**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**



**SEMINAR**

**CHƯƠNG 6: QUẢN LÝ GIAO TÁC VÀ ĐIỀU KHIỂN ĐỒNG THỜI**

**TRONG MÔI TRƯỜNG PHÂN TÁN**

**Giảng viên: Thái Bảo Trân**

**Nhóm sinh viên thực hiện:**

Lưu Vĩnh Phát - 20521292

Đỗ Huỳnh Mỹ Tâm - 20520746

Trần Văn Thế -

Nguyễn Minh Duy - 21522005

Mục lục

[**I.** **Nội dung** 3](#_Toc119617504)

[**1.** **Giới thiệu giao tác /Quản lý giao tác:** 3](#_Toc119617505)

[**1.1** **Khái niệm giao tác:** 3](#_Toc119617506)

[**1.2** **Các tính chất của giao tác:** 3](#_Toc119617507)

[**1.3** **Các loại giao tác** 4](#_Toc119617508)

[**2.** **Điều khiển đồng bộ/ đồng thời** 5](#_Toc119617509)

[**2.1** **Điều khiển đồng thời là gì ?** 5](#_Toc119617510)

[**2.2** **Hiện tượng race condition :** 5](#_Toc119617511)

[**2.3** **Các kỹ thuật điều khiển đồng thời** 12](#_Toc119617512)

[**2.3.1 Kỹ thuật sử dụng khóa (lock) cho điều khiển đồng thời trong phân tán** 12](#_Toc119617513)

[**2.3.2 Khóa 2 pha phân tán (Distributed two-phase locking 2PL)** 13](#_Toc119617514)

[**2.3.3 Nghi thức lock 2 giai đoạn** 14](#_Toc119617515)

[**2.3.4 Điều khiển đồng thời dựa vào mốc thời gian (Timestamps** 14](#_Toc119617516)

[**Ordering)** 14](#_Toc119617517)

[**3.** **Phân biệt quản lý giao tác với điều khiển đồng thời so với môi** 16](#_Toc119617518)

[**trường tập trung** 16](#_Toc119617519)

[**II.** **Demo** 17](#_Toc119617520)

1. **Nội dung**
2. **Giới thiệu giao tác /Quản lý giao tác:**
   1. **Khái niệm giao tác:**

Một giao tác là môṭ đơn vị logic xử lý CSDL, nó bao gồm một hoặc nhiều thao tác truy cập dữ liệu như chèn, xoá, sửa đổi hoặc lấy thông tin ra.

Giao tác là một dãy các thao tác cần thực hiện trên cơ sở dữ liệu dưới một đơn vị duy nhất, nghĩa là hoặc thực hiện tất cả các thao tác hoặc không thực hiện thao tác nào cả.

***Ví dụ:***

* Hệ thống giao dịch ngân hàng
* Hệ thống đặt vé máy bay

Các thao tác trong cơ sở dữ liệu cấu thành một giao tác có thể được nhúng vào một chương trình ứng dụng hoặc được chỉ rõ một cách tương tác thông qua một ngôn ngữ truy vấn bậc cao như SQL

Trong ngữ cảnh CSDL, có thể xem sự thực hiện một chương trình ứng dụng như một dãy các giao tác và xen kẽ giữa chúng là những thao tác không đòi hỏi xử lí CSDL (thao tác CPU).

* 1. **Các tính chất của giao tác:**

Để đảm bảo tính nhất quán của dữ liệu, các hệ QTCSDL cần đảm bảo các tính chất sau đây cho giao tác, tính chất ACID

**Tính nguyên tố (Atomicity):** toàn bộ các thao tác trong cùng một giao tác đều được thực hiện hoặc không một thao tác nào trong chúng được thực hiện.

+ Có 2 chiều hướng thực hiện:

Hoặc nó sẽ được kết thúc bằng cách hoàn tất các hành động còn lại.

Hoặc có thể kết thúc bằng cách hồi lại tất cả các hành động đã được thực hiện.

