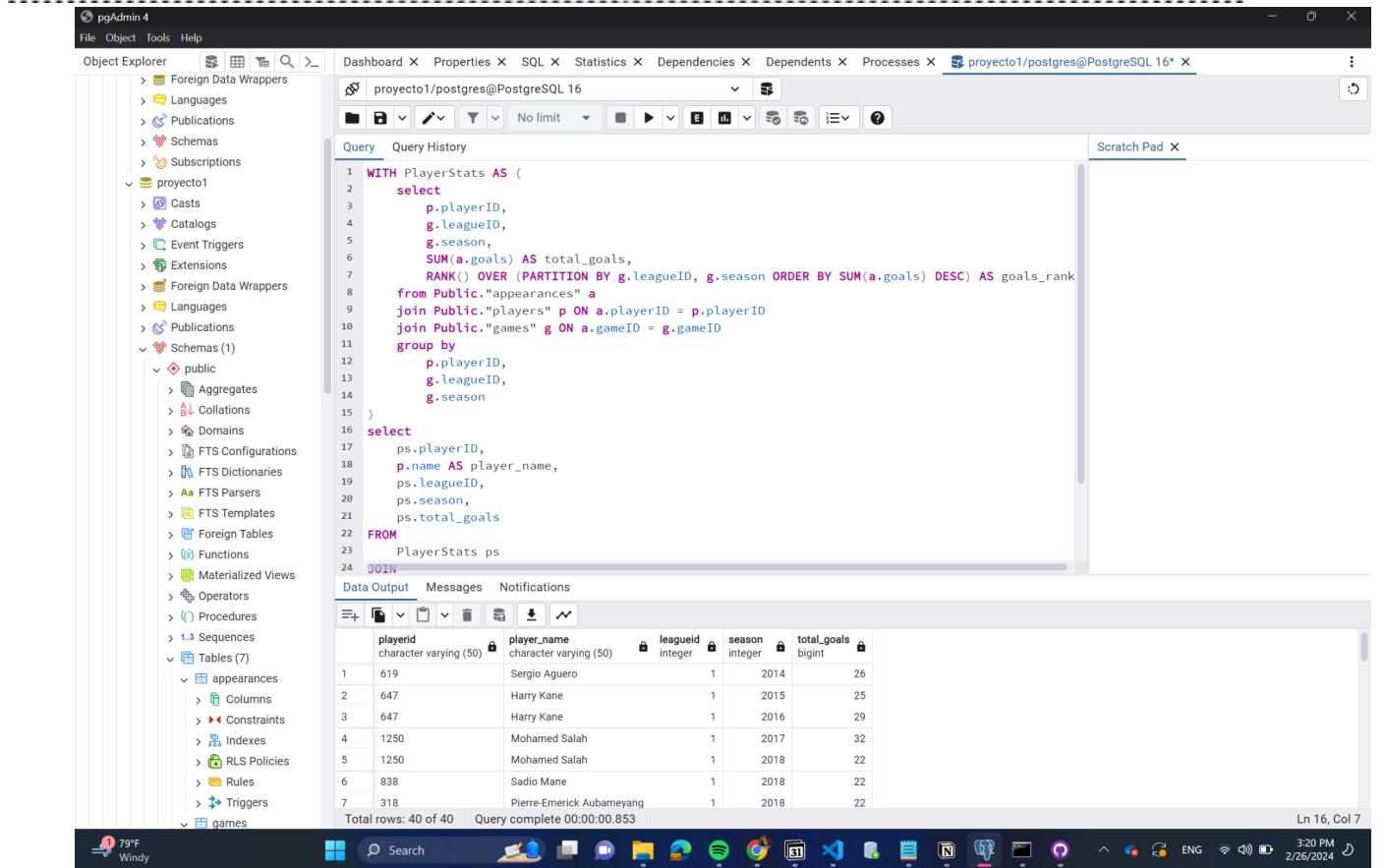
Below the query, the Data Output tab is active, showing a table with one row:

| teamid | name |
|--------|---------------------|
| 161 | Paris Saint Germain |

Total rows: 1 of 1 Query complete 00:00:00.224

Otros:

6. ¿Quiénes son los jugadores de cada liga y cada temporada que tienen los mejores atributos – características de juego -pases, goles, etc.? ¿De acuerdo a este inciso, y comparándolo con el inciso 2 y 5 anteriores, alguno de los jugadores más valiosos se encuentra dentro del mejor equipo?



The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database schema with various objects like tables, functions, and procedures. The main area contains a SQL query window and a data output window.

```

WITH PlayerStats AS (
    select
        p.playerID,
        g.leagueID,
        g.season,
        SUM(a.goals) AS total_goals,
        RANK() OVER (PARTITION BY g.leagueID, g.season ORDER BY SUM(a.goals) DESC) AS goals_rank
    from Public.appearances a
    join Public.players p ON a.playerID = p.playerID
    join Public.games g ON a.gameID = g.gameID
    group by
        p.playerID,
        g.leagueID,
        g.season
)
select
    ps.playerID,
    p.name AS player_name,
    ps.leagueID,
    ps.season,
    ps.total_goals
FROM
    PlayerStats ps
JOIN
    
```

Data Output

| | playerid | player_name | leagueid | season | total_goals |
|---|----------|---------------------------|----------|--------|-------------|
| 1 | 619 | Sergio Aguero | 1 | 2014 | 26 |
| 2 | 647 | Harry Kane | 1 | 2015 | 25 |
| 3 | 647 | Harry Kane | 1 | 2016 | 29 |
| 4 | 1250 | Mohamed Salah | 1 | 2017 | 32 |
| 5 | 1250 | Mohamed Salah | 1 | 2018 | 22 |
| 6 | 838 | Sadio Mane | 1 | 2018 | 22 |
| 7 | 318 | Pierre-Emerick Aubameyang | 1 | 2018 | 22 |

Total rows: 40 of 40 Query complete 00:00:00.853 Ln 16, Col 7

7. Obtenga el rendimiento de los equipos en promedio, comparando goles metidos contra la expectativa de goles, determinando qué equipo era quien tenía más expectativa de goles contra quien fue en realidad el que acertó más goles (goals vs expected goals, x_{goals}) en general, pero también es necesario que lo muestre si dichos equipos jugaron como locales o como extranjeros.

pgAdmin 4

File Object Tools Help

Object Explorer

- gameid
- shooterid
- assisterid
- minute
- situation
- lastaction
- shottype
- shotresult
- xgoal
- positionx
- positiony
- ▶ Constraints
- ▶ Indexes
- ▶ RLS Policies
- ▶ Rules
- ▶ Triggers
- teams
- ▶ Columns
- ▶ Constraints
- ▶ Indexes
- ▶ RLS Policies
- ▶ Rules
- ▶ Triggers
- teamstats
- ▶ Columns
- ▶ Constraints
- ▶ Indexes
- ▶ RLS Policies
- ▶ Rules
- ▶ Triggers
- ▶ Trigger Functions
- ▶ Types
- ▶ Views
- ▶ Subscriptions
- ▶ Login/Group Roles
- ▶ Tablespaces

Dashboard X Properties X SQL X Statistics X Dependencies X Dependents X Processes X proyector1/postgres@PostgreSQL 16*

