

Báo cáo Bài tập lớn Cơ sở Dữ liệu

Mã học phần: INT2211_37.

Giảng viên: TS. Trần Hồng Việt.

Tên nhóm: 5555.

Các thành viên trong nhóm:

1. Đoàn Quang Huy - 23020374.
2. Nguyễn Đức Huy - 23020376.
3. Ngô Nguyễn Khải Hưng - 23020382.
4. Tạ Quang Linh - 23020396.

Chủ đề: Hệ thống dữ liệu quản lý về thứ hạng, trận đấu, giải đấu của các tay vợt tennis thuộc Hiệp hội Quần vợt Chuyên nghiệp (ATP).

Thứ tự các nhiệm vụ:

1. Tạo hệ thống cơ sở dữ liệu:

Người thực hiện: Đoàn Quang Huy.

a. Class: players (vận động viên):

- **player_id** (Mã tay vợt): Mã số riêng biệt của các tay vợt, được biểu diễn bởi một số nguyên.
- **first_name** (Tên): Tên của các tay vợt, được biểu diễn bởi một xâu có tối đa 100 kí tự.
- **last_name** (Họ): Họ của các tay vợt, được biểu diễn bởi một xâu có tối đa 100 kí tự.
- **hand** (Tay): Tay thuận của các tay vợt, được biểu diễn bởi 1 kí tự chữ duy nhất là một trong 3 chữ: L - Trái, R - Phải, U - Chưa biết.
- **date_of_birth** (Ngày sinh): Ngày sinh các tay vợt, được biểu diễn bởi một ngày được viết dưới dạng: yyyy-mm-dd.
- **nationality** (Quốc tịch): Quốc tịch của các tay vợt, được biểu diễn bởi một xâu có 3 kí tự.

- **height** (Chiều cao): Chiều cao của các tay vợt, được biểu diễn bởi một số nguyên.

b. Class: **single_ranking** (thứ hạng đánh đơn):

- **ranking_date** (Ngày xếp hạng): Ngày cập nhật bảng xếp hạng, được biểu diễn bởi một ngày được viết dưới dạng: yyyy-mm-dd.
- **ranking** (Thứ hạng): Thứ hạng trên bảng xếp hạng được cập nhật, được biểu diễn bởi một số nguyên lớn hơn 0 và nhỏ hơn bằng 200.
- **player** (Tay vợt): Mã số của tay vợt tương ứng với thứ hạng và ngày cập nhật bảng xếp hạng, được biểu diễn bởi một số nguyên.
- **points** (Điểm): Số điểm mà tay vợt tích lũy được qua các giải đấu, được biểu diễn bởi một số nguyên.

c. Class: **matches_94_04_14_24** (các trận đấu trong năm 1994, 2004, 2014 và 2024 - dữ liệu trong năm 2024 được cập nhật đến ngày 27/05/2024):

- **tournament_key** (Mã giải đấu): Mã số của giải đấu nơi trận đấu diễn ra, được biểu diễn bởi một số nguyên.
- **surface** (Mặt sân): Mặt sân của giải đấu vào năm đó, được biểu diễn bằng một xâu có tối đa 10 kí tự.
- **draw_size** (Kích thước nhánh): Số tay vợt / đội tuyển tham dự giải, được biểu diễn bởi một số nguyên.
- **tournament_date** (Ngày diễn ra giải): Ngày giải đấu khởi tranh, được biểu diễn bởi một ngày được viết dưới dạng: yyyy-mm-dd.
- **winner_id** (Mã tay vợt chiến thắng): Mã số của tay vợt giành chiến thắng trận đấu, được biểu diễn bởi một số nguyên.
- **loser_id** (Mã tay vợt thất bại): Mã số của tay vợt thất bại trận đấu, được biểu diễn bởi một số nguyên.
- **score** (Tỉ số): Tỉ số cuối cùng của trận đấu, được biểu diễn bởi một xâu có tối đa 100 kí tự.
- **best_of** : Số set tối đa trong một trận đấu, thể hiện điều kiện để một tay vợt giành chiến thắng trận đấu (nếu không có trường hợp bỏ cuộc hoặc bị xử thua), đó là giành chiến thắng hơn nửa số set được ghi nhận; được biểu diễn bởi một số nguyên.

