

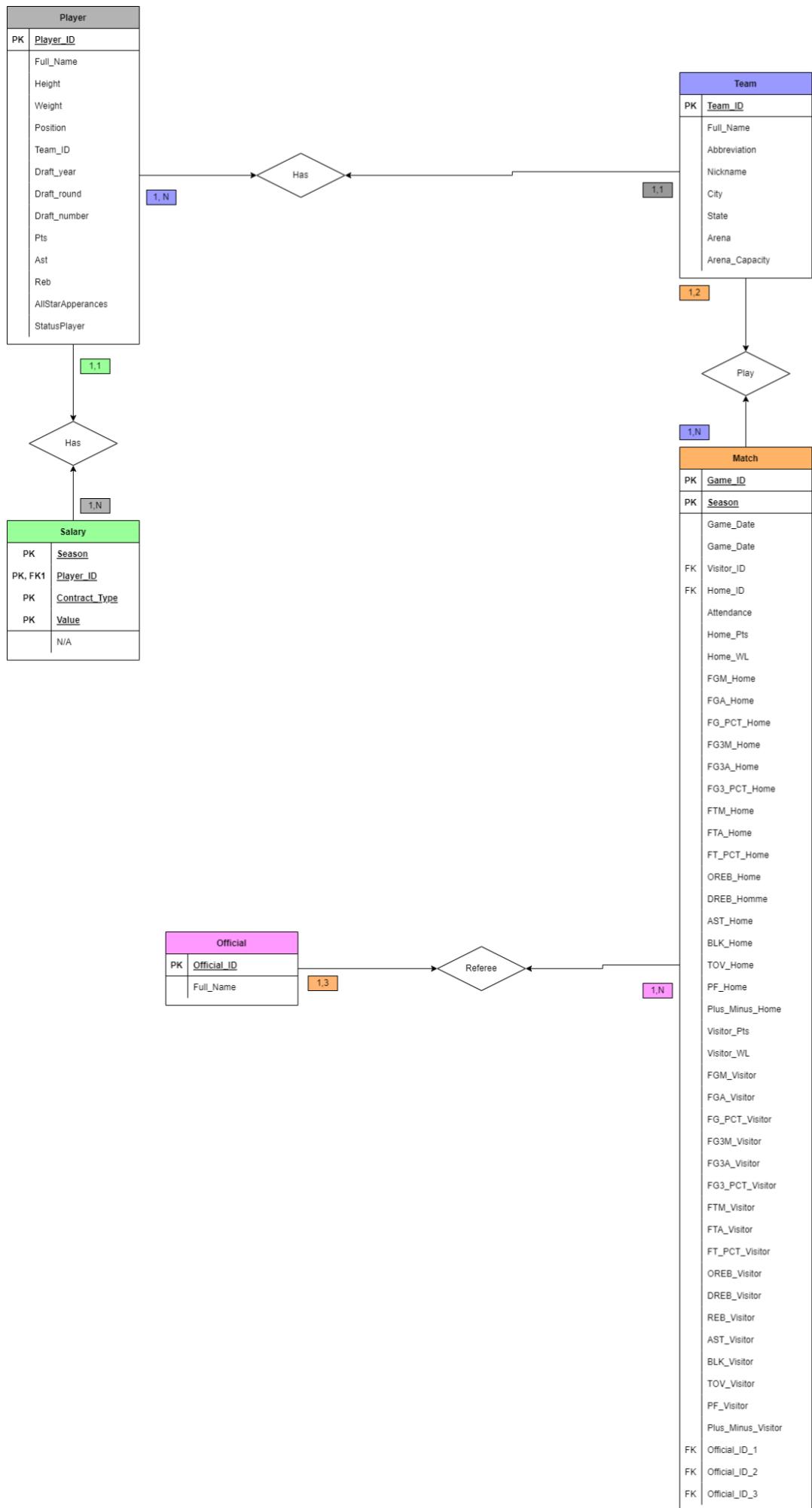
Universidad del Valle de Guatemala
Bases de datos
Ing. Jorge Yass

Diego Alonzo 20172
Paulo Sanchez 21401

Proyecto #1

Parte 1

Diagrama entidad relación



Pruebas de que se logró realizar la parte 01 del proyecto.

The screenshot shows a Visual Studio Code interface with the following details:

- File Explorer:** Shows a project structure for "PROYECTO1" containing files like "makingTablesInCSV.py", "connectDB.py", "Others", "ERDProyecto01DB.drawio", and various CSV files such as "DictionaryProject01.xlsx", "Draft_Combine.csv", "DraftCSV", "Game_Inactive_Players.csv", "GameOfficials.csv", "Game.csv", "News_Missing.csv", "News.csv", "OfficialsTable.csv", "Player_Attributes.csv", "Player_Salary.csv", "Player.csv", "PlayerTable.csv", "Salary.csv", "SalaryTable.csv", "Team_Attributes.csv", "Team_History.csv", "Team_Salary.csv", "Team.csv", "TeamCSV", and "TeamTable.csv".
- Code Editor:** Displays Python code for connecting to a database and creating tables. The code includes imports for `sqlite3` and `os`. It defines a function `createTableIfNotExists` that creates tables for Match, Team, Player, and Salary. The `Match` table structure is defined with columns: Game_ID (int), Season (int), Game_Date (date), Visitor_ID (int references Team(Team_ID) on delete cascade), Home_ID (int references Team(Team_ID) on delete cascade), Attendace (float), Home_Pts (varchar(6)), Home_WL (varchar(3)), FGM_Home (varchar(6)), FGA_Home (varchar(6)), FG_PCT_Home (varchar(6)), FG3M_Home (varchar(6)), FG3A_Home (varchar(6)), FG3_PCT_Home (varchar(6)), FTM_Home (varchar(6)), FTA_Home (varchar(6)), FT_PCT_Home (varchar(6)), OREB_Home (varchar(6)), DREB_Home (varchar(6)), BFR_Home (varchar(6)).
- Terminal:** Shows the command `python connectDB.py` being run, resulting in the message "Insertando Datos Salario".
- Output:** Shows the output of the terminal command.
- Problems:** Shows no errors or warnings.
- Status Bar:** Shows the current file path as "C:\Users\Windows 10\AppData\Local\Programs\Python\Python39\python.exe" "c:/Users/Windows 10/Documents/IMG/CODING/Semestre 4/D81/Proyectos/Proyecto1/Others/connectDB.py", the line number "Ln 105", column "Col 29", and the file "connectDB.py". It also shows the status "You now + Uncommitted changes".
- Bottom Bar:** Includes icons for file operations, search, and navigation.

The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database structure under 'Servers (1)'. The 'PostgreSQL 14' server contains 'Databases (3)': 'Lab05', 'postgres', and 'proyecto01'. 'proyecto01' is selected, revealing its contents: 'Casts', 'Catalogs', 'Event Triggers', 'Extensions', 'Foreign Data Wrappers', 'Languages', 'Publications', 'Schemas', and 'Subscriptions'. Below this is 'Login/Group Roles (12)' with several system roles listed. The main window shows a query editor with the following SQL code:

```
2 drop table Team cascade constraints
3 drop table Official CASCADE
4 drop table Salary
drop table Match

7 select *
from match
8
10 select*
11 from player
12
13 select *
from official
```

The 'Data output' tab is active, displaying a table with the following data:

game_id	season	game_date	visitor_id	home_id	attendance	home_pts	home_wl	fgm_home		
33	21400519	[PK] integer	2014	2015-01-05	1610612737	1610612746	9060	98	L	33.0
34	21400517	2014	2015-01-05	1610612747	1610612757	19827	98	W	33.0	
35	21400516	2014	2015-01-05	1610612754	1610612762	17378	101	L	35.0	
36	21400514	2014	2015-01-05	1610612743	1610612750	10386	101	L	38.0	
37	21400513	2014	2015-01-05	1610612752	1610612763	16888	105	W	43.0	
38	21400515	2014	2015-01-05	1610612764	1610612740	16182	85			

The status bar at the bottom right indicates 'Activate Windows' and 'PostgreSQL 14/proyecto01 - Database connected.' The bottom taskbar shows various application icons.

pgAdmin 4

File Object Tools Help

Servers (1) PostgreSQL 14 Databases (3) Lab05 postgres proyecto01

Properties SQL Statistics Dependencies proyecto01/postgres@PostgreSQL 14*

Query Query History

```

2 drop table Team cascade constraints
3 drop table Official CASCADE
4 drop table Salary
5 drop table Match
6
7 select *
8 from match
9
10 select*
11 from player
12
13 select *
14 from official

```

Data output Messages Notifications

player_id	[PK] integer	full_name	height	weight	position	team_id	draft_year	draft_round
1	76001	Alaa Abdelnaby	82	240	Forward	1610612757	1990	1
2	76002	Zaid Abdul-Aziz	81	235	Center	1610612745	1968	1
3	76003	Kareem Abdul-Jabbar	86	225	Center	1610612747	1969	1
4	51	Mahmoud Abdul-Rauf	73	162	Guard	1610612743	1990	1
5	1505	Tariq Abdul-Wahad	78	235	Forward-Guard			
6	949	Shareef Abdur-Rahim	81	245	Forward			

Total rows: 1000 of 3570 Query complete 00:00:00.211

Successfully run Total query runtime: 211 msec. 3570 PostgreSQL 14/proyecto01 - Database connected.

Ln 10, Col 1

pgAdmin 4

File Object Tools Help

Servers (1) PostgreSQL 14 Databases (3) Lab05 postgres proyecto01

Properties SQL Statistics Dependencies proyecto01/postgres@PostgreSQL 14*

Query Query History

```

5 drop table Match
6
7 select *
8 from match
9
10 select*
11 from player
12
13 select *
14 from official
15
16 select *
17 from salary

```

Data output Messages Notifications

official_id	[PK] integer	full_name
1	1165	Luis Grillo
2	1140	Bruce Alexander
3	1153	Joe Crawford
4	1146	Tony Brothers
5	1157	Terry Durham
6	1177	Ed Middleton

Total rows: 143 of 143 Query complete 00:00:00.112

Successfully run Total query runtime: 112 msec. 143 rows affected. PostgreSQL 14/proyecto01 - Database connected.

