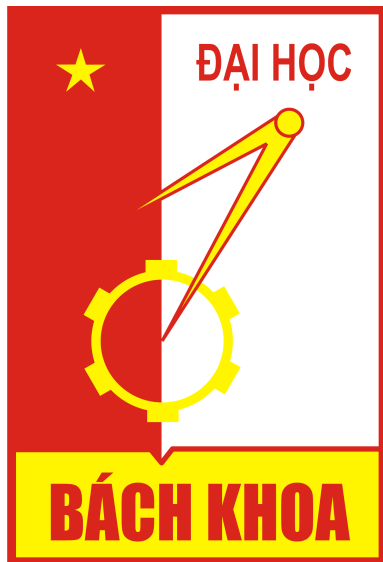


**ĐẠI HỌC BÁCH KHOA HÀ NỘI**  
**TRƯỜNG CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**



**BÁO CÁO KẾT QUẢ**  
**BÀI TẬP LỚN**

**Thiết kế cơ sở dữ liệu quản lý giải đấu bóng đá**

**Giảng viên hướng dẫn:** TS. Trần Văn Đặng

**Nhóm thực hiện:** 07

**Sinh viên thực hiện:** Hồ Tuấn Anh  
Nguyễn Tài Hoan  
Trần Mạnh Hùng

**Lớp:** Thực hành Cơ sở dữ liệu

**Hà Nội, Ngày 19 tháng 6 năm 2024**

# LỜI CẢM ƠN

Trong thời gian nghiên cứu, thiết kế và phát triển dự án, nhóm 07 đã nhận được nhiều sự hướng dẫn, chia sẻ và góp ý đến từ thầy cô và bạn bè. Tôi, với vai trò là trưởng nhóm đại diện, trước hết xin bày tỏ sự kính trọng và lòng biết ơn sâu sắc đến thầy **Trần Văn Đặng**, giảng viên bộ môn Thực hành Cơ sở dữ liệu, cùng với những thầy cô bộ môn nói chung và bạn bè trong chuyên ngành, đã tận tình chỉ dạy những kiến thức và kinh nghiệm quý báu để nhóm chúng tôi có thể hoàn thiện dự án một cách hiệu quả và tối ưu nhất.

Cuối cùng, quá trình xây dựng dự án không thể tránh khỏi sai sót. Chúng tôi mong rằng có thể nhận được nhiều ý kiến từ thầy cô và bạn bè để dự án được hoàn thiện tốt hơn.

## DANH SÁCH THÀNH VIÊN

STT	Họ và tên	Mã sinh viên	Vai trò
1	Trần Mạnh Hùng	20226108	Thiết kế mô hình dữ liệu. Xây dựng hệ thống bảng.
2	Nguyễn Tài Hoan	20226106	Quản lý hoạt động tìm kiếm, xử lý dữ liệu. Đề xuất chức năng nghiệp vụ và các câu truy vấn tương ứng.
3	Hồ Tuấn Anh	20226100	Đề xuất các phương án tối ưu. Kiểm thử và thống kê kết quả. Viết báo cáo thu hoạch.

## DANH MỤC THUẬT NGỮ VÀ TỪ VIẾT TẮT

Thuật ngữ	Ý nghĩa
SQL	Ngôn ngữ truy vấn mang tính cấu trúc (Structured Query Language)
ERD	Sơ đồ thực thể - liên kết (Entity-Relationship Diagram)
DBMS	Hệ quản trị cơ sở dữ liệu (Database Management System)

# MỤC LỤC

<b>1</b>	<b>GIỚI THIỆU ĐỀ TÀI</b>	<b>5</b>
1.1	Đặt vấn đề . . . . .	5
1.2	Phạm vi đề tài . . . . .	5
1.3	Đặc tả hệ thống . . . . .	5
1.4	Công nghệ sử dụng . . . . .	5
<b>2</b>	<b>THIẾT KẾ MÔ HÌNH HỆ THỐNG</b>	<b>6</b>
2.1	Xác định thực thể và thuộc tính . . . . .	6
2.2	Xác định quan hệ và ràng buộc . . . . .	7
2.3	Xây dựng lược đồ quan hệ . . . . .	7
2.3.1	Sơ đồ thực thể - liên kết . . . . .	7
2.3.2	Sơ đồ quan hệ . . . . .	8
<b>3</b>	<b>XÂY DỰNG CƠ SỞ DỮ LIỆU TRONG POSTGRESQL</b>	<b>9</b>
3.1	Tạo cơ sở dữ liệu và bảng . . . . .	9
3.2	Tạo Views, Functions, Triggers . . . . .	12
3.2.1	View . . . . .	12
3.2.2	Hàm và các trigger tương ứng . . . . .	13
3.2.3	Một số hàm quan trọng khác . . . . .	15
3.3	Danh sách chức năng và các câu truy vấn tương ứng . . . . .	16
3.4	Các phương án tối ưu . . . . .	32
<b>4</b>	<b>LỜI KẾT</b>	<b>34</b>

# CHƯƠNG 1. GIỚI THIỆU ĐỀ TÀI

## 1.1 Đặt vấn đề

Hiện nay, các mô hình dữ liệu quan hệ đã trở nên phổ biến và được áp dụng rộng rãi trong các lĩnh vực cần lưu trữ và quản lý dữ liệu lớn. Trong số đó có công tác tổ chức quản lý các giải thi đấu thể thao, đặc biệt là giải đấu bóng đá, một mô hình quan hệ phức tạp và đa dạng.

Khi tổ chức các giải đấu, một nhu cầu đặt ra là đảm bảo nguồn dữ liệu có thể tồn tại lâu dài và phát huy tối đa vai trò của nó, nhằm đẩy mạnh quá trình theo dõi, thống kê và phân tích. Do quy mô của các giải đấu bóng đá ngày càng lớn nên lượng dữ liệu cần lưu trữ cũng tăng nhanh theo từng năm nên việc tổ chức và quản lý cần được xây dựng một cách hợp lý và tối ưu. Như vậy, việc thiết kế xây dựng hệ thống quản lý tốt là điều cần thiết. Nắm bắt được yêu cầu này, nhóm chúng tôi quyết định đề xuất thiết kế một mô hình quản lý và lưu trữ dữ liệu liên quan đến giải đấu bóng đá.

## 1.2 Phạm vi đề tài

Xây dựng cơ sở dữ liệu lưu trữ dữ liệu thuộc các giải đấu bóng đá hàng đầu Châu Âu từ năm 2020 đến nay. Dữ liệu được thu thập và lưu trữ từ 1000 đến 10000 bản ghi trong mỗi bảng để kiểm thử. Đồng thời hệ thống tích hợp các chức năng tính toán, thống kê tự động giúp ban quản lý và người dùng có thể tra cứu thuận tiện.

## 1.3 Đặc tả hệ thống

Về mục đích xây dựng: Giúp ban quản lý giải đấu, cầu thủ, huấn luyện viên, người hâm mộ,... theo dõi thông tin một cách nhanh chóng và tiện lợi.