- `round` (Vòng): Vòng diễn ra trận đấu trong giải đấu tương ứng, được biểu diễn bởi một chuỗi có tối đa 50 ký tự.

d. Class: `tournaments` (giải đấu):

- `tournament_key` (Mã giải đấu): Mã số của giải đấu, được biểu diễn bởi một số nguyên.
- `tournament_name` (Tên giải đấu): Tên giải đấu hoặc địa điểm nơi giải đấu diễn ra, được biểu diễn bằng văn bản.

2. Mô hình thực thể - liên kết:

a. Các thực thể:

- `players` (vận động viên):
`player_id`, `first_name`, `last_name`, `hand`, `date_of_birth`, `nationality`, `height`.
- `single_ranking` (thứ hạng đánh đơn):
`ranking_date`, `ranking`, `player`, `points`.
- `matches_94_04_14_24` (các trận đấu trong năm 94, 04, 14, 24):
`tournament_key`, `surface`, `draw_size`, `tournament_date`, `winner_id`, `loser_id`, `score`, `best_of`, `round`.
- `tournaments` (giải đấu):
`tournament_key`, `tournament_name`.

b. Các liên kết:

- `players` và `single_ranking`: Quan hệ Một-Nhiều (Một tay vợt có thể xuất hiện ở bảng xếp hạng Top 200 ATP ở 2 thời điểm khác nhau).
- `players` và `matches_94_04_14_24`: Quan hệ Một-Nhiều (Một tay vợt có thể tham gia nhiều trận đấu khác nhau).
- `tournaments` và `matches_94_04_14_24`: Quan hệ Một-Nhiều (Một giải đấu có nhiều trận đấu khác nhau).

3. Mô hình quan hệ:

a. Các thực thể và thuộc tính:

- `players`:
 - `player_id` (Primary Key).

- `first_name` .
- `last_name` .
- `hand` .
- `date_of_birth` .
- `nationality` .
- `height` .
- `single_ranking` :
 - `ranking_date` .
 - `ranking` .
 - `player` (Foreign Key referencing `player_id` in `players`).
 - `points` .
- `matches_94_04_14_24` :
 - `tournament_key` (Foreign Key referencing `tournament_key` in `tournaments`).
 - `surface` .
 - `draw_size` .
 - `tournament_date` .
 - `winner_id` (Foreign Key referencing `player_id` in `players`).
 - `loser_id` (Foreign Key referencing `player_id` in `players`).
 - `score` .
 - `best_of` .
 - `round` .
- `tournaments` :
 - `tournament_key` (Primary Key).
 - `tournament_name` .

Phụ thuộc hàm:

- `players`:

$\{ \text{player_id} \} \rightarrow \{ \text{first_name}, \text{last_name}, \text{hand}, \text{date_of_birth}, \text{nationality}, \text{height} \}.$

- tournaments:

$\{ \text{tournament_key} \} \rightarrow \{ \text{tournament_name} \}.$

b. Các quan hệ trong 3NF:

- players:

- $R1(\text{players}) = (\text{player_id}, \text{first_name}, \text{last_name}, \text{hand}, \text{date_of_birth}, \text{nationality}, \text{height}).$

- Phụ thuộc hàm: $\{ \text{player_id} \} \rightarrow \{ \text{first_name}, \text{last_name}, \text{hand}, \text{date_of_birth}, \text{nationality}, \text{height} \}.$

- single_ranking:

- $R2(\text{single_ranking}) = (\text{ranking_date}, \text{ranking}, \text{player}, \text{points}).$

- matches_94_04_14_24:

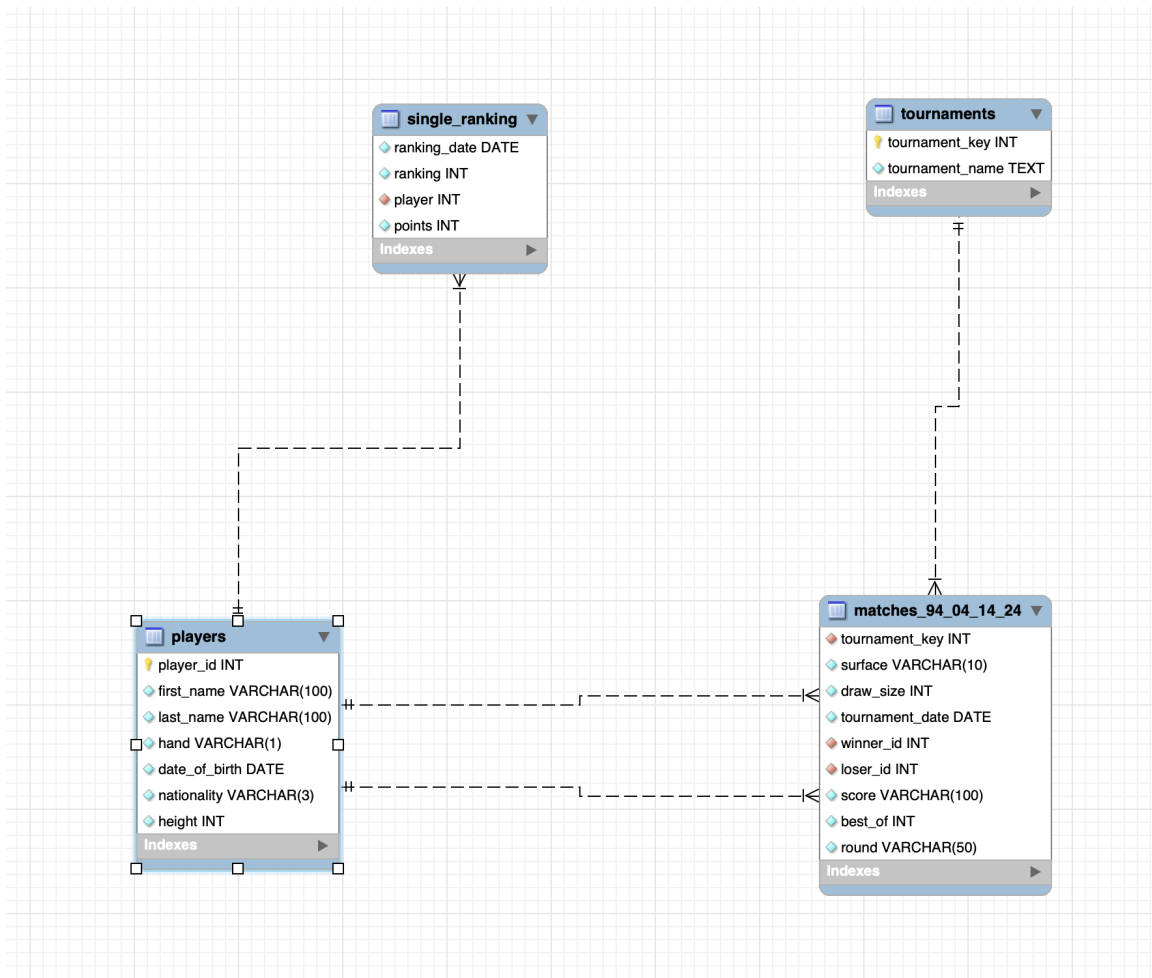
- $R3(\text{matches_94_04_14_24}) = (\text{tournament_key}, \text{surface}, \text{draw_size}, \text{tournament_date}, \text{winner_id}, \text{loser_id}, \text{score}, \text{best_of}, \text{round}).$

- tournaments:

- $R4(\text{tournaments}) = (\text{tournament_key}, \text{tournament_name}).$

- Phụ thuộc hàm: $\{ \text{tournament_key} \} \rightarrow \{ \text{tournament_name} \}.$

c. Sơ đồ quan hệ:



4. Thêm data vào hệ thống:

Hệ thống cập nhật dữ liệu của hơn 2800 tay vợt lớn nhỏ từ các thể hệ khác nhau. Các trận đấu được cập nhật là các trận đấu quần vợt diễn ra trong năm 1994, 2004, 2014 và từ ngày 01/01/2024 đến hết ngày 27/05/2024. Thứ hạng được cập nhật là thứ hạng của các tay vợt Top 200 Thế giới được ghi nhận vào ngày 19/12/1994, 27/12/2004, 29/12/2014 và 27/05/2024.

5. Thực hiện các truy vấn:

Người thực hiện: Ngô Nguyễn Khải Hưng.

a. Inner Join:

```

SELECT p.first_name, p.last_name, s.ranking_date, s.ranking
FROM players p
INNER JOIN single_ranking s ON p.player_id = s.player
WHERE s.ranking <= 10;
  
```

- Giải thích: Câu truy vấn này thực hiện phép **INNER JOIN** giữa hai bảng `players` và `single_ranking`. Nó lấy ra thông tin về tên và họ của các tay vợt, cùng với ngày và thứ hạng của họ trong bảng xếp hạng. Chỉ những tay vợt có thứ hạng trong top 10 được chọn (thứ hạng nhỏ hơn hoặc bằng 10).

b. Outer join:

```
SELECT p.player_id, p.first_name, p.last_name, s.ranking_date, s.ranking
FROM players p
LEFT OUTER JOIN single_ranking s ON p.player_id = s.player
ORDER BY p.player_id;
```

- Giải thích: Câu truy vấn sử dụng **LEFT OUTER JOIN** để kết nối bảng `players` với bảng `single_ranking`. Dữ liệu từ bảng `players` sẽ được lấy đầy đủ, ngay cả khi không có thông tin tương ứng từ bảng `single_ranking` (đối với những tay vợt chưa có thứ hạng). Câu truy vấn này cũng sắp xếp kết quả theo `player_id`.

c. Truy vấn với where:

```
SELECT first_name, last_name, height
FROM players
WHERE height > (SELECT AVG(height) FROM players);
```

- Giải thích: Trong câu truy vấn này, **subquery** được sử dụng trong điều kiện `WHERE` để lấy ra danh sách các tay vợt có chiều cao lớn hơn chiều cao trung bình của tất cả các tay vợt. Subquery `(SELECT AVG(height) FROM players)` tính chiều cao trung bình của tất cả các tay vợt trong bảng `players`.

d. Truy vấn với from:

```
SELECT nationality, AVG(avg_height) AS average_height_by_nationality
FROM (
    SELECT nationality, AVG(height) AS avg_height
    FROM players
    GROUP BY nationality
) subquery
GROUP BY nationality;
```

- Giải thích: Câu truy vấn sử dụng **subquery trong FROM** để tính chiều cao trung bình của tay vợt theo quốc tịch. Subquery trong nhóm các tay vợt theo `nationality` và tính chiều cao trung bình của từng nhóm. Sau đó, truy vấn bên ngoài tính lại chiều cao trung bình của từng nhóm quốc tịch từ kết quả của subquery.

e. Sử dụng group by và hàm:

```
select nationality, COUNT(player_id) AS num_players, AVG(height) AS avg_height
from players
group by nationality
having avg_height > 183;
```

- Giải thích: Câu truy vấn này sử dụng **GROUP BY** để nhóm các tay vợt theo quốc tịch và tính toán số lượng tay vợt (`COUNT(player_id)`) và chiều cao trung bình (`AVG(height)`) cho mỗi nhóm. Điều kiện **HAVING** được sử dụng để chỉ chọn những quốc tịch có chiều cao trung bình lớn hơn 183.