Ln 13, Col 1

pgAdmin 4

File Object Tools Help

Properties SQL Statistics Dependencies proyecto01/postgres@PostgreSQL 14*

Servers (1) PostgreSQL 14 Databases (3) Lab05 postgres proyecto01 Casts Catalogs Event Triggers Extensions Foreign Data Wrappers Languages Publications Schemas Subscriptions Login/Group Roles (12) pg_database_owner pg_execute_server_program pg_monitor pg_read_all_data pg_read_all_settings pg_read_all_stats pg_read_server_files pg_signal_backend pg_stat_scan_tables pg_write_all_data no_write_server_files

```

9
10 select*
11 from player
12
13 select *
14 from official
15
16 select *
17 from salary
18
19 select *
20 from team

```

Data output Messages Notifications

session	player_id	contract_type	value
1 2020-21	203992	Guaranteed	18000000
2 2021-22	203992	Guaranteed	18000000
3 2022-23	203992	Guaranteed	18000000
4 2023-24	203992	Player Option	18000000
5 2020-21	1629164	Guaranteed	1701593
6 2021-22	1629164	Qualifying Offer	2126991

Total rows: 922 of 922 Query complete 00:00:00.143

Activate Windows

PostgreSQL 14/proyecto01 - Database connected.

12:15 AM 8/23/2022 Ln 16, Col 1

pgAdmin 4

File Object Tools Help

Properties SQL Statistics Dependencies proyecto01/postgres@PostgreSQL 14*

Servers (1) PostgreSQL 14 Databases (3) Lab05 postgres proyecto01 Casts Catalogs Event Triggers Extensions Foreign Data Wrappers Languages Publications Schemas Subscriptions Login/Group Roles (12) pg_database_owner pg_execute_server_program pg_monitor pg_read_all_data pg_read_all_settings pg_read_all_stats pg_read_server_files pg_signal_backend pg_stat_scan_tables pg_write_all_data no_write_server_files

```

9
10 select*
11 from player
12
13 select *
14 from official
15
16 select *
17 from salary
18
19 select *
20 from team

```

Data output Messages Notifications

team_id	full_name	abbreviation	nickname	city	state	arena
1 1610612737	Atlanta Hawks	ATL	Hawks	Atlanta	Atlanta	State Farm Arena
2 1610612738	Boston Celtics	BOS	Celtics	Boston	Massachusetts	TD Garden
3 1610612739	Cleveland Cavaliers	CLE	Cavaliers	Cleveland	Ohio	Rocket Mortgage FieldHouse
4 1610612740	New Orleans Pelicans	NOP	Pelicans	New Orleans	Louisiana	Smoothie King Center
5 1610612741	Chicago Bulls	CHI	Bulls			
6 1610612742	Dallas Mavericks	DAL	Mavericks	Dallas		

Total rows: 30 of 30 Query complete 00:00:00.164

Successfully run. Total query runtime: 164 msec. 30 rows affected.

PostgreSQL 14/proyecto01 - Database connected.

12:15 AM 8/23/2022 Ln 19, Col 1

Parte 2

1. ¿Quién es el jugador más alto?

The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database structure for 'proyecto01'. The main window contains a SQL query and its results. The SQL code is:

```
6
7 select a.Player_ID, a.Full_Name, a.Height
8 from Player as a
9 where a.StatusPlayer = 'Active'
10 order by a.Height desc
11 limit (1)
12
13 select a.Player_ID, a.Full_Name, a.Height
14 from Player as a
15 where a.StatusPlayer = 'Active'
16 order by a.Height asc
17 limit (1)
```

The results table shows one row:

	player_id	full_name	height
1	1629607	Jared Harper	70

Below the table, the status bar indicates "Total rows: 1 of 1 Query complete 00:00:00.179".

1629605 "Tacko Fall" 89

2. ¿Quién es el jugador más bajo?

The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database structure for 'proyecto01'. The main window contains a SQL query and its results. The SQL code is identical to the previous query:

```
6
7 select a.Player_ID, a.Full_Name, a.Height
8 from Player as a
9 where a.StatusPlayer = 'Active'
10 order by a.Height desc
11 limit (1)
12
13 select a.Player_ID, a.Full_Name, a.Height
14 from Player as a
15 where a.StatusPlayer = 'Active'
16 order by a.Height asc
17 limit (1)
```

The results table shows one row:

	player_id	full_name	height
1	1629605	Tacko Fall	89

Below the table, the status bar indicates "Total rows: 1 of 1 Query complete 00:00:00.165". A message bar at the bottom right says "Successfully run. Total query runtime: 165 msec. 1 rows affected."

1629607 "Jared Harper"70

3. ¿Cuál fue el promedio de puntos anotados y recibidos en cada una de las temporadas relevantes? El promedio de puntos anotados en cada temporada relevante [2015-2020] para cada equipo es:

pgAdmin 4

PgAdmin File Object Tools Help

Browser Properties SQL Statistics Dependencies proyecto01/postgres@PostgreSQL 14*

Query Query History

```

224     order by avg(cast(m pts as INT)) desc
225
226 select m.season, t.full_name as Team, avg(cast(m pts as INT)) as Promedio_Puntos_Anotados
227 from team as t
228 inner join (
229     select home_pts as pts, home_id as id, season from match where season >=2015
230 union all
231     select visitor_pts as pts, visitor_id as id, season from match where season >=2015
232 ) as m
233 on m.id = t.team_id
234 group by m.season, m.id, t.full_name
235 order by m.id, m.season asc

```

Data output Messages Notifications

	season	team	promedio_puntos_anotados
1	2015	Atlanta Hawks	102.84146341463415
2	2016	Atlanta Hawks	103.15853658536585
3	2017	Atlanta Hawks	103.35365853658537
4	2018	Atlanta Hawks	113.34146341463415
5	2019	Atlanta Hawks	111.76119402985074
6	2020	Atlanta Hawks	113.04166666666667

Total rows: 180 of 180 Query complete 00:00:00.127

Activate Windows Go to Settings to activate Windows.

Ln 226, Col 1

2015	"Atlanta Hawks"	102.84146341463415
2016	"Atlanta Hawks"	103.15853658536585
2017	"Atlanta Hawks"	103.35365853658537
2018	"Atlanta Hawks"	113.34146341463415
2019	"Atlanta Hawks"	111.76119402985074
2020	"Atlanta Hawks"	113.04166666666667
2015	"Boston Celtics"	105.71951219512195
2016	"Boston Celtics"	108.01219512195122
2017	"Boston Celtics"	104.01219512195122
2018	"Boston Celtics"	112.39024390243902
2019	"Boston Celtics"	113.65277777777777
2020	"Boston Celtics"	112.625
2015	"Cleveland Cavaliers"	104.32926829268293
2016	"Cleveland Cavaliers"	110.34146341463415
2017	"Cleveland Cavaliers"	110.86585365853658
2018	"Cleveland Cavaliers"	104.47560975609755
2019	"Cleveland Cavaliers"	106.8923076923077
2020	"Cleveland Cavaliers"	103.83333333333333
2015	"New Orleans Pelicans"	102.71951219512195
2016	"New Orleans Pelicans"	104.34146341463415
2017	"New Orleans Pelicans"	111.71951219512195
2018	"New Orleans Pelicans"	115.4390243902439
2019	"New Orleans Pelicans"	115.84722222222223
2020	"New Orleans Pelicans"	114.34722222222223
2015	"Chicago Bulls"	101.64634146341463
2016	"Chicago Bulls"	102.86585365853658
2017	"Chicago Bulls"	102.92682926829268
2018	"Chicago Bulls"	104.9390243902439
2019	"Chicago Bulls"	106.84615384615384

2020	"Chicago Bulls"	110.68055555555556
2015	"Dallas Mavericks"	102.29268292682927
2016	"Dallas Mavericks"	97.91463414634147
2017	"Dallas Mavericks"	102.3170731707317
2018	"Dallas Mavericks"	108.86585365853658
2019	"Dallas Mavericks"	117.01333333333334
2020	"Dallas Mavericks"	110.97222222222223
2015	"Denver Nuggets"	101.8641975308642
2016	"Denver Nuggets"	111.55555555555556
2017	"Denver Nuggets"	110
2018	"Denver Nuggets"	110.67073170731707
2019	"Denver Nuggets"	111.28767123287672
2020	"Denver Nuggets"	112.4722222222223
2015	"Golden State Warriors"	114.89024390243902
2016	"Golden State Warriors"	115.89024390243902
2017	"Golden State Warriors"	113.46341463414635
2018	"Golden State Warriors"	117.6829268292683
2019	"Golden State Warriors"	106.33846153846154
2020	"Golden State Warriors"	110.80555555555556
2015	"Houston Rockets"	106.54878048780488
2016	"Houston Rockets"	115.34146341463415
2017	"Houston Rockets"	112.35365853658537
2018	"Houston Rockets"	113.91463414634147
2019	"Houston Rockets"	117.80555555555556
2020	"Houston Rockets"	105.90277777777777
2015	"Los Angeles Clippers"	104.5
2016	"Los Angeles Clippers"	108.67073170731707
2017	"Los Angeles Clippers"	108.98780487804878
2018	"Los Angeles Clippers"	115.14634146341463
2019	"Los Angeles Clippers"	116.3472222222223
2020	"Los Angeles Clippers"	110.8472222222223
2015	"Los Angeles Lakers"	97.34146341463415
2016	"Los Angeles Lakers"	104.57317073170732
2017	"Los Angeles Lakers"	108.07317073170732
2018	"Los Angeles Lakers"	111.76829268292683
2019	"Los Angeles Lakers"	113.43661971830986
2020	"Los Angeles Lakers"	104.5972222222223
2015	"Miami Heat"	100.04878048780488
2016	"Miami Heat"	103.17073170731707
2017	"Miami Heat"	103.41463414634147
2018	"Miami Heat"	105.70731707317073
2019	"Miami Heat"	112.04109589041096
2020	"Miami Heat"	108.06944444444444
2015	"Milwaukee Bucks"	99.04878048780488
2016	"Milwaukee Bucks"	103.6219512195122
2017	"Milwaukee Bucks"	106.47560975609755

