데이터베이스론	국가 전산 7급	2020년 9월 26일

♠ 필기합격인원/합격선(46명/76.66점) - 선발예정인원 33명 ♠

1. 관계 데이터 모델의 키와 제약조건에 대한 설명으로 옳은 것만을 모두 고르면? [2020년 국가 7급]

ㄱ. R(A, B, C) 릴레이션에서 기본키가 복합키 (A, B)이고 B가 외래키라면, 참조 무결성에 의해 B는 널값을 가질 수 있다.

ㄴ. 주어진 릴레이션 R의 속성들의 부분집합 X에 대해, 어떤 튜플도 동일한 값을 가지지 않는다면, 이러한 속성의 집합 X를 그 릴레이션의 슈퍼키라고 한다.

ㄷ. 참조 무결성 제약은 참조할 수 없는 외래키 값을 가져서는 안 된다는 것을 의미한다.

ㄹ. 대체키와 외래키는 유일성과 최소성을 모두 만족해야 한다.

---

① ㄱ, ㄴ ② ㄱ, ㄹ

③ ㄴ, ㄷ ④ ㄷ, ㄹ

♤ **관계 데이터 모델의 키와 제약조건**

ㄱ. R(A, B, C) 릴레이션에서 기본키가 복합키 (A, B)이고 B가 외래키라면, 참조 무결성에 의해 B는 널값을 가질 수 **있다.(×)**

→ 복합키를 구성하는 키도 기본키이므로 널값을 가질 수 **없다.**

ㄹ. 대체키와 외래키는 유일성과 최소성을 모두 **만족해야 한다.(×)**

→ 외래키는 유일성을 **만족하지 않는다.**

// 외래키가 기본키의 일부로 참가하는 경우(밑줄은 기본키, 이태리체는 외래키)

수강

학번	*과목*	점수
100	디비	90
100	소공	80
200	디비	90

교과목

과목	학점	구분
디비	3	필수
소공	3	선택
보안	2	필수

- 수강 테이블의 과목은 외래키이다.(학번도 외래키가 될 수 있다)
- 수강 테이블의 기본키인 복합키 (학번, 과목)에서 학번이나 과목은 모두 널값을 가질 수 **없다.**
- 기본키는 널(null) 값을 가질 수 **없으므로**(개체 무결성)
- 외래키가 **기본키**의 일부로 **참가하는 경우**는 **널(null)** 값을 가질 수 **없다.**
- 외래키 과목은 유일성을 **만족하지 않는다.** 여러 개가 존재할 수 있다.

정답 : ③

2. 릴레이션 스키마의 표현은 릴레이션명(속성명1:도메인1, 속성명2:도메인2, …, 속성명n:도메인n)이다. 이 표현을 따르는 릴레이션 스키마 A(a:int, x:int, c:int)와 B(b:int, x:int, d:int)에 대한 관계대수식 $\Pi_{a,b}(\sigma_{a>10}(A \bowtie_{A.x=B.x} B))$와 동등한 관계대수식은? [2020년 국가 7급]

① $\Pi_{a,b}(\sigma_{a>10}(A) \times B)$
② $\sigma_{a>10}(\Pi_{a,b}(A) \bowtie_x B)$
③ $\Pi_{a,b}(A \bowtie_a \sigma_{a>10}(B))$
④ $\Pi_{a,b}(\sigma_{a>10}(A) \bowtie_{A.x=B.x} B)$

♤ 동등한 관계대수식 – 질의최적화

// 주어진 내용을 테이블로 나타내면 다음과 같다.

A

a	x	c
10	1	100
20	2	200
30	3	300

B

b	x	d
40	1	400
50	2	500
60	3	600

↓ $(A \bowtie_{A.x = B.x} B)$

a	x	c	b	d
10	1	100	40	400
20	2	200	50	500
30	3	300	60	600

↓ $(\sigma_{a>10}(A \bowtie_{A.x = B.x} B))$

a	x	c	b	d
20	2	200	50	500
30	3	300	60	600

↓ $\pi_{a,b}(\sigma_{a>10}(A \bowtie_{A.x = B.x} B))$

a	b
20	50
30	60

- $\pi_{a,b}(\sigma_{a>10}(A \bowtie_{A.x = B.x} B))$ //테이블 A, B를 조인한 후에 선택 연산 실시

↓

- Select a, b From A, B Where A.x=B.x And A.a>10;

