CHƯƠNG 4

Lương Vũ Đình Duy - 21133018

1. Transaction là gì?

* Transaction được định nghĩa là bất kỳ một lần thực thi chương trình người dùng nào trong DBMS và khác với việc thực thi chương trình bên ngoài DBMS.

1. Transaction giống hay Transaction khác với một chương trình được thực thi bên ngoài DBMS

* Thực thi: Transaction được thực thi bên trong DBMS, trong khi chương trình được thực thi bên ngoài DBMS. Điều này có nghĩa là transaction có thể truy cập và sửa đổi dữ liệu trực tiếp trong cơ sở dữ liệu, trong khi chương trình phải truy cập dữ liệu thông qua một API.
* Kiểm soát: Transaction được kiểm soát bởi DBMS, trong khi chương trình được kiểm soát bởi người lập trình. Điều này có nghĩa là DBMS chịu trách nhiệm đảm bảo rằng transaction được thực thi thành công hoặc thất bại toàn bộ. Người lập trình chịu trách nhiệm đảm bảo rằng chương trình được thực thi đúng cách.
* Thuộc tính: Transaction có các thuộc tính ACID, trong khi chương trình không có. Điều này có nghĩa là transaction đảm bảo tính toàn vẹn dữ liệu, ngay cả khi có lỗi xảy ra trong quá trình thực thi. Chương trình không có các thuộc tính ACID, do đó dữ liệu có thể bị hỏng nếu có lỗi xảy ra.

1. Phát biểu nào dưới đây đúng
2. DBMS xử lý transaction như một loạt các thao tác đọc ghi các đối tượng CSDL
3. Thao tác đọc một đối tượng CSDL là đưa đối tượng đó từ đĩa vào bộ nhớ (buffer bool), sau đó copy giá trị của nó vào biến trong chương trình.
4. Thao tác ghi một đối tượng CSDL là ghi bản sao của đối tượng trong bộ nhớ lên đĩa
5. Tất cả ý trên
6. Nêu 4 tính chất quan trọng của transaction? Cho ví dụ

* **Atomicity** (nguyên tử) : Một transaction phải được thực hiện như một đơn vị nguyên tử. Điều này có nghĩa là tất cả các câu lệnh trong một transaction phải thành công hoặc thất bại cùng nhau.
* **Consistency** (nhất quán): Các ràng buộc cơ sở dữ liệu được bảo tồn. Database phải ở trạng thái nhất quán sau khi thực hiện một transaction. Điều này có nghĩa là tất cả các dữ liệu trong database phải tuân theo các ràng buộc toàn vẹn.
* **Isolation** (cách ly): Các transaction phải được thực hiện cách ly với nhau. Điều này có nghĩa là các thay đổi của một transaction không được nhìn thấy bởi các transaction khác cho đến khi transaction đó hoàn thành..
* **Durability** (độ bền): một transaction đã được hoàn thành thành công, ảnh hưởng của nó sẽ vẫn tồn tại ngay cả khi hệ thống gặp sự cố trước khi tất cả các thay đổi của nó được phản ánh trên đĩa. Các thay đổi của một transaction phải được lưu trữ vĩnh viễn. Điều này có nghĩa là các thay đổi của một transaction phải được ghi vào database ngay cả khi có sự cố xảy ra

