

● DeadLock이 무엇인지 설명해라

▶ 데이터베이스 deadlock(교착 상태)이란, 여러 transaction들이 각각 자신의 데이터에 대하여 lock을 획득한 상태에서 상대방 데이터에 대하여 접근하고자 대기를 할 때, 교차 대기를 하게 되면서 서로 영원히 기다리는 상태를 말합니다. 두 transaction이 각각 lock을 설정하고, unlock을 하지 않은 상태에서 서로의 lock이 걸린 데이터에 접근하려고 할 때, 서로 대기를 계속하여 영원히 처리되지 않는 상황이 발생합니다. deadlock을 해결하는 방법으로는 각 transaction이 실행되기 전에 사용될 모든 데이터를 미리 locking을 해주는 예방기법이 있고, 자원 할당시 timestamp를 사용하여 deadlock이 발생하지 않도록 회피하는 기법이 있고, 또한 deadlock이 발생하면 이를 감지하고 회복시키는 탐지/회복기법이 있습니다.

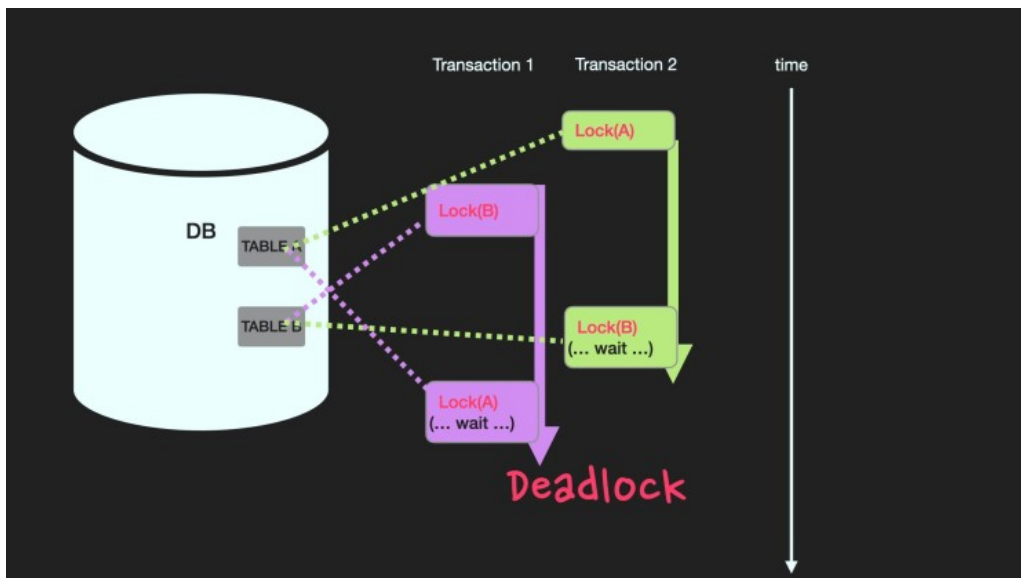
▶ 예방기법: 각 transaction이 실행되기 전에 필요한 데이터를 모두 Locking 해주는 것입니다. 하지만 locking해야 하는 데이터가 많다면 사실상 모든 데이터를 전부 locking한 것과 동일하여 transaction의 병행성을 보장하지 못할 수 있습니다. (Transactions들이 동시에 진행되지 못할 수도 있다)

▶ 회피기법: 자원을 할당할 때 timestamp를 사용하여 deadlock가 일어나지 않도록 회피하는 방법입니다. deadlock을 예방하기 위해 트랜잭션이 시작된 시간을 timestamp로 삼고, 이를 기준으로 크게 두 방식으로 처리합니다.

1) Wait-die: T1, T2 두 개의 transaction이 있다고 해볼게요. T2가 선점하고 있는 data에 T1이 접근을 요청합니다. T1의 timestamp가 T2의 timestamp보다 더 older일 경우에는 wait 할 수 있습니다. 그렇지 않은 경우에 T1은 그냥 die 합니다 (roll back).

2. Wound-wait: T1, T2 두 개의 transaction이 있다고 해볼게요. T2가 선점하고 있는 data에 T1이 접근을 요청합니다. T1의 timestamp가 T2의 timestamp보다 더 younger일 경우에는 wait 할 수 있습니다. 그렇지 않은 경우에 T2는 그냥 die 합니다(roll back).

▶ 탐지/회복 기법: Transaction이 실행되기 전에는 아무런 검사를 하지 않고, deadlock이 발생하면 이를 감지하고 회복시키는 방법입니다.



Resources: inflearn 개발남 노씨