Query History

```
1 select ts.teamid, t.name, (ts.goals), avg(ts.xgoals) * ts.goals / 10 as probabilidad from public.
2 join public.teams t on (ts.teamid = t.teamid)
3 group by ts.teamid, t.name, ts.goals, ts.xgoals
4 order by goals desc
```

Scratch Pad X

Data Output Messages Notifications

| teamid | name | goals | probabilidad |
|--------|---------------------|-------|--------------------|
| 1 | Real Madrid | 10 | 5.71778 |
| 2 | Leicester | 9 | 3.9424140000000003 |
| 3 | Manchester United | 9 | 4.529249999999999 |
| 4 | Real Madrid | 9 | 5.949819 |
| 5 | Paris Saint Germain | 9 | 3.8729970000000002 |
| 6 | Paris Saint Germain | 9 | 5.273531999999999 |
| 7 | Southampton | 8 | 2.968776 |
| 8 | Manchester City | 8 | 5.304392 |
| 9 | Bayern Munich | 8 | 2.8939440000000003 |

Total rows: 1000 of 25359 Query complete 00:00:00.237 Ln 1, Col 70

79°F Windy

File Object Tools Help

Object Explorer

- gameid
- shooterid
- assisterid
- minute
- situation
- lastaction
- shottype
- shotresult
- xgoal
- positionx
- positiony
- ▶ Constraints
- ▶ Indexes
- ▶ RLS Policies
- ▶ Rules
- ▶ Triggers
- teams
- ▶ Columns
- ▶ Constraints
- ▶ Indexes
- ▶ RLS Policies
- ▶ Rules
- ▶ Triggers
- teamstats
- ▶ Columns
- ▶ Constraints
- ▶ Indexes
- ▶ RLS Policies
- ▶ Rules
- ▶ Triggers
- ▶ Trigger Functions
- ▶ Types
- ▶ Views
- ▶ Subscriptions
- ▶ Login/Group Roles
- ▶ Tablespaces

Dashboard X Properties X SQL X Statistics X Dependencies X Dependents X Processes X proyector1/postgres@PostgreSQL 16*

Query History

```
1 select ts.teamid, t.name, (ts.goals), avg(ts.xgoals) * ts.goals / 10 as goles Esperados from public.
2 join public.teams t on (ts.teamid = t.teamid)
3 group by ts.teamid, t.name, ts.goals, ts.xgoals
4 order by goles Esperados desc
```

Scratch Pad X

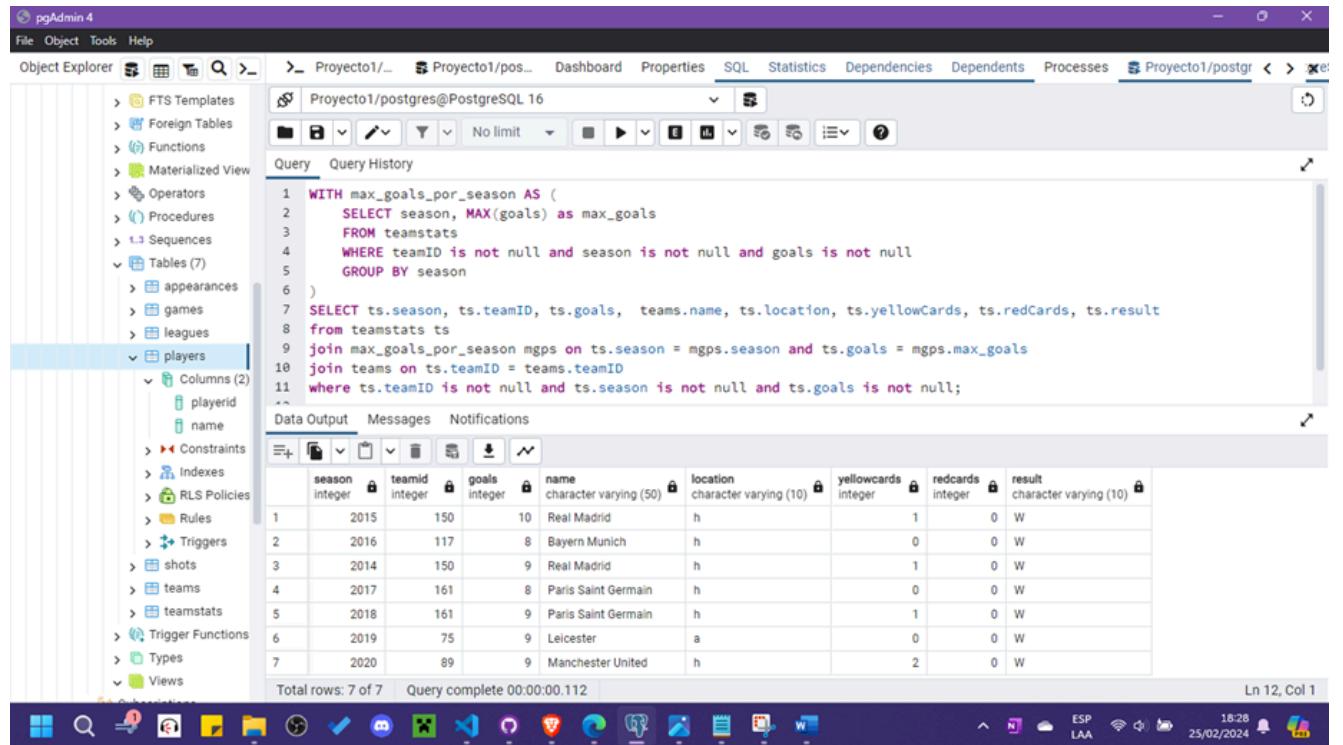
Data Output Messages Notifications

| teamid | name | goals | golest Esperados |
|--------|---------------------|-------|-------------------|
| 1 | Real Madrid | 9 | 5.949819 |
| 2 | Real Madrid | 10 | 5.71778 |
| 3 | Manchester City | 8 | 5.304392 |
| 4 | Paris Saint Germain | 9 | 5.273531999999999 |
| 5 | Barcelona | 8 | 4.949568 |
| 6 | Barcelona | 8 | 4.83344 |
| 7 | Barcelona | 8 | 4.608896 |
| 8 | Manchester United | 9 | 4.529249999999999 |
| 9 | Atalanta | 7 | 4.083079 |

Total rows: 1000 of 25359 Query complete 00:00:00.281 Ln 4, Col 25

El que mas probabilidad de ganar tenia era Manchester City, pero el que mas goles hizo fue el Real Madrid.

8. ¿Cuáles son las características/atributos de los equipos que han sido los líderes de sus ligas en las distintas temporadas? ¿Sus comportamientos son similares?



