

Bài tập tuần 08

Chuẩn hoá dữ liệu

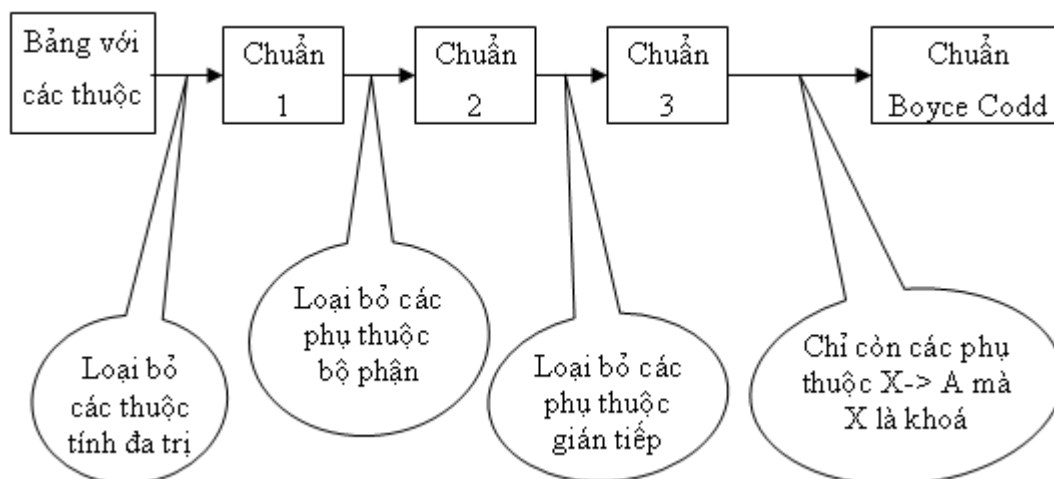
Nội dung

Chuẩn hoá là quá trình tách bảng (phân rã) thành các bảng nhỏ hơn dựa vào các phụ thuộc hàm. Các dạng chuẩn là các chỉ dẫn để thiết kế các bảng trong CSDL.

Mục đích của chuẩn hoá là loại bỏ các dư thừa dữ liệu và các lỗi khi thao tác dư thừa và các lỗi khi thao tác dữ liệu (Insert, Delete, Update). Tuy nhiên chuẩn hoá làm tăng thời gian truy vấn.

Có rất nhiều dạng chuẩn cho thiết kế dữ liệu; tuy nhiên, chúng ta sẽ cố gắng thực hiện ba loại chuẩn hóa quan trọng sau đây.

- **Dạng chuẩn một:** toàn bộ các miền giá trị của các cột có mặt trong bảng (quan hệ) đều chỉ chứa các giá trị nguyên tử (nguyên tố).
- **Dạng chuẩn hai:** bảng quan hệ phải thoả mãn dạng chuẩn 1 và các thuộc tính không khoá phải phụ thuộc hàm đầy đủ vào khoá chính.
- **Dạng chuẩn ba:** bảng quan hệ phải thoả mãn dạng chuẩn 2 và các thuộc tính không khoá phải phụ thuộc trực tiếp vào khoá chính.



Quá trình chuẩn hóa sẽ được thực hiện trong quy trình sau. Các thuộc tính của các thực thể và mối quan hệ của chúng cũng sẽ được tinh chỉnh từng bước bằng cách sử dụng kỹ thuật lập mô hình E-R.

Phần I: Câu hỏi và bài tập

Bài 1.1

a) Tác nhân ca sử dụng luôn là con người, không phải là các thiết bị hệ thống?

1. Sai

2. Đúng

b) Use-cases là một kịch bản mà mô tả?

1. Phần mềm thực hiện như thế nào khi được dùng trong một tình huống cho trước

2. Những công cụ CASE sẽ được dùng như thế nào để xây dựng hệ thống

3. Kế hoạch xây dựng cho sản phẩm phần mềm

4. Những test-case cho sản phẩm phần mềm

c) Phát biểu nào sau đây là đúng về yêu cầu phần mềm? Trả lời bằng cách đánh dấu T (đúng) hoặc F (sai).

(1) T / F Độ tin cậy và bảo mật là ví dụ về các yêu cầu chức năng.

(2) T / F Yêu cầu đóng vai trò như một cơ sở cho việc kiểm tra và xác minh.

(3) T / F Yêu cầu mô tả kiến trúc phần mềm.

(4) T / F Các yêu cầu về hành vi thường mang tính chủ quan và không thể đo lường được.

(5) T / F Các trường hợp sử dụng nắm bắt các yêu cầu chức năng.

