BỘ GIÁO DỤC VÀ ĐÀO TẠO

**TRƯỜNG ĐẠI HỌC THĂNG LONG**

---o0o---

**BÀI TẬP LỚN MÔN   
CƠ SỞ DỮ LIỆU PHÂN TÁN**

**XÂY DỰNG CƠ SỞ DỮ LIỆU QUẢN LÝ   
CHO THUÊ TRUYỆN PHÂN TÁN**

**Giáo viên hướng dẫn: Nhóm 1 – Sinh viên thực hiện:**

**Đậu Hải Phong A30093 – Nguyễn Tuấn Sơn**

**A30337 – Bùi Đức Toàn**

**A30819 – Nguyễn Đăng Tú**

**Hà Nội – 2021**

**MỤC LỤC**

[Chương 1. Tổng quan về cơ sở dữ liệu phân tán 4](#_Toc74607554)

[1.1. Xử lý dữ liệu phân tán 4](#_Toc74607555)

[1.2. Hệ cơ sở dữ liệu phân tán 5](#_Toc74607556)

[1.3. Kiến trúc phân tán 8](#_Toc74607557)

[1.3.1. Khái niệm 8](#_Toc74607558)

[1.3.2. Các kiến trúc phân tán 8](#_Toc74607559)

[1.4. Khả năng của cơ sở dữ liệu phân tán 10](#_Toc74607560)

[1.4.1. Quản lý minh bạch dữ liệu phân tán và dữ liệu nhân bản 10](#_Toc74607561)

[1.4.2. Độ tin cậy thông qua các giao dịch phân tán 12](#_Toc74607562)

[1.4.3. Cải thiện hiệu suất 13](#_Toc74607563)

[1.4.4. Dễ dàng mở rộng hệ thống 13](#_Toc74607564)

[1.5. Các vấn đề cần quan tâm trong phân tán 13](#_Toc74607565)

[1.5.1. Thiết kế cơ sở dữ liệu phân tán 13](#_Toc74607566)

[1.5.2. Quản lý thư mục phân tán 14](#_Toc74607567)

[1.5.3. Xử lý truy vấn phân tán 14](#_Toc74607568)

[1.5.4. Kiểm soát đồng thời phân tán 14](#_Toc74607569)

[1.5.5. Quản lý bế tắc phân tán 14](#_Toc74607570)

[1.5.6. Độ tin cậy của DBMS phân tán 14](#_Toc74607571)

[1.5.7. Nhân bản 15](#_Toc74607572)

[Chương 2. Xây dựng cơ sở dữ liệu phân tán cho ứng dụng quản lý cho thuê truyện 16](#_Toc74607573)

[2.1. Mô tả bài toán 16](#_Toc74607574)

[2.1.1. Mô tả yêu cầu dữ liệu 16](#_Toc74607575)

[2.1.2. Mô hình thực thể quan hệ 19](#_Toc74607576)

[2.1.3. Lược đồ quan hệ 20](#_Toc74607577)

[2.2. Thiết kế phân tán 21](#_Toc74607578)

[2.2.1. Kịch bản phân mảnh dọc 21](#_Toc74607579)

[2.2.2. Kịch bản phân mảnh ngang 30](#_Toc74607580)

[2.3. Kiểm soát dữ liệu 33](#_Toc74607581)

[2.3.1. Xây dựng trigger chèn 33](#_Toc74607582)

[2.3.2. Xây dựng trigger xóa 34](#_Toc74607583)

[2.4. Tối ưu câu truy vấn 36](#_Toc74607584)

[2.5. Xây dựng các thao tác dữ liệu 65](#_Toc74607585)

[2.5.1. Xây dựng thủ tục chèn 65](#_Toc74607586)

[2.5.2. Xây dựng thủ tục sửa 66](#_Toc74607587)

[2.5.3. Xây dựng thủ tục xóa 67](#_Toc74607588)

# Tổng quan về cơ sở dữ liệu phân tán

## Xử lý dữ liệu phân tán

Rất khó để có thể định nghĩa chính xác thuật ngữ xử lý phân tán (hay tính toán phân tán). Có thể thấy một số mức độ xử lý phân tán diễn ra trong bất kỳ hệ thống máy tính nào, ngay cả trên các máy tính một bộ xử lý có bộ xử lý trung tâm (CPU) và các chức năng đầu vào/đầu ra (I/O) được tách biệt và chồng chéo lên nhau. Sự tách biệt và chồng chéo này có thể được coi là một trong những hình thức xử lý phân tán. Sự xuất hiện rộng rãi của các máy tính song song đã làm điều này thêm phức tạp, vì sự khác biệt giữa hệ thống máy tính phân tán và một số dạng máy tính song song khá là mơ hồ.

Thuật ngữ xử lý phân tán có thể là thuật ngữ được lạm dụng nhiều nhất trong khoa học máy tính trong những năm qua. Nó thường được dùng để chỉ những hệ thống gồm nhiều loại thiết bị khác nhau chẳng hạn như: hệ đa bộ xử lý, xử lý dữ liệu phân tán, mạng máy tính, …

Có thể định nghĩa xử lý phân tán là một tập hợp các phần tử xử lý tự trị (không nhất thiết đồng nhất) được kết nối với nhau bởi một mạng máy tính và cùng phối hợp thực hiện những công việc gán cho chúng. Phần tử xử lý ở đây chỉ một thiết bị tính toán có khả năng thực hiện chương trình trên nó.

Có hai khái niệm xử lý phân tán liên quan với nhau:

* Khái niệm liên quan đến việc tính toán trên Client/Server. Trong đó ứng dụng được chia thành hai phần (phần của Server và phần của Client), được vận hành ở hai nơi. Tính toán phân tán này cho phép truy nhập trực tiếp dữ liệu và xử lý dữ liệu trên Server và Client.
* Khái niệm thứ hai là việc thực hiện các tác vụ xử lý phức tạp trên nhiều hệ thống. Không gian nhớ và bộ xử lý của nhiều máy cùng hoại động chia nhau tác vụ xử lý. Máy trung tâm sẽ giám sát và quản lý các tiến trình này. Có trường hợp thông qua Internet, hàng nghìn máy cùng xử lý một tác vụ.

Một câu hỏi cơ bản cần được đặt ra là: Cái gì đang được phân phối? Một trong những thứ có thể được phân phối là xử lý logic. Trên thực tế định nghĩa của một hệ thống máy tính phân tán được đưa ra ở trên mặc nhiên giả định rằng logic xử lý hoặc các phần tử xử lý được phân phối. Một phân phối khác là theo chức năng. Các chức năng khác nhau của hệ thống máy tính có thể được ủy quyền cho các phần cứng hoặc phần mềm khác nhau. Phương thức phân phối thứ ba là theo dữ liệu. Dữ liệu được sử dụng bởi một số ứng dụng có thể được phân phối cho số lượng trạm xử lý. Cuối cùng, kiểm soát có thể được phân phối. Sự kiểm soát của việc thực hiện các nhiệm vụ khác nhau có thể được phân phối thay vì được thực hiện bởi một hệ thống máy tính. Từ quan điểm của hệ thống cơ sở dữ liệu phân tán, các phân phối này đều cần thiết và quan trọng.

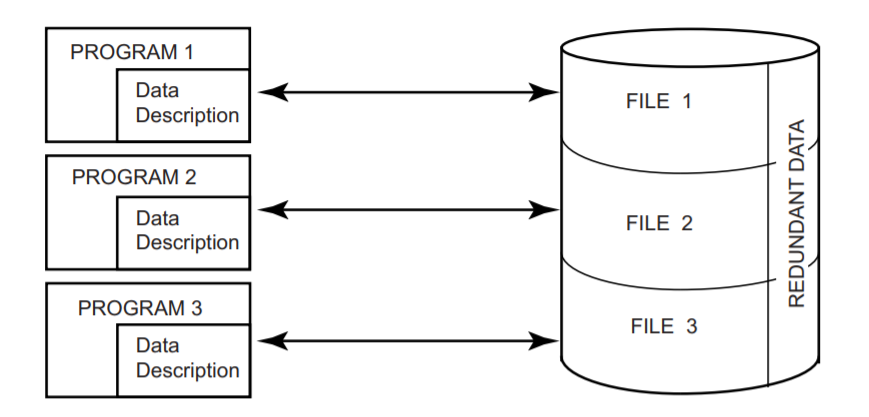
Một câu hỏi hợp lý khác cần đặt ra là: Tại sao chúng ta lại phân phối? Các câu trả lời cổ điển cho câu hỏi này chỉ ra rằng xử lý phân tán tốt hơn tương ứng với cơ cấu tổ chức của các doanh nghiệp phân tán rộng rãi hiện nay và một hệ thống như vậy đáng tin cậy hơn và phản hồi nhanh hơn. Quan trọng hơn, nhiều ứng dụng hiện tại của công nghệ máy tính vốn đã được phân phối. Các ứng dụng dựa trên web, kinh doanh thương mại điện tử qua Internet, các ứng dụng đa phương tiện như tin tức theo yêu cầu hoặc hình ảnh y tế, hệ thống điều khiển sản xuất là tất cả các ví dụ về các ứng dụng đó.

Tuy nhiên, từ góc độ toàn cầu, có thể nói rằng lý do cơ bản đằng sau quá trình xử lý phân tán là để có thể đối phó tốt hơn với các vấn đề quản lý dữ liệu quy mô lớn mà chúng ta phải đối mặt ngày nay, bằng cách sử dụng một biến thể của phép phân chia nổi tiếng - chia để trị. Nếu có thể phát triển phần mềm hỗ trợ cần thiết cho quá trình xử lý phân tán, có thể giải quyết những vấn đề phức tạp bằng cách chia chúng thành các phần nhỏ hơn và gán chúng cho các nhóm phần mềm khác nhau, hoạt động trên các máy tính khác nhau và tạo ra một hệ thống chạy trên nhiều phần tử xử lý nhưng có thể hoạt động hiệu quả để thực hiện một nhiệm vụ chung.

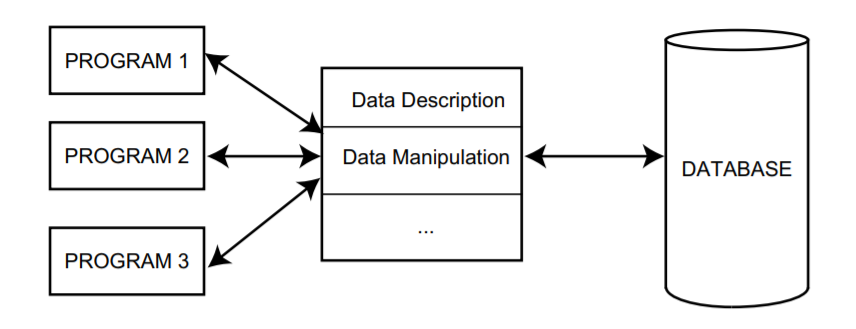
Các hệ thống cơ sở dữ liệu phân tán cũng được xem là nằm trong khuôn khổ này và được coi như các công cụ có thể làm cho quá trình xử lý phân tán dễ dàng hơn và hiệu quả hơn. Sẽ là hợp lý khi rút ra một sự tương tự giữa những gì cơ sở dữ liệu phân tán có thể cung cấp cho thế giới xử lý dữ liệu và những gì công nghệ cơ sở dữ liệu đã cung cấp. Không còn nghi ngờ gì nữa, việc phát triển các hệ thống cơ sở dữ liệu phân tán có mục đích chung, có thể thích ứng và hiệu quả đã hỗ trợ rất nhiều cho nhiệm vụ phát triển phần mềm phân tán.

## Hệ cơ sở dữ liệu phân tán

Công nghệ các hệ cơ sở dữ liệu phân tán (DDBS) là sự kết hợp của hai cách tiếp cận đối lập hoàn toàn đối với việc xử lý dữ liệu: hệ thống cơ sở dữ liệu và công nghệ mạng máy tính. Các hệ thống cơ sở dữ liệu đã đưa chúng ta từ một mô hình xử lý dữ liệu trong đó mỗi ứng dụng xác định và duy trì dữ liệu của riêng mình (Hình 1.1) sang một mô hình mà dữ liệu được xác định và quản lý tập trung (Hình 1.2). Định hướng mới này dẫn đến tính độc lập của dữ liệu, theo đó các chương trình ứng dụng miễn nhiễm với những thay đổi trong tổ chức logic hoặc vật lý của dữ liệu, và ngược lại.



Hình 1.1. Xử lý dữ liệu truyền thống

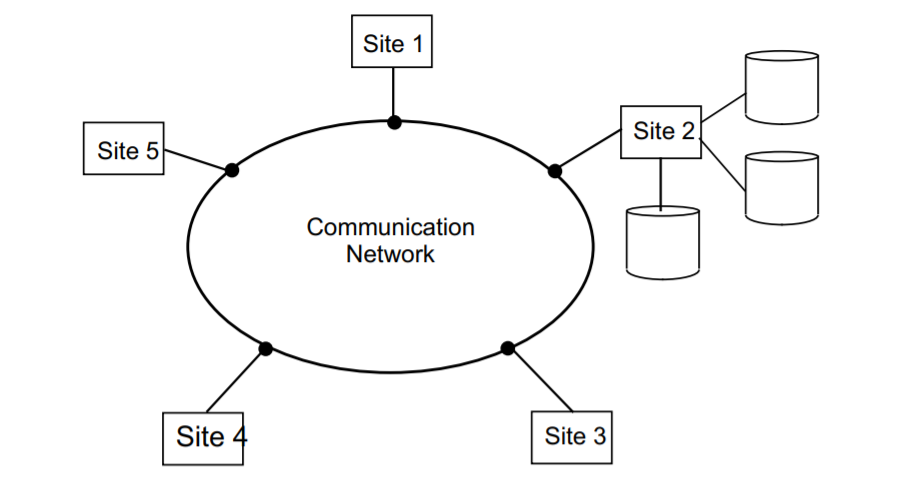


Hình 1.2. Xử lý cơ sở dữ liệu

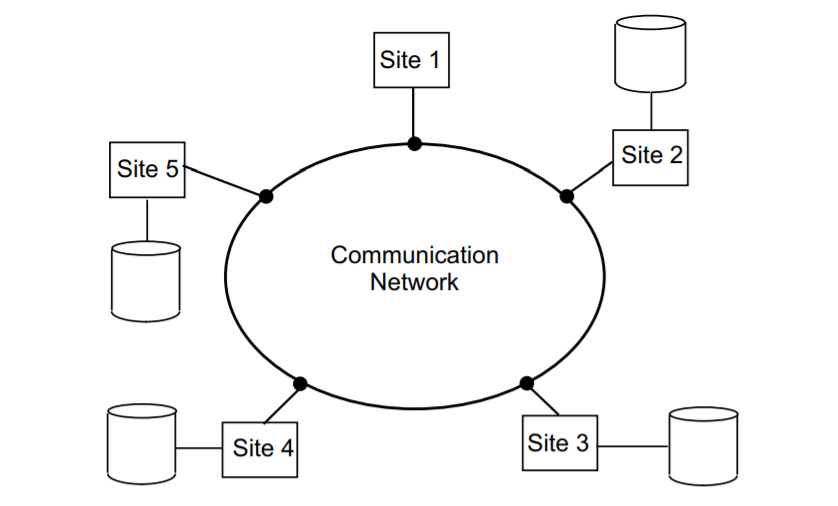
Hay nói cách khác, hệ cơ sở dữ liệu phân tán là một tập hợp nhiều cơ sở dữ liệu có liên quan với nhau về mặt logic được phân tán trên mạng máy tính. Sau đó, một hệ quản trị cơ sở dữ liệu phân tán (DBMS phân tán) được định nghĩa là hệ thống phần mềm cho phép quản lý cơ sở dữ liệu phân tán và làm cho việc phân phối trở nên minh bạch với người dùng. Đôi khi hệ cơ sở dữ liệu phân tán (DDBS) được sử dụng để chỉ cơ sở dữ liệu phân tán và DBMS phân tán. Hai thuật ngữ quan trọng trong các định nghĩa này là “có liên quan với nhau về mặt logic” và “được phân phối qua mạng máy tính”. Chúng giúp loại bỏ một số trường hợp nhất định mà đôi khi được chấp nhận để đại diện cho một DDBS.

Đôi khi người ta cho rằng việc phân phối dữ liệu vật lý không phải là vấn đề quan trọng nhất. Do đó, những người ủng hộ quan điểm này sẽ cảm thấy thoải mái khi gắn nhãn là cơ sở dữ liệu phân tán một số cơ sở dữ liệu (có liên quan) nằm trong cùng một hệ thống máy tính. Tuy nhiên, phân phối vật lý của dữ liệu là quan trọng. Nó tạo ra các vấn đề không gặp phải khi các cơ sở dữ liệu nằm trong cùng một máy tính. Phân phối vật lý không nhất thiết phải là các hệ thống máy tính cách xa nhau về mặt địa lý. Nó chỉ đơn giản là giao tiếp giữa các máy tính được thực hiện qua mạng thay vì thông qua bộ nhớ dùng chung hoặc đĩa dùng chung (như trường hợp với các hệ thống đa xử lý), với mạng là tài nguyên được chia sẻ duy nhất.

