

**10 bài tập tổng hợp mức Expert** về chuẩn hóa 1NF, 2NF, 3NF trong thiết kế CSDL SQL, yêu cầu phân tích sâu về phụ thuộc hàm, khóa chính và các vấn đề phức tạp trong thực tế.

---

## Bài 1: Hệ thống đặt phòng khách sạn

Bảng HotelBookings: (booking\_id, guest\_id, guest\_phone, room\_id, room\_type, room\_price, check\_in, check\_out, hotel\_id, hotel\_city)

- **Khóa chính:** (booking\_id, room\_id, check\_in).
- **Phụ thuộc hàm:**
  - room\_id → room\_type, room\_price, hotel\_id
  - hotel\_id → hotel\_city
  - guest\_id → guest\_phone

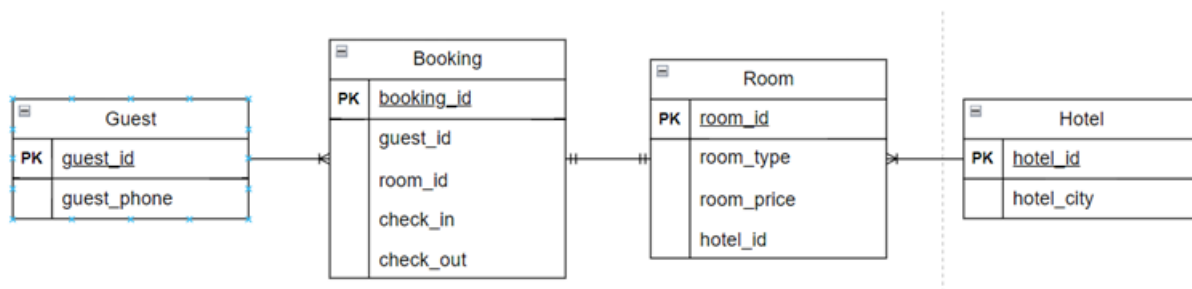
**Yêu cầu:**

1. Chỉ ra vi phạm 1NF (nếu có).
2. Chuẩn hóa lần lượt đến 3NF, giải thích từng bước.

Bảng HotelBookings không vi phạm chuẩn 1NF, vì tất cả các thuộc tính đều có giá trị nguyên tử, không chứa tập giá trị lặp hoặc đa trị. Do đó, bảng này đã đáp ứng các yêu cầu của chuẩn 1NF.

Tiếp theo, khi kiểm tra chuẩn 2NF, ta nhận thấy rằng một số thuộc tính không phụ thuộc đầy đủ vào toàn bộ khóa chính, mà chỉ phụ thuộc vào một phần của khóa chính tổng hợp (booking\_id, room\_id, check\_in). Đây là dấu hiệu của phụ thuộc hàm từng phần, vi phạm chuẩn 2NF.

Vì vậy, cần phân tách bảng thành các quan hệ mới, loại bỏ các phụ thuộc từng phần để đảm bảo cấu trúc dữ liệu đạt chuẩn 2NF, đồng thời vẫn bảo toàn đầy đủ thông tin ban đầu. Đây là kết quả:



Tiếp theo, tiến hành đánh giá mức độ thỏa mãn chuẩn 3NF trên thiết kế đã chuẩn hóa trước đó. Để đạt chuẩn 3NF, ngoài việc bảng phải đạt chuẩn 2NF, mọi thuộc tính không khóa phải phụ thuộc trực tiếp vào khóa chính, tức là không tồn tại phụ thuộc bắc cầu thông qua các thuộc tính không khóa.

Dựa trên sơ đồ thực thể–liên kết (ERD) đã thiết kế ở bước chuẩn hóa 2NF, ta nhận thấy không tồn tại bất kỳ phụ thuộc bắc cầu nào giữa các thuộc tính không khóa. Các thuộc tính đều phụ thuộc trực tiếp và duy nhất vào khóa chính của từng bảng, không thông qua trung gian. Do đó, thiết kế cơ sở dữ liệu hiện tại đã thỏa mãn các yêu cầu của chuẩn 3NF.

