

EXERCISES & SOLUTIONS

10장 연습문제 풀이

회복과 병행 제어

총 41문항 · 객관식 23 + 주관식 18

박상돈 조교수 · 대전대학교 컴퓨터공학과

10



객관식 문제

Multiple Choice · Q1 – Q23

데이터베이스에서 논리적인 작업의 단위로, 하나의 기능을 수행하기 위해 필요한 일련의 연산들을 의미하는 것은?

1

뷰

2

트랜잭션

3

튜플

4

로킹

정답 ②

1 뷰

3 튜플

2 트랜잭션 ✓

4 로킹

해설

트랜잭션(transaction)은 하나의 작업을 수행하는 데 필요한 데이터베이스 연산들을 모아 놓은 논리적인 작업 단위입니다.

트랜잭션에 대한 설명으로 옳지 않은 것은?

- 1 트랜잭션의 연산은 모두 정상적으로 수행되거나 하나도 수행되지 않아야 한다.
- 2 성공적으로 수행된 후에도 데이터베이스가 일관성 있는 상태를 유지해야 한다.
- 3 수행 중 생성된 중간 연산 결과를 다른 트랜잭션의 연산이 접근할 수 있다.
- 4 성공적으로 완료된 후 반영된 수행 결과는 영구적으로 유지되어야 한다.

정답 ③

1

트랜잭션의 연산은 모두 정상적으로 수행되거나 하나도 수행되지 않아야 한다.

2

성공적으로 수행된 후에도 데이터베이스가 일관성 있는 상태를 유지해야 한다.

3

수행 중 생성된 중간 연산 결과를 다른 트랜잭션의 연산이 접근할 수 있다. ✓

4

성공적으로 완료된 후 반영된 수행 결과는 영구적으로 유지되어야 한다.

해설

③은 격리성(isolation)에 위배됩니다. 수행 중인 트랜잭션의 중간 결과는 다른 트랜잭션이 접근할 수 없어야 합니다.

트랜잭션에 대한 설명으로 옳은 것은?

1 트랜잭션의 일부 연산만 데이터베이스에 반영할 수 있다.

2 트랜잭션은 병행 제어와 회복 작업의 단위다.

3 트랜잭션은 하나의 기능을 수행하기 위한 물리적 작업 단위다.

4 작업 완료를 뜻하는 rollback과 작업 취소를 뜻하는 commit 연산이 있다.

정답 ②

1 트랜잭션의 일부 연산만 데이터베이스에 반영할 수 있다.

3 트랜잭션은 하나의 기능을 수행하기 위한 물리적 작업 단위다.

2 트랜잭션은 병행 제어와 회복 작업의 단위다. ✓

4 작업 완료를 뜻하는 rollback과 작업 취소를 뜻하는 commit 연산이 있다.

해설

트랜잭션은 병행 제어와 회복의 기본 단위입니다. ①은 원자성 위배, ③은 '논리적' 작업 단위, ④는 commit↔rollback 의미가 뒤바뀌었습니다.

다음 설명이 의미하는 트랜잭션의 특성은?

여러 트랜잭션이 동시에 수행되더라도 각 트랜잭션이 독립적으로 수행될 수 있도록, 현재 수행 중인 트랜잭션이 완료될 때까지 다른 트랜잭션의 연산은 끼어들 수 없다. 즉, T1과 T2가 있다면 T1 시작 전에 T2가 끝나거나, T1이 끝난 후 T2가 시작되어야 한다.

1 원자성(atomicity)

2 일관성(consistency)

3 격리성(isolation)

4 지속성(durability)

정답 ③

1 원자성(atomicity)

2 일관성(consistency)

3 격리성(isolation) ✓

4 지속성(durability)

해설

여러 트랜잭션이 동시에 수행되어도 서로 간섭 없이 독립적으로 실행되는 특성은 격리성(isolation)입니다.

데이터 무결성을 보장하기 위한 트랜잭션의 특성으로 옳바르지 않은 것은?

1

자율성(autonomy)

2

일관성(consistency)

3

격리성(isolation)

4

지속성(durability)

정답 ①

1 자율성(autonomy) ✓

2 일관성(consistency)

3 격리성(isolation)

4 지속성(durability)

해설

트랜잭션의 특성(ACID)은 원자성·일관성·격리성·지속성입니다. '자율성'은 ACID에 포함되지 않습니다.

다음 설명이 의미하는 트랜잭션의 특성은?

