FOOTBALL ANALYSIS USING AGGREGATION AND OLAP QUERIES

Abhishek Singh (Id : 202118004)
DAIICT
202118004@daiict.ac.in

Devarsh Antani (Id : 202118044)
DAIICT
202118004@daiict.ac.in

INTRODUCTION:

OLAP stands for Online Analytical Processing Server. It is a software technology that allows users to analyze information from multiple database systems at the same time. It is based on a multidimensional data model and allows the user to query on multi-dimensional data. OLAP databases are divided into one or more cubes and these cubes are known as Hyper-cubes.

There are five basic analytical operations that can be performed on an OLAP cube:

- **Slice**: It selects a single dimension from the OLAP cube which results in a new sub-cube creation.
- **Dice**: It selects a sub-cube from the OLAP cube by selecting two or more dimensions.
- **Drill down**: In drill-down operation, the less detailed data is converted into highly detailed data. It can be done by:
    1. Moving down in the concept hierarchy
    2. Adding a new dimension
- **Roll up**: It is just opposite of the drill-down operation. It performs aggregation on the OLAP cube. It can be done by:
    1. Climbing up in the concept hierarchy
    2. Reducing the dimensions
- **Pivot**: It is also known as rotation operation as it rotates the current view to get a new view of the representation. In the sub-cube obtained after the slice operation, performing pivot operation gives a new view of it.

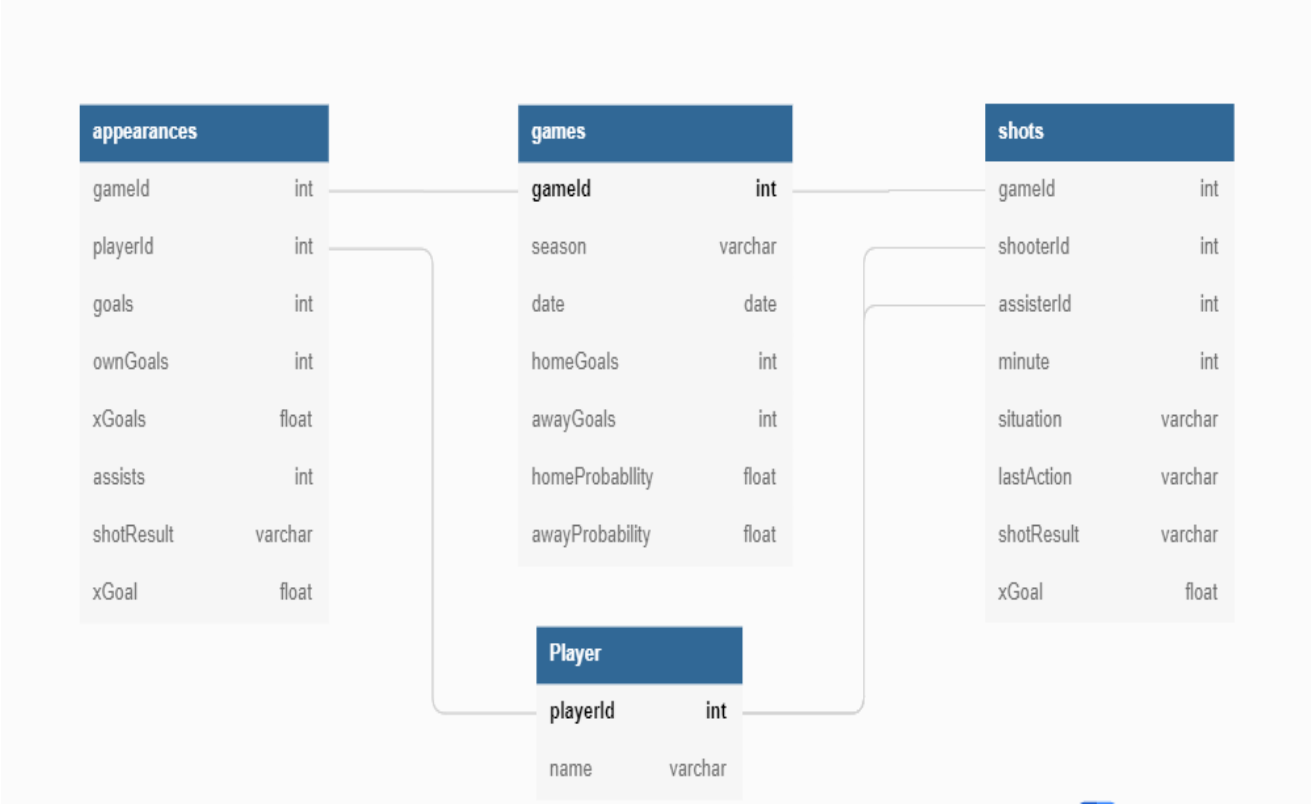
AIM OF THE PROJECT:

We are performing various aggregation and OLAP queries on football dataset taken from kaggle to draw helpful insights from the dataset.

DATASET:

This dataset contains football-related data covering the Top5 leagues in Europe from 2014-2020. It is structured like a relational database, which makes it easy to work with, regardless of the problem you want to solve. The dataset contains 4 tables that are appearances, games, shots, and player.

SCHEMA:



As you can see we have four tables here: appearances, games, shots, player. gameId which is a primary key of table games is a foreign key seen in table appearances, shots. PlayerId which is a primary key of table player, it is a foreign key seen in table appearances, shots.

QUERIES:

CUBE BUILDING

1) Get player id name goals position

```
1
2 sql_str = "SELECT a.playerId,p.name, SUM(goals)as goals, position FROM appearances a  " \
3   + " JOIN players p ON( a.playerId = p.playerId)"\
4   + "GROUP BY p.name, a.playerID,position WITH CUBE"
5 appearances_cube = spark.sql( sql_str )
6 appearances_cube.show()
7 appearances_cube.write.mode("overwrite").csv('appearances_cube3d.csv')
```

```
+--------+------------------+-----+--------+
|playerID|              name|goals|position|
+--------+------------------+-----+--------+
|     927|  Sebasti@n Coates|    0|      DC|
|     863|  Jonathan Walters|    6|     AMR|
|     601|       Arouna Kon@|    3|     AML|
|     815|    Alex Pritchard|    0|     Sub|
|     897|            Juanmi|   12|     Sub|
|     823|Vadis Odjidja-Ofoe|    0|     Sub|
|     939|      Mauro Z@rate|    0|      MR|
|    1764|     Ramadan Sobhi|    2|     Sub|
|    1954|  Samuel Souprayen|    0|      DL|
|    1888| Antonio Di Natale|   14|      FW|
|    1953|      Karim Laribi|    0|     Sub|
|    1279|  Roberto Vitiello|    0|     Sub|
|    1442|     Saphir Ta@der|    0|     DMC|
|    1815|              Alex|    0|     Sub|
|    1859|    Matteo Ciofani|    0|      DR|
|    1864|        Mirko Gori|    0|     Sub|
|    1480|   Gianluca Pegolo|    0|      GK|
|    1899|  Lorenzo Pasciuti|    0|      ML|
|    1899|  Lorenzo Pasciuti|    2|      MR|
|    1446|   Sergio Floccari|    0|     FWL|
+--------+------------------+-----+--------+
only showing top 20 rows
```