---

## Bài 2: Quản lý đơn hàng với thuộc tính đa giá trị

Bảng Orders: (order\_id, customer\_id, items, order\_date)

- items là JSON chứa danh sách sản phẩm:

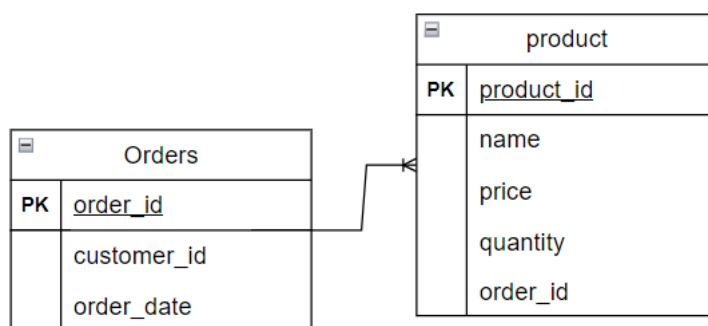
```
[{"product_id": 1, "name": "Laptop", "price": 1000, "quantity": 2}, ...]
```

### Yêu cầu:

1. Đưa về 1NF bằng cách khử thuộc tính đa giá trị.
2. Phân tích phụ thuộc hàm và áp dụng 2NF, 3NF.

Bảng ban đầu **Orders**(order\_id, customer\_id, items, order\_date) vi phạm chuẩn 1NF do thuộc tính **items** chứa dữ liệu đa giá trị được biểu diễn dưới dạng JSON mảng. Cụ thể, **items** chứa nhiều sản phẩm, mỗi sản phẩm bao gồm các thuộc tính: **product\_id**, **name**, **price**, và **quantity**. Điều này vi phạm yêu cầu của chuẩn 1NF, trong đó mọi thuộc tính phải có giá trị nguyên tử (atomic) – tức là mỗi ô trong bảng phải chứa một giá trị duy nhất.

Để đưa bảng này về chuẩn 1NF, ta tách các sản phẩm trong **items** thành từng dòng riêng biệt, mỗi dòng chỉ chứa một sản phẩm trong đơn hàng. Đây là kết quả:



Tiếp theo, để một quan hệ đạt chuẩn 2NF, trước tiên nó phải thỏa mãn 1NF (đã làm ở bước trước), và mọi thuộc tính không khóa phải phụ thuộc đầy đủ vào toàn bộ khóa chính, không được phụ thuộc vào một phần khóa. Thiết kế trên đã thỏa mãn điều đó.

Cuối cùng, ta sẽ xem xét liệu thiết kế có đạt chuẩn 3NF hay không. Rõ ràng tất cả các thuộc tính ở các bảng đều phụ thuộc trực tiếp vào khóa chính, và không có thuộc tính nào phụ thuộc

bắc cầu thông qua thuộc tính không khóa khác. Do đó, thiết kế hiện tại đã đạt chuẩn 3NF mà không cần phải tách bảng sản phẩm riêng biệt.

---

### Bài 3: Hệ thống giáo dục (quan hệ phức tạp)

Bảng StudentCourses: (student\_id, student\_name, department\_id, department\_head, course\_id, course\_name, instructor\_id, instructor\_email, grade)

- **Khóa chính:** (student\_id, course\_id).
- **Phụ thuộc hàm:**
  - student\_id → student\_name, department\_id
  - department\_id → department\_head
  - course\_id → course\_name, instructor\_id
  - instructor\_id → instructor\_email