트랜잭션의 연산이 데이터베이스에 모두 반영되거나, 모두 반영되지 않아야 하는 *all-or-nothing* 방식이어야 한다.

1 원자성(atomicity)

2 일관성(consistency)

3 격리성(isolation)

4 지속성(durability)

정답 ①

1 원자성(atomicity) ✓

2 일관성(consistency)

3 격리성(isolation)

4 지속성(durability)

해설

연산이 전부 반영되거나 전혀 반영되지 않아야 하는 all-or-nothing 특성은 원자성(atomicity)입니다.

트랜잭션 상태에 대한 다음 설명 중 옳은 것은?

- 1 commit 연산을 실행한 상태를 활동 상태라고 한다.
- 2 트랜잭션의 모든 연산이 처리된 상태를 부분 완료 상태라고 한다.
- 3 장애가 발생하여 수행이 중단된 상태를 철회 상태라고 한다.
- 4 수행이 실패하여 rollback 연산을 실행한 상태를 실패 상태라고 한다.

정답 ②

1 commit 연산을 실행한 상태를 활동 상태라고 한다.

3 장애가 발생하여 수행이 중단된 상태를 철회 상태라고 한다.

2 트랜잭션의 모든 연산이 처리된 상태를 부분 완료 상태라고 한다. ✓

4 수행이 실패하여 rollback 연산을 실행한 상태를 실패 상태라고 한다.

해설

마지막 연산까지 처리하고 commit 직전인 상태가 부분 완료 상태입니다. ①은 완료, ③은 실패, ④는 철회 상태에 해당합니다.

트랜잭션 수행 도중 장애가 발생했을 때 데이터베이스를 장애 발생 전의 일관된 상태로 복구시키는 것은?

1

재시작(restart)

2

철회(abort)

3

회복(recovery)

4

정지(stop)

정답 ③

1 재시작(restart)

2 철회(abort)

3 회복(recovery) ✓

4 정지(stop)

해설

장애 발생 전의 일관된 상태로 데이터베이스를 되돌리는 작업을 회복(recovery)이라고 합니다.

트랜잭션 장애가 발생하는 원인으로 거리가 먼 것은?

1

처리 대상 데이터의 부재

2

디스크 헤드의 고장

3

잘못된 데이터의 입력

4

논리적 연산 오류

정답 ②

1 처리 대상 데이터의 부재

3 잘못된 데이터의 입력

2 디스크 헤드의 고장 ✓

4 논리적 연산 오류

해설

디스크 헤드 고장은 저장 장치 손상으로 인한 미디어 장애의 원인입니다. 나머지는 트랜잭션 자체의 장애 원인입니다.

데이터베이스 시스템에 발생하는 장애의 유형으로 거리가 먼 것은?

1

트랜잭션 장애

2

시스템 장애

3

네트워크 장애

4

미디어 장애

정답 ③

1 트랜잭션 장애

2 시스템 장애

3 네트워크 장애 ✓

4 미디어 장애

해설

데이터베이스 장애 유형은 트랜잭션 장애·시스템 장애·미디어 장애로 구분합니다. 네트워크 장애는 포함되지 않습니다.

QUESTION 11

객관식

여러 비소멸성 저장 장치로 데이터 복사본을 만들어, 어떤 장애가 발생해도 데이터를 손실 없이 영구 저장할 수 있는 저장 장치는?

1

캐시 저장 장치

2

휘발성 저장 장치

3

비휘발성 저장 장치

4

안정 저장 장치

정답 ④

1 캐시 저장 장치

2 휘발성 저장 장치

3 비휘발성 저장 장치

4 안정 저장 장치 ✓

해설

여러 비휘발성 저장 장치에 복사본을 두어 데이터 손실을 막는 것이 안정 저장 장치(stable storage)입니다.

QUESTION 12

객관식

디스크 블록에 저장되어 있는 데이터 X를 메인 메모리의 버퍼 블록으로 이동시키는 연산은?

1 input(X)

2 output(X)

3 read(X)

4 write(X)

정답 ①

1 input(X)



2 output(X)

3 read(X)

4 write(X)

해설

디스크 → 메인 메모리 버퍼로의 블록 이동은 input(X)입니다. (반대 방향은 output(X))

QUESTION 13

객관식

프로그램의 변수 값을 메인 메모리 버퍼 블록에 있는 데이터 X에 기록하는 연산은?

