UEMF
الجامعة الأورومتوسطية بفاس
EUROMED UNIVERSITY OF FES
UNIVERSITÉ EUROMED DE FÈS

EIDIA
École d'Ingénierie Digitale et d'Intelligence Artificielle

Modern Football Data Pipeline

Case Study: Football Data Analysis

Final Project Report

Prepared by:

Imane MALIKI
Hanae TALEBI
Lahcen ASSNAOUI
Ayman BOUATTANE

January 17, 2026

Program: Data Engineering & Analytics

Academic Year: 2024-2025

Contents

Chapter 1

Introduction

1.1 Context of the Project

Football clubs and analysts generate vast amounts of data on a daily basis, including match results, individual player statistics, tracking data, and detailed training metrics. This wealth of information provides valuable insights but also presents challenges in terms of organization, processing, and analysis.

The objective of this project is to design and implement a modern data pipeline capable of efficiently handling football-related data. The pipeline will systematically collect data from multiple sources, store it in a structured and scalable manner, perform necessary transformations to ensure consistency and usability, and finally provide intuitive visualizations and reports. By doing so, clubs and analysts will be able to make data-driven decisions, optimize player performance, and gain strategic insights into team dynamics and match outcome

1.2 Objectives

The main objectives of this project are to design and implement a comprehensive football data pipeline that supports analysis and decision-making. Specifically, the goals are:

- **Build a modern cloud data architecture:** Design a scalable and robust infrastructure that can store large volumes of football-related data, ensuring high availability, security, and efficient data management in the cloud.
- **Automate data ingestion and transformation:** Develop automated processes to collect data from multiple sources (e.g., match results, player statistics, tracking data) and transform it into a consistent and structured format, reducing manual effort and minimizing errors.
- **Centralize football statistics:** Consolidate disparate datasets into a unified repository, enabling analysts and decision-makers to access complete and reliable football data from a single source.
- **Prepare data for BI dashboards:** Transform and organize the data to be readily consumable by business intelligence tools, supporting the creation of interactive dashboards, visualizations, and reports for performance analysis, tactical evaluation, and strategic planning.

1.3 Football Use Case

The data pipeline in this project is specifically designed to process and analyze football-related data. This includes a datasets that provide both performance metrics and contextual insights. Key types of data processed by the pipeline include:

- **Match results from the AFCON:** This includes scores, match dates. Analyzing these results over time helps identify team performance trends, win/loss patterns, and league standings dynamics.
- **Team and player statistics:** Detailed metrics such as possession, passes completed, and player appearances are collected. These statistics enable in-depth analysis of individual player contributions, team strategies, and tactical performance.
- **Goals, cards, and expected goals (xG):** Tracking goals scored, yellow/red cards, and advanced metrics like expected goals provides insights into scoring efficiency, discipline, and the underlying quality of attacking opportunities. xG, in particular, helps evaluate whether a team or player is performing above or below expectation, independent of raw results.

By centralizing and structuring these datasets, the pipeline allows analysts to uncover hidden patterns, compare team and player performance, and generate actionable insights for coaches, analysts, and decision-makers in Moroccan football.

1.4 Global Architecture

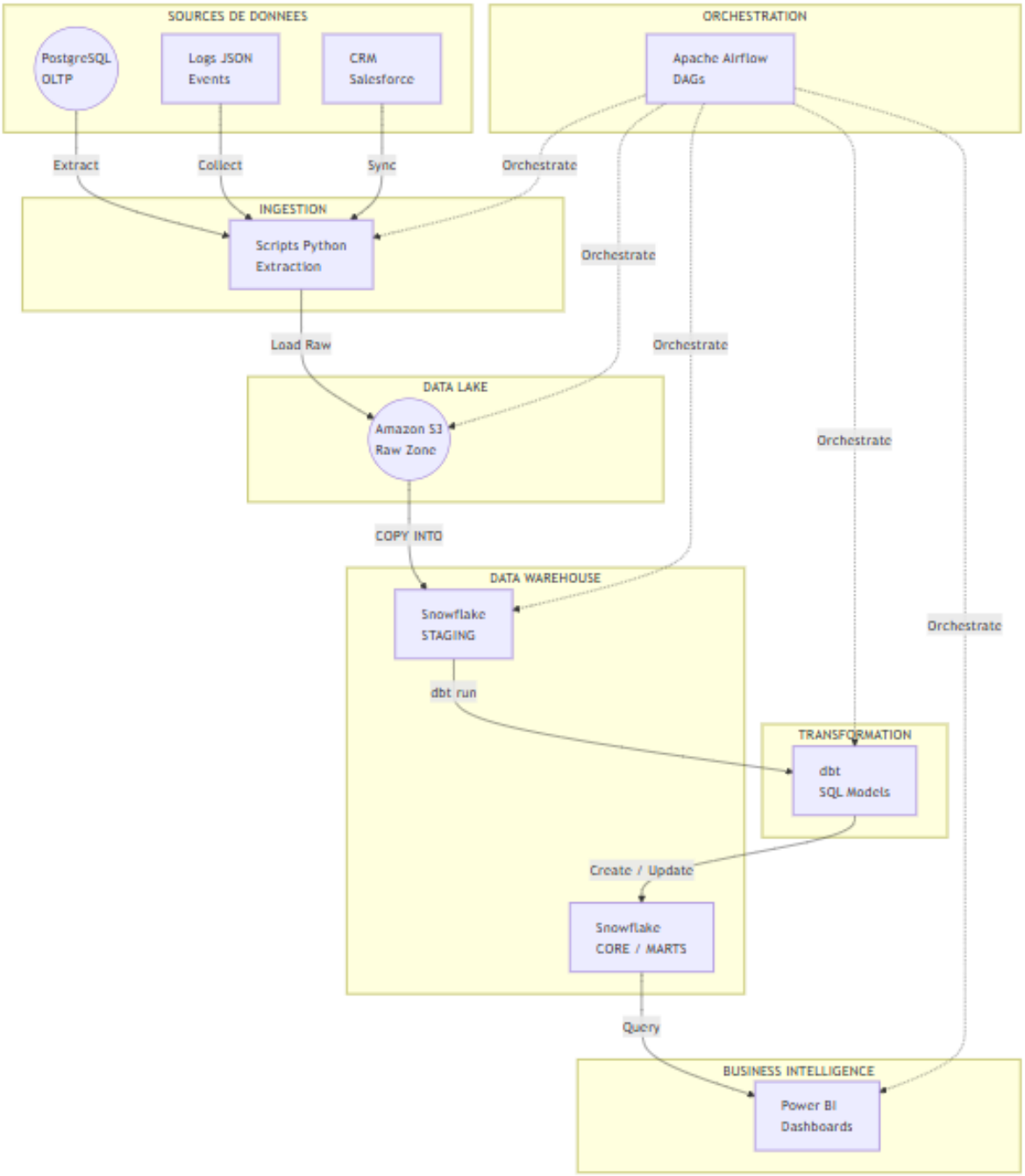


Figure 1 : Architecture du Modern Data Stack pour ShopStream

Figure 1.1: Overall architecture of the football data pipeline

Chapter 2

Data Source: PostgreSQL

2.1 Description

The raw football match data is stored in a PostgreSQL relational database. Several tables have been created to model the different entities involved in the game as well as their relationships.

- Creation of Data base :

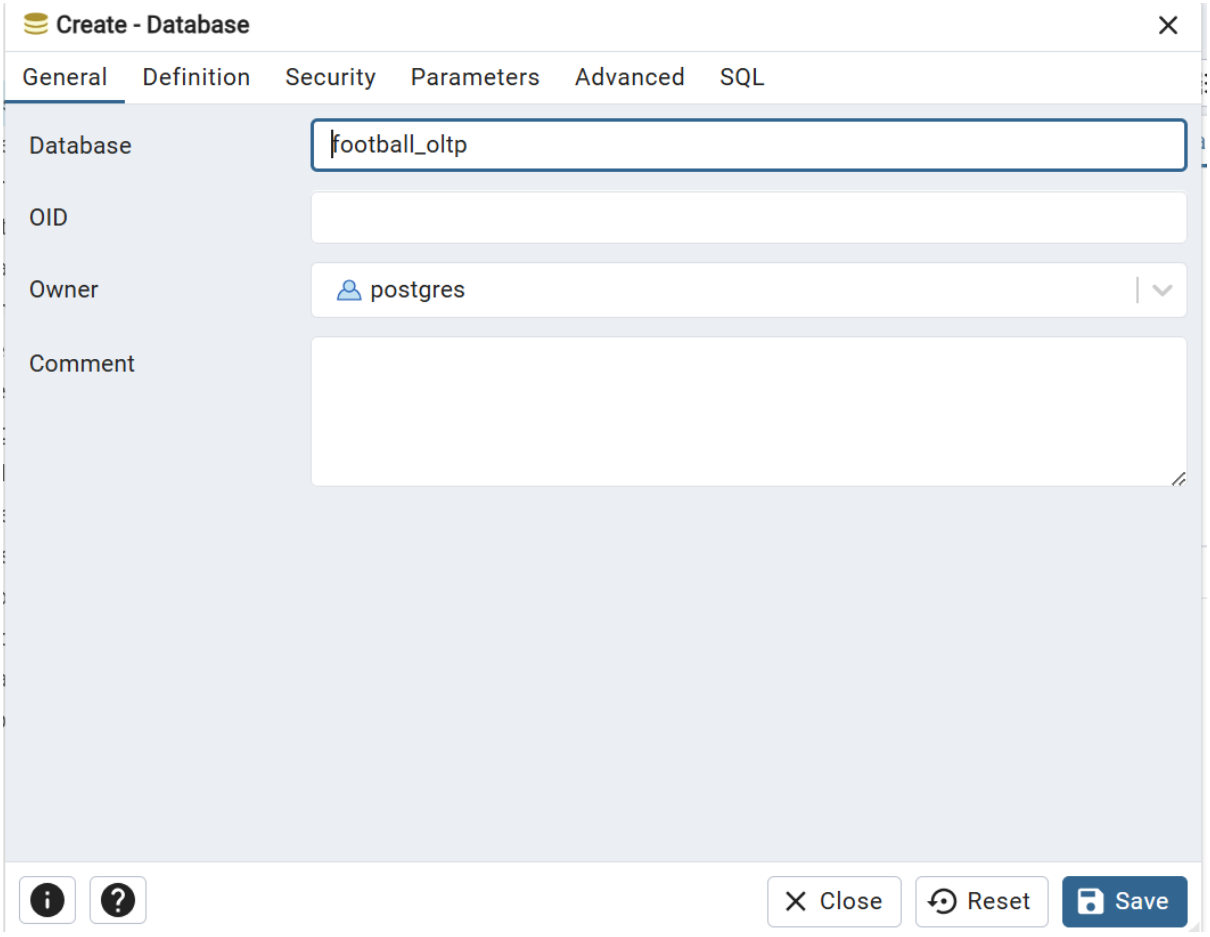


Figure 2.1: Creation the database football$_oltpinPostgreSQL$

- Creation of Tables ,the main tables are:

- **players:** contains personal and professional information about players , like : player_id,name,national_team,position,age,height_cm,weight_kg,club_example ,speed,shooting,passing,dribbling,defending,physical,overall_rating, matches_played_season,minutes_played_season ,goals,assists,yellow_cards,red_cards,injury_risk_index,availability_flag
- **matches**: represents each official match, including : match_id,match_date,opponent, competition,venue,stadium_name,result .
- **matches_players_stats**: stores detailed statistics for each player participating in a match (minutes played, goals, assists, cards, etc.).
- **injuries**: records injuries suffered by players, including tinjury_id,player_id, injury_type,start_date,end_date,severity .
- **training**: stores information related to players' training sessions such : id,player_id, session_date,duration_min,rpe,total_distance_km,sprint_distance_km,avg_heart_rate . .

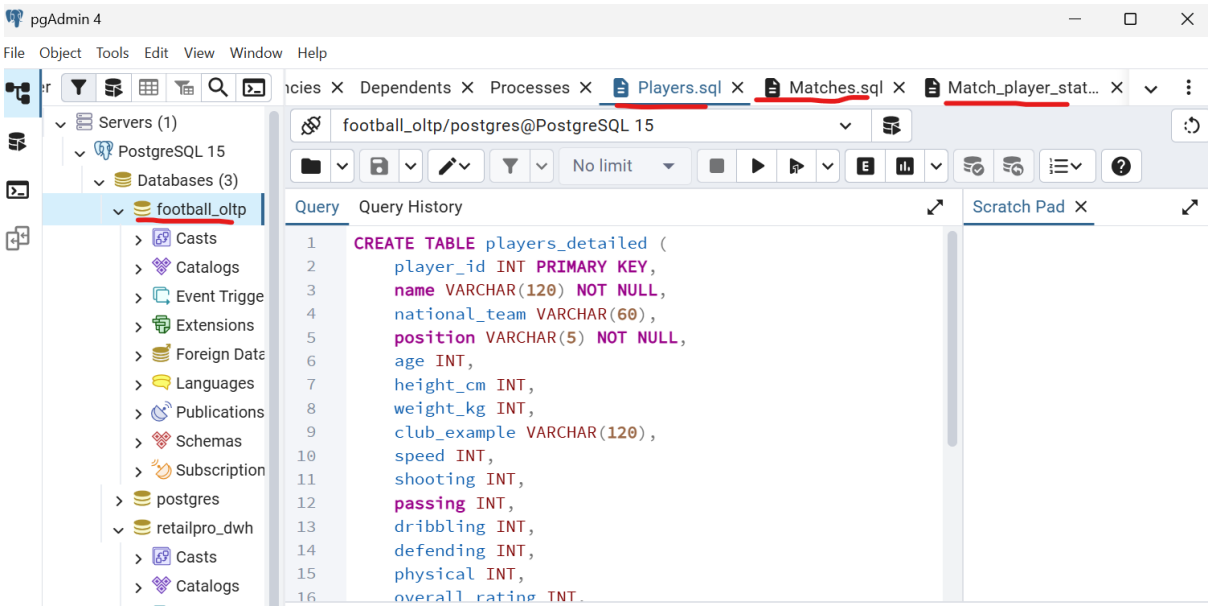


Figure 2.2: Example of Creation of Arrays players

- Insertation of Data in tables :

Query Query History

```
1  INSERT INTO players_detailed (
2      player_id,name,national_team,position,age,height_cm,we
3      speed,shooting,passing,dribbling,defending,physical,ov
4      matches_played_season,minutes_played_season,goals,assi
5      injury_risk_index,availability_flag
6  )
7  VALUES
8  (1,'Yassine Bounou','Morocco','GK',30,187,93,'Fake FC',57,
9  (2,'Munir Mohamedi','Morocco','GK',26,189,88,'Fake FC',41,
10 (3,'Mehdi Benabid','Morocco','GK',35,188,84,'Fake FC',38,3
11 (4,'Achraf Hakimi','Morocco','RB',24,178,85,'Fake FC',79,3
12 (5,'Noussair Mazraoui','Morocco','LB',33,179,76,'Fake FC',
13 (6,'Romain Saiss','Morocco','CB',35,185,72,'Fake FC',62,29
14 (7,'Nayef Aguerd','Morocco','CB',32,186,82,'Fake FC',56,32
15 (8,'Chadi Riad','Morocco','CB',26,190,84,'Fake FC',89,49,8
```

Figure 2.3: Insertion of data in players Table

Query Query History

```
1  INSERT INTO matches (
2      match_date, opponent, competition, venue, stadium_name,
3  ) VALUES
4  ('2025-01-10', 'Egypt', 'AFCON 2025', 'neutral', 'Stade de M
5  ('2025-01-15', 'Algeria', 'AFCON 2025', 'neutral', 'Stade de
6  ('2025-01-20', 'Nigeria', 'AFCON 2025', 'neutral', 'Stade Mo
7  ('2025-01-25', 'Senegal', 'AFCON 2025', 'neutral', 'Stade de
8  ('2025-01-30', 'Morocco', 'AFCON 2025', 'home', 'Stade Moham
9  ('2025-02-05', 'Ghana', 'AFCON 2025', 'away', 'Accra Sports
10 ('2025-02-10', 'Cameroon', 'AFCON 2025', 'neutral', 'Stade I
11 ('2025-02-15', 'Tunisia', 'AFCON 2025', 'neutral', 'Stade de
12
```

Data Output Messages Notifications

```
INSERT 0 8

Query returned successfully in 149 msec.
```