Về chức năng nghiệp vụ: Hệ thống được xây dựng với chức năng chính là lưu trữ và hỗ trợ truy vấn những thông tin về giải đấu, đội bóng, cầu thủ, lịch sử đối đầu,... Bên cạnh đó hệ thống còn hỗ trợ tính toán điểm số, thống kê thứ hạng tự động với sự hỗ trợ của ngôn ngữ truy vấn cấu trúc và một số ngôn ngữ lập trình khác.

Về yêu cầu: Xây dựng hệ thống một cách khoa học, tối ưu, trực quan, đáp ứng đầy đủ nhu cầu của người dùng. Đồng thời cần có nguồn dữ liệu chính xác được kiểm chứng trên trang chủ của các đội bóng cũng như của giải đấu và có sự cập nhật liên tục kịp thời.

## 1.4 Công nghệ sử dụng

DBMS: PostgreSQL (phiên bản mới nhất)

Hệ điều hành: Windows, Linux

Mã nguồn: <https://github.com/tuan6100/football-mangement-project>

## CHƯƠNG 2. THIẾT KẾ MÔ HÌNH HỆ THỐNG

### 2.1 Xác định thực thể và thuộc tính

#### 1. Thực thể quốc gia

STT	Tên thuộc tính	Ký hiệu
1	Mã quốc gia	nation_id
2	Tên quốc gia	nation_name
3	Châu lục	continent

#### 2. Thực thể giải đấu

STT	Tên thuộc tính	Ký hiệu
1	Mã giải đấu	league_id
2	Tên giải đấu	league_name
3	Thể thức	formula
4	Website	website

#### 3. Thực thể đội bóng

STT	Tên thuộc tính	Ký hiệu
1	Mã đội bóng	club_id
2	Tên đội bóng	club_name
3	Website	website

#### 4. Thực thể cầu thủ

STT	Tên thuộc tính	Ký hiệu
1	Mã cầu thủ	player_id
2	Tên cầu thủ	player_name
3	Ngày sinh	date_of_birth
4	Chiều cao	height
5	Chân thuận	preferred foot

#### 5. Thực thể trận đấu

STT	Tên thuộc tính	Ký hiệu
1	Mã trận đấu	match_id
2	Đội chủ nhà	home
3	Đội khách	away
4	Vòng đấu	round
5	Ngày tổ chức	date_of_match
6	Sân vận động	stadium
7	Trọng tài	referee

## 6. Thực thể huấn luyện viên

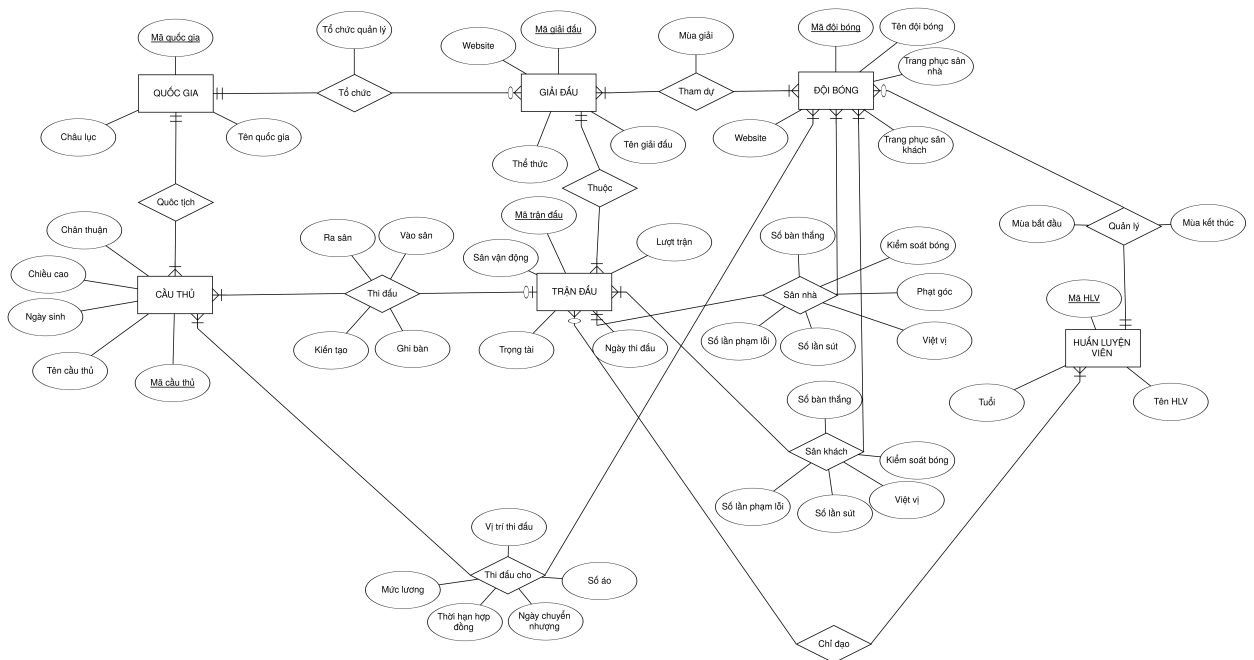
STT	Tên thuộc tính	Ký hiệu
1	Mã huấn luyện viên	manager_id
2	Tên huấn luyện viên	manager_name
3	Ngày sinh	date_of_birth

### 2.2 Xác định quan hệ và ràng buộc

- Mỗi giải đấu, câu lạc bộ, cầu thủ, huấn luyện viên đều xác định một quốc gia tương ứng.
- Mỗi giải đấu có từ 18 đến 20 câu lạc bộ tham dự, thi đấu theo thể thức đấu vòng tròn hoặc đấu cup. Đối với thể thức đấu vòng tròn, đội thắng trong mỗi trận đấu được cộng 3 điểm, nếu hòa được cộng 1 điểm, nếu thua thì không cộng điểm nào; sau mỗi trận đấu sẽ thực hiện xếp hạng theo số điểm nhận được.
- Mỗi đội bóng có tối thiểu 15 cầu thủ và 01 huấn luyện viên. Tỷ số của mỗi trận đấu được tính bằng tổng số bàn thắng mà cầu thủ ghi được trong trận đấu ấy.