1 input(X)

2 output(X)

3 read(X)

4 write(X)

정답 ④

1 input(X)

3 read(X)

2 output(X)

4 **write(X)** ✓

해설

프로그램 변수 값을 버퍼 블록의 X에 기록하는 연산은 write(X)입니다. (버퍼 → 변수는 read(X))

QUESTION 14

객관식

데이터베이스의 회복 기법에 해당하지 않는 것은?

1

로그 회복 기법

2

미디어 회복 기법

3

검사 시점 회복 기법

4

로킹 회복 기법

정답 ④

1 로그 회복 기법

2 미디어 회복 기법

3 검사 시점 회복 기법

4 로킹 회복 기법 ✓

해설

로킹은 병행 제어 기법입니다. 회복 기법에는 로그 기반·검사 시점·미디어 회복 기법 등이 있습니다.

QUESTION 15

객관식

덤프를 이용하는 회복 기법으로, 장애가 발생하면 가장 최근에 복사해 둔 덤프로 장애 발생 이전의 일관된 상태로 복구하는 것은?

1

로그 이용 회복 기법

2

미디어 회복 기법

3

검사 시점 회복 기법

4

로킹 회복 기법

정답 ②

1 로그 이용 회복 기법

3 검사 시점 회복 기법

2 미디어 회복 기법 ✓

4 로킹 회복 기법

해설

덤프(전체 복사본)를 이용해 복구하는 것은 미디어 회복 기법입니다.

병행 제어 없이 여러 트랜잭션을 동시에 수행할 때 발생할 수 있는 문제점이 아닌 것은?

1

갱신 분실

2

중복성

3

모순성

4

연쇄 복귀

정답 ②

1 갱신 분실

3 모순성

2 중복성 ✓

4 연쇄 복귀

해설

병행 수행 시 발생 문제는 갱신 분실·모순성·연쇄 복귀 등입니다. '중복성'은 해당하지 않습니다.

병행 제어 없이 여러 트랜잭션을 동시에 수행할 때 발생할 수 있는 문제점 중 다음 설명이 의미하는 것은?

하나의 트랜잭션이 수행한 데이터 변경 연산의 결과를 다른 트랜잭션이 덮어 써, 변경 연산이 무효화되는 것

1 갱신 분실

2 중복성

3 모순성

4 연쇄 복귀

정답 ①

1 갱신 분실



2 중복성

3 모순성

4 연쇄 복귀

해설

한 트랜잭션의 갱신 결과가 다른 트랜잭션에 의해 덮어쓰여 사라지는 현상은 갱신 분실(lost update)입니다.

병행 제어의 목적으로 옳지 않은 것은?

1 시스템 활용도 최대화

2 응답 시간 최대화

3 데이터베이스 공유 최대화

4 데이터베이스 일관성 유지

정답 ②

1 시스템 활용도 최대화

3 데이터베이스 공유 최대화

2 응답 시간 최대화 ✓

4 데이터베이스 일관성 유지

해설

응답 시간은 '최소화'해야 합니다. 병행 제어는 활용도·공유 최대화와 일관성 유지를 목적으로 합니다.

로킹 기법에서 상대가 독점한 데이터의 unlock 연산을 기다리며 수행을 중단하고 있는 상태를 무엇이라 하는가?

1 중단 상태

2 회피 상태

3 대기 상태

4 교착 상태

정답 ④

1 중단 상태

3 대기 상태

2 회피 상태

4 교착 상태



해설

서로 상대가 잠근 데이터의 해제를 무한정 기다리며 진행하지 못하는 상태가 교착 상태(deadlock)입니다.

로킹(locking) 단위에 대한 설명으로 옳지 않은 것은?

1 로킹 단위가 커질수록 병행성이 낮아진다.

2 로킹 단위가 커질수록 lock 연산의 개수가 적어진다.

3 로킹 단위가 작아질수록 제어 기법이 단순해진다.

4 데이터베이스도 로킹 단위가 될 수 있다.

정답 ③

1 로킹 단위가 커질수록 병행성이 낮아진다.

2 로킹 단위가 커질수록 lock 연산의 개수가 적어진다.

3 로킹 단위가 작아질수록 제어 기법이 단순해진다. ✓

4 데이터베이스도 로킹 단위가 될 수 있다.

해설

로킹 단위가 작아질수록 관리할 lock이 많아져 제어 기법이 오히려 복잡해집니다.