Figure 2.4: Insertion of data in matches Table

Query Query History

```
1  INSERT INTO injuries (
2      player_id, injury_type, start_date, end_date, severity
3  ) VALUES
4  (2, 'Ankle Sprain', '2025-02-03', '2025-02-20', 'medium'),
5  (5, 'Knee Ligament', '2025-01-25', NULL, 'severe'),
6  (7, 'Hamstring', '2025-02-10', '2025-02-28', 'medium'),
7  (10, 'Shoulder Dislocation', '2025-01-30', '2025-02-15', 'l
8  (12, 'Concussion', '2025-02-12', '2025-02-22', 'light'),
9  (13, 'Groin Strain', '2025-01-18', '2025-02-05', 'medium'),
10 (14, 'Muscle Tear', '2025-02-15', NULL, 'severe'),
11 (16, 'Back Pain', '2025-01-20', '2025-02-07', 'medium'),
12 (18, 'Finger Fracture', '2025-02-01', '2025-02-18', 'light'
13 (19, 'Calf Strain', '2025-02-08', '2025-02-25', 'medium'),
14 (22, 'Knee Sprain', '2025-01-28', '2025-02-12', 'medium'),
15 (23, 'Hamstring', '2025-02-05', '2025-02-20', 'medium')
```

Data Output Messages Notifications

```
INSERT 0 13

Query returned successfully in 157 msec.
```

Figure 2.5: Insertion of data in Injuries Table

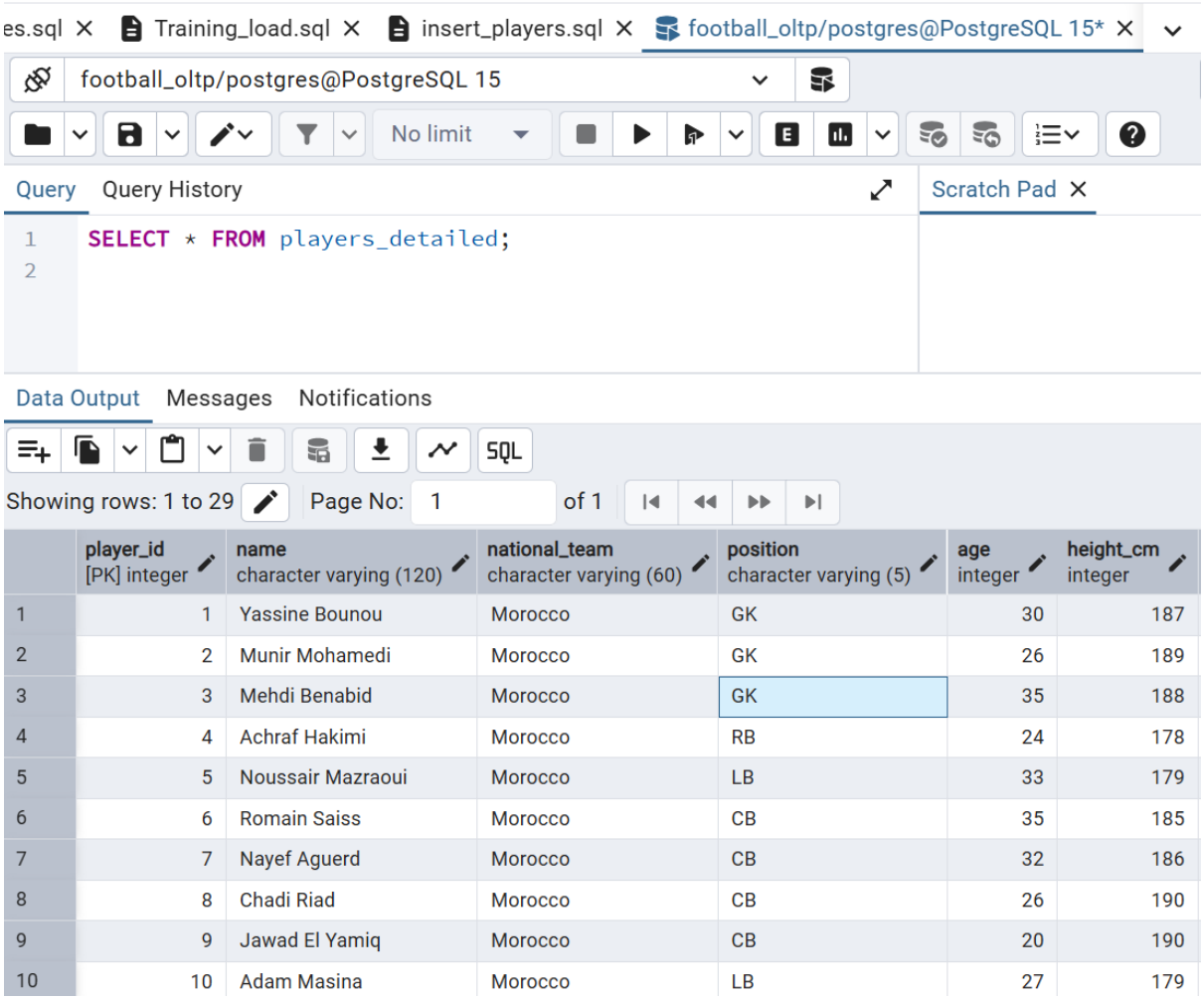
```
INSERT INTO training_load (
    player_id, session_date, duration_min, rpe,
    total_distance_km, sprint_distance_km, avg_heart_rate
) VALUES
(4, '2025-01-03', 85, 7, 9.8, 1.2, 158),
(4, '2025-01-05', 70, 6, 8.4, 1.0, 152),
(4, '2025-01-07', 90, 8, 10.2, 1.5, 160),
(4, '2025-01-09', 60, 5, 7.5, 0.8, 150),

(21, '2025-01-03', 75, 8, 8.1, 0.9, 165),
(21, '2025-01-05', 65, 7, 7.4, 0.7, 160),
(21, '2025-01-08', 80, 8, 8.9, 1.1, 162),
(21, '2025-01-10', 70, 7, 7.8, 0.9, 159),

(1, '2025-01-04', 60, 4, 5.2, 0.1, 140),
```

Figure 2.6: Insertion of data in Training Table

-All these tables were created in PostgreSQL using SQL CREATE TABLE statements,Data was then inserted into each table in order to populate the database and enable queries and analysis of player performance, match results and team management.



	player_id [PK] integer	name character varying (120)	national_team character varying (60)	position character varying (5)	age integer	height_cm integer
1	1	Yassine Bounou	Morocco	GK	30	187
2	2	Munir Mohamedi	Morocco	GK	26	189
3	3	Mehdi Benabid	Morocco	GK	35	188
4	4	Achraf Hakimi	Morocco	RB	24	178
5	5	Noussair Mazraoui	Morocco	LB	33	179
6	6	Romain Saiss	Morocco	CB	35	185
7	7	Nayef Aguerd	Morocco	CB	32	186
8	8	Chadi Riad	Morocco	CB	26	190
9	9	Jawad El Yamiq	Morocco	CB	20	190
10	10	Adam Masina	Morocco	LB	27	179

Figure 2.7: Example of Verification if the data is really created

Chapter 3

Data Lake: Amazon S3

3.1 Role of Amazon S3

Amazon S3 is used as the central **data lake** of the project. It allows the storage of large volumes of football-related data in a scalable and cost-efficient way. The data stored in S3 comes mainly from the PostgreSQL operational database.

The following types of data are stored in the bucket:

- CSV exports of football matches
- CSV files containing players and teams information
- player match statistics (per match and per season)
- JSON logs describing football events (goals, fouls, substitutions, VAR decisions, etc.)
- backup files and historical snapshots of the database

S3 is used as a separation layer between **data production** (PostgreSQL) and **data consumption** (analytics tools, dashboards, notebooks). This architecture follows the principles of modern data engineering and enables future integration with data warehouses or machine learning models.

3.2 Bucket Structure

The S3 bucket is organized according to a directory structure that reflects the football data model: - Creation of the bucket :

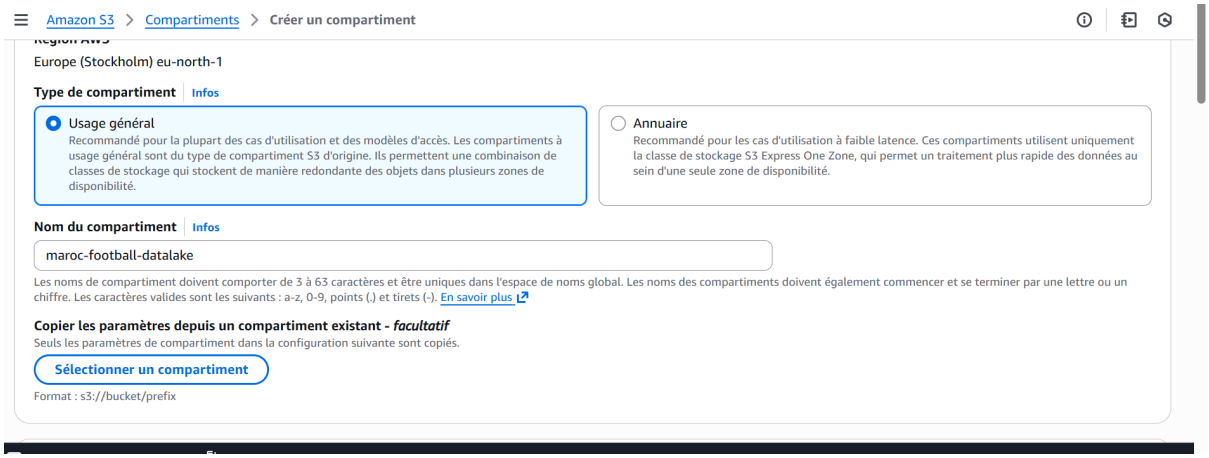


Figure 3.1: Creation of the bucket S3 :maroc-football-datalake

```
football-datalake/
 raw/
     postgres/
         players/
         teams/
         matches/
         matches_players_stats/
         injuries/
         training/
```

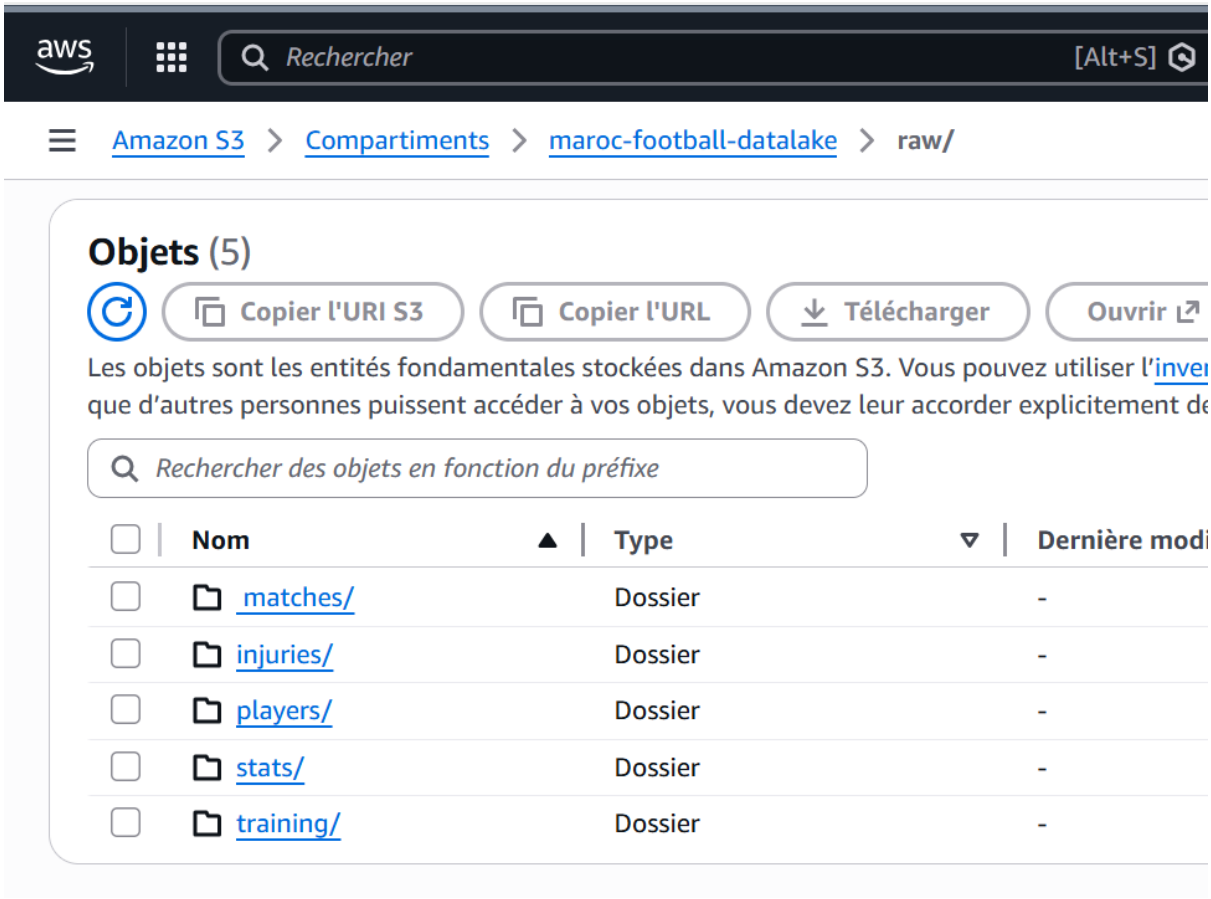


Figure 3.2: Amazon S3 bucket containing raw football data exported from PostgreSQL

Each subfolder contains daily partitions generated during the data export process, allowing easy time-based queries.

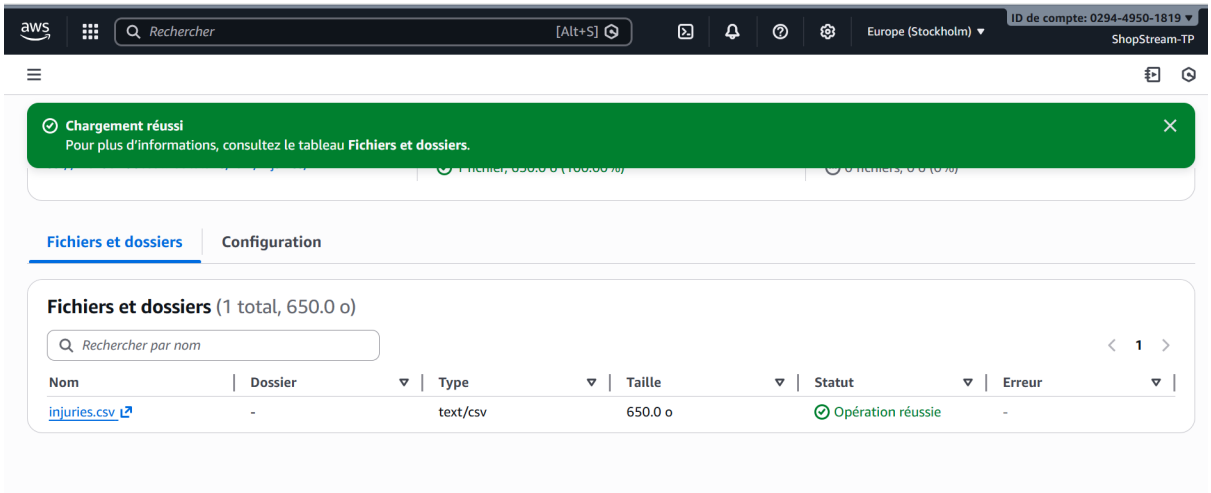


Figure 3.3: Example of file .csv inside each folder of maroc-football-datalake (injuries Example)

3.3 Role of IAM User

In order to securely access Amazon S3 from external applications such as Python scripts or ETL tools, a dedicated **IAM user** was created. This IAM user is not a human user, but a technical account whose sole purpose is to enable automated exports of football data from PostgreSQL to S3.

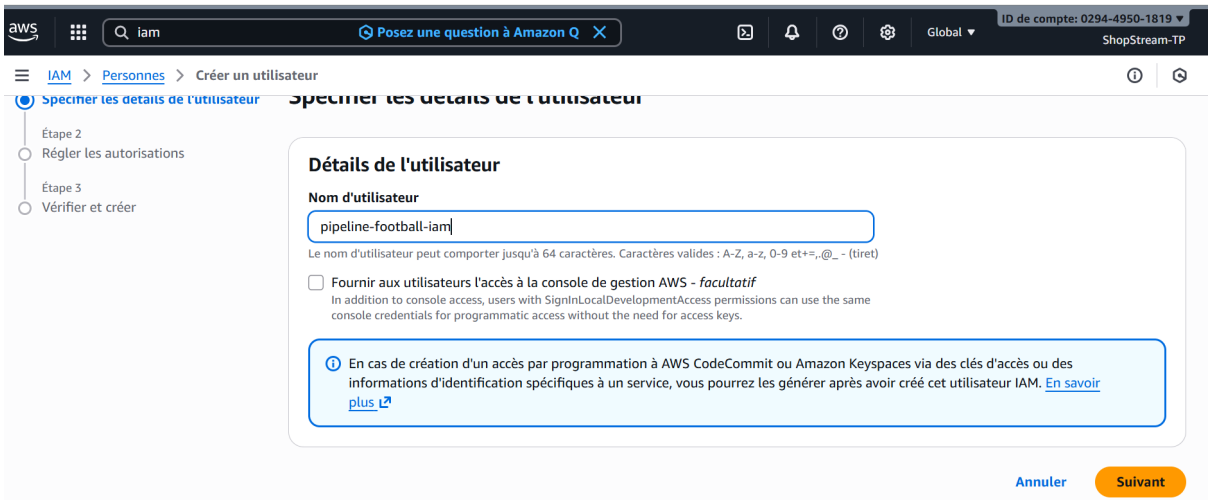


Figure 3.4: Creation of IAM User

The IAM user has the following characteristics:

- programmatic access through an Access Key ID and Secret Access Key;
- permissions restricted exclusively to the S3 bucket used in the project;
- policies configured according to the principle of least privilege.

