

# ETL

**GitHub Link:** [bishnu123-rgb/football\\_etl](https://github.com/bishnu123-rgb/football_etl)

This manual provides a guide for setting up and running the Football ETL Pipeline and Superset Dashboard.

By following this guide, you will:

1. Download the project from GitHub.
2. Set up the required environment on your local computer.
3. Run the ETL pipeline to prepare the football dataset.
4. Launch Apache Superset to explore the dashboard and charts.

## Prerequisites

Before starting, please make sure you have:

- A computer with Internet access
- Installed software:
  - [Python 3.11+](#)



- [PostgreSQL 14+](#)



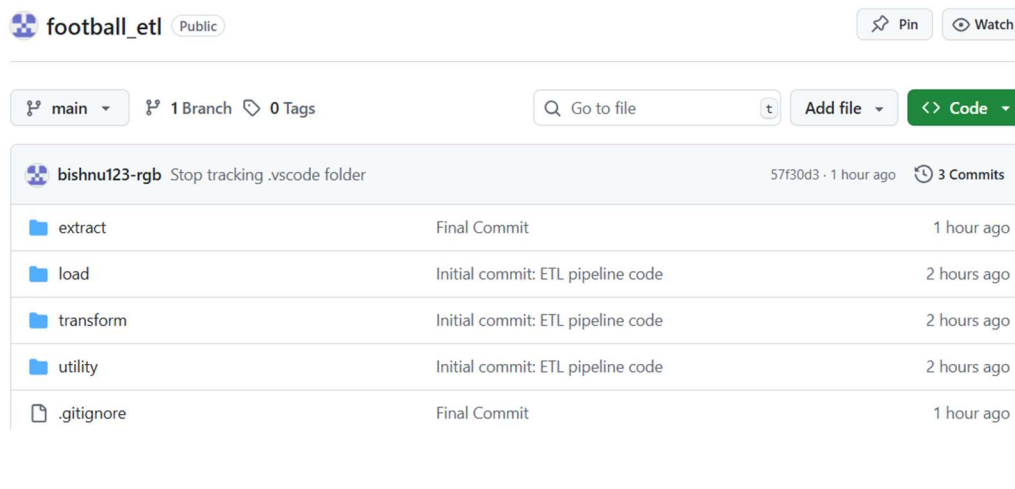
- [Git](#)



## Step 1 – Download the Project from GitHub

- Open a terminal (Linux/Mac) or PowerShell (Windows).
- Run the following command:

```
(venv) bishnu@Bishnu:~$ git clone https://github.com/bishnu123-rgb/football_etl.git
cd football_etl
```



## Step 2 – Set Up Python Environment

- Create a virtual environment.
- Activate the virtual environment.

```
bishnu@Bishnu:~/football_etl$ source venv/bin/activate
(venv) bishnu@Bishnu:~/football_etl$
```

- Install required dependencies.

### Step 3 – Setup PostgreSQL Database

- Open PostgreSQL terminal:

```
(venv) bishnu@Bishnu:~/football_etl$ psql -U postgres
Password for user postgres:
psql (15.13 (Debian 15.13-0+deb12u1))
Type "help" for help.

postgres=#
```

- Create a new database:

```
postgres=# CREATE DATABASE football;
CREATE USER bishnu WITH PASSWORD 'bishnu';
GRANT ALL PRIVILEGES ON DATABASE football TO bishnu;
ERROR:  database "football" already exists
ERROR:  role "bishnu" already exists
GRANT
postgres=#
```

- Exit PostgreSQL with \q.

