2. 장애와 회복

회복 정의

회복은 데이터베이스 운영 도중에 예상하지 못한 **장애(failure)**가 발생하였을 때,
데이터베이스를 장애 발생 이전의 상태로 **복원**시키는 것이다.

• 데이터베이스 관리 시스템에서 회복을 담당하는 것이 회복관리자(recovery manager)이다.

// 장애(failure) 종류

시스템 장애 **(system failure)** **전역적 고장**	• **하드웨어 오동작** • **하드웨어 오동작**으로 주기억장치에 저장된 데이터가 손실된 경우 • 교착상태가 발생하여 트랜잭션이 더 이상 작업을 수행할 수 없는 경우 • 중앙처리장치, 주기억장치, 전원공급장치 등이 고장나는 경우
트랜잭션 장애 **(transaction failure)** **지역적 고장**	• **트랜잭션 실행 오류** • **트랜잭션**이 정상적으로 **실행**을 계속할 수 없는 경우 • 잘못된 매개변수 • 불량한 입력데이터 • 데이터를 찾을 수 없는 경우 • 오버플로(overflow) 발생 • 0으로 나누는 연산(논리적 오류) • 가용 자원(resource)의 초과 요청
미디어 장애 **(media failure)**	• **하드디스크 고장**으로 저장된 데이터베이스가 손상된 경우 • 디스크 헤드 파손 • 디스크 입출력 오류 • 디스크 블록의 데이터 손실

// 회복을 위해 사용되는 것

• 백업(backup)
• 로그(log)
• 검사점(checkpoint)
• 그림자 페이징(shadow paging) 등

// 회복 기법

- 즉시갱신 / 지연갱신 회복 : **로그(log)** 이용
- 검사점(checkpoint) 회복 : 로그와 검사점 이용
- ARIES 회복 : 로그와 검사점 이용, **undo** 중에 로깅을 함(3단계 회복)
- 그림자 페이징(shadow paging) 회복 : **로그(log)**와 검사점을 이용하지 않는 회복 기법
- 미디어(디스크) 회복 : 데이터베이스 전체 내용을 안정 저장장치에 주기적으로 **덤프(dump)**

// redo와 undo

redo/undo는 데이터베이스에 장애가 발생했을 때, **회복**을 위해 취할 수 있는 조치이다.

redo **재수행**	• **전진 회복 :** 로그레코드를 **순차적**으로 실행 • redo는 로그 내용을 이용하여 데이터베이스를 **갱신**하는 것 • 트랜잭션이 commit된 상태에서 장애가 발생했을 때, redo 수행 • redo 수행 이유: **output()** 연산이 확실하게 수행되었는지 모르므로 • **완료된** 트랜잭션이 변경한 데이터베이스 페이지는 **유지(durability)**되어야 한다. • 완료된 트랜잭션의 수정을 데이터베이스에 반영하는 복구 작업을 redo라 한다.
undo **취소**	• **후진 회복 :** 로그레코드를 **역순**으로 실행(대부분 앱에서 사용하는 Ctrl+Z과 같음) • **미완료된** 트랜잭션이 변경한 데이터베이스 페이지는 **원상 복구**되어야 한다. • 이러한 복구를 **undo**라 한다. • 시스템 장애가 발생하면 **미완료된** 트랜잭션이 변경한 내용은 취소(undo)한다.

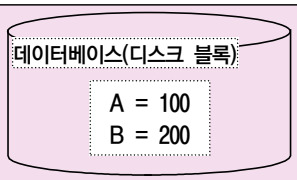
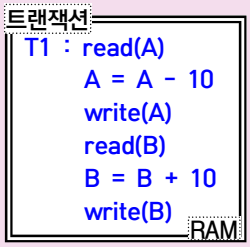
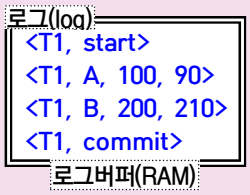
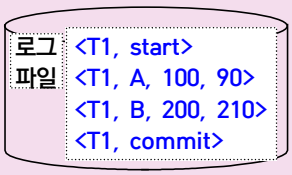
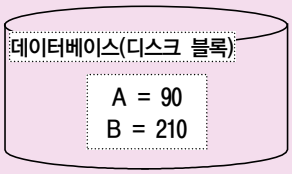
- 회복 작업이 완료될 때까지 시스템은 새로운 트랜잭션을 받아들일 수 없다.