2) Get shooter id, situation, minute, shot result

```
1 sql_str = "SELECT shooterId, situation, minute , shotResult FROM shots group by shooterId, situation, shotResult,minute WITH CUBE ORDER BY shooterId"
2 shots_cube = spark.sql( sql_str )
3 shots_cube.show()
4 shots_cube.write.mode("overwrite").csv('shots_cube3d.csv')
```

```
+---------+--------------+------+----------+
|shooterId|     situation|minute| shotResult|
+---------+--------------+------+----------+
|     null|      OpenPlay|    82| ShotOnPost|
|     null|      SetPiece|    62|BlockedShot|
|     null|    FromCorner|    62|      Goal|
|     null|      SetPiece|    14|MissedShots|
|     null|      OpenPlay|    12|      null|
|     null|          null|  null|BlockedShot|
|     null|    FromCorner|     2|      Goal|
|     null|    FromCorner|     9| ShotOnPost|
|     null|DirectFreekick|    80| ShotOnPost|
|     null|       Penalty|    59|  SavedShot|
|     null|      OpenPlay|    75|      Goal|
|     null|      SetPiece|    68|MissedShots|
|     null|          null|    57|      null|
|     null|      OpenPlay|    15|      null|
|     null|      OpenPlay|     9|      null|
|     null|          null|     3|  SavedShot|
|     null|      OpenPlay|    76|MissedShots|
|     null|    FromCorner|    44|      null|
|     null|          null|    58|      Goal|
|     null|      SetPiece|    40|MissedShots|
+---------+--------------+------+----------+
only showing top 20 rows
```

3) get game_id, season, homeGoals, awayGoals from games table

```
[14]  1 sql_str = "SELECT gameId, season, SUM(homeGoals) as home_goals , SUM(awayGoals) as away_goals FROM games"\
      2 +" group by gameId, season  WITH CUBE ORDER BY gameId"
      3 games_cube = spark.sql( sql_str )
      4 games_cube.show()
      5 games_cube.write.mode("overwrite").csv('games_cube3d.csv')
```

```
+------+------+----------+----------+
|gameId|season|home_goals|away_goals|
+------+------+----------+----------+
|  null|  2016|      2967|      2206|
|  null|  2017|      2794|      2153|
|  null|  2014|      2736|      2062|
|  null|  2020|      2696|      2418|
|  null|  2018|      2819|      2200|
|  null|  2019|      2663|      2153|
|  null|  null|     19441|     15300|
|  null|  2015|      2766|      2108|
|    81|  null|         1|         0|
|    81|  2015|         1|         0|
|    82|  null|         0|         1|
|    82|  2015|         0|         1|
|    83|  null|         2|         2|
|    83|  2015|         2|         2|
|    84|  null|         4|         2|
|    84|  2015|         4|         2|
|    85|  2015|         1|         3|
|    85|  null|         1|         3|
|    86|  null|         2|         2|
|    86|  2015|         2|         2|
+------+------+----------+----------+
only showing top 20 rows
```

SLICING

4) Total goals and assists of FW players

```
[16]  1 sql_str = "SELECT * from appearances_cube where position = 'FW' "
      2 spark.sql( sql_str ).show()
```

```
+--------+------------------+-----+--------+
|playerID|              name|goals|position|
+--------+------------------+-----+--------+
|    1888| Antonio Di Natale|   14|      FW|
|    1473|   Matteo Politano|    8|      FW|
|    6036|Rajiv van La Parra|    0|      FW|
|    5169| Youssef En-Nesyri|   32|      FW|
|    1577|       Marko Pjaca|    2|      FW|
|     594|              null|  123|      FW|
|     940|              null|    7|      FW|
|    null|       Libor Koz�k|    0|      FW|
|    null|         Marco Sau|   12|      FW|
|    null|       Timo Werner|   74|      FW|
|     276|              null|    4|      FW|
|    null|      Sergi Enrich|   28|      FW|
|    null|         Casades�s|    8|      FW|
|    3683|              null|    1|      FW|
|    null|   Julian Schieber|    6|      FW|
|    null|     Rub�n Sobrino|    6|      FW|
|    null| Prejuce Nakoulma|    8|      FW|
|    null|    George Puscas|    1|      FW|
|    4366|              null|    0|      FW|
|    4718|              null|    0|      FW|
+--------+------------------+-----+--------+
only showing top 20 rows
```