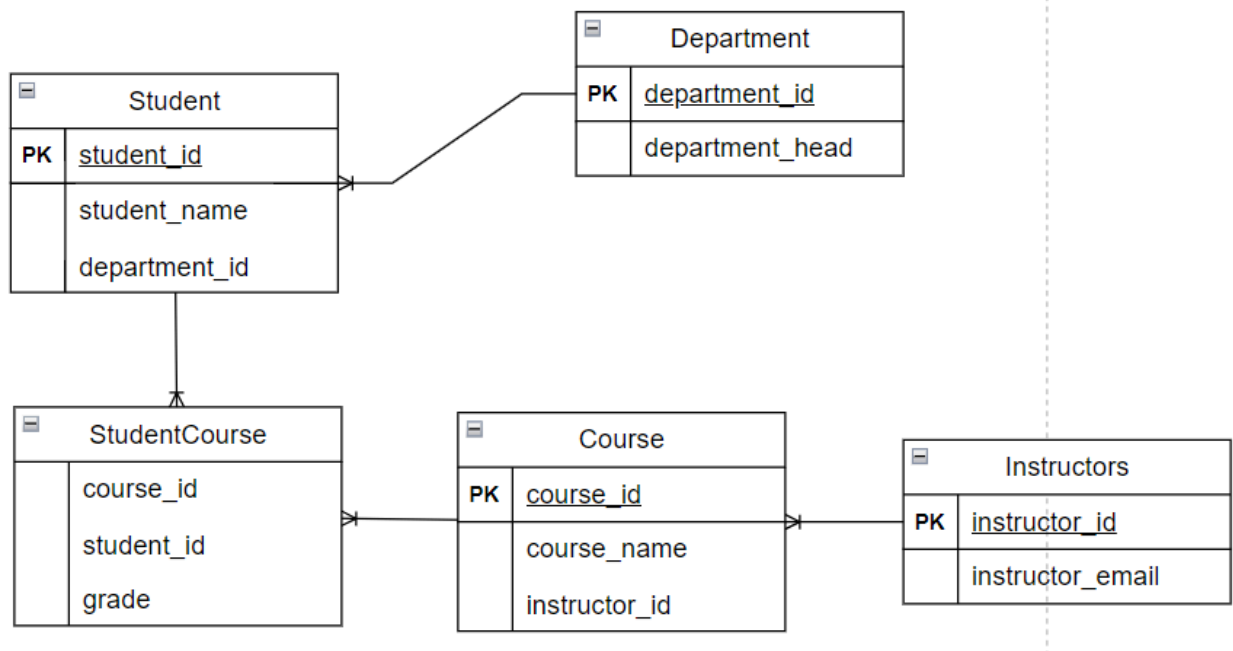
**Yêu cầu:**

1. Chứng minh vi phạm 2NF/3NF.
2. Thiết kế lại đạt 3NF.

Trong bảng **StudentCourses**, khóa chính là tổ hợp (student\_id, course\_id). Tuy nhiên, nhiều thuộc tính chỉ phụ thuộc vào một phần của khóa chính, dẫn đến vi phạm chuẩn 2NF. Cụ thể, các thuộc tính như student\_name và department\_id chỉ phụ thuộc vào student\_id, trong khi course\_name và instructor\_id chỉ phụ thuộc vào course\_id. Điều này chứng tỏ các thuộc tính không phụ thuộc vào toàn bộ khóa chính, mà chỉ vào một phần khóa, vi phạm nguyên tắc của 2NF, vốn yêu cầu tất cả các thuộc tính không khóa phải phụ thuộc toàn bộ vào khóa chính.

Tiếp theo, xét đến chuẩn 3NF, ta phát hiện các phụ thuộc bắc cầu trong bảng. Ví dụ, department\_head không phụ thuộc trực tiếp vào khóa chính mà phụ thuộc gián tiếp thông qua department\_id (tức là student\_id → department\_id → department\_head). Tương tự, instructor\_email phụ thuộc gián tiếp vào course\_id thông qua instructor\_id (tức là course\_id → instructor\_id → instructor\_email). Do các thuộc tính này không phụ thuộc trực tiếp vào toàn bộ khóa chính mà thông qua một thuộc tính trung gian không phải khóa chính, nên bảng tiếp tục vi phạm chuẩn 3NF.