6. Thực hiện transaction có tích hợp lệnh rollback:

- Người thực hiện: Đoàn Quang Huy.
- Khái niệm Transaction: Transaction là một **đơn vị công việc** hoặc một nhóm các thao tác **được thực hiện như một thể thống nhất**. Transaction đảm bảo rằng **dữ liệu trong cơ sở dữ liệu** luôn ở trạng thái **nhất quán** và **an toàn**.
- Khái niệm Rollback: **Rollback** là **quá trình hoàn tác (undo)** các thao tác trong một transaction khi có lỗi xảy ra hoặc transaction không thể hoàn thành.

```
START TRANSACTION; # initialize
# Set the hand of right-handed players whose first name is Carlos as left hand
UPDATE players SET hand = 'L' where first_name = 'Carlos' and hand = 'R';
ROLLBACK; # cancel all the changes
```

- Giải thích: Transaction thực hiện sửa dữ liệu tay thuận của các tay vợt tên là "Carlos" thuận tay trái sang tay phải. Sau đó sử dụng lệnh Rollback để đảo ngược lại toàn bộ quá trình Transaction và trả lại hiện trạng ban đầu cho dữ liệu.

6. Thực hiện trigger:

Người thực hiện: Nguyễn Đức Huy.

▼ Khái niệm về trigger:

Trigger là một **thành phần tự động** trong cơ sở dữ liệu, được kích hoạt khi xảy ra một hành động cụ thể như **INSERT**, **UPDATE**, hoặc **DELETE** trên một bảng. Chúng được thiết kế để thực hiện một loạt các thao tác tự động nhằm đảm bảo dữ liệu được xử lý đúng và nhất quán.

▼ Trigger được tạo trong cơ sở dữ liệu:

1. Trigger after_player_insert dùng cho bảng player

a. Mục đích

Trigger này được tạo ra để ghi nhật lại các thao tác **thêm mới** (INSERT) vào bảng `players`

b. Chi tiết hoạt động

- Các thông tin ghi nhận gồm:
 - `player_id` : Mã người chơi.
 - `first_name` và `last_name` : Họ và tên.
 - `action` : Ghi nhận là "INSERT".
 - `action_time` : Thời gian thêm mới.
- Trigger tự động thêm một dòng mới vào bảng `player_audit` mỗi khi có bản ghi được chèn vào bảng `players`.

c. Ý nghĩa của trigger:

- Theo dõi lịch sử** thêm người chơi mới.
- Hỗ trợ truy vết dữ liệu** khi cần kiểm tra hoặc phân tích các thao tác trong bảng `players`.

d. Cách kiểm tra:

- Chèn dữ liệu mới** vào bảng `players` :

```
INSERT INTO players (player_id, first_name, last_name, hand, date_of_birth, nationality, height)
VALUES (3, 'Novak', 'Djokovic', 'R', '1987-05-22', 'SRB', 188);
```

```
INSERT INTO players (player_id, first_name, last_name, hand, date_of_birth, nationality, height)
VALUES (4, 'Novak', 'Djokovic', 'L', '1984-05-22', 'SRB', 188);
```

```
SELECT * FROM player_audit;
```

- Kiểm tra bảng** `player_audit` :

Dữ liệu được lưu lại sẽ bao gồm ID, tên người chơi, hành động (INSERT) và thời gian.

2. Trigger after_ranking_update

a. Mục đích

Theo dõi và ghi nhật thay đổi thứ hạng trong bảng

`single_ranking`.

b. Chi tiết hoạt động

- Trigger kích hoạt sau khi bảng `single_ranking` có thao tác **UPDATE**.
- Trigger sẽ kiểm tra xem cột `ranking` trước và sau khi cập nhật có thay đổi hay không:
 - Nếu **không thay đổi**, trigger không thực hiện gì.
 - Nếu **có thay đổi**, trigger thêm một dòng vào bảng `ranking_history` gồm:
 - `ranking_date` : Ngày thay đổi.
 - `player` : Mã người chơi.
 - `old_ranking` : Thứ hạng cũ.
 - `new_ranking` : Thứ hạng mới.
 - `points` : Điểm của người chơi.

