

Database Design Report: Football Data Warehouse

Author: Tran Hung Vi
Date: October 18, 2025
Version: 1.0

1.0 Introduction

This document provides a comprehensive overview of the relational database schema designed to store and manage football data from the API-Football service. The objective of this design is to create a structured, scalable, and efficient data warehouse that accurately models the relationships between various football entities such as countries, leagues, teams, players, and matches. This report details each table, its columns, and the critical relationships that connect the schema.

2.0 Table Definitions

This section describes the purpose and structure of each table in the database.

2.1 countries

- Purpose:** Stores a unique list of all countries. This is a foundational table referenced by leagues, teams, and venues.

Column	Type	Description
--------	------	-------------

country_id	INT (PK)	The unique identifier for the country.
name	VARCHAR	The common name of the country (e.g., "England").
code	VARCHAR	The official country code (e.g., "GB").
flag_url	VARCHAR	A URL pointing to an image of the country's flag.

2.2 leagues

- **Purpose:** Stores information about each football league or competition.

Column	Type	Description
league_id	INT (PK)	The unique identifier for the league.
name	VARCHAR	The name of the league (e.g., "Premier League").
country_id	INT (FK)	References countries.country_id.
logo_url	VARCHAR	A URL to the league's official logo.

2.3 seasons

- **Purpose:** Defines the specific seasons for each league, including start and end dates.

Column	Type	Description
league_id	INT (PK, FK)	Part of the composite key; references leagues.league_id.
year	INT (PK)	Part of the composite key; the starting year of the season.
start_date	DATE	The official start date of the season.
end_date	DATE	The official end date of the season.

2.4 venues

- **Purpose:** Stores details about the stadiums or venues where matches are played.

Column	Type	Description
venue_id	INT (PK)	The unique identifier for the venue.
name	VARCHAR	The name of the stadium (e.g., "Old Trafford").
city	VARCHAR	The city where the venue is located.
country_id	INT (FK)	References countries.country_id.
capacity	INT	The seating capacity of the venue.

2.5 teams

- **Purpose:** Contains information about each football team, both club and national.

Column	Type	Description
team_id	INT (PK)	The unique identifier for the team.
name	VARCHAR	The name of the team (e.g., "Manchester United").
country_id	INT (FK)	References countries.country_id.
venue_id	INT (FK)	The team's home stadium; references venues.venue_id.
is_national	BOOLEAN	A flag indicating if it's a national team.

2.6 players

- **Purpose:** Stores core, static biographical information for each player.

Column	Type	Description
player_id	INT (PK)	The unique identifier for the player.
first_name	VARCHAR	The player's first name.

last_name	VARCHAR	The player's last name.
birth_date	DATE	The player's date of birth.
nationality	VARCHAR	The player's country of citizenship.

2.7 player_team_history

- **Purpose:** A crucial linking table that tracks a player's career by mapping them to teams for specific periods.

Column	Type	Description
history_id	INT (PK)	The unique identifier for this career entry.
player_id	INT (FK)	References players.player_id.
team_id	INT (FK)	References teams.team_id.
start_date	DATE	The date the player joined the team.
end_date	DATE	The date the player left the team (NULL if current).

2.8 fixtures

- **Purpose:** Stores all information related to a single match.

Column	Type	Description
--------	------	-------------

fixture_id	INT (PK)	The unique identifier for the match.
league_id	INT (FK)	References leagues.league_id.
season_year	INT (FK)	The season in which the match was played.
match_date	DATETIME	The specific date and time of the match.
home_team_id	INT (FK)	The home team; references teams.team_id.
away_team_id	INT (FK)	The away team; references teams.team_id.
home_score	INT	The final score for the home team.
away_score	INT	The final score for the away team.

2.9 player_match_stats

- **Purpose:** Records the performance statistics of a single player in a single match.

Column	Type	Description
fixture_id	INT (PK, FK)	Part of the composite key; references fixtures.fixture_id.
player_id	INT (PK, FK)	Part of the composite key; references

		players.player_id.
minutes_played	INT	Number of minutes the player was on the field.
goals	INT	Number of goals scored by the player.
assists	INT	Number of assists made by the player.

2.10 player_season_stats

- **Purpose:** Stores aggregated statistics for a player across an entire season for a specific league and team.

Column	Type	Description
stat_id	INT (PK)	The unique identifier for the stat summary row.
player_id	INT (FK)	References players.player_id.
team_id	INT (FK)	References teams.team_id.
league_id	INT (FK)	References leagues.league_id.
season_year	INT	The season the stats apply to.
appearances	INT	Total number of matches played.
goals_total	INT	Total goals scored in the

		season.
--	--	---------

(Note: Other stat columns like `passes_total`, `cards_yellow`, etc., are included as defined in the schema).

(The report would continue with similar descriptions for the `player_trophies`, `player_injuries`, and `transfers` tables.)

3.0 Database Schema and Relationships

The schema is designed around a set of core entities and the relationships between them. Foreign key constraints are used to maintain data integrity.

Key Relationships:

- **Teams and Countries:** A teams record has one `country_id`, establishing a many-to-one relationship. Many teams can belong to one country.
- **Players and Teams:** This is a many-to-many relationship managed through the `player_team_history` table. A player can play for many teams over their career, and a team has many players.
- **Fixtures, Leagues, and Teams:** Each fixtures record is linked to one leagues, one seasons, a home teams, and an away teams. This creates the structure for match schedules and results.
- **Players and Statistics:** Player performance is tracked at two levels. The `player_match_stats` table links a players record to a fixtures record. The `player_season_stats` table provides a higher-level summary, linking a players record to a leagues and seasons.

4.0 Conclusion

This database design provides a robust and normalized foundation for a comprehensive football application. By separating entities into distinct tables and using linking tables for complex relationships, the schema ensures data integrity, minimizes redundancy, and allows for efficient and powerful querying. It effectively captures the dynamic nature of player careers, match results, and seasonal statistics.