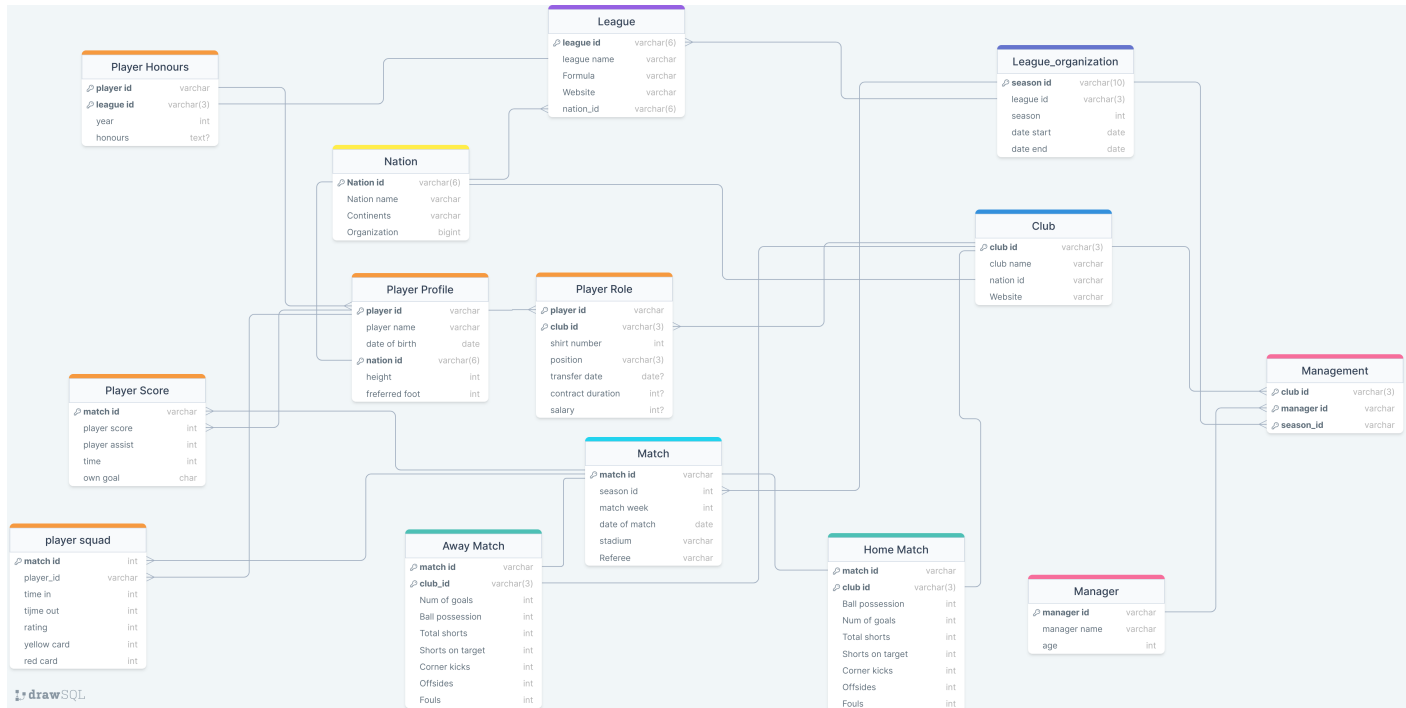
### 2.3 Xây dựng lược đồ quan hệ

#### 2.3.1 Sơ đồ thực thể - liên kết



**Hình 2.1:** Sơ đồ ERD biểu diễn mối quan hệ trong giải đấu bóng đá

### 2.3.2 Sơ đồ quan hệ



**Hình 2.2:** Sơ đồ quan hệ

**Nhận xét:** Sơ đồ quan hệ biểu diễn các bảng được hình thành từ sơ đồ thực thể - liên kết thông qua các phép biến đổi về dạng chuẩn 3NF hoặc dạng chuẩn Boyce-Codd nhằm tránh dư thừa dữ liệu và hiệu quả trong việc tìm kiếm.



## CHƯƠNG 3. XÂY DỰNG CƠ SỞ DỮ LIỆU TRONG POSTGRESQL

### 3.1 Tạo cơ sở dữ liệu và bảng

**Listing 3.1:** Tạo bảng

```
-- CREATE DATABASE --
CREATE DATABASE footballdb;

-- CREATE TABLE --
CREATE TABLE nation
(
    nation_id VARCHAR(6) PRIMARY KEY,
    nation_name VARCHAR(255) NOT NULL,
    continent VARCHAR(255) NOT NULL,
    organization VARCHAR
);

CREATE TABLE league
(
    league_id VARCHAR(6) PRIMARY KEY,
    league_name VARCHAR(255) NOT NULL,
    formula VARCHAR(255) NOT NULL,
    website VARCHAR(255) NOT NULL,
    nation_id VARCHAR(6) REFERENCES nation(nation_id)
);

CREATE TABLE league_organ
    season_id VARCHAR(10) PRIMARY KEY,
    league_id VARCHAR(6) REFERENCES league(league_id) NOT NULL,
    season VARCHAR(10) NOT NULL,
    date_start DATE,
    date_end DATE
);

CREATE TABLE club
(
    club_id VARCHAR(3) PRIMARY KEY,
    club_name VARCHAR(255) NOT NULL,
    home_kit VARCHAR(255) NOT NULL,
    away_kit VARCHAR(255) NOT NULL,
    website VARCHAR(255) NOT NULL
```

);

**CREATE TABLE match**

```
(
    match_id INT SERIAL PRIMARY KEY,
    season_id VARCHAR(10) REFERENCES league_organ(season_id),
    round VARCHAR(255),
    date_of_match DATE NOT NULL,
    stadium VARCHAR(255) NOT NULL,
    referee VARCHAR(255) NOT NULL
);
```

**CREATE TABLE home**

```
(
    match_id INT REFERENCES match(match_id) ON DELETE CASCADE,
    club_id VARCHAR(3) REFERENCES club(club_id) ,
    ball_possession INT L,
    num_of_goals INT ,
    total_shots INT ,
    shots_on_target INT ,
    corner_kicks INT ,
    offsides INT ,
    fouls INT ,
    penalties INT,
    PRIMARY KEY (match_id)
);
```

**CREATE TABLE away**

```
(
    match_id INT REFERENCES match(match_id) ON DELETE CASCADE,
    club_id VARCHAR(3) REFERENCES club(club_id) ,
    ball_possession INT ,
    num_of_goals INT,
    total_shots INT ,
    shots_on_target INT L,
    corner_kicks INT ,
    offsides INT ,
    fouls INT ,
    penalties INT,
    PRIMARY KEY (match_id)
);
```

```

CREATE TABLE player_profile
(
    player_id VARCHAR(6) PRIMARY KEY,
    player_name VARCHAR(255) NOT NULL,
    date_of_birth DATE NOT NULL,
    nation_id VARCHAR(6) REFERENCES nation(nation_id),
    height INT NOT NULL,
    preferred_foot CHAR NOT NULL CHECK (preferred_foot IN ('L', 'R'))
);

```

```

CREATE TABLE player_role
(
    season_id VARCHAR(10) REFERENCES league_organ(season_id),
    player_id VARCHAR(6) REFERENCES player_profile(player_id),
    club_id VARCHAR(3) REFERENCES club(club_id),
    shirt_number INT NOT NULL,
    position VARCHAR(3) NOT NULL,
    contract_duration DATE,
    total_goals INT,
    total_assists INT,
    PRIMARY KEY (player_id, club_id, season_id)
);

```

```

CREATE TABLE match_squad
(
    match_id INT REFERENCES match(match_id) ON DELETE CASCADE,
    player_id VARCHAR(6) REFERENCES player_profile(player_id),
    time_in INT CHECK (time_in >= 0 AND time_in <= 90),
    time_out INT CHECK (time_out >= 0 AND time_out <= 90),
    yellow_card INT CHECK (yellow_card IN (0, 1)),
    red_card INT CHECK (red_card IN (0, 1)),
    rating INT CHECK (rating >= 0 AND rating <= 10),
    PRIMARY KEY (match_id, player_id)
);