5) Highest goal ratio playerid and his goals in season 2020.

```
[47]  1
      2 sql_str = "SELECT shooterId, p.name, SUM(a.goals), MAX(xGoal) from shots"\
      3  +" s JOIN players p ON(s.shooterId = p.playerId) JOIN games g ON(s.gameId = g.gameId)"\
      4  +" JOIN appearances a ON(s.shooterId = a.playerId)"\
      5  +" where g.season = '2020' GROUP BY s.shooterId, p.name"\
      6  +" ORDER BY s.shooterId"
      7
      8 spark.sql( sql_str ).show()
```

```
+---------+--------------------+----------+------------------+
|shooterId|                name|sum(goals)|        max(xGoal)|
+---------+--------------------+----------+------------------+
|        3|      Luca Caldirola|        21| 0.130910724401474|
|       22|         John Brooks|       180|  0.51889580488205|
|       23| Marvin Plattenhardt|        18|0.0501803532242775|
|       27|     Vladimir Darida|       459| 0.474260807037354|
|       28|     Mitchell Weiser|        18|0.0583156235516071|
|       29|      Vedad Ibisevic|        90| 0.646249175071716|
|       38|    Daniel Brosinski|       100|  0.75777667760849|
|       40|         Stefan Bell|        88|0.0900668501853943|
|       42|Julian Baumgartli...|        49| 0.689103305339813|
|       43|         Danny Latza|       132| 0.605999708175659|
|       47|      Yoshinori Muto|       484| 0.675653636455536|
|       48|        Jhon C�rdoba|      1632| 0.583697021007538|
|       52|      Alexander Hack|        21|0.0974528416991234|
|       60|   Christian Gentner|       105|0.0539998114109039|
|       64|        Filip Kostic|      1612| 0.572721719741821|
|       65|         Timo Werner|      7533| 0.918899639129639|
|       69|       Carlos Gruezo|        16|0.0699499845504761|
|       74|     Pavel Kader�bek|        70| 0.406523674726486|
|       75|         Niklas S�le|       160| 0.104078404605389|
|       76|       Fabian Sch�r|       120| 0.761168837547302|
+---------+--------------------+----------+------------------+
only showing top 20 rows
```

6) Gameid, awayGoals, homeProbability, awayProbability where a home team lost by 4 or more goals.

```
[51]  1 sql_str = "SELECT Gameid,awayGoals, homeProbability, awayProbability from games"\
      2 " where awayGoals >=4 "
      3 spark.sql( sql_str ).show()
```

```
+------+---------+---------------+---------------+
|Gameid|awayGoals|homeProbability|awayProbability|
+------+---------+---------------+---------------+
|   106|        4|         0.1041|         0.7202|
|   144|        5|         0.0518|         0.8594|
|   178|        5|         0.0068|          0.961|
|   208|        4|         0.0916|         0.7435|
|   265|        4|         0.5398|         0.2041|
|   286|        4|         0.0471|         0.7914|
|   301|        5|         0.2233|         0.4551|
|   339|        6|         0.0134|         0.8843|
|   384|        4|         0.0442|         0.8396|
|   386|        4|         0.4982|         0.2203|
|   414|        4|         0.0203|         0.9004|
|   421|        4|         0.0697|         0.8242|
|   422|        4|         0.0702|         0.7652|
|   439|        4|         0.3543|         0.3887|
|   469|        4|         0.2393|         0.4748|
|   472|        4|         0.1427|         0.6019|
|   493|        4|         0.0711|         0.7978|
|   497|        4|         0.2437|         0.5076|
|   502|        4|         0.0142|         0.9209|
|   608|        4|         0.0044|         0.9415|
+------+---------+---------------+---------------+
only showing top 20 rows
```