Dưới đây là thiết kế lại đạt 3NF:



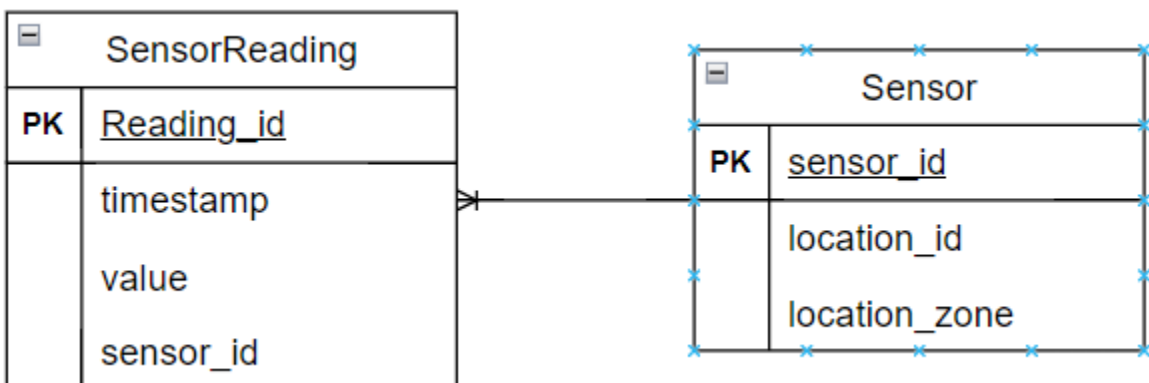
#### Bài 4: Hệ thống IoT (dữ liệu phi cấu trúc)

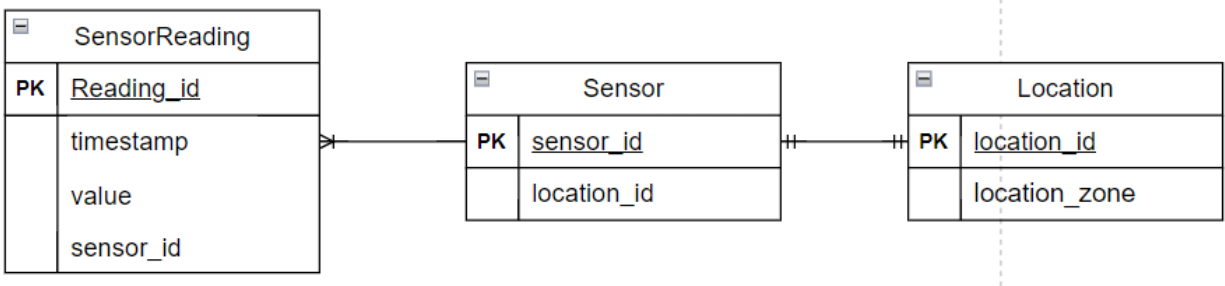
Bảng SensorReadings: (sensor\_id, timestamps, values, location\_id, location\_zone)

- timestamps: Chuỗi phân cách bằng dấu phẩy (ví dụ: "2023-01-01 12:00, 2023-01-01 12:05").
- values: Chuỗi giá trị tương ứng (ví dụ: "23.5, 24.1").

**Yêu cầu:**

- Áp dụng 1NF bằng cách tách timestamps và values.
- Giả sử location\_id → location\_zone, chuẩn hóa tiếp đến 3NF.





## Bài 5: Quản lý dự án với phụ thuộc bắc cầu

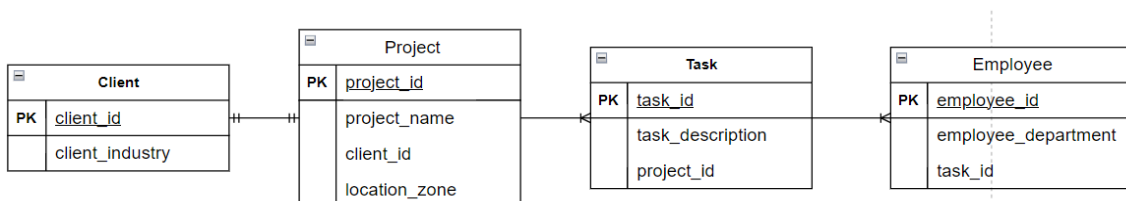
Bảng ProjectTasks: (project\_id, project\_name, client\_id, client\_industry, task\_id, task\_description, employee\_id, employee\_department)