### Step 4 – Run the ETL Pipeline

- Navigate to the project folder.
- Run Extract, Transform, and Load scripts step by step:

```
(venv) bishnu@Bishnu:~/football_etl$ python extract/execute.py
2025-08-25 15:23:57 - INFO - Extracting /home/bishnu/football_etl/extracted_data/soccer.zip...
2025-08-25 15:23:57 - INFO - Extraction complete.
2025-08-25 15:23:57 - INFO - Reading database.sqlite...
2025-08-25 15:24:02 - INFO - matches.csv created! (25979 rows)
2025-08-25 15:24:02 - INFO - teams.csv created! (299 rows)
2025-08-25 15:24:02 - INFO - All CSVs verified successfully
2025-08-25 15:24:02 - INFO - Extraction finished in 5 seconds
```

```
(venv) bishnu@Bishnu:~/football_etl$ python transform/execute.py
WARNING: Using incubator modules: jdk.incubator.vector
Using Spark's default log4j profile: org/apache/spark/log4j2-defaults.properties
25/08/25 15:26:07 WARN Utils: Your hostname, Bishnu, resolves to a loopback address: 127.0.1.1; using 10.255.255.254 instead (on interface lo)
25/08/25 15:26:07 WARN Utils: Set SPARK_LOCAL_IP if you need to bind to another address
Using Spark's default log4j profile: org/apache/spark/log4j2-defaults.properties
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLevel(newLevel).
25/08/25 15:26:09 WARN NativeCodeLoader: Unable to load native-hadoop library for your platform... using builtin-java classes where applicable
2025-08-25 15:26:12 - INFO - Starting transformation...
2025-08-25 15:26:15 - INFO - Matches loaded: 27383 rows
2025-08-25 15:26:15 - INFO - Teams loaded: 299 rows
2025-08-25 15:26:17 - INFO - Final transformed dataset: 25979 rows
25/08/25 15:26:17 WARN SparkStringUtils: Truncated the string representation of a plan since it was too large. This behavior can be adjusted by setting 'spark.sql.debug.maxToStringFields'.
25/08/25 15:26:18 WARN MemoryManager: Total allocation exceeds 95.00% (1,020,054,720 bytes) of heap memory
Scaling row group sizes to 95.00% for 8 writers
2025-08-25 15:26:21 - INFO - Saved cleaned matches to /home/bishnu/football_etl/parquet_output/matches_cleaned.parquet
2025-08-25 15:26:23 - INFO - Saved league standings to /home/bishnu/football_etl/parquet_output/league_standings.parquet
2025-08-25 15:26:25 - INFO - Saved team stats to /home/bishnu/football_etl/parquet_output/team_stats.parquet
2025-08-25 15:26:25 - INFO - Phase 1 Transformation completed in 13 seconds
(venv) bishnu@Bishnu:~/football_etl$
```

```
(venv) bishnu@Bishnu:~/football_etl$ python load/execute.py
2025-08-25 15:27:57 - INFO - Starting Phase 1 PostgreSQL load...
2025-08-25 15:27:58 - INFO - Loaded 27383 rows from parquet
2025-08-25 15:27:58 - INFO - Inserted 27383 rows into matches_teams
2025-08-25 15:27:58 - INFO - Phase 1 PostgreSQL load completed in 1 seconds
(venv) bishnu@Bishnu:~/football_etl$
```

- Verify data is loaded into PostgreSQL:

```
football=> \dt
          List of relations
 Schema |      Name      | Type  | Owner
-----+-----+-----+-----
 public | league_map     | table | bishnu
 public | league_standings | table | bishnu
 public | matches_cleaned | table | bishnu
 public | matches_teams  | table | bishnu
 public | player_stats   | table | bishnu
 public | team_stats     | table | bishnu
(6 rows)

football=>
```

## Step 5 – Setup Apache Superset

- Initialize Superset database:

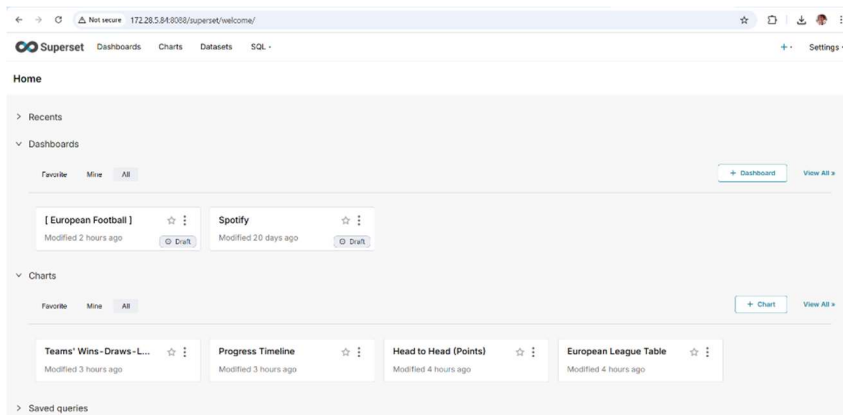
```
(venv) bishnu@Bishnu:~/football_etl$ superset db upgrade
2025-08-25 15:35:07,342:INFO:superset.utils.screenshots:No PIL installation found
2025-08-25 15:35:07,810:INFO:superset.utils.pdf:No PIL installation found
WARNI [alembic.env] SQLite Database support for metadata databases will be removed in a future version of Superset.
INFO [alembic.env] Starting the migration scripts.
INFO [alembic.runtime.migration] Context impl SQLiteImpl.
INFO [alembic.runtime.migration] Will assume transactional DDL.
INFO [alembic.env] Migration scripts completed. Duration: 00:00:00
(venv) bishnu@Bishnu:~/football_etl$
```

- Create an admin user.
- Initialize and run:

```
(venv) bishnu@Bishnu:~/football_etl$ superset init
2025-08-25 15:35:43,387:INFO:superset.utils.screenshots:No PIL installation found
2025-08-25 15:35:43,647:INFO:superset.utils.pdf:No PIL installation found
2025-08-25 15:35:44,548:INFO:superset.security.manager:Syncing role definition
2025-08-25 15:35:44,560:INFO:superset.security.manager:Syncing Admin perms
2025-08-25 15:35:44,562:INFO:superset.security.manager:Syncing Alpha perms
2025-08-25 15:35:44,564:INFO:superset.security.manager:Syncing Gamma perms
2025-08-25 15:35:44,567:INFO:superset.security.manager:Syncing sql_lab perms
2025-08-25 15:35:44,569:INFO:superset.security.manager:Fetching a set of all perms to lookup which ones are missing
2025-08-25 15:35:44,571:INFO:superset.security.manager:Creating missing datasource permissions.
2025-08-25 15:35:44,578:INFO:superset.security.manager:Creating missing database permissions.
2025-08-25 15:35:44,579:INFO:superset.security.manager:Cleaning faulty perms
```

```
(venv) bishnu@Bishnu:~/football_etl$ superset run -p 8088 --with-threads --reload --debugger
2025-08-25 15:36:01,956:INFO:superset.utils.screenshots:No PIL installation found
2025-08-25 15:36:02,213:INFO:superset.utils.pdf:No PIL installation found
* Serving Flask app 'superset'
* Debug mode: off
2025-08-25 15:36:02,621:INFO:werkzeug:WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:8088
2025-08-25 15:36:02,621:INFO:werkzeug:Press CTRL+C to quit
2025-08-25 15:36:02,622:INFO:werkzeug: * Restarting with stat
2025-08-25 15:36:03,745:INFO:superset.utils.screenshots:No PIL installation found
2025-08-25 15:36:04,000:INFO:superset.utils.pdf:No PIL installation found
2025-08-25 15:36:04,404:WARNING:werkzeug: * Debugger is active!
2025-08-25 15:36:04,410:INFO:werkzeug: * Debugger PIN: 113-979-907
```

- Open in your browser:



## Step 6 – Connect Superset to Database

- Log in with your Superset admin account.
- Go to Settings → Data → Databases → + Database.
- Select Postgres.
- Enter connection string:

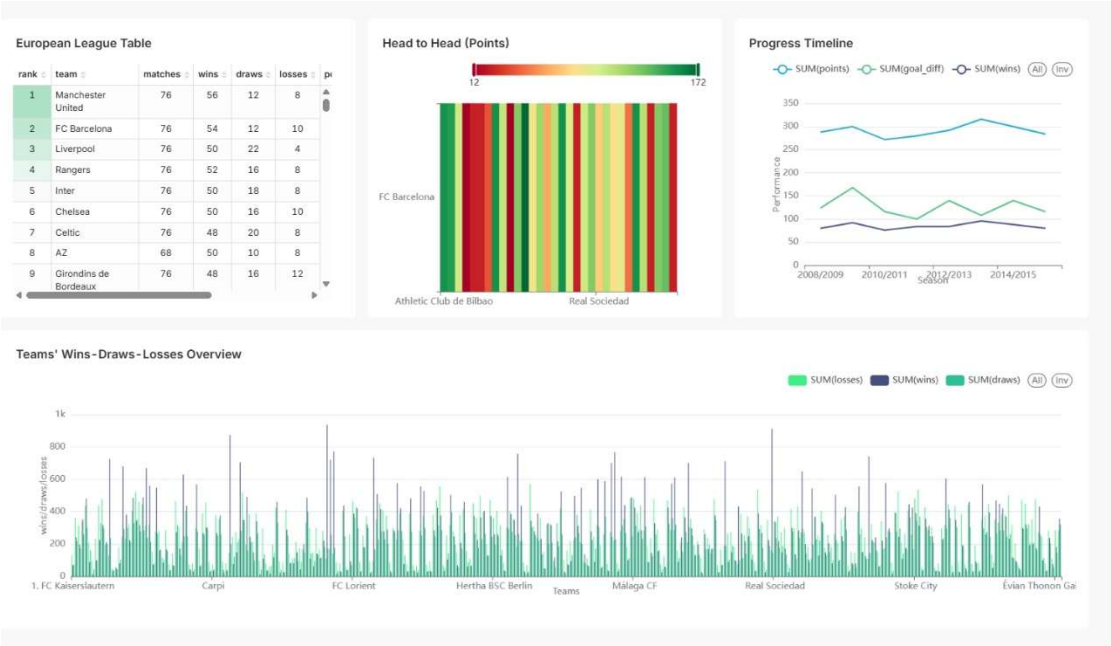
postgresql+psycopg2://bishnu:bishnu@localhost:5432/football

- Test and save.

## Step 7 – Load or Create Virtual Dataset and Create charts through queries and save charts into dashboard.

In my case,

**Dashboard:**



**Filter Section:**

## Filters



### Season

2008/2009 ×



### Team

FC Barcelona ×



### Trends

Arsenal ×



Apply filters

Clear all

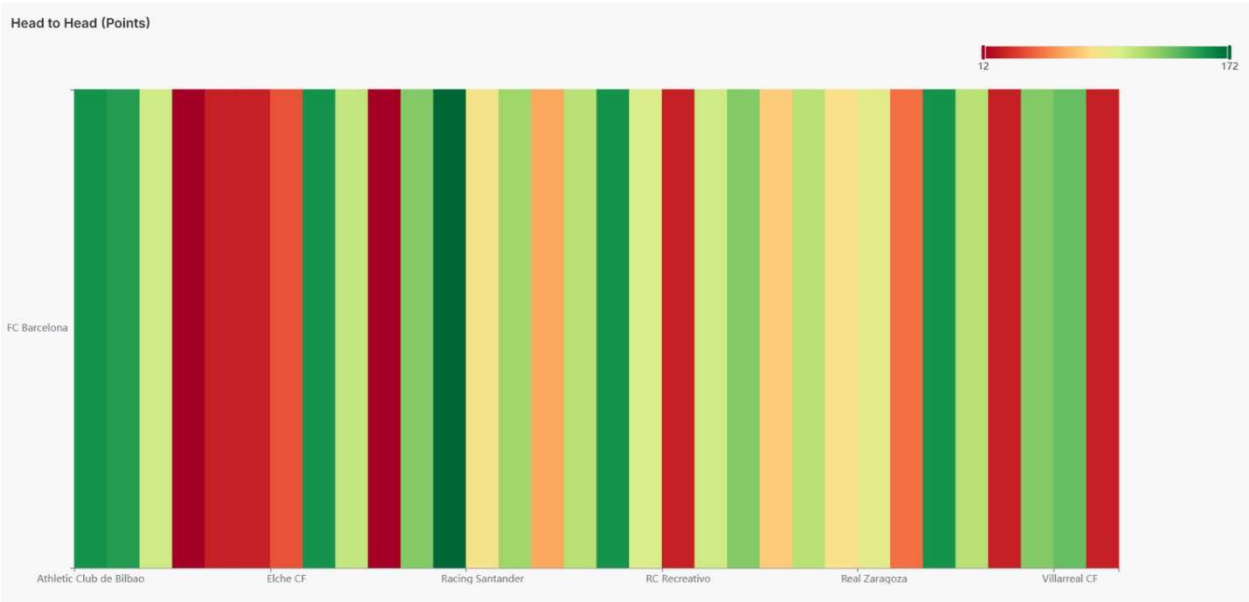
## Charts



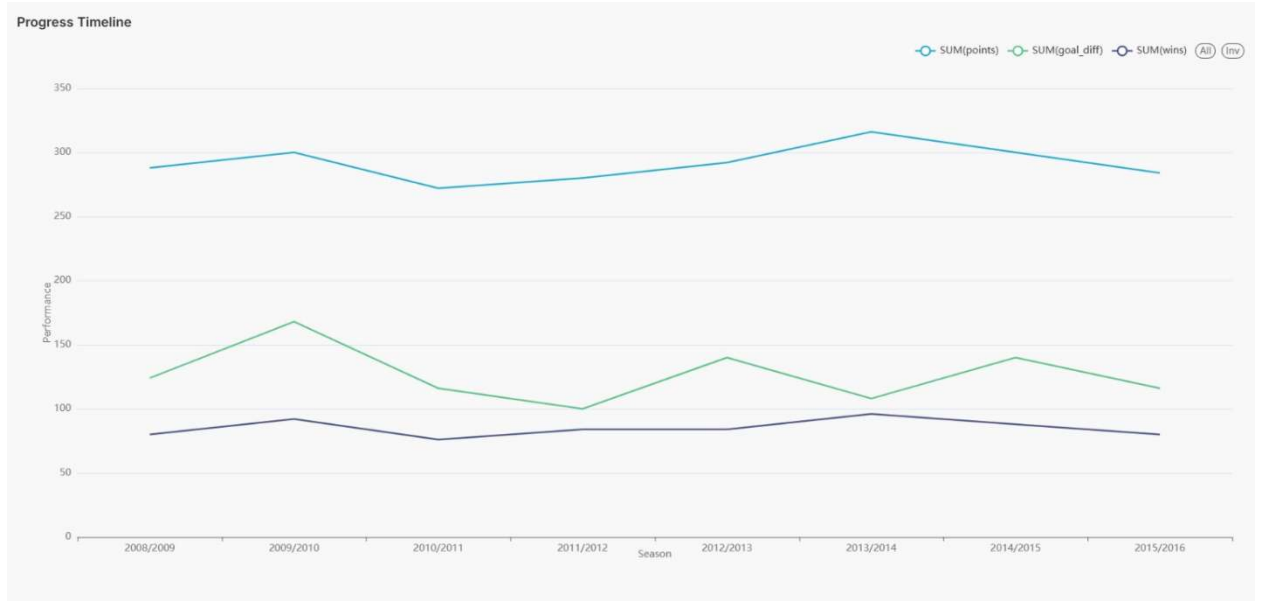
1) European League Table

European League Table									
rank	team	matches	wins	draws	losses	points	gf	ga	gd
1	Manchester United	76	56	12	8	180	136	48	88
2	FC Barcelona	76	54	12	10	174	210	70	140
3	Liverpool	76	50	22	4	172	154	54	100
4	Rangers	76	52	16	8	172	154	56	98
5	Inter	76	50	18	8	168	140	64	76
6	Chelsea	76	50	16	10	166	136	48	88
7	Celtic	76	48	20	8	164	160	66	94
8	AZ	68	50	10	8	160	132	44	88
9	Girondins de Bordeaux	76	48	16	12	160	128	68	60
10	FC Zürich	72	48	14	10	158	160	72	88
11	Real Madrid CF	76	50	6	20	156	166	104	62
12	RSC Anderlecht	68	48	10	10	154	150	60	90
13	Standard de Liège	68	48	10	10	154	132	52	80
14	Olympique de Marseille	76	44	22	10	154	134	70	64
15	Milan	76	44	16	16	148	140	70	70
16	Juventus	76	42	22	12	148	138	74	64
17	BSC Young Boys	72	44	14	14	146	170	92	78
18	Arsenal	76	40	24	12	144	136	74	62
19	FC Basel	72	44	12	16	144	144	88	56
20	Olympique Lyonnais	76	40	22	14	142	104	58	46
21	FC Porto	60	42	14	4	140	122	36	86
22	Sevilla FC	76	42	14	20	140	108	78	30

2) Head-to-Head



3) Progress Timeline



#### 4) Overview