```

```

CREATE TABLE player_score
(
    match_id INT REFERENCES match(match_id) ON DELETE CASCADE,
    player_goal VARCHAR(6) REFERENCES player_profile(player_id),
    player_assist VARCHAR(6) REFERENCES player_profile(player_id),

```

```

        time_goal INT CHECK (time_goal >= 0 AND time_goal <= 90),
        PRIMARY KEY (match_id, time_goal)
    );

```

```

CREATE TABLE manager
(
    manager_id VARCHAR(20) PRIMARY KEY,
    manager_name VARCHAR(255) NOT NULL,
    date_of_birth VARCHAR NOT NULL
);

```

```

CREATE TABLE management
(
    club_id VARCHAR(3) REFERENCES club(club_id),
    manager_id VARCHAR(20) REFERENCES manager(manager_id),
    season_id VARCHAR(20) REFERENCES league_organ(season_id),
    PRIMARY KEY (club_id, manager_id, season_id)
);

```

## 3.2 Tạo Views, Functions, Triggers

### 3.2.1 View

Tạo view lưu trữ tỉ số của các trận đấu:

**Listing 3.2:** Tạo bảng

```

CREATE MATERIALIZED VIEW match_result AS
SELECT
    m.match_id,
    hm.club_id AS home_team,
    hm.num_of_goals AS home_score,
    am.club_id AS away_team,
    am.num_of_goals AS away_score,
    CASE
        WHEN hm.num_of_goals > am.num_of_goals THEN hm.club_id
        WHEN hm.num_of_goals < am.num_of_goals THEN am.club_id
        ELSE 'Draw'
    END AS match_winner
FROM
    match m
    JOIN home hm ON m.match_id = hm.match_id
    JOIN away am ON m.match_id = am.match_id;

```

### 3.2.2 Hàm và các trigger tương ứng

1. Trigger tự động thêm trường match\_id trong bảng home và away khi thêm mới bản ghi vào bảng match:

**Listing 3.3:** Tạo bảng

```
CREATE OR REPLACE FUNCTION update_match_id()  
RETURNS TRIGGER AS $$  
BEGIN  
    INSERT INTO home(match_id) VALUES (NEW.match_id);  
    INSERT INTO away(match_id) VALUES (NEW.match_id);  
    RETURN NEW;  
END;  
$$ LANGUAGE plpgsql;
```

```
CREATE OR REPLACE TRIGGER trigger_update_match_id  
AFTER INSERT ON match  
FOR EACH ROW  
EXECUTE FUNCTION update_match_id();
```

2. Trigger tự động tính toán tỷ lệ kiểm soát bóng của đội khách khi biết tỷ lệ kiểm soát bóng của đội nhà:

**Listing 3.4:** Tạo bảng

```
CREATE OR REPLACE FUNCTION calculate_ball_possession()  
RETURNS TRIGGER AS $$  
BEGIN  
    NEW.ball_possession = 100 - (SELECT ball_possession FROM home  
    RETURN NEW;  
END;  
$$ LANGUAGE plpgsql;
```

```
CREATE OR REPLACE TRIGGER trigger_calculate_ball_possession  
BEFORE INSERT ON away  
FOR EACH ROW  
EXECUTE FUNCTION calculate_ball_possession();
```

3. Trigger tự động cập nhật tỷ số của trận đấu khi cầu thủ trong trận ghi bàn:

**Listing 3.5:** Tạo bảng

```
CREATE OR REPLACE FUNCTION calculate_num_of_goals()  
RETURNS TRIGGER AS $$  
DECLARE  
    var_club_id VARCHAR(3);
```

```

BEGIN
    -- Check if the club_id is in the home or away team
    IF TG_OP = 'INSERT' OR TG_OP = 'UPDATE' THEN
        SELECT club_id
        INTO var_club_id
        FROM player_role
        INNER JOIN match ON player_role.season_id = match.season_id
        WHERE player_role.player_id = NEW.player_goal AND match.match_id = NEW.match_id;

        IF EXISTS (SELECT 1 FROM home WHERE home.match_id = NEW.match_id) THEN
            UPDATE home
            SET num_of_goals = num_of_goals + 1
            WHERE home.match_id = NEW.match_id;
        ELSIF EXISTS (SELECT 1 FROM away WHERE away.match_id = NEW.match_id) THEN
            UPDATE away
            SET num_of_goals = num_of_goals + 1
            WHERE away.match_id = NEW.match_id;
        END IF;

    ELSIF TG_OP = 'DELETE' THEN
        SELECT club_id
        INTO var_club_id
        FROM player_role
        INNER JOIN match ON player_role.season_id = match.season_id
        WHERE player_role.player_id = OLD.player_goal AND match.match_id = OLD.match_id;

        IF EXISTS (SELECT 1 FROM home WHERE home.match_id = OLD.match_id) THEN
            UPDATE home
            SET num_of_goals = num_of_goals - 1
            WHERE home.match_id = OLD.match_id;
        ELSIF EXISTS (SELECT 1 FROM away WHERE away.match_id = OLD.match_id) THEN
            UPDATE away
            SET num_of_goals = num_of_goals - 1
            WHERE away.match_id = OLD.match_id;
        END IF;
    END IF;

    REFRESH MATERIALIZED VIEW match_result;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

```

```

CREATE OR REPLACE TRIGGER trigger_calculate_num_of_goals
AFTER INSERT OR DELETE OR UPDATE ON player_score
FOR EACH ROW
EXECUTE FUNCTION calculate_num_of_goals();

```

### 3.2.3 Một số hàm quan trọng khác

1. Hàm tính toán số điểm nhận được của câu lạc bộ sau mỗi trận đấu:

**Listing 3.6:** Tạo bảng

```

CREATE OR REPLACE FUNCTION calculate_point(var_club_id VA
RETURNS INT AS $$
DECLARE point INT;
BEGIN
    SELECT
        SUM(CASE
            WHEN match_result.home_score > match_result.away_score
            WHEN match_result.home_score = match_result.away_score
            ELSE 0
        END)
    INTO point
    FROM match_result
    INNER JOIN match on match_result.match_id = match.match_id
    WHERE match.season_id = var_season_id
    AND match_result.home_team = var_club_id OR match_result.awa
    RETURN point;
END;
$$ LANGUAGE plpgsql;

```

2. Hàm tính toán hiệu số bàn thắng thua:

```

CREATE OR REPLACE FUNCTION calculate_goal_diff(var_club_id
RETURNS INT AS $$
DECLARE goal_diff INT;
BEGIN
    SELECT
        (SUM(match_result.home_score) - SUM(match_result.away_score))
    INTO goal_diff
    FROM match_result
    INNER JOIN match on match_result.match_id = match.match_id
    WHERE match.season_id = var_season_id
    AND match_result.home_team = var_club_id OR match_result.awa
    RETURN goal_diff;
END;