병행 제어에서 로킹 단위가 작은 경우에 대한 설명으로 옳지 않은 것은?

1 lock 연산의 개수가 많아진다.

2 병행성의 수준이 낮아진다.

3 병행 제어 기법이 복잡해진다.

4 교착 상태가 발생하는 경우가 많아진다.

정답 ②

1 lock 연산의 개수가 많아진다.

3 병행 제어 기법이 복잡해진다.

2 병행성의 수준이 낮아진다. ✓

4 교착 상태가 발생하는 경우가 많아진다.

해설

로킹 단위가 작으면 병행성 수준은 오히려 '높아집니다'. 나머지 ①③④는 모두 작은 단위의 특징입니다.

로킹 기법에서 트랜잭션이 데이터에 대한 독점권을 요청하는 연산은?

1 lock 연산

2 unlock 연산

3 redo 연산

4 undo 연산

정답 ①

1 lock 연산



2 unlock 연산

3 redo 연산

4 undo 연산

해설

데이터에 대한 독점권(잠금)을 요청하는 연산은 lock 연산이고, 해제는 unlock 연산입니다.

2단계 로킹 규약에 대한 설명으로 옳은 것은?

1 lock만 실행할 수 있고 unlock은 실행할 수 없는 축소 단계가 있다.

2 unlock만 실행할 수 있고 lock은 실행할 수 없는 확장 단계가 있다.

3 직렬 가능성이 보장된다.

4 교착 상태가 발생하지 않도록 예방할 수 있다.

정답 ③

1

lock만 실행할 수 있고 unlock은 실행할 수 없는 축소 단계가 있다.

2

unlock만 실행할 수 있고 lock은 실행할 수 없는 확장 단계가 있다.

3

직렬 가능성이 보장된다. ✓

4

교착 상태가 발생하지 않도록 예방할 수 있다.

해설

2단계 로킹 규약(2PL)은 직렬 가능성을 보장합니다. 확장 단계는 lock만, 축소 단계는 unlock만 가능하며, 교착 상태는 예방하지 못합니다.

B

주관식 문제

Short Answer & Descriptive · Q24 – Q41

QUESTION 24

주관식

트랜잭션이 무엇인지 설명하시오.

정답

하나의 작업을 수행하기 위해 필요한 데이터베이스 연산들을 모아 놓은 논리적인 작업의 단위입니다.

QUESTION 25

주관식

데이터 무결성과 일관성을 보장하기 위한 트랜잭션의 특성 네 가지가 무엇인지 설명하시오.

정답

원자성

Atomicity — 연산은 모두 반영되거나 전혀 반영되지 않는다(all-or-nothing)

일관성

Consistency — 수행 후에도 일관된 데이터베이스 상태를 유지한다

격리성

Isolation — 동시 수행 트랜잭션이 서로 간섭하지 않는다

지속성

Durability — 완료된 결과는 영구적으로 반영된다

한빛 은행에서 A가 고객 인증을 거쳐 잔액을 조회한 후, 타인에게 송금하는 도중에 장애가 발생했을 경우의 부작용을 방지할 수 있는 트랜잭션의 특성은?

정답

원자성 (atomicity)

해설

송금 연산의 일부만 반영되면 출금은 되고 입금은 안 되는 잔액 불일치가 생깁니다. 연산을 전부 반영하거나 전혀 반영하지 않는 all-or-nothing(원자성)을 보장하면 장애 시 트랜잭션 전체가 취소되어 부작용을 막을 수 있습니다.

다음 설명에서 ㉠과 ㉡가 각각 무엇인지 쓰시오.

트랜잭션의 수행이 성공적으로 완료됐음을 선언하는 연산은 (㉠)이고, 트랜잭션을 수행하기 이전 상태로 되돌리기 위해 작업 취소를 선언하는 연산은 (㉡)이다.

