

Football Data Warehouse

MVP

Disciplina: Sprint: Engenharia de Dados

Aluno: Lucas Araujo Abbade

1. Objetivo

Identificar os times e jogadores de maior destaque no cenário do futebol mundial, através dos seguintes critérios:

Quais os times que ganharam mais títulos?

Quais times se mantiveram nas primeiras posições por mais tempo?

Quais os times/jogadores fizeram mais gols?

Quais os jogadores que têm mais assistências?

Quais são os jogadores com maior participação em gol (assistência/gol)?

Quais os times que jogam melhor dentro e fora de casa?

Quais os times/jogadores com melhor aproveitamento de chutes?

Quais são os times/jogadores mais violentos (cartões amarelos/vermelhos e nº de faltas)?

Quais os times/jogadores fizeram mais gols contra?

O número de gols está aumentando ou caindo com o passar do tempo?

Os gols acontecem mais em que período do jogo?

Quais são os tipos de gols mais comuns?

Quais são os tipos de jogadas que mais resultam em gols?

Quais times fazem mais gols de escanteio? E de cabeça?

2. Coleta

Foi realizada uma pesquisa no repositório aberto de dados da Kaggle onde foi escolhido um dataset relacionado a futebol. Os dados apresentam informações à respeito dos times e dos jogadores das 5 principais ligas de futebol europeias no período de 2014-2020. Os dados se encontram no link [Football Database | Kaggle](https://www.kaggle.com/datasets/technika148/football-database) e foram baixados para a máquina local. O software definido para o projeto foi o ambiente em nuvem Google Cloud Platform (GCP). Sendo assim, os dados foram inseridos manualmente no Storage do GCP, conforme mostra a Figura 2.

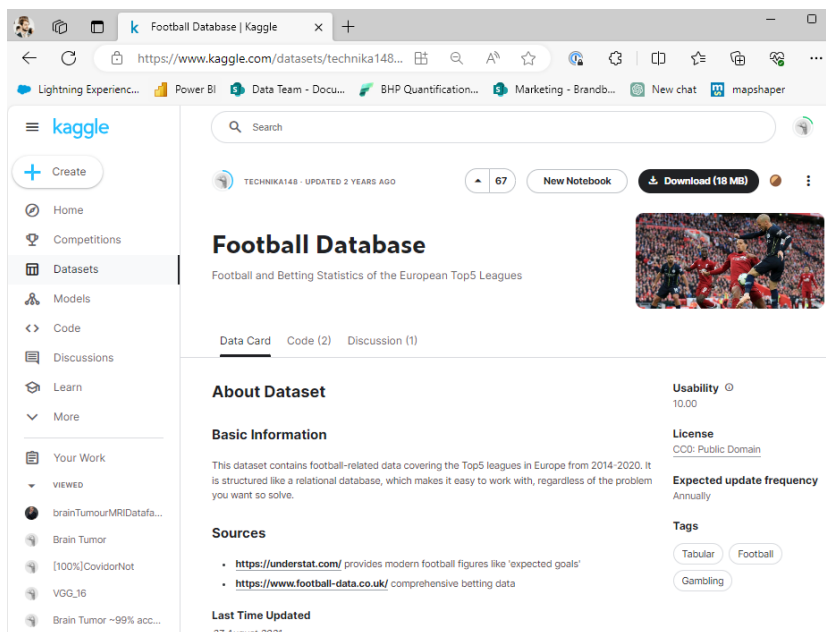


Figura 1

☰

Google Cloud

My First Project

Search (/) for resources, docs, products, and more

Cloud Storage

Buckets

Monitoring

Settings

← Bucket details

lucas-abbade_mvp3

Location

us (multiple regions in United States)

Storage class

Standard

Public access

Not public

Protection

None

OBJECTS

CONFIGURATION

PERMISSIONS

PROTECTION

LIFECYCLE

OBSERVABILITY

INVENTORY REPORTS

Buckets

>

lucas-abbade_mvp3

UPLOAD FILES

UPLOAD FOLDER

CREATE FOLDER

TRANSFER DATA

MANAGE HOLDS

DOWNLOAD

DELETE

Filter by name prefix only

Filter

Filter objects and folders








<input type="checkbox"/>	Name	Size	Type	Created	Storage class	Last modified
<input type="checkbox"/>	 appearances.csv	33.7 MB	text/csv	Sep 23, 2023, 9:41:08 PM	Standard	Sep 23, 2023, 9:41:08 PM
<input type="checkbox"/>	 games.csv	2.1 MB	text/csv	Sep 23, 2023, 9:41:02 PM	Standard	Sep 23, 2023, 9:41:02 PM
<input type="checkbox"/>	 leagues.csv	173 B	text/csv	Sep 23, 2023, 9:40:58 PM	Standard	Sep 23, 2023, 9:40:58 PM
<input type="checkbox"/>	 players.csv	182.6 KB	text/csv	Sep 23, 2023, 9:40:58 PM	Standard	Sep 23, 2023, 9:40:58 PM
<input type="checkbox"/>	 shots.csv	36 MB	text/csv	Sep 23, 2023, 9:41:08 PM	Standard	Sep 23, 2023, 9:41:08 PM
<input type="checkbox"/>	 teams.csv	2.7 KB	text/csv	Sep 23, 2023, 9:41:00 PM	Standard	Sep 23, 2023, 9:41:00 PM
<input type="checkbox"/>	 teamstats.csv	2 MB	text/csv	Sep 23, 2023, 9:41:00 PM	Standard	Sep 23, 2023, 9:41:00 PM

Figura 2

Os dados brutos deste banco de dados são compostos por 7 tabelas em csv conforme a imagem acima ilustra. Cada um destes arquivos será carregado na Data Warehouse deste projeto, sofrerá transformações e será estruturado de acordo com a modelagem descrita na próxima etapa.

3. Modelagem

A linhagem dos dados segue um padrão muito simples. Os dados escolhidos no repositório Kaggle estão em csv, são baixados na máquina local e transferidos manualmente para o GCP Storage em nuvem. Em seguida, foi utilizada a ferramenta de ETL do GCP Data Fusion para transformar e carregar os dados no BigQuery. A imagem abaixo descreve o processo.



Figura 3

O Google BigQuery, embora não seja um banco de dados relacional tradicional, é uma plataforma de análise em nuvem que permite a manipulação de dados usando SQL. Ele armazena tabelas e views em conjuntos de datasets e possui uma notável capacidade de processamento de dados, possibilitando consultas em grandes volumes de dados em segundos. Portanto, é possível armazenar e estruturar os dados do projeto no BigQuery, possibilitando a simulação de um modelo em esquema de constelação de fatos.

Os dados estão divididos em 7 arquivos os quais foram carregados em 7 tabelas distintas, sem a necessidade de aplicação de joins. Os dados baixados do Kaggle já estão relativamente bem estruturados, sendo necessário apenas a aplicação de algumas restrições, transformações e criação de chaves primárias únicas para a integridade do modelo, os quais estão bem detalhados no tópico 4 do projeto (Carga).

Sendo assim, o modelo descrito corresponde a um esquema de constelação de fatos no qual existem 4 tabelas fato (appearances, games, shots e teamstats) e 3 dimensão (players, leagues e teams). As tabelas fato possuem relacionamentos tanto com as tabelas dimensão quanto entre si.

- Tabelas dimensão:

1) leagues

Esta tabela traz o nome das 5 principais ligas que estão presentes nesta data warehouse ("La Liga", "Ligue 1", "Serie A", "Premier League" e "Bundesliga"). Cada linha corresponde à uma das ligas. A tabela possui relação 1xN com as tabelas fato games e appearances.

2) teams

Esta tabela traz o nome dos times das 5 principais ligas que estão presentes nesta data warehouse durante o período de 2014-2020. Cada linha corresponde à um dos times. A tabela possui relação 1xN com as tabelas fato games e teamstats.

3) players

Esta tabela traz o nome dos jogadores dos times das 5 principais ligas que estão presentes nesta data warehouse durante o período de 2014-2020. Cada linha corresponde à um jogadores. A tabela possui relação 1xN com as tabelas fato appearances e shots.

- Tabelas fato:

1) games

Esta tabela traz métricas referentes ao número de gols de cada time em uma determinada partida. Cada linha corresponde à uma partida. Faz conexão Nx1 com a dimensão leagues e 1xN com a fato appearances. Esta tabela apresenta dois atributos (date e season) que podem ser entendidos como dimensões, porém não foram normalizadas em uma nova tabela dimensão pois date apresenta uma granularidade muito grande, por se tratar de uma variável datetime e season é diretamente relacionada à variável datetime.

2) shots

Esta tabela traz métricas que qualificam os chutes em determinada partida de futebol. As linhas da tabela correspondem à cada chute ocorrido na partida. Apesar de se tratarem de atributos string ainda assim devem ser entendidos como fato pois são termos qualitativos e descritivos do evento principal, de modo que não há métricas numéricas nesta tabela. A tabela possui relação Nx1 com a fato games e relação Nx1 com a dimensão players.

3) appearances

Esta tabela traz métricas a respeito de participação de um jogador de futebol em uma determinada partida, como por exemplo número de chutes, cartões amarelos e vermelhos, gols, assistências, dentre outros. Cada linha desta tabela corresponde à um jogador que entrou em campo em determinado jogo. A tabela possui relação Nx1 com a fato games, relação Nx1 com a dimensão players e relação Nx1 com a dimensão leagues. O campo time pode ser entendido como uma dimensão, porém como ele é um atributo numérico que varia de 1 a 90, ao criarmos uma surrogate key os valores seriam iguais à própria variável em si.

4) teamstats

Esta tabela traz métricas referentes aos principais indicadores de um time em uma partida de futebol, como por exemplo número de chutes, cartões amarelos e vermelhos, gols, escanteios, faltas, resultado do jogo, dentre outros. Cada linha corresponde à um time que participou do jogo, ou seja, cada jogo aparece exatamente em duas linhas, uma para cada equipe. A tabela possui relação Nx1 com a fato games, relação Nx1 com a dimensão teams.

O diagrama de Modelo de Entidade-Relacionamento (DER) abaixo representa de forma visual toda a estrutura do modelo.

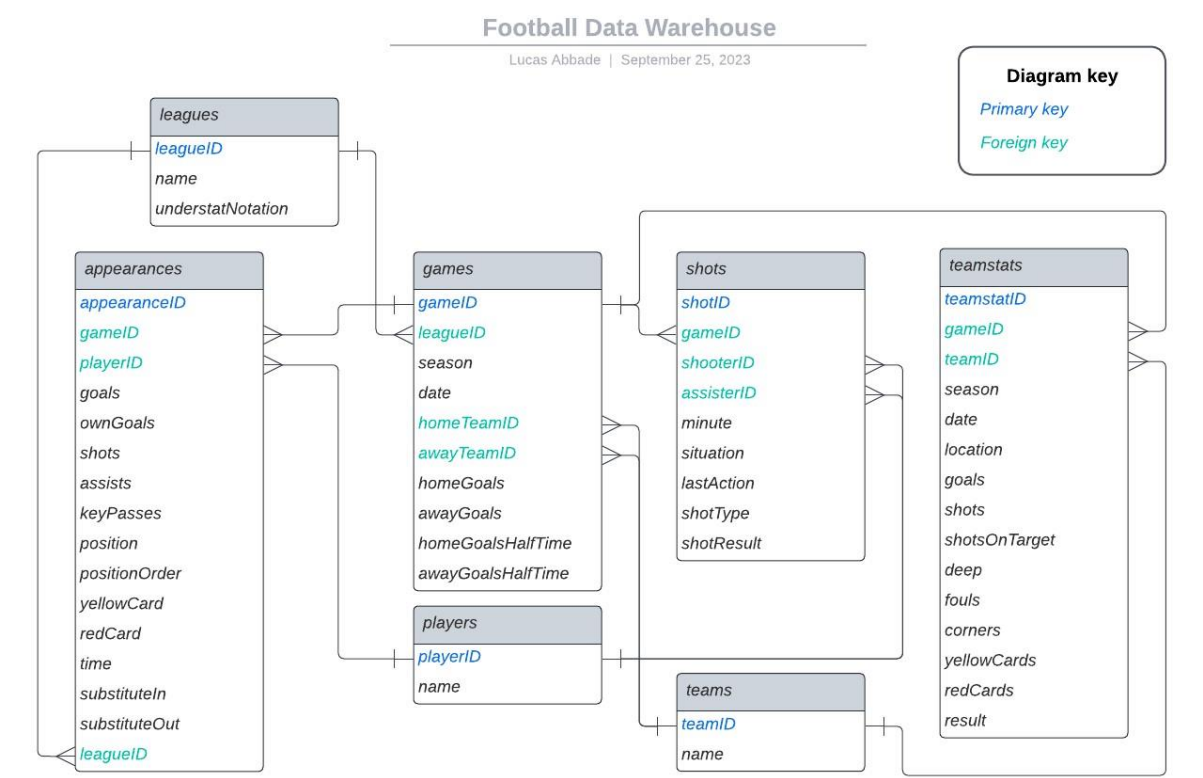


Figura 4

Por fim, as 7 imagens abaixo trazem as descrições de todas as variáveis do modelo, bem como os seus domínios e restrições de não nulidade.

