

**TRƯỜNG ĐẠI HỌC HÀNG HẢI
KHOA CÔNG NGHỆ THÔNG TIN
BỘ MÔN HỆ THỐNG THÔNG TIN**
-----***-----



BÀI GIẢNG

CƠ SỞ DỮ LIỆU NÂNG CAO

TÊN HỌC PHẦN : CƠ SỞ DỮ LIỆU NÂNG CAO
MÃ HỌC PHẦN : 17406
TRÌNH ĐỘ ĐÀO TẠO : ĐẠI HỌC CHÍNH QUY
DÙNG CHO SV NGÀNH: CÔNG NGHỆ THÔNG TIN

HẢI PHÒNG – 2011

MỤC LỤC

CHƯƠNG 1: CÁC MÔ HÌNH DỮ LIỆU	7
1.1. Các mô hình dữ liệu	7
1.1.1. Mô hình thực thể - quan hệ	7
1.1.2. Mô hình mạng	8
1.1.3. Mô hình phân cấp	8
1.2. Tóm tắt về cơ sở dữ liệu quan hệ	9
1.2.1. Các khái niệm	9
1.2.2. Tóm tắt các phép toán đại số quan hệ	10
1.2.3. Phương pháp xây dựng cơ sở dữ liệu quan hệ	10
1.3. Một số mô hình cơ sở dữ liệu mở rộng	11
1.3.1. Mô hình cơ sở dữ liệu thời gian	11
1.3.2. Mô hình cơ sở dữ liệu tích cực	12
1.4. Câu hỏi ôn tập chương 1	14
CHƯƠNG 2: TỐI ƯU HÓA TRUY VẤN	15
2.1. Giới thiệu	15
2.2. Các phép biến đổi tương đương	15
2.3. Thuật toán tối ưu hóa cây đại số quan hệ	17
2.3.1. Thuật toán	17
2.3.2. Ví dụ	17
2.4. Câu hỏi ôn tập chương 2	19
CHƯƠNG 3: GIAO DỊCH TRONG CƠ SỞ DỮ LIỆU	20
3.1. Giới thiệu	20
3.2. Các tính chất của giao dịch	20
3.3. Các trạng thái của giao dịch	21
3.4. Lịch biểu	21
3.5. Tính khả tuần tự của lịch biểu	22
3.6. Thuật toán kiểm tra tính khả tuần tự của lịch biểu	23
3.7. Câu hỏi ôn tập chương 3	24
CHƯƠNG 4: ĐIỀU KHIỂN ĐỒNG THỜI VÀ KHÔI PHỤC HỆ THỐNG	25
4.1. Các giao thức dựa vào khóa	25
4.1.1. Mô hình khóa nhị phân	25
4.1.2. Mô hình khóa đọc – ghi (chia sẻ – độc quyền)	25

4.1.3. Giao thức khóa 2 pha.....	27
4.1.4. Deadlock.....	29
4.2. Giao thức thứ tự nhãn thời gian (Timestamp – Ordering protocol).....	31
4.2.1 Nhãn thời gian (Timestamp).....	31
4.2.2. Giao thức thứ tự nhãn thời gian (Timestamp – Ordering Protocol)	31
4.3. Phục hồi hệ thống dựa vào nhật ký giao dịch (Log-based).....	32
4.3.1 Cập nhật trì hoãn cơ sở dữ liệu (Deferred Database Modification)	32
4.3.2. Cập nhật tức thời (Immediate Database Modification)	34
4.4. Kỹ thuật phân trang bóng (Shadow Paging)	34
4.5. Câu hỏi ôn tập chương 4	37
CHƯƠNG 5: CƠ SỞ DỮ LIỆU PHÂN TÁN	38
5.1. Giới thiệu	38
5.2. Ưu nhược điểm của cơ sở dữ liệu phân tán	39
5.3. Cấu trúc chung của một cơ sở dữ liệu phân tán	39
5.4. Phân đoạn dữ liệu (Fragmentation).....	40
5.4.1. Phân đoạn ngang.....	42
5.4.2. Phân đoạn ngang dẫn xuất	42
5.4.3. Phân đoạn dọc.....	43
5.4.4. Phân đoạn hỗn hợp	43
5.5. Thiết kế định vị dữ liệu	44
5.6. Biến đổi truy vấn toàn cục thành truy vấn đoạn	45
5.7. Câu hỏi ôn tập chương 5	47
MỘT SỐ ĐỀ THI MẪU.....	48

Tên học phần: Cơ sở dữ liệu nâng cao

Loại học phần: 2

Bộ môn phụ trách giảng dạy: Hệ thống Thông tin

Khoa phụ trách: CNTT.

Mã học phần: 17406

Tổng số TC: 2

Tổng số tiết	Lý thuyết	Thực hành/Xemina	Tự học	Bài tập lớn	Đồ án môn học
45	30	15	0	không	không

Học phần học trước: Cơ sở dữ liệu.

Học phần tiên quyết: Cơ sở dữ liệu.

Học phần song song: Không yêu cầu.

Mục tiêu của học phần:

Cung cấp cho sinh viên những kiến thức nâng cao về cơ sở dữ liệu quan hệ, kiến thức về hệ cơ sở dữ liệu phân tán.

Nội dung chủ yếu:

Các vấn đề nâng cao về cơ sở dữ liệu: Các mô hình dữ liệu mở rộng, tối ưu hóa câu hỏi truy vấn, quản lý giao dịch trong cơ sở dữ liệu, điều khiển tương tranh và phục hồi hệ thống; Các vấn đề về cơ sở dữ liệu phân tán: kiến trúc hệ cơ sở dữ liệu phân tán, thiết kế cơ sở dữ liệu phân tán, xử lý truy vấn phân tán.

Nội dung chi tiết:

TÊN CHƯƠNG MỤC	PHÂN PHỐI SỐ TIẾT				
	TS	LT	TH	BT	KT
Chương 1: Các mô hình dữ liệu	6	4	2		
1.1. Các mô hình dữ liệu					
1.2. Tóm tắt về cơ sở dữ liệu quan hệ					
1.3. Một số mô hình cơ sở dữ liệu mở rộng					
Chương 2: Tối ưu hóa truy vấn	6	5			1
2.1. Giới thiệu					
2.2. Các phép biến đổi tương đương					
2.3. Thuật toán tối ưu hóa cây đại số quan hệ					
Chương 3: Giao dịch trong cơ sở dữ liệu	7	3	3		1
3.1. Giới thiệu					

TÊN CHƯƠNG MỤC	PHÂN PHỐI SỐ TIẾT				
	TS	LT	TH	BT	KT
3.2. Các tính chất của giao dịch					
3.3. Các trạng thái của giao dịch					
3.4. Lịch biểu					
3.5. Tính khả tuần tự của lịch biểu					
3.6. Thuật toán kiểm tra tính khả tuần tự của lịch biểu					
Chương 4: Điều khiển đồng thời và khôi phục hệ thống	6	6			
4.1. Các giao thức dựa vào khóa					
4.2. Giao thức thứ tự nhãn thời gian					
4.3. Phục hồi hệ thống dựa vào nhật ký giao dịch (Log-based)					
4.4. Kỹ thuật phân trang bóng (Shadow Paging)					
Chương 5: Cơ sở dữ liệu phân tán	20	9	10		1
5.1. Giới thiệu					
5.2. Ưu nhược điểm của cơ sở dữ liệu phân tán					
5.3. Cấu trúc chung của một cơ sở dữ liệu phân tán					
5.4. Phân đoạn dữ liệu (Fragmentation)					
5.5. Thiết kế định vị dữ liệu					
5.6. Biến đổi truy vấn toàn cục thành truy vấn đoạn					

Nhiệm vụ của sinh viên:

Tham dự các buổi học lý thuyết và thực hành, làm các bài tập được giao, làm các bài thi giữa học phần và bài thi kết thúc học phần theo đúng quy định.

Tài liệu học tập:

1. Nguyễn Xuân Huy, *Giáo trình về cơ sở dữ liệu*, Đại học Quốc Gia Hà Nội, 2000.
2. Raghu Ramakrishnan, *Database Management Systems*, McGraw – Hill, 1998.
3. Silberschatz, Korth, Sudarshan, *Database System Concepts*, 4th Edition, McGraw-Hill, 2001.

Hình thức và tiêu chuẩn đánh giá sinh viên:

- Hình thức thi: thi viết.
- Tiêu chuẩn đánh giá sinh viên: căn cứ vào sự tham gia học tập của sinh viên trong các buổi học lý thuyết và thực hành, kết quả làm các bài tập được giao, kết quả của các bài thi giữa học phần và bài thi kết thúc học phần.

Thang điểm: Thang điểm chữ A, B, C, D, F.

Điểm đánh giá học phần: $Z = 0,3X + 0,7Y$.

Bài giảng này là tài liệu **chính thức và thống nhất** của Bộ môn Hệ thống Thông tin, Khoa Công nghệ Thông tin và được dùng để giảng dạy cho sinh viên.

Ngày phê duyệt: / /

Trưởng Bộ môn

CHƯƠNG 1: CÁC MÔ HÌNH DỮ LIỆU

1.1. Các mô hình dữ liệu

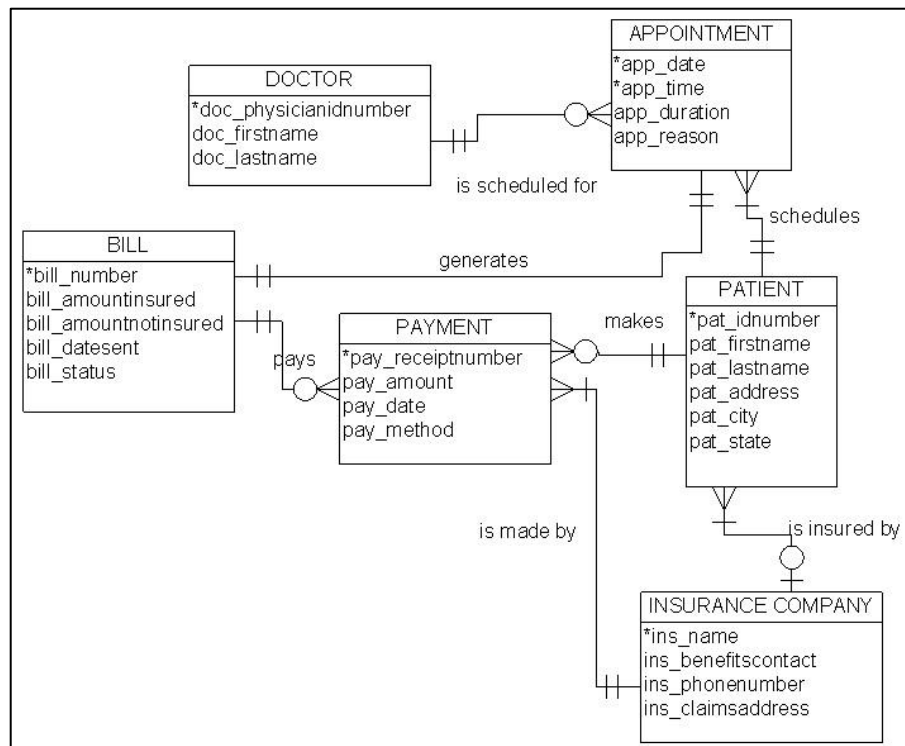
Mô hình dữ liệu là sự trừu tượng hóa môi trường thực, đó là tập hợp các mô tả dữ liệu ở mức cao mà không quan tâm đến chi tiết lưu trữ dữ liệu ở mức thấp. Trong phạm vi môn học ta xét các mô hình dữ liệu sau: Mô hình thực thể - quan hệ, mô hình mạng, mô hình phân cấp, mô hình hướng đối tượng.

1.1.1. Mô hình thực thể - quan hệ

Mô hình thực thể - liên kết (ER Model) là thể hiện logic của dữ liệu trong tổ chức hay trong công tác kinh doanh. Nó mô tả thế giới thực gồm những đối tượng cơ bản được gọi là các thực thể và những quan hệ giữa chúng.

Sơ đồ thực thể quan hệ (Entity Relationship Diagram - ERD) là thể hiện đồ họa của ER.

- Thực thể là thành phần của thế giới thực, có khả năng tồn tại độc lập. Ví dụ một tài khoản ngân hàng là một thực thể (Entity). Một nhóm các thực thể cùng chung một số tính chất, một số đặc trưng cơ bản tạo ra một tập thực thể (Entity Set). Ví dụ tập các nhân viên trong một cơ quan.
- Thuộc tính dùng để mô tả thực thể, ví dụ: Với thực thể khách hàng thì thuộc tính là: tên khách hàng, địa chỉ, số điện thoại, ... Thuộc tính có thể nhận một hoặc nhiều giá trị.
- Các tập thực thể có thể liên kết với nhau theo mỗi liên kết 1 – 1, 1 – n, n – m.



Hình 1.1: Ví dụ mô hình ER (Crow's Foot)

1.1.2. Mô hình mạng

Mô hình mạng (Network Model) là mô hình thực thể - quan hệ trong đó các mối liên hệ bị hạn chế trong kiểu nhị phân và nhiều – một. Như vậy chúng ta có thể dùng một đồ thị có hướng cho các dữ liệu. Ở vị trí của các tập thực thể, mô hình mạng đưa ra kiểu bản ghi (record type). Một kiểu bản ghi là tên gán cho một tập các bản ghi (record). Bản ghi được cấu tạo bởi các trường (field) chứa các giá trị cơ bản như số nguyên, chuỗi kí tự, ... Tập các tên trường và kiểu của chúng tạo nên khuôn dạng bản ghi (record format). Nếu so sánh với mô hình quan hệ ta thấy sự tương ứng như sau:

Mô hình mạng	Mô hình quan hệ
Kiểu bản ghi	Tên quan hệ
Bản ghi	Bộ của quan hệ
Khuôn dạng bản ghi	Lược đồ quan hệ

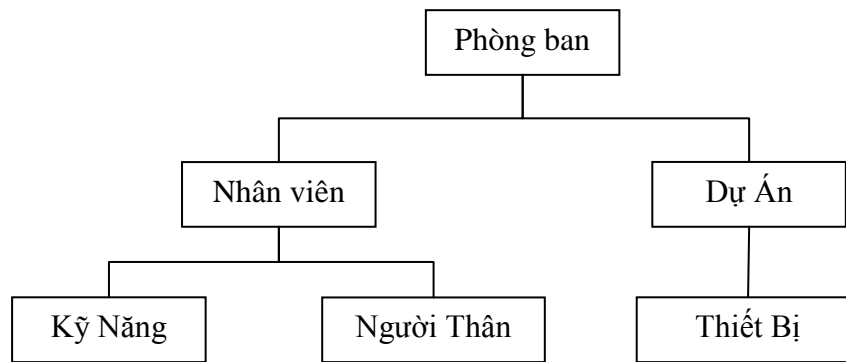
- Đường nối (Link): dùng để biểu diễn mối quan hệ nhiều – một giữa hai đối tượng. Chúng ta dùng đồ thị có hướng, gọi là mạng, đó thực sự là một sơ đồ thực thể - quan hệ đã được đơn giản hóa để biểu diễn các kiểu bản ghi và đường nối giữa chúng. Các nút tương ứng với các kiểu bản ghi.
- Biểu diễn tập thực thể trong mô hình mạng: Các tập thực thể được biểu diễn bằng các kiểu bản ghi; thuộc tính của tập thực thể trở thành các trường của khuôn dạng bản ghi.
- Biểu diễn các mối quan hệ: Chỉ những mối quan hệ một – nhiều mới có thể biểu diễn trực tiếp bằng các đường nối. Tuy nhiên ta có thể dùng một số phương pháp để biểu diễn một mối quan hệ bất kỳ.

1.1.3. Mô hình phân cấp

Trong mô hình này dữ liệu được tổ chức thành cấu trúc cây, các nút (node) là tập các thực thể, các cạnh là các mối quan hệ giữa hai nút theo mối quan hệ nhất định, cứng nhắc. Mô hình phân cấp:

- Là mô hình dữ liệu trong đó các bản ghi được sắp xếp theo cấu trúc top-down(tree).
- Một nút con chỉ liên kết tới một nút cha.

Tập dữ liệu được tổ chức theo cấu trúc của mô hình dữ liệu phân cấp gọi là CSDL phân cấp. Hình 1.2 là một ví dụ về mô hình phân cấp quản lý nhân sự của một công ty.



Hình 1.2: Ví dụ mô hình phân cấp trong quản lý nhân sự

1.2. Tóm tắt về cơ sở dữ liệu quan hệ

1.2.1. Các khái niệm

Cho $\Omega = \{A_1, A_2, \dots, A_n\}$ là một tập hữu hạn các thuộc tính, khác rỗng và các phần tử không nhất thiết phân biệt nhau. Mỗi thuộc tính A_i có một miền giá trị xác định $D(A_i) = \text{Dom}(A_i)$. Tích Đề các của các miền $\text{Dom}(A_1), \text{Dom}(A_2), \dots, \text{Dom}(A_n)$ được biểu diễn như sau:

$$\text{Dom}(A_1) \times \text{Dom}(A_2) \times \dots \times \text{Dom}(A_n) = \{(a_1, a_2, \dots, a_n) \mid a_i \in A_i, i = 1 \div n\}.$$

Nói rằng R là một quan hệ trên tập $\Omega \leftrightarrow R \subseteq \text{Dom}(A_1) \times \text{Dom}(A_2) \times \dots \times \text{Dom}(A_n)$.

Ví dụ: $\Omega = \{A_1, A_2\}$, $\text{Dom}(A_1) = \{0, 1\}$ và $\text{Dom}(A_2) = \{a, b, c\}$

Như vậy tích Đề các $\text{Dom}(A_1) \times \text{Dom}(A_2) = \{(0,a), (0,b), (0,c), (1,a), (1,b), (1,c)\}$. Khi đó tập $R = \{(0,a), (0,b)\}$ là một quan hệ trên tập Ω .