2018	"Milwaukee Bucks"	118.1219512195122
2019	"Milwaukee Bucks"	118.67123287671232
2020	"Milwaukee Bucks"	120.125
2015	"Minnesota Timberwolves"	102.41463414634147
2016	"Minnesota Timberwolves"	105.57317073170732
2017	"Minnesota Timberwolves"	109.51219512195122
2018	"Minnesota Timberwolves"	112.47560975609755
2019	"Minnesota Timberwolves"	113.25
2020	"Minnesota Timberwolves"	111.23611111111111
2015	"Brooklyn Nets"	98.64634146341463
2016	"Brooklyn Nets"	105.76829268292683
2017	"Brooklyn Nets"	106.59756097560975
2018	"Brooklyn Nets"	112.2439024390244
2019	"Brooklyn Nets"	111.77777777777777
2020	"Brooklyn Nets"	117.33333333333333
2015	"New York Knicks"	98.35365853658537
2016	"New York Knicks"	104.34146341463415
2017	"New York Knicks"	104.46341463414635
2018	"New York Knicks"	104.57317073170732
2019	"New York Knicks"	105.8030303030303
2020	"New York Knicks"	106.22222222222223
2015	"Orlando Magic"	102.0609756097561
2016	"Orlando Magic"	100.93827160493827
2017	"Orlando Magic"	103.40243902439025
2018	"Orlando Magic"	107.3170731707317
2019	"Orlando Magic"	107.27397260273973
2020	"Orlando Magic"	104
2015	"Indiana Pacers"	102.15853658536585
2016	"Indiana Pacers"	105.09756097560975
2017	"Indiana Pacers"	105.5609756097561
2018	"Indiana Pacers"	108.01219512195122
2019	"Indiana Pacers"	109.43835616438356
2020	"Indiana Pacers"	113.95833333333333
2015	"Philadelphia 76ers"	97.34567901234568
2016	"Philadelphia 76ers"	102.4390243902439
2017	"Philadelphia 76ers"	109.8048780487805
2018	"Philadelphia 76ers"	115.1829268292683
2019	"Philadelphia 76ers"	110.73972602739725
2020	"Philadelphia 76ers"	113.63888888888889
2015	"Phoenix Suns"	100.86585365853658
2016	"Phoenix Suns"	107.6951219512195
2017	"Phoenix Suns"	103.92682926829268
2018	"Phoenix Suns"	107.5
2019	"Phoenix Suns"	113.61643835616438
2020	"Phoenix Suns"	113
2015	"Portland Trail Blazers"	105.14634146341463

2016	"Portland Trail Blazers"	107.9390243902439
2017	"Portland Trail Blazers"	105.6219512195122
2018	"Portland Trail Blazers"	114.65853658536585
2019	"Portland Trail Blazers"	114.97297297297297
2020	"Portland Trail Blazers"	113.34722222222223
2015	"Sacramento Kings"	106.58536585365853
2016	"Sacramento Kings"	102.8048780487805
2017	"Sacramento Kings"	98.82926829268293
2018	"Sacramento Kings"	114.1829268292683
2019	"Sacramento Kings"	110.09722222222223
2020	"Sacramento Kings"	109.18055555555556
2015	"San Antonio Spurs"	103.53658536585365
2016	"San Antonio Spurs"	105.32926829268293
2017	"San Antonio Spurs"	102.73170731707317
2018	"San Antonio Spurs"	111.65853658536585
2019	"San Antonio Spurs"	114.05633802816901
2020	"San Antonio Spurs"	109.69444444444444
2015	"Oklahoma City Thunder"	110.21951219512195
2016	"Oklahoma City Thunder"	106.59756097560975
2017	"Oklahoma City Thunder"	107.85365853658537
2018	"Oklahoma City Thunder"	114.47560975609755
2019	"Oklahoma City Thunder"	110.41666666666667
2020	"Oklahoma City Thunder"	103.88888888888889
2015	"Toronto Raptors"	102.70731707317073
2016	"Toronto Raptors"	106.85365853658537
2017	"Toronto Raptors"	111.65853658536585
2018	"Toronto Raptors"	114.4390243902439
2019	"Toronto Raptors"	112.75
2020	"Toronto Raptors"	110.13888888888889
2015	"Utah Jazz"	97.6829268292683
2016	"Utah Jazz"	100.70731707317073
2017	"Utah Jazz"	104.14634146341463
2018	"Utah Jazz"	111.71951219512195
2019	"Utah Jazz"	111.29166666666667
2020	"Utah Jazz"	112.55555555555556
2015	"Memphis Grizzlies"	99.09756097560975
2016	"Memphis Grizzlies"	100.47560975609755
2017	"Memphis Grizzlies"	99.32926829268293
2018	"Memphis Grizzlies"	103.53658536585365
2019	"Memphis Grizzlies"	112.63013698630137
2020	"Memphis Grizzlies"	112.73611111111111
2015	"Washington Wizards"	104.07317073170732
2016	"Washington Wizards"	109.1829268292683
2017	"Washington Wizards"	106.60975609756098
2018	"Washington Wizards"	114.02439024390245
2019	"Washington Wizards"	114.41666666666667

2020	"Washington Wizards"	116.63888888888889
2015	"Detroit Pistons"	101.96341463414635
2016	"Detroit Pistons"	101.32926829268293
2017	"Detroit Pistons"	103.76829268292683
2018	"Detroit Pistons"	107.04878048780488
2019	"Detroit Pistons"	107.24242424242425
2020	"Detroit Pistons"	105.31944444444444
2015	"Charlotte Hornets"	103.40243902439025
2016	"Charlotte Hornets"	104.89024390243902
2017	"Charlotte Hornets"	108.21951219512195
2018	"Charlotte Hornets"	110.7439024390244
2019	"Charlotte Hornets"	102.87692307692308
2020	"Charlotte Hornets"	109.45833333333333

Mientras que el promedio de puntos recibidos para cada equipo en cada temporada relevante [2015-2020] es:

```

pgAdmin 4
File Object Tools Help
Browser Properties SQL Statistics Dependencies proyecto01/postgres@PostgreSQL 14*
Query Query History
--+
233    on m.id = t.team_id
234    group by m.season, m.id, t.full_name
235    order by m.id, m.season asc
236
237 select m.season, t.full_name as Team, avg(cast(m pts as INT)) as Promedio_Puntos_Recibidos
238 from team as t
239 inner join (
240     select visitor_pts as pts, home_id as id, visitor_id, season from match where season >=2015
241 ) as m
242 on m.id = t.team_id
243 group by m.season, m.id, t.full_name
244 order by m.id, m.season asc
245

```

Data output

season	team	promedio_puntos_recibidos
1	2015 Atlanta Hawks	97.6829268292683
2	2016 Atlanta Hawks	105.82926829268293
3	2017 Atlanta Hawks	108.8048780487805
4	2018 Atlanta Hawks	119.8048780487805
5	2019 Atlanta Hawks	117.3529411764706
6	2020 Atlanta Hawks	109.08333333333333

Total rows: 180 of 180 | Query complete 00:00:00.128

2015	"Atlanta Hawks"	97.6829268292683
2016	"Atlanta Hawks"	105.82926829268293
2017	"Atlanta Hawks"	108.8048780487805
2018	"Atlanta Hawks"	119.8048780487805
2019	"Atlanta Hawks"	117.3529411764706
2020	"Atlanta Hawks"	109.08333333333333
2015	"Boston Celtics"	100.48780487804878
2016	"Boston Celtics"	105.4390243902439
2017	"Boston Celtics"	100.65853658536585
2018	"Boston Celtics"	107.3170731707317
2019	"Boston Celtics"	108.05555555555556
2020	"Boston Celtics"	111.08333333333333
2015	"Cleveland Cavaliers"	98.39024390243902
2016	"Cleveland Cavaliers"	104.5609756097561

2017	"Cleveland Cavaliers"	110.36585365853658
2018	"Cleveland Cavaliers"	112.51219512195122
2019	"Cleveland Cavaliers"	115.05555555555556
2020	"Cleveland Cavaliers"	114.33333333333333
2015	"New Orleans Pelicans"	106.58536585365853
2016	"New Orleans Pelicans"	107.07317073170732
2017	"New Orleans Pelicans"	111.8780487804878
2018	"New Orleans Pelicans"	116
2019	"New Orleans Pelicans"	116.30555555555556
2020	"New Orleans Pelicans"	113.72222222222223
2015	"Chicago Bulls"	100.1951219512195
2016	"Chicago Bulls"	100.6829268292683
2017	"Chicago Bulls"	107.90243902439025
2018	"Chicago Bulls"	113.39024390243902
2019	"Chicago Bulls"	107.61764705882354
2020	"Chicago Bulls"	111.13888888888889
2015	"Dallas Mavericks"	101.95121951219512
2016	"Dallas Mavericks"	97.8780487804878
2017	"Dallas Mavericks"	103.90243902439025
2018	"Dallas Mavericks"	107.97560975609755
2019	"Dallas Mavericks"	112.8157894736842
2020	"Dallas Mavericks"	109.55555555555556
2015	"Denver Nuggets"	106.075
2016	"Denver Nuggets"	110.675
2017	"Denver Nuggets"	105.63414634146342
2018	"Denver Nuggets"	103.60975609756098
2019	"Denver Nuggets"	107.5945945945946
2020	"Denver Nuggets"	108.02777777777777
2015	"Golden State Warriors"	101.8780487804878
2016	"Golden State Warriors"	102.8780487804878
2017	"Golden State Warriors"	105.5609756097561
2018	"Golden State Warriors"	111.51219512195122
2019	"Golden State Warriors"	115.67647058823529
2020	"Golden State Warriors"	105.86111111111111
2015	"Houston Rockets"	104.58536585365853
2016	"Houston Rockets"	108.34146341463415
2017	"Houston Rockets"	104.97560975609755
2018	"Houston Rockets"	109.6829268292683
2019	"Houston Rockets"	112.36111111111111
2020	"Houston Rockets"	114.88888888888889
2015	"Los Angeles Clippers"	97.92682926829268
2016	"Los Angeles Clippers"	100.7560975609756
2017	"Los Angeles Clippers"	108.6829268292683
2018	"Los Angeles Clippers"	114.60975609756098
2019	"Los Angeles Clippers"	109.02777777777777
2020	"Los Angeles Clippers"	103.63888888888889