**Tính nhất quán (Consistency):** tính đúng đắn của nó, một giao tác phải chuyển CSDL từ một trạng thái nhất quán này sang một trạng thái nhất quán khác

**Tính cô lập (Isolation):** các giao tác phải được thực hiện một cách độc lập với nhau. Nói một cách khác những tác động của giao tác này sẽ không thể thấy được đối với những giao tác khác, khi giao tác này chưa thực hiện xong hoàn toàn.

**Tính bền vững (Durability):** Để bảo đảm rằng mỗi khi giao tác uỷ thác, kết quả của nó sẽ được duy trì và không bị xoá ra khỏi CSDL.Tính bền vững được sử dụng như là một điều kiện để khôi phục dữ liệu (database recovery), nghĩa là cách khôi phục CSDL về trạng thái nhất quán mà ở đó mọi hành động đã ủy thác đều được phản ánh.

* 1. **Các loại giao tác**

Có 2 loại giao tác:

* Giao tác phẳng:

+ Giao tác phẳng (flat transaction) có một khởi điểm duy nhất (Begin transaction) và một điểm kết thúc duy nhất (Endtransaction).

+ Tất cả các ví dụ của chúng ta đã xem xét đều nằm trong nhóm này.

+ Phần lớn các nghiên cứu về quản lý giao tác trong cơ sở dữ liệu đều tập trung vào các giao tác phẳng.

* Giao tác lồng nhau

+ Đây là mô hình giao tác cho phép một giao tác chứa giao tác khác với điểm bắt đầu và ủy thác của riêng chúng.Những giao tác như thế được gọi là giao tác lồng (nestedtransaction).

+ Những giao tác được đặt vào trong giao tác khác thường được gọi là giao tác con (subtransaction)

- Các giao tác được nhiều người dùng khác nhau đệ trình có thể thực hiện một cách đồng thời và có thể truy cập và cập nhật các mục cơ sở dữ liệu giống nhau.

- Vì vậy, cần phải có quản lý giao tác nếu không có thể sẽ dẫn đến nhiều vấn đề, chẳng hạn như việc cơ sở dữ liệu không nhất quán.

|  |
| --- |
| **QUẢN LÝ GIAO TÁC PHÂN TÁN:**  + Trong CSDL phân tán, database thường được sử dụng bởi nhiều người dùng. Các hệ thống thường cho phép nhiều giao dịch chạy cùng lúc.  + Thực hiện nhiều quy trình trên các máy khác nhau. -> CSDL phân tán cung cấp xử lý giao tác song song, giúp rút gọn độ trễ của thời gian.  + Hệ thống xử lý giao dịch phân tán dành cho các hệ thống không đồng nhất có bộ phận quản lý tài nguyên nhận biết giao dịch.  + Thực hiện giao dịch phân tán yêu cầu sự phối hợp giữa các nhà quản lý tài nguyên cục bộ và hệ thống quản lý giao dịch toàn cầu.  + Hai yếu tố cơ bản của một giao dịch phân tán là trình giám sát xử lý giao dịch (transaction manager) và trình quản lý tài nguyên (resource manager)  + Tính chất ACID cũng được tuân theo bởi gtpt, tuy vậy rất khó để duy trì chúng trong giao tác phân tán vì ở đây bất kì quá trình nào cũng có thể thất bại. |

1. **Điều khiển đồng bộ/ đồng thời**
   1. **Điều khiển đồng thời là gì ?**

Kiểm soát tính đồng thời là một cách thức trong hệ quản trị cơ sở dữ liệu, được sử dụng để quản lý các hoạt động hay sự kiện đồng thời xảy ra trong cơ sở dữ liệu mà không làm xung đột lẫn nhau.

Kiểm soát tính đồng thời đảm bảo rằng các giao dịch được thực hiện một cách an toàn và tuân theo các quy tắc ACID – gồm tính nguyên tố, nhất quán, cô lập, bền vững.

* 1. **Hiện tượng race condition :**

Trong các hệ thống database có hỗ trợ transactions, khi 2 (hay nhiều) transactions đồng thời cùng thực hiện truy cập một (hoặc nhiều) dữ liệu có thể gây ra hiện tượng race conditions như:

1. **Lost update**

Các cập nhật dữ liệu sẽ bị mất khi giao dịch cùng truy cập vào một dữ liệu và cùng cập nhật dữ liệu đó.