Trong mô hình cơ sở dữ liệu quan hệ người ta thường biểu diễn một quan hệ trên tập các thuộc tính Ω bằng một bảng hai chiều. Các thuộc tính của quan hệ là các cột A_1, A_2, \dots, A_n . Các hàng của quan hệ gọi là các bộ, kí hiệu $r \in R(\Omega)$.

- Ví dụ: Bảng sau thể hiện quan hệ Can_bo bao gồm các thuộc tính Ma_So, Ho_ten, Ngay_sinh, Que_quan, Chuc_vu

	Cán Bộ				
	Ma_so	Ho_ten	Ngay_sinh	Que_quan	Chuc_vu
t1	0001	Lê Thanh Hằng	12/04/1970	Quảng Ninh	Trưởng phòng
t2	0002	Hoàng Văn Cảnh	15/10/1981	Nam Định	Nhân viên
t3	0003	Trần Bình Dương	20/01/1979	Ninh Bình	Nhân viên

Khoá của một quan hệ là một hoặc một tập thuộc tính dùng để định tên duy nhất các bộ trong quan hệ.

Khoá ngoại: Trong cơ sở dữ liệu có thể có rất nhiều các quan hệ, và đôi khi quan hệ này có những thuộc tính lấy dữ liệu từ một quan hệ khác, hay nói các khác là các thuộc tính này có miền giá trị là những giá trị được lưu tại một hoặc nhiều trường trong quan hệ khác hoặc chính nó.

1.2.2. Tóm tắt các phép toán đại số quan hệ

- Phép chọn (Selection): $\delta_E(R) = \{t \mid t \in R \ \& \ t[E] = \text{“True”}\}$
- Phép chiếu (Projection): $\Pi_X(R) = \{t[X] \mid t \in R\}$
- Phép hợp (Union): $R_1 \cup R_2 = \{t \mid t \in R_1 \vee t \in R_2\}$
- Phép giao (Intersection): $R_1 \cap R_2 = \{t \mid t \in R_1 \wedge t \in R_2\}$
- Phép trừ (Minus): $R_1 - R_2 = \{t \mid t \in R_1 \wedge t \notin R_2\}$
- Tích đề các (Cartesian Product): $R \times S = \{t \mid t = (a,b) \wedge a \in R \wedge b \in S\}$
- Phép kết nối θ (Join): $R \bowtie S = \{(t,u) \mid t \in R \wedge u \in S \wedge t[\Omega] \theta u[\Sigma]\}$
- Kết nối tự nhiên (Natural Join): $R * S = P(\Omega \cup \Sigma) = \{t[\Omega \cup \Sigma] \mid t[\Omega \cap \Sigma]_R = t[\Omega \cap \Sigma]_S\}$
- Phép chia (Division): $R \div S = \{t \mid \forall s \in S, (t,s) \in R\}$

1.2.3. Phương pháp xây dựng cơ sở dữ liệu quan hệ

1.2.3.1. Phương pháp Top – Down

- Phân tích thực tiễn:
 - Xác định các đối tượng có trong cơ sở dữ liệu
 - Các mối liên hệ giữa chúng
 - Các quy tắc quản lý
- Xây dựng lược đồ liên kết thực thể - ER:
 - Kiểu thực thể: Một lớp đối tượng
 - Kiểu liên kết: Một lớp các mối liên hệ: 1 – 1, 1 – n, n – m
 - Dùng thuật toán chuyển đổi từ lược đồ ER sang lược đồ quan hệ.

1.2.3.2. Phương pháp Bottom – Up

- Tạo ra lược đồ phổ quát $R(A_1, A_2, \dots, A_n)$
- Tìm ra tập phụ thuộc hàm F
- Tách R thành các R_i đều ở dạng chuẩn BC (BCNF)

1.3. Một số mô hình cơ sở dữ liệu mở rộng

1.3.1. Mô hình cơ sở dữ liệu thời gian

- Cơ sở dữ liệu thời gian (Temporal database) là cơ sở dữ liệu được xây dựng dựa trên khía cạnh thời gian. Ví dụ:
 - CSDL chăm sóc sức khỏe: bệnh án, lịch sử bệnh.
 - Bảo hiểm: bảo hiểm xe, liên quan đến đền bù, sửa chữa.
 - Các hệ thống đặt vé, giữ chỗ ...
- Valid time, transaction time, bitemporal time:
 - Valid Time: Thời gian sự việc xảy ra trong thực tế.
 - VST: Valid Start Time: Thời gian bắt đầu có hiệu lực.
 - VET: Valid End Time: Thời gian kết thúc hiệu lực.
 - Transaction Time: Thời gian khi sự thay đổi được lưu trong CSDL.
 - TST: Transaction Start Time: Thời gian giao dịch được tạo ra.
 - TET: Transaction End Time: Thời gian giao dịch bị hủy bỏ.
 - Bitemporal time: Valid time + Transaction time.
- Các kiểu cập nhật:
 - Proactive: Thông tin được cập nhật vào cơ sở dữ liệu trước khi nó xảy ra trong thế giới thực.
 - Retroactive: Thông tin được cập nhật vào cơ sở dữ liệu sau khi nó xảy ra trong thế giới thực.
 - Simultaneous: Thông tin được cập nhật vào cơ sở dữ liệu cùng thời điểm nó xảy ra trong thế giới thực.
- Ví dụ:
 - 03/04/1975: John ra đời (con trai của Jack và Jane).
 - 04/04/1975: Jack đăng ký khai sinh cho con trai ở Smallville City.
 - 26/08/1994: John chuyển đến sống ở Bigtown (sau khi tốt nghiệp trung học) nhưng quên chưa khai báo thay đổi địa chỉ nơi ở mới.
 - 27/12/1994: John khai báo địa chỉ nơi ở mới.
 - 01/04/2001: John bị tai nạn. Nhân viên điều tra báo John mất cùng ngày.

Với mô hình cơ sở dữ liệu truyền thống: Ta có bảng Person lưu thông tin nhân khẩu: Person (Name, Address):

Ngày	Sự việc trong thực tế	Thao tác CSDL	Ý nghĩa CSDL
03/04/1975	John chào đời.	Không có gì	Không có ai là John
04/04/1975	Jack đăng ký khai sinh cho John.	Inserted:Person(John, Smallville)	John sống ở Smallville
26/08/1994	John chuyển đến Bigtown nhưng chưa khai báo địa chỉ mới.	Không có gì	John sống ở Smallville
27/12/1994	John đăng ký địa chỉ mới.	Updated:Person(John, Bigtown)	John sống ở Bigtown
01/04/2001	John mất.	Deleted:Person(John Doe)	Không có ai là John

Với mô hình cơ sở dữ liệu thời gian (Valid time):

- 3/4/1975, John chào đời (Jack đăng ký khai sinh cho con vào ngày 4/4/1975):
Person(John, Smallville, 3-Apr-1975, ∞).
- 27/12/1994, John khai báo địa chỉ nơi ở mới (John sống ở Bigtown từ 26/8/1994):
Person(John, Smallville, 3-Apr-1975, 26-Aug-1994)
Person (John, Big Town, 26-Aug-1994, ∞).
- 1/4/2001, John mất:
Person(John, Smallville, 3-Apr-1975, 26-Aug-1994)
Person(John, Bigtown, 26-Aug-1994, 1-Apr-2001)

1.3.2. Mô hình cơ sở dữ liệu tích cực

- Với cơ sở dữ liệu thụ động: Mọi hành động trên dữ liệu kết quả chỉ thực hiện với các chương trình ứng dụng được chỉ ra nhưng với cơ sở dữ liệu tích cực: Các hành động có thể xảy ra tự động trong trả lời để giám sát các sự kiện và đảm bảo tính toàn vẹn của dữ liệu. Ví dụ một CSDL tích cực có thể được sử dụng để giám sát nhiệt độ của ngành công nghiệp luyện kim:
 - Làm tăng dự tích hợp các ràng buộc.
 - Tự động bảo vệ dữ liệu nhận được.
 - Bảo vệ các bảng sao nhất quán.
- Cơ sở dữ liệu tích cực là một CSDL chứa các qui tắc tích cực (hoặc các qui tắc gây nên), chủ yếu là các qui tắc dạng ECA (Event – Condition – Action).
- Quy tắc ECA: có tác dụng đảm bảo tính toàn vẹn của cơ sở dữ liệu. Một quy tắc được cấu tạo bởi 3 thành phần:
 - Event: Sự kiện làm kích hoạt quy tắc. Chúng cũng có thể là sự kiện thời gian hoặc một kiểu sự kiện mở rộng khác.

- Condition: Điều kiện làm vi phạm tính toàn vẹn của dữ liệu.
 - Action: Một hành động nhằm loại bỏ sự vi phạm về tính toàn vẹn của dữ liệu. Hành động thường là một câu truy vấn, hoặc một chương trình mở rộng được thực hiện một cách tự động.
- ➔ Mô hình cơ sở dữ liệu tích cực còn được gọi là mô hình ECA.
- Trong cơ sở dữ liệu quan hệ các quy tắc tích cực được viết dưới dạng các Trigger. Ví dụ: Với hai bảng:
 - DonVi (MaDV, TenDV, TongLuong, MaNQL)
 - NhanVien (MaNV, TenNV, NgaySinh, GioiTinh, Luong, MaDV)
 Một quy tắc sẽ có dạng như sau:
 - Event: Hành động Insert một bản ghi trong bảng NhanVien.
 - Condition:
 - Giá trị của MaDV được thêm vào không có trong tập giá trị hiện có trong bảng DonVi.
 - Nếu MaDV được thêm vào là đã có trong bảng DonVi → Tổng lương của đơn vị đó cần được thay đổi.
 - Action:
 - Sửa đổi sai sót do sự kiện gây ra.
 - Thông báo lỗi cho người dùng.
 - Khi thiết kế và cài đặt cơ sở dữ liệu tích cực cần lưu ý một số vấn đề sau:
 - Cần phân tích và xác định đúng đắn các ràng buộc dữ liệu, từ đó mới có tập quy tắc phục vụ đầy đủ cho cơ sở dữ liệu.
 - Phải có phương pháp để các quy tắc có thể được kích hoạt trước hay sau khi sự kiện xảy ra.
 - Tập các quy tắc của một cơ sở dữ liệu tích cực thường rất lớn nên tại một thời điểm không phải tất cả các quy tắc đều được áp dụng, do vậy mỗi quy tắc cần phải có hai trạng thái: Active, Deactive.

1.4. Câu hỏi ôn tập chương 1

- Mô hình cơ sở dữ liệu là gì?
- Trình bày các khái niệm lược đồ quan hệ, quan hệ, trường, thuộc tính trong mô hình cơ sở dữ liệu quan hệ.
- Trình bày các khái niệm khóa, siêu khóa, khóa chính, khóa ngoại, thuộc tính khóa trong mô hình cơ sở dữ liệu quan hệ.
- Trình bày các loại ràng buộc trong mô hình cơ sở dữ liệu quan hệ.
- Trình bày các phép toán đại số quan hệ: phép chọn, phép chiếu, phép hợp, phép giao, phép trừ, phép kết nối, phép chia.
- Mô hình cơ sở dữ liệu thời gian là gì?
- Phân biệt các kiểu thời gian: Valid time, Transaction time, Bitemporal time?
- Mô hình cơ sở dữ liệu tích cực là gì?
- Quy tắc ECA là gì?

CHƯƠNG 2: TỐI ƯU HÓA TRUY VẤN

2.1. Giới thiệu

Trong chương này chúng ta sẽ nghiên cứu một kỹ thuật tối ưu hóa vấn tin cơ bản. Khi diễn tả truy vấn bằng một ngôn ngữ khai báo, như ngôn ngữ vấn tin quan hệ, hệ thống cơ sở dữ liệu rất khó thực thi nhanh chóng. Với nhiều cách cài đặt một câu vấn tin, giai đoạn tối ưu hóa phải chọn ra được một cách cài đặt tốt nhất.

Quá trình xử lý một truy vấn:

- Bước 1:
 - Bộ quét: Truy vấn được bằng một biểu thức. Bộ quét sẽ duyệt truy vấn để biết truy vấn được viết bằng ngôn ngữ nào.
 - Bộ kiểm tra: Kiểm tra cú pháp của truy vấn xem có hợp lệ hay không.
 - Xác nhận tính hợp lệ (các quan hệ, thuộc tính sử dụng trong truy vấn đã được khai báo hay chưa? Sau bước 1 truy vấn sẽ được biểu diễn bằng một biểu thức đại số quan hệ)
- Bước 2:
 - Bộ tối ưu: Tìm ra phương pháp thực hiện tối ưu cho truy vấn.
 - Sau bước này sẽ cho ra một biểu thức đại số quan hệ với chi phí thực hiện nhỏ nhất.
- Bước 3:
 - Bộ tạo mã sẽ tạo ra chương trình bằng ngôn ngữ trong để thực hiện truy vấn.
 - Thực thi chương trình để lấy về kết quả.

2.2. Các phép biến đổi tương đương

- Các truy vấn được viết bằng các ngôn ngữ bậc cao, ví dụ SQL, sau bước 1 của quá trình xử lý truy vấn, truy vấn được biểu diễn bằng biểu thức đại số quan hệ. Ví dụ:

✓ SELECT	*		
FROM	R	↔	$\delta_E(R)$
WHERE	E		
✓ SELECT	A_1, A_2, \dots, A_i		
FROM R		↔	$\Pi_F(R)$
✓ SELECT	*		
FROM R, S		↔	$R \bowtie_F S$
WHERE	F		

- Tối ưu bằng biến đổi biểu thức đại số quan hệ: Biến đổi thứ tự thực hiện các phép toán của biểu thức đại số quan hệ sao cho các phép toán một ngôi được thực hiện trước các phép toán hai ngôi, do các phép toán chiếu, chọn thì có chi phí nhỏ hơn so với các phép kết nối, tích đề các.
- Một số phép biến đổi tương đương:
 - o Tách điều kiện trong phép chọn:

$$\delta_{C_1 \wedge C_2}(E) = \delta_{C_1}(\delta_{C_2}(E))$$
 - o Tính chất giao hoán của phép chọn:

$$\delta_{C_1}(\delta_{C_2}(E)) = \delta_{C_2}(\delta_{C_1}(E))$$
 - o Dãy các phép chiếu liên tiếp:

$$\Pi_{L_1}(\Pi_{L_2}(\dots(\Pi_{L_n}(E))\dots)) = \Pi_{L_1}(E), L_1 \subseteq L_2 \dots \subseteq L_n$$
 - o Phép chọn – phép tích đề các và phép kết nối θ :
 - $\delta_\theta(E_1 \times E_2) = E_1 \bowtie_\theta E_2$
 - $\delta_{\theta_1}(E_1 \bowtie_{\theta_2} E_2) = E_1 \bowtie_{\theta_1 \wedge \theta_2} E_2$
 - o Tính chất giao hoán của phép kết nối θ :

$$E_1 \bowtie_\theta E_2 = E_2 \bowtie_\theta E_1$$
 - o Tính chất giao hoán của phép kết nối tự nhiên:

$$E_1 * E_2 = E_2 * E_1$$
 - o Tính chất kết hợp của phép kết nối tự nhiên:

$$(E_1 * E_2) * E_3 = E_1 * (E_2 * E_3)$$
 - o Tính chất kết hợp của phép kết nối θ :
 - $(E_1 \bowtie_{\theta_1} E_2) \bowtie_{\theta_2 \wedge \theta_3} E_3 = E_1 \bowtie_{\theta_1 \wedge \theta_3} (E_2 \bowtie_{\theta_2} E_3)$, nếu các thuộc tính trong θ_2 chỉ thuộc E_2 và E_3 .
 - o Phép chọn và phép kết nối θ :
 - $\delta_{\theta_0}(E_1 \bowtie_\theta E_2) = (\delta_{\theta_0}(E_1)) \bowtie_\theta E_2$, nếu các thuộc tính trong θ_0 chỉ thuộc E_1 .
 - $\delta_{\theta_1 \wedge \theta_2}(E_1 \bowtie_\theta E_2) = (\delta_{\theta_1}(E_1)) \bowtie_\theta (\delta_{\theta_2}(E_2))$, nếu các thuộc tính trong θ_1 chỉ thuộc E_1 , các thuộc tính trong θ_2 chỉ thuộc E_2 .
 - o Phép chiếu và phép kết nối θ :
 - $\Pi_{L_1 \cup L_2}(E_1 \bowtie_\theta E_2) = (\Pi_{L_1}(E_1)) \bowtie_\theta (\Pi_{L_2}(E_2))$, nếu:
 - ✓ Các thuộc tính trong L_1 chỉ thuộc E_1 .
 - ✓ Các thuộc tính trong L_2 chỉ thuộc E_2 .