appearances	QUERY	SHARE	COPY	SNAPSHOT	DELETE	EXPORT	
SCHEMA	DETAILS	PREVIEW	LINEAGE	DATA PROFILE	DATA QUALITY		
Filter: Enter property name or value							
<input type="checkbox"/>	Field name	Type	Mode	Key	Collation	Default Value	Policy Tags
<input type="checkbox"/>	appearanceID	STRING	REQUIRED				
<input type="checkbox"/>	gameID	INTEGER	REQUIRED				
<input type="checkbox"/>	playerID	INTEGER	REQUIRED				
<input type="checkbox"/>	goals	INTEGER	REQUIRED				
<input type="checkbox"/>	ownGoals	INTEGER	REQUIRED				
<input type="checkbox"/>	shots	INTEGER	REQUIRED				
<input type="checkbox"/>	assists	INTEGER	REQUIRED				
<input type="checkbox"/>	keyPasses	INTEGER	REQUIRED				
<input type="checkbox"/>	position	STRING	REQUIRED				
<input type="checkbox"/>	positionOrder	INTEGER	REQUIRED				
<input type="checkbox"/>	yellowCard	INTEGER	REQUIRED				
<input type="checkbox"/>	redCard	INTEGER	REQUIRED				
<input type="checkbox"/>	time	INTEGER	REQUIRED				
<input type="checkbox"/>	substituteIn	INTEGER	REQUIRED				
<input type="checkbox"/>	substituteOut	INTEGER	REQUIRED				
<input type="checkbox"/>	leagueID	INTEGER	REQUIRED				

Figura 5

games

QUERYSHARECOPYSNAPSHOTDELETEDELETEEXPORT

SCHEMADETAILEDPREVIEWLINEAGEDATA PROFILEDATA QUALITY

Filter

Enter property name or value

Field name	Type	Mode	Key	Collation	Default Value	Policy Tags	Description
gameID	INTEGER	REQUIRED					Chave Primária da tabela. Composta por um número inteiro aleatório.
leagueID	INTEGER	REQUIRED					Chave Estrangeira que conecta à tabela leagues.
season	INTEGER	REQUIRED					Ano de início da temporada. Valores entre 2014 e 2020.
date	DATETIME	REQUIRED					Data e hora do jogo. Os jogos vão de 2014-08-08 19:30 a 2021-05-23 19:00.
homeTeamID	INTEGER	REQUIRED					Chave Estrangeira que conecta à tabela teams. Traz o nome do time da casa.
awayTeamID	INTEGER	REQUIRED					Chave Estrangeira que conecta à tabela teams. Traz o nome do time visitante.
homeGoals	INTEGER	REQUIRED					Número de gols do time da casa. Valores vão de 0 a 10.
awayGoals	INTEGER	REQUIRED					Número de gols do time visitante. Valores vão de 0 a 9.
homeGoalsHalfTime	INTEGER	REQUIRED					Número de gols do time da casa no primeiro tempo. Valores vão de 0 a 6.
awayGoalsHalfTime	INTEGER	REQUIRED					Número de gols do time visitante no primeiro tempo. Valores vão de 0 a 5.

Figura 6

leagues

QUERYSHARECOPYSNAPSHOTDELETEDELETEEXPORT

SCHEMADETAILEDPREVIEWLINEAGEDATA PROFILEDATA QUALITY

Filter

Enter property name or value

Field name	Type	Mode	Key	Collation	Default Value	Policy Tags	Description
leagueID	INTEGER	REQUIRED					Chave Primária da tabela. Composta por um número inteiro aleatório.
name	STRING	REQUIRED					Nome das ligas. As ligas presentes são: "La Liga", "Ligue 1", "Serie A", "Premier League" e "Bundesliga"
understatNotation	STRING	REQUIRED					Notação do nome das ligas. Underline no lugar de espaços.

Figura 7

players

QUERYSHARECOPYSNAPSHOTDELETEDELETEEXPORT

SCHEMADETAILEDPREVIEWLINEAGEDATA PROFILEDATA QUALITY

Filter

Enter property name or value

Field name	Type	Mode	Key	Collation	Default Value	Policy Tags	Description
playerID	INTEGER	REQUIRED					Chave Primária da tabela. Composta por um número inteiro aleatório.
name	STRING	REQUIRED					Nome de todos os jogadores dos times que participaram das ligas presentes na tabela leagues no período de 2014-2020.

Figura 8

shots

QUERYSHARECOPYSNAPSHOTDELETEDELETEEXPORT

SCHEMADETAILEDPREVIEWLINEAGEDATA PROFILEDATA QUALITY

Filter

Enter property name or value

Field name	Type	Mode	Key	Collation	Default Value	Policy Tags	Description
shotID	INTEGER	REQUIRED					Chave Primária da tabela. Composta por um número inteiro aleatório.
gameID	INTEGER	REQUIRED					Chave Estrangeira que conecta à tabela games.
shooterID	INTEGER	REQUIRED					Chave Estrangeira que conecta à tabela players. Traz o nome do jogador que deu o chute.
assisterID	INTEGER	REQUIRED					Chave Estrangeira que conecta à tabela players. Traz o nome do jogador que deu à assistência. Caso não haja uma assistência o valor é 0.
minute	INTEGER	REQUIRED					Minuto em que o chute aconteceu. Valores vão de 0 a 104.
situation	STRING	REQUIRED					Situação em que ocorreu o chute. Os valores podem ser: "DirectFreekick", "SetPiece", "OpenPlay", "FromCorner" e "Penalty".
lastAction	STRING	REQUIRED					Última ação antes do chute ser realizado. Os valores podem ser: "Standard", "Pass", "Tackle", "BallRecovery", "None", "Cross", "Chipped", "Rebound", "Aerial", "Dispossessed", "Throughball", "HeadPass", "TakeOn", "LayOff", "BallTouch", "Interception", "Foul", "Save", "BlockedPass", "Challenge", "End", "Goal", "Clearance", "CornerAwarded", "GoodSkill", "OffsidePass", "Error", "KeeperSweeper", "Card", "SubstitutionOn", "PenaltyFaced", "Start", "Punch", "ShieldBallOpp", "CrossNotClaimed", "OffsideProvoked", "FormationChange", "KeeperPickup", "ChanceMissed", "Smother" e "SubstitutionOff".
shotType	STRING	REQUIRED					Tipo de chute. Os valores podem ser: "LeftFoot", "RightFoot", "Head" e "OtherBodyPart".
shotResult	STRING	REQUIRED					Resultado do chute. Os valores podem ser: "BlockedShot", "MissedShots", "SavedShot", "OwnGoal", "Goal" e "ShotOnPost".

Figura 9

teams

QUERY

SHARE

COPY

SNAPSHOT

DELETE

EXPORT

SCHEMA

DETAILS

PREVIEW

LINEAGE

DATA PROFILE

DATA QUALITY

Filter

Enter property name or value

<input type="checkbox"/>	Field name	Type	Mode	Key	Collation	Default Value	Policy Tags	Description
<input type="checkbox"/>	teamID	INTEGER	REQUIRED					Chave Primária da tabela. Composta por um número inteiro aleatório.
<input type="checkbox"/>	name	STRING	REQUIRED					Nome de todos os times que participaram das ligas presentes na tabela leagues no período de 2014-2020.

Figura 10

teamstats

QUERY

SHARE

COPY

SNAPSHOT

DELETE

EXPORT

SCHEMA

DETAILS

PREVIEW

LINEAGE

DATA PROFILE

DATA QUALITY

Filter

Enter property name or value

<input type="checkbox"/>	Field name	Type	Mode	Key	Collation	Default Value	Policy Tags	Description
<input type="checkbox"/>	teamstatID	STRING	REQUIRED					Chave Primária da tabela. Composta pela combinação das variáveis gameId e teamID.
<input type="checkbox"/>	gameID	INTEGER	REQUIRED					Chave Estrangeira que conecta à tabela games.
<input type="checkbox"/>	teamID	INTEGER	REQUIRED					Chave Estrangeira que conecta à tabela teams.
<input type="checkbox"/>	season	INTEGER	REQUIRED					Ano de início da temporada. Valores entre 2014 e 2020.
<input type="checkbox"/>	date	DATETIME	REQUIRED					Data e hora do jogo. Os jogos vão de 2014-08-08 19:30 a 2021-05-23 19:00.
<input type="checkbox"/>	location	STRING	REQUIRED					Localização do jogo. Em casa 'h' ou visitante 'a'.
<input type="checkbox"/>	goals	INTEGER	REQUIRED					Número de gols do time. Valores vão de 0 a 10.
<input type="checkbox"/>	shots	INTEGER	REQUIRED					Número de chutes do time. Valores vão de 0 a 47.
<input type="checkbox"/>	shotsOnTarget	INTEGER	REQUIRED					Número de chutes no gol do time. Valores vão de 0 a 18.
<input type="checkbox"/>	deep	INTEGER	REQUIRED					Passes completados pelo time a uma distância maior do que 20 jardas do gol (cruzamentos excluídos). Valores vão de 0 a 42.
<input type="checkbox"/>	fouls	INTEGER	REQUIRED					Faltas cometidas pelo time. Valores vão de 0 a 33.
<input type="checkbox"/>	corners	INTEGER	REQUIRED					Escanteios cobrados pelo time. Valores vão de 0 a 20.
<input type="checkbox"/>	yellowCards	INTEGER	REQUIRED					Cartões amarelos recebidos pelo time. Valores vão de 0 a 9.
<input type="checkbox"/>	redCards	INTEGER	REQUIRED					Cartões vermelhos recebidos pelo time. Valores vão de 0 a 3.
<input type="checkbox"/>	result	STRING	REQUIRED					Resultado do jogo para o time. Vitória 'W', empate 'D' e derrota 'L'.

Figura 11

4. Carga

Para realizar a carga dos dados na Data Warehouse foi utilizado a ferramenta de ETL do GCP chamada Data Fusion. Nesta, foi criado um pipeline de dados para cada um dos arquivos em csv no qual os dados são extraídos do Storage, recebem um sequência de transformações e são carregados no Google BigQuery, onde por fim é alocada a nossa DW.

Seguem abaixo as figuras que demostram os pipelines mencionados.