Hai giao dịch cùng truy cập vào 1 dữ liệu và cập nhật dữ liệu đó, giao dịch trẻ hơn sẽ ghi đè dữ liệu lên giao dịch cũ, do đó làm mất dữ liệu cập nhật.

***Mô tả tình huống:*** Khi 2 người khác nhau cùng đặt vé một bộ phim (2 hay nhiều vé bởi những người khác nhau) có khả năng nhảy ra mất dữ liệu.

|  |  |
| --- | --- |
| **Transaction T1**  (Người 1 đặt vé trên bộ phim a) | **Transaction T2**  (Người 1 đặt vé trên bộ phim a) |
| SET TRANSACTION ISOLATION LEVEL Read committed;      sp\_datve(1,1,51,51,'ONLINE','The'); |  |
|  | SET TRANSACTION ISOLATION LEVEL Read committed;      sp\_datve(4,1,51,53,'ONLINE','The'); |
| COMMIT; |  |
|  | COMMIT; |
| SELECT DOANHTHU FROM PHIM join LICHCHIEU ON PHIM.MAPHIM = LICHCHIEU.MAPHIM WHERE MALICHCHIEU = 51;  Result:  95000  ---- Với mỗi bộ phim là 95000 thì kết quả đúng phải là 190000 |  |

***Vấn đề xảy ra***: Khi transaction T1 đặt vé update doanh thu nhưng chưa commit dữ liệu. Transaction T2 update ghi đè doanh thu lên bảng phim làm mất dữ liệu.

***Mô tả kết quả:***

* Ví dụ người 1 đặt phim vé trên bộ phim A với giá là 95000 đồng. Người 2 đặt vé trên bộ phim A cũng với giá 95000 đồng.
* Khi cả 2 người đặt xong. Người quản trị thống kê doanh thu phim A thì thấy chỉ có 95000 đồng. (Mất dữ liệu)

***Nguyên nhân:*** Transaction T1 đang trong trạng thái SLEEP. Transaction T2 nhảy vào và update đè lên doanh thu phim. Gây ra tình trạng mất dữ liệu (Lost update)

***Cách khắc phục***: Thay mức cô lập thành Serializable để giải quyết vấn đề Lost update.

1. **Dirty read**

Một giao dịch sẽ đọc được dữ liệu đang được cập nhật bởi một giao dịch khác trong khi nó vẫn chưa hoàn tất

Trong hệ quản trị cơ sở dữ liệu Oracle mức cô lập mặc định là read commited nên không bao giờ xảy ra trường hợp Dirty Read

Khi UPDATE, INSERT hoặc DELETE khỏi giao dịch khác được đọc và giao dịch khác chưa cam kết dữ liệu. Nó đang đọc dữ liệu "đang xử lý", dữ liệu này có thể không hoàn chỉnh và có thể không bao giờ thực sự được cam kết.

***Mô tả tình huống:*** Khi một khách hàng đang đặt một vé nhưng chưa commit. Tài khoản của khách hàng tại một phiên khác xem điểm tích lũy của mình xảy ra trường hợp Unreadtable Read.

***Cách khắc phục***: Thay mức cô lập thành Read Committed hoặc Serializable để giải quyết vấn đề Dirty Read.

Ngày nay, đa số các Hệ quản trị Cơ sở dữ liệu đề đặt mức cô lập mặc định là Read Committed nên trường hợp Dirty Read ít khi xảy ra.

