



BÀI 1. HỆ THỐNG THÔNG TIN

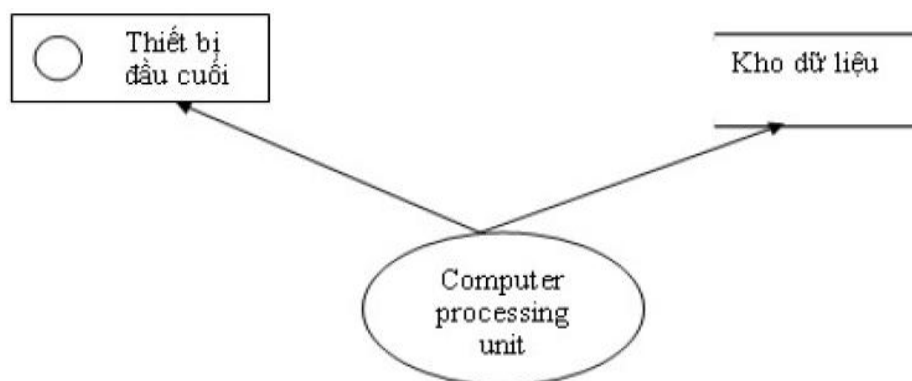
TÓM TẮT

Mục đích chính của bài này cung cấp cho chúng ta các khái niệm về hệ thống, hệ thống thông tin, các mô hình hệ thống thông tin, cách xử lý thông tin trên máy tính

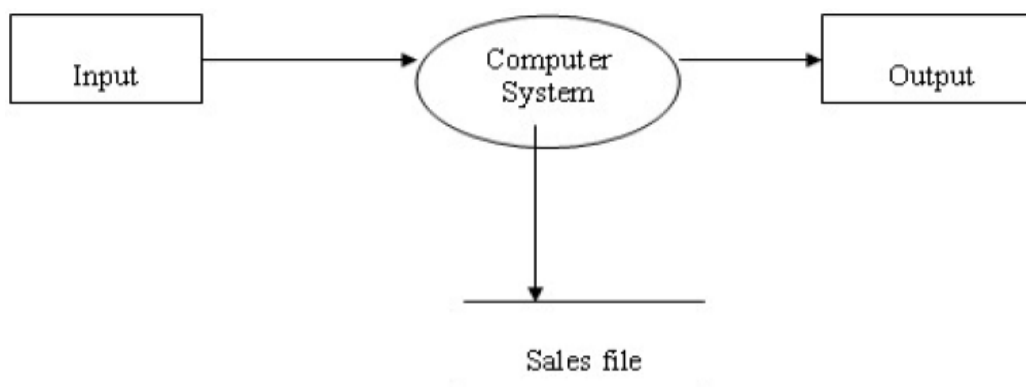
1.1. HỆ THỐNG THÔNG TIN

1.1.1. Các khái niệm

- **Hệ thống** (system): Trong các hoạt động của con người, các thuật ngữ như hệ thống triết học, hệ thống pháp luật, hệ thống kinh tế, hệ thống thông tin đã trở nên quen thuộc. Một cách đơn giản và vắn tắt, ta có thể hiểu: Hệ thống là một tập hợp vật chất và phi vật chất như người, máy móc, thông tin, dữ liệu, các phương pháp xử lý, các qui tắc, quy trình xử lý, gọi là các phần tử của hệ thống. Trong hệ thống, các phần tử tương tác với nhau và cùng hoạt động để hướng tới mục đích chung.
- **Hệ thống thông tin** (information system): Hệ thống thông tin là một hệ thống mà mục tiêu tồn tại của nó là cung cấp thông tin phục vụ cho hoạt động của con người trong một tổ chức nào đó. Ta có thể hiểu hệ thống thông tin là hệ thống mà mối liên hệ giữa các thành phần của nó cũng như mối liên hệ giữa nó với các hệ thống khác là sự trao đổi thông tin. Một số ví dụ về hệ thống thông tin.



Hình 1.1. Máy tính là một hệ thống thông tin



Hình 1.2. Hệ thống thông tin có máy tính tham gia

1.1.2. Các loại hệ thống thông tin cơ bản

- **Hệ thống xử lý giao dịch** (Transaction processing system – TPS) là một hệ thống thông tin có chức năng thu thập và xử lý dữ liệu về các giao dịch nghiệp vụ.
- **Hệ thống thông tin quản lý** (Management information system - MIS) là một hệ thống thông tin cung cấp thông tin cho việc báo cáo hướng quản lý dựa trên việc xử lý giao dịch và các hoạt động của tổ chức.
- **Hệ thống hỗ trợ quyết định** (Decision support system – DSS) là một hệ thống thông tin vừa có thể trợ giúp xác định các thời cơ ra quyết định, vừa có thể cung cấp thông tin để trợ giúp việc ra quyết định.
- **Hệ thống thông tin điều hành** (Executive information system – EIS) là một hệ thống thông tin hỗ trợ nhu cầu lập kế hoạch và đánh giá của các nhà quản lý điều hành.
- **Hệ thống chuyên gia** (Expert System) là hệ thống thông tin thu thập tri thức chuyên môn của các chuyên gia rồi mô phỏng tri thức đó nhằm đem lại lợi ích cho người sử dụng bình thường.
- **Hệ thống truyền thông và cộng tác** (Communication and collaboration system) là một hệ thống thông tin làm tăng hiệu quả giao tiếp giữa các nhân viên, đối tác, khách hàng và nhà cung cấp để củng cố khả năng cộng tác giữa họ.
- **Hệ thống tự động văn phòng** (Office automation system) là một hệ thống thông tin hỗ trợ các hoạt động nghiệp vụ văn phòng nhằm cải thiện luồng công việc giữa các nhân viên.

1.1.3. Những đặc trưng cơ bản của hệ thống:

- Hệ thống phải được thiết kế, tổ chức trong ngữ cảnh chung của nhiều mặt kinh tế xã hội, tức là nó ứng dụng lý thuyết hệ thống.
- Hệ thống phải đạt được mục tiêu là ra các quyết định.
- Hệ thống phải dựa trên các kỹ thuật và công nghệ tiên tiến về xử lý thông tin.
- Hệ thống có kết cấu mềm dẻo, phát triển được, dễ bảo trì, nâng cấp về sau.



1.2. PHƯƠNG THỨC XỬ LÝ THÔNG TIN TRONG MÁY TÍNH

1.2.1. Xử lý tương tác:

Xử lý tương tác là xử lý thực hiện từng phần, phần xử lý bởi con người và bởi máy tính được thực hiện xen kẽ nhau. Trong quá trình xử lý tương tác, máy tính đóng vai trò trợ giúp tích cực. Xử lý tương tác là phương thức được lựa chọn cho các hệ thống phải xử lý nhiều thông tin có mối quan hệ phức tạp với nhau, khó mô tả bằng các công thức, các phương trình toán học.

Con người phải thường xuyên vận dụng những kinh nghiệm công tác của mình vào trong quá trình xử lý.

1.2.2. Xử lý giao dịch:

Xử lý giao dịch là xử lý một yêu cầu cho đến khi ra kết quả, không có sự can thiệp từ ngoài vào. Xử lý giao dịch thích hợp với những tiến trình có nhiều khâu độc lập với nhau để kiểm tra và xử lý thông tin. Thủ tục rút tiền từ ngân hàng là một ví dụ về xử lý giao dịch. Khởi đầu là kiểm tra lỗi các thông tin nhập vào, tiếp theo kiểm tra sự tương thích của các thông tin này với các dữ liệu đã có trong hệ thống, trên cơ sở kết quả kiểm tra, hệ thống sẽ đáp ứng yêu cầu của khách hàng.

1.2.3. Xử lý theo lô:

Xử lý theo lô hay còn gọi là xử lý trọn gói, hoặc xử lý theo mẻ, là tiến trình tập hợp những thông tin sẵn có hoặc tạo ra thông tin mới theo định kỳ. Ví dụ về xử lý theo lô là lập báo cáo định kỳ hàng tuần, hàng tháng. Phương thức xử lý theo lô thích hợp với những tiến trình xử lý thông tin mà trong đó: Việc truy cập thông tin diễn ra định kỳ, Khuôn dạng và kiểu dữ liệu hoàn toàn xác định, Thông tin khá ổn định trong khoảng thời gian giữa hai tiến trình xử lý liên tiếp.

1.2.4. Xử lý trực tuyến:

Xử lý trực tuyến được sử dụng trong những hệ thống mà tại đó đòi hỏi xử lý từng dòng thông tin, từng mẫu tin ngay tại thời điểm nó mới xuất hiện, một cách trực tiếp trong đối thoại giữa các đối tác. Ví dụ như dịch vụ gửi tiền tại ngân hàng, các xử lý tại phòng bán vé máy bay, tàu hoả, hoặc dịch vụ thông tin tại tổng đài thường là các xử lý trực tuyến. Đặc trưng của các xử lý trực tuyến là: Việc truy cập thông tin xảy ra hoàn toàn ngẫu nhiên, Khuôn dạng và kiểu thông tin không hoàn toàn xác định, Thông tin thay đổi liên tục ngay trong khi thực hiện tiến trình xử lý.

1.2.5. Xử lý thời gian thực:

Xử lý thời gian thực là các tiến trình máy tính phải đảm bảo các yêu cầu rất ngặt nghèo của hệ thống về thời gian. Thông thường các xử lý thời gian thực xuất hiện trong các hệ thống có liên kết với các hệ thống ngoài như hệ thống điều khiển nhiệt độ lò luyện thép hoặc lò nấu sợi, hệ thống điều khiển đường bay của tên lửa hoặc các hệ thống mô phỏng. Xử lý thời gian



thực phải đảm bảo đồng bộ các tiến trình máy tính với các hoạt động diễn ra trong thực tế.

1.2.6. Xử lý phân tán:

Xử lý có thể diễn ra rai các bộ phận ở những vị trí khác nhau, có những yêu cầu khác nhau vào những thời điểm cũng có thể khác nhau. Nói chung, với những hệ thống có sự xử lý phân tán, dữ liệu thường được bố trí ở những vị trí địa lý khác nhau. Trong xử lý phân tán, với một thành phần dữ liệu, có thể cùng một lúc xảy ra nhiều thao tác như cập nhập, sửa chữa hoặc khai thác khác nhau. Vì vậy, một trong những vấn đề cần phải quan tâm đối với các xử lý phân tán là đảm bảo tính đồng bộ trong hệ thống.

1.3. CÁC GIAI ĐOẠN TRIỂN KHAI DỰ ÁN XÂY DỰNG HTTT

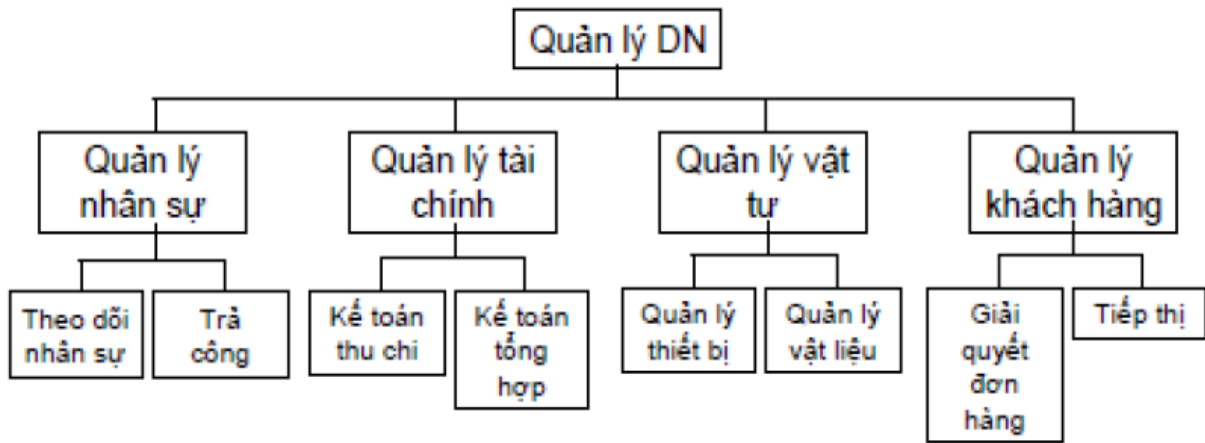
Hệ thống thông tin cũng như bất kỳ hệ thống nào khác, nó có một cuộc sống cùng với các chu kỳ sống có những đặc trưng riêng. Nó được sinh ra, phát triển và cuối cùng thì bị thay thế (loại bỏ) bởi một hệ thống khác tiên tiến hơn, hiện đại hơn. Ta có thể chia cuộc sống, hay còn gọi là vòng đời (life cycle), của hệ thống thông tin ra làm các giai đoạn như sau:

- **Giai đoạn chuẩn bị:** Giai đoạn này tính từ khi trong tổ chức xuất hiện nhu cầu xây dựng hệ thống thông tin mới nhằm cung cấp thông tin chính xác, kịp thời cho việc điều hành các hoạt động sản xuất trong tổ chức.
- **Giai đoạn hình thành và phát triển:** Trong giai đoạn này, các dự định xây dựng hệ thống thông tin được triển khai thực hiện trong thực tế. Các chuyên gia phân tích hệ thống, nhà quản lý và các lập trình viên cùng nghiên cứu, khảo sát, phân tích, thiết kế và xây dựng hệ thống thông tin quản lý. Hệ thống thông tin được thử nghiệm, cài đặt và chuẩn bị đưa vào sử dụng.
- **Giai đoạn khai thác và sử dụng:** Thông thường đây là giai đoạn dài nhất trong vòng đời của hệ thống thông tin quản lý. Trong giai đoạn này hệ thống được vận hành phục vụ cho nhu cầu khai thác và sử dụng thông tin trong tổ chức. Trong quá trình sử dụng, hệ thống được bảo trì hoặc sửa chữa để phù hợp với sự thay đổi về thông tin hoặc nhu cầu thông tin.
- **Giai đoạn thay thế:** Trong quá trình sử dụng và khai thác hệ thống, luôn gặp phải sự thay đổi về thông tin (thay đổi về dung lượng và về cấu trúc), những sửa chữa và thay đổi trong hệ thống làm cho nó trở nên cồng kềnh, hoạt động kém hiệu quả. Vì vậy, hệ thống thông tin cũ cần phải được thay thế bởi hệ thống thông tin mới hoặc nâng cấp hệ thống cũ.

1.4. CÁC MÔ HÌNH BIỂU DIỄN HỆ THỐNG

1.4.1. Biểu đồ phân cấp chức năng (BFD - Business Flow Diagram):

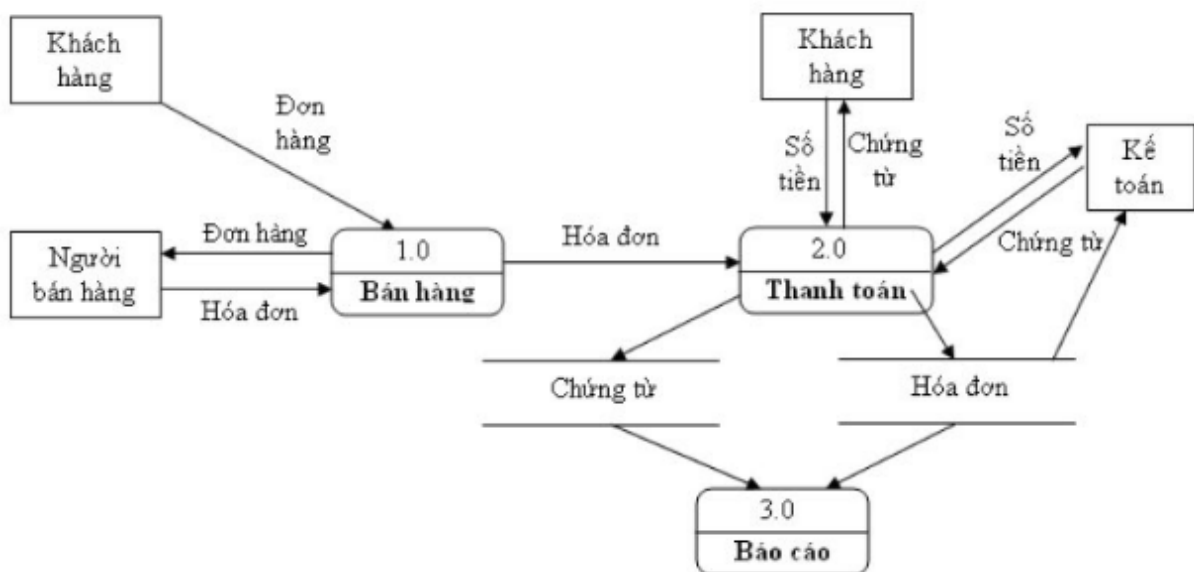
Diễn tả sự phân rã dần dần các chức năng từ tổng thể đến chi tiết, mỗi nút trong biểu đồ là một chức năng. Đặc điểm của biểu đồ phân rã chức năng cho cái nhìn khái quát, từ tổng thể đến chi tiết; dễ thành lập; có tính chất tĩnh (không thấy trình tự xử lý); thiếu sự trao đổi giữa các thông tin chức năng. Ví dụ về biểu đồ phân cấp chức năng:



Hình 1.3. Biểu đồ phân cấp chức năng

1.4.2. Biểu đồ luồng dữ liệu (DFD - Data Flow Diagram):

- Biểu đồ luồng dữ liệu là một loại biểu đồ nhằm mục đích diễn tả một quá trình xử lý thông tin với các yêu cầu sau: Sự diễn tả là ở mức logic, nghĩa là nhằm trả lời câu hỏi: “Làm gì?” mà bỏ qua câu hỏi “Làm như thế nào?”. Chỉ rõ các chức năng (con) phải thực hiện để hoàn tất quá trình xử lý cần mô tả. Chỉ rõ các thông tin được chuyển giao giữa các chức năng đó, và qua đó phần nào thấy được trình tự thực hiện của chúng.
- Biểu đồ luồng dữ liệu là một công cụ dùng để trợ giúp cho bốn hoạt động chính: Phân tích: được dùng để xác định yêu cầu của người sử dụng. Thiết kế DFD dùng để vạch kế hoạch và minh họa các phương án cho phân tích viên hệ thống và người dùng khi thiết kế hệ thống mới. Biểu đồ DFD là công cụ đơn giản, dễ hiểu đối với phân tích viên hệ thống và người dùng. Tài liệu DFD cho phép biểu diễn tài liệu phân tích hệ thống một cách đầy đủ, súc tích và ngắn gọn. DFD cung cấp cho người sử dụng một cái nhìn tổng thể về hệ thống và cơ chế lưu chuyển thông tin trong hệ thống đó.



Hình 1.4. Biểu đồ luồng dữ liệu



- Mục đích của DFD là giúp chúng ta thấy được đằng sau những cái gì thực tế xảy ra trong hệ thống.
- Biểu đồ DFD dựa vào phương pháp phát triển hệ thống có cấu trúc bao gồm ba kỹ thuật phân tích chính:
 - Biểu đồ DFD mô tả quan hệ giữa quá trình xử lý và các luồng dữ liệu.
 - Từ điển định nghĩa dữ liệu mô tả các phần tử luồng dữ liệu.
 - Xác định quá trình xử lý, mô tả quá trình xử lý một cách chi tiết.
- DFD diễn tả ở hai mức: mức vật lý và mức logic.
 - Mức vật lý: Mô tả hệ thống làm như thế nào?
 - Mức logic: Mô tả hệ thống làm gì?

1.4.3. Phân mức các chức năng của hệ thống:

- Có ba mức cơ bản được ứng dụng:
 - Mức 0: DFD mức ngữ cảnh (Context Level DFD).
 - Mức 1: DFD mức đỉnh (Top Level DFD).
 - Mức 2: DFD mức dưới đỉnh (Levelling DFD).
- DFD *mức ngữ cảnh (mức 0)*: là mô hình hệ thống ở mức tổng quát nhất, ta xem cả hệ thống như một chức năng. Tại mức này hệ thống chỉ có duy nhất một chức năng.
- DFD *mức đỉnh (mức 1)*: gồm nhiều chức năng được phân rã từ DFD mức ngữ cảnh với các chức năng phân rã tương ứng mức 1 của biểu đồ phân cấp chức năng BFD.
- DFD *mức dưới đỉnh (mức 2)*: phân rã từ DFD mức đỉnh. Các chức năng được định nghĩa riêng từng biểu đồ hoặc ghép lại thành một biểu đồ trong trường hợp biểu đồ đơn giản.

CÂU HỎI THẢO LUẬN

- 1.1 Nêu khái niệm về hệ thống và hệ thống thông tin? Cho ví dụ minh họa về hệ thống và hệ thống thông tin?
- 1.2 Phân loại và nêu ý nghĩa của các hệ thống thông tin cơ bản ?
- 1.3 Nêu khái niệm về các phương thức xử lý thông tin trong máy tính? Cho ví dụ minh họa ?

Một số bài tập lớn: là những bài tập nhằm đánh giá kết quả học tập. Yêu cầu là thực hiện xây dựng một hệ thống thông tin quản lý qua các bước như: Khảo sát thực tế và xác lập các yêu cầu; Phân tích hệ thống; Thiết kế hệ



thông; Xây dựng và cài đặt hệ thống. Học viên chọn một trong các đề tài gợi ý dưới đây hoặc tự đưa ra đề tài.

STT	Tên hệ thống thông tin quản lý	Mô tả
1	Hệ thống quản lý học sinh phổ thông trung học	Quản lý hồ sơ học sinh, kết quả học tập các môn học, kết quả tổng kết cuối năm...
2	Hệ thống quản lý khách sạn	Quản lý phòng, khách thuê phòng, các dịch vụ khách đã sử dụng...
3	Hệ thống quản lý giáo viên phổ thông trung học	Quản lý thông tin giáo viên, các quá trình, sự kiện liên quan đến giáo viên...
4	Hệ thống quản lý vật tư	Quản lý tình hình xuất nhập vật tư...
5	Hệ thống quản lý thư viện	Quản lý sách, độc giả, tình hình mượn trả...
6	Hệ thống quản lý sinh viên đại học	Quản lý hồ sơ sinh viên, kết quả học tập, quản lý các môn học của học sinh...
7	Hệ thống quản lý dược	Quản lý quá trình xuất nhập thuốc, bán thuốc cho bệnh nhân,....
8	Hệ thống quản lý tuyển sinh đại học	Quản lý thông tin tuyển sinh đại học, kết quả thi đại học....
9	Hệ thống quản lý bán điện	Quản lý các hợp đồng mua điện của khách hàng, phân loại hợp đồng, quản lý mức tiêu thụ điện





BÀI 2. KHẢO SÁT HIỆN TRẠNG VÀ THU THẬP THÔNG TIN

Tóm tắt: Nghiên cứu các phương pháp để khảo sát thực tế nhằm thu thập các thông tin phục vụ cho việc phân tích hệ thống. Một số phương pháp khảo sát thu thập thông tin cụ thể như sau: phỏng vấn, sử dụng phiếu hỏi, lấy mẫu, phân tích tài liệu định lượng / định tính, quan sát.