DICING

7) Get the player name, situation and last action when shot result is goal and minute is 90

```
[29] 1 sql_str = "SELECT s.shooterId, p.name, situation, lastAction  FROM shots s JOIN players p ON (s.shooterId = p.playerId)"\
     2 + "WHERE shotResult='Goal' and minute = 90 " \
     3  + "ORDER BY s.shooterId"
     4 spark.sql( sql_str ).show()
```

```
+---------+--------------------+-------------+----------+
|shooterId|                name|    situation|lastAction|
+---------+--------------------+-------------+----------+
|       26|      Salomon Kalou|      Penalty|  Standard|
|       26|      Salomon Kalou|   FromCorner|  HeadPass|
|       26|      Salomon Kalou|      Penalty|  Standard|
|       29|     Vedad Ibisevic|     OpenPlay|      Pass|
|       47|     Yoshinori Muto|   FromCorner|     Cross|
|       48|       Jhon C@rdoba|     OpenPlay|    Aerial|
|       58|  Georg Niedermeier|   FromCorner|      Pass|
|       62|         Lukas Rupp|   FromCorner|      None|
|       65|        Timo Werner|     OpenPlay|      Pass|
|       81|    Jonathan Schmid|     OpenPlay|  HeadPass|
|       83|      Kevin Volland|     OpenPlay|   Rebound|
|       88|           Mark Uth|     OpenPlay|      Pass|
|      104|      Mevl@t Erdinc|     OpenPlay|   Rebound|
|      110|Theodor Gebre Sel...|   FromCorner|     Cross|
|      131|       Jonas Hector|   FromCorner|     Cross|
|      134|       Marcel Risse|DirectFreekick|  Standard|
|      153|  Maximilian Arnold|     OpenPlay|      None|
|      169|       Lewis Holtby|     OpenPlay|Throughball|
|      169|       Lewis Holtby|     OpenPlay|      None|
|      179|     Zolt@n Stieber|     OpenPlay|  HeadPass|
+---------+--------------------+-------------+----------+
only showing top 20 rows
```

8) Getting the values of shot result at minute 60 from 2019 season

```
[18] 1 sql_str = "SELECT g.gameID, season, s.minute, s.shotResult from games_cube AS g JOIN shots AS s ON(g.gameId = s.gameID)"\
     2  +"where g.season = '2019' AND s.minute = 60 "
     3 spark.sql( sql_str ).show()
```

```
+------+------+------+-----------+
|gameID|season|minute| shotResult|
+------+------+------+-----------+
| 11645|  2019|    60|MissedShots|
| 11653|  2019|    60|MissedShots|
| 11655|  2019|    60|       Goal|
| 11657|  2019|    60|  SavedShot|
| 11657|  2019|    60|BlockedShot|
| 11659|  2019|    60|  SavedShot|
| 11659|  2019|    60| ShotOnPost|
| 11661|  2019|    60|MissedShots|
| 11662|  2019|    60|BlockedShot|
| 11664|  2019|    60|  SavedShot|
| 11666|  2019|    60|BlockedShot|
| 11670|  2019|    60|BlockedShot|
| 11672|  2019|    60|  SavedShot|
| 11677|  2019|    60|BlockedShot|
| 11680|  2019|    60|  SavedShot|
| 11681|  2019|    60|MissedShots|
| 11683|  2019|    60|  SavedShot|
| 11684|  2019|    60|MissedShots|
| 11686|  2019|    60|  SavedShot|
| 11688|  2019|    60|BlockedShot|
+------+------+------+-----------+
only showing top 20 rows
```