- **Khóa chính:** (project\_id, task\_id).
- **Phụ thuộc hàm:**
  - project\_id → project\_name, client\_id
  - client\_id → client\_industry
  - task\_id → task\_description
  - employee\_id → employee\_department

**Yêu cầu:**

1. Chỉ ra phụ thuộc bắc cầu vi phạm 3NF.
2. Chuẩn hóa triệt để.

Trong bảng **ProjectTasks**, tồn tại các phụ thuộc bắc cầu (transitive dependencies) như sau: đầu tiên, ta có **project\_id** → **client\_id** và tiếp theo **client\_id** → **client\_industry**, suy ra **project\_id** → **client\_industry**. Điều này vi phạm 3NF bởi vì **client\_industry** không phụ thuộc trực tiếp vào khóa chính (**project\_id**, **task\_id**) mà thông qua **client\_id**. Tương tự, **task\_id** → **task\_description** và **employee\_id** → **employee\_department** cũng là các phụ thuộc từ một thuộc tính không phải là toàn bộ khóa chính hoặc không phải khóa chính (**task\_id** và **employee\_id**), nên việc giữ các cột này trong cùng bảng **ProjectTasks** cũng vi phạm 3NF. Mục tiêu của 3NF là đảm bảo mọi thuộc tính không khóa chỉ phụ thuộc trực tiếp vào toàn bộ khóa chính và không qua trung gian bất kỳ thuộc tính không khóa nào khác.



## Bài 6: Cửa hàng bán lẻ (khóa phức hợp)

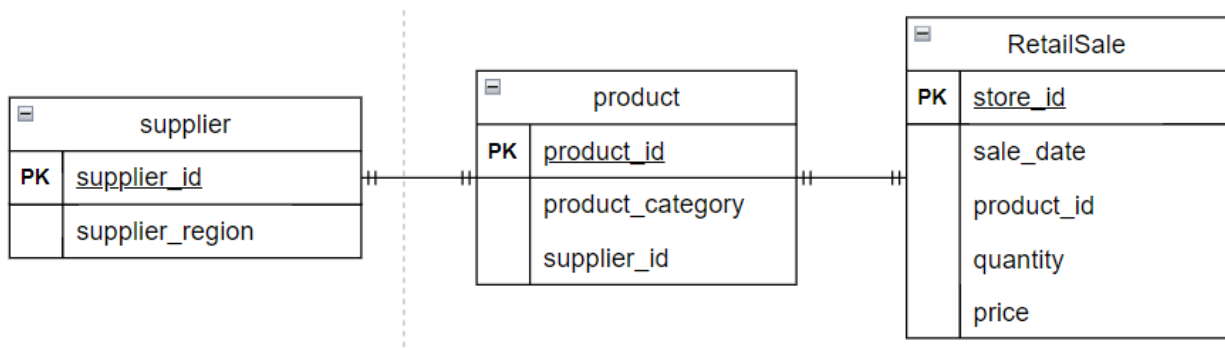
Bảng RetailSales: (store\_id, product\_id, product\_category, supplier\_id, supplier\_region, sale\_date, quantity, price)

- **Khóa chính:** (store\_id, product\_id, sale\_date).
- **Phụ thuộc hàm:**
  - product\_id → product\_category, supplier\_id
  - supplier\_id → supplier\_region

**Yêu cầu:**

1. Tại sao bảng này chưa đạt 2NF?
2. Đưa về 3NF và vẽ sơ đồ quan hệ.

Bảng **RetailSales** chưa đạt 2NF vì trong đó tồn tại các phụ thuộc hàm mà một thuộc tính không phải khóa (**product\_category, supplier\_id, supplier\_region**) lại chỉ phụ thuộc vào một phần của khóa phức hợp (**store\_id, product\_id, sale\_date**) hoặc không phụ thuộc vào khóa chính. Cụ thể, **product\_id → product\_category, supplier\_id** và **supplier\_id → supplier\_region**, thay vì phụ thuộc vào toàn bộ ba thành phần **store\_id, product\_id** và **sale\_date**. Điều này vi phạm nguyên tắc của 2NF, theo đó mọi thuộc tính không khóa phải phụ thuộc đầy đủ vào toàn bộ khóa chính.