↓ 질의최적화를 하면

- $\pi_{a,b}(\sigma_{a>10}(A) \bowtie_{A.x = B.x} B)$ //보다 제한적인 선택 연산 후에 조인 실시

정답 : ④

3. 스키마가 사원(사원번호, 직급, 보너스)인 사원 테이블의 인스턴스가 〈보기 1〉과 같을 때, 〈보기 2〉의 ㉠, ㉡에 들어갈 말로 옳게 짝지은 것은? [2020년 국가 7급]

<보기 1>

사원

사원번호	직급	보너스
1000	과장	500
1001	과장	500
1002	사원	200

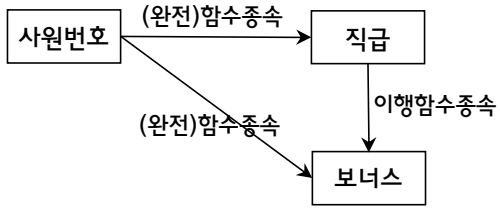
<보기 2>

이 사원 테이블에서 보너스는 직급별로 결정된다.
이때 직급과 보너스 속성은 모두 기본키인 사원번호에 의해 결정되지만 사원번호가 직급을 결정하고 직급이 보너스를 결정하는 (㉠) 관계를 가지고 있으므로, 이 테이블은 (㉡)이라고 할 수 있다.

	㉠	㉡
①	완전함수종속	제2정규형
②	완전함수종속	제3정규형
③	이행함수종속	제2정규형
④	이행함수종속	제3정규형

♤ 정규형

// 함수종속(FD)의 관계를 그림으로 그리면 다음과 같다.



- 함수종속 FD : 사원번호→직급, 직급→보너스
- **이행함수종속**이 존재하므로 제3정규형이 아니고, **제2정규형**에 속한다.

// 제3정규형(3NF; third normal form)

어떤 릴레이션이 2NF이고,
기본키에 속하지 않은 모든 속성이 기본키에 이행함수종속이 아니면 제3정규형에 속한다.

정답 : ③

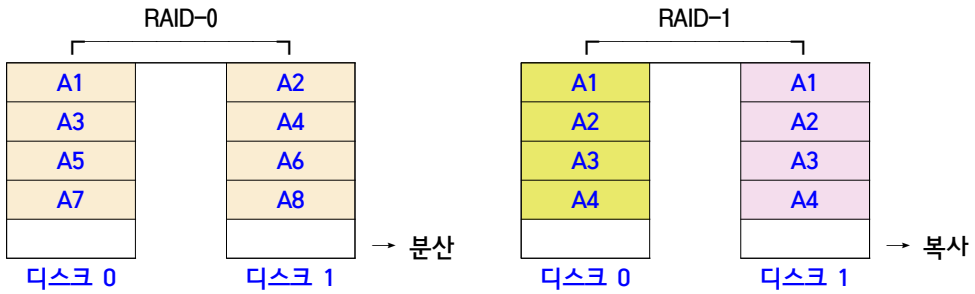
4. 데이터베이스 시스템에서 데이터 저장 요구량이 빠르게 증가하고 있어서 많은 수의 디스크가 요구된다. 다수의 디스크 드라이브를 사용하여 저장 용량을 늘리고 읽기와 쓰기를 병렬로 수행하기도 하며, 디스크의 고장에 대비하기 위해 RAID(Redundant Arrays of Independent Disks)를 구성하여 활용한다. 이와 같은 RAID에서 1 TByte 디스크 드라이브 6개를 이용하여 RAID를 구성할 때, 구성된 RAID의 저장 용량이 가장 작은 구성 방법은? [2020년 국가 7급]

① 레벨 0　　② 레벨 1
③ 레벨 5　　④ 레벨 6

♤ RAID

RAID 0 : 적어도 2개의 디스크, **분산**, 패리티(오류 검출 기능) 없음 → 신속한 입출력
RAID 1 : 적어도 2개의 디스크, **복사**, 패리티(오류 검출 기능) 없음 → 안정성
RAID 2 : 적어도 3개의 디스크, **분산**, **비트** 레벨, 전용 해밍코드 패리티 디스크 사용
RAID 3 : 적어도 3개의 디스크, **분산**, **바이트** 레벨, 전용 패리티 디스크 사용
RAID 4 : 적어도 3개의 디스크, **분산**, **블록** 레벨, 전용 패리티 디스크 사용
RAID 5 : 적어도 3개의 디스크, **분산**, 블록 레벨, **패리티 각 디스크에 분산** → RAID 4 **개선**
RAID 6 : 적어도 4개의 디스크, **분산**, 블록 레벨, **패리티 정보 2중 분산 기록**

