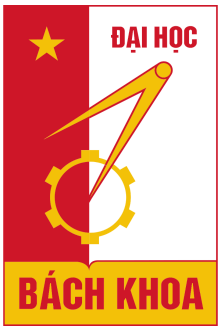
NGUYỄN PHÚ VINH

ĐẠI HỌC BÁCH KHOA HÀ NỘI
KHOA TOÁN - TIN




NGUYỄN PHÚ VINH

# XÂY DỰNG KHO DỮ LIỆU CHO PHÂN TÍCH BÓNG ĐÁ

ĐỒ ÁN II

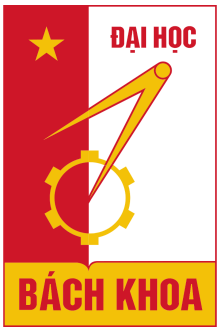
Chuyên ngành: TOÁN TIN

HÀ NỘI - 2025

NGUYỄN PHÚ VINH

HÀ NỘI - 2025

ĐẠI HỌC BÁCH KHOA HÀ NỘI
KHOA TOÁN - TIN




# XÂY DỰNG KHO DỮ LIỆU CHO PHÂN TÍCH BÓNG ĐÁ

## ĐỒ ÁN II

Chuyên ngành: TOÁN TIN

Giảng viên hướng dẫn: TS. Nguyễn Đình Hân
Chữ kí của GVHD

Sinh viên thực hiện: Nguyễn Phú Vinh

MSSV: 20227169

Lớp: Toán-Tin 01 – K67

HÀ NỘI - 2025

# NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

**1. Mục tiêu và nội dung của đồ án**

.....

.....

.....

.....

.....

.....

**2. Kết quả đạt được**

.....

.....

.....

.....

.....

.....

**3. Ý thức làm việc của sinh viên**

.....

.....

.....

.....

.....

.....

*Hà Nội, ngày ... tháng ... năm 2026*

Giảng viên hướng dẫn

Danh sách lớp học / Đồ án II / 756214

**Thông tin lớp học mã 756214**

Kì học: 20251 Mã học phần: MI3390 Tên học phần: Đồ án II

Mã lớp: 756214

**Đồ án**

Giáo viên hướng dẫn: Nguyễn Đình Hân

Tên đồ án: Xây dựng kho dữ liệu cho phân tích bóng đá

Nội dung:
1) Khảo sát và mô tả hệ thống ứng dụng
2) Xác định và phân tích yêu cầu
3) Thiết kế chi tiết
4) Cài đặt hệ thống
5) Lập tài liệu và báo cáo kết quả

Các mốc kiểm soát chính: Tổng thời gian 15 tuần.
+) 22/9/2025: Nộp bản đề xuất đề tài/dự án
+) 27/10/2025: Nộp cuốn báo cáo đồ án lần 1 <Phiên bản 1.0>
+) 1/12/2025: Nộp cuốn báo cáo đồ án lần 2 <Phiên bản 2.0>
+) 22/12/2025: Nộp cuốn báo cáo đồ án, slide và chương trình <Phiên bản hoàn thiện>

Giáo viên phản biện:

**Danh sách đánh giá đồ án**

| Ngày đánh giá | Lần | Nội dung kế hoạch | Nội dung đã thực hiện | Điểm tích cực | Điểm nội dung | Ghi chú |
|---|---|---|---|---|---|---|
| 28/11/2025 | 1 | Nội dung 1-3 | Đảm bảo tiến độ | 10 | 9 | |
| 14/12/2025 | 2 | Nội dung 4-5 | Đảm bảo tiến độ | 10 | 9 | |

Hình 1: Báo cáo tiến độ đồ án

# Mục lục

# Danh sách hình vẽ

# Bảng ký hiệu và chữ viết tắt

| | |
|---|---|
| $xG$ | Xác suất một cú sút thành bàn (Expected Goals) |
| $G - xG$ | Hiệu số giữa tổng số bàn thắng thực tế và tổng xG của cầu thủ hoặc đội bóng (Goals minus Expected Goals) |
| $xA$ | Xác suất một đường chuyền trở thành kiến tạo (Expected Assists) |
| $PPDA$ | Số đường chuyền trung bình của đội B trong khu vực 2/3 sân cuối cùng (có tọa độ $x \geq 40$ trên sân có kích cỡ $120 \times 80$) trước khi đội A thực hiện một hành động phòng ngự (Passes Per Defensive Action) |
| $TiB/90$ | Số lần chạm bóng trong vòng cấm của đối phương, được chuẩn hóa theo 90 phút thi đấu (Touches in Box/90) |
| $PAdjI/90$ | Số lần cắt bóng đã điều chỉnh theo quyền kiểm soát bóng (Possession-Adjusted Interceptions/90) |
| $TSR$ | Tỷ lệ tắc bóng thành công (Tackles Success Rate) |
| **DW** | Kho dữ liệu (Data Warehouse) |
| **OLTP** | Hệ thống xử lý giao dịch trực tuyến (Online Transaction Processing) |
| **OLAP** | Hệ thống phân tích trực tuyến (Online Analytical Processing) |
| **BI** | Kinh doanh thông minh (Business Intelligence) |
| **EDW** | Kho dữ liệu doanh nghiệp (Enterprise Data Warehouse) |

# Lời mở đầu

Trong thời đại số hóa, dữ liệu đã trở thành một trong những yếu tố then chốt trong việc đưa ra quyết định và định hướng chiến lược ở mọi lĩnh vực, bao gồm cả thể thao.

Trong bóng đá, việc phân tích hiệu suất cầu thủ, tối ưu hóa chiến thuật, điều chỉnh giáo án tập luyện và đánh giá trận đấu đều phụ thuộc vào khả năng thu thập, xử lý và phân tích lượng dữ liệu khổng lồ (bao gồm dữ liệu sự kiện, dữ liệu theo dõi vị trí, dữ liệu vật lý, ...). Để khai thác tối đa giá trị của nguồn dữ liệu này, việc xây dựng một kho dữ liệu (Data Warehouse) hiệu quả và đầy đủ là rất cần thiết.

Xuất phát từ lý do trên, em quyết định lựa chọn đề tài "Xây dựng kho dữ liệu cho phân tích bóng đá". Đồ án mong muốn xây dựng một kho dữ liệu giúp giải quyết các bài toán phân tích, dự báo, và cung cấp những góc nhìn đa chiều về dữ liệu bóng đá, hỗ trợ công tác huấn luyện và quản lý bóng đá hiệu quả hơn.

Ngoài phần Mở đầu và Kết luận, đồ án của em sẽ bao gồm 4 chương chính:

- Chương I: Cơ sở lý thuyết.
- Chương II: Khảo sát hệ thống.
- Chương III: Thiết kế hệ thống.
- Chương IV: Cài đặt hệ thống.

*Hà Nội, tháng 10 năm 2025*

Sinh viên

**Nguyễn Phú Vinh**

# Lời cảm ơn

Em xin gửi lời cảm ơn chân thành đến thầy Nguyễn Đình Hân, người đã tận tình hướng dẫn và đồng hành cùng em trong suốt quá trình thực hiện đồ án này. Sự chỉ bảo tận tâm cùng những ý kiến đóng góp quý giá của thầy đã giúp em xác định hướng đi đúng đắn và vượt qua nhiều thử thách trong quá trình phát triển phần mềm. Em cũng xin gửi lời cảm ơn chân thành đến các thầy cô Khoa Toán-Tin, Đại học Bách khoa Hà Nội. Sự tận tâm giảng dạy và kiến thức mà các thầy cô truyền đạt đã giúp em tự tin áp dụng lý thuyết vào thực tiễn, góp phần quan trọng vào việc hoàn thiện đồ án này.

Dù đã nỗ lực hết mình để thực hiện và hoàn thành đồ án, em nhận thấy sản phẩm của mình vẫn còn những thiếu sót. Vì vậy, em rất mong nhận được những ý kiến nhận xét quý báu từ thầy cô để có thể cải thiện đồ án tốt hơn, đồng thời tích lũy thêm kinh nghiệm thực tế cho bản thân.

Em xin chân thành cảm ơn!

# Chương 1

# Cơ sở lý thuyết

## 1.1 Giới thiệu về phân tích dữ liệu bóng đá

**Bóng đá** (hay còn gọi là túc cầu, đá bóng, đá banh) là một môn thể thao đồng đội được chơi với quả bóng hình cầu giữa hai đội gồm 11 cầu thủ mỗi bên. Môn thể thao này là môn thể thao phổ biến nhất trên thế giới với khoảng hơn 250 triệu người chơi ở hơn 200 quốc gia và vùng lãnh thổ. Môn này chơi trên một mặt sân hình chữ nhật với một khung thành ở mỗi đầu. Mục tiêu là ghi bàn vào khung thành đối phương. Đội nào có số bàn thắng nhiều hơn sẽ giành chiến thắng. [1].

Trong bóng đá hiện đại, các kỹ thuật, công nghệ hỗ trợ cho việc phân tích, đánh giá ngày càng trở nên phổ biến hơn vì những lợi ích mà chúng mang lại. Rất nhiều đội bóng trên toàn thế giới, đặc biệt là các đội bóng giàu thành tích tại các giải đấu hàng đầu châu Âu, có thể sẵn sàng chi những số tiền rất lớn để đầu tư vào những công nghệ này nhằm cải thiện thành tích cho đội bóng, nâng cao hiệu quả trong công tác huấn luyện, thi đấu, đào tạo các cầu thủ trẻ tài năng hay thậm chí là để có thể mang về những bản hợp đồng chất lượng, đáng tiền trong mỗi kì chuyển nhượng căng thẳng. Nhờ vậy, dữ liệu được tổng hợp từ các trận đấu lại trở thành nguồn tài nguyên vô cùng quý giá đối với họ, điều này đã phần nào phản ánh tầm quan trọng của một kho dữ liệu lưu trữ nguồn tài nguyên này để phục vụ cho sự phân tích, đánh giá của các chuyên gia.

Trong một trận đấu, điều mà những cổ động viên cuồng nhiệt lưu tâm đến không chỉ là những bàn thắng. Đó còn là phong cách chơi bóng độc đáo của các cầu thủ trên sân, những đường chuyền, đường kiến tạo đẹp mắt, những tình huống tranh chấp quyết liệt, những pha cản phá xuất thần của hậu vệ hoặc thủ môn hay những tình huống cố định, tình huống phản công,... Tất cả đều có thể được hiểu

đơn giản là những sự kiện diễn ra trong một trận đấu. Nhưng ẩn sâu trong những dữ liệu sự kiện đó, các chuyên gia phân tích thường quan tâm đến các chỉ số sau:

- $xG$ (Expected Goals): Xác suất một cú sút thành bàn, với giá trị dao động từ 0 đến 1, được tính dựa trên dữ liệu lịch sử của hàng nghìn cú sút có đặc điểm (vị trí tọa độ, góc sút, khoảng cách tới khung thành, bộ phận cơ thể, loại cơ hội,...) tương tự. Chỉ số này giúp đánh giá chất lượng cơ hội.

- $G - xG$ (Goals minus Expected Goals): Hiệu số giữa tổng số bàn thắng thực tế và tổng xG của cầu thủ hoặc đội bóng. Chỉ số này giúp đánh giá khả năng dứt điểm thành bàn của cầu thủ hoặc đội bóng.

- $xA$ (Expected Assists): Xác suất một đường chuyền trở thành kiến tạo, được tính bằng cách lấy $xG$ của cú sút ngay sau đường chuyền đó. Chỉ số này giúp đánh giá khả năng tạo cơ hội.

- $PPDA$ (Passes Per Defensive Action): Số đường chuyền trung bình của đội B trong khu vực 2/3 sân cuối cùng (có tọa độ $x \geq 40$ trên sân có kích cỡ $120 \times 80$) trước khi đội A thực hiện một hành động phòng ngự. Đây là chỉ số đo lường mức độ bị ép sân của đội A.

$$\textbf{PPDA}_A = \frac{\text{Số đường chuyền của B trong khu vực } x \geq 40}{\text{Số sự kiện phòng ngự của A trong khu vực } x \geq 40} \tag{1.1}$$

- $TiB/90$ (Touches in Box/90): Số lần chạm bóng trong vòng cấm của đối phương, được chuẩn hóa theo 90 phút thi đấu. Chỉ số này giúp đánh giá khả năng chọn vị trí và độ nguy hiểm khi tham gia tấn công của cầu thủ.

$$\textbf{TiB/90} = \frac{\text{Tổng số lần chạm bóng trong vòng cấm}}{\text{Tổng số phút đã chơi}} \times 90 \tag{1.2}$$

- $PAdjI/90$ (Possession-Adjusted Interceptions/90): Số lần cắt bóng đã điều chỉnh theo quyền kiểm soát bóng. Chỉ số này đo lường số lần một cầu thủ cắt đường chuyền của đối phương trong 90 phút, sau đó điều chỉnh bằng một hệ số dựa trên thời gian đội đó không kiểm soát bóng.

$$\textbf{PAdjI/90} = \frac{\text{Tổng số lần cắt bóng}}{\text{Tổng số phút đã chơi}} \times 90 \times \frac{\text{Tỷ lệ \% kiểm soát bóng đội bạn}}{\text{Tỷ lệ \% kiểm soát bóng đội nhà}} \tag{1.3}$$

- *TSR* (Tackles Success Rate): Tỷ lệ tắc bóng thành công. Chỉ số này đánh giá khả năng tắc bóng chính xác, sự quyết đoán trong phòng ngự của cầu thủ.

$$\textbf{TSR} = \frac{\text{Tổng số lần tắc bóng thành công}}{\text{Tổng số lần tắc bóng}} \times 100\% \tag{1.4}$$

Sử dụng những chỉ số như vậy, các chuyên gia có thể thực hiện các công việc như: đánh giá phong độ của cầu thủ, tìm kiếm và phát hiện tài năng; điều chỉnh giáo án tập luyện, chiến thuật, vị trí thi đấu; đánh giá điểm mạnh và rủi ro trong hệ thống vận hành của đội bóng; tư vấn chuyển nhượng; dự đoán kết quả thi đấu và phong độ của cầu thủ, đội bóng trong tương lai.

## 1.2 Tổng quan về Kho dữ liệu (Data Warehouse) và Phân tích xử lý trực tuyến (OLAP)

### 1.2.1 Kho dữ liệu (Data Warehouse)

**Khái niệm**

Kho dữ liệu (Data Warehouse - DW) là một cơ sở dữ liệu lớn, tập trung, lưu trữ dữ liệu lịch sử từ nhiều nguồn khác nhau đã được tích hợp và cấu trúc hóa riêng biệt cho mục đích phân tích. Khái niệm này phân biệt rõ ràng DW với các hệ thống tác nghiệp (Online Transaction Processing - OLTP):

- **Mục tiêu**: Trong khi các hệ thống OLTP được tối ưu cho việc vận hành kinh doanh hàng ngày (xử lý các giao dịch nhỏ, nhanh), DW được tối ưu cho việc phân tích và ra quyết định (Online Analytical Processing - OLAP).
- **Chức năng**: DW hoạt động như trái tim của kinh doanh thông minh (Business Intelligence - BI). Nó giúp hợp nhất dữ liệu từ nhiều nguồn, đảm bảo tính nhất quán và cung cấp cái nhìn toàn cảnh.
- **Triết lý thiết kế**: DW thường sử dụng mô hình phi chuẩn hóa, phổ biến nhất là mô hình đa chiều (Dimensional Model), như lược đồ sao (Star Schema). Triết lý này ưu tiên tốc độ truy vấn phân tích bằng cách giảm số lượng phép JOIN, vốn là vấn đề của các cơ sở dữ liệu chuẩn hóa cao (3NF) trong OLTP.

**Tính chất**

- **Hướng chủ đề**: Dữ liệu trong DW được tổ chức xoay quanh các chủ đề kinh doanh chính thay vì theo các quy trình nghiệp vụ của từng phòng ban như hệ

thống OLTP, giúp cung cấp một cái nhìn toàn diện về một chủ đề cụ thể, hợp nhất dữ liệu liên quan từ nhiều hệ thống nguồn khác nhau.

- **Tích hợp**: Dữ liệu từ các nguồn khác nhau phải được tổng hợp và nhất quán hóa. Sự tích hợp này thể hiện ở việc áp dụng các quy ước đặt tên chung, đơn vị đo lường thống nhất và định dạng dữ liệu chuẩn trên toàn bộ kho dữ liệu.
- **Bất biến**: Khi có sự thay đổi dữ liệu trong hệ thống nguồn (ví dụ: khách hàng đổi địa chỉ), DW không ghi đè mà sẽ thêm một bản ghi mới để ghi nhận sự thay đổi theo thời gian.
- **Tính thời gian**: Mọi dữ liệu trong DW đều được gắn với một yếu tố thời gian. Kiến trúc của DW luôn được thiết kế để cho phép phân tích theo dòng thời gian (ví dụ: so sánh doanh thu quý này so với cùng kỳ năm ngoái), trong khi hệ thống OLTP thường chỉ quan tâm đến trạng thái hiện tại.

**Ưu điểm**

- **Hỗ trợ ra quyết định chiến lược**: DW/BI là công cụ mạnh mẽ thúc đẩy chuyển đổi dữ liệu thô thành trí tuệ để dẫn dắt chiến lược. Nó hỗ trợ ra quyết định ở cả ba cấp độ: chiến lược, chiến thuật và tác nghiệp.
- **Tính nhất quán, độ tin cậy**: DW giúp đảm bảo tính đúng đắn của dữ liệu.
- **Phân tích lịch sử**: Khả năng lưu trữ dữ liệu lịch sử chi tiết cho phép phân tích xu hướng dài hạn.
- **Hiệu năng phân tích cao**: Thiết kế theo mô hình đa chiều (phi chuẩn hóa) giúp tăng tốc độ truy vấn, giảm các phép JOIN khi tổng hợp dữ liệu.
- **Nền tảng cho Machine Learning/AI**: Dữ liệu sạch, tích hợp và có tính lịch sử giúp giảm thời gian chuẩn bị dữ liệu cho các mô hình học máy.

**Nhược điểm**

- **Độ trễ dữ liệu**: Dữ liệu trong DW thường được cập nhật theo lô, dẫn đến độ trễ nhất định so với dữ liệu thời gian thực.
- **Chi phí và thời gian triển khai ban đầu**: Việc xây dựng một DW truyền thống đòi hỏi chi phí đầu tư ban đầu lớn cho cơ sở hạ tầng, phần mềm, nhân lực và hỗ trợ kỹ thuật, có thể mất nhiều thời gian để thấy được giá trị.

- **Khả năng xử lý dữ liệu phi cấu trúc**: DW truyền thống gặp khó khăn khi xử lý dữ liệu bán cấu trúc và phi cấu trúc.
- **Tính cứng nhắc**: Mô hình phải được định nghĩa trước khi dữ liệu được nạp vào. Việc thay đổi cấu trúc DW sau này có thể phức tạp và tốn kém.

**Kiến trúc kho dữ liệu cơ bản**

| Mô hình kiến trúc | Đặc điểm chính | Hạn chế |
|---|---|---|
| **Một tầng** | Hoạt động như một hệ thống ảo hoặc lớp trung gian để tổng hợp dữ liệu, nhằm giảm thiểu sự dư thừa dữ liệu trong quá trình lưu trữ. | Ít được sử dụng vì hạn chế về khả năng mở rộng và tích hợp dữ liệu, bao gồm việc hợp nhất dữ liệu và loại bỏ trùng lặp. |
| **Hai tầng** | Gồm Nguồn dữ liệu, Vùng đệm, Lớp kho dữ liệu (lưu trữ dữ liệu đã xử lý, Data Marts, Meta-data), và Lớp phân tích/báo cáo. | Đơn giản hơn, nhưng khả năng tích hợp dữ liệu có thể chưa tối ưu. |
| **Ba tầng** | Mô hình phổ biến nhất, đặc biệt trong các doanh nghiệp lớn. Gồm 3 lớp:<br>1. Lớp nguồn dữ liệu.<br>2. Lớp xử lý trung gian.<br>3. Lớp kho dữ liệu. | Đòi hỏi không gian lưu trữ lớn cho lớp xử lý trung gian và có thể gặp hạn chế trong việc phân tích dữ liệu theo thời gian thực. |

**Kiến trúc BI tổng thể**

**Các thành phần cốt lõi**

1. **Tầng nguồn và tích hợp dữ liệu**:
   - **Nguồn dữ liệu (Data Sources)**: Là điểm khởi đầu, bao gồm các hệ thống tác nghiệp (OLTP), hoặc các nguồn bên ngoài (file Excel, dữ liệu mạng xã hội). Tầng này có thể chứa nhiều loại dữ liệu: có cấu trúc, bán cấu trúc (JSON, XML), và phi cấu trúc (video, log server).
   - **Vùng đệm (Staging Area)**: Là khu vực lưu trữ trung gian.