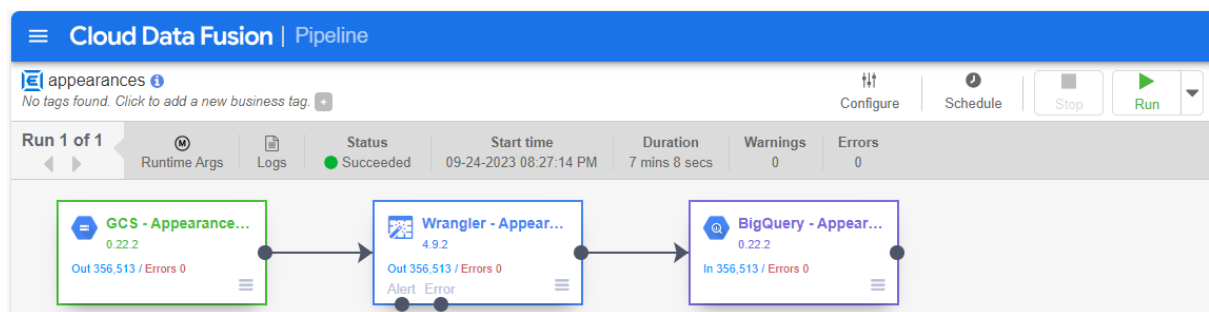


Figura 12

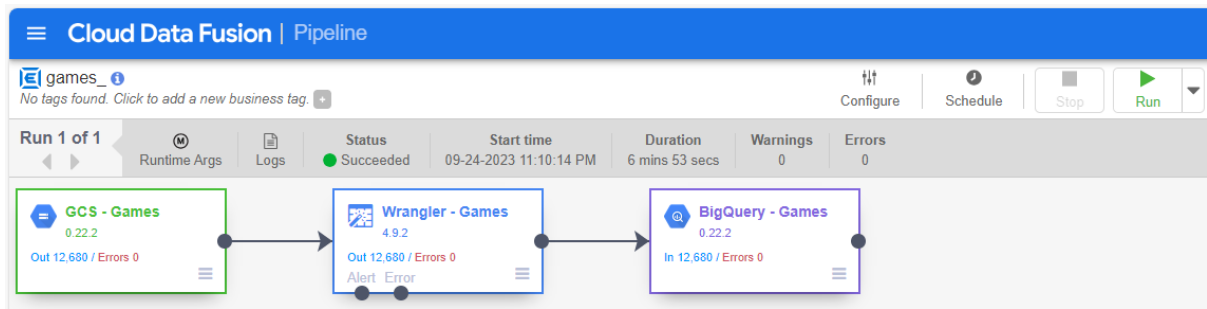


Figura 13

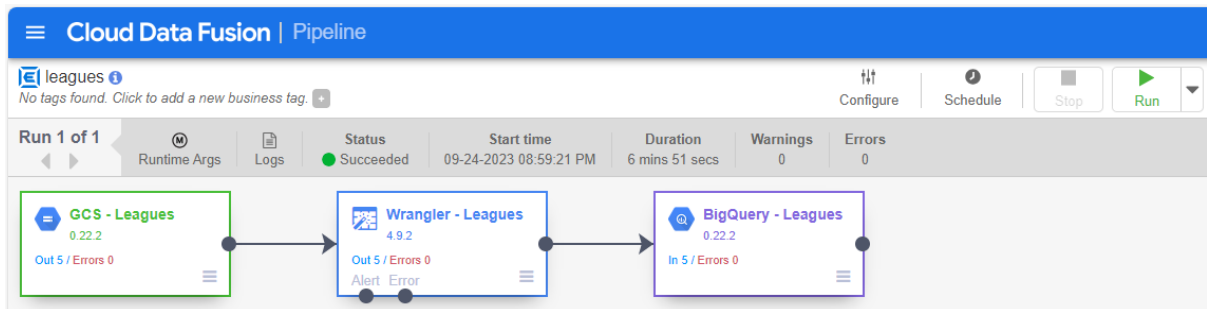


Figura 14

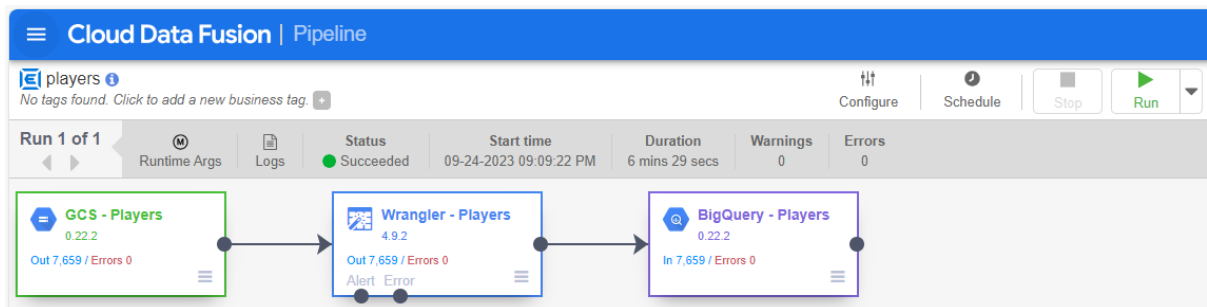


Figura 15

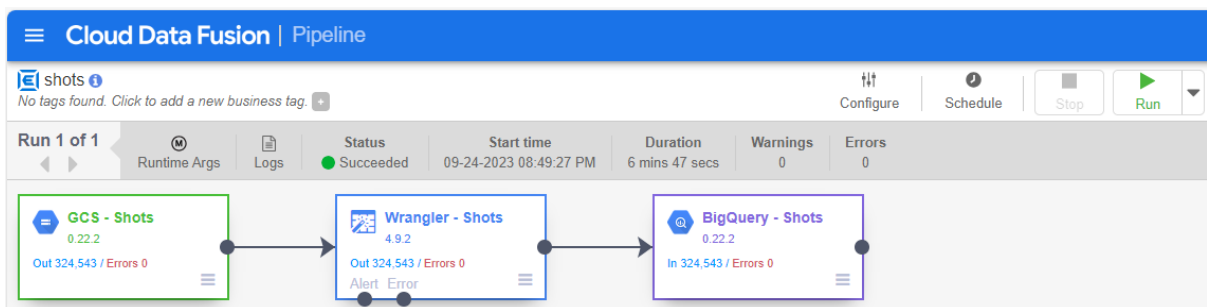


Figura 16

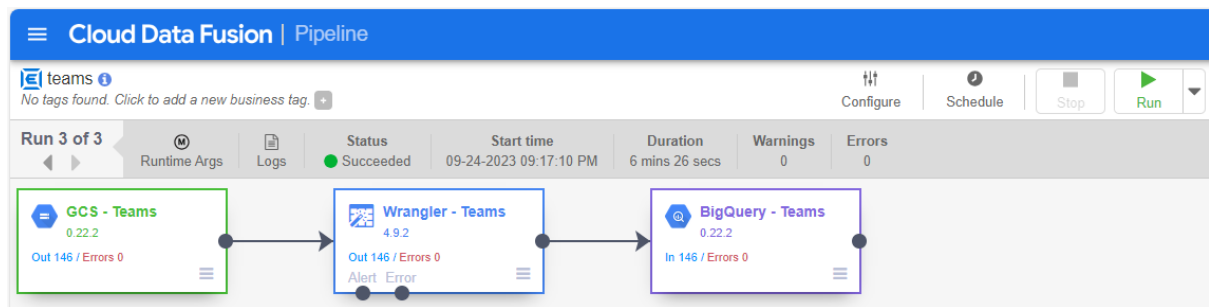


Figura 17

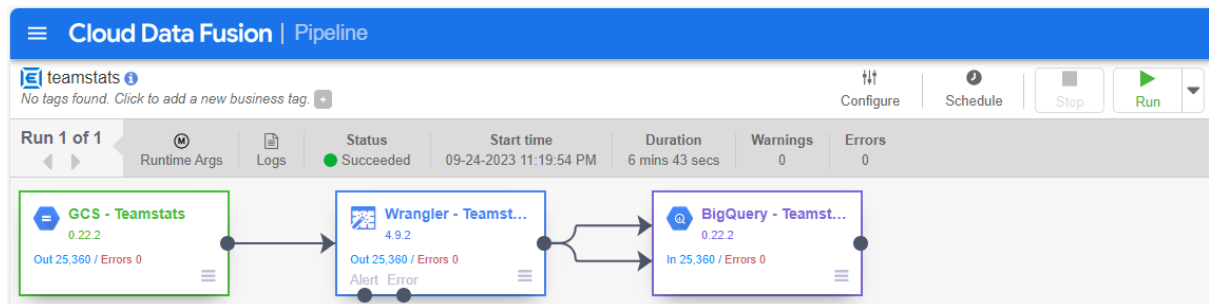


Figura 18

Após a execução de cada pipeline individualmente obteve-se sucesso em todos conforme a imagem abaixo.

Cloud Data Fusion | Pipeline

Deployed | Drafts

Search by pipeline name Showing 7 of 7 pipelines

Pipeline name	Type	Status	Last start time
appearances	Batch	Succeeded	09-24-2023 08:27:14 PM
games_	Batch	Succeeded	09-24-2023 11:10:14 PM
leagues	Batch	Succeeded	09-24-2023 08:59:21 PM
players	Batch	Succeeded	09-24-2023 09:09:22 PM
shots	Batch	Succeeded	09-24-2023 08:49:27 PM
teams	Batch	Succeeded	09-24-2023 09:17:10 PM
teamstats	Batch	Succeeded	09-24-2023 11:19:54 PM

Figura 19

Deste modo, todas as tabelas foram carregadas no BigQuery, conforme o a imagem abaixo.

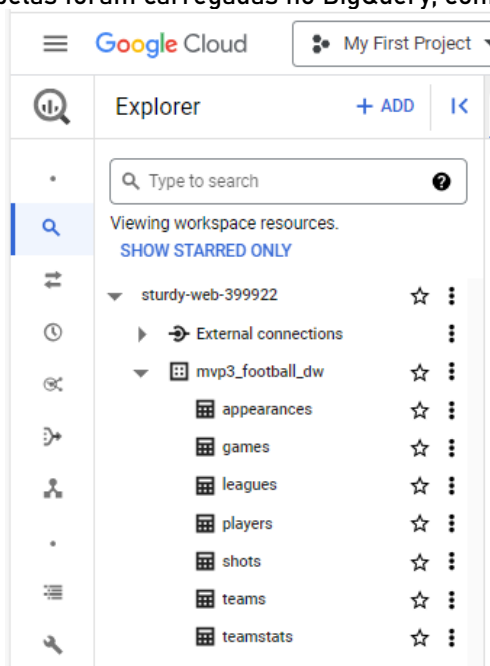


Figura 20

Abaixo foram destacadas as transformações realizadas dentro do Wrangler (box de transformação no pipeline de dados) de cada uma das tabelas e também as transformações realizadas já no ambiente do BigQuery (nos casos em que foi necessário).

4.1. appearances

Nesta transformação do Wrangler removeram-se algumas colunas que não seriam importantes para a análise do projeto e em seguida transformou-se o domínio de 5 outros atributos para números inteiros.

Cloud Data Fusion Studio														
Cloud Storage Default - lucas-abbade_mvp3/appearances.csv														
appearances.csv Columns: 15 Rows: 1000														
	gameID	playerID	goals	ownGoals	shots	assists	keyPasses	position	positionOrder	yellowCard	redCard			
1	81	560	0	0	0	0	0	GK	1	0	0	90		
2	81	557	0	0	0	0	1	DR	2	0	0	82		
3	81	548	0	0	0	0	0	DC	3	0	0	90		
4	81	628	0	0	0	0	0	DC	3	0	0	90		
5	81	1006	0	0	0	0	0	DL	4	0	0	90		
6	81	551	0	0	0	0	0	DMC	7	0	0	90		
7	81	654	0	0	0	0	1	DMC	7	0	0	61		

Figura 21

Em seguida, no BigQuery foi preciso criar uma chave primária para esta tabela. Para isto, foi feita uma chave a partir da combinação entre gameId e playerId, uma vez que esta chave é única, dado que cada jogador só pode aparecer uma única vez dentro de uma partida. Para realizar esta tarefa, foi criada uma tabela temporária com todas as colunas mais a chave criada, em seguida criou-se uma tabela nova, para inserir a restrição de valores não nulos e seus domínios em todos os atributos. Por fim os valores foram inseridos. Deste modo, garantiu-se que as restrições de chave, entidade e domínio sejam respeitadas. Segue abaixo a query utilizada.

```
-- appearances
create or replace table `sturdy-web-399922.mvp3_football_dw.appearances_2` as
select
| CONCAT(gameID, '_', playerId) as appearanceID
| *
from `sturdy-web-399922.mvp3_football_dw.appearances`;

drop table `sturdy-web-399922.mvp3_football_dw.appearances`;

CREATE TABLE `sturdy-web-399922.mvp3_football_dw.appearances` (
appearanceID STRING NOT NULL,
gameID INT64 NOT NULL,
playerID INT64 NOT NULL,
goals INT64 NOT NULL,
ownGoals INT64 NOT NULL,
shots INT64 NOT NULL,
assists INT64 NOT NULL,
keyPasses INT64 NOT NULL,
position STRING NOT NULL,
positionOrder INT64 NOT NULL,
yellowCard INT64 NOT NULL,
redCard INT64 NOT NULL,
time INT64 NOT NULL,
substituteIn INT64 NOT NULL,
substituteOut INT64 NOT NULL,
leagueID INT64 NOT NULL
);

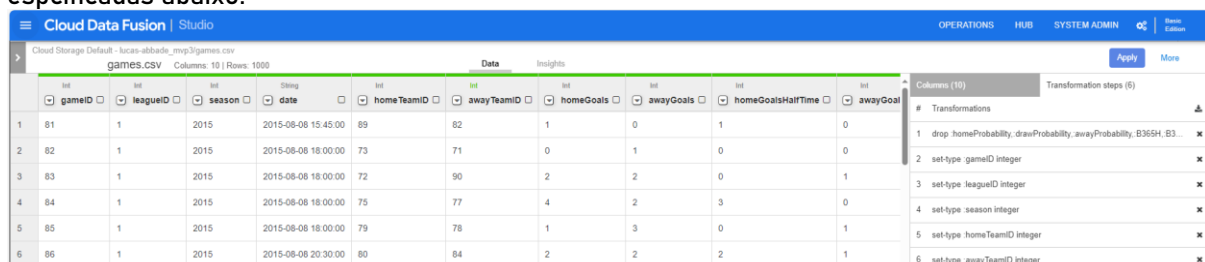
insert into `sturdy-web-399922.mvp3_football_dw.appearances`
SELECT * FROM `sturdy-web-399922.mvp3_football_dw.appearances_2`;

drop table `sturdy-web-399922.mvp3_football_dw.appearances_2`;
```