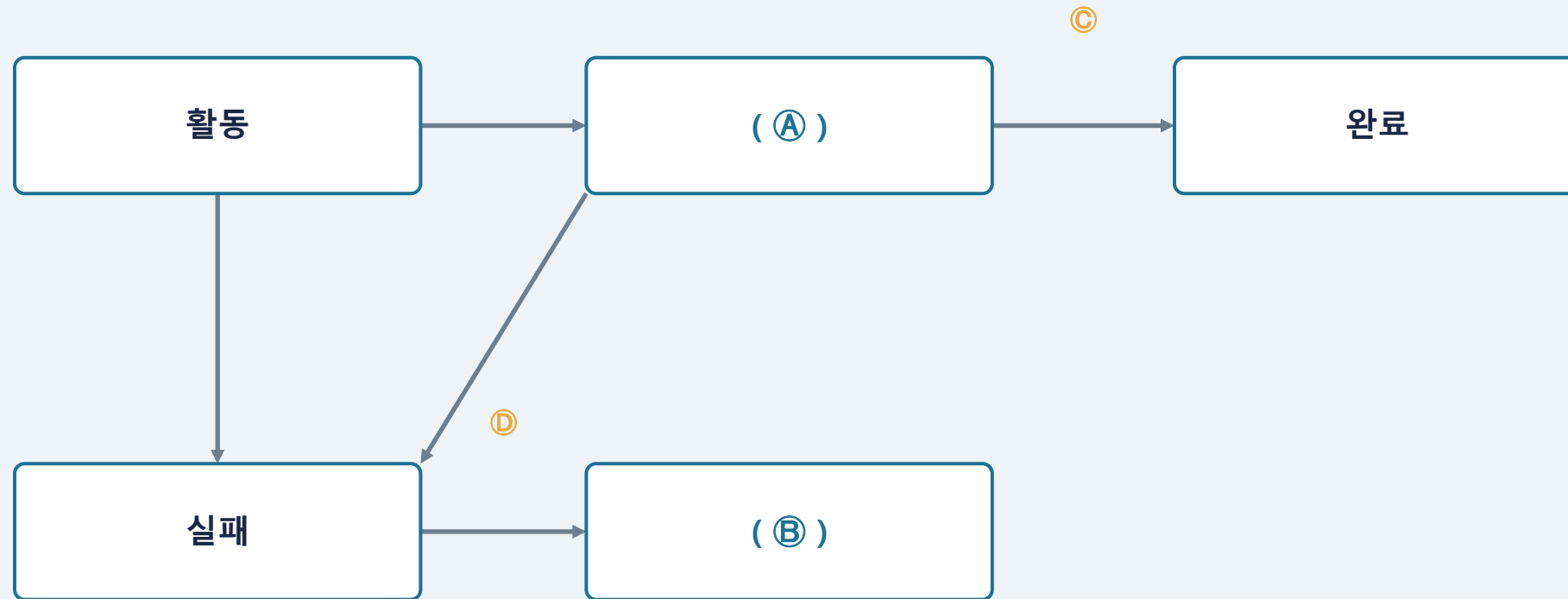
정답

**COMMIT****ROLLBACK**

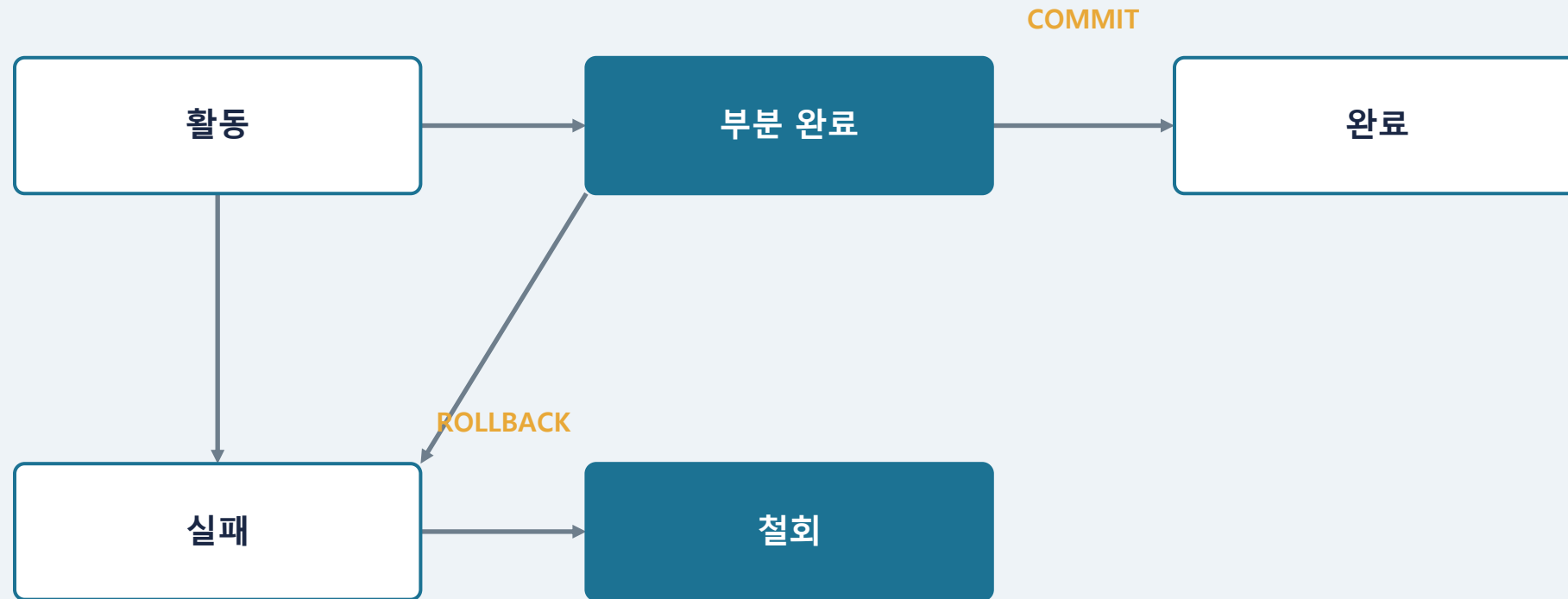
해설

COMMIT은 트랜잭션의 성공적 완료를 확정해 결과를 영구 반영하고, ROLLBACK은 실패한 트랜잭션의 작업을 취소해 이전 상태로 되돌립니다.

트랜잭션의 5가지 상태를 나타내는 다음 그림에서 ㉠·㉡는 어떤 상태이고, ㉢·㉣는 어떤 연산인가?



정답



Ⓐ 부분 완료

Ⓑ 철회

Ⓒ COMMIT 연산

Ⓓ ROLLBACK 연산

데이터베이스 회복을 위해 복사본을 만드는 방법에 대한 다음 설명에서 ㉠과 ㉡를 채우시오.

(㉠) : 데이터베이스 전체 내용을 다른 저장 장치에 주기적으로 복사하는 방법

(㉡) : 변경 연산이 수행될 때마다 변경된 데이터의 이전 값과 새로운 값을 별도의 파일에 기록하는 방법

정답



덤프(dump)



로그(log)

해설

덤프는 전체 내용을 통째로 복사하는 미디어 회복용 방법이고, 로그는 변경 연산마다 이전 값과 새 값을 기록해 redo·undo에 사용하는 로그 기반 회복용 방법입니다.

데이터베이스 회복을 위한 기본 연산에 대한 다음 설명에서 ㉠과 ㉡를 채우시오.

- (㉠) 연산 : 로그에 기록된 변경 연산 후의 새로운 값을 이용해 변경 연산을 재실행
- (㉡) 연산 : 로그에 기록된 변경 연산 이전의 값을 이용해 변경 연산을 취소