  - **Vai trò**: Dữ liệu thô sau khi trích xuất (Extract) sẽ được đưa vào đây. Mọi quá trình biến đổi (Transform) như làm sạch, chuẩn hóa, kết hợp và định hình lại dữ liệu sẽ diễn ra tại đây.
  - **Mục đích**: Cách ly quá trình xử lý nặng khỏi hệ thống nguồn để không làm chậm hệ thống tác nghiệp.

2. **Tầng lưu trữ (Storage Layer)**: Nơi dữ liệu đã được xử lý và tích hợp được lưu trữ.

   - **Kho dữ liệu doanh nghiệp (Enterprise Data Warehouse - EDW)**:
     - Là một cơ sở dữ liệu tập trung, lớn, lưu trữ dữ liệu lịch sử đã được tích hợp và cấu trúc hóa cho mục đích phân tích.
   - **Kho dữ liệu chủ đề (Data Marts)**:
     - Là các tập con nhỏ hơn, chuyên biệt, trích xuất từ kho dữ liệu doanh nghiệp hoặc được xây dựng riêng.
     - Được thiết kế để phục vụ nhu cầu phân tích của một phòng ban hoặc lĩnh vực nghiệp vụ cụ thể.
   - **Kho siêu dữ liệu (Metadata Repository)**: Nơi lưu trữ thông tin về nguồn gốc, cấu trúc bảng, các phép biến đổi, và cách truy cập dữ liệu.

3. **Tầng phân tích và trình bày (Analytics and Presentation Layer)**: Nơi dữ liệu được chuyển hóa thành tri thức và giao tiếp đến người dùng cuối.

   - **Khối OLAP**: Tầng xử lý các truy vấn phức tạp trên dữ liệu, thường sử dụng các kỹ thuật OLAP, cho phép người dùng thực hiện các thao tác phân tích như Drill-Down, Roll-Up, Slice, Dice và Pivot.
   - **Lớp ngữ nghĩa**: Một lớp trừu tượng ánh xạ cấu trúc bảng, cột sang các thuật ngữ kinh doanh dễ hiểu (ví dụ: "doanh thu", "lợi nhuận").
   - **Công cụ BI và Báo cáo**: Giao diện người dùng cuối tương tác, bao gồm các báo cáo (Reports) và dashboard tương tác.

**Luồng dữ liệu và kiến trúc Hiện đại (ELT/Lakehouse)**

- **Quy trình ETL/ELT**: ETL (Extract, Transform, Load) là quy trình nơi biến đổi dữ liệu diễn ra ở máy chủ trung gian. ELT (Extract, Load, Transform)

là mô hình hiện đại hơn, nơi dữ liệu thô được tải vào kho dữ liệu đám mây trước, sau đó dùng sức mạnh xử lý của chính kho dữ liệu để biến đổi.

- **Sự kết hợp Data Lake**: Một kiến trúc hiện đại thường bao gồm Hồ dữ liệu (Data Lake), nơi lưu trữ mọi loại dữ liệu ở định dạng thô.
- **Kiến trúc Lakehouse**: Là sự hợp nhất của Data Lake và Data Warehouse, nhằm phá bỏ sự phức tạp và dư thừa của kiến trúc hai tầng truyền thống. Data Lakehouse cung cấp một nền tảng duy nhất để phục vụ cho cả phân tích kinh doanh (BI) và khoa học dữ liệu (Machine Learning/AI).

### 1.2.2 Hệ thống phân tích xử lý trực tuyến (OLAP)

Trong kiến trúc kho dữ liệu, OLAP là thành phần chủ đạo của tầng phân tích và trình bày. OLAP là cơ chế chuyển hóa dữ liệu lịch sử đã được tích hợp thành tri thức có thể hành động được.

#### Định nghĩa và vai trò của hệ thống OLAP

OLAP (Online Analytical Processing) là một giải pháp phân tích dữ liệu mạnh mẽ, được thiết kế để xử lý và khai thác thông tin từ nhiều góc độ khác nhau với hiệu suất cao, ngay cả khi làm việc với khối lượng dữ liệu khổng lồ. Các hệ thống OLAP được sinh ra để giải quyết những câu hỏi phân tích phức tạp, hỗ trợ các truy vấn trên một khối lượng lớn dữ liệu lịch sử.

Vai trò cốt lõi của OLAP bao gồm:

- **Hỗ trợ phân tích và ra quyết định**: Hệ thống giúp tổ chức đưa ra các quyết định kinh doanh tốt hơn.
- **Phân tích đa chiều**: Cung cấp khả năng "nhìn" dữ liệu từ nhiều góc độ khác nhau (đa chiều) để tìm ra xu hướng, mẫu và tri thức ẩn.
- **Hỗ trợ dự báo và lập kế hoạch**: Giúp doanh nghiệp xây dựng các kế hoạch chiến lược và dự báo các kịch bản trong tương lai.
- **Tối ưu hóa hoạt động**: Giúp nhận diện các vấn đề cần cải thiện trong quy trình kinh doanh, từ đó tăng hiệu quả vận hành.
- **Người dùng**: Nhà phân tích dữ liệu, nhà quản lý, lãnh đạo cấp cao, v.v.

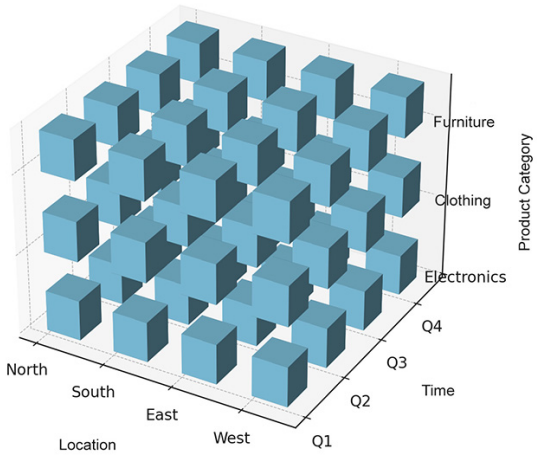
**Phân biệt với hệ thống xử lý giao dịch trực tuyến (OLTP)**

- **Thiết kế cơ sở dữ liệu**: OLAP sử dụng mô hình dữ liệu phi chuẩn hóa, như lược đồ sao (Star Schema), để giảm số lượng phép JOIN và tăng tốc độ truy vấn phân tích. Ngược lại, OLTP yêu cầu chuẩn hóa cao (ví dụ: 3NF) để đảm bảo tính toàn vẹn dữ liệu.
- **Loại thao tác**: OLAP chủ yếu thực hiện các thao tác đọc và tổng hợp trên hàng triệu bản ghi. Các thao tác ghi (INSERT, UPDATE) rất hạn chế và thường diễn ra theo lô.

**Mô hình khối dữ liệu đa chiều**

Khối OLAP (OLAP Cube) là một cấu trúc dữ liệu đa chiều được tối ưu hóa để truy vấn và phân tích nhanh, là hiện thực hóa của lớp ngữ nghĩa.

- **Lớp ngữ nghĩa**: Là lớp trừu tượng nằm giữa người dùng và cơ sở dữ liệu. Nó chuyển đổi các cấu trúc bảng phức tạp thành các thuật ngữ kinh doanh dễ hiểu (ví dụ: "doanh thu", "lợi nhuận"). Điều này giải quyết vấn đề người dùng không quen thuộc với SQL hoặc cấu trúc bảng Fact/Dimension.
- **Tính toán trước**: Khối OLAP thường tính toán trước các giá trị tổng hợp ở nhiều cấp độ khác nhau để các truy vấn có thể được trả về gần như tức thời.




Hình 1.1: Khối OLAP

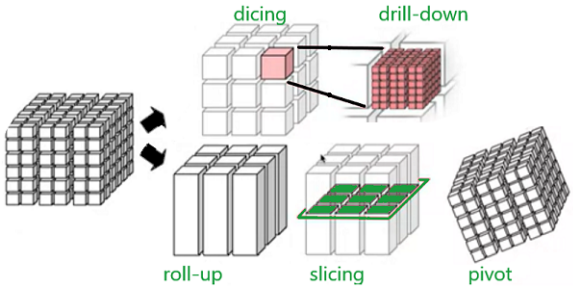
**Thành phần khối dữ liệu**

Khối dữ liệu OLAP được xây dựng dựa trên lược đồ sao, bao gồm hai thành phần chính:

1. **Chỉ số đo lường (Measures)**:
    - Là các cột Fact trong Bảng Fact.
    - Đây là các giá trị số mà ta muốn đo đếm và phân tích.
2. **Các chiều (Dimensions)**:
    - Là các bảng Dimension.
    - Chúng trở thành các trục dùng để phân tích Measures, cung cấp bối cảnh cho các con số.
    - Các thuộc tính phân cấp (ví dụ: Năm, Quý, Tháng) tạo thành các hệ thống phân cấp (Hierarchies) bên trong chiều.

**Các phép toán phân tích OLAP**

1. **Drill-Down (Đào sâu)**:
    - Điều hướng từ một cấp độ cao hơn xuống một cấp độ thấp hơn trong hệ thống phân cấp (ví dụ: từ xem Doanh thu theo Năm xuống theo Quý).
2. **Roll-Up (Tổng hợp)**:
    - Tổng hợp dữ liệu từ một cấp độ chi tiết lên một cấp độ tổng quan hơn (ví dụ: từ xem theo Thành phố, lên xem theo Vùng).
3. **Slice (Cắt lát)**:
    - Lọc dữ liệu theo một chiều, chọn một giá trị duy nhất cho một chiều để xem một "lát cắt" của khối dữ liệu (ví dụ: bộ lọc chỉ xem một khu vực).
4. **Dice (Cắt khối)**:
    - Lọc dữ liệu theo nhiều chiều, chọn các giá trị cụ thể trên hai hoặc nhiều chiều khác nhau để xem một "khối con" của dữ liệu (ví dụ: xem doanh thu của ngành hàng "Thời trang Nữ" tại "TP.HCM" trong "Quý 4").
5. **Pivot (Xoay)**:
    - Thay đổi cách dữ liệu được hiển thị mà không thay đổi giá trị của dữ liệu. Nó cho phép hoán đổi vị trí của các chiều giữa trục hàng và trục cột (ví dụ: thay vì xem doanh thu theo sản phẩm ở hàng và khu vực ở cột, người dùng chuyển sang khu vực ở hàng và sản phẩm ở cột).




Hình 1.2: Các phép toán trên khối OLAP

## 1.3 Quy trình tích hợp dữ liệu ETL và ELT

Quy trình tích hợp dữ liệu là xương sống vận hành của một kho dữ liệu. Nhiệm vụ của nó là di chuyển và chuẩn bị dữ liệu từ các hệ thống cơ sở dữ liệu tác nghiệp (OLTP) sang mô hình phân tích (OLAP) có trật tự và nhất quán. Có hai kiến trúc chính được sử dụng: ETL truyền thống và ELT hiện đại.

### 1.3.1 Kiến trúc ETL Kinh điển

ETL (Extract - Trích xuất, Transform - Biến đổi, Load - Tải) là quy trình cốt lõi chịu trách nhiệm di chuyển và chuẩn bị dữ liệu cho kho dữ liệu. Kiến trúc ETL điển hình sử dụng một vùng đệm (Staging Area), nơi các phép biến đổi phức tạp diễn ra. Việc này giúp giảm thiểu tác động lên hệ thống nguồn và đảm bảo kho dữ liệu đích chỉ nhận vào dữ liệu đã sạch.

#### Giai đoạn E - Trích xuất (Extract)

**Mục tiêu**: Đọc và lấy dữ liệu từ một hoặc nhiều hệ thống nguồn. Dữ liệu nguồn có thể từ cơ sở dữ liệu quan hệ, file phẳng (CSV, Excel) cho đến các API.

**Các phương pháp:**

- **Trích xuất Toàn bộ**: Sao chép toàn bộ bảng mỗi lần chạy, chỉ phù hợp với các bảng dữ liệu nhỏ, ít thay đổi.
- **Trích xuất tăng trưởng**: Chỉ trích xuất những dữ liệu đã thay đổi kể từ lần cuối cùng, tối ưu cho các bảng lớn.
- **Trích xuất từ các nguồn phức tạp**: Đối với các nguồn như API hoặc file, quy trình trích xuất phải xử lý các thách thức như giới hạn số lần gọi, phân trang và cấu trúc không nhất quán.

**Giai đoạn T - Biến đổi (Transform)**

**Mục tiêu**: Chuyển đổi dữ liệu thô, không nhất quán và phân mảnh thành một bộ dữ liệu sạch, tuân thủ các quy tắc nghiệp vụ và có cấu trúc phù hợp với mô hình lược đồ sao đã thiết kế. Các tác vụ chính diễn ra tại vùng đệm bao gồm:

- **Làm sạch và chuẩn hóa dữ liệu**:
  - **Phân tách cấu trúc**: Tách các cấu trúc phức tạp (như dòng log web hoặc JSON) thành các cột riêng biệt có ý nghĩa.
  - **Chuẩn hóa**: Đưa các giá trị khác nhau nhưng cùng ngữ nghĩa về một dạng chuẩn duy nhất (ví dụ: ánh xạ "HN" về "Hà Nội").
  - **Xử lý giá trị NULL**: Thay thế bằng giá trị mặc định hoặc loại bỏ.
  - **Xác thực**: Kiểm tra dữ liệu có vi phạm các quy tắc nghiệp vụ không.
- **Tích hợp dữ liệu và Tạo khóa**:
  - **Loại bỏ trùng lặp và hợp nhất**: Định nghĩa các quy tắc để xác định và hợp nhất các bản ghi trùng lặp từ nhiều nguồn.
  - **Tạo khóa thay thế**: Quy trình ETL phải gán một khóa thay thế mới, đơn giản và có thứ tự cho mỗi giá trị của bảng Dimension, thay vì sử dụng khóa nghiệp vụ từ hệ thống nguồn.
  - **Hiện thực hóa Logic SCD**: Áp dụng logic SCD loại 1, loại 2, hoặc loại 3 để theo dõi lịch sử thay đổi của các thuộc tính Dimension (ví dụ: địa chỉ khách hàng thay đổi theo thời gian).
- **Biến đổi cho Bảng Fact**:
  - **Tra cứu và thay thế khóa**: Thay thế tất cả các khóa nghiệp vụ từ nguồn bằng các khóa thay thế tương ứng từ các bảng Dimension.
  - **Tính toán chỉ số đo lường**: Tính toán trước các chỉ số đo lường mới và lưu trữ chúng trong bảng Fact để tăng hiệu năng truy vấn.

**Giai đoạn L - Tải (Load)**

**Mục tiêu**: Di chuyển dữ liệu đã được biến đổi từ vùng đệm vào các bảng Fact và Dimension trong kho dữ liệu đích một cách hiệu quả, an toàn.

**Chiến lược tải và tối ưu hóa**:

- **Tải ban đầu và tăng trưởng**:
  - **Tải ban đầu**: Tải toàn bộ dữ liệu lịch sử lần đầu tiên.
  - **Tải tăng trưởng**: Chỉ tải các dữ liệu mới hoặc đã thay đổi kể từ lần tải cuối cùng, phải hoàn thành trong cửa sổ tải cho phép.
- **Tối ưu hóa tải bảng Fact lớn**: Đối với các bảng Fact khổng lồ (hàng tỷ dòng), kỹ thuật tối ưu hóa gồm: vô hiệu hóa hoặc xóa chỉ mục (indexes) trước khi bắt đầu tải, tải dữ liệu hàng loạt, sau đó xây dựng lại chỉ mục.

### 1.3.2 Sự chuyển dịch sang kiến trúc ELT hiện đại

Sự ra đời của các kho dữ liệu đám mây như Google BigQuery hay Snowflake đã tạo ra sự chuyển dịch mạnh mẽ từ ETL sang ELT.

**Kiến trúc ELT (Extract, Load, Transform)**

1. **E (Extract)**: Tương tự như ETL, trích xuất dữ liệu từ nguồn.
2. **L (Load)**: Thay vì biến đổi trước, dữ liệu thô, kể cả dữ liệu bán cấu trúc được tải thẳng vào kho dữ liệu đám mây đích.
3. **T (Transform)**: Các phép biến đổi phức tạp được thực hiện ngay bên trong kho dữ liệu đám mây.

**Ưu điểm của các nền tảng đám mây trong ELT**

- **Sức mạnh tính toán**: Các nền tảng cho phép chạy các phép biến đổi phức tạp trên hàng tỷ dòng dữ liệu nhanh hơn so với một máy chủ ETL riêng biệt.
- **Chi phí linh hoạt**: Chi phí lưu trữ trên đám mây thấp và mô hình chi phí dựa trên nhu cầu khiến việc lưu trữ dữ liệu thô khả thi về mặt kinh tế.
- **Linh hoạt với dữ liệu thô**: ELT cho phép lưu trữ dữ liệu thô ở định dạng gốc, giúp các nhà khoa học dữ liệu dễ dàng khám phá và xây dựng mô hình.

# Chương 2

# Khảo sát hệ thống

## 2.1 Khảo sát nhu cầu của các bên liên quan

### 2.1.1 Nhu cầu của các bên liên quan

Trong bối cảnh phân tích bóng đá chuyên nghiệp, các nhóm người dùng chính và nhu cầu đặc thù của họ được xác định như sau:

1. **Ban huấn luyện (Huấn luyện viên trưởng, Trợ lý, Giám đốc kỹ thuật, Bác sĩ)**

   - **Nhu cầu về công cụ phân tích hiệu suất đội nhà**:
     - Đánh giá hiệu suất của từng cầu thủ sau mỗi trận đấu thông qua các chỉ số thống kê cơ bản và nâng cao (ví dụ: $xG$, $xA$, tỷ lệ chuyền bóng chính xác, số lần thu hồi bóng).
     - Xác định chiến thuật, điểm mạnh, điểm yếu trong lối chơi của toàn đội (ví dụ: phân tích các pha chuyển đổi trạng thái, khả năng tận dụng tình huống cố định).
     - Cần các báo cáo trực quan cho phép so sánh hiệu suất của cầu thủ và toàn đội qua nhiều trận đấu, giai đoạn khác nhau của mùa giải.
   - **Nhu cầu về hệ thống hỗ trợ phân tích đối thủ chuyên sâu**:
     - Nghiên cứu lối chơi, sơ đồ chiến thuật ưa thích, xu hướng tấn công/phòng ngự của đối thủ.
     - Xác định các cầu thủ then chốt, các mối đe dọa chính và các điểm yếu có thể khai thác của đối thủ.
     - Truy vấn được dữ liệu lịch sử đối đầu và hiệu suất của đối thủ khi chạm trán các đội có lối chơi tương tự.

- **Nhu cầu về dữ liệu để tối ưu hóa kế hoạch tập luyện**: Dữ liệu về thể chất và hiệu suất của cầu thủ cần được cung cấp để Ban huấn luyện có thể thiết kế, điều chỉnh các giáo án tập luyện, dinh dưỡng phù hợp, cá nhân hóa nhằm cải thiện điểm yếu và tránh quá tải. Ngoài ra còn giúp dự báo phòng tránh chấn thương, theo dõi quá trình hồi phục.

2. **Bộ phận tuyển trạch và quản lý thể thao (Tuyển trạch viên, Giám đốc thể thao)**

   - **Nhu cầu hỗ trợ quá trình tuyển trạch và tìm kiếm tài năng một cách khoa học**:
     - Sàng lọc cầu thủ từ một tập dữ liệu lớn (hàng trăm cầu thủ từ nhiều đội bóng) dựa trên các tiêu chí cụ thể (ví dụ: độ tuổi, vị trí, quốc tịch, các chỉ số hiệu suất).
     - So sánh khách quan các cầu thủ tiềm năng ở cùng một vị trí để tìm ra lựa chọn tối ưu nhất.
     - Phát hiện các cầu thủ có chỉ số thống kê ấn tượng nhưng chưa được thị trường chuyển nhượng chú ý.
   - **Nhu cầu về việc xây dựng hồ sơ dữ liệu đa chiều về cầu thủ**: Cho phép tạo ra một cái nhìn toàn diện về một cầu thủ, bao gồm lịch sử thi đấu, sự tiến bộ qua các mùa giải, phong cách chơi, sự phù hợp với triết lý của câu lạc bộ. Hệ thống phải cho phép thực hiện các truy vấn phức tạp.

3. **Ban lãnh đạo (Chủ tịch, Giám đốc điều hành)**

   - **Nhu cầu về cái nhìn tổng quan, mang tính chiến lược**: Cung cấp các báo cáo cấp cao về hiệu suất tổng thể của đội bóng, sự phát triển của các tài năng trẻ, và hiệu quả hoạt động của Ban huấn luyện.
   - **Nhu cầu đánh giá hiệu quả đầu tư**: Hệ thống cần cung cấp dữ liệu để đánh giá mức độ thành công của các thương vụ chuyển nhượng, so sánh giữa chi phí bỏ ra và đóng góp chuyên môn của cầu thủ.
   - **Nhu cầu hỗ trợ việc ra quyết định dài hạn**: Dữ liệu từ kho phải là một nguồn tham khảo quan trọng cho các quyết định chiến lược như gia hạn hợp đồng với cầu thủ, đầu tư vào học viện đào tạo trẻ, hay định hướng phát triển chuyên môn của câu lạc bộ trong 3-5 năm tới.