**Ví dụ:**

|  |
| --- |
| READ(A)  A := A – 50  WRITE(A)  READ(B)  B := B + 50  WRITE(B) |

* + Yêu cầu tính nhất quán: tổng của A và B không thay đổi khi thực hiện transaction.
  + Yêu cầu về tính nguyên tử: nếu transaction không thành công sau bước 3 và trước bước 6, hệ thống phải DBMS phải đảm bảo là kết quả các thao tác cập nhật của transaction không được ghi lên CSDL, nếu không sẽ dẫn đến sự không nhất quán.
  + Yêu cầu về độ bền: một khi người dùng đã được thông báo rằng transaction đã hoàn tất (tức là việc chuyển 50 đô la đã diễn ra), các thông tin bản cập nhật cho cơ sở dữ liệu của transaction phải tồn tại mặc dù không thành công hay gặp lỗi.
  + Yêu cầu cách ly: nếu giữa bước 3 và 6, một transaction khác được phép truy cập cơ sở dữ liệu được cập nhật một phần, nó sẽ thấy cơ sở dữ liệu không nhất quán (tổng A + B sẽ nhỏ hơn mức cần thiết).Có thể được đảm bảo tầm thường bằng cách chạy các transaction tuần tự, đó là lần lượt. Tuy nhiên, thực hiện nhiều transaction đồng thời có những lợi ích đáng kể, như chúng ta sẽ thấy.

1. Cho 1 chương trình gồm các hành động theo thứ tự sau: (B1) Read(A); (B2)A = A – 50; (B3) Write(A); (B4) Read(B); (B5) B = B + 50; (B6) Write(B). Phát biểu nào dưới đây mô tả tính nguyên tố, tính nhất quán, tính cô lập, tính bền vững
2. Tổng A và B không được thay đổi bởi sự thực thi chương trình

* Tính nhất quán

1. Nếu transaction không thành công sau bước 3 và trước bước 6, DBMS phải đảm bảo là kết quả các thao tác cập nhật của transaction không được ghi lên CSDL.

* Tính nguyên tố

1. Khi người dùng đã được thông báo là đã chuyển 50$ sang B thành công thì các cập nhật lên CSDL bởi transaction phải vẫn tồn tại dù hệ thống có gặp sự cố.

* Tình bền vững

1. Hai người thực thi chương trình trên đồng thời thì mỗi người vẫn thấy như chỉ có một mình chạy chương trình đó.

* Tính cô lập

1. Hệ thống cho phép nhiều transaction chạy đồng thời để:
2. Tăng khả năng sử dụng bộ xử lý và ổ đĩa: một transaction có thể đang sử dụng CPU trong khi một transaction khác đang đọc hoặc ghi vào đĩa.
3. Giảm thời gian phản hồi trung bình của các transaction: các transaction ngắn không cần chờ đợi sau các transaction dài
4. Cả 2 ý trên
5. Phát biểu nào dưới đây đúng về Lịch biểu
6. Là một lịch trình thực thi các câu lệnh trong các transaction khác nhau theo thời gian
7. Một lịch biểu cho một tập các transaction phải bao gồm tất cả các câu lệnh trong các transaction đó.
8. Lịch biểu phải duy trì thứ tự thực hiện các câu lệnh trong một transaction riêng biệt.
9. Tất cả ý trên
10. Lịch biểu tuần tự là gì?

* Lịch trình trong đó các hành động của các transaction khác nhau không xen kẽ. Nói cách khác, các transaction được thực hiện tuần tự, một transaction sau khi transaction trước đó hoàn thành.

1. Cho 2 lịch biểu L1 và L2. L1 được nói là tương đương với L2 nếu
2. Tác động của việc thực thi L1 lên CSDL giống với tác động của việc thực thi L2 lên CSDL
3. Kết quả của việc thực thi L1 giống với kết quả thực thi L2
4. Cả 2 ý trên
5. Lịch biểu khả tuần tự?

* Lịch trình tương đương với một số thực hiện nối tiếp của các transaction. Nói cách khác, có thể sắp xếp lại các transaction trong lịch trình thành một chuỗi nối tiếp mà vẫn giữ nguyên hiệu ứng của lịch trình.