DDBS cũng không phải là một hệ thống mà mặc dù có sự tồn tại của mạng nhưng cơ sở dữ liệu chỉ nằm ở một nút của mạng (Hình 1.3). Trong trường hợp này, các vấn đề về quản lý cơ sở dữ liệu không khác gì các vấn đề gặp phải trong môi trường cơ sở dữ liệu tập trung. Cơ sở dữ liệu được quản lý tập trung bởi một hệ thống máy tính (trạm 2 trong Hình 1.3) và tất cả các yêu cầu được chuyển đến trạm đó. Việc xem xét bổ sung duy nhất là sự chậm trễ của đường truyền. Rõ ràng là sự tồn tại của một mạng máy tính hoặc một tập hợp “các file” là không đủ để tạo thành một hệ thống cơ sở dữ liệu phân tán. Điều chúng ta cần quan tâm là một môi trường nơi dữ liệu được phân phối giữa một số trạm (Hình 1.4).



Hình 1.3. Các giải pháp thay thế cung cấp dữ liệu



Hình 1.4. Môi trường DDBS

Hệ CSDL phân tán không đơn thuần bao gồm nhiều file dữ liệu được tổ chức lưu trữ riêng lẻ trên các thiết bị nhớ của mạng máy tính. Để tạo một hệ CSDL phân tán, các file không chỉ có quan hệ với nhau về mặt logic mà còn cần có một cấu trúc giao diện chung giữa chúng để các file có thể truy cập lẫn nhau. Có rất nhiều ứng dụng yêu cầu các hệ quản trị CSDL thao tác trên dữ liệu bán cấu trúc hoặc không cấu trúc, như các file web trên Internet.

## Kiến trúc phân tán

### Khái niệm

Kiến trúc của một hệ thống xác định cấu trúc của nó. Điều này có nghĩa là các thành phần của hệ thống đã được xác định, chức năng của từng thành phần đã được xác định và các mối quan hệ và tác động qua lại giữa các thành phần này đã được xác định. Đặc tả kiến trúc của một hệ thống yêu cầu xác định các mô-đun khác nhau, với các giao diện và mối quan hệ qua lại của chúng, về dữ liệu và luồng điều khiển qua hệ thống.

Đây là những quan điểm “lý tưởng hóa” của DBMS trong đó nhiều hệ thống thương mại có thể đi chệch hướng với chúng. Tuy nhiên, các kiến ​​trúc sẽ đóng vai trò như một khuôn khổ hợp lý trong đó các vấn đề liên quan đến DBMS phân tán có thể được thảo luận.

Trước tiên, chúng ta bắt đầu với phần trình bày ngắn gọn về “kiến trúc ANSI/SPARC” (American National Standards Institute – Standards Planning and Requirements Committee: Viện tiêu chuẩn quốc gia Mỹ – Ủy ban nhu cầu và kế hoạch), đây là một phương pháp tiếp cận theo danh mục dữ liệu để xác định kiến ​​trúc DBMS. Nó tập trung vào các lớp và vai trò người dùng khác nhau và các quan điểm khác nhau của họ về dữ liệu. Kiến trúc này rất hữu ích trong việc đưa một số khái niệm mà chúng ta đã thảo luận cho đến nay vào quan điểm thích hợp của chúng.

### Các kiến trúc phân tán

Có 3 kiểu kiến ​​trúc tham chiếu­[[1]](#footnote-1)\* được phát triển cho DBMS phân tán:

* Hệ thống Client/Server:
  + Phân chia các chức năng của một hệ thống thành 2 lớp: lớp Server và lớp Client. Server thực hiện hầu hết các công việc quản lý dữ liệu. Điều này có nghĩa là tất cả các công việc: xử lý truy vấn, tối ưu hóa truy vấn, quản lý giao dịch, quản lý thiết bị lưu trữ đều được thực hiện ở phía Server. Phía Client sẽ gồm các công việc: giao tiếp với người dùng, đệm và quản lý bộ đệm dữ liệu, kiểm tra tính nhất quán, tính phù hợp, bảo mật của các truy vấn người dùng tạo ra.
  + Có một số kiểu kiến trúc Client/Server khác nhau. Kiểu đơn giản nhất – đơn Server, đa Client – chỉ có một Server và nhiều Client truy vấn vào Server. Nói chung, kiểu này không khác nhiều so với hệ tập trung ngoài việc phân tán một số chức năng cho Client và thậm chí dành phần đệm dữ liệu cho phía Client. Một loại kiến trúc phức tạp hơn, đó là kiến trúc đa Server – đa Client. Trong kiến trúc này có hai hướng quản lý được đề cập:
    - Mỗi Client quản lý một kết nối của nó tới một Server phù hợp.
    - Mỗi Client có một “home” Server, Server này sẽ chịu trách nhiệm giao tiếp với các Server khác.
* Phân tán ngang hàng (Peer-to-Peer):
  + Kiến trúc này được coi thật sự là kiến trúc hệ CSDL phân tán. Chúng ta bắt đầu mô tả kiến trúc này bằng việc xem xét việc tổ chức dữ liệu. Phải chú ý rằng tổ chức dữ liệu vật lý ở các nút khác nhau có thể khác nhau. Điều này có nghĩa là cần phải có lược đồ cục bộ (local internal schema). Mô hình dữ liệu của cả tổ chức được biểu diễn bởi lược đồ khái niệm toàn cục (global conceptual schema).
  + Dữ liệu trong một hệ CSDL phân tán thường được phân mảnh và nhân bản, như vậy chúng ta cũng cần mô tả cách tổ chức dữ liệu logic ở các trạm, chúng ta sẽ xây dựng thêm tầng thứ ba trong kiến trúc đó là lược đồ khái niệm cục bộ (local conceptual schema). Trong kiến trúc này, lược đồ khái niệm toàn cục sẽ là tập hợp của các lược đồ khái niệm cục bộ. Cuối cùng, những ứng dụng của người sử dụng được hỗ trợ thông qua lược đồ bên ngoài (ESs), được định nghĩa ở mức trên lược đồ khái niệm toàn cục.
* Hệ thống đa cơ sở dữ liệu:
  + Sự khác nhau giữa kiến trúc đa CSDL và kiến trúc ngang hàng được quyết định bởi mức độ tự trị. Sự khác nhau cơ bản nằm ở định nghĩa lược đồ khái niệm toàn cục. Trong kiến trúc ngang hàng, lược đồ khái niệm toàn cục là mô hình dữ liệu cho toàn bộ hệ thống CSDL còn trong kiến trúc đa CSDL, lược đồ khái niệm toàn cục chỉ là một tập hợp của một vài CSDL mà các hệ quản trị CSDL cục bộ muốn chia sẻ. Kiến trúc này có hai mô hình:
    - Mô hình sử dụng một lược đồ khái niệm toàn cục: Lược đồ khái niệm toàn cục được xây dựng bằng cách tích hợp những lược đồ bên ngoài hay một phần của những lược đồ khái niệm cục bộ. Ngoài ra, người dùng ở những CSDL cục bộ vẫn có thể sử dụng những ứng dụng cũ để truy nhập vào CSDL đã có bên cạnh những người dùng toàn cục truy nhập vào nhiều CSDL.
    - Mô hình không sử dụng một lược đồ khái niệm toàn cục: Mô hình kiến trúc này gồm 2 tầng: Tầng hệ thống cục bộ và tầng đa CSDL. Tầng hệ thống cục bộ bao gồm một số DBMS. Những hệ quản trị này sẽ đem đến cho tầng trên những phần CSDL mà chúng muốn chia sẻ. Những dữ liệu được chia sẻ này ở dạng lược đồ khái niệm cục bộ thực sự hay một lược đồ ngoài cục bộ.

## Khả năng của cơ sở dữ liệu phân tán

### Quản lý minh bạch dữ liệu phân tán và dữ liệu nhân bản

Tính minh bạch đề cập đến việc tách biệt ngữ nghĩa cấp cao hơn của hệ thống khỏi các vấn đề triển khai cấp thấp hơn. Nói cách khác, một hệ thống minh bạch "ẩn đi" các chi tiết triển khai với người dùng. Ưu điểm của DBMS hoàn toàn minh bạch là mức độ hỗ trợ cao mà nó cung cấp cho việc phát triển các ứng dụng phức tạp.

* Độc lập dữ liệu:
  + Độc lập dữ liệu là một dạng minh bạch cơ bản trong DBMS. Nó là một kiểu duy nhất, quan trọng trong bối cảnh của một DBMS tập trung. Nó đề cập đến khả năng miễn nhiễm của các ứng dụng người dùng đối với những thay đổi trong định nghĩa, tổ chức dữ liệu và ngược lại.
    - Như đã biết, định nghĩa dữ liệu xảy ra ở hai cấp độ: cấu trúc logic của dữ liệu được chỉ định và cấu trúc vật lý. Cấu trúc logic của dữ liệu được chỉ định được gọi là định nghĩa lược đồ, trong khi cấu trúc vật lý được gọi là mô tả dữ liệu vật lý. Do đó, chúng ta có thể nói về hai loại độc lập dữ liệu: độc lập dữ liệu logic và độc lập dữ liệu vật lý. Tính độc lập dữ liệu logic đề cập đến khả năng miễn nhiễm của các ứng dụng người dùng đối với những thay đổi trong cấu trúc logic (tức là lược đồ) của cơ sở dữ liệu. Mặt khác, tính độc lập dữ liệu vật lý giải quyết việc che giấu các chi tiết của cấu trúc lưu trữ khỏi các ứng dụng của người dùng. Khi một ứng dụng người dùng được viết, nó không nên quan tâm đến các chi tiết của tổ chức dữ liệu vật lý. Do đó, ứng dụng người dùng không cần phải sửa đổi khi các thay đổi về tổ chức dữ liệu xảy ra do cân nhắc về hiệu suất.
* Tính minh bạch của mạng (Network):
  + Trong hệ thống cơ sở dữ liệu tập trung, tài nguyên sẵn có duy nhất cần được bảo vệ khỏi người dùng là dữ liệu (hệ thống lưu trữ). Tuy nhiên, trong môi trường cơ sở dữ liệu phân tán, có một tài nguyên thứ hai cần được quản lý theo cách tương tự: mạng. Tốt hơn là người dùng nên được bảo vệ khỏi các chi tiết hoạt động của mạng; thậm chí có thể che giấu sự tồn tại của mạng. Sau đó, sẽ không có sự khác biệt giữa các ứng dụng cơ sở dữ liệu sẽ chạy trên cơ sở dữ liệu tập trung và những ứng dụng sẽ chạy trên cơ sở dữ liệu phân tán. Loại minh bạch này được gọi là minh bạch mạng hoặc minh bạch phân phối.
  + Người ta có thể xem xét tính minh bạch của mạng từ quan điểm của các dịch vụ được cung cấp hoặc dữ liệu. Từ góc độ DBMS, tính minh bạch của phân phối yêu cầu người dùng không phải chỉ định dữ liệu được đặt ở đâu.
* Tính minh bạch của nhân bản:
  + Giả sử rằng dữ liệu được nhân bản, vấn đề minh bạch là liệu người dùng có nhận thức được sự tồn tại của các bản sao hay không hoặc liệu hệ thống có nên xử lý việc quản lý các bản sao hay không và người dùng nên hành động như thể có một bản sao dữ liệu duy nhất. Từ quan điểm của người dùng, câu trả lời là hiển nhiên. Tốt hơn là không nên tham gia vào việc xử lý các bản sao và phải nêu rõ thực tế là một hành động nhất định có thể và/hoặc nên được thực hiện trên nhiều bản sao. Tuy nhiên, từ quan điểm hệ thống, câu trả lời không đơn giản như vậy. Khi trách nhiệm chỉ định rằng một hành động cần được thực hiện trên nhiều bản sao được giao cho người dùng, điều đó làm cho việc quản lý giao dịch trở nên đơn giản hơn đối với các DBMS phân tán. Mặt khác, làm như vậy chắc chắn sẽ làm mất đi một số tính linh hoạt. Nó không phải là hệ thống quyết định có hay không có bản sao và có bao nhiêu bản sao, mà là ứng dụng của người dùng. Bất kỳ thay đổi nào trong các quyết định này do các cân nhắc khác nhau chắc chắn ảnh hưởng đến ứng dụng của người dùng và do đó, làm giảm đáng kể tính độc lập của dữ liệu. Với những cân nhắc này, điều mong muốn là tính minh bạch của bản sao được cung cấp như một tính năng tiêu chuẩn của DBMS. Tính minh bạch của bản sao chỉ đề cập đến sự tồn tại của các bản sao chứ không phải vị trí thực tế của chúng.
* Tính minh bạch của phân mảnh:
  + Sự phân mảnh có thể làm giảm tác động tiêu cực của việc sao chép. Mỗi bản sao không phải là quan hệ đầy đủ mà chỉ là một tập con của nó; do đó cần ít dung lượng hơn và cần quản lý ít mục dữ liệu hơn.
  + Có hai loại lựa chọn thay thế phân mảnh chung. Loại thứ nhất được gọi là phân mảnh ngang, một quan hệ được phân chia thành một tập hợp các quan hệ con, mỗi quan hệ có một tập con gồm các bộ giá trị (hàng) của quan hệ ban đầu. Loại thứ hai là phân mảnh dọc trong đó mỗi quan hệ con được xác định trên một tập hợp con của các thuộc tính (cột) của quan hệ ban đầu.
  + Khi các đối tượng của cơ sở dữ liệu bị phân mảnh, chúng ta phải giải quyết vấn đề xử lý các truy vấn của người dùng được chỉ định trên toàn bộ quan hệ nhưng phải được thực thi trên các quan hệ con. Nói cách khác, vấn đề nằm ở việc tìm chiến lược xử lý truy vấn dựa trên các phân mảnh chứ không phải các quan hệ, mặc dù các truy vấn được chỉ định trên phân mảnh sau. Thông thường, điều này yêu cầu một bản dịch từ những gì được gọi là truy vấn toàn cục sang một số truy vấn phân phân mảnh. Vì vấn đề cơ bản của việc xử lý tính trong suốt của phân mảnh là một trong những xử lý truy vấn.

### Độ tin cậy thông qua các giao dịch phân tán

* Một giao dịch là một đơn vị cơ bản về tính toán nhất quán và đáng tin cậy, bao gồm một chuỗi các hoạt động cơ sở dữ liệu được thực thi dưới dạng hành động nguyên tử. Nó biến đổi trạng thái cơ sở dữ liệu nhất quán này sang trạng thái cơ sở dữ liệu nhất quán khác ngay cả khi một số giao dịch như vậy được thực thi đồng thời (đôi khi được gọi là độ trong suốt đồng thời) và ngay cả khi xảy ra lỗi (còn gọi là nguyên tử thất bại). Do đó, một DBMS cung cấp hỗ trợ giao dịch đầy đủ đảm bảo rằng việc thực hiện đồng thời các giao dịch người dùng sẽ không vi phạm tính nhất quán của cơ sở dữ liệu khi đối mặt với các lỗi hệ thống miễn là mỗi giao dịch là chính xác, tức là, tuân theo các quy tắc toàn vẹn được chỉ định trên cơ sở dữ liệu.
* Giao dịch phân tán thực thi tại một số trạm mà chúng truy cập cơ sở dữ liệu cục bộ.

### Cải thiện hiệu suất

* Để cải thiện hiệu suất, các DBMS phân tán thường được thực hiện dựa trên hai điểm. Đầu tiên, một đoạn DBMS phân tán Các cơ sở dữ liệu khái niệm, cho phép dữ liệu được lưu trữ gần với các điểm sử dụng của nó (còn gọi là bản địa hóa dữ liệu). Điều này có hai lợi thế tiềm năng:
  + Vì mỗi trạm chỉ xử lý một phần cơ sở dữ liệu, tranh chấp các dịch vụ CPU và I/O không nghiêm trọng như đối với cơ sở dữ liệu tập trung.
  + Bản địa hóa giảm độ trễ truy cập từ xa, thường tham gia vào các mạng diện rộng (ví dụ: độ trễ lan truyền thông báo khứ hồi tối thiểu trong các hệ thống dựa trên vệ tinh là khoảng 1 giây).
* Hầu hết các DBMS phân phối được cấu trúc để đạt được lợi ích tối đa từ việc bản địa hóa dữ liệu. Lợi ích đầy đủ của việc giảm tranh chấp và giảm chi phí giao tiếp chỉ có thể được lấy bằng cách phân mảnh và phân phối cơ sở dữ liệu thích hợp.
* Điểm này liên quan đến chi phí điện toán phân tán nếu dữ liệu phải nằm ở các trạm từ xa và người ta phải truy cập vào giao tiếp từ xa.