4. **Các cầu thủ chuyên nghiệp**

    - **Nhu cầu tự đánh giá và phát triển cá nhân**: Cầu thủ cần truy cập vào dashboard cá nhân để xem lại hiệu suất của mình sau mỗi trận đấu. Dữ liệu giúp họ nhận ra điểm mạnh và điểm yếu của bản thân.
    - **Nhu cầu so sánh và đặt mục tiêu**: Hệ thống cần cho phép cầu thủ so sánh các chỉ số của mình với chính họ trong quá khứ hoặc với những cầu thủ khác ở cùng vị trí, từ đó đặt ra các mục tiêu phát triển cụ thể.
    - **Nhu cầu về dữ liệu trong đàm phán hợp đồng**: Các số liệu thống kê về hiệu suất là bằng chứng thuyết phục để cầu thủ (và người đại diện) sử dụng trong các cuộc đàm phán về lương thưởng hoặc gia hạn hợp đồng.

5. **Bộ phận truyền thông và marketing** Bộ phận này có nhiệm vụ xây dựng hình ảnh câu lạc bộ, kết nối với người hâm mộ và tối đa hóa các cơ hội thương mại. Dữ liệu là nguồn tài nguyên quý giá để họ sáng tạo nội dung.

    - **Nhu cầu tìm kiếm các câu chuyện và thống kê thú vị**: Hệ thống cần cho phép truy vấn để tìm ra các cột mốc, kỷ lục hoặc các chỉ số thống kê đặc biệt (ví dụ: "Cầu thủ X sắp có bàn thắng thứ 100 cho câu lạc bộ").
    - **Nhu cầu sản xuất nội dung số hấp dẫn**: Dữ liệu là nền tảng để tạo ra các sản phẩm đồ họa thông tin, video phân tích ngắn cho các nền tảng mạng xã hội, giúp tăng tương tác với cộng đồng người hâm mộ.
    - **Nhu cầu cá nhân hóa trải nghiệm người hâm mộ**: Phân tích dữ liệu về các cầu thủ được yêu thích có thể giúp đưa ra các chiến dịch quảng bá sản phẩm (áo đấu, vật phẩm lưu niệm,...) hiệu quả hơn.

### 2.1.2 Các báo cáo cần xây dựng và chủ điểm phân tích

**Nhóm báo cáo phân tích diễn biến trận đấu và chiến thuật:** Đây là nhóm báo cáo phục vụ việc đánh giá thế trận, kiểm soát bóng và hiệu quả chiến thuật. Từ đó trả lời các câu hỏi phân tích:

- Đội bóng kiểm soát thế trận ra sao trong từng giai đoạn của trận đấu?
- Mức độ gây áp lực lên đối thủ và khả năng đoạt lại bóng hiệu quả ra sao?
- Chất lượng các cơ hội tạo ra và nguy cơ nhận bàn thua là bao nhiêu?

**Nhóm báo cáo đánh giá hiệu suất cầu thủ:** Nhóm báo cáo này cung cấp dữ liệu chi tiết về đóng góp của từng cá nhân, phục vụ cho việc đánh giá phong độ và điều chỉnh nhân sự. Từ đó trả lời các câu hỏi sau:

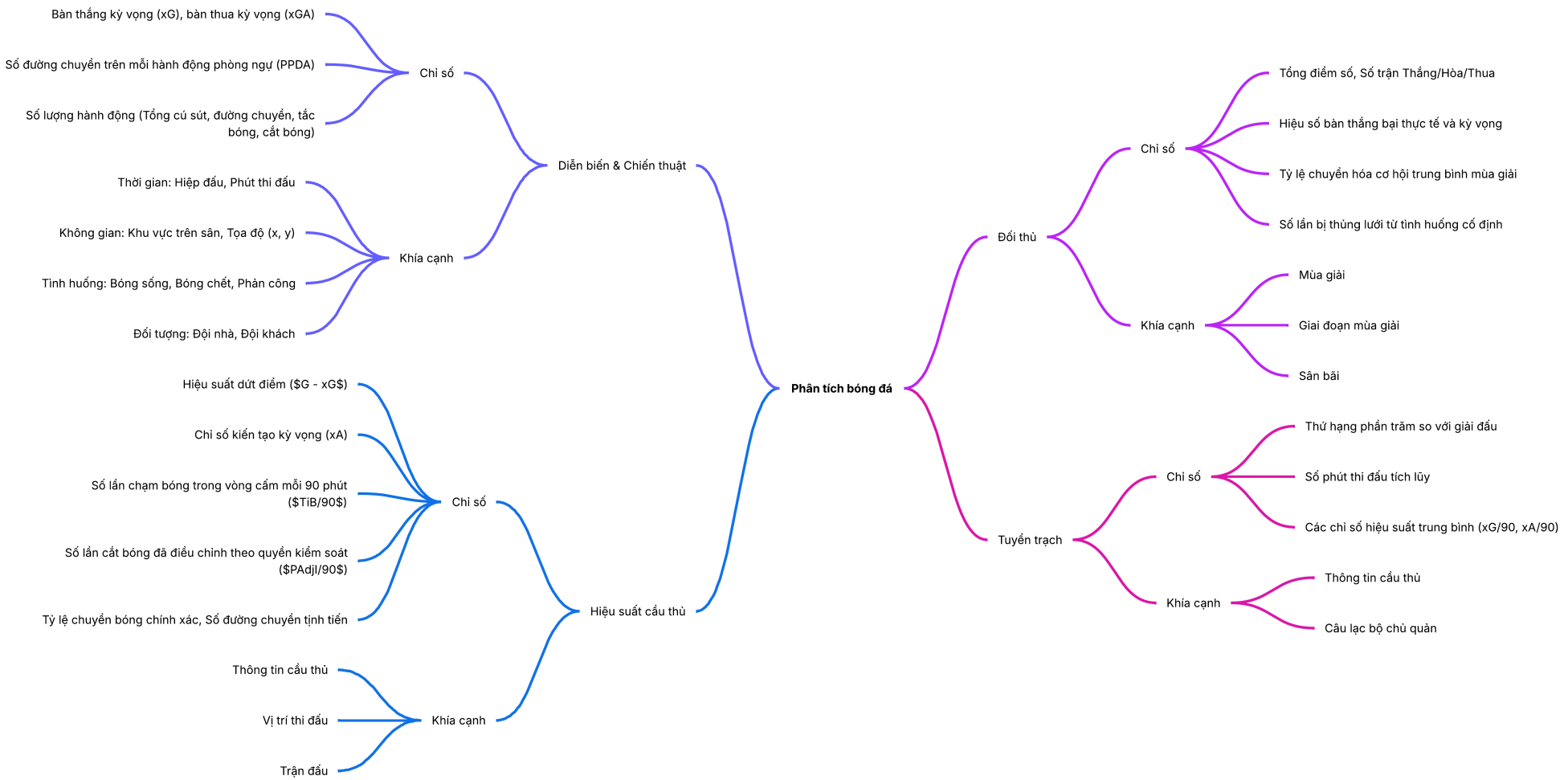
- Cầu thủ nào có khả năng dứt điểm thành bàn tốt hơn so với kỳ vọng?
- Mức độ đóng góp vào mặt trận tấn công và khả năng chọn vị trí trong vòng cấm ra sao?
- Hiệu quả phòng ngự của cầu thủ khi đã tính đến thời lượng kiểm soát bóng của đối phương?

**Nhóm báo cáo phân tích đội nhà, đối thủ:** Chức năng này cung cấp dữ liệu lịch sử của đội nhà hoặc đối thủ sắp tới, hỗ trợ Ban huấn luyện xây dựng đấu pháp phù hợp. Từ đó trả lời các câu hỏi phân tích:

- Phong độ tổng thể của đội nhà, đối thủ gần đây như thế nào?
- Điểm mạnh và điểm yếu trong lối chơi của đội nhà, đối thủ là gì?
- Hiệu quả thi đấu của đội nhà, đối thủ khi đá sân nhà so với sân khách ra sao?

**Nhóm báo cáo tuyển trạch:** Giúp bộ phận tuyển trạch có thể sàng lọc và tìm kiếm các ứng viên tiềm năng. Từ đó trả lời các câu hỏi sau:

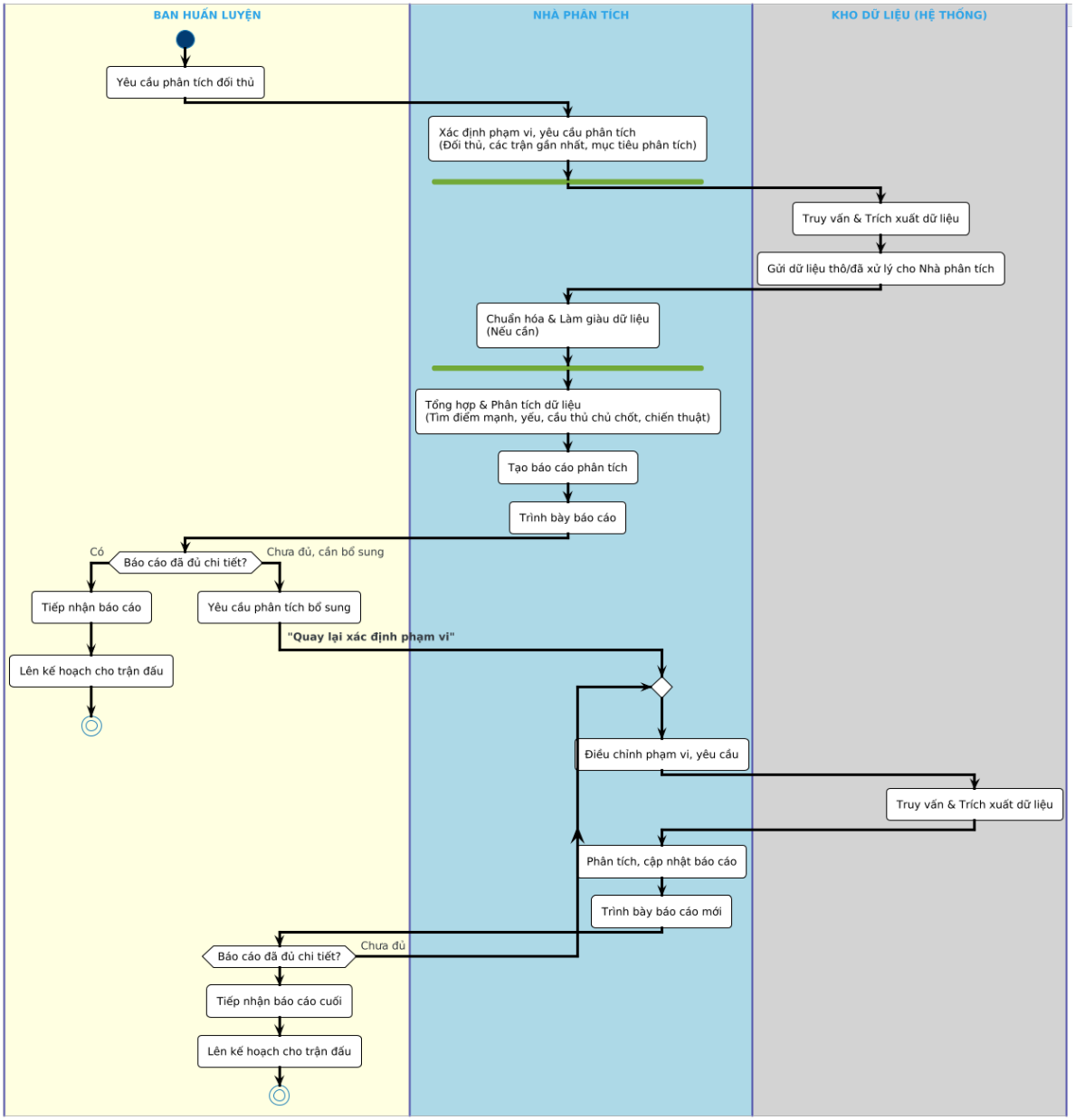
- Cầu thủ mục tiêu đang đứng ở đâu so với mặt bằng chung của giải đấu?
- Kinh nghiệm thi đấu và sự ổn định phong độ của cầu thủ thể hiện qua các chỉ số trung bình như thế nào?




Hình 2.1: Mindmap nhu cầu phân tích

## 2.2 Các luồng nghiệp vụ khai thác dữ liệu bóng đá

### 2.2.1 Luồng nghiệp vụ phân tích đối thủ




Hình 2.2: Luồng nghiệp vụ phân tích đối thủ

1. **Mục tiêu**: Thu thập, tổng hợp và phân tích dữ liệu về các đối thủ sắp tới nhằm phân tích chiến thuật, điểm mạnh, điểm yếu và các nhân sự chủ chốt. Kết quả của nghiệp vụ này là các báo cáo phục vụ cho Ban huấn luyện trong việc xây dựng chiến lược và kế hoạch chuẩn bị cho trận đấu.

2. **Các bên liên quan**

   - **Ban huấn luyện**: Đưa ra yêu cầu phân tích và sử dụng các báo cáo để

ra quyết định về chiến thuật, nhân sự và phương án thi đấu.

- **Nhà phân tích**: Sử dụng các công cụ để khai thác thông tin và chuyển hóa dữ liệu thô thành các báo cáo.
- **Kho dữ liệu**: Nguồn cung cấp dữ liệu tập trung, chứa thông tin về các trận đấu, cầu thủ, đội bóng,...

3. **Mô tả quy trình**

**Bước 1: Khởi tạo yêu cầu**: Khi có lịch thi đấu, Ban huấn luyện sẽ gửi yêu cầu cho Nhà phân tích để bắt đầu tìm hiểu về đối thủ cụ thể.

**Bước 2: Xác định phạm vi và thu thập dữ liệu**: Nhà phân tích làm việc với Ban huấn luyện để xác định phạm vi cần phân tích, dựa vào đó để thực hiện truy vấn và trích xuất dữ liệu thô cần thiết từ Kho dữ liệu.

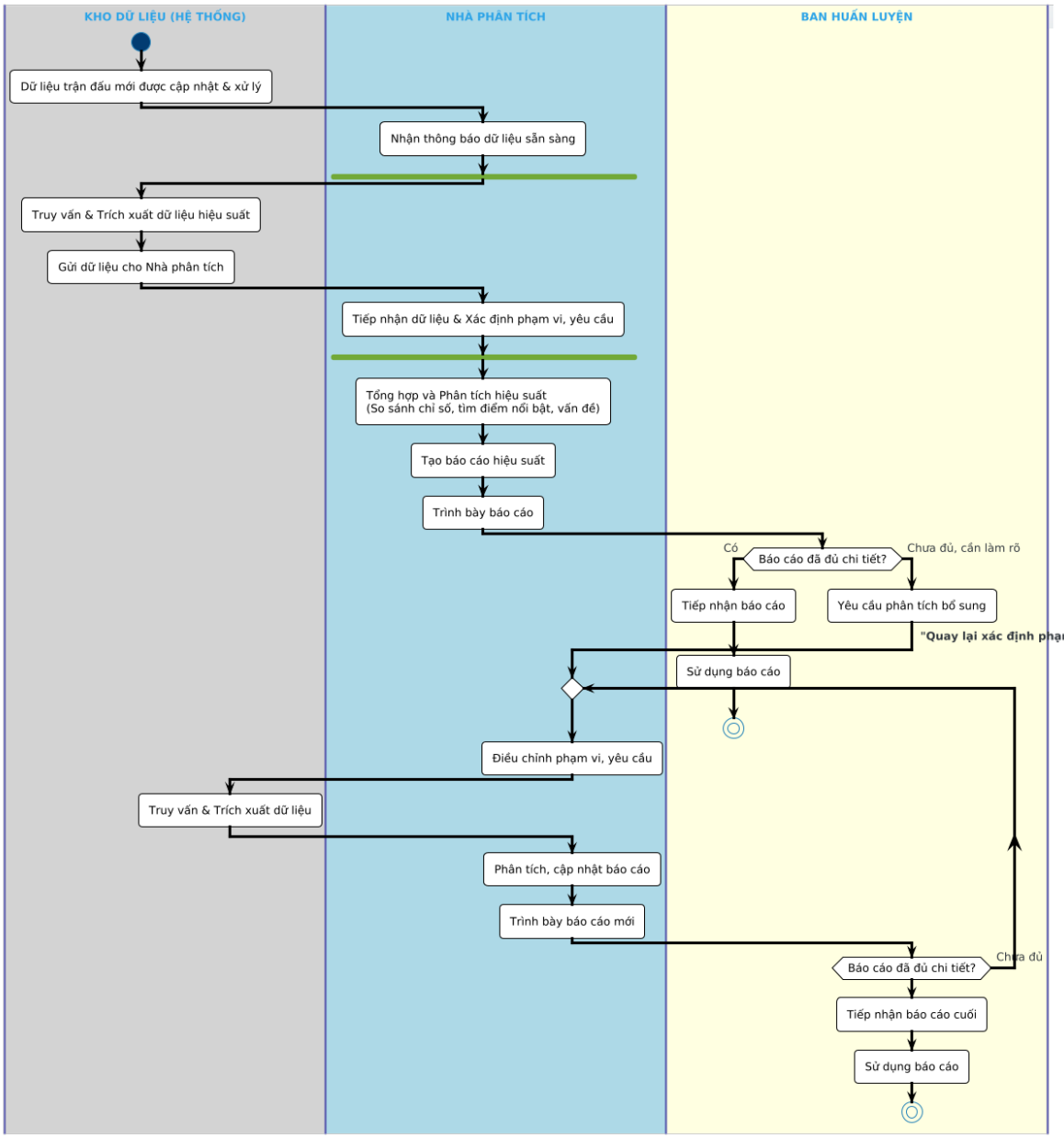
**Bước 3: Tổng hợp, xử lý và phân tích**: Dữ liệu thô được trích xuất sẽ được làm sạch, chuẩn hóa và tổng hợp. Nhà phân tích phân tích các xu hướng trong lối chơi, hiệu suất cầu thủ và điểm mạnh/yếu của đối thủ.

**Bước 4: Tạo và trình bày báo cáo**: Kết quả phân tích được trình bày dưới dạng một báo cáo hoàn chỉnh, bao gồm các số liệu, biểu đồ trực quan và nhận định chuyên môn, sau đó được gửi đến cho Ban huấn luyện.

**Bước 5: Phản hồi và hiệu chỉnh**: Ban huấn luyện xem xét báo cáo. Nếu chưa đạt, họ sẽ yêu cầu bổ sung. Quy trình quay lại bước 2 hoặc 3 để Nhà phân tích thực hiện phân tích sâu hơn và cập nhật lại báo cáo.

**Bước 6: Hoàn tất và ứng dụng**: Khi báo cáo cuối cùng được phê duyệt, Ban huấn luyện sử dụng báo cáo để lên kế hoạch cho các buổi tập và xây dựng chiến thuật cho trận đấu. Nghiệp vụ kết thúc.

### 2.2.2 Luồng nghiệp vụ phân tích hiệu suất đội nhà




Hình 2.3: Luồng nghiệp vụ phân tích hiệu suất đội nhà

1. **Mục tiêu**: Đánh giá hiệu suất thi đấu của đội nhà sau mỗi trận đấu. Kết quả phân tích cung cấp dữ liệu khách quan để Ban huấn luyện điều chỉnh chiến thuật, giáo án tập luyện và chuẩn bị cho các trận đấu trong tương lai.

2. **Các bên liên quan**

   - **Ban huấn luyện**: Đưa ra các yêu cầu phân tích và áp dụng kết quả phân tích.

- **Nhà phân tích**: Thực hiện quy trình phân tích, trích xuất dữ liệu, xử lý, tìm kiếm thông tin và tạo các báo cáo.
- **Kho dữ liệu**: Tự động cập nhật, xử lý dữ liệu từ các trận đấu mới nhất và cung cấp nguồn dữ liệu sẵn sàng cho Nhà phân tích khai thác.

3. **Mô tả quy trình**

   **Bước 1: Cập nhật dữ liệu sau trận đấu**: Sau khi một trận đấu kết thúc, Kho dữ liệu tự động cập nhật và xử lý dữ liệu liên quan.

   **Bước 2: Khởi tạo phân tích và thu thập dữ liệu**: Nhà phân tích tiếp nhận yêu cầu, xác định phạm vi phân tích ban đầu và tiến hành truy vấn, trích xuất dữ liệu hiệu suất chi tiết của đội nhà từ Kho dữ liệu.

