**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.

**GIẢI**

* Đạt chuẩn 1 vì không có cột đa trị và lặp lại
* Chưa đạt chuẩn 2 do có phụ thuộc 1 phần: room\_id → room\_type
* Chưa đạt chuẩn 3 do phụ thuộc bắc cầu: room\_id → hotel\_id → hotel\_city
* Tách bảng:
* Room(room\_id, room\_type, room\_price, hotel\_id)
* Hotel(hotel\_id, hotel\_city)
* Guest(guest\_id, guest\_phone)
* HotelBookings(booking\_id, guest\_id, room\_id, check\_in, check\_out)

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

Bảng Orders:

sql

Copy

(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.
3. **GIẢI**

* Chưa đạt chuẩn 1 vì có giá trị đa trị: items là JSON
* OrderItems(order\_id, product\_id, product\_name, price, quantity)
* Chưa đạt chuẩn 2 do phụ thuộc 1 phần:
* product\_id → product\_name, price
* order\_id → customer\_id, order\_date
* Tách để đạt chuẩn 3:
* Product(product\_id, product\_name, price)
* Orders(order\_id, customer\_id, order\_date)
* OrderItems(order\_id, product\_id, quantity)

### **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.
3. **GIẢI**

* Có nhiều phụ thuộc không trực tiếp vào toàn bộ khóa như:
* student\_id → student\_name, department\_id
* department\_id → department\_head
* course\_id → course\_name, instructor\_id
* instructor\_id → instructor\_email
* Vi phạm 2NF/3NF

Student(student\_id, student\_name, department\_id)

Department(department\_id, department\_head)

Course(course\_id, course\_name, instructor\_id)

Instructor(instructor\_id, instructor\_email)

StudentCourses(student\_id, course\_id, grade)

### **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**:

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

* Readings(sensor\_id, timestamp, value, location\_id)
* location\_zone không phụ thuộc vào toàn bộ khóa chính (sensor\_id, timestamp) mà location\_zone phụ thuộc vào location\_id
* Location(location\_id, location\_zone)

### **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.
3. **GIẢI**

* Phụ thuộc bắc cầu: project\_id → client\_id → client\_industry
* project\_id → project\_name, client\_id
* client\_id → client\_industry
* task\_id → task\_description
* employee\_id → employee\_department
* Project(project\_id, project\_name, client\_id)
* Client(client\_id, client\_industry)
* Task(task\_id, task\_description)
* Employee(employee\_id, employee\_department)
* ProjectTasks(project\_id, task\_id, employee\_id)

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

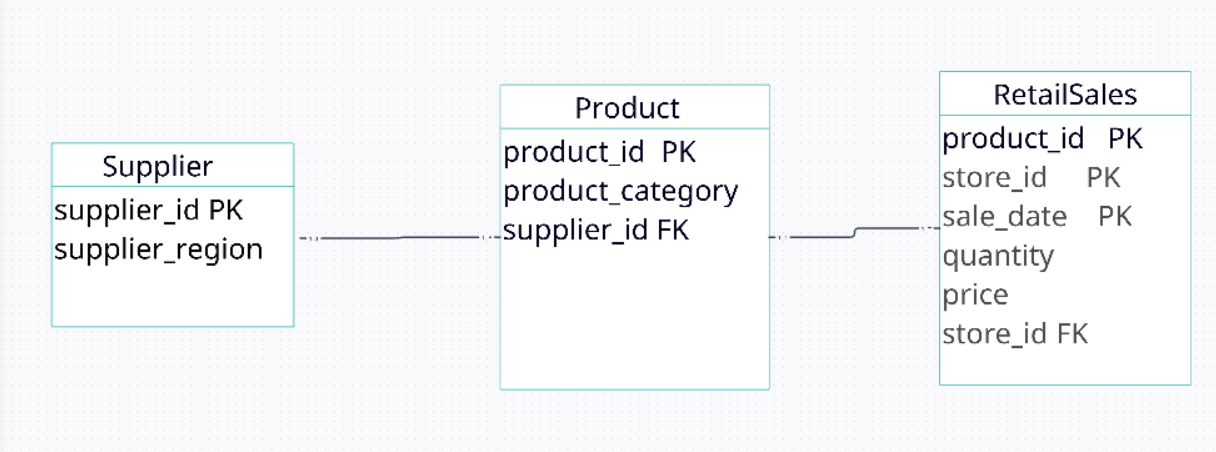
Bảng RetailSales:

(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.
3. **GIẢI**

* Chưa đạt chuẩn 2 vì:
* product\_id → product\_category, supplier\_id (phụ thuộc 1 phần)
* supplier\_id → supplier\_region (phụ thuộc bắc cầu)
* Product(product\_id, product\_category, supplier\_id)
* Supplier(supplier\_id, supplier\_region)
* RetailSales(store\_id, product\_id, sale\_date, quantity, price)



### **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>
* Run HTML

**Yêu cầu**:

1. Khử 1NF bằng cách "làm phẳng" XML.
2. Giả sử doctor\_id → doctor\_name, áp dụng 3NF.
3. **GIẢI**

* Visit(visit\_id, patient\_id, visit\_date, doctor\_id, diagnosis)
* Patient(patient\_id, patient\_name)
* Doctor(doctor\_id, doctor\_name)

### **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.
3. **GIẢI**

* Bids(auction\_id, bidder\_id, bid\_amount)
* Hàm phụ thuộc:
* auction\_id → item\_name (1 phiên đấu giá chỉ có bán 1 món đồ duy nhất)
* auction\_id → current\_winner\_id (1 phiên đấu giá chỉ có 1 người dẫn đầu)

Auction(auction\_id, item\_name, current\_winner\_id) -> thông tin phiên đấu giá

Bids(auction\_id, bidder\_id, bid\_amount) -> thông tin các lượt đấu giá

### **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.
3. **GIẢI**

* Chưa đạt chuẩn 2 do phụ thuộc 1 phần:
* product\_id → product\_name
* warehouse\_id → warehouse\_location
* chưa đạt chuẩn 3 do phụ thuộc bắc cầu: product\_id → product\_name và warehouse\_id → warehouse\_location phụ thuộc bắc cầu qua khóa phụ
* reorder\_level không phụ thuộc product\_id hay warehouse\_id. Mà nó phụ thuộc cả 2 nên nó phù hợp để trong bảng chính sau khi chuẩn hóa
* Product (product\_id, product\_name)
* Warehouse (warehouse\_id, warehouse\_location)
* Inventory (product\_id, warehouse\_id, current\_stock, reorder\_level, last\_restock\_date)

### **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?
3. **GIẢI**

* Đạt chuẩn 1
* Chưa đạt chuẩn 2 vì
* user\_name phụ thuộc vào user\_id, nhưng user\_id không phải là khóa chính mà là một phần trong khóa ngoài (khóa phụ).
* parent\_post\_user phụ thuộc vào parent\_post\_id, nhưng parent\_post\_id không phải là khóa chính mà chỉ là một khóa ngoài.

User (user\_id, user\_name)

Post (post\_id, user\_id, post\_content, parent\_post\_id)