정답



Redo (재실행)



Undo (취소)

해설

Redo는 로그의 '새로운 값'으로 연산을 다시 실행해 완료 트랜잭션을 복구하고, Undo는 로그의 '이전 값'으로 연산을 취소해 미완료 트랜잭션을 되돌립니다.

트랜잭션이 데이터를 변경하면 부분 완료되기 전이라도 즉시 데이터베이스에 반영하고, 회복을 대비해 변경 내용을 로그에 기록하며, 회복 시 redo와 undo가 모두 가능한 회복 기법은?

정답

즉시 갱신 회복 기법

해설

변경을 즉시 DB에 반영하므로, 회복 시 완료된 트랜잭션은 Redo로, 미완료 트랜잭션은 Undo로 처리합니다. (반대로 지연 갱신 회복 기법은 commit 전까지 DB에 반영하지 않아 Undo가 필요 없습니다.)

QUESTION 32

주관식

rollback 연산에 대해 간단히 설명하시오.

정답

트랜잭션 수행에 실패했음을 선언하는 연산입니다. 트랜잭션이 실패한 경우 지금까지 수행한 작업을 모두 취소하고, 트랜잭션 수행 이전의 상태로 데이터베이스를 되돌리는 작업을 수행합니다.

검사 시점 회복 기법이 검사 시점을 이용하지 않는 일반 로그 회복 기법에 비해 어떤 장점이 있는지 설명하십시오.

정답

장애가 발생하면 가장 가까운 검사 시점 이전의 트랜잭션은 회복 대상에서 제외하고, 검사 시점 이후의 트랜잭션에 대해서만 회복 작업을 수행합니다. 따라서 불필요한 회복 작업을 줄여 데이터베이스 회복 시간이 단축됩니다.