2.1. PHƯƠNG PHÁP PHỎNG VẤN

- Phỏng vấn là một phương pháp quan trọng để thu thập dữ liệu về các yêu cầu của hệ thống thông tin.
- Việc phỏng vấn nhằm phát hiện thông tin về:
 - Các ý kiến của người được phỏng vấn.
 - Cảm giác của người được phỏng vấn.
 - Trạng thái hiện tại của hệ thống.
 - Các mục tiêu của con người và tổ chức.
 - Các thủ tục nghiệp vụ không chính thức
- Việc lập kế hoạch phỏng vấn thường trải qua 5 bước:
 - Đọc các tài liệu cơ bản.
 - Thiết lập các mục tiêu phỏng vấn.
 - Xác định người đi phỏng vấn.
 - Chuẩn bị người được phỏng vấn.
 - Quyết định cấu trúc và kiểu câu hỏi.
- Xây dựng câu hỏi phỏng vấn phải thuộc hai dạng cơ bản:
 - Câu hỏi mở.
 - Câu hỏi đóng.

2.1.1. Dạng câu hỏi mở

- Các câu hỏi phỏng vấn mở cho phép những người được phỏng vấn trả lời những gì họ mong muốn và mức độ mong muốn của họ.
- Các câu hỏi mở phù hợp khi người phân tích quan tâm tới độ rộng và sâu của câu trả lời.
- Các ưu điểm:
 - Làm cho người được phỏng vấn cảm thấy thoải mái.
 - Cho phép người phỏng vấn tập trung vào cách biểu đạt của người được phỏng vấn:
 - Phản ánh trình độ văn hóa, các giá trị, thái độ và niềm tin.
 - Cung cấp mức độ chi tiết cao.
 - Phát hiện các câu hỏi mới mà chưa được khai thác.
 - Làm cho người được phỏng vấn thấy thú vị hơn.
 - Cho phép tính tự nhiên cao hơn.



Giúp người phỏng vấn dễ điều chỉnh nhịp độ hơn.

Hữu ích khi người phỏng vấn không chuẩn bị trước.

- Các nhược điểm:

Có thể thu được quá nhiều chi tiết không liên quan.

Có thể mất đi tính điều khiển cuộc phỏng vấn.

Có thể mất quá nhiều thời gian để thu được thông tin có ích.

Có khả năng thể hiện rằng người phỏng vấn không chuẩn bị.

Có thể gây ấn tượng rằng người phỏng vấn đang trong “cuộc hành trình đi câu”

2.1.2. Dạng câu hỏi đóng

- Câu hỏi đóng hạn chế số câu trả lời có thể có.

- Câu hỏi đóng phù hợp để tạo ra dữ liệu đáng tin cậy và chính xác, dễ dàng để phân tích.

- Phương pháp lập luận hiệu quả và đòi hỏi ít kỹ năng đối với người phỏng vấn

- Các ưu điểm:

Tiết kiệm thời gian phỏng vấn.

Dễ dàng so sánh giữa các lần phỏng vấn.

Dễ đạt đúng mục đích.

Kiểm soát được cuộc phỏng vấn.

Thu hoạch được các dữ liệu liên quan.

Bao phủ một phạm vi rộng lớn một cách nhanh chóng.

- Các nhược điểm:

Nhàm chán đối với người được phỏng vấn.

Khó thu được nhiều chi tiết.

Có thể mất đi các ý tưởng chính.

Khó tạo được mối giao tiếp tốt giữa người phỏng vấn và người được phỏng vấn

2.1.3. Các dạng câu hỏi mở khác

- Các câu hỏi lưỡng cực:

Là những câu hỏi có thể trả lời với các từ “có” hoặc “không” hoặc “đồng ý” hoặc “không đồng ý”.

Các câu hỏi này chỉ nên dùng khi thật cần thiết.

- Các câu hỏi thăm dò:

Các câu hỏi thăm dò gợi ra tính chi tiết hơn về câu hỏi trước đó.

Mục đích của câu hỏi thăm dò là: Thu được nhiều ý nghĩa hơn, Làm sáng rõ, Khai thác và mở rộng các quan điểm của người được phỏng vấn

2.1.4. Thứ tự đặt câu hỏi

- Ba cách cơ bản để cấu trúc cuộc phỏng vấn là:

Dạng 1: mở đầu với các câu hỏi đóng và tiếp tục với các câu hỏi mở



Mở đầu rất chi tiết, thường là bằng các câu hỏi đóng.

Mở rộng bằng các câu hỏi mở và những câu trả lời tổng quát hơn.

Biết người được phỏng vấn cần được tập trung đi vào chủ đề hoặc không hướng tới chủ đề.

Dạng 2: mở đầu với các câu hỏi mở và tiếp tục với các câu hỏi đóng

Mở đầu với các câu hỏi mở, mang tính tổng quát.

Kết thúc bằng cách thu hẹp các câu trả lời có thể có bằng việc sử dụng các câu hỏi đóng.

Cung cấp cách thức dễ dàng, không gây áp lực để bắt đầu một cuộc phỏng vấn.

Có ích khi người được phỏng vấn cảm thấy thích với chủ đề.

Dạng 3: mở đầu với các câu hỏi đóng, tiếp tục với các câu hỏi mở và kết thúc bằng các câu hỏi đóng

Tiếp theo các vấn đề tổng quát hơn được xem xét.

Kết thúc với các câu hỏi cụ thể.

Dạng này kết hợp thế mạnh của cả 2 dạng trên.

Mất nhiều thời gian hơn các dạng khác

2.1.5. Một số việc phải làm khi kết thúc và sau phỏng vấn:

Luôn luôn hỏi “Liệu còn có gì khác mà bạn muốn bổ sung không?”.

Tóm tắt và cung cấp phản hồi về ấn tượng của người phỏng vấn.

Hỏi xem người tiếp theo nên phỏng vấn là ai.

Thiết lập các cuộc hẹn gặp tiếp theo.

Cảm ơn người được phỏng vấn và bắt tay.

Báo cáo phỏng vấn.

Viết càng sớm càng tốt ngay sau khi phỏng vấn.

Cung cấp một bản tóm tắt ban đầu, sau đó thì chi tiết hơn.

Xem lại báo cáo với người được phỏng vấn

2.2. PHƯƠNG PHÁP DÙNG PHIẾU HỎI

- Phiếu hỏi để thu thập thông tin từ các thành viên chủ đạo trong tổ chức về:

Thái độ

Niềm tin

Hành vi

Tính cách

Phiếu hỏi có giá trị nếu:

Các thành viên của tổ chức phân tán rộng

Nhiều thành viên tham gia vào dự án

Cần việc có tính thăm dò

Các câu hỏi được thiết kế theo một trong hai kiểu

Câu hỏi mở



Cố gắng đoán trước câu trả lời sẽ nhận được

Phù hợp để thu được các ý kiến

Câu hỏi đóng

Sử dụng khi tất cả các lựa chọn đều liệt kê được.

Khi các lựa chọn loại trừ lẫn nhau

2.2.1. Thiết kế phiếu hỏi

- Ngôn ngữ dùng trong phiếu hỏi nên:

Đơn giản

Cụ thể

Không thành kiến

Không có vẻ bề trên

Chính xác về mặt kỹ thuật

Hướng đến những người có hiểu biết

Phù hợp với khả năng đọc hiểu của người trả lời

- Phiếu hỏi phải chính xác và đáng tin cậy:

Tính tin cậy thể hiện sự nhất quán trong trả lời – nghĩa là thu được cùng các kết quả nếu như cùng một phiếu hỏi được phân phát trong cùng điều kiện

Tính chính xác là mức độ câu hỏi đo được những gì người phân tích muốn đánh giá.

- Tỷ lệ câu trả lời tốt có thể có được nhờ sự điều chỉnh phù hợp phiếu hỏi

Đề ra nhiều khoảng trống

Bố trí khoảng trống lớn để viết/gõ câu trả lời

Tạo điều kiện cho người trả lời dễ dàng bày tỏ rõ câu trả lời của họ

Nhất quán về hình thức

- Thứ tự câu hỏi:

Đặt các câu hỏi quan trọng nhất lên đầu tiên

Nhóm các câu hỏi có cùng nội dung lại với nhau

Đưa các câu hỏi ít gây tranh luận lên trên

2.2.2. Các phương pháp phát phiếu hỏi

- Tập hợp tất cả những người trả lời vào cùng một thời gian

- Phát phiếu hỏi cho từng cá nhân

- Gửi phiếu hỏi qua đường bưu điện

- Phát phiếu hỏi qua Web hoặc thư điện tử, có các ưu điểm:

Giảm chi phí

Thu thập và lưu trữ các kết quả dễ dàng hơn

- Phiếu hỏi dạng web thường gồm:

Hộp văn bản đơn dòng

Hộp văn bản cuộn, dùng một hoặc nhiều đoạn văn bản



Hộp chọn dành cho các câu trả lời có/không hoặc đúng/sai

Nút tùy chọn cho các câu trả lời mang tính loại trừ lẫn nhau có/không hoặc đúng/sai

Hộp danh sách thả xuống để chọn

Nút Submit (xác nhận) hoặc Reset (xác lập lại)

2.3. PHƯƠNG PHÁP LẤY MẪU

- Lấy mẫu là quá trình lựa chọn một cách có hệ thống các phần tử đại diện của một mẫu. Thay vì nghiên cứu tất cả các thể hiện của các biểu mẫu và bản ghi trong các tệp hoặc cơ sở dữ liệu thì người phân tích chỉ cần sử dụng kỹ thuật lấy mẫu để chọn ra một phần đủ lớn các phần tử đại diện phục vụ cho việc xác định thông tin diễn ra trong hệ thống.
- Bao gồm hai quyết định quan trọng:
 - Những tài liệu và website quan trọng nào nên được lấy mẫu
 - Những người nào nên được phỏng vấn và gửi phiếu hỏi
- Lý do người phân tích cần lấy mẫu là:
 - Giảm chi phí
 - Tăng tốc quá trình thu thập dữ liệu
 - Cải thiện hiệu quả
 - Giảm việc tập trung thu thập dữ liệu

2.3.1. Các phương pháp thiết kế mẫu

- Để thiết kế một mẫu tốt, người phân tích hệ thống cần tuân theo các bước sau:
 - Xác định dữ liệu cần được thu thập hoặc mô tả
 - Xác định tập cần được lấy mẫu
 - Chọn loại mẫu
 - Quyết định kích thước mẫu
- Quyết định kích thước mẫu nên được thực hiện theo những điều kiện cụ thể mà người phân tích hệ thống làm việc:
 - Lấy mẫu dữ liệu trên các thuộc tính
 - Lấy mẫu dữ liệu trên các biến
 - Lấy mẫu dữ liệu định tính

2.3.2. Các kiểu lấy mẫu

- Lấy mẫu tùy ý:
 - Các mẫu không giới hạn, không mang tính xác suất
 - Dễ sắp xếp
 - Không đáng tin cậy nhất
- Lấy mẫu có mục đích:
 - Dựa trên sự đánh giá
 - Người phân tích chọn nhóm các cá nhân để lấy mẫu
 - Dựa trên các tiêu chuẩn



Mẫu không mang tính xác suất

Đáng tin cậy ở mức độ vừa phải

- Lấy mẫu ngẫu nhiên đơn giản:

Dựa trên danh sách các con số của tập lấy mẫu

Mỗi người hoặc tài liệu đều có cơ hội được lựa chọn ngang nhau

- Lấy mẫu ngẫu nhiên phức tạp, có ba hình thức là:

Lấy mẫu có hệ thống

Là phương pháp đơn giản nhất của lấy mẫu theo xác suất

Chọn mọi cá nhân thứ k trong danh sách

Không hay nếu danh sách được sắp thứ tự

Lấy mẫu phân tầng

Là phương pháp quan trọng nhất đối với người phân tích

Xác định các tập lấy mẫu con

Chọn các đối tượng hoặc con người để lấy mẫu từ tập lấy mẫu con

Bù vào số lượng không cân đối các nhân viên trong một nhóm nhất định.

Chọn các phương pháp khác nhau để thu thập dữ liệu từ các nhóm con khác nhau

Lấy mẫu theo nhóm

Chọn nhóm các tài liệu hoặc con người để nghiên cứu

Chọn các nhóm điển hình đại diện cho số còn lại

2.4. PHÂN TÍCH TÀI LIỆU ĐỊNH TÍNH/ ĐỊNH LƯỢNG

2.4.1. Phân tích tài liệu định lượng

- Nghiên cứu dữ liệu cứng là một phương pháp hữu hiệu để người phân tích thu thập thông tin

- Dữ liệu cứng có thể thu thập từ:

Phân tích các tài liệu định lượng như các hồ sơ được sử dụng để ra quyết định

Các báo cáo thực thi

Các hồ sơ

Các mẫu thu thập dữ liệu

Các giao dịch nghiệp vụ

2.4.2. Phân tích tài liệu định tính

- Xem xét các tài liệu định tính để thu được:

Các thông tin tiềm ẩn quan trọng

Trạng thái tâm lý

Những gì được coi là tốt/xấu

Hình ảnh, logo, biểu tượng

- Tài liệu định tính bao gồm:

Các bản ghi nhớ



Dấu hiệu trên các bản tin Website của tổ chức

Các tài liệu chỉ dẫn

Sổ tay về chính sách của tổ chức

2.5. PHƯƠNG PHÁP QUAN SÁT

- Việc quan sát cung cấp sự hiểu biết về những gì các thành viên của tổ chức thực sự đang làm
- Nhìn nhận trực tiếp các quan hệ tồn tại giữa những người ra quyết định và các thành viên khác của tổ chức
- Kỹ thuật quan sát phân tích bảy phần tử môi trường:
 - Vị trí văn phòng
 - Vị trí bàn làm việc
 - Thiết bị văn phòng
 - Tài sản
 - Các nguồn thông tin bên ngoài
 - Màu sắc và ánh sáng văn phòng
 - Trang phục của người ra quyết định

CÂU HỎI THẢO LUẬN

- 2.1. Nêu các mục tiêu của việc phỏng vấn.
- 2.2. Kể tên các dạng câu hỏi có thể dùng trong quá trình phỏng vấn.
- 2.3. Nêu ý nghĩa của việc lấy mẫu.





BÀI 3. PHÂN TÍCH HỆ THỐNG

3.1. KHÁI NIỆM PHÂN TÍCH HỆ THỐNG

3.1.1. Phân tích hệ thống (System Analysis):

Là giai đoạn phát triển trong một dự án, tập trung vào các vấn đề nghiệp vụ, ví dụ như những gì hệ thống phải làm về mặt dữ liệu, các thủ tục xử lý và giao diện, độc lập với kỹ thuật có thể được dùng để cài đặt giải pháp cho vấn đề đó.

3.1.2. Thiết kế hệ thống (System Design):

Là giai đoạn phát triển tập trung vào việc xây dựng và cài đặt mang tính kỹ thuật của hệ thống (cách thức mà công nghệ sẽ được sử dụng trong hệ thống).

3.2. CÁC HƯỚNG TIẾP CẬN PHÂN TÍCH HỆ THỐNG

3.2.1. Tiếp cận phân tích hướng mô hình

- Nhấn mạnh việc vẽ các mô hình hệ thống dạng đồ họa để tài liệu hóa và kiểm tra hệ thống hiện tại cũng như hệ thống được đề xuất.
- Cuối cùng thì mô hình hệ thống trở thành bản thiết kế chi tiết cho việc thiết kế và xây dựng một hệ thống được cải thiện.
- Phân tích hướng cấu trúc (Structured Analysis): là kiểu phân tích hướng mô hình, là một kỹ thuật lấy quá trình làm trung tâm để phân tích một hệ thống đang có và xác định các yêu cầu nghiệp vụ cho hệ thống mới.

Chú ý: Đưa ra mô hình minh họa các thành phần của hệ thống: các quá trình (các chức năng, thao tác) và những thành phần liên quan là đầu vào, đầu ra và các file.

3.2.2. Tiếp cận phân tích hệ thống nhanh

- Nhấn mạnh việc xây dựng các bản mẫu thử để xác định nhanh các yêu cầu nghiệp vụ và của người dùng đối với một hệ thống mới.
- Xây dựng bản mẫu tìm hiểu một kỹ thuật dùng để xác định các yêu cầu nghiệp vụ của người dùng bằng cách để họ phản ứng với một bản cài đặt nhanh và thô của các yêu cầu đó.

3.3. CÁC GIAI ĐOẠN PHÂN TÍCH HỆ THỐNG

3.3.1. Giai đoạn xác định phạm vi

Bước 1.1: xác định các vấn đề, cơ hội và yếu tố chi phối theo các tiêu chí sau:

Tính khẩn cấp: trong khoảng thời gian nào thì vấn đề cần được giải quyết hoặc cơ hội hoặc yếu tố chi phối cần được nhận ra?.

Tính rõ ràng: Mức độ thấy được của của một giải pháp hoặc hệ thống mới đối với khách hàng hoặc người quản lý điều hành?.



Tính hữu ích: Một hệ thống mới hoặc giải pháp có thể tăng lợi nhuận hoặc giảm chi phí hàng năm lên/xuống bao nhiêu?

Tính ưu tiên: dựa vào những câu trả lời trên, mức ưu tiên giữa các vấn đề, cơ hội và yếu tố chi phối là như thế nào?

Giải pháp khả thi: vào giai đoạn đầu của dự án, giải pháp khả thi có thể diễn đạt ở dạng giản đơn sau: Đề nguyên, Sửa nhanh, Thay đổi đơn giản để củng cố hệ thống hiện có, Thiết kế lại hệ thống hiện có, Thiết kế một hệ thống mới.

Bước 1.2: Thảo luận sơ bộ phạm vi

Kết quả: Báo cáo phạm vi dự án (giới hạn của dự án).

Những loại dữ liệu nào cần nghiên cứu.

Những quy trình nghiệp vụ nào cần đưa vào.

Hệ thống giao tiếp như thế nào với người dùng và các hệ thống khác.

Chú ý: nếu sau này phạm thay đổi thì ngân sách và lịch biểu cũng nên được thay đổi phù hợp.

Bước 1.3: Đánh giá tính khả thi của dự án

“Liệu dự án này có đáng được xem xét?”

Phân tích chi phí / lợi ích.

Quyết định.

Phê duyệt dự án.

Hủy bỏ dự án.

Xem xét lại phạm vi dự án (với ngân sách và lịch biểu đã được điều chỉnh).

Bước 1.4: lập biểu và lập kế hoạch ngân sách cho dự án

Kết quả: báo cáo dự án.

Lập kế hoạch thực hiện cho toàn bộ dự án: lập biểu và phân bổ tài nguyên, bố trí nhân sự.

Bước 1.5: Trình bày dự án và kế hoạch

Trình bày và bảo vệ dự án, kế hoạch trước hội đồng thẩm định.

Khởi đầu chính thức dự án và thông báo về dự án, các mục tiêu và lịch biểu cho dự án.

3.3.2. Giai đoạn phân tích vấn đề

Bước 2.1: Nghiên cứu lĩnh vực vấn đề

Tìm hiểu lĩnh vực của vấn đề và các thuật ngữ nghiệp vụ.

Dữ liệu: dữ liệu đang được lưu trữ, các thuật ngữ nghiệp vụ.

Các quá trình: các sự kiện nghiệp vụ hiện có.

Các giao diện: các vị trí và người dùng hiện tại.

Kết quả: xác định về lĩnh vực hệ thống / các mô hình của các hệ thống hiện có.

Bước 2.2: Phân tích các vấn đề và cơ hội



Nghiên cứu các nguyên nhân và hệ quả của từng vấn đề (chú ý: một hệ quả có thể lại là nguyên nhân của những vấn đề khác).

Kết quả: các báo cáo vấn đề được cập nhật và các phân tích nguyên nhân-hệ quả của từng vấn đề và cơ hội.

Bước 2.3: Phân tích các quá trình nghiệp vụ (chỉ danh cho việc tái cấu trúc quy trình nghiệp vụ)

Đánh giá giá trị gia tăng hoặc giảm bớt của các quá trình đối với toàn bộ tổ chức.

Số lượng đầu vào, thời gian đáp ứng, các khâu đình trệ, chi phí, giá trị gia tăng, các hệ quả của việc loại bỏ hoặc hợp lý hóa quá trình.

Kết quả: các mô hình quá trình nghiệp vụ hiện tại.

Bước 2.4: Xác lập các mục tiêu cải thiện hệ thống

Xác định các mục tiêu cụ thể cải thiện hệ thống và các ràng buộc đối với mỗi vấn đề.

Các mục tiêu phải chính xác, có thể đo được.

Các ràng buộc về lịch biểu, chi phí, công nghệ và chính sách.

Kết quả: các mục tiêu cải thiện hệ thống và báo cáo đề xuất.

Bước 2.5: Cập nhật kế hoạch dự án

Cập nhật dự án:

Thu hẹp phạm vi, chỉ giữ những mục tiêu ưu tiên cao để phù hợp với thời hạn/ngân sách.

Mở rộng phạm vi và điều chỉnh lịch biểu và ngân sách phù hợp.

Kết quả: kế hoạch dự án đã được cập nhật.

Bước 2.6: trình bày các nhận xét và đề xuất

Kết quả: các mục tiêu cải thiện hệ thống.

Quyết định: tiếp tục/điều chỉnh/hủy bỏ dự án hiện tại.

3.3.3. Giai đoạn phân tích yêu cầu

Bước 3.1: xác định các yêu cầu hệ thống

Các yêu cầu chức năng: các hoạt động và dịch vụ cung cấp bởi hệ thống: các chức năng nghiệp vụ, các đầu vào, đầu ra, dữ liệu được lưu trữ.

Các yêu cầu phi chức năng: các đặc trưng, đặc điểm xác định một hệ thống thỏa đáng: hiệu suất, tài liệu, ngân sách, tính dễ học và sử dụng, tiết kiệm chi phí, tiết kiệm thời gian, an toàn.

Kết quả: phác thảo các yêu cầu chức năng và phi chức năng: các mục tiêu cải thiện và đầu vào, đầu ra, các quá trình, dữ liệu được lưu trữ liên quan để đạt được mục tiêu.

Bước 3.2: Phân mức ưu tiên cho các yêu cầu

Các yêu cầu mang tính bắt buộc có ưu tiên cao hơn các yêu cầu khác.

Time boxing: đưa ra hệ thống dưới dạng một tập các phiên bản kế tiếp nhau trong một khoảng thời gian.



Phiên bản đầu tiên đáp ứng các yêu cầu thiết yếu và có mức ưu tiên cao nhất.

Bước 3.3: Cập nhật kế hoạch dự án

Nếu các yêu cầu vượt quá phiên bản đầu tiên: thu hẹp phạm vi hoặc tăng ngân sách

Kết quả: các yêu cầu hệ thống đã được thống nhất (các yêu cầu và mức ưu tiên đã được bổ sung)

3.3.4. Giai đoạn mô hình hóa logic

Bước 4.1: Phân tích các yêu cầu mang tính chức năng

Các mô hình hệ thống logic: hệ thống phải làm gì (chứ không phải làm như thế nào).

Xây dựng các bản mẫu để xác lập các yêu cầu giao diện người dùng. Kết quả: các mô hình dữ liệu (ERD), các mô hình quá trình (DFD), các mô hình giao diện (biểu đồ ngữ cảnh, biểu đồ Use case), các mô hình đối tượng (các biểu đồ UML) của hệ thống được đề xuất.

Bước 4.2: Kiểm tra các yêu cầu mang tính chức năng

Kiểm tra tính đầy đủ, xem xét lại, thực hiện các thay đổi và bổ sung đối với các mô hình hệ thống và các bản mẫu để đảm bảo rằng các yêu cầu đã được xác định thỏa đáng.