### Dễ dàng mở rộng hệ thống

* Trong một môi trường phân tán, khá dễ dàng để chứa kích thước cơ sở dữ liệu ngày một tăng. Đại tu hệ thống chính hiếm khi cần thiết. Việc mở rộng thường có thể được xử lý bằng cách thêm công suất xử lý và lưu trữ vào mạng.
* Một khía cạnh của việc mở rộng hệ thống dễ dàng hơn là kinh tế. Nó thường tốn ít chi phí hơn để kết hợp một hệ thống máy tính "nhỏ hơn" với sức mạnh tương đương của một máy lớn duy nhất. Vấn đề là đối với nhiều ứng dụng, điều đó là tiết kiệm hơn để kết hợp một hệ thống máy tính phân tán (cho dù bao gồm các máy tính lớn hoặc máy trạm) với năng lượng đủ hơn là thiết lập một hệ thống duy nhất, tập trung để chạy các tác vụ này.

## Các vấn đề cần quan tâm trong phân tán

### Thiết kế cơ sở dữ liệu phân tán

Có hai lựa chọn thay thế cơ bản để đặt vị trí của dữ liệu: phân mảnh (không nhân bản) và nhân bản. Trong sơ đồ phân mảnh, cơ sở dữ liệu được chia thành một số phân mảnh được phân chia đặt tại một số trạm khác. Thiết kế nhân bản có thể được nhân bản đầy đủ (còn được gọi là trùng lặp hoàn toàn) trong đó toàn bộ cơ sở dữ liệu được lưu trữ tại mỗi trạm hoặc được nhân bản một phần (hoặc trùng lặp một phần) trong đó mỗi phân mảnh của cơ sở dữ liệu được lưu trữ ở nhiều trạm, nhưng không phải là tất cả các trạm. Hai vấn đề thiết kế cơ bản là phân mảnh, việc tách cơ sở dữ liệu thành nhiều mảnh được gọi là các mảnh, phân phối, phân phối tối ưu các mảnh.

### Quản lý thư mục phân tán

Một thư mục chứa thông tin (như mô tả và vị trí) về các mục dữ liệu trong cơ sở dữ liệu. Các vấn đề liên quan đến quản lý thư mục giống như bản chất của vấn đề vị trí cơ sở dữ liệu. Một thư mục chung cho toàn bộ DDBS hoặc cục bộ đến từng trạm, nó có thể được tập trung tại một trạm hoặc phân phối trên một số trạm, có thể có một bản sao hoặc nhiều bản sao.

### Xử lý truy vấn phân tán

Xử lý truy vấn liên quan đến việc thiết kế các thuật toán phân tích các truy vấn và chuyển đổi chúng thành một chuỗi các thao tác xử lý dữ liệu. Vấn đề là làm thế nào để quyết định một chiến lược thực hiện từng truy vấn qua mạng một cách hiệu quả nhất về chi phí, tuy nhiên chi phí được xác định. Các yếu tố được xem là sự phân bố của dữ liệu, chi phí liên lạc và thiếu thông tin có sẵn tại cục bộ. Mục tiêu là tối ưu hóa nơi mà tính song song vốn có được sử dụng để cải thiện hiệu suất thực hiện giao dịch, tuân theo các ràng buộc nêu trên.

### Kiểm soát đồng thời phân tán

Kiểm soát đồng thời liên quan đến việc đồng bộ hóa các quyền truy cập vào cơ sở dữ liệu phân tán, sao cho tính toàn vẹn của cơ sở dữ liệu được duy trì. Đây là một trong những vấn đề được nghiên cứu rộng rãi nhất trong lĩnh vực DDBS. Vấn đề kiểm soát đồng thời trong bối cảnh phân tán hơi khác so với trong khuôn khổ tập trung. Người ta không chỉ phải lo lắng về tính toàn vẹn của một cơ sở dữ liệu đơn lẻ, mà còn về tính nhất quán của nhiều bản sao của cơ sở dữ liệu. Điều kiện yêu cầu tất cả các giá trị của nhiều bản sao của mọi dữ liệu hội tụ về cùng một giá trị được gọi là tính nhất quán lẫn nhau.

### Quản lý bế tắc phân tán

Vấn đề bế tắc trong DDBS có bản chất tương tự như vấn đề gặp phải trong hệ điều hành. Sự cạnh tranh giữa những người dùng để truy cập vào một tập hợp tài nguyên (trong trường hợp này là dữ liệu) có thể dẫn đến bế tắc nếu cơ chế đồng bộ hóa dựa trên việc khóa. Các lựa chọn thay thế nổi tiếng về phòng ngừa, tránh và phát hiện/phục hồi cũng áp dụng cho DDBS.

### Độ tin cậy của DBMS phân tán

Một trong những lợi thế tiềm năng của hệ thống phân tán là cải thiện độ tin cậy và tính khả dụng. Tuy nhiên, đây không phải là một tính năng tự động đi kèm. Điều quan trọng là phải cung cấp các cơ chế để đảm bảo tính nhất quán của cơ sở dữ liệu cũng như phát hiện các lỗi và khôi phục chúng. Hàm ý đối với DDBS là khi xảy ra lỗi và các trạm khác nhau không thể hoạt động hoặc không thể truy cập được, cơ sở dữ liệu tại các điểm hoạt động vẫn nhất quán và cập nhật. Hơn nữa, khi hệ thống máy tính hoặc mạng phục hồi sau sự cố, các DDBS sẽ có thể khôi phục và cập nhật cơ sở dữ liệu tại các trạm cập nhật thất bại. Điều này có thể đặc biệt khó khăn trong trường hợp phân mảnh mạng, trong đó các trạm được chia thành hai hoặc nhiều nhóm mà không có thông tin liên lạc giữa chúng.

### Nhân bản

Nếu cơ sở dữ liệu phân tán được phân tán (một phần hoặc toàn bộ), thì cần phải triển khai các giao thức đảm bảo tính nhất quán của các bản sao, tức là các bản sao của cùng một mục dữ liệu có cùng giá trị. Các giao thức này có thể buộc các bản cập nhật phải được áp dụng cho tất cả các nhân bản trước khi giao dịch hoàn tất hoặc chúng có thể để giao dịch cập nhật một bản sao (được gọi là bản chính) mà từ đó các bản cập nhật được truyền cho các bản sao khác sau khi giao dịch hoàn tất.

# Xây dựng cơ sở dữ liệu phân tán cho ứng dụng quản lý cho thuê truyện

## Mô tả bài toán

Với mong muốn có thể hiểu và giải quyết được những yêu cầu cấp thiết trên thực tế với các cửa hàng cho thuê truyện hiện nay để giảm bớt được sự không cần thiết về nhân lực, cũng như sự bất tiện trong việc quản lý sổ sách và tìm kiếm. Việc xây dựng một cơ sở dữ liệu phân tán sẽ là một giải pháp hợp lý về chi phí và thời gian cho vấn đề này.

### Mô tả yêu cầu dữ liệu

Căn cứ vào khảo sát thực tế ta có thể thấy được các thành phần cơ bản của hệ thống quản lý cho thuê truyện như sau:

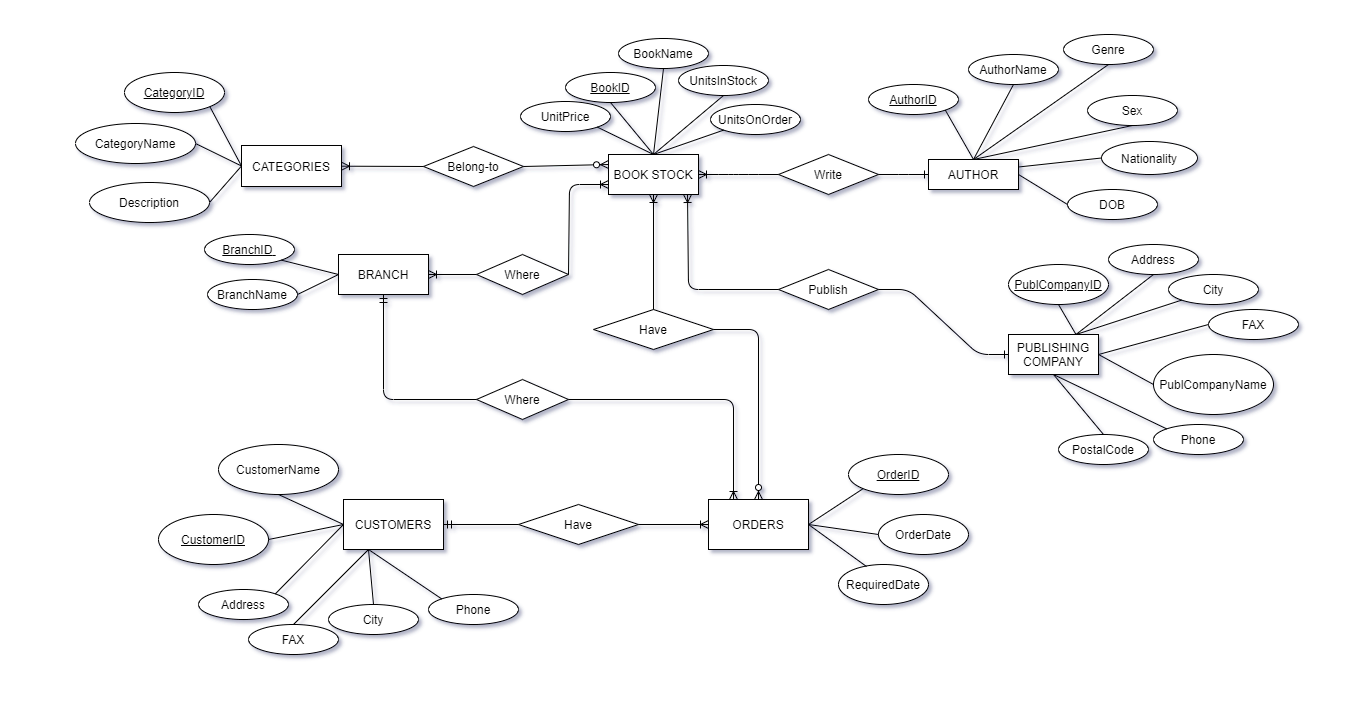
* Tập thực thể **BOOK STOCK** gồm các thuộc tính:
* Mã sách (BookID: varchar(2))
* Tên sách (BookName: nvarchar(30))
* Giá (UnitPrice: money)
* Số lượng trong kho (UnitsInStock: smallint)
* Số lượng đã cho thuê (UnitsOnOrder: smallint)
* Tập thực thể **PUBLISHING COMPANY**
* Mã nhà xuất bản (PublCompanyID: varchar(2))
* Tên nhà xuất bản (PublCompanyName: nvarchar(30))
* Địa chỉ (Address: nvarchar(60))
* Thành phố (City: nvarchar(20))
* Số fax (FAX: nvarchar(20))
* Mã bưu chính (PostalCode: nvarchar(10))
* Số điện thoại (Phone: nvarchar(15))
* Tập thực thể **AUTHOR**
* Mã tác giả (AuthorID: varchar(2))
* Tên tác giả (AuthorName: nvarchar(30))
* Thể loại sáng tác (Genre: nvarchar(60))
* Giới tính (Sex: nvarchar(20))
* Quốc tịch (Nationality: nvarchar(20))
* Ngày sinh (DOB: date)
* Tập thực thể **CUSTOMERS**
* Mã khách hàng (CustomerID: varchar(2))
* Tên khách hàng (CustomerName: nvarchar(30))
* Địa chỉ (Address: nvarchar(60))
* Thành phố (City: nvarchar(20))
* Số fax (FAX: nvarchar(20))
* Số điện thoại (Phone: nvarchar(15))
* Tập thực thể **ORDERS**
* Mã thuê sách (OrderID: varchar(2))
* Ngày thuê (OrderDate: datetime)
* Hạn trả (RequiredDate: datetime)
* Tập thực thể **CATEGORIES**
* Mã thể loại (CategoryID: varchar(2))
* Tên thể loại (CategoryName: nvarchar(15))
* Mô tả (Description: ntext)
* Tập thực thể **BRANCH**
* Mã chi nhánh (BranchID: varchar(2))
* Tên chi nhánh (BranhName: nvarchar(20))

Mối quan hệ giữa các tập thực thể:

* Giữa tập thực thể **BOOK STOCK** và **AUTHOR** có các liên kết như là một quyển sách có một tác giả, một tác giả có thể có một hoặc nhiều quyển sách.
* Giữa tập thực thể **BOOK STOCK** và **PUBLISHING COMPANY** có liên kết là một quyển sách có một nhà xuất bản, một nhà xuất bản có thể xuất bản một hoặc nhiều quyển sách.
* Giữa tập thực thể **BOOK STOCK** và **CATEGORIES** có liên kết là một quyển sách có một hoặc nhiều thể loại, một thể loại có thể không có hoặc có nhiều loại sách sở hữu.
* Giữa tập thực thể **BOOK STOCK** và **ORDERS** có liên kết sau: một quyển sách có thể có nhiều đơn thuê hoặc không có đơn thuê nào, một đơn thuê có thể thuê một hoặc nhiều sách
* Giữa tập thực thể **CUSTOMERS** và tập thực thể **ORDERS** có những liên kết sau: một khách hàng có thể có nhiều đơn thuê, một đơn thuê chỉ thuộc về một khách hàng.
* Giữa tập thực thể **BRANCH** và **BOOKSTOCK** có liên kết: một quyển sách có thể có ở một hoặc nhiều chi nhánh, một chi nhánh cũng sở hữu một hoặc nhiều quyển sách.
* Giữa tập thực thể **BRANCH** và **ORDERS** có liên kết là một chi nhánh có thể có một hoặc nhiều đơn thuê, một đơn thuê chỉ thuộc một chi nhánh.

### Mô hình thực thể quan hệ

Từ mối quan hệ giữa các tập thực thể, chúng ta có mô hình thực thể ER (Entity Relationship Diagram) sau:

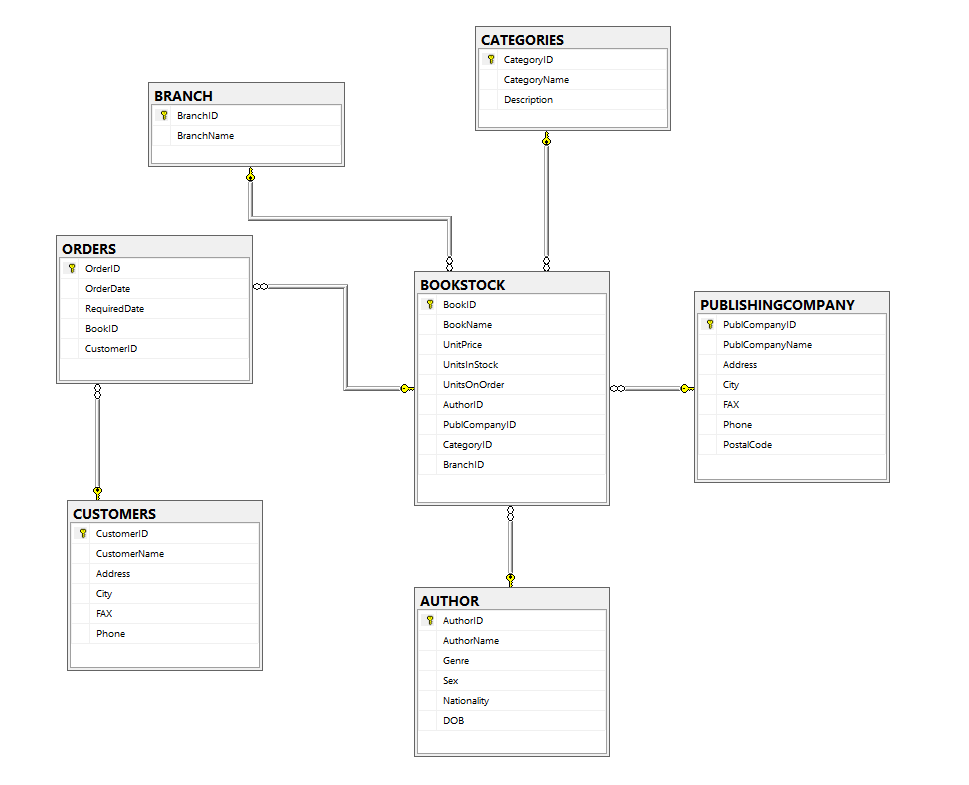


### Lược đồ quan hệ

Chuyển mô hình ER về lược đồ quan hệ ta có các bảng như sau:

* **BOOKSTOCK**(#BookID,BookName,UnitPrice,UnitsInStock, UnitsOnOrder,@AuthorID,@PublCompanyID,@CategoriesID, @BranchID)
* **AUTHOR**(#AuthorID, AuthorName, Genre, Sex, Nationality, DOB)
* **PUBLISHINGCOMPANY**(#PublCompanyID,PublCompanyName, PostalCode, Address, City, FAX, Phone)
* **ORDERS**(#OrderID,OrderDate,RequiredDate,@BookID,@CustomerID,@BranchID)
* **CUSTOMERS**(#CustomerID,CustomerName,Address,City,FAX, Phone)
* **CATEGORIES**(#CategoryID, CategoryName, Description)
* **BRANCH**(#BranchID, BranchName)

Ta có lược đồ cơ sở dữ liệu (database diagrams) sau:



## Thiết kế phân tán

### Kịch bản phân mảnh dọc

Giả sử số trạm (Site), số ứng dụng (AP) bằng số thành viên trong nhóm (3 người). Phân mảnh dọc bảng AUTHOR có 6 thuộc tính được đặt tên C, C1, C2, C3, C4, C5, với C là khóa chính.