9) No of shots of players made where result = goal minute =90 and shots > 5.

```
[71]  1 sql_str = "SELECT shooterId , p.name, SUM(a.shots) as shots , shotResult, minute  FROM shots s " \
      2   + "JOIN appearances a ON(s.shooterId = a.playerID)"\
      3   +" JOIN players p ON(s.shooterId = p.playerID) WHERE a.shots > 5 AND shotResult = 'Goal' AND minute = 90"\
      4   +" GROUP BY shooterId, p.name, shotResult, minute"
      5 spark.sql( sql_str ).show()
```

```
+---------+--------------------+-----+----------+------+
|shooterId|                name|shots|shotResult|minute|
+---------+--------------------+-----+----------+------+
|      337|          Leroy San�|   18|      Goal|    90|
|      942|     Florian Thauvin|  206|      Goal|    90|
|     1518|     Rodrigo Palacio|   13|      Goal|    90|
|      717|     Baf�timbi Gomis|   38|      Goal|    90|
|     3577|      Cheick Diabat�|    7|      Goal|    90|
|      360|         �d�m Szalai|   21|      Goal|    90|
|     1723|     Roberto Pereyra|    6|      Goal|    90|
|     1728|   Fernando Llorente|   24|      Goal|    90|
|     4119|       Alberto Bueno|   40|      Goal|    90|
|     5222|     Joel Pohjanpalo|   13|      Goal|    90|
|     3544|       Jeremie Boga|    6|      Goal|    90|
|      888|     Xherdan Shaqiri|    6|      Goal|    90|
|     1291|           Hernanes|   12|      Goal|    90|
|     3570|  Benjamin Moukandjo|   12|      Goal|    90|
|      802|        Diego Costa|   46|      Goal|    90|
|      318|Pierre-Emerick Au...|  765|      Goal|    90|
|     2328|       Thomas Partey|   12|      Goal|    90|
|     1253|         Edin Dzeko|  307|      Goal|    90|
|     1580|       Mattia Destro|   12|      Goal|    90|
|     1219|        Luis Alberto|  134|      Goal|    90|
+---------+--------------------+-----+----------+------+
only showing top 20 rows
```

10) Gameid, homegoals, awaygoals and awayprobability where awayprobability was greater than homeprobability and awayteam goals where 3 more than home team goals

```
[28]  1 sql_str = "SELECT gameID, homeGoals, awayGoals, awayProbability from games"\
      2 " where awayProbability > homeProbability AND awayGoals > 3*homeGoals "
      3 spark.sql( sql_str ).show()
```

```
+------+---------+---------+---------------+
|gameID|homeGoals|awayGoals|awayProbability|
+------+---------+---------+---------------+
|    90|        0|        3|         0.7687|
|    91|        0|        1|         0.4291|
|    92|        0|        3|         0.4619|
|   108|        0|        2|         0.6532|
|   111|        0|        1|         0.9348|
|   116|        0|        3|          0.467|
|   123|        0|        1|         0.9174|
|   128|        0|        1|         0.4623|
|   149|        0|        1|         0.7996|
|   152|        0|        1|         0.3847|
|   166|        0|        3|         0.5927|
|   168|        0|        3|         0.9274|
|   170|        0|        1|         0.7365|
|   174|        0|        1|         0.4872|
|   175|        0|        2|         0.3846|
|   178|        1|        5|          0.961|
|   185|        0|        3|         0.8712|
|   196|        0|        1|         0.8173|
|   203|        0|        1|         0.4862|
|   205|        0|        3|         0.9674|
+------+---------+---------+---------------+
only showing top 20 rows
```

ROLL UP

11) Get id, season, homeGoals, homeProbability from games table with roll up

```
1 sql_str = "SELECT gameId, season, homeGoals , homeProbability FROM games"\
2 +" group by gameId, season, homeGoals, homeProbability WITH ROLLUP ORDER BY gameId "
3 shots_roll_up = spark.sql( sql_str )
4 shots_roll_up.show()
```

```
+------+------+---------+--------------+
|gameId|season|homeGoals|homeProbability|
+------+------+---------+--------------+
|  null|  null|     null|          null|
|    81|  2015|     null|          null|
|    81|  null|     null|          null|
|    81|  2015|        1|        0.2843|
|    81|  2015|        1|          null|
|    82|  2015|     null|          null|
|    82|  2015|        0|        0.3574|
|    82|  null|     null|          null|
|    82|  2015|        0|          null|
|    83|  2015|     null|          null|
|    83|  null|     null|          null|
|    83|  2015|        2|          null|
|    83|  2015|        2|        0.2988|
|    84|  2015|        4|          null|
|    84|  2015|        4|        0.6422|
|    84|  2015|     null|          null|
|    84|  null|     null|          null|
|    85|  null|     null|          null|
|    85|  2015|        1|          null|
|    85|  2015|        1|        0.1461|
+------+------+---------+--------------+
only showing top 20 rows
```