Liên kết các yêu cầu phi chức năng với các yêu cầu mang tính chức năng.

3.3.5. Giai đoạn phân tích quyết định

Là giai đoạn chuyển tiếp giữa phân tích hệ thống và thiết kế hệ thống.

Bước 5.1: xác định các giải pháp đề cử

Xác định tất cả các giải pháp đề cử có thể có.

Kết quả: ma trận các hệ thống (giải pháp) đề cử

Bước 5.2: Phân tích các giải pháp đề cử

Việc phân tích tính khả thi được thực hiện với từng đề cử mà không quan tâm tới tính khả thi của các đề cử khác.

Phân tích tính khả thi:

Tính khả thi về kỹ thuật: Liệu giải pháp có phù hợp với thực tế công nghệ? Liệu đội ngũ dự án có chuyên gia kỹ thuật để thiết kế và xây dựng giải pháp?.

Tính khả thi về hoạt động: Liệu giải pháp có thực hiện được yêu cầu của người dùng? Ở mức độ nào? Giải pháp sẽ thay đổi môi trường làm việc của người dùng như thế nào? Người dùng sẽ cảm thấy như thế nào về giải pháp như vậy?.

Tính khả thi về kinh tế: Liệu giải pháp có chi phí hiệu quả?.

Tính khả thi lịch biểu: Liệu giải pháp có thể được thiết kế và xây dựng trong một khoảng thời gian chấp nhận được hay không?

Bước 5.3: So sánh các giải pháp đề cử



Chọn giải pháp đề cử có sự kết hợp “toàn diện tốt nhất” của các tính khả thi về kỹ thuật, hoạt động, kinh tế và lịch biểu Ma trận tính khả thi.

Kết quả: giải pháp được đề xuất.

Bước 5.4: Cập nhật kế hoạch dự án

Đầu vào: giải pháp đề xuất

Xem xét và cập nhật lịch biểu mới nhất của dự án và phân bổ tài nguyên.

Kết quả: cập nhật kế hoạch dự án.

Bước 5.5: đề xuất một giải pháp

Kết quả: đề xuất dự án

3.4. XÁC ĐỊNH CÁC YÊU CẦU CỦA NGƯỜI DÙNG

3.4.1. Vai trò của việc xác định yêu cầu

- **Yêu cầu hệ thống** (yêu cầu nghiệp vụ) là một mô tả các nhu cầu và mong muốn đối với một hệ thống thông tin. Một yêu cầu có thể mô tả các chức năng, đặc trưng (thuộc tính) và các ràng buộc.
- **Các yêu cầu mang tính chức năng:** các chức năng hoặc đặc trưng có thể có trong một hệ thống thông tin để nó thỏa mãn nhu cầu nghiệp vụ và có thể chấp nhận được đối với người dùng
- **Các yêu cầu phi chức năng:** các đặc trưng, đặc điểm và thuộc tính của các hệ thống cũng như bất kỳ các ràng buộc nào có thể giới hạn ranh giới của giải pháp được đề xuất.

Chú ý: Xác định các yêu cầu không chính xác dẫn đến một số hậu quả sau:

- Hệ thống có thể tốn nhiều chi phí hơn.
- Hệ thống có thể hoàn thành muộn hơn thời gian đã định.
- Hệ thống có thể không phù hợp với những gì người dùng mong muốn và sự không hài lòng đó có thể khiến họ không sử dụng hệ thống.
- Chi phí bảo trì và nâng cấp hệ thống có thể quá cao.
- Hệ thống có thể không chắc chắn và dễ có lỗi và ngừng hoạt động.
- Uy tín của các chuyên gia trong đội dự án có thể bị giảm sút bởi bất kỳ thất bại nào, cho dù là do ai gây ra thì cũng sẽ bị xem là lỗi của cả đội dự án

3.4.2. Tiêu chuẩn của việc xác định yêu cầu

- **Nhất quán:** các yêu cầu không mâu thuẫn hay nhập nhằng lẫn nhau.
- **Toàn diện:** các yêu cầu mô tả mọi đầu vào và đáp ứng có thể có của hệ thống.
- **Khả thi:** các yêu cầu có thể được thỏa mãn dựa trên các tài nguyên và ràng buộc sẵn có.
- **Cần thiết:** các yêu cầu là thực sự cần thiết và đáp ứng mục đích của hệ thống.
- **Chính xác:** các yêu cầu được phát biểu chính xác.



- **Dễ theo dõi:** các yêu cầu liên kết trực tiếp tới các chức năng và đặc trưng của hệ thống.
- **Có thể kiểm tra:** các yêu cầu đã được chứng minh.

3.4.3. Quy trình xác định yêu cầu

- Phân tích các yêu cầu để giải quyết các vấn đề về: yêu cầu bị thiếu, mâu thuẫn nhau, không khả thi, trùng lặp, mơ hồ,...
- Chính thức hóa các yêu cầu: lập tài liệu yêu cầu, truyền đạt đến các nhân sự tham gia. Việc lập tài liệu yêu cầu gồm: các chức năng và dịch vụ mà hệ thống có thể cung cấp.

CÂU HỎI THẢO LUẬN

- 3.1. Phân biệt Phân tích hệ thống và Thiết kế hệ thống.
- 3.2. Thế nào là kỹ thuật phân tích hướng cấu trúc.
- 3.3. Nêu và phân tích các giai đoạn phân tích hệ thống.





BÀI 4. MÔ HÌNH HÓA CHỨC NĂNG

Tóm tắt: Bài này tập trung vào việc mô hình hóa các chức năng trong quá trình phân tích. Nội dung bao gồm khái niệm và các nguyên tắc xây dựng biểu đồ phân cấp chức năng và biểu đồ luồng dữ liệu.

4.1. MÔ HÌNH HÓA HỆ THỐNG

4.1.1. Các bước mô hình hóa hệ thống

Trong bài 3, chúng ta đã biết về các hoạt động phân tích hệ thống, những hoạt động đó là nhằm mục đích vẽ các mô hình hệ thống. Các mô hình hệ thống đóng vai trò quan trọng trong phát triển hệ thống. Dù là người sử dụng hay người phân tích hệ thống thì bạn đều phải giải quyết những vấn đề phi cấu trúc. Và một cách để cấu trúc vấn đề là vẽ các mô hình.

Mô hình là một biểu diễn hình tượng của thực tế. Các mô hình có thể được xây dựng cho các hệ thống hiện có để giúp chúng ta hiểu kỹ hơn về những hệ thống đó. Hoặc cũng có thể xây dựng mô hình cho các hệ thống được đề xuất nhằm tài liệu hóa các yêu cầu nghiệp vụ hoặc thiết kế kỹ thuật.

Mô hình hóa chức năng (Process Modeling) với biểu đồ luồng dữ liệu (Data Flow Diagram - DFD) nhằm mục đích trả lời câu hỏi: Hệ thống làm gì? Mô hình hóa chức năng là kỹ thuật dùng để tổ chức và tài liệu hóa cấu trúc và luồng dữ liệu xuyên qua các quá trình của một hệ thống hoặc các chức năng được thực hiện bởi các quá trình hệ thống.

Mô hình hóa dữ liệu (Data Modeling) với biểu đồ quan hệ thực thể (Entity Relationship Diagram - ERD) nhằm trả lời câu hỏi: Hệ thống có những dữ liệu nào? Mô hình hóa dữ liệu là kỹ thuật dùng để tổ chức và mô hình hóa dữ liệu của một hệ thống nhằm xác định các yêu cầu nghiệp vụ cho một cơ sở dữ liệu. Đôi khi mô hình hóa dữ liệu còn được gọi là mô hình hóa cơ sở dữ liệu.

Mô hình hóa đối tượng (Object Modeling) với ngôn ngữ mô hình hợp nhất (Unified Modeling Language - UML) nhằm trả lời câu hỏi: Cái gì và tại sao? (lôgic của hệ thống)

4.1.2. Mục đích của mô hình hóa hệ thống

Để hiểu rõ hơn về hệ thống: các cơ hội để đơn giản hóa, tối ưu hóa (Tái cấu trúc quy trình).

Để liên kết các hành vi và cấu trúc của hệ thống (các yêu cầu nghiệp vụ về: thông tin/dữ liệu và chức năng/quy trình).

Để trực quan hóa và điều khiển kiến trúc hệ thống (thiết kế).

Để kiểm soát những rủi ro trong quá trình phát triển



4.1.3. Các thao tác mô hình hóa chức năng

4.1.3.1. Lập kế hoạch chiến lược hệ thống:

Các mô hình quá trình nghiệp vụ của tổ chức mô tả các chức năng nghiệp vụ quan trọng

4.1.3.2. Tái cấu trúc quy trình nghiệp vụ:

Các mô hình chức năng “As is” làm đơn giản việc phân tích các điểm yếu (Hệ thống hiện tại).

Các mô hình chức năng “To be” làm đơn giản việc cải thiện (Hệ thống mới được đề xuất).

4.1.3.3. Phân tích hệ thống

Mô hình hóa hệ thống hiện có bao gồm những thiếu sót của nó (DFD logic)

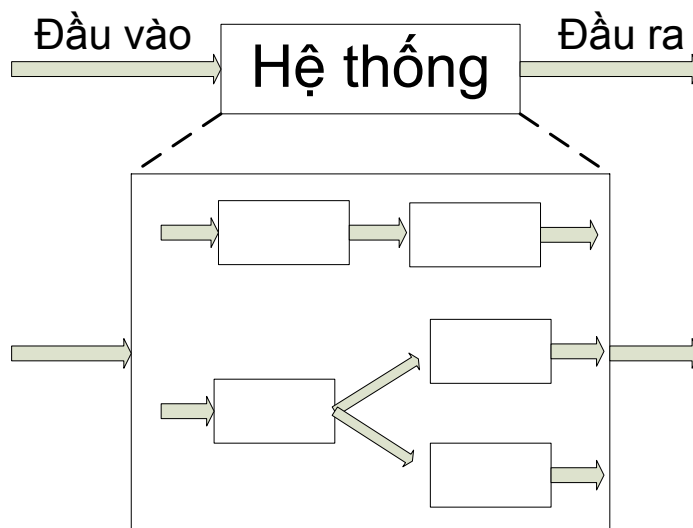
Mô hình hóa các yêu cầu logic (các quá trình và luồng dữ liệu cần có dù hệ thống được xây dựng thế nào – DFD logic) của hệ thống được đề xuất.

Mô hình hóa các giải pháp kỹ thuật đề cử (DFD vật lý)

Mô hình hóa giải pháp được chọn (DFD vật lý)

4.1.4. Khái niệm hệ thống

Một hệ thống tồn tại bằng việc lấy đầu vào từ môi trường, biến đổi (xử lý) đầu vào này và tạo ra một đầu ra. Một hệ thống có thể được phân rã thành nhiều hệ thống con. Một hệ thống con có đầu vào và đầu ra của riêng nó. Đầu ra của một hệ thống con có thể trở thành đầu vào của những hệ thống con khác.



Hình 4.1. Hệ thống và Hệ thống con



4.2. MÔ HÌNH LOGIC

4.2.1. Phân biệt mô hình logic và mô hình vật lý

Mô hình logic cho biết hệ thống là gì và làm gì. Nó độc lập với việc cài đặt kỹ thuật. Nó minh họa bản chất của hệ thống. Mô hình logic còn có thể được gọi là mô hình bản chất, mô hình khái niệm mô hình nghiệp vụ.

Mô hình vật lý không chỉ thể hiện hệ thống là gì và làm gì mà còn thể hiện cách thức hệ thống được cài đặt một cách vật lý và kỹ thuật. Nó phản ánh các lựa chọn công nghệ. Mô hình vật lý còn có thể được gọi là mô hình cài đặt hay mô hình kỹ thuật.

4.2.2. Sự cần thiết của mô hình logic

Các nhà phân tích hệ thống đã nhận thấy giá trị của việc tách riêng việc nghiên cứu nghiệp vụ với việc nghiên cứu kỹ thuật. Đó là lý do tại sao họ sử dụng các mô hình hệ thống logic để minh họa các yêu cầu nghiệp vụ và các mô hình hệ thống vật lý để minh họa các thiết kế kỹ thuật.

Các hoạt động của người phân tích hệ thống tập trung chủ yếu vào các mô hình hệ thống logic vì những lý do sau:

- Các mô hình logic loại bỏ tư tưởng thiên lệch do ảnh hưởng bởi cách thức cài đặt hệ thống đã có hoặc ý kiến chủ quan của một người nào đó về cách cài đặt cho hệ thống. Do đó, chúng khuyến khích tính sáng tạo.
- Các mô hình logic làm giảm khả năng bỏ sót các yêu cầu nghiệp vụ trong trường hợp con người bị chi phối quá nhiều vì các kết quả mang tính kỹ thuật. Nhờ việc tách biệt những gì hệ thống phải làm với cách thức hệ thống thực hiện mà chúng ta có thể phân tích tốt hơn các yêu cầu nhằm đảm bảo tính hoàn thiện, chính xác và nhất quán.
- Các mô hình logic cho phép truyền đạt với người dùng cuối dưới dạng ngôn ngữ phi kỹ thuật hoặc ít kỹ thuật hơn.

4.3. BIỂU ĐỒ PHÂN RÃ CHỨC NĂNG (Business Functional Digram)

4.3.1. Khái niệm BFD

BFD là sơ đồ phân rã có thứ bậc các chức năng của hệ thống từ tổng thể đến chi tiết. Mỗi chức năng có thể có một hoặc nhiều chức năng con, tất cả được thể hiện trong một khung của sơ đồ.

Ý nghĩa của BFD:

- Giới hạn phạm vi của hệ thống cần phải phân tích.
- Tiếp cận hệ thống về mặt logic nhằm làm rõ các chức năng mà hệ thống thực hiện để phục vụ cho các bước phân tích tiếp theo.
- Phân biệt các chức năng và nhiệm vụ của từng bộ phận trong hệ thống, từ đó loại bỏ những chức năng trùng lặp, dư thừa.

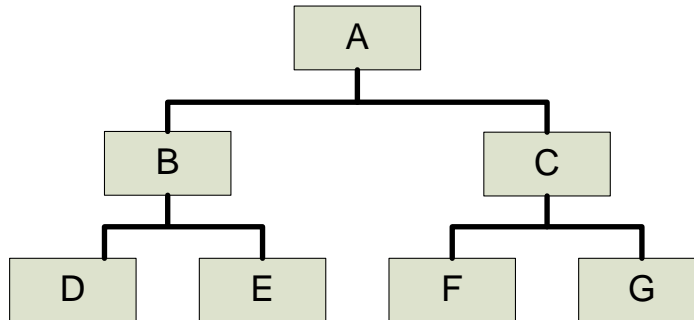
Tuy nhiên, BFD không có tính động, nó chỉ cho thấy các chức năng mà không thể hiện trình tự xử lý của các chức năng đó cũng như là sự trao đổi thông tin giữa các chức năng. Do đó, BFD thường được sử dụng làm mô hình chức năng trong bước đầu phân tích.



4.3.2. Phương pháp xây dựng BFD

4.3.2.1. Phân mức các chức năng:

BFD được xây dựng theo mô hình phân mức, mỗi một chức năng có thể gồm một hoặc nhiều chức năng con trong sơ đồ.



Hình 4.2. Mô hình BFD phân mức

Nhìn mô hình trên, dễ thấy chức năng A bao gồm chức năng B và C; chức năng B lại bao gồm D và E.

Chú ý là với một sơ đồ, không nên có quá 6 mức, thông thường thì 3 mức là phù hợp với các hệ thống trung bình. Với mỗi chức năng không nên có quá 6 chức năng con vì như vậy sẽ làm sơ đồ trở nên phức tạp và khó kiểm soát.

Nếu gặp trường hợp có quá nhiều chức năng con thì có thể giải quyết bằng cách tạo thêm mức trung gian để nhóm các chức năng con lại. Cần đảm bảo tính cân bằng của sơ đồ, nghĩa là các chức năng thuộc cùng một mức nên có sự tương đương nhau về kích thước và độ phức tạp

Mỗi chức năng phải mang một tên duy nhất, không trùng lặp với chức năng khác; tên phải thể hiện khái quát các chức năng con của nó, phản ánh được thực tế nghiệp vụ mà nó thực hiện. Tên của chức năng phải bắt đầu bằng động từ, ví dụ như “lập đơn hàng”.

4.3.2.2. Xác định các chức năng.

Ở mức cao nhất của nghiệp vụ, chức năng chính có thể là một trong các loại sau:

- Sản xuất sản phẩm.
- Cung cấp dịch vụ (bán hàng, bảo dưỡng).
- Quản lý tài nguyên (tài sản, nguồn nhân lực, con người...).

Khi đã xác định được loại mà nó thuộc vào thì sẽ đặt tên cho chức năng cao nhất này. Tiếp theo, để xác định các chức năng con thì từ chức năng chính, ta đặt nó trong chu kỳ sống gồm các giai đoạn:

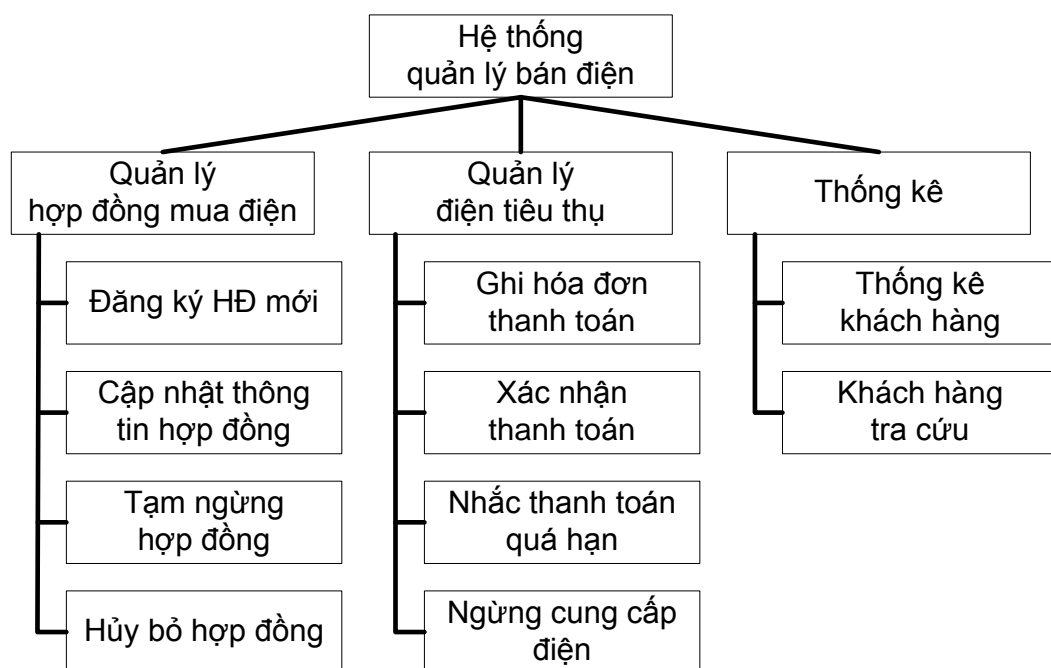
- Xác định nhu cầu.
- Mua bán.
- Bảo hành, bảo dưỡng.
- Thanh lý hoặc chuyển nhượng.



Mỗi giai đoạn có thể có một hoặc nhiều chức năng con. Ví dụ, với chức năng Bán hàng thì ở giai đoạn xác định nhu cầu có thể có chức năng con là Quản lý thông tin khách hàng, ở giai đoạn mua bán thì có thể là Cập nhật đơn hàng...

Người phân tích phải xác định được mức nào là thấp nhất, khi đó sẽ dừng việc phân tích chức năng. Để nhận biết một chức năng mức thấp nhất bằng cách xét xem có phải chức năng đó chỉ có một nhiệm vụ hoặc một nhóm các nhiệm vụ nhỏ.

Khi xây dựng BFD cần đảm bảo tính đơn giản, rõ ràng và chính xác của sơ đồ. Với các hệ thống lớn, có thể trình bày BFD trên nhiều trang, trang 1 là BFD mức cao nhất (mức 0), tiếp theo ứng với mỗi chức năng sẽ được phân tích ở các trang sau tới chức năng mức thấp nhất thì dừng.



Hình 4.3. Ví dụ biểu đồ phân cấp chức năng Hệ thống quản lý bán điện

4.4. BIỂU ĐỒ LUỒNG DỮ LIỆU (Data Flow Diagram)

4.4.1. Mô hình hóa chức năng với DFD

Trong chương này, chúng ta tập trung vào việc mô hình hóa chức năng logic trong giai đoạn phân tích hệ thống. Như trên đã nói, mô hình hóa chức năng là một kỹ thuật để tổ chức và tài liệu hóa cấu trúc và luồng dữ liệu cũng như logic, đường lối và các thủ tục được thực hiện bởi các quá trình của một hệ thống. Một trong các mô hình chức năng phân tích hệ thống chính là sơ đồ luồng dữ liệu.

Một sơ đồ luồng dữ liệu (Data Flow Diagram – DFD) là một công cụ đồ họa để mô tả luồng dữ liệu luân chuyển trong một hệ thống và những hoạt động xử lý được thực hiện bởi hệ thống đó. Sơ đồ luồng dữ liệu còn có các tên gọi khác là *biểu đồ bọt*, *biểu đồ biến đổi* và *mô hình chức năng*.



Tại sao sử dụng DFD?

- Sự mô tả bằng ngôn ngữ hướng tới sự giải thích, nó có thể bỏ sót những thông tin quan trọng.
- Sự mô tả đồ họa minh họa được luồng dữ liệu trong một tổ chức thông qua DFD.

Biểu đồ luồng dữ liệu đã rất phổ biến hơn 20 năm nay nhưng lợi ích của DFD đã được đổi mới nhờ vào tính ứng dụng của nó trong việc tái cấu trúc quy trình nghiệp vụ (business process redesign – BPR). Khi mà tổ chức nhận thấy rằng hầu hết các hệ thống xử lý dữ liệu đã trở nên lỗi thời, không hiệu quả và rườm rà về thủ tục thì đó là lúc có thể thu lợi ích mới nhờ vào việc tổ chức lại các quy trình nghiệp vụ. Điều này được tiến hành trước tiên bằng việc mô hình hóa các quy trình nghiệp vụ nhằm mục đích phân tích, thiết kế lại và/hoặc cải thiện chúng. Tiếp theo, công nghệ thông tin có thể được áp dụng một cách sáng tạo cho các quy trình nghiệp vụ đã được cải thiện nhằm tối đa hóa giá trị thu về cho tổ chức.

4.4.2. Vai trò của DFD

DFD tài liệu hóa một thao tác/hoạt động/chức năng nghiệp vụ của một hệ thống thành một quá trình. DFD mô tả cách thức dữ liệu được xử lý trong và tại biên giới của hệ thống. DFD thể hiện chi tiết sự phụ thuộc lẫn nhau giữa các quá trình của hệ thống, các sự dịch chuyển dữ liệu hoặc thông tin giữa các quá trình.

DFD logic mô tả luồng thông tin của một hệ thống.

DFD vật lý mô tả cách thức một hệ thống thông tin được cài đặt vật lý (ai làm, bằng cách nào, bằng công cụ nào).