```

```
$$ LANGUAGE plpgsql;
```

### 3. Hàm đếm số trận đã thi đấu của câu lạc bộ:

**Listing 3.7:** Tạo bảng

```
CREATE OR REPLACE FUNCTION count_match_played(var_club_id VAR
RETURNS INT AS $$
DECLARE num INT;
BEGIN
    SELECT
        COUNT(match_result.match_id)
    INTO num
    FROM match_result
    INNER JOIN match on match_result.match_id = match.match_id
    WHERE match.season_id = var_season_id
    AND match_result.home_team = var_club_id OR match_result.away_tea
    RETURN num;
END;
$$ LANGUAGE plpgsql;
```

### 3.3 Danh sách chức năng và các câu truy vấn tương ứng

#### 1. Thêm một trận đấu mới

**Listing 3.8:** Tạo bảng

```
INSERT INTO match(match_id, season_id, round, date_of_match, stadium,
VALUES ('6119', 'LIGUE12324', '38', '09/01/2024', 'Stade Bollaert-Delelis
```

#### 2. Cập nhật số liệu thống kê thu được trong các trận đấu sau mỗi trận đấu

**Listing 3.9:** Tạo bảng

```
-- trigger t ng c p n h t m t r n u v o b
CREATE OR REPLACE FUNCTION UPDATE_match_id()
RETURNS TRIGGER AS $$
BEGIN
    INSERT INTO home(match_id) VALUES (NEW.match_id);
    NSERT INTO away(match_id) VALUES (NEW.match_id);
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE OR REPLACE TRIGGER trigger_UPDATE_match_id
AFTER INSERT ON match
```



```

FOR EACH ROW
EXECUTE FUNCTION UPDATE_match_id();

-- trigger t_nh_t_l_kim_sot_bng
CREATE OR REPLACE FUNCTION calculate_ball_possession()
RETURNS TRIGGER AS $$
BEGIN
    NEW.ball_possession = 100 - (SELECT ball_possession FROM home
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE OR REPLACE TRIGGER trigger_calculate_ball_possession
BEFORE INSERT ON away
FOR EACH ROW
EXECUTE FUNCTION calculate_ball_possession();

```

### 3. Cập nhật tỷ số cho trận đấu

**Listing 3.10:** Tạo bảng

```

CREATE OR REPLACE FUNCTION calculate_num_of_goals()
RETURNS TRIGGER AS $$
DECLARE
    var_club_id VARCHAR(3);
BEGIN
    IF TG_OP = 'INSERT' OR TG_OP = 'UPDATE' THEN
        SELECT club_id
        INTO var_club_id
        FROM player_role
        INNER JOIN match ON player_role.season_id = match.season_id
        WHERE player_role.player_id = NEW.player_goal AND match.m

    IF EXISTS (SELECT 1 FROM home WHERE home.match_id = NEW.m
        UPDATE home
        SET num_of_goals = num_of_goals + 1
        WHERE home.match_id = NEW.match_id;
    ELSIF EXISTS (SELECT 1 FROM away WHERE away.match_id = NEW.m
        UPDATE away
        SET num_of_goals = num_of_goals + 1
        WHERE away.match_id = NEW.match_id;
    END IF;

```

```

ELSIF TG_OP = 'DELETE' THEN
    SELECT club_id
    INTO var_club_id
    FROM player_role
    INNER JOIN match ON player_role.season_id = match.season_id
    WHERE player_role.player_id = OLD.player_id AND match.match_id = OLD.match_id;

    IF EXISTS (SELECT 1 FROM home WHERE home.match_id = OLD.match_id) THEN
        UPDATE home
        SET num_of_goals = num_of_goals - 1
        WHERE home.match_id = OLD.match_id;
    ELSEIF EXISTS (SELECT 1 FROM away WHERE away.match_id = OLD.match_id) THEN
        UPDATE away
        SET num_of_goals = num_of_goals - 1
        WHERE away.match_id = OLD.match_id;
    END IF;
END IF;
REFRESH MATERIALIZED VIEW match_result;
RETURN NEW;
END;
$$ LANGUAGE plpgsql;

CREATE OR REPLACE TRIGGER trigger_calculate_num_of_goals
AFTER INSERT OR DELETE OR UPDATE ON player_score
FOR EACH ROW
EXECUTE FUNCTION calculate_num_of_goals();

-- view      t h e o   d i   t       s       t r n       u
CREATE MATERIALIZED VIEW match_result AS
SELECT
    m.match_id ,
    hm.club_id AS home_team ,
    hm.num_of_goals AS home_score ,
    am.club_id AS away_team ,
    am.num_of_goals AS away_score ,
    CASE
        WHEN hm.num_of_goals > am.num_of_goals THEN hm.club_id
        WHEN hm.num_of_goals < am.num_of_goals THEN am.club_id
        ELSE 'Draw'
    END AS match_winner

```

```

match m
JOIN home hm ON m.match_id = hm.match_id
JOIN away am ON m.match_id = am.match_id;

```

- ```
UPDATE participation SET num_of_matches = count_match_played(club_id, season_id);
UPDATE participation SET point = calculate_point(club_id, season_id);
UPDATE participation SET goal_diff = calculate_goal_diff(club_id, season_id);
```

- ```
DELETE FROM match WHERE match_id = '6119';  
DELETE FROM match WHERE date_of_match = '09/01/2024' AND stadium
```

- ```
UPDATE player_role
SET club_id = 'RMA' AND season_id = 'LALIGA2425'
WHERE player_id = (SELECT player_id FROM player_profile WHERE pla
```

- ```
-- toi uu
SELECT
    club.club_name ,
    SUM(CASE
        WHEN match_result.home_team = club.club_id AND match_resu
        WHEN match_result.away_team = club.club_id AND match_resu
        ELSE 0
    END) as win ,
    SUM(CASE
        WHEN match_result.home_team = club.club_id AND match_resu
        WHEN match_result.away_team = club.club_id AND match_resu
        ELSE 0
    END) as draw ,
    SUM(CASE
        WHEN match_result.home_team = club.club_id AND match_resu
```

```

        WHEN match_result.away_team = club.club_id AND match_result.a
        ELSE 0
    END) as lose
FROM match_result
INNER JOIN match ON match_result.match_id = match.match_id
INNER JOIN club ON club.club_id = match_result.home_team OR club.club
WHERE club.club_name = '...'
GROUP BY club.club_name;

-- chua toi uu
WITH MatchResults AS (
    SELECT
        hm.club_id AS club_id ,
        m.match_id ,
        CASE
            WHEN hm.num_of_goals > am.num_of_goals THEN 'Win'
            WHEN hm.num_of_goals < am.num_of_goals THEN 'Loss'
            ELSE 'Draw'
        END AS result
    FROM
        match m
        JOIN home hm ON m.match_id = hm.match_id
        JOIN away am ON m.match_id = am.match_id
    UNION ALL
    SELECT
        am.club_id AS club_id ,
        m.match_id ,
        CASE
            WHEN am.num_of_goals > hm.num_of_goals THEN 'Win'
            WHEN am.num_of_goals < hm.num_of_goals THEN 'Loss'
            ELSE 'Draw'
        END AS result
    FROM
        match m
        JOIN home hm ON m.match_id = hm.match_id
        JOIN away am ON m.match_id = am.match_id
)
SELECT
    club.club_name ,
    COUNT(CASE WHEN mr.result = 'Win' THEN 1 END) AS Wins ,
    COUNT(CASE WHEN mr.result = 'Draw' THEN 1 END) AS Draws ,