1. **Unrepeateable read**

Một giao dịch đọc dữ liệu lúc chưa cập nhật và dữ liệu có thể bị thay đổi bởi giao dịch khác đang thực hiện việc cập nhật

Xảy ra khi trong quá trình giao dịch, một hàng được truy xuất hai lần và các giá trị trong hàng khác nhau giữa các lần đọc

***Mô tả tình huống:*** Một nhân viên đang xem thông tin của bộ phim cùng lúc đó có một nhân viên khác cập nhật thông tin phim. Nhân viên xem lại thông tin phim một lần nữa hai thì thấy thông tin đã được thay đổi. (Trường hợp Unrepeatable Read).

|  |  |
| --- | --- |
| **Transaction T1**  (Nhân viên xem thông tin phim) | **Transaction T2**  (Nhân viên cập nhật thông tin phim) |
| SET TRANSACTION ISOLATION LEVEL read committed;      SP\_INTHONGTINPHIM(4);  COMMIT;      SLEEP(10);  **Result:**  Bo phim can tim là  \*\* Tên phim: Đẳng Cấp Thú Cưng (Dao dien: Chris Renaud)  \*\* Ngày phát hành: 07-06-2019 Nhà phát hành: CGV  Danh sách dien viên tham gia  Lake Bell, Hannibal Buress, Dana Carvey, Harrison Ford,...  \*\* Doanh thu phim: 0  ----- Lich chieu: 12 PHONG SO: 1  ----- Lich chieu: 22 PHONG SO: 21  ----- Lich chieu: 48 PHONG SO: 1  ----- Lich chieu: 51 PHONG SO: 1 |  |
|  | BEGIN  UPDATE\_PHIM(4,'DCTC',1,'ABC','DEF');  COMMIT  END; |
| SP\_INTHONGTINPHIM(4);  COMMIT;  **Result:**  Bo phim can tim là  \*\* Tên phim: DCTC (Dao dien: ABC)  \*\* Ngày phát hành: 07-06-2019 Nhà phát hành: CGV  Danh sách dien viên tham gia  DEF  \*\* Doanh thu phim: 0  ----- Lich chieu: 12 PHONG SO: 1  ----- Lich chieu: 22 PHONG SO: 21  ----- Lich chieu: 48 PHONG SO: 1  ----- Lich chieu: 51 PHONG SO: 1 |  |

***Vấn đề xảy ra:*** Hai lần xem thông tin doanh thu phim cho ra hai kết quả phim khác nhau.

***Mô tả kết quả:*** Như trên tình huống

***Nguyên nhân:*** Đây là vấn đề Unrepeatable Read khi một transaction T1 đọc dữ liệu 2 lần, transaction T2 cập nhật dữ liệu giữa hai lần đọc. Gây ra cho hai kết quả khác nhau.

***Cách khắc phục:*** Thay mức cô lập SET TRANSACTION ISOLATION LEVEL READ COMMITTED thành mức cô lập SET TRANSACTION ISOLATION LEVEL Serializable.

1. **Phantom read**

Một giao dịch vừa thực hiện xong việc đọc (chưa commit) thì giao dịch khác lại thay đổi (ghi) trên chính đơn vị dữ liệu đó khiến kết quả lần đọc sau bị thay đổi.

Đọc ảo xảy ra trong quá trình giao dịch, hai truy vấn giống hệt nhau được thực hiện và tập hợp các hàng được trả về bởi truy vấn thứ hai khác với truy vấn đầu tiên.

Mô tả tình huống: Khi một nhân viên đang xem bộ phim đó có bao nhiêu lịch chiếu, thì một nhân viên khác lại thêm một lịch chiếu của bộ phim đó. Nhân viên xem lịch chiếu của bộ phim lúc đầu xem lại lần nữa thì thấy nhiều hơn dữ liệu. Đây là trường hợp Phantom Read.