- ✓ Các thuộc tính trong θ đều có trong L_1 và L_2 .
- $\prod_{L_1 \cup L_2}(E_1 \bowtie_{\theta} E_2) = \prod_{L_1 \cup L_2} \left(\left(\prod_{L_1 \cup L_3}(E_1) \right) \bowtie_{\theta} \left(\prod_{L_2 \cup L_4}(E_2) \right) \right)$
 - ✓ Các thuộc tính trong L_1 chỉ thuộc E_1 .
 - ✓ Các thuộc tính trong L_2 chỉ thuộc E_2 .
 - ✓ Các thuộc tính trong L_3 chỉ có trong E_1 , liên quan đến θ nhưng không thuộc $L_1 \cup L_2$.
 - ✓ Các thuộc tính trong L_4 chỉ có trong E_2 , liên quan đến θ nhưng không thuộc $L_1 \cup L_2$.
- o Phép hợp – Phép giao và phép trừ
 - Tính chất giao hoán:
 - ✓ $E_1 \cup E_2 = E_2 \cup E_1$
 - ✓ $E_1 \cap E_2 = E_2 \cap E_1$
 - Tính chất kết hợp:
 - ✓ $(E_1 \cup E_2) \cup E_3 = E_1 \cup (E_2 \cup E_3)$
 - ✓ $(E_1 \cap E_2) \cap E_3 = E_1 \cap (E_2 \cap E_3)$
 - Tính chất phân phối của phép chọn qua phép hợp – giao – trừ:
 - ✓ $\delta_P(E_1 - E_2) = \delta_P(E_1) - \delta_P(E_2)$
 - ✓ $\delta_P(E_1 - E_2) = \delta_P(E_1) - E_2$
 - Tính chất phân phối của phép chiếu qua phép hợp:

$$\prod_L(E_1 \cup E_2) = (\prod_L(E_1)) \cup (\prod_L(E_2))$$

2.3. Thuật toán tối ưu hóa cây đại số quan hệ

2.3.1. Thuật toán

- Bước 1: Biểu diễn truy vấn dưới dạng cây với lá là các quan hệ, đỉnh trong là các phép toán đại số quan hệ.
- Bước 2: Áp dụng các phép biến đổi tương đương đẩy các phép toán một ngôi xuống dưới các phép toán hai ngôi.
- Bước 3: Thêm vào các phép chiếu để giảm bớt kích thước của các quan hệ.

2.3.2. Ví dụ

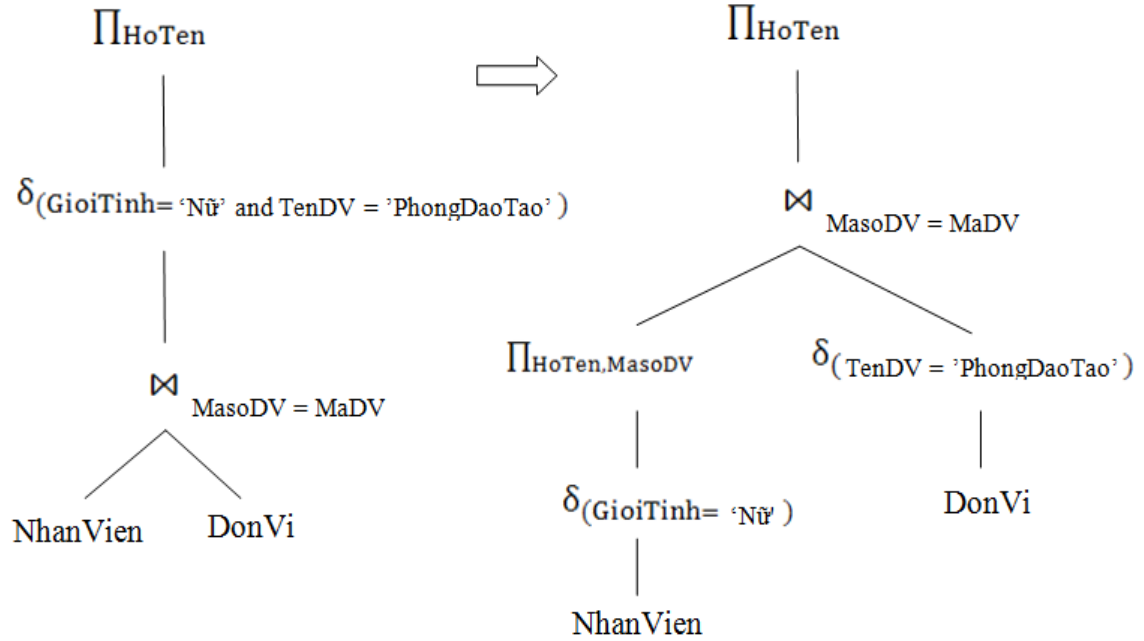
Xét hai bảng nhân viên và đơn vị sau:

- NhanVien (MaNV, MasoDV, HoTen, NgaySinh, GioiTinh, Luong)
- DonVi(MaDV, TenDV)

Truy vấn đưa ra họ tên của các nhân viên nữ ở đơn vị có tên là 'PhongDaoTao':

$$\Pi_{\text{HoTen}} (\delta_{\text{GioiTinh} = \text{'Nữ'} \text{ and } \text{TenDV} = \text{'PhongDaoTao'}} (\text{NhanVien} \bowtie \text{DonVi}))$$

$\text{MasodV} = \text{MaDV}$



2.4. Câu hỏi ôn tập chương 2

- Trình bày các bước trong quá trình xử lý truy vấn.
- Trình bày các phép biến đổi tương đương.
- Trình bày thuật toán tối ưu hóa cây đại số quan hệ.
- Cho hai bảng nhân viên và đơn vị sau:
 - NhanVien (MaNV, MasoDV, HoTen, NgaySinh, GioiTinh, Luong)
 - DonVi(MaDV, TenDV)

Hãy viết biểu thức đại số quan hệ thực hiện truy vấn sau: Đưa ra họ tên của các nhân viên nữ ở đơn vị có tên là 'PhongDaoTao'. Tối ưu hóa truy vấn trên sử dụng phương pháp biến đổi đại số quan hệ.

- Cho các quan hệ:
 - DonVi(MDV, TenDV, MaNQL, DiaDiem)
 - DuAn (MaDA, TenDA, KinhPhi, MaDV)

Truy vấn nào dưới đây đã tối ưu?

$$Q1. \Pi_{TenDA} \left(\delta_{TenDV='XD2'}(DonVi) \bowtie_{MDV=MaDV} \Pi_{TenDA, MaDV} (\delta_{KinhPhi > 9000}(DuAn)) \right)$$

$$Q2. \Pi_{TenDA} \left(\delta_{TenDV='XD2' \wedge KinhPhi > 9000}(DonVi \bowtie_{MDV=MaDV} DuAn) \right)$$

$$Q3. \Pi_{TenDA} \left(\Pi_{MDV} (\delta_{TenDV='XD2'}(DonVi)) \bowtie_{MDV=MaDV} \Pi_{TenDA, MaDV} (\delta_{KinhPhi > 9000}(DuAn)) \right)$$

$$Q4. \Pi_{TenDA} (\delta_{TenDV='XD2'}(DonVi) \bowtie_{MDV=MaDV} \delta_{KinhPhi > 9000}(DuAn))$$

CHƯƠNG 3: GIAO DỊCH TRONG CƠ SỞ DỮ LIỆU

3.1. Giới thiệu

Thuật ngữ giao dịch (Transaction) đề cập tới một tập hợp các lệnh tạo thành một đơn vị làm việc logic, hoặc nó được thực hiện một cách đầy đủ hoặc bị hủy bỏ hoàn toàn.

Ví dụ 3.1: Chuyển tiền từ tài khoản A sang tài khoản B là một giao dịch:

- Kiểm tra tiền trong tài khoản A (có X không?).
- $A = A - X$.
- $B = B + X$.

Thông thường giao dịch được tạo ra bởi các chương trình người dùng được viết trong ngôn ngữ xử lý dữ liệu bậc cao hoặc ngôn ngữ lập trình (vd: SQL, C++, Java).

Mục dữ liệu (data item) là đơn vị dữ liệu trong CSDL. Bản chất, kích thước của các mục dữ liệu do nhà thiết kế chọn. Chúng được lựa chọn sao cho việc truy xuất dữ liệu đạt hiệu quả nhất. Ví dụ trong mô hình dữ liệu quan hệ, ta có thể chọn mục dữ liệu là: các quan hệ, các bộ, hay các thành phần của bộ. Kích thước của các mục dữ liệu mà hệ thống sử dụng gọi là độ mịn của hệ thống (granularity).

Bộ lập lịch (Scheduler) là một thành phần của hệ thống CSDL chịu trách nhiệm tạo ra một lịch biểu (thứ tự thực hiện) các thao tác trong các giao dịch.

3.2. Các tính chất của giao dịch

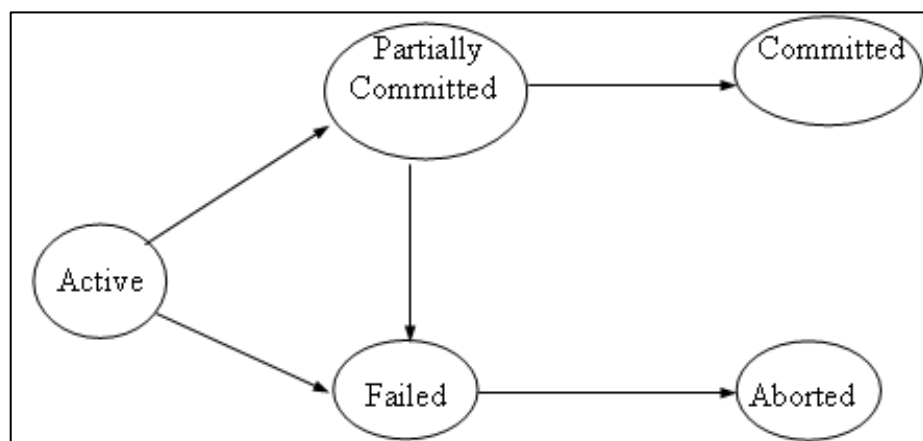
Để đảm bảo tính toàn vẹn của dữ liệu, hệ cơ sở dữ liệu cần duy trì các tính chất sau của giao dịch:

- Tính nguyên tử (Atomicity): Hoặc toàn bộ các thao tác của giao dịch được phản ánh đúng đắn trong cơ sở dữ liệu hoặc không có gì cả.
- Tính nhất quán (Consistency): Sự thực hiện của một giao dịch phải bảo đảm tính nhất quán của cơ sở dữ liệu.
- Tính cô lập (Isolation): Nhiều giao dịch có thể thực thi đồng thời nhưng mỗi giao dịch không cần biết đến các giao dịch khác đang thực hiện đồng thời. Các kết quả trung gian của giao dịch phải được che giấu trước các giao dịch khác.
- Tính lâu bền (Durability): Sau khi giao dịch hoàn thành, các thay đổi đã được tạo ra đối với cơ sở dữ liệu vẫn còn ngay cả khi xảy ra sự cố hệ thống.

3.3. Các trạng thái của giao dịch

Khi giao dịch bắt đầu thực hiện nó vào trạng thái active và có thể thực hiện thao tác đọc, ghi các mục dữ liệu. Khi kết thúc thao tác cuối cùng giao dịch đạt tới trạng thái partially committed. Tại thời điểm này, giao dịch đã hoàn thành sự thực hiện của nó, nhưng nó vẫn có thể bị bỏ dở do các thay đổi có thể vẫn lưu tạm thời trong bộ nhớ chính và như thế một sự cố phần cứng vẫn có thể ngăn cản sự hoàn tất của giao dịch. Nếu không có sự cố, tất cả các giao dịch đều hoàn tất thành công (tất cả các thao tác trong giao dịch đều được phản ánh trong cơ sở dữ liệu), khi đó giao dịch được “bàn giao” (committed). Tuy nhiên, trong thực tế một giao dịch có thể không hoàn tất sự thực hiện của nó. Giao dịch như vậy được gọi là bị bỏ dở (abort). Để đảm bảo được tính nguyên tử, một giao dịch bị bỏ dở không được phép làm ảnh hưởng tới trạng thái của cơ sở dữ liệu. Như vậy, tất cả thay đổi được tạo ra trước đó phải bị huỷ bỏ (rollback).

- Active: Giao dịch sẽ đi vào trạng thái Active khi bắt đầu và trạng thái này sẽ được duy trì trong khi giao dịch đang thực hiện.
- Partially Committed: Sau khi thao tác cuối cùng trong giao dịch được thực hiện.
- Failed: Sau khi phát hiện rằng sự thực hiện không thể tiếp tục được nữa.
- Aborted: Sau khi rollback và cơ sở dữ liệu đã phục hồi lại trạng thái của nó trước khi khởi động giao dịch.
- Committed: Giao dịch hoàn tất thành công (tất cả các thao tác được phản ánh đầy đủ trong cơ sở dữ liệu).
- Sơ đồ trạng thái tương ứng với một giao dịch như sau:



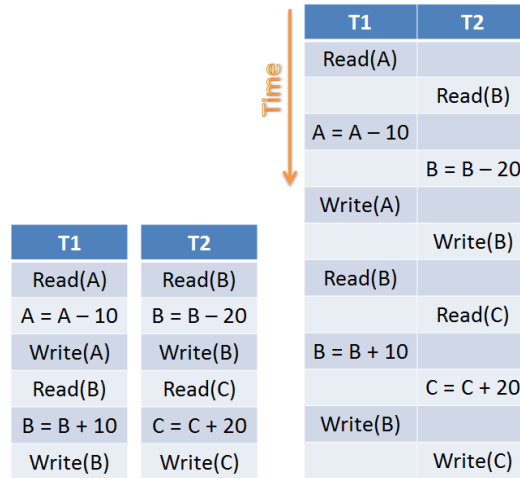
Hình 3.1: Các trạng thái của giao dịch

3.4. Lịch biểu

Lịch biểu: Là một dãy (có thứ tự) các thao tác của một tập các giao dịch mà trong đó thứ tự của các thao tác trong mỗi giao dịch được bảo toàn.

Ví dụ 3.2: Xét hai giao dịch chuyển tiền:

- 10\$ từ tài khoản A sang tài khoản B
- 20\$ từ tài khoản B sang tài khoản C.

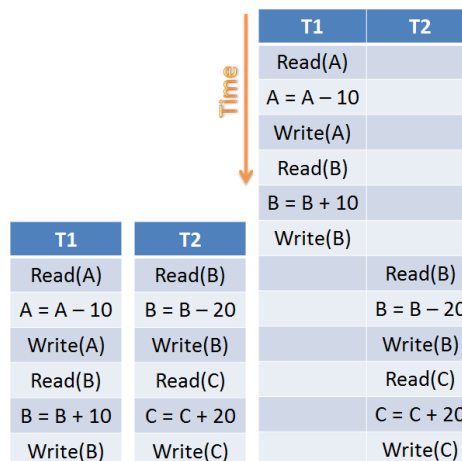


Hình 3.2: Lịch biểu

3.5. Tính khả tuần tự của lịch biểu

Lịch biểu tuần tự (Serial Schedule): Là lịch biểu mà trong mỗi giao dịch các thao tác được thực hiện kế tiếp nhau, không có thao tác của giao dịch khác xen vào (Thực hiện lần lượt, hết giao dịch này đến giao dịch khác). Không thể có đựng độ trong một lịch biểu tuần tự.

- Với một tập S gồm n giao dịch $\{T1, T2, \dots, Tn\}$ sẽ có $n!$ lịch biểu tuần tự. Hình 3.3 là một ví dụ về lịch biểu tuần tự của hai giao dịch T1 và T2.

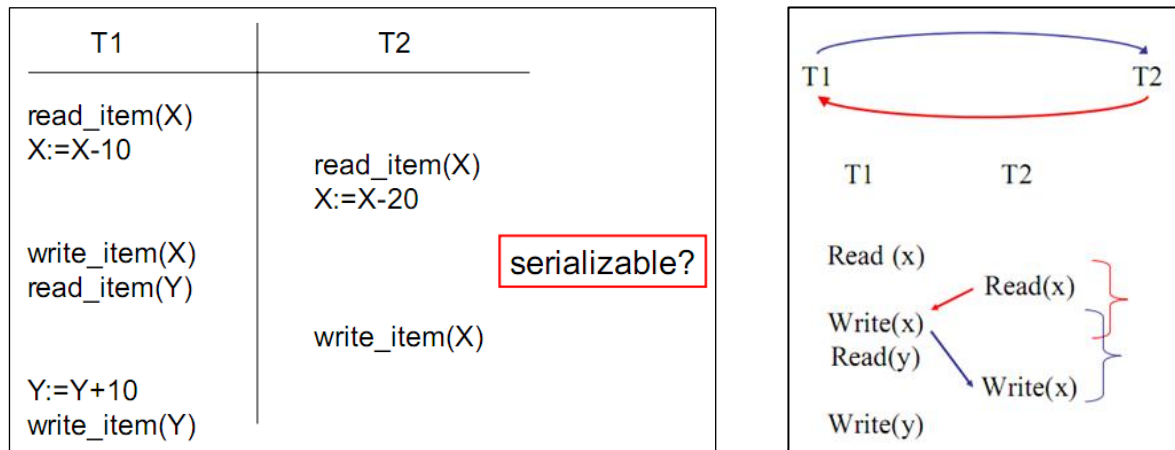


Hình 3.3: Lịch biểu tuần tự

- Lịch biểu gọi là khả tuần tự (Serializable): nếu nó tương đương với một lịch biểu tuần tự. Tương đương theo nghĩa cho ra cùng một trạng thái CSDL sau khi kết thúc việc thực hiện lịch biểu).
- Lịch biểu bất khả tuần tự nếu nó không tương đương với một lịch biểu tuần tự.

