MỤC LỤC

thức cũng như kinh nghiệm sư phạm đã được *TS.* ***Nguyễn Kim Anh*** thể hiện trong cuốn sách này. Đây không phải là cuốn sách (tiếng Việt) duy nhất và cũng không phải là cuốn sách cuối cùng về CSDL ở Việt Nam, tuy nhiên chắc chắn rằng nó sẽ là cuốn cẩm nang kiến thức đáng tin cậy và thực sự bổ ích cho bạn đọc, đặc biệt là những sinh viên các ngành *Toán-Tin, Tin học, Công nghệ Thông tin* ở các trường đại học và cao đẳng trong cả nước.

Xin trân trọng giới thiệu cùng bạn đọc.

GS.TS. Nguyễn Thúc Hải

CHƯƠNG 1
ĐẠI CƯƠNG VỀ CÁC HỆ CƠ SỞ DỮ LIỆU

1.1. Các hệ thống xử lý tệp truyền thống

Cách tiếp cận truyền thống đối với các hệ thống thông tin thường chỉ tập trung vào các nhu cầu xử lý dữ liệu của các phòng riêng lẻ trong một tổ chức mà không xem xét tổ chức này như một tổng thể. Các hệ thống thông tin này đáp ứng các yêu cầu của những người sử dụng bằng cách viết một chương trình mới đối với mỗi ứng dụng mới đơn lẻ, thường là một chương trình được phát triển tại một thời điểm. Mỗi chương trình ứng dụng hay một hệ thống theo yêu cầu sẽ được thiết kế để đáp ứng các yêu cầu của một phòng đặc biệt hay một nhóm người sử dụng cụ thể. Do vậy, không có một kế hoạch tổng thể hay một mô hình hướng đến sự tăng trưởng các ứng dụng mới trong tương lai.

Mỗi ứng dụng tin học mới được thiết kế một cách điển hình với một tập các tệp dữ liệu riêng của nó. Nhiều dữ liệu trong các tệp mới có thể đã xuất hiện trong các tệp đang tồn tại sẽ phải được cấu trúc lại, mà đến lượt nó, các chương trình đang tồn tại sử dụng cùng các tệp này cũng phải được sửa lại hay viết lại hoàn toàn. Với lý do này, thiết kế các tệp mới với mỗi ứng dụng được yêu cầu thường là đơn giản hơn và cũng ít rủi ro hơn.

Trước tiên, chúng ta hãy xem xét ví dụ về một phần của hệ thống Ngân hàng tiết kiệm (Quỹ tiết kiệm) lưu giữ thông tin về khách hàng và các tài khoản tiết kiệm của khách hàng. Một cách để lưu trữ thông tin này trên máy tính là lưu trữ nó trong các tệp hệ thống bền vững. Để cho phép người sử dụng thao tác những thông tin này, hệ thống có một số các chương trình ứng dụng thao tác các tệp đó, chẳng hạn như:

- Chương trình thêm một tài khoản mới;
- Chương trình ghi nợ (ghi có) một tài khoản;
- Chương trình tính số dư của một tài khoản.

Các chương trình ứng dụng này được viết bởi các nhà lập trình hệ thống để đáp ứng các yêu cầu tin học hoá của Ngân hàng tiết kiệm.

Các chương trình ứng dụng mới có thể bổ sung thêm vào hệ thống khi các nhu cầu mới nảy sinh.

Hệ thống xử lý tệp điển hình vừa mô tả được hỗ trợ bởi một hệ điều hành nào đó. Các bản ghi khá ổn định và tồn tại lâu dài được lưu trữ trong các tệp khác nhau và các chương trình ứng dụng khác nhau được viết để trích ra các bản ghi từ các tệp thích hợp hay bổ sung thêm các bản ghi mới vào các tệp đó. Như vậy, trước khi xuất hiện các phần mềm hệ quản trị cơ sở dữ liệu, trong quá khứ các hệ thống trên cơ sở tệp đã được tạo lập để xử lý một số lượng lớn các dữ liệu của các tổ chức một cách hiệu quả. Trong các hệ thống như vậy, việc xử lý dữ liệu được hoàn thành bởi việc tạo các tệp trên các đĩa từ hay các băng từ.