// 로그(log)

- 로그는 데이터베이스 **변경에 대한 기록**이다.
- 로그는 **트랜잭션** 수행 결과 **생성**되는 **로그레코드** 집합이다.
- 로그레코드는 데이터베이스의 모든 갱신 작업을 기록한 것이고 **연속적**이다.
- 로그는 덧붙이는(append) 방식으로 기록되며, 각 로그레코드는 고유의 식별자를 가진다.
- 로그는 뒤에 덧붙이는 방식으로 기록되므로, 로그식별자는 **단조증가**하는 성질을 가진다.
- 로그레코드 식별자를 LSN(log sequence number) 또는 LSA(log sequence address)라 한다.
- 시스템 장애가 발생하면, 로그를 참고하여 트랜잭션의 **redo** 또는 **undo**를 결정한다.
- 이론적으로 로그는 **안정 저장장치**(stable storage)에 기록되어야 하는 것으로 말한다.
- 안정 저장장치는 어떤 경우에도 정보 손실이 발생하지 않는 이상적인 매체이다.
- 이러한 이상적인 장치는 현실에 존재하지 않지만 **RAID** 시스템은 안정 저장장치로 분류한다.
- DBMS 스스로 여러 개의 로그를 유지하는 방법으로 안정 저장장치처럼 동작하기도 한다.
- 하지만, 대부분 DBMS는 성능 상의 이유로 하나의 로그를 유지한다.

// 회복 기법을 이해하기 위한 몇 가지 개념

다음은 트랜잭션 수행 과정을 개략적으로 보여준다.

<table>
<tr><td> 
</td><td>// 데이터베이스
· 데이터베이스는 안전 저장장치에 저장해야 한다.(이론적)
· 안전한 저장은 정보 손실이 발생하지 않는 비소멸성 저장이다.
· 안전 저장장치 : RAID system이나 Mirrored disk
· 데이터베이스는 실제로 디스크에 상주하고 있다.</td></tr>
<tr><td>↓</td><td></td></tr>
<tr><td> 
</td><td>// 트랜잭션 수행
· input : 트랜잭션 수행에 필요한 데이터, 디스크에서 주기억장치로
· 트랜잭션은 입력된 데이터를 처리하고(연산)
· output : 트랜잭션 처리 결과를 주기억장치에서 디스크로 출력
· 디스크 출력에 해당하는 코드가 write(A), write(B)이다.
· 여기서, 주의할 것은 write()의 출력은 보통 즉시 수행되지 않는다.
· 로그를 생성하고, 로그를 디스크에 기록하는 작업을 먼저 수행한다.</td></tr>
<tr><td>↓</td><td></td></tr>
<tr><td> 
</td><td>// 로그 생성
· 트랜잭션 수행 결과인 로그는 로그버퍼에 임시 기록된다. - RAM
· 로그는 데이터베이스의 모든 갱신 작업에 대한 기록이다.
· 로그는 트랜잭션의 갱신 작업에 대한 연속적인 기록이다.
· 트랜잭션의 갱신 작업에 해당하는 코드가 write(A), write(B)이다.</td></tr>
<tr><td>↓</td><td></td></tr>
<tr><td> 
</td><td>// 로그우선기록(WAL)
· DBMS는 로그레코드를 디스크의 로그파일에 먼저 기록한다.
· 로그파일은 시스템 장애가 발생했을 때 회복하기 위한 것이다.
· 로그파일은 트랜잭션의 원자성을 보장한다.</td></tr>
<tr><td>↓</td><td></td></tr>
<tr><td> 
</td><td>// 데이터베이스 갱신
· 데이터베이스 갱신은 로그우선기록 후에 진행된다.
· 데이터베이스 갱신은 트랜잭션의 완료(commit) 연산으로 진행된다.
· 트랜잭션 실행이 성공적으로 완료된 것이다.(영속성 보장)</td></tr>
</table>

· 트랜잭션 수행은 디스크에서 주기억장치로 데이터를 입력시켜 처리하고 출력한다.
· 입출력은 블록 단위로 수행된다.
· 디스크에 있는 블록을 디스크 블록, 주기억장치에 있는 블록을 버퍼 블록이라 한다.

기출문제 분석

1. 데이터베이스 시스템의 회복(recovery)에 대한 설명으로 가장 옳지 않은 것은? [2021년 서울 7급]

① 장애로 인해 손상된 데이터베이스를 손상되기 이전의 정상적 상태로 복구하는 것이다.
② SW 오류는 물론 HW 오류까지 대비한다.
③ 검사점(checkpoint) 기법의 목적은 복구 작업에 소요되는 시간을 줄이기 위한 것이다.
④ WAL(Write-Ahead Logging) 기법은 데이터를 갱신한 후에, 로그에 기록을 남기는 방법이다.

♤ **데이터베이스 시스템의 회복 – 로그우선기록(WAL, write ahead logging)**

- DBMS는 데이터베이스 **갱신 전**에 로그(log)를 **디스크**의 **로그파일**에 **먼저 기록**한다.
- 시스템 장애가 발생했을 때 회복하기 위한 것이다.(트랜잭션 원자성 보장)
- 트랜잭션은 **〈Ti, commit〉** 로그레코드를 안정 저장장치에 출력시켜야만 완료 상태로 들어갈 수 있다.
- **〈Ti, commit〉**를 출력하려며, 먼저 **Ti**와 관련된 모든 로그레코드부터 안정 저장장치에 출력해야 한다.
- 로그가 임시 저장된 주기억장치 블록을 로그버퍼라 한다.
- 로그버퍼와 로그파일에 기록된 로그레코드들의 순서는 같아야 한다.
- 로그버퍼는 주기억장치에서 운영되므로 시스템이 붕괴되면 버퍼 내용을 잃을 수도 있다.