3.6. Thuật toán kiểm tra tính khả tuần tự của lịch biểu

- Input: S {T1, T2, ..., Tn}
- Output: S khả tuần tự hay k?
- Thuật toán:
 - Xây dựng đồ thị phụ thuộc G:
 - Nút: là các giao dịch.
 - Cung $T_i \rightarrow T_j$: $\begin{cases} T_j \text{ read}(X) \text{ sau } T_i \text{ write}(X) \\ T_j \text{ write}(X) \text{ sau } T_i \text{ read}(X) \\ T_j \text{ write}(X) \text{ sau } T_i \text{ write}(X) \end{cases}$
 - Nếu G không có chu trình thì lịch biểu S là khả tuần tự, ngược lại lịch biểu S là bất khả tuần tự.
- Ví dụ: Lịch biểu S của hai giao dịch T1, T2 và đồ thị phụ thuộc G của S:

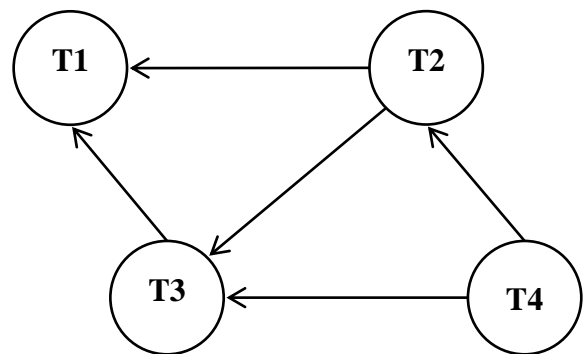


Hình 3.4: Ví dụ kiểm tra tính khả tuần tự của lịch biểu

Trong ví dụ này ta thấy đồ thị G có chu trình, do vậy lịch biểu S là bất khả tuần tự.

3.7. Câu hỏi ôn tập chương 3

- Trình bày khái niệm về giao dịch. Cho ví dụ.
- Trình bày bốn tính chất của giao dịch.
- Trình bày khái niệm lịch biểu. Cho ví dụ.
- Trình bày các khái niệm lịch biểu tuần tự, lịch biểu bất tuần tự và lịch biểu khả tuần tự. Cho ví dụ.
- Trình bày thuật toán kiểm tra tính khả tuần tự của lịch biểu.
- Lịch biểu nào dưới đây là khả tuần tự?
 - $R2(Y), R1(X), R1(Y), R3(X), W3(X), W2(Y), W1(X)$
 - $R2(Y), R1(X), R1(Y), R3(Z), W3(Z), W2(Y), W1(X)$
 - $R2(Y), R1(X), R1(Y), R3(X), W2(Y), W1(X), W3(X)$
 - $R1(X), R1(Y), R3(X), R2(Y), W2(Y), W1(X), W3(X)$
- Cho đồ thị phụ thuộc của lịch biểu S. Lịch biểu nào dưới đây là lịch biểu tuần tự tương đương với S:
 - $T1 \rightarrow T2 \rightarrow T3 \rightarrow T4$
 - $T2 \rightarrow T1 \rightarrow T3 \rightarrow T4$
 - $T4 \rightarrow T2 \rightarrow T3 \rightarrow T1$
 - $T4 \rightarrow T3 \rightarrow T1 \rightarrow T2$



CHƯƠNG 4: ĐIỀU KHIỂN ĐỒNG THỜI VÀ KHÔI PHỤC HỆ THỐNG

4.1. Các giao thức dựa vào khóa

4.1.1. Mô hình khóa nhị phân

- Một khóa nhị phân (Binary Lock) có hai trạng thái (giá trị): Locked và Unlocked (1 hoặc 0).
- Lock(X): Khóa trên mục dữ liệu X. Lock(X) = 1 nghĩa là mục dữ liệu X đã bị khóa bởi một giao dịch; Các giao dịch khác nếu có yêu cầu truy cập mục dữ liệu này sẽ phải đợi đến khi Lock(X) = 0.
- Hai thao tác lock_item, unlock_item được sử dụng với khóa nhị phân.
 - **lock_item(X);**
B: if LOCK(X)=0 (*item is unlocked*)
 then LOCK(X) \leftarrow 1 (*lock the item*)
 else begin
 wait (until LOCK(X)=0 and
 the lock manager wakes up the transaction);
 go to B;
 end;
 - **unlock_item(X);**
LOCK(X) \leftarrow 0 (*unlock the item*)
if any transactions are waiting
 then wakeup one of the waiting transactions;
- Trong mô hình khóa nhị phân, mọi giao dịch phải tuân theo các luật sau:
 - Một giao dịch phải thực hiện lock_item(X) trước các thao tác read_item(X), write_item(X).
 - Một giao dịch phải thực hiện unlock_item(X) sau khi đã hoàn tất các thao tác read_item(X), write_item(X).
 - Một giao dịch không được thực hiện lock_item(X) nếu nó đang giữ khóa trên X (Hold the lock).
 - Một giao dịch không được thực hiện unlock_item(X) nếu nó không giữ khóa trên X.

4.1.2. Mô hình khóa đọc – ghi (chia sẻ – độc quyền)

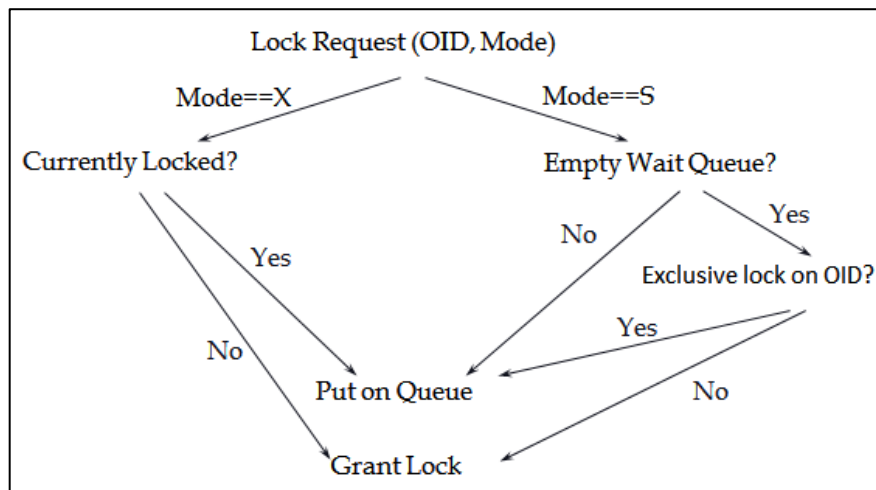
- Cho phép nhiều hơn một giao dịch có thể truy cập trên cùng một mục dữ liệu X nếu các giao dịch đó chỉ đọc dữ liệu.

- Nếu một giao dịch muốn thực hiện ghi trên mục dữ liệu X, nó phải có độc quyền truy cập trên X.
- Một khóa trên mục dữ liệu X: LOCK(X) có 3 trạng thái: “read – locked” (“share – locked”), “write – locked” (“exclusive – locked”) và “unlocked”.
- Các thao tác: read_lock, write_lock, unlock được sử dụng với khóa đọc – ghi.
- Trong mô hình khóa đọc – ghi, một bản ghi trong bảng khóa (Lock Table) gồm 4 trường:

Data item name	LOCK	No_of_reads	Locking Transactions
----------------	------	-------------	----------------------

- Data item name: Tên mục dữ liệu.
- LOCK: read_locked hoặc write_locked.
- Nếu LOCK là write_locked: Locking transactions là một giao dịch duy nhất đang giữ khóa ghi trên mục dữ liệu.
- Nếu LOCK là read_locked: Locking transactions là một danh sách các giao dịch giữ khóa đọc trên mục dữ liệu. No_of_reads: Số lượng các giao dịch đang giữ khóa đọc trên X.
- Trong mô hình khóa đọc – ghi, mọi giao dịch phải tuân theo các luật sau:
 - Một giao dịch phải thực hiện read_lock(X) hoặc write_lock(X) trước khi read_item(X).
 - Một giao dịch phải thực hiện write_lock(X) trước khi write_item(X).
 - Một giao dịch phải thực hiện unlock_item(X) sau khi đã hoàn tất các thao tác read_item(X), write_item(X).
 - Một giao dịch không được thực hiện read_lock(X) nếu nó đang giữ khóa đọc, hoặc ghi trên X (Hold the lock).¹
 - Một giao dịch không được thực hiện write_lock(X) nếu nó đang giữ một khóa đọc trên X.¹
 - Một giao dịch không được thực hiện unlock_item(X) nếu nó không giữ khóa trên X.
- Hình 4.1 dưới đây chỉ ra quá trình xử lý yêu cầu khóa trong mô hình khóa chia sẻ - độc quyền.

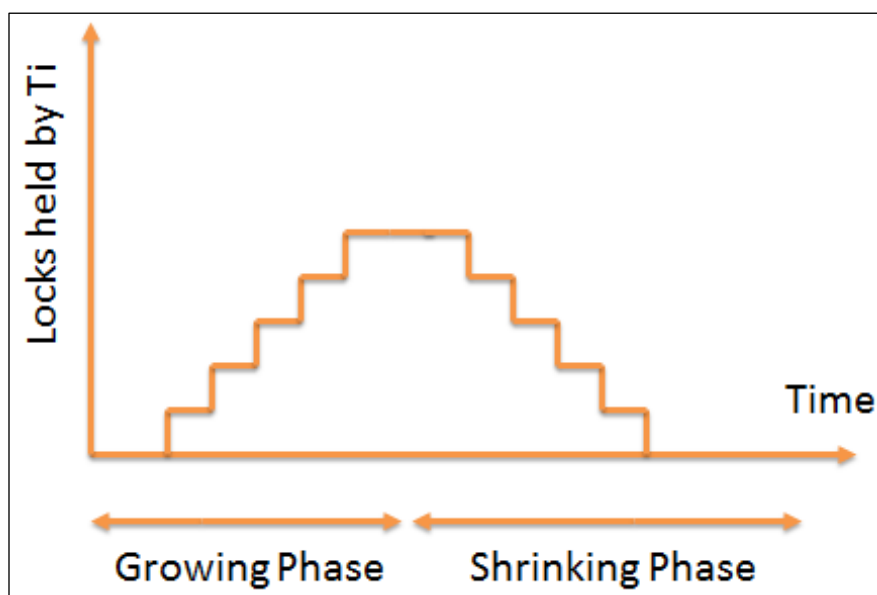
¹ Các luật này có thể được nói lỏng.



Hình 4.1: Xử lý yêu cầu khóa trong mô hình khóa chia sẻ – độc quyền

4.1.3. Giao thức khóa 2 pha

- Giao thức khóa hai pha (Two – phase locking protocol – 2PLP) là một giao thức đảm bảo tính khả tuần tự. Một giao dịch T được gọi là tuân theo giao thức khóa hai pha nếu trong T tất cả các yêu cầu khóa (read_lock, write_lock) được đưa ra trước yêu cầu mở khóa (unlock).
- Với giao thức này, các giao dịch được chia thành hai pha:
 - Pha tăng trưởng: (expanding or growing phase): Trong pha này giao dịch được phép khóa (read_lock, write_lock) trên các mục dữ liệu nhưng không được phép mở khóa (unlock) bất cứ một mục dữ liệu nào.
 - Pha thu lại (shrinking phase): Trong pha này, giao dịch được phép mở khóa trên các mục dữ liệu nhưng không được phép khóa thêm bất kỳ một mục dữ liệu nào.



Hình 4.2: Giao thức khóa 2 pha

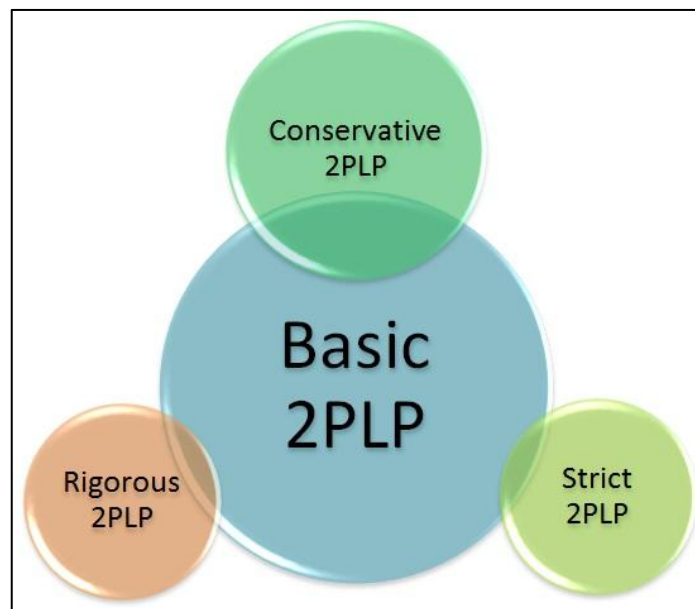
- Ví dụ:

T1	T2	T1'	T2'
read_lock(Y)	read_lock(X)	read_lock(Y)	read_lock(X)
read_item(Y)	read_item(X)	read_item(Y)	read_item(X)
unlock(Y)	unlock(X)	write_lock(X)	write_lock(Y)
write_lock(X)	write_lock(Y)	unlock(Y)	unlock(X)
read_item(X)	read_item(Y)	read_item(X)	read_item(Y)
$X = X + Y$	$Y = X + Y$	$X = X + Y$	$Y = X + Y$
write_item(X)	write_item(Y)	write_item(X)	write_item(Y)
unlock(X)	unlock(Y)	unlock(X)	unlock(Y)

Hình 4.3: Giao dịch T1 và T2 không tuân theo 2PLP

Giao dịch T1' và T2' tuân theo 2PLP

- Các biến thể của giao thức khóa 2 pha:
 - Conservative 2PLP: Yêu cầu giao dịch phải khóa tất cả các mục dữ liệu cần truy cập trước khi thực thi giao dịch, bằng cách đưa ra các tập: read-set và write-set. (Giao thức này có khả năng ngăn ngừa deadlock – trình bày trong các slide kế tiếp)
 - Strict 2PLP: Yêu cầu giao dịch không được giải phóng bất kỳ khóa độc quyền nào (exclusive lock) cho đến khi commit hoặc abort.
 - Rigorous 2PLP: Yêu cầu giao dịch không được giải phóng bất kỳ khóa nào cho đến khi commit hoặc abort.



Hình 4.4: Giao thức khóa 2 pha và các biến thể

4.1.4. Deadlock

- Deadlock là hiện tượng xảy ra khi mỗi giao dịch T trong một tập S (gồm hai hay nhiều giao dịch) đợi một số mục bị khóa bởi các giao dịch khác trong S.
- Ví dụ: Hai giao dịch đang trong trạng thái Deadlock:

T1	T2
read_lock(Y)	
read_item(Y)	
	read_lock(X)
	read_item(X)
write_lock(X)	
	write_lock(Y)

Trong ví dụ này giao dịch T1 đang đợi khóa ghi trên mục dữ liệu X nhưng X lại đang bị khóa bởi giao dịch T2; còn giao dịch T2 đang đợi khóa ghi trên mục dữ liệu Y nhưng Y lại đang bị khóa bởi giao dịch T1. Hai giao dịch này đợi lẫn nhau và ta gọi hiện tượng này là Deadlock.

4.1.4.1. Các chiến lược ngăn cản Deadlock

Một chiến lược được sử dụng để giải quyết vấn đề Deadlock là sử dụng các giao thức có khả năng ngăn cản Deadlock (Deadlock prevention protocol). Tuy nhiên các giao thức này ít được sử dụng trong thực tế.

Conservative two – phase locking là giao thức được sử dụng để ngăn cản Deadlock. Giao thức này yêu cầu giao dịch phải khóa tất cả các mục dữ liệu cần truy cập trước khi thực thi giao dịch. Nếu một yêu cầu khóa nào đó không được đáp ứng thì tất cả yêu cầu khóa còn lại sẽ bị bỏ qua, giao dịch sẽ phải đợi và thử lại ở lần sau. Tuy nhiên, giao thức này lại giới hạn việc thực thi đồng thời các giao dịch.

Một số chiến lược khác để ngăn cản Deadlock là quyết định làm gì với các giao dịch liên quan trong tình huống Deadlock: có thể là chặn lại và bắt phải chờ đợi, hoặc bỏ qua. Những chiến lược này sử dụng khái niệm nhãn thời gian của giao dịch (Transaction Timestamp). Nhãn thời gian của một giao dịch T (Ký hiệu $TS(T)$) là một định danh duy nhất được gán cho T khi T bắt đầu. Nếu giao dịch T1 bắt đầu trước giao dịch T2 thì $TS(T1) < TS(T2)$. Hai chiến lược ngăn ngừa Deadlock sử dụng nhãn thời gian là Wait – Die và Wound – Wait. Giả sử giao dịch Ti cố gắng khóa

mục dữ liệu X nhưng không thể vì X đang bị khóa bởi Tj với một khóa xung đột (Conflicting lock)². Khi đó:

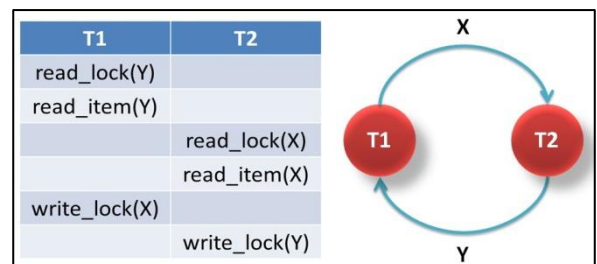
- Với Wait – Die: Nếu $TS(T_i) < TS(T_j)$ thì Ti được phép đợi (*wait*); ngược lại bỏ qua Ti (Ti *die*) và khởi động lại Ti sau với cùng nhãn thời gian.
- Với Wound – Wait: Nếu $TS(T_i) < TS(T_j)$ thì bỏ qua Tj (Ti *wound* Tj) và khởi động lại Tj sau với cùng nhãn thời gian; ngược lại Ti phải đợi (*wait*).

Một nhóm các giao thức ngăn cản Deadlock mà không dùng tới nhãn thời gian là no waiting (NW) và cautious waiting (CW). Với NW, nếu một giao dịch không nhận được khóa nó sẽ lập tức bị bỏ qua và khởi động lại sau một khoảng thời gian trễ mà không cần kiểm tra Deadlock có thật sự xảy ra hay không, như vậy NW có thể bỏ qua các giao dịch không cần thiết (các giao dịch không gây ra Deadlock). Để giảm số lượng các giao dịch bị bỏ qua không cần thiết CW được đề xuất. Giả sử Ti cố gắng khóa mục dữ liệu X nhưng không được bởi X đang bị khóa bởi Tj với một khóa xung đột. Với Cautious Waiting: Nếu Tj không bị chặn (Không đợi khóa một mục dữ liệu) thì Ti sẽ được phép đợi; ngược lại Ti sẽ bị bỏ qua.

