CHƯƠNG 4

1. Transaction là gì?

Trả lời: Một Giao dịch (Transaction) là một thực thi của một chương trình người dùng, và được DBMS xem như một chuỗi các thao tác. Các thao tác này có thể được thực hiện bằng một giao dịch nào đó bao gồm: các phép đọc và viết lên các đối tượng cơ sở dữ liệu, trong khi đó các thao tác trong một chương trình thông thường có thể bao gồm dữ liệu đầu vào của người dùng, truy cập các thiết bị mạng, xây dựng giao diện người dùng, …

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

Trả lời: Transaction khác mới một chương trình được thực thi bên ngoài DBMS.

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. Nêu 4 tính chất quan trọng của transaction? Cho ví dụ

Trả lời:

* **Atomic (Nguyên tử):** Tất cả các thao tác trong 1 Transaction được thực hiện thành công hoặc thất bại hoàn toàn. Người dùng không phải lo lắng về ảnh hưởng của các Transaction chưa được thành công (Giả sử có sự cố xảy ra trong quá trình Transaction thực hiện).

***Ví dụ:*** Khi chuyển tiền từ A -> B. Nếu giao dịch không thành công trong một hoặc nhiều bước trong Transaction, hệ thống phải đảm bảo rằng các bản cập nhật của nó không được phản ánh trong cơ sở dữ liệu, nếu không sẽ dẫn đến sự không nhất quán.

* **Consistent (Nhất quán)**: Mỗi giao dịch được thực thi không tranh chấp với các giao dịch khác, phải đảm bảo tính chất nhất quán của cơ sở dữ liệu. DBMS thừa nhận rằng tính nhất quán được đảm bảo trên mỗi giao dịch. Việc đảm bảo tính chất này của giao dịch là trách nhiệm của người dùng.

***Ví dụ:*** Khi chuyền tiền từ A -> B. Thì tổng tiền trong 2 tài khoản A và B sau khi chuyển và trước khi chuyển phải bằng nhau.

* **Isolated (Cô lập):** Một Transaction khi được người dùng thực hiện thì nó thực hiện như thể là chỉ có một Transaction được thực thi trong 1 thời điểm. Nhưng thực tế thì trong 1 thời điểm có thể có nhiều Transaction cùng được thực hiện.

***Ví dụ:*** Khi chuyển tiền từ A -> B. Nếu giữa các bước 3 và 6, một giao dịch khác được phép truy cập vào CSDL được cập nhật một phần, nó sẽ thấy một CSDL không nhất quán (tổng A + B sẽ nhỏ hơn mức cần thiết). Có thể được đảm bảo một cách nhỏ nhặt bằng cách chạy các giao dịch nối tiếp nhau, cái này đến cái kia. Tuy nhiên, thực hiện đồng thời nhiều giao dịch có những lợi ích đáng kể, như chúng ta sẽ thấy.

* **Durable (Bền vững):** Khi DBMS thông báo cho người dùng biết rằng giao dịch đã thành công hoàn toàn, những ảnh hưởng của nó nên được duy trì ngay cả khi hệ thống gặp sự cố trước khi tất cả những thay đổi này kịp lưu lại trên đĩa.

***Ví dụ:*** Khi chuyển tiền từ A -> B. Sau khi người dùng được thông báo rằng giao dịch đã hoàn tất (tức là quá trình chuyển tiền đã diễn ra), các cập nhật cho cơ sở dữ liệu của giao dịch phải vẫn tồn tại bất chấp lỗi.

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

Trả lời: Lịch biểu tuần tự (*Serial schedule*): là lịch biểu chỉ cho phép thực hiện xong Transaction này, rồi mới tiếp tục thực hiện Transaction khác.

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. Lịch biểu khả tuần tự?

Trả lời: Lịch biểu khả tuần tự (*Serializable schedul*): là 1 lịch biểu tương đương với 1 lịch biểu tuần tự nào đó.

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. Định nghĩa 2 lịch biểu L1 và L2 được gọi là tương đương xung đột

Trả lời: L1 và L2 được gọi là tương đương xung đột nếu L2 là L1 sau khi thực hiện một loạt cách lệnh hoán đổi không xung đột. Hoặc ngược lại.

1. Định nghĩa lịch biểu khả tuần tự xung đột

Trả lời: Lịch biểu tuần tự xung đột (*Conflict Serializable*): là lịch biểu tương đương xung đột với một lịch biểu tuần tự.

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:
2. T1:R(X), T2:R(Y), T1: R(Y), T2:W(Y), T2:R(X), T1:W(X), T2:W(X), T1:W(Y)
3. T2:R(Y), T1:R(X), T1: R(Y), T2:W(Y), T2:R(X), T1:W(X), T1:W(Y), T2:W(X)
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?