2015	"Los Angeles Lakers"	103
2016	"Los Angeles Lakers"	109.1219512195122
2017	"Los Angeles Lakers"	105.95121951219512
2018	"Los Angeles Lakers"	111.17073170731707
2019	"Los Angeles Lakers"	106.37142857142857
2020	"Los Angeles Lakers"	101.38888888888889
2015	"Miami Heat"	97.85365853658537
2016	"Miami Heat"	102.51219512195122
2017	"Miami Heat"	102.58536585365853
2018	"Miami Heat"	105.78048780487805
2019	"Miami Heat"	106.97222222222223
2020	"Miami Heat"	108.58333333333333
2015	"Milwaukee Bucks"	101.8780487804878
2016	"Milwaukee Bucks"	103.53658536585365
2017	"Milwaukee Bucks"	105.8048780487805
2018	"Milwaukee Bucks"	107.78048780487805
2019	"Milwaukee Bucks"	107.97142857142858
2020	"Milwaukee Bucks"	113.08333333333333
2015	"Minnesota Timberwolves"	105.3170731707317
2016	"Minnesota Timberwolves"	103.6829268292683
2017	"Minnesota Timberwolves"	105.7560975609756
2018	"Minnesota Timberwolves"	111.7560975609756
2019	"Minnesota Timberwolves"	115.34375
2020	"Minnesota Timberwolves"	116.38888888888889
2015	"Brooklyn Nets"	104
2016	"Brooklyn Nets"	109.92682926829268
2017	"Brooklyn Nets"	108.95121951219512
2018	"Brooklyn Nets"	111.29268292682927
2019	"Brooklyn Nets"	110.61111111111111
2020	"Brooklyn Nets"	111.36111111111111
2015	"New York Knicks"	100.3170731707317
2016	"New York Knicks"	106.46341463414635
2017	"New York Knicks"	105.73170731707317
2018	"New York Knicks"	113.7560975609756
2019	"New York Knicks"	110.48484848484848
2020	"New York Knicks"	105.13888888888889
2015	"Orlando Magic"	102.17073170731707
2016	"Orlando Magic"	103.8048780487805
2017	"Orlando Magic"	107.3170731707317
2018	"Orlando Magic"	106.6829268292683
2019	"Orlando Magic"	106.4
2020	"Orlando Magic"	114.41666666666667
2015	"Indiana Pacers"	98.85365853658537
2016	"Indiana Pacers"	101.6829268292683
2017	"Indiana Pacers"	103.92682926829268
2018	"Indiana Pacers"	101.04878048780488

2019	"Indiana Pacers"	105.91666666666667
2020	"Indiana Pacers"	113
2015	"Philadelphia 76ers"	107.70731707317073
2016	"Philadelphia 76ers"	105.95121951219512
2017	"Philadelphia 76ers"	103.41463414634147
2018	"Philadelphia 76ers"	110.1951219512195
2019	"Philadelphia 76ers"	104.65714285714286
2020	"Philadelphia 76ers"	108.52777777777777
2015	"Phoenix Suns"	105.8048780487805
2016	"Phoenix Suns"	111.92682926829268
2017	"Phoenix Suns"	113.46341463414635
2018	"Phoenix Suns"	113.85365853658537
2019	"Phoenix Suns"	112.61538461538461
2020	"Phoenix Suns"	103.80555555555556
2015	"Portland Trail Blazers"	102.34146341463415
2016	"Portland Trail Blazers"	106.8780487804878
2017	"Portland Trail Blazers"	102.1951219512195
2018	"Portland Trail Blazers"	109.82926829268293
2019	"Portland Trail Blazers"	115.33333333333333
2020	"Portland Trail Blazers"	110
2015	"Sacramento Kings"	108.6829268292683
2016	"Sacramento Kings"	106.90243902439025
2017	"Sacramento Kings"	105.95121951219512
2018	"Sacramento Kings"	113.48780487804878
2019	"Sacramento Kings"	112.22857142857143
2020	"Sacramento Kings"	111.83333333333333
2015	"San Antonio Spurs"	91.17073170731707
2016	"San Antonio Spurs"	97.46341463414635
2017	"San Antonio Spurs"	98.1219512195122
2018	"San Antonio Spurs"	105.95121951219512
2019	"San Antonio Spurs"	112.94117647058823
2020	"San Antonio Spurs"	113.58333333333333
2015	"Oklahoma City Thunder"	99.6829268292683
2016	"Oklahoma City Thunder"	103.58536585365853
2017	"Oklahoma City Thunder"	103.5609756097561
2018	"Oklahoma City Thunder"	109.63414634146342
2019	"Oklahoma City Thunder"	109.02702702702703
2020	"Oklahoma City Thunder"	117.97222222222223
2015	"Toronto Raptors"	98.39024390243902
2016	"Toronto Raptors"	103.1951219512195
2017	"Toronto Raptors"	101.73170731707317
2018	"Toronto Raptors"	107.58536585365853
2019	"Toronto Raptors"	107.02777777777777
2020	"Toronto Raptors"	109.58333333333333
2015	"Utah Jazz"	93.04878048780488
2016	"Utah Jazz"	94.8048780487805

2017	"Utah Jazz"	96.73170731707317
2018	"Utah Jazz"	104.78048780487805
2019	"Utah Jazz"	107.6
2020	"Utah Jazz"	100.30555555555556
2015	"Memphis Grizzlies"	101.48780487804878
2016	"Memphis Grizzlies"	97.1219512195122
2017	"Memphis Grizzlies"	102.95121951219512
2018	"Memphis Grizzlies"	104.17073170731707
2019	"Memphis Grizzlies"	112.13513513513513
2020	"Memphis Grizzlies"	110.61111111111111
2015	"Washington Wizards"	103.48780487804878
2016	"Washington Wizards"	105.48780487804878
2017	"Washington Wizards"	105.29268292682927
2018	"Washington Wizards"	114.65853658536585
2019	"Washington Wizards"	115.86111111111111
2020	"Washington Wizards"	119.08333333333333
2015	"Detroit Pistons"	100.04878048780488
2016	"Detroit Pistons"	98.92682926829268
2017	"Detroit Pistons"	102.60975609756098
2018	"Detroit Pistons"	106.09756097560975
2019	"Detroit Pistons"	111.36363636363636
2020	"Detroit Pistons"	108.13888888888889
2015	"Charlotte Hornets"	99.09756097560975
2016	"Charlotte Hornets"	103.36585365853658
2017	"Charlotte Hornets"	108
2018	"Charlotte Hornets"	109.46341463414635
2019	"Charlotte Hornets"	109.51612903225806
2020	"Charlotte Hornets"	108.5

4. Muestre el top 5 árbitros que cuando pitán pierde el equipo visitante.

```

pgAdmin 4
File Object Tools Help
Browser Properties SQL Statistics Dependencies proyecto01/postgres@PostgreSQL 14*
Query History
125 select o.full_name as Official_Name, count(m.wl) as Number_Of_Loses_With_This_Referee
126 from official as o
127 inner join (
128 select official_ID_3 as id, visitor_WL as wl from match where visitor_WL='L' and season>=2015
129 union all
130 select official_ID_2, visitor_WL from match where visitor_WL='L' and season>=2015
131 union all
132 select official_ID_1, visitor_WL from match where visitor_WL='L' and season>=2015 ) as m
133 on m.id = o.official_id
134 group by m.id, o.full_name
135 order by count(m.wl) desc
136 Limit (5)
137
Data output Messages Notifications
+-----+-----+
| official_name | number_of_losses_with_this_referee |
+-----+-----+
| James Capers | 218 |
| Mark Ayotte | 212 |
| Scott Foster | 212 |
| Tony Brothers | 212 |
| John Goble | 211 |
+-----+
Total rows: 5 of 5 Query complete 00:00:00.185

```

"James Capers"	218
"Mark Ayotte"	212
"Scott Foster"	212
"Tony Brothers"	212
"John Goble"	211

5. ¿Qué equipo maneja los salarios más altos actualmente?

```

pgAdmin 4
File Object Tools Help
Browser Properties SQL Statistics Dependencies proyecto01/postgres@PostgreSQL 14*
Query History
88 group by s.player_id
89
90 select t.full_name as Team, sum(s.value) as Total_Salaries
91 from salary as s
92 inner join player as p
93 on s.player_id = p.player_id
94 inner join team as t
95 on t.team_id = p.team_id
96 where s.season = '2020-21'
97 group by p.team_id, t.full_name
98 order by sum(s.value) desc
99 Limit(1)
100
Data output Messages Notifications
+-----+-----+
| team | total_salaries |
+-----+-----+
| Brooklyn Nets | 195260760 |
+-----+
Total rows: 1 of 1 Query complete 00:00:00.136

```

El equipo que tiene los salarios más altos es "Brooklyn Nets" con un total de \$195260760 invertidos en salarios para la temporada 2020-21.
 Así mismo, esto es comprobado debido a que en la media de salarios, también tienen la más alta con un total de \$13017384 por jugador.

The screenshot shows the pgAdmin 4 interface. On the left is the 'Browser' pane with a tree view of database objects. In the center is the 'Query' pane displaying a SQL query and its results. The query retrieves the average salary for each team in the 2020-21 season. The results table has two columns: 'team' and 'total_salaries'. The result is for the Brooklyn Nets with a total salary of 13017384.

```

99 Limit(1)
100
101 select t.full_name as Team, avg(s.value) as Total_Salaries
102 from salary as s
103 inner join player as p
104 on s.player_id = p.player_id
105 inner join team as t
106 on t.team_id = p.team_id
107 where s.season ='2020-21'
108 group by p.team_id, t.full_name
109 order by avg(s.value) desc
110 Limit(1)
111

```

team	total_salaries
Brooklyn Nets	13017384

6. ¿Cuál fue la temporada con más partidos en la historia de la NBA? Se debe de destacar que solo se tomará en cuenta las temporadas entre [2015-2020] debido a que en las instrucciones del proyecto se especificó que eran de estas temporadas de las que nos teníamos que basar.