Mô tả cách thức một hệ thống thông tin được cài đặt vật lý (ai làm, bằng cách nào, bằng công cụ nào)

4.5. CÁC PHẦN TỬ CỦA DFD

4.5.1. Các bộ ký hiệu của DFD

Có vài bộ ký hiệu DFD mà phổ biến là Gane/Sarson và Demarco/Yourdon. Trong đó, bộ ký hiệu Gane & Sarson được sử dụng phổ biến hơn.

	Gane/Sarson	Demarco/Yourdon
Tác nhân ngoài		
Luồng dữ liệu		
Quá trình		
Kho dữ liệu		

Hình 4.4. Hai bộ ký hiệu DFD của Gane/Sarson và Demarco/Yourdon



4.5.2. Tác nhân ngoài

Khái niệm:

Một tác nhân ngoài là một nguồn cung cấp hoặc nhận thông tin dữ liệu của hệ thống.

Một tác nhân ngoài không phải là một phần của hệ thống, nó thể hiện mối quan hệ giữa hệ thống với môi trường bên ngoài.

Nhãn: Tên của tác nhân ngoài phải là một **danh từ**.

Một tác nhân ngoài xác định một người, một đơn vị của tổ chức hay một tổ chức khác nằm ngoài phạm vi của dự án nhưng có tương tác với hệ thống đang được nghiên cứu.

Các tác nhân ngoài xác định “biên giới” hay phạm vi của hệ thống đang được mô hình hóa. Khi phạm vi thay đổi, các tác nhân ngoài có thể trở thành các quá trình và ngược lại

Tác nhân ngoài thường là:

Phòng ban, bộ phận trong tổ chức nhưng nằm ngoài phạm vi hệ thống.

Một chi nhánh hoặc tổ chức bên ngoài

Một hệ thống thông tin khác của tổ chức

Người dùng cuối hoặc người quản lý của hệ thống

Nhà cung cấp

Phòng kế toán

4.5.3. Kho dữ liệu

Khái niệm:

- Một kho dữ liệu là một kho lưu trữ dữ liệu, nó chứa thông tin.

- Kho chứa vật lý là phi vật chất, nó có thể là một tủ hồ sơ, sách hoặc file máy tính.

Nhãn: Tên của kho dữ liệu phải bắt đầu bằng **danh từ**, nó nói lên nội dung thông tin.

Một kho dữ liệu là “dữ liệu tĩnh” khác với luồng dữ liệu là “dữ liệu chuyển động”. Một kho dữ liệu cần biểu diễn cho “những thứ” mà tổ chức muốn lưu trữ dữ liệu, “những thứ” thường là:

- Con người: ví dụ như khách hàng, phòng, nhân viên, thầy giáo, sinh viên, nhà cung cấp...

- Các địa điểm: ví dụ như nơi sinh, tòa nhà, phòng, chi nhánh...

- Các đối tượng: ví dụ như sách, máy móc, sản phẩm, nguyên liệu thô, bản quyền phần mềm, gói phần mềm, công cụ, phương tiện vận tải...

- Các sự kiện (dữ liệu được thu thập về chúng): ví dụ như việc bán hàng, giải thưởng, sự trì hoãn, lớp học, chuyến bay, hóa đơn, đơn hàng, đăng ký, đặt chỗ...

- Các khái niệm (dữ liệu về chúng rất quan trọng): ví dụ như việc giảm giá, tài khoản, khóa học, chất lượng...



Có thể xác định các kho dữ liệu với các yếu tố Tài nguyên – Sự kiện – Tác nhân – Địa điểm. Các kho dữ liệu được mô tả trong một DFD chứa tất cả các thể hiện của các thực thể dữ liệu (được mô tả trong một biểu đồ quan hệ quan hệ ERD).

Danh mục hàng hóa

Tài khoản phải thu

4.5.4. Luồng dữ liệu

Khái niệm:

- Một luồng dữ liệu biểu diễn một **sự di chuyển của dữ liệu (thông tin)** giữa các quá trình hoặc kho dữ liệu.
- Một luồng dữ liệu **không** biểu diễn một tài liệu hay một vật thể vật lý: nó biểu diễn sự trao đổi **thông tin** trong tài liệu hoặc về vật thể. Nhận: Phải có tên và không trùng lặp với các luồng dữ liệu khác. Tên phải thể hiện logic của thông tin chứ không phải dạng vật lý của nó và phải bắt đầu bằng **danh từ**.
- Một luồng dữ liệu biểu diễn một đầu vào dữ liệu tới một quá trình hoặc đầu ra dữ liệu từ một quá trình.
- Một luồng dữ liệu cũng có thể được dùng để biểu diễn việc tạo, đọc, xóa hoặc cập nhật dữ liệu trong một file hoặc cơ sở dữ liệu (được gọi là kho dữ liệu).
- Một luồng dữ liệu ghép (gói) là một luồng dữ liệu chứa các luồng dữ liệu khác.

Số tiền phải trả →

4.5.5. Quá trình

Khái niệm:

- Một quá trình là một hoạt động được thực hiện trên luồng dữ liệu vào để tạo một luồng dữ liệu ra.
- Là chức năng được thực hiện bởi hệ thống để đáp ứng lại các luồng dữ liệu hoặc điều kiện vào. Nhận: Sử dụng một **động từ** để gán nhãn cho hành động thực hiện bởi quá trình (không phải là tên của người hay phòng ban thực hiện nó trong DFD vật lý).
- Một quá trình phải có **ít nhất một** luồng dữ liệu **vào** và **ít nhất một** luồng dữ liệu **ra**.

Thanh toán
Hóa đơn

Các loại quá trình logic:

- Chức năng: một tập các hoạt động tiếp diễn liên quan tới một nghiệp vụ; ví dụ như việc bán hàng



- Sự kiện (hay giao dịch, hoạt động): một công việc phải được hoàn thành toàn bộ (hoặc như một phần của một chức năng); ví dụ như việc thu tiền thanh toán (là một công đoạn trong việc bán hàng)
- Quá trình cơ bản (hay thao tác): một hoạt động hoặc thao tác chi tiết, rời rạc được yêu cầu để đáp lại một sự kiện. Thông thường, một số thao tác như vậy phải được hoàn thành để đáp ứng một sự kiện; ví dụ như ghi tiền thanh toán

Cách tách các quá trình:

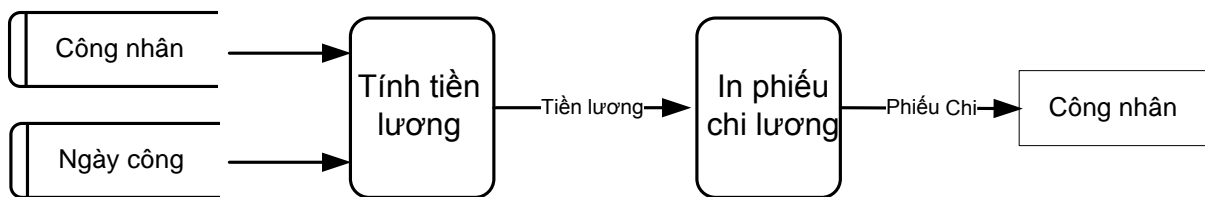
- Mỗi hệ thống có thể được chia thành nhiều quá trình khác nhau bằng các cách khác nhau. Các quá trình có thể được tách nếu có một luồng thông tin đi giữa chúng.
- Điều kiện để tách: nếu các tiến trình này không thực hiện đồng thời hoặc không cùng một nơi hoặc không do một người thực hiện. Khi đó, ta kiểm tra quá trình tách bằng cách cuối luồng dữ liệu ta đặt câu hỏi:

Tiến trình tiếp theo có thể thực hiện ở thời gian khác được không?

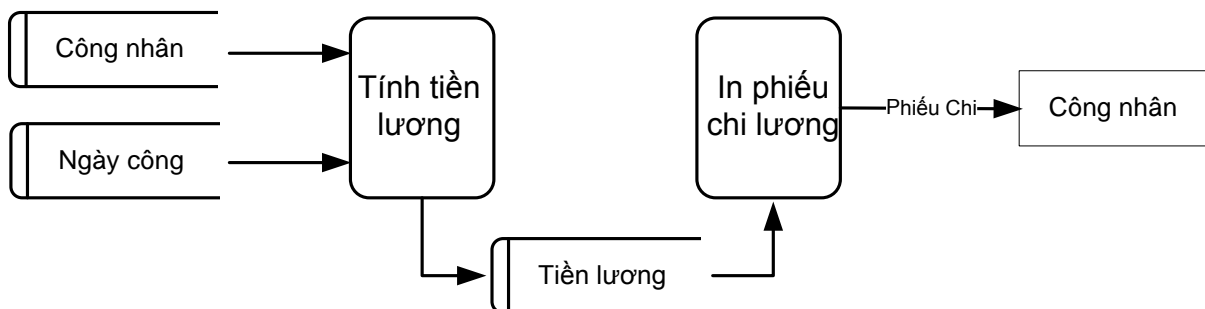
Tiến trình tiếp theo có thể thực hiện ở nơi khác được không?

Tiến trình tiếp theo có thể được thực hiện bởi người khác được không?

Nếu một trong ba câu hỏi trên là có thì ta tách chúng bằng cách đặt một tệp dữ liệu ở giữa.



Hình 4.5. *Quá trình thanh toán lương chưa được tách*



Hình 4.6. *Cách tách quá trình thanh toán lương*

4.6. BIỂU ĐỒ LUỒNG DỮ LIỆU MỨC NGỮ CẢNH

Biểu đồ luồng dữ liệu mức ngữ cảnh (Context data flow diagram) là một mô hình chức năng được dùng để tài liệu hóa phạm vi của một hệ thống. Nó còn được gọi là *mô hình môi trường*. Để xây dựng biểu đồ ngữ cảnh, cần phải:

Xác định biên giới của hệ thống

Xác định các tác nhân ngoài



Không chi tiết về các quá trình và kho dữ liệu của hệ thống

Chiến lược cụ thể xây dựng biểu đồ ngữ cảnh:

- Coi cả hệ thống là một “hộp đen”, xem nó là một chức năng duy nhất và chỉ quan tâm tới phần bên ngoài của nó mà không phải xét tới những hoạt động bên trong.
- Hỏi người sử dụng cuối của hệ thống về những giao dịch nghiệp vụ mà hệ thống phải đáp ứng. Đó chính là các luồng vào của hệ thống. Với mỗi luồng vào, cần xác định nguồn tương ứng của nó. Các nguồn sẽ trở thành các tác nhân ngoài trong sơ đồ ngữ cảnh.
- Hỏi người sử dụng cuối về những đáp ứng phải được sinh ra bởi hệ thống. Đó chính là các luồng ra của hệ thống, Với mỗi luồng ra, xác định đích của nó. Các đích đó cũng sẽ trở thành các tác nhân ngoài.
- Xác định các kho dữ liệu ngoài. Rất nhiều hệ thống đòi hỏi truy nhập vào các tệp hoặc cơ sở dữ liệu của hệ thống khác.

Vẽ một biểu đồ ngữ cảnh dựa trên tất cả các thông tin đã xử lý. Chú ý chỉ minh họa những luồng dữ liệu thể hiện những mục tiêu chính của hệ thống nhằm tránh việc biểu đồ ngữ cảnh có quá nhiều luồng dữ liệu vào/ra.

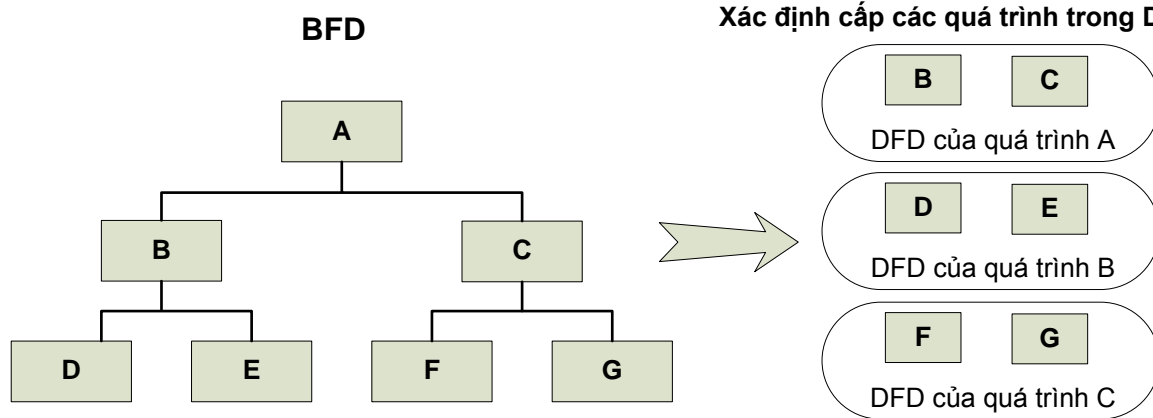
4.7. TRÌNH TỰ VÀ QUY TẮC XÂY DỰNG DFD

4.7.1. Các bước xây dựng DFD

Kỹ thuật phổ biến được dùng để xây dựng DFD là kỹ thuật phân mức. Dựa theo BFD của hệ thống, chúng ta sẽ xây dựng DFD theo nhiều mức, mỗi mức thể hiện trên một hoặc nhiều trang. Nên đặt tên cho mỗi trang bằng tên của chức năng đang được phân tích trên trang đó. Như vậy với trang phân tích tại mức 0 thì tên của nó chính là tên của hệ thống.

Biểu đồ ngữ cảnh

- Xác định hệ thống và giới hạn của nó (ngữ cảnh).
 - Xác định các tác nhân ngoài (người cung cấp, người nhận thông tin hệ thống của).
 - Xác định các luồng dữ liệu ngoài (đầu vào, đầu ra).
 - Chú ý: bản thân toàn bộ hệ thống là một quá trình (nó nhận đầu vào và biến đổi nó thành đầu ra) DFD mức 0.
 - Xác định những gì đang được thực hiện ra giữa từng đầu vào và đầu ra tương ứng
 - Xác định các quá trình.
 - Xác định các luồng dữ liệu ngoài giữa các tác nhân ngoài với các quá trình.
 - Xác định các luồng dữ liệu ngoài giữa các quá trình với các kho dữ liệu
- Các DFD mức 1.
- Là các quá trình con của các quá trình mức 0



Hình 4.7. Kỹ thuật phân mức DFD dựa trên BFD

4.7.2. Các quy tắc xây dựng DFD

- Quy tắc 1: mỗi biểu tượng có riêng một nhãn duy nhất để tránh gây hiểu nhầm
- Quy tắc 2: Sử dụng một ĐỘNG TỪ để gán nhãn cho một quá trình (vì một quá trình là một hành động!!!)
- Quy tắc 3: Mỗi luồng dữ liệu phải được liên kết với ít nhất một quá trình
- Quy tắc 4: Phải có một góc tô đậm trong tất cả các thể hiện của một biểu tượng lặp trong cùng một biểu đồ.
- Quy tắc 5: Một quá trình phải luôn có luồng dữ liệu vào và ra.
- Quy tắc 6: Không cần có một luồng dữ liệu (mà không có sự biến đổi) liên kết với một quá trình (vì hoạt động như vậy là vô giá trị).
- Quy tắc 7: Các quá trình cha và các quá trình con tương ứng của nó phải có các luồng dữ liệu vào ra giống nhau (nhưng các quá trình con có thể có luồng dữ liệu của riêng nó).
- Quy tắc 8: Các luồng dữ liệu không thể tự phân tách được.
- Quy tắc 9: Một gói dữ liệu có thể gồm nhiều phần tử dữ liệu được truyền đi đồng thời tới cũng một đích.
- Quy tắc 10: Không được sử dụng mũi tên hai chiều vì luồng vào (cập nhật) và luồng ra (trích thông tin) của một kho dữ liệu mang nội dung thông tin khác nhau.

4.7.3. Phân tích hướng cấu trúc cổ điển (top-down)

1. Vẽ DFD logic dạng top-down của hệ thống hiện có: Vẽ các biểu đồ DFD vật lý dạng top-down để biểu diễn sự cài đặt vật lý hiện tại của hệ thống bao gồm các giới hạn của nó.
2. Chuyển đổi các DFD vật lý thành các DFD logic tương ứng của nó.
3. Vẽ các DFD logic dạng top-down biểu diễn một hệ thống được cải thiện.
4. Mô tả tất cả các luồng dữ liệu, kho dữ liệu, quy tắc và thủ tục trong một từ điển dữ liệu.



5. Một cách tùy chọn, đánh dấu các bản sao của DFD logic để biểu diễn các giải pháp vật lý khác nhau.
6. Vẽ các DFD vật lý dạng top-down biểu diễn giải pháp được lựa chọn.

4.7.4. Phân tích hướng cấu trúc hiện đại (hướng sự kiện)

1. Vẽ một DFD ngữ cảnh để xác lập phạm vi ban đầu của dự án.
2. Vẽ một biểu đồ phân rã chức năng để phân chia hệ thống thành các hệ thống con.
3. Tạo một danh sách các đáp ứng sự kiện hay use-case cho hệ thống để xác định các sự kiện mà hệ thống phải có đáp ứng.
4. Vẽ một DFD sự kiện (hay bộ xử lý sự kiện) cho từng sự kiện.
5. Kết hợp các DFD sự kiện thành một biểu đồ hệ thống (hay đối với các hệ thống lớn thì là các biểu đồ hệ thống con).
6. Vẽ các DFD chi tiết cho các bộ xử lý sự kiện phức tạp hơn.
7. Tài liệu hoá các luồng dữ liệu và quá trình trong từ điển dữ liệu.

4.7.5. So sánh DFD với biểu đồ luồng tiến trình (flowchart)

Các quá trình trong DFD có thể thực hiện song song (vào cùng một thời gian)

- Các quá trình trong flowchart thực hiện mỗi cái vào một thời điểm DFD thể hiện luồng dữ liệu xuyên suốt hệ thống
- Flowchart thể hiện luồng điều khiển (trình tự và sự chuyển giao điều khiển) Các quá trình trong một DFD có thể có thời gian khác nhau đáng kể (hàng ngày, hàng tuần, cấp bách)
- Các quá trình trong flowchart là một phần của một chương trình đơn nhất với thời gian nhất quán.

4.7.6. Sự bảo toàn dữ liệu

Là việc bảo đảm rằng một luồng dữ liệu chỉ chứa dữ liệu cần thiết cho quá trình nhận nó.

- Xác định và loại bỏ những phần không hiệu quả
- Đơn giản hoá việc giao tiếp giữa các quá trình
- Phải xác định chính xác kết cấu dữ liệu của từng luồng dữ liệu, được biểu diễn dưới dạng các cấu trúc dữ liệu (trong phần Mô hình hoá dữ liệu).

	Chức năng	Luồng dữ liệu	Kho dữ liệu	Tác nhân ngoài	Tác nhân trong
Định nghĩa	Nhiệm vụ xử lý thông tin	Thông tin vào / ra một chức năng xử lý	Nơi lưu trữ thông tin trong một thời gian	Người hay tổ chức ngoài hệ thống có giao tiếp với hệ thống	Một chức năng hay một hệ con của hệ thống nhưng được mô tả ở trang khác
Tên đi	Động từ	Danh từ	Danh từ	Danh từ	Động từ



kèm	(+ <i>bổ ngữ</i>)	(+ <i>tính từ</i>)	(+ <i>tính từ</i>)		
Biểu đồ		Tên →	_____ Tên _____		
Ví dụ		Hoá đơn đã → Xác nhận chi	_____ Đơn hàng _____		

Bảng 4. 1. Bảng tóm tắt các thành phần và ý nghĩa của DFD

CÂU HỎI THẢO LUẬN

- 4.1. Phân biệt mô hình logic và mô hình vật lý.
- 4.2. Ý nghĩa của biểu đồ phân cấp chức năng là gì?
- 4.3. Nêu vai trò của DFD.
- 4.4. Khi xây dựng DFD, có thể gặp những sai sót nào?





BÀI 5. MÔ HÌNH HOÁ DỮ LIỆU

Tóm tắt: Tập trung vào việc mô hình hóa dữ liệu bao gồm khái niệm và phương pháp xây dựng biểu đồ thực thể quan hệ và biểu đồ dữ liệu quan hệ.

5.1. MÔ HÌNH HÓA DỮ LIỆU

5.1.1. Khái niệm

Mô hình hoá dữ liệu (mô hình hoá cơ sở dữ liệu, mô hình hoá thông tin) là một kỹ thuật để tổ chức và tài liệu hoá dữ liệu của hệ thống trong một mô hình. Kỹ thuật này xác định các yêu cầu nghiệp vụ đối với một cơ sở dữ liệu. Mô hình hoá dữ liệu thường được gọi là mô hình hoá cơ sở dữ liệu vì cuối cùng một mô hình dữ liệu luôn được cài đặt thành cơ sở dữ liệu.

Biểu đồ quan hệ thực thể (Entity Relationship Diagram - ERD) mô tả dữ liệu dưới dạng các thực thể và các quan hệ được mô tả bởi dữ liệu. ERD xác định các đơn vị thông tin cơ sở cần thiết cho hệ thống (các thực thể) và các mối quan hệ giữa chúng. Nghĩa là tất cả các dữ liệu chỉ được lưu giữ một lần trong toàn bộ hệ thống.

5.1.2. Từ mô hình dữ liệu tới cài đặt cơ sở dữ liệu

ERD: là một mô hình khái niệm của các thực thể dữ liệu, các thuộc tính (đặc điểm) và các quan hệ (với các thực thể khác) của chúng trong một hệ thống thông tin (độc lập kỹ thuật). **(Phân tích)**

Mô hình dữ liệu quan hệ (Relational Data Model - RDM): một bản thiết kế cho việc cài đặt của một mô hình dữ liệu khái niệm (ERD) trong môi trường cơ sở dữ liệu quan hệ (độc lập phần mềm). **(Thiết kế)**

Sơ đồ quan hệ : một sơ đồ thể hiện cách thức một mô hình dữ liệu được cài đặt với hệ quản trị cơ sở dữ liệu (như Microsoft Access hay MS SQL Server...). **(Cài đặt)**

5.1.3. Vai trò của biểu đồ quan hệ thực thể

Cơ sở dữ liệu = dữ liệu + quan hệ

ERD được dùng để mô hình hoá dữ liệu và quan hệ của chúng

ERD là một biểu diễn đồ hoạ của mô hình dữ liệu khái niệm

ERD là độc lập tài nguyên: nó không gắn với bất cứ môi trường cơ sở dữ liệu cụ thể nào.

5.2. CÁC PHẦN TỬ CỦA BIỂU ĐỒ QUAN HỆ THỰC THỂ (ERD)

5.2.1. Thực thể (Entity)

Thực thể là một nhóm các thuộc tính tương ứng với một đối tượng khái niệm mà chúng ta cần thu thập và lưu trữ dữ liệu; là các vật thể, con người, địa điểm, sự kiện, khái niệm mà sự tồn tại của nó không phụ thuộc vào các thực thể khác.

Thực thể là một tập các thể hiện của đối tượng mà nó biểu diễn.