c. Ý nghĩa của trigger này

- **Lưu trữ lịch sử** thay đổi thứ hạng người chơi qua thời gian.
- **Phân tích xu hướng**, giúp theo dõi phong độ thi đấu của người chơi.
- **Hỗ trợ báo cáo**, nhật ký lại thông tin quan trọng cho nhóm phân tích hoặc quản trị.

d. Cách kiểm tra trigger:

- **Cập nhật dữ liệu để kích hoạt trigger:**

```

1 • UPDATE single_ranking
2   SET ranking = 2, points = 1200
3   WHERE player = 1;
4
5 • SELECT * FROM ranking_history;
6

```

- **Kiểm tra bảng** `ranking_history` :

Trigger sẽ tự động thêm bản ghi lưu thông tin thay đổi thứ hạng.

7. Thực hiện procedure:

Người thực hiện: Tạ Quang Linh

▼ Khái niệm về Stored Procedure:

Stored Procedure (Thủ tục lưu trữ) là một tập hợp các câu lệnh SQL được lưu trữ dưới dạng một chương trình trong cơ sở dữ liệu. Thủ tục giúp thực hiện các thao tác phức tạp, tái sử dụng mã lệnh, và tối ưu hóa hiệu suất của các truy vấn.

▼ Stored Procedures được tạo trong cơ sở dữ liệu:

Procedure: getPlayerByNation

1. Mục đích

- Thủ tục này được tạo ra để lấy thông tin người chơi dựa trên quốc tịch.

2. Chi tiết hoạt động

- Thủ tục nhận một tham số đầu vào `nationName` là kiểu `VARCHAR(3)`, đại diện cho quốc tịch của người chơi.
- Thực hiện truy vấn để lấy tất cả các thông tin từ bảng `players` mà cột `nationality` bằng với `nationName`.

3. Ý nghĩa của thủ tục

- **Tối ưu hóa truy vấn:** Thủ tục giúp tối ưu hóa và đơn giản hóa việc truy vấn thông tin người chơi theo quốc tịch.
- **Tăng cường bảo mật:** Thủ tục giúp kiểm soát và bảo mật các thao tác truy vấn, tránh lỗi và đảm bảo dữ liệu chính xác.
- **Tái sử dụng mã lệnh:** Giảm thiểu việc viết lại mã lệnh, giúp tiết kiệm thời gian và công sức của nhà phát triển.

4. Cách kiểm tra

- Gọi thủ tục với một giá trị cụ thể cho `nationName` :

```
26  
27  
28 • CALL getPlayerByNation('USA');  
29  
30
```

Kết quả sẽ trả về tất cả các dòng trong bảng `players` có quốc tịch là 'USA'.

Procedure: getPlayerByHand

1. Mục đích

- Thủ tục này được tạo ra để lấy thông tin người chơi dựa trên tay thuận.

2. Chi tiết hoạt động

- Thủ tục nhận một tham số đầu vào `thisHand` là kiểu `VARCHAR(1)`, đại diện cho tay thuận của người chơi (ví dụ: 'L' cho tay trái, 'R' cho tay phải).
- Thực hiện truy vấn để lấy tất cả các thông tin từ bảng `players` mà cột `hand` bằng với `thisHand`.

3. Ý nghĩa của thủ tục

- **Tối ưu hóa truy vấn:** Thủ tục giúp tối ưu hóa và đơn giản hóa việc truy vấn thông tin người chơi theo tay thuận.
- **Tăng cường bảo mật:** Thủ tục giúp kiểm soát và bảo mật các thao tác truy vấn, tránh lỗi và đảm bảo dữ liệu chính xác.
- **Tái sử dụng mã lệnh:** Giảm thiểu việc viết lại mã lệnh, giúp tiết kiệm thời gian và công sức của nhà phát triển.

4. Cách kiểm tra

- Gọi thủ tục với một giá trị cụ thể cho `thisHand` :

```
26  
27  
28 • CALL getPlayerByHand('R');  
29 |  
30
```

Kết quả sẽ trả về tất cả các dòng trong bảng
`players` có tay thuận là 'R' (tay phải).