The screenshot shows the pgAdmin 4 interface. The 'Browser' pane lists database objects. The 'Query' pane displays a SQL query that joins the 'match' table with itself to calculate the count of distinct games per season. It filters for seasons from 2015 to 2020 and orders the results by the count of games in descending order. The results table shows seasons from 2017 to 2019 with their respective counts of NBA matches.

```

239 inner join (
240     select visitor_pts as pts, home_id as id, visitor_id, season from match where season >=2015
241 ) as m
242 on m.id = t.team_id
243 group by m.season, m.id, t.full_name
244 order by m.id, m.season asc
245
246 select season, count(distinct m.game_id) as Count_Of_NBA_Matches
247 from match as m
248 where season >= '2015'
249 group by season
250 Order by count(distinct m.game_id) desc

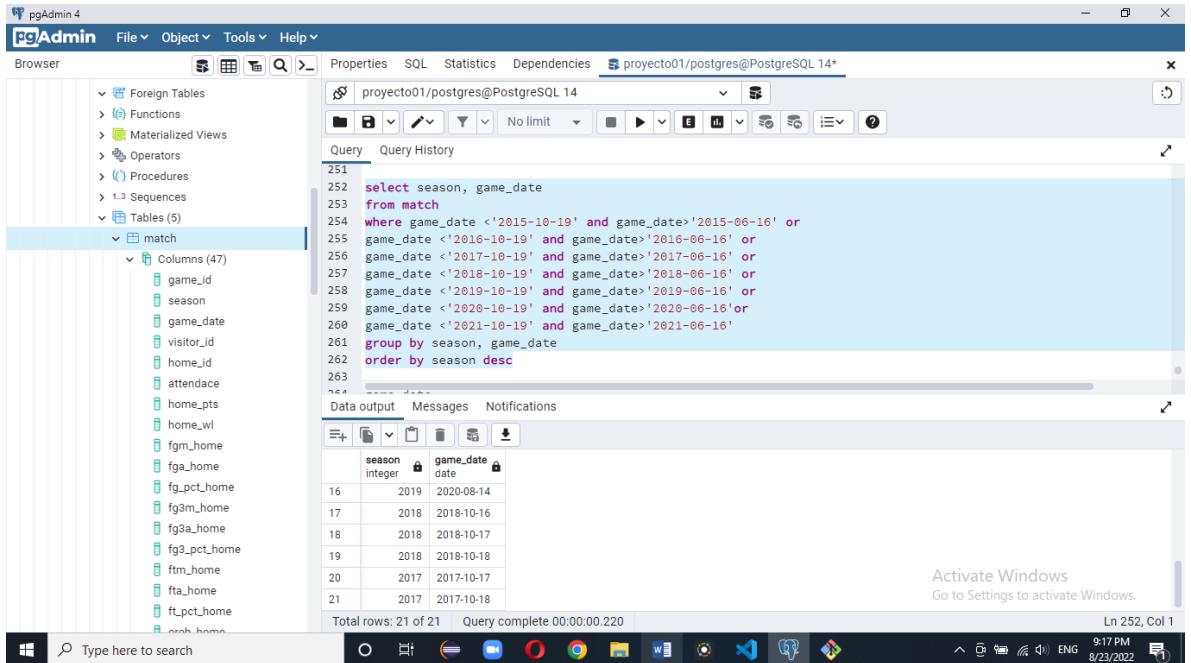
```

season	count_of_nba_matches
1	1230
2	1230
3	1229
4	1229
5	1080
6	1059

2017 1230
 2018 1230
 2015 1229
 2016 1229
 2020 1080
 2019 1059

Por ende la temporada 2017 y la temporada 2018 son las temporadas con más partidos con 1230 partidos realizados.

7. ¿Cuál fue la temporada en la que más se han prolongado las fechas de los juegos?
 Basándonos en el hecho de que la temporada regular de la NBA abarca desde el 19 de octubre de 2021 hasta el 10 de abril de 2022, luego hay play in del 12 al 15 de abril, play offs del 16 de abril al 30 de mayo y las finales son del 2 al 16 de Junio de 2022, una temporada regular estaría desde el año anterior-10-19 hasta el año actual-06-16.



```

    select season, game_date
    from match
   where game_date <'2015-10-19' and game_date>'2015-06-16' or
         game_date <'2016-10-19' and game_date>'2016-06-16' or
         game_date <'2017-10-19' and game_date>'2017-06-16' or
         game_date <'2018-10-19' and game_date>'2018-06-16' or
         game_date <'2019-10-19' and game_date>'2019-06-16' or
         game_date <'2020-10-19' and game_date>'2020-06-16'or
         game_date <'2021-10-19' and game_date>'2021-06-16'
  group by season, game_date
  order by season desc
  
```

season	game_date
16	2019-08-14
17	2018-10-16
18	2018-10-17
19	2018-10-18
20	2017-10-17
21	2017-10-18

Por ende, la temporada que más se prolongaron las fechas fue, la temporada 2017 y 2018 dado que la temporada tuvo fechas hasta el 18 de octubre. Un día antes que terminara la temporada.

```

2019 "2020-07-30"
2019 "2020-07-31"
2019 "2020-08-01"
2019 "2020-08-02"
2019 "2020-08-03"
2019 "2020-08-04"
2019 "2020-08-05"
2019 "2020-08-06"
2019 "2020-08-07"
2019 "2020-08-08"
2019 "2020-08-09"
2019 "2020-08-10"
2019 "2020-08-11"
2019 "2020-08-12"
2019 "2020-08-13"
2019 "2020-08-14"
2018 "2018-10-16"
2018 "2018-10-17"
2018 "2018-10-18"
2017 "2017-10-17"
2017 "2017-10-18"
  
```

8. ¿Cuál es el equipo que más diferencia de puntos a favor obtuvo en promedio por partido para la temporada 2017?

The screenshot shows the pgAdmin 4 interface with a query editor and a results grid. The query retrieves the team with the highest average difference in points for the 2017 season.

```

260 game_date < '2021-10-19' AND game_date > '2021-06-16'
261 GROUP BY season, game_date
262 ORDER BY season DESC
263
264 SELECT team.full_name AS team, AVG(CAST(home_pts AS INT) - CAST(visitor_pts AS INT)) AS Average_To_Favor_Points
265 FROM match
266 INNER JOIN team
267 ON team.team_id = match.home_id
268 WHERE match.season = '2017'
269 GROUP BY match.home_id, team.full_name
270 ORDER BY Average_To_Favor_Points DESC
271 LIMIT (1)
  
```

team	average_to_favor_points
Toronto Raptors	10.439024390243903

Total rows: 1 of 1 | Query complete 00:00:00.188 | Ln 271, Col 10 | Successfully run. Total query runtime: 188 msec. 1 rows affected.

Por ende fue Toronto Raptors quien tuvo la diferencia de puntos más favorable en la temporada 2017 con 10.44 puntos a favor por partido jugado.

9. ¿Cuál es el equipo con más diferencia de punto a favor para la temporada 2018?

The screenshot shows the pgAdmin 4 interface with a query editor and a results grid. The query retrieves the team with the highest average difference in points for the 2018 season.

```

269 GROUP BY match.home_id, team.full_name
270 ORDER BY Average_To_Favor_Points DESC
271 LIMIT (1)
272
273 SELECT team.full_name AS team, AVG(CAST(home_pts AS INT) - CAST(visitor_pts AS INT)) AS Average_To_Favor_Points
274 FROM match
275 INNER JOIN team
276 ON team.team_id = match.home_id
277 WHERE match.season = '2018'
278 GROUP BY match.home_id, team.full_name
279 ORDER BY Average_To_Favor_Points DESC
280 LIMIT (1)
  
```

team	average_to_favor_points
Milwaukee Bucks	12.073170731707316

Total rows: 1 of 1 | Query complete 00:00:00.121 | Ln 273, Col 1 | Successfully run. Total query runtime: 121 msec. 1 rows affected.

Para la temporada 2018 el equipo con la mayor diferencia de puntos a favor en promedio por partido jugado fue, Milwaukee Bucks, con 12.07 puntos a favor por partido jugado.

10. ¿Quién es el jugador más valioso del draft del 2018 hoy en día?

```

pgAdmin 4
File Object Tools Help
Server proyecto01/postgres@PostgreSQL 14*
Properties SQL Statistics Dependencies
Query History
291 inner join team
292 on team.team_id = player.team_id
293 where draft_year = '2018' and player.statusplayer = 'Active'
294 group by team.full_name, player.full_name, player pts, player.ast, player.reb, player.allstarappearances
295 order by avg(player.pts+player.ast+player.reb+player.allstarappearances)
296
297 select team.full_name as team, player.full_name as player, ((player.pts+player.ast+player.reb+player.allstarappearances)/4) as NBA_Value
298 from player
299 inner join team
300 on team.team_id = player.team_id
301 where draft_year = '2018' and ((player.pts+player.ast+player.reb+player.allstarappearances)/4) is not null
302 order by NBA_Value desc
303 Limit (1)
Data output Messages Notifications
team character varying (30) player character varying (60) nba_value double precision
1 Los Angeles Lakers Jemerrio Jones 3.7249999999999996
Total rows: 1 of 1 Query complete 00:00:00.115 Ln 297, Col 1

```

Por ende, el jugador más valioso de la NBA para el año 2018 fue Jemerrio Jones con un valor de nba de 3.72 perteneciente a Los Angeles Lakers.

11. Calcule el top 5 de los estados que más salarios pagaron durante las temporadas 2020-2021 y 2021-2022

```

pgAdmin 4
File Object Tools Help
Browser proyecto01/postgres@PostgreSQL 14*
Properties SQL Statistics Dependencies
Query History
317 select t.state as state, avg(s.value) as Average_Salary_In_Seasons_202021_202122
318 from player as p
319 inner join salary as s
320 on p.player_id = s.player_id
321 inner join team as t
322 on p.team_id = t.team_id
323 where s.season = '2020-21' or s.season = '2021-22'
324 group by t.state
325 order by Average_Salary_In_Seasons_202021_202122 desc
326 Limit(5)
327
328
329
Data output Messages Notifications
state character varying (30) average_salary_in_seasons_202021_202122 double precision
1 Wisconsin 13074887.47368421
2 Oregon 12325360.05263158
3 Colorado 12168411.578947369
4 New York 11945270.42857143
5 Utah 11920169.090909092
Total rows: 5 of 5 Query complete 00:00:00.148 Ln 318, Col 1

```

"Wisconsin"	13074887.47368421
"Oregon"	12325360.05263158
"Colorado"	12168411.578947369
"New York"	11945270.42857143
"Utah"	11920169.090909092