## Bài 7: Hệ thống y tế (dữ liệu lồng nhau)

Bảng PatientRecords: (patient\_id, patient\_name, visits)

- visits là XML:

<visits>

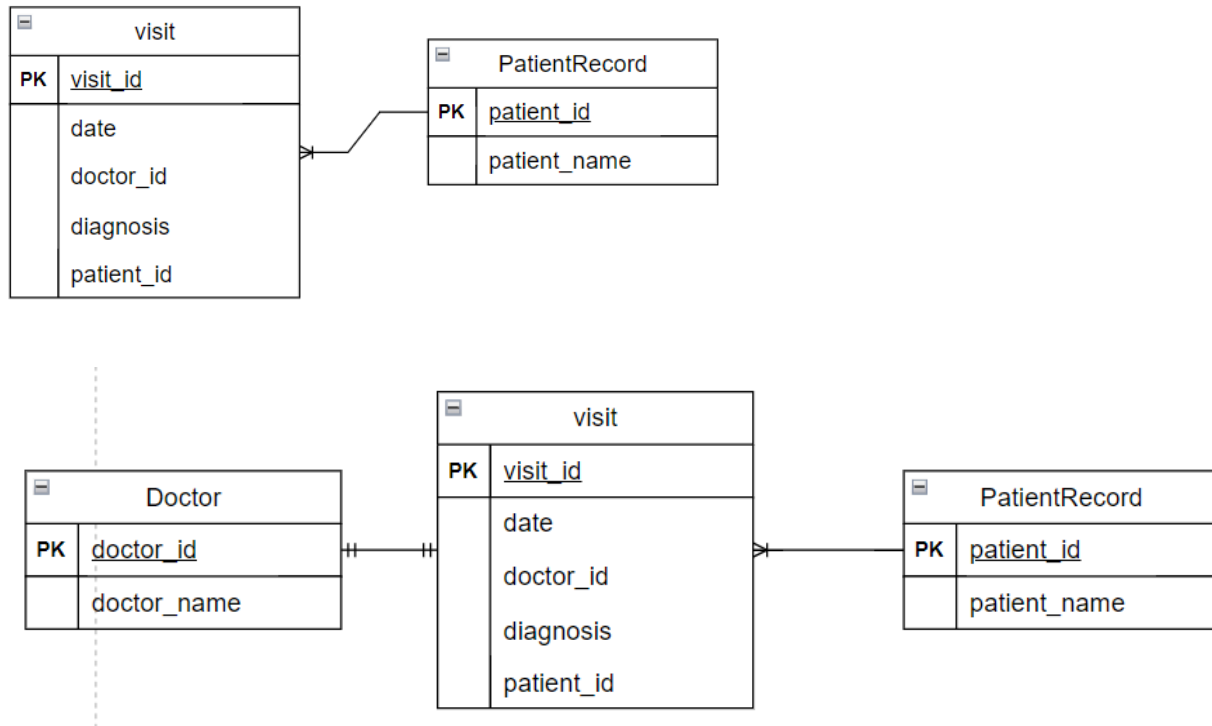
<visit date="2023-01-01" doctor\_id="D1" diagnosis="Fever"/>