1. Cho 2 transaction Ti và Tj. Hành động Li thuộc Ti và Lj thuộc Tj. Phát biểu nào dưới đây đúng
2. Li và Lj là 2 hành động xung đột nếu Li và Lj cùng truy cập đến một đối tượng CSDL
3. Li và Lj là 2 hành động xung đột nếu ít nhất một trong 2 hành động Li và Lj là hành động ghi
4. Li và Lj là 2 hành động xung đột nếu Li và Lj cùng truy cập đến một đối tượng CSDL và ít nhất một trong 2 hành động Li và Lj là hành động ghi
5. Lịch biểu L1 có 2 transaction gồm các hành động theo thứ tự sau T1: R(X), T1:R(Y), T2:R(X), T1:W(X), T2:W(X), T1:W(Y). Phát biểu nào dưới đây đúng
6. T1: R(X) và T2:R(X) là 2 hành động không xung đột
7. T2:R(X) và T1:W(X) là 2 hành động xung đột
8. T2:W(X) và T1:W(Y) là 2 hành động không xung đột
9. Phát biểu nào dưới đây đúng
10. Nếu đổi thứ tự thực hiện 2 hành động xung đột thì kết quả của lịch biểu sẽ thay đổi
11. Nếu đổi thứ tự thực hiện 2 hành động không xung đột thì kết quả của lịch biểu sẽ không thay đổi
12. Cả 2 ý trên
13. Định nghĩa 2 lịch biểu L1 và L2 được gọi là tương đương xung đột

* L1 và L2 được gọi là tương đương xung đột khi từ L1 bằng 1 loạt các phép hoán đổi lệnh không xung đột và trở thành L2

1. Định nghĩa lịch biểu khả tuần tự xung đột

* Lịch biểu khả tuần tự xung đột là lịch trình có thể xung đột tương đương với lịch trình nối tiếp

1. Lịch biểu

T1:R(X), T2:R(Y), T1: R(Y), T2:W(Y), T2:R(X), T1:W(X), T1:W(Y), T2:W(X) tương đương xung đột với lịch biểu nào dưới đây:

1. T1:R(X), T2:R(Y), T1: R(Y), T2:W(Y), T2:R(X), T1:W(X), T2:W(X), T1:W(Y)
2. T2:R(Y), T1:R(X), T1: R(Y), T2:W(Y), T2:R(X), T1:W(X), T1:W(Y), T2:W(X)
3. Cả 2 ý trên vì cả 2 đều hoán đổi các lệnh không xung đột
4. Cho lịch biểu T1:R(X), T2:R(Y), T1: R(Y), T2:W(Y), T2:R(X), T1:W(X), T1:W(Y), T2:W(X). Lịch biểu trên là lịch biểu gì

* Lịch biểu tương đương xung đột

1. Cho lịch biểu T1:R(X), T2:R(Y), T1: W(X), T2:R(X), T2:W(X), T2:W(Y). Phát biểu nào dưới đây đúng về lịch biểu này
2. Lịch biểu này khả tuần tự xung đột
3. Lịch biểu này khả tương đương xung đột
4. Cả 2 ý trên
5. Lịch biểu T1:R(X), T2:R(Y), T1: W(X), T2:R(X), T2:W(X), T2:W(Y) tương đương xung đột với lịch biểu nào dưới đây
6. T1:R(X), T1: W(X), T2:R(Y), T2:R(X), T2:W(X), T2:W(Y)
7. T2:R(Y), T1:R(X), T1: W(X), T2:R(X), T2:W(X), T2:W(Y)
8. Cả 2 ý trên
9. Định nghĩa 2 Lịch biểu tương đương view

* S và S 'là hai lịch trình với cùng một tập hợp các transaction. S và S' được xem tương đương view nếu đáp ứng ba điều kiện sau:
* Đối với mỗi mục dữ liệu Q, nếu transaction Ti đọc giá trị ban đầu của Q trong biểu S, thì transaction Ti (trong biểu S') cũng phải đọc giá trị ban đầu của Q.
* Đối với mỗi mục dữ liệu Q, nếu transaction Ti thực hiện read(Q) trong biểu S và giá trị đó được tạo ra bởi transaction Tj (nếu có), thì transaction Ti trong biểu S 'cũng phải đọc giá trị của Q được tạo ra bởi transaction Tj.
* Đối với mỗi mục dữ liệu Q, transaction (nếu có) thực hiện thao tác write(Q) cuối cùng trong phụ lục S phải thực hiện thao tác write(Q) cuối cùng trong biểu S'.

1. Định nghĩa lịch biểu khả tuần tự view

* Lịch trình có thể xem tuần tự hóa nếu nó được xem tương đương với lịch nối tiếp.