Queries:

```
drop table Player cascade  
drop table official cascade  
drop table salary cascade  
drop table team cascade  
drop table match cascade
```

```
select a.Player_ID, a.Full_Name, a.Height  
from Player as a  
where a.StatusPlayer = 'Active'  
order by a.Height desc  
limit (1)
```

```
select * from match
```

```
select m.home_id, cast(m.home_pts as INT)  
from match as m  
order by m.home_id
```

```
select sum(cast(m.visitor_pts as INT)) as suma, m.visitor_id from match as m  
group by m.visitor_id  
order by m.visitor_id
```

```
select sum(cast(m.home_pts as INT)) as suma, m.home_id, l.suma, l.visitor_id  
from match as m, (  
    select sum(cast(m.visitor_pts as INT)) as suma, m.visitor_id from match as m  
    group by m.visitor_id  
    order by m.visitor_id  
)as l  
group by m.home_id, l.suma, l.visitor_id
```

```
order by m.home_id, l.visitor_id
```

```
select sum(coalesce(m.suma2)+coalesce(m.suma1)) as , m.id
from match, (
    select sum(cast(m.home_pts as INT)) as suma1, m.home_id as id, l.suma as suma2,
l.visitor_id
    from match as m,
        select sum(cast(m.visitor_pts as INT)) as suma, m.visitor_id from match as m
        group by m.visitor_id
        order by m.visitor_id
    )as l
    group by m.home_id, l.suma, l.visitor_id
    order by m.home_id, l.visitor_id
) as m
group by m.id
order by m.id asc
```

```
select avg(coalesce(m.suma2)+coalesce(m.suma1)) as promedio, m.id
```

```
from match, (
```

```
    select avg(cast(m.home_pts as INT)) as suma1, m.home_id as id, l.suma as suma2,
l.visitor_id
    from match as m,
        select avg(cast(m.visitor_pts as INT)) as suma, m.visitor_id from match as m
        group by m.visitor_id
        order by m.visitor_id
    )as l
    group by m.home_id, l.suma, l.visitor_id
    order by m.home_id, l.visitor_id
```

```
) as m  
group by m.id  
order by promedio asc
```

```
select count(visitor_WL), official_ID_1  
from match  
where visitor_WL='L'  
group by official_ID_1  
order by official_ID_1
```

```
select count(visitor_WL), official_ID_2  
from match  
where visitor_WL='L'  
group by official_ID_2  
order by official_ID_2
```

```
select count(visitor_W1L), official_ID_2  
from match  
where visitor_WL='L'  
group by official_ID_3  
order by official_ID_3
```

```
select count(l.visitor_WL) as Loses_Counter, l.official_ID_3 as Official_ID  
from official, (  
    select official_ID_3, visitor_WL from match where visitor_WL='L' and season>=2015  
    group by official_ID_3, visitor_WL  
union all  
    select official_ID_2, visitor_WL from match where visitor_WL='L' and season>=2015 group  
    by official_ID_2, visitor_WL  
union all
```

```
select official_ID_1, visitor_WL from match where visitor_WL='L' and season>=2015 group by official_ID_1, visitor_WL
```

```
) as l
```

```
group by l.official_ID_3
```

```
order by Loses_Counter desc
```

```
Limit 5
```

```
select p.Team_ID
```

```
from Salary as s, player as p
```

```
where p.player_id = s.player_id
```

```
select sum(s.value), s.player_id
```

```
from salary as s
```

```
where s.season ='2020-21'
```

```
group by s.player_id
```

```
select t.full_name as Team, sum(s.value) as Total_Salaries
```

```
from salary as s
```

```
inner join player as p
```

```
on s.player_id = p.player_id
```

```
inner join team as t
```

```
on t.team_id = p.team_id
```

```
where s.season ='2020-21'
```

```
group by p.team_id, t.full_name
```

```
order by sum(s.value) desc
```

```
Limit(1)
```

```
select t.full_name as Team, avg(s.value) as Total_Salaries
```

```
from salary as s
```

```
inner join player as p
```

```
on s.player_id = p.player_id
```

```
inner join team as t  
on t.team_id = p.team_id  
where s.season ='2020-21'  
group by p.team_id, t.full_name  
order by avg(s.value) desc  
Limit(1)
```

```
select o.full_name, count(m.home_wl)  
from official as o  
inner join match as m  
on m.official_id_1 = o.official_id  
where m.home_wl = 'W'and m.season >=2014  
group by m.official_id_1, o.full_name
```

```
select official_ID_3, visitor_WL from match where visitor_WL='L' and season>=2015  
union all  
select official_ID_2, visitor_WL from match where visitor_WL='L' and season>=2015  
union all  
select official_ID_1, visitor_WL from match where visitor_WL='L' and season>=2015
```

```
select o.full_name as Official_Name, count(m.wl) as  
Number_Of_Loses_Of_Visitor_With_This_Official  
from official as o  
inner join (  
select official_ID_3 as id, visitor_WL as wl from match where visitor_WL='L' and  
season>=2015  
union all  
select official_ID_2, visitor_WL from match where visitor_WL='L' and season>=2015  
union all  
select official_ID_1, visitor_WL from match where visitor_WL='L' and season>=2015 ) as m  
on m.id = o.official_id
```

```
group by m.id, o.full_name
```

```
order by count(m.wl) desc
```

```
Limit (5)
```

```
select t.full_name as Team, avg(cast(m.pts as INT)) as Promedio_Puntos_2014
```

```
from team as t
```

```
inner join (
```

```
    select home_pts as pts, home_id as id from match where season =2014
```

```
union all
```

```
    select visitor_pts as pts, visitor_id as id from match where season =2014
```

```
) as m
```

```
on m.id = t.team_id
```

```
group by m.id, t.full_name
```

```
order by avg(cast(m.pts as INT)) desc
```

```
select t.full_name as Team, avg(cast(m.pts as INT)) as Promedio_Puntos_2015
```

```
from team as t
```

```
inner join (
```

```
    select home_pts as pts, home_id as id from match where season =2015
```

```
union all
```

```
    select visitor_pts as pts, visitor_id as id from match where season =2015
```

```
) as m
```

```
on m.id = t.team_id
```

```
group by m.id, t.full_name
```

```
order by avg(cast(m.pts as INT)) desc
```

```
select t.full_name as Team, avg(cast(m.pts as INT)) as Promedio_Puntos_2016
```

```
from team as t
```

```
inner join (
```

```
    select home_pts as pts, home_id as id from match where season =2016
```

```
union all
select visitor_pts as pts, visitor_id as id from match where season =2016
) as m
on m.id = t.team_id
group by m.id, t.full_name
order by avg(cast(m pts as INT)) desc

select t.full_name as Team, avg(cast(m pts as INT)) as Promedio_Puntos_2017
from team as t
inner join (
    select home_pts as pts, home_id as id from match where season =2017
union all
select visitor_pts as pts, visitor_id as id from match where season =2017
) as m
on m.id = t.team_id
group by m.id, t.full_name
order by avg(cast(m pts as INT)) desc

select t.full_name as Team, avg(cast(m pts as INT)) as Promedio_Puntos_2018
from team as t
inner join (
    select home_pts as pts, home_id as id from match where season =2018
union all
select visitor_pts as pts, visitor_id as id from match where season =2018
) as m
on m.id = t.team_id
group by m.id, t.full_name
order by avg(cast(m pts as INT)) desc

select t.full_name as Team, avg(cast(m pts as INT)) as Promedio_Puntos_2019
```

```
from team as t
inner join (
    select home_pts as pts, home_id as id from match where season =2019
union all
    select visitor_pts as pts, visitor_id as id from match where season =2019
) as m
on m.id = t.team_id
group by m.id, t.full_name
order by avg(cast(m pts as INT)) desc
```

```
select t.full_name as Team, avg(cast(m pts as INT)) as Promedio_Puntos_2020
```

```
from team as t
```

```
inner join (

```

```
    select home_pts as pts, home_id as id from match where season =2020
union all
    select visitor_pts as pts, visitor_id as id from match where season =2020
) as m
on m.id = t.team_id
group by m.id, t.full_name
order by avg(cast(m pts as INT)) desc
```

```
select t.full_name as Team, avg(cast(m pts as INT)) as Promedio_Puntos_2021
```

```
from team as t
```

```
inner join (

```

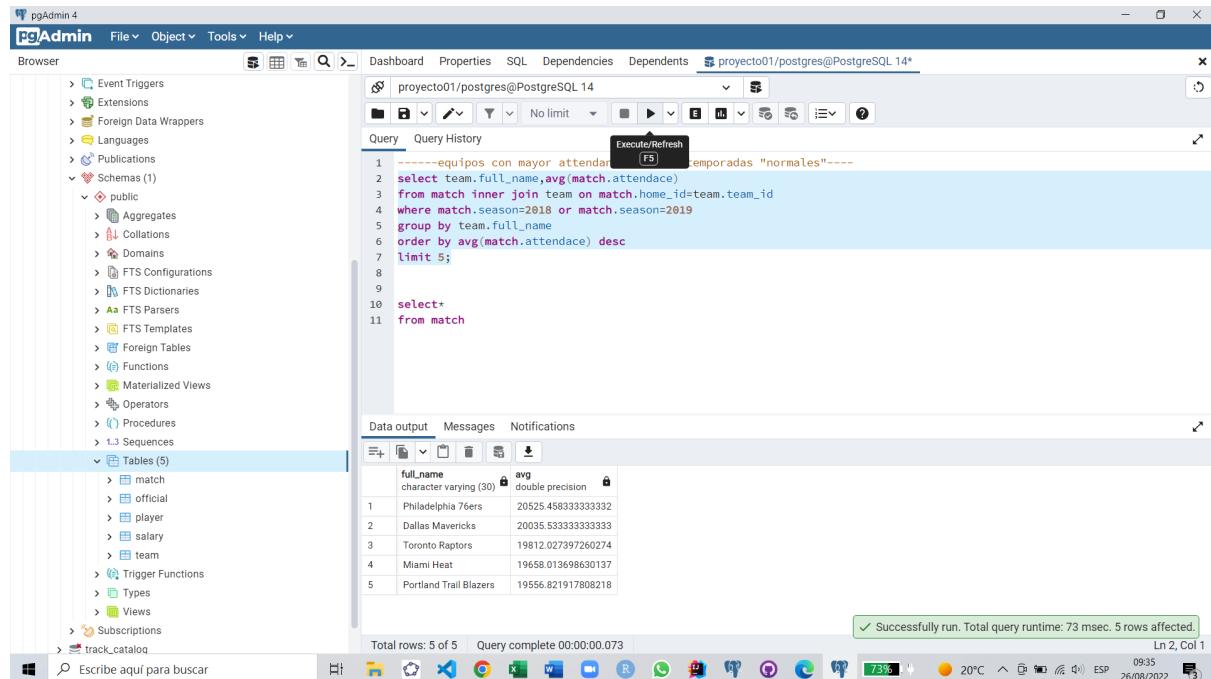
```
    select home_pts as pts, home_id as id from match where season =2021
```

Parte 3

1. ¿Cuáles son los 5 equipos con más asistencia de fanáticos en la temporada 2018 y 2019?