The screenshot shows the pgAdmin 4 interface. The left pane is the Object Explorer, displaying a tree structure of database objects. The 'Tables' node under 'players' is expanded, showing columns: playerid and name. The right pane contains a SQL editor with the following query:

```
1 WITH max_goals_por_season AS (
2     SELECT season, MAX(goals) as max_goals
3     FROM teamstats
4     WHERE teamID is not null and season is not null and goals is not null
5     GROUP BY season
6 )
7 SELECT ts.season, ts.teamID, ts.goals, teams.name, ts.location, ts.yellowCards, ts.redCards, ts.result
8 from teamstats ts
9 join max_goals_por_season mgps on ts.season = mgps.season and ts.goals = mgps.max_goals
10 join teams on ts.teamID = teams.teamID
11 where ts.teamID is not null and ts.season is not null and ts.goals is not null;
```

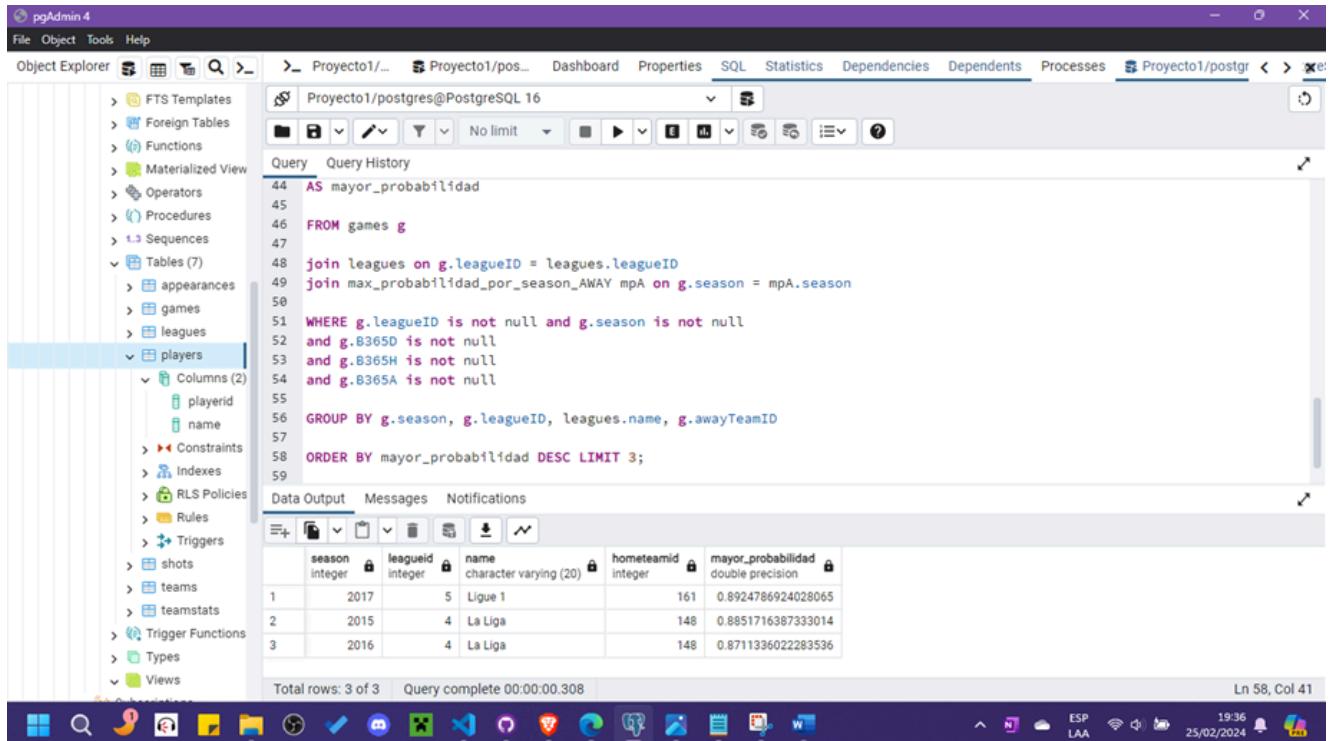
Below the query is a Data Output grid showing the results:

| season | teamid | goals | name | location | yellowcards | redcards | result |
|--------|--------|-------|-----------------------|----------|-------------|----------|--------|
| 1 | 2015 | 150 | 10 Real Madrid | h | 1 | 0 | W |
| 2 | 2016 | 117 | 8 Bayern Munich | h | 0 | 0 | W |
| 3 | 2014 | 150 | 9 Real Madrid | h | 1 | 0 | W |
| 4 | 2017 | 161 | 8 Paris Saint Germain | h | 0 | 0 | W |
| 5 | 2018 | 161 | 9 Paris Saint Germain | h | 1 | 0 | W |
| 6 | 2019 | 75 | 9 Leicester | a | 0 | 0 | W |
| 7 | 2020 | 89 | 9 Manchester United | h | 2 | 0 | W |

Total rows: 7 of 7 Query complete 00:00:00.112 Ln 12, Col 1

Algo que se puede notar es que ninguno de los equipos tiene tarjetas rojas, además que han sido ganadores y que la mayoría ha jugado en casa

9. ¿Según la casa de apuesta Beat365 (tome la mejor probabilidad de las 3 medidas), cuales deberían de ser los equipos que tenían la mayor probabilidad de ganar en cada una de las temporadas (seasons)?



The screenshot shows the pgAdmin 4 interface. In the Object Explorer, the 'players' table under the 'Tables' section is selected. In the main window, a SQL query is being run:

```
AS mayor_probabilidad
FROM games g
join leagues on g.leagueID = leagues.leagueID
join max_probabilidad_por_season_AWAY mpA on g.season = mpA.season
WHERE g.leagueID is not null and g.season is not null
and g.B365D is not null
and g.B365H is not null
and g.B365A is not null
GROUP BY g.season, g.leagueID, leagues.name, g.awayTeamID
ORDER BY mayor_probabilidad DESC LIMIT 3;
```

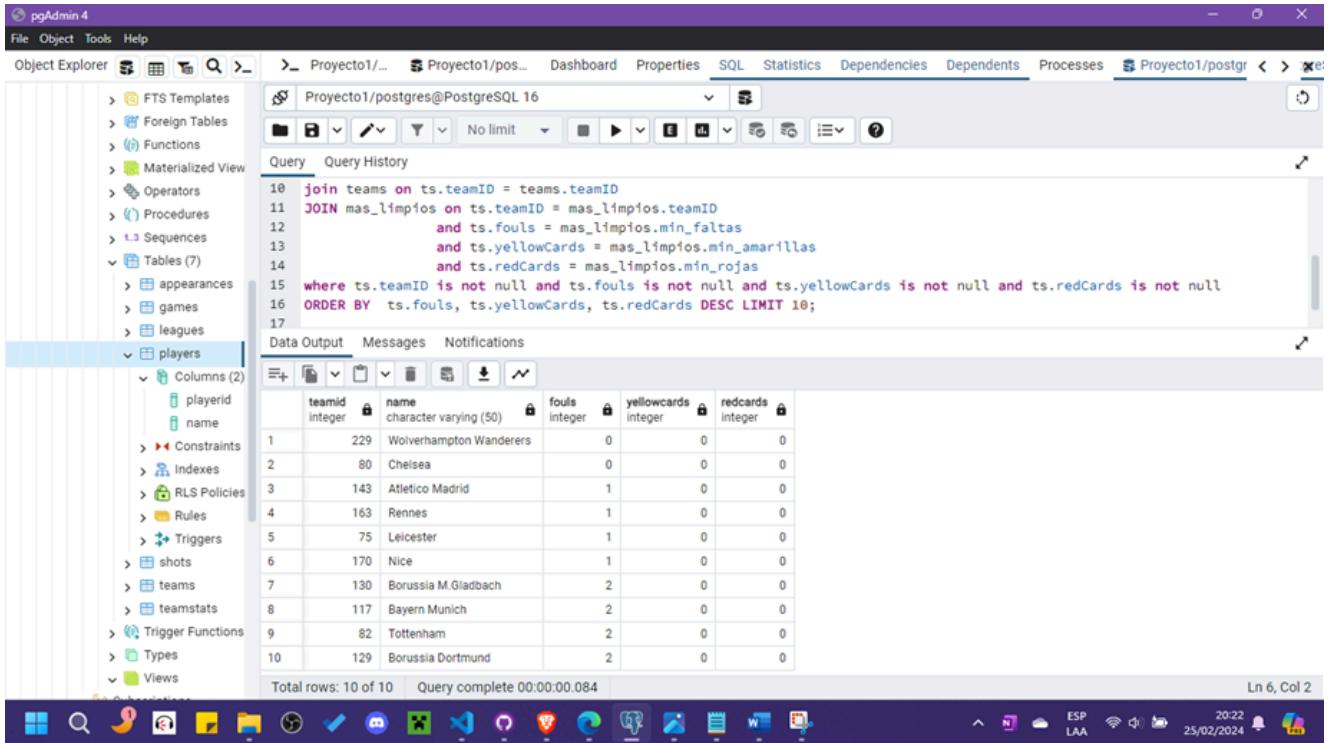
The Data Output tab displays the results of the query:

| | season | leagueID | name | hometeamID | mayor_probabilidad |
|---|--------|----------|---------|------------|--------------------|
| 1 | 2017 | 5 | Ligue 1 | 161 | 0.8924786924028065 |
| 2 | 2015 | 4 | La Liga | 148 | 0.8851716387333014 |
| 3 | 2016 | 4 | La Liga | 148 | 0.8711336022283536 |

Total rows: 3 of 3 Query complete 00:00:00.308

10. Obtenga el top 10 de estadísticas de los equipos más limpios en jugar (mejor faltas, menos tarjetas amarillas, menos tarjetas rojas) y también el top 10 de los equipos más sucios.

Más limpios:



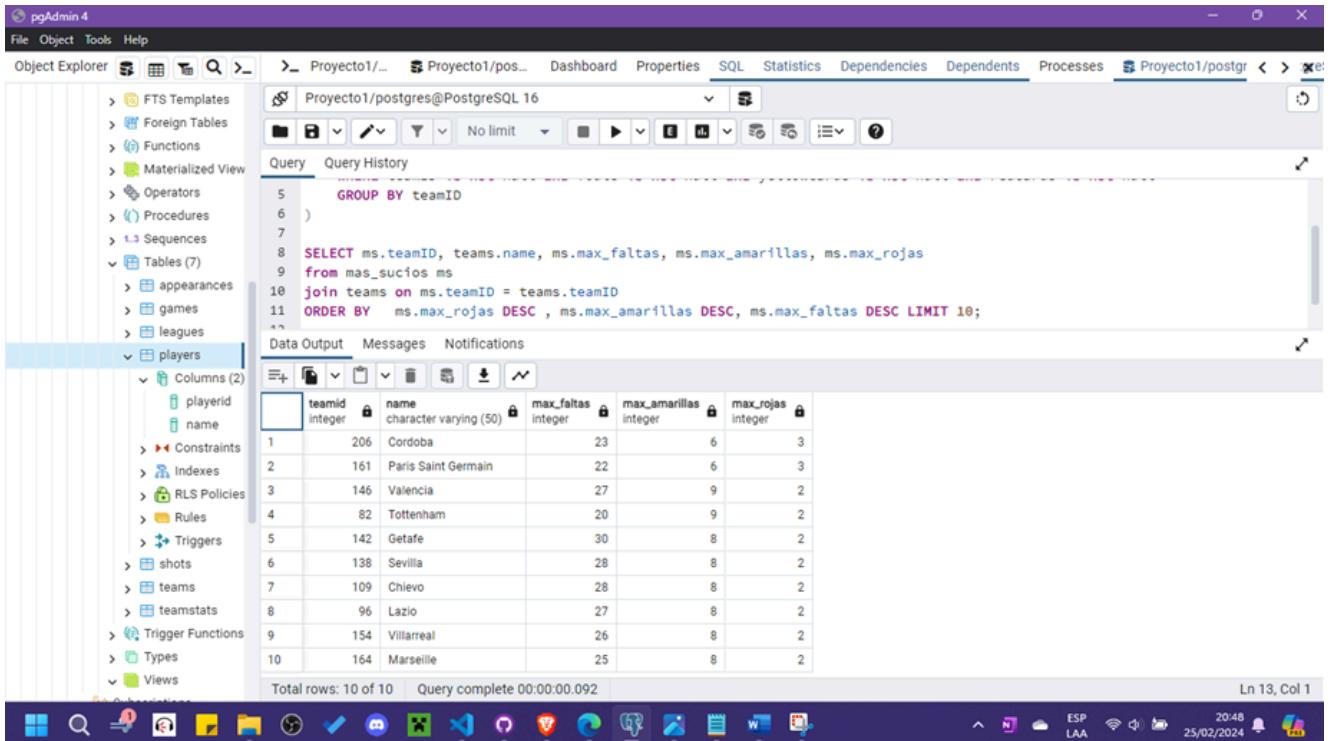
The screenshot shows the pgAdmin 4 interface with the following details:

- Object Explorer:** Shows the database structure with the 'players' table selected.
- SQL Tab:** Contains the following SQL query:


```
10 join teams on ts.teamID = teams.teamID
11 JOIN mas_limpios on ts.teamID = mas_limpios.teamID
12         and ts.fouls = mas_limpios.min_faltas
13         and ts.yellowCards = mas_limpios.min_amarillas
14         and ts.redCards = mas_limpios.min_rojas
15 where ts.teamID is not null and ts.fouls is not null and ts.yellowCards is not null and ts.redCards is not null
16 ORDER BY ts.fouls, ts.yellowCards, ts.redCards DESC LIMIT 10;
17
```
- Data Output Tab:** Displays the results of the query as a table:

| teamid | name | fouls | yellowcards | redcards |
|--------|-----------------------------|-------|-------------|----------|
| 1 | 229 Wolverhampton Wanderers | 0 | 0 | 0 |
| 2 | 80 Chelsea | 0 | 0 | 0 |
| 3 | 143 Atletico Madrid | 1 | 0 | 0 |
| 4 | 163 Rennes | 1 | 0 | 0 |
| 5 | 75 Leicester | 1 | 0 | 0 |
| 6 | 170 Nice | 1 | 0 | 0 |
| 7 | 130 Borussia M.Gladbach | 2 | 0 | 0 |
| 8 | 117 Bayern Munich | 2 | 0 | 0 |
| 9 | 82 Tottenham | 2 | 0 | 0 |
| 10 | 129 Borussia Dortmund | 2 | 0 | 0 |
- Bottom Status Bar:** Shows the message "Total rows: 10 of 10 Query complete 00:00:00.084".