1. Giải thích các khái niệm ghi mù (blind write), đọc bẩn (dirty read)

* Ghi mù: Transaction thực hiện write đối tượng nhưng trước đó nó không read đuọc đối tượng đó
* Đọc bẩn: Xảy ra khi một Transaction đọc một đối tượng mà đối tượng này đang được thay đổi giá trị bằng một Transaction chưa thành công khác.

1. Lịch biểu: T1:R(X), T2:R(Y), T1:W(X), T2:W(Y) tương đương view với lịch biểu nào dưới đây
2. T1:R(X), T1:W(X), T2:R(Y), T2:W(Y)
3. T1:R(X), T2:R(Y), T2:W(Y), T1:W(X)
4. ~~T1:R(Y), T2:R(X),~~ T1:W(X), T2:W(Y)
5. Lịch biểu: T1:R(X), T2:R(Y), T1:W(X), T2:W(Y) tương đương xung đột với lịch biểu nào dưới đây
6. T1:R(X), T1:W(X), T2:R(Y), T2:W(Y)
7. T1:R(X), T2:R(Y), T2:W(Y), T1:W(X)
8. ~~T1:R(Y), T2:R(X)~~, T1:W(X), T2:W(Y)
9. Lịch biểu nào dưới đây có hành động ghi mù
10. T1:R(X), T1:W(X), T2:R(Y), T2:W(Y)
11. T1:R(X), T2:R(Y), T2:W(Y), T1:W(X)
12. T1:R(Y), T2:R(X), T1:W(X), T2:W(X)
13. Định nghĩa lịch biểu được gọi là khả phục hồi

* Một transaction Tj đọc một mục dữ liệu được viết trước đó bởi một transaction Ti , hoạt động cam kết của Ti phải xuất hiện trước hoạt động cam kết của Tj thì lịch biểu đó khả phục hồi