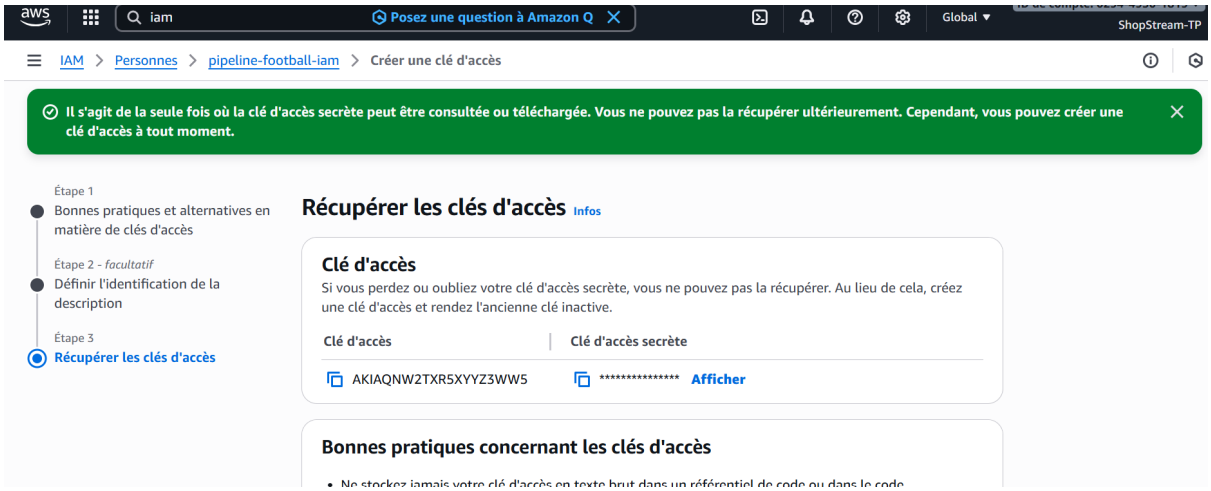


Figure 3.5: Creation of key access for IAM User

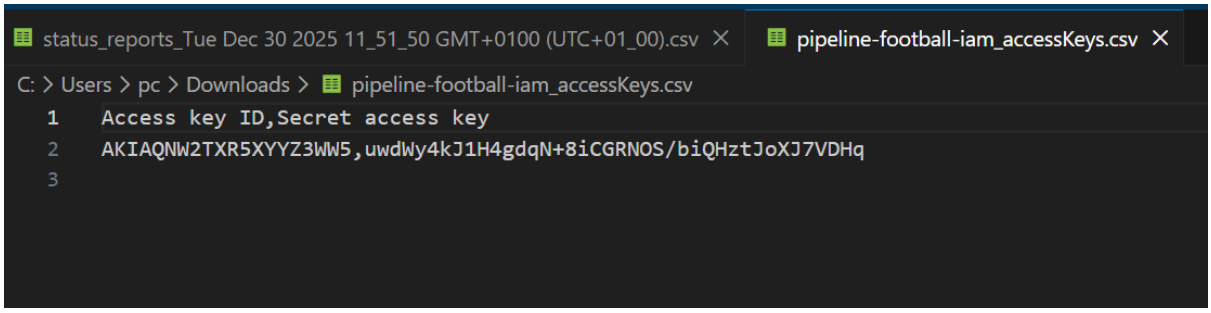


Figure 3.6: key access for IAM User

The credentials of the IAM user are configured in the export script so that it can:

- connect securely to AWS
- upload CSV files of matches, players, teams and statistics
- upload JSON event logs

The IAM user ensures **security best practices**, since:

- credentials are not shared with real users
- permissions are limited according to the principle of least privilege
- access can be revoked at any time without impacting database users

Chapter 4

Cloud Data Warehouse: Snowflake

4.1 Role of Snowflake

Snowflake is used as the central cloud data warehouse of the project. Data exported from PostgreSQL and stored in Amazon S3 is loaded into Snowflake, where it is transformed and modeled for football analytics.

Snowflake enables:

- scalable storage of historical football data,
- fast analytical queries,
- separation of storage and compute,
- secure access control through roles and warehouses.

4.2 Data Modeling Architecture

The data modeling strategy in Snowflake follows a layered architecture. The transformation logic is implemented using **dbt**, as reflected in the project structure.

4.3 Creation of Database ,Warehouse and Role for Snowflake:

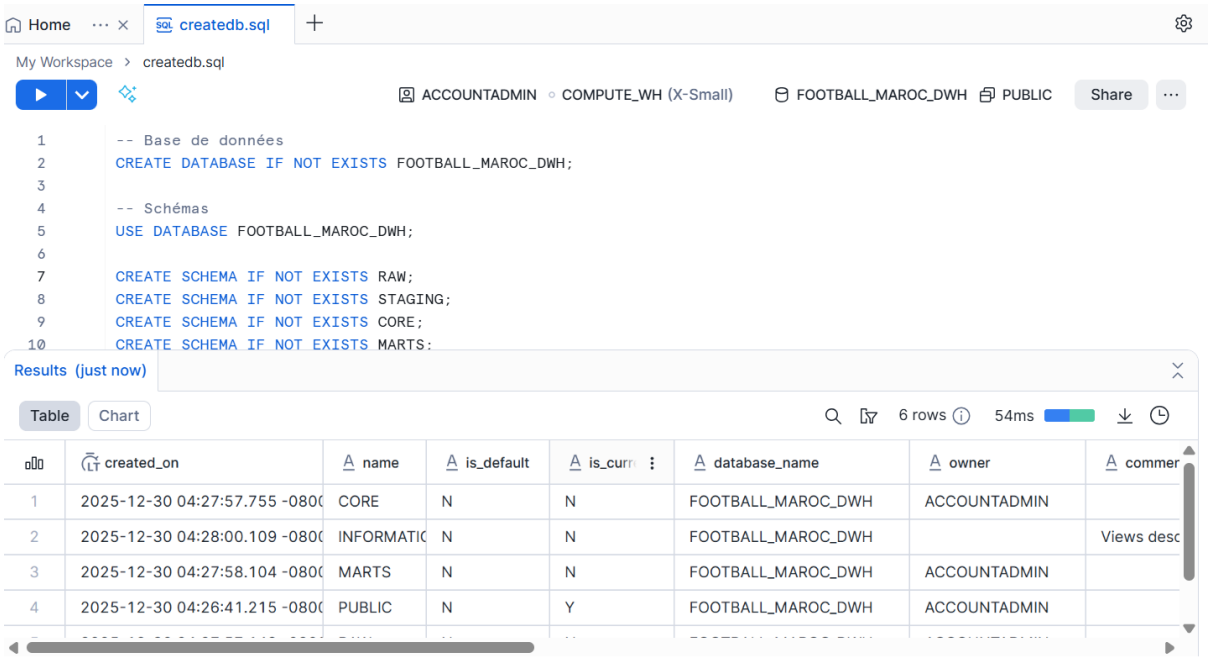


Figure 4.1: Snowflake DataBase

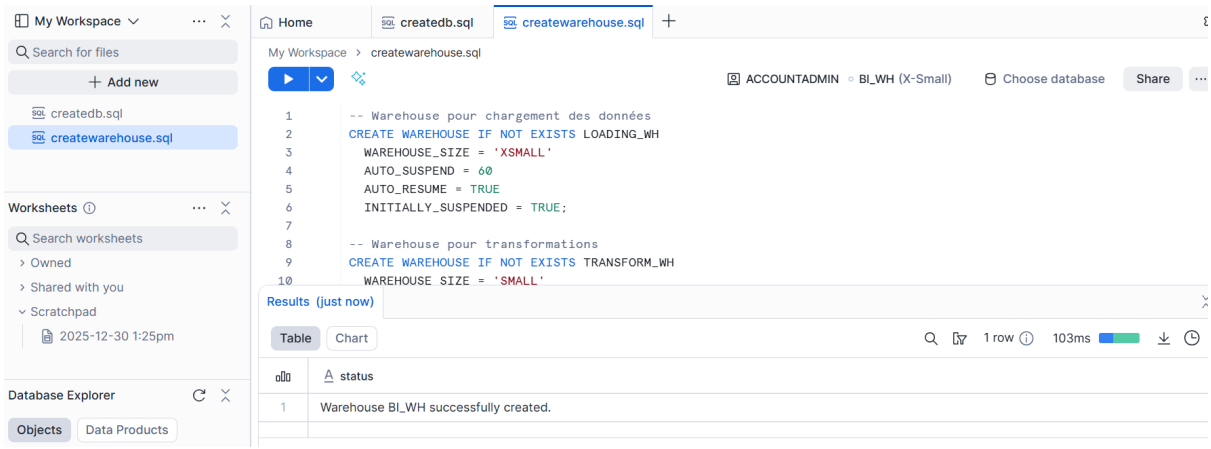


Figure 4.2: Snowflake Warehouse

Creation of Storage Integration 1.IAM ROLE

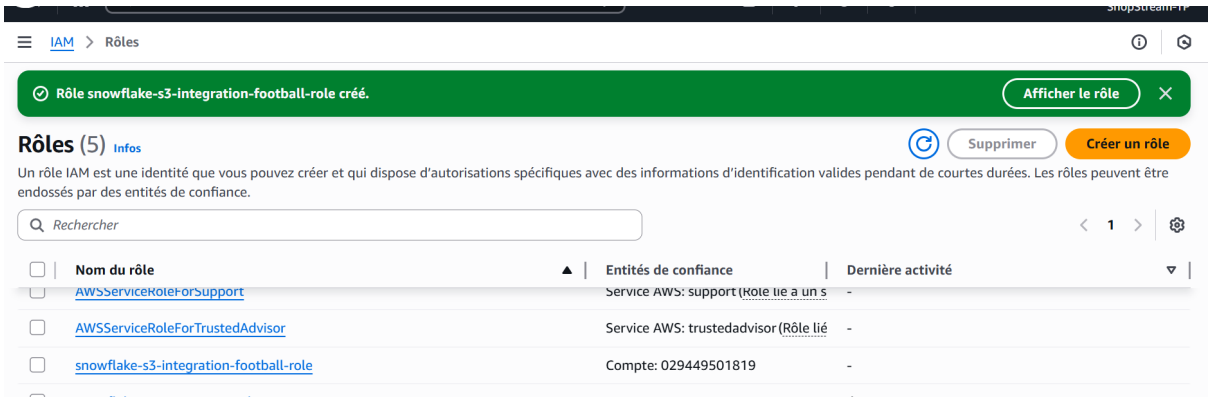


Figure 4.3: Creation of IAM role for Snowflake

2.Storage Integration

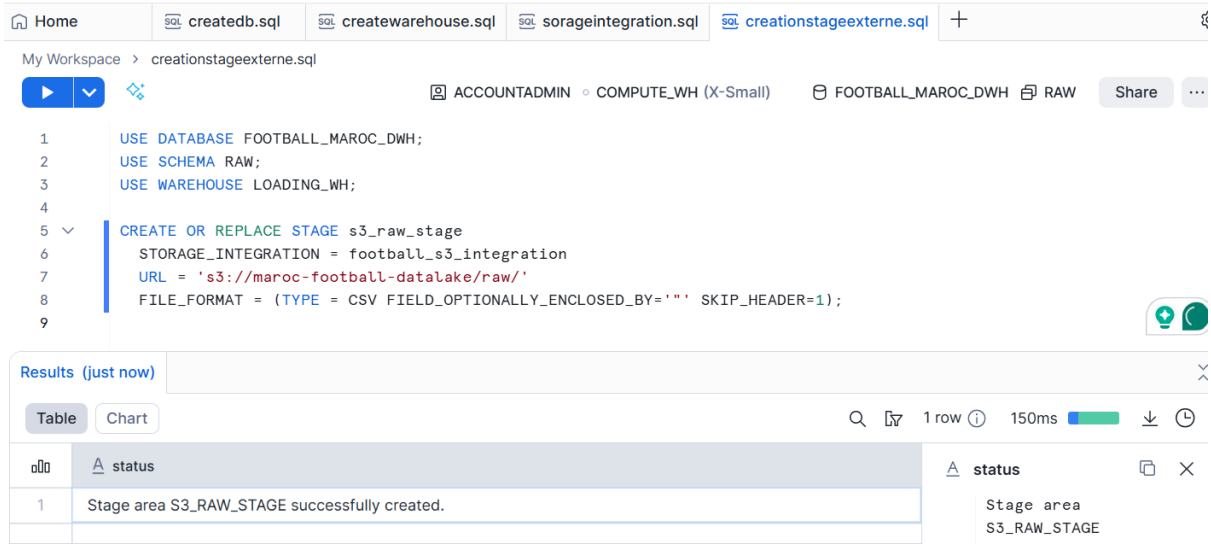


Figure 4.4: Creation of IAM role for Snowflake

2.Verfication of the Integration

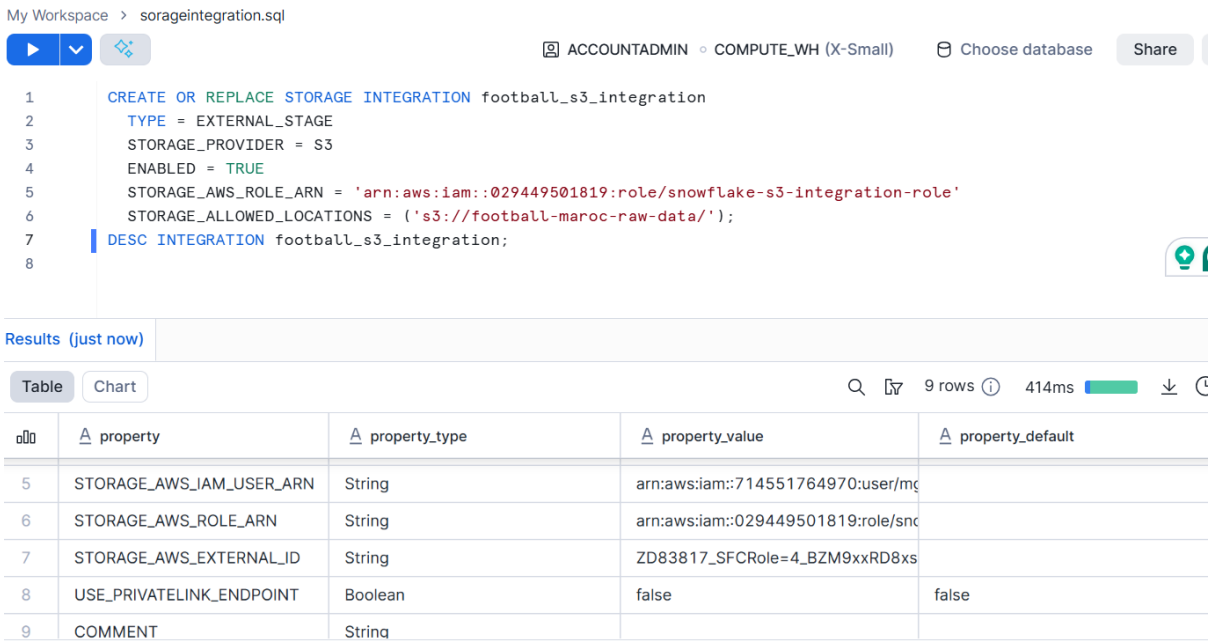


Figure 4.5: Creation of IAM role for Snowflake

4.4 Creation of externe Stage

Figure 4.6: Creation of externe Stage

4.4.1 Staging Layer

The **staging** schema contains tables that are direct, cleaned copies of the raw data coming from Amazon S3.