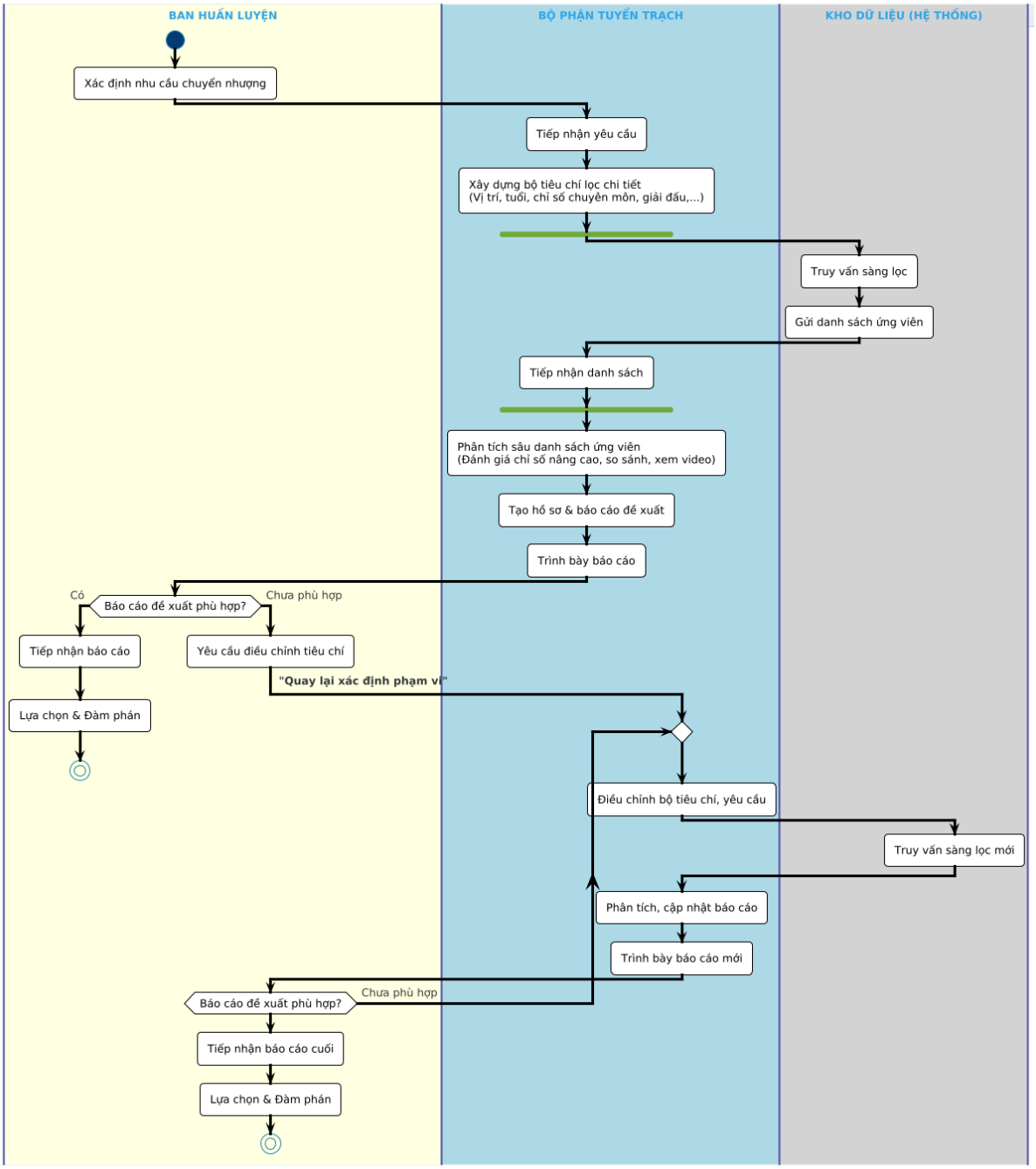
   **Bước 3: Phân tích và tạo báo cáo**: Nhà phân tích tổng hợp dữ liệu, thực hiện so sánh các chỉ số, tìm ra những điểm tích cực, tiêu cực và các vấn đề tồn đọng, từ đó tạo ra báo cáo hiệu suất.

   **Bước 4: Trình bày và xem xét**: Báo cáo được trình bày cho Ban huấn luyện để đánh giá xem báo cáo đã đáp ứng yêu cầu chuyên môn chưa.

   **Bước 5: Phản hồi và hiệu chỉnh**: Nếu báo cáo chưa đạt, Ban huấn luyện sẽ yêu cầu bổ sung. Nhà phân tích điều chỉnh phạm vi, khai thác dữ liệu, cập nhật báo cáo. Quá trình lặp lại đến khi báo cáo đạt yêu cầu.

   **Bước 6: Hoàn tất và ứng dụng**: Khi báo cáo cuối cùng được phê duyệt, Ban huấn luyện sẽ tiếp nhận và sử dụng báo cáo để phục vụ cho công tác chuyên môn. Nghiệp vụ kết thúc.

### 2.2.3 Luồng nghiệp vụ tuyển trạch cầu thủ




Hình 2.4: Luồng nghiệp vụ tuyển trạch cầu thủ

1. **Mục tiêu**: Tìm kiếm và xác định các cầu thủ tiềm năng phù hợp đội bóng. Cung cấp cho Ban huấn luyện một danh sách các ứng viên đã qua sàng lọc, làm cơ sở cho các quyết định chuyển nhượng.

2. **Các bên liên quan**

   - **Ban huấn luyện**: Đưa ra nhu cầu chuyển nhượng và ra quyết định cuối cùng trong việc lựa chọn và đàm phán.

- **Bộ phận tuyển trạch**: Xây dựng tiêu chí lọc, phân tích dữ liệu, đánh giá chuyên môn và tạo báo cáo đề xuất các ứng viên tiềm năng.
- **Kho dữ liệu**: Cung cấp một cơ sở dữ liệu lớn về cầu thủ để sàng lọc, truy vấn theo các tiêu chí phức tạp do bộ phận tuyển trạch xây dựng.

3. **Mô tả quy trình**

   **Bước 1: Xác định nhu cầu**: Ban huấn luyện xác định nhu cầu nhân sự và gửi yêu cầu đến Bộ phận tuyển trạch.

   **Bước 2: Xây dựng tiêu chí và sàng lọc**: Bộ phận tuyển trạch tiếp nhận yêu cầu và cụ thể hóa thành một bộ tiêu chí lọc, sau đó thực hiện truy vấn trên Kho dữ liệu để có được danh sách ứng viên sơ bộ.

   **Bước 3: Phân tích và tạo báo cáo đề xuất**: Bộ phận tuyển trạch tiến hành phân tích sâu danh sách ứng viên, đánh giá các chỉ số, so sánh các cầu thủ để tạo ra báo cáo đề xuất ban đầu.

   **Bước 4: Trình bày và xem xét**: Báo cáo được trình bày cho Ban huấn luyện để đánh giá mức độ phù hợp của các ứng viên được đề xuất.

   **Bước 5: Phản hồi và hiệu chỉnh**: Nếu báo cáo chưa phù hợp hoặc cần tìm kiếm thêm, bộ phận tuyển trạch cập nhật lại bộ lọc, thực hiện truy vấn mới và lặp lại quá trình phân tích để tổng hợp lại báo cáo mới.

   **Bước 6: Hoàn tất và lựa chọn**: Khi báo cáo cuối cùng đã đáp ứng được yêu cầu, Ban huấn luyện sẽ tiếp nhận, lựa chọn ứng viên phù hợp và bắt đầu quá trình đàm phán chuyển nhượng. Nghiệp vụ kết thúc.

## 2.3 Mô hình kinh doanh và Luồng dữ liệu

### 2.3.1 Mô hình kinh doanh


Hình 2.5: Mô hình kinh doanh

1. **Đối tác chính (Key Partners):**
   - **Nhà cung cấp dữ liệu:** Các đơn vị như StatsBomb cung cấp dữ liệu sự kiện thô, là nguyên liệu đầu vào quan trọng cho hệ thống phân tích.
   - **Nhà tài trợ:** Các thương hiệu đồng hành cung cấp nguồn tài chính.
   - **Liên đoàn/Ban tổ chức giải:** Đơn vị quản lý, tổ chức giải đấu và phân chia bản quyền truyền hình.
2. **Hoạt động chính (Key Activities):**
   - **Tập luyện, thi đấu:** Hoạt động thường nhật.
   - **Tuyển trạch, chuyển nhượng:** Tìm kiếm, sàng lọc và mua bán cầu thủ để nâng cấp đội hình hoặc kiếm lời.
   - **Quản lý sức khỏe, y tế:** Duy trì thể trạng, ngăn ngừa chấn thương.
   - **Thương mại:** Quảng bá, khai thác giá trị thương hiệu.
3. **Giá trị cung cấp (Value Proposition):**
   - **Thành tích thi đấu cao:** Chiến thắng và các danh hiệu là sản phẩm cốt lõi thu hút người hâm mộ.

- **Hoạt động chuyển nhượng thông minh:** Mua cầu thủ tiềm năng với giá rẻ, phát triển họ và bán lại với giá cao.
- **Thương hiệu mạnh, giải trí:** Cống hiến lối chơi đẹp mắt và trải nghiệm giải trí đỉnh cao.
- **Đào tạo trẻ chất lượng:** Hệ thống lò đào tạo bài bản cung cấp nguồn nhân lực kế cận.

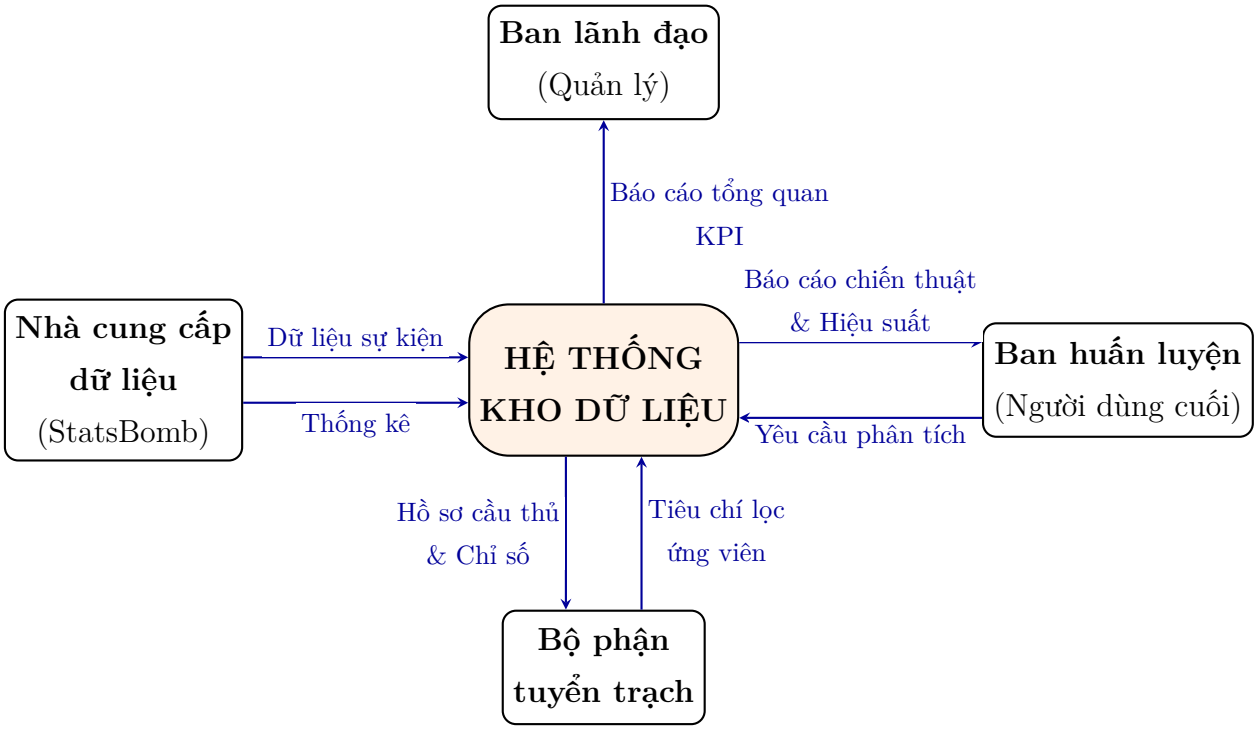
4. **Quan hệ khách hàng (Customer Relationships):**
   - **Cộng đồng người hâm mộ:** Tương tác liên tục qua các kênh mạng xã hội, hội cổ động viên.
   - **Thành viên thân thiết:** Cung cấp các gói ưu đãi cho cổ động viên trung thành.
5. **Phân khúc khách hàng (Customer Segments):**
   - **Người hâm mộ:** Khán giả đến sân hoặc theo dõi qua truyền hình.
   - **Nhà tài trợ:** Các doanh nghiệp muốn quảng bá thương hiệu gắn liền với hình ảnh đội bóng.
   - **Các đài truyền hình/Đơn vị bản quyền:** Đối tác mua bản quyền phát sóng giải đấu.
   - **Các CLB khác:** Đối tác mua/bán cầu thủ.
6. **Tài nguyên chính (Key Resources):**
   - **Cầu thủ:** Tài sản giá trị nhất của đội bóng.
   - **Hệ thống dữ liệu, phân tích:** Kho dữ liệu và đội ngũ phân tích.
   - **Sân vận động, cơ sở tập luyện:** Hạ tầng vật chất phục vụ thi đấu.
   - **Thương hiệu, hình ảnh:** Giá trị vô hình giúp thu hút tài trợ.
7. **Kênh phân phối (Channels):**
   - **Sân vận động:** Nơi diễn ra trận đấu.
   - **Truyền thông số:** Website, Ứng dụng, Mạng xã hội của CLB.
   - **Cửa hàng:** Kênh bán vé và vật phẩm lưu niệm.
8. **Cơ cấu chi phí (Cost Structure):**

- **Lương:** Khoản chi phí vận hành lớn nhất.
- **Phí chuyển nhượng:** Chi phí khấu hao khi mua cầu thủ.
- **Chi phí vận hành hệ thống:** Hạ tầng máy chủ, nhân sự phân tích.
- **Chi phí vận hành sân bãi và học viện đào tạo.**

9. **Nguồn doanh thu (Revenue Streams):**

- **Doanh thu chuyển nhượng:** CLB kiếm lợi nhuận từ việc bán cầu thủ.
- **Tiền thưởng, tiền bản quyền.**
- **Doanh thu ngày thi đấu:** Bán vé vào sân xem trận đấu.
- **Tài trợ, quảng cáo.**

### 2.3.2 Luồng dữ liệu




Hình 2.6: Sơ đồ luồng dữ liệu

Dữ liệu đầu vào được thu thập từ GitHub của StatsBomb, sau đó được xử lý và lưu trữ tập trung trong kho dữ liệu.

Ban lãnh đạo khai thác các báo cáo tổng quan và chỉ số KPI nhằm phục vụ công tác quản lý và đánh giá. Ban huấn luyện sử dụng các báo cáo phân tích chiến thuật và hiệu suất thi đấu dựa trên các yêu cầu phân tích cụ thể để hỗ trợ công

tác huấn luyện và thi đấu. Đồng thời, bộ phận tuyển trạch khai thác hồ sơ cầu thủ và các chỉ số chuyên môn để xây dựng tiêu chí lọc và đánh giá ứng viên.

Luồng dữ liệu hai chiều giữa các bộ phận nghiệp vụ và hệ thống kho dữ liệu nhằm đáp ứng các yêu cầu phân tích khác nhau, đồng thời đảm bảo dữ liệu được khai thác nhất quán, chính xác và hiệu quả.

## 2.4 Đặc tả yêu cầu kỹ thuật

### 2.4.1 Yêu cầu về quy trình xử lý dữ liệu

Hệ thống phải đảm bảo khả năng vận hành tự động toàn bộ vòng đời dữ liệu, bao gồm các năng lực cụ thể:

- **Khả năng tích hợp đa nguồn:**
  - Hệ thống phải tự động kết nối và trích xuất dữ liệu định kỳ từ nguồn dữ liệu của StatsBomb.
  - Hỗ trợ cơ chế tải dữ liệu tăng trưởng để chỉ cập nhật các trận đấu mới diễn ra, tối ưu băng thông và thời gian xử lý.
- **Khả năng biến đổi và làm giàu dữ liệu:**
  - Thực hiện quy trình ETL/ELT để làm sạch, chuẩn hóa tên cầu thủ/đội bóng và xử lý các giá trị thiếu hoặc sai lệch.
  - Tính toán tự động các chỉ số nâng cao không có sẵn trong dữ liệu gốc để phục vụ trực tiếp cho tầng phân tích.
- **Khả năng phục vụ phân tích:**
  - Tổ chức dữ liệu theo mô hình đa chiều (Star Schema) tại tầng Data Warehouse để tối ưu hiệu năng cho các truy vấn phức tạp của công cụ BI.
  - Cung cấp các Data Mart chuyên biệt cho từng nghiệp vụ: Tuyển trạch, Phân tích trận đấu, và Quản trị chiến lược.

### 2.4.2 Tiêu chuẩn chất lượng và hiệu năng

Hệ thống phải đáp ứng các tiêu chuẩn kỹ thuật sau để đảm bảo trải nghiệm người dùng và độ tin cậy:

1. **Tính toàn vẹn và Chính xác:**
    - Đảm bảo toàn vẹn dữ liệu trong quá trình nạp từ nguồn vào Data Lake.
    - Dữ liệu sau khi xử lý phải đảm bảo tính nhất quán giữa các bảng Fact và Dimension.
2. **Khả năng mở rộng và Ổn định:**
    - Hệ thống phải có khả năng xử lý khối lượng dữ liệu tăng dần theo từng mùa giải mà không làm giảm hiệu năng truy vấn.
    - Luồng dữ liệu phải có cơ chế tự động thử lại khi gặp lỗi kết nối và gửi cảnh báo đến kỹ sư vận hành.

### 2.4.3 Công nghệ sử dụng

- **Docker**: Được sử dụng để container hóa toàn bộ các thành phần của hệ thống, đảm bảo tính nhất quán giữa các môi trường.
- **Apache Airflow**: Được sử dụng làm công cụ điều phối, lập lịch và giám sát các luồng xử lý dữ liệu.
- **MinIO**: Được sử dụng làm Data Lake (lưu trữ đối tượng) để chứa dữ liệu thô và dữ liệu trung gian.
- **Apache Spark**: Được sử dụng làm công cụ xử lý dữ liệu phân tán, chịu trách nhiệm cho các tác vụ biến đổi dữ liệu phức tạp và quy mô lớn.
- **PostgreSQL**: Được sử dụng làm Data Warehouse, lưu trữ dữ liệu có cấu trúc đã được làm sạch và sẵn sàng cho việc truy vấn phân tích.
- **Microsoft PowerBI**: Được sử dụng làm công cụ BI để kết nối tới PostgreSQL, xây dựng các mô hình dữ liệu và tạo các báo cáo, dashboard.

## 2.5 Đặc điểm và quy mô dữ liệu

Nguồn dữ liệu từ GitHub của StatsBomb là nền tảng cho Data Warehouse.

- **Đặc điểm:**
    - **Định dạng:** JSON bán cấu trúc.

  - **Cấu trúc:** Phức tạp, lồng nhau nhiều cấp. Một bản ghi sự kiện chứa nhiều object con như `tactics.lineup`, `shot.freeze_frame` (vị trí 22 cầu thủ), `location[x,y]`.

- **Quy mô:**

  - **Số lượng bản ghi:** Khoảng **2.000.000 – 3.000.000** sự kiện. Trung bình một trận đấu chứa khoảng 3.500 sự kiện.
  - **Dung lượng lưu trữ:** Khoảng **1.5 GB – 2.0 GB** dữ liệu thô (JSON).

- **Thách thức kỹ thuật:**

  - Tuy dung lượng lưu trữ không quá lớn nhưng độ phức tạp của cấu trúc JSON yêu cầu tài nguyên tính toán lớn để thực hiện quá trình làm phẳng. Đây là lý do chính cho việc sử dụng **Apache Spark**.

**Ghi chú:** Dữ liệu sử dụng trong đồ án được lấy từ **StatsBomb Free dataset** cho câu lạc bộ **FC Barcelona** thuộc giải đấu La Liga. Do giới hạn dữ liệu mở, đề tài chọn Barcelona làm *case study* để minh hoạ quy trình xây dựng kho dữ liệu và báo cáo phân tích.

# Chương 3

# Thiết kế hệ thống

## 3.1 Khám phá dữ liệu

### 3.1.1 Tổng quan về cấu trúc dữ liệu

```
1. Bảng Matches (1 Mùa giải): 36 bản ghi (trận đấu)
2. Bảng Events (1 Trận):    3831 bản ghi (sự kiện)
3. Bảng Lineups (1 Trận):   2 bản ghi (2 đội bóng)
4. Định dạng file: JSON (Nested Structure)
```

Hình 3.1: Tổng quan về cấu trúc các file dữ liệu dạng JSON

Quá trình khảo sát với Apache Spark cho thấy dữ liệu từ StatsBomb được tổ chức thành 3 nhóm đối tượng chính: Matches (Thông tin trận đấu), Lineups (Danh sách đăng ký thi đấu) và Events (Chi tiết sự kiện). Dữ liệu này được lưu trữ dưới dạng JSON lồng nhau thay vì dạng bảng phẳng truyền thống, phản ánh độ phức tạp cao của các tình huống trong bóng đá.