|  |  |
| --- | --- |
| **Transaction T1**  (Nhân viên xem thông tin phim) | **Transaction T2**  (Nhân viên thêm lịch chiếu) |
| SET TRANSACTION ISOLATION LEVEL read committed;      SP\_INTHONGTINPHIM(4);  COMMIT;      SLEEP(10);  **Result:**  Bo phim can tim là  \*\* Tên phim: Đẳng cấp thú cưng (Dao dien: Chris Renaud)  \*\* Ngày phát hành: 07-06-2019 Nhà phát hành: CGV  Danh sách dien viên tham gia  Lake Bell, Hannibal Buress, Dana Carvey, Harrison Ford,...  \*\* Doanh thu phim: 190000  ----- Lich chieu: 12 PHONG SO: 1  ----- Lich chieu: 22 PHONG SO: 21  ----- Lich chieu: 48 PHONG SO: 1  ----- Lich chieu: 51 PHONG SO: 1 |  |
|  | BEGIN      Insert\_LICHCHIEU(22,4,2);  END; |
| SP\_INTHONGTINPHIM(4);  COMMIT;  **Result:**  Bo phim can tim là  \*\* Tên phim: Đẳng cấp thú cưng (Dao dien: Chris Renaud)  \*\* Ngày phát hành: 07-06-2019 Nhà phát hành: CGV  Danh sách dien viên tham gia  Lake Bell, Hannibal Buress, Dana Carvey, Harrison Ford,...  \*\* Doanh thu phim: 190000  ----- Lich chieu: 12 PHONG SO: 1  ----- Lich chieu: 22 PHONG SO: 21  ----- Lich chieu: 48 PHONG SO: 1  ----- Lich chieu: 51 PHONG SO: 1  ----- Lich chieu: 62 PHONG SO: 2 |  |

***Vấn đề xảy ra:*** Hai lần xem đọc thông tin lịch chiếu cho ra hai kết quả khác nhau. Lần thứ 2 nhiều hơn lần thứ nhất

***Mô tả kết quả:*** Như trên tình huống

***Nguyên nhân:*** Đây là vấn đề Phantom Read khi một transaction T1 đọc dữ liệu 2 lần, transaction T2 cập nhật dữ liệu giữa hai lần đọc. Lần thứ 2 cho ra kết quả nhiều hơn lần đọc thứ nhất.

***Cách khắc phục:*** Thay mức cô lập thành SET TRANSACTION ISOLATION LEVEL Serializable

1. **Deadlock**

Mỗi giao dịch đang nắm giữ khoá của riêng mình và chờ khoá của giao dịch kia, khi đó cả hai giao dịch chờ đợi vô hạn định.

***Mô tả tình huống:*** Khi một nhân viên 1 update thông tin bộ phim 1 chưa commit, nhân viên 2 update thông tin của bộ phim 2 chưa commit. Sau đó nhân viên 1 lại update thông tin bộ phim 2 và nhân viên 2 lại update thông tin của bộ phim 1. Gây ra trường hợp deadlock.

|  |  |
| --- | --- |
| **Transaction T1**  (Update movie 2 then movie 1) | **Transaction T2**  (Update  1 rồi phim 2) |
| /\* Call api PATCH ${baseUrl}/phim/2/\*/      SLEEP(10); |  |
|  | /\* Call api PATCH ${baseUrl}/phim/1 \*/      SLEEP(10); |
| /\* Call api PATCH ${baseUrl}/phim/1 \*/  COMMIT; |  |
|  | /\* Call api PATCH ${baseUrl}/phim/2 \*/ |
| **Result:**  Procedure UPDATE\_PHIM compiled | **Result:**  Error report -  ORA-00060: deadlock detected while waiting for resource  00060. 00000 -  "deadlock detected while waiting for resource"  \*Cause:    Transactions deadlocked one another while waiting for resources.  \*Action:   Look at the trace file to see the transactions and resources involved. Retry if necessary. |

***Vấn đề xảy ra:***  Transaction 1 dự khóa trên đơn vị dữ liệu của bộ phim 4, chờ khóa trên đơn vị dữ liệu của bộ phim 1 và Transaction 2 dự khóa trên đơn vị dữ liệu của bộ phim 1, chờ khóa trên đơn vị dữ liệu của bộ phim 2.

***Mô tả kết quả***: Như hình trên.

***Nguyên nhân:*** Hai transaction chờ khóa vô hạn định gây ra tình trạng deadlock.

***Cách khắc phục:*** Các hệ quản trị cơ sở dữ liệu hiện đại sẽ tự động Rollback transaction bị lỗi.

**2.3 Các kỹ thuật điều khiển đồng thời**

Hầu hết các phương thức điều khiển đồng thời phân tán sẽ gồm loại cơ bản:

× Kỹ thuật sử dụng khóa cho điều khiển đồng thời

× Điều khiển đồng thời dựa vào timestamps ordering (thứ tự nhãn/ tem thời gian)

× Kỹ thuật sắp xếp theo lịch

**2.3.1 Kỹ thuật sử dụng khóa (lock) cho điều khiển đồng thời trong phân tán**

- Tập các giao thức quan trọng đầu tiên là dùng kỹ thuật khóa (lock) để ngăn chặn nhiều giao tác truy cập đồng thời vào mục dữ liệu, đảm bảo tính tuần tự.