Examples of staging models:

- `stg_matches`
- `stg_players`
- `stg_match_player_stats`
- `stg_injuries`
- `stg_training_load`

4.5 Creation of staging tables

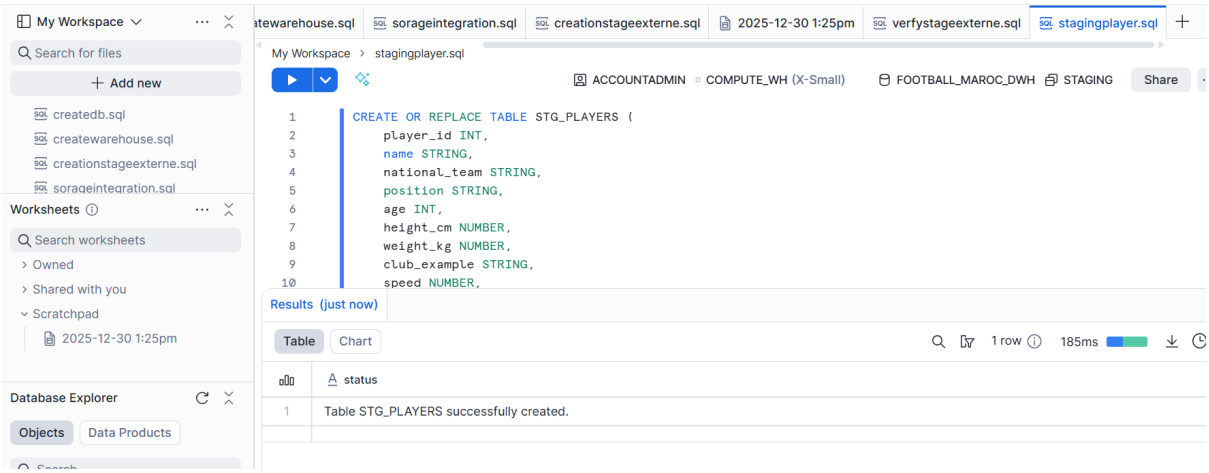


Figure 4.7: Creation of Staging Players

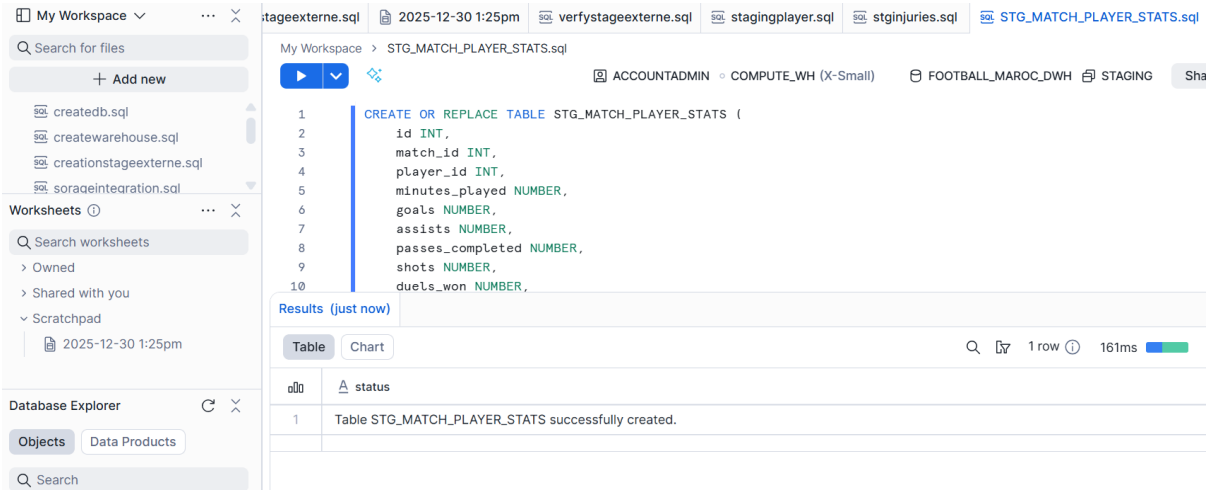


Figure 4.8: Creation of Staging Players_MATCHS_STATE

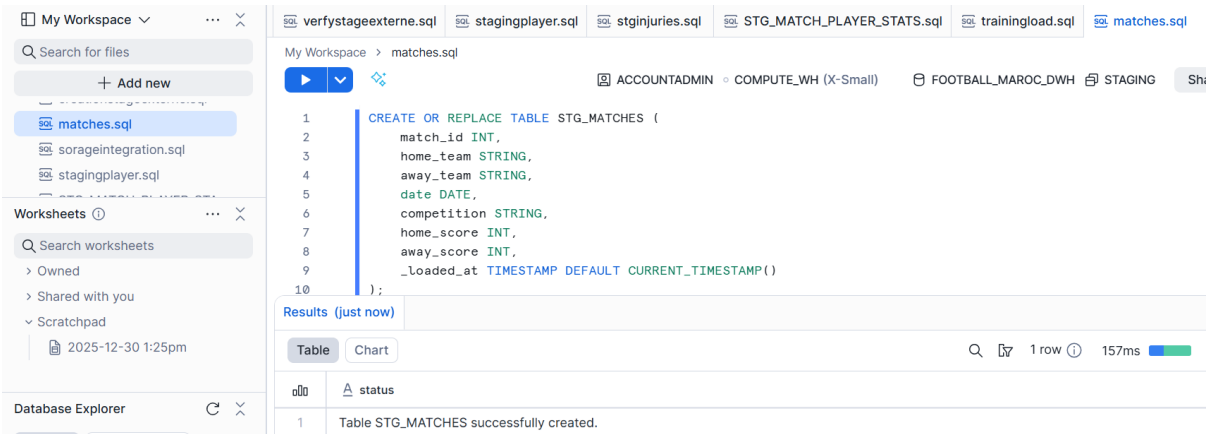


Figure 4.9: Creation of Staging Matches

Verfication of the creation of all Staging Tables

```
);
SHOW TABLES;
 Ctrl+I to generate
```

ults (just now)

able | Chart | 5 rows | 74ms

created_on	name	database_name	schema_name	kind	comment
2025-12-30 11:33:08.799 -080(	STG_MATCHES	FOOTBALL_MAROC_DWH	STAGING	TABLE	
2025-12-30 11:30:55.947 -080(	STG_MATCH_PLAYE	FOOTBALL_MAROC_DWH	STAGING	TABLE	
2025-12-30 11:23:51.088 -080(	STG_PLAYERS	FOOTBALL_MAROC_DWH	STAGING	TABLE	
2025-12-30 11:32:24.143 -080(	STG_TRAINING_LOAI	FOOTBALL_MAROC_DWH	STAGING	TABLE	

Figure 4.10: Creation of Staging Matches

- Filling Staging Tables :

My Workspace > loadstaggingtables.sql

ACCOUNTADMIN | LOADING_WH (X-Small) | FOOTBALL_MAROC_DWH | STAGING | Share

```
USE DATABASE FOOTBALL_MAROC_DWH;
USE SCHEMA STAGING;
USE WAREHOUSE LOADING_WH;

-- ========================================
-- 1 STG_PLAYERS
-- ========================================
COPY INTO STG_PLAYERS
FROM @RAW.S3_RAW_STAGE/players/
FILE_FORMAT = (
    TYPE = CSV
    SKIP_HEADER = 1
    FIELD_OPTIONALLY_ENCLOSED_BY='"'
    ERROR_ON_COLUMN_COUNT_MISMATCH = FALSE
)
```

Results (just now)

Table | Chart | 1 row | 50ms

	TABLE_NAME	ROW_COUNT
1	STG_PLAYERS rows:	29

Figure 4.11: Inserting data into Staging tables- Players as an Example

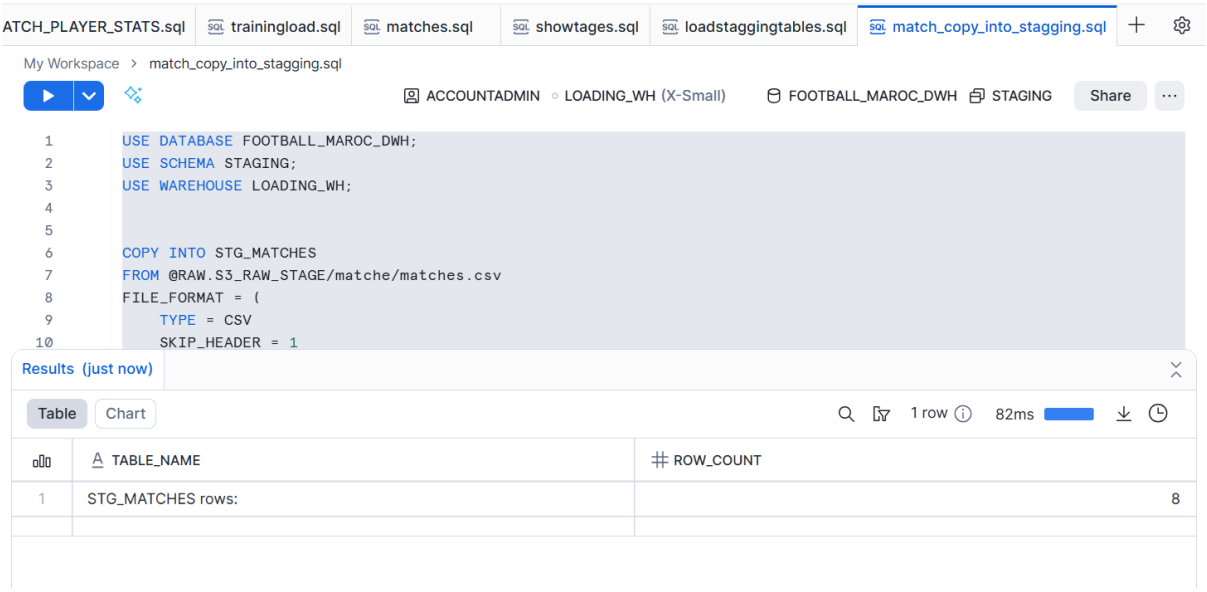


Figure 4.12: Inserting data into Staging tables- Matches as an Example

Their role is to:

- standardize column names,
- cast data types,
- remove duplicates,
- prepare data for dimensional modeling.

4.6 Summary

Snowflake serves as the analytical heart of the project. Raw football data is ingested from S3, cleaned in the staging layer, modeled into dimensions and facts in the core layer, and finally exposed through data marts for analytics and business intelligence tools.

Chapter 5

Transformation Layer: dbt (Data Build Tool)

5.1 Role of dbt

Once the raw football data has been ingested into Snowflake, the transformation phase is handled by **dbt (data build tool)**. dbt allows us to transform raw data into clean, structured and analytics-ready tables using SQL.

dbt is responsible for:

- cleaning and standardising raw data,
- building staging tables from Snowflake raw layer,
- generating dimension and fact tables,
- creating business-oriented data marts,
- applying data quality tests and documentation.

This makes dbt the central component of the transformation layer of the football analytics platform.

5.1.1 dbt-snowflake Installation

```
C:\Users\pc>cd C:\Users\pc\football_maroc

C:\Users\pc\football_maroc>dbt init football_maroc
10:44:28  Running with dbt=1.11.0-rc3
10:44:28
Your new dbt project "football_maroc" was created!

For more information on how to configure the profiles.yml file,
please consult the dbt documentation here:

  https://docs.getdbt.com/docs/configure-your-profile

One more thing:

Need help? Don't hesitate to reach out to us via GitHub issues or on Slack:

  https://community.getdbt.com/

Happy modeling!

10:44:28  Setting up your profile.
Which database would you like to use?
[1] snowflake

(Don't see the one you want? https://docs.getdbt.com/docs/available-adapters)

Enter a number: 1
```

Figure 5.1: dbt Install

5.1.2 dbt-snowflake Configuration

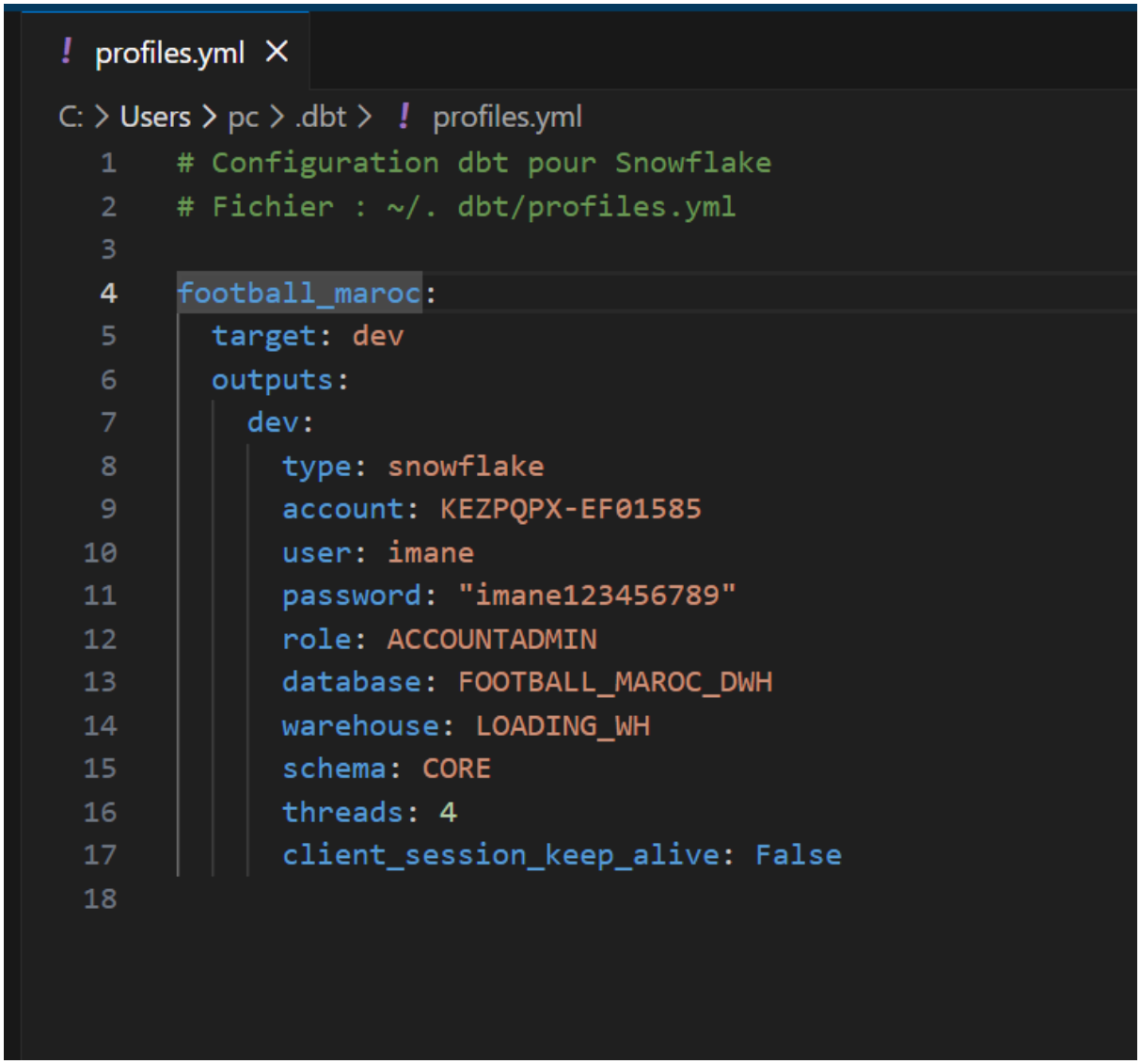


Figure 5.2: dbt Configuration

```
11:14:35  Connection:
11:14:35    account: KEZPQPX-EF01585
11:14:35    user: imane
11:14:35    database: FOOTBALL_MAROC_DWH
11:14:35    warehouse: COMPUTE_WH
11:14:35    role: ACCOUNTADMIN
11:14:35    schema: CORE
11:14:35    authenticator: None
11:14:35    oauth_client_id: None
11:14:35    query_tag: None
11:14:35    client_session_keep_alive: False
11:14:35    host: None
11:14:35    port: None
11:14:35    proxy_host: None
11:14:35    proxy_port: None
11:14:35    protocol: None
11:14:35    connect_retries: 1
11:14:35    connect_timeout: None
11:14:35    retry_on_database_errors: False
11:14:35    retry_all: False
11:14:35    insecure_mode: False
11:14:35    reuse_connections: True
11:14:35    s3_stage_vpce_dns_name: None
11:14:35    platform_detection_timeout_seconds: 0.0
11:14:35  Registered adapter: snowflake=1.10.4
11:14:40    Connection test: [OK connection ok]

11:14:40  All checks passed!

C:\Users\pc\football_maroc\football_maroc>
```