12) Get shooterid, situation, minute, shotResult with roll up

```
[20] 1 sql_str = "SELECT shooterId, situation, minute , shotResult FROM shots"\
     2 +" group by shooterId, situation, shotResult,minute WITH ROLLUP ORDER BY shooterId"
     3 shots_roll_up = spark.sql( sql_str )
     4 shots_roll_up.show()
```

```
+---------+----------+------+-----------+
|shooterId| situation|minute| shotResult|
+---------+----------+------+-----------+
|     null|      null|  null|       null|
|        2|FromCorner|  null|       null|
|        2|  SetPiece|    67|    OwnGoal|
|        2|  SetPiece|  null|       null|
|        2|  SetPiece|  null|  SavedShot|
|        2|  SetPiece|  null|    OwnGoal|
|        2|FromCorner|    93|MissedShots|
|        2|      null|  null|       null|
|        2|FromCorner|  null|MissedShots|
|        2|  SetPiece|     1|  SavedShot|
|        2|FromCorner|    58|MissedShots|
|        3|FromCorner|  null|  SavedShot|
|        3|  OpenPlay|  null|       Goal|
|        3|FromCorner|    53|MissedShots|
|        3|  SetPiece|    57|MissedShots|
|        3|  OpenPlay|    21|  SavedShot|
|        3|FromCorner|    15|MissedShots|
|        3|FromCorner|    89|BlockedShot|
|        3|  OpenPlay|  null|MissedShots|
|        3|FromCorner|  null| ShotOnPost|
+---------+----------+------+-----------+
only showing top 20 rows
```

13) Playerid, his name, total own goals in different season

```
sql_str = "SELECT a.playerId, p.name, SUM(ownGoals) as ownGoals, g.season  FROM appearances a JOIN players p ON"\
+" (a.playerId = p.PlayerId) JOIN games g ON(a.gameId = g.gameId) GROUP BY a.playerId, p.name, g.season WITH ROLLUP"\
+" ORDER BY a.playerId, g.season"

shots_roll_up = spark.sql( sql_str )
shots_roll_up.show()
```

```
+--------+------------------+--------+------+
|playerId|              name|ownGoals|season|
+--------+------------------+--------+------+
|    null|              null|    1014|  null|
|       1|Christian Mathenia|       0|  null|
|       1|              null|       0|  null|
|       1|Christian Mathenia|       0|  2015|
|       1|Christian Mathenia|       0|  2016|
|       1|Christian Mathenia|       0|  2017|
|       1|Christian Mathenia|       0|  2018|
|       2|              null|       1|  null|
|       2|     Gy?rgy Garics|       1|  null|
|       2|     Gy?rgy Garics|       1|  2015|
|       3|    Luca Caldirola|       0|  null|
|       3|              null|       0|  null|
|       3|    Luca Caldirola|       0|  2014|
|       3|    Luca Caldirola|       0|  2015|
|       3|    Luca Caldirola|       0|  2016|
|       3|    Luca Caldirola|       0|  2017|
|       3|    Luca Caldirola|       0|  2020|
|       4|        Aytac Sulu|       3|  null|
|       4|              null|       3|  null|
|       4|        Aytac Sulu|       2|  2015|
+--------+------------------+--------+------+
only showing top 20 rows
```