- Một phương pháp để đảm bảo tính khả năng tuần tự là yêu cầu việc truy cập đến các mục dữ liệu được tiến hành theo kiểu loại trừ lẫn nhau; có nghĩa là trong khi một giao tác đang truy cập một mục dữ liệu, không một giao tác nào khác có thể sửa đổi mục dữ liệu này.

- Lock là đặc quyền truy xuất (Access privilege) lên các đơn vị dữ liệu của các thao tác mà bộ quản lý khoá có thể trao đổi cho một giao tác hay thu hồi lại.

- Khi một thao tác đã khoá (lock) trên một đơn vị dữ liệu thì các thao tác khác không được phép truy cập đến đơn vị dữ liệu đó cho đến khi nó nhả khoá (unlock)

- Ý tưởng chính của kĩ thuật này để đảm bảo dữ liệu mà được chia sẻ bởi nhiều hệ thống đang mâu thuẫn thì được truy cập vào một hệ thống cùng 1 thời điểm

***- Cách thức hoạt động:***

+ Trước khi nó hoạt động , khóa này được thiết lập bởi giao tác và được reset vào lúc kết thúc giao tác đó.

+ Có 2 loại khóa là khóa đọc (read lock) và khóa ghi (write lock)

+ Trong một hệ thống phân tán thì có các trang web được chỉ định làm bộ phận quản lý khóa.

+ Trình quản lý khóa sẽ kiểm soát các yêu cầu khóa từ các trình giám sát giao dịch. ( CC Manager)

+ Để thực thi sự phối hợp giữa những bộ phận quản lý khóa trong các trang web khác nhau thì ít nhất một trang web phải được cấp quyền xem tất cả các giao dịch và phát hiện xung đột khóa.

**2.3.2 Khóa 2 pha phân tán (Distributed two-phase locking 2PL)**

- Ở các hệ quản trị CSDL thì Serializability là mức isolation (chữ cái I trong ACID) mà ở đó Database đảm bảo kết quả khi thực hiện đồng thời các transactions sẽ tương tự như khi các transactions đó được thực hiện một cách tuần tự.

- Để hỗ trợ serializability, 2PL là thuật toán được lựa chọn để cài đặt.

- Ý tưởng chính của 2PL là chia một transaction theo thứ tự lần lượt:

+ Kỳ xin khóa (Growing phase): Một giao tác có thể nhận được các khoa nhưng nó không thể mở bất kỳ khóa nào

+ Kỳ khóa (Locked phase): Giao dịch có đủ các khóa nó cần và thực hiện các thao tác. Khi giao dịch giải phóng khóa đầu tiên của nó, kỳ thứ ba sẽ bắt đầu.

+ Kỳ mở khóa (Shrinking phase): Một giao tác có thể mở các khóa nhưng không thể nhận được một khóa mới nào.

**2.3.3 Nghi thức lock 2 giai đoạn**

Một giao tác thực hiện cơ chế lock 2 phase là một giao tác không thực hiện một lock nào nữa sau khi đã unlock (thực hiện xong hết tất cả các yêu cầu lock rồi mới tiến hanh unlock). Các giao tác tuân theo nghi thức này được gọi là các giao tác hai pha: pha ban đầu là pha khoa, pha sau là pha mở khóa.

- Ý nghĩa:

+ Khóa 2 kỳ có thể hạn chế số lượng giao tác đồng thời mà có thể xuất hiện trong một lịch biểu

+ Các giao dịch 2PL sẽ nhận được tất cả các khóa cần thiết trước khi chuyển sang trạng thái mở khóa.

+ 2PL đảm bảo khả năng tuần tự hóa của dữ liệu nhưng nó không đảm bảo rằng deadlock hay khóa chết sẽ không xảy ra.