Tuy nhiên, khi qui mô kinh doanh của các tổ chức và nhu cầu xử lý thao tác dữ liệu với các mục đích khác nhau theo các qui cánh khác nhau tăng lên, một số các vấn đề nghiêm trọng có thể nảy sinh.

Các hệ thống trên cơ sở tệp được phát triển với các ứng dụng đặc biệt, do vậy việc lưu trữ thông tin của các tổ chức trong các hệ thống tệp có một số bất lợi chính sau:

- Dư thừa dữ liệu và tính không nhất quán dữ liệu: Do các tệp và các chương trình ứng dụng được tạo lập bởi các nhà lập trình khác nhau trong một khoảng thời gian dài, các tệp khác nhau có thể có các qui cách khác nhau và các chương trình có thể được viết bằng một số ngôn ngữ lập trình khác nhau. Hơn nữa, vì các tệp được tạo lập đối với mỗi một chương trình ứng dụng riêng, do vậy cùng một tập các dữ liệu của tổ chức có thể được lưu trữ lại trong mỗi chương trình ứng dụng, có nghĩa là một số thông tin có thể lặp lại trên một số tệp khác nhau và các tài nguyên phần cứng như các đĩa từ có thể bị lãng phí. Ví dụ, địa chỉ và số điện thoại của của một khách hàng đặc biệt có thể xuất hiện trong một tệp chứa các bản ghi tài khoản tiết kiệm và trong một tệp chứa các bản ghi số dư tài khoản. Sự dư thừa này dẫn đến giá truy nhập và giá lưu trữ cao hơn. Thêm vào đó, vì các dữ liệu được lưu trữ trong các tệp có thể được thay đổi một cách độc lập bởi các chương trình ứng dụng sử dụng chúng, nội dung của cùng các khoản mục dữ liệu trong chương trình ứng dụng này có thể không trùng với cùng các khoản mục dữ liệu đó trong một chương trình ứng dụng khác. Điều này dẫn đến tính không nhất quán dữ liệu, có nghĩa là các bản sao khác nhau của cùng một khoản mục dữ liệu không giống nhau.Ví dụ, một địa chỉ khách hàng bị thay đổi có thể được phản ánh trong bản ghi tài khoản tiết kiệm nhưng không được phản ánh trong bản ghi số dư tài khoản đối với khách hàng đó.

- Khó khăn trong truy nhập dữ liệu: Giả sử rằng một trong các nhân viên của Ngân hàng cần tìm danh sách tên các khách hàng sống trong một thành phố nào đó. Nhân viên này hỏi phòng xử lý dữ liệu để đưa ra một danh sách như vậy. Vì yêu cầu này không được dự tính trước, hệ thống gốc được thiết kế không có sẵn một chương trình ứng dụng để đáp ứng yêu cầu này.Tuy nhiên, có một chương trình ứng dụng đưa ra danh sách tất cả các khách hàng. Bấy giờ, nhân viên Ngân hàng này đứng trước hai lựa chọn: Hoặc sử dụng danh sách tất cả các khách hàng và trích ra những thông tin cần thiết bằng tay hoặc yêu cầu phòng xử lý dữ liệu đề nghị một nhà lập trình hệ thống viết chương trình ứng dụng mới đáp ứng yêu cầu mới này. Cả hai lựa chọn hiển nhiên đều không được hài lòng. Giả sử rằng, một chương trình như vậy được viết, và sau vài ngày, cũng nhân viên đó cần một danh sách khách hàng khác. Cũng như vậy, một chương trình đưa ra danh sách đó không tồn tại và nhân viên này lại đứng trước hai lựa chọn và không có lựa chọn nào được hài lòng. Điều chúng tôi muốn chỉ ra ở đây là các môi trường xử lý tệp không cho phép các dữ liệu cần thiết được tìm kiếm trong một phương pháp hiệu quả và tiện lợi. Do vậy, các hệ thống tìm kiếm dữ liệu thích hợp hơn cần phải được phát triển và phải đáp ứng được các ứng dụng khác nhau của hệ thống.
- Cô lập và hạn chế chia sẻ dữ liệu: Với cách tiếp cận truyền thống, mỗi ứng dụng có các tệp dữ liệu riêng của nó và những người sử dụng ít có cơ hội chia sẻ dữ liệu ngoài các ứng dụng riêng của họ. Một hậu quả của việc chia sẻ dữ liệu bị hạn chế là cùng các dữ liệu phải lưu trữ trong nhiều tệp ứng với các ứng dụng khác nhau do khi thiết kế phát triển các ứng dụng mới, người thiết kế thường khó khai thác các dữ liệu chứa trong các tệp đang tồn tại trong khi các tệp mới được thiết kế lặp lại nhiều dữ liệu đang tồn tại. Hơn nữa, các dữ liệu được lưu trữ trong các tệp khác nhau và các tệp có thể có các qui cách khác nhau, điều này dẫn đến khó khăn khi viết các chương trình ứng dụng mới để tìm kiếm các dữ liệu thích hợp trên nhiều tệp.
- Các vấn đề về toàn vẹn: Các giá trị dữ liệu được lưu trữ trong cơ sở dữ liệu phải thoả mãn các kiểu ràng buộc toàn vẹn nhất định. Các nhà phát triển làm hiệu lực các ràng buộc này trong hệ thống bằng cách thêm các đoạn mã tương ứng vào các chương trình ứng dụng. Tuy nhiên, khi các ràng buộc mới được bổ sung thêm đối với cơ sở dữ liệu, sẽ rất khó thay đổi các chương trình để làm hiệu lực các ràng buộc mới này. Vấn đề này phức tạp hơn khi các ràng buộc đó lôi kéo một số khoản mục dữ liệu từ các tệp khác nhau.