<visit date="2023-02-01" doctor\_id="D2" diagnosis="Flu"/>

</visits>

### Yêu cầu:

1. Khử 1NF bằng cách "làm phẳng" XML.
2. Giả sử  $\text{doctor\_id} \rightarrow \text{doctor\_name}$ , áp dụng 3NF.



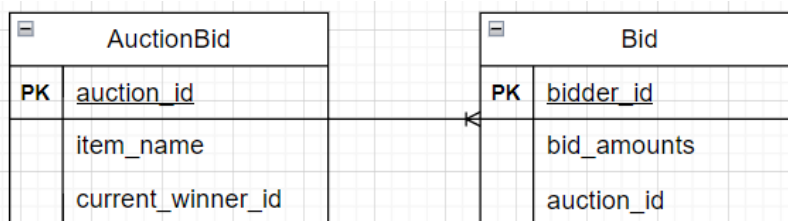
## Bài 8: Hệ thống đấu giá (phụ thuộc đa trị)

Bảng AuctionBids: (auction\_id, item\_name, bidder\_ids, bid\_amounts, current\_winner\_id)

- bidder\_ids và bid\_amounts là mảng (ví dụ: "B001,B002", "100,150").

### Yêu cầu:

1. Thiết kế lại để đạt 1NF.
2. Phân tích phụ thuộc hàm để đạt 3NF.



Sau khi thiết kế lại bảng dữ liệu để đạt chuẩn 1NF, tất cả các thuộc tính đã đảm bảo tính nguyên tử, loại bỏ hoàn toàn các giá trị lặp và đa trị. Tiếp theo, trong quá trình chuẩn hóa đến chuẩn 3NF, ta xem xét các phụ thuộc hàm giữa các thuộc tính. Theo định nghĩa của 3NF, mọi thuộc tính không thuộc khóa chính trong một quan hệ phải phụ thuộc hàm trực tiếp và đầy đủ vào khóa chính, đồng thời không được phụ thuộc bắc cầu thông qua một thuộc tính không khóa khác.

Thiết kế hiện tại đã đáp ứng đầy đủ yêu cầu này. Cụ thể, trong mỗi quan hệ được tách ra, các thuộc tính không khóa như `item_name`, `bid_amount` và `current_winner_id` đều phụ thuộc trực tiếp vào khóa chính của quan hệ tương ứng, và không tồn tại phụ thuộc bắc cầu nào. Do đó, thiết kế đã đạt chuẩn ba của chuẩn hóa (3NF), đảm bảo tính toàn vẹn dữ liệu và giảm thiểu tối đa sự dư thừa.

---

## Bài 9: Quản lý kho (thuộc tính tính toán)

Bảng Inventory: (product\_id, product\_name, warehouse\_id, warehouse\_location, current\_stock, reorder\_level, last\_restock\_date)

- **Khóa chính:** (product\_id, warehouse\_id).
- **Phụ thuộc hàm:**
  - product\_id → product\_name
  - warehouse\_id → warehouse\_location
  - reorder\_level phụ thuộc vào product\_id và warehouse\_id.

**Yêu cầu:**

1. Giải thích vi phạm 2NF/3NF.
2. Chuẩn hóa và đưa reorder\_level vào bảng phù hợp.

### Vi phạm chuẩn 2NF:

Theo định nghĩa, chuẩn 2NF yêu cầu mọi thuộc tính không khóa trong quan hệ phải phụ thuộc hàm hoàn toàn vào toàn bộ khóa chính, không được phụ thuộc vào một phần khóa. Tuy nhiên, trong quan hệ hiện tại:

- Thuộc tính `product_name` chỉ phụ thuộc vào `product_id`, không phụ thuộc vào toàn bộ khóa (`product_id`, `warehouse_id`).
- Tương tự, `warehouse_location` chỉ phụ thuộc vào `warehouse_id`.

Điều này dẫn đến vi phạm chuẩn 2NF do tồn tại các phụ thuộc vào một phần khóa chính.