Figure 5.3: dbt Connexion succeded

5.2 Project Structure

The dbt project is organized into three main layers, which correspond to the folders:

- **staging** – light cleaning and normalization of raw data,
- **core** – creation of dimensions and fact tables,
- **marts** – aggregated tables for reporting and dashboards.

dbt Project Initialisation

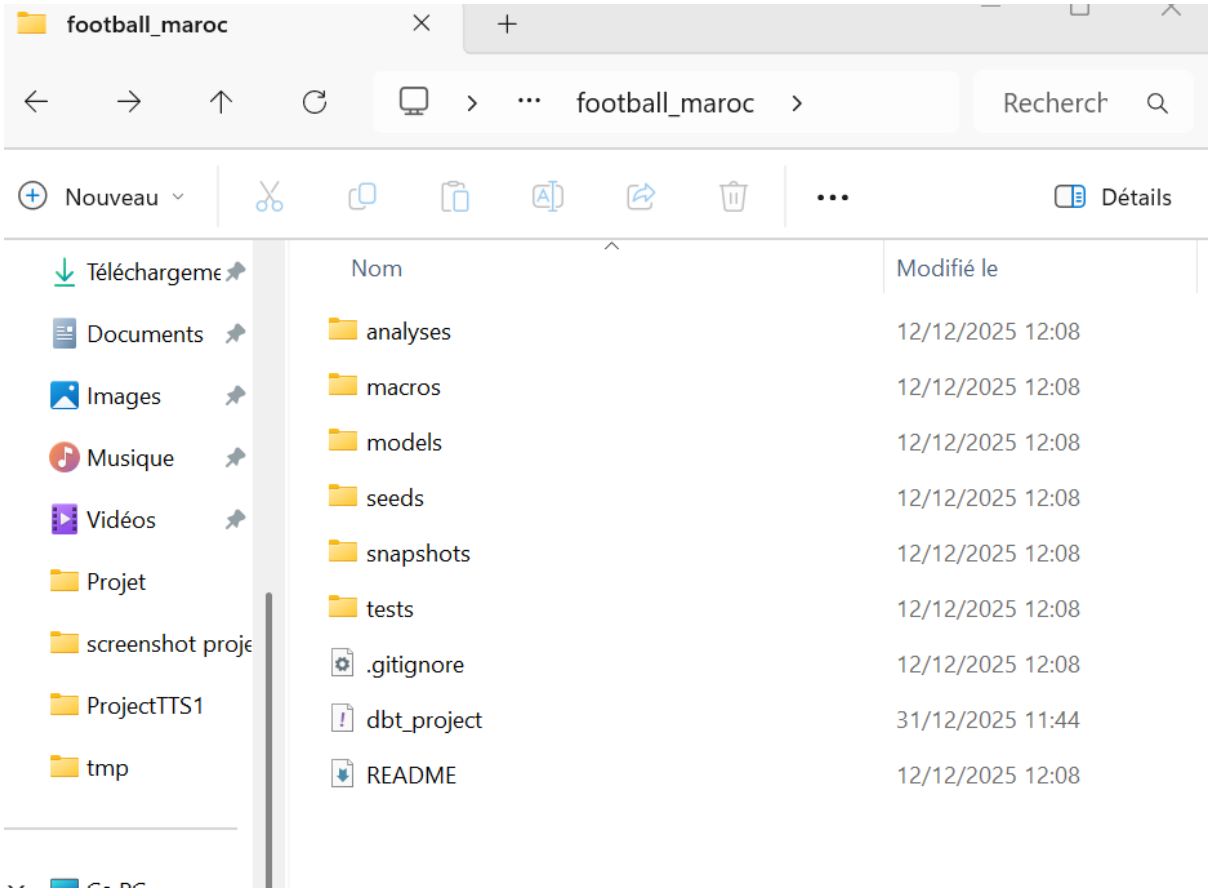


Figure 5.4: Initialisation of the dbt Project

5.3 Staging Layer

The **staging layer** contains models that are direct cleaned versions of the raw Snowflake tables. They mainly standardise field names, apply data types and remove duplicates.

Typical staging models in the project include:

- `stg_matches`
- `stg_players`
- `stg_match_player_stats`
- `stg_injuries`
- `stg_training_load`

```
11:14:35  Connection:
11:14:35    account: KEZPQPX-EF01585
11:14:35    user: imane
11:14:35    database: FOOTBALL_MAROC_DWH
11:14:35    warehouse: COMPUTE_WH
11:14:35    role: ACCOUNTADMIN
11:14:35    schema: CORE
11:14:35    authenticator: None
11:14:35    oauth_client_id: None
11:14:35    query_tag: None
11:14:35    client_session_keep_alive: False
11:14:35    host: None
11:14:35    port: None
11:14:35    proxy_host: None
11:14:35    proxy_port: None
11:14:35    protocol: None
11:14:35    connect_retries: 1
11:14:35    connect_timeout: None
11:14:35    retry_on_database_errors: False
11:14:35    retry_all: False
11:14:35    insecure_mode: False
11:14:35    reuse_connections: True
11:14:35    s3_stage_vpce_dns_name: None
11:14:35    platform_detection_timeout_seconds: 0.0
11:14:35  Registered adapter: snowflake=1.10.4
11:14:40    Connection test: [OK connection ok]

11:14:40  All checks passed!

C:\Users\pc\football_maroc\football_maroc>
```

Figure 5.5: Model Staging created successfully

These models form the foundation for dimensional modeling, We've already create them in the previous chapter.

5.4 Core Layer: Dimensions and Facts

The **core layer** implements the star schema inside Snowflake.

Dimension tables

Dimension tables describe football entities:

- `dim_teams` – team attributes (team name, league, country, stadium),
- `dim_players` – player information (age, position, nationality, current club),
- `dim_matches` – contextual match information (competition, round, date, venue).

They contain surrogate keys and descriptive attributes used for analysis.

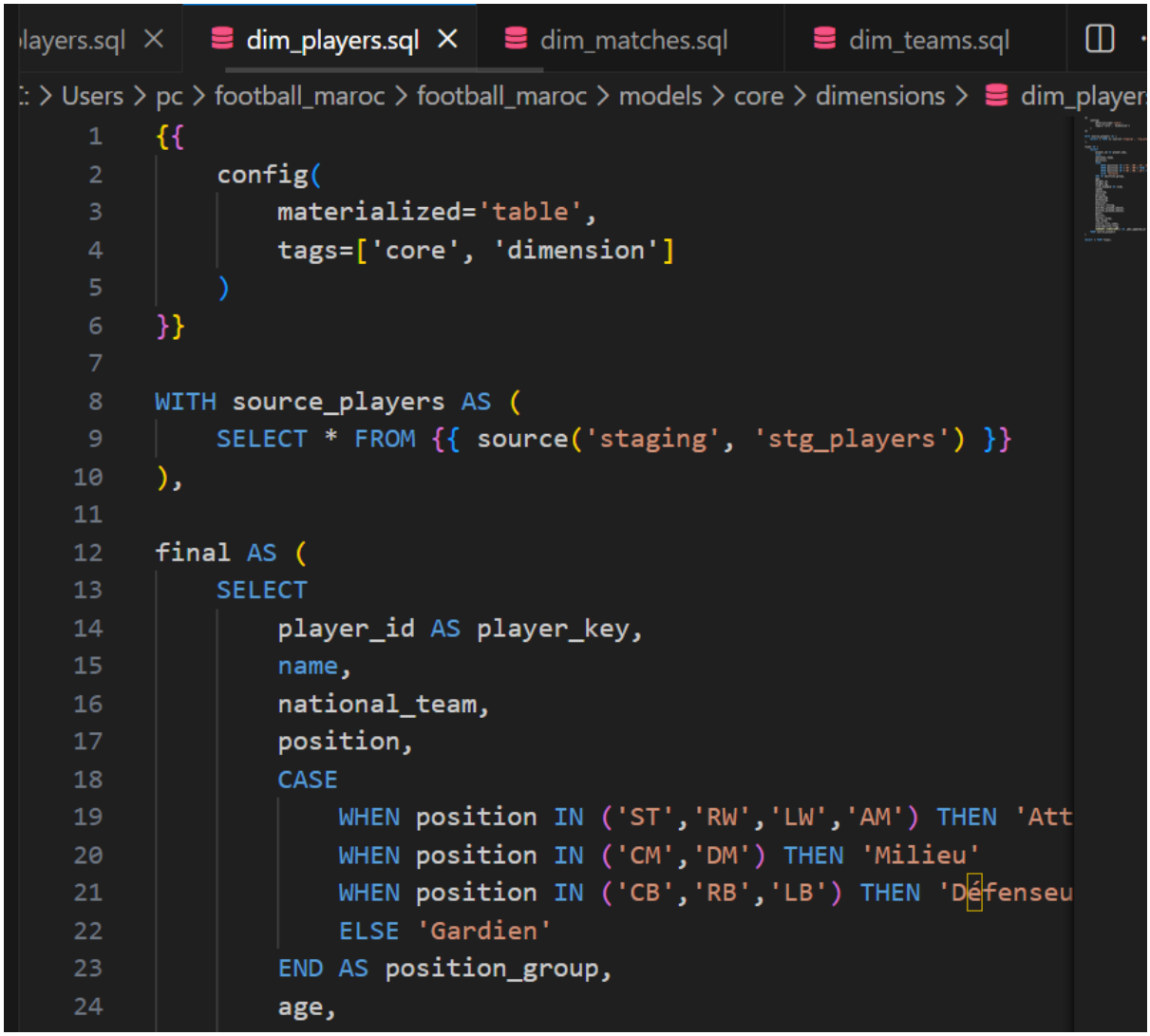


Figure 5.6: Code of dimensions Tables

Fact tables

Fact tables store measurable football events:

- `fact_match_stats` – goals, assists, cards, minutes played per player per match,
- `fact_injuries` – injuries per player with duration and injury type,
- `fact_training_load` – training sessions and workload metrics.

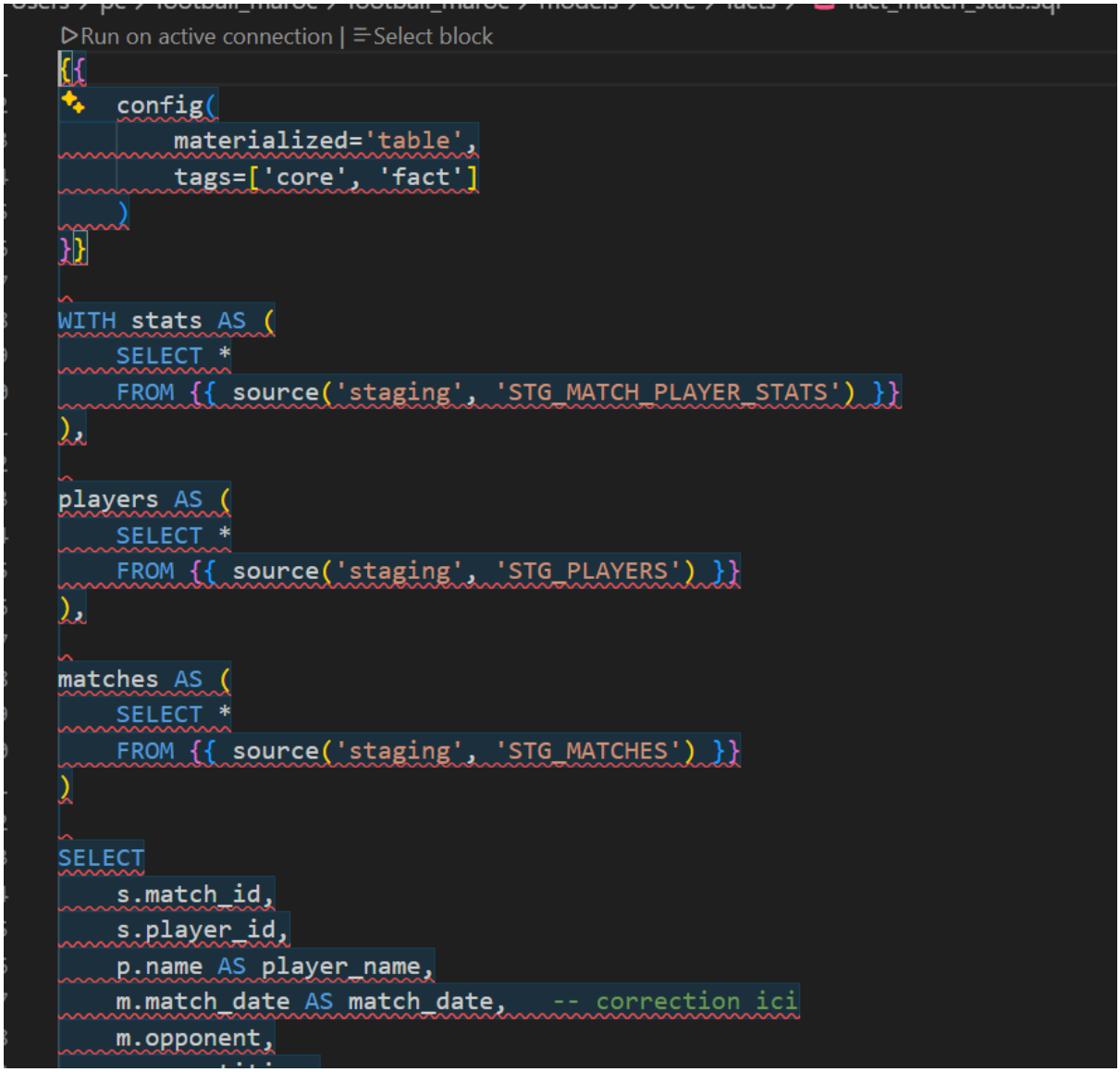


Figure 5.7: Example of code of Fact Tables

Facts are linked to dimensions through foreign keys, enabling analytical queries.

5.5 Marts Layer

The **marts layer** contains high-level aggregated models optimised for reporting tools such as Power BI.

Examples include:

- `mart_team_performance`
- `mart_players_performance`
- `mart_training_load_summary`
- `mart_player_injuries`