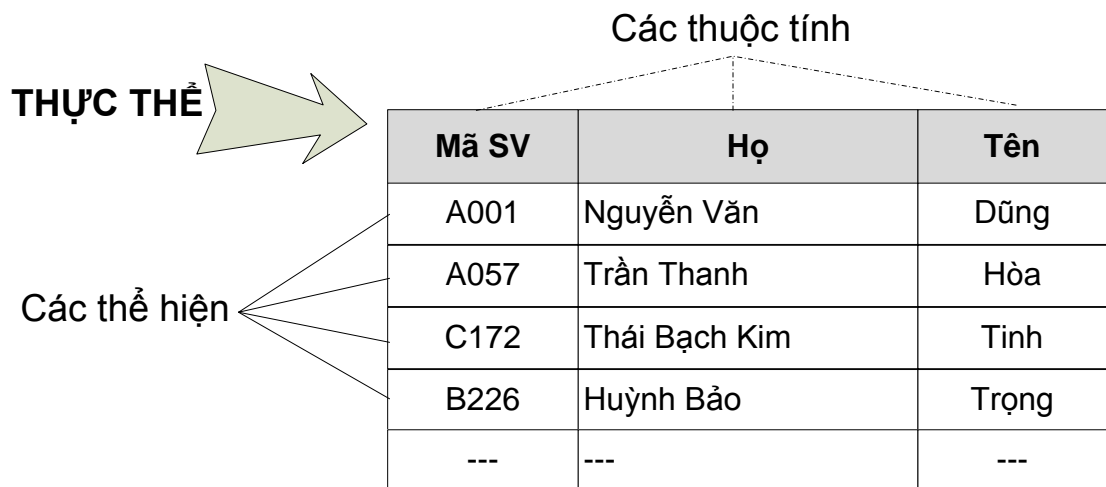


Thực thể phải có một tên duy nhất (một danh từ số ít), từ định danh duy nhất và ít nhất một thuộc tính (chính là từ định danh).

Các loại thực thể có thể có:

- **Con người:** là những người thực hiện chức năng nào đó trong hoặc ngoài hệ thống. Ví dụ: công ty, khách hàng, phòng ban, bộ phận, nhân viên, giáo viên, sinh viên, nhà cung cấp...
- **Địa điểm:** là nơi được sử dụng bởi con người. Ví dụ: nơi bán hàng, toà nhà, phòng, chi nhánh...
- **Vật thể:** là những đối tượng vật lý thấy rõ. Ví dụ: sách, tạp chí, sản phẩm, nguyên liệu thô, công cụ...
- **Sự kiện:** là những gì xảy ra theo thời gian hoặc theo một quy trình nhất định. Ví dụ: giải thưởng, sự huỷ bỏ, chuyến bay, giờ học, việc lập hoá đơn, việc đặt hàng, việc đăng ký, sự gia hạn, sự đặt chỗ, việc bán hàng...
- **Khái niệm:** là những gì không thể nhìn thấy được. Ví dụ: tài khoản, khoảng thời gian, khoá học, nguồn tài chính, quy tắc, luật lệ...

Trong ERD, thực thể được ký hiệu là một hình chữ nhật, mỗi thực thể tương đương với một bảng dữ liệu trong cơ sở dữ liệu của hệ thống. Thẻ hiện của thực thể: là một thực thể cụ thể. Ví dụ thực thể SinhVien có thể có nhiều thẻ hiện như Dũng, Trọng, Hòa...



Hình 5.1. Ví dụ về thực thể SinhVien

5.2.2. Thuộc tính (Properties)

Mỗi thực thể bao gồm nhiều thông tin, mỗi thông tin là một thuộc tính của tập thực thể, ứng với một trường trong bảng dữ liệu tương ứng. Ví dụ: khách hàng Nguyễn Văn A có năm sinh là 1981, có số điện thoại là 8534... . Tập thực thể khách hàng sẽ có các thuộc tính “năm sinh”, “số điện thoại”. Một thuộc tính là một đặc tính mô tả hoặc đặc điểm quan tâm của một thực thể.

Kiểu dữ liệu (Data type) của một thuộc tính xác định kiểu dữ liệu có thể lưu trữ được trong thuộc tính đó.



Phạm vi (Domain) của một thuộc tính xác định các giá trị mà thuộc tính đó có thể chứa một cách hợp lệ.

Giá trị mặc định (default value) của một thuộc tính là giá trị sẽ được ghi vào nếu không được xác định bởi người dùng.

Một số kiểu dữ liệu điển hình:

Kiểu dữ liệu	Ý nghĩa
NUMBER	Có thể là số thực hoặc số nguyên
TEXT	Một chuỗi ký tự, bao gồm cả các con số
MEMO	Tương tự như TEXT nhưng có kích thước không xác định. Một số hệ thống đòi hỏi khả năng này để lưu các văn bản dài trong bản ghi cơ sở dữ liệu
DATE	Ngày dưới bất kỳ định dạng nào
TIME	Giờ dưới bất kỳ định dạng nào
YES/NO	Một thuộc tính chỉ có thể nhận một trong hai giá trị bên
VALUE SET	Một tập hữu hạn các giá trị. Trong hầu hết các trường hợp, một lược đồ mã sẽ được xây dựng (ví dụ M=Nam giới, F=Nữ giới)
IMAGE	Các loại hình ảnh

Bảng 5. 1. Một số kiểu dữ liệu điển hình cho thuộc tính của thực thể

Kiểu dữ liệu	Phạm vi giá trị	Ví dụ
NUMBER	Đối với số nguyên, xác định phạm vi Đối với số thực, xác định phạm vi và độ chính xác	{ 10-99 } { 1.000-799.999 }
TEXT	Kích thước lớn nhất của thuộc tính Các giá trị thực tế thường là vô hạn; tuy nhiên, người dùng có thể xác định các hạn chế nào đó	Text(30)
DATE	Sự biến đổi trên các định dạng MMDDYYYY MMYYYY	MMDDYYYY MMYYYY
TIME	Đối với thời gian AM/PM: HHMMT Đối với thời gian 24 giờ: HHMM	HHMMT HHMM
YES/NO	{ YES, NO }	{ YES, NO } { ON, OFF }
VALUE SET	{ giá trị#1, giá trị#2,... giá trị#n } { bảng các mã và ý nghĩa }	{ M=Nam giới F=Nữ giới }

Bảng 5. 2. Phạm vi giá trị hợp lệ của các kiểu dữ liệu

Giá trị mặc định	Diễn giải	Ví dụ
Một giá trị hợp lệ từ phạm vi	Đối với một thể hiện của thuộc tính, nếu người dùng không xác định giá trị thì sử dụng giá trị này	0 1.00
NONE hoặc	Đối với một thể hiện của thuộc tính,	NONE NULL



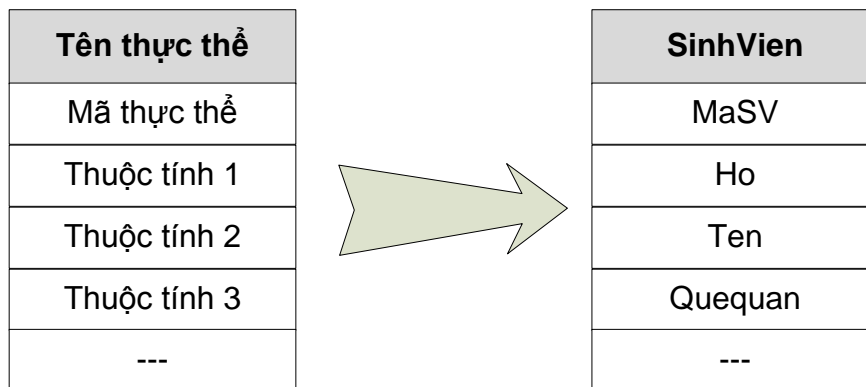
NULL	nếu người dùng không xác định giá trị thì để trống	
Required hay NOT NULL	Đối với một thể hiện của thuộc tính, đòi hỏi người dùng phải nhập vào một giá trị hợp lệ từ phạm vi. (Điều này được dùng khi không có giá trị nào trong phạm vi có thể làm giá trị mặc định nhưng lại nhất thiết phải có giá trị được nhập)	REQUIRED NOT NULL

Bảng 5.3. Các giá trị mặc định chấp nhận được cho các thuộc tính

Có 3 loại thuộc tính sau:

- **Thuộc tính khóa:** gồm một hoặc nhiều thuộc tính trong thực thể được dùng để gán cho mỗi thể hiện thực thể một cách tham khảo duy nhất. Ví dụ thuộc tính Mã sinh viên trong thực thể Sinh viên
- **Thuộc tính mô tả:** là các thuộc tính dữ liệu mô tả về một đối tượng và không được chọn làm thuộc tính khóa. Ví dụ các thuộc tính Tên sinh viên, Địa chỉ...
- **Thuộc tính kết nối:** là thuộc tính mà với thực thể này thì là thuộc tính mô tả nhưng với thực thể khác thì là thuộc tính khóa, nó đóng vai trò kết nối các thực thể có quan hệ với nhau

Ví dụ về các loại thuộc tính:



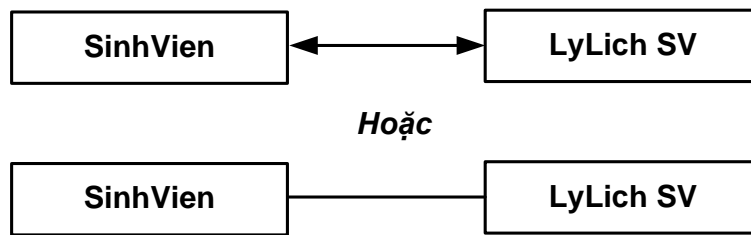
Hình 5.2. Ví dụ các thuộc tính của thực thể

5.2.3. Mỗi quan hệ

Một quan hệ tài liệu hoá một liên kết giữa một, hai hoặc nhiều thực thể. Nó phải có một cái tên (và có thể mang dữ liệu).

5.2.3.1. Quan hệ 1-1:

Là mối quan hệ trong đó một thực thể của tập thực thể này tương ứng với duy nhất một thực thể của tập thực thể kia và ngược lại. Ví dụ, một thực thể **Sinh viên** chỉ ứng với duy nhất một thực thể **Lý lịch sinh viên** mô tả nó. Quan hệ 1-1 được biểu diễn bằng một mũi tên hai đầu hoặc là một đoạn thẳng...

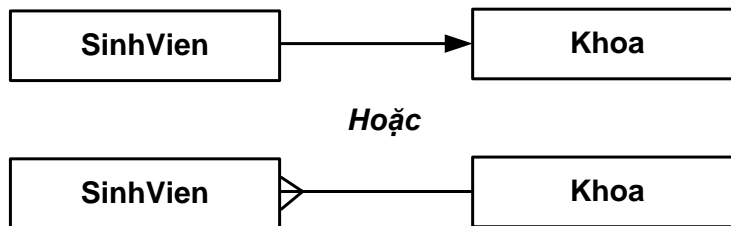


Hình 5.3. Quan hệ 1-1

Quan hệ này sẽ dẫn tới việc nhập chung hai tập thực thể thành một tập thực thể, tập thực thể mới phải bao gồm các thuộc tính của hai tập thực thể cũ.

5.2.3.2. Quan hệ 1-n:

Là mối quan hệ mà trong đó một thực thể của tập thực thể này có quan hệ với nhiều thực thể của tập thực thể kia. Ví dụ, một **Khoa** có thể có nhiều **Sinh viên** nên *một thực thể Khoa trong tập thực thể Khoa có quan hệ với nhiều thực thể Sinh viên trong tập thực thể Sinh viên*. Quan hệ 1- nhiều được biểu diễn bằng một mũi tên 1 đầu hướng từ bên nhiều tới bên 1 hoặc là một đoạn thẳng với một đầu là trục ba hướng về bên nhiều...



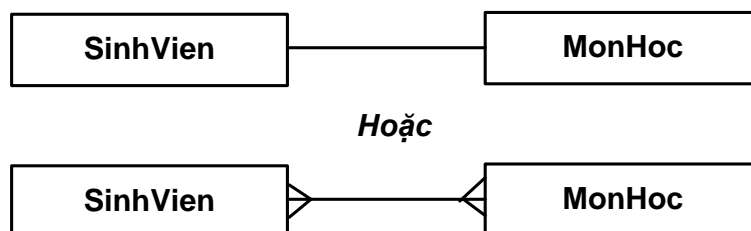
Hình 5.4. Quan hệ 1-n

Quan hệ này đóng vai trò rất quan trọng thể hiện mối liên hệ giữa các thực thể trong mô hình. Ở đây, thuộc tính khóa của bên một sẽ là thuộc tính kết nối của bên nhiều.

5.2.3.3. Quan hệ n-n:

Là mối quan hệ mà trong đó một thực thể của tập thực thể này có quan hệ với nhiều thực thể của tập thực thể kia và ngược lại. Ví dụ, **một Sinh viên** trong *tập thực thể Sinh viên* có thể có **nhiều Môn học** trong *tập thực thể Môn học* và ngược lại **một Môn học** có thể được học bởi **nhiều Sinh viên**.

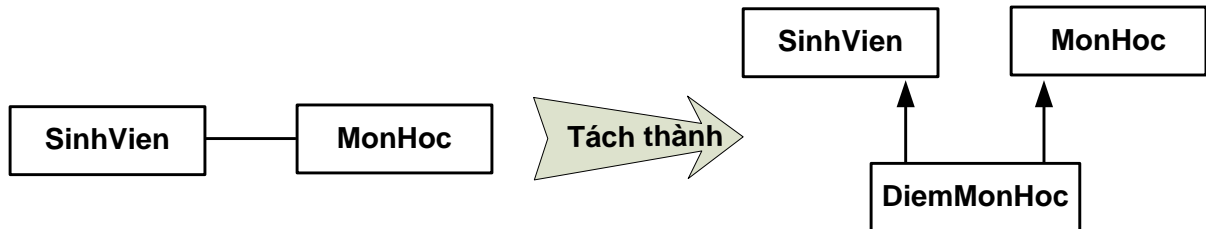
Quan hệ nhiều - nhiều được biểu diễn bằng một đoạn thẳng hoặc là một đoạn thẳng có trục ba ở cả hai đầu...



Hình 5.5. Hình quan hệ n-n



Quan hệ này không thể hiện được mối quan hệ giữa hai thực thể cũng như không cho thấy điều gì về mặt nghiệp vụ, nên thường tách thành hai quan hệ 1- n bằng cách tạo một tập thực thể trung gian có quan hệ 1- n với cả hai tập thực thể đã có. Ví dụ với quan hệ n-n giữa **Sinh viên** và **Môn học**, ta sẽ tạo tập thực thể **Điểm môn học** có quan hệ là một Sinh viên gồm nhiều Điểm môn học và một Môn học lại ứng với nhiều Điểm môn học.



Hình 5.6. Tách quan hệ n-n thành các quan hệ 1-n

Số yếu tố tài liệu hoá số lượng các thể hiện của một thực thể có thể có quan hệ với một thể hiện của thực thể khác trong một quan hệ.

- Bao gồm số lớn nhất và nhỏ nhất các thể hiện.
- Phản ánh quy tắc nghiệp vụ hoặc thực tế nghiệp vụ nó chung (ví dụ có bao nhiêu lớp học mà một sinh viên có thể tham gia, có bao nhiêu sinh viên có thể có trong một lớp học).

Bậc của quan hệ xác định số lượng thực thể tham gia vào một quan hệ:

- Một ngôi
- Hai ngôi
- Ba ngôi
-

Quan hệ một ngôi (đệ qui) chỉ gồm một thực thể (các thể hiện trong cùng một thực thể)

Quan hệ hai ngôi.

Quan hệ ba ngôi.

5.3. CÁC BƯỚC XÂY DỰNG BIỂU ĐỒ QUAN HỆ THỰC THỂ

5.3.1. Các bước mô hình hóa dữ liệu logic

1. Mô hình dữ liệu ngữ cảnh

Đề thiết lập phạm vi dự án

2. Mô hình dữ liệu dựa trên khoá

Loại bỏ các quan hệ không cụ thể

Thêm các thực thể có liên quan

Bao gồm các khoá chính

Xác định chính xác số yếu tố

3. Mô hình dữ liệu với thuộc tính đầy đủ

Tất cả các thuộc tính còn lại.

Các tiêu chuẩn nhóm con



4. Mô hình dữ liệu được chuẩn hoá

Thế nào là một mô hình dữ liệu tốt?

- Đơn giản

Các thuộc tính dữ liệu mô tả bất cứ thực thể đã cho nào thì chỉ nên mô tả thực thể đó thôi.

Mỗi thuộc tính của một thể hiện của thực thể chỉ có thể có một giá trị.

- Không dư thừa

Mỗi thuộc tính dữ liệu, không phải là khoá ngoại, mô tả tối đa một thực thể.

Tìm cùng một thuộc tính được ghi lại nhiều lần dưới các tên khác nhau.

- Linh động và dễ điều chỉnh cho những nhu cầu phát sinh trong tương lai.

5.3.2. Trình tự xây dựng ERD

Xác định các thực thể (Top-down).

Xác định bậc của các quan hệ giữa các thực thể (rõ ràng ngữ cảnh).

Hoàn thiện các quan hệ với các số yếu tố (rõ ràng ngữ cảnh).

Xây dựng mô hình.

5.3.2.1. Xác định các thực thể

Cách 1:

Một tập thực thể có thể thuộc một trong 3 loại sau đây.

- Thông tin liên quan tới một giao dịch chủ yếu của hệ thống, ví dụ như hóa đơn bán hàng thuộc về quá trình bán hàng, đơn đặt hàng thuộc về quá trình mua hàng.

- Thông tin liên quan tới thuộc tính hoặc tài nguyên của hệ thống, ví dụ khách hàng, nhà cung cấp, vị trí kho hàng...

- Thông tin đã được khái quát dưới dạng thống kê liên quan tới lập kế hoạch hoặc quản lý như bảng chấm công, lịch trực...

Để nhận ra tập thực thể, phải đặt câu hỏi để ghi nhận thông tin về thực thể:

- Cái gì mà ta cần lưu thông tin về nó?

- Cái gì là cốt yếu trong hệ thống?

- Cái gì mà ta nói về nó trong hệ thống?

- Cái gì có thể dùng để phân biệt sự kiện của tập thực thể này với sự kiện của một tập thực thể khác?

Cách 2:

Lấy một bản mô tả về hệ thống hiện tại hoặc cần có trong tương lai, xem xét các danh từ có trong đó xem có phải là thông tin cần lưu giữ không. Chú ý loại bỏ các từ đồng nghĩa. Lưu ý là có những danh từ mang tính mô tả nhưng lại không trở thành một tập thực thể, một số khác lại có thể là tập thực thể tiềm năng.

Ví dụ: chương trình quản lý kho hàng



- Theo dõi **hàng** tồn trong một **kho** nào đó tại một thời điểm nào đó.
- Theo dõi chi tiết xuất nhập tồn của mỗi loại **hàng hóa**.
- In chi tiết xuất nhập **vật tư** cho mỗi **khách hàng**.

Vậy các tập thực thể xác định được từ mô tả này chính là: vật tư, kho hàng, khách hàng.

5.3.2.2. Xác định mối quan hệ

Quan hệ giữa các tập thực thể thường được diễn tả bởi các động từ, nó xác định sự tác động của các thực thể với nhau. Để xác định được các mối quan hệ giữa các tập thực thể, cần chú ý:

Nếu cần phải lưu giữ thông tin về tập thực thể này trong tập thực thể kia thì sẽ có một quan hệ xuất hiện để tạo mối liên kết.

Khi quan hệ giữa hai thực thể là gián tiếp thì ta không cần phải xây dựng mối quan hệ giữa chúng.

Ví dụ: Hệ thống thông tin quản lý tồn kho gồm 4 thực thể cơ bản sau:

Tên dữ liệu	Bao gồm
Danh mục tồn kho	+ Thẻ giá + Hồ sơ tồn kho + Báo cáo hàng tồn kho cũ + Danh sách đặt hàng bổ sung + Báo cáo mức tồn kho + Bản kiểm kê tồn kho
Danh mục hàng bán	+ Hoá đơn bán hàng + Báo cáo bán hàng
Danh mục nhà cung cấp	+ Hoá đơn nhà cung cấp + Đơn đặt hàng bổ sung
Danh mục bảo hành	+ Đơn bảo hành

Quan hệ giữa Danh mục tồn kho và Danh mục nhà cung cấp là quan hệ n-n vì: Mỗi loại sản phẩm trong danh mục tồn kho có thể có hơn một nhà cung cấp trong danh mục nhà cung cấp.

Mỗi nhà cung cấp trong danh mục nhà cung cấp có thể cung cấp nhiều hơn một loại sản phẩm.

Chúng ta đưa ra tập thực thể mới là **Danh mục đặt hàng** có quan hệ 1 - n với cả hai tập thực thể ban đầu. Việc này nhằm tách 1 quan hệ n-n thành 2 quan hệ 1-n. Tương tự, có quan hệ n - n giữa danh mục đặt hàng và danh mục tồn kho. Chúng ta đưa ra tập thực thể **Chi tiết đặt hàng**.

Chúng ta đã thu thập được danh sách một số phần tử dữ liệu, cần xác định những thực thể còn thiếu. Bắt đầu với dữ liệu bán hàng, có một vài trường của hoá đơn bán mô tả một khách hàng. Vậy khách hàng là một đối tượng mà dữ liệu tập hợp vào đó. Do vậy **Danh mục khách hàng** là một tập thực thể.

Nhận xét rằng quan hệ giữa danh mục khách hàng và danh mục tồn kho là quan hệ n-n vì:



Một khách hàng có thể mua nhiều loại sản phẩm.

Một loại sản phẩm có thể được bán cho nhiều khách hàng.

Với các thuộc tính còn lại trong dữ liệu bán hàng ta có **Danh mục bán hàng**, từ đó chúng ta có quan hệ n-n giữa Danh mục bán hàng với Danh mục tồn kho vì:

Một giao dịch bán có thể có nhiều hơn một loại hàng.

Một loại hàng có thể có trong nhiều giao dịch bán.

Do vậy, ta đưa ra tập thực thể mới là **Chi tiết bán hàng**

5.3.3. Quy tắc xây dựng ERD

Mỗi thực thể phải có tên.

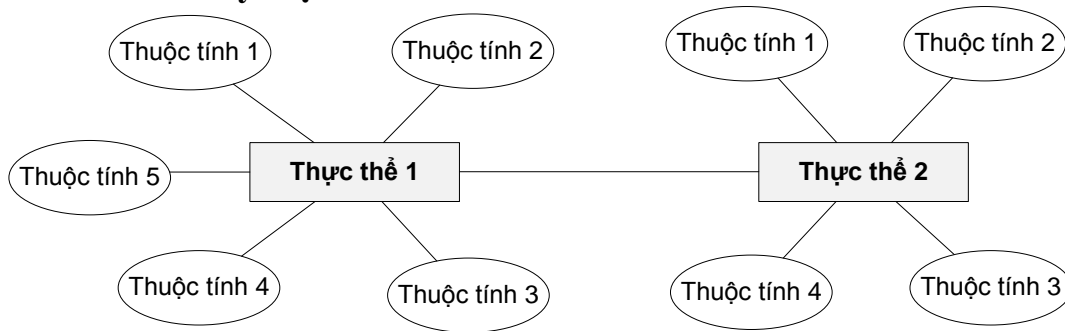
Mỗi thực thể phải có định danh.

Mỗi thể hiện không thể là một thực thể.