```

```

COUNT(CASE WHEN mr.result = 'Loss' THEN 1 END) AS Losses
FROM
    MatchResults mr
JOIN club ON mr.club_id = club.club_id
WHERE club.club_name = '...'
GROUP BY
    club.club_name
ORDER BY
    club.club_name;

```

8. Thống kê tổng số bàn thắng, kiến tạo của cầu thủ sau từng trận đấu

**Listing 3.15:** Tạo bảng

```

-- tinh tong so ban thang
alter table player_role add column total_scores int;

create or replace function UPDATE_total_goals(var_playerid varchar)
returns int as $$
declare total int;
begin
    select count(*) into total
    from player_score
    where player_score.player_goal = var_playerid
    and (select season_id from match where match.match_id = player_role.match_id)
    RETURN total;
end;
$$ language plpgsql;

UPDATE player_role SET total_goals = UPDATE_total_goals(player_id);

-- tinh tong so kien tao
alter table player_role add column total_assists int;

create or replace function UPDATE_total_assists(var_playerid varchar)
returns int as $$
declare total int;
begin
    select count(*) into total
    from player_score
    where player_score.player_assist = var_playerid
    and (select season_id from match where match.match_id = player_role.match_id)
    RETURN total;

```

```
end ;
$$ language plpgsql;
```

```
UPDATE player_role SET total_assists = UPDATE_total_assists(player_id
```

9. Thống kê số phút thi đấu trung bình của cầu thủ trong mùa giải S?

**Listing 3.16:** Tạo bảng

```
select player_profile.player_name ,
       round(avg(match_squad.time_out - match_squad.time_in ),2) as a
from match_squad
inner join player_profile on match_squad.player_id = player_profile.p
inner join match on match_squad.match_id = match.match_id
where match.season_id = '...'
group by player_profile.player_name
order by avg_time desc;
```

10. Tìm cầu thủ xuất sắc nhất trong trận đấu (MOTM)

**Listing 3.17:** Tạo bảng

```
SELECT player_profile.player_name , match_squad.rating AS max_rating
FROM match_squad
INNER JOIN player_profile ON match_squad.player_id = player_profile.p
WHERE match_squad.match_id = '...'
AND match_squad.rating =
(
    SELECT MAX(rating)
    FROM match_squad
    WHERE match_id = '...'
);
```

11. Trả về nhà vô địch của các giải đấu

**Listing 3.18:** Tạo bảng

```
WITH ranked_clubs AS (
    SELECT club_id , season_id ,
           RANK() OVER (PARTITION BY season_id ORDER BY point DESC, g
    FROM participation
    WHERE season_id = '...'
)
UPDATE participation
SET state= 'Champion'
WHERE (club_id , season_id) IN (
```

```

SELECT club_id , season_id
FROM ranked_clubs
WHERE rank = 1
);

```

12. . Trả về danh sách các CLB tham dự giải đấu UEFA Champion League mùa sau:

**Listing 3.19:** Tạo bảng

```

WITH ranked_clubs AS (
    SELECT club_id , season_id ,
           RANK() OVER (PARTITION BY season_id ORDER BY point DESC)
    FROM participation
    WHERE season_id = '...'
)
UPDATE participation
SET state= 'C1'
WHERE (club_id , season_id) IN (
    SELECT club_id , season_id
    FROM ranked_clubs
    WHERE rank IN (2, 3 ,4)
);

```

13. Trả về danh sách các CLB tham dự giải đấu UEFA Europa League mùa sau

**Listing 3.20:** Tạo bảng

```

WITH ranked_clubs AS (
    SELECT club_id , season_id ,
           RANK() OVER (PARTITION BY season_id ORDER BY point DESC)
    FROM participation
    WHERE season_id = 'EPL2324'
)
UPDATE participation
SET state= 'C2'
WHERE (club_id , season_id) IN (
    SELECT club_id , season_id
    FROM ranked_clubs
    WHERE rank IN (5, 6)
    UNION
    SELECT match.season_id , match_result.match_winner
    FROM match
    INNER JOIN match_result ON match.match_id = match_result.match_id
    INNER JOIN club ON match_result.match_winner = club.club_id
    INNER JOIN league_organ ON match.season_id = league_organ.season_id
);

```

```

        INNER JOIN league ON league_organ.league_id = league.league_id
WHERE match.round = 'Chung kết' and league.league_id like '%0002'
);

```

14. Tỷ lệ thắng thua của đội bóng A trong giải đấu B

**Listing 3.21:** Tạo bảng

```

SELECT
    'LIV' AS team_a,
    'MCI' AS versus,
    COUNT(*) AS total_matches,
    COUNT(CASE WHEN match_winner = 'LIV' THEN 1 END) AS wins,
    COUNT(CASE WHEN match_winner = 'MCI' THEN 1 END) AS losses,
    COUNT(CASE WHEN match_winner = 'Draw' THEN 1 END) AS draws,
    ROUND(COUNT(CASE WHEN match_winner = 'LIV' THEN 1 END) * 100.0 / COUNT(*)) AS win_percent,
    ROUND(COUNT(CASE WHEN match_winner = 'MCI' THEN 1 END) * 100.0 / COUNT(*)) AS loss_percent
FROM
    match_result mr
WHERE
    (mr.home_team = 'LIV' AND mr.away_team = 'MCI')
    OR (mr.home_team = 'MCI' AND mr.away_team = 'LIV');

```

15. Trả về Tỷ lệ thắng thua của đội bóng A trong giải đấu B

**Listing 3.22:** Tạo bảng

```

SELECT
    COUNT(*) AS total_matches,
    COUNT(CASE WHEN match_winner = 'LIV' THEN 1 END) AS wins,
    COUNT(CASE WHEN match_winner != 'LIV' THEN 1 END) AS losses,
    COUNT(CASE WHEN match_winner = 'Draw' THEN 1 END) AS draws,
    ROUND(COUNT(CASE WHEN match_winner = 'LIV' THEN 1 END) * 100.0 / COUNT(*)) AS win_percent,
    ROUND(COUNT(CASE WHEN match_winner != 'LIV' THEN 1 END) * 100.0 / COUNT(*)) AS loss_percent
FROM
    match_result mr
    INNER JOIN match ON mr.match_id = match.match_id
WHERE
    match.season_id = 'EPL2324' AND
    (home_team = 'LIV' OR away_team = 'LIV');