직렬 스케줄과 직렬 가능 스케줄의 공통점과 차이는 무엇인가?

정답

공통점

두 스케줄의 실행 결과(데이터베이스 최종 상태)가 서로 동일합니다.

차이점

직렬 스케줄은 트랜잭션을 순차 실행하지만, 직렬 가능 스케줄은 인터리빙을 사용하는 '비직렬' 스케줄입니다.

2단계 로킹 규약에 대한 다음 설명에서 ㉠과 ㉡를 채우시오.

기본 로킹 규약의 문제를 해결하고 직렬 가능성을 보장하기 위해 lock·unlock 연산의 실행 시점에 규약을 추가한 것이 2단계 로킹 규약이다.

(㉠) 단계 : lock 연산만 수행할 수 있고 unlock 연산은 수행할 수 없는 단계

(㉡) 단계 : unlock 연산만 수행할 수 있고 lock 연산은 수행할 수 없는 단계

정답



확장(growing) 단계



축소(shrinking) 단계

해설

확장 단계에서는 필요한 lock을 모두 획득하기만 하고, 한 번 unlock을 시작하면 축소 단계로 들어가 더 이상 lock을 걸 수 없습니다. 이 규칙이 직렬 가능성을 보장합니다.

A→B로 100원을 이체하는 T1과 C에서 200원을 출금하는 T2의 로그·DB 기록 상태이다. 각 장애 시점의 회복 연산을 구하시오.

로그(Log)

데이터베이스 반영 / 장애 시점

⟨T1, start⟩

⟨T1, A, 1000, 900⟩

⟨T1, B, 2000, 2100⟩

⟨T1, commit⟩

A = 900, B = 2100 (DB 반영)

⟨T2, start⟩

⟨T2, C, 3000, 2800⟩

C = 2800 (DB 반영)

⟨T2, commit⟩

◀ 장애1

◀ 장애2

◀ 장애3

◀ 장애4

정답

(1) T1

지연 갱신 회복 기법

(2) T2

즉시 갱신 회복 기법

(3) 장애1

회복 작업 불필요 (T1 commit 이전)

(4) 장애2

Redo(T1)

(5) 장애3

Redo(T1), Undo(T2)