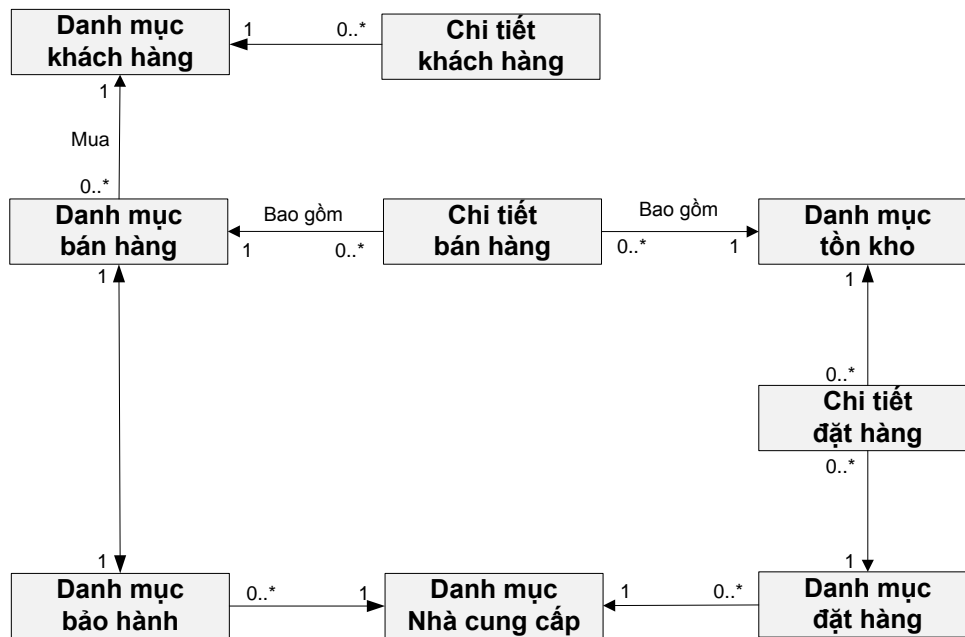
Mỗi quan hệ phải có tên (có thể mang hoặc không mang dữ liệu).

Số yếu tố phải hợp lý (rõ ràng ngữ cảnh).

5.3.4. Các kiểu ký hiệu ERD khác



Hình 5.7. Ví dụ các ký hiệu trong biểu đồ ERD



Hình 5.8. Biểu đồ quan hệ thực thể hệ thống quản lý kho



5.4. XÂY DỰNG BIỂU ĐỒ DỮ LIỆU QUAN HỆ (RDM)

Mô hình dữ liệu quan hệ (RDM – Relational Data Model) là công cụ tiếp theo sau ERD được dùng trong việc mô hình hóa dữ liệu nhằm mục đích xác định danh sách các thuộc tính của các thực thể. Quá trình xây dựng RDM bao gồm các bước:

- Xác định các thuộc tính cần thiết.
- Chuẩn hóa các thực thể.
- Xác định các mối quan hệ giữa các thuộc tính của các thực thể.

5.4.1. Xác định thuộc tính

Để xác định các thuộc tính của các thực thể, cần dựa vào những yếu tố sau:

- Sự hiểu biết về hệ thống đang phân tích.
- Quá trình phỏng vấn, trao đổi với người sử dụng.
- Các báo cáo, biểu mẫu được dùng trong hệ thống hiện tại.

Từ những thông tin thu thập được, ta sẽ lập danh sách các thuộc tính cho các thực thể đã được xác định trong ERD. Phân biệt các thuộc tính khóa bằng cách gạch dưới.

5.4.2. Phân tích và chuẩn hóa dữ liệu

Phân tích dữ liệu là một quá trình chuẩn bị một mô hình dữ liệu cho việc cài đặt thành một cơ sở dữ liệu đơn giản, không dư thừa, mềm dẻo và dễ thích ứng. Kỹ thuật cụ thể được gọi là sự chuẩn hóa.

Chuẩn hóa là một kỹ thuật tổ chức các thuộc tính dữ liệu sao cho chúng được nhóm thành các thực thể không dư thừa, ổn định, mềm dẻo và dễ thích ứng:

- Không có sự lặp lại các thuộc tính ở các bảng khác nhau, trừ thuộc tính khóa và thuộc tính kết nối
- Không chứa các thuộc tính có giá trị là kết quả tính được của các thuộc tính khác. Ví dụ, thuộc tính giá thành là kết quả của thuộc tính số lượng nhân với thuộc tính đơn giá nên cần phải loại bỏ.
- Không có vai trò giống nhau giữa các thực thể.

Khái niệm phụ thuộc hàm:

- **Phụ thuộc hàm đơn trị:** từ 1 giá trị của khóa trong bảng, ta chỉ xác định được 1 giá trị cho các thuộc tính khác. VD: với mỗi mã khách hàng, chỉ có duy nhất một giá trị Họ tên, số điện thoại, địa chỉ,...
- **Phụ thuộc hàm đa trị:** 1 giá trị của khóa trong bảng lại ứng với nhiều giá trị của các thuộc tính khác. Ví dụ: ứng với một mã số học sinh lại có nhiều môn học khác nhau vì một học sinh có thể học nhiều môn học.
- Như vậy, nếu có thuộc tính không phụ thuộc hàm vào khóa thì nó phải nằm trong một thực thể khác. Quá trình chuẩn hóa được thực hiện dựa trên khái niệm phụ thuộc hàm nêu trên.



5.4.2.1. Chuẩn hóa dạng 1:

- Yêu cầu: Các thuộc tính nào có thể xuất hiện nhiều lần với cùng một thực thể thì loại bỏ ra.
- Các thuộc tính bị loại ra sẽ cùng với thuộc tính khóa của thực thể ban đầu tạo thành một tập thực thể mới.

5.4.2.2. Chuẩn hóa dạng 2:

- Yêu cầu: Tất cả các thuộc tính trong thực thể phải phụ thuộc hàm vào toàn bộ khóa.
- Đối với các thực thể chỉ có một trường là khóa thì đương nhiên thỏa mãn dạng chuẩn 2.
- Đối với các thực thể có khóa bao gồm 2 thuộc tính trở lên, nếu trong đó có những thuộc tính phụ thuộc hàm đơn trị vào một bộ phận của khóa thì tách các thuộc tính đó ra thành 1 thực thể mới với khóa là bộ phận khóa của thực thể ban đầu mà nó phụ thuộc hàm.

5.4.2.3. Chuẩn hóa dạng 3:

- Yêu cầu: Tất cả các thuộc tính phải phụ thuộc đơn trị vào khóa và không phụ thuộc hàm đơn trị vào bất kỳ thuộc tính nào không phải là khóa trong thực thể.
- Tách những thuộc tính phụ thuộc hàm đơn trị vào thuộc tính không phải là khóa, đưa chúng vào thực thể mới có khóa chính là thuộc tính mà nó phụ thuộc hàm.

Ví dụ: Xét quá trình xây dựng các thuộc tính cho các tập thực thể dựa trên mẫu hóa đơn bán hàng của một công ty

Số HD:
HÓA ĐƠN BÁN
 Ngày.....
 Họ tên khách hàng:.....Mã số khách hàng:.....
 Địa chỉ:.....

Mã số mặt hàng	Tên hàng	Số lượng	Đơn giá	Thành tiền

Quá trình chuẩn hóa diễn ra như sau:

Thuộc tính nguyên bản chưa chuẩn hóa	Chuẩn hóa dạng 1 1NF	Chuẩn hóa dạng 2 2NF	Chuẩn hóa dạng 3 3NF
--------------------------------------	--------------------------------	--------------------------------	--------------------------------



<u>Số hiệu đơn</u> Mã số khách hàng Ngày đặt hàng Tên khách hàng Địa chỉ Mã số mặt hàng Tên mặt hàng Số lượng Đơn giá	<u>Số hiệu đơn</u> Mã số khách hàng Ngày đặt hàng Tên khách hàng Địa chỉ <u>Số hiệu đơn</u> <u>Mã số mặt hàng</u> Tên mặt hàng Số lượng Đơn giá	<u>Số hiệu đơn</u> Mã số khách hàng Ngày đặt hàng Tên khách hàng Địa chỉ <u>Số hiệu đơn</u> <u>Mã số mặt hàng</u> Số lượng Đơn giá <u>Mã số mặt hàng</u> Tên mặt hàng	<u>Số hiệu đơn</u> Mã số khách hàng Ngày đặt hàng <u>Mã số khách hàng</u> Tên khách hàng Địa chỉ <u>Số hiệu đơn</u> Mã số mặt hàng Số lượng Đơn giá <u>Mã số mặt hàng</u> Tên mặt hàng
--	---	---	---

Sau khi chuẩn hóa, thu được các thực thể sau:

Đơn hàng bán (Số hiệu đơn, Mã số khách hàng, Ngày đặt hàng)

Khách hàng (Mã số khách hàng, Tên khách hàng, Địa chỉ khách hàng)

Dòng đơn hàng (Số hiệu đơn, Mã số mặt hàng, Số lượng, Đơn giá)

Mặt hàng (Mã số mặt hàng, Tên mặt hàng)

Kết hợp các tập thực thể chung.

Do việc chuẩn hóa xuất phát từ nhiều tài liệu khác nhau nên có thể sau khi chuẩn hóa sẽ xuất hiện các thực thể giống nhau. Cần phải hợp nhất chúng thành một thực thể mà chứa đủ các thuộc tính.

Rất có thể sau giai đoạn này thì thực thể thu được sẽ không còn ở dạng chuẩn 3 nên cần phải thực hiện chuẩn hóa lại các thực thể mới.

Ví dụ: Có 2 tập thực thể đơn đặt hàng được chuẩn hóa từ 2 tài liệu là đơn đặt hàng và tài liệu giao hàng như sau:

Đơn hàng (Số hiệu đơn, Mã số khách hàng, Ngày đặt hàng)

Đơn đặt hàng (Số hiệu đơn, Tình trạng đơn hàng, Địa chỉ giao hàng)

Sau khi kết hợp có:

Đơn hàng (Số hiệu đơn, Mã số khách hàng, Ngày đặt hàng, Tình trạng đơn hàng, Địa chỉ giao hàng)

5.4.3. Xác định các mối quan hệ

Theo 2 ví dụ trên, sau khi chuẩn hóa, thu được các thực thể sau:

Đơn hàng bán (Số hiệu đơn hàng, Mã số khách hàng, Ngày đặt hàng)

Khách hàng (Mã số khách hàng, Tên khách hàng, Địa chỉ khách hàng)

Dòng đơn hàng (Số hiệu đơn hàng, Mã số mặt hàng, Số lượng, Đơn giá)

Mặt hàng (Mã số mặt hàng, Tên mặt hàng)

Giao nhận (Số hiệu giao nhận, Mã số khách hàng, Ngày giao)

Dòng giao hàng (Số hiệu giao nhận, Số hiệu đơn hàng, Mã số mặt hàng, Số lượng giao)

**5.4.3.1. Ma trận thực thể/khóa.**

Để xác định các mối quan hệ giữa các thực thể, ta cần lập bảng ma trận thực thể/khóa. Trong đó, các cột liệt kê các tập thực thể, các hàng liệt kê các thuộc tính khóa của các thực thể.

Ứng với mỗi ô giao giữa hàng và cột, nếu thuộc tính khóa có trong thực thể, ta đánh dấu X, nếu không là khóa của thực thể nhưng có xuất hiện trong đó thì đánh dấu O.

Ví dụ:

Thực thể \ Thuộc tính khóa	Đơn hàng	Khách hàng	Dòng đơn hàng	Mặt hàng	Giao nhận	Dòng giao hàng
Số hiệu đơn hàng	X		X			X
Mã số khách hàng	O	X			X	
Mã số mặt hàng			X	X		X
Số hiệu giao nhận					X	X

5.4.3.2. Thiết lập các mối quan hệ:

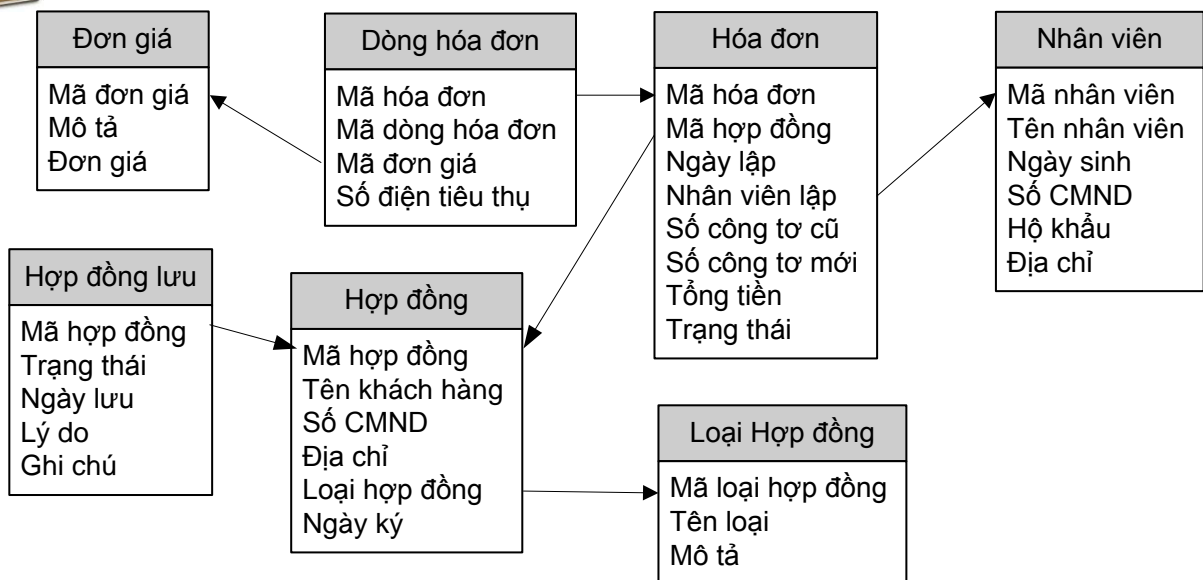
Dựa theo bảng ma trận thực thể/khóa, ta xác định các mối quan hệ bằng cách bắt đầu từ cột thứ nhất, từ ô chứa khóa của nó, ta chiếu qua các ô kế tiếp trên cùng một hàng để xem nếu ô nào có chứa dấu X hoặc O thì ta có một liên kết giữa thực thể đang xét với thực thể có ô chứa dấu trên hàng đó.

5.4.4. Xây dựng mô hình RDM

Sau khi đã thực hiện xong 3 bước trên, ta lập mô hình quan hệ.

Tiếp theo, cần điều chỉnh để có được mô hình dữ liệu hoàn chỉnh:

- So sánh ERD được xây dựng ban đầu với RDM để chỉnh lý những điểm khác biệt sao cho hai mô hình phải phản ánh chính xác lẫn nhau.
- Trong một số trường hợp, nhà phân tích có thể đưa vào hoặc loại bỏ những quan hệ phụ để làm trong sáng mô hình.



Hình 5.9. Biểu đồ quan hệ dữ liệu của hệ thống bán điện

CÂU HỎI THẢO LUẬN

- 5.1. Phân biệt thuộc tính khóa và thuộc tính kết nối.
- 5.2. Quan hệ 1 - 1 xuất hiện trong các trường hợp nào? Hãy cho ví dụ.
- 5.3. Quan hệ nhiều - nhiều được xử lý như thế nào trong mô hình ERD?
- 5.4. Quan hệ 1 - nhiều thường xuất hiện trong những trường hợp nào? Hãy cho ví dụ.
- 5.5. Phân biệt phụ thuộc hàm đơn trị và phụ thuộc hàm đa trị.
- 5.6. Phân tích mối quan hệ giữa các thực thể trong ERD của hệ thống quản lý kho ở hình 5.8.
- 5.7. Phân tích RDM của hệ thống quản lý bán điện trong hình 5.9.
- 5.8. Xây dựng RDM của hệ thống quản lý kho.





BÀI 6. THIẾT KẾ HỆ THỐNG

6.1. TỔNG QUAN VỀ THIẾT KẾ HỆ THỐNG

6.1.1. Các hướng tiếp cận thiết kế hệ thống

6.1.1.1. Các tiếp cận hướng mô hình

Thiết kế hướng mô hình (Model-driven) là một cách tiếp cận thiết kế hệ thống nhấn mạnh vào việc vẽ các mô hình hệ thống để tài liệu hóa các khía cạnh cài đặt và kỹ thuật của một hệ thống.

Các mô hình thiết kế thường được dẫn xuất từ các mô hình logic được phát triển trước đó theo cách phân tích hướng mô hình. Cuối cùng thì các mô hình thiết kế hệ thống sẽ trở thành các bản thiết kế phục vụ cho việc xây dựng và cài đặt hệ thống mới.

Trong tiếp cận hướng mô hình có 3 kỹ thuật là thiết kế hướng cấu trúc, kỹ thuật thông tin và thiết kế hướng đối tượng. Ngày nay, các tiếp cận hướng mô hình thường được củng cố nhờ vào việc sử dụng các công cụ tự động hóa. Các công cụ thường dùng:

- Công cụ đi kèm bộ công cụ lập trình: Oracle Designer

- Các công cụ đơn giản: MS.Word, MS.Visio, Smartdraw...

- Các công cụ chuyên dụng: Rational Rose, Rational XDE for platforms...

Thiết kế hướng cấu trúc hiện đại (Modern Structured Design):

- Là kỹ thuật phân rã chức năng hệ thống ra thành nhiều phần, mỗi thành phần lại được thiết kế chi tiết hơn ở các bước sau. Thiết kế hướng cấu trúc còn được gọi là thiết kế chương trình từ tổng quan đến chi tiết (top-down).
- Mỗi modun ở mức thấp nhất chỉ thực hiện một phần việc nhất định, ít liên quan đến công việc của các modun khác.
- Thường được sử dụng vì đơn giản, dễ hiểu, thuận tiện trong triển khai và nâng cấp.
- Mô hình phần mềm được dẫn xuất từ thiết kế hướng cấu trúc được gọi là biểu đồ cấu trúc (structure chart). Biểu đồ này được xây dựng từ các luồng dữ liệu trong chương trình. Thiết kế hướng cấu trúc được thực hiện trong giai đoạn phân tích hệ thống. Tuy nhiên, nó không bao trùm mọi khía cạnh của việc thiết kế, như thiết kế đầu vào/đầu ra hay cơ sở dữ liệu. Các ký hiệu trong biểu đồ cấu trúc:
 - Môđun: được biểu diễn bằng hình chữ nhật có nhãn là tên của môđun.
 - Dữ liệu được chuyển giao giữa các môđun: biểu diễn bởi mũi tên có đầu tròn rỗng.
 - Thông tin điều khiển: biểu diễn bằng mũi tên với đầu tròn đặc.

Kỹ thuật thông tin (Information Engineering):



- Là cách tiếp cận hướng mô hình và lấy dữ liệu làm trung tâm nhưng , chú trọng đến việc tổ chức các thông tin: nội dung thông tin quan hệ giữa các thành phần thông tin.
- Công cụ chủ yếu là sơ đồ mô hình dữ liệu
- Được sử dụng khi thiết kế chương trình dựa trên mô hình cơ sở dữ liệu quan hệ.

Làm bản mẫu (Prototyping)

- Bản mẫu là một chương trình nhỏ, chưa hoàn chỉnh nhưng đủ để cho người xem hình dung về chức năng, hoạt động của chương trình cần thực hiện:
- Làm bản mẫu là phương pháp mô hình hoá trên mã nguồn chứ không trên bản vẽ
- Thuận tiện để làm bản demo cho người dùng cuối xem (không đòi hỏi phải hiểu những ngôn ngữ mô hình hoá)
- Sớm phát hiện những sai khác về nghiệp vụ
- Chỉ phù hợp với các dự án nhỏ, ít phức tạp

Thiết kế hướng đối tượng (Object Oriented Design):

- Sử dụng cách tiếp cận theo tư duy hướng đối tượng - phân biệt rõ ràng hai yếu tố: dữ liệu (thuộc tính) chức năng (hành vi) và các mối tương tác (sự kiện)
- Là bước tiếp theo của Phân tích hệ thống hướng đối tượng, do đó đòi hỏi những kết quả từ bước trước: định nghĩa đối tượng, thuộc tính, hành vi, sự kiện...

6.1.1.2. Phát triển ứng dụng nhanh

Kỹ thuật xây dựng ứng dụng nhanh chóng bằng cách phối hợp sử dụng nhiều kỹ thuật:

- Tổ hợp thông tin
- Làm bản mẫu
- Kỹ thuật phát triển ứng dụng kết hợp (Joint Application Development): phát triển ứng dụng bằng cách gộp chung hai giai đoạn phân tích và thiết kế. Nhấn mạnh sự tham gia đồng thời của nhà phân tích, thiết kế, người dùng cuối, chuyên gia hệ thống trong quá trình xây dựng. Thường được dùng phát triển các ứng dụng nhỏ trong thời gian ngắn.

6.1.2. Các công việc cụ thể trong giai đoạn thiết kế

- Thiết kế kiến trúc ứng dụng: Lựa chọn công nghệ sử dụng cho dự án, Đưa ra mô hình vật lý của hệ thống.
- Thiết kế cơ sở dữ liệu
 - Đưa ra mô hình dữ liệu.
 - Lựa chọn hệ quản trị CSDL và tối ưu hoá mô hình dữ liệu theo hệ quản trị đã lựa chọn
- Thiết kế giao diện hệ thống: đầu ra, đầu vào, giao diện người dùng,...



- Đưa ra các đặc tả hệ thống cho lập trình viên.

Câu hỏi thảo luận

- 6.1. Đặc điểm của thiết kế hướng cấu trúc hiện đại là gì?
- 6.2. Nêu ý nghĩa của việc làm bản mẫu.
- 6.3. Nêu các công việc cụ thể trong giai đoạn thiết kế.

6.2. KIẾN TRÚC ỨNG DỤNG VÀ VIỆC MÔ HÌNH HOÁ

6.2.1. Kiến trúc ứng dụng

Kiến trúc hệ thống thông tin (KTHT) là một đặc tả về mặt công nghệ của một hệ thống thông tin.

KTHT dùng làm phương tiện để: Trao đổi về đặc tính của hệ thống (tập trung hay phân tán, CSDL, tính tích hợp, giao diện hệ thống...).

KTHT là cơ sở để triển khai hệ thống theo thiết kế và cũng là cơ sở để bảo trì hệ thống sau này.

6.2.2. Biểu đồ luồng dữ liệu vật lý

Biểu đồ luồng dữ liệu vật lý (Physical Data Flow Diagram - PDFD) là mô hình chức năng dùng để mô hình hoá kiến trúc hệ thống.

PDFD biểu diễn các thuộc tính của từng thành phần trong KTHT cũng như mô tả mối quan hệ, tương tác giữa các thành phần đó.

Dưới đây sẽ giới thiệu cách PDFD diễn tả từng đối tượng trong KTHT.

6.2.2.1. Chức năng vật lý

Chức năng vật lý là nơi thực hiện các công việc của hệ thống, đó có thể là con người, máy tính cá nhân, server, máy tính cầm tay...