* Tạo ma trận tần suất theo Msv của thành viên nhóm:
  + Mã sinh viên của các thành viên:
    - A30093 - Nguyễn Tuấn Sơn
    - A30337 - Bùi Đức Toàn
    - A30819 - Nguyễn Đăng Tú
* Ma trận tần suất:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **C1** | **C2** | **C3** | **C4** | **C5** | **Tần suất (Aff)** | | |
| **S1** | **S2** | **S3** |
| **AP1** | 3 | 0 | 0 | 9 | 3 | 0 | 9 | 3 |
| **AP2** | 3 | 0 | 3 | 3 | 7 | 3 | 3 | 7 |
| **AP3** | 3 | 0 | 8 | 1 | 9 | 8 | 1 | 9 |

* Đơn giản ma trận tần suất (số lẻ thành 1, số chẵn thành 0):

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **C1** | **C2** | **C3** | **C4** | **C5** | **Tần suất (Aff)** | | |
| **S1** | **S2** | **S3** |
| **AP1** | 1 | 0 | 0 | 1 | 1 | 0 | 9 | 3 |
| **AP2** | 1 | 0 | 1 | 1 | 1 | 3 | 3 | 7 |
| **AP3** | 1 | 0 | 0 | 1 | 1 | 8 | 1 | 9 |

* Đơn giản ma trận tần suất (Gộp tần suất giữa các Site):

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **C1** | **C2** | **C3** | **C4** | **C5** | **Aff** |
| **AP1** | 1 | 0 | 0 | 1 | 1 | 12 |
| **AP2** | 1 | 0 | 1 | 1 | 1 | 13 |
| **AP3** | 1 | 0 | 0 | 1 | 1 | 18 |

* Ma trận hấp dẫn:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **C1** | **C2** | **C3** | **C4** | **C5** |
| **C1** | 99 | 0 | 13 | 43 | 43 |
| **C2** | 0 | 0 | 0 | 0 | 0 |
| **C3** | 13 | 0 | 39 | 13 | 13 |
| **C4** | 43 | 0 | 13 | 99 | 43 |
| **C5** | 43 | 0 | 13 | 43 | 99 |

* Tính Cont và Bond:
* Xếp C3:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | C0 | C3 | Product | | 0 | 13 | 0 | | 0 | 0 | 0 | | 0 | 39 | 0 | | 0 | 13 | 0 | | 0 | 13 | 0 | | Bond | | 0 | | |  |  |  | | --- | --- | --- | | C3 | C1 | Product | | 13 | 99 | 1287 | | 0 | 0 | 0 | | 39 | 13 | 507 | | 13 | 43 | 559 | | 13 | 43 | 559 | | Bond | | 2912 | | |  |  |  | | --- | --- | --- | | C1 | C0 | Product | | 99 | 0 | 0 | | 0 | 0 | 0 | | 13 | 0 | 0 | | 43 | 0 | 0 | | 43 | 0 | 0 | | Bond | | 0 | |

Cont (C0, C1, C3) = 2\*0 + 2\*2912 – 2\*0 = 5824

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | C1 | C3 | Product | | 99 | 13 | 1287 | | 0 | 0 | 0 | | 13 | 39 | 507 | | 43 | 13 | 559 | | 43 | 13 | 559 | | Bond | | 2912 | | |  |  |  | | --- | --- | --- | | C3 | C2 | Product | | 13 | 0 | 0 | | 0 | 0 | 0 | | 39 | 0 | 0 | | 13 | 0 | 0 | | 13 | 0 | 0 | | Bond | | 0 | | |  |  |  | | --- | --- | --- | | C1 | C2 | Product | | 99 | 0 | 0 | | 0 | 0 | 0 | | 13 | 0 | 0 | | 43 | 0 | 0 | | 43 | 0 | 0 | | Bond | | 0 | |

Cont(C1, C3, C2) = 2\*2912 + 2\*0 - 2\*0 = 5824

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | C2 | C3 | Product | | 0 | 13 | 0 | | 0 | 0 | 0 | | 0 | 39 | 0 | | 0 | 13 | 0 | | 0 | 13 | 0 | | Bond | | 0 | | |  |  |  | | --- | --- | --- | | C3 | C0 | Product | | 13 | 0 | 0 | | 0 | 0 | 0 | | 39 | 0 | 0 | | 13 | 0 | 0 | | 13 | 0 | 0 | | Bond | | 0 | | |  |  |  | | --- | --- | --- | | C2 | C0 | Product | | 0 | 0 | 0 | | 0 | 0 | 0 | | 0 | 0 | 0 | | 0 | 0 | 0 | | 0 | 0 | 0 | | Bond | | 0 | |

Cont(C2, C3, C0) = 2\*0 + 2\*0 - 2\*0 = 0

Chọn Cont lớn nhất: Cont(C0, C3, C1) = 5824

Ma trận sau khi sắp xếp lại là:

|  |  |  |  |
| --- | --- | --- | --- |
|  | C3 | C1 | C2 |
| C3 | 39 | 13 | 0 |
| C1 | 13 | 99 | 0 |
| C2 | 0 | 0 | 0 |

* Xếp C4:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | C0 | C4 | Product | | 0 | 43 | 0 | | 0 | 0 | 0 | | 0 | 13 | 0 | | 0 | 99 | 0 | | 0 | 43 | 0 | | Bond | | 0 | | |  |  |  | | --- | --- | --- | | C4 | C3 | Product | | 43 | 13 | 559 | | 0 | 0 | 0 | | 13 | 39 | 507 | | 99 | 13 | 1287 | | 43 | 13 | 559 | | Bond | | 2912 | | |  |  |  | | --- | --- | --- | | C0 | C3 | Product | | 0 | 13 | 0 | | 0 | 0 | 0 | | 0 | 39 | 0 | | 0 | 13 | 0 | | 0 | 13 | 0 | | Bond | | 0 | |

Cont(C0, C4, C3) = 2\*0 + 2\*2912 - 2\*0 = 5824

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | C3 | C4 | Product | | 13 | 43 | 559 | | 0 | 0 | 0 | | 39 | 13 | 507 | | 13 | 99 | 1287 | | 13 | 43 | 559 | | Bond | | 2912 | | |  |  |  | | --- | --- | --- | | C4 | C1 | Product | | 43 | 99 | 4257 | | 0 | 0 | 0 | | 13 | 13 | 169 | | 99 | 43 | 4257 | | 43 | 43 | 1849 | | Bond | | 10532 | | |  |  |  | | --- | --- | --- | | C3 | C1 | Product | | 13 | 99 | 1287 | | 0 | 0 | 0 | | 39 | 13 | 507 | | 13 | 43 | 559 | | 13 | 43 | 559 | | Bond | | 2912 | |

Cont(C3, C4, C1) = 2\*2912 + 2\*10532 - 2\*2912 = 21064

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | C1 | C4 | Product | | 99 | 43 | 4257 | | 0 | 0 | 0 | | 13 | 13 | 169 | | 43 | 99 | 4257 | | 43 | 43 | 1849 | | Bond | | 10532 | | |  |  |  | | --- | --- | --- | | C4 | C2 | Product | | 43 | 0 | 0 | | 0 | 0 | 0 | | 13 | 0 | 0 | | 99 | 0 | 0 | | 43 | 0 | 0 | | Bond | | 0 | | |  |  |  | | --- | --- | --- | | C1 | C2 | Product | | 99 | 0 | 0 | | 0 | 0 | 0 | | 13 | 0 | 0 | | 43 | 0 | 0 | | 43 | 0 | 0 | | Bond | | 0 | |

Cont(C1, C4, C2) = 2\*10532 + 2\*0 - 2\*0 = 21064

Chọn Cont lớn nhất: Cont(C3, C4, C1) = 21064

Ma trận sau khi sắp xếp lại là:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | C3 | C4 | C1 | C2 |
| C3 | 39 | 13 | 13 | 0 |
| C4 | 13 | 99 | 43 | 0 |
| C1 | 13 | 43 | 99 | 0 |
| C2 | 0 | 0 | 0 | 0 |

* Xếp C5:

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | C0 | C5 | Product | | 0 | 43 | 0 | | 0 | 0 | 0 | | 0 | 13 | 0 | | 0 | 43 | 0 | | 0 | 99 | 0 | | Bond | | 0 | | |  |  |  | | --- | --- | --- | | C5 | C3 | Product | | 43 | 13 | 559 | | 0 | 0 | 0 | | 13 | 39 | 507 | | 43 | 13 | 559 | | 99 | 13 | 1287 | | Bond | | 2912 | | |  |  |  | | --- | --- | --- | | C0 | C3 | Product | | 0 | 13 | 0 | | 0 | 0 | 0 | | 0 | 39 | 0 | | 0 | 13 | 0 | | 0 | 13 | 0 | | Bond | | 0 | |

Cont(C0, C5, C3) = 2\*0 + 2\*2912 - 2\*0 = 5824

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | C3 | C5 | Product | | 13 | 43 | 559 | | 0 | 0 | 0 | | 39 | 13 | 507 | | 13 | 43 | 559 | | 13 | 99 | 1287 | | Bond | | 2912 | | |  |  |  | | --- | --- | --- | | C5 | C4 | Product | | 43 | 43 | 1849 | | 0 | 0 | 0 | | 13 | 13 | 169 | | 43 | 99 | 4257 | | 99 | 43 | 4257 | | Bond | | 10532 | | |  |  |  | | --- | --- | --- | | C3 | C4 | Product | | 13 | 43 | 559 | | 0 | 0 | 0 | | 39 | 13 | 507 | | 13 | 99 | 1287 | | 13 | 43 | 559 | | Bond | | 2912 | |

Cont(C3, C5, C4) = 2\*2912 + 2\*10532 - 2\*2912 = 21064

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | C4 | C5 | Product | | 43 | 43 | 1849 | | 0 | 0 | 0 | | 13 | 13 | 169 | | 99 | 43 | 4257 | | 43 | 99 | 4257 | | Bond | | 10532 | | |  |  |  | | --- | --- | --- | | C5 | C1 | Product | | 43 | 99 | 4257 | | 0 | 0 | 0 | | 13 | 13 | 169 | | 43 | 43 | 1849 | | 99 | 43 | 4257 | | Bond | | 10532 | | |  |  |  | | --- | --- | --- | | C4 | C1 | Product | | 43 | 99 | 4257 | | 0 | 0 | 0 | | 13 | 13 | 169 | | 99 | 43 | 4257 | | 43 | 43 | 1849 | | Bond | | 10532 | |

Cont(C4, C5, C1) = 2\*10532 + 2\*10532 - 2\*10532 = 21064

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | C1 | C5 | Product | | 99 | 43 | 4257 | | 0 | 0 | 0 | | 13 | 13 | 169 | | 43 | 43 | 1849 | | 43 | 99 | 4257 | | Bond | | 10532 | | |  |  |  | | --- | --- | --- | | C5 | C2 | Product | | 43 | 0 | 0 | | 0 | 0 | 0 | | 13 | 0 | 0 | | 43 | 0 | 0 | | 99 | 0 | 0 | | Bond | | 0 | | |  |  |  | | --- | --- | --- | | C1 | C2 | Product | | 99 | 0 | 0 | | 0 | 0 | 0 | | 13 | 0 | 0 | | 43 | 0 | 0 | | 43 | 0 | 0 | | Bond | | 0 | |

Cont(C1, C5, C2) = 2\*10532 + 2\*0 - 2\*0 = 21064

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  |  | | --- | --- | --- | | C2 | C5 | Product | | 0 | 43 | 0 | | 0 | 0 | 0 | | 0 | 13 | 0 | | 0 | 43 | 0 | | 0 | 99 | 0 | | Bond | | 0 | | |  |  |  | | --- | --- | --- | | C5 | C0 | Product | | 43 | 0 | 0 | | 0 | 0 | 0 | | 13 | 0 | 0 | | 43 | 0 | 0 | | 99 | 0 | 0 | | Bond | | 0 | | |  |  |  | | --- | --- | --- | | C2 | C0 | Product | | 0 | 0 | 0 | | 0 | 0 | 0 | | 0 | 0 | 0 | | 0 | 0 | 0 | | 0 | 0 | 0 | | Bond | | 0 | |

Cont(C2, C5, C0) = 2\*0 + 2\*0 - 2\*0 = 0

Chọn Cont lớn nhất: Cont(C3, C5, C4) = 21064

Ma trận sau khi sắp xếp lại là:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | C3 | C5 | C4 | C1 | C2 |
| C3 | 39 | 13 | 13 | 13 | 0 |
| C5 | 13 | 99 | 43 | 43 | 0 |
| C1 | 13 | 43 | 99 | 43 | 0 |
| C4 | 13 | 43 | 43 | 99 | 0 |
| C2 | 0 | 0 | 0 | 0 | 0 |

* Tính Z:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | C3 | C5 | C4 | C1 | C2 |
| C3 | 39 | 13 | 13 | 13 | 0 |
| C5 | 13 | 99 | 43 | 43 | 0 |
| C1 | 13 | 43 | 99 | 43 | 0 |
| C4 | 13 | 43 | 43 | 99 | 0 |
| C2 | 0 | 0 | 0 | 0 | 0 |

TCW = AFF(1) + AFF(2) + AFF(3) = 12 + 13 + 18 = 43

BCW = 0

BOCW = 0

Z2 = TCW\*BCW - BOCW^2 = 43\*0 - 0^2 = 0

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | C3 | C5 | C4 | C1 | C2 |
| C3 | 39 | 13 | 13 | 13 | 0 |
| C5 | 13 | 99 | 43 | 43 | 0 |
| C1 | 13 | 43 | 99 | 43 | 0 |
| C4 | 13 | 43 | 43 | 99 | 0 |
| C2 | 0 | 0 | 0 | 0 | 0 |

TCW = 0

BCW = 0

BOCW = AFF(1) + AFF(2) + AFF(3) = 12 + 13 + 18 = 43

Z1 = TCW\*BCW - BOCW^2 = 0\*0 - 43^2 = -1849

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | C3 | C5 | C4 | C1 | C2 |
| C3 | 39 | 13 | 13 | 13 | 0 |
| C5 | 13 | 99 | 43 | 43 | 0 |
| C1 | 13 | 43 | 99 | 43 | 0 |
| C4 | 13 | 43 | 43 | 99 | 0 |
| C2 | 0 | 0 | 0 | 0 | 0 |

TCW = 0

BCW = 0

BOCW = AFF(1) + AFF(2) + AFF(3) = 12 + 13 + 18 = 43

Z4 = TCW\*BCW - BOCW^2 = 0\*0 - 43^2 = -1849

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | C3 | C5 | C4 | C1 | C2 |
| C3 | 39 | 13 | 13 | 13 | 0 |
| C5 | 13 | 99 | 43 | 43 | 0 |
| C1 | 13 | 43 | 99 | 43 | 0 |
| C4 | 13 | 43 | 43 | 99 | 0 |
| C2 | 0 | 0 | 0 | 0 | 0 |

TCW = 0

BCW = AFF(1) + AFF(3) = 12 + 18 = 30

BOCW = AFF(2) = 13

Z5 = TCW\*BCW - BOCW^2 = 0\*30 - 13^2 = -169

* Chuyển InnerBlock:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | C3 | C5 | C4 | C1 | C2 |
| C3 | 39 | 13 | 13 | 13 | 0 |
| C5 | 13 | 99 | 43 | 43 | 0 |
| C1 | 13 | 43 | 99 | 43 | 0 |
| C4 | 13 | 43 | 43 | 99 | 0 |
| C2 | 0 | 0 | 0 | 0 | 0 |

TCW = AFF(1) + AFF(3) = 12 + 18 = 30

BCW = 0

BOCW = AFF(2) = 13

ZInnerBlock = TCW\*BCW - BOCW^2 = 30\*0 - 13^2 = -169

* Phân mảnh:
* Ta thấy Z2 = 0 là lớn nhất
* Ta chọn Z = 0 là Z2
* Sau khi thêm khóa C cho vùng trên ta được kết quả hai mảnh dọc sau: VF1['C', 'C2'], VF2['C', 'C3', 'C5', 'C4', 'C1']
* Trong bảng AUTHOR có 2 phân mảnh dọc sau:
  + Mảnh AUTHOR\_Gerne. Ta có VF1['C', 'C2'] tương đương là AuthorID, Genre.
  + Mảnh AUTHOR\_No\_Gerne. Ta có VF2['C', 'C3', 'C5', 'C4', 'C1'] tương đương là AuthorID, AuthorName, Sex, Nationality, DOB.
* Trong bảng CUSTOMER có 2 phân mảnh dọc sau:
  + Mảnh CUSTOMER\_No\_Address. Ta có VF1['C', 'C2'] tương đương là CustomerID, Address.
  + Mảnh CUSTOMER\_Address. Ta có VF2['C', 'C3', 'C5', 'C4', 'C1'] tương đương là CustomerID, CustomerName, City, FAX, Phone.