**2.3.4 Điều khiển đồng thời dựa vào mốc thời gian (Timestamps** **Ordering)**

Time Stamp về cơ bản là một phương pháp thừa nhận hoặc ghi chú thời gian tương ứng của một thời điểm đặc biệt đáng chú ý hay nói đơn giản nó như một con dấu công chứng xác minh của bên thứ ba rằng chữ ký là hợp pháp tại thời điểm nó được thực hiện.

Nhãn thời gian của đơn vị dữ liệu là nhãn thời gian của các giao tác cuối cùng có truy cập đến đơn vị dữ liệu có thành công

+ Để thiết lập thứ tự, bộ quản lý giao tác sẽ giao cho từng giao tác một cái Tem thời gian (timestamp) độc nhất

- **Tem thời gian (timestamp)** là định danh duy nhất được tạo ra bởi DBMS để nhận dạng một giao tác. Các giá trị đặc trưng của tem thời gian được gán theo trật tự các giao tác được đưa vào hệ thống xem xét, bởi vậy tem thời gian có thể được nghĩ đến như thời gian bắt đầu giao tác.

- Nhãn thời gian của đơn vị dữ liệu là nhãn thời gian của các giao tác cuối cùng có truy cập đến đơn vị dữ liệu có thành công, hay là nhãn thời gian cao nhất trong số các giao tác có truy câph thành công đến đơn vị dữ liệu đó

- Tem thời gian có thể được tạo ra theo vài cách. Cách thứ nhất là sử dụng bộ đếm và giá trị được tăng lên mỗi khi gán cho một giao tác.

- Tuy nhiên, việc duy trì các biến đếm toàn cục là một vấn đề trong cơ sở dữ liệu phân tán do đó thích hợp hơn là giao cho các timestamp dựa trên các biến cục bộ

- Trong một hệ thống tập trung, timestamp của bất kỳ giao dịch nào được xác định bằng cách đọc đồng hồ vật lý. Tuy nhiên, trong một hệ thống phân tán, các số đọc đồng hồ logic / vật lý cục bộ của bất kỳ trang web nào không thể được sử dụng làm timestamp toàn cầu, vì có nhiều mốc thời gian khác nhau trên toàn cầu, không thể sử dụng chung

- Vì vậy, tem thời gian bao gồm sự kết hợp của ID trang web và số đọc đồng hồ của trang web đó.

🡪 Để triển khai các thuật toán sắp xếp timestamp, mỗi trang web có một bộ lập lịch để duy trì một hàng đợi riêng cho mỗi bộ phận quản lý giao tác. Trong quá trình giao dịch, bộ phận quản lý giao dịch sẽ gửi yêu cầu khoa đến bộ lập lịch của trang web. Bộ lập lịch đưa yêu cầu vào hàng đợi tương ứng theo thứ tự timestamp tăng dần. Yêu cầu được xử lý từ phía trước hàng đợi theo thứ tự timestamp của chúng, tức là yêu cầu cũ nhất đầu tiên

🡪 Giao thức này đảm bảo rằng mọi hoạt động đọc và ghi xung đột đều được thực thi theo thứ tự các mốc thời gian. Giao thức sử dụng Phép tính thời gian của hệ thống hoặc Phép đếm logic làm mốc thời gian. Giao dịch vào trước luôn được ưu tiên hơn trong phương pháp này. Nó sử dụng thời giancủa hệ thống để xác định các mốc thời gian của giao dịch.

***Thuật toán sắp xếp tem thời gian cơ bản - Basic timestamp ordering (BTO)***

+ BTO liên kết timestamp với tất cả các mục dữ liệu được truy cập gần đây và yêu cầu các giao dịch truy cập dữ liệu xung đột được thực hiện theo thứ tự timestamp thay vì sử dụng phương pháp khóa

+ Trong trường hợp dữ liệu sao chép, phương pháp "write all" và "read any" được sử dụng, do đó yêu cầu ghi phải được gửi đến tất cả các bản sao trong khi yêu cầu đọc có thể được gửi đến bất kỳ bản sao nào.