- Các vấn đề về độ tin cậy: Một hệ thống máy tính cũng như một thiết bị điện tử hay cơ học nào đó có thể bị lỗi. Trong nhiều ứng dụng, vấn đề mấu chốt là khi một lỗi xuất hiện và được dò tìm phát hiện, dữ liệu phải được phục hồi đến trạng thái toàn vẹn tồn tại trước khi xảy ra lỗi. Ví dụ, xét chương trình ứng dụng chuyển 50 đô la từ tài khoản A đến tài khoản B. Nếu hệ thống xảy ra lỗi trong khi thực hiện chương trình, có thể rằng, 50 đô la đã chuyển đi từ tài khoản A nhưng chưa được ghi có vào tài khoản B, kết quả là cơ sở dữ liệu ở trong một trạng thái không toàn vẹn dữ liệu.
- Các dị thường truy nhập đồng thời: Để tăng hiệu năng tổng thể của hệ thống và thời gian đáp ứng nhanh nhất có thể, nhiều hệ thống cho phép nhiều người sử dụng truy nhập dữ liệu đồng thời. Trong một môi trường như vậy, sự tương tác của các truy nhập đồng thời có thể dẫn đến kết quả trong một trạng thái không toàn vẹn dữ liệu. Ví dụ, xét tài khoản Ngân hàng A chứa 500 đô la. Nếu hai khách hàng đến rút quỹ (rút 50 đô la và 100 đô la tương ứng) từ tài khoản A vào cùng một thời điểm. Kết quả của các thực hiện đồng thời này có thể dẫn đến tài khoản này ở trong một trạng thái không đúng đắn (hay không toàn vẹn). Giả sử rằng, các chương trình rút thực hiện yêu cầu rút bao gồm các thao tác : đọc giá trị tài khoản, giảm giá trị này đi một giá trị bằng số cần rút và ghi kết quả trả lại. Nếu hai chương trình chạy đồng thời, cả hai có thể cùng đọc giá trị 500 đô la và ghi lại 450 đô la , 400 đô la tương ứng. Phụ thuộc vào chương trình nào ghi giá trị sau cùng, tài khoản A có thể chứa hoặc 450 đô la hoặc 400 đô la chứ không phải giá trị đúng của nó là 350 đô la. Để cảnh giác với những tình huống như vậy, hệ thống phải duy trì một dạng giám sát nào đó. Bởi vì dữ liệu có thể được truy nhập bởi nhiều chương trình ứng dụng khác nhau mà không được điều phối từ trước, một dạng giám sát như vậy khó có thể được cung cấp.
- Các vấn đề về an toàn: Không phải mọi người sử dụng của hệ thống đều có thể truy nhập tất cả dữ liệu trong hệ. Ví dụ, trong hệ thống Ngân hàng, các nhân viên tài vụ của Ngân hàng chỉ cần nhìn thấy một phần của cơ sở dữ liệu chứa thông tin về các nhân viên Ngân hàng. Họ không cần thiết truy nhập vào thông tin về các tài khoản khách hàng. Do các chương trình ứng dụng được bổ sung vào hệ thống theo phương pháp thêm dần, việc bắt tuân thủ các ràng buộc an toàn như vậy là khá khó khăn.
- Sự phụ thuộc dữ liệu của các chương trình ứng dụng: Do định nghĩa tệp được chứa trong các chương trình ứng dụng, nếu các nội dung tệp và các khuôn dạng bản ghi cần được thay đổi, các chương trình ứng dụng cũng phải thay đổi theo.