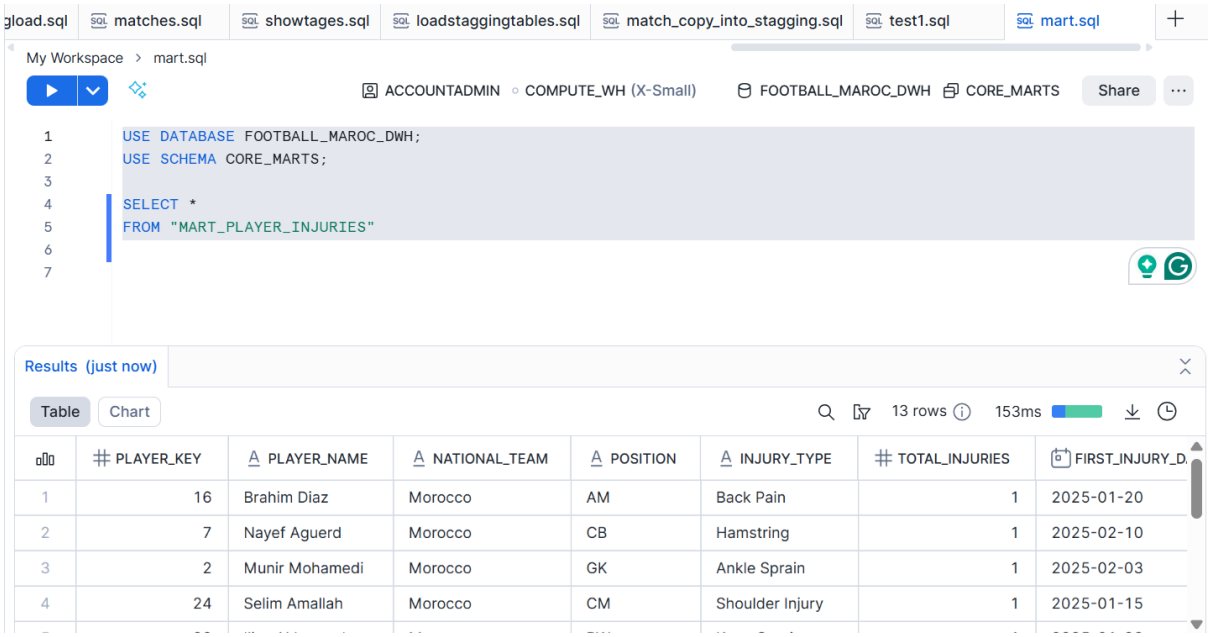


Figure 5.8: Example of Data_Mart : Injuries

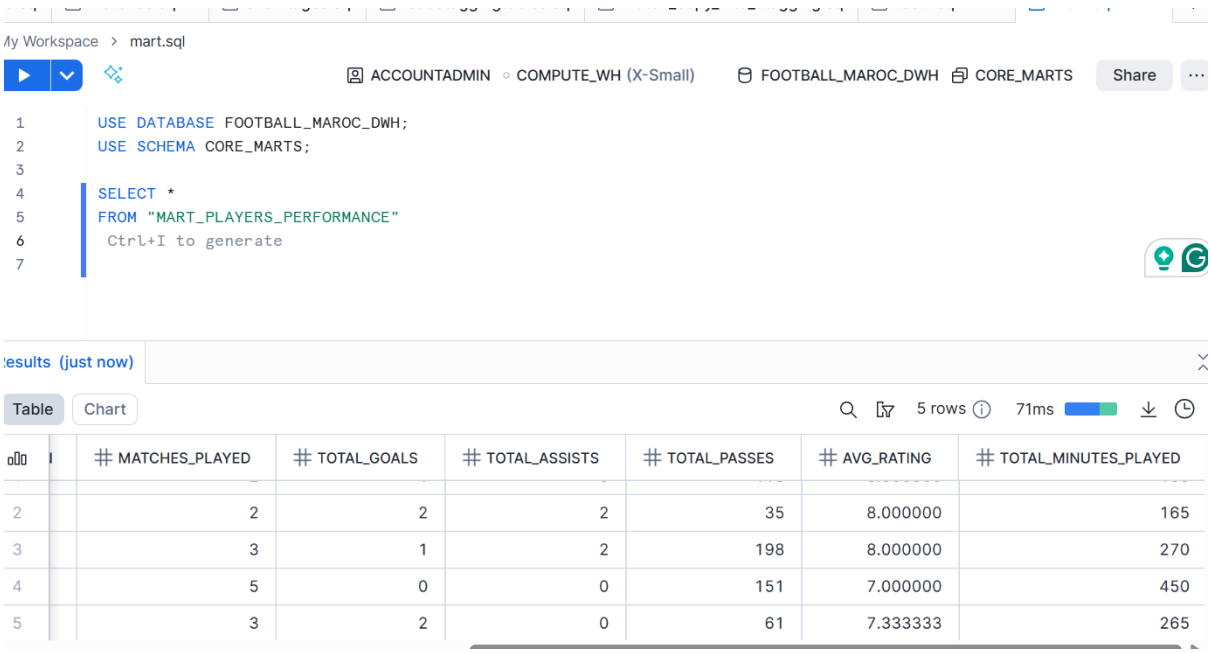


Figure 5.9: Example of Data/$_{M}art$: $Playersperformance$

These data marts answer business questions such as:

- Which players have the best performance index?
- How does team performance evolve across seasons?
- What is the relationship between workload and injury risk?

5.6 Data Quality and Documentation

dbt integrates automatic:

- uniqueness and non-null tests on keys,
- referential integrity tests between facts and dimensions,
- generated technical documentation,
- lineage graphs showing dependencies between models.

- dbt Execution :

```
14:56:05  4 of 13 START sql table model CORE.fact_injuries ............................ [RUN]
14:56:08  2 of 13 OK created sql table model CORE.dim_players ........................... [SUCCESS 1 in 2.12s]
14:56:08  3 of 13 OK created sql table model CORE.dim_teams ............................. [SUCCESS 1 in 2.15s]
14:56:08  4 of 13 OK created sql table model CORE.fact_injuries ......................... [SUCCESS 1 in 2.13s]
14:56:08  5 of 13 START sql table model CORE.fact_match_stats ........................... [RUN]
14:56:08  6 of 13 START sql table model CORE.fact_training_load ......................... [RUN]
14:56:08  1 of 13 OK created sql table model CORE.dim_matches ........................... [SUCCESS 1 in 2.18s]
14:56:08  7 of 13 START sql view model CORE.stg_injuries ................................ [RUN]
14:56:08  8 of 13 START sql view model CORE.stg_match_player_stats ...................... [RUN]
14:56:08  7 of 13 OK created sql view model CORE.stg_injuries ............................ [SUCCESS 1 in 0.61s]
14:56:08  8 of 13 OK created sql view model CORE.stg_match_player_stats .................. [SUCCESS 1 in 0.55s]
14:56:08  9 of 13 START sql view model CORE.stg_matches .................................. [RUN]
14:56:08  10 of 13 START sql view model CORE.stg_players ................................. [RUN]
14:56:09  6 of 13 OK created sql table model CORE.fact_training_load ..................... [SUCCESS 1 in 0.94s]
14:56:09  5 of 13 OK created sql table model CORE.fact_match_stats ....................... [SUCCESS 1 in 0.95s]
14:56:09  11 of 13 START sql view model CORE.stg_training_load ........................... [RUN]
14:56:09  12 of 13 START sql table model CORE_MARTS.mart_player_injuries ................. [RUN]
14:56:09  9 of 13 OK created sql view model CORE.stg_matches ............................. [SUCCESS 1 in 0.34s]
14:56:09  10 of 13 OK created sql view model CORE.stg_players ........................... [SUCCESS 1 in 0.33s]
14:56:09  13 of 13 START sql table model CORE_MARTS.mart_players_performance ............. [RUN]
14:56:09  11 of 13 OK created sql view model CORE.stg_training_load ...................... [SUCCESS 1 in 0.36s]
14:56:09  12 of 13 OK created sql table model CORE_MARTS.mart_player_injuries ............ [SUCCESS 1 in 0.74s]
14:56:10  13 of 13 OK created sql table model CORE_MARTS.mart_players_performance ........ [SUCCESS 1 in 0.86s]
14:56:11
14:56:11  Finished running 8 table models, 5 view models in 0 hours 0 minutes and 9.55 seconds (9.55s).
14:56:11
14:56:11  Completed successfully
14:56:11
14:56:11  Done. PASS=13 WARN=0 ERROR=0 SKIP=0 NO-OP=0 TOTAL=13
```

Figure 5.10: dbt Execution

```
C:\Users\pc\football_maroc\football_maroc>dbt test
15:02:29  Running with dbt=1.11.0-rc3
15:02:32  Registered adapter: snowflake=1.10.4
15:02:35  [WARNING][MissingArgumentsPropertyInGenericTestDeprecation]: Deprecated
functionality
Found top-level arguments to test `relationships` defined on 'fact_match_stats'
in package 'football_maroc' (models\core\schema.yaml). Arguments to generic
tests should be nested under the `arguments` property.
15:02:36  Found 13 models, 10 data tests, 5 sources, 504 macros
15:02:36
15:02:36  Concurrency: 4 threads (target='dev')
15:02:36
15:02:37  1 of 10 START test not_null_dim_matches_match_key .............................. [RUN]
15:02:37  2 of 10 START test not_null_dim_players_player_key ............................. [RUN]
15:02:37  4 of 10 START test not_null_mart_player_injuries_player_key .................... [RUN]
15:02:37  3 of 10 START test not_null_fact_match_stats_match_id .......................... [RUN]
15:02:38  2 of 10 PASS not_null_dim_players_player_key ................................... [PASS in 1.15s]
15:02:38  1 of 10 PASS not_null_dim_matches_match_key .................................... [PASS in 1.16s]
15:02:38  3 of 10 PASS not_null_fact_match_stats_match_id ................................ [PASS in 1.17s]
15:02:38  5 of 10 START test not_null_mart_player_injuries_total_injuries ................ [RUN]
15:02:38  6 of 10 START test not_null_mart_players_performance_competition ............... [RUN]
15:02:38  7 of 10 START test not_null_mart_players_performance_player_key ................ [RUN]
15:02:38  4 of 10 PASS not_null_mart_player_injuries_player_key .......................... [PASS in 1.27s]
15:02:38  8 of 10 START test relationships_fact_match_stats_player_id__player_key__ref_dim_players_  [RUN]
15:02:38  5 of 10 PASS not_null_mart_player_injuries_total_injuries ...................... [PASS in 0.27s]
15:02:38  9 of 10 START test unique_dim_matches_match_key ................................ [RUN]
15:02:38  7 of 10 PASS not_null_mart_players_performance_player_key ...................... [PASS in 0.29s]
15:02:38  6 of 10 PASS not_null_mart_players_performance_competition ..................... [PASS in 0.31s]
15:02:38  10 of 10 START test unique_dim_players_player_key ............................... [RUN]
```

Figure 5.11: dbt test

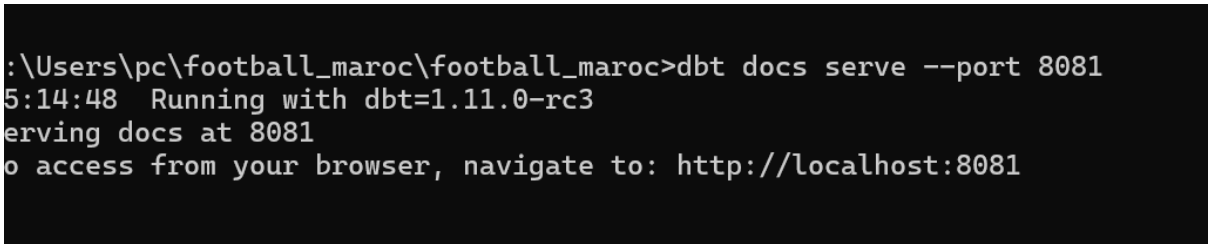


Figure 5.12: dbt Server launch

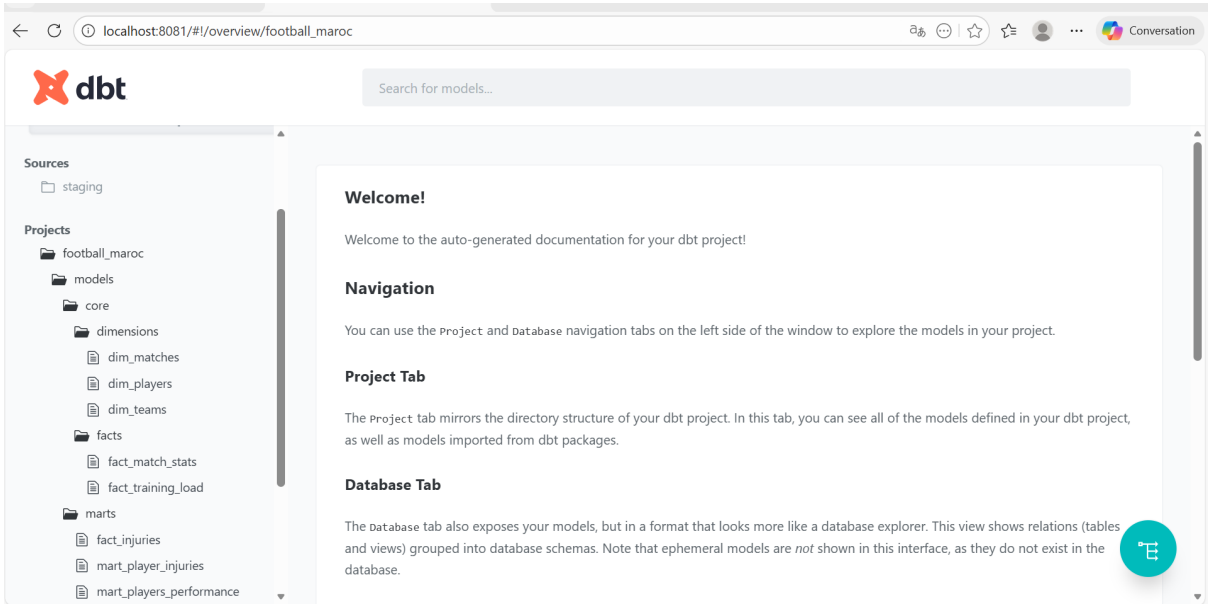


Figure 5.13: dbt Website

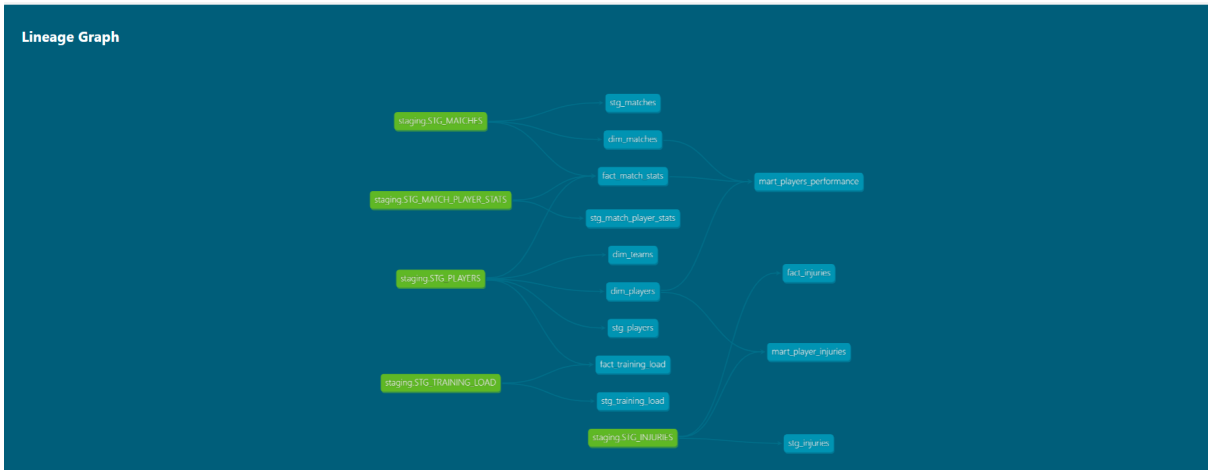


Figure 5.14: Visualisation du Dag dans dbt Website

This guarantees reliable football analytics and full transparency of the transformation pipeline.

Chapter 6

Workflow Orchestration: Apache Airflow

6.1 Role of Apache Airflow in the Football Data Pipeline

Up to this stage, the execution of each processing step was done manually. Apache Airflow is introduced in the project to **orchestrate the entire football data pipeline automatically**.

Airflow schedules, executes and monitors the end-to-end workflow that transports data from operational systems to analytical dashboards.

The orchestrated pipeline covers:

- extraction of football data from PostgreSQL,
- loading of raw files into Amazon S3 (data lake),
- ingestion of data from S3 into Snowflake staging layer,
- transformation of data with dbt (facts and dimensions),
- refresh of BI dashboards.

Airflow therefore acts as the **central control plane** of the architecture.

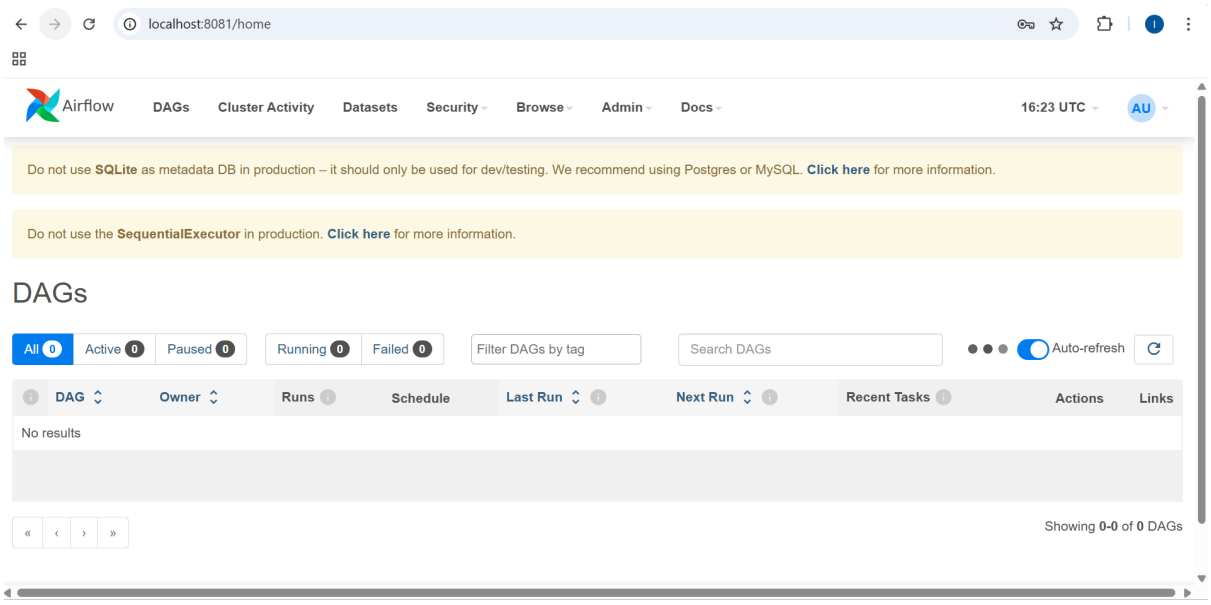


Figure 6.1: Connexion to Air-Flow

6.2 What is Apache Airflow?

Apache Airflow is an open-source platform designed to define, schedule and monitor workflows. Workflows are represented as **Directed Acyclic Graphs (DAGs)**, where each node is a task and each edge defines a dependency between tasks.

