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)

Là những thao tác của chương trình người dùng trên một DB, nó là một chương trình nằm trong cơ sở dữ liệu.

1. 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.

* Atomic: một transaction hoặc là thực hiện hoàn toàn hoặc không thực hiện
* Consistent: DB có thể có các ràng buộc, đảm bảo cho các ràng buộc này thuộc về trách nhiệm của người dùng
* Isolated: thể hiện với người dùng như thể chỉ có một tiến trình đang chạy
* Durable: khi một transaction được thực hiện thành công, thì kết quả của transaction đó được lưu trong hệ thống bất kể hệ thống có gặp trục trặc gì
* Schedule: Là một lịch trình thực thi các câu lệnh trong các transaction khác nhau theo thời gian
* Blind write: là hiện tượng một lịch biểu khả tuần tự view chứ không khả tuần tự xung đột
* Dirty read: là một trong các trường hợp xảy ra khi sử dụng các thao tác đọc (Read) và ghi (Write) dữ liệu. Trong trường hợp đọc bẩn, một giao tác nào đó đọc dữ liệu nhưng chưa kết thúc, trong khi giao tác khác đã thay đổi dữ liệu đó và chưa được xác nhận. Khi đó, giao tác đọc sẽ nhận được kết quả không chính xác hoặc "bẩn" (dirty).
* Serializable schedule: là lịch biểu tương đương với một lịch biểu tuần tự
* Recoverable shedule: lịch biểu trong đó nếu một giao tác cần phải bị khôi phục trở lại thành trạng thái trước khi được commit, nó sẽ không phá vỡ tính nhất quán của cơ sở dữ liệu.

1. Mô tả Strict 2PL.

* Một transaction muốn thao tác trên một đối tượng, nó sẽ yêu cầu được chia sẻ một lock trên đối tượng đó
* Tất cả các lock được lưu bởi một transaction sẽ được giải phóng khi transaction đó kết thúc

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.

T2 R(X), R(Y), W(X), R(Y)

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

Đó là khi một transaction đang giữ lock của một đối tượng, không có transaction nào khác có thể truy cập vào đối tượng đó.

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

* Nhằm kiểm soát các transaction khi có sự truy cập đồng thời
* Đảm bảo tính nhất quán, nguyên tử, bền vững.

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)  W(X)  R(Y) | R(X)  R(Y)  W(Y)  W(X) |

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) | W(X)  R(Y)  R(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).
2. Hãy lý giải Strict 2 PL sẽ không cho phép lịch biểu nào 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)
2. Lịch biểu cho kết quả có thể phục hồi
3. Lịch biểu cho kết quả là xung đột-khả tuần tự (conflict-serializable)

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

1. Mô tả hai nghi thức lock sau: 2PL, conservative 2PL.
2. Tại sao Lock và Unlock phải là các thao tác atomic.
3. 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.
4. 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.

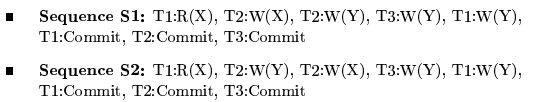
Câu 6. Xác định các lớp lịch biểu dưới đây thuộc các lớp lịch biểu nào trong các lớp lịch biểu: serializable, conflict-serializable, view-serializable, recoverable, avoids-cascading-aborts, strict.

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.



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



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.