### 3.1.2 Cấu trúc schema

Khi đi sâu vào cấu trúc schema, có thể thấy dữ liệu không tồn tại độc lập mà có tính liên kết chặt chẽ. Các trường thông tin quan trọng như location (tọa độ), shot (cú sút) hay pass (đường chuyền) không phải là kiểu dữ liệu nguyên thủy mà là các cấu trúc phức tạp (Struct hoặc Array). Vì vậy, một dòng sự kiện đơn lẻ chứa hàng chục thông tin con cần được bóc tách kỹ lưỡng.

**Bảng Matches (Trận đấu)**

| **Tên trường** | **Kiểu dữ liệu** | **Mô tả** |
|---|---|---|
| match_id | Long | Khóa chính của trận đấu. |
| match_date | String | Ngày diễn ra trận đấu (YYYY-MM-DD). |
| kick_off | String | Thời gian bắt đầu trận đấu. |
| home_team | Struct | Đội nhà (home_team_id, home_team_name,...). |
| away_team | Struct | Đội khách (away_team_id, away_team_name,...). |
| home_score | Long | Số bàn thắng của đội nhà. |
| away_score | Long | Số bàn thắng của đội khách. |
| competition | Struct | Giải đấu (id, name, country_name). |
| season | Struct | Mùa giải (season_id, season_name). |

Bảng 3.1: Tóm tắt cấu trúc dữ liệu bảng Matches

**Bảng Events (Sự kiện)**

| **Tên trường** | **Kiểu dữ liệu** | **Mô tả** |
|---|---|---|
| id | String | Khóa chính của sự kiện. |
| index | Long | Số thứ tự của sự kiện trong trận đấu. |
| timestamp | String | Thời điểm xảy ra sự kiện (phút:giây.miligiây). |
| type | Struct | Loại sự kiện (Pass, Shot,...). |
| possession_team | Struct | Đội đang kiểm soát bóng tại thời điểm đó. |
| play_pattern | Struct | Tình huống bóng (From Corner,...). |
| player | Struct | Thông tin cầu thủ thực hiện hành động (id, name). |
| location | Array<Double> | Tọa độ trên sân dạng mảng $[x, y]$. |
| shot | Struct | Chi tiết cú sút: statsbomb_xg, outcome, body_part,... |
| pass | Struct | Chi tiết đường chuyền: length, angle, height,... |
| tactics | Struct | Thông tin đội hình chiến thuật và vị trí. |

Bảng 3.2: Tóm tắt cấu trúc dữ liệu bảng Events

**Bảng Lineups (Đội hình)**

| Tên trường | Kiểu dữ liệu | Mô tả |
|---|---|---|
| team_id | Long | ID của đội bóng. |
| team_name | String | Tên đội bóng. |
| lineup | Array<Struct> | Danh sách cầu thủ đăng ký thi đấu. Dữ liệu là một mảng chứa thông tin cầu thủ. |
| – *element* | *Struct* | *Thông tin chi tiết của cầu thủ trong mảng lineup:* |
| .player_id | Long | ID cầu thủ. |
| .player_name | String | Tên đầy đủ cầu thủ. |
| .jersey_number | Long | Số áo thi đấu. |
| .country | Struct | Quốc tịch cầu thủ. |
| .cards | Array | Danh sách thẻ phạt (nếu có). |

Bảng 3.3: Tóm tắt cấu trúc dữ liệu bảng Lineups

### 3.1.3 Chất lượng dữ liệu

```
1. Số lượng giá trị Null:
+--------------+--------------+---------------+
|Null Location|Null Player ID|Null Timestamp|
+--------------+--------------+---------------+
|            30|            13|              0|
+--------------+--------------+---------------+

2. Số lượng ID sự kiện bị trùng lặp: 0
3. Sự kiện là 'Shot' nhưng thiếu dữ liệu 'shot': 0
```

Hình 3.2: Chất lượng dữ liệu sự kiện

Phân tích trên tập dữ liệu đại diện (một trận El Clásico ở mùa giải 2017/2018) cho thấy chất lượng dữ liệu tương đối tốt nhưng vẫn tồn tại Null. Cụ thể, trường location xuất hiện các giá trị Null ở các sự kiện mang tính thủ tục (như tiếng còi bắt đầu hiệp đấu). Không phát hiện trùng lặp khóa chính (ID) trong mẫu thử.

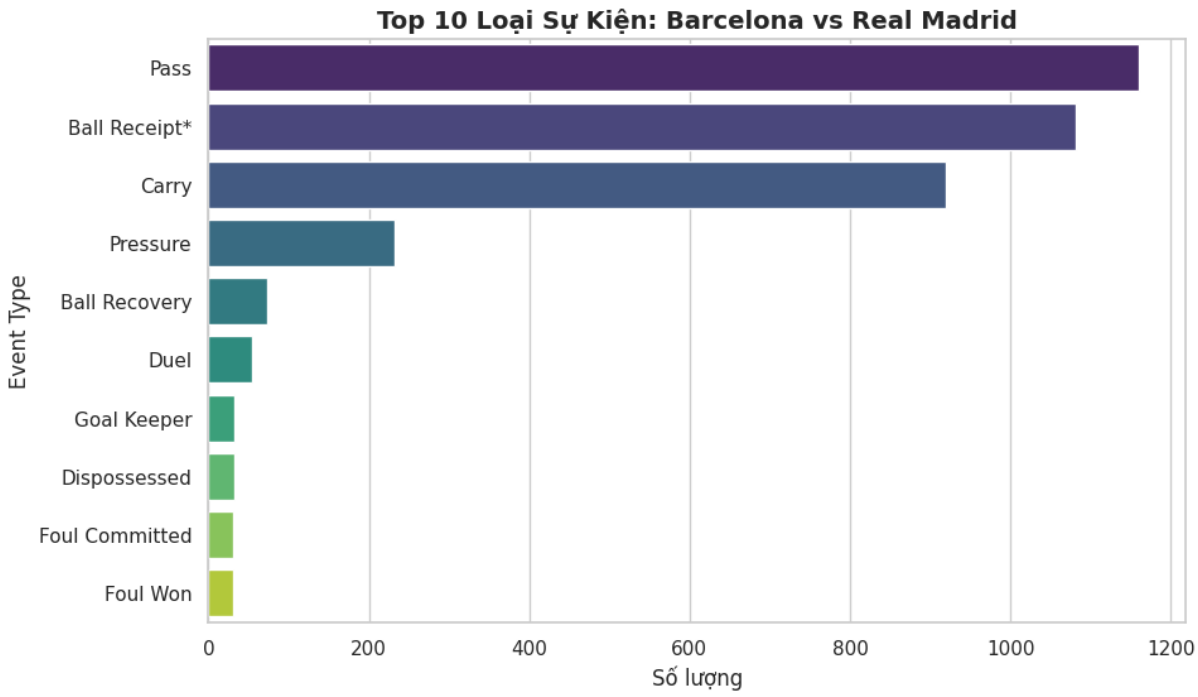
### 3.1.4 Khám phá dữ liệu

Việc hiểu rõ đặc điểm dữ liệu trước khi đưa vào kho là bước quan trọng để định hình chiến lược phân tích. Dựa trên dữ liệu JSON từ StatsBomb, ta thực hiện phân

tích trên 4 khía cạnh chính dưới đây.

### Tổng quan sự kiện trận đấu

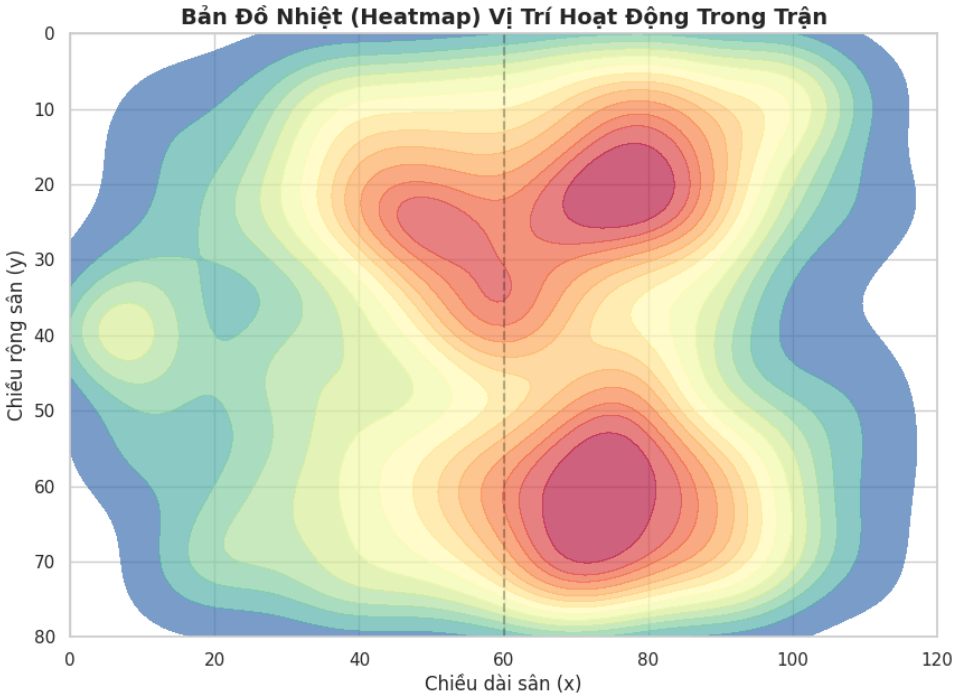
Dữ liệu sự kiện cho thấy sự phân bố không đồng đều giữa các loại hành động. Theo hình 3.3, các hành động mang tính kiểm soát như *Pass* (Chuyền bóng) và *Ball Receipt* (Nhận bóng) chiếm tỷ trọng áp đảo. Trong khi đó, các sự kiện mang tính quyết định trận đấu như *Shot* hay *Goal* là các sự kiện hiếm khi xảy ra. Điều này phản ánh đúng tính chất của bóng đá hiện đại và đặt ra yêu cầu xử lý mất cân bằng dữ liệu khi xây dựng các mô hình dự báo.




Hình 3.3: Top 10 loại sự kiện phổ biến nhất trong một trận đấu mẫu

### Phân tích không gian

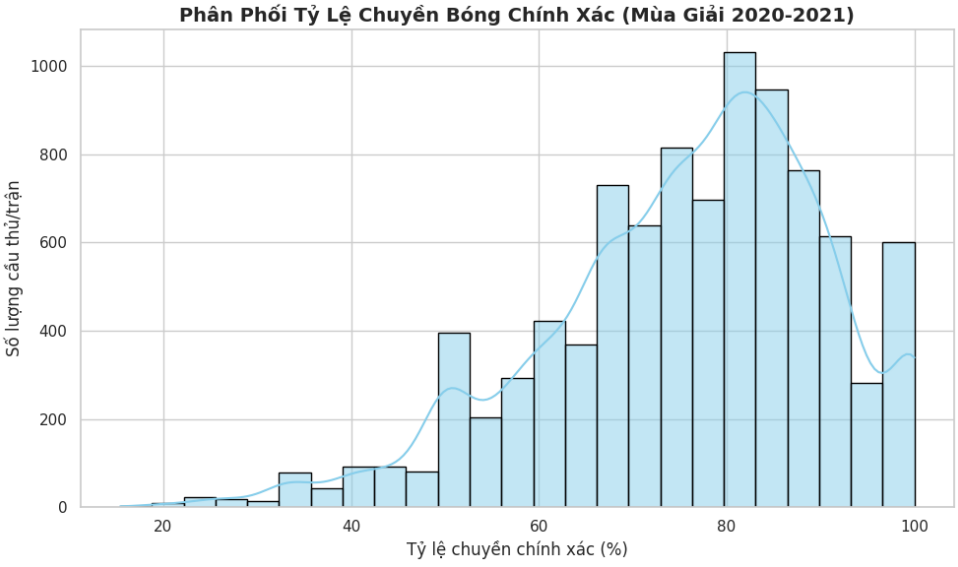
Tận dụng trường thông tin *location* (tọa độ $[x, y]$) được trích xuất từ cấu trúc JSON lồng nhau, ta có thể xây dựng Bản đồ nhiệt (Heatmap) để quan sát mật độ di chuyển của cầu thủ (Hình 3.4). Việc trực quan hóa này chứng minh hệ thống đã xử lý thành công dữ liệu tọa độ thô, tạo tiền đề cho các bài toán phân tích chiến thuật và kiểm soát không gian ở các chương sau.




Hình 3.4: Bản đồ nhiệt (Heatmap) vị trí hoạt động trên sân

### Phân phối kỹ thuật cầu thủ

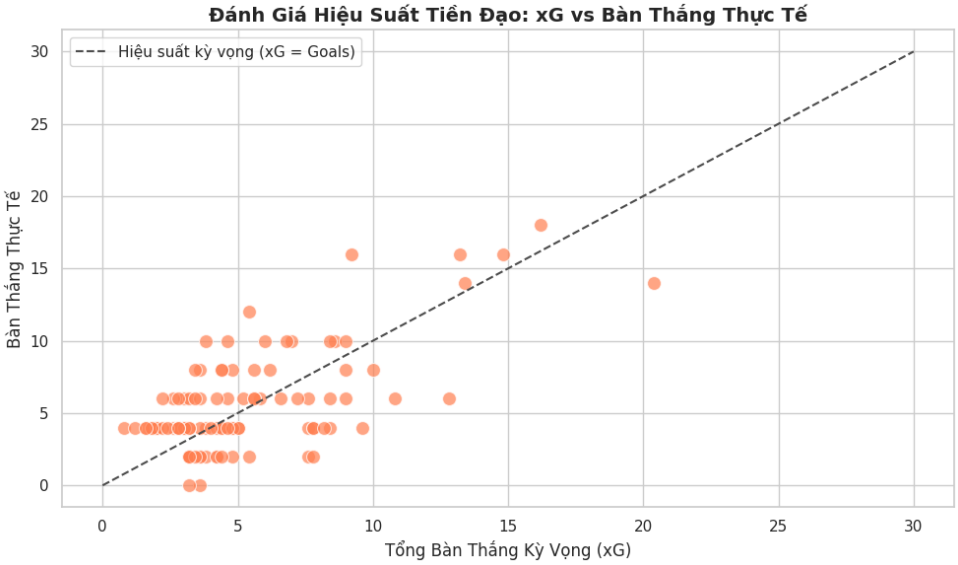
Biểu đồ Histogram (Hình 3.5) mô tả phân phối tỷ lệ chuyền bóng chính xác của các cầu thủ tại giải đấu. Biểu đồ có dạng lệch trái rõ rệt, với đa số cầu thủ duy trì tỷ lệ chuyền bóng thành công trên 75%. Điều này cho thấy mặt bằng kỹ thuật tại giải đấu La Liga là rất cao, đòi hỏi hệ thống phân tích phải có độ nhạy lớn để phân loại được các cầu thủ xuất sắc.




Hình 3.5: Phân phối tỷ lệ chuyền bóng chính xác của cầu thủ

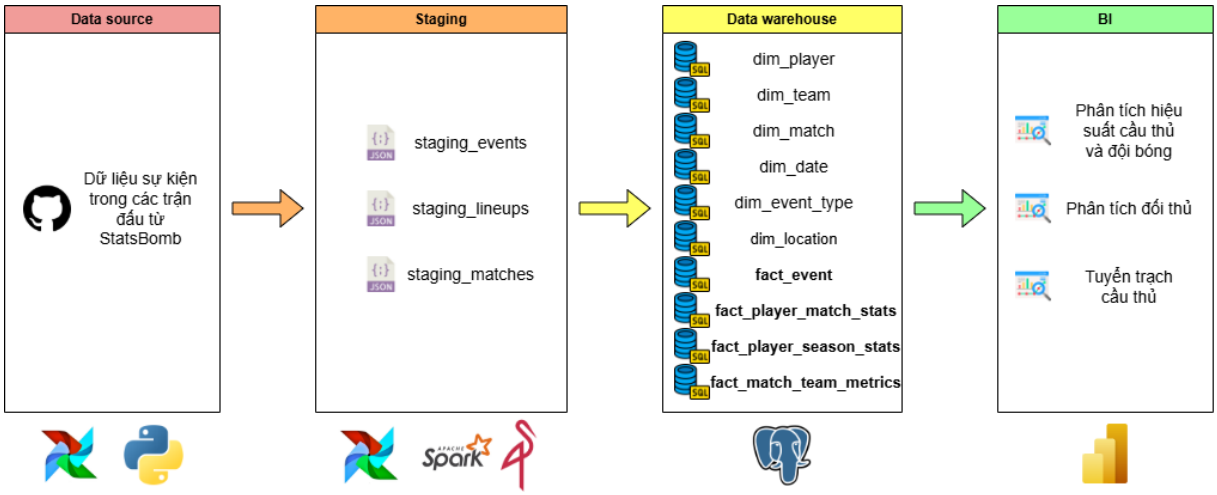
**Kiểm chứng chỉ số nâng cao**

Để đánh giá độ tin cậy của các chỉ số hiện đại, ta có thể phân tích tương quan tuyến tính giữa *Bàn thắng kỳ vọng (xG)* và *Bàn thắng thực tế* (Hình 3.6). Kết quả cho thấy mối tương quan thuận chặt chẽ, các điểm dữ liệu phân bố bám sát đường chéo tham chiếu. Như vậy, $xG$ là một chỉ số dự báo đáng tin cậy cho hiệu suất ghi bàn và được sử dụng làm một trong những chỉ số Fact chính trong Kho dữ liệu.




Hình 3.6: Tương quan giữa Bàn thắng kỳ vọng (xG) và Bàn thắng thực tế

## 3.2 Kiến trúc Data Warehouse




Hình 3.7: Kiến trúc Data Warehouse

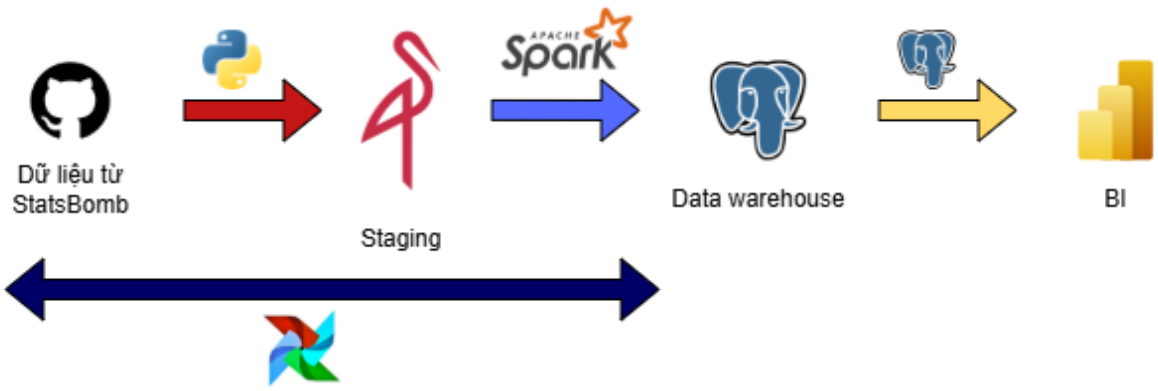
Kiến trúc Data Warehouse được chia làm 4 tầng:

1. **Nguồn dữ liệu (Data source):**

- Dữ liệu từ GitHub của StatsBomb.
- Đây là nguyên liệu thô đầu vào, chứa thông tin đa dạng về trận đấu, cầu thủ và sự kiện kỹ thuật.

2. **Vùng đệm (Staging/Data Lake):**
   - Sử dụng **MinIO** để lưu trữ nguyên bản dữ liệu thô vừa thu thập được.
   - Đảm bảo toàn vẹn dữ liệu và phục vụ truy vết hoặc xử lý lại nếu cần.
3. **Kho dữ liệu (Data Warehouse):**
   - **Xử lý:** Sử dụng **Apache Spark** để đọc dữ liệu từ MinIO, thực hiện làm sạch, chuẩn hóa và làm phẳng cấu trúc JSON lồng nhau.
   - **Lưu trữ:** Dữ liệu sạch được nạp vào **PostgreSQL** và tổ chức theo mô hình lược đồ sao gồm các bảng Fact và bảng Dimension.
4. **Phân tích và báo cáo (BI):**
   - Sử dụng **Microsoft PowerBI** kết nối trực tiếp với PostgreSQL.
   - Cung cấp các dashboard tương tác phục vụ nhu cầu phân tích chiến thuật, hiệu suất cầu thủ và tuyển trạch cho người dùng cuối.

## 3.3 Đường ống dữ liệu




Hình 3.8: Đường ống dữ liệu

Hệ thống sử dụng đường ống dữ liệu tự động hóa được điều phối bởi **Apache Airflow**. Quy trình xử lý dữ liệu được chia thành các giai đoạn tuần tự như sau:

1. **Giai đoạn 1: Trích xuất và tập kết (Extract & Ingest)**

- Các tác vụ Python được kích hoạt để tải các tệp JSON từ StatsBomb.
- Dữ liệu được giữ nguyên định dạng gốc và lưu trữ vào vùng Staging trên **MinIO** (Data Lake). Bước này đảm bảo tách biệt giữa quá trình thu thập và xử lý, giảm thiểu rủi ro ảnh hưởng đến nguồn dữ liệu.