### Kịch bản phân mảnh ngang

Phân mảnh ngang trên 2 bảng:

* **BOOKSTOCK** (BookID, BookName, UnitPrice, UnitsInStock, UnitsOnOrder, AuthorID, PublCompanyID, CategoryID, BranchID) với BookID là khóa chính.
* **ORDERS** (OrderID, OrderDate, RequiredDate, BookID, CustomerID, BranchID) với OrderID là khóa chính

Với 3 ứng dụng (AP), tạo ra 2 phân mảnh ngang trên 2 bảng BOOKSTOCK và ORDERS:

* **Phân mảnh ngang trên bảng BOOKSTOCK:**
  + Các vị từ đơn giản:
    - AP1 có vị từ đơn giản là p1: UnitPrice > 60000
    - AP2 có vị từ đơn giản là p2: Branch = 'B0'
    - AP3 có vị từ đơn giản là p3: Branch = 'B1'
  + Các hội vị từ:
    - m1 = {BranchID = 'B0' ^ UnitPrice > 60000 ^ BranchID = 'B1'}
    - m2 = {BranchID <> 'B0' ^ UnitPrice > 60000 ^ BranchID = 'B1'}
    - m3 = {BranchID <> 'B0' ^ UnitPrice <= 60000 ^ BranchID = 'B1'}
    - m4 = {BranchID <> 'B0' ^ UnitPrice > 60000 ^ BranchID <> 'B1'}
    - m5 = {BranchID = 'B0' ^ UnitPrice <= 60000 ^ BranchID = 'B1'}
    - m6 = {BranchID = 'B0' ^ UnitPrice <= 60000 ^ BranchID <> 'B1'}
    - m7 = {BranchID = 'B0' ^ UnitPrice > 60000 ^ BranchID <> 'B1'}
    - m8 = {BranchID <> 'B0' ^ UnitPrice <= 60000 ^ BranchID <> 'B1'}
  + Lọc các vị từ không thể xảy ra đồng thời:
    - Lọc m1, m5 vì BranchID không thể cùng là ‘B0’ và ‘B1’.
    - Lọc m4, m8 vì BranchID không thể cùng khác ‘B0’ và ‘B1’.
  + Các vị tự còn lại:
    - m2 = {BranchID <> 'B0' ^ UnitPrice > 60000 ^ BranchID = 'B1'}
    - m3 = {BranchID <> 'B0' ^ UnitPrice <= 60000 ^ BranchID = 'B1'}
    - m6 = {BranchID = 'B0' ^ UnitPrice <= 60000 ^ BranchID <> 'B1'}
    - m7 = {BranchID = 'B0' ^ UnitPrice > 60000 ^ BranchID <> 'B1'}
  + Rút gọn các vị từ:
    - m2 = {UnitPrice > 60000 ^ BranchID = 'B1'}
    - m3 = {UnitPrice <= 60000 ^ BranchID = 'B1'}
    - m6 = {UnitPrice <= 60000 ^ BranchID = 'B0'}
    - m7 = {UnitPrice > 60000 ^ BranchID = 'B0'}
  + Phân mảnh:
    - Phân mảnh 2 vị từ m2, m7 tại Site 1 (Giá > 60000 tại Hà Nội và TP.HCM).
    - Phân mảnh vị từ m6 tại Site 2 (Giá <= 6000 tại Hà Nội).
    - Phân mảnh vị từ m3 tại Site 3 (Giá <= 6000 tại TP.HCM).
* **Phân mảnh ngang trên bảng ORDERS:**
  + Các vị từ đơn giản:
    - AP1 có vị từ đơn giản là: p1: Branch = 'B0'
    - AP2 có vị từ đơn giản là: p2: Branch = 'B1'
    - AP3 có vị từ đơn giản là: p3: OrderDate < 1-1-2021
  + Các hội vị từ:
    - m1 = {BranchID = 'B0' ^ OrderDate < 1-1-2021 ^ BranchID = 'B1'}
    - m2 = {BranchID <> 'B0' ^ OrderDate < 1-1-2021 ^ BranchID = 'B1'}
    - m3 = {BranchID <> 'B0' ^ OrderDate >= 1-1-2021 ^ BranchID = 'B1'}
    - m4 = {BranchID <> 'B0' ^ OrderDate < 1-1-2021 ^ BranchID <> 'B1'}
    - m5 = {BranchID = 'B0' ^ OrderDate >= 1-1-2021 ^ BranchID = 'B1'}
    - m6 = {BranchID = 'B0' ^ OrderDate >= 1-1-2021 ^ BranchID <> 'B1'}
    - m7 = {BranchID = 'B0' ^ OrderDate < 1-1-2021 ^ BranchID <> 'B1'}
    - m8 = {BranchID <> 'B0' ^OrderDate >= 1-1-2021 ^ BranchID <> 'B1'}
  + Lọc các vị từ không thể xảy ra đồng thời:
    - Lọc m1, m5 vì BranchID không thể cùng là ‘B0’ và ‘B1’.
    - Lọc m4, m8 vì BranchID không thể cùng khác ‘B0’ và ‘B1’.
  + Các vị tự còn lại:
    - m2 = {BranchID <> 'B0' ^ OrderDate < 1-1-2021 ^ BranchID = 'B1'}
    - m3 = {BranchID <> 'B0' ^ OrderDate >= 1-1-2021 ^ BranchID = 'B1'}
    - m6 = {BranchID = 'B0' ^ OrderDate >= 1-1-2021 ^ BranchID <> 'B1'}
    - m7 = {BranchID = 'B0' ^ OrderDate < 1-1-2021 ^ BranchID <> 'B1'}
  + Rút gọn các vị từ:
    - m2 = {OrderDate < 1-1-2021 ^ BranchID = 'B1'}
    - m3 = {OrderDate >= 1-1-2021 ^ BranchID = 'B1'}
    - m6 = {OrderDate >= 1-1-2021 ^ BranchID = 'B0'}
    - m7 = {OrderDate < 1-1-2021 ^ BranchID = 'B0'}
  + Phân mảnh:
    - Phân mảnh vị từ m6 tại Site 1 (Ngày thuê từ 1-1-2021 trở đi tại   
      Hà Nội).
    - Phân mảnh vị từ m3 tại Site 2 (Ngày thuê từ 1-1-2021 trở đi tại TP.HCM).
    - Phân mảnh 2 vị từ m2, m7 tại Site 3 (Ngày thuê từ trước 1-1-2021 tại Hà Nội và TP.HCM).

## Kiểm soát dữ liệu

### Xây dựng trigger chèn

Các trigger đảm bảo được:

* **Đảm bảo khóa chính là duy nhất**: Trong các phân mảnh ngang, 1 bảng được phân mảnh ở nhiều trạm. Nên cần đảm bảo khi chèn giá trị khóa vào 1 mảnh thì giá trị khóa đó chưa tồn tại ở các mảnh còn lại.
* **Đảm bảo tham chiếu**: Chỉ được chèn các giá trị cho khóa ngoại khi giá trị đó tồn tại ở bảng tham chiếu đến. Không được xóa giá trị khóa chính khi đang tồn tại một giá trị tham chiếu đến
* Site 1:
* Trigger chèn cho bảng BRANCH: *trigger insertBranch*
* Trigger chèn cho bảng CATEGORIES: *trigger insertCategories*
* Trigger chèn cho bảng PUBLISHINGCOMPANY: *trigger insertPC*
* Trigger chèn cho bảng AUTHOR: *trigger insertAuthor*
* Trigger chèn cho bảng CUSTOMERS: *trigger insertCustomer*
* Trigger chèn cho bảng BOOKSTOCK: *trigger insertBookStockHanoi, insertBookStockHCM*
* Trigger chèn cho bảng ORDERS: *trigger insertOrder*
* Site 2:
* Trigger chèn cho bảng BRANCH: *trigger insertBranch*
* Trigger chèn cho bảng CATEGORIES: *trigger insertCategories*
* Trigger chèn cho bảng PUBLISHINGCOMPANY: *trigger insertPC*
* Trigger chèn cho bảng AUTHOR: *trigger insertAuthor*
* Trigger chèn cho bảng CUSTOMERS: *trigger insertCustomer*
* Trigger chèn cho bảng BOOKSTOCK: *trigger insertBookStock*
* Trigger chèn cho bảng ORDERS: *trigger insertOrder*
* Site 3:
* Trigger chèn cho bảng BRANCH: *trigger insertBranch*
* Trigger chèn cho bảng CATEGORIES: *trigger insertCategories*
* Trigger chèn cho bảng PUBLISHINGCOMPANY: *trigger insertPC*
* Trigger chèn cho bảng AUTHOR: *trigger insertAuthor*
* Trigger chèn cho bảng CUSTOMERS: *trigger insertCustomer*
* Trigger chèn cho bảng BOOKSTOCK: *trigger insertBookStock*
* Trigger chèn cho bảng ORDERS: *trigger insertOrderHanoi, insertOrderHCM*

### Xây dựng trigger xóa

* Site 1:
* Trigger xóa cho bảng BRANCH: *trigger deleteBranch*
* Trigger xóa cho bảng CATEGORIES: *trigger deleteCategory*
* Trigger xóa cho bảng PUBLISHINGCOMPANY: *trigger deletePC*
* Trigger xóa cho bảng AUTHOR: *trigger deleteAuthor*
* Trigger xóa cho bảng CUSTOMERS: *trigger deleteCustomer*
* Trigger xóa cho bảng BOOKSTOCK: *trigger deleteBookstockHanoi, deleteBookstockHCM*
* Trigger xóa cho bảng ORDERS: *trigger insertOrder*
* Site 2:
* Trigger xóa cho bảng BRANCH: trigger deleteBranch
* Trigger xóa cho bảng CATEGORIES: trigger deleteCategory
* Trigger xóa cho bảng PUBLISHINGCOMPANY: trigger deletePC
* Trigger xóa cho bảng AUTHOR: trigger deleteAuthor
* Trigger xóa cho bảng CUSTOMERS: trigger deleteCustomer
* Trigger xóa cho bảng BOOKSTOCK: trigger deleteBookstockHCM
* Site 3:
* Trigger xóa cho bảng BRANCH: trigger deleteBranch
* Trigger xóa cho bảng CATEGORIES: trigger deleteCategory
* Trigger xóa cho bảng PUBLISHINGCOMPANY: trigger deletePC
* Trigger xóa cho bảng AUTHOR: trigger deleteAuthor
* Trigger xóa cho bảng CUSTOMERS: trigger deleteCustomer
* Trigger xóa cho bảng BOOKSTOCK: trigger deleteBookstockHCM

## Tối ưu câu truy vấn

Giả sử có lược đồ sau:

**PUBLISHINGCOMPANY** (PublCompanyID, PublCompanyName, PostalCode, Address, City, FAX, Phone)

**BOOKSTOCK** (BookID, BookName, UnitPrice, UnitsInStock, UnitsOnOrder, AuthorID, PublCompanyID, CategoriesID, BranchID)

**ORDERS** (OrderID, OrderDate, RequiredDate, BookID, CustomerID, BranchID)

Các mảnh được phân như sau:

* **BẢNG BOOKSTOCK:**

**BOOKSTOCK = BOOKSTOCK\_HanoiMoreThan60000 U BOOKSTOCK\_HCMMoreThan60000 U BOOKSTOCK\_HanoiNoMoreThan60000 U BOOKSTOCK\_HCMNoMoreThan60000**

* Trong đó:
* BOOKSTOCK\_HanoiMoreThan60000 chứa sách có giá lớn hơn 60000 và có ở chi nhánh Hà Nội.
* BOOKSTOCK\_HCMMoreThan60000 chứa sách có giá lớn hơn 60000 và có ở chi nhánh HCM.
* BOOKSTOCK\_HanoiNoMoreThan60000 chứa sách có ít lớn hơn 60000 và có ở chi nhánh Hà Nội.
* BOOKSTOCK\_HCMNoMoreThan60000 chứa sách có ít lớn hơn 60000 và có ở chi nhánh HCM.
* **BẢNG ORDERS:**

**ORDERS = ORDERS\_HanoiLater2021 U ORDERS\_HCMLater2021 U ORDERS\_HanoiBefore2021 U ORDERS\_HCMBefore2021**

* Trong đó:
  + ORDERS\_HanoiLater2021 chứa những đơn thuê tại Hà Nội và thuê từ ngày 1/1/2021
  + ORDERS\_HCMLater2021 chứa những đơn thuê tại HCM và thuê từ ngày 1/1/2021
  + ORDERS\_HanoiBefore2021 chứa những đơn thuê tại Hà Nội và thuê trước ngày 1/1/2021
  + ORDERS\_HCMBefore2021 chứa những đơn thuê tại HCM và thuê trước ngày 1/1/2021
* **BẢNG PUBLISHINGCOMPANY** (Phân mảnh ngang dẫn xuất)**:**

**PUBLISHINGCOMPANY\_HanoiMoreThan60000 = PUBLISHINGCOMPANY⋉ BOOKSTOCK\_HanoiMoreThan60000**

**PUBLISHINGCOMPANY\_HCMMoreThan60000 = PUBLISHINGCOMPANY⋉ BOOKSTOCK\_HCMMoreThan60000**

**PUBLISHINGCOMPANY\_HanoiNoMoreThan60000 = PUBLISHINGCOMPANY⋉ BOOKSTOCK\_HanoiNoMoreThan60000**

**PUBLISHINGCOMPANY\_HCMNoMoreThan60000 = PUBLISHINGCOMPANY⋉ BOOKSTOCK\_HCMNoMoreThan60000**

* **Câu truy vấn 1:**

SELECT PublCompanyName

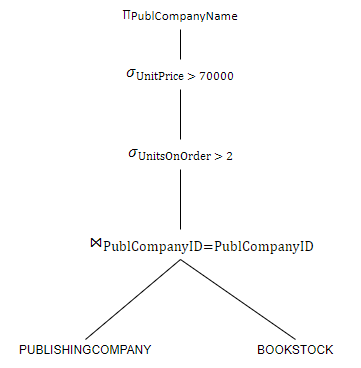
FROM PUBLISHINGCOMPANY pc, BOOKSTOCK bs

WHERE pc.PublCompanyID = bs.PublCompanyID AND UnitPrice > 70000 AND UnitsOnOrder > 2

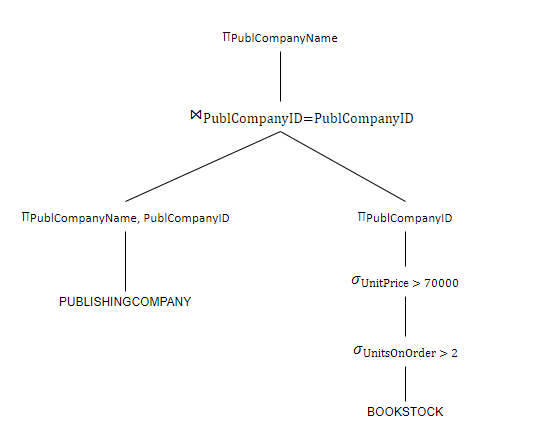
* Chuyển đổi thành biểu thức đại số quan hệ:

∏PublCompanyName (σ(UnitPrice >70000)˄(UnitsOnOrder >2) (PUBLISHINGCOMPANY⋈PublCompanyID=PublCompanyID BOOKSTOCK))