4.1.4.2. Chiến lược phát hiện Deadlock

Chiến lược này tỏ ra hiệu quả khi các giao dịch là ngắn và mỗi giao dịch chỉ yêu cầu khóa một vài mục dữ liệu. Ý tưởng của chiến lược này là xây dựng và duy trì một đồ thị đợi (Wait – for – graph):

- Một nút tương ứng với một active transaction.
- Ti cố gắng khóa mục X nhưng không thể do X đang bị khóa bởi Tj với một khóa xung đột: Tạo một cung từ Ti đến Tj; Khi Tj loại bỏ khóa xung đột thì xóa cung tương ứng đi.
- Đồ thị có chu trình \rightarrow Deadlock xảy ra. Khi đó một số giao dịch gây ra Deadlock sẽ bị loại bỏ.



Hình 4.5: Wait – for – graph

² Hai khóa được gọi là xung đột nếu tồn tại một khóa là khóa ghi (Write lock).

4.1.4.3. Starvation

- Starvation: Là hiện tượng một giao dịch không được thực hiện trong một thời gian dài trong khi các giao dịch khác vẫn thực hiện bình thường.
- Starvation có thể xảy ra do:
 - Chiến lược chờ đợi khóa không công bằng. Giải pháp cho vấn đề này là đưa ra một chiến lược chờ đợi công bằng như sử dụng hàng đợi “đến trước phục vụ trước” (first – come – first – served queue), hoặc tăng độ ưu tiên của các giao dịch chờ đợi lâu.
 - Thuật toán chọn các giao dịch để bỏ qua (victim selection) không tốt: Chọn lặp lại một giao dịch để bỏ qua.

4.2. Giao thức thứ tự nhãn thời gian (Timestamp – Ordering protocol)

4.2.1 Nhãn thời gian (Timestamp)

Ta kết hợp với mỗi giao dịch T_i trong hệ thống với một nhãn thời gian cố định. Nhãn thời gian của giao dịch T_i , kí hiệu $TS(T_i)$. Nhãn thời gian này được gán bởi hệ CSDL khi giao dịch T_i bắt đầu thực hiện. Giả sử một giao dịch T_i đã được gán nhãn thời gian $TS(T_i)$ và một giao dịch mới T_j đi vào hệ thống, khi đó $TS(T_i) < TS(T_j)$. Nhãn thời gian có thể được tạo ra theo một vài cách:

- Sử dụng giá trị của đồng hồ hệ thống như nhãn thời gian: Một nhãn thời gian của một giao dịch bằng giá trị của đồng hồ khi giao dịch đi vào hệ thống.
- Sử dụng bộ đếm logic: bộ đếm được tăng lên mỗi khi một nhãn thời gian đã được gán, nhãn thời gian của một giao dịch bằng với giá trị của bộ đếm khi giao dịch đi vào hệ thống.

Nhãn thời gian của các giao dịch xác định thứ tự khả tuần tự. Như vậy, nếu $TS(T_i) < TS(T_j)$, hệ thống phải đảm bảo rằng lịch trình được sinh ra là tương đương với một lịch trình tuần tự trong đó T_i xuất hiện trước T_j . Giao thức duy trì trên mỗi mục dữ liệu hai giá trị TS:

- Read_TS(X): read timestamp của X; là TS lớn nhất trong số các TS của các giao dịch đã read_lock(X) thành công.
- Write_TS(X): read timestamp của X; là TS lớn nhất trong số các TS của các giao dịch đã write_lock(X) thành công.

4.2.2. Giao thức thứ tự nhãn thời gian (Timestamp – Ordering Protocol)

- Giao dịch T muốn thực hiện write_item(X):
 1. Nếu Read_TS(X) > TS(T) hoặc Write_TS(X) > TS(T) thì bỏ qua và rollback.
 2. Nếu điều kiện 1 không thỏa mãn, cho phép T write_item(X) và gán lại: Write_TS(X) = TS(T).

- Giao dịch T muốn thực hiện $read_item(X)$:
 1. Nếu $Write_TS(X) > TS(T)$ thì bỏ qua và rollback.
 2. Nếu điều kiện 1 không thỏa mãn, cho phép T $read_item(X)$ và gán lại:
 $Read_TS(X) = \text{Max}\{Read_TS(X), TS(T)\}.$

4.3. Phục hồi hệ thống dựa vào nhật ký giao dịch (Log-based)

Một cấu trúc thường được dùng để ghi lại những thay đổi trên cơ sở dữ liệu là sổ ghi lộ trình (log). Log là một dãy các mẫu tin lộ trình (log records). Một thao tác cập nhật trên cơ sở dữ liệu sẽ được ghi nhận bằng một log record. Một log record kiểu mẫu chứa các trường sau:

- Định danh giao dịch (transaction identifier): định danh duy nhất của giao dịch thực hiện hoạt động write.
- Định danh hạng mục dữ liệu (Data-item identifier): định danh duy nhất của hạng mục dữ liệu được viết (thường là vị trí của hạng mục dữ liệu trên đĩa).
- Giá trị cũ (Old value): giá trị của hạng mục dữ liệu trước thao tác write.
- Giá trị mới (New value): giá trị hạng mục dữ liệu sẽ có sau khi write.

Có một vài log record đặc biệt mang các ý nghĩa riêng. Bảng sau đây chỉ ra một số loại log record và ý nghĩa của chúng:

LOẠI LOG RECORD	Ý NGHĨA
< Ti start >	Giao dịch Ti đã khởi động.
< Ti, Xj, V1, V2 >	Giao dịch Ti đã thực hiện thao tác ghi trên hạng mục dữ liệu Xj, Xj có giá trị V1 trước khi ghi và nhận giá trị V2 sau khi ghi.
< Ti commit >	Giao dịch Ti đã bàn giao.
< Ti abort >	Giao dịch Ti đã hủy bỏ.

Giao dịch muốn thực hiện một thao tác ghi, trước tiên phải tạo ra một log record cho thao tác ghi đó (trong log file), trước khi giao dịch thay đổi cơ sở dữ liệu. Như vậy, hệ thống có cơ sở để hủy bỏ (undo) một thay đổi đã được làm trên cơ sở dữ liệu bằng cách sử dụng trường Old-value trong log record. Log phải được lưu trong những thiết bị lưu trữ bền. Mỗi một log record mới ngằm định sẽ được thêm vào cuối tập tin log.

4.3.1 Cập nhật trì hoãn cơ sở dữ liệu (Deferred Database Modification)

- Tư tưởng chính của kỹ thuật cập nhật trì hoãn là trì hoãn việc cập nhật thực sự vào csdl cho đến khi giao dịch đạt tới trạng thái commit.

- Một giao dịch không thể thay đổi csdl trên đĩa cho đến khi nó đạt tới trạng thái commit.
- Một giao dịch không thể đạt tới trạng thái commit cho đến khi tất cả các thao tác cập nhật của nó được ghi lại trên file log (WAL: write – ahead logging).
- Các thao tác CSDL sẽ không được cập nhật cho đến khi giao dịch hoàn tất nên không cần các thao tác UNDO. Có thể phải REDO trong trường hợp hệ thống lỗi sau khi giao dịch đã hoàn tất nhưng trước khi tất cả các thay đổi được ghi vào cơ sở dữ liệu. Giao dịch Ti cần được REDO nếu trong file log chứa cả hai bản ghi: <Ti start> và <Ti commit>. Cũng vì vậy mà kỹ thuật phục hồi này còn được gọi là NO – UNDO/ REDO recovery algorithm. Bảng sau chỉ ra các loại bản ghi được tạo ra trong file log:

Thao tác của giao dịch	Bản ghi trong file Log
Giao dịch Ti bắt đầu	Ti, Start
Ti Write(X)	Ti, X, New_Value
Giao dịch Ti hoàn tất	Ti, Commit

- Ví dụ: Xét ba mục dữ liệu A, B, C với các giá trị ban đầu lần lượt là 1000, 2000, 700 và hai giao dịch T1, T2:

T0:	Read(A)	Bản ghi trong file log
	A = A – 50	
	Write(A)	
	Read(B)	
	B = B + 50	
	Write(B)	
T1:	Read(C)	
	C = C – 100	
	Write(C)	
		T0, Start
		T0, A, 950
		T0, B, 2050
		T0, Commit
		T1, Start
		T1, C, 600
		T1, Commit

Sau đây là một số tình huống xảy ra và hành động của hệ thống phục hồi sử dụng kỹ thuật cập nhật tri hoãn cơ sở dữ liệu:

Tình huống	Hệ thống phục hồi	Giá trị các mục dữ liệu		
		A	B	C
Lỗi sau khi T0 vừa thực hiện thao tác write B.	Không làm gì.	1000	2000	700
Lỗi sau khi T1 thực hiện Write(C).	REDO(T0)	950	2050	700

4.3.2. Cập nhật tức thời (Immediate Database Modification)

Kỹ thuật cập nhật tức thời cho phép các thao tác sửa đổi cơ sở dữ liệu có quyền xuất dữ liệu tức thời ra đĩa trong khi giao dịch vẫn còn ở trong trạng thái hoạt động (active state).

Trong kỹ thuật này khi giao dịch thực hiện thao tác ghi, cơ sở dữ liệu sẽ được cập nhật tức thời mà không cần đợi đến khi giao dịch đạt tới trạng thái commit. Các thao tác cập nhật vẫn phải được ghi lại trong file log trước khi ghi lại trong csdl.

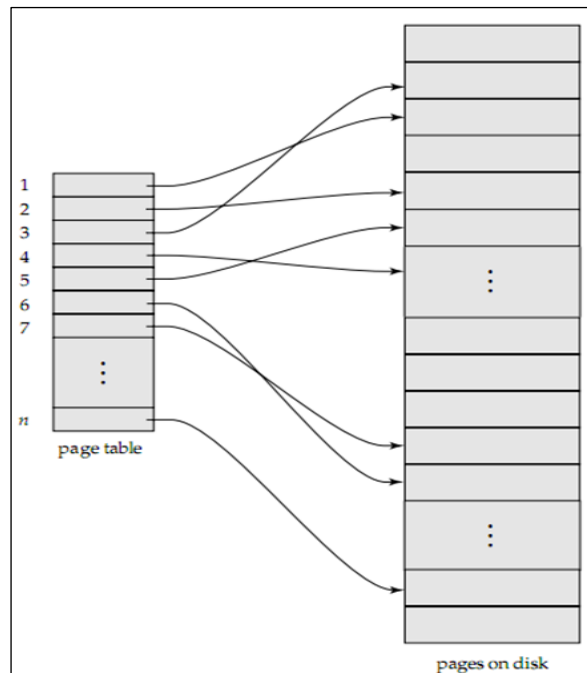
Bảng sau chỉ ra các loại bản ghi được tạo ra trong file log:

Thao tác của giao dịch	Bản ghi trong file Log
Giao dịch Ti bắt đầu	Ti, Start
Ti Write(X)	Ti, X, Old_Value, New_Value
Giao dịch Ti hoàn tất	Ti, Commit

4.4. Kỹ thuật phân trang bóng (Shadow Paging)

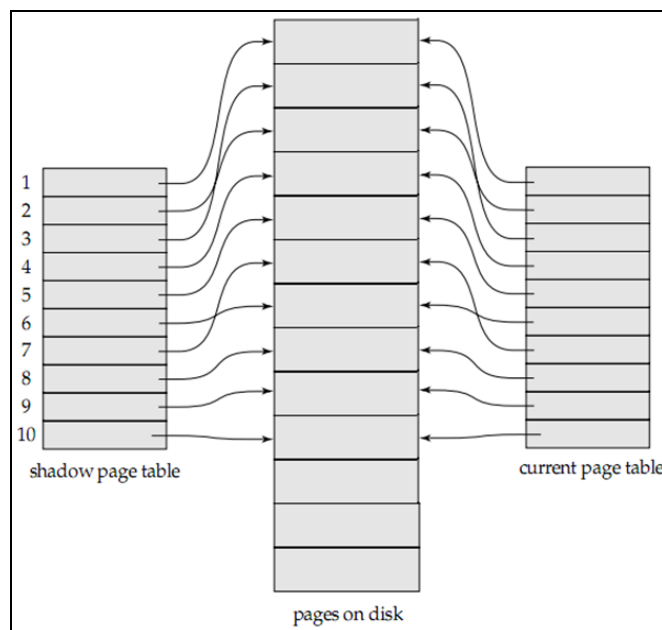
Kỹ thuật “Phân trang bóng” cũng là kỹ thuật cho phép phục hồi sau lỗi, nhưng ý tưởng thực hiện khác với các kỹ thuật dựa trên sổ ghi lộ trình vừa trình bày ở phần trên.

Bảng trang và ý nghĩa của nó: Khái niệm trang đã nói được mượn từ lý thuyết về Hệ điều hành. Cách quản lý trang cũng được thừa kế từ đó. Giả sử rằng cơ sở dữ liệu được phân thành n trang và sự phân bố trên đĩa của chúng có thể không theo một thứ tự cụ thể nào cả. Tuy nhiên, phải có cách để tìm ra nhanh một trang. Người ta dùng bảng trang cho mục đích này. Bảng trang có n đầu vào (entry). Mỗi đầu vào ứng với một trang. Một đầu vào chứa một con trỏ, trỏ đến một trang trên đĩa. Đầu vào đầu tiên chỉ đến trang đầu tiên của cơ sở dữ liệu, đầu vào thứ hai chỉ đến trang thứ hai ...



Hình 4.3: Ví dụ về bảng trang

Ý tưởng then chốt của kỹ thuật “Phân trang bóng” là người ta sẽ duy trì hai bảng trang trong suốt chu kỳ sống của giao dịch, một bảng trang gọi là “bảng trang hiện hành” (current page table), bảng trang còn lại gọi là “bảng trang bóng” (shadow page table). Khi giao dịch khởi động, hai bảng trang này giống nhau. Bảng trang bóng sẽ không thay đổi suốt quá trình hoạt động của giao dịch. Bảng trang hiện hành sẽ bị thay đổi mỗi khi giao dịch thực hiện tác vụ write. Tất cả các thao tác input và output đều sử dụng bảng trang hiện hành để định vị các trang trong đĩa. Điểm quan trọng khác là nên lưu bảng trang bóng vào thiết bị lưu trữ bền.



Hình 4.4: Current Page và Shadow Page

Giả sử giao dịch thực hiện tác vụ write(X) và hạng mục dữ liệu X được chứa trong trang thứ i. Tác vụ write được thực thi như sau:

1. Nếu trang thứ i chưa có trong bộ nhớ chính, thực hiện input(X).
2. Nếu đây là lệnh ghi được thực hiện lần đầu tiên trên trang thứ i bởi giao dịch, sửa đổi bảng trang hiện hành như sau:
 - a) Tìm một trang chưa được dùng trên đĩa.
 - b) Xoá trang vừa được tìm xong ở bước 2.a khỏi danh sách các khung trang tự do.
 - c) Sửa lại bảng trang hiện hành sao cho đầu vào thứ i trở đến trang mới vừa tìm được trong bước 2.a.
 - d) Gán giá trị xi cho X trong trang đệm (buffer page).

Để bàn giao một giao dịch, cần làm các bước sau:

- B1. Đảm bảo rằng tất cả các trang đệm trong bộ nhớ chính đã được giao dịch sửa đổi phải được xuất ra đĩa.
- B2. Xuất bảng trang hiện hành ra đĩa (không được ghi đè lên trang bóng).
- B3. Xuất địa chỉ đĩa của bảng trang hiện hành ra vị trí cố định trong thiết bị lưu trữ bền. Vị trí này chính là nơi chứa địa chỉ của bảng trang bóng. Hành động này sẽ ghi đè lên địa chỉ của bảng trang bóng cũ. Như vậy, bảng trang hiện hành sẽ trở thành bảng trang bóng và giao dịch được bàn giao.

Nếu sự cố xảy ra trước khi hoàn thành bước thứ 3, hệ thống sẽ trở về trạng thái trước khi giao dịch được thực hiện. Nếu sự cố xảy ra sau khi bước thứ 3 hoàn thành, hiệu quả của giao dịch được bảo tồn; không cần thực hiện thao tác redo nào cả.

- Kỹ thuật phân trang bóng có một số điểm lợi hơn so với các kỹ thuật dựa vào file log:
 - Không mất thời gian ghi các log record.
 - Khôi phục sau sự cố nhanh hơn, do không cần các thao tác undo hoặc redo.
- Tuy nhiên kỹ thuật phân trang bóng lại có nhiều nhược điểm:
 - Tốn chi phí bàn giao, xuất nhiều khối ra đĩa: các khối dữ liệu hiện tại, bảng trang hiện hành, địa chỉ của bảng trang hiện hành. Trong khi kỹ thuật dựa vào file log chỉ cần xuất ra các log record.
 - Gây ra hiện tượng phân mảnh dữ liệu: Do kỹ thuật phân trang bóng đổi vị trí của trang khi trang này bị sửa đổi.
 - Ngoài ra, kỹ thuật phân trang bóng sẽ gặp nhiều khó khăn khi cần đáp ứng yêu cầu phục vụ song song cho nhiều giao dịch. Vì những lý do trên, kỹ thuật phân trang bóng không được sử dụng rộng rãi.