- RAID 0은 **분산** 기술로 데이터를 여러 디스크에 분산 저장한다. 저장 용량이 가장 **크다.**
- RAID 1은 **복사** 기술로 데이터를 이중으로 저장한다. 해서, 저장 용량이 가장 **작다.**



- RAID 0에서 사용 가능한 최대 용량(**분산**) = 전체 디스크 수 = **6** TByte
- RAID 1에서 사용 가능한 최대 용량(**복사**) = 전체 디스크 수 / 2 = 6 / 2 = **3** TByte
- RAID 5에서 사용 가능한 최대 용량(**분산**) = 전체 디스크 수 − 1 = 6 − 1 = **5** TByte
  ↳ RAID 5는 패리티 정보를 기록하기 위한 하나의 디스크 용량이 필요하다.
- RAID 6에서 사용 가능한 최대 용량(**분산**) = 전체 디스크 수 − 2 = 6 − 2 = **4** TByte
  ↳ RAID 6은 패리티 정보를 기록하기 위한 2개 디스크 용량이 필요하다.**(패리티 2중 기록)**

정답 : ②

5. 다음은 산업통계 테이블이다. 이 테이블을 대상으로 아래 결과를 출력하고자 한다. 이를 위한 SQL 질의문은? [2020년 국가 7급]

산업통계

대분류	소분류	종사자수	평균연봉
제조업	반도체	10000	4000
제조업	스마트폰	5000	5000
제조업	가전	7000	6000
IT	SW개발	6000	4000
IT	게임개발	3000	3000
서비스업	미용	4000	3500
서비스업	유통	9000	4500

↓SQL 실행결과

대분류	소분류	종사자수	평균연봉
제조업	반도체	10000	4000
IT	SW개발	6000	4000
서비스업	유통	9000	4500

---

① SELECT * FROM 산업통계 WHERE 종사자수 >= 600;
② SELECT * FROM 산업통계
WHERE 종사자수 = (SELECT MAX(종사자수) FROM 산업통계);
③ SELECT A.* FROM 산업통계 A,
(SELECT 대분류, MAX(종사자수) AS 종사자수 FROM 산업통계
GROUP BY 대분류) B
WHERE A.종사자수 = B.종사자수 AND A.대분류 = B.대분류;
④ SELECT 대분류, 소분류, MAX(종사자수), 평균연봉 FROM 산업통계
GROUP BY 대분류;

♤ SQL 실행결과

• SELECT A.* FROM 산업통계 A,
(SELECT **대분류, MAX(종사자수)** AS 종사자수 FROM 산업통계 **GROUP BY 대분류**) B
↓
대분류, MAX(종사자수) : 종사자수가 **최대**인 레코드를 출력

정답 : ③

6. 버킷 용량 C=2인 버킷들의 주소공간이 0~(N-1)인 해시에서 키를 k로 하는 해시함수 h(k) = k mod N이고, N값은 5이다. 이때 일련의 k 값들이 3, 5, 7, 9, 1, 13, 18, 14 순서로 삽입된 후 4번 버킷의 값은? (단, 버킷에서의 충돌은 충돌 발생 버킷주소 i에 대하여 (i+1) mod N을 통해 지정하는 선형탐색 개방주소법을 이용하여 해결한다) [2020년 국가 7급]

① 5, 14 ② 3, 13
③ 9, 14 ④ 9, 18

♤ **선형탐색 개방주소법**

- 주어진 문제에서 버킷 용량 C=2인 버킷은 각 버킷이 2개의 슬롯으로 구성된 것이다.
- 선형탐색 개방주소법은 오버플로(overflow)가 발생하면, 색인 증가를 1로 처리한다.