1. Lịch biểu nào dưới đây không khả phục hồi
2. T1:R(X), T1:W(X), T2:R(Y), T2:W(Y), T1: commit, T2: commit
3. T1:R(X), T2:R(X), T1: W(X), T2:W(X), T1: commit, T2: commit
4. T1:R(X), T1:W(X), T2:R(X), T2:W(X), T2: commit, T1: commit
5. T1:R(X), T1:W(X), T2:R(X), T2:W(X), T1: commit, T2: commit
6. Phát biểu nào dưới đây đúng
7. Trong 1 lịch biểu khi 1 transaction thực hiện không thành công dẫn đến phải phục hồi lại các transaction khác thì lịch biểu đó được gọi là khả phục hồi lan truyền
8. Phục hồi lan truyền có thể dẫn đến phải phục hồi một lượng đáng kể công việc.
9. Cần phải tránh việc phục hồi lan truyền
10. Tất cả các ý trên
11. Lịch biểu nào dưới đây dẫn đến việc phục hồi lan truyền
12. T1:R(X), T1:W(X), T2:R(X), T2:W(X), T1: commit, T2: commit
13. T1:R(X), T1:W(X), T2:R(Y), T2:W(Y), T1: commit, T2: commit
14. T1:R(X), T1:W(X), T2:R(X), T2:W(X), T2: commit, T1: commit
15. T1:R(X), T2:R(X), T1: W(X), T2:W(X), T1: commit, T2: commit
16. Một lịch biểu là khả phục hồi không lan truyền nếu
17. Transaction Tj đọc đối tượng X mà trước đó X được ghi bởi Ti thì hành động commit của Ti phải thực hiện trước hành động đọc X của Tj
18. Transaction Tj đọc đối tượng X mà trước đó X được ghi bởi Ti thì hành động commit của Tj phải thực hiện trước commit của Ti 🡪 khả phục hồi lan truyền
19. Lịch biểu nào dưới đây dẫn đến việc phục hồi không lan truyền
20. T1:R(X), T1:W(X), T2:R(X), T2:W(X), T1: commit, T2: commit
21. T1:R(X), T1:W(X), T1: commit, T2:R(X), T2:W(X), T2: commit
22. T1:R(X), T1:W(X), T2:R(X), T1: commit, T2:W(X), T2: commit
23. Phát biểu nào dưới đây đúng với nghi thức khóa 2 giai đoạn (2 phase locking)
24. Dùng để quản lý sự truy cập đồng thời
25. Khi một transaction muốn đọc (hoặc chỉnh sửa) một đối tượng, đầu tiên nó phải yêu cầu share lock (hoặc exclusive lock) trên đối tượng đó,
26. Chỉ khi nhận được lock transaction mới đọc (hoặc chỉnh sửa đối tượng)
27. Tất cả các ý trên
28. Phát biểu nào dưới đây đúng
29. Trong nghi thức khóa 2 giai đoạn nghiêm ngặt, tất cả khóa được giữ bởi 1 transaction sẽ được giải phóng khi transaction đó đã hoàn thành
30. Trong nghi thức khóa 2 giai đoạn không nghiêm ngặt, các khóa được giữ bởi 1 transaction có thể được giải phóng bất kỳ lúc nào
31. Cả 2 ý trên
32. Phát biểu nào dưới đây đúng
33. Trong SQL Server có 3 loại transaction: tự động commit, tường minh và không tường minh
34. Trong SQL Server có 2 loại transaction: Tường minh và không tường minh
35. Trong SQL Server chỉ có 1 loại transaction duy nhất
36. Phát biểu nào dưới đây đúng về Autocommit transaction
37. Là transaction mặc định
38. Mỗi câu lệnh T-SQL được xem là 1 transaction. Chúng được commit hay rollback tùy vào kết quả của chúng.
39. Cả 2 ý trên
40. Phát biểu nào dưới đây đúng về Implicit transaction
41. Được kích hoạt bằng câu lệnh SET IMPLICIT\_TRANSACTIONS ON
42. Khi đã được kích hoạt, SQl server sẽ bắt đầu transaction bất cứ khi nào gặp câu lệnh thao tác dữ liệu (DML-Data Manipulation Language)
43. ~~Phải dùng~~ câu lệnh commit hoặc rollback tường minh ở cuối transaction
44. Phát biểu nào dưới đây đúng về Explicit transaction
45. Bắt đầu bằng Begin transaction
46. Dùng lệnh commit transaction để kết thúc transaction thành công
47. Dùng Rollback transaction để hủy bỏ transaction
48. Tất cả các ý trên

Bài tập chương 4

Câu 1. Trả lời các câu hỏi sau:

1. Transaction là gì? Nó khác gì khi so với một chương trình thông thường (chẳng hạn một chương trình viết bằng ngôn ngữ C)

* So với một chương trình thông thường (ví dụ, một chương trình viết bằng ngôn ngữ C), transaction khác biệt ở việc đảm bảo tính nhất quán và an toàn cho dữ liệu trong một môi trường đa người dùng và đa nhiệm như hệ thống quản lý cơ sở dữ liệu. Trong một chương trình thông thường, bạn cần tự mình quản lý việc đọc và ghi dữ liệu từ và đến cơ sở dữ liệu, trong khi một hệ thống quản lý cơ sở dữ liệu tự động quản lý các transaction để đảm bảo tính nhất quán của dữ liệu.Định nghĩa các thuật ngữ sau: atomicity, consistency, isolation, durability, schedule, blide write, dirty read, unrepeatable read, serializable schedule, recoverable schedule, avoids-cascading-aborts schedule.

1. Mô tả Strict 2PL.

* Giao thức khóa được sử dụng rộng rãi nhất, được gọi là Khóa hai pha nghiêm ngặt, hoặc 2PL nghiêm ngặt, có hai quy tắc
  + Nếu một giao dịch T muốn đọc (tương ứng, sửa đổi) một đối tượng, trước tiên nó yêu cầu một khóa được chia sẻ (tương ứng, độc quyền) trên đối tượng.
  + Tất cả các khóa được giữ bởi một giao dịch được giải phóng khi giao dịch hoàn tất.
  + Nếu một Xact giữ một khóa X trên một đối tượng, không có Xact nào khác có thể có được khóa (S hoặc X) trên đối tượng đó.