### Vi phạm chuẩn 3NF:

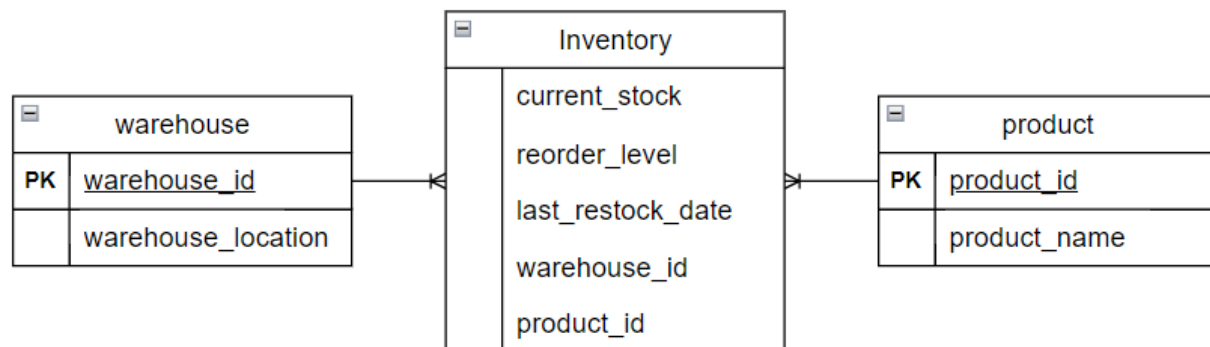
Sau khi loại bỏ các vi phạm chuẩn 2NF bằng cách tách các thuộc tính phụ thuộc vào một phần khóa thành các quan hệ riêng biệt, cần tiếp tục kiểm tra chuẩn 3NF. Theo chuẩn này, mọi thuộc



tính không khóa trong một quan hệ phải phụ thuộc hàm trực tiếp vào khóa chính và không được phụ thuộc bắc cầu thông qua một thuộc tính không khóa khác.

Trong trường hợp này, nếu còn tồn tại các thuộc tính phụ thuộc gián tiếp vào khóa chính thông qua một thuộc tính trung gian không khóa, thì quan hệ vẫn vi phạm chuẩn 3NF.

Do đó, để đảm bảo quan hệ đạt chuẩn 3NF, ta cần tách các thuộc tính có phụ thuộc bắc cầu sang các quan hệ riêng biệt, đảm bảo tất cả các thuộc tính không khóa đều phụ thuộc trực tiếp và duy nhất vào khóa chính.



---

## Bài 10: Mạng xã hội (quan hệ đệ quy)

Bảng UserPosts: (post\_id, user\_id, user\_name, post\_content, parent\_post\_id, parent\_post\_user)

- **Khóa chính:** post\_id.
- **Phụ thuộc hàm:**
  - user\_id → user\_name
  - parent\_post\_id → parent\_post\_user

**Yêu cầu:**

1. Phân tích phụ thuộc hàm và chuẩn hóa đến 3NF.
2. Xử lý quan hệ đệ quy (self-referencing) thế nào?

Dựa trên cấu trúc dữ liệu và ngữ nghĩa của quan hệ `UserPosts(post_id, user_id, user_name, post_content, parent_post_id, parent_post_user)`, ta xác định được các phụ thuộc hàm như sau:

- `user_id → user_name`: Mỗi người dùng có một tên duy nhất, do đó `user_name` phụ thuộc vào `user_id`.
- `parent_post_id → parent_post_user`: Mỗi bài viết cha được đăng bởi một người dùng xác định, nên `parent_post_user` phụ thuộc vào `parent_post_id`.

- $post\_id \rightarrow user\_id, post\_content, parent\_post\_id$ : Vì  $post\_id$  là khóa chính của quan hệ, nó xác định đầy đủ thông tin về bài viết, bao gồm người đăng, nội dung và bài viết cha (nếu có).

Tuy nhiên, cần lưu ý rằng các thuộc tính  $user\_name$  và  $parent\_post\_user$  không phụ thuộc trực tiếp vào khóa chính  $post\_id$ , mà phụ thuộc gián tiếp thông qua các thuộc tính trung gian  $user\_id$  và  $parent\_post\_id$ . Do đó, các phụ thuộc này mang tính bắc cầu, làm cho quan hệ vi phạm chuẩn 3NF, bởi chuẩn này yêu cầu mọi thuộc tính không khóa phải phụ thuộc trực tiếp vào khóa chính và không thông qua thuộc tính không khóa khác.