(6) T / F Lý do số một mà các dự án thành công là sự tham gia của nhà phát triển.

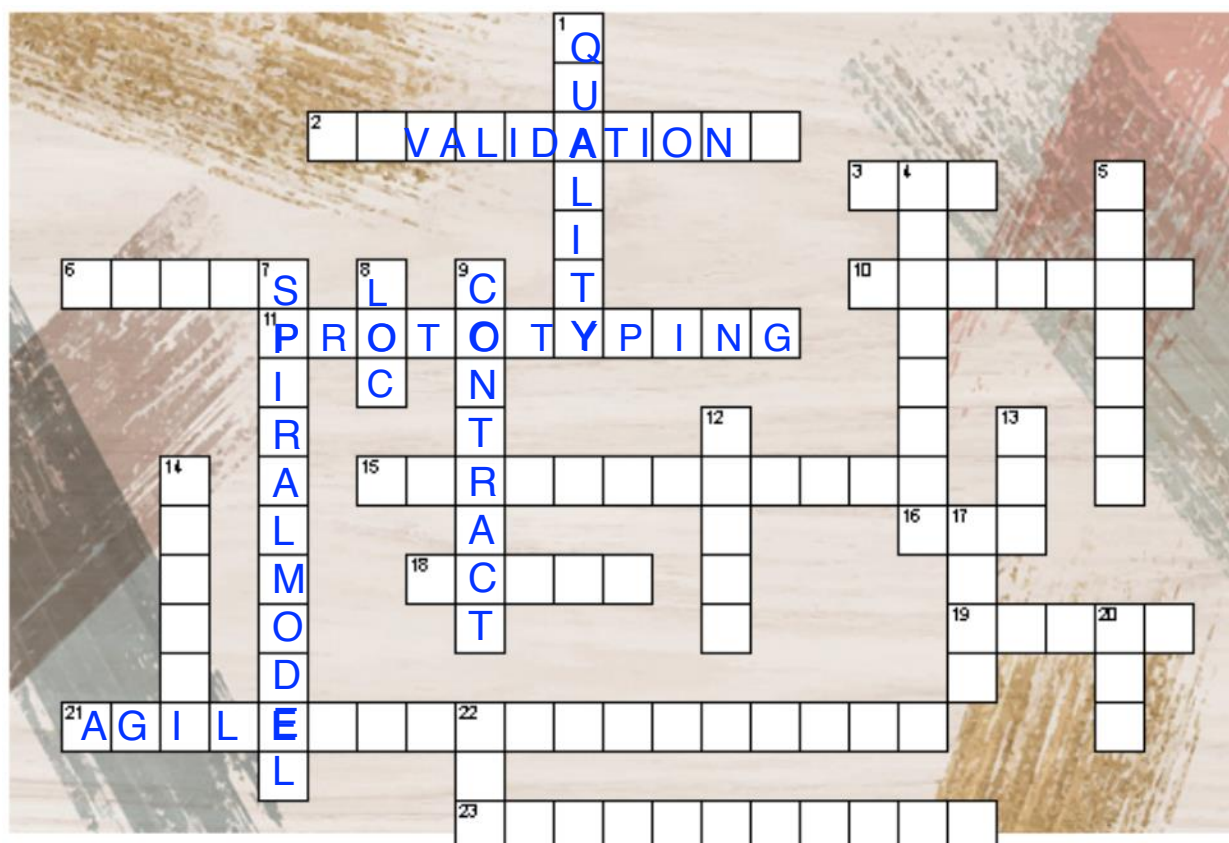
(7) T / F Một ca sử dụng đại diện cho một hành vi ví dụ của hệ thống.

(8) T / F Các tình huống mở rộng của một ca sử dụng thiết lập sự hiểu biết giữa khách hàng và nhà phát triển hệ thống về các yêu cầu.

(9) T / F Trong hầu hết các trường hợp sử dụng, gần như mọi bước đều có thể bị lỗi.

Bài 1.2

Hãy giải ô chữ tổng hợp kiến thức dưới đây với các gợi ý kèm theo?