```

16. Trả về vua phá lưới của giải đấu trong mùa giải

**Listing 3.23:** Tạo bảng

```

SELECT

```



```

        player_profile.player_name ,
        player_role.total_goals as total_goals
FROM player_role
INNER JOIN player_profile ON player_role.player_id = player_profi
WHERE player_role.season_id = '...'
AND player_role.total_goals =
(
    SELECT MAX(total_goals)
    FROM player_role
    WHERE season_id = '...'
);

```

#### 17. Kiểm tra điều kiện thi đấu của cầu thủ

**Listing 3.24:** Tạo bảng

```

ALTER TABLE player_role ADD column total_cards INT;

```

```

CREATE FUNCTION UPDATE_total_cards()
RETURNS TRIGGER AS $$
BEGIN
    UPDATE plater_role
    SET total_cards = total_cards + NEW.yellow_card
    WHERE player_role.player_id = NEW.player_id;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;

```

```

CREATE TRIGGER trigger_UPDATE_total_card
AFTER INSERT OR UPDATE ON match_squad
FOR EACH ROW
EXECUTE FUNCTION UPDATE_total_cards();

```

```

-- k i m   t r a   s       t h   v n g   l       b i   c a   5

```

```

CREATE OR REPLACE FUNCTION check_cond_player()
RETURNS TRIGGER AS $$
BEGIN
    IF (SELECT total_cards FROM plater_role WHERE player_id = NEW
        RAISE EXCEPTION 'Player % is banned in this match', NEW.p
    END IF;
    RETURN NEW;
END;

```

```
$$ LANGUAGE plpgsql;
```

```
CREATE OR REPLACE TRIGGER trigger_check_cond_player  
BEFORE INSERT ON match_squad  
FOR EACH ROW  
EXECUTE FUNCTION check_cond_player();
```

18. Liệt kê danh sách cầu thủ thuộc biên chế CLB C? trong mùa giải L?

**Listing 3.25:** Tạo bảng

```
SELECT  
pr.club_id , pp.player_name , pp.date_of_birth , pp.nation_id , p  
FROM  
player_role pr  
JOIN player_profile pp ON pr.player_id = pp.player_id  
WHERE  
pr.club_id = 'C?' AND pr.season_id = 'L?';
```

19. Liệt kê danh sách cầu thủ có quốc tịch N? và đang thi đấu ở giải đấu L?

**Listing 3.26:** Tạo bảng

```
-- chua toi uu  
SELECT player_profile.player_name  
FROM player_profile  
INNER JOIN player_role ON player_profile.player_id = player_role.play  
INNER JOIN nation ON player_profile.nation_id = nation.nation_id  
INNER JOIN league_organ ON player_role.season_id = league_organ.seaso  
INNER JOIN league ON league_organ.league_id = league.league_id  
WHERE nation.nation_name = 'N?'  
AND league.league_name = 'L?'  
AND CURRENT_DATE >= league_organ.date_start;  
  
--toi uu  
CREATE INDEX idx_nation_name on nation (nation_name);  
CREATE INDEX idx_league_name on league (league_name);
```

20. Liệt kê danh sách cầu thủ dưới ... tuổi đang thi đấu ở giải đấu L?

**Listing 3.27:** Tạo bảng

```
--chua toi uu  
SELECT player_profile.player_name ,  
AGE(now() , player_profile.date_of_birth) as age  
FROM player_profile
```

```

INNER JOIN player_role ON player_profile.player_id = player_role.player_id
INNER JOIN league_organ ON player_role.season_id = league_organ.season_id
INNER JOIN league ON league_organ.league_id = league.league_id
WHERE EXTRACT(YEAR FROM AGE(player_profile.date_of_birth)) < 23;

--toi uu
ALTER TABLE player_profile ADD COLUMN age INT;

CREATE OR REPLACE FUNCTION dob_to_age(var_dob DATE)
RETURNS INT AS $$
BEGIN
    RETURN EXTRACT(YEAR FROM AGE(var_dob));
END;
$$ LANGUAGE plpgsql;

create index idx_player_age on player_profile (age) ;

SELECT player_profile.player_name ,
       AGE(CURRENT_DATE, player_profile.date_of_birth) as age
FROM player_profile
INNER JOIN player_role ON player_role.player_id = player_profile.player_id
INNER JOIN league_organ ON player_role.season_id = league_organ.season_id
INNER JOIN league ON league_organ.league_id = league.league_id
WHERE player_profile.age < 23;

```

21. Tìm cầu thủ có chiều cao lớn nhất và thi đấu ở vị trí ?

**Listing 3.28:** Tạo bảng

```

-- chua toi uu
SELECT player_profile.player_name , player_profile.height
FROM player_profile
WHERE player_profile.height =
(
    SELECT MAX(height)
    FROM player_profile
    INNER JOIN player_role ON player_profile.player_id = player_role.player_id
    WHERE player_role.position = 'ST'
);

-- toi uu
CREATE INDEX idx_player_height on player_profile (height);
CREATE INDEX idx_player_position on player_role using hash (position);

```

22. Liệt kê danh sách cầu thủ được chuyển nhượng sau mùa giải S?

**Listing 3.29:** Tạo bảng

```
-- ham tim mua giai tiep theo
CREATE OR REPLACE FUNCTION next_season(var_season VARCHAR(10))
RETURNS VARCHAR(10) AS $$
DECLARE
    year1 INT;
    year2 INT;
BEGIN
    year1 := CAST(SPLIT_PART(var_season, '-', 1) AS INT);
    year2 := CAST(SPLIT_PART(var_season, '-', 2) AS INT);
    RETURN TO_CHAR(year1 + 1, 'FM0000') || '-' || TO_CHAR(year2 + 1,
END;
$$ LANGUAGE plpgsql;

-- tim cau thu co 2 ma clb khac nhau
CREATE OR REPLACE FUNCTION transferred_players(var_season VARCHAR(20))
RETURNS TABLE (
    player_name VARCHAR(50),
    old_club_id VARCHAR(50),
    new_club_id VARCHAR(50)
) AS $$
BEGIN
    RETURN QUERY
    WITH old_club AS (
        SELECT pr.player_id, pr.club_id
        FROM player_role pr
        INNER JOIN league_organ lo ON pr.season_id = lo.season_id
        WHERE lo.season = var_season
    ),
    new_club AS (
        SELECT pr.player_id, pr.club_id
        FROM player_role pr
        INNER JOIN league_organ lo ON pr.season_id = lo.season_id
        WHERE lo.season = next_season(var_season)
    )
SELECT
    pp.player_name,
    old.club_name AS old_club_name,
    new.club_name AS new_club_name
```

```

FROM player_profile pp
INNER JOIN old_club oc ON pp.player_id = oc.player_id
INNER JOIN new_club nc ON pp.player_id = nc.player_id
INNER JOIN club old ON oc.club_id = old.club_id
INNER JOIN club new ON nc.club_id = new.club_id
WHERE oc.club_id != nc.club_id;
END;
$$ LANGUAGE plpgsql;

```

```

SELECT * FROM transferred_players('2022-2023');

```

23. (Hoan) liệt kê danh sách cầu thủ còn hợp đồng ... năm

**Listing 3.30:** Tạo bảng

```

SELECT pp.player_name , pr.club_id , pr.contract_duration
FROM player_profile pp
JOIN player_role pr ON pp.player_id = pr.player_id
WHERE season_id LIKE '%2324' AND pr.contract_duration = ...