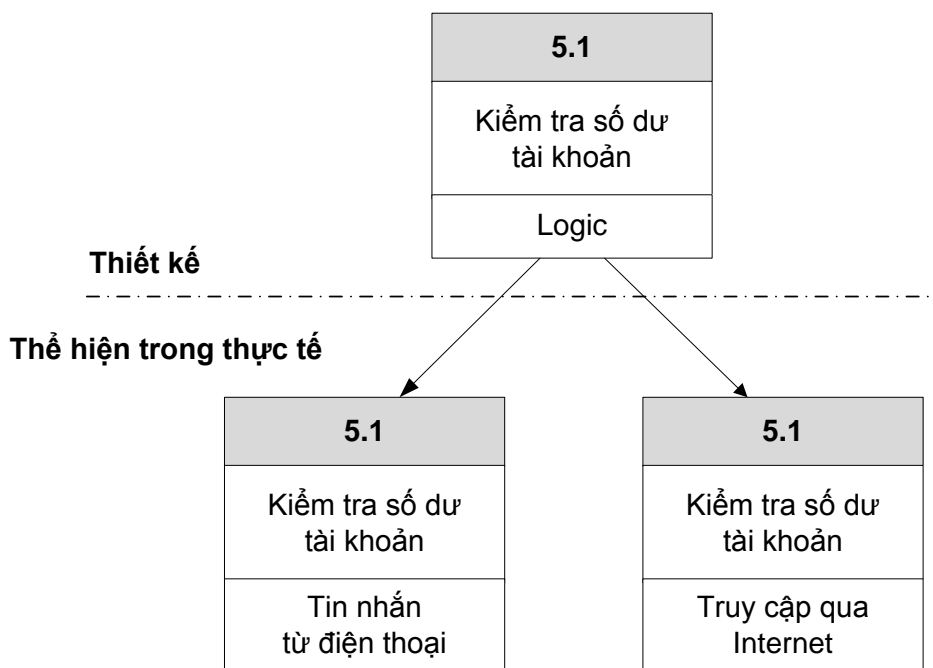
Mỗi hệ thống cần một hoặc một số chức năng vật lý.

PDFD giúp chúng ta thấy rõ: thông tin được xử lý bởi con người hay máy móc, thông tin được xử lý bởi công nghệ nào...

Ký hiệu:



Ví dụ PDFD biểu diễn một chức năng vật lý:

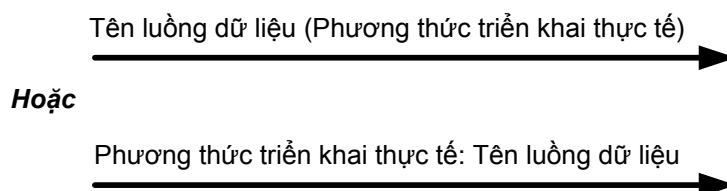


6.2.2.2. Luồng dữ liệu vật lý

Luồng dữ liệu vật lý:

- Mô tả các luồng dữ liệu đi luân chuyển trong hệ thống
- Các lệnh tương tác với CSDL: tạo, đọc, cập nhật, xóa các đối tượng csdl
- Nhập/xuất các phần tử dữ liệu giữa các thành phần trong mạng

Ký hiệu:



6.2.2.3. Kho dữ liệu vật lý

Các kho dữ liệu vật lý dùng để mô tả

- Một cơ sở dữ liệu
- Một bảng trong cơ sở dữ liệu
- Một file máy tính
- File tạm
- Một phương tiện lưu trữ dự phòng
- Một dạng lưu trữ dữ liệu phi máy tính (mã vạch, RFID, thẻ từ...)

Ký hiệu:

ID	Phương thức triển khai Tên kho dữ liệu
----	---



6.2.3. Kiến trúc công nghệ thông tin

Kiến trúc công nghệ thông tin (Information technology architecture) là một chủ đề phức tạp. Trong mục này, chúng ta chỉ nêu tóm tắt những xu thế công nghệ thông tin hiện đại có tác động tới các quyết định trong giai đoạn thiết kế.

6.2.3.1. Kiến trúc hệ thống:

- **Hệ phân tán** (Distributed system): là hệ thống trong đó các thành phần phân tán giữa những địa điểm, mạng, máy tính khác nhau: tính toán lưới (grid-computing, mạng máy tính dựa trên PC...).
- **Hệ tập trung** (Centralized system): *Đối lập với hệ phân tán*, Hệ tập trung là hệ thống trong đó các thành phần, các tác vụ xử lý tập trung tại một nơi (Mainframe).

Các hệ thống hiện đại là các hệ phân tán, nó giúp phân phối dữ liệu và các dịch vụ đến gần người dùng cuối hơn, cắt giảm sự phức tạp và chi phí đầu tư, bảo trì. Có 3 loại kiến trúc hệ thống phân tán:

Kiến trúc máy chủ tập (File server architecture)

- Là một mạng cục bộ (LAN) trong đó có một máy chủ chứa dữ liệu của một hệ thống thông tin. Mạng LAN là mạng nội bộ kết nối các máy tính(PC, Server, PDA...) trong một phạm vi hẹp (văn phòng, toà nhà...). Mạng LAN giúp tổng hợp năng lực các máy tính đơn lẻ trong mạng khi cho phép bất kỳ máy nào cũng có thể là máy chủ, bất kể máy nào cũng có thể là máy khách.
- Kiến trúc này cho phép nhiều máy tính cá nhân và máy trạm được kết nối để chia sẻ dữ liệu và giao tiếp với nhau.

Kiến trúc khách/chủ (Client/Server architecture): Là kiến trúc trong đó có một hay nhiều máy tính đóng vai trò máy chủ cung cấp các dịch vụ, dữ liệu cho một hay nhiều máy khách. Một số máy chủ điển hình:

- Máy chủ cơ sở dữ liệu (Database server): là máy chủ logic lưu trữ một hay nhiều cơ sở dữ liệu đồng thời cung cấp một hệ thống các câu lệnh cho phép thao tác với những cơ sở dữ liệu nói trên.
- Máy chủ ứng dụng (Application server): là máy chủ logic lưu trữ phần xử lý logic của một hay nhiều ứng dụng, cho phép các máy khách truy cập vào để thực thi ứng dụng.
- Máy chủ nhắn tin hoặc phần mềm nhóm (Message hoặc Groupware server): là máy chủ logic cung cấp các dịch vụ như email, lịch làm việc, các chức năng hỗ trợ làm việc nhóm.
- Máy chủ web (Webserver): là máy chủ logic lưu trữ và vận hành các website trên mạng internet hoặc intranet.

Kiến trúc tính toán dựa trên Internet (Internet-Based computing architecture)

- Là một dạng khác của kiến trúc phân tán đang góp phần định hình lại ý tưởng thiết kế của các nhà phân tích hệ thống và chuyên gia thông tin.



- Một hệ thống tính toán mạng là hệ thống trong đó các hệ thống thông tin đều chạy trên trình duyệt (ví dụ như hệ thống tài chính, hệ thống quản lý nhân sự...), lấy dữ liệu từ máy chủ web.

6.2.3.2. Kiến trúc dữ liệu

Cơ sở dữ liệu quan hệ lưu trữ dữ liệu dưới dạng bảng. Mỗi bảng bao gồm nhiều cột (giống các trường trong cơ sở dữ liệu dựa trên file), giao giữa các dòng và cột là các bản ghi (tương tự khái niệm bản ghi trên cơ sở dữ liệu file). Cơ sở dữ liệu quan hệ có một cơ sở toán học vững chắc và được dùng làm cơ sở dữ liệu của hầu hết các hệ thống hiện nay.

Cơ sở dữ liệu quan hệ phân tán là cơ sở dữ liệu quan hệ trong đó một hay nhiều bảng được nhân rộng và phân tán trên nhiều máy chủ cơ sở dữ liệu ở các nơi khác nhau.

Hệ quản trị cơ sở dữ liệu (HQTCSDL) là hệ thống quản lý việc lưu trữ, truy vấn, phân quyền truy nhập một hay nhiều cơ sở dữ liệu. HQTCSDL phân tán là một HQTCSDL làm thêm chức năng quản lý sự đồng bộ, kiểm soát truy nhập đối với các bảng dữ liệu phân tán.

Có 2 kỹ thuật:

- Data partitioning: phân mảnh và phân tán một hay nhiều trường dữ liệu giữa các server mà không có hoặc có rất ít sự trùng lặp.
- Data Replication: không phân mảnh mà nhân bội một hay nhiều trường rồi phân tán giữa các server.

6.2.3.3. Kiến trúc giao diện

Là các kênh giao tiếp giữa các trung tâm xử lý trong hệ thống hay giữa các hệ thống máy tính với nhau. Các kiểu giao tiếp: dữ liệu vào ra trực tuyến/theo bó, nhập liệu không cần bàn phím (mã vạch, thẻ từ, RFID), nhập liệu bằng bút cảm ứng, dữ liệu EDI, dữ liệu có được thông qua nhận dạng (vân tay, scan...), thông qua middleware...

6.3. THIẾT KẾ CƠ SỞ DỮ LIỆU

6.3.1. Các phương thức lưu trữ dữ liệu

Có hai phương thức lưu trữ dữ liệu phổ biến:

- File
- Cơ sở dữ liệu

6.3.1.1. File

Là một tập hợp của các bản ghi tương tự nhau. Các file không có liên quan với nhau trừ khi được liên kết trong code của chương trình ngoài.

Ưu điểm:

- Dễ dàng thiết kế nếu chỉ dùng cho một ứng dụng.
- Tối ưu về hiệu năng nếu chỉ dùng cho một ứng dụng.

Nhược điểm:

- Khó thích ứng hoặc khó dùng chung giữa nhiều ứng dụng.



- Hay bị dư thừa dữ liệu (cùng một thông tin lại được lưu trữ trên nhiều file khác nhau).

6.3.1.2. Cơ sở dữ liệu

Là một tập hợp của nhiều files (bảng) có quan hệ với nhau. Bản ghi của một file (hay bảng) có thể có mối quan hệ vật lý với một hay nhiều bản ghi ở các file (hay bảng) khác.

Ưu điểm:

- Tách biệt dữ liệu khỏi logic chương trình do đó tăng tính thích ứng, khả chuyên của chương trình.
- Kiểm soát được quy mô, độ lớn của dữ liệu
- Tối ưu trong việc chia sẻ dùng chung giữa nhiều ứng dụng
- Giảm thiểu dư thừa dữ liệu

Nhược điểm:

- Phức tạp hơn công nghệ file rất nhiều
- Ở khía cạnh nào đó truy xuất cơ sở dữ liệu thường chậm hơn so với truy xuất file.
 - o Cần tuân thủ nhiều nguyên tắc khi thiết kế để có thể khai thác được lợi ích của cơ sở dữ liệu quan hệ
 - o Cần có chuyên gia sử dụng hệ quản trị cơ sở dữ liệu

6.3.2. Kiến trúc dữ liệu

6.3.2.1. Kiến trúc dữ liệu mô tả cách thức:

- Sử dụng file/cơ sở dữ liệu để lưu trữ dữ liệu
- Công nghệ file/cơ sở dữ liệu được lựa chọn sử dụng
- Cơ cấu quản lý được thiết lập để quản lý các nguồn dữ liệu Thông thường dữ liệu được lưu trữ đồng thời bởi nhiều cách thức, phương tiện:
 - Các files,
 - Cơ sở dữ liệu cá nhân, cơ sở dữ liệu chung của nhóm, cơ sở dữ liệu giao dịch,
 - Nhà kho dữ liệu (tổng hợp các nguồn)...

6.3.2.2. Hệ quản trị CSDL:

- Là một phần mềm dùng để quản lý việc tạo, truy nhập, kiểm soát, quản lý các đối tượng dữ liệu của một hay nhiều cơ sở dữ liệu.
- Phần nền tảng của một HQTCSDL là một bộ máy dữ liệu - data engine
- Ngôn ngữ định nghĩa dữ liệu (Data Definition Language - DDL) là một phần của bộ máy dùng để định nghĩa các bảng, trường, quan hệ
- Ngôn ngữ thao tác dữ liệu (Data Manipulation Language - DML) dùng để thêm, sửa, xóa và di chuyển giữa các trường trong cơ sở dữ liệu.



6.3.3. Triển khai mô hình dữ liệu logic dựa trên một CSDL quan hệ

6.3.3.1. Cơ sở dữ liệu quan hệ

Là cơ sở dữ liệu lưu trữ và quản lý dữ liệu trong những bảng 2 chiều. Các bảng này có thể có quan hệ với nhau thông qua các trường khoá.

Đặc thù của cơ sở dữ liệu quan hệ:

- Mô hình dữ liệu vật lý (Schema) ở DDL và DML được thể hiện bởi ngôn ngữ SQL
- Triggers là các chương trình được nhúng cùng cơ sở dữ liệu và tự động thực thi khi cơ sở dữ liệu được cập nhật
- Thủ tục thường trú (Stored procedure) là chương trình được nhúng cùng cơ sở dữ liệu và thực thi từ câu lệnh của ứng dụng.

6.3.3.2. Mô hình hoá dữ liệu

Một mô hình dữ liệu tốt là mô hình trong đó:

- Mỗi thuộc tính mô tả một và chỉ một thực thể
- Mỗi thuộc tính chỉ tồn tại ở duy nhất một thực thể (trừ thuộc tính khoá ngoại)

Để có được một mô hình dữ liệu tốt, ta tiến hành các bước chuẩn hoá (*xem thêm phần phân tích hệ thống*)

Chuẩn hoá dữ liệu - Một thực thể logic hay một bảng vật lý được gọi là:

- Ở dạng chuẩn thứ nhất nếu không có thuộc tính (trường) nào có hai giá trị trong cùng một thể hiện
- Ở dạng chuẩn thứ hai nếu nó đã ở dạng chuẩn thứ nhất và giá trị các trường không phải là khoá chính hoàn toàn phụ thuộc vào khoá chính.
- Ở dạng chuẩn thứ ba nếu nó đã ở dạng chuẩn thứ hai và giá trị các trường không phải khoá chính không phụ thuộc các trường không phải khoá chính khác

Các bước tạo mô hình dữ liệu vật lý

- Xem lại mô hình dữ liệu logic
- Tạo bảng cho mỗi thực thể
- Tạo trường cho mỗi thuộc tính
- Tạo chỉ mục (index) cho mỗi khoá
- Thiết kế khoá ngoại cho các quan hệ
- Định nghĩa kích thước/kiểu dữ liệu, thuộc tính null, giá trị mặc định
- Đánh giá và thiết lập các ràng buộc
- Chú ý các công nghệ khác nhau cho các kiểu dữ liệu khác nhau

Tạo câu lệnh SQL

- Tùy vào từng hệ quản trị CSDL, tạo CSDL tương ứng trên ngôn ngữ DDL.



6.4. THIẾT KẾ ĐẦU VÀO

6.4.1. Tổng quan về thiết kế đầu vào

6.4.1.1. Các khái niệm

Có thể phân loại các phương thức nhập liệu theo hai đặc trưng: **cách thức** dữ liệu được thu thập, đưa vào và xử lý và **phương pháp và công nghệ** được dùng để thu thập và nhập dữ liệu.

Thu thập dữ liệu (data capture): nhận dạng và tạo dữ liệu mới từ nguồn tạo tin.

Nhập liệu (data entry): chuyển dữ liệu từ nguồn tạo tin vào máy tính

Xử lý dữ liệu (data processing): là quá trình biến đổi trực tiếp trên dữ liệu trước khi đưa nó về dạng máy tính có thể đọc được. Xử lý bó là thu thập 1 khối lượng dữ liệu và xử lý đồng thời cả bó. Xử lý trực tuyến là xử lý ngay lập tức dữ liệu vừa thu thập được.

6.4.1.2. Các phương thức nhập liệu

- Bàn phím
- Chuột
- Màn hình cảm ứng (màn hình tương tác)
- Nhận dạng âm thanh, tiếng nói
- Tự động nhập liệu: mã vạch, nhận dạng quang học, mực từ, thẻ từ, thẻ thông minh, sinh trắc học...

6.4.1.3. Các nguyên tắc thiết kế đầu vào

Nên tuân theo những nguyên tắc dưới đây khi thiết kế phương thức nhập liệu:

- Không nên nhập những dữ liệu có thể tính toán được từ những dữ liệu khác.

Ví dụ: Số lượng x Đơn giá = Thành tiền

- Không nhập những dữ liệu có thể lưu trong máy tính như những hằng số.
- Sử dụng mã lấy từ cơ sở dữ liệu đối với những thuộc tính phù hợp.
- Sử dụng các chỉ dẫn nhập liệu khi thiết kế các form nhập liệu (tooltip).
- Giảm thiểu số lượng ký tự gõ vào để tránh gây sai sót. Thay vào đó, cố gắng dùng các hộp check chọn càng nhiều càng tốt.
- Dữ liệu nhập vào theo trình tự từ trên xuống dưới, trái qua phải.

6.4.1.4. Kiểm soát nhập liệu

Việc kiểm soát dữ liệu đầu vào rất cần thiết trong tất cả các hệ thống ứng dụng trên máy tính. Các điều khiển đầu vào đảm bảo rằng dữ liệu đầu vào là chính xác và hệ thống được bảo vệ khỏi các lỗi vô ý hoặc hữu ý.

Số lượng đầu vào cần phải được theo dõi, đặc biệt là trong trường hợp nhập dữ liệu theo bó:



- Lưu mã số giao dịch cho bó các dữ liệu nhập liệu theo bó.
- Ghi các log file cho các dữ liệu được nhập trực tuyến

Phải kiểm soát tính đúng đắn của dữ liệu nhập vào. Phải làm các kiểm tra về: trùng lặp thực thể, kiểu dữ liệu, định dạng, tính ràng buộc với các dữ liệu khác.

Ví dụ: Khi nhập liệu thành phố và quốc gia cho một hồ sơ nhân sự, nếu đã chọn tỉnh là Đắk Lắk thì chỉ cho phép chọn huyện và thành phố là Buôn Ma Thuột, EaKar hoặc các huyện và thành phố khác ở tỉnh Đắk Lắk ... chứ không cho phép chọn thành phố thuộc tỉnh khác như Nha Trang, Tuy Hòa chẳng hạn.

6.4.2. Các điều khiển giao diện cho thiết kế đầu vào

6.4.2.1. Một số điều khiển phổ biến

- **Hộp văn bản** (Text box): chứa một hộp hình chữ nhật kèm theo tên, cho phép nhập dữ liệu vào.
- **Nút chọn loại trừ** (Radio button): chứa một hình trong nhỏ kèm theo một đoạn văn bản mô tả tương ứng với giá trị lựa chọn. Trong một nhóm các nút này thì chỉ cho phép chọn một nút mà thôi.
- **Hộp chọn kiểm tra** (Check box): chứa một hộp hình vuông kèm theo đoạn văn bản mô tả trường dữ liệu vào, người dùng sẽ chọn giá trị Yes/No. Trong một nhóm các hộp chọn thì có thể chọn nhiều hộp.
- **Hộp danh sách** (List box): là một hình chữ nhật chứa một hoặc nhiều dòng dữ liệu.
- **Danh sách thả** (Drop down list): chứa hộp chọn hình chữ nhật và một nút bên cạnh. Khi nhấn vào nút đó thì danh sách sẽ được thả xuống.
- **Hộp thả kết hợp** (Combination box): cũng là một danh sách thả nhưng cho phép người dùng nhập thêm dữ liệu ngoài những dữ liệu có sẵn trong đó.
- **Nút lệnh** (Button): các nút lệnh không phải là điều khiển vào. Chúng không dành cho việc lựa chọn dữ liệu vào. Mục đích của chúng là cho phép người dùng xác nhận rằng tất cả các dữ liệu cần được xử lý hay hủy bỏ một giao dịch hoặc cần gọi chức năng trợ giúp... Tóm lại, nút lệnh đóng vai trò gọi tới một chức năng nào đó.

6.4.2.2. Một số điều khiển cao cấp

- **Hộp lịch thả** (Drop down calendar): là một ô dữ liệu có chứa một nút mũi tên. Khi nhấn chuột vào đó thì hộp lịch được thả xuống để chọn ngày.
- **Điều khiển hiệu chỉnh trượt** (Slider edit control): cho phép lựa chọn giá trị bằng cách trượt con trỏ.
- **Điều khiển hiệu chỉnh mặt nạ** (Masked edit control): điều khiển này tạo ra định dạng để buộc dữ liệu nhập vào phải tuân theo.
- **Hộp danh sách chọn** (Check list box): điều khiển này được dùng để kết hợp nhiều hộp chọn kiểm tra.



- **Hộp cây chọn** (Check tree box): điều khiển này được dùng để biểu diễn các lựa chọn dữ liệu dưới dạng cây phân cấp.

6.4.3. Quy trình thiết kế đầu vào

- Xác định các dữ liệu đầu vào của hệ thống và các yêu cầu nhập liệu logic.
- Lựa chọn các điều khiển thích hợp.
- Thiết kế, lập cơ chế kiểm soát nhập liệu, lưu vết
- Nếu cần thiết, lập hồ sơ đặc tả đầu vào.

6.5. THIẾT KẾ ĐẦU RA

6.5.1. Tổng quan về thiết kế đầu ra

6.5.1.1. Phân loại đầu ra

Một cách để phân loại đầu ra là dựa vào hình thức phân phối chúng trong hay ngoài tổ chức và đối tượng người sẽ đọc và sử dụng chúng. Hình thức đầu ra chủ yếu là dưới dạng các báo cáo.

Báo cáo nội bộ: là các báo cáo được cung cấp cho người dùng hệ thống trong tổ chức

- Báo cáo chi tiết: thông tin trực tiếp truy xuất từ dữ liệu hệ thống, ví dụ: danh sách khách hàng.
- Báo cáo tóm lược: thông tin sau khi truy xuất đã được sắp xếp theo thứ tự thuận tiện cho người dùng quan sát, đôi khi kết quả được thể hiện dưới dạng đồ họa, ví dụ: khách hàng theo khu vực
- Báo cáo ngoại lệ: thông tin cảnh báo, đột xuất theo sự kiện thay đổi về chất lượng, điều kiện của hệ thống.

Báo cáo bên ngoài: là các báo cáo cung cấp cho khách hàng, nhà cung cấp, cơ quan pháp luật...

Báo cáo quay vòng: là các loại báo cáo bên ngoài sau đó lại trở về hệ thống như là một phương thức thu thập dữ liệu, chẳng hạn bản điều tra, hoá đơn...

6.5.1.2. Các phương thức cài đặt đầu ra

- In ra trên giấy
- Hiển thị trên màn hình, trên trang web
- Xuất dưới dạng đa phương tiện
- Gửi thư trực tiếp
- Tạo các đường liên kết

6.5.2. Cách thức thiết kế đầu ra

6.5.2.1. Các nguyên tắc thiết kế đầu ra

Báo cáo phải đơn giản, dễ hiểu, dễ giải thích, bao gồm:

- Tiêu đề rõ ràng.
- Ghi rõ ngày giờ phát hành
- Có các phân ghi thông tin chung



Thông tin phải được thể hiện ở dạng người dùng bình thường không được tùy ý sửa chữa.

Thông tin hiển thị phải hài hoà giữa các trang

Cung cấp cách di chuyển giữa các ô thật sự đơn giản

Thời gian xuất báo cáo phải được kiểm soát

Một số hình thức báo cáo phải được sự đồng ý của công ty.