Gợi ý:

Across

2. Kiểm tra để đảm bảo rằng phần mềm tuân thủ các yêu cầu của nó (10)
3. Sự phân tách thứ bậc của các hoạt động công việc trong một dự án phần mềm (3)
6. Một khái niệm mô tả cho việc đóng gói các thuộc tính và hoạt động trong OOP (5)
10. Một chuỗi các giao dịch trong cuộc đối thoại giữa một tác nhân và một thành phần hoặc hệ thống (7)
11. Tạo bản mô phỏng nhanh cho ứng dụng (11)
15. Hoạt động kiểm tra để đảm bảo tính đúng đắn (12)

Down

1. Sản phẩm đáp ứng các tiêu chí của khách hàng, phù hợp với đặc điểm thiết kế (7)
4. Điểm mà một số sản phẩm có thể phân phối được đặt dưới sự kiểm soát thay đổi chính thức (8)
5. Một tập các hoạt động nhằm cố gắng phát hiện và loại bỏ lỗi (7)
7. Một mô hình phát triển phần mềm tiến hóa với pha xác định và kiểm soát rủi ro (11)
8. Lines of code (3)
9. Một thỏa thuận ràng buộc lẫn nhau (8)
12. Loại biểu đồ được sử dụng phổ biến trong quản lý dự án (5)

Across

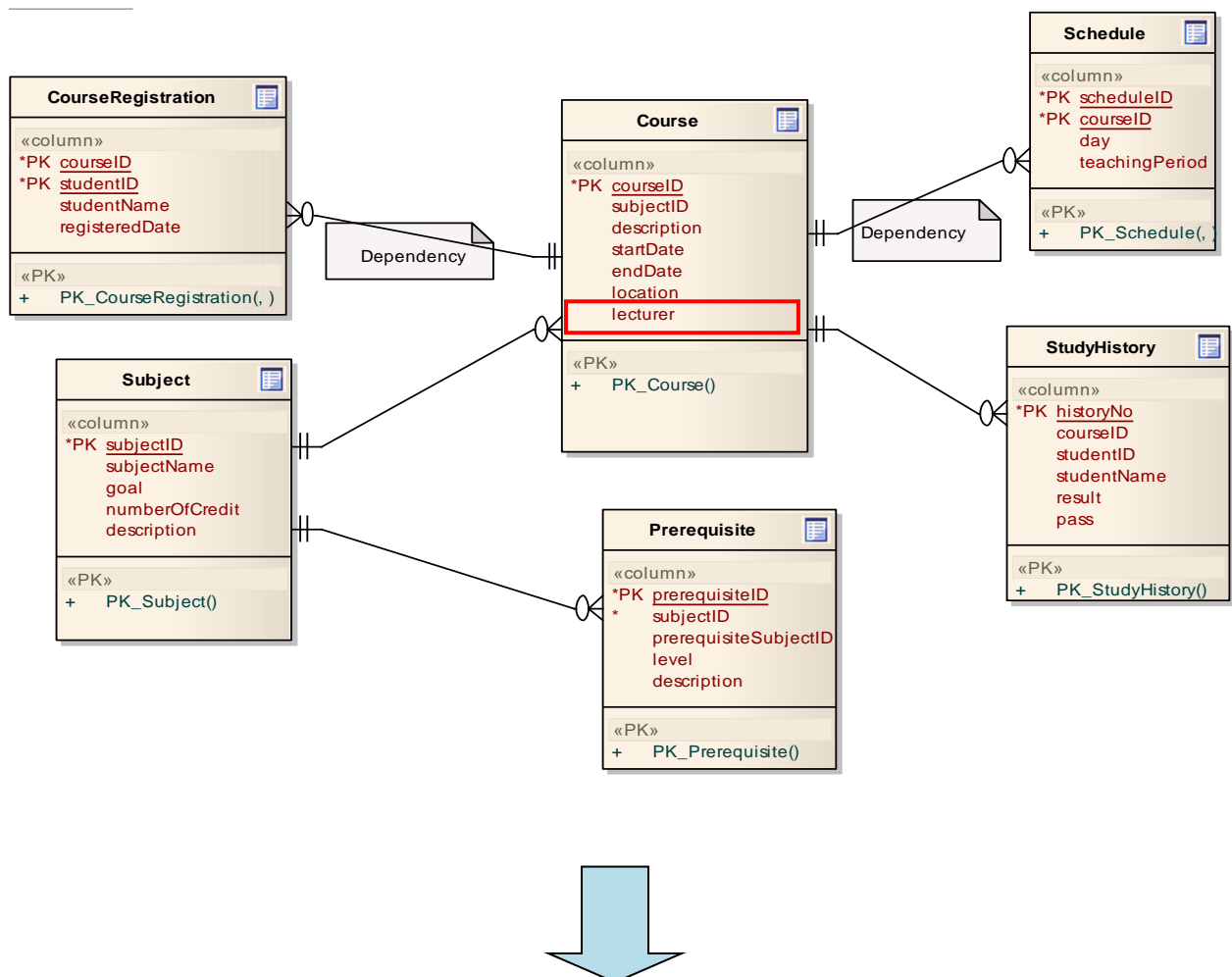
16. Loại sơ đồ thể hiện thiết kế cơ sở dữ liệu về thực thể và quan hệ (3)
18. Một loại vai trò được thực hiện bởi một thực thể trong tương tác với hệ thống (5)
19. Một phương pháp Agile dành cho quản lý tiến trình phần mềm với các vòng lặp ngắn gọi là sprint (5)
21. Một phương pháp kỹ thuật phần mềm được sử dụng trong Agile có thể sử dụng lập trình cặp đôi (18)
23. Khả năng chuyển phần mềm có thể hoạt động từ môi trường này sang môi trường khác (11)

