**1. Extract Phase**

* Collected multiple datasets from two sources:
  + **Dataset 1**: Player performance statistics split into multiple CSVs (player\_defense, player\_gca, player\_misc, player\_shooting, player\_possession, player\_passing\_type, player\_passing, player\_standard\_stats).
  + **Dataset 2**: Player valuation data with supporting metadata (players.csv, clubs.csv, competitions.csv).

**2. Transform Phase**

**2.1 Data Cleaning**

* **Standardized column names and formats** across all CSVs.
* Converted all dates to YYYY format (e.g., valuation\_date -> year).
* Cleaned up string inconsistencies in player and club names.
* Removed rows that had missing critical identifiers (e.g., missing player names).

**2.2 Player Matching & Enrichment**

* Substituted player\_id in the valuation dataset with full player names (first\_name + last\_name).
* Used the players.csv file to fill missing fields like:
  + nation from country\_of\_citizenship
  + country from country\_of\_birth
  + born from date\_of\_birth (converted to year only)
* Handled cases where names were mismatched due to formatting (e.g., only surname or extra spaces).

**2.3 Club Name Normalization**

* Resolved inconsistent club names between dataset\_1 and dataset\_2 using:
  + Manual mapping for known mismatches (e.g., Manchester United Football -> Manchester Utd).
  + Fuzzy string matching with fallback logic.

**2.4 Competition Name Mapping**

* Mapped verbose competition names to standardized versions using a manual dictionary.
* Dropped rows for competitions outside the top 5 leagues (Premier League, Serie A, La Liga, Bundesliga, Ligue 1).

**2.5 Enriching Club-Year Information**

* In valuations\_with\_season\_club, added a club\_in\_year column by:
  + Searching each dataset\_1 file for matching (player\_name, season).
  + Extracting the squad (club name) from the year of interest.
  + Using fallback to current club if none was found.

**2.6 Global Missing Value Resolution**

* Repaired missing values in key fields (born, country, continent, nation, age) using:
  + Secondary dataset joins (players.csv)
  + Value inference from same-country rows in other files.
  + Computed age using season - born.

**3. Row Selection and Optimization**

**3.1 Global Pair Scoring**

* Computed (player, season) pairs present in **all** stats files **and** the valuation file.
* Counted total missing cells across all stats files **per pair**.
* Selected the **8,000 best player-season pairs** with the lowest null count.

**3.2 Filtering & Export**

* Filtered every stats file and the valuation file to retain **only the top 8,000 pairs**.
* Ensured all files had **identical sets of player-season keys**.
* Exported filtered files to global\_selected\_8000/ folder.

**4. Load Preparation**

* Final CSVs are now clean, aligned, and optimized.
* Each file is ready for import into PostgreSQL.
* Keys (player, season) can be used to join fact and dimension tables.

**✅ Outcome**

You now have a fully transformed, cleaned, and optimized dataset across multiple files that can:

* Support a dimensional model (DFM)
* Enable analytical queries (OLAP)
* Be easily loaded into a Data Warehouse system