// 키 : 3, 5, 7, 9, 1, 13, 18, 14

0	1	2	3	4	→ 버킷주소
5	1	7	3	9	
14			13	18	
			↑ 18	↑ 14	

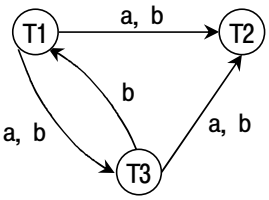
h(3) = 3 mod 5 = 3	키 3은 버킷주소 3에 저장
h(5) = 5 mod 5 = 0	키 5는 버킷주소 0에 저장
h(7) = 7 mod 5 = 2	키 7은 버킷주소 2에 저장
h(9) = 9 mod 5 = 4	키 9는 버킷주소 4에 저장
h(1) = 1 mod 5 = 1	키 1은 버킷주소 1에 저장
h(13) = 13 mod 5 = 3	1차 충돌 발생, 버킷크기가 2이므로 키 13은 버킷주소 3에 저장
h(18) = 18 mod 5 = 3	1차 충돌 발생 및 오버플로 발생 색인 증가를 1, 키 18은 버킷주소 4에 저장
h(14) = 14 mod 5 = 4	1차 충돌 발생 및 오버플로 발생 색인 증가를 1 키 14는 버킷주소 0에 저장

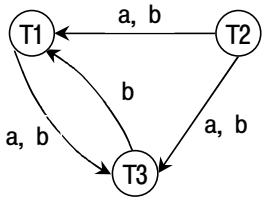
- 4번 버킷의 값은 **9, 18**이다.

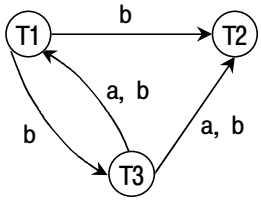
정답 : ④

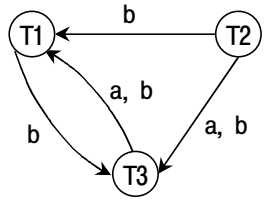
7. 다음 트랜잭션 스케줄의 선행그래프(precedence graph)로 옳은 것은? (단, t_i는 시간단위, T_j는 트랜잭션, r(a)는 a항목 읽기, w(a)는 a 항목 쓰기를 나타낸다) [2020년 국가 7급]

시간	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}
T_1	r(a)	w(a)				r(b)		w(b)		
T_2					r(a)					r(b)
T_3			r(a)	w(a)			r(b)		w(b)	

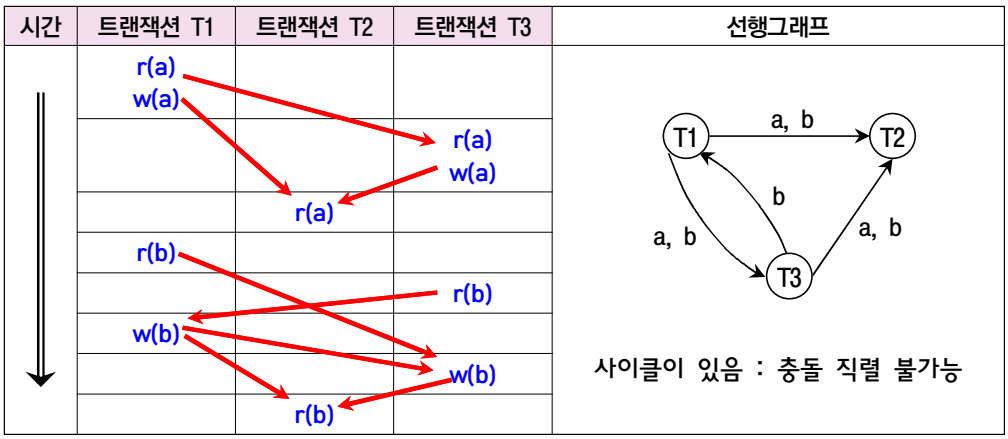
① 

② 

③ 

④ 

♤ 트랜잭션 스케줄의 선행그래프(precedence graph)



- 트랜잭션 Ta에서 read(x)한 것을 Tb에서 write(x)하면 간선을 생성한다.(Ta→Tb)
- 트랜잭션 Ta에서 write(x)한 것을 Tb에서 read(x)하면 간선을 생성한다.(Ta→Tb)
- 트랜잭션 Ta가 write(x)한 것을 바로 Tb가 write(x)하면 간선을 생성한다.(Ta→Tb)

정답 : ①

8. 다음 SQL 구문에 대한 설명으로 옳지 않은 것은? [2020년 국가 7급]

```
CREATE ASSERTION NO_DESIGNERS_IN_SEOUL
AS CHECK
( NOT EXISTS
    ( SELECT * FROM EMPLOYEE E, DEPARTMENT D
      WHERE E.DEPTNO = D.DEPTNO
         AND E.JOB = 'DESIGNER'
         AND D.LOC = 'SEOUL' )
);
```

① 명시된 이벤트가 발생할 때마다 DBMS가 자동적으로 수행하는 구문이다.
② 대상 테이블과 관련된 SQL의 갱신문 수행 시 위의 구문이 검사된다.
③ 위의 SQL이 불필요할 경우 DROP문으로 제거할 수 있다.
④ 위의 SQL에서 명세하고 있는 조건에 따라 DB연산이 수행되지 않을 수 있다.