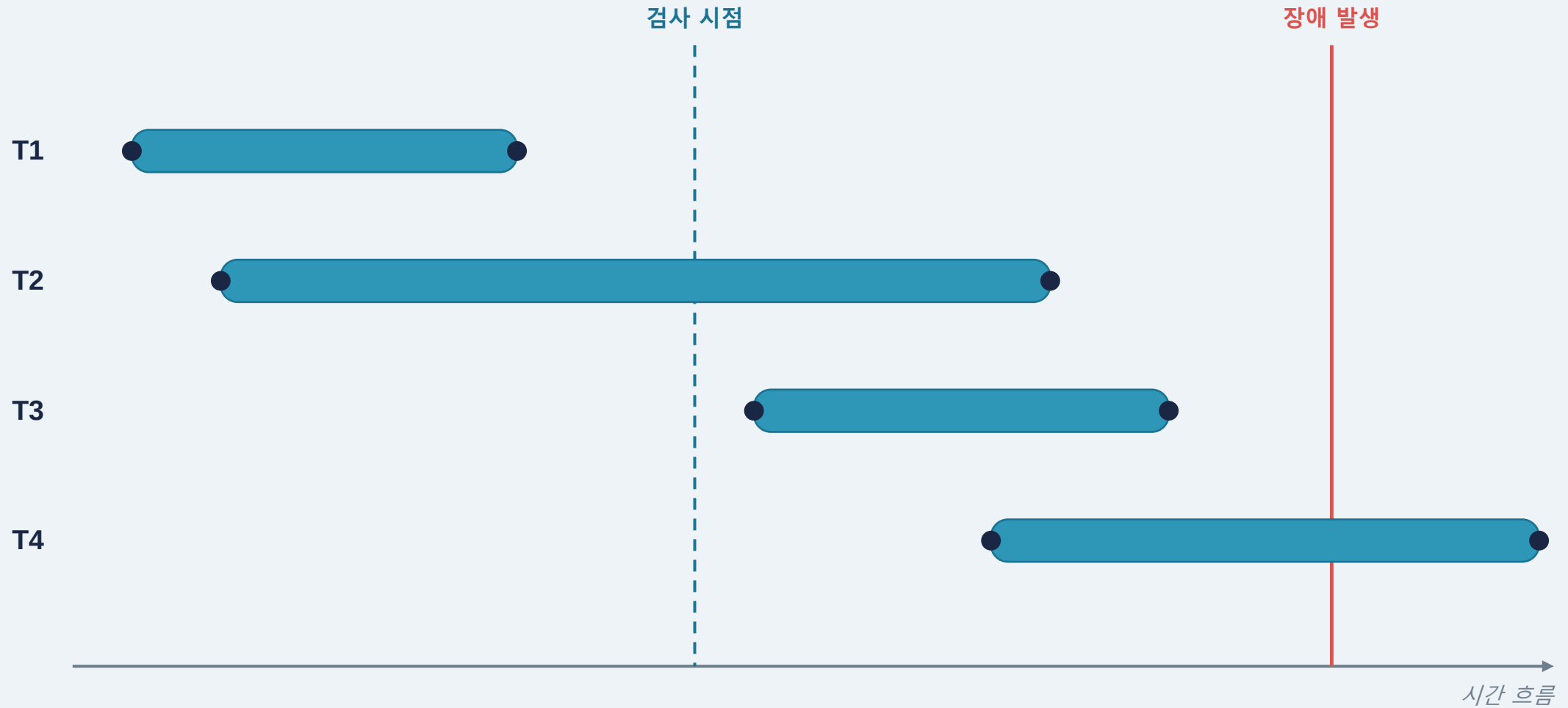
(6) 장애4

Redo(T1), Redo(T2)

QUESTION 37

주관식

네 개의 트랜잭션이 수행되는 다음 그림(검사 시점·장애 발생 표시)을 보고 회복 연산을 구하시오.



정답

(1) **Undo 필요** (검사 시점 미고려)

T4

(2) **Redo 필요** (검사 시점 미고려)

T1, T2, T3

(3) 검사 시점을 고려한 경우

T1 : 회복 작업 불필요

T2 : 검사 시점 이후 연산만 Redo

T3 : 트랜잭션 전체 연산 Redo

T4 : 트랜잭션 전체 연산 Undo

QUESTION 38

주관식

병행 제어가 무엇인지 설명하시오.

정답

여러 개의 트랜잭션이 동시에 수행되면서 같은 데이터에 접근하여 연산을 수행하더라도, 문제가 발생하지 않고 정확한 결과를 얻을 수 있도록 트랜잭션의 수행을 제어하는 것을 의미합니다.

QUESTION 39

주관식

트랜잭션 스케줄이 무엇인지 설명하시오.

정답

트랜잭션에 포함되어 있는 연산들을 실행하는 순서를 의미합니다.

세 가지 유형의 트랜잭션 스케줄에 대한 다음 설명에서 ㉠은 무엇인가?

직렬 스케줄 : 인터리빙 없이 트랜잭션별로 연산을 순차 실행하는 것

비직렬 스케줄 : 인터리빙을 이용해 트랜잭션들을 병행 실행하는 것

(㉠) : 직렬 스케줄과 동일한 정확한 결과를 생성하는 비직렬 스케줄

정답

㉠ 직렬 가능 스케줄

해설

직렬 가능 스케줄은 비직렬 스케줄이지만 그 실행 결과가 어떤 직렬 스케줄의 결과와 같아, 병행 수행을 허용하면서도 정확성을 보장하는 스케줄입니다.

다음과 같이 병행 제어 없이 두 트랜잭션 T1과 T2가 수행될 때, 최종 X의 값은? (처음 X = 100)

처음 X = 100

트랜잭션 T1	트랜잭션 T2
read(X);	
X = X + 30;	
write(X);	
	read(X);
rollback;	
	X = X - 50;
	write(X);

시간 ↓

정답

최종 $X = 80$

해설

T1이 write(X)로 130을 기록한 뒤 T2가 read(X)로 그 값(130)을 읽습니다. 이후 T1이 rollback되어도 T2는 이미 읽은 130으로 $X = 130 - 50 = 80$ 을 계산해 기록하므로, 최종 X는 80이 됩니다. (T1의 갱신이 무효화되는 갱신 분실 상황)



수고하셨습니다

10장 회복과 병행 제어 · 연습문제 풀이

박상돈 조교수 · 대전대학교 컴퓨터공학과