2. **Giai đoạn 2: Chuyển đổi và làm sạch (Transform)**
   - **Apache Spark** đọc dữ liệu thô từ MinIO.
   - Thực hiện làm phẳng các cấu trúc JSON lồng nhau của dữ liệu sự kiện, chuẩn hóa tên cầu thủ và loại bỏ các bản ghi không hợp lệ.
   - Tính toán các chỉ số phát sinh như: Bàn thắng kỳ vọng (xG), tỷ lệ chuyền bóng thành công, số lần gây áp lực,...
3. **Giai đoạn 3: Nạp dữ liệu (Load)**
   - Dữ liệu sau khi xử lý sẽ được ghi vào cơ sở dữ liệu **PostgreSQL**.
   - Tuân theo mô hình *Upsert* (Update/Insert) để đảm bảo không trùng lặp dữ liệu khi chạy lại pipeline.
4. **Giai đoạn 4: Khai thác và phân phối (Serving)**
   - Tại tầng này, dữ liệu được truy vấn trực tiếp bởi **Microsoft PowerBI**.
   - Các kết nối dữ liệu được thiết lập để tự động làm mới các dashboard ngay khi dữ liệu mới được nạp vào kho thành công.

## 3.4 Hệ thống chiều khái niệm

Dựa vào quá trình khảo sát và phân tích, ta có thể phân chia dữ liệu thành các nhóm chiều khái niệm chính:

- **dim_date**: cung cấp trục thời gian cho phân tích.
- **dim_match**: cung cấp thông tin ngữ cảnh cho các trận đấu.
- **dim_team**: cung cấp thông tin cơ bản của các đội bóng.
- **dim_player**: cung cấp thông tin cơ bản của các cầu thủ.
- **dim_event_type, dim_play_pattern**: các loại hành động, tình huống bóng trong trận đấu (chuyền bóng, sút, tình huống cố định,...).

- **dim_location**: mô tả vị trí trên sân.

| date_value | year | month | day | is_weekend |
|---|---|---|---|---|
| 2017-08-20 | 2017 | 1 | 1 | False |
| 2017-08-26 | 2018 | 10 | 10 | True |
| 2017-09-09 | | 11 | 11 | |
| 2017-09-16 | | 12 | 14 | |
| 2017-09-19 | | 2 | 16 | |
| 2017-09-23 | | 3 | 17 | |
| 2017-10-01 | | 4 | 18 | |
| 2017-10-14 | | 5 | 19 | |
| 2017-10-21 | | 8 | 2 | |
| 2017-10-28 | | 9 | 20 | |
| 2017-11-04 | | | 21 | |
| 2017-11-18 | | | 23 | |
| 2017-11-26 | | | 24 | |
| 2017-12-02 | | | 26 | |
| 2017-12-10 | | | 28 | |
| 2017-12-17 | | | 29 | |
| 2017-12-23 | | | 31 | |
| 2018-01-07 | | | 4 | |
| 2018-01-14 | | | 6 | |
| 2018-01-21 | | | 7 | |
| 2018-01-28 | | | 9 | |

Hình 3.9: Nhóm chiều thời gian

| season | competition | kickoff_tim | stadium | referee | roun | season_stage |
|---|---|---|---|---|---|---|
| 2017/2018 | La Liga | 13:00:00.000 | Estádio Cívitas Metropolitano | Alberto Undiano Mallenco | 1 | Regular Season |
| | | 16:15:00.000 | Abanca-Balaídos | Alejandro José Hernández Hernández | 10 | |
| | | 18:15:00.000 | Coliseum Alfonso Pérez | Antonio Miguel Mateu Lahoz | 11 | |
| | | 20:00:00.000 | Estadi Municipal de Montilivi | Carlos del Cerro Grande | 12 | |
| | | 20:15:00.000 | Estadio Abanca-Riazor | Daniel Jesús Trujillo Suárez | 13 | |
| | | 20:45:00.000 | Estadio Benito Villamarín | David Fernández Borbalan | 14 | |
| | | 21:00:00.000 | Estadio Municipal de Butarque | Ignacio Iglesias Villanueva | 15 | |
| | | 22:00:00.000 | Estadio Municipal de Ipurúa | Jesús Gil Manzano | 16 | |
| | | | Estadio Ramón Sánchez Pizjuán | José Luis González González | 17 | |
| | | | Estadio Santiago Bernabéu | José Luis Munuera Montero | 18 | |
| | | | Estadio de Gran Canaria | José María Sánchez Martínez | 19 | |
| | | | Estadio de Mendizorroza | Juan Martínez Munuera | 2 | |
| | | | Estadio de Mestalla | N/A | 20 | |
| | | | Estadio de la Cerámica | Ricardo De Burgos Bengoetxea | 21 | |
| | | | RCDE Stadium | Santiago Jaime Latre | 22 | |
| | | | Reale Arena | | 23 | |
| | | | San Mamés Barria | | 24 | |
| | | | Spotify Camp Nou | | 25 | |

Hình 3.10: Nhóm chiều thông tin trận đấu

| team_name | manager_name | home_stadium |
|---|---|---|
| Athletic Club | Abelardo Fernández Antuña | Estádio Cívitas Metropolitano |
| Atlético Madrid | Asier Garitano Aguirrezábal | Abanca-Balaídos |
| Barcelona | Clarence Seedorf | Coliseum Alfonso Pérez |
| Celta Vigo | Cristóbal Parralo Aguilera | Estadi Municipal de Montilivi |
| Deportivo Alavés | Diego Pablo Simeone | Estadio Abanca-Riazor |
| Eibar | Eder Sarabia Armesto | Estadio Benito Villamarín |
| Espanyol | Enrique Setién Solar | Estadio Municipal de Butarque |
| Getafe | Enrique Sánchez Flores | Estadio Municipal de Ipurúa |
| Girona | Ernesto Valverde Tejedor | Estadio Ramón Sánchez Pizjuán |
| Las Palmas | Eusebio Sacristán Mena | Estadio Santiago Bernabéu |
| Leganés | Francisco Jémez Martín | Estadio de Gran Canaria |
| Levante UD | Francisco Martín Ayestarán Barandiarán | Estadio de Mendizorroza |
| Málaga | Imanol Alguacil Barrenetxea | Estadio de Mestalla |
| RC Deportivo La Coruña | Javier Calleja Revilla | Estadio de la Cerámica |
| Real Betis | José Bordalás Jiménez | N/A |
| Real Madrid | José Luis Mendilibar Etxebarria | RCDE Stadium |
| Real Sociedad | José Miguel González Martín del Campo | Reale Arena |
| Sevilla | José Ángel Ziganda Lacunza | San Mamés Barria |
| Valencia | Juan Carlos Unzué Labiano | Spotify Camp Nou |
| Villarreal | Juan Ramón López Muñiz | |

Hình 3.11: Nhóm chiều thông tin đội bóng

| player_name | nationality | position_group | specific_position |
|---|---|---|---|
| Andrés Iniesta Luján | Argentina | Defender | Center Defensive Midfield |
| Carlos Henrique Casimiro | Belgium | Forward | Center Forward |
| Cristiano Ronaldo dos Santos Aveiro | Brazil | Goalkeeper | Goalkeeper |
| Daniel Ceballos Fernández | Costa Rica | Midfielder | Left Back |
| Denis Suárez Fernández | Croatia | Other | Left Center Back |
| Francisco Alcácer García | France | | Left Center Forward |
| Francisco Casilla Cortés | Germany | | Left Center Midfield |
| Gareth Frank Bale | Netherlands | | Left Defensive Midfield |
| Gerard Piqué Bernabéu | Portugal | | Left Midfield |
| Ivan Rakitić | Spain | | Left Wing |
| Jasper Cillessen | Uruguay | | Right Back |
| Jesús Vallejo Lázaro | Wales | | Right Center Back |
| Jordi Alba Ramos | | | Right Center Forward |
| José Ignacio Fernández Iglesias | | | Right Center Midfield |
| José Paulo Bezzera Maciel Júnior | | | Right Defensive Midfield |
| Karim Benzema | | | Right Midfield |
| Keylor Navas Gamboa | | | Right Wing |
| Lionel Andrés Messi Cuccittini | | | Substitute/Unknown |

Hình 3.12: Nhóm chiều thông tin cầu thủ

| event_type | event_category | outcome |
|---|---|---|
| Bad Behaviour | Attack | Blocked |
| Ball Receipt* | Contest | Complete |
| Ball Recovery | Defense | Goal |
| Block | Discipline | Incomplete |
| Carry | Distribution | Off T |
| Clearance | General Play | Out |
| Dispossessed | Goalkeeping | Pass Offside |
| Dribble | | Saved |
| Dribbled Past | | Success |
| Duel | | Unknown |
| Foul Committed | | Wayward |
| Foul Won | | |
| Goal Keeper | | |
| Half End | | |
| Half Start | | |
| Injury Stoppage | | |
| Interception | | |
| Miscontrol | | |
| Offside | | |

Hình 3.13: Nhóm chiều thông tin sự kiện

| play_pattern_name |
|---|
| From Corner |
| From Counter |
| From Free Kick |
| From Goal Kick |
| From Keeper |
| From Kick Off |
| From Throw In |
| Regular Play |

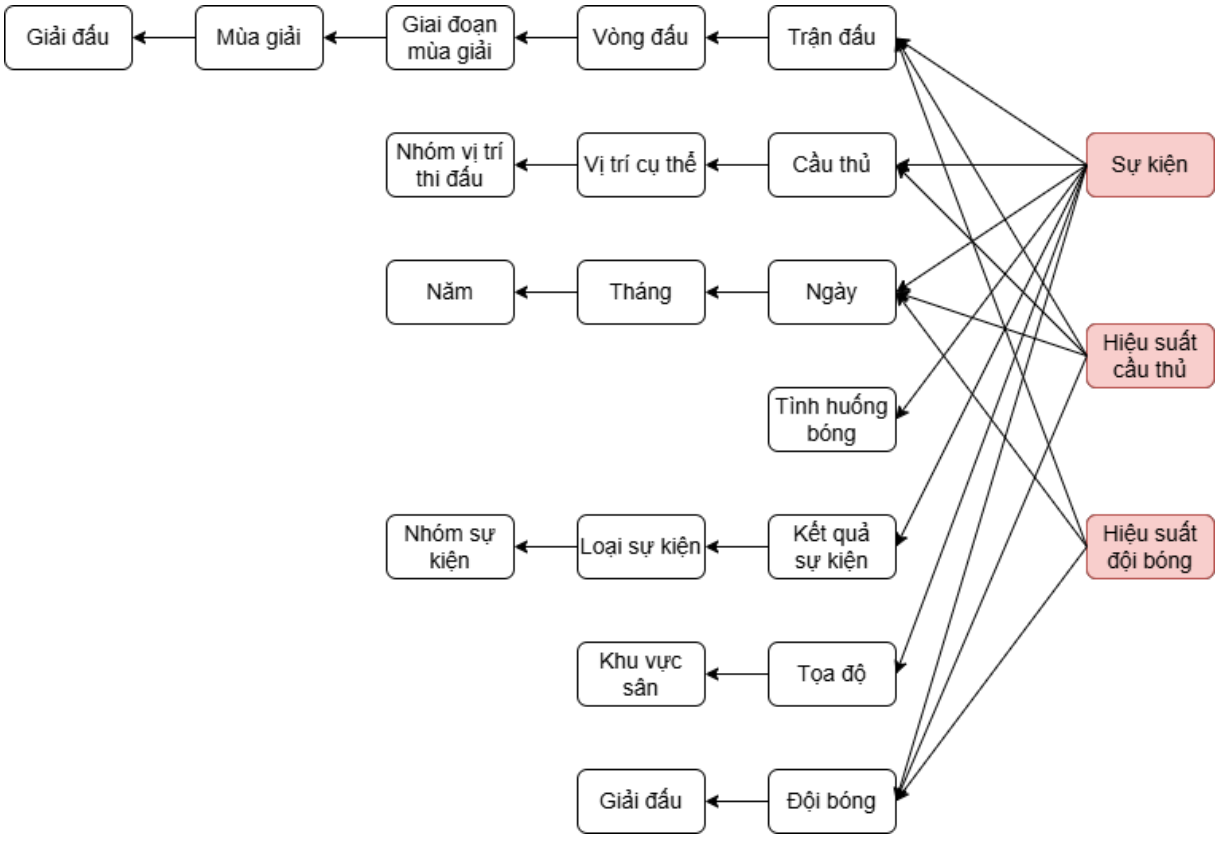
Hình 3.14: Nhóm chiều tình huống bóng

| field_zone | is_box |
|---|---|
| Attacking Third | False |
| Defensive Third | True |
| Midfield | |

Hình 3.15: Nhóm chiều khu vực sân

## 3.5 Mô hình dữ liệu logic

Căn cứ vào quá trình khảo sát và phân tích, mô hình dữ liệu logic cho hệ thống kho dữ liệu có thể được minh họa cụ thể như sau:




Hình 3.16: Mô hình dữ liệu logic

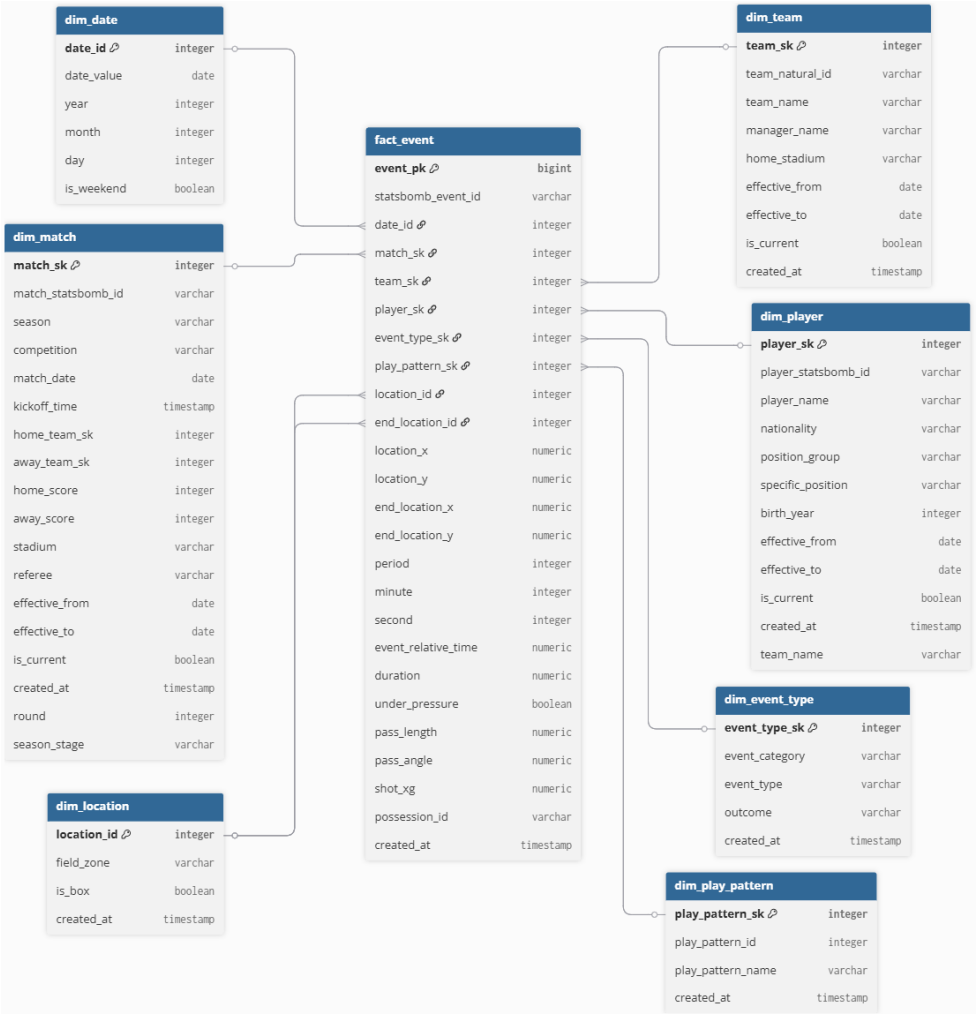
## 3.6 Mô hình dữ liệu vật lý

Từ hệ thống chiều khái niệm và mô hình dữ liệu logic đã xây dựng, mô hình dữ liệu vật lý được thiết kế nhằm hiện thực hóa cấu trúc dữ liệu trên hệ quản trị cơ sở dữ liệu PostgreSQL. Các bảng dữ liệu được xác định chi tiết về kiểu dữ liệu, khóa

chính, khóa ngoại và các ràng buộc toàn vẹn, đảm bảo dữ liệu được lưu trữ nhất quán và hiệu quả. Đây là cơ sở để triển khai các quy trình ETL, xây dựng Data Mart và kết nối với các công cụ BI trong các bước tiếp theo.

| **Tên cột** | **Kiểu dữ liệu** | **Ý nghĩa & Diễn giải** |
|---|---|---|
| **Bảng: dim_player** | | |
| player_sk | Integer | Khóa thay thế (Surrogate Key) |
| player_name | Varchar | Tên cầu thủ |
| effective_from | Date | Ngày bắt đầu hiệu lực |
| effective_to | Date | Ngày kết thúc (NULL nếu hiện tại) |
| is_current | Boolean | Đánh dấu dòng dữ liệu mới nhất |
| **Bảng: dim_team** | | |
| team_sk | Integer | Khóa thay thế của đội bóng |
| team_name | Varchar | Tên đội bóng |
| manager_name | Varchar | Tên HLV trưởng tại thời điểm đó |
| home_stadium | Varchar | Sân vận động nhà |
| **Bảng: dim_match** | | |
| match_sk | Integer | Khóa thay thế trận đấu |
| match_date | Date | Ngày thi đấu |
| season | Varchar | Mùa giải (VD: 2020/2021) |
| competition | Varchar | Tên giải đấu |
| **Bảng: dim_location** | | |
| location_id | Integer | Khóa chính |
| field_zone | Varchar | Tên khu vực (VD: Zone 14, Cánh trái) |
| is_box | Boolean | Có nằm trong vòng cấm không |

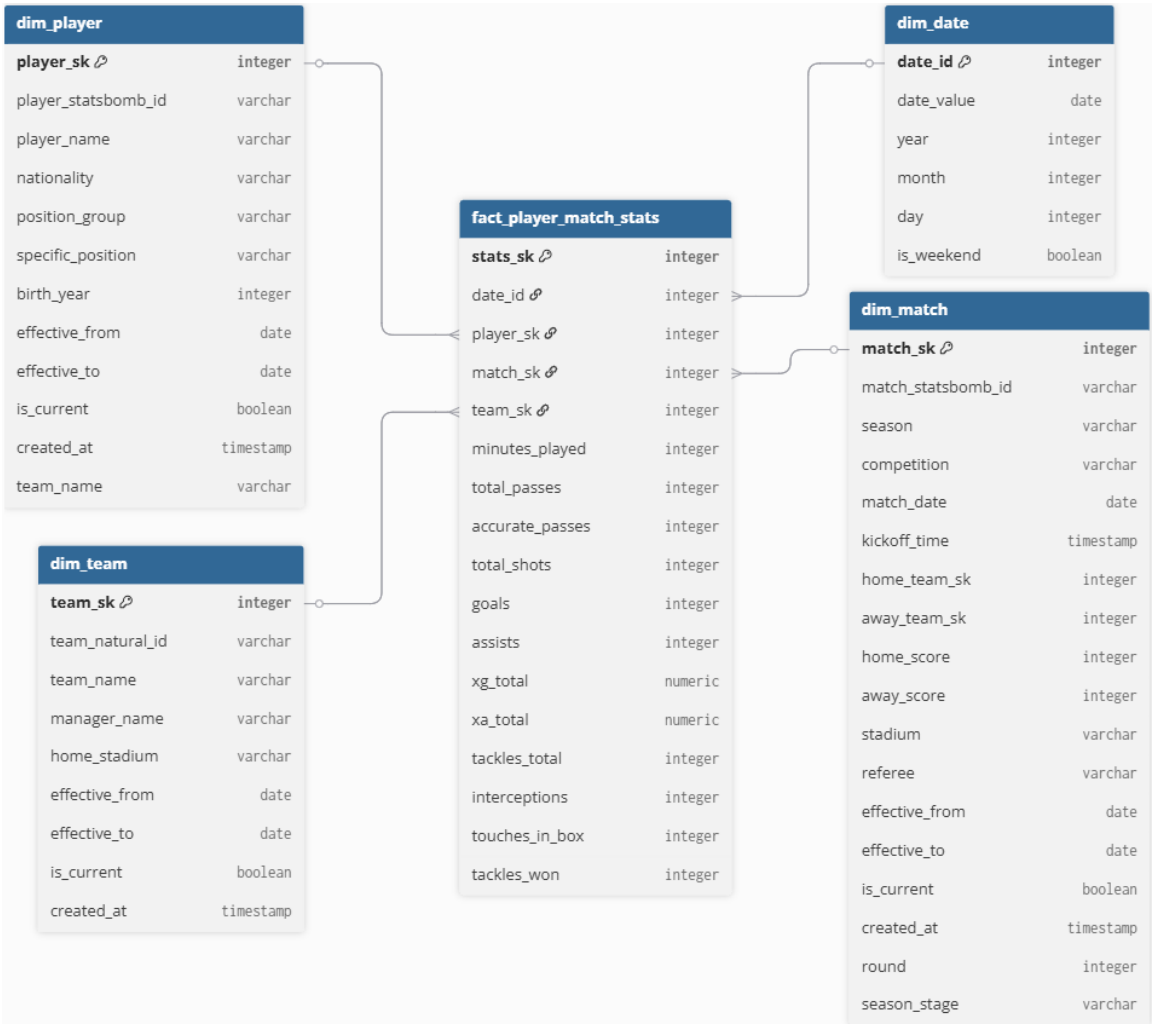
Bảng 3.4: Đặc tả các bảng Dimension dùng chung