PIVOT

14) Get game id, home goals from season 2015,2016,2017,2018

```

sql_str = "SELECT * FROM "\
  +"(SELECT gameId ,season as season, homeGoals FROM games  )  " \
  + "PIVOT ( sum(homeGoals) FOR season IN ( '2015','2016','2017','2018' ))"
pivoted = spark.sql( sql_str )
pivoted.show()
```

```
+------+----+----+----+----+
|gameId|2015|2016|2017|2018|
+------+----+----+----+----+
|  7340|null|null|   0|null|
| 11858|null|null|null|null|
|  2866|null|   1|null|null|
|  3997|null|   0|null|null|
| 15790|null|null|null|null|
| 14832|null|null|null|null|
|  7253|null|null|   1|null|
| 12799|null|null|null|null|
| 13285|null|null|null|null|
|  1088|   2|null|null|null|
|  3175|null|   3|null|null|
|  7554|null|null|   4|null|
| 10206|null|null|null|   0|
|   471|null|   2|null|null|
| 11748|null|null|null|null|
|   148|   2|null|null|null|
|  9376|null|null|null|   3|
|  3794|null|   1|null|null|
|   496|null|   1|null|null|
|  1580|   1|null|null|null|
+------+----+----+----+----+
only showing top 20 rows
```

15) Get shooter id, shot result, total Excepted goals from 75, 34, 90, 13

```
[22]  1
      2
      3 sql_str = "SELECT * FROM "\
      4   +"(SELECT shooterId, xGoal, minute , shotResult FROM shots ) " \
      5   + "PIVOT ( SUM(xGoal) FOR minute IN ( 75,34,90,13 ))"
      6 pivoted = spark.sql( sql_str )
      7 pivoted.show()
```

```
+---------+-----------+------------------+------------------+------------------+-------------------+
|shooterId| shotResult|                75|                34|                90|                 13|
+---------+-----------+------------------+------------------+------------------+-------------------+
|     2370|       Goal|0.4882115051150328|              null|0.5338043384253979| 0.5510657429695132|
|     2233|BlockedShot|              null|              null|              null|               null|
|     3795|MissedShots|0.0822890996932983|              null|1.2884900569915778| 0.3453409960493449|
|     3422|  SavedShot|              null| 0.105036497116089|  0.0537331513134|               null|
|      301|BlockedShot| 0.107572317123413|0.0515943244099617|              null|               null|
|      300|  SavedShot|0.0297455172985792|0.0677571892738342|              null| 0.0501434840261936|
|     4297|MissedShots|              null|              null|              null|  0.246945753693581|
|     1029|       Goal|              null|              null|              null|               null|
|      633|MissedShots|              null|              null|              null|               null|
|     1604|MissedShots| 0.384754240512848|0.0534644387662411|              null| 0.0161092840135098|
|      453|  SavedShot|              null|0.8364370167255404|              null|               null|
|      185|MissedShots|              null|              null|              null|               null|
|      313|BlockedShot|              null|              null|0.0158107243478298|               null|
|     1726|  SavedShot|              null|              null|0.0443860851228237| 0.0641456246376038|
|      878|  SavedShot|0.1181230247020721|              null|              null|               null|
|     4060|  SavedShot|              null|              null|              null|               null|
|      603|MissedShots|0.0835353508591652| 0.045158039778471|              null|               null|
|       64|MissedShots| 0.116914980113506|0.0130163738504052| 0.436274163424968|0.09385521896183491|
|     1101|       Goal|              null|              null|              null|               null|
|     5619|  SavedShot|              null|0.0624480918049812|              null|               null|
+---------+-----------+------------------+------------------+------------------+-------------------+
only showing top 20 rows
```

Conclusion:

We have successfully executed about 15 queries to analyze player's states, game info. From this project we learned how to deal with big data using aggregation and OLAP queries and we will try to improve this project in the future.

References :

1. https://www.geeksforgeeks.org/olap-operations-in-dbms/
2. https://www.kaggle.com/datasets/technika148/football-database
3. https://dbdiagram.io/