Câu 2. Xét các hành động được thực hiện bởi transaction T1 trên hai đối tượng CSDL như sau: R(X), W(X), R(Y), W(Y).

1. Hãy cho một ví dụ về transaction T2 sao cho nếu thực hiện đồng thời hai transaction mà không có cơ chế kiểm soát đồng thời thì có thể ngăn cản việc thực hiện T1.

|  |  |
| --- | --- |
| T1 | T2 |
| R(X)  W(X)  R(Y)  W(Y) | R(X)  W(X) |

* T1 thực hiện R(X) và nhận được giá trị 100. Sau đó, T2 bắt đầu thực hiện. T2 thực hiện R(X) và nhận được giá trị 100. Sau đó, T2 cập nhật giá trị của X thành 200. Khi T1 thực hiện W(X), nó sẽ ghi giá trị 100 vào X. Tuy nhiên, giá trị của X hiện đã là 200, vì vậy T1 sẽ không thể ghi giá trị 100 của mình. Điều này sẽ khiến T1 thất bại.

1. Giải thích Strict 2PL sẽ thực hiện việc ngăn cản sự ảnh hưởng giữa hai transaction.

* T2 sẽ không thực hiện được lệnh R(X), W(X) khi mà T1 đang nắm giữ khoá Strict 2PL để thực thi R(X), W(X) thì từ đó việc ảnh hưởng giữa 2 transaction bị ngăn cản

1. Strict 2PL được sử dụng trong nhiều hệ CSDL. Hãy nêu hai lý do tại sao?

* Khoá Strict 2PL đảm bảo tính chắc chắn bằng cách đảm bảo rằng mọi transaction sẽ tuân thủ quy tắc "strictness", tức là không cho phép đọc hoặc ghi vào dữ liệu đã được đánh dấu bởi các khoá đã được giữ.
* Khoá Strict 2PL được triển khai một cách đơn giản, giúp tối giản hóa quản lý các khoá và kiểm soát việc truy cập dữ liệu. Điều này làm cho việc triển khai và duy trì hệ thống dễ dàng hơn.

Câu 3. Xét một CSDL có hai đối tượng X và Y. Giả sử có hai transaction T1 và T2. Transaction T1 thực hiện : R(X), R(Y) và W(X). Transaction T2 thực hiện : R(X), R(Y), W(X), W(Y).

1. Hãy cho một lịch biểu với các hành động của T1 và T2 trên đối tượng X và Y mà nó gây ra xung đột ghi- đọc (write-read conflict).

|  |  |
| --- | --- |
| T1 | T2 |
| R(X) R(Y)  W(X) | R(X) R(Y)  W(X)  W(Y) |

1. Hãy cho một lịch biểu với các hành động của T1 và T2 trên đối tượng X và Y mà nó gây ra xung đột đọc-ghi (read- write conflict).

|  |  |
| --- | --- |
| T1 | T2 |
| R(X) R(Y)  W(X) | R(X)  R(Y)  W(X)  W(Y) |

1. Hãy cho một lịch biểu với các hành động của T1 và T2 trên đối tượng X và Y mà nó gây ra xung đột đọc-ghi (write- write conflict).

|  |  |
| --- | --- |
| T1 | T2 |
| R(X) R(Y)  W(X) | R(X)  R(Y)  W(X)  W(Y) |

1. Hãy lý giải Strict 2 PL sẽ không cho phép lịch biểu nào thực thi.

Write – Write: nếu T1 hoặc T2 thực hiện Write thì hệ thống sẽ cập một Xlock cho T1 hoặc T2 đó. Và trước khi commit thì sẽ không cấp xlock cho tuyến trình khác nên sẽ không thể Write nó. Nên 2PL sẽ không cho lich biểu thực thi.