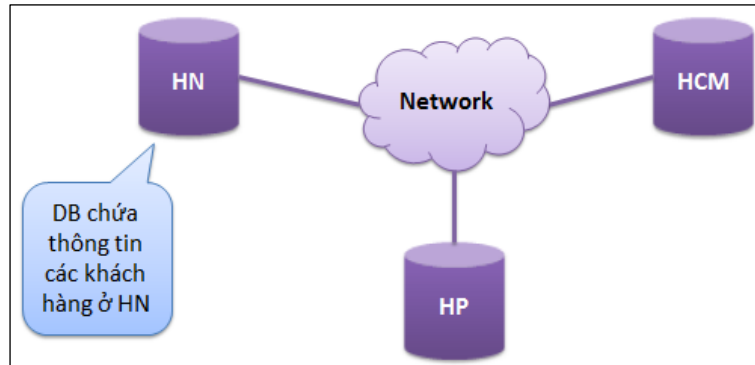
4.5. Câu hỏi ôn tập chương 4

- Trình bày mô hình khóa nhị phân.
- Trình bày các luật trong mô hình khóa nhị phân.
- Trình bày mô hình khóa đọc – ghi.
- Trình bày các luật trong mô hình khóa đọc – ghi.
- Trình bày mô hình khóa 2 pha.
- Trình bày về các biến thể của giao thức khóa hai pha.
- Trình bày hiện tượng Deadlock.
- Trình bày các chiến lược ngăn cản và phát hiện Deadlock.
- Trình bày về giao thức thứ tự nhân thời gian.
- Phân biệt hai kỹ thuật cập nhật trì hoãn và cập nhật tức thời.
- Trình bày kỹ thuật Shadow Paging.

CHƯƠNG 5: CƠ SỞ DỮ LIỆU PHÂN TÁN

5.1. Giới thiệu

- Cơ sở dữ liệu phân tán (Distributed Database) là cơ sở dữ liệu được phân đoạn và được lưu trữ trên các trạm trong hệ thống mạng. Hình 6.1 dưới đây là một ví dụ về cơ sở dữ liệu phân tán.



Hình 5.1: Ví dụ cơ sở dữ liệu phân tán

- Thiết kế hệ thống thông tin có CSDL phân tán bao gồm:
 - Phân tán và chọn những vị trí đặt dữ liệu;
 - Các chương trình ứng dụng tại các điểm;
 - Thiết kế tổ chức khai thác hệ thống đó trên nền mạng.
- Khi thiết kế các hệ thống CSDL phân tán người ta thường tập trung xoay quanh các câu hỏi?
 - Tại sao lại cần phải phân mảnh?
 - Làm thế nào để thực hiện phân mảnh?
 - Phân mảnh nên thực hiện đến mức độ nào?
 - Có cách gì kiểm tra tính đúng đắn của việc phân mảnh?
 - Các mảnh sẽ được cấp phát trên mạng như thế nào?
 - Những thông tin nào sẽ cần thiết cho việc phân mảnh và cấp phát?
- Hệ quản trị cơ sở dữ liệu phân tán (DDBMS)
 - Cho phép người dùng tạo, sử dụng csdl.
 - Đảm bảo an ninh (cấp phát quyền, 1 nhóm người được sử dụng, ...)
 - Đảm bảo tính trong suốt của csdl (Transperence):
 - Người dùng sử dụng như csdl tập trung.
 - Truy vấn tập trung → Truy vấn phân tán.
- Các ứng dụng:
 - Ứng dụng cục bộ (Local App): Chỉ quan tâm tới dữ liệu ở 1 trạm.
 - Ứng dụng toàn cục (Global App): Liên quan đến nhiều trạm.

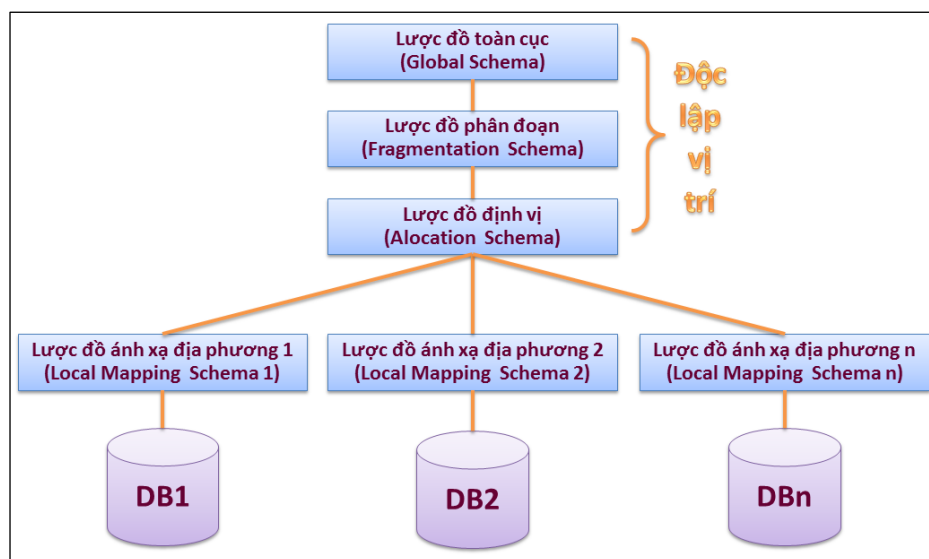
- Các bước thiết kế cơ sở dữ liệu phân tán
 - Thiết kế lược đồ quan niệm: Đối tượng + Mối quan hệ → ER (Giống csdl tập trung).
 - Thiết kế logic – vật lý:
 - Thiết kế phân đoạn: Chia các quan hệ thành các đoạn → LĐồ phân đoạn CSDL.
 - Thiết kế định vị: Đặt các đoạn lên các trạm → LĐồ định vị.

5.2. Ưu nhược điểm của cơ sở dữ liệu phân tán

- Ưu điểm:
 - Dữ liệu gần với nơi xử lý → Hiệu suất cao.
 - Tính sẵn sàng của hệ thống cao: Nếu một trạm bị lỗi sẽ không ảnh hưởng tới các trạm khác trong hệ thống.
 - Việc tăng các trạm sử dụng trong hệ thống là đơn giản nên việc mở rộng CSDL là dễ dàng.
- Nhược điểm:
 - Lưu trữ: Ngoài lược đồ CSDL như trong CSDL tập trung (Thuộc tính, kiểu dữ liệu, ...) còn thêm các lược đồ phân đoạn CSDL, lược đồ định vị CSDL (cho biết các đoạn được lưu trữ ở đâu).
 - Xử lý: Truy vấn tập trung là đơn giản còn truy vấn phân tán phức tạp.
 - An toàn: CSDL được lưu trữ ở nhiều nơi nảy sinh vấn đề: đảm bảo an toàn dữ liệu khi truyền qua mạng.

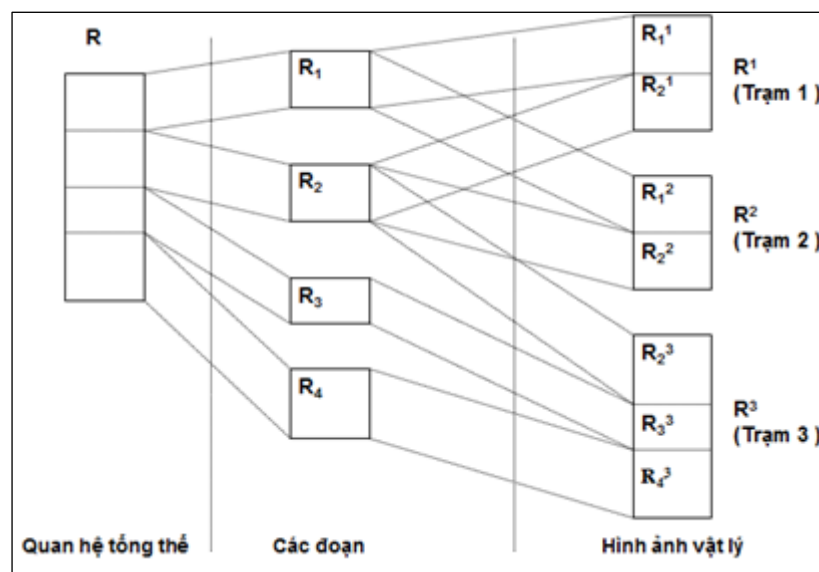
5.3. Cấu trúc chung của một cơ sở dữ liệu phân tán

Cấu trúc mẫu của một cơ sở dữ liệu phân tán có dạng:



Hình 5.2: Cấu trúc mẫu của một cơ sở dữ liệu phân tán

- **Lược đồ toàn cục:**
 - Xác định toàn bộ dữ liệu được lưu trữ trong CSDLPT.
 - Được định nghĩa như trong CSDL tập trung.
 - Trong mô hình quan hệ: lược đồ toàn cục là các quan hệ và mối liên kết giữa chúng.
- **Lược đồ phân đoạn:**
 - Mỗi quan hệ tổng thể có thể được chia thành các phần không giao nhau gọi là phân đoạn (Fragment).
 - Có nhiều cách khác nhau để phân đoạn: Phân đoạn dọc, phân đoạn ngang, phân đoạn hỗn hợp.
 - Các đoạn được mô tả bằng tên của quan hệ tổng thể cùng với chỉ mục đoạn. Ví dụ R_i là đoạn thứ i của quan hệ toàn cục R .
- **Lược đồ định vị:**
 - Xác định đoạn dữ liệu nào được định vị tại trạm nào trên mạng.
 - R_i^j : Cho biết đoạn thứ i của quan hệ tổng thể R được định vị trên trạm j .
- **Lược đồ ánh xạ địa phương:**
 - Ánh xạ các ảnh vật lý và các đối tượng được lưu trữ tại một trạm.



Hình 5.3: Các đoạn và hình ảnh vật lý của một quan hệ tổng thể

5.4. Phân đoạn dữ liệu (Fragmentation)

Trước tiên việc phân tán dữ liệu được thực hiện trên cơ sở cấp phát các tập tin cho các nút trên một mạng máy tính. Các nút mạng thường nằm ở các vị trí địa lý khác nhau trải rộng trên một

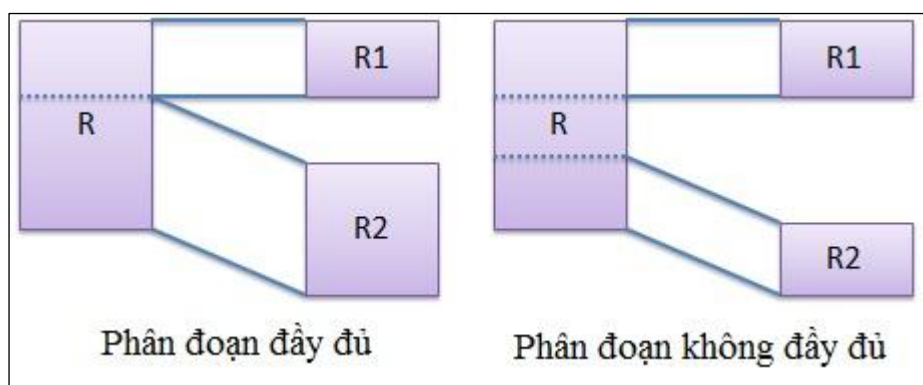
diện tích lớn. Do vậy để tối ưu việc khai thác thông tin thì dữ liệu không thể để tập trung mà phải phân tán trên các nút của mạng.

Hơn nữa một quan hệ không phải là một đơn vị truy xuất dữ liệu tốt nhất. Ví dụ như, nếu ứng dụng được thực hiện trên một bộ phận nhỏ các dữ liệu của quan hệ mà quan hệ đó nằm tại các vị trí khác nhau thì có thể gây ra những truy xuất thừa và hơn thế việc nhân bản các quan hệ làm tốn không gian bộ nhớ. Do vậy phân rã một quan hệ thành nhiều mảnh, mỗi mảnh được xử lý như một đơn vị sẽ cho phép thực hiện nhiều giao dịch đồng thời. Một câu truy vấn ban đầu có thể được chia ra thành một tập các truy vấn con, các truy vấn này có thể được thực hiện song song trên các mảnh sẽ giúp cải thiện tốc độ hoạt động của hệ thống.

Tuy nhiên chúng ta cũng sẽ gặp những rắc rối của việc phân mảnh, ví dụ nếu các ứng dụng có những xung đột sẽ ngăn cản hoặc gây khó khăn cho việc truy xuất dữ liệu. Phân rã các mảnh nói chung làm tăng chi phí trong việc truy xuất dữ liệu. Một vấn đề nữa liên quan đến việc kiểm soát ngữ nghĩa và tính toàn vẹn dữ liệu.

Thể hiện của các quan hệ chính là các bảng, vì thế vấn đề là tìm những cách khác nhau để chia một bảng thành nhiều bảng nhỏ hơn. Rõ ràng có hai phương pháp khác nhau: Chia bảng theo chiều dọc và chia bảng theo chiều ngang. Chia dọc ta được các quan hệ con mà mỗi quan hệ chứa một tập con các thuộc tính của quan hệ gốc – gọi là phân mảnh dọc. Chia ngang một quan hệ ta được các quan hệ con mà mỗi quan hệ chứa một số bộ của quan hệ gốc – gọi là phân mảnh ngang. Ngoài ra còn có một khả năng hỗn hợp, đó là phân mảnh kết hợp cách phân mảnh ngang và dọc.

- Phân đoạn một quan hệ R là chia R thành các đoạn R_i : $F_R = \{R_i\}, i = 1, n$
- Điều kiện phân đoạn đúng đắn:
 - Điều kiện đầy đủ: Một mục dữ liệu trong R phải được đưa vào một R_i nào đó. Đảm bảo dữ liệu trong R được ánh xạ vào các đoạn mà không bị mất.



Hình 5.4: Phân đoạn đầy đủ và phân đoạn không đầy đủ

- Điều kiện xây dựng lại: Nếu R được phân đoạn thành các R_i thì phải định nghĩa được một toán tử quan hệ ∇ sao cho: $R = \nabla R_i$.
- Điều kiện tách rời

- Nếu R tách thành các R_i và d là một mục dữ liệu trong R_i thì d không nằm trong R_k ($\forall k \neq i$).
- Trong phân đoạn ngang: Điều kiện này đảm bảo các đoạn ngang là tách biệt.
- Trong phân đoạn dọc tính tách biệt chỉ áp dụng cho các thuộc tính không khóa.

5.4.1. Phân đoạn ngang

- Đoạn ngang (Horizontal Fragmentation) của một quan hệ R là một nhóm các bộ trong R có cùng một tính chất nào đó. $R_i = \delta_{C_i}(R)$, C_i là điều kiện phân đoạn.
- Ví dụ: KháchHang (MaKH, HoTen, NgaySinh, SoDT, DiaChi)
 - $KH_1 = \delta_{DiaChi='Hà Nội'}(KhachHang)$
 - $KH_2 = \delta_{DiaChi='Hà Nội'}(KhachHang)$
 - $KH_3 = \delta_{DiaChi='Tp. HCM'}(KhachHang)$

Đảm bảo điều kiện phân đoạn đúng đắn:

- Điều kiện đầy đủ: Mỗi bộ đều thuộc một đoạn.
- Điều kiện xây dựng lại: $KhachHang = KH_1 \cup KH_2 \cup KH_3$
- Điều kiện tách rời: $KH_i \cap KH_j = \phi, \forall i \neq j$.

5.4.2. Phân đoạn ngang dẫn xuất

- Phân đoạn ngang dẫn xuất (Derived Horizontal Fragmentation) Là phân đoạn ngang dựa trên một quan hệ khác đã được phân đoạn ngang.
- Ví dụ: Quan hệ nhân viên được phân đoạn theo đơn vị:
 - $NV_1 = \delta_{MaDV='PKT'}(NhanVien)$
 - $NV_2 = \delta_{MaDV='PTC'}(NhanVien)$
 - $NV_3 = \delta_{MaDV='PHC'}(NhanVien)$

Có quan hệ NV_DA(MaNV, MaDA, SoGio). Cần phân đoạn quan hệ này thành 3 đoạn theo NV1, NV2, NV3.

Ta sử dụng phép nửa kết nối (Semi – Join): $R \bowtie S = \prod_{\text{các thuộc tính của } R} (R \bowtie S)$. Khi đó quan hệ NV_DA được phân đoạn thành:

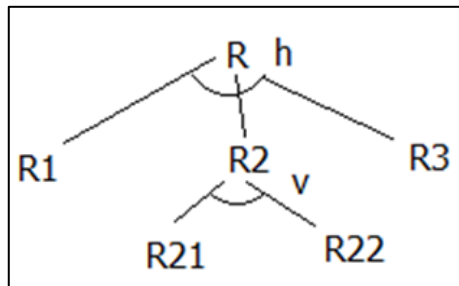
- $NV_DA_1 = NV_DA \bowtie_{MaNV} NV_1$
- $NV_DA_2 = NV_DA \bowtie_{MaNV} NV_2$
- $NV_DA_3 = NV_DA \bowtie_{MaNV} NV_3$

5.4.3. Phân đoạn dọc

- Phân đoạn dọc (Vertical Fragmentation): Là sự phân chia tập thuộc tính của quan hệ toàn cục thành các tập thuộc tính con.
- Các đoạn dọc có được bằng cách chiếu quan hệ toàn cục trên mỗi tập thuộc tính con:
 $R_i = \Pi_{L_i}(R)$.
- Để xây dựng lại R: Mỗi R_i đều phải chứa khóa chính của R: $R_i \cap R_j = \text{Primary Key}_R$, $\forall i \neq j$.
- Khi đó biểu thức xây dựng lại: $R = R_1 * R_2 * \dots * R_n$.
- Ví dụ: NhanVien(MaNV, TenNV, Luong, Thue, MaNQL, MaPB)
 - $NV_1 = \Pi_{MaNV, TenNV, MaNQL, MaPB}(NhanVien)$
 - $NV_2 = \Pi_{MaNV, Luong, Thue}(NhanVien)$Khi đó, $NhanVien = NV_1 * NV_2$

5.4.4. Phân đoạn hỗn hợp

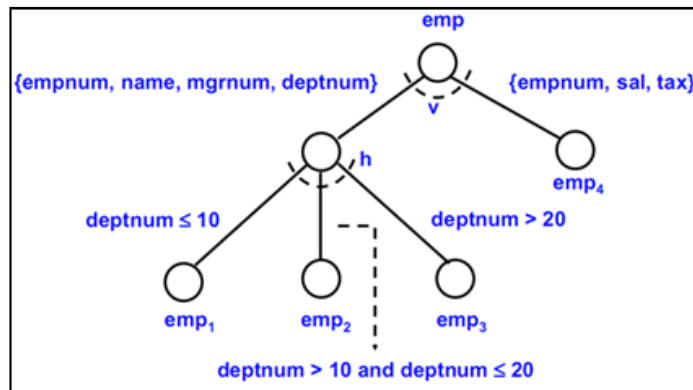
Một mảnh ngang được phân đoạn dọc hoặc một mảnh dọc được phân đoạn ngang.
Biểu diễn dưới dạng cây \rightarrow Chỉ rõ phân đoạn như thế nào.