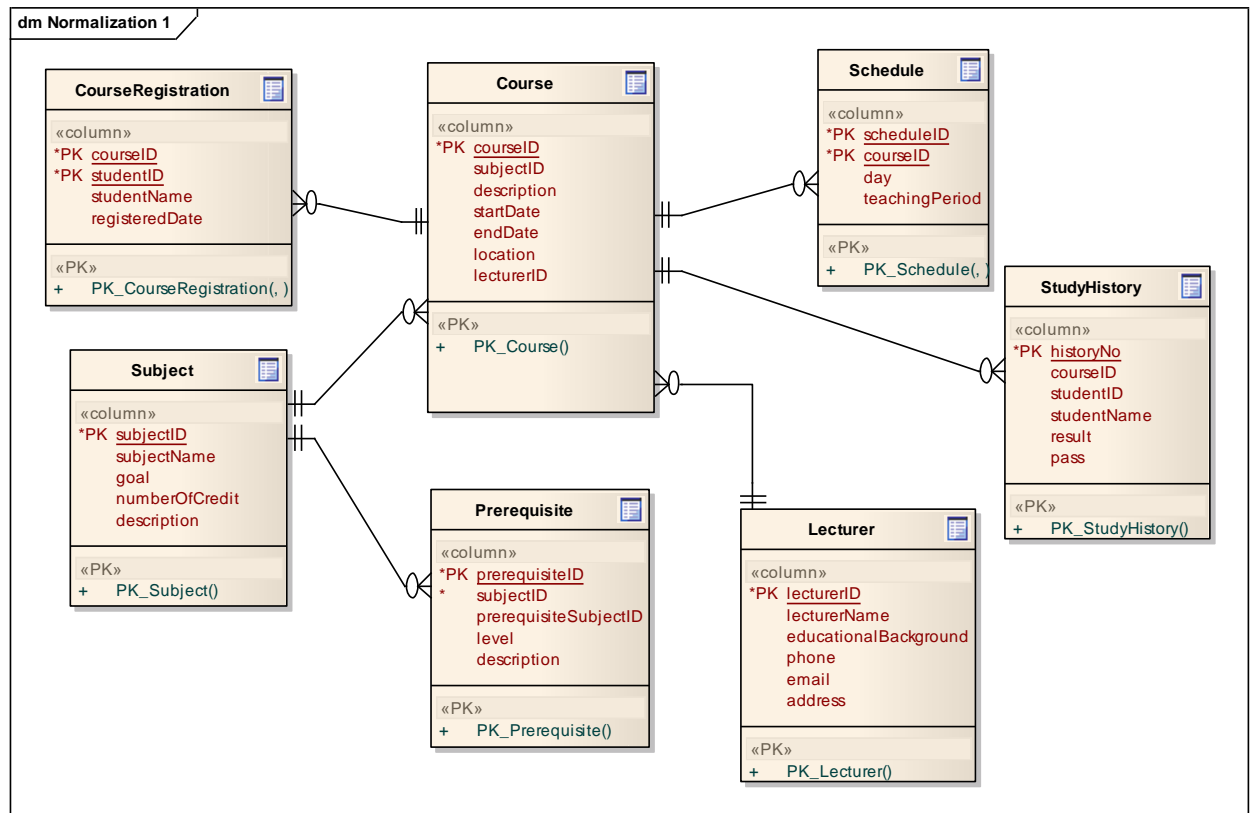
Down

13. Loại biểu đồ thể hiện luồng dữ liệu trong hệ thống (3)
14. Một lỗi hỏng trong một bộ phận hoặc hệ thống gây ra việc không thực hiện được chức năng cần thiết (6)
17. Một sự kiện hoặc điều kiện không chắc chắn (4)
20. Một ngôn ngữ mô hình hóa mục đích chung được chuẩn hóa (3)
22. Khung quy trình phát triển phần mềm hợp nhất có tính lặp có thể thích ứng, thường sử dụng UML kèm theo (3)