Se planteó esta pregunta ya que para que un equipo represente una ganancia hacia el inversor este tiene que tener una asistencia de fanáticos bastante fuerte para que la mercadería y negocios en el interior de los estadios tengan una ganancia. Asimismo la pregunta fue dirigida a esas dos temporadas ya que fueron las últimas temporadas con cierta “normalidad” ya que la de 2020 fue interrumpida por

la pandemia mundial. De acuerdo con (Young,2021) la NBA proyecta en total alrededor de 10 billones en ganancias para las próximas temporadas debido a que la asistencia de los fanáticos sigue aumentando. No vamos a tomar en cuenta cuantos juegos transmitidos en televisión tiene cada equipo ya que es un área que ha estado inestable debido a las nuevas formas de transmitir los juegos.



```

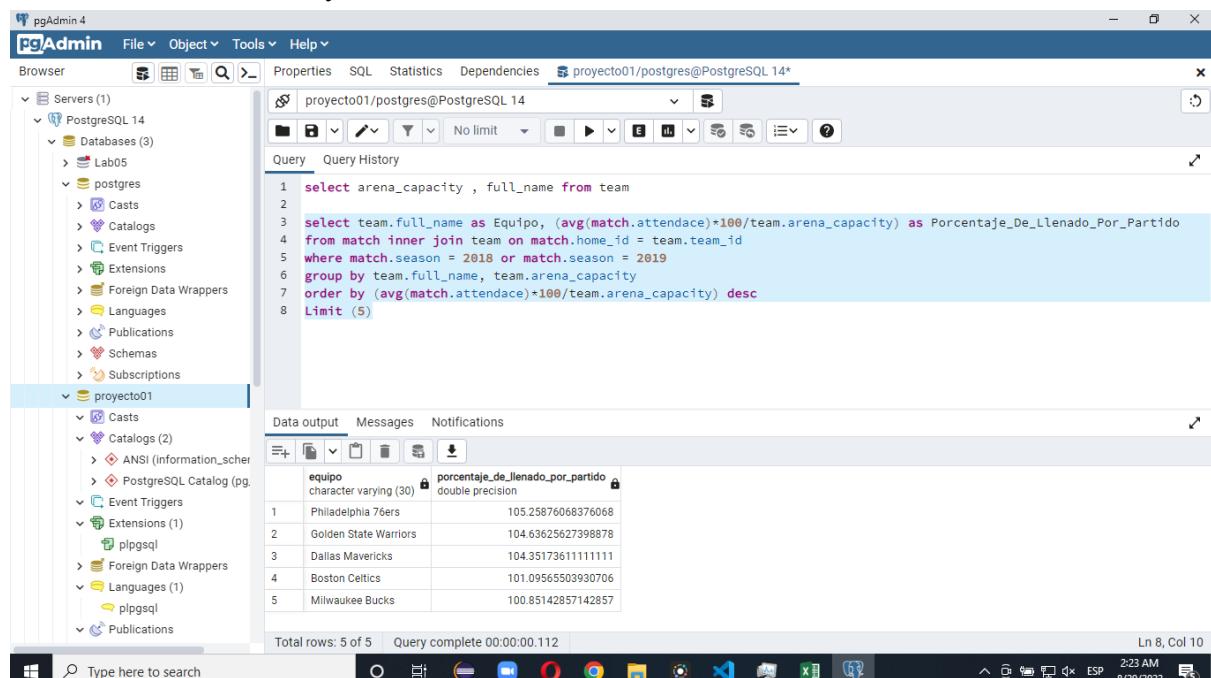
pgAdmin 4
File Object Tools Help
Browser
Schemas (1)
Tables (5)
match
official
player
salary
team
Trigger Functions
Types
Views
Subscriptions
track_catalog
Escribe aquí para buscar
Query Query History
1 -----equipos con mayor asistencia en temporadas "normales"-----
2 select team.full_name,avg(match.attendance)
3 from match inner join team on match.home_id=team.team_id
4 where match.season=2018 or match.season=2019
5 group by team.full_name
6 order by avg(match.attendance) desc
7 limit 5;
8
9
10 select*
11 from match

```

full_name	avg
Philadelphia 76ers	20525.458333333332
Dallas Mavericks	20035.533333333333
Toronto Raptors	19812.027397260274
Miami Heat	19658.013698630137
Portland Trail Blazers	19556.821917808218

Total rows: 5 of 5 Query complete 00:00:00.073 Successfully run. Total query runtime: 73 msec. 5 rows affected. Ln 2, Col 1

Los 5 equipos con mayor asistencia de fanáticos son los Philadelphia 76ers, Dallas Mavericks, Toronto Raptors, Miami Heat y Portland TrailBlazer. Así mismo, los equipos que suelen llenar de media la mayor parte de su estadio en promedio son: Philadelphia 76ers, Golden State Warriors, Dallas Mavericks, Boston Celtics y Milwaukee Bucks.



```

pgAdmin 4
File Object Tools Help
Browser
Servers (1)
PostgreSQL 14
Databases (3)
Lab05
postgres
Casts
Catalogs
Event Triggers
Extensions
Foreign Data Wrappers
Languages
Publications
Schemas
Subscriptions
projeto01
Casts
Catalogs (2)
ANSI (information_schema)
PostgreSQL Catalog (pg)
Event Triggers
Extensions (1)
plpgsql
Foreign Data Wrappers
Languages (1)
plpgsql
Publications
Escribe aquí para buscar
Query Query History
1 select arena_capacity , full_name from team
2
3 select team.full_name as Equipo, (avg(match.attendance)*100/team.arena_capacity) as Porcentaje_De_Llenado_Por_Partido
4 from match inner join team on match.home_id = team.team_id
5 where match.season = 2018 or match.season = 2019
6 group by team.full_name, team.arena_capacity
7 order by (avg(match.attendance)*100/team.arena_capacity) desc
8 Limit (5)

Data output Messages Notifications
equipo porcentaje_de_llenado_por_partido
1 Philadelphia 76ers 105.25876068376068
2 Golden State Warriors 104.63625627398878
3 Dallas Mavericks 104.35173611111111
4 Boston Celtics 101.09565503930706
5 Milwaukee Bucks 100.85142857142857

Total rows: 5 of 5    Query complete 00:00:00.112    Ln 8, Col 10

```

2. ¿Cuál es la cantidad de juegos ganados en casa y de visita por Philadelphia 76ers, Dallas Mavericks, Toronto Raptors, Miami Heat y Portland TrailBlazers?

La cantidad de juegos ganados en total en casa desde la temporada 2014 hasta la temporada 2020 son

```

projecto01/postgres@PostgreSQL 14
No limit
Order by avg(match.attendance) desc
Limit 5;
----Cantidad de juegos ganados por los equipos mencionados anteriormente desde la temporada 2014 en casa-----
select full_name,count(match.home_wl)
from match inner join team on match.home_id=team.team_id
where team.full_name in (
    select full_name
    from match inner join team on match.home_id=team.team_id
    where match.season=2018 or match.season=2019
    group by full_name
    order by avg(match.attendance) desc
    limit 5
)
and home_wl='W'
group by team.full_name
order by count(match.home_wl) desc
Limit 5;

```

full_name	count
Toronto Raptors	181
Portland Trail Blazers	168
Miami Heat	160
Philadelphia 76ers	157
Dallas Mavericks	137

Total rows: 5 of 5 Query complete 00:00:00.050 Successfully run. Total query runtime: 50 msec. 5 rows affected. Ln 9, Col 1

Los Toronto Raptors son los que más han ganado desde casa luego le sigue Portland Trail Blazers, Miami Heat, Philadelphia 76ers y por último los Dallas Mavericks.
Y la cantidad de juegos ganados de visita desde el 2014 por los equipos con mayor asistencia de fanáticos son:

```

projecto01/postgres@PostgreSQL 14
No limit
Over by count(match.visitor_wl) desc
----Juegos ganados desde 2014 de visita-----
select full_name,count(match.visitor_wl)
from match inner join team on match.visitor_id=team.team_id
where team.full_name in (
    select full_name
    from match inner join team on match.home_id=team.team_id
    where match.season=2018 or match.season=2019
    group by full_name
    order by avg(match.attendance) desc
    limit 5
)
and visitor_wl='W'
group by team.full_name
order by count(match.visitor_wl) desc
Limit 5;

```

full_name	count
Toronto Raptors	147
Miami Heat	119
Portland Trail Blazers	117
Dallas Mavericks	105
Philadelphia 76ers	90

Total rows: 5 of 5 Query complete 00:00:00.147 Successfully run. Total query runtime: 147 msec. 5 rows affected. Ln 23, Col 1

Los Toronto Raptors son los que más juegos han ganado desde 2014 de visita, seguidos por el Miami Heat, los Portland TrailBlazers, los Dallas Mavericks y de último los Philadelphia 76ers.

Como es evidente los equipos más consistentes de juegos ganados independientemente de si se encuentran de visita o de locales son los Toronto Raptors, Miami Heat y los Portland Trail Blazers.

3. ¿Que equipo tiene la mayor cantidad de jugadores All-Stars históricamente entre los Philadelphia 76ers, Dallas Mavericks, Toronto Raptors, Miami Heat y Portland Trail Blazers?

Se busca esto ya que según (Arlauckas,2022) existe una mayor tendencia de que televisen los juegos a partir de la cantidad de AllStars que los equipos han tenido históricamente, debido el que tengan o hayan tenido estrellas hace el equipo que tenga un nivel más alto de popularidad.

```

PgAdmin 4
PgAdmin File Object Tools Help
Browser Dashboard Properties SQL Dependencies Dependents QueriesProyectos.sql* lab040202.sql lab04_01.sql
Query History
36 ----> Que equipos históricamente han tenido la mayor cantidad de jugadores seleccionados en AllStar--
37 select team.full_name, count(player.allstarappearances) as AparicionesAS
38 from player inner join team on player.team_id=team.team_id
39 where team.full_name in(
40     select full_name
41         from match inner join team on match.home_id=team.team_id
42         where match.season=2018 or match.season=2019
43         group by full_name
44         order by avg(match.attendance) desc
45         limit 5
46 )
47 group by team.full_name
48 having count(player.allstarappearances)>0
49
50
51
Data output Messages Notifications
full_name          aparicionesAS
character varying(30)    bigint
1 Dallas Mavericks           74
2 Miami Heat                 49
3 Philadelphia 76ers          151
4 Portland Trail Blazers      100
5 Toronto Raptors             34
Total rows: 5 of 5   Query complete 00:00:00.141

```

Successfully run. Total query runtime: 141 msec. 5 rows affected.
Ln 37, Col 1
11:46 24°C 66% 26/08/2022

Historicamente los Philadelphia 76ers han tenido la mayor cantidad de All Stars seguidos por los Portland Trailblazers, luego los Dallas Mavericks, seguido por el Miami Heat y por último los Toronto Raptors.