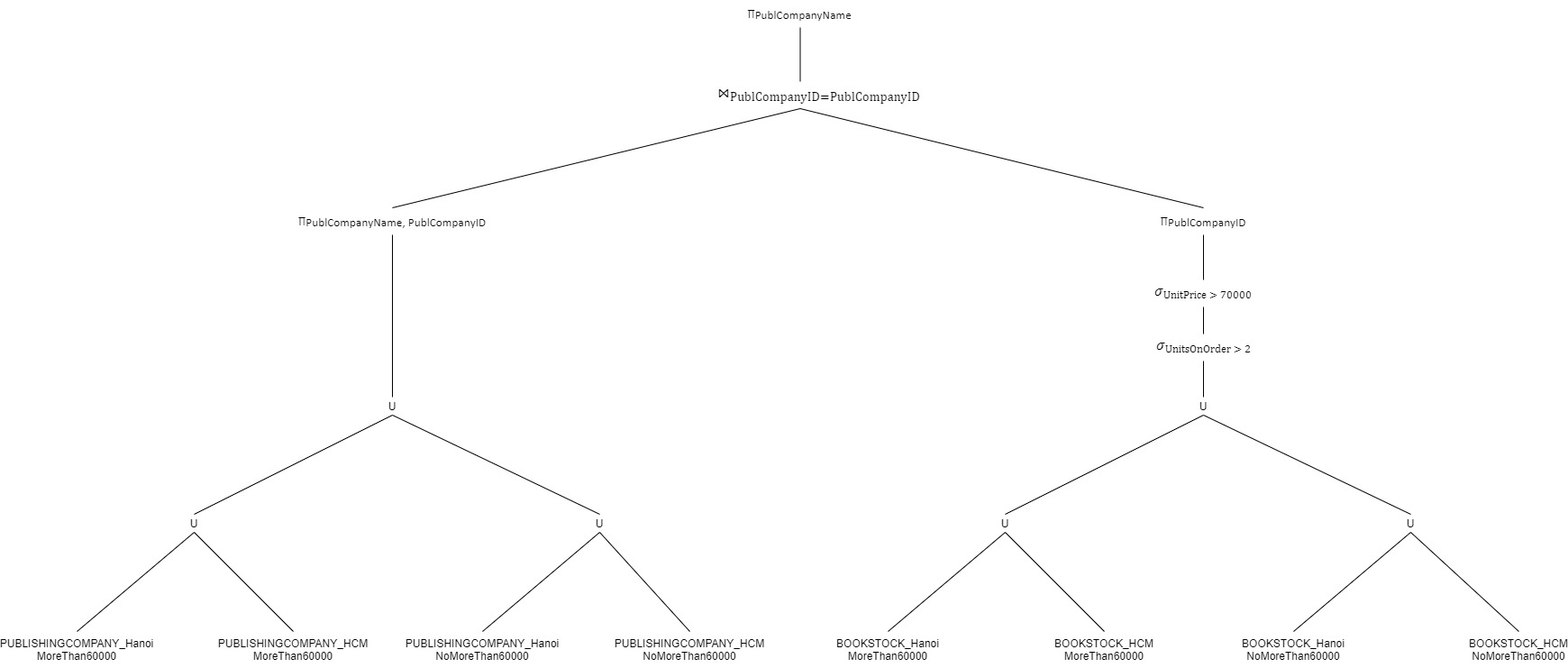
* Xây dựng cây toán tử:



* Đơn giản cây truy vấn:



* Chuyển cây truy vấn trên lược đồ toàn cục sang lược đồ phân mảnh:



* Chuyển phép chiếu và phép chọn xuống dưới phép hợp ta có:
* ∏PublCompanyID ((UnitPrice >70000)˄(UnitsOnOrder >2)( BOOKSTOCK\_HanoiMoreThan60000 U BOOKSTOCK\_HCMMoreThan60000 U BOOKSTOCK\_HanoiNoMoreThan60000 U BOOKSTOCK\_HCMNoMoreThan60000))

= ∏PublCompanyID ((UnitPrice >70000)˄(UnitsOnOrder >2)( BOOKSTOCK\_HanoiMoreThan60000)) U

∏PublCompanyID ((UnitPrice >70000)˄(UnitsOnOrder >2)( BOOKSTOCK\_HCMMoreThan60000)) U

∏PublCompanyID ((UnitPrice >70000)˄(UnitsOnOrder >2)( BOOKSTOCK\_HanoiNoMoreThan60000)) U

∏PublCompanyID ((UnitPrice >70000)˄(UnitsOnOrder >2)( BOOKSTOCK\_HCMNoMoreThan60000))

* ∏PublCompanyID, PublCompanyName( PUBLISHINGCOMPANY\_HanoiMoreThan60000 U PUBLISHINGCOMPANY\_HCMMoreThan60000 U PUBLISHINGCOMPANY\_HanoiNoMoreThan60000 U PUBLISHINGCOMPANY\_HCMNoMoreThan60000)

= ∏PublCompanyID, PublCompanyName( PUBLISHINGCOMPANY\_HanoiMoreThan60000) U

∏PublCompanyID, PublCompanyName( PUBLISHINGCOMPANY\_HCMMoreThan60000) U

∏PublCompanyID, PublCompanyName( PUBLISHINGCOMPANY\_HanoiNoMoreThan60000) U

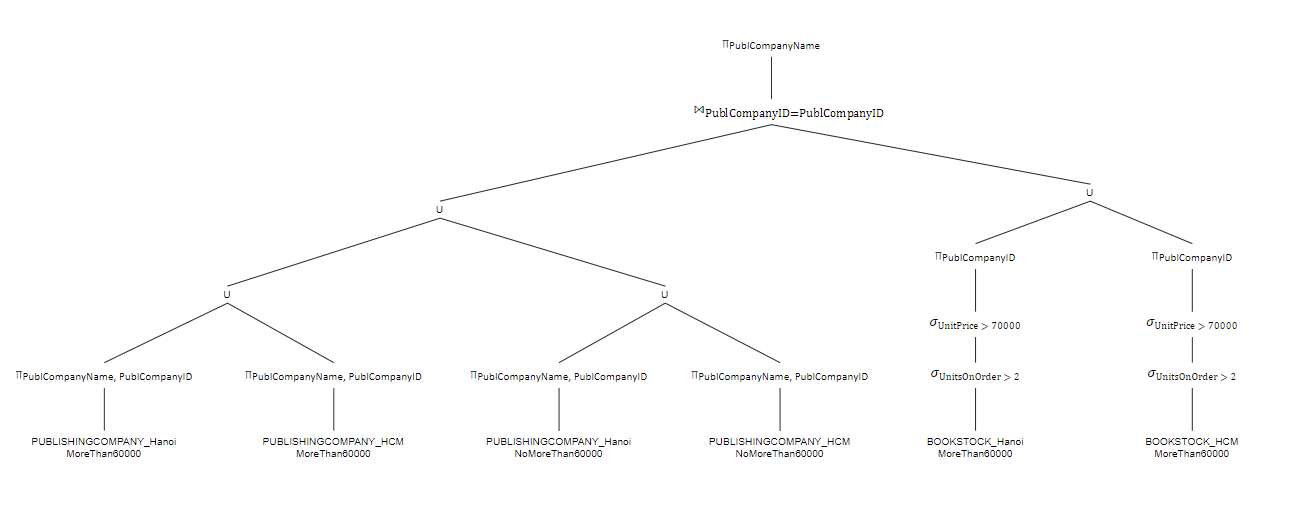
∏PublCompanyID, PublCompanyName( PUBLISHINGCOMPANY\_HCMNoMoreThan60000)

* Nhận thấy:

∏PublCompanyID ((UnitPrice >70000)˄(UnitsOnOrder >2)(BOOKSTOCK\_HanoiNoMoreThan60000)) và

∏PublCompanyID ((UnitPrice >70000)˄(UnitsOnOrder >2)(BOOKSTOCK\_HCMNoMoreThan60000)) là ∅

* Được cây như sau:



* Phân phối phép nối và phép hợp:
* Nhận thấy:

PUBLISHINGCOMPANY\_HanoiMoreThan60000 BOOKSTOCK\_HCMMoreThan60000,

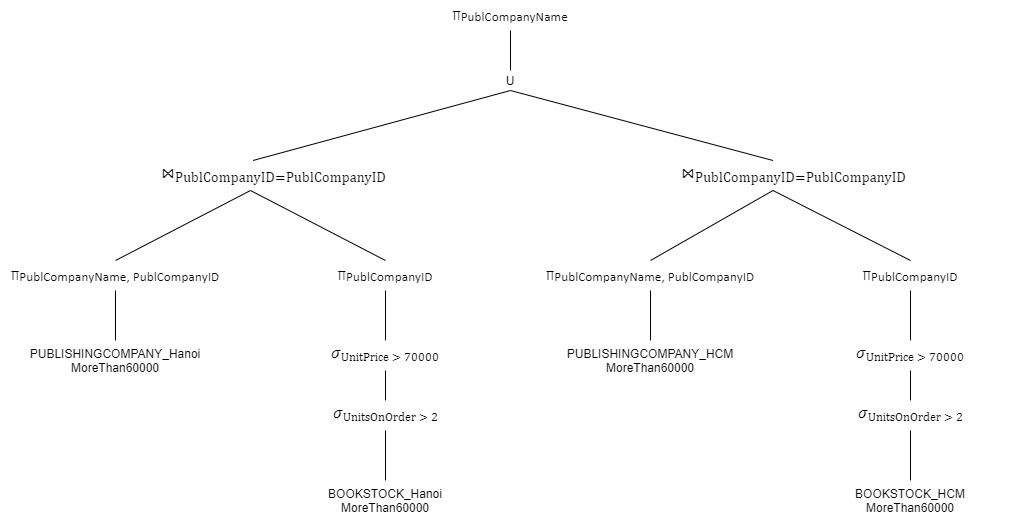
PUBLISHINGCOMPANY\_HCMMoreThan60000 BOOKSTOCK\_HanoiMoreThan60000,

PUBLISHINGCOMPANY\_HanoiNoMoreThan60000 BOOKSTOCK\_HanoiMoreThan60000,

PUBLISHINGCOMPANY\_HanoiNoMoreThan60000 BOOKSTOCK\_HCMMoreThan60000,

PUBLISHINGCOMPANY\_HCMNoMoreThan60000 BOOKSTOCK\_HanoiMoreThan60000,

PUBLISHINGCOMPANY\_HCMNoMoreThan60000 BOOKSTOCK\_HCMMoreThan60000 đều là ∅

* Cuối cùng ta được cây truy vấn sau: 

* Viết lại truy vấn sau khi đã tối ưu:

SELECT PublCompanyName

FROM PUBLISHINGCOMPANY\_HanoiMoreThan60000 a, BOOKSTOCK\_HanoiMoreThan60000 b

WHERE a.PublCompanyID = b.PublCompanyID AND b.UnitPrice > 70000 AND b.UnitsOnOrder > 2

UNION

SELECT PublCompanyName

FROM PUBLISHINGCOMPANY\_HCMMoreThan60000 c, BOOKSTOCK\_HCMMoreThan60000 d

WHERE c.PublCompanyID = d.PublCompanyID AND d.UnitPrice > 70000 AND d.UnitsOnOrder > 2

* **Câu truy vấn 2:**

SELECT BookName

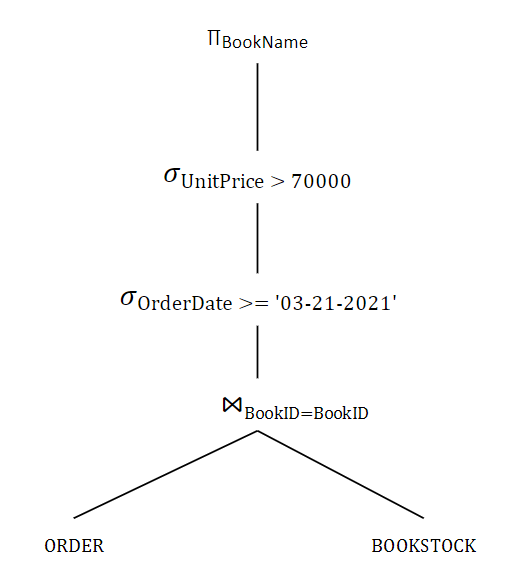
FROM ORDERS o, BOOKSTOCK bs

WHERE o.BookID = bs.BookID AND OrderDate >= ‘03-21-2021’ AND UnitPrice > 70000

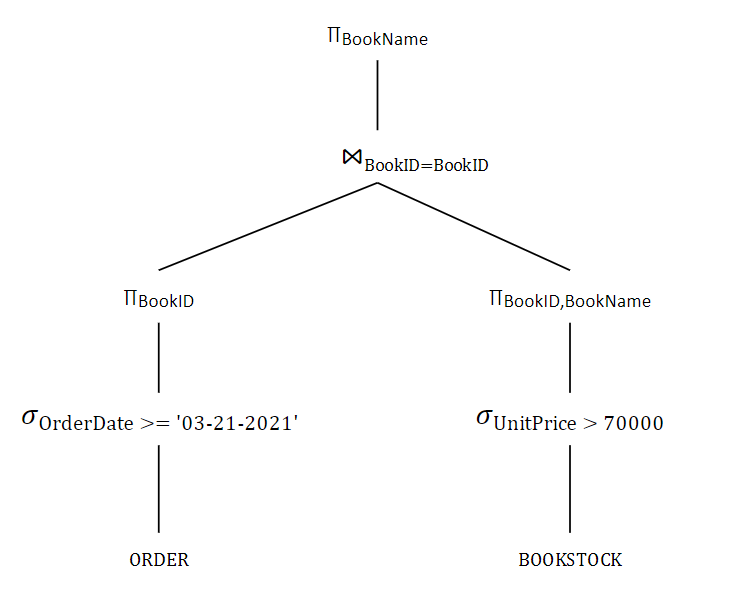
* Chuyển câu truy vấn về dạng đại số quan hệ:

∏BookName (σ(OrderDate >= ‘03-21-2021’)˄(UnitPrice > 70000) (ORDERS⋈BookID=BookID BOOKSTOCK))

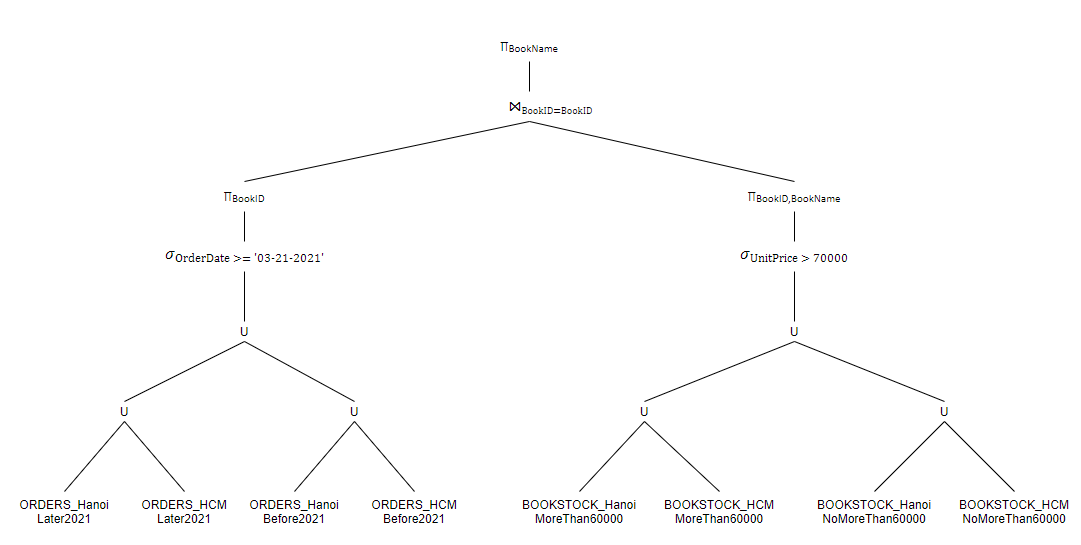
* Xây dựng cây toán tử:



* Đơn giản cây truy vấn:



* Chuyển cây truy vấn từ lược đồ toàn cục sang lược đồ phân mảnh:



* Chuyển phép chiếu và phép chọn xuống dưới phép hợp ta có:
* ∏BookID,BookName (σ(UnitPrice >70000)( BOOKSTOCK\_HanoiMoreThan60000 U BOOKSTOCK\_HCMMoreThan60000 U BOOKSTOCK\_HanoiNoMoreThan60000 U BOOKSTOCK\_HCMNoMoreThan60000))

= ∏BookID,BookName (σ(UnitPrice >70000)( BOOKSTOCK\_HanoiMoreThan60000)) U

∏BookID,BookName (σ(UnitPrice >70000)( BOOKSTOCK\_HCMMoreThan60000)) U

∏BookID,BookName (σ(UnitPrice >70000)( BOOKSTOCK\_HanoiNoMoreThan60000)) U

∏BookID,BookName (σ(UnitPrice >70000)( BOOKSTOCK\_HCMNoMoreThan60000))

* ∏BookID(σ (OrderDate >= ’03-21-2021’)(ORDERS\_HanoiLater2021 U ORDERS\_HCMLater2021 U ORDERS\_HanoiBefore2021 U ORDERS\_HCMBefore2021))

= ∏BookID(σ(OrderDate >= ’03-21-2021’)(ORDERS\_HanoiLater2021)) U

∏BookID(σ(OrderDate >= ’03-21-2021’)(ORDERS\_HCMLater2021)) U

∏BookID(σ(OrderDate >= ’03-21-2021’)(ORDERS\_HanoiBefore2021)) U

∏BookID(σ(OrderDate >= ’03-21-2021’)(ORDERS\_HCMBefore2021))

* Nhận thấy:

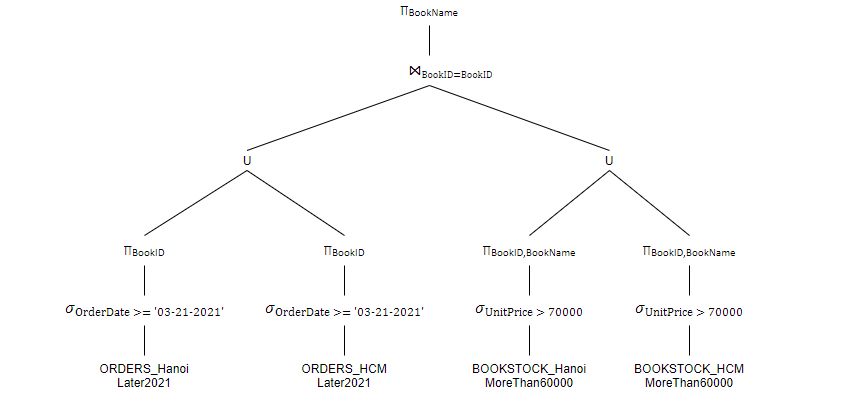
∏BookID (σ(UnitPrice >70000)( BOOKSTOCK\_HanoiNoMoreThan60000)),

∏BookID (σ(UnitPrice >70000)( BOOKSTOCK\_HCMNoMoreThan60000)),

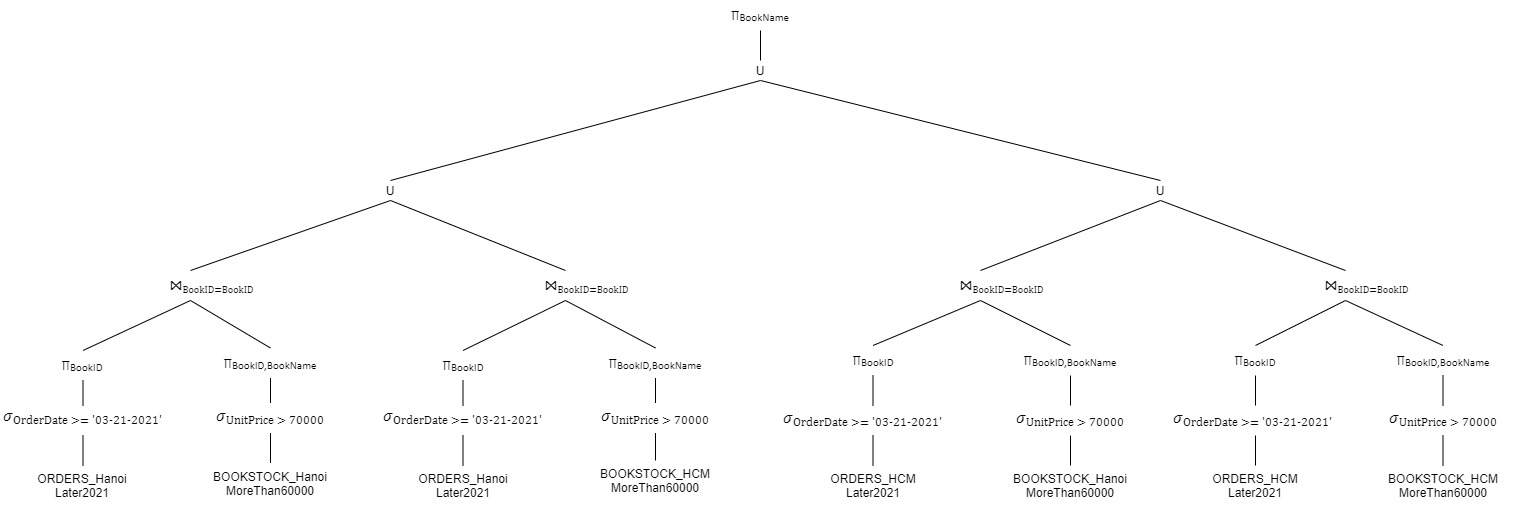
∏BookID(σ(OrderDate >= ’03-21-2021’)(ORDERS\_HanoiBefore2021)) và

∏BookID(σ(OrderDate >= ’03-21-2021’)(ORDERS\_HCMBefore2021)) đều là ∅

* Ta được cây như sau:



* Phân phối phép nối và phép hợp ta được:



* Viết lại câu truy vấn sau khi đã tối ưu:

SELECT BookName

FROM ORDERS\_HanoiLater2021 a, BOOKSTOCK\_HanoiMoreThan60000 b

WHERE a.BookID = b.BookID AND OrderDate >= ’03-21-2021’ AND UnitPrice > 70000

UNION

SELECT BookName

FROM ORDERS\_HanoiLater2021 a, BOOKSTOCK\_HCMMoreThan60000 b

WHERE a.BookID = b.BookID AND OrderDate >= ’03-21-2021’ AND UnitPrice > 70000

UNION

SELECT BookName

FROM ORDERS\_HCMLater2021 a, BOOKSTOCK\_HanoiMoreThan60000 b

WHERE a.BookID = b.BookID AND OrderDate >= ’03-21-2021’ AND UnitPrice > 70000

UNION

SELECT BookName

FROM ORDERS\_HCMLater2021 a, BOOKSTOCK\_HCMMoreThan60000 b

WHERE a.BookID = b.BookID AND OrderDate >= ’03-21-2021’ AND UnitPrice > 70000

* **Câu truy vấn 3:**

SELECT BookName

FROM PUBLISHINGCOMPANY a, BOOKSTOCK b

WHERE a.PublCompanyID = b.PublCompanyID AND PostalCode = 100000 AND

UnitPrice < 45000 AND BookID IN ( SELECT c.BookID

FROM BOOKSTOCK c, ORDERS d

WHERE c.BookID = d.BookID AND

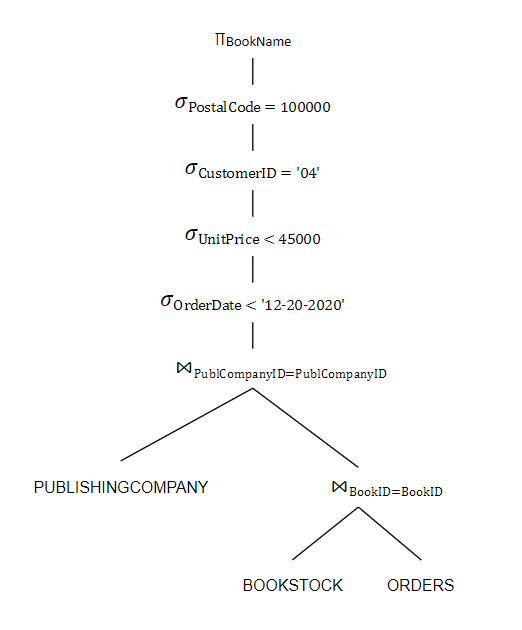
CustomerID =’04' AND

OrderDate < ’12-20-2020’)

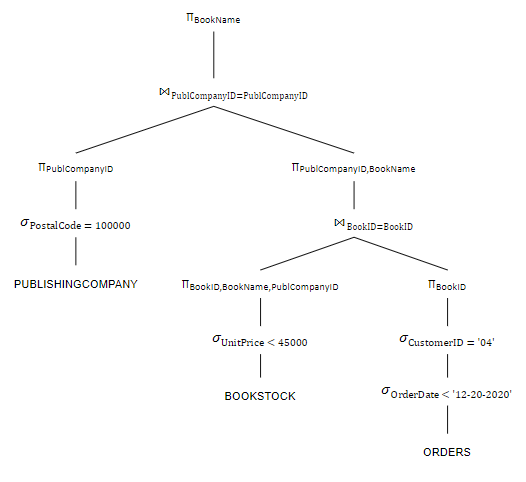
* Chuyển câu truy vấn về dạng đại số quan hệ:

∏BookName (σ(PostalCode = 100000)˄(CustomerID =’04’)˄(UnitPrice < 45000)˄(OrderDate < ’12-20-2020’)(PUBLISHINGCOMPANY⋈PublCompanyID=PublCompanyID (BOOKSTOCK⋈BookID=BookID ORDERS))

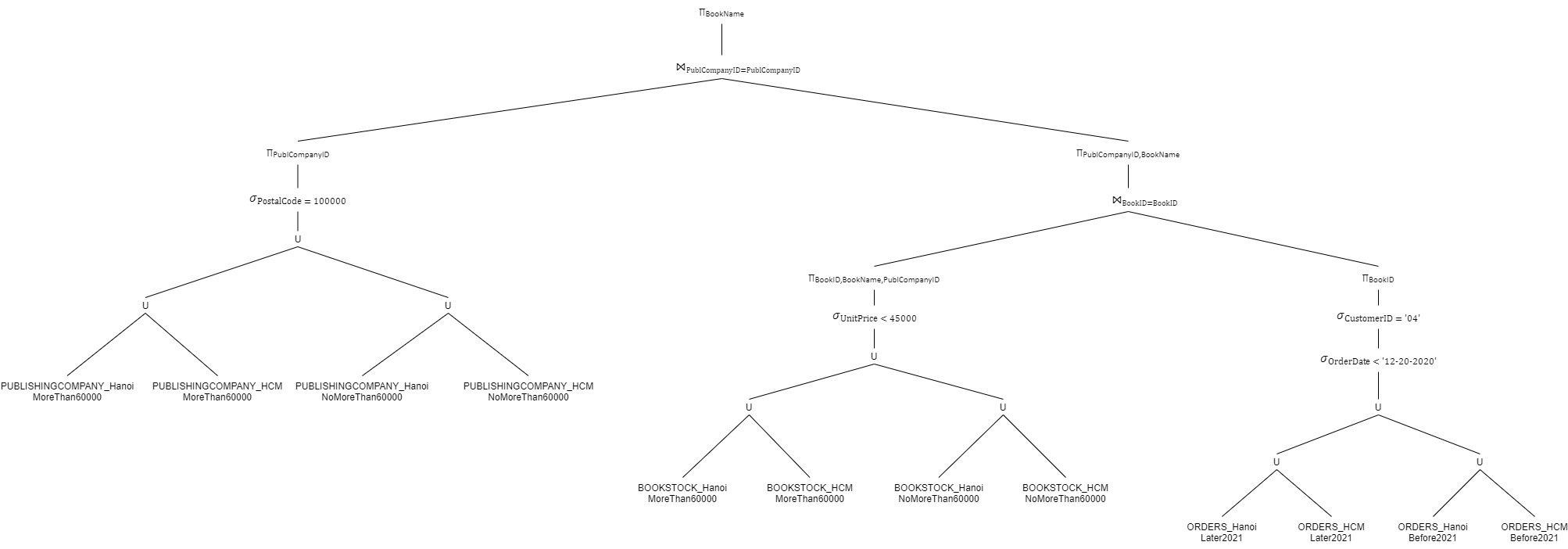
* Xây dựng cây truy vấn:



* Đơn giản cây truy vấn:



* Chuyển cây truy vấn từ biểu đồ toàn cục sang biểu đồ phân mảnh:



* Chuyển phép chiếu và phép chọn xuống dưới phép hợp ta có:
* ∏BookID,BookNamePublCompanyID(σ(UnitPrice < 45000)( BOOKSTOCK\_HanoiMoreThan60000 U BOOKSTOCK\_HCMMoreThan60000 U BOOKSTOCK\_HanoiNoMoreThan60000 U BOOKSTOCK\_HCMNoMoreThan60000))

= ∏BookID,BookNamePublCompanyID(σ(UnitPrice < 45000)( BOOKSTOCK\_HanoiMoreThan60000)) U

∏BookID,BookNamePublCompanyID(σ(UnitPrice < 45000)( BOOKSTOCK\_HCMMoreThan60000)) U

∏BookID,BookNamePublCompanyID(σ(UnitPrice < 45000)( BOOKSTOCK\_HanoiNoMoreThan60000)) U

∏BookID,BookNamePublCompanyID(σ(UnitPrice < 45000)( BOOKSTOCK\_HCMNoMoreThan60000))

* ∏BookID(σ(OrderDate < ’12-20-2020’) ˄ (CustomerID = ’04’)(ORDERS\_HanoiLater2021 U ORDERS\_HCMLater2021 U ORDERS\_HanoiBefore2021 U ORDERS\_HCMBefore2021))

= ∏BookID(σ(OrderDate < ’12-20-2020’) ˄ (CustomerID = ’04’)(ORDERS\_HanoiLater2021)) U

∏BookID(σ(OrderDate < ’12-20-2020’) ˄ (CustomerID = ’04’)(ORDERS\_HCMLater2021)) U

∏BookID(σ(OrderDate < ’12-20-2020’) ˄ (CustomerID = ’04’)(ORDERS\_HanoiBefore2021)) U

∏BookID(σ(OrderDate < ’12-20-2020’) ˄ (CustomerID = ’04’)(ORDERS\_HCMBefore2021))

* ∏PublCompanyID(σ(PostalCode = 100000)(PUBLISHINGCOMPANY\_HanoiMoreThan60000 U PUBLISHINGCOMPANY\_HCMMoreThan60000 U PUBLISHINGCOMPANY\_HanoiNoMoreThan60000 U PUBLISHINGCOMPANY\_HCMNoMoreThan60000))

= ∏PublCompanyID(σ(PostalCode = 100000)(PUBLISHINGCOMPANY\_HanoiMoreThan60000)) U

∏PublCompanyID(σ(PostalCode = 100000)(PUBLISHINGCOMPANY\_HCMMoreThan60000)) U

∏PublCompanyID(σ(PostalCode = 100000)(PUBLISHINGCOMPANY\_HanoiNoMoreThan60000)) U

∏PublCompanyID(σ(PostalCode = 100000)(PUBLISHINGCOMPANY\_HCMNoMoreThan60000))

* Nhận thấy:

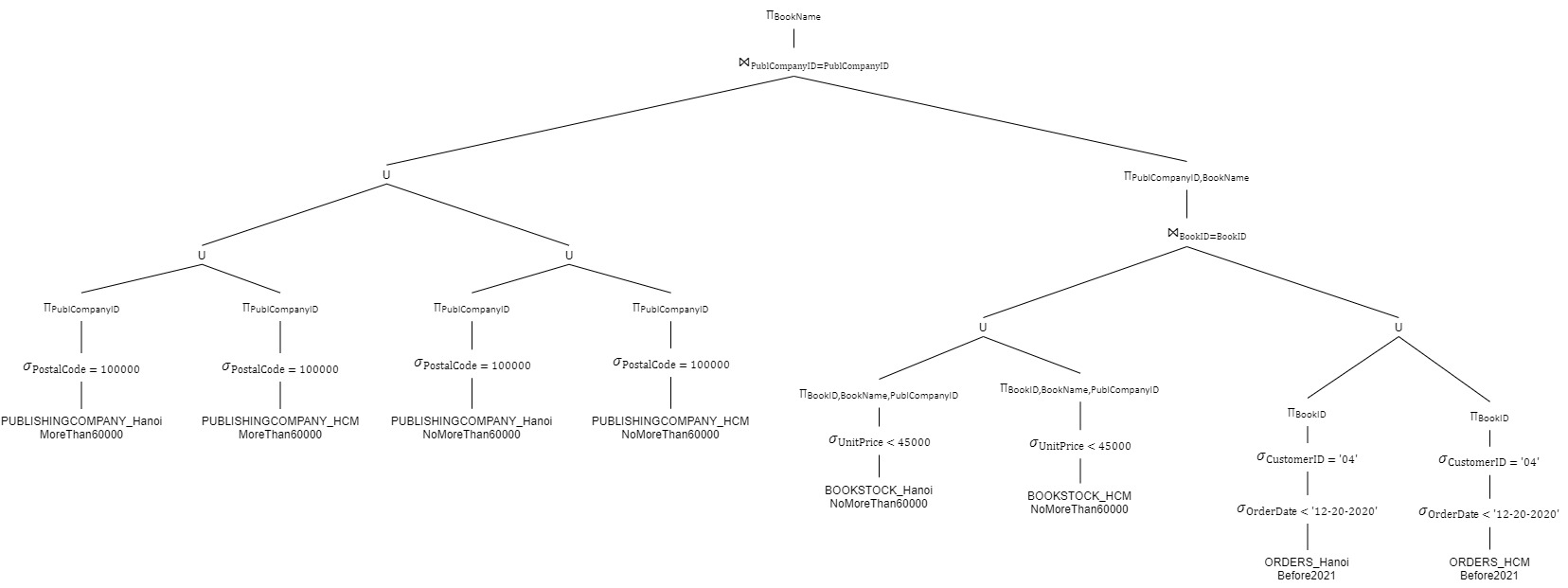
∏BookID,BookNamePublCompanyID(σ(UnitPrice < 45000)( BOOKSTOCK\_HanoiMoreThan60000)),

∏BookID,BookNamePublCompanyID(σ(UnitPrice < 45000)( BOOKSTOCK\_HCMMoreThan60000)),