Phần II: Thực hiện các bước chuẩn hoá cho bài toán Quản lý đăng ký học tập

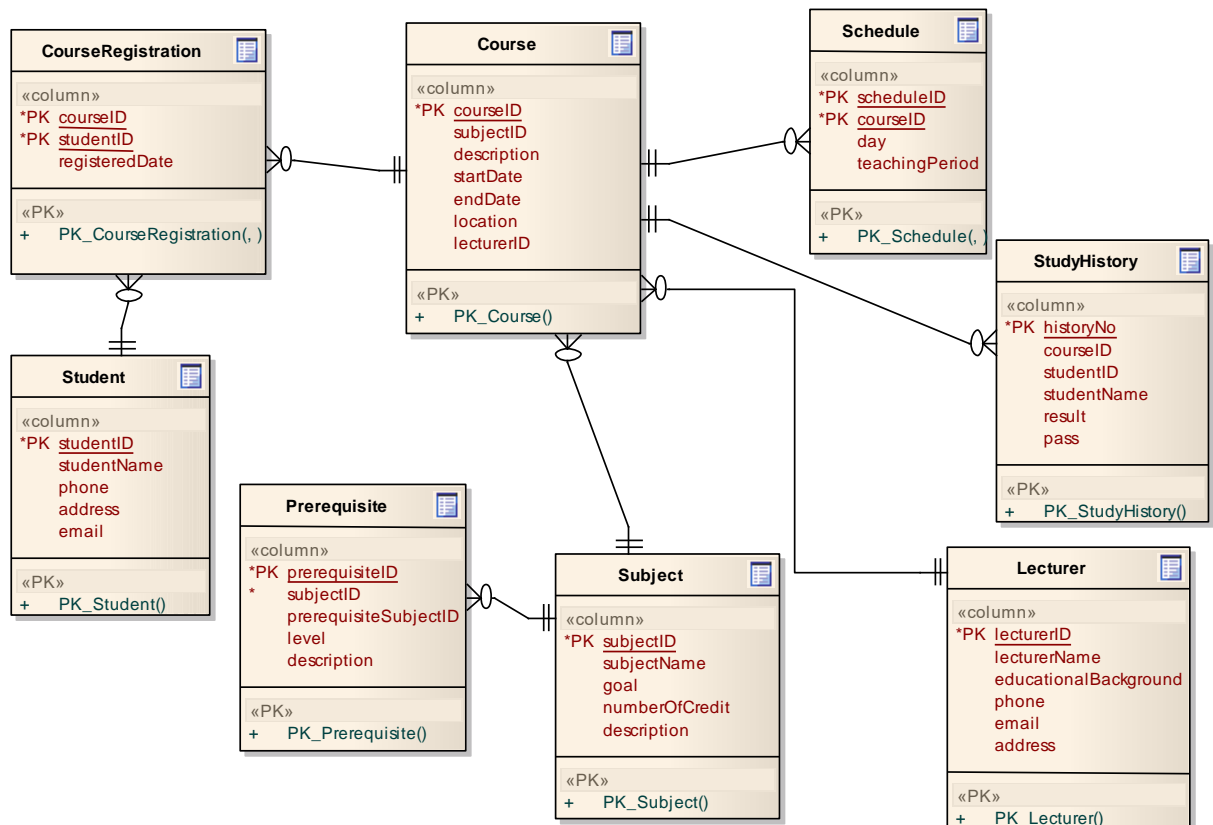
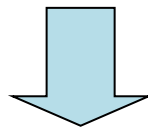
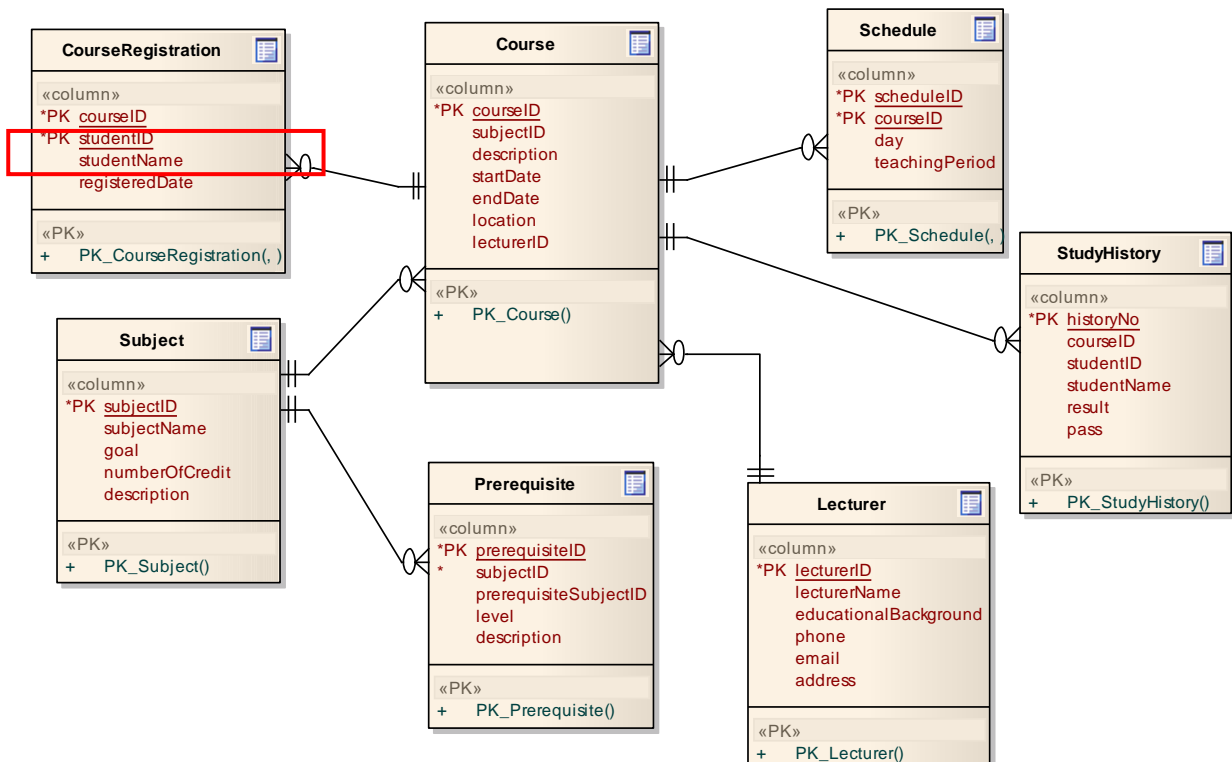
Biến đổi về dạng chuẩn 1:





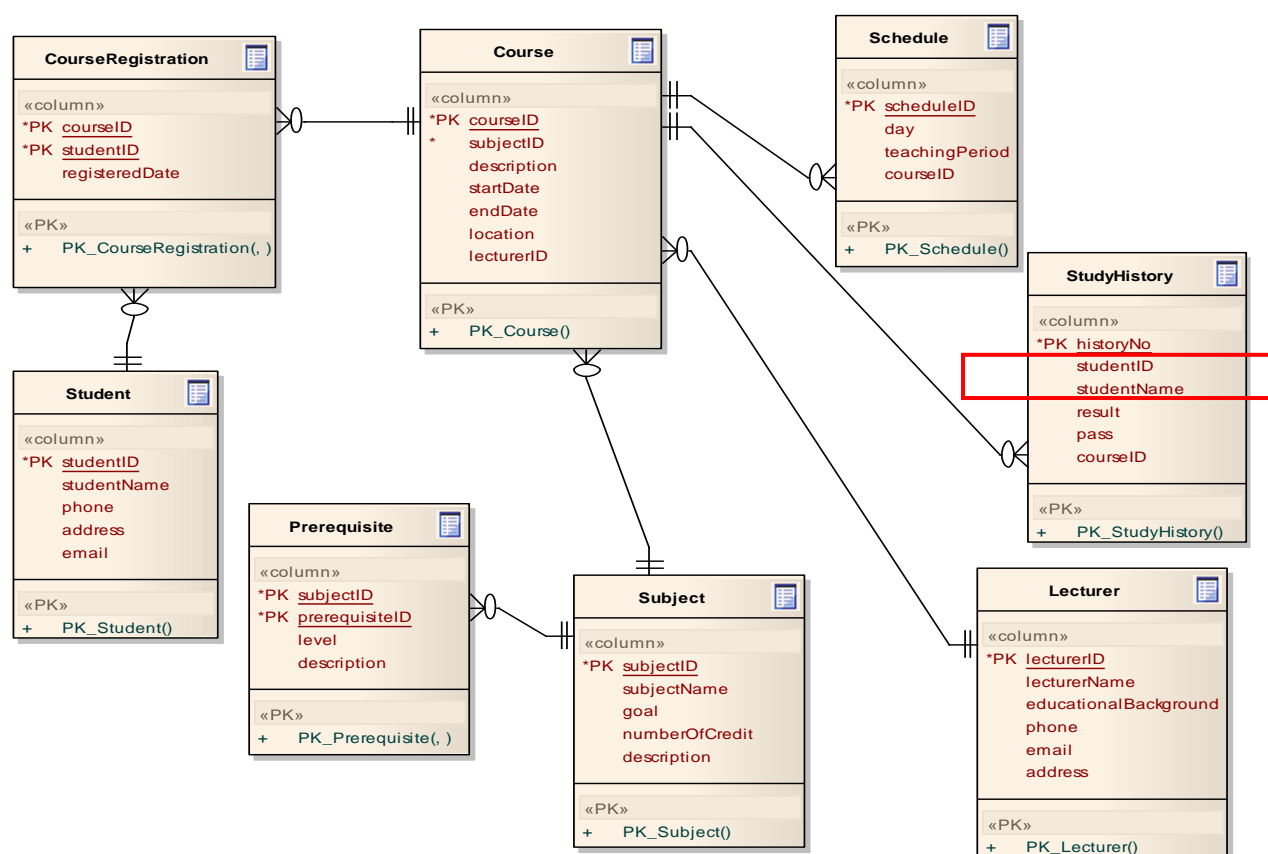
- Thuộc tính “lecturer” của thực thể “Course” là một thuộc tính lặp: để đáp ứng yêu cầu của dạng chuẩn 1, thuộc tính này được tách ra thành một thực thể mới, thực thể “Lecturer”, thực thể này có thuộc tính “lecturer ID” là khoá chính.
- Thực thể “Course” chứa khoá ngoài tham chiếu đến khoá chính của thực thể “Lecturer”. Mỗi quan hệ giữa “Lecturer” entity và “Course” entity là quan hệ độc lập.