6.5.2.2. Quy trình thiết kế đầu ra

- Xem xét tất cả các đầu ra của hệ thống và các yêu cầu logic.
- Làm rõ đặc tả (vật lý) của các yêu cầu báo cáo.
- Thiết kế các bản mẫu trước (nếu cần).
- Thiết kế, kiểm thử và kiểm tra đầu ra của báo cáo

6.6. THIẾT KẾ GIAO DIỆN NGƯỜI DÙNG

6.6.1. Tổng quan về giao diện người dùng

Giao diện người dùng hiệu quả phải phù hợp với trình độ và kinh nghiệm của người dùng. Những nguyên nhân sau đây khiến cho người dùng sử dụng sai hay cảm thấy nhầm chán, lẫn lộn thậm chí hoảng sợ quay sang chối bỏ phần mềm:

- Sử dụng nhầm lẫn các thuật ngữ, khái niệm
- Giao diện không trực quan
- Cách tiếp cận giải quyết vấn đề bị lẫn lộn
- Thiết kế giao diện rắc rối

Các nguyên tắc nên áp dụng khi thiết kế giao diện người dùng:

- Phải hiểu rõ trình độ người sử dụng cũng như đặc thù các công việc của họ.
- Lôi kéo người dùng vào việc thiết kế giao diện
- Kiểm tra và thử nghiệm việc thiết kế trên người dùng thật
- Áp dụng các quy ước, thói quen trong thiết kế giao diện, tuân thủ style chung cho toàn chương trình.
- Người dùng cần được chỉ dẫn những công việc họ sẽ đối mặt tiếp theo.

Chỉ cho người dùng hệ thống đang mong đợi họ làm gì

Chỉ cho người dùng dữ liệu họ nhập đúng hay sai

Giải thích cho người dùng hệ thống đang đứng yên do có công việc cần xử lý chứ không treo

Khẳng định với người dùng hệ thống đã hay chưa hoàn thành một công việc nào đó

- Nên định hình giao diện sao cho các thông điệp, chỉ dẫn luôn xuất hiện tại cùng một vị trí
- Định hình các thông điệp và chỉ dẫn đủ dài để người dùng có thể đọc được, đủ ngắn để họ có thể hiểu được



- Các giá trị mặc định cần được hiển thị
- Lường trước những sai sót người dùng có thể gặp phải để phòng tránh
- Không cho phép xử lý tiếp nếu lỗi chưa được khắc phục

6.6.2. Kỹ thuật giao diện người dùng

6.6.2.1. Hệ điều hành và trình duyệt web

Những hệ điều hành đồ họa phổ biến cho các máy khách hiện nay là Windows, Macintosh, Unix, Linux và cho các máy cầm tay là Palm OS, Windows CE. Tuy nhiên, hệ điều hành ngày càng không còn là nhân tố chính trong thiết kế giao diện người dùng nữa. Các ứng dụng Internet và Intranet chạy trên các trình duyệt web. Hầu hết các trình duyệt có thể chạy trên nhiều hệ điều hành. Điều này cho phép thiết kế giao diện người dùng ít phụ thuộc vào hệ điều hành. Tính năng này được gọi là độc lập nền tảng (platform independence). Thay vì viết giao diện riêng cho từng hệ điều hành thì chỉ cần viết giao diện cho một hoặc hai trình duyệt. Hiện tại, hai trình duyệt phổ biến nhất là Microsoft Internet Explorer và Firefox nhưng vẫn còn tồn tại một khó khăn khác đó là vấn đề về các phiên bản trình duyệt.

6.6.2.2. Màn hình hiển thị

Kích thước vùng hiển thị là vấn đề then chốt khi thiết kế giao diện. Không phải màn hình hiển thị nào cũng là dạng màn hình máy tính cá nhân. Có rất nhiều thiết bị hiển thị không phải là máy tính cá nhân.

Đối với màn hình máy tính cá nhân, chúng ta có đơn vị đo lường là độ phân giải đồ họa. Độ phân giải đồ họa được tính theo pixel, đó là số điểm sáng phân biệt được hiển thị trên màn hình. Hiện nay, độ phân giải phổ biến là 800.000 pixel theo chiều ngang và 600.000 pixel theo chiều dọc trong một màn hình 17 inch. Những kích thước hiển thị lớn hơn hỗ trợ nhiều pixel hơn; tuy nhiên, người thiết kế nên thiết kế giao diện theo loại màn hình có độ phân giải phổ biến nhất.

Rõ ràng, các máy tính cầm tay và một số thiết bị hiển thị đặc biệt (ví dụ như màn hình máy rút tiền tự động ATM) hỗ trợ màn hình hiển thị nhỏ hơn nhiều cũng phải được xem xét khi thiết kế giao diện.

Cách thức thể hiện vùng hiển thị đối với người dùng được điều khiển bởi cả khả năng kỹ thuật của màn hình và khả năng của hệ điều hành. Hai cách tiếp cận phổ biến nhất là paging và scrolling. **Paging** hiển thị một màn hình hoàn chỉnh các ký tự vào cùng một lần. Toàn bộ vùng hiển thị được gọi là một trang (hay màn hình). Các trang được hiển thị theo nhu cầu của người dùng bằng cách nhấn nút lệnh, tương tự như lật các trang trong một cuốn sách. **Scrolling** dịch chuyển phần thông tin hiển thị lên hoặc xuống trên màn hình, thường là mỗi lần 1 dòng. Các màn hình máy tính cá nhân còn cho phép nhiều tùy chọn paging và scrolling.



6.6.2.3. Bàn phím và các thiết bị trợ

Hầu hết (nhưng không phải tất cả) các thiết bị hiển thị và màn hình đều được tích hợp với bàn phím. Những tính năng chủ yếu của bàn phím là tập ký tự và các khóa chức năng.

Tập ký tự của hầu hết các máy tính cá nhân đều theo chuẩn. Những tập ký tự đó có thể được mở rộng với phần mềm để hỗ trợ thêm các ký tự và biểu tượng. Các khóa chức năng nên được sử dụng một cách nhất quán. Nghĩa là, bất kỳ chương trình nào cũng nên sử dụng nhất quán các khóa chức năng cho cùng mục đích. Ví dụ, F1 thường được dùng để gọi chức năng trợ giúp trong cả hệ điều hành và các ứng dụng.

Hầu hết các giao diện (bao gồm các hệ điều hành và trình duyệt) đều sử dụng thiết bị trợ như chuột, bút và màn hình cảm ứng. Tất nhiên, thiết bị trợ phổ biến nhất vẫn là chuột.

Bút đang trở nên quan trọng trong các ứng dụng chạy trên các thiết bị cầm tay. Bởi lý do là những thiết bị đó thường không có bàn phím. Do đó, giao diện có thể cần được thiết kế để cho phép “gõ” trên một bàn phím được hiển thị trên màn hình hoặc sử dụng một chuẩn viết tay như Graffiti hoặc Jot.

6.6.3. Các phong cách thiết kế giao diện người dùng

6.6.3.1. Giao diện dựa trên cửa sổ và frame

Phần cơ bản nhất của một giao diện là cửa sổ. Một cửa sổ có thể nhỏ hoặc lớn hơn vùng màn hình hiển thị. Nó thường chứa các điều khiển chuẩn ở góc trên bên phải như phóng to, thu nhỏ hay đóng cửa sổ.

Phần dữ liệu hiển thị bên trong cửa sổ có thể lớn hoặc nhỏ hơn kích thước cửa sổ. Trong trường hợp lớn hơn, có thể dùng thanh cuộn để dịch chuyển.

Một cửa sổ có thể được chia thành các vùng gọi là frame. Mỗi frame có thể hoạt động độc lập với các frame khác trong cùng một cửa sổ. Mỗi frame có thể được xác định để phục vụ cho một mục đích nhất định.

Trong một cửa sổ, chúng ta có thể sử dụng tất cả các điều khiển giao diện đã giới thiệu trong các mục 6.4 và 6.5.

6.6.3.2. Giao diện dựa trên menu

Chiến lược đối thoại phổ biến nhất và cổ điển nhất là menu. Có nhiều loại menu nhưng tư tưởng chung đều là yêu cầu người dùng chọn một hành động từ menu:

- Menu kéo thả, menu xếp tầng
- Menu pop-up
- Thanh công cụ và menu icon
- Menu siêu liên kết: thanh menu bar, menu xếp tầng, menu kéo thả,...



6.6.3.3. Giao diện dựa trên dòng lệnh

Thay cho menu hoặc cũng có thể bổ sung thêm cho menu, một số ứng dụng được thiết kế sử dụng đối thoại dựa trên tệp lệnh (còn gọi là giao diện ngôn ngữ lệnh – command language interface). Tuy nhiên, người sử dụng phải học cú pháp tập lập nên cách tiếp cận này chỉ phù hợp với đối tượng người dùng chuyên gia. Có 3 loại cú pháp, lựa chọn loại nào là phụ thuộc vào công nghệ có thể dùng:

Cú pháp dựa trên ngôn ngữ (ví dụ như SQL)

Cú pháp mnemonic: người sử dụng được cung cấp một màn hình giao tiếp trong đó họ có thể nhập các lệnh gọi tới các hành động. Các câu lệnh phải có nghĩa với người sử dụng.

Cú pháp ngôn ngữ tự nhiên: cho phép người dùng nhập các câu hỏi các lệnh bằng ngôn ngữ tự nhiên. Hệ thống thông dịch các lệnh đó theo cú pháp đã biết và có thể yêu cầu người dùng nhập lại rõ ràng hơn nếu nó không hiểu được ý muốn của người dùng.

Cách thiết kế giao diện dựa trên dòng lệnh từng phổ biến trong các ứng dụng máy tính lớn và các ứng dụng máy tính cá nhân trên DOS trước đây. Nhưng phong cách tương tác này vẫn được sử dụng trong một số ứng dụng hiện nay. Ví dụ như Microsoft Access có phần soạn thảo câu lệnh truy vấn.

Thiết kế giao diện người dùng là việc đặc tả sự đối thoại giữa người sử dụng chương trình và máy tính. Sự đối thoại này thường cho kết quả là dữ liệu đầu vào và thông tin đầu ra. Có một số phong cách thiết kế giao diện người dùng. Trước đây, những hình thức đó được xem là loại bỏ nhau nhưng ngày nay, chúng đang pha trộn lẫn nhau. Mục này giới thiệu tổng quan một số phong cách và chiến lược được dùng để thiết kế giao diện người dùng và cách thức chúng được kết hợp vào các ứng dụng.

6.6.3.4. Đối thoại hỏi – đáp

Hình thức đối thoại hỏi đáp được dùng chủ yếu để hỗ trợ cho đối thoại dựa trên menu hoặc dựa trên câu lệnh. Người dùng được gợi ý bằng câu hỏi mà họ cần cho câu trả lời. Câu hỏi đơn giản nhất là Yes/No. Chiến lược này yêu cầu chúng ta phải xét mọi câu trả lời đúng có thể có và chuẩn bị mọi hành động nếu xuất hiện câu trả lời sai. Rõ ràng đây là một hình thức giao diện khó thiết kế. Tuy nhiên, hình thức này phổ biến trong các ứng dụng trên web.

6.6.3.5. Một số tính năng đặc biệt

Xác thực và phân quyền

Trong hầu hết các hệ thống, người sử dụng phải được xác thực trước khi họ được phép sử dụng hệ thống. Nói một cách khác, người sử dụng phải đăng nhập vào hệ thống. Hầu hết việc đăng nhập đều yêu cầu tên người dùng (username) và mật khẩu (password). Mỗi người dùng được cấp một



quyền hạn sử dụng một số chức năng nhất định. Người dùng có quyền cao nhất thường là người quản trị hệ thống.

Trợ giúp

- Tooltip: xuất hiện khi người dùng đưa chuột vào vị trí biểu tượng (hoặc đối tượng) trên màn hình. Tooltip chứa một đoạn mô tả ngắn gọn về chức năng thể hiện bởi đối tượng tương ứng.
- Help Wizard: hướng dẫn người sử dụng thông qua một quá trình phức tạp bằng cách đưa ra một chuỗi các hộp đối thoại yêu cầu người dùng phải đưa ra đầu vào và trả về phản hồi của hệ thống.
- Tác tử (Agents): là các đối tượng phần mềm có thể hoạt động trên nhiều ứng dụng phần mềm và thậm chí là trên các mạng. Ví dụ như tác tử trợ giúp của Microsoft (có thể hiểu là trợ lý) cung cấp một trợ lý chung trong các ứng dụng Office. Nó cho phép người dùng đặt câu hỏi bằng một đoạn ngôn ngữ tự nhiên. Đoạn ngôn ngữ sẽ được thông dịch bởi tác tử để đưa ra đáp ứng phù hợp. Sau đó, người dùng có thể lựa chọn một trong các đáp ứng để chọn ra chỉ mục trợ giúp chi tiết hơn nữa.

6.6.4. Cách thức thiết kế giao diện người dùng

6.6.4.1. Các công cụ tạo giao diện

Microsoft Access
CASE Tools
Visual Basic
Visio,...

6.6.4.2. Quy trình thiết kế giao diện người dùng

- Bước 1: Lập sơ đồ phân cấp giao tiếp người dùng hoặc sử dụng lược đồ biến đổi trạng thái
- Bước 2: Lập bản mẫu đối thoại và giao diện người dùng
- Bước 3: Tham khảo và tiếp thu ý kiến phản hồi của người dùng. Nếu cần thiết quay lại bước 1 và bước 2.

CÂU HỎI THẢO LUẬN

- 6.1. Đặc điểm của thiết kế hướng cấu trúc hiện đại là gì?
- 6.2. Nêu ý nghĩa của việc làm bản mẫu.
- 6.3. Nêu các công việc cụ thể trong giai đoạn thiết kế.
- 6.4. Ý nghĩa của biểu đồ luồng dữ liệu vật lý.
- 6.5. Nêu khái niệm, phân tích ưu và nhược điểm của hệ thống phân tán.
- 6.6. Nêu ưu nhược điểm của phương thức lưu trữ dữ liệu dạng file.



- 6.7. Nêu ưu nhược điểm của phương thức lưu trữ dữ liệu sử dụng cơ sở dữ liệu.
- 6.8. Nêu tóm tắt các phương thức nhập liệu.
- 6.9. Nêu các nguyên tắc thiết kế nhập liệu.
- 6.10. Nêu các phương thức cài đặt đầu ra.
- 6.11. Nêu các nguyên tắc thiết kế đầu ra.
- 6.12. Nêu các nguyên tắc trong thiết kế giao diện.
- 6.13. Nêu các kỹ thuật giao diện.
- 6.14. So sánh các phong cách giao diện.



**MỤC LỤC**

BÀI 1. HỆ THỐNG THÔNG TIN	1
1.1. HỆ THỐNG THÔNG TIN	1
1.1.1. Các khái niệm.....	1
1.1.2. Các loại hệ thống thông tin cơ bản.....	2
1.1.3. Những đặc trưng cơ bản của hệ thống:.....	2
1.2. PHƯƠNG THỨC XỬ LÝ THÔNG TIN TRONG MÁY TÍNH.....	3
1.2.1. Xử lý tương tác:	3
1.2.2. Xử lý giao dịch:.....	3
1.2.3. Xử lý theo lô:	3
1.2.4. Xử lý trực tuyến:.....	3
1.2.5. Xử lý thời gian thực:.....	3
1.2.6. Xử lý phân tán:.....	4
1.3. CÁC GIAI ĐOẠN TRIỂN KHAI DỰ ÁN XÂY DỰNG HTTT	4
1.4. CÁC MÔ HÌNH BIỂU DIỄN HỆ THỐNG	4
1.4.1. Biểu đồ phân cấp chức năng (BFD - Business Flow Diagram):.....	4
1.4.2. Biểu đồ luồng dữ liệu (DFD - Data Flow Diagram):.....	5
1.4.3. Phân mức các chức năng của hệ thống:	6
BÀI 2. KHẢO SÁT HIỆN TRẠNG VÀ THU THẬP THÔNG TIN	8
2.1. PHƯƠNG PHÁP PHÒNG VẤN	8
2.1.1. 2.1.1. Dạng câu hỏi mở.....	8
2.1.2. Dạng câu hỏi đóng.....	9
2.1.3. Các dạng câu hỏi mở khác	9
2.1.4. Thứ tự đặt câu hỏi	9
2.1.5. Một số việc phải làm khi kết thúc và sau phỏng vấn:	10
2.2. PHƯƠNG PHÁP DÙNG PHIẾU HỎI.....	10
2.2.1. Thiết kế phiếu hỏi.....	11
2.2.2. Các phương pháp phát phiếu hỏi.....	11
2.3. PHƯƠNG PHÁP LẤY MẪU	12
2.3.1. Các phương pháp thiết kế mẫu.....	12
2.3.2. Các kiểu lấy mẫu	12
2.4. PHÂN TÍCH TÀI LIỆU ĐỊNH TÍNH/ ĐỊNH LƯỢNG.....	13
2.4.1. Phân tích tài liệu định lượng.....	13
2.4.2. Phân tích tài liệu định tính	13
2.5. PHƯƠNG PHÁP QUAN SÁT	14
BÀI 3. PHÂN TÍCH HỆ THỐNG.....	15
3.1. KHÁI NIỆM PHÂN TÍCH HỆ THỐNG	15
3.1.1. Phân tích hệ thống (System Analysis):	15
3.1.2. Thiết kế hệ thống (System Design):.....	15
3.2. CÁC HƯỚNG TIẾP CẬN PHÂN TÍCH HỆ THỐNG.....	15
3.2.1. Tiếp cận phân tích hướng mô hình	15
3.2.2. Tiếp cận phân tích hệ thống nhanh.....	15
3.3. CÁC GIAI ĐOẠN PHÂN TÍCH HỆ THỐNG.....	15
3.3.1. Giai đoạn xác định phạm vi.....	15
3.3.2. Giai đoạn phân tích vấn đề.....	16
3.3.3. Giai đoạn phân tích yêu cầu.....	17
3.3.4. Giai đoạn mô hình hóa logic	18
3.3.5. Giai đoạn phân tích quyết định.....	18



3.4. XÁC ĐỊNH CÁC YÊU CẦU CỦA NGƯỜI DÙNG	19
3.4.1. Vai trò của việc xác định yêu cầu	19
3.4.2. Tiêu chuẩn của việc xác định yêu cầu.....	19
3.4.3. Quy trình xác định yêu cầu.....	20
BÀI 4. MÔ HÌNH HÓA CHỨC NĂNG	21
4.1. MÔ HÌNH HÓA HỆ THỐNG.....	21
4.1.1. Các bước mô hình hóa hệ thống	21
4.1.2. Mục đích của mô hình hóa hệ thống.....	21
4.1.3. Các thao tác mô hình hóa chức năng.....	22
4.1.4. Khái niệm hệ thống	22
4.2. MÔ HÌNH LOGIC	23
4.2.1. Phân biệt mô hình logic và mô hình vật lý.....	23
4.2.2. Sự cần thiết của mô hình logic	23
4.3. BIỂU ĐỒ PHÂN RÃ CHỨC NĂNG (BUSSINESS FUNCTIONAL DIAGRAM)	23
4.3.1. Khái niệm BFD	23
4.3.2. Phương pháp xây dựng BFD.....	24
4.4. BIỂU ĐỒ LƯỒNG DỮ LIỆU (DATA FLOW DIAGRAM)	25
4.4.1. Mô hình hóa chức năng với DFD	25
4.5. CÁC PHẦN TỬ CỦA DFD.....	26
4.5.1. Các bộ ký hiệu của DFD	26
4.5.2. Tác nhân ngoài.....	27
4.5.3. Kho dữ liệu	27
4.5.4. Luồng dữ liệu.....	28
4.5.5. Quá trình	28
4.6. BIỂU ĐỒ LƯỒNG DỮ LIỆU MỨC NGŨ CẢNH	29
4.7. TRÌNH TỰ VÀ QUY TẮC XÂY DỰNG DFD.....	30
4.7.1. Các bước xây dựng DFD	30
4.7.2. Các quy tắc xây dựng DFD.....	31
4.7.3. Phân tích hướng cấu trúc cổ điển (top-down).....	31
4.7.4. Phân tích hướng cấu trúc hiện đại (hướng sự kiện).....	32
4.7.5. So sánh DFD với biểu đồ luồng tiến trình (flowchart)	32
4.7.6. Sự bảo toàn dữ liệu	32
BÀI 5. MÔ HÌNH HOÁ DỮ LIỆU	34
5.1. MÔ HÌNH HÓA DỮ LIỆU.....	34
5.1.1. Khái niệm.....	34
5.1.2. Từ mô hình dữ liệu tới cài đặt cơ sở dữ liệu.....	34
5.1.3. Vai trò của biểu đồ quan hệ thực thể.....	34
5.2. CÁC PHẦN TỬ CỦA BIỂU ĐỒ QUAN HỆ THỰC THỂ (ERD)	34
5.2.1. Thực thể (Entity).....	34
5.2.2. Thuộc tính (Properties).....	35
5.2.3. Mối quan hệ	37
5.3. CÁC BƯỚC XÂY DỰNG BIỂU ĐỒ QUAN HỆ THỰC THỂ	39
5.3.1. Các bước mô hình hóa dữ liệu logic	39
5.3.2. Trình tự xây dựng ERD	40
5.3.3. Qui tắc xây dựng ERD.....	42
5.3.4. Các kiểu ký hiệu ERD khác	42
5.4. XÂY DỰNG BIỂU ĐỒ DỮ LIỆU QUAN HỆ (RDM)	43
5.4.1. Xác định thuộc tính	43
5.4.2. Phân tích và chuẩn hóa dữ liệu	43
5.4.3. Xác định các mối quan hệ.....	45



5.4.4. Xây dựng mô hình RDM 46

BÀI 6. THIẾT KẾ HỆ THỐNG 48

6.1. TỔNG QUAN VỀ THIẾT KẾ HỆ THỐNG 48

 6.1.1. Các hướng tiếp cận thiết kế hệ thống 48

 6.1.2. Các công việc cụ thể trong giai đoạn thiết kế 49

6.2. KIẾN TRÚC ỨNG DỤNG VÀ VIỆC MÔ HÌNH HOÁ 50

 6.2.1. Kiến trúc ứng dụng 50

 6.2.2. Biểu đồ luồng dữ liệu vật lý 50

 6.2.3. Kiến trúc công nghệ thông tin 52

6.3. THIẾT KẾ CƠ SỞ DỮ LIỆU 53

 6.3.1. Các phương thức lưu trữ dữ liệu 53

 6.3.2. Kiến trúc dữ liệu 54

 6.3.3. Triển khai mô hình dữ liệu logic dựa trên một CSDL quan hệ 55

6.4. THIẾT KẾ ĐẦU VÀO 56

 6.4.1. Tổng quan về thiết kế đầu vào 56

 6.4.2. Các điều khiển giao diện cho thiết kế đầu vào 57

 6.4.3. Quy trình thiết kế đầu vào 58

6.5. THIẾT KẾ ĐẦU RA 58

 6.5.1. Tổng quan về thiết kế đầu ra 58

 6.5.2. Cách thức thiết kế đầu ra 58

6.6. THIẾT KẾ GIAO DIỆN NGƯỜI DÙNG 59

 6.6.1. Tổng quan về giao diện người dùng 59

 6.6.2. Kỹ thuật giao diện người dùng 60

 6.6.3. Các phong cách thiết kế giao diện người dùng 61

 6.6.4. Cách thức thiết kế giao diện người dùng 63

MỤC LỤC