Write – Read: nếu T1 hoặc T2 thực hiện Write thì hệ thống sẽ cập một Xlock cho T1 hoặc T2 đó. Và trước khi commit thì sẽ không cấp slock cho tuyến trình khác. Nên 2PL sẽ không cho lich biểu thực thi

Câu 4. Xét lịch biểu S (chưa đầy đủ) sau :

T1: R(X), T1: R(Y), T1: W(X), T2: R(Y), T3: W(Y), T1: W(X), T2: R(Y)

Với mỗi yêu cầu dưới đây, hãy chỉnh sửa S để tạo một lịch biểu đầy đủ thỏa mãn các điều kiện đã cho. Nếu có một chỉnh sửa nào là không thể thực hiện hãy giải thích lý do. Nếu nó có thể hãy dùng số lượng hành động có thể nhỏ nhất (Read, Write, Commit hay Abort). Bạn có thể tùy ý thêm hành động ở bất kỳ chỗ nào trong lịch biểu S.

1. Lịch biểu cho kết quả tránh đươc việc hủy bỏ dây chuyền (cascading abort) nhưng không thể phục hồi (not recorverable)

* T1: R(X), T1: R(Y), T1: W(X), T2: R(Y), T3: W(Y), T1: W(X), T2: R(Y), T1: Comit, T2: Abort, T3: Comit

1. Lịch biểu cho kết quả có thể phục hồi

* T1: R(X), T1: R(Y), T1: W(X), T2: R(Y), T3: W(Y), T3 : Commit, T1: W(X), T2: R(Y), T2 : Commit

1. Lịch biểu cho kết quả là xung đột-khả tuần tự (conflict-serializable)

* Không thể thực hiện được lịch biểu này. Vì T2 : R(Y) xung đột với T3 : W(Y) do đó không thể đổi chỗ cho nhau.

Câu 5. Định nghĩa các thuật ngữ sau: conflict-serializable schedule, View-serializable schedule, strict schedule.

* Conflict-serializable schedule là một lịch trình khả tuần tự xung đột nếu nó là xung đột tương đương với một lịch trình nối tiếp
* View- serializable schedule nghĩa là lịch trình khả tuần tự view. Một lịch trình khả tuần tự view khi nó tương đương với một lịch trình tuần tự

1. Mô tả hai nghi thức lock sau: 2PL, conservative 2PL.

* Nghi thức 2PL cung cấp 2 loại khóa là S và X.
  + Khóa X: khi transaction T1 yêu cầu khóa X để truy cập đối tượng, các transaction T2 sẽ không thể thực hiện thao tác với đối tượng mà transaction T1 đang giữ khóa X, thường sẽ thực hiện thao tác write
  + Khóa S: khi transaction T1 yêu cầu khóa S để truy cập đối tượng, các transaction T2 sẽ có thể thực hiện thao tác với đối tượng mà transaction T1 đang giữ khóa S, thường là thao tác read. Đây gọi là khóa dùng chung
* Transaction phải yêu cầu khóa tất cả các mục dữ liệu cần thiết trước khi transaction bắt đầu thực hiện.

1. Tại sao Lock và Unlock phải là các thao tác atomic.

* Lock và Unlock phải là các thao tác atomic vì 2 thao tác này yêu cầu 2 trạng thái là thành công hoặc thất bại. Không được phép thực hiện nửa chừng vì như vậy sẽ không thể quyết định cấp quyền hoặc không cấp quyền truy cập dữ liệu cho các transaction.

1. Vấn đề phantom (phantom problem) là gì? Có phải nó xảy ra trong CSDL mà tập các đối tượng CSDL là cố định và chỉ có giá trị của đối tượng có thể được thay đổi.

* Phantom problem là các vấn đề có thể xảy ra trong lúc thực hiện các lệnh transaction và xảy ra khi các transaction tiến hành thay đổi các giá trị của đối tượng( viết) nhưng không đồng nhất khiến kết quả tổng không đúng

1. Trình bày điểm khác biệt giữa các thời biểu (timestamps) được gán cho các transaction được khởi động lại khi thời biểu được dùng để ngăn cản deadLock so với thời biểu được dùng để kiểm soát đồng thời.