🡪 Tóm gọn: Quy luật thuật toán timestamp

• Quy tắc Truy cập - Khi hai giao dịch cố gắng truy cập đồng thời vào cùng một mục dữ liệu, đối với các hoạt động xung đột, giao dịch cũ hơn sẽ được ưu tiên. Điều này khiến giao dịch trẻ hơn phải đợi giao dịch cũ hơn để cam kết trước.

• Quy tắc giao dịch muộn - Nếu một giao dịch trẻ hơn đã ghi một mục dữ liệu, thì một giao dịch cũ hơn không được phép đọc hoặc ghi mục dữ liệu đó. Quy tắc này ngăn giao dịch cũ hơn cam kết sau khi giao dịch trẻ hơn đã được cam kết.

• Quy tắc giao dịch trẻ hơn - Một giao dịch trẻ hơn có thể đọc hoặc ghi một mục dữ liệu đã được ghi bởi một giao dịch cũ hơn.

**3.** **Phân biệt quản lý giao tác với điều khiển đồng thời so với môi** **trường tập trung**

● Nếu HQT CSDL trong môi trường phân tán là bao gồm 1 CSDL mà được phân chia thành nhiều mảnh . Mà trong đó mỗi mảnh được tích hợp bởi 1 hoặc nhiều máy tính và điều khiển bởi 1 CSDL độc lập (DBMS) thì Hệ CSDL mt tập trung thì dữ liệu được lưu trữ trên máy tính trung tâm, nhiều người sử dụng từ xa có thể truy cập CSDL này thông qua các thiết bị đầu cuối.

● Toàn bộ CSDL được lưu ở một dàn máy tính (có thể bao gồm một hoặcnhiều máy được điều hành chung như một máy).

● Các hệ CSDL trung tâm thường rất lớn và có nhiều người dùng, ví dụ các hệ thống đăng kí và bán vé máy bay, các hệ thống thông tin của tổ chức tài chính,…

● Ở môi trường tập trung thì việc quản lý giao tác cũng như điều khiển đồng thời cũng sẽ ít phức tạp hơn ở môi trường phân tán.

● Đối với việc quản lý giao tác ở môi trường tập trung thì vẫn đảm bảo 4 tính chất ACID về (tính nguyên tử ,nhất quán, cô lập và tính bền vững) của các giao tác

● Đối với điều khiển đồng thời ở môi trường tập trung thì vẫn đảm bảo bảo các giao tác diễn ra đồng thời mà không xảy ra xung đột với nhau, để kết quả sau cùng được tạo ra là chính xác và không vi phạm tính toàn vẹn của dữ liệu. Sử dụng cơ chế khóa, nhãn thời gian để các giao tác diễn ra song song với nhau.

|  |  |  |
| --- | --- | --- |
|  | Quản lí giao tác | Điều khiển đồng thời |
| Khái niệm | là một đơn vị xử lý logic trong DBMS đòi hỏi một hoặc nhiều hoạt động truy cập cơ sở dữ liệu | Là một Procedure đề quản lý các giao tác diễn ra đồng thời |
| Mục đích | Dùng để đảm bảo tính chất ACID của các giao tác | Đảm bảo các giao tác diễn ra đồng thời mà không xảy ra xung đột với nhau, để kết quả sau cùng được tạo ra là chính xác và không vi phạm tính toàn vẹn của dữ liệu |
| Tầm quan trọng | Thiếu quản lí giao tác có thể gây lỗi phần cứng và sập hệ thống | Thiếu điều khiển đồng thời có thê gây ra Lost Updates, Dirty Read, Non-Repeatable Read, Phantom Read,… |
| Cách thức hoạt động | Tạo ra lịch biểu để các giao tác hoạt động tuần tự nhau | Sử dụng cơ chế khóa, nhãn thời gian để các giao tác diễn ra song song với nhau. |

1. **Demo**
2. **Chuẩn bị**
3. ***Môi trường sử dụng:***

Hệ điệu hành Windows 10

Hệ quản trị cơ sở dữ liệu SQL SERVER 2019

1. ***Cơ sở dữ liệu: QLVT (Quản lý vật tư)***

Các bảng trong cơ sở dữ liệu