Hình 5.5: Cây phân đoạn hỗn hợp của một quan hệ

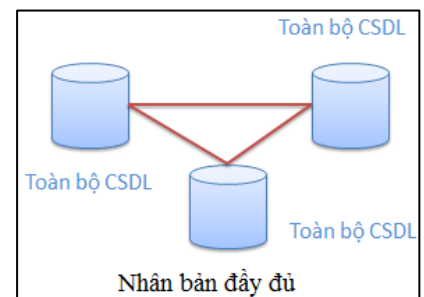
Ví dụ: Xét quan hệ emp (empnum, name, sal, tax, mgrnum, deptnum) và thông tin phân đoạn:

- emp được phân đoạn dọc thành 2 đoạn: emp4 và emp5.
 - ✓ emp4: các thông tin về lương: empnum, sal, tax.
 - ✓ emp5: Các thông tin về nhân sự: empnum, name, mgrnum, deptnum.
- Emp5: phân mảnh ngang thành 3 đoạn emp1, emp2, emp3 theo deptnum: ≤ 10 , $(10,20]$ và >20 .



5.5. Thiết kế định vị dữ liệu

- Thiết kế định vị dữ liệu dựa trên nguyên tắc:
 - Nếu tần số sử dụng của đoạn Ri tại trạm j vượt quá một ngưỡng θ nào đó thì đoạn Ri sẽ được đặt tại trạm j.
 - Mỗi đoạn có thể được đặt trên nhiều trạm.
- Có 3 cách định vị:
 - Nhân bản đầy đủ: Mỗi trạm chứa toàn bộ CSDL.
 - Ưu điểm: Xử lý nhanh.
 - Nhược điểm: Mất thời gian cập nhật, sửa đổi.
 - Không nhân bản:
 - Ví dụ: $\begin{cases} R \rightarrow R1, R2, R3 \\ S \rightarrow S1, S2, S3 \end{cases}$
 - Ưu điểm: Cập nhật nhanh.
 - Nhược điểm: Xử lý chậm trong việc thống kê, tổng hợp dữ liệu.
 - Kết hợp.



Ví dụ một công ty có hệ thống mạng máy tính với 3 trạm tương ứng với 3 đơn vị có mã số: 1, 2 và 3.

- Trạm 1: Ban quản lý dự án.
- Trạm 2 và 3 dành cho hai đơn vị 2, 3. Trong mỗi trạm, phòng nhân sự thường xuyên cần các thông tin: MasoNV, HoDem, Ten, NgaySinh, GioiTinh, DiaChi. Phòng kế toán thường xuyên cần các thông tin về: MasoNV, MasoNGS, MasoDV, Luong.

Các bảng cơ sở dữ liệu như sau:

- DonVi(MasoDV, TenDV, MasoNQL, Ngay)
- NhanVien(MasoNV, HoDem, Ten, NgaySinh, GioiTinh, DiaChi, Luong, MaSoNGS, MasoDV)

- DuAn(MasoDA, TenDA, DiaDiemDA, MasoDV)
- NV_DA (MasoNV, MasoDA, SoGio)

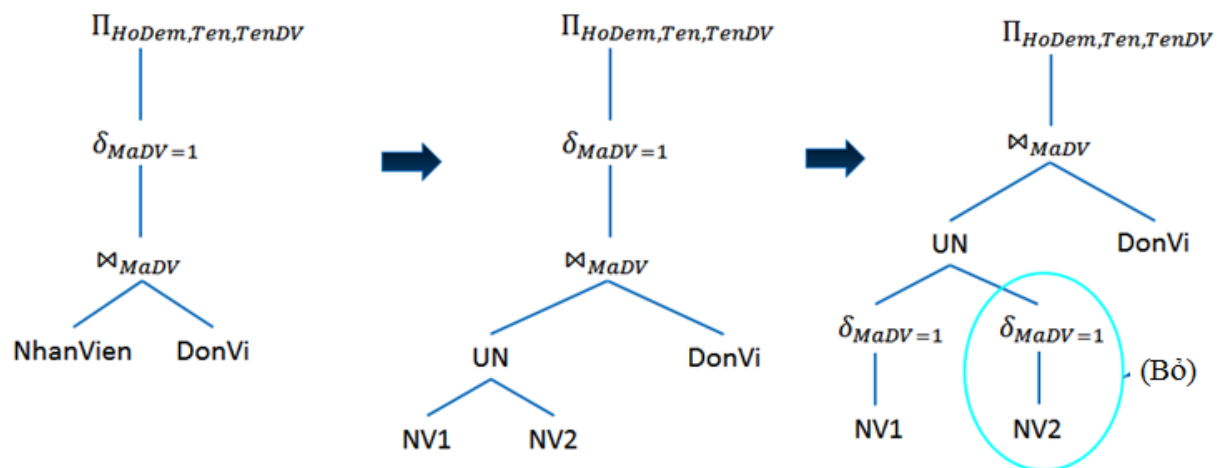
Thiết kế định vị như sau:

- Trạm 1: Toàn bộ CSDL
- Trạm 2:
 - $NV2 = \delta_{MasoDV=2}(NhanVien)$
 - $NV2$ tách thành: $\begin{cases} NV21 = \Pi_{MasoNV, HoDem, Ten, NgaySinh, GioiTinh, DiaChi}(NV2) \\ NV22 = \Pi_{MasoNV, MasoNGS, MasoDV, Luong}(NV2) \end{cases}$
 - $DA2 = \delta_{MasoDV=2}(DuAn)$
 - $NV_DA2 = NV_DA \bowtie_{MasoDV} DA2$
- Trạm 3: Tương tự trạm 2.

5.6. Biến đổi truy vấn toàn cục thành truy vấn đoạn

- Kỹ thuật:
 - Biểu diễn truy vấn toàn cục dưới dạng cây
 - Thay thế mối quan hệ bằng biểu thức xây dựng lại:
 - Phân đoạn ngang: $R = R_1 \cup R_2 \cup \dots \cup R_n$.
 - Phân đoạn dọc: $R = R_1 * R_2 * \dots * R_n$.
 - Biến đổi truy vấn: Sử dụng các phép biến đổi tương đương.
 - Bỏ các nhánh của cây trong đó điều kiện truy vấn và điều kiện phân đoạn mâu thuẫn với nhau.
- Ví dụ: Xét hai bảng NhanVien, DonVi:
 - NhanVien(MaNV, HoDem, Ten, Luong, MaDV)
 - DonVi(MaDV, TenDV, MaNQL)
 - NhanVien được phân đoạn ngang làm 2 theo MaDV = 1, 2:
 - $NV1 = \delta_{MaDV=1}(NhanVien)$
 - $NV2 = \delta_{MaDV=2}(NhanVien)$
 - Cho truy vấn toàn cục: Đưa ra họ đệm, tên, tên đơn vị của các nhân viên có MaDV = 1.

Truy vấn toàn cục Q: $\Pi_{HoDem, Ten, TenDV}(\delta_{MaDV=1}(NhanVien \bowtie_{MaDV} DonVi))$. Chuyển Q thành truy vấn đoạn:



5.7. Câu hỏi ôn tập chương 5

- Định nghĩa cơ sở dữ liệu phân tán. Cho ví dụ.
- Trình bày ưu nhược điểm của CSDLPT.
- Phân đoạn ngang là gì. Cho ví dụ.
- Phân đoạn ngang dẫn xuất là gì. Cho ví dụ.
- Phân đoạn dọc là gì. Cho ví dụ.
- Phân đoạn hỗn hợp là gì. Cho ví dụ.
- Trình bày các cách định vị cơ sở dữ liệu.
- Trình bày phương pháp chuyển đổi truy vấn toàn cục thành truy vấn đoạn.
- Cho các quan hệ toàn cục:
 - KhoHang (MaKH, TenKH, ViTri)
 - LoaiHang (MaLH, TenLH, MoTa, MaKH)

Giả sử KhoHang, LoaiHang được phân đoạn ngang theo $\text{MaKH} = 1$ và $\text{MaKH} = 2$ thành KhoHang1, KhoHang2, LoaiHang1, LoaiHang2.

- ✓ Viết biểu thức đại số quan hệ của các đoạn trên.
- ✓ Viết biểu thức xây dựng lại của 2 quan hệ toàn cục DonVi, NhanVien.
- ✓ Hãy viết truy vấn toàn cục Q_{TC} : Lập danh sách gồm TenLH, ViTri của loại hàng có $\text{MaLH} = \text{'LH01'}$ và ở kho hàng có $\text{MaKH} = 2$.
- ✓ Chuyển truy vấn Q_{TC} ở trên thành truy vấn đoạn.

MỘT SỐ ĐỀ THI MẪU

Trường Đại Học Hàng Hải Việt Nam
Khoa Công nghệ Thông tin
BỘ MÔN HỆ THỐNG THÔNG TIN
 -----***-----

THI KẾT THÚC HỌC PHẦN

Tên học phần: CƠ SỞ DỮ LIỆU NÂNG CAO Năm học: 2010 – 2011	Đề thi số: 1	Ký duyệt đề:
Thời gian: 60 phút Thang điểm: 100		

PHẦN I – TRẮC NGHIỆM (30 ĐIỂM)

Câu 1: (10 điểm)

Cho các quan hệ:

- Khoa (MKhoa, TenKhoa, DiaDiem)
- Lop (MaLop, TenLop, SiSo, MaKhoa)

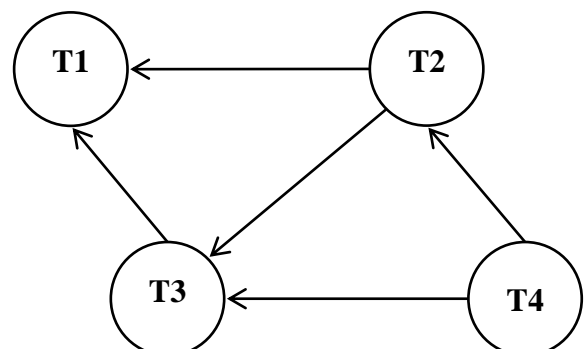
và truy vấn Q: Hiển thị TenLop của các lớp có SiSo > 50 và thuộc khoa 'Kinh Tế'. Hỏi biểu thức đại số quan hệ nào dưới đây đã tối ưu?

- A. $\Pi_{TenLop}(\delta_{TenKhoa='Kinh\ Tế' \wedge SiSo > 50}(Lop \bowtie_{MKhoa=MaKhoa} Khoa))$
- B. $\Pi_{TenLop}(\Pi_{MKhoa}(\delta_{TenKhoa='Kinh\ Tế'}(Khoa)) \bowtie_{MKhoa=MaKhoa} \Pi_{TenLop, MaKhoa}(\delta_{SiSo > 50}(Lop)))$
- C. $\Pi_{TenLop}(\delta_{TenKhoa='Kinh\ Tế'}(Khoa) \bowtie_{MKhoa=MaKhoa} \delta_{SiSo > 50}(Lop))$
- D. $\Pi_{TenLop}(\delta_{TenKhoa='Kinh\ Tế'}(Khoa) \bowtie_{MKhoa=MaKhoa} \Pi_{TenLop, MaKhoa}(\delta_{SiSo > 50}(Lop)))$

Câu 2: (10 điểm)

Cho đồ thị phụ thuộc của lịch biểu S. Lịch biểu nào dưới đây là lịch biểu tuần tự tương đương với S:

- A. $T1 \rightarrow T2 \rightarrow T3 \rightarrow T4$
- B. $T2 \rightarrow T1 \rightarrow T3 \rightarrow T4$
- C. $T4 \rightarrow T2 \rightarrow T3 \rightarrow T1$
- D. $T4 \rightarrow T3 \rightarrow T1 \rightarrow T2$



Câu 3: (10 điểm)

Lịch biểu nào dưới đây là khả tuần tự:

- A. $R1(X), R2(Y), R1(Y), W1(X), R3(Y), W2(Y), W3(Y)$
- B. $R1(X), R2(Y), R1(Y), W1(X), R3(Z), W2(Y), W3(Z)$
- C. $R1(X), R2(Z), R1(Y), W1(X), R3(Z), W2(Z), W3(Z)$
- D. $R1(Z), R2(Z), R1(Y), W1(Z), R3(Y), W2(Z), W3(Y)$

PHẦN II – TỰ LUẬN (70 ĐIỂM)

Câu 4: (30 điểm)

Cho các quan hệ sau:

- LoaiHang (MaLH, TenLH, MoTa)
- HangHoa (MaHH, TenHH, HinhThuc, SoLuong, MaLH)

và truy vấn Q: Lập danh sách gồm MaHH, TenHH của các mặt hàng có số lượng >100 và thuộc loại hàng ‘Mỹ phẩm’.

- a. Viết biểu thức đại số quan hệ thực hiện truy vấn Q.
- b. Sử dụng phương pháp tối ưu Heuristic để tối ưu hóa Q.

Câu 5: (40 điểm)

Cho các quan hệ toàn cục:

- Khoa (MaKhoa, TenKhoa, DiaDiem)
- Lop (MaLop, TenLop, SiSo, MaKhoa)

Giả sử Khoa và Lop được phân đoạn ngang theo $MaKhoa = 1$ và $MaKhoa = 2$ thành Khoa1, Khoa2, Lop1, Lop2.

- a. Viết biểu thức đại số quan hệ của các đoạn trên.
- b. Viết biểu thức xây dựng lại của 2 quan hệ toàn cục Khoa, Lop.
- c. Hãy viết truy vấn toàn cục Q_{TC} : Lập danh sách gồm TenLop, TenKhoa của các lớp có $MaKhoa = 2$.
- d. Chuyển truy vấn Q_{TC} ở câu c thành truy vấn đoạn.

-----***HẾT***-----

Lưu ý: - Không sửa, xóa đề thi, nộp lại đề sau khi thi

Trường Đại Học Hàng Hải Việt Nam
Khoa Công nghệ Thông tin
BỘ MÔN HỆ THỐNG THÔNG TIN
 -----***-----

THI KẾT THÚC HỌC PHẦN

Tên học phần: CƠ SỞ DỮ LIỆU NÂNG CAO	Đề thi số:	Ký duyệt đề:
Năm học: 2010 – 2011	2	
Thời gian: 60 phút		
Thang điểm: 100		

PHẦN I – TRẮC NGHIỆM (30 ĐIỂM)

Câu 1: (10 điểm)

Cho các quan hệ:

- Khoa (MKH, TenKH, DiaDiem, MaCNK)
- Lop (MaLH, TenLH, SiSo, MaKH)

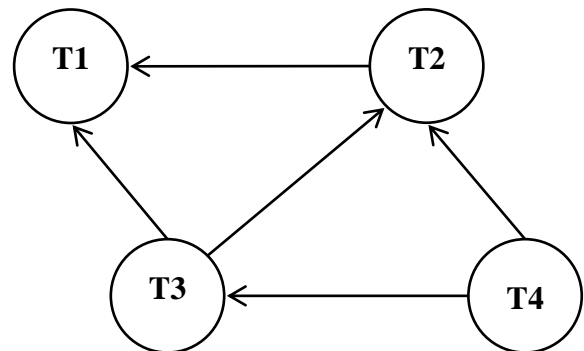
và truy vấn Q: Hiển thị MaLH, TenLH, SiSo của các lớp thuộc khoa ‘CNT’. Hỏi biểu thức đại số quan hệ nào dưới đây đã tối ưu?

- A. $\Pi_{MaLH, TenLH, SiSo} (Lop \bowtie_{MaKH=MKH} \Pi_{MKH} (\delta_{TenKH='CNT'} (Khoa)))$
- B. $\Pi_{MaLH, TenLH, SiSo} (\delta_{TenKH='CNT'} (Lop \bowtie_{MaKH=MKH} Khoa))$
- C. $\Pi_{MaLH, TenLH, SiSo} (Lop \bowtie_{MaKH=MKH} \delta_{TenKH='CNT'} (Khoa))$
- D. $\Pi_{MaLH, TenLH, SiSo} (\delta_{TenKH='CNT'} (Lop \bowtie_{MaKH=MKH} \Pi_{MKH, TenKH} (Khoa)))$

Câu 2: (10 điểm)

Cho đồ thị phụ thuộc của lịch biểu S.
 Lịch biểu nào dưới đây là lịch biểu tuần tự tương đương với S?

- A. $T4 \rightarrow T1 \rightarrow T2 \rightarrow T3$
- B. $T4 \rightarrow T3 \rightarrow T2 \rightarrow T1$
- C. $T3 \rightarrow T1 \rightarrow T4 \rightarrow T2$
- D. $T4 \rightarrow T3 \rightarrow T1 \rightarrow T2$



Câu 3: (10 điểm)

Lịch biểu nào dưới đây là khả tuần tự?