정답 : ④

2. 로그버퍼에 대한 설명으로 옳지 않은 것은? [2015년 국가 7급]

① 로그파일은 안정 저장장치(stable storage)에서 운영되며 로그버퍼는 주기억장치에서 운영된다. 따라서 시스템 고장 발생 시 로그버퍼의 내용을 잃을 수 있다.
② 로그레코드 〈Ti, commit〉가 로그파일에 기록되기 전에 로그버퍼내의 Ti와 관련된 모든 로그레코드들은 로그파일에 기록되어야 한다.
③ 데이터베이스 버퍼에 있는 블록을 데이터베이스 파일에 기록하는 것과 로그버퍼에 있는 블록을 로그파일에 기록하는 것은 순서적으로 독립적이다.
④ 로그버퍼에 기록된 로그레코드들의 순서와 로그파일의 이들의 순서는 동일하여야 한다.

♤ **로그우선기록 (WAL)**

- DBMS는 로그를 디스크상의 로그파일에 먼저 기록한다.

정답 : ③

3. 데이터베이스 재해복구(disaster recovery) 기술에 대한 설명으로 옳은 것은? [2021년 국가 7급]

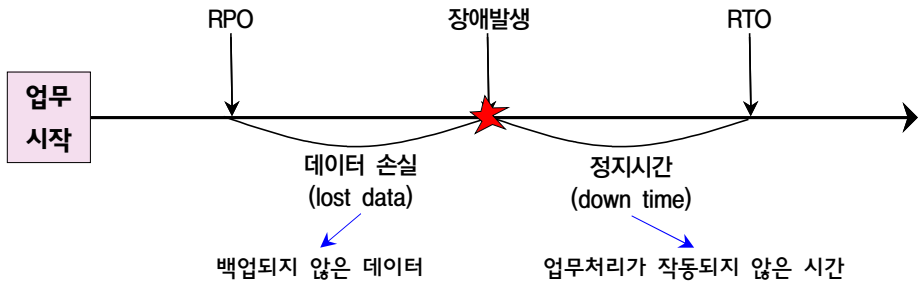
① 재해적 실패가 발생하면 가장 최근의 백업 사본이 디스크에서 백업장치로 적재된다.
② 은행, 보험, 주식 등과 같은 중요한 응용에서는 데이터 전체를 주기적으로 백업하여 안전한 장소에 보관한다.
③ 시스템 로그를 백업하면 사용자는 마지막 데이터베이스 백업 이후에 수행한 모든 트랜잭션을 잃게 된다.
④ 백업된 시스템 로그에 기록되어 있는 모든 완료된 트랜잭션의 실행결과는 데이터베이스를 undo하는 데 사용될 수 있다.

♤ **데이터베이스 재해복구(disaster recovery)**

① 재해적 실패가 발생하면 가장 최근의 백업 사본이 디스크에서 **백업장치**로 적재된다.**(×)**
→ **백업 사본**은 **디스크**에서 해당 **운영 시스템**으로 적재해야 한다.

③ 시스템 로그를 백업하면 사용자는 마지막 데이터베이스 백업 이후에 수행한 모든 트랜잭션을 **잃게 된다.(×)**
→ 백업 이후에 수행한 모든 트랜잭션을 그냥 잃게 되는 것은 아니다.
→ 백업은 지속적으로 추가로 수행될 수 있다.
→ 만약, 장애가 발생되지 않으면, 손실이 나지 않는다.

④ 백업된 시스템 로그에 기록되어 있는 모든 완료된 트랜잭션의 실행결과는 데이터베이스를 **undo** 하는 데 사용될 수 있다.**(×)**
→ 모든 완료된 트랜잭션의 실행결과는 데이터베이스를 **redo**하는 데 사용될 수 있다.



// 복구목표시점(RPO, Recovery Point Objective) – 데이터 관점
• RPO는 어느 시점에 **백업**할 것인지?를 결정하는 지표이다.**(백업시점)**
• RPO는 수용 가능한 데이터 손실의 양을 결정하는데 효과적이다.**(손실되어도 무방)**
// 복구목표시간(RTO, Recovery Time Objective) – 업무 관점
• 간단히 말하면, 복구목표시간(RTO)은 **복구하는데 걸리는 시간**이다.**(업무 관점)**
• RTO는 재해 발생 이후에 원 상태로 복구하는데 소요되는 시간이다.

정답 : ②