Hình 3.17: Mô hình dữ liệu vật lý của bảng fact_event

| **Trường** | **Kiểu dữ liệu** | **Ý nghĩa** |
|---|---|---|
| event_pk | Bigint | Khóa chính sự kiện |
| match_sk | Integer | FK: Trận đấu |
| player_sk | Integer | FK: Cầu thủ thực hiện |
| team_sk | Integer | FK: Đội bóng thực hiện |
| location_x | Numeric | Tọa độ X điểm bắt đầu (0-120) |
| location_y | Numeric | Tọa độ Y điểm bắt đầu (0-80) |
| end_location_x | Numeric | Tọa độ X điểm đến (0-120) |
| end_location_y | Numeric | Tọa độ Y điểm đến (0-90) |
| location_id | Integer | FK: Khu vực trên sân |
| event_type_sk | Integer | FK: Loại sự kiện |
| minute | Integer | Thời điểm diễn ra (Phút) |
| under_pressure | Boolean | Có bị áp lực khi xử lý không |

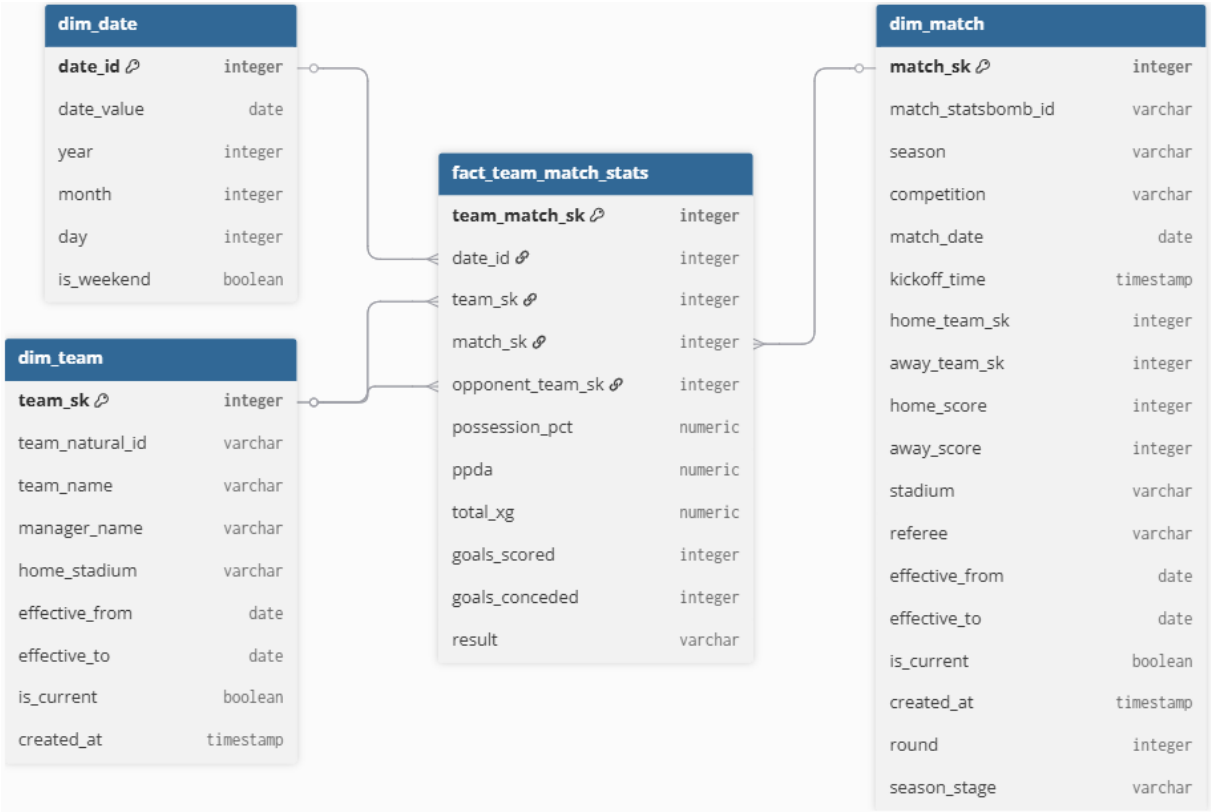
Bảng 3.5: Cấu trúc bảng fact_event (Sự kiện chi tiết)




Hình 3.18: Mô hình dữ liệu vật lý của bảng fact_player_match_stats

| **Trường** | **Kiểu dữ liệu** | **Ý nghĩa** |
|---|---|---|
| stats_sk | Integer | Khóa chính |
| player_sk | Integer | FK: Cầu thủ |
| match_sk | Integer | FK: Trận đấu |
| minutes_played | Integer | Số phút thi đấu thực tế |
| total_passes | Integer | Tổng số đường chuyền |
| accurate_passes | Integer | Số đường chuyền chính xác |
| xg_total | Numeric | Tổng xG (Bàn thắng kỳ vọng) |
| xa_total | Numeric | Tổng xA (Kiến tạo kỳ vọng) |
| goals | Integer | Số bàn thắng ghi được |
| assists | Integer | Số kiến tạo thành bàn |
| tackles | Integer | Số lần tắc bóng thành công |

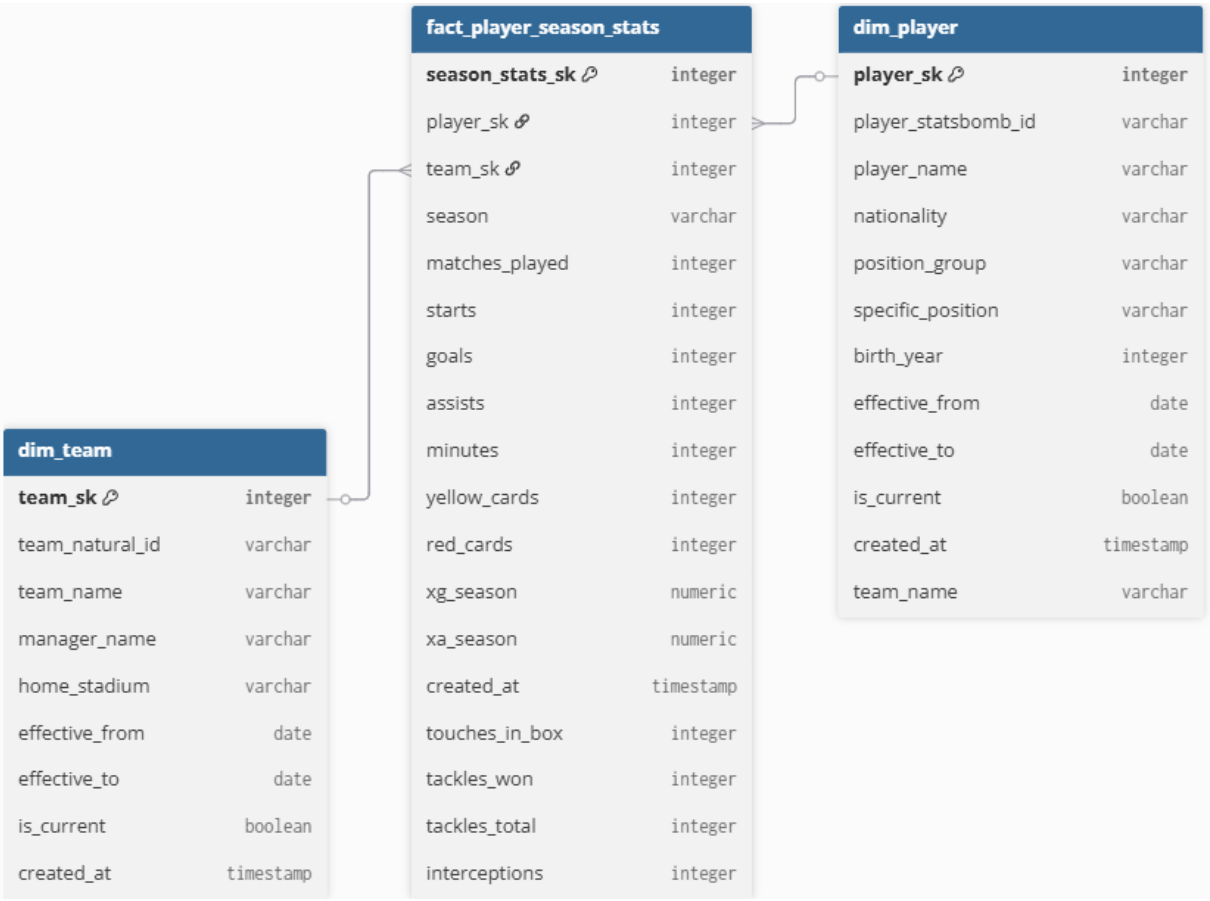
Bảng 3.6: Cấu trúc bảng fact_player_match_stats




Hình 3.19: Mô hình dữ liệu vật lý của bảng fact_team_match_stats

| **Trường** | **Kiểu dữ liệu** | **Ý nghĩa** |
|---|---|---|
| team_match_sk | Integer | Khóa chính |
| team_sk | Integer | FK: Đội bóng chủ thể |
| opponent_team_sk | Integer | FK: Đội đối thủ |
| match_sk | Integer | FK: Trận đấu |
| possession_pct | Numeric | Tỉ lệ kiểm soát bóng (%) |
| ppda | Numeric | Chỉ số PPDA (Đo cường độ Pressing) |
| total_xg | Numeric | Tổng xG của cả đội |
| goals_scored | Integer | Bàn thắng ghi được |
| goals_conceded | Integer | Bàn thua phải nhận |
| result | Varchar | Kết quả (Thắng/Hòa/Thua) |

Bảng 3.7: Cấu trúc bảng fact_team_match_stats




Hình 3.20: Mô hình dữ liệu vật lý của bảng fact_player_season_stats

| **Trường** | **Kiểu dữ liệu** | **Ý nghĩa** |
|---|---|---|
| season_stats_sk | Integer | Khóa chính |
| player_sk | Integer | FK: Cầu thủ |
| team_sk | Integer | FK: Đội bóng |
| season | Varchar | Mùa giải |
| matches_played | Integer | Số trận ra sân |
| starts | Integer | Số trận đá chính |
| goals | Integer | Tổng bàn thắng cả mùa |
| assists | Integer | Tổng kiến tạo cả mùa |
| yellow_cards | Integer | Tổng số thẻ vàng |
| xg_season | Numeric | Tổng xG tích lũy |
| xa_season | Numeric | Tổng xA tích lũy |

Bảng 3.8: Cấu trúc bảng fact_player_season_stats

# Chương 4

# Cài đặt hệ thống

## 4.1 Quá trình xử lý dữ liệu

### 4.1.1 Cấu hình môi trường và kết nối dữ liệu

Để đảm bảo tính nhất quán và khả năng mở rộng, hệ thống được xây dựng trên nền tảng Apache Spark, kết nối với Data Lake (MinIO) và Data Warehouse (PostgreSQL) thông qua các giao thức chuẩn.

Sử dụng thư viện `hadoop-aws` để Spark có thể giao tiếp trực tiếp với MinIO thông qua giao thức S3 (`s3a://`). Cấu hình `fs.s3a.path.style.access` được đặt là `true` để đảm bảo tương thích với kiến trúc MinIO chạy trên Docker nội bộ.

Việc ghi dữ liệu vào PostgreSQL được thực hiện thông qua JDBC Driver (`org.postgresql.Driver`). Các cấu hình kết nối được tham số hóa để đảm bảo bảo mật và dễ dàng thay đổi môi trường.

```
spark = SparkSession.builder \
    .appName("ETL_Dim_Date_Fixed") \
    .config("spark.hadoop.fs.s3a.endpoint", MINIO_CONF["endpoint"]) \
    .config("spark.hadoop.fs.s3a.access.key", MINIO_CONF["access_key"]) \
    .config("spark.hadoop.fs.s3a.secret.key", MINIO_CONF["secret_key"]) \
    .config("spark.hadoop.fs.s3a.path.style.access", "true") \
    .config("spark.hadoop.fs.s3a.impl", "org.apache.hadoop.fs.s3a.S3AFileSystem") \
    .config("spark.hadoop.fs.s3a.connection.ssl.enabled", "false") \
    .getOrCreate()
```

Hình 4.1: Cấu hình kết nối Apache Spark với MinIO

### 4.1.2 Xử lý dữ liệu cho các bảng Dim

Một trong những thách thức lớn nhất của dữ liệu bóng đá là tính biến động theo thời gian (ví dụ: cầu thủ chuyển đội, đội bóng thay huấn luyện viên). Thuật

toán SCD Type 2 (Slowly Changing Dimension) được cài đặt bằng PySpark có thể giúp giải quyết vấn đề này. Thuật toán giúp phát hiện và xử lý thay đổi:

- **Phân hoạch dữ liệu:** Dữ liệu nguồn được gom nhóm theo khóa (ví dụ: `player_id` hoặc `team_id`) và sắp xếp tăng dần theo thời gian.

```
window_spec = Window.partitionBy("team_natural_id").orderBy("match_date")
```

- **Phát hiện thay đổi:** Sử dụng Window Function `lag()` để so sánh giá trị của bản ghi hiện tại với bản ghi liền trước. Một bản ghi mới được xác định khi:
  - Là bản ghi đầu tiên trong lịch sử.
  - Có sự thay đổi ở các trường quan trọng (ví dụ: `team_name`, `manager_name`).

```
df_scd = df_full.withColumn("prev_manager", lag("manager_name").over(window_spec)) \
                .withColumn("prev_name", lag("team_name").over(window_spec)) \
df_changes = df_scd.filter(
    (col("prev_manager").isNull()) |
    (col("manager_name") != col("prev_manager")) |
    (col("team_name") != col("prev_name"))
)
```

- **Tính toán thời gian hiệu lực:**
  - `effective_from`: Là ngày diễn ra của trận đấu đầu tiên (`match_date`) xuất hiện sự thay đổi.
  - `effective_to`: Sử dụng hàm `lead()` để lấy ngày bắt đầu của bản ghi kế tiếp trừ đi 1 ngày. Nếu không có bản ghi kế tiếp (dữ liệu là bản ghi mới nhất), giá trị được gán mặc định là "9999-12-31".
  - **Đánh dấu hiện hành:** Cột `is_current` là `true` nếu `effective_to` là "9999-12-31".

```
window_next = Window.partitionBy("team_natural_id").orderBy("match_date")

df_final = df_changes.withColumn("effective_from", col("match_date")) \
    .withColumn("next_start_date", lead("match_date").over(window_next)) \
    .withColumn("effective_to",
        when(col("next_start_date").isNotNull(), date_sub(col("next_start_date"), 1))
        .otherwise(to_date(lit("9999-12-31")))
    ) \
    .withColumn("is_current",
        when(col("effective_to") == to_date(lit("9999-12-31")), True).otherwise(False)
    ) \
```

**Kết quả:** Bảng dim_player và dim_team lưu trữ lịch sử chuyển nhượng và thay đổi nhân sự, cho phép truy vấn chính xác trạng thái của đối tượng tại bất kỳ thời điểm nào trong quá khứ.

### 4.1.3 Xử lý dữ liệu cho các bảng Fact

#### Chuẩn hóa dữ liệu sự kiện

Tự động quét schema của DataFrame để tìm tất cả các cấu trúc chứa trường `outcome`, giúp hợp nhất cấu trúc với các trường lồng nhau phức tạp trong file JSON gốc thành trường `outcome_name` duy nhất.

```
outcome_columns = []
for field in df_raw.schema.fields:
    if isinstance(field.dataType, StructType) and 'outcome' in field.dataType.names:
        outcome_columns.append(col(f"{field.name}.outcome.name"))
final_outcome_col = coalesce(*outcome_columns) if outcome_columns else lit(None)
```

Chuẩn hóa tọa độ (x, y) thành các ID từ 1 đến 18 và khu vực đặc biệt (Penalty Box). Logic này sử dụng chuỗi điều kiện when-otherwise lồng nhau, giúp tối ưu tốc độ truy vấn phân tích không gian sau này.

```
def calculate_zone_id(x_col, y_col):
    return when((col(x_col) >= 102) & (col(y_col) >= 18) & (col(y_col) <= 62), 19) \
            .when((col(x_col) < 20), \
                when(col(y_col) < 26.6, 1).when(col(y_col) < 53.3, 2).otherwise(3)) \
            .when((col(x_col) < 40), \
                when(col(y_col) < 26.6, 4).when(col(y_col) < 53.3, 5).otherwise(6)) \
            .when((col(x_col) < 60), \
                when(col(y_col) < 26.6, 7).when(col(y_col) < 53.3, 8).otherwise(9)) \
            .when((col(x_col) < 80), \
                when(col(y_col) < 26.6, 10).when(col(y_col) < 53.3, 11).otherwise(12)) \
            .when((col(x_col) < 100), \
                when(col(y_col) < 26.6, 13).when(col(y_col) < 53.3, 14).otherwise(15)) \
            .otherwise( \
                when(col(y_col) < 26.6, 16).when(col(y_col) < 53.3, 17).otherwise(18))
```

Thời gian xảy ra sự kiện được chuyển đổi từ dạng "HH:mm:ss.SSS" sang dạng số thực (giây) để phục vụ các tính toán khoảng cách thời gian giữa các sự kiện.

```
df_ready = df_final_join \
    .withColumn("date_id", date_format(col("match_date"), "yyyyMMdd").cast(IntegerType())) \
    .withColumn("t_parts", split(col("event_relative_time_str"), ":")) \
    .withColumn("calc_seconds",
        col("t_parts")[0].cast("float") * 3600 +
        col("t_parts")[1].cast("float") * 60 +
        col("t_parts")[2].cast("float")
    ) \
    .withColumn("event_relative_time",
        when(col("calc_seconds") > 18000, lit(None)).otherwise(col("calc_seconds"))
    )
```

**Sử dụng Broadcast Join để tối ưu hiệu năng**

Khi thực hiện Lookup dữ liệu từ các bảng Dimension có kích thước nhỏ (như dim_event_type, dim_play_pattern) vào bảng Fact khổng lồ (fact_event), hệ thống sử dụng kỹ thuật Broadcast Join.

- **Cơ chế:** Spark sẽ gửi bản sao của bảng Dimension đến tất cả các node worker thay vì thực hiện Sort-Merge Join (yêu cầu shuffle cả bảng Fact lớn).
- **Cài đặt:** Sử dụng hàm `broadcast()` bao quanh các DataFrame bảng Dimension trong câu lệnh join.

```
cond_type = (
    (df_j_player.event_type_name == dim_event_type.event_type) &
    (df_j_player.outcome_name.eqNullSafe(dim_event_type.outcome))
)
df_j_type = df_j_player.join(broadcast(dim_event_type), cond_type, "left") \
    .select(df_j_player["*"], dim_event_type["event_type_sk"])
```

- **Hiệu quả:** Giảm lưu lượng mạng và loại bỏ hiện tượng phân bổ dữ liệu không đồng đều trên các phân vùng khi join.

**Chuẩn hóa dữ liệu thống kê tổng hợp**

Thuật toán tính số phút thi đấu thực tế:

- **Xác định thời điểm vào sân:** 0 phút cho cầu thủ đá chính, hoặc phút thay người cho cầu thủ dự bị.

```
# A. Tìm thời lượng trận đấu
df_match_duration = df_events.groupBy("match_id") \
    .agg(max("minute").alias("match_end_min"))

# B. Xác định thời điểm VÀO SÂN
df_starters = df_events.filter(col("type.name") == "Starting XI") \
    .select("match_id", explode("tactics.lineup").alias("l")) \
    .select(
        col("match_id"),
        col("l.player.id").cast(StringType()).alias("player_statsbomb_id"),
        lit(0).alias("entry_min")
    )

df_subs_in = df_events.filter(col("type.name") == "Substitution") \
    .select(
        col("match_id"),
        col("substitution.replacement.id").cast(StringType()).alias("player_statsbomb_id"),
        col("minute").alias("entry_min")
    )

df_entries = df_starters.unionByName(df_subs_in)
```

- **Xác định thời điểm rời sân:** Phút thay người (nếu bị thay ra) hoặc phút bị thẻ đỏ.

```
# C. Xác định thời điểm RỜI SÂN
df_subs_out = df_events.filter(col("type.name") == "Substitution") \
    .select(
        col("match_id"),
        col("player.id").cast(StringType()).alias("player_statsbomb_id"),
        col("minute").alias("exit_min")
    )

df_red_cards = df_events.filter(
    col("bad_behaviour.card.name").isin("Red Card", "Second Yellow") |
    col("foul_committed.card.name").isin("Red Card", "Second Yellow")
).select(
    col("match_id"),
    col("player.id").cast(StringType()).alias("player_statsbomb_id"),
    col("minute").alias("exit_min")
)

df_exits = df_subs_out.unionByName(df_red_cards)
```

- **Công thức:** Số phút = Thời điểm rời sân/hết trận − Thời điểm vào sân.