Más sucios:



The screenshot shows the pgAdmin 4 interface with the following details:

- Object Explorer:** Shows the database structure with the 'players' table selected.
- SQL Tab:** Contains the following modified SQL query:


```
5     GROUP BY teamID
6 )
7
8 SELECT ms.teamID, teams.name, ms.max_faltas, ms.max_amarillas, ms.max_rojas
9 from mas_sucios ms
10 join teams on ms.teamID = teams.teamID
11 ORDER BY ms.max_rojas DESC , ms.max_amarillas DESC, ms.max_faltas DESC LIMIT 10;
```
- Data Output Tab:** Displays the results of the query as a table:

| teamid | name | max_faltas | max_amarillas | max_rojas |
|--------|---------------------|------------|---------------|-----------|
| 1 | Cordoba | 23 | 6 | 3 |
| 2 | Paris Saint Germain | 22 | 6 | 3 |
| 3 | Valencia | 27 | 9 | 2 |
| 4 | Tottenham | 20 | 9 | 2 |
| 5 | Getafe | 30 | 8 | 2 |
| 6 | Sevilla | 28 | 8 | 2 |
| 7 | Chievo | 28 | 8 | 2 |
| 8 | Lazio | 27 | 8 | 2 |
| 9 | Villarreal | 26 | 8 | 2 |
| 10 | Marseille | 25 | 8 | 2 |
- Bottom Status Bar:** Shows the message "Total rows: 10 of 10 Query complete 00:00:00.092".

Etapa 3

A continuación, debe plantear sus propias preguntas que le permitan justificar la decisión que tomará acerca de en qué equipo invertirá. Todas sus conclusiones deben estar basadas en el resultado de consultas SQL. Por ejemplo (sugerencias):

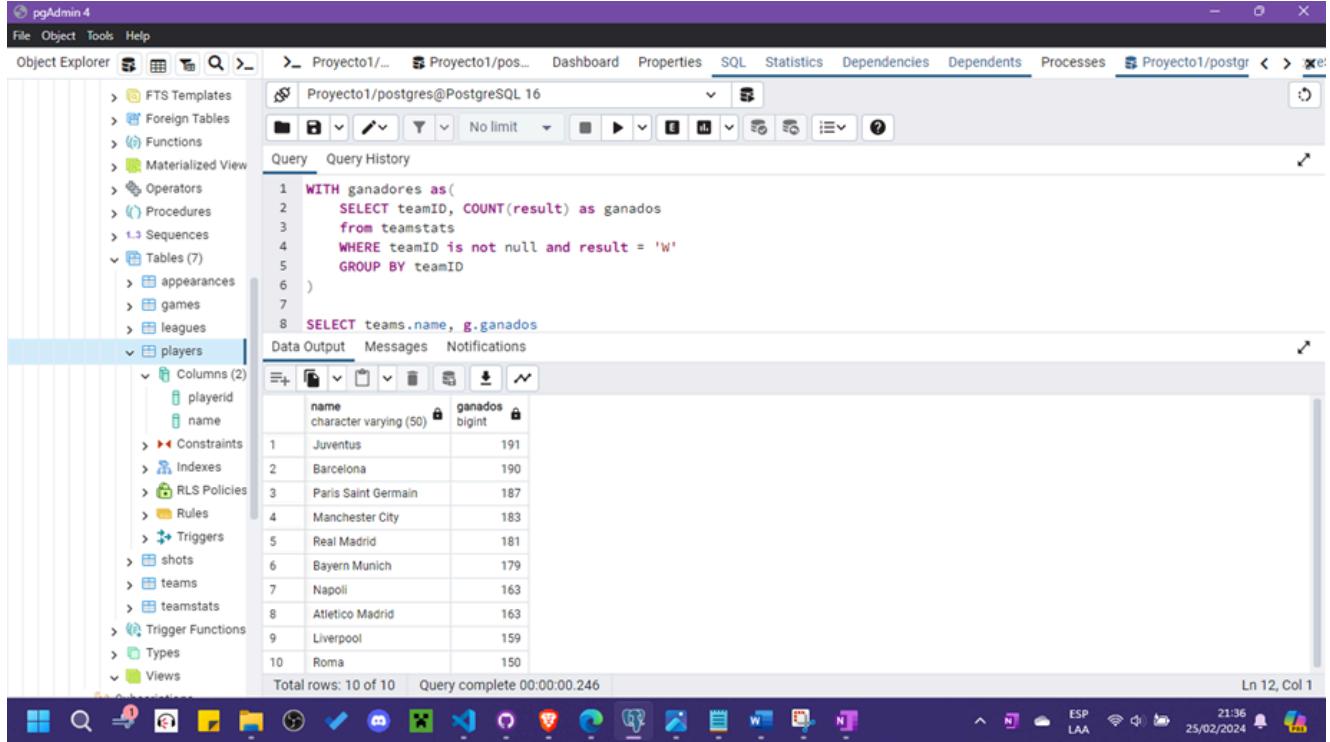
- Podría plantearse apostar en el equipo que sea más consistente en la cantidad de partidos que gana por temporada
- Podría plantearse apostar en el equipo que haya mejorado en las últimas tres temporadas
- Podría plantearse invertir en el equipo tienen características que tienen más goles, menos faltas, más pases, etc
- Etc.

Número de faltas, tarjetas rojas, amarillas, goles esperados y goles del equipo con más faltas de cada temporada. (Para saber que equipos definitivamente no se deberían de apostar)

```
1 WITH max_fouls_por_season AS (
2     SELECT season, MAX(fouls) as max_fouls
3     FROM teamstats
4     WHERE teamID is not null and season is not null and fouls is not null
5     GROUP BY season
6 )
7 SELECT ts.season, ts.teamID, teams.name, ts.fouls, ts.yellowCards, ts.redCards, ts.xGoals, ts.goals, ts.result
8 from teamstats ts
9 join max_fouls_por_season mgps on ts.season = mgps.season and ts.fouls = mgps.max_fouls
10 join teams on ts.teamID = teams.teamID
11 where ts.teamID is not null and ts.season is not null and ts.fouls is not null;
```

| season | teamid | name | fouls | yellowcards | redcards | xgoals | goals | result |
|--------|--------|------------|-------|-------------|----------|----------|-------|--------|
| 2015 | 101 | Genoa | 30 | 3 | 0 | 0.184361 | 0 | L |
| 2016 | 102 | Sampdoria | 32 | 4 | 0 | 1.04646 | 1 | W |
| 2014 | 141 | Espanyol | 33 | 3 | 0 | 0.197774 | 0 | D |
| 2017 | 158 | Alaves | 29 | 7 | 0 | 1.01634 | 1 | W |
| 2017 | 144 | Las Palmas | 29 | 4 | 0 | 0.351625 | 0 | D |
| 2018 | 121 | Augsburg | 29 | 7 | 0 | 3.18937 | 3 | L |
| 2019 | 142 | Getafe | 30 | 2 | 0 | 1.18616 | 1 | L |
| 2020 | 156 | Eibar | 30 | 0 | 0 | 0.839508 | 1 | D |

TOP 10 equipos con resultados de haber ganado partidos



The screenshot shows the pgAdmin 4 interface with a query editor displaying the following SQL code:

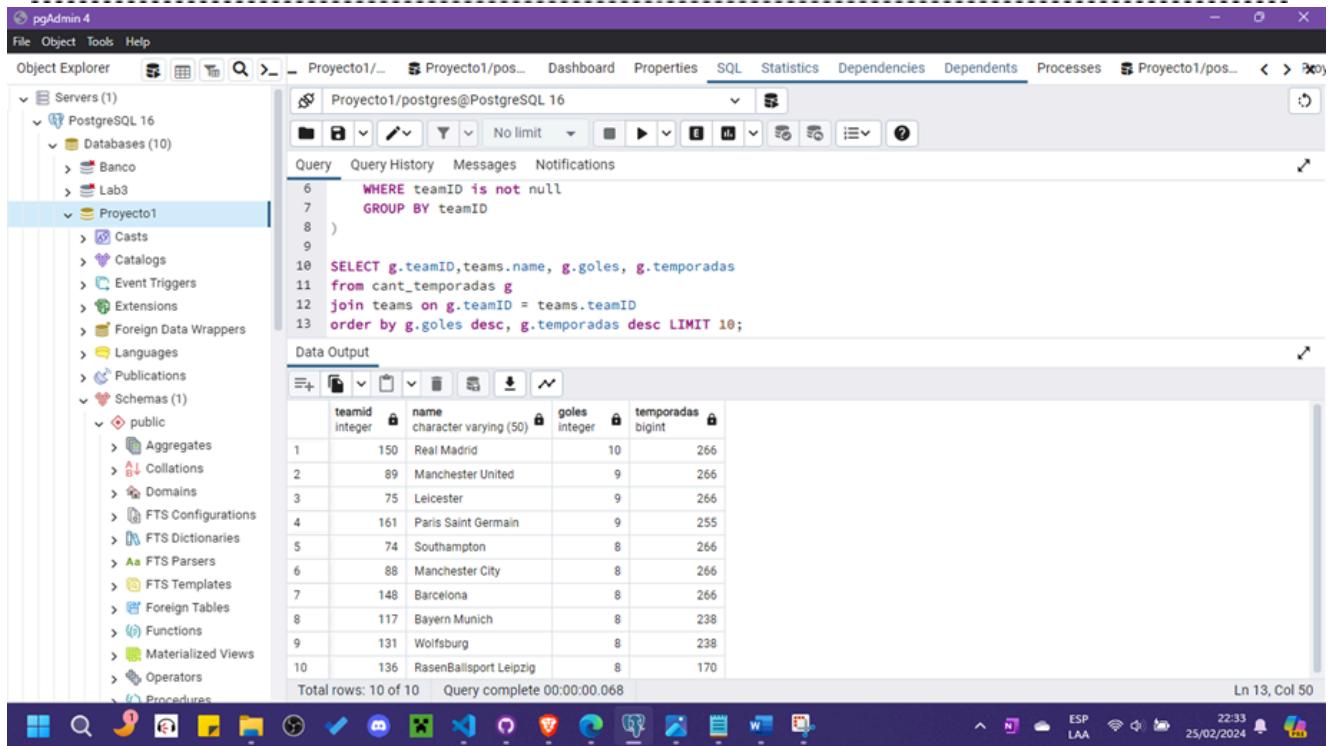
```
1 WITH ganadores AS(
2     SELECT teamID, COUNT(result) AS ganados
3     FROM teamstats
4     WHERE teamID IS NOT NULL AND result = 'W'
5     GROUP BY teamID
6 )
7
8 SELECT teams.name, g.ganados
```

The results are displayed in a table:

| name | ganados |
|---------------------|---------|
| Juventus | 191 |
| Barcelona | 190 |
| Paris Saint Germain | 187 |
| Manchester City | 183 |
| Real Madrid | 181 |
| Bayern Munich | 179 |
| Napoli | 163 |
| Atletico Madrid | 163 |
| Liverpool | 159 |
| Roma | 150 |

Total rows: 10 of 10 Query complete 00:00:00.246

TOP 10 equipos que han participado en las temporadas y tienen mayor cantidad de goles



The screenshot shows the pgAdmin 4 interface with the following details:

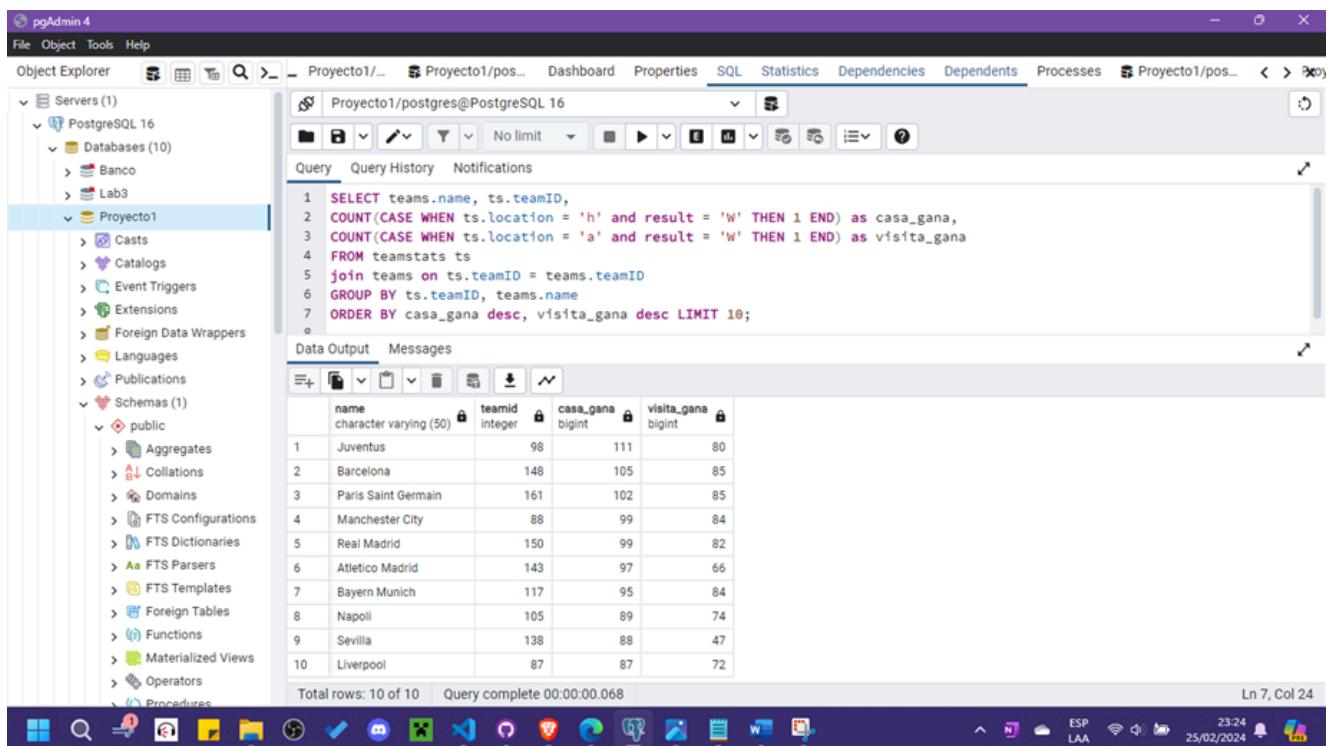
- Object Explorer:** Shows a tree view of servers, databases, and schema objects.
- Query Editor:** Displays a SQL query and its results.


```

6 WHERE teamID is not null
7 GROUP BY teamID
8 )
9
10 SELECT g.teamID, teams.name, g.goles, g.temporadas
11   from cant_temporadas g
12  join teams on g.teamID = teams.teamID
13  order by g.goles desc, g.temporadas desc LIMIT 10;
            
```
- Data Output:** Shows the results of the query as a table.