Trả lời: 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 tuần tự
4. 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
5. T1:R(X), T1: W(X), T2:R(Y), T2:R(X), T2:W(X), T2:W(Y)
6. T2:R(Y), T1:R(X), T1: W(X), T2:R(X), T2:W(X), T2:W(Y)
7. Định nghĩa 2 Lịch biểu tương đương view

Trả lời: Hai lịch biểu được gọi là tương đương view (*View Equivalent*) nếu đạt ba điều kiện sau:

1. Với mỗi đối tượng Q, nếu Tracsaction Ti đọc giá trị ban đầu của Q trong lịch biểu S thì Transantion Ti trong lịch biểu S’ cũng phải đọc giá trị ban đầu của Q.
2. Với mỗi đối tượng Q, nếu Transaction Ti thực hiện đọc Q trong lịch biểu S, mà giá trị Q này trước đó được ghi bởi Transaction Tj (nếu có) thì Transaction Ti trong lịch biểu S’ cũng phải đọc giá trị Q, mà giá trị Q trước đó cũng được ghi bởi Transaction Tj của S’.
3. Với mỗi đối tượng Q, nếu Transaction thực hiện việc ghi giá trị Q cuối cùng (nếu có) trong lịch biểu S, thì Transaction đó cũng phải thực hiện việc ghi giá trị Q cuối cùng trong lịch biểu S’.
4. Định nghĩa lịch biểu khả tuần tự view

Trả lời: Lịch biểu khả tuần tự view (*View Serializable*): nếu lịch biểu này tương đương view với lịch biểu tuần tự.

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

Trả lời:

* *Blind Write (Ghi mù):* Là việc một Transaction thực hiện ghi lên một đối tượng khi chưa đọc đối tượng đó.
* *Dirty Read (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

Trả lời: Lịch biểu được gọi là khả phục hồi (Recoverable Schedule): nếu 1 Transaction Tj đọc 1 đối tượng dữ liệu mà trước đó đối tượng này được ghi bởi Transaction Ti thì thao tác Commit của Ti phải xuất hiện trước thao tác Commit của Tj.

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. \* Lịch biểu nào dưới đây dẫn đến việc phục hồi lan truyền
11. T1:R(X), T1:W(X), T2:R(X), T2:W(X), T1: commit, T2: commit
12. T1:R(X), T1:W(X), T2:R(Y), T2:W(Y), T1: commit, T2: commit
13. T1:R(X), T1:W(X), T2:R(X), T2:W(X), T2: commit, T1: commit
14. T1:R(X), T2:R(X), T1: W(X), T2:W(X), T1: commit, T2: commit
15. Một lịch biểu là khả phục hồi không lan truyền nếu
16. 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
17. 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
18. Lịch biểu nào dưới đây dẫn đến việc phục hồi không lan truyền
19. T1:R(X), T1:W(X), T2:R(X), T2:W(X), T1: commit, T2: commit
20. T1:R(X), T1:W(X), T1: commit, T2:R(X), T2:W(X), T2: commit
21. T1:R(X), T1:W(X), T2:R(X), T1: commit, T2:W(X), T2: commit
22. Phát biểu nào dưới đây đúng với nghi thức khóa 2 giai đoạn (2 phase locking)
23. Dùng để quan lý sự truy cập đồng thời
24. 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,
25. Chỉ khi nhận được lock transaction mới đọc (hoặc chỉnh sửa đối tượng)
26. Phát biểu nào dưới đây đúng
27. 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
28. 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
29. Phát biểu nào dưới đây đúng
30. Trong SQL Server có 3 loại transaction: tự động commit, tường minh và không tường minh
31. Trong SQL Server có 2 loại transaction: Tường minh và không tường minh
32. Trong SQL Server chỉ có 1 loại transaction duy nhất
33. Phát biểu nào dưới đây đúng về Autocommit transaction
34. Là transaction mặc định
35. Mỗi câu lệnh T-SQL được xem là 1 transaction. Chúng được commit hay rollback tùy vào kết quả của chúng.
36. Phát biểu nào dưới đây đúng về Implicit transaction
37. Được kích hoạt bằng câu lệnh SET IMPLICIT\_TRANSACTIONS ON
38. 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)
39. Phải dùng câu lệnh commit hoặc rollback tường minh ở cuối transaction
40. Phát biểu nào dưới đây đúng về Explicit transaction
41. Bắt đầu bằng Begin transaction
42. Dùng lệnh commit transaction để kết thúc transaction thành công
43. Dùng Rollback transaction để hủy bỏ transaction