Tính toán các chỉ số nâng cao:

- **xG/xA:** Tổng hợp từ dữ liệu sự kiện chi tiết có sẵn trong nguồn dữ liệu gốc.
- **Touches in Box:** Đếm số lần chạm bóng có tọa độ nằm trong vòng cấm địa đối phương.
- **TSR:** Tính toán dựa trên kết quả của các sự kiện tranh chấp (Duel).

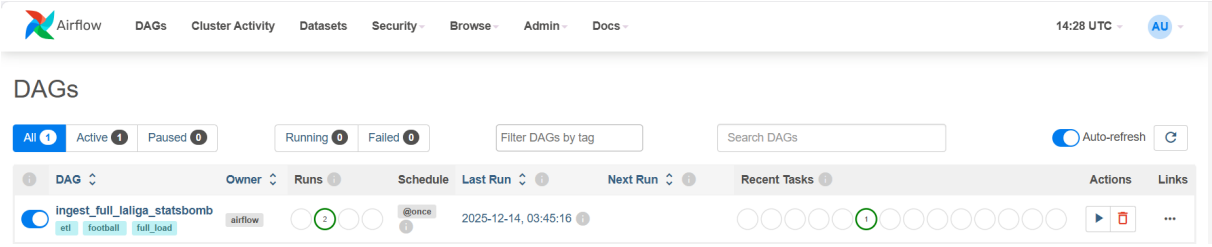
```
# TiB
sum(when(is_in_box, 1).otherwise(0)).alias("touches_in_box"),
# TSR (Total)
sum(when((col("type_name") == "Duel") & (col("duel_type") == "Tackle"), 1).otherwise(0)).alias("tackles_total"),
# TSR (Won)
sum(when(
    (col("type_name") == "Duel") &
    (col("duel_type") == "Tackle") &
    (col("duel_outcome").isin(tackle_won_outcomes)), 1
).otherwise(0)).alias("tackles_won")
```

- **PPDA:** Sử dụng Window Functions để tính toán số đường chuyền của đối thủ trực tiếp trên dòng dữ liệu mà không cần Self-Join gây tốn kém tài nguyên.

```
w_match = Window.partitionBy("match_id")

df_calc = df_agg_basic \
    .withColumn("match_total_duration", sum("my_duration").over(w_match)) \
    .withColumn("match_total_passes", sum("my_pass_count").over(w_match)) \
    .withColumn("opponent_pass_count", col("match_total_passes") - col("my_pass_count"))
df_agg_basic = df_metrics.groupBy("match_id", "team_natural_id") \
    .agg(
        # Tổng xG
        sum(coalesce(col("xg"), lit(0))).cast("numeric(6,3)").alias("total_xg"),

        # Thời gian cầm bóng (để tính %) - Clean overflow duration > 999
        sum(when(abs(col("duration")) > 999, 0).otherwise(coalesce(col("duration"), lit(0)))).alias("my_duration"),

        # Số đường chuyền của MÌNH (để tính PPDA cho đối thủ)
        sum(when(col("type_name") == "Pass", 1).otherwise(0)).alias("my_pass_count"),

        # Số hành động phòng ngự của MÌNH (để tính PPDA cho mình)
        sum(when(col("type_name").isin(def_actions), 1).otherwise(0)).alias("my_def_action_count")
    )
# Possession % = My Duration / Total Duration * 100
when(col("match_total_duration") > 0,
     (col("my_duration") / col("match_total_duration") * 100))
.otherwise(50).cast("numeric(5,2)").alias("possession_pct"),

# PPDA = Opponent Passes / My Def Actions
when(col("my_def_action_count") > 0,
     col("opponent_pass_count") / col("my_def_action_count"))
.otherwise(None).cast("numeric(6,2)").alias("ppda")
```

## 4.2 Tự động hóa quy trình xử lý với Apache Airflow

Để quản lý sự phụ thuộc phức tạp giữa các job PySpark và đảm bảo quy trình ETL vận hành ổn định định kỳ, hệ thống sử dụng **Apache Airflow** làm công cụ điều phối (Orchestration). Toàn bộ quy trình được định nghĩa dưới dạng một Đồ thị không chu trình có hướng (DAG - Directed Acyclic Graph).

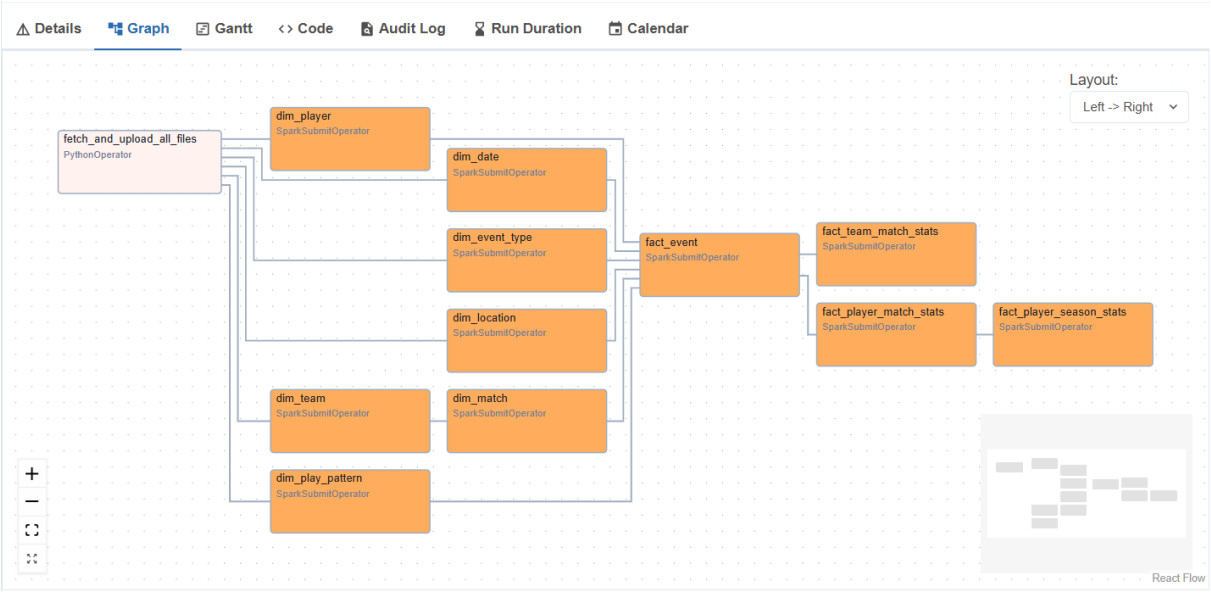



Hình 4.2: Giao diện của Apache Airflow

### 4.2.1 Thiết kế luồng dữ liệu

Quy trình xử lý dữ liệu bóng đá cần phải tuân thủ nghiêm ngặt thứ tự ưu tiên để đảm bảo tính toàn vẹn. DAG được chia thành 4 giai đoạn xử lý tuần tự:

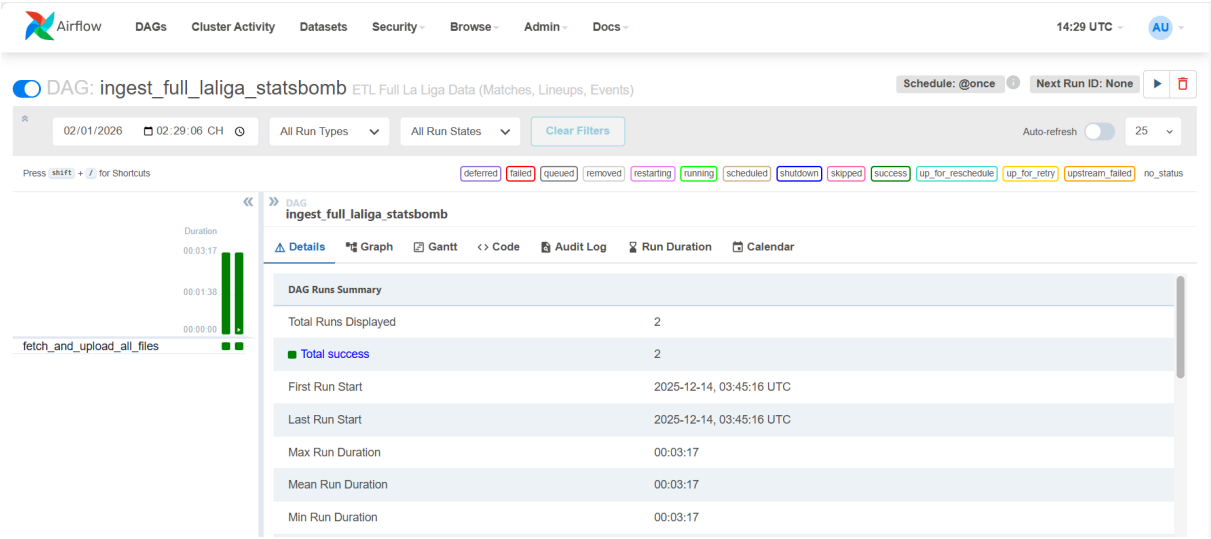
1. **Giai đoạn 1: Thu thập dữ liệu**
    - Task: `ingest_statsbomb_data`
    - Sử dụng `PythonOperator` để tải dữ liệu JSON mới nhất từ GitHub và đẩy vào MinIO (khu vực Staging).
2. **Giai đoạn 2: Xử lý Dimensions (Chạy song song)**
    - Các bảng dim_date, dim_location, dim_event_type, dim_play_pattern được xử lý song song vì chúng độc lập với nhau.
    - dim_player và dim_team cũng được kích hoạt trong giai đoạn này để sẵn sàng cho các bảng Fact.
    - Sử dụng `SparkSubmitOperator` để submit các job PySpark lên cluster.
3. **Giai đoạn 3: Xử lý Fact chi tiết**
    - Task: `fact_event`
    - Task này chỉ được phép chạy khi tất cả các task ở Giai đoạn 2 đã chạy thành công. Điều này đảm bảo khi bảng fact_event thực hiện Lookup ID, các khóa ngoại đã tồn tại trong bảng Dimension.
4. **Giai đoạn 4: Tổng hợp dữ liệu**
    - Task: `fact_player_match_stats` và `fact_team_match_stats` chạy song song, lấy dữ liệu nguồn từ `fact_event` vừa tạo.
    - Task: `fact_player_season_stats` chạy cuối cùng, tổng hợp dữ liệu từ bảng stats theo trận đấu.




Hình 4.3: Luồng thực hiện các task trên Apache Airflow

### 4.2.2 Cấu hình kỹ thuật và Giám sát

**SparkSubmitOperator:** Mỗi bước biến đổi dữ liệu tương ứng với một file mã nguồn PySpark độc lập. Airflow kích hoạt, gửi lệnh `spark-submit` tới Spark Master container kèm theo các cấu hình tài nguyên (Driver Memory, Executor Memory) phù hợp với độ nặng của từng task.

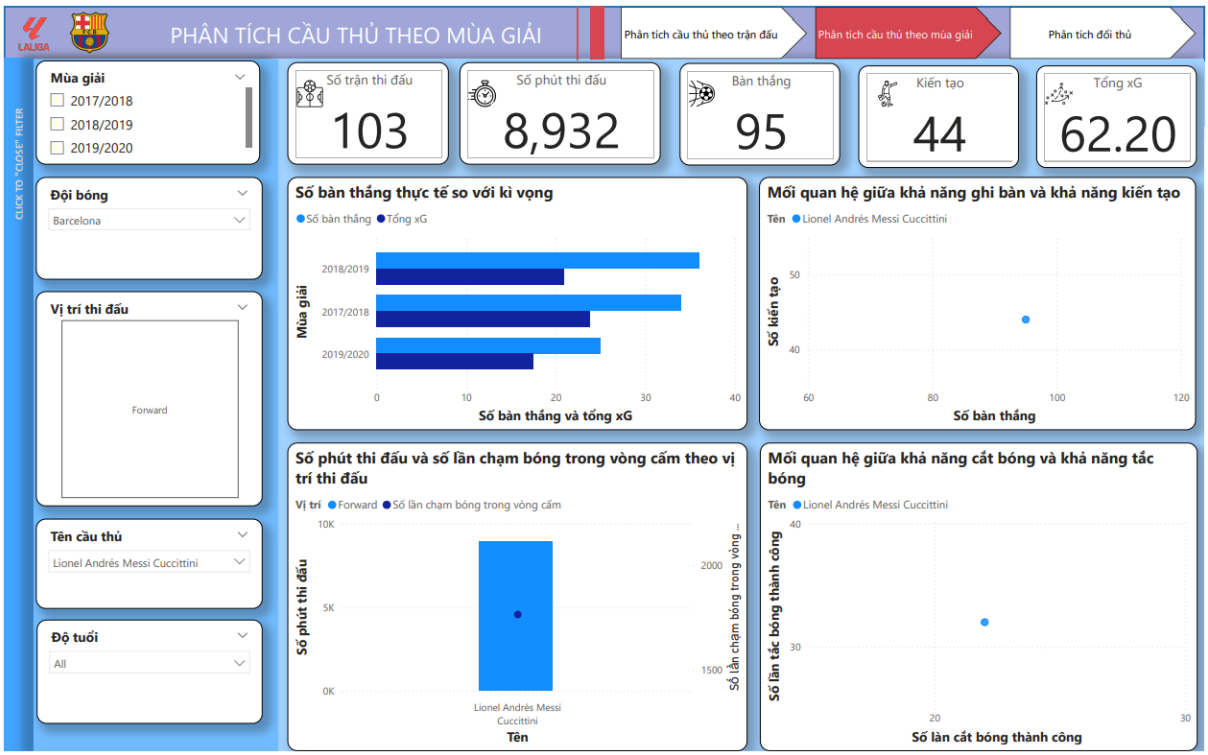



Hình 4.4: Kết quả thực thi của các task trên Apache Airflow

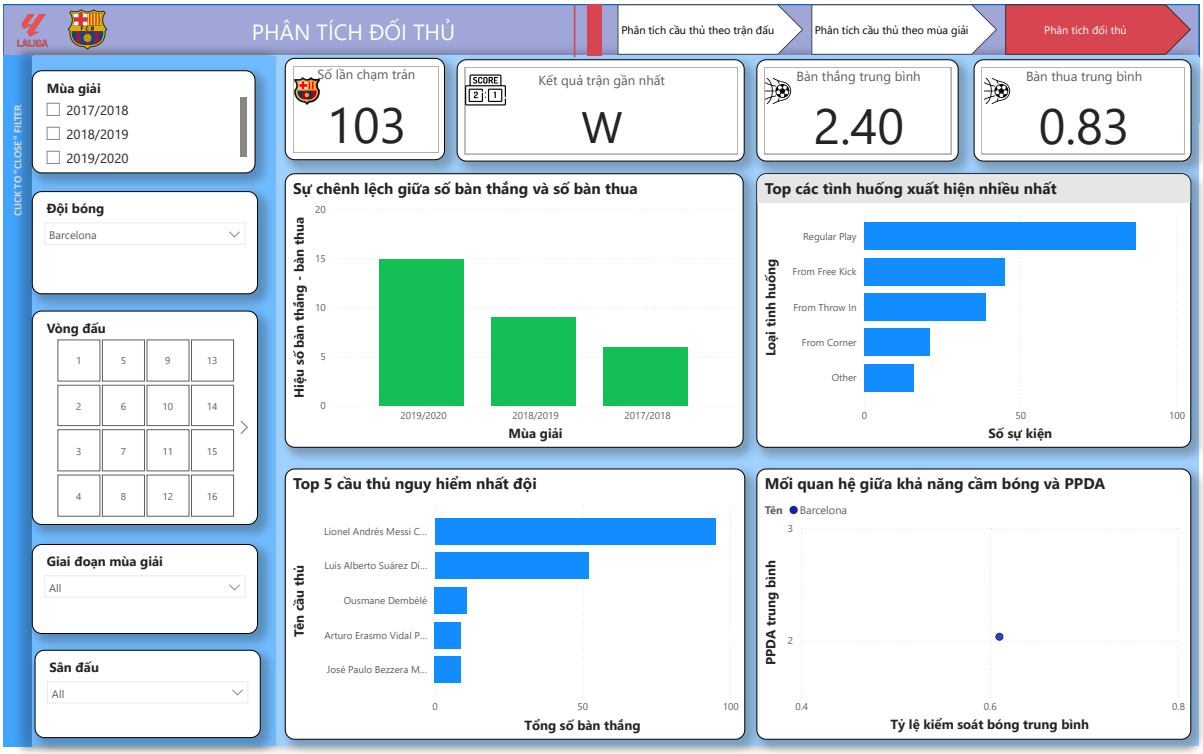
## 4.3 Xây dựng báo cáo phân tích


Hình 4.5: Dashboard phân tích trận đấu


Hình 4.6: Dashboard phân tích cầu thủ theo trận đấu




Hình 4.7: Dashboard phân tích cầu thủ theo mùa giải




Hình 4.8: Dashboard phân tích đội bóng đối thủ

# Kết luận

## Các kết quả đạt được

Đồ án đã giải quyết được các bài toán kỹ thuật và nghiệp vụ cốt lõi sau:

- **Xây dựng quy trình xử lý dữ liệu hiện đại:** Triển khai thành công đường ống dữ liệu tự động hóa sử dụng **Apache Airflow**, tự động trích xuất dữ liệu thô, lưu trữ tại Data Lake (**MinIO**) và chuyển đổi dữ liệu đa tầng.
- **Xử lý dữ liệu bán cấu trúc:** Sử dụng **Apache Spark** để làm phẳng và chuẩn hóa dữ liệu từ định dạng JSON lồng nhau của StatsBomb. Dữ liệu sau xử lý đảm bảo tính toàn vẹn và sẵn sàng cho các truy vấn phân tích.
- **Thiết kế Kho dữ liệu tối ưu:** Xây dựng thành công mô hình dữ liệu dạng lược đồ sao trên hệ quản trị **PostgreSQL**. Các bảng được thiết kế tối ưu cho các chỉ số như bàn thắng kỳ vọng (xG), kiến tạo kỳ vọng (xA) và PPDA.
- **Trực quan hóa và hỗ trợ ra quyết định:** Hệ thống báo cáo trên **Microsoft PowerBI** cung cấp các Dashboard trực quan.

## Hạn chế

Bên cạnh những kết quả đạt được, đồ án vẫn còn tồn tại một số hạn chế:

- **Độ trễ dữ liệu:** Hệ thống hiện tại hoạt động theo cơ chế xử lý theo lô. Dữ liệu chỉ được cập nhật sau khi trận đấu kết thúc, chưa hỗ trợ phân tích thời gian thực ngay trong khi trận đấu đang diễn ra.
- **Phạm vi dữ liệu:** Do giới hạn của nguồn dữ liệu mở StatsBomb, đồ án mới chỉ tập trung phân tích sâu cho một số giải đấu và câu lạc bộ cụ thể (như Barcelona, La Liga).

## Hướng phát triển trong tương lai

Để nâng cao tính ứng dụng và hoàn thiện hệ thống, các hướng phát triển tiếp theo được đề xuất bao gồm:

- **Tích hợp xử lý thời gian thực:** Sử dụng công nghệ **Apache Kafka** hoặc **Spark Streaming** vào kiến trúc hệ thống để thu thập và xử lý sự kiện ngay lập tức, phục vụ cho các quyết định chiến thuật trực tiếp trong trận đấu.
- **Mở rộng và tối ưu hóa hạ tầng:** Triển khai hệ thống lên các nền tảng đám mây để tận dụng khả năng mở rộng linh hoạt, đồng thời tối ưu hóa chi phí lưu trữ và hiệu năng tính toán khi khối lượng dữ liệu lịch sử tăng lên.

# Tài liệu tham khảo

[1] Wikipedia. *Bóng đá*. `https://vi.wikipedia.org/wiki/Football`. 2025.

[2] TS. Phạm Huyền Linh. *Bài giảng Phân tích và thiết kế hệ thống*. 2025.

[3] TS. Lê Hải Hà. *Bài giảng Phân tích và thiết kế hệ thống*. 2024.

[4] ThS. Nguyễn Danh Tú. *Giáo trình Kho dữ liệu và Kinh doanh thông minh*. 2025.