Core concepts

- **DAG** – representation of the complete football data pipeline,
- **Task** – processing step (e.g., extraction, loading, transformation),
- **Operator** – predefined task template (Python, Bash, Snowflake, etc.),
- **Sensor** – waits for the availability of a resource (e.g., file in S3),
- **Schedule** – execution frequency (daily football data refresh),
- **Executor** – execution backend running the tasks.

In this project, Airflow automates the daily football data workflow:

1. extract match, player and team data from PostgreSQL;
2. export the data as CSV files to Amazon S3;
3. detect the arrival of files in S3 using S3 sensors;
4. load S3 files into Snowflake staging tables;
5. launch dbt transformations to build facts and dimensions;
6. update aggregated football performance data marts;

7. log execution history and monitor failures.

This ensures that analytical data is always up-to-date without manual intervention.

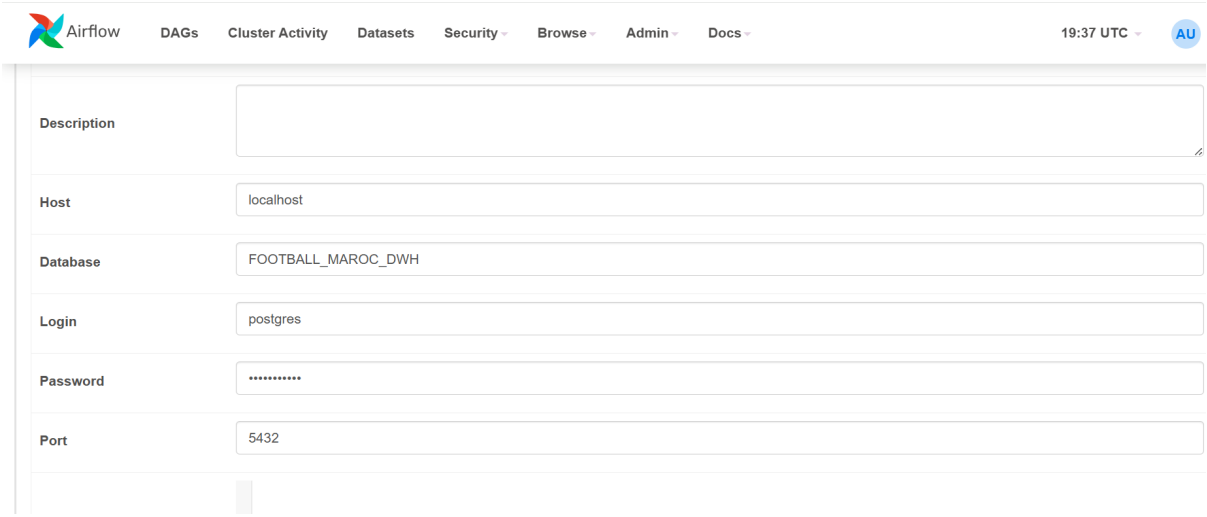


Figure 6.2: Connexion to Air-Flow

6.3 Airflow Connections

To securely interact with external systems, several Airflow connections are configured:

- **AWS connection** – access to Amazon S3 bucket,
- **Snowflake connection** – loading staging tables and running SQL,
- **PostgreSQL connection** – extracting operational football data.

Each connection stores credentials in Airflow's metadata database instead of code, improving security and maintainability.

6.4 Football Data Pipeline DAG

A dedicated DAG was created to orchestrate the daily football data refresh. It executes the following main phases:

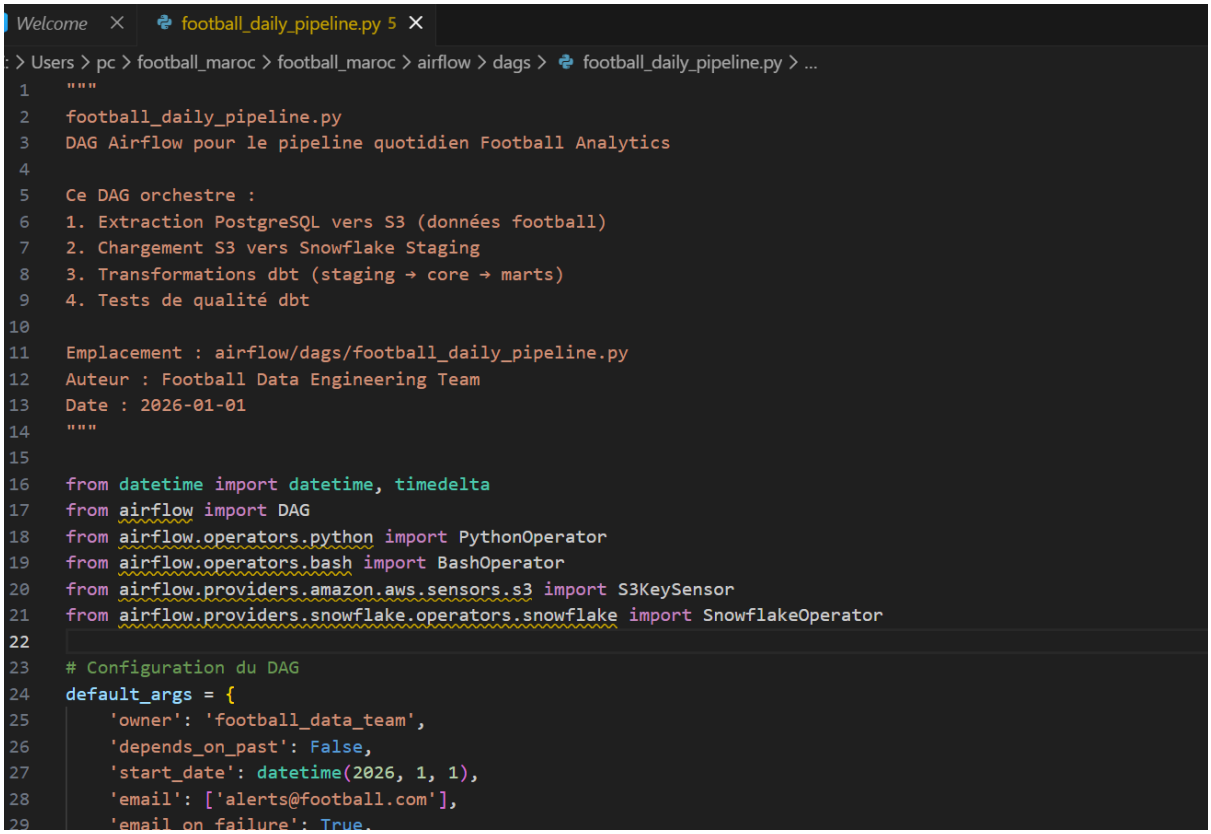


Figure 6.3: Code of dag creation

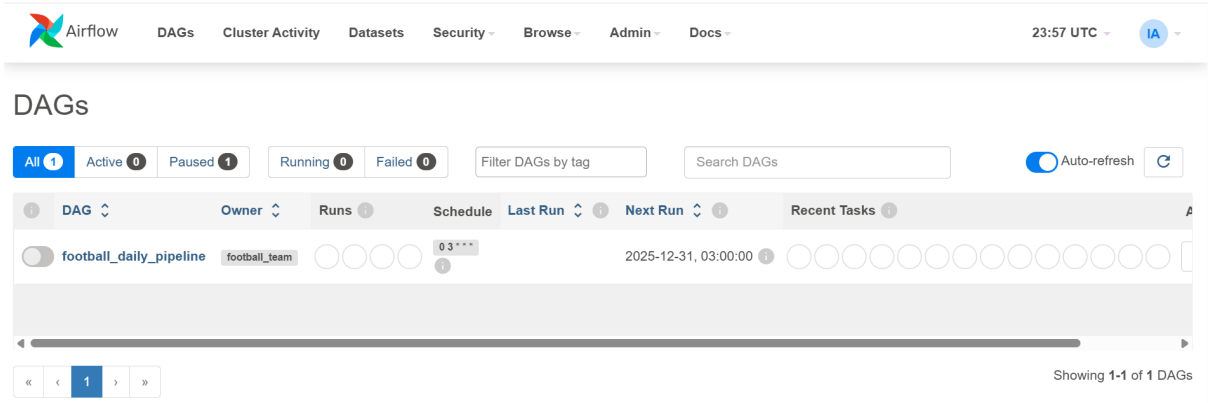


Figure 6.4: Dag of Football Data pipeline creation

Phase 1 – Data Extraction

PostgreSQL tables such as:

- matches,
- players,
- teams,
- match_player_stats,

- training and injuries tables

are exported to CSV files and written to Amazon S3.

Phase 2 – S3 Monitoring

Airflow sensors wait until exported files are available in the data lake before continuing.

Phase 3 – Snowflake Loading

The data is copied from S3 into Snowflake STAGING schemas using bulk COPY INTO operations.

Phase 4 – Transformations with dbt

dbt is triggered to:

- build dimension tables (players, teams, matches),
- build fact tables (match statistics, injuries, training load),
- generate marts used by BI tools.

Phase 5 – Monitoring and Notifications

Airflow provides:

- execution logs for each task,
- graphical view of the DAG,
- alerting if a task fails,
- historical run tracking.

This guarantees reliability and reproducibility of the football analytics platform.

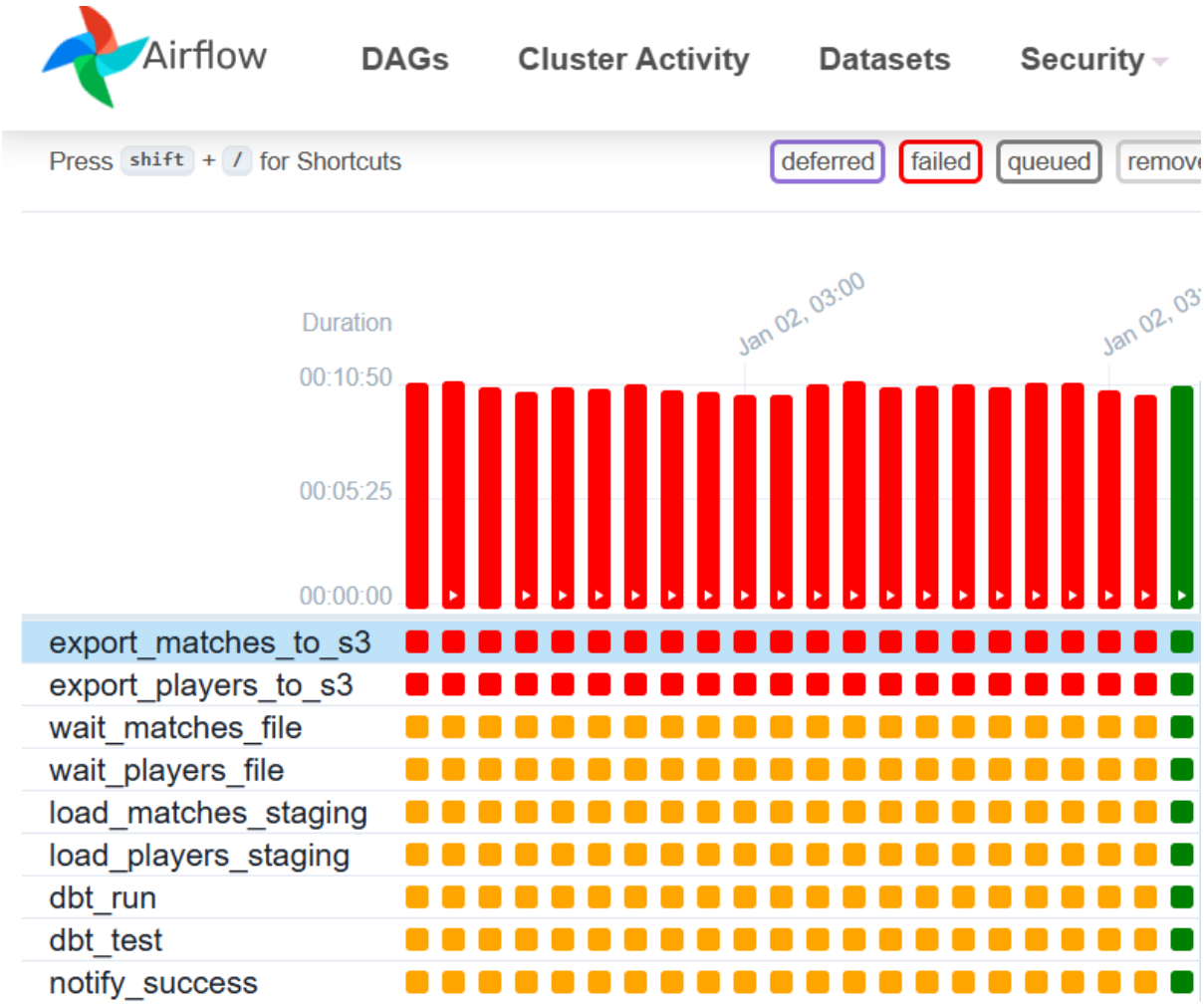


Figure 6.5: Dag Execution

The Airflow DAG is configured to run automatically every day. The scheduling policy ensures that:

- match results are updated frequently,
- player statistics remain current,
- dashboards reflect the latest competitions.

Thus, Airflow transforms the data pipeline into a fully automated production workflow.

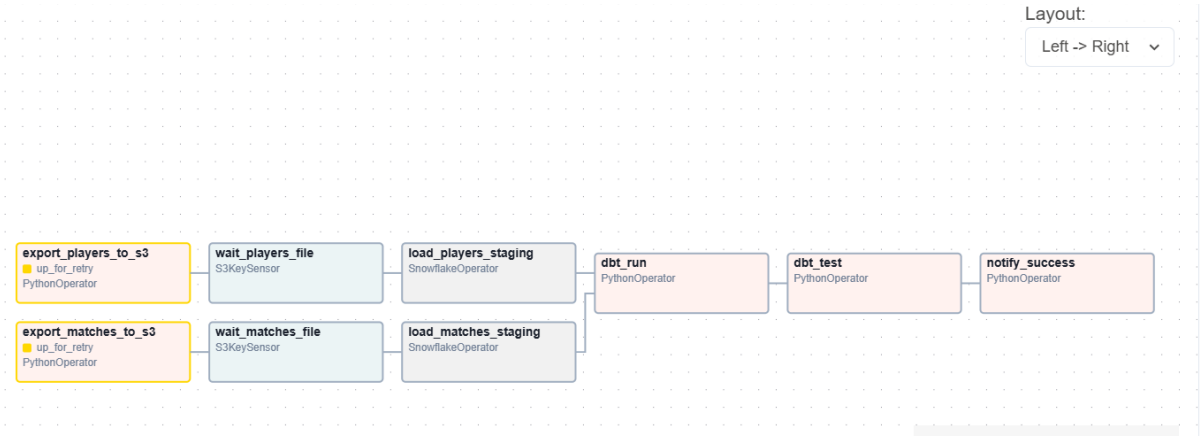


Figure 6.6: Dag Pipeline

Chapter 7

Intégration Power BI et Snowflake

7.1 Configuration de la connexion Snowflake

Pour connecter Power BI à notre entrepôt de données Snowflake, nous utilisons le connecteur natif disponible dans Power BI Desktop. La boîte de dialogue 'Obtenir les données' permet de rechercher et sélectionner Snowflake parmi les sources de données disponibles. Cette étape initiale est cruciale pour établir la connexion avec notre data warehouse cloud et accéder aux tables analytiques préparées.