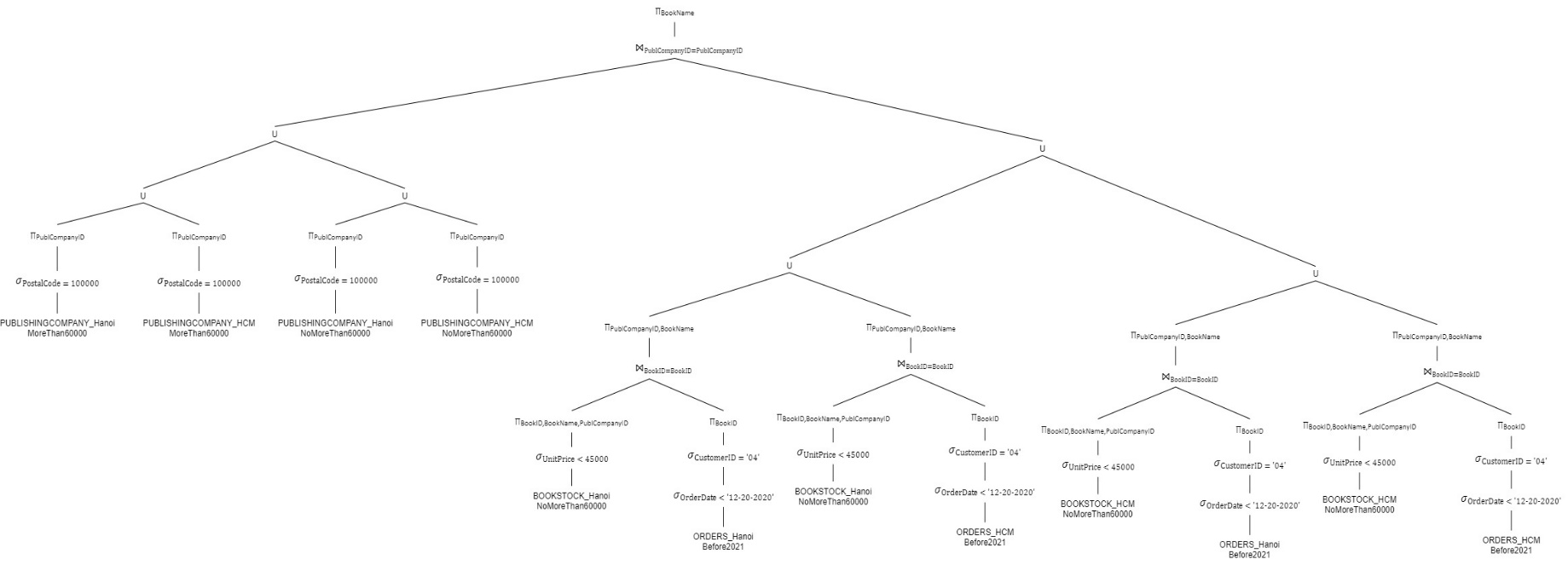
∏BookID(σ(OrderDate < ’12-20-2020’) ˄ (CustomerID = ’04’)(ORDERS\_HanoiLater2021)),

∏BookID(σ(OrderDate < ’12-20-2020’) ˄ (CustomerID = ’04’)(ORDERS\_HCMLater2021)) đều là ∅

* Ta được cây sau:



* Tiếp tục chuyển phép chiếu xuống dưới phép hợp:



* Tiếp tục dựa theo tính chất phân phối của phép hợp chuyển phép giao xuống dưới phép hợp
* Nhận thấy:

PUBLISHINGCOMPANY\_HanoiMoreThan60000 BOOKSTOCK\_HanoiNoMoreThan60000,

PUBLISHINGCOMPANY\_HanoiMoreThan60000 BOOKSTOCK\_HCMNoMoreThan60000,

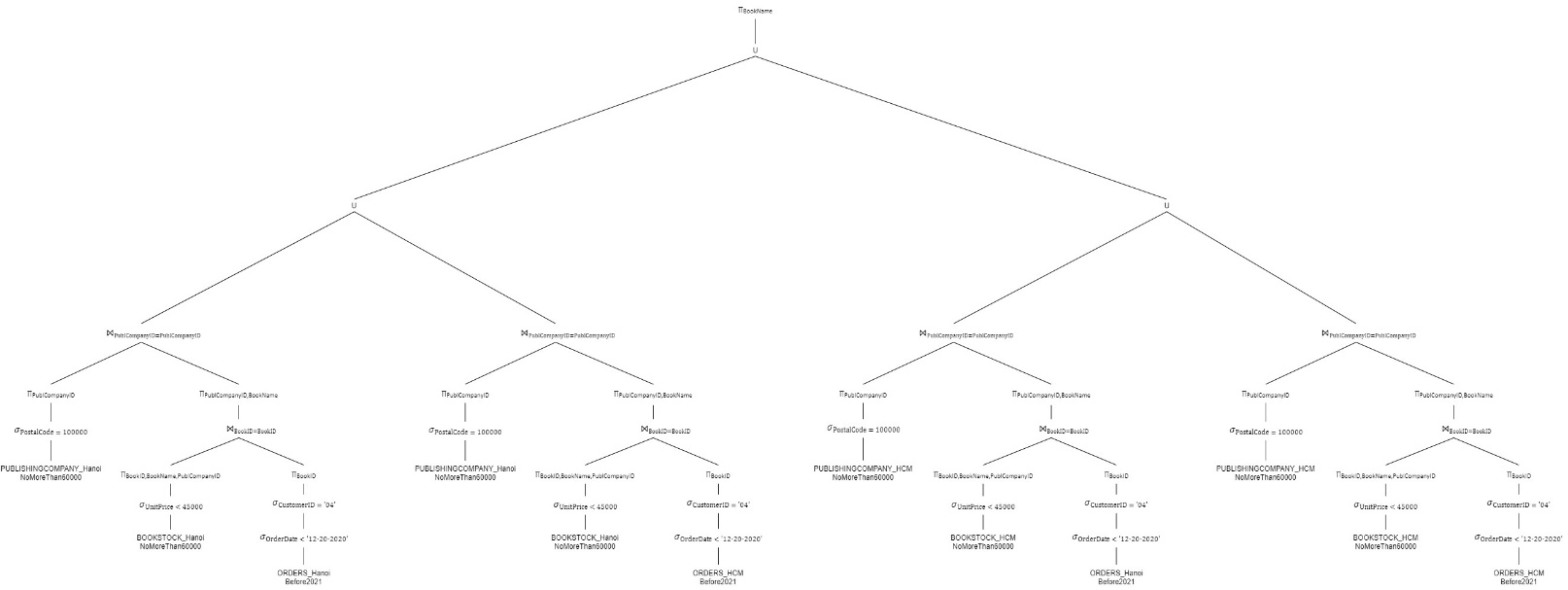
PUBLISHINGCOMPANY\_HCMMoreThan60000 BOOKSTOCK\_HanoiNoMoreThan60000,

PUBLISHINGCOMPANY\_HCMMoreThan60000 BOOKSTOCK\_HCMNoMoreThan60000,

PUBLISHINGCOMPANY\_HanoiNoMoreThan60000 BOOKSTOCK\_HCMNoMoreThan60000,

PUBLISHINGCOMPANY\_HCMNoMoreThan60000 BOOKSTOCK\_HanoiNoMoreThan60000 đều là ∅

* Cuối cùng ta được cây truy vấn:



* Viết lại câu truy vấn dựa trên cây truy vấn đã tối ưu:

SELECT BookName

FROM PUBLISHINGCOMPANY\_HanoiNoMoreThan60000 a,

BOOKSTOCK\_HanoiNoMoreThan60000 b

WHERE a.PublCompanyID = b.PublCompanyID AND PostalCode = 100000 AND

UnitPrice < 45000 AND BookID IN

( SELECT c.BookID

FROM BOOKSTOCK\_HanoiNoMoreThan60000 c, ORDERS\_HanoiBefore2021 d

WHERE c.BookID = d.BookID AND

CustomerID =’04' AND OrderDate < ’12-20-2020’)

UNION

SELECT BookName

FROM PUBLISHINGCOMPANY\_HanoiNoMoreThan60000 a,

BOOKSTOCK\_HanoiNoMoreThan60000 b

WHERE a.PublCompanyID = b.PublCompanyID AND PostalCode = 100000 AND

UnitPrice < 45000 AND BookID IN

( SELECT c.BookID

FROM BOOKSTOCK\_HanoiNoMoreThan60000 c, ORDERS\_HCMBefore2021 d

WHERE c.BookID = d.BookID AND

CustomerID =’04' AND OrderDate < ’12-20-2020’)

UNION

SELECT BookName

FROM PUBLISHINGCOMPANY\_HCMNoMoreThan60000 a,

BOOKSTOCK\_HCMNoMoreThan60000 b

WHERE a.PublCompanyID = b.PublCompanyID AND PostalCode = 100000 AND

UnitPrice < 45000 AND BookID IN

( SELECT c.BookID

FROM BOOKSTOCK\_HCMNoMoreThan60000 c, ORDERS\_HanoiBefore2021 d

WHERE c.BookID = d.BookID AND

CustomerID =’04' AND OrderDate < ’12-20-2020’)

UNION

SELECT BookName

FROM PUBLISHINGCOMPANY\_HCMNoMoreThan60000 a,

BOOKSTOCK\_HCMNoMoreThan60000 b

WHERE a.PublCompanyID = b.PublCompanyID AND PostalCode = 100000 AND

UnitPrice < 45000 AND BookID IN

( SELECT c.BookID

FROM BOOKSTOCK\_HCMNoMoreThan60000 c, ORDERS\_HCMBefore2021 d

WHERE c.BookID = d.BookID AND

CustomerID =’04' AND OrderDate < ’12-20-2020’)

## Xây dựng các thao tác dữ liệu

### Xây dựng thủ tục chèn

* Thủ tục chèn được nhân bản:
* Thủ tục chèn cho bảng BRANCH: *proc insert\_branch*
* Thủ tục chèn cho bảng CATEGORIES: *proc insert\_categories*
* Thủ tục chèn cho bảng PUBLISHINGCOMPANY: *proc insert\_publCom*
* Site 1:
* Thủ tục chèn cho bảng BRANCH: *proc insert\_branch*
* Thủ tục chèn cho bảng CATEGORIES: *proc insert\_categories*
* Thủ tục chèn cho bảng PUBLISHINGCOMPANY: *proc insert\_publCom*
* Thủ tục chèn cho bảng AUTHOR: *proc insert\_author*
* Thủ tục chèn cho bảng CUSTOMERS: *proc insert\_customers*
* Thủ tục chèn cho bảng BOOKSTOCK: *proc insert\_bookstock*
* Thủ tục chèn cho bảng ORDERS: *proc insert\_order*
* Site 2:
* Thủ tục chèn cho bảng AUTHOR: *proc insert\_author*
* Thủ tục chèn cho bảng CUSTOMERS: *proc insert\_customers*
* Thủ tục chèn cho bảng BOOKSTOCK: *proc insert\_bookstock*
* Thủ tục chèn cho bảng ORDERS: *proc insert\_order*
* Site 3:
* Thủ tục chèn cho bảng AUTHOR: *proc insert\_author*
* Thủ tục chèn cho bảng CUSTOMERS: *proc insert\_customers*
* Thủ tục chèn cho bảng BOOKSTOCK: *proc insert\_bookstock*
* Thủ tục chèn cho bảng ORDERS: *proc insert\_order*

### Xây dựng thủ tục sửa

* Thủ tục sửa được nhân bản:
* Thủ tục sửa cho bảng BRANCH: *proc edit\_branch*
* Thủ tục sửa cho bảng CATEGORIES: *proc edit\_categories*
* Thủ tục sửa cho bảng PUBLISHINGCOMPANY: *proc edit\_pc*
* Thủ tục sửa cho bảng BOOKSTOCK: *proc edit\_bookstockName*
* Thủ tục sửa cho bảng ORDERS: *proc edit\_order*
* Site 1:
* Thủ tục sửa cho bảng BRANCH: *proc edit\_branch*
* Thủ tục sửa cho bảng CATEGORIES: *proc edit\_categories*
* Thủ tục sửa cho bảng PUBLISHINGCOMPANY: *proc edit\_pc*
* Thủ tục sửa cho bảng BOOKSTOCK: *proc edit\_bookstockName*
* Thủ tục sửa cho bảng AUTHOR: *proc edit\_author*
* Thủ tục sửa cho bảng CUSTOMERS: *proc edit\_customer*
* Thủ tục sửa cho bảng ORDERS: *proc edit\_order*
* Site 2:
* Thủ tục sửa cho bảng AUTHOR: *proc edit\_author*
* Thủ tục sửa cho bảng CUSTOMERS: *proc edit\_customer*
* Site 3:
* Thủ tục sửa cho bảng AUTHOR: *proc edit\_author*
* Thủ tục sửa cho bảng CUSTOMERS: *proc edit\_customer*

### Xây dựng thủ tục xóa

* Thủ tục xóa được nhân bản:
* Thủ tục xóa cho bảng BRANCH: proc delete\_branch
* Thủ tục xóa cho bảng CATEGORIES: proc delete\_categories
* Thủ tục xóa cho bảng PUBLISHINGCOMPANY: proc delete\_publCom
* Site 1:
* Thủ tục xóa cho bảng BRANCH: *proc delete\_branch*
* Thủ tục xóa cho bảng CATEGORIES: *proc delete\_categories*
* Thủ tục xóa cho bảng PUBLISHINGCOMPANY: *proc delete\_publCom*
* Thủ tục xóa cho bảng AUTHOR: *proc delete\_author*
* Thủ tục xóa cho bảng CUSTOMERS: *proc delete\_customers*
* Thủ tục xóa cho bảng BOOKSTOCK: *proc delete\_bookstock*
* Thủ tục xóa cho bảng ORDERS: *proc delete\_orders*
* Site 2:
* Thủ tục xóa cho bảng AUTHOR: *proc delete\_author*
* Thủ tục xóa cho bảng CUSTOMERS: *proc delete\_customers*
* Thủ tục xóa cho bảng BOOKSTOCK: *proc delete\_bookstock*
* Thủ tục xóa cho bảng ORDERS: *proc delete\_orders*
* Site 3:
* Thủ tục xóa cho bảng AUTHOR: *proc delete\_author*
* Thủ tục xóa cho bảng CUSTOMERS: *proc delete\_customers*
* Thủ tục xóa cho bảng BOOKSTOCK: *proc delete\_bookstock*
* Thủ tục xóa cho bảng ORDERS: *proc delete\_orders*

**KẾT LUẬN**

Theo như ta thấy, việc sử dụng cơ sở dữ liệu phân tán là một bước tiến lớn trong việc xử lý và lưu trữ dữ liệu, nhưng bên cạnh đó vẫn còn tồn tại những bất cập trong việc bảo mật và cần kỹ năng lập trình vững chắc cho nhà xử lý dữ liệu hoặc lập trình viên hay người làm IT nói chung.

Bên cạnh những bất cập đó, chúng ta cũng không thể phủ nhận được những lợi ích to lớn mà nó đem lại, nó giúp các công ty có nguồn dữ liệu lớn trên thế giới mà dữ liệu được phân bổ tại nhiều đất nước khác nhau có thể truy cập được một cách nhanh chóng.

Tài liệu này trình bày tổng quan về CSDL phân tán, bao gồm xử lý dữ liệu phân tán, khái niệm hệ CSDL phân tán, kiến trúc phân tán, khả năng của CSDL phân tán, một số vấn đề cần quan tâm trong CSDL phân tán và cách xây dựng cơ sở dữ liệu phân tán cho thuê truyện bao gồm mô tả bài toán cho thuê truyện phân tán, thiết kế phân tán, kiểm soát dữ liệu, tối ưu truy vấn, xây dựng các thao tác dữ liệu. Đây là cơ sở lý thuyết để giúp những người muốn tìm hiểu và nghiên cứu về CSDL phân tán, giúp người sử dụng có thể áp dụng để phát triển các hệ thống lớn trong môi trường phân tán và phạm vi sử dụng lớn.

Dựa trên tài liệu về cơ sở dữ liệu cho thuê truyện phân tán, một vài hướng phát triển có thể được đặt ra. Về mặt lý thuyết, tài liệu có thể được dùng để tham khảo làm luận án, khóa luận tốt nghiệp nghiên cứu về hệ cơ sở dữ liệu phân tán. Về mặt ứng dụng, có thể phát triển lên hỗ trợ cho một thư viện cho thuê sách, truyện cấp quốc gia mà có nhiều chi nhanh trên toàn quốc. Cần ứng dụng thêm thuật toán và tối ưu truy vấn để khiến việc truy vấn vào các dữ liệu ở nhiều nơi một cách mượt mà hơn.

**TÀI LIỆU THAM KHẢO**

1. Sách “Principles of distributed database systems” (1991) – Tác giả: M. Tamer Özsu
2. Tài liệu “Cơ sở dữ liệu phân tán” – Biên soạn: TS. Phạm Thế Quế. <http://dulieu.tailieuhoctap.vn/books/cong-nghe-thong-tin/co-so-du-lieu/file_goc_767856.pdf>

1. \* Kiến trúc tham chiếu thường được các nhà phát triển tiêu chuẩn tạo ra để xác định rõ ràng các giao diện cần được tiêu chuẩn hóa. [↑](#footnote-ref-1)