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)

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, vv.

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: Toàn bộ giao dịch hoặc không có giao dịch nào được thực hiện. Người dùng không phải lo lắng về ảnh hưởng của các giao dịch không hoàn tất.

Nhất quán (consistency): 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.

Isolated: Nó xuất hiện với người dùng như thể chỉ có một quá trình thực thi tại một thời điểm.

Durable: một giao dịch 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.

Lịch trình (schedule): là một chuỗi các giao dịch (có thể xếp chồng).

Viết mù (blind write ): là việc một giao dịch nào đó viết lên một đối tượng mà thậm chí không đọc đối tượng này.

Đọc không thể lặp lại (unrepeatable read ): xảy ra khi một giao dịch nào đó không thể đọc cùng một giá trị đối tượng nhiều hơn một lần, thậm chí giao dịch này không được phép thay đổi giá trị. Giả sử giao dịch T2 thay đổi giá trị của đối tượng A – đối tượng đang được đọc bằng một giao dịch T1 trong khi T1 vẫn đang trong quá trình xử lý. Nếu T1 cố gắng đọc giá trị A một lần nữa, nó sẽ có một kết quả khác, mặc dù nó không thay đổi A.

Lịch trình tuần tự (serializable schedule ): trên một tập S của các giao dịch là một lịch trình mà đều giống với lịch trình tuần tự hoàn toàn trên tập các giao dịch thành công trong S.



Lịch trình tránh hủy bỏ chồng (avoids-cascading-aborts schedule ): là một trong số các giao dịch chỉ đọc những thay đổi của các giao dịch đã thành công. Một lịch trình như vậy không chỉ có khả năng phục hồi, việc hủy bỏ một giao dịch có thể được hoàn thành mà không hủy bỏ chồng các giao dịch khác.

1. Mô tả Strict 2PL.

* Một giao dịch yêu cầu một khóa chia sẻ/ độc quyền trên một đối tượng trước khi nó (đọc/ sửa) đối tượng đó.
* Tất cả các khóa mà giao dịch nắm bắt được giải phóng khi giao dịch đó thành công.

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

1. Giải thích Strict 2PL sẽ thực hiện việc ngăn cản sự ảnh hưởng giữa hai transaction.
2. Strict 2PL được sử dụng trong nhiều hệ CSDL. Hãy nêu hai lý do tại sao?

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).
2. 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).
3. 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).
4. 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.

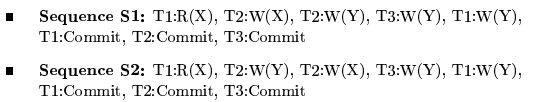
Câu 6. Xác định các lớp lịch biểu dưới đây thuộc các lớp lịch biểu nào trong các lớp lịch biểu: serializable, conflict-serializable, view-serializable, recoverable, avoids-cascading-aborts, strict.

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.