**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:

sql

Copy

(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.

Trả lời:

Chuẩn hóa 2NF:

Các phụ thuộc một phần:

room\_id -> room\_type, room\_price, hotel\_id -> phụ thuộc một phần vào room\_id.

guest\_id -> guest\_phone -> phụ thuộc một phần vào guest\_id.

hotel\_id -> hotel\_city -> phụ thuộc một phần vào hotel\_id.

Tách bảng:

Room( room\_id(PK), room\_type, room\_price, hotel \_id(FK)) (Hotel-Room 1:n)

Guest(guest\_id(PK), guest\_phone)

Hotel(hotel\_id(PK), hotel\_city)

HotelBooking(booking\_id(PK), guest\_id(FK), room\_id(FK), check\_in, check\_out)

Guest-HotelBookings: 1:n

Room-HotelBookings 1:n

Chuẩn hóa 3NF:

Phụ thuộc bắc cầu:

Room\_id -> hotel\_id -> hotel\_city phụ thuộc bắc cầu room\_id -> hotel\_city (qua hotel\_id)

Tách bảng tương tự chuẩn 2NF

### **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:
* json
* Copy
* [{"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.

**Trả lời:**

Tách bảng:

Order(order\_id(PK), customer\_id, order\_date)

OrderItems( order\_id(FK), product\_id(PK), name, price, quantity)

Chuẩn hóa 2NF:

Phụ thuộc một phần:

Product\_id -> name, price

Bảng mới:

Orders(order\_id(PK), customer\_id, order\_date)

Products(product\_id(PK), name ,price)

OrderItems(order\_id(PK,FK), product\_id(PK,FK), quantity)

(Orders-OrderItems 1:n, Products-OrderItem 1:n)

Vì chưa có thêm phụ thuộc nên dữ liệu đã đạt 3NF

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

Bảng StudentCourses:

sql

Copy

(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.

**Trả lời:**

Thiết kế lại đạt 3NF

Students(student\_id (PK), student\_name, department\_id (FK)) (Department-Student 1:n)

Department(department\_id (PK), department\_head)

Course (course\_id (PK), course\_name, instructor\_id (FK)) Instructor-Course(1:n)

Instructors( instructor\_id (PK), instructor\_email)

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

Student-StudentCourses (1:n)

Course-Studentcourses(1:n)

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

Bảng SensorReadings:

sql

Copy

(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.

**Trả lời:**

Tách mỗi cặp timestamp và value thành 1 dòng riêng biệt mỗi bản ghi chỉ còn 1 giá trị duy nhất cho mỗi thuộc tính

Chuẩn hóa 3NF

SensorReadings( sensor\_id(PK), timestamp, value, location\_id(FK))

Locations (location\_id (PK), location\_zone)

SensorReadings-Location (n:1)

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

Bảng ProjectTasks:

sql

Copy

(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.

**Trả lời:**

Project\_id -> project\_name, client\_id

Client\_id -> client\_industry phụ thuộc bắc cầu -> project\_id -> client\_id ->client\_id ->client\_industry

Chuẩn hóa 3NF

Projects( project\_id(PK), project\_name, client\_id(FK) )

Clients (client\_id(PK), client\_industry)

Tasks (task\_id(PK), task\_description)

Employees(employee\_id(PK), emloyee\_department)

ProjectTasks (project\_id(PK,FK), task\_id(PK, FK), employee\_id(FK))

Projects-Clients n:1

ProjectTasks-Projects n:1

ProjectTasks-Tasks n:1

ProjectTasks-Employees n:1

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

Bảng RetailSales:

sql

Copy

(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.

**Trả lời:**

Product\_id -> product\_catagory, suppliers\_id phụ thuộc 1 phần vào khóa chính

Tách bảng

Products (product\_id(PK), product\_category, supplier\_id(FK)

Suppliers( supplier\_id(PK), supplier\_region)

RetailSales (store\_id(PK), product\_id ( FK), sale\_date (FK), quantity, price)

RetailSales-Products n:1

Products-Suppliers n:1

RetailSales-Suppliers n:1

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

Bảng PatientRecords:

sql

Copy

(patient\_id, patient\_name, visits)

* visits là XML:
* xml
* Copy

<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.

**Trả lời:**

Khử 1NF

Patients(patient\_id(PK), patient\_name)

Visits(visit\_id(PK), patient\_id(FK), visit\_date, docter\_id, diagnosis)

Patients-Visits 1:n

Chuẩn 3NF

Patients(patient\_id (PK), patient\_name)

Doctors(doctor\_id (PK), doctor\_name)

Visits(patient\_id (PK,FK), visit\_date, doctor\_id (PK,FK), diagnosis)

Patients-Visits 1:n

Doctors-Visits 1:n

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

Bảng AuctionBids:

sql

Copy

(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.

**Trả lời:**

Thiết kế lại

Auctions(auction\_id(PK), item\_name, currrent\_winner\_id)

Bidders(bidder\_id(PK), bidderr\_name)

Bids(auction\_id(PK,FK), bidder\_id(PK,FK), bid\_amount)

Auctions-Bids 1:n

Bidders-Bids 1:n

Phụ thuộc hàm từ bảng gốc

auction\_id ->item\_nateme, current\_winner\_id

auction\_id, bidder\_id -> bid\_amount

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

Bảng Inventory:

sql

Copy

(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.

**Trả lời:**

Vi phạm 2NF

Thuộc tính product\_name chỉ phụ thuộc vào product\_id (một phần của khóa chính)

warehouse\_location chỉ phụ thuộc vào warehouse\_id (một phần của khóa chính)

Chuẩn hóa

Product(product\_id(PK), product\_name)

Warehouse(warehouse\_id (PK), warehouse\_location)

Inventory(product\_id (PK,FK), warehouse\_id(PK,FK), current\_stock,reorder\_level, last\_restock\_date)

Product-Inventory 1:n

Warehohouse-Inventory 1:n

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

Bảng UserPosts:

sql

Copy

(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?**

**Trả lời:**

Vi phạm vì user\_name phụ thuộc vào user\_id, đây là phụ thuộc vào một phần khóa chính (khóa chính là post\_id, nhưng user\_name chỉ phụ thuộc vào user\_id, không phải post\_id).

parent\_post\_user phụ thuộc vào parent\_post\_id và không phụ thuộc vào post\_id, cũng là phụ thuộc một phần.

Chuẩn hóa 3NF

Users(user\_id(PK), user\_name)

Posts(post\_id (PK), user\_id(FK), post\_content, parent\_post\_id (FK references Posts) // quan hệ đệ quy

Users-Posts 1:n

Posts-Posts 1:n