Figura 22

4.2. games

Nesta transformação do Wrangler foram feitas alterações iguais às da tabela anterior. Removeram-se algumas colunas e transformaram-se outras em números inteiros. Seguem as transformações especificadas abaixo.



	gameID	leagueID	season	date	homeTeamID	awayTeamID	homeGoals	awayGoals	homeGoalsHalfTime	awayGoalsHalfTime
1	81	1	2015	2015-08-08 15:45:00	89	82	1	0	1	0
2	82	1	2015	2015-08-08 18:00:00	73	71	0	1	0	0
3	83	1	2015	2015-08-08 18:00:00	72	90	2	2	0	1
4	84	1	2015	2015-08-08 18:00:00	75	77	4	2	3	0
5	85	1	2015	2015-08-08 18:00:00	79	78	1	3	0	1
6	86	1	2015	2015-08-08 20:30:00	80	84	2	2	2	1

Figura 23

Nesta tabela precisou-se formatar a coluna de data em datetime no BigQuery pois esta transformação no Data Fusion incorreu em diversos erros. Realizou-se essa transformação em três etapas conforme a tabela anterior. Criou-se uma tabela temporária com a formatação da coluna date e em seguida criou-se a nova tabela com as restrições adequadas e inseriu nesta os dados. Segue a query.

```
-- games
create or replace table `sturdy-web-399922.mvp3_football_dw.games_2` as
select
| gameID
, leagueID
, season
, datetime(date) as date
, homeTeamID
, awayTeamID
, homeGoals
, awayGoals
, homeGoalsHalfTime
, awayGoalsHalfTime
from `sturdy-web-399922.mvp3_football_dw.games`;

drop table `sturdy-web-399922.mvp3_football_dw.games`;

CREATE TABLE `sturdy-web-399922.mvp3_football_dw.games` (
gameID INT64 NOT NULL,
leagueID INT64 NOT NULL,
season INT64 NOT NULL,
date DATETIME NOT NULL,
homeTeamID INT64 NOT NULL,
awayTeamID INT64 NOT NULL,
homeGoals INT64 NOT NULL,
awayGoals INT64 NOT NULL,
homeGoalsHalfTime INT64 NOT NULL,
awayGoalsHalfTime INT64 NOT NULL
);

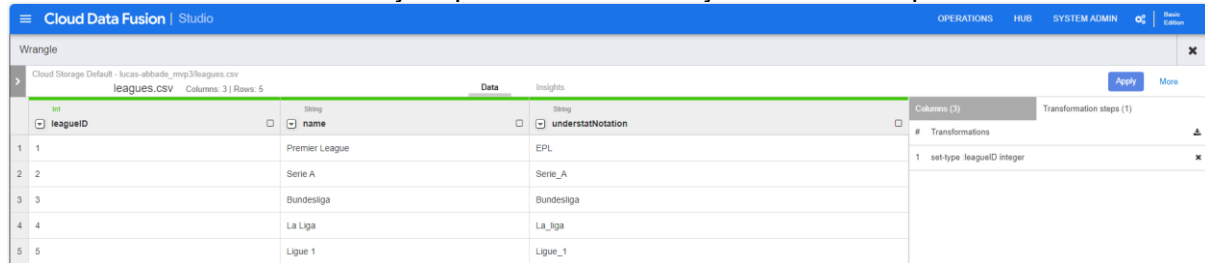
insert into `sturdy-web-399922.mvp3_football_dw.games`
SELECT * FROM `sturdy-web-399922.mvp3_football_dw.games_2`;

drop table `sturdy-web-399922.mvp3_football_dw.games_2`;
```

Figura 24

4.3. leagues

Nesta tabela a única transformação aplicada foi a formatação da sua chave primária em inteiro.

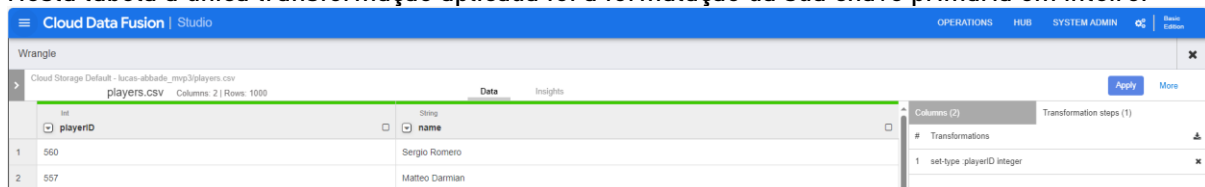


leagueID	name	understatNotation
1	Premier League	EPL
2	Serie A	Serie_A
3	Bundesliga	Bundesliga
4	La Liga	La_liga
5	Ligue 1	Ligue_1

Figura 25

4.4. players

Nesta tabela a única transformação aplicada foi a formatação da sua chave primária em inteiro.

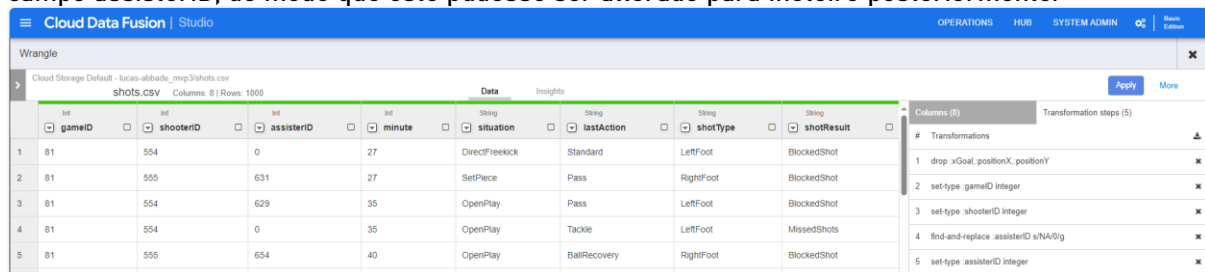


playerID	name
560	Sergio Romero
557	Matteo Darmian

Figura 26

4.5. shots

Nesta transformação do Wrangler três colunas sem importância para análise foram removidas. Por fim, gameId e shooterID foram transformados em inteiros e foi realizado um replace de 'NA' por '0' no campo assisterID, de modo que este pudesse ser alterado para ineteiro posteriormente.



gameID	shooterID	assisterID	minute	situation	lastAction	shotType	shotResult
81	554	0	27	DirectFreekick	Standard	LeftFoot	BlockedShot
81	555	631	27	SetPiece	Pass	RightFoot	BlockedShot
81	554	629	35	OpenPlay	Pass	LeftFoot	BlockedShot
81	554	0	35	OpenPlay	Tackle	LeftFoot	MissedShots
81	555	654	40	OpenPlay	BallRecovery	RightFoot	BlockedShot

Figura 27

Para esta tabela realizamos alterações no BigQuery com as mesmas três etapas das outras tabelas que necessitaram modificações posteriores. Esta alteração foi realizada somente para inserir uma chave primária para a tabela. Como esta tabela não possui chaves candidatas, foi necessário a criação de uma surrogate key. Segue a Query.

```
-- shots
create or replace table `sturdy-web-399922.mvp3_football_dw.shots_2` as
select
| row_number() OVER() as shotID
| *
from `sturdy-web-399922.mvp3_football_dw.shots`;

drop table `sturdy-web-399922.mvp3_football_dw.shots`;

CREATE TABLE `sturdy-web-399922.mvp3_football_dw.shots` (
shotID INT64 NOT NULL,
gameID INT64 NOT NULL,
shooterID INT64 NOT NULL,
assisterID INT64 NOT NULL,
minute INT64 NOT NULL,
situation STRING NOT NULL,
lastAction STRING NOT NULL,
shotType STRING NOT NULL,
shotResult STRING NOT NULL
);

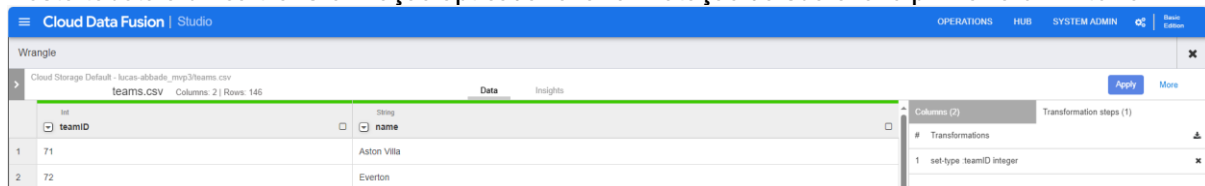
insert into `sturdy-web-399922.mvp3_football_dw.shots`
SELECT * FROM `sturdy-web-399922.mvp3_football_dw.shots_2`;

drop table `sturdy-web-399922.mvp3_football_dw.shots_2`;
```

Figura 28

4.6. teams

Nesta tabela a única transformação aplicada foi a formatação da sua chave primária em inteiro.



Int	String
teamID	name
71	Aston Villa
72	Everton

Figura 29

4.7. teamstats

Nesta transformação foram removidas duas colunas e formatadas outras três como inteiro. Por fim foi realizado um replace de 'NA' por '0' na coluna yellowCards.

Cloud Data Fusion

Studio

OPERATIONS

HUB

SYSTEM ADMIN

QC

Recent Editors

Wrangle

Cloud Storage Default - lucas-abbade_mvp3@teamstats.csv

teamstats.csv

Columns: 14 | Rows: 1000

Data

Insights

Apply

More

Int	String	String	Int	Int	Int	Int	Int	Int	Int	Int	Int	String
season	date	location	goals	shots	shotsOnTarget	deep	fouls	corners	yellowCards	redCards	result	
2015	2015-08-08 15:45:00	h	1	9	1	4	12	1	2	0	W	
2015	2015-08-08 15:45:00	a	0	9	4	10	12	2	3	0	L	
2015	2015-08-08 18:00:00	h	0	11	2	11	13	6	3	0	L	
2015	2015-08-08 18:00:00	a	1	7	3	2	13	3	4	0	W	
2015	2015-08-08 18:00:00	h	2	10	5	5	7	8	1	0	D	

Columns (14)

Transformation steps (5)

Transformations

1 drop xGoals.ppd

2 set-type: gameId Integer

3 set-type: teamID Integer

4 set-type: season Integer

5 find-and-replace: yellowCards s/NA/0/g

Figura 30

Já no BigQuery, para esta tabela foi criada uma chave primária a partir da combinação de gameId e teamID, uma vez que cada equipe é única de uma mesma partida de futebol. Além disso, formatou-se a coluna date como datetime. Estas alterações foram implementadas no mesmo modelo de três etapas aplicado nas demais que também tiveram modificações no BigQuery.

```
-- teamstats
create or replace table `sturdy-web-399922.mvp3_football_dw.teamstats_2` as
select
  CONCAT(gameID, '_', teamID) as teamstatID
, gameId
, teamID
, season
, datetime(date) as date
, location
, goals
, shots
, shotsOnTarget
, deep
, fouls
, corners
, yellowCards
, redCards
, result
from `sturdy-web-399922.mvp3_football_dw.teamstats`;

drop table `sturdy-web-399922.mvp3_football_dw.teamstats`;

CREATE TABLE `sturdy-web-399922.mvp3_football_dw.teamstats` (
  teamstatID STRING NOT NULL,
  gameId INT64 NOT NULL,
  teamID INT64 NOT NULL,
  season INT64 NOT NULL,
  date DATETIME NOT NULL,
  location STRING NOT NULL,
  goals INT64 NOT NULL,
  shots INT64 NOT NULL,
  shotsOnTarget INT64 NOT NULL,
  deep INT64 NOT NULL,
  fouls INT64 NOT NULL,
  corners INT64 NOT NULL,
  yellowCards INT64 NOT NULL,
  redCards INT64 NOT NULL,
  result STRING NOT NULL
);

insert into `sturdy-web-399922.mvp3_football_dw.teamstats`
SELECT * FROM `sturdy-web-399922.mvp3_football_dw.teamstats_2`;

drop table `sturdy-web-399922.mvp3_football_dw.teamstats_2`;
```

Figura 31

5. Análise

Para realizar a análise dos dados carregados na data warehouse, foram criadas 4 views em SQL no BigQuery com os joins e agregações necessárias para extrair as informações que respondem as perguntas levantadas no objetivo do projeto. Para gerar uma análise mais dinâmica, visual e interativa, estas 4 views foram carregadas no Power BI para a construção de gráficos e tabelas.