- A. $R1(X), R1(Y), R3(Z), R2(Y), W3(Z), W2(Y), W1(X)$
- B. $R2(Y), R1(X), R1(Y), R3(X), W3(X), W2(Y), W1(X)$
- C. $R3(X), R2(Y), R1(X), R1(Y), W2(Y), W1(Y), W3(X)$
- D. $R1(X), R1(Y), R3(X), R2(Y), W2(Y), W1(Y), W3(X)$

PHẦN II – TỰ LUẬN (70 ĐIỂM)

Câu 4: (30 điểm)

Cho các quan hệ sau:

- HoaDon (MaHD, MaKH, NgayLap)
- ChiTiet (MaHD, MaHang, SoLuong, GiaBan, ChietKhau)

và truy vấn Q: Lập danh sách gồm MaHang, SoLuong của các mặt hàng bán vào ngày ‘29/04/2011’.

- a. Viết biểu thức đại số quan hệ thực hiện truy vấn Q.
- b. Sử dụng phương pháp tối ưu Heuristic để tối ưu hóa Q.

Câu 5: (40 điểm)

Cho các quan hệ toàn cục:

- KhoHang (MaKH, TenKH, ViTri)
- LoaiHang (MaLH, TenLH, MoTa, MaKH)

Giả sử KhoHang, LoaiHang được phân đoạn ngang theo $MaKH = 1$ và $MaKH = 2$ thành KhoHang1, KhoHang2, LoaiHang1, LoaiHang2.

- a. Viết biểu thức đại số quan hệ của các đoạn trên.
- b. Viết biểu thức xây dựng lại của 2 quan hệ toàn cục DonVi, NhanVien.
- c. Hãy viết truy vấn toàn cục Q_{TC} : Lập danh sách gồm TenLH, ViTri của loại hàng có $MaLH = 'LH01'$ và ở kho hàng có $MaKH = 2$.
- d. Chuyển truy vấn Q_{TC} ở câu c thành truy vấn đoạn.

-----***HẾT***-----

Lưu ý: - Không sửa, xóa đề thi, nộp lại đề sau khi thi

Trường Đại Học Hàng Hải Việt Nam
Khoa Công nghệ Thông tin
BỘ MÔN HỆ THỐNG THÔNG TIN
 -----***-----

THI KẾT THÚC HỌC PHẦN

Tên học phần: CƠ SỞ DỮ LIỆU NÂNG CAO	Đề thi số:	Ký duyệt đề:
Năm học: 2010 – 2011	3	
Thời gian: 60 phút		
Thang điểm: 100		

PHẦN I – TRẮC NGHIỆM (30 ĐIỂM)

Câu 1: (10 điểm)

Cho các quan hệ:

- LoaiHang (MLH, TenLH, MoTa)
- Hang (MaHH, TenHH, HinhThuc, SoLuong, MaLH)

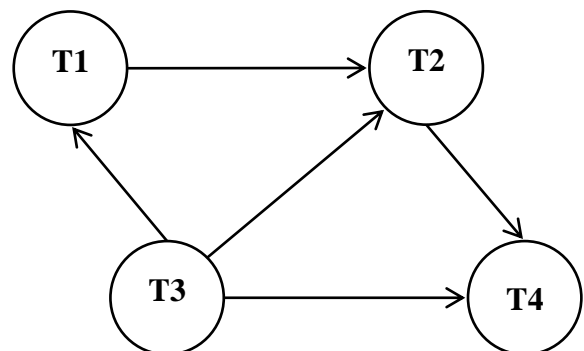
và truy vấn Q: Hiển thị TenHH của các mặt hàng có SoLuong < 10 và thuộc loại hàng ‘Điện Tử’. Hỏi biểu thức đại số quan hệ nào dưới đây đã tối ưu?

- A. $\Pi_{TenHH} \left(\delta_{TenLH='Điện Tử'} \wedge SoLuong < 10 (LoaiHang \bowtie_{MLH=MaLH} Hang) \right)$
- B. $\Pi_{TenHH} \left(\delta_{TenLH='Điện Tử'} (LoaiHang) \bowtie_{MLH=MaLH} \delta_{SoLuong < 10} (Hang) \right)$
- C. $\Pi_{TenHH} \left(\delta_{TenLH='Điện Tử'} (LoaiHang) \bowtie_{MLH=MaLH} \Pi_{TenHH, MLH} \left(\delta_{SoLuong < 10} (Hang) \right) \right)$
- D. $\Pi_{TenHH} \left(\Pi_{MLH} \left(\delta_{TenLH='Điện Tử'} (LoaiHang) \right) \bowtie_{MLH=MaLH} \Pi_{TenHH, MaLH} \left(\delta_{SoLuong < 10} (Hang) \right) \right)$

Câu 2: (10 điểm)

Cho đồ thị phụ thuộc của lịch biểu S.
 Lịch biểu nào dưới đây là lịch biểu tuần tự tương đương với S:

- A. $T3 \rightarrow T1 \rightarrow T2 \rightarrow T4$
- B. $T3 \rightarrow T2 \rightarrow T1 \rightarrow T4$
- C. $T4 \rightarrow T2 \rightarrow T3 \rightarrow T1$
- D. $T4 \rightarrow T3 \rightarrow T1 \rightarrow T2$



Câu 3: (10 điểm)

Lịch biểu nào dưới đây là khả tuần tự:

- A. $R1(X), R2(Y), R1(Y), W2(Y), R3(Y), W1(X), W3(Y)$
- B. $R1(X), R2(Y), R1(Y), R3(Y), W1(X), W3(Y), W2(Y)$
- C. $R1(X), R2(Y), R1(Y), R3(Y), W3(Y), W2(Y), W1(X)$
- D. $R1(Y), R3(Y), R1(X), R2(Y), W3(Y), W2(Y), W1(X)$

PHẦN II – TỰ LUẬN (70 ĐIỂM)

Câu 4: (30 điểm)

Cho các quan hệ sau:

- Khoa (MKhoa, TenKhoa, DiaDiem)
- Lop (MaLop, TenLop, SiSo, MaKhoa)

và truy vấn Q: Lập danh sách gồm MaLop, TenLop của các lớp có sĩ số nhỏ hơn 30 và thuộc khoa ‘Điện tự động’.

- a. Viết biểu thức đại số quan hệ thực hiện truy vấn Q.
- b. Sử dụng phương pháp tối ưu Heuristic để tối ưu hóa Q.

Câu 5: (40 điểm)

Cho các quan hệ toàn cục:

- DonVi (MaDV, TenDV, DiaDiem)
- NhanVien (MaNV, HoTen, NgaySinh, GioiTinh, Luong, MaDV)

Giả sử DonVi và NhanVien được phân đoạn ngang theo $MaDV = 1$ và $MaDV = 2$ thành DonVi1, DonVi2, NhanVien1, NhanVien2.

- a. Viết biểu thức đại số quan hệ của các đoạn trên.
- b. Viết biểu thức xây dựng lại của 2 quan hệ toàn cục DonVi, NhanVien.
- c. Hãy viết truy vấn toàn cục Q_{TC} : Lập danh sách gồm HoTen, TenDV của các nhân viên thuộc đơn vị có $MaDV = 1$.
- d. Chuyển truy vấn Q_{TC} ở câu c thành truy vấn đoạn.

-----***HẾT***-----

Lưu ý: - Không sửa, xóa đề thi, nộp lại đề sau khi thi

Trường Đại Học Hàng Hải Việt Nam
Khoa Công nghệ Thông tin
BỘ MÔN HỆ THỐNG THÔNG TIN
 -----***-----

THI KẾT THÚC HỌC PHẦN

Tên học phần: CƠ SỞ DỮ LIỆU NÂNG CAO	Đề thi số:	Ký duyệt đề:
Năm học: 2010 – 2011	4	
Thời gian: 60 phút		
Thang điểm: 100		

PHẦN I – TRẮC NGHIỆM (30 ĐIỂM)

Câu 1: (10 điểm)

Cho các quan hệ:

- SinhVien (MSV, TenSV, NgaySinh, DiaChi, MaLop)
- KetQua (MaSV, MaMH, Diem)

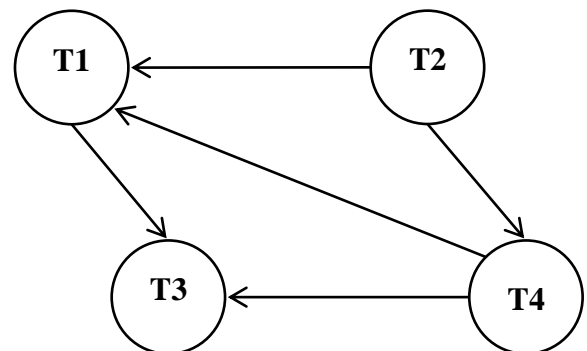
và truy vấn Q: Hiển thị MaMH, Diem của sinh viên ‘Trần Huyền’. Hỏi biểu thức đại số quan hệ nào dưới đây đã tối ưu?

- A. $\Pi_{MaMH, Diem} \left(\delta_{TenSV='Trần Huyền'} (SinhVien \bowtie_{MSV=MaSV} KetQua) \right)$
- B. $\Pi_{MaMH, Diem} \left(KetQua \bowtie_{MaSV=MSV} \delta_{TenSV='Trần Huyền'} (SinhVien) \right)$
- C. $\Pi_{MaMH, Diem} \left(\delta_{TenSV='Trần Huyền'} \left(KetQua \bowtie_{MaSV=MSV} \Pi_{MSV, TenSV} (SinhVien) \right) \right)$
- D. $\Pi_{MaMH, Diem} \left(\Pi_{MSV} \left(\delta_{TenSV='Trần Huyền'} (SinhVien) \right) \bowtie_{MSV=MaSV} KetQua \right)$

Câu 2: (10 điểm)

Cho đồ thị phụ thuộc của lịch biểu S. Lịch biểu nào dưới đây là lịch biểu tuần tự tương đương với S?

- A. $T2 \rightarrow T4 \rightarrow T1 \rightarrow T3$
- B. $T4 \rightarrow T3 \rightarrow T2 \rightarrow T1$
- C. $T2 \rightarrow T1 \rightarrow T4 \rightarrow T3$
- D. $T4 \rightarrow T3 \rightarrow T1 \rightarrow T2$



Câu 3: (10 điểm)

Lịch biểu nào dưới đây là khả tuần tự?

- A. R1(X), R1(Z), R3(Z), R2(Y), W3(Z), W2(Y), W1(Z)
- B. R1(X), R2(Y), R1(Y), R3(X), W3(X), W1(X), W2(Y)
- C. R3(Z), R2(Y), R1(X), R1(Y), W1(X), W3(Z), W2(Y)
- D. R3(X), R1(X), R1(Y), R2(Y), W2(Y), W1(Y), W3(X)

PHẦN II – TỰ LUẬN (70 ĐIỂM)

Câu 4: (30 điểm)

Cho các quan hệ sau:

- DonVi(MDV, TenDV, MaNQL, DiaDiem)
- DuAn (MaDA, TenDA, KinhPhi, MaDV)

và truy vấn Q: Hiển thị TenDA của các dự án có KinhPhi > 100000 và do đơn vị có TenDV = 'XD1' quản lý.

- a. Viết biểu thức đại số quan hệ thực hiện truy vấn Q.
- b. Sử dụng phương pháp tối ưu Heuristic để tối ưu hóa Q.

Câu 5: (40 điểm)

Cho các quan hệ toàn cục:

- NhanVien (MaNV, TenNV, DiaChi, Luong, PhuCap, MaNQL, MaPB)

và thông tin phân đoạn như sau:

- NhanVien được phân đoạn dọc thành:
 - NV1 chứa các thông tin: MaNV, TenNV, DiaChi, MaNQL, MaPB
 - NV2 chứa các thông tin: MaNV, Luong, PhuCap
- NV1 được phân đoạn ngang thành:
 - NV11: Những nhân viên có MaPB < 5.
 - NV12: Những nhân viên có MaPB ≥ 5.

- a. Viết biểu thức đại số quan hệ của các đoạn trên.
- b. Viết biểu thức xây dựng lại của 2 quan hệ toàn cục NhanVien.
- c. Hãy viết truy vấn toàn cục Q_{TC}: Lập danh sách gồm TenNV, Luong, PhuCap của các nhân viên có MaPB = 8.
- d. Chuyển truy vấn Q_{TC} ở câu c thành truy vấn đoạn.

-----***HẾT***-----

Lưu ý: - Không sửa, xóa đề thi, nộp lại đề sau khi thi

Trường Đại Học Hàng Hải Việt Nam
Khoa Công nghệ Thông tin
BỘ MÔN HỆ THỐNG THÔNG TIN
 -----***-----

THI KẾT THÚC HỌC PHẦN

Tên học phần: CƠ SỞ DỮ LIỆU NÂNG CAO	Đề thi số:	Ký duyệt đề:
Năm học: 2010 – 2011	5	
Thời gian: 60 phút		
Thang điểm: 100		

PHẦN I – TRẮC NGHIỆM (30 ĐIỂM)

Câu 1: (10 điểm)

Cho các quan hệ:

- DonVi (MDV, TenDV, DiaDiem)
- NhanVien (MaNV, HoTen, NgaySinh, GioiTinh, Luong, MaDV)

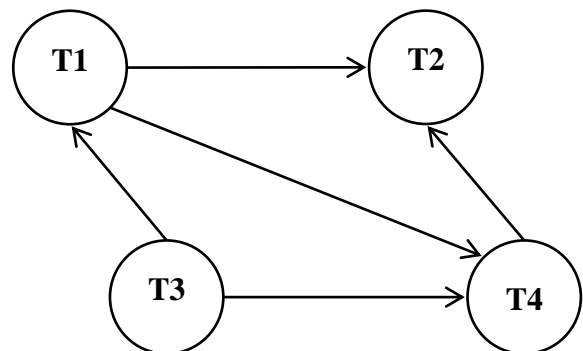
và truy vấn Q: Hiển thị HoTen của các nhân viên nữ thuộc đơn vị 'PKD'. Hỏi biểu thức đại số quan hệ nào dưới đây đã tối ưu?

- A. $\Pi_{HoTen} \left(\Pi_{MDV} \left(\delta_{TenDV='PKD'}(DonVi) \right) \bowtie_{MDV=MaDV} \Pi_{HoTen, MaDV} \left(\delta_{GioiTinh='Nữ'}(NhanVien) \right) \right)$
- B. $\Pi_{HoTen} \left(\delta_{TenDV='PKD'}(DonVi) \bowtie_{MDV=MaDV} \Pi_{HoTen, MaDV} \left(\delta_{GioiTinh='Nữ'}(NhanVien) \right) \right)$
- C. $\Pi_{HoTen} \left(\delta_{TenDV='PKD' \wedge GioiTinh='Nữ'}(DonVi \bowtie_{MDV=MaDV} NhanVien) \right)$
- D. $\Pi_{HoTen} \left(\delta_{TenDV='PKD'}(DonVi) \bowtie_{MDV=MaDV} \delta_{GioiTinh='Nữ'}(NhanVien) \right)$

Câu 2: (10 điểm)

Cho đồ thị phụ thuộc của lịch biểu S.
 Lịch biểu nào dưới đây là lịch biểu tuần tự tương đương với S:

- A. $T3 \rightarrow T1 \rightarrow T2 \rightarrow T4$
- B. $T3 \rightarrow T2 \rightarrow T1 \rightarrow T4$
- C. $T3 \rightarrow T1 \rightarrow T4 \rightarrow T2$
- D. $T4 \rightarrow T3 \rightarrow T1 \rightarrow T2$



Câu 3: (10 điểm)

Lịch biểu nào dưới đây là khả tuần tự:

- A. $R2(X), R1(X), R1(Y), R3(Z), W3(Z), W2(X), W1(X)$
- B. $R1(Y), R2(X), R1(X), R3(Z), W3(Z), W2(X), W1(X)$
- C. $R1(Z), R2(X), R1(Y), W2(X), R3(Z), W1(Y), W3(Z)$
- D. $R1(Y), R2(X), R1(X), R3(Z), W1(X), W3(Z), W2(X)$

PHẦN II – TỰ LUẬN (70 ĐIỂM)

Câu 4: (30 điểm)

Cho các quan hệ sau:

- DonHang (MaDH, MaNV, MaNhaCC, NgayLap, TongTien)
- ChiTiet (MaDH, MaHang, SoLuong, DonGia, ChietKhau)

và truy vấn Q: Lập danh sách gồm MaHang, SoLuong của các mặt hàng có DonGia > 90000 trong đơn hàng có NgayLap = '29/04/2011'.

- a. Viết biểu thức đại số quan hệ thực hiện truy vấn Q.
- b. Sử dụng phương pháp tối ưu Heuristic để tối ưu hóa Q.

Câu 5: (40 điểm)

Cho các quan hệ toàn cục:

- DonVi(MaDV, TenDV, MaNQL)
- DuAn (MaDA, TenDA, KinhPhi, MaDV)

Giả sử DonVi và DuAn được phân đoạn ngang theo MaDV = 1 và MaDV = 2 thành DonVi1, DonVi2, DuAn1, DuAn2.

- a. Viết biểu thức đại số quan hệ của các đoạn trên.
- b. Viết biểu thức xây dựng lại của 2 quan hệ toàn cục DonVi, DuAn.
- c. Hãy viết truy vấn toàn cục Q_{TC} : Lập danh sách gồm TenDA, TenDV của các dự án có KinhPhi > 900000 và do đơn vị có MaDV = 1 quản lý.
- d. Chuyển truy vấn Q_{TC} ở câu c thành truy vấn đoạn.

-----***HẾT***-----

Lưu ý: - Không sửa, xóa đề thi, nộp lại đề sau khi thi