| teamid | name | goles | temporadas |
|--------|------|------------------------|------------|
| 1 | 150 | Real Madrid | 10 |
| 2 | 89 | Manchester United | 9 |
| 3 | 75 | Leicester | 9 |
| 4 | 161 | Paris Saint Germain | 9 |
| 5 | 74 | Southampton | 8 |
| 6 | 88 | Manchester City | 8 |
| 7 | 148 | Barcelona | 8 |
| 8 | 117 | Bayern Munich | 8 |
| 9 | 131 | Wolfsburg | 8 |
| 10 | 136 | RasenBallsport Leipzig | 8 |

Comparación de los primeros 10 los equipos en casa y fuera



The screenshot shows the pgAdmin 4 interface with the following details:

- Object Explorer:** Shows a tree view of servers, databases, and schema objects.
- Query Editor:** Displays a SQL query and its results.


```

1 SELECT teams.name, ts.teamID,
2 COUNT(CASE WHEN ts.location = 'h' and result = 'W' THEN 1 END) as casa_gana,
3 COUNT(CASE WHEN ts.location = 'a' and result = 'W' THEN 1 END) as visita_gana
4 FROM teamstats ts
5 join teams on ts.teamID = teams.teamID
6 GROUP BY ts.teamID, teams.name
7 ORDER BY casa_gana desc, visita_gana desc LIMIT 10;
            
```
- Data Output:** Shows the results of the query as a table.

| name | teamid | casa_gana | visita_gana |
|---------------------|--------|-----------|-------------|
| Juventus | 98 | 111 | 80 |
| Barcelona | 148 | 105 | 85 |
| Paris Saint Germain | 161 | 102 | 85 |
| Manchester City | 88 | 99 | 84 |
| Real Madrid | 150 | 99 | 82 |
| Atletico Madrid | 143 | 97 | 66 |
| Bayern Munich | 117 | 95 | 84 |
| Napoli | 105 | 89 | 74 |
| Sevilla | 138 | 88 | 47 |
| Liverpool | 87 | 87 | 72 |

Los equipos que más han ganado y empatado de todas las ligas y las season, sin importar la diferencia de goles

The screenshot shows the pgAdmin 4 interface with a database connection to 'proyecto1/postgres@PostgreSQL 16*'. The left sidebar lists tables: games, teamstats, teams, and teamstat. The 'teams' table is selected. The main area displays a query in the 'Query' tab:

```

1 select t.name, g.leagueID ,count(*) as num_wins_empates
2 from public.games g
3 join public.teamstats ts on (g.gameID = ts.gameID)
4 join public.teams t on (t.teamid = ts.teamid)
5 where ts.result = 'W' or ts.result = 'D'
6 group by t.name, g.leagueID
7 order by num_wins_empates desc

```

The 'Data Output' tab shows the results:

| | name | leagueid | num_wins_empates |
|---|---------------------|----------|------------------|
| 1 | Barcelona | 4 | 236 |
| 2 | Juventus | 2 | 233 |
| 3 | Atletico Madrid | 4 | 229 |
| 4 | Real Madrid | 4 | 228 |
| 5 | Paris Saint Germain | 5 | 226 |
| 6 | Manchester City | 1 | ??? |

Total rows: 146 Query complete 00:00:00.115 Ln 1, Col 26

Los equipos que más partidos ganados tiene y cuantos goles

The screenshot shows the pgAdmin 4 interface with the same database connection. The 'teams' table is selected. The main area displays a query in the 'Query' tab:

```

1 select t.name, count(*) as total_ganados, sum(ts.goals) as goles from
2 public.teamstats ts
3 join public.teams t on (t.teamid = ts.teamid)
4 where ts.result = 'W'
5 group by t.name
6 order by total_ganados desc

```

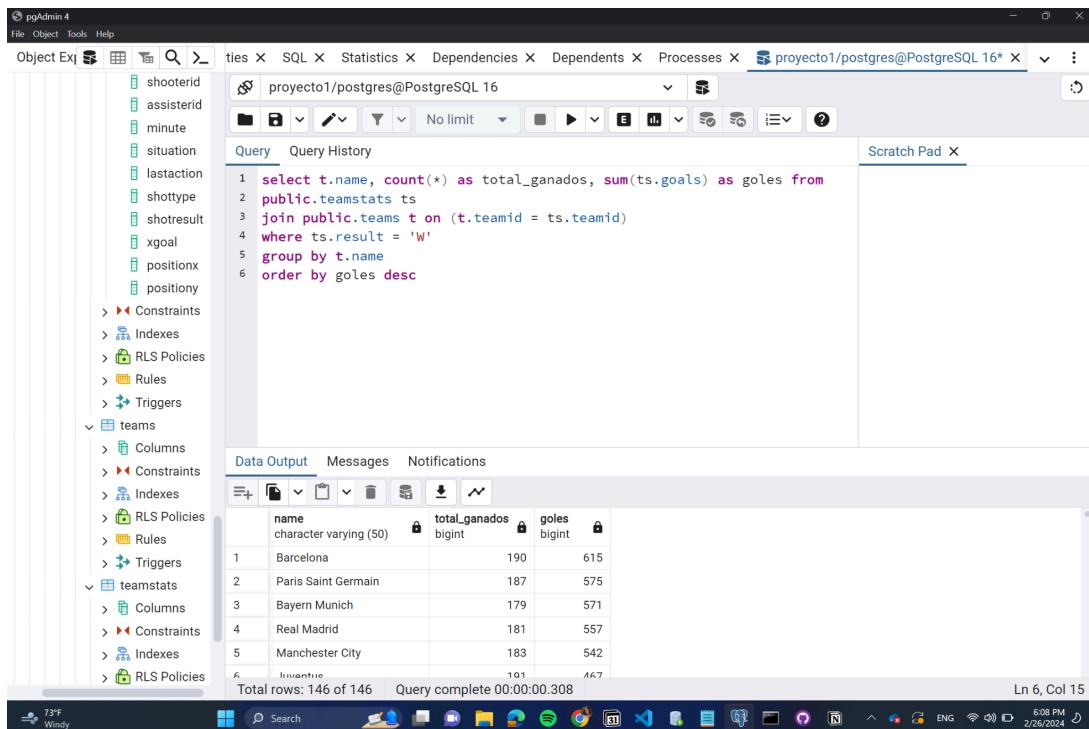
The 'Data Output' tab shows the results:

| | name | total_ganados | goles |
|---|---------------------|---------------|-------|
| 1 | Juventus | 191 | 467 |
| 2 | Barcelona | 190 | 615 |
| 3 | Paris Saint Germain | 187 | 575 |
| 4 | Manchester City | 183 | 542 |
| 5 | Real Madrid | 181 | 557 |
| 6 | Bayern Munich | 170 | 571 |

Total rows: 146 Query complete 00:00:00.129 Ln 6, Col 28

De los equipos que más goles tienen y cuantos partidos tienen ganado (se compara con el resultado

anterior)



The screenshot shows the pgAdmin 4 interface. In the top navigation bar, there are tabs for Object Explorer, SQL, Statistics, Dependencies, Dependents, Processes, and the current connection (proyecto1/postgres@PostgreSQL 16*). Below the tabs is a toolbar with various icons for database management. The main area has two tabs: 'Query' (selected) and 'Query History'. The 'Query' tab contains the following SQL code:

```
1 select t.name, count(*) as total_ganados, sum(ts.goals) as goles from
2 public.teamstats ts
3 join public.teams t on (t.teamid = ts.teamid)
4 where ts.result = 'W'
5 group by t.name
6 order by goles desc
```