A view abaixo monta uma tabela de classificação de cada uma das ligas em cada uma das temporadas trazendo a colocação de cada time no campeonato, sua pontuação, saldo de gols, número de vitórias, cartões dentre outras informações.

```
-- CRIAR VIEW COM AS TABELAS DAS LIGAS POR TEMPORADA
create or replace view `sturdy-web-399922.mvp3_football_dw.tabela_classificacao_ligas` as
with tabela as (
SELECT
  concat(cast(g.season as string), "-", substr(cast((g.season+1) as string),3,2)) as temporada
, l.name as liga
, t.name as equipe
, sum(case when ts.result = 'W' then 3 when ts.result = 'D' then 1 else 0 end) as pts
, sum(ts.goals)-sum(gc.goals) as sg
, sum(ts.goals) as gm
, sum(gc.goals) as gc
, sum(case when ts.result = 'W' then 1 else 0 end) as vit
, sum(case when ts.result = 'D' then 1 else 0 end) as emp
, sum(case when ts.result = 'L' then 1 else 0 end) as der
, sum(ts.redCards) as car_verm
, sum(ts.yellowCards) as car_am
, sum(ts.fouls) as faltas
, sum(ts.corners) as escanteios
, sum(ts.shots) as chutes
, sum(ts.shotsOnTarget) as chutes_no_gol
, sum(case when ts.result = 'W' and ts.location = 'h' then 1 else 0 end) as vit_casa
, sum(case when ts.result = 'W' and ts.location = 'a' then 1 else 0 end) as vit_fora
, sum(case when ts.result = 'D' and ts.location = 'h' then 1 else 0 end) as emp_casa
, sum(case when ts.result = 'D' and ts.location = 'a' then 1 else 0 end) as emp_fora
, sum(case when ts.result = 'L' and ts.location = 'h' then 1 else 0 end) as der_casa
, sum(case when ts.result = 'L' and ts.location = 'a' then 1 else 0 end) as der_fora

from `sturdy-web-399922.mvp3_football_dw.teamstats` ts
inner join `sturdy-web-399922.mvp3_football_dw.games` g on g.gameID = ts.gameID
inner join `sturdy-web-399922.mvp3_football_dw.leagues` l on g.leagueID = l.leagueID
inner join `sturdy-web-399922.mvp3_football_dw.teams` t on t.teamID = ts.teamID
inner join `sturdy-web-399922.mvp3_football_dw.teamstats` gc on ts.gameID = gc.gameID and ts.teamID <> gc.teamID

group by 1,2,3
order by 1,2, 4 desc, 5 desc, 6 desc , 8 desc)

select
  tabela.temporada,tabela.liga
, row_number() over(partition by temporada, liga order by pts desc, sg desc, gm desc, vit desc, car_verm, car_am) as colocacao
, * except(temporada,liga)
from tabela;
```

Figura 32

Posteriormente, foi consruída a view abaixo, na qual agregam-se o número de chutes de cada jogador por temporada, liga tipo e situação do chute, resultado do chute (0 = não gol, 1 = gol, -1 = gol contra), dentre outros.

```
-- CRIAR VIEW SOBRE OS CHUTES
create or replace view `sturdy-web-399922.mvp3_football_dw.chutes` as
select
  concat(cast(g.season as string), "-", substr(cast((g.season+1) as string),3,2)) as temporada
, l.name as liga
, ch.name as chute_de
, a.name as assistencia_de
, CASE
    WHEN minute BETWEEN 0 AND 14 THEN '0-14'
    WHEN minute BETWEEN 15 AND 29 THEN '15-29'
    WHEN minute BETWEEN 30 AND 44 THEN '30-44'
    WHEN minute BETWEEN 45 AND 59 THEN '45-59'
    WHEN minute BETWEEN 60 AND 74 THEN '60-74'
    WHEN minute BETWEEN 75 AND 89 THEN '75-89'
    ELSE '90+'
  END AS periodo_jogo
, s.lastAction as ultima_acao
, s.situation as situacao
, s.shotType as tipo_chute
, case when shotResult = "OwnGoal" then -1 when shotResult = "Goal" then 1 else 0 end as resultado_chute
, count(distinct shotID) as n_chutes
from `sturdy-web-399922.mvp3_football_dw.shots` s
inner join `sturdy-web-399922.mvp3_football_dw.games` g on g.gameID = s.gameID
inner join `sturdy-web-399922.mvp3_football_dw.leagues` l on g.leagueID = l.leagueID
inner join `sturdy-web-399922.mvp3_football_dw.players` ch on ch.playerID = s.shooterID
left join `sturdy-web-399922.mvp3_football_dw.players` a on a.playerID = s.assisterID

group by 1,2,3,4,5,6,7,8,9
;
```

Figura 33

Em seguida, montou-se uma view para extrair as informações de cada jogador por liga e temporada, como por exemplo o número de chutes, assistências, gols, cartões, dentre outras.

```
-- CRIAR VIEW SOBRE PARTICIPAÇÕES DOS JOGADORES
create or replace view `sturdy-web-399922.mvp3_football_dw.participacoes` as
select
  concat(cast(g.season as string), "-", substr(cast((g.season+1) as string),3,2)) as temporada
, l.name as liga
, p.name as jogador
, count(distinct appearanceID) as jogos
, sum(a.keyPasses) as passes_importantes
, sum(a.shots) as chutes
, sum(a.goals) as gols
, sum(a.assists) as assistencias
, sum(a.goals) + sum(a.assists) as participacao_gol
, sum(a.ownGoals) as gols_contra
, sum(a.yellowCard) as cartao_am
, sum(a.redCard) as cartao_ver
from `sturdy-web-399922.mvp3_football_dw.appearances` a
inner join `sturdy-web-399922.mvp3_football_dw.games` g on g.gameID = a.gameID
inner join `sturdy-web-399922.mvp3_football_dw.leagues` l on l.leagueID = a.leagueID
inner join `sturdy-web-399922.mvp3_football_dw.players` p on p.playerID = a.playerID
group by 1,2,3
order by 1,2,3
;
```

Figura 34

Por fim, foi construída a view para extrair o número de gols por data e liga.

```
-- evolução de gols com o tempo
create or replace view `sturdy-web-399922.mvp3_football_dw.gols_tempo` as
select
| l.name as liga
, date(g.date) as date
, g.homeGoals + g.awayGoals as gols
from `sturdy-web-399922.mvp3_football_dw.games` g
inner join `sturdy-web-399922.mvp3_football_dw.leagues` l on g.leagueID = l.leagueID
order by 1,2
;
```

Figura 35

Após isso, as quatro views foram carregadas no Power BI conforme as imagens abaixo.

```
= Value.NativeQuery(
    GoogleBigQuery.Database()[[Name="sturdy-web-399922"]][Data],
    "select * from `sturdy-web-399922.mvp3_football_dw.tabela_classificacao_ligas`"
, null, [EnableFolding=true])
```

Figura 36

```
= Value.NativeQuery(
    GoogleBigQuery.Database()[[Name="sturdy-web-399922"]][Data],
    "select * from `sturdy-web-399922.mvp3_football_dw.chutes`"
, null, [EnableFolding=true])
```

Figura 37

```
= Value.NativeQuery(
    GoogleBigQuery.Database()[[Name="sturdy-web-399922"]][Data],
    "select * from `sturdy-web-399922.mvp3_football_dw.participacoes`"
, null, [EnableFolding=true])
```

Figura 38

```
= Value.NativeQuery(
    GoogleBigQuery.Database()[[Name="sturdy-web-399922"]][Data],
    "select * from `sturdy-web-399922.mvp3_football_dw.gols_tempo`"
, null, [EnableFolding=true])
```

Figura 39

No Power BI não houve a necessidade de construir nenhuma medida nem tabela extra. As views em SQL já vieram bem completas, necessitando apenas colocar cada campo nos visuais construídos.

a. Qualidade de dados

Antes de solucionar o problema levantado no projeto, é fundamental analisar a qualidade dos dados carregados na Data Warehouse. Sendo assim, foi realizado um processo de validação dos dados de cada uma das tabelas.

a.1. Tabelas dimensão

As tabelas leagues e teams são tabelas pequenas com 5 e 146 linhas cada. Foi realizada uma validação manual nas mesmas e não foi verificada nenhuma divergência com relação ao nome das ligas nem dos times.

Já na tabela players, foram encontrados dois tipos de problemas. Todos os jogadores que possuíam apóstrofe no nome estavam com o código `'` no lugar da apóstrofe, o que já estava assim no próprio csv. Isto pode ser observado na figura 40. Além disso, 13 jogadores apresentaram erro em alguns caracteres especiais durante o processo de carga, aparecendo o símbolo de erro ❖ no lugar do caractere mesmo o processo de carga ter sido realizado com sucesso e utilizando o encoding correto UTF-8.

```
17 SELECT * FROM `sturdy-web-399922.mvp3_football_dw.players`
18 where name like "%&#039;%"
19 ;
```

Query results

JOB INFORMATION	RESULTS	JSON	EXECUTION DETAIL
dw	playerID	name	
1	731	Dame N'Doye	
2	8756	Dara O'Shea	
3	796	Gary O'Neil	
4	3587	Guy N'Gosso	
5	729	John O'Shea	
6	732	Yann M'Vila	
7	4845	Deme N'Diaye	
8	472	Eunan O'Kane	
9	918	Joey O'Brien	
10	9018	M'Bala Nzola	
11	1126	M'Baye Niang	
12	5688	Rais M'bolhi	
13	3814	Samuel Eto'o	

Figura 40

```
17 SELECT * FROM `sturdy-web-399922.mvp3_football_dw.players`
18 where name like '%❖%'
19 ;
```

Query results

JOB INFORMATION	RESULTS	JSON	EXECUTION DETAILS
Row	playerID	name	
1	2294	Se❖é	
2	5136	Estupi❖án	
3	6149	Unai N❖ñez	
4	2403	❖ñigo Lekue	
5	5058	Carles Ale❖á	
6	2372	Rubén Y❖ñez	
7	2266	Saúl ❖íguez	
8	4071	❖ñigo López	
9	4110	Aarón ❖íguez	
10	2118	Paco Monta❖és	
11	2089	Robert Ib❖ñez	
12	5233	Andersson Ord❖ñez	
13	4384	Jussi J❖äskeläinen	

Figura 41

Para ajustar a tabela players foi preciso realizar o primeiro UPDATE na query da figura 42 para incluir as apóstrofes. Para ajustar o caractere especial dos 13 jogadores com erro foi preciso ver um a um qual era o caractere correto e rodar o segundo UPDATE da query da figura 42

```
UPDATE `sturdy-web-399922.mvp3_football_dw.players`
SET name = REPLACE(name, "&#039;", "'")
where name like "%&#039;%";

UPDATE `sturdy-web-399922.mvp3_football_dw.players`
SET name = CASE
  WHEN playerID = 2294 THEN 'Señé'
  WHEN playerID = 5136 THEN 'Estupiñán'
  WHEN playerID = 6149 THEN 'Unai Núñez'
  WHEN playerID = 2403 THEN 'Íñigo Lekue'
  WHEN playerID = 5058 THEN 'Carles Aleñá'
  WHEN playerID = 2372 THEN 'Rubén Yáñez'
  WHEN playerID = 2266 THEN 'Saúl Níguez'
  WHEN playerID = 4071 THEN 'Íñigo López'
  WHEN playerID = 4110 THEN 'Aarón Níguez'
  WHEN playerID = 2118 THEN 'Paco Montañés'
  WHEN playerID = 2089 THEN 'Robert Ibáñez'
  WHEN playerID = 5233 THEN 'Andersson Ordóñez'
  WHEN playerID = 4384 THEN 'Jussi Jäskeläinen'
  ELSE name -- Keep the existing name for all other playerID values
END
WHERE playerID IN (2294, 5136, 6149, 2403, 5058, 2372, 2266, 4071, 4110, 2118, 2089, 5233, 4384);
```

Figura 42

a.2. Tabelas fato

Foram realizadas validações dos números totais das 4 tabelas fato por amostragem. As tabelas games, teamstats e appearances não apresentaram divergência em relação à números totais encontrados em diversos portais na internet. Abaixo pode-se comparar os números do Lionel Messi pelo Barcelona na Wikipedia (à esquerda) e os números da Data Warehouse (à direita). Todos os números são iguais, com exceção do número de assistências da temporada 2019-20, no qual o número do banco de dados consta uma assistência a menos. Isto não deve ser considerado um erro, pois o conceito de assistência envolve um certo nível de subjetividade e fontes diferentes podem apresentar números ligeiramente diferentes neste quesito. Este exemplo validação das figuras 43 e 44 serve para verificar a qualidade dos dados das tabelas games e appearances, uma vez que estes dados são oriundos da view ``sturdy-web-399922.mvp3_football_dw.participacoes``.