```

24. Liệt kê top 10 cầu thủ thi đấu nhiều trận nhất ở giải đấu L? trong mùa giải S?

**Listing 3.31:** Tạo bảng

```

SELECT player_profile.player_name , COUNT(match_squad.player_id) as num_of_matches
FROM match_squad
INNER JOIN player_profile ON match_squad.player_id = player_profile.player_id
INNER JOIN match ON match_squad.match_id = match.match_id
INNER JOIN league_organ ON match.season_id = league_organ.season_id
INNER JOIN league ON league_organ.league_id = league.league_id
WHERE league.league_name = 'L?'
AND league_organ.season = 'S?'
GROUP BY player_profile.player_id
ORDER BY num_of_matches DESC LIMIT 10;

```

25. Top tỉ lệ cầm bóng cao nhất của giải đấu A

**Listing 3.32:** Tạo bảng

```

SELECT
    c.club_name ,
    c.club_id ,
    ROUND(AVG(ball_possession),2) AS average_ball_possession
FROM (
    SELECT
        hm.club_id ,

```

```

        hm.ball_possession
FROM home hm
JOIN match m ON hm.match_id = m.match_id
WHERE m.season_id = 'BUNDES2324'
UNION ALL
SELECT
        am.club_id ,
        am.ball_possession
FROM away am
JOIN match m ON am.match_id = m.match_id
WHERE m.season_id = 'BUNDES2324'
) AS cm
JOIN club c ON c.club_id = cm.club_id
GROUP BY
        c.club_id ,
        c.club_name
ORDER BY average_ball_possession DESC
LIMIT 10;

```

26. Liệt kê danh sách cầu thủ ghi được hattrick trong 1 trận đấu trong mùa giải S?

**Listing 3.33:** Tạo bảng

```

SELECT player_profile.player_name
FROM player_profile
INNER JOIN player_score ON player_profile.player_id = player_score.pl
INNER JOIN match ON player_score.match_id = match.match_id
INNER JOIN league_organ ON match.season_id = league_organ.season_id
WHERE league_organ.season = '2023-2024'
GROUP BY player_profile.player_name , match.match_id
HAVING COUNT(player_score.player_goal) >= 3;

```

27. Liệt kê tỉ số của các trận đấu diễn ra trong ngày D?

**Listing 3.34:** Tạo bảng

```

-- chưa tối ưu
SELECT match_result.*
FROM match_result
INNER JOIN match ON match_result.match_id = match.match_id
WHERE match.date_of_match = 'D?';

-- tối ưu
CREATE INDEX idx_date on match (date_of_match);

```

28. Lịch sử đối đầu của 2 đội bóng

**Listing 3.35:** Tạo bảng

```
SELECT
    m.season_id ,
    m.round ,
    m.stadium ,
    m.referee ,
    mr.home_team AS home_club ,
    mr.away_team AS away_club ,
    mr.home_score AS home_goals ,
    mr.away_score AS away_goals ,
CASE
    WHEN mr.home_score > mr.away_score THEN mr.home_team
    WHEN mr.home_score < mr.away_score THEN mr.away_team
    ELSE 'Draw'
END AS match_winner
FROM
    match m
JOIN match_result mr ON m.match_id = mr.match_id
WHERE
    (mr.home_team = 'LIV' AND mr.away_team = 'MCI')
    OR (mr.home_team = 'MCI' AND mr.away_team = 'LIV')
ORDER BY
    m.season_id ,
    m.round ;
```

29. Liệt kê danh sách những CLB vô địch từ mùa giải S? đến nay ở giải đấu L?

**Listing 3.36:** Tạo bảng

```
-- chua toi uu
SELECT club.club_name
FROM club
INNER JOIN participation ON club.club_id = participation.club_id
INNER JOIN league_organ ON participation.season_id = league_organ.season_id
INNER JOIN league ON league_organ.league_id = league.league_id
WHERE league_organ.date_end >= 'S?'
AND league.league_name = 'L?'
AND participation.state = 'Champion';

-- toi uu
CREATE INDEX idx_state on participation (state);
```

30. Liệt kê danh sách các CLB mà HLV M? đã chỉ đạo từ năm ? Đến nay

**Listing 3.37:** Tạo bảng

```
-- chưa tối ưu
SELECT DISTINCT manager.manager_name , club.club_name
FROM manager
INNER JOIN management ON manager.manager_id = management.manager_id
INNER JOIN club ON management.club_id = club.club_id
INNER JOIN league_organ ON management.season_id = league_organ.season_id
WHERE manager.manager_name = '...' AND league_organ.date_start >= '...'

-- tối ưu
CREATE INDEX idx_manager_name on manager (manager_name);
```

### 3.4 Các phương án tối ưu

- Tạo index trong câu truy vấn cần tìm kiếm
- Tạo materialized view thay vì view để nhanh chóng truy vấn



<b>Nội dung truy vấn</b>	<b>Chi phí thực hiện (S)</b>	
	<b>Trước tối ưu</b>	<b>S</b>
Thống kê tỷ lệ thắng, hòa, thua của clb	393.58	
Liệt kê danh sách cầu thủ dưới ... tuổi đang thi đấu ở giải đấu L?	23.38	
Liệt kê danh sách các CLB mà HLV M? đã chỉ đạo từ năm ? Đến nay	25.74	
Tìm kiếm lịch sử đối đầu giữa 2 CLB C? và C?	338.7	
Liệt kê danh sách cầu thủ dưới ... tuổi đang thi đấu ở giải đấu L?	855.08	
Tìm cầu thủ có chiều cao lớn nhất và thi đấu ở vị trí P?	1031.15	
Liệt kê tỉ số của các trận đấu diễn ra trong ngày D?	345.95	

**Hình 3.1:** Biểu đồ so sánh chi phí tối ưu và chưa tối ưu

## CHƯƠNG 4. LỜI KẾT

Báo cáo đã trình bày những quy trình và phương án thiết kế cơ sở dữ liệu lưu trữ giải đấu bóng đá theo định hướng tự động hóa và tối ưu các quy trình, các chức năng nghiệp vụ nhằm hướng tới những trải nghiệm tốt nhất cho người dùng. Trong thời gian thiết kế dự án, nhóm chúng tôi đã tích lũy và học hỏi thêm nhiều kiến thức, kinh nghiệm trong thiết kế cơ sở dữ liệu và kỹ năng triển khai dự án theo nhóm. Song vì thời gian và trình độ hạn chế nên trong quá trình xây dựng hệ thống, nhóm chúng tôi còn gặp phải một số khó khăn như kích thước dữ liệu còn nhỏ để có thể thấy rõ sự tối ưu. Đồng thời nhóm chưa có khả năng xây dựng website hoàn thiện nhằm mang tới những trải nghiệm tốt nhất cho người dùng. Nhóm sẽ tiếp tục nâng cao chất lượng hệ thống trong tương lai gần.