The 'Data Output' tab below displays the results of the query:

| | name | total_ganados | goles |
|---|---------------------|---------------|-------|
| 1 | Barcelona | 190 | 615 |
| 2 | Paris Saint Germain | 187 | 575 |
| 3 | Bayern Munich | 179 | 571 |
| 4 | Real Madrid | 181 | 557 |
| 5 | Manchester City | 183 | 542 |
| 6 | Internazionale | 181 | 467 |

Total rows: 146 of 146 Query complete 00:00:00.308 Ln 6, Col 15

De los equipos que más han ganado, sacar cuantos goles tiene en partidos empatados

The screenshot shows the pgAdmin 4 interface. On the left is a tree view of database objects under the schema 'proyecto1'. The main window contains a SQL query editor with the following code:

```
11
12 join
13
14 (select ts.teamid, t.name, sum(ts.goals) as goles, count(*) as total_ganados from
public.teamstats ts
15 join public.teams t on (t.teamid = ts.teamid)
16 where ts.result = 'W'
17 group by t.name, ts.teamid
18 order by total_ganados desc) as ganadores
19
20
21 on (empates.teamID = ganadores.teamid)
22 group by ganadores.name, empates.sum_goals
23 order by sum_goals desc
24
25
```

The Data Output tab shows a table with two columns: 'sum_goals' (bigint) and 'name' (character varying(50)). The data is as follows:

| | sum_goals | name |
|---|-----------|------------|
| 1 | 93 | Torino |
| 2 | 88 | Sassuolo |
| 3 | 84 | Atalanta |
| 4 | 84 | Celta Vigo |
| 5 | 84 | West Ham |

A green message bar at the bottom right indicates: "Successfully run. Total query runtime: 85 msec. 146 rows affected." The status bar at the bottom shows: "Total rows: 146 of 146 Query complete 00:00:00.085 Ln 23, Col 19".

-
- Barcelona es el mejor equipo al que apostar, ya que es el más persistente en los querys realizados. Es el que más partidos ganados tiene en todas las ligas. Además, tienen al mejor jugador, con las mejores características, por lo que es más probable que este sea un buen indicio de ganador para variar ligas.
 - Paris Saint Germain a pesar de parecer un buen candidato, no fue seleccionado debido a que aparece en el top 10 de los equipos más sucios.
 - Hay querys que no se incluyeron en la selección del equipo que debe de apostar debido a que hay querys que sirven de descarte, pero los equipos que aparecen allí no necesariamente aparecían en el top 10 de equipos posibles a seleccionar.

Requerimientos mínimos a completar:

- Se debe presentar el resultado de al menos 18 queries en todo el proyecto
- Deben presentarse al menos tres queries *diferentes* con agrupaciones (GROUP BY)
- Deben presentarse al menos tres queries *diferentes* con JOINs entre dos o más tablas
- Debe presentarse al menos una consulta que haga uso de subqueries

Especificación de tecnología:

- Sistema gestor de base de datos: PostgreSQL
- Interfaz de interacción con base de datos: a discreción

Temas a reforzar:

- Lenguaje SQL: DDL / DML
- PostgreSQL
- Consultas SQL hacia lógica de negocio

Documentos a entregar:

- Archivo comprimido con:
 - a. Diagrama Entidad / Relación de la base de datos construida
 - b. Script desarrollado para procesar archivos CSV y alimentar base de datos
 - c. Documento PDF con las preguntas, queries y resultados obtenidos; que incluya la respuesta a la pregunta de negocio planteada y su justificación

Evaluación:

1. Diseño y construcción de base de datos: 10 puntos
2. Diseño y construcción de script para procesar archivos CSV y alimentar base de datos: 10 puntos
3. Diseño de queries iniciales e interacción con base de datos: 25 puntos
4. Diseño de preguntas propias y queries para responderlas: 40 puntos
5. Análisis de resultados y presentación de solución a pregunta de negocio: 15 puntos

Total: 100 puntos

Puntos extras:

- Creatividad para presentación de análisis
- Creatividad para presentación de resultados de queries de forma gráfica

Glosario:

Datos de probabilidades de apuestas

B365H = Bet365 home win odds
B365D = Bet365 draw odds
B365A = Bet365 away win odds
BSH = Blue Square home win
odds BSD = Blue Square draw
odds BSA = Blue Square away win
odds BWH = Bet&Win home win
odds BWD = Bet&Win draw odds
BWA = Bet&Win away win odds
GBH = Gamebookers home win
odds GBD = Gamebookers draw
odds GBA = Gamebookers away win
odds IWH = Interwetten home win
odds IWD = Interwetten draw odds
IWA = Interwetten away win
odds LBH = Ladbrokes home
win odds LBD = Ladbrokes draw
odds LBA = Ladbrokes away win
odds
PSH = Pinnacle Sports home win
odds PSD = Pinnacle Sports draw
odds PSA = Pinnacle Sports away win
odds SOH = Sporting Odds home win
odds SOD = Sporting Odds draw odds
SOA = Sporting Odds away win odds
SBH = Sportingbet home win odds
SBD = Sportingbet draw odds
SBA = Sportingbet away win odds
SJH = Stan James home win
odds SJD = Stan James draw
odds SJA = Stan James away win
odds SYH = Stanleybet home win
odds SYD = Stanleybet draw
odds
SYA = Stanleybet away win odds
VCH = VC Bet home win odds
VCD = VC Bet draw odds
VCA = VC Bet away win odds
WHH = William Hill home win odds
WHD = William Hill draw odds
WHA = William Hill away win odds

Indicadores

XGOALS OR XG: Los goles esperados le asignan un valor a cada disparo entre 0.00 y 1.00, para reflejar la probabilidad de que ese golpeo termine en gol. Un disparo que mide 0.01 xG sugiere que podría ser un gol en cada 100 oportunidades que se repita. En otras palabras,

hay solo un 1% de posibilidades de que termine en gol y, por ende, no es una muy buena opción. Un disparo que mida 0.99 xG, debería terminar en acierto en 99 de 100 veces por un jugador. Así, es una opción de gol que rara vez puede ser fallada.

Si bien la métrica de los goles esperados se utiliza para calcular valores de disparo individuales, suele ser más útil cuando se utiliza en períodos prolongados, como una temporada completa. También es útil para equipos enteros, en vez de jugadores individuales.

XGOALCHAIN: Esta métrica se define por la **cantidad de Goles Esperados en los que ha participado un jugador; si el jugador ha participado en una jugada que acaba en tiro, el xG de la jugada se sumará a esta métrica**. De esta forma, damos valor a jugadores que con frecuencia forman parte de secuencias de pase que acaban en tiro.

XGOALSBUILDUP: se basa en conocer qué jugadores forman parte de cadenas de pase que acaban en tiro pero sin tener en cuenta los dos últimos eslabones -tiro y último pase-, midiendo con ello a centrocampistas, centrales o laterales que tienen una influencia importante en las posesiones de su equipo.

XASSISTS: El modelo de Expected Assists (xA) de Stats Perform mide la probabilidad de que un pase se convierta en una asistencia de gol. El modelo premia a los jugadores que dan un pase en zonas de peligro, independientemente de si el receptor realiza o no un disparo. La xA se mide en una escala entre cero y uno, donde cero representa un pase que nunca resultará en asistencia y uno representa un pase del que se esperaría que el receptor anotara siempre.