Equipe	Temporada	Campeonato nacional		
		Jogos	Gols	Assist.
Barcelona	2004-05	7	1	0
	2005-06	17	6	2
	2006-07	26	14	2
	2007-08	28	10	12
	2008-09	31	23	11
	2009-10	35	34	10
	2010-11	33	31	18
	2011-12	37	50	16
	2012-13	32	46	12
	2013-14	31	28	11
	2014-15	38	43	18
	2015-16	33	26	16
	2016-17	34	37	9
	2017-18	36	34	12
	2018-19	34	36	13
	2019-20	33	25	21
	2020-21	35	30	9

Figura 43

Gols e Assistências do Linonel Messi				
temporada	Jogos	Gols	Assit.	
2014-15		38	43	18
2015-16		33	26	16
2016-17		34	37	9
2017-18		36	34	12
2018-19		34	36	13
2019-20		33	25	20
2020-21		35	30	9

Figura 44

Em seguida, também realizou-se validação por amostragem para a view ``sturdy-web-399922.mvp3_football_dw.tabela_classificacao_ligas`` (tabelas games e teamstats). Foram montadas tabelas de classificações para todas temporadas de todas as ligas e comparadas com os seus resultados finais oficiais e todas bateram em todos os quesitos. Segue abaixo nas figuras 46 e 47 um exemplo de validação da liga Serie A para a temporada 2020-21.

Temporada 2020-21									
Clube	Pts	PJ	VIT	E	DER	GM	GC	SG	
1 Inter	91	38	28	7	3	89	35	54	
2 Milan	79	38	24	7	7	74	41	33	
3 Atalanta	78	38	23	9	6	90	47	43	
4 Juventus	78	38	23	9	6	77	38	39	
5 Napoli	77	38	24	5	9	86	41	45	
6 Lazio	68	38	21	5	12	61	55	6	
7 Roma	62	38	18	8	12	68	58	10	
8 Sassuolo	62	38	17	11	10	64	56	8	
9 Sampdoria	52	38	15	7	16	52	54	-2	
10 Verona	45	38	11	12	15	46	48	-2	
11 Genoa	42	38	10	12	16	47	58	-11	
12 Bologna	41	38	10	11	17	51	65	-14	
13 Fiorentina	40	38	9	13	16	47	59	-12	
14 Udinese	40	38	10	10	18	42	58	-16	
15 Spezia	39	38	9	12	17	52	72	-20	
16 Cagliari	37	38	9	10	19	43	59	-16	
17 Torino	37	38	7	16	15	50	69	-19	
18 Benevento	33	38	7	12	19	40	75	-35	
19 Crotone	23	38	6	5	27	45	92	-47	
20 Parma	20	38	3	11	24	39	83	-44	

Figura 46

Classificação das Ligas por temporada

temporada

2020-21

liga

Serie A

#	equipe	pts	gm	gc	sg	vit	emp	der
1	Inter	91	89	35	54	28	7	3
2	AC Milan	79	74	41	33	24	7	7
3	Atalanta	78	90	47	43	23	9	6
4	Juventus	78	77	38	39	23	9	6
5	Napoli	77	86	41	45	24	5	9
6	Lazio	68	61	55	6	21	5	12
7	Roma	63	68	55	13	18	9	11
8	Sassuolo	62	64	56	8	17	11	10
9	Sampdoria	52	52	54	-2	15	7	16
10	Verona	43	43	48	-5	10	13	15
11	Genoa	42	47	58	-11	10	12	16
12	Bologna	41	51	65	-14	10	11	17
13	Fiorentina	40	47	59	-12	9	13	16
14	Udinese	40	42	58	-16	10	10	18
15	Spezia	39	52	72	-20	9	12	17
16	Cagliari	37	43	59	-16	9	10	19
17	Torino	37	50	69	-19	7	16	15
18	Benevento	33	40	75	-35	7	12	19
19	Crotone	23	45	92	-47	6	5	27
20	Parma Calcio 1913	20	39	83	-44	3	11	24

Figura 47

Além dessas validações realizadas, é importante destacar alguns pontos observados a respeito da Data Warehouse construída que afetam negativamente sua qualidade dos dados. Nas tabelas shots e appearances não há a chave teamID só gameId, de modo que não se pode construir a chave teamstatID, o que impede que os dados destas duas tabelas possam ser agrupados no nível de time. Para solucionar este problema seria fundamental a inclusão da variável teamID em ambas as tabelas.

Por fim, no que diz respeito a tabela shots foram encontradas algumas inconsistências. Certos números gerados a partir dela apresentavam valores maiores do que sua correspondência na tabela appearances. Segue abaixo um exemplo que evidenciam o problema. O jogador Luis Suárez apresentava o dobro do número de assistências quando calculadas via tabela shots (Figura 48).

Assistências (tabela appearances)					
temporada	Cristiano Ronaldo	Lionel Messi	Luis Suárez	Neymar	Total
2014-15	16	18	14	7	55
2015-16	11	16	16	12	55
2016-17	6	9	13	11	39
2017-18	5	12	12	13	42
2018-19	8	13	6	7	34
2019-20	5	20	8	6	39
2020-21	3	9	4	5	21

Assistências (tabela shots)					
temporada	Cristiano Ronaldo	Lionel Messi	Luis Suárez	Neymar	Total
2014-15	16	18	28	7	69
2015-16	11	16	32	12	71
2016-17	6	9	26	11	52
2017-18	5	12	24	13	54
2018-19	8	13	12	7	40
2019-20	5	20	16	6	47
2020-21	3	9	8	5	25

Figura 48

Como os demais jogadores apresentavam os mesmos valores, a suspeita foi que houvesse duplicidades para certos jogadores. Ao resultado das consultas abaixo comprovavam a suspeita.

```
1 SELECT * FROM `sturdy-web-399922.mvp3_football_dw.players`
2 where name like '%Luis Su%'
```

Query results

JOB INFORMATION		RESULTS	JSON	EXECUTION DETAIL
Row	playerID	name		
1	2098	Luis Suárez		
2	8978	Luis Suárez		

Figura 49

```
1 SELECT * FROM `sturdy-web-399922.mvp3_football_dw.shots`
2 WHERE assisterID IN (2098, 8978)
3 ORDER BY gameID
```

Press Alt+F1 for Accessibility Option

Query results

SAVE RESULTS

EXPLORE DATA



JOB INFORMATION		RESULTS	JSON	EXECUTION DETAILS	CHART	PREVIEW	EXECUTION GRAPH
Row	shotID	gameID	shooterID	assisterID	minute	situation	
1	56413	1411	2095	2098	8	OpenPlay	
2	56414	1411	2095	8978	8	OpenPlay	
3	57820	1422	2097	8978	76	OpenPlay	
4	57819	1422	2097	2098	76	OpenPlay	

Figura 50

Desse modo, foi inserida uma subquery com distinct (após substituir os ID pelos devidos nomes dos jogadores) dentro do from na query que gera a view de chutes para retirar as duplicidades:

```
-- CRIAR VIEW SOBRE OS CHUTES -- RETIRAR DUPLICIDADES
create or replace view `sturdy-web-399922.mvp3_football_dw.chutes` as
select
concat(cast(season as string), "-", substr(cast((season+1) as string),3,2)) as temporada
,liga
,chute_de
,assistencia_de
,CASE
WHEN minute BETWEEN 0 AND 14 THEN '0-14'
WHEN minute BETWEEN 15 AND 29 THEN '15-29'
WHEN minute BETWEEN 30 AND 44 THEN '30-44'
WHEN minute BETWEEN 45 AND 59 THEN '45-59'
WHEN minute BETWEEN 60 AND 74 THEN '60-74'
WHEN minute BETWEEN 75 AND 89 THEN '75-89'
ELSE '90+'
END AS periodo_jogo
,lastAction as ultima_acao
,situation as situacao
,shotType as tipo_chute
,case when shotResult = "OwnGoal" then -1 when shotResult = "Goal" then 1 else 0 end as resultado_chute
,count(distinct shotID) as n_chutes
from
(select row_number() over () as shotID
,* FROM
(select distinct
s.gameID
,g.season
,l.name as liga
,ch.name as chute_de
,a.name as assistencia_de
,s.* except (shotID, gameID, shooterID, assisterID)
from `sturdy-web-399922.mvp3_football_dw.shots` s
inner join `sturdy-web-399922.mvp3_football_dw.games` g on g.gameID = s.gameID
inner join `sturdy-web-399922.mvp3_football_dw.leagues` l on g.leagueID = l.leagueID
inner join `sturdy-web-399922.mvp3_football_dw.players` ch on ch.playerID = s.shooterID
left join `sturdy-web-399922.mvp3_football_dw.players` a on a.playerID = s.assisterID)
)
group by 1,2,3,4,5,6,7,8,9
;
```

Figura 51

Após reprocessar a view, os valores das duas tabelas (comparação da Figura 48) passaram a ficarem totalmente iguais, o que indica a correção dos valores da tabela shots.

b. Solução do problema

Nesta etapa do projeto serão utilizadas tabelas e visuais do Power BI, alimentado pelas 4 views criadas, para responder aos questionamentos levantados no objetivo do projeto.

b.1. Quais os times que ganharam mais títulos?

liga	equipe	2014-15	2015-16	2016-17	2017-18	2018-19	2019-20	2020-21	Total
Bundesliga	Bayern Munich	1	1	1	1	1	1	1	7
La Liga	Barcelona	1	1		1	1			4
	Real Madrid			1			1		2
	Atletico Madrid							1	1
Ligue 1	Paris Saint Germain	1	1		1	1	1		5
	Lille							1	1
	Monaco			1					1
Premier League	Manchester City				1	1		1	3
	Chelsea	1		1					2
	Leicester		1						1
	Liverpool						1		1
Serie A	Juventus	1	1	1	1	1	1		6
	Inter							1	1

Figura 52

Bayern Munich (7) e Juventus (6) foram os times que ganharam mais títulos no período. A equipe alemã ganhou todas as temporadas de sua liga nas 7 temporadas presentes no banco de dados. Ainda cabe um destaque para o campeonato inglês, o qual teve 4 campeões diferentes no período, indicando haver um nível de acirramento maior do que os demais.

b.2. Quais times se mantiveram nas primeiras posições por mais tempo?

Top 4 Equipes por temporada																														
	temporada	2014-15				2015-16				2016-17				2017-18				2018-19				2019-20				2020-21				Total
liga	equipe	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	Total				
Bundesliga	Bayern Munich	1				1				1				1				1				1				7				
	Borussia Dortmund						1				1				1		1				1				1	6				
	RasenBallsport Leipzig								1											1					1	4				
	Bayer Leverkusen					1			1									1								3				
	Borussia M.Gladbach			1					1													1				3				
	Hoffenheim										1			1												2				
	Wolfsburg		1																						1	2				
	Schalke 04													1												1				
La Liga	Atletico Madrid			1				1			1				1				1			1				7				
	Barcelona	1				1			1			1			1					1				1		7				
	Real Madrid		1				1		1				1				1	1					1			7				
	Sevilla										1										1				1	3				
	Valencia				1									1				1								3				
	Villarreal							1																		1				
Ligue 1	Paris Saint Germain	1				1			1		1				1			1					1			7				
	Lyon		1				1				1		1			1									1	6				
	Monaco			1				1				1												1		5				
	Lille															1						1	1			3				
	Marseille				1									1					1							3				
	Nice							1		1																2				
	Rennes																				1					1				
	Saint-Etienne																	1								1				
Premier League	Manchester City		1					1		1	1				1				1			1				7				
	Chelsea	1							1							1					1				1	5				
	Liverpool										1		1		1		1						1			5				
	Manchester United				1							1								1			1			4				
	Tottenham						1		1				1					1								4				
	Arsenal			1			1																			2				
	Leicester					1																				1				
Serie A	Juventus	1				1					1				1			1							1	7				
	Atalanta										1						1			1			1			4				
	Inter							1										1		1			1			4				
	Napoli					1				1		1			1											4				
	Roma		1				1		1			1														4				
	Lazio			1										1								1				3				
	AC Milan																							1		1				
	Fiorentina				1																					1				

Figura 53

Ao observar a figura 53, destaca-se que Bayern Munich, Atletico Madrid, Barcelona, Real Madrid, PSG, Manchester City e Juventus se estiveram nas 4 primeiras posições em todas as temporadas, garantindo classificação para a Champions League. Ainda é importante destacar que a Serie A e Ligue 1 tiveram 8 equipes diferentes entre os 4 primeiros ao longo do período, demonstrando alto nível de acirramento para a classificação para a Champions League, o que indica a forte competitividade entre as segundas potências do campeonato.

b.3. Quais os times/jogadores fizeram mais gols?

Equipes com maior nº de gols								
equipe	2014-15	2015-16	2016-17	2017-18	2018-19	2019-20	2020-21	Total
Barcelona	110	112	116	99	90	86	85	698
Paris Saint Germain	83	102	83	108	105	75	86	642
Bayern Munich	80	80	89	92	88	100	99	628
Real Madrid	118	110	106	94	63	70	67	628
Manchester City	83	71	80	106	95	102	83	620
Napoli	70	80	94	77	74	61	86	542
Juventus	72	75	77	86	70	76	77	533
Liverpool	52	63	78	84	89	85	68	519
Borussia Dortmund	47	82	72	64	81	84	75	505
Roma	54	83	90	61	66	77	68	499

Figura 54

Estas são as 10 equipes com maior número de gols no período. Barcelona se destaca por ter 9% a mais de gols que o segundo colocado, liderando isoladamente este quesito. É interessante traçar um paralelo com a figura 52, que apesar de ser líder em quesito gols, o time ganhou 4 dos 7 títulos disponíveis no período.

Jogadores com maior nº de gols								
jogador	2014-15	2015-16	2016-17	2017-18	2018-19	2019-20	2020-21	Total
Lionel Messi	43	26	37	34	36	25	30	231
Cristiano Ronaldo	48	35	25	26	21	31	29	215
Robert Lewandowski	17	30	30	29	22	34	41	203
Luis Suárez	16	40	29	25	21	16	26	173
Harry Kane	21	25	29	30	17	18	23	163
Pierre-Emerick Aubameyang	16	25	31	23	22	22	10	149
Ciro Immobile	3	7	23	29	15	36	20	133
Edinson Cavani	18	19	35	28	18	4	10	132
Sergio Agüero	26	24	20	21	21	16	4	132
Mohamed Salah	6	14	15	32	22	19	22	130

Figura 55

Lionel Messi liderou o quesito gol entre os jogadores, seguido por Cristiano Ronaldo, Robert Lewandowski, Luis Suárez e Harry Kane. É interessante ressaltar que 2 dos 4 primeiros são jogadores do Barcelona no período, o time com melhor ataque.

b.4. Quais os jogadores que têm mais assistências?

Jogadores com maior nº de assistências								
jogador	2014-15	2015-16	2016-17	2017-18	2018-19	2019-20	2020-21	Total
Kevin De Bruyne	20	9	18	16	2	20	12	97
Lionel Messi	18	16	9	12	13	20	9	97
Thomas Müller	10	5	12	13	9	20	18	87
Ángel Di María	10	18	7	6	11	14	9	75
Luis Suárez	14	16	13	12	6	8	4	73
Dimitri Payet	16	12	9	13	6	4	10	70
Neymar	7	12	11	13	7	6	5	61
Alejandro Gomez	2	7	10	10	11	16	3	59
David Silva	7	11	7	11	8	10	5	59
Christian Eriksen	2	13	15	10	12	4	0	56

Figura 56

Kevin de Bruyne e Lionel Messi lideram este quesito com 97 assistências, 10 a mais que o segundo colocado Thomas Muller. É impressionante o fato que eles possuem 73% mais assistências que o décimo colocado no quesito. Lionel Messi lidera tanto em gols quanto em assistências.

b.5. Quais são os jogadores com maior participação em gol (assistência/gol)?

Jogadores com maior nº de participações em gol								
jogador	2014-15	2015-16	2016-17	2017-18	2018-19	2019-20	2020-21	Total
Lionel Messi	61	42	46	46	49	45	39	328
Cristiano Ronaldo	64	46	31	31	29	36	32	269
Luis Suárez	30	56	42	37	27	24	30	246
Robert Lewandowski	22	32	35	31	29	38	48	235
Harry Kane	25	26	36	32	21	20	37	197
Mohamed Salah	9	20	26	42	30	29	27	183
Pierre-Emerick Aubameyang	22	30	33	30	27	25	13	180
Karim Benzema	25	31	16	15	27	29	32	175
Neymar	29	36	24	32	22	18	14	175
Ciro Immobile	4	9	26	38	21	45	26	169

Figura 57

Lionel Messi lidera novamente, com 59 participações a mais que o segundo colocado, Cristiano Ronaldo, cujo número de participações é próximo à dos 3 e 4 colocados. É notável a liderança isolada de Lionel Messi no quesito.

b.6. Quais os times que jogam melhor dentro e fora de casa?

Equipes com maior nº de vitórias em casa								
equipe	2014-15	2015-16	2016-17	2017-18	2018-19	2019-20	2020-21	Total
Juventus	16	16	18	16	15	16	14	111
Barcelona	16	16	15	16	15	16	11	105
Paris Saint Germain	15	15	13	17	17	12	13	102
Manchester City	14	12	11	16	18	15	13	99
Real Madrid	16	16	14	12	13	15	13	99
Atletico Madrid	14	15	14	12	15	12	15	97
Bayern Munich	14	15	13	14	13	13	13	95
Napoli	11	16	13	14	13	10	12	89
Sevilla	13	14	14	11	12	10	14	88
Liverpool	10	8	12	12	17	18	10	87

Equipes com maior nº de vitórias fora de casa								
equipe	2014-15	2015-16	2016-17	2017-18	2018-19	2019-20	2020-21	Total
Barcelona	14	13	13	12	11	9	13	85
Paris Saint Germain	9	15	14	12	12	10	13	85
Bayern Munich	11	13	12	13	11	13	11	84
Manchester City	10	7	12	16	14	11	14	84
Real Madrid	14	12	15	10	8	11	12	82
Juventus	10	13	11	14	13	10	9	80
Napoli	7	9	13	14	11	8	12	74
Liverpool	8	8	10	9	13	14	10	72
Chelsea	11	7	13	10	9	9	10	69
Atletico Madrid	9	13	9	11	7	6	11	66

Figura 58

Juventus é o melhor mandante, seguido do Barcelona. Os melhores visitantes são Barcelona e PSG, seguidos com uma vitória a menos por Bayern Munich e Manchester City. É notável que o Barcelona é o segundo melhor mandante e o melhor visitante. Ainda é impressionante destacar a diferença de aproveitamento da Juventus e do Atletico Madrid dentro e fora de casa, há uma diferença de 31 vitórias para ambos os times.

b.7. Quais os times/jogadores com melhor aproveitamento de chutes?

Não é possível chegar no número aproveitamento de chutes dos times, pois não dá para agrupar a tabela appearances por teamID, de modo que não se consegue trazer o número de chutes de um time para poder calcular o aproveitamento de chutes (gols/nº de chutes). Sendo assim, só foi possível trazer o aproveitamento dos jogadores. Foram selecionados só 100 jogadores com maior número de chutes para fazer este cálculo, a fim de evitar casos em que um jogador tem poucos chutes e acertados, por exemplo um jogador que chutou 3x e acertou todas, tendo um aproveitamento de 100%.

Jogadores com maior aproveitamento de chutes

jogador	2014-15	2015-16	2016-17	2017-18	2018-19	2019-20	2020-21	Total
Alexandre Lacazette	28.1%	22.1%	32.9%	20.9%	16.0%	19.2%	28.9%	24.2%
Kylian Mbappe-Lottin		20.0%	28.3%	16.5%	26.4%	20.5%	26.2%	23.6%
Edinson Cavani	16.2%	22.1%	24.5%	26.4%	34.6%	14.3%	29.4%	23.6%
Mauro Icardi	18.2%	28.1%	22.0%	29.6%	14.9%	33.3%	17.5%	22.6%
Iago Aspas	13.3%	17.5%	24.4%	23.4%	28.6%	16.7%	27.5%	22.2%
Pierre-Emerick Aubameyang	15.7%	21.2%	26.7%	25.6%	23.4%	23.7%	17.5%	22.2%
Jamie Vardy	10.4%	20.9%	24.5%	28.6%	22.8%	25.8%	18.3%	22.0%
Wissam Ben Yedder	17.1%	18.3%	25.0%	18.0%	23.7%	28.1%	24.7%	21.8%
Luis Suárez	21.3%	29.2%	24.2%	20.7%	18.8%	20.3%	17.2%	21.8%
Robert Lewandowski	16.3%	19.7%	21.0%	22.8%	15.3%	24.6%	30.4%	21.5%

Figura 59

Destacam-se Alexandre Lacazette, Mbappe e Cavani. Estes jogadores obtiveram uma média de quase 1 gol marcado para cada 4 chutes.

b.8. Quais são os times/jogadores mais violentos (cartões amarelos/vermelhos e nº de faltas)?

Equipes com maior nº de faltas

equipe	2014-15	2015-16	2016-17	2017-18	2018-19	2019-20	2020-21	Total
Torino	543	656	587	565	602	601	577	4131
Genoa	634	632	623	560	545	563	517	4074
Atalanta	629	636	603	491	429	528	581	3897
Sevilla	604	557	581	562	536	522	492	3854
Fiorentina	537	536	535	547	542	538	522	3757
Getafe	541	531		674	641	705	630	3722
Eibar	569	593	515	493	515	538	494	3717
Lazio	678	594	543	435	495	457	503	3705
Sassuolo	587	571	518	483	510	499	496	3664
AC Milan	541	585	521	498	444	544	510	3643

Figura 60

Equipes com maior nº de cartões vermelhos

equipe	2014-15	2015-16	2016-17	2017-18	2018-19	2019-20	2020-21	Total
AC Milan	13	3	12	7	5	6	4	50
Genoa	6	9	8	5	7	10	2	47
Valencia	9	7	6	5	3	8	5	43
Bologna		7	11	6	6	7	4	41
Lazio	9	10	4	6	4	2	5	40
Marseille	6	9	1	3	8	3	10	40
Sassuolo	9	7	2	7	8	2	4	39
Atalanta	9	14	3	4	3	2	3	38
Celta Vigo	5	7	8	1	6	6	5	38
Monaco	5	4	3	4	7	10	5	38

Figura 61

Equipes com maior nº de cartões amarelos

equipe	2014-15	2015-16	2016-17	2017-18	2018-19	2019-20	2020-21	Total
Getafe	93	122		132	112	136	119	714
Sevilla	115	95	105	101	118	98	78	710
Celta Vigo	115	109	98	86	84	103	108	703
Atletico Madrid	109	88	80	96	103	97	98	671
Valencia	101	101	108	106	90	81	82	669
Lazio	104	94	90	77	90	101	107	663
Villarreal	94	96	89	117	118	78	66	658
Athletic Club	96	83	87	93	120	90	82	651
Genoa	102	82	89	76	96	107	88	640
Fiorentina	81	93	92	88	91	106	86	637

Figura 62

É notável que nos três quesitos para medir a violência de um jogo praticamente todas as equipes no top 10 são ou do campeonato italiano ou espanhol, o que deixa claro que estes são os campeonatos mais violentos. Em termos de faltas, 70% das equipes no top 10 são italianas e em cartões vermelhos 60%, evidenciando que esta liga é ainda mais violenta que a espanhola.

Para jogadores, não temos o número de faltas. Serão pontuados só cartões. Segue abaixo.

Jogadores com maior nº de cartões vermelhos								
jogador	2014-15	2015-16	2016-17	2017-18	2018-19	2019-20	2020-21	Total
Yannick Cahuzac	2	2	4	0	1		0	9
Facundo Roncaglia	2	0	1	0	1	2	1	7
Gabriel Paletta	0	2	5					7
Granit Xhaka	1	3	2	0	0	0	1	7
Nicolas Pallois	1	1	2	2	0	0	1	7
Afriyie Acquah	2	0	3	1	0			6
Danilo	0	2	1	0	1	2	0	6
Jacques-Alaixys Romao	1	2			1	2		6
Jeison Murillo	1	3	0	0	0	1	1	6
Mario Balotelli	0	0	3	1	1	1		6

Jogadores com maior nº de cartões amarelos								
jogador	2014-15	2015-16	2016-17	2017-18	2018-19	2019-20	2020-21	Total
Danilo	7	19	11	10	6	15	15	83
Raúl García	8	10	17	11	14	13	8	81
Dani García	10	15	9	14	13	11	8	80
Álvaro González	13	10	8	13	12	4	14	74
Damián Suárez	11	7		14	13	15	11	71
Recio	10	16	9	9	9	11	5	69
Felipe	2	13	9	6	17	12	8	67
Daniel Carvajal	12	6	11	11	10	11	5	66
Rubén Pérez	9	17	11	13	7	9		66
Tomás Rincón	8	9	4	12	15	10	8	66

Figura 63

Ao observar a tabela acima, percebe-se que praticamente todos dois top 10 são compostos por jogadores diferentes, o que evidencia que os cartões amarelos contêm a violência. Cahuzac teve 9 cartões vermelhos no período, com uma média de mais de um por ano. Roncaglia, Paletta, Chaka e Pallois têm uma média de um por ano.

b.9. Quais os times/jogadores fizeram mais gols contra?

Este dado só temos para jogadores. Como não é possível linkar os jogadores a seus clubes devido à ausência de teamID em appearances, não se pode levar esta informação de appearances para teamstats.

Jogadores com maior nº de gols contra								
jogador	2014-15	2015-16	2016-17	2017-18	2018-19	2019-20	2020-21	Total
Damien Da Silva	0	0	1	3	0	1	0	5
Federico Fernández	0	2	0	1	0	1	1	5
Lewis Dunk				4	0	0	1	5
Martin Hinteregger		2	0	0	1	1	1	5
Anthony Lopes	1	0	0	0	0	1	2	4
Ben Mee	0		2	0	2	0	0	4
Bostjan Cesar	2	0	0	2	0			4
Francesco Vicari				2	0	2		4
Luca Ceperelli	0		1	1	1	1	0	4
Niklas Süle	0	0	1	3	0	0	0	4

Figura 64

Damien Da Silva, Federico Fernández, Lewis Dunk e Martin Hinteregger se destacam por terem marcado 5 gols contra em todo o período, com uma média de 0,7 gols contra por ano.

b.10. O número de gols está aumentando ou caindo com o passar do tempo?

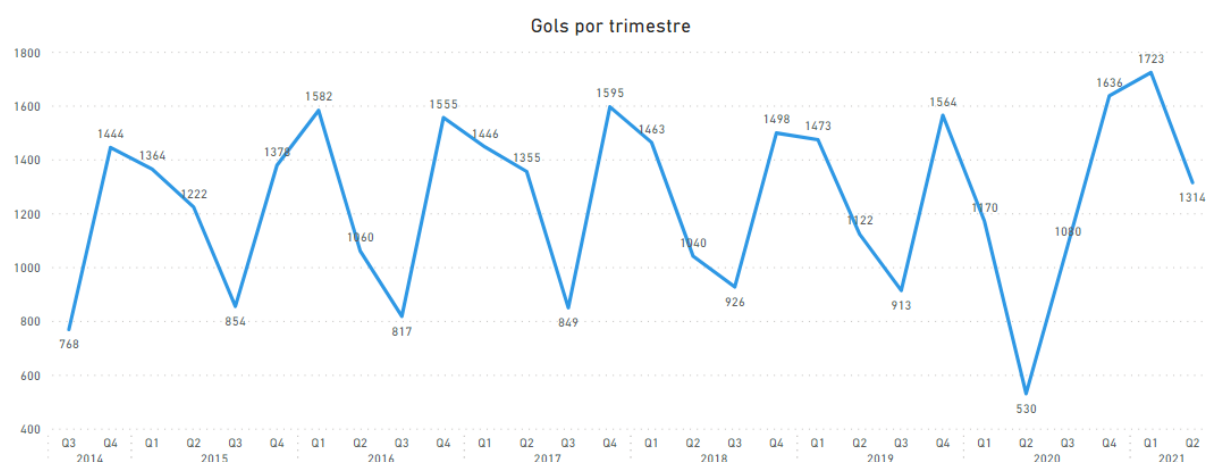


Figura 65

O número de gols não parece estar aumentando com o passar do tempo, parece girar em torno de uma média de 1.200 por trimestre, com sazonalidade de baixa nos Q2 e Q3 quando há o período de férias nas ligas. É importante destacar o pico de baixa em 2020 Q2 devido à paralisação dos campeonatos em virtude da pandemia de COVID-19. Posterior à isto, houveram dois picos de alta em 2020 Q4 e 2021 Q1 devido ao alto número de jogos ocorridos neste período para compensar a paralisação anterior.

b.11. Os gols acontecem mais em que período do jogo?

Período do jogo com maior nº de gols								
período_jogo	2014-15	2015-16	2016-17	2017-18	2018-19	2019-20	2020-21	Total
0-14	12.3%	12.8%	12.6%	12.3%	11.9%	12.2%	13.0%	12.4%
15-29	15.1%	14.4%	13.5%	15.6%	13.5%	14.2%	15.1%	14.5%
30-44	16.6%	15.3%	14.7%	14.0%	14.5%	15.0%	14.7%	14.9%
45-59	17.1%	18.0%	18.3%	18.2%	18.3%	19.5%	18.6%	18.3%
60-74	16.9%	16.7%	17.4%	17.5%	18.0%	17.0%	16.5%	17.1%
75-89	17.7%	18.2%	17.8%	17.2%	17.7%	16.6%	17.1%	17.5%
90+	4.4%	4.6%	5.7%	5.2%	6.1%	5.6%	5.0%	5.2%
Total	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%	100.0%

Figura 66

A maior parte dos gols ocorre no segundo tempo (58%), com destaque especial para os 15 primeiros minutos do segundo tempo e os 15 últimos minutos do segundo tempo.

b.12. Quais são os tipos de gols mais comuns?

Tipo de chute com maior nº de gols								
tipo_chute	2014-15	2015-16	2016-17	2017-18	2018-19	2019-20	2020-21	Total
Head	18%	17%	17%	17%	17%	15%	16%	17%
LeftFoot	28%	29%	30%	31%	29%	30%	31%	30%
OtherBodyPart	1%	1%	1%	1%	0%	0%	0%	1%
RightFoot	54%	54%	52%	52%	53%	54%	52%	53%
Total	100%	100%	100%	100%	100%	100%	100%	100%

Figura 67

A maior parte dos gols é com pé direito 53%, porém pé esquerdo e cabeça apresentam um percentual considerável, totalizando praticamente a outra metade do total de gols.

b.13. Quais são os tipos de jogadas que mais resultam em gols?

Situações com maior nº de gols								
situacao	2014-15	2015-16	2016-17	2017-18	2018-19	2019-20	2020-21	Total
OpenPlay	70%	71%	70%	71%	71%	71%	71%	71%
FromCorner	11%	12%	12%	12%	12%	11%	11%	12%
Penalty	8%	8%	9%	8%	9%	10%	11%	9%
SetPiece	7%	6%	6%	6%	5%	6%	5%	6%
DirectFreekick	3%	3%	3%	3%	3%	3%	2%	3%
Total	100%	100%	100%	100%	100%	100%	100%	100%

Figura 68

A grande maioria dos gols sai de bola rolando (71%), ficando em segundo lugar escanteio (12%) e terceiro pênalti (9%).

b.14. Quais times fazem mais gols de escanteio? E de cabeça?

Não é possível calcular este dado, uma vez que não há uma conexão entre os jogadores a seus clubes devido à ausência de teamID em shots, não permitindo a agregação desta informação por clube.

A partir das análises feitas em cima das perguntas específicas, é possível chegar a uma resposta para o objetivo principal do projeto, identificar os times e jogadores de maior destaque no cenário do futebol mundial.

Entre os clubes o principal destaque é o Bayern Munich, por ter se sagrado campeão em todo o período, possuir o terceiro melhor ataque, um alto número de vitórias tanto dentro como fora de casa e ter o terceiro maior artilheiro do período (Robert Lewandowski) e o terceiro maior assistente (Thomas Muller). Além disso, o clube alemão possui a segunda melhor defesa de entre todos os times que se sagraram campeões no período, conforme a figura abaixo.

Equipes com melhor defesa entre os times campeões no período								
equipe	2014-15	2015-16	2016-17	2017-18	2018-19	2019-20	2020-21	Total
Atletico Madrid	29	18	27	22	29	27	25	177
Bayern Munich	18	17	22	28	32	32	44	193
Paris Saint Germain	36	19	27	29	35	24	28	198
Juventus	24	20	27	24	30	43	38	206
Barcelona	21	29	37	29	36	38	38	228
Manchester City	38	41	39	27	23	35	32	235
Real Madrid	38	34	41	44	46	25	28	256
Lille	42	27	47	67	33	27	23	266
Inter	48	38	49	30	33	36	35	269
Liverpool	48	50	42	38	22	33	42	275
Chelsea	32	53	33	38	39	54	36	285
Monaco	26	50	31	45	57	44	42	295
Leicester	55	36	63	60	48	41	50	353

Figura 69

Por outro lado, também é destacável a Juventus, por ter sido campeã em 6 das 7 temporadas e por possuir uma força muito grande jogando em casa, chegando à uma marca de 111 vitórias em casa no período. E por fim, há de se destacar o Barcelona por possuir o melhor ataque e ter sido campeão 4x, além de possuir o primeiro e o terceiro jogadores com maior número de participações em gol (Lionel Messi e Luis Suárez).

Por fim, o maior destaque entre os jogadores fica para Lionel Messi, por ser o maior artilheiro, assistente e maior em participações em gol. Ainda se destacam Cristiano Ronaldo e Robert Lewandowski por serem o segundo e terceiro maiores artilheiros no período, com marca superior a 200 gols. Cristiano Ronaldo ainda foi o segundo colocado em participações em gols e Lewandowski o quarto. Em seguida, é fundamental o destaque de Luis Suárez como o terceiro maior em participações em gol. Ainda há de se ressaltar Kevin de Bruyne que foi líder em assistências ao lado de Messi, com 97 assistências no período, tendo papel crucial nos 3 títulos de sua equipe, Manchester City.

Autoavaliação

Acredito que o projeto conseguiu responder de forma consistente às perguntas previamente definidas. Os dados coletados, bem como as manipulações exercidas foram capazes de responder a grande maioria das perguntas específicas e principalmente o objetivo central.

No que diz respeito às dificuldades enfrentadas, destaca-se o processo de ETL desenvolvido no Data Fusion, no qual foi preciso identificar minuciosamente o log dos erros nas cargas que não deram certo de primeira até que todos os pipelines de dados fossem executados com sucesso. Além disso, destaca-se como maior desafio encontrado o processo de consertar o erro de duplicidade de jogador em uma mesma partida na tabela shots.

Por fim, para enriquecer este projeto, seria fundamental que a fonte de dados fosse conectada de forma automática à o GCP Storage e que o Data Fusion fosse agendado para proporcionar atualizações incrementais no processo de ETL. Ainda neste sentido, deve-se agendar as consultas do BigQuery que fazem alterações nas tabelas geradas pelo Data Fusion para que ocorram em sequência à conclusão do pipeline de dados. Para que ao final de tudo formate-se o layout e os visuais do Power Bi para publicá-lo e agendar atualizações diárias.