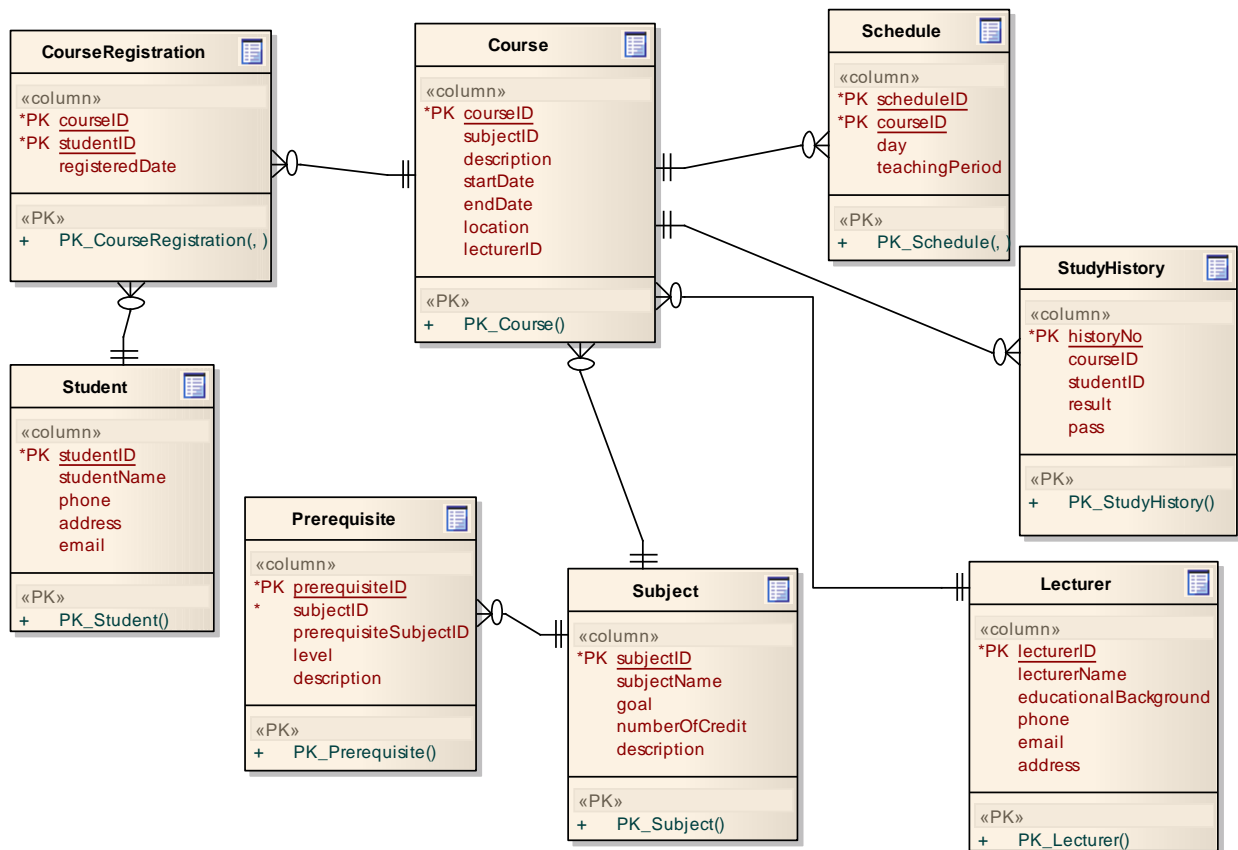
Biến đổi về dạng chuẩn 2:



- Thực thể “Course Registration” có khoá chính là tổ hợp của {CourseID, StudentID}, tuy nhiên trong thực thể này thuộc tính “student name” chỉ phụ thuộc vào một bộ phận của khoá chính, là thuộc tính StudentID → Thực thể này chưa thoả mãn tiêu chí của dạng chuẩn 2
- Để thực hiện chuẩn hoá về dạng chuẩn 2, phụ thuộc bộ phận giữa thuộc tính “student name” vào StudentID được tách ra thành một quan hệ mới. Thực thể mới này có tên là “Student” và có khoá chính là “StudentID”. Thực thể “Course Registration” vẫn duy trì khoá chính là tổ hợp của {CourseID, StudentID}, trong đó StudentID là khoá ngoài tham chiếu đến khoá chính của thực thể “Student”.
→ quan hệ giữa thực thể “Student” và “Course Registration” là mối quan hệ phụ thuộc.

Biến đổi về dạng chuẩn 3:





- Thuộc tính “student name” của thực thể “Study history” phụ thuộc vào thuộc tính không khoá là “student ID” → vi phạm tiêu chí của dạng chuẩn 3; phụ thuộc giữa hai thuộc tính này được tách ra thành một thực thể mới, thực thể mới này trùng với thực thể “Student” đã tạo từ bước trước.
- Kiểm tra các thực thể khác trong sơ đồ, tất cả đều đảm bảo thoả mãn tiêu chí của dạng chuẩn 3.

Xây dựng cơ sở dữ liệu vật lý và đặc tả các bảng dữ liệu:

- Xây dựng các bảng dữ liệu, xác định quan hệ giữa các bảng và thiết kế chi tiết cho mỗi trường (cột) trong bảng.
- Một số lưu ý khi thiết kế các trường dữ liệu: Field là đơn vị nhỏ nhất của dữ liệu → Field tương ứng với 1 thuộc tính (attribute) trong mô hình dữ liệu logic. Quyết định cần làm khi thiết kế là phải **chọn kiểu dữ liệu cho field, kiểm soát tính toàn vẹn dữ liệu** và Hệ quản trị dữ liệu sẽ quản lý các giá trị bị thiếu cho field như thế nào?

Loại đặc tả	Mô tả nội dung
Tên trường (field name)	Theo quy định về cách đặt tên hoặc quy ước viết code của nhóm
Kiểu trường (data type)	Kiểu dữ liệu phù hợp với các giá trị cần lưu trữ trong các trường
Kích cỡ (size)	Kích thước / Khoảng giá trị tối đa và tối thiểu
Mã hóa (Coding)	Mã hoá các thông tin lưu trữ / Khuôn dạng của dữ liệu
Các quy tắc toàn vẹn dữ liệu (data integrity rules)	Đặc tả về các hạn chế đặt lên giá trị của trường: NULL / NOT NULL/ KHOÁ CHÍNH,...
Các kiểm soát bảo trì (maintenance controls)	Chỉ ra những giá trị nào được phép thay đổi
Công thức (Formular)	Mô tả công thức tính toán giá trị với những trường cần tính toán.
Toàn vẹn tham chiếu (references integrity)	Đặc tả giá trị của trường có liên quan đến giá trị của trường khác, ví dụ: KHOÁ NGOẠI,...
Sở hữu (Ownership)	Ai có quyền đối với dữ liệu.

- Xây dựng sơ đồ quan hệ giữa các bảng dữ liệu
- Đặc tả thiết kế dữ liệu cho từng bảng theo mẫu sau:

Tên bảng:

Tên trường	Kiểu dữ liệu	Kích thước	Ràng buộc toàn vẹn	Khuôn dạng	Ghi chú

HẾT