* Điểm khác biệt chính giữa các thời biểu (timestamps) được gán cho các transaction được khởi động lại khi thời biểu được dùng để ngăn cản deadlock so với thời biểu được dùng để kiểm soát đồng thời là ở cách gán thời biểu.
  + Khi sử dụng thời biểu để ngăn cản deadlock, các timestamps được gán cho các transaction theo thứ tự thời gian thực tế. Transaction nào được tạo trước sẽ có timestamps nhỏ hơn transaction nào được tạo sau. Điều này giúp đảm bảo rằng các transaction có timestamps nhỏ hơn sẽ luôn được thực hiện trước các transaction có timestamps lớn hơn.
  + Khi sử dụng thời biểu để kiểm soát đồng thời, các timestamps được gán cho các transaction một cách ngẫu nhiên. Điều này giúp đảm bảo rằng các transaction sẽ được thực hiện theo một thứ tự ngẫu nhiên, tránh được tình trạng một số transaction chiếm dụng tài nguyên quá lâu.

Câu 6. Xác định các lớp lịch biểu dưới đây thuộc các lớp lịch biểu nào trong các lớp lịch biểu: serializable, conflict-serializable, view-serializable, recoverable, avoids-cascading-aborts, strict.

Nếu bạn không thể xác định một lịch biểu nào đó thuộc lớp nào dựa trên danh sách các hành động, hãy giải thích lý do.

Các hành động được liệt kê theo thứ tự chúng được lập lịch. Nếu một lịch không có hành động commit hay abort thì lịch đó không đầy đủ. Giả sử hành động abort/commit phải đứng sau các hành động được liệt kê.

Ảnh có chứa văn bản, ảnh chụp màn hình, Phông chữ, số

Mô tả được tạo tự động

1. view-serializable

2. conflict-serializable

3. conflict-serializable

4. conflict-serializable

5. serializable, conflict-serializable, and view-serializable; recoverable and avoid cascading

6. aborts

7. Recoverable

8. Không xác định vì nó có tất cả các tính chất trên

9. avoids-cascading-aborts

10. avoids-cascading-aborts

11. không xác định vì nó có tất cả các tính chất trên

12. strict

Câu 7. Xét các chuỗi hành động được liệt kê theo thứ tự được đệ trình tới DBMS sau:

Ảnh có chứa văn bản, Phông chữ, màu trắng, ảnh chụp màn hình

Mô tả được tạo tự động

Với mỗi chuỗi và với mỗi cơ chế kiểm soát đồng thời (Wait-die policy, deadLock detection, Conservative and strict 2PL, Optimistic concurrency control), hãy mô tả cơ chế kiểm soát đồng thời xử lý chuỗi hành động như thế nào.

1. Với S1

T1 yêu cầu khóa S(X) và nhận được để đọc dữ liệu từ X

T2 yêu cầu khóa X(X), X(Y) và nhận được để ghi dữ liệu X, Y vào hệ thống

T3 yêu cầu khóa X(Y)

T2 trả khóa X(Y) để T3 ghi dữ liệu

T3 nhận được khóa X(Y)

T1 yêu cầu khóa X(Y)

T3 trả khóa X(Y) để T1 ghi dữ liệu

T1 nhận được khóa X(Y)

Khi T1 commit, T1 sẽ trả S(X) và X(Y)

T2 commit, T3 commit

2. Với S2

T1 yêu cầu khóa S(X) và nhận được để đọc dữ liệu từ X

T2 yêu cầu khóa X(X), X(Y) và nhận được để ghi dữ liệu X, Y vào hệ thống

T3 yêu cầu khóa X(Y)

T2 trả khóa X(Y) để T3 ghi dữ liệu

T3 nhận được khóa X(Y)

T1 yêu cầu khóa X(Y)

T3 trả khóa X(Y) để T1 ghi dữ liệu

T1 nhận được khóa X(Y)

Khi T1 commit, T1 sẽ trả S(X) và X(Y)

T2 commit, T3 commit