4. ¿Los 5 máximos anotadores de los Philadelphia 76ers, Dallas Mavericks, Toronto Raptors, Miami Heat y Portland TrailBlazers ?

De acuerdo con Scaletta(2010) cuando la distribución de anotación es lo más cercano a uniforme los equipos tienden a ganar más. Por lo que observaremos si hay mucha diferencia entre el máximo anotador y el que le sigue para determinar si el equipo solo depende de un jugador o tiene varios que aportan para ganar los partidos.

Philadelphia 76ers

pgAdmin 4

File Object Tools Help

Browser

```

46 )
47 group by team.full_name
48 having count(player.allstarappearances)>0
49
50 ----top 5 anotadores de los equipos mas atendidos en la temporada 2018 y 2019----
51 ---Philadelphia 76ers---
52 select player.full_name,player pts
53 from player inner join team on player.team_id=team.team_id
54 where team.full_name='Philadelphia 76ers' and player.statusplayer='Active'
55 group by player.full_name, player.pts
56 order by player.pts desc
57 limit 5
  
```

Data output Messages Notifications

full_name	pts
character varying (60)	double precision
1 Joel Embiid	29.9
2 Tobias Harris	20.8
3 Wayne Simen	16.1
4 Shake Milton	13.8
5 Seth Curry	12.9

Total rows: 5 of 5 Query complete 00:00:00.120 ✓ Successfully run. Total query runtime: 120 msec. 5 rows affected. Ln 52, Col 1

Escribe aquí para buscar

Como se puede observar entre el mejor y el segundo mejor solamente existen 9 puntos de diferencia.
Dallas Mavericks

pgAdmin 4

File Object Tools Help

Browser

```

55 group by player.full_name, player.pts
56 order by player.pts desc
57 limit 5
58 ---Dallas Mavericks---
59 select player.full_name,player pts
60 from player inner join team on player.team_id=team.team_id
61 where team.full_name='Dallas Mavericks' and player.statusplayer='Active'
62 group by player.full_name, player.pts
63 order by player.pts desc
64 limit 5
65 ---Toronto Raptors---
66 select player.full_name,player pts
  
```

Data output Messages Notifications

full_name	pts
character varying (60)	double precision
1 Luka Doncic	29
2 Bob Portman	19.7
3 Tim Hardaway Jr.	16.4
4 Jeremy Richardson	13.8
5 Jalen Brunson	12.2

Total rows: 5 of 5 Query complete 00:00:00.121 ✓ Successfully run. Total query runtime: 121 msec. 5 rows affected. Ln 59, Col 1

Escribe aquí para buscar

Como se puede observar entre el mejor y el segundo mejor existen únicamente 9.3 puntos de diferencia

Toronto Raptors

Únicamente existe una diferencia de 0.3 puntos entre el máximo anotador y el segundo máximo

```

64 limit 5
65      ---Toronto Raptors---
66      select player.full_name,player pts
67      from player inner join team on player.team_id=team.team_id
68      where team.full_name='Toronto Raptors' and player.statusplayer='Active'
69      group by player.full_name, player pts
70      order by player pts desc
71 limit 5
72      ---Miami Heat---
73      select player.full_name,player pts
74      from player inner join team on player.team_id=team.team_id
75      where team.full_name='Miami Heat' and player.statusplayer='Active'
    
```

	full_name	pts
1	Norm Van Lier	20.1
2	Alexey Shved	19.8
3	Kasib Powell	19.5
4	Kyle Lowry	17.6
5	OG Anunoby	14.0

Miami Heat

Existe solamente una diferencia de 1 punto entre el máximo anotador y el segundo máximo anotador.

```

69 group by player.full_name,player pts
70 order by player pts desc
71 limit 5
72      ---Miami Heat---
73      select player.full_name,player pts
74      from player inner join team on player.team_id=team.team_id
75      where team.full_name='Miami Heat' and player.statusplayer='Active'
76      group by player.full_name, player pts
77      order by player pts desc
78 limit 5
79      ---Portland TrailBlazers
80      select player.full_name,player pts
    
```

	full_name	pts
1	Jimmy Butler	21.3
2	Bam Adebayo	19.2
3	Tyler Herro	15.1
4	Goran Dragic	13.9
5	Frank Ntilikina	13.3

Portland TrailBlazers

The screenshot shows the pgAdmin 4 interface with a query editor containing SQL code to find the top 5 players by points for Miami Heat and Portland Trail Blazers. The results table shows the full name and points for each player.

```

75 where team.full_name='Miami Heat' and player.statusplayer='Active'
76 group by player.full_name, player pts
77 order by player pts desc
78 limit 5
79 ---Portland TrailBlazers
80 select player.full_name, player pts
81 from player inner join team on player.team_id=team.team_id
82 where team.full_name='Portland Trail Blazers' and player.statusplayer='Active'
83 group by player.full_name, player pts
84 order by player pts desc
85 limit 5
86

```

full_name	pts
Damian Lillard	30.1
CJ McCollum	23.8
Gary Trent	15
Carmelo Anthony	14
Enes Kanter	12.1

Hay una diferencia de 6.4 puntos entre el máximo anotador y el segundo máximo anotador

5. ¿Cuál es la proyección de salarios para la temporada 2022-2023 por parte de Philadelphia 76ers, Dallas Mavericks, Toronto Raptors, Miami Heat y Portland TrailBlazer?

The screenshot shows the pgAdmin 4 interface with a query editor containing SQL code to calculate the total projected salary for the top 5 teams based on attendance. The results table shows the team name and total projected salary.

```

88 -----Cantidad de dinero programada para los jugadores de los 5 equipos más atendidos por fanáticos-----
89
90 select t.full_name as Equipo, sum(s.value) as Salarios_Proyectados
91 from salary as s
92 inner join player as p
93 on s.player_id=p.player_id
94 inner join team as t
95 on t.team_id=p.team_id
96 where s.season='2022-23' and t.full_name in(
97     select full_name
98         from match inner join team on match.home_id=team.team_id
99         where match.season=2018 or match.season=2019
100        group by full_name
101        order by avg(match.attendance) desc
102        limit 5
103    )
104 group by p.team_id,t.full_name
105 order by sum(s.value) desc
106

```

equipo	salarios_proyectados
Philadelphia 76ers	122640590
Portland Trail Blazers	85755925
Toronto Raptors	78411082
Miami Heat	78183748
Dallas Mavericks	70637326

Como es evidente Philadelphia y Portland cuentan con la mayor cantidad de dinero comprometido. Según la página oficial de la NBA(2022) el máximo de dinero que tiene un equipo para pagarle a los jugadores es 123 millones de dólares, por lo que Philadelphia y Portland quedan definitivamente descartados debido a que no tienen espacio a que haya nuevas contrataciones para que los equipos sean mejores.

6. ¿Cuál es el tamaño promedio de Philadelphia 76ers, Dallas Mavericks, Toronto Raptors, Miami

Heat y Portland TrailBlazer?

De acuerdo con (Akabas,2022) los equipos tienen un mejor desempeño cuando los jugadores miden en promedio 6 pies con nueve pulgadas o 81 pulgadas por lo que con este query se va determinar que equipo se encuentra en ese parámetro o cercano a ese parámetro. El que el equipo tenga un buen tamaño implica

```

pgAdmin 4
PgAdmin File Object Tools Help
Browser Dashboard Properties SQL Dependencies Dependents QueriesProyectos.sql* lab040202.sql lab04_01.sql
Query History Execute/Refresh
101     order by avg(match.attendanc
102     limit 5
103 )
104 group by p.team_id,t.full_name
105 order by sum(s.value) desc
106 -----tamaño promedio de los equipos-----
107 select team.full_name as Equipo,avg(player.height) as Tamaño_Promedio
108 from team inner join player on team.team_id=player.team_id
109 where statusplayer='Active' and team.full_name in(
110     select full_name
111         from match inner join team on match.home_id=team.team_id
112         where match.season=2018 or match.season=2019
113         group by full_name
114         order by avg(match.attendance) desc
115         limit 5
116 )
117 group by team.full_name
118 order by avg(player.height) desc
119
Data output Messages Notifications
+-----+
| equipo | tamaño_promedio |
| character varying(30) | double precision |
+-----+
| 1 Dallas Mavericks | 79.76923076923077 |
| 2 Philadelphia 76ers | 79.42857142857143 |
| 3 Portland Trail Blazers | 78.76923076923077 |
| 4 Miami Heat | 78.25 |
| 5 Toronto Raptors | 77.71428571428571 |
+-----+
Total rows: 5 of 5 Query complete 00:00:00.075

```

The screenshot shows the pgAdmin 4 interface. The left sidebar displays the database schema with various objects like triggers, extensions, and tables. The main window has a query editor with a SQL script and a data output tab showing the results of the executed query. The results table has two columns: 'equipo' and 'tamaño_promedio'. The data shows five rows corresponding to NBA teams: Dallas Mavericks, Philadelphia 76ers, Portland Trail Blazers, Miami Heat, and Toronto Raptors, along with their average heights.

References

- Alauckas, D. (2022, Agosto 17). *NBA schedule 2022-23: Which teams have the most games on national TV? Warriors, Lakers, Celtics... AS USA*. Retrieved Agosto 26, 2022, from
<https://en.as.com/nba/nba-schedule-2022-23-which-teams-have-the-most-games-on-national-tv-warriors-lakers-celestics-n/>
- NBA. (2022, Junio 30). *NBA salary cap for 2022-23 season set at just over \$123 million*. NBA.com. Retrieved Agosto 28, 2022, from
<https://www.nba.com/news/nba-salary-cap-for-2022-23-season-set-at-just-over-123-million>
- Scaletta, K. (2010, Noviembre 9). *NBA What Makes a Champion? The Five Most Common Factors From the Past*. Bleacher Report. Retrieved Agosto 26, 2022, from
<https://bleacherreport.com/articles/512356-what-makes-a-champion-the-five-most-common-factors-from-the-past>
- Young, J. (2021, Octubre 18). *NBA 2021-2022 season: \$10 billion revenue, TV viewership rebound?* CNBC. Retrieved Agosto 28, 2022, from
<https://www.cnbc.com/2021/10/18/nba-2021-2022-season-10-billion-revenue-tv-viewership-rebound.html>