Những khó khăn này dẫn đến cần phải phát triển một phần mềm đặc biệt, một hệ quản trị cơ sở dữ liệu với đầy đủ các tính năng cho phép khắc phục các bất lợi nêu trên.

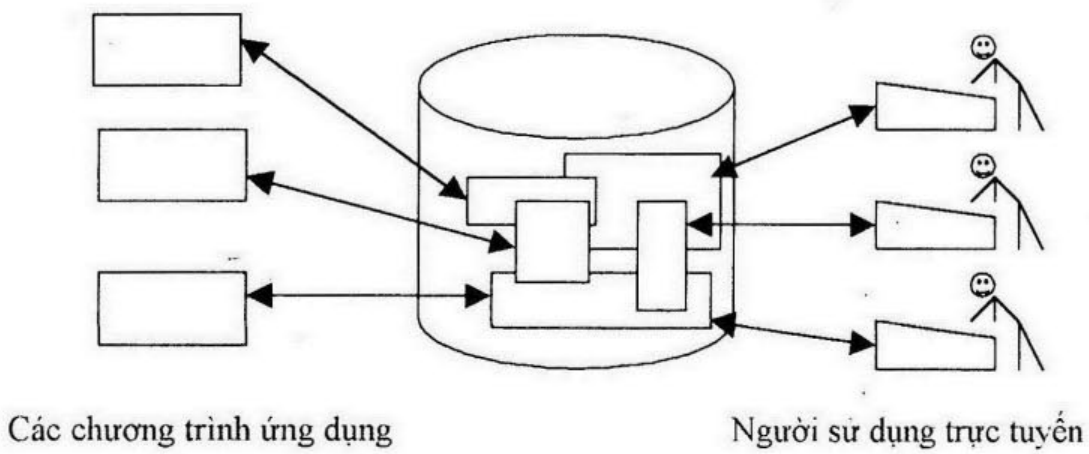
1.2. Các hệ cơ sở dữ liệu

Trong thời gian gần đây ngành tin học đã có nhiều thành tựu hết sức to lớn. Trong số những thành tựu đó phải kể tới việc sử dụng ngày càng rộng rãi và có hiệu quả các hệ cơ sở dữ liệu. Trước tiên chúng ta hãy xem xét các khái niệm cơ bản về hệ cơ sở dữ liệu.

1.2.1. Các khái niệm cơ bản

Cơ sở dữ liệu là gì ?

Một tổ chức (xí nghiệp, ngân hàng, bệnh viện, cơ quan...) có nhiều ứng dụng tin học khác nhau nhưng tất cả các ứng dụng đó cùng được tiến hành trên một nguồn dữ liệu chung thì không gì tốt hơn là triển khai các ứng dụng đó trên một cơ sở dữ liệu hợp nhất cho phép quản lý tập trung tất cả dữ liệu xí nghiệp.



Hình 1.1. Cơ sở dữ liệu hợp nhất

Trên hình 1.1, chúng ta nhìn thấy thành phần cơ sở dữ liệu hợp nhất là một bộ sưu tập các dữ liệu chứa trên các phương tiện lưu trữ như đĩa từ hay băng từ... Những người sử dụng trực tuyến hay các chương trình ứng dụng có thể sử dụng chung hay chia sẻ cơ sở dữ liệu này .