♤ SQL 구문 – ASSERTION

• CREATE ASSERTION : 주장 생성(일반적인 제약조건을 명시)

```
CREATE ASSERTION NO_DESIGNERS_IN_SEOUL   //제약조건 이름
AS CHECK                                 //Check 다음에 제약조건 명시
( NOT EXISTS                             //제약조건 명시
    ( SELECT * FROM EMPLOYEE E, DEPARTMENT D
      WHERE E.DEPTNO = D.DEPTNO
         AND E.JOB = 'DESIGNER' AND D.LOC = 'SEOUL' )
);
```

// 트리거(trigger)
• 명시된 이벤트가 발생할 때마다 **DBMS가 자동으로 실행**하는 (사용자가 정의하는) 프로시저
• 트리거는 데이터베이스 무결성을 유지하기 위한 강력한 도구이다.
• 표준 SQL3에 포함되어 있다, 대부분의 상용 DBMS에서 제공한다.

// 주장(assertion)
• 주장은 트리거와 다르게, 제약조건을 위반하는 연산은 수행되지 않도록 하기 위한 것이다.
• 주장은 DB가 만족하길 바라는 조건을 직접적으로 표현한다.
• 만약, 주장의 조건을 검사하여 참이면 변경을 허용한다.

정답 : ①

9. XML Schema와 DTD에 대한 비교 중 옳지 않은 것으로만 묶은 것은? [2020년 국가 7급]

	구분	XML Schema	DTD
ㄱ	구조	복잡함	상대적으로 간결함
ㄴ	문법	전용문법	XML문법
ㄷ	Namespace	지원함	지원하지 못함
ㄹ	확장성	문자열 치환을 통한 확장성	객체지향적인 확장성

① ㄱ, ㄴ ② ㄱ, ㄷ ③ ㄴ, ㄹ ④ ㄷ, ㄹ

♤ XML Schema와 DTD

// 올바르게 고치면

	구분	XML Schema	DTD
ㄱ	구조	복잡함	상대적으로 간결함
ㄴ	문법	XML 문서와 같은 구문 규칙을 사용	자신 고유의 구문
ㄷ	Namespace	지원함	지원하지 못함
ㄹ	확장성	객체지향적인 확장성 우수	문자열 치환을 통한 확장

DTD	• DTD는 **XML 문서 형식**을 기술해 둔 것이다. • DTD는 같은 타입의 여러 XML 문서가 참조하기 위한 것이다. • 예 : 두 회사가 같은 상품에 대한 상거래에서는 하나의 DTD만 필요하다. • 한번 만들어둔 DTD를 다음의 XML 문서에 적용하면 된다. • DTD는 다양한 종류의 원소를 가진 트리구조에 대응된다.**(계층모델)** **// DTD의 제한점** ① DTD는 문자 스트링 타입만 허용한다,(타입 지원이 거의 없다) → DTD는 **재사용성이나 확장성은 부족**하다.**(문자열 치환을 통한 확장)** ② DTD는 **자신 고유의 구문**을 가지고 있어서 전문 처리기가 필요하다. ③ DTD 원소들은 일정한 순서를 명세하고 있다. → XML 문서에서는 정확히 DTD에 명세된 순서로 나타나야 한다. ④ DTD는 데이터 무결성을 표현하는데 취약하다.
XML 스키마	• XML 문서구조를 명세하기 위한 고급 데이터 정의 언어이다.**(DTD 문제점 보완)** → 문서구조는 원소, 데이터 타입, 관계타입, 범위, 기정의 값 등 • 사용자 정의 타입, 기본키, 외래키, 유일성, 무결성 제약조건을 명세할 수 있다. • XML 스키마는 데이터베이스 용어 및 기능과 매우 비슷하다. • XML 스키마는 **XML 문서와 같은 구문 규칙을 사용**한다.**(확장성 우수)** → 하나의 처리기로 XML 스키마와 XML 문서를 모두 처리할 수 있다. • XML 스키마는 **네임스페이스(namespace)**를 지원한다. • 네임스페이스는