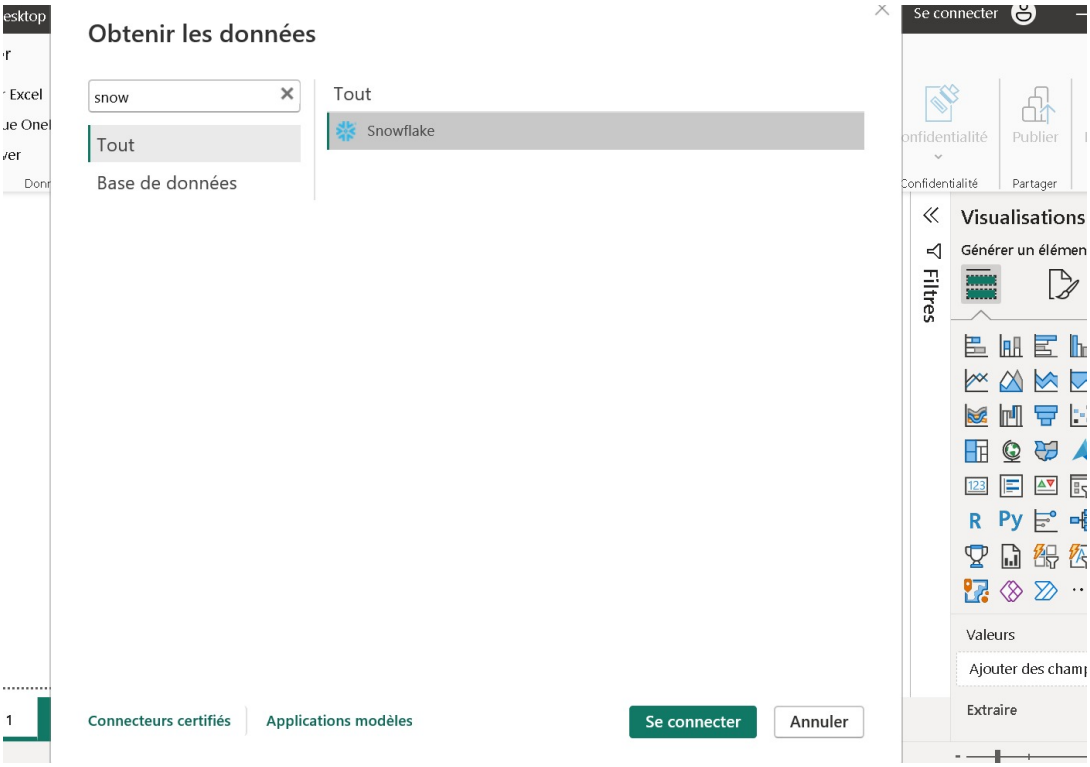


Figure 7.1: Boîte de dialogue de connexion à Snowflake

7.1.1 Paramétrage de la connexion

La configuration finale de la connexion nécessite deux paramètres essentiels :

- **Server** : l'URL complète du compte Snowflake `zd83817.eu-north-1.aws.snowflakecomputing.com`
- **Warehouse** : le nom du warehouse virtuel `BI_WH` qui exécutera les requêtes

Le warehouse `BI_WH` est dimensionné pour supporter les charges analytiques de Power BI. Les options avancées permettent de configurer des paramètres supplémentaires comme le rôle d'utilisateur ou le schéma par défaut. Une fois validée, cette connexion permet à Power BI d'importer ou d'interroger directement les données via DirectQuery.

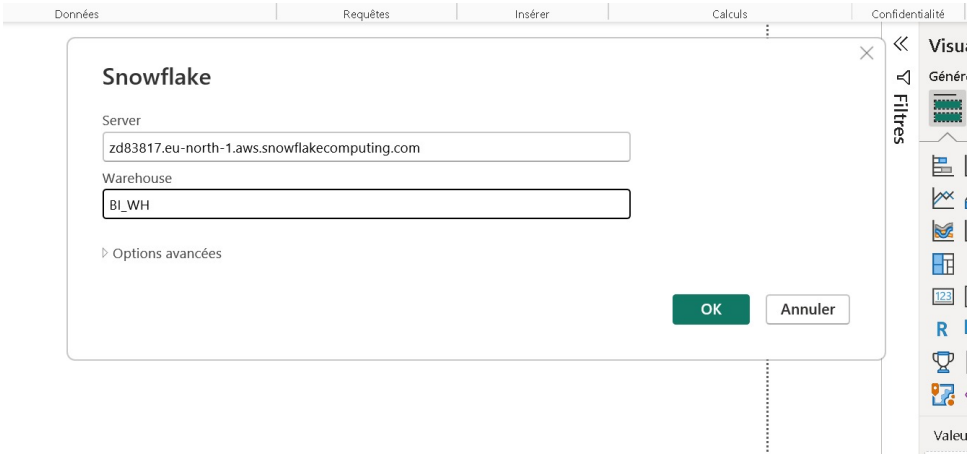


Figure 7.2: Configuration des paramètres de connexion Snowflake

7.2 Navigation et sélection des données

Cette organisation en couches reflète une architecture de données moderne et structurée selon les bonnes pratiques de data warehousing, facilitant la maintenance et l'évolutivité du système.

7.2.1 Prévisualisation des données

La table `MART_TRAINING_LOAD_SUMMARY` sélectionnée affiche les données d'entraînement des joueurs, incluant :

- `PLAYER_ID` : Identifiant unique du joueur
- `PLAYER_NAME` : Nom complet du joueur
- `TRAINING_WEEK` : Semaine d'entraînement (format date)
- `TOTAL_DURATION` : Durée totale d'entraînement en minutes
- `AVG_RPE` : Charge moyenne perçue (Rating of Perceived Exertion)

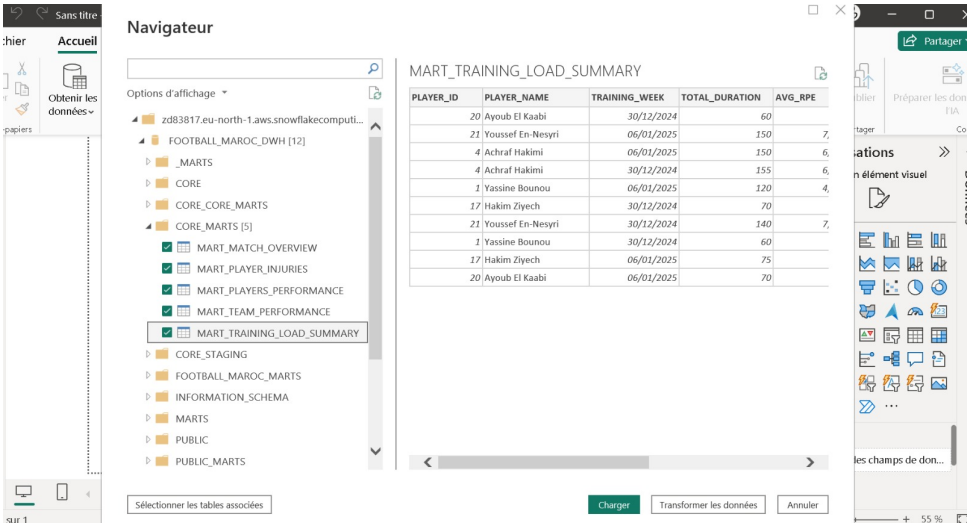


Figure 7.3: Connexion a DataMart de Snowflake

Cette prévisualisation permet de valider la qualité des données avant leur importation dans Power BI.

7.3 Création du Dashboard CAN

Le dashboard principal offre une vue d'ensemble complète des performances de la Coupe d'Afrique des Nations (CAN). Il présente plusieurs visualisations clés organisées de manière ergonomique et intuitive.

Figure 7.4: Dashboard CAN - Vue d'ensemble des performances

7.3.1 Composants du tableau de bord

Le dashboard se compose des éléments suivants :

Indicateurs clés (KPIs)

- **Total des buts** : 6,00 buts marqués
- **Joueurs impliqués** : 15 joueurs
- **Premier pays** : Maroc identifié comme équipe principale

Visualisations analytiques

1. **Diagramme circulaire - Disponibilité des joueurs**
   Représente la répartition du ratio de disponibilité par nom de joueur, permettant d'identifier rapidement les joueurs les plus disponibles pour les sélections.
2. **Graphique à barres - Comparaison buts/tirs par équipe**
   Compare le nombre total de buts marqués et le nombre total de tirs effectués par équipe nationale (Sénégal, Tunisie, Maroc, Cameroun, Ghana), offrant une vision de l'efficacité offensive.
3. **Graphique linéaire - Évolution temporelle**
   Montre l'évolution des buts totaux par équipe au fil du temps, permettant d'identifier les tendances de performance.
4. **Tableau détaillé des joueurs marocains**
   Liste exhaustive des joueurs avec les colonnes suivantes :
   - `NATIONAL_TEAM` : Équipe nationale (Morocco)
   - `PLAYER_NAME` : Nom du joueur
   - `POSITION` : Poste (GK, RB, RW, ST)
   - `TOTAL_GOALS` : Nombre de buts marqués
   - `MATCHES_PLAYED` : Nombre de matchs disputés
5. **Graphique par compétition**
   Affiche le nombre de matchs, joueurs impliqués et buts totaux pour la compétition AFCON 2025.

7.3.2 Fonctionnalités interactives

Le dashboard intègre plusieurs fonctionnalités interactives :

- **Filtres dynamiques** : Possibilité de filtrer par équipe, joueur ou période
- **Interactivité croisée** : Les sélections sur un graphique filtrent automatiquement les autres visualisations
- **Interface conversationnelle** : Section "Poser une question sur vos données" permettant des requêtes en langage naturel
- **Navigation intuitive** : Onglets organisés (Calculs, Calendriers, Actualisation, Paramètres, Sécurité)

7.3.3 Cas d'usage

Ce tableau de bord permet aux analystes sportifs de :

- Identifier rapidement les performances individuelles et collectives
- Comparer l'efficacité offensive entre différentes équipes nationales
- Suivre l'évolution des performances au fil de la compétition
- Analyser la disponibilité et l'utilisation des joueurs
- Prendre des décisions data-driven pour la sélection et la stratégie d'équipe

7.3.4 Avantages de l'intégration Snowflake-Power BI

L'intégration entre Snowflake et Power BI offre plusieurs avantages significatifs :

Avantage	**Description**
Performance	Snowflake permet des requêtes rapides sur de grands volumes de données grâce à son architecture de séparation compute/storage
Scalabilité	Le warehouse peut être dimensionné dynamiquement selon les besoins analytiques
Sécurité	Authentification robuste et gestion fine des droits d'accès
Actualisation	Les données peuvent être rafraîchies automatiquement selon un planning défini
Gouvernance	Traçabilité complète des accès et des modifications

Table 7.1: Avantages de l'intégration Snowflake-Power BI

Chapter 8

Conclusion

8.1 Synthèse du projet

Ce projet a permis de concevoir et de mettre en œuvre un pipeline de données moderne et complet pour l'analyse des performances footballistiques, en prenant comme cas d'étude les données du football marocain et de la Coupe d'Afrique des Nations (CAN).

L'architecture développée intègre l'ensemble des composants essentiels d'une plateforme analytique cloud moderne :

- **PostgreSQL** comme système de gestion de base de données opérationnelles, stockant les données brutes des matchs, joueurs, blessures et entraînements
- **Amazon S3** comme data lake centralisé, assurant le stockage scalable et économique des données exportées
- **Snowflake** comme entrepôt de données cloud, permettant des analyses performantes sur de grands volumes
- **dbt** pour la transformation et la modélisation des données selon une architecture en couches (staging, core, marts)
- **Apache Airflow** pour l'orchestration automatisée du pipeline de bout en bout
- **Power BI** pour la visualisation interactive et la production de tableaux de bord décisionnels

8.2 Objectifs atteints

Les objectifs initiaux du projet ont été pleinement réalisés :

Architecture cloud moderne et scalable L'infrastructure mise en place repose sur des technologies cloud éprouvées (AWS S3, Snowflake) qui garantissent la scalabilité, la haute disponibilité et la sécurité des données. L'architecture permet de traiter des volumes croissants de données sans dégradation de performance.

Automatisation complète du pipeline Grâce à Apache Airflow, l'ensemble du processus d'ingestion, transformation et publication des données est automatisé. Le pipeline s'exécute quotidiennement sans intervention manuelle, assurant la fraîcheur des données analytiques.

Centralisation des statistiques footballistiques Toutes les données relatives aux performances des équipes et joueurs sont consolidées dans un référentiel unique (Snowflake), facilitant l'accès et garantissant la cohérence des analyses.

Données prêtes pour l'analyse décisionnelle La modélisation dimensionnelle (schéma en étoile) et la création de data marts spécialisés permettent aux analystes sportifs d'accéder facilement aux indicateurs de performance via des dashboards Power BI interactifs.

8.3 Apports et bénéfices

Ce projet démontre concrètement comment une architecture de données moderne peut transformer la gestion et l'analyse des performances sportives :

- **Prise de décision data-driven** : Les entraîneurs et analystes disposent d'indicateurs fiables pour optimiser les sélections, gérer la charge d'entraînement et prévenir les blessures
- **Réduction du temps d'analyse** : L'automatisation du pipeline libère les analystes des tâches de préparation de données, leur permettant de se concentrer sur l'interprétation des résultats
- **Qualité et traçabilité** : Les tests automatiques de dbt et la documentation générée garantissent la fiabilité des données et facilitent la maintenance
- **Évolutivité** : L'architecture modulaire permet d'intégrer facilement de nouvelles sources de données (données GPS, analyses vidéo, données médicales)
- **Gouvernance et sécurité** : La gestion des accès via IAM (AWS) et les rôles Snowflake assure la protection des données sensibles

8.4 Limites et perspectives d'amélioration

Bien que le projet soit fonctionnel et réponde aux objectifs fixés, plusieurs axes d'amélioration peuvent être envisagés :

8.4.1 Limites actuelles

- **Périmètre des données** : Le projet se concentre principalement sur les données structurées (statistiques de matchs, entraînements). L'intégration de données non structurées (vidéos, images) n'est pas encore implémentée

- **Analyses prédictives** : Le pipeline actuel se limite à des analyses descriptives. Les modèles de machine learning pour prédire les performances ou les risques de blessure ne sont pas encore développés
- **Temps réel** : L'actualisation quotidienne est suffisante pour la plupart des analyses, mais certains cas d'usage (suivi en direct des matchs) nécessiteraient un traitement en streaming
- **Monitoring avancé** : Bien qu'Airflow fournisse des alertes de base, un système de monitoring plus complet (métriques de qualité de données, SLA) pourrait être mis en place

8.4.2 Perspectives d'évolution

Plusieurs pistes prometteuses peuvent être explorées pour enrichir la plateforme :

Intégration de l'intelligence artificielle

- Modèles de prédiction des performances individuelles et collectives
- Algorithmes de détection des risques de blessure basés sur la charge d'entraînement
- Systèmes de recommandation pour l'optimisation des compositions d'équipe
- Analyse vidéo automatisée pour extraire des métriques tactiques avancées

Extension des sources de données

- Intégration de données GPS et tracking (vitesse, distance parcourue, zones de terrain)
- Connexion avec des API externes (réseaux sociaux, médias, données météo)
- Données biométriques et médicales pour un suivi holistique des joueurs
- Analyses adverses pour la préparation tactique

Amélioration de l'infrastructure

- Mise en place d'un streaming pipeline pour l'analyse en temps réel
- Déploiement de dbt Cloud pour faciliter la collaboration entre data engineers
- Implémentation de tests de performance et d'optimisation des requêtes
- Mise en place d'un data catalog pour améliorer la découvrabilité des données

Enrichissement des analyses

- Développement de dashboards spécialisés par rôle (entraîneur, médecin, recruteur)
- Analyses comparatives avec d'autres équipes et compétitions
- Indicateurs avancés (xG, PPDA, pressing intensity)
- Rapports automatisés post-match et pré-match

8.5 Compétences acquises

La réalisation de ce projet a permis de développer des compétences techniques et méthodologiques essentielles en data engineering et data analytics :

- Maîtrise des technologies cloud (AWS S3, Snowflake)
- Compétences en modélisation de données (schéma en étoile, architecture en couches)
- Pratique de l'orchestration de workflows avec Apache Airflow
- Transformation de données avec dbt et SQL avancé
- Visualisation de données avec Power BI
- Gestion de projet data et architecture de solutions analytiques
- Compréhension des bonnes pratiques en data engineering (tests, documentation, versioning)

8.6 Conclusion générale

Ce projet démontre qu'une architecture de données moderne, bien conçue et correctement implémentée, peut transformer radicalement la manière dont les organisations sportives exploitent leurs données. En automatisant l'ingestion, la transformation et la publication des données, nous avons suit une pipeline robuste qui permet aux analystes de se concentrer sur ce qui compte vraiment : générer des insights actionnables pour améliorer les performances sportives.

L'approche modulaire et scalable adoptée garantit que cette infrastructure pourra évoluer avec les besoins futurs, qu'il s'agisse d'intégrer de nouvelles sources de données, d'ajouter des modèles prédictifs avancés, ou d'étendre l'analyse à d'autres équipes et compétitions.

Au-delà des aspects techniques, ce projet illustre l'importance croissante de la data dans le sport professionnel moderne. Les équipes qui sauront exploiter efficacement leurs données disposeront d'un avantage concurrentiel significatif, tant pour l'optimisation des performances que pour la prévention des blessures et la gestion des carrières des joueurs.

Finalement, cette expérience confirme que les méthodologies et outils du data engineering moderne sont parfaitement applicables au domaine sportif, ouvrant la voie à une nouvelle génération d'analyses permettant de prendre des décisions plus éclairées, plus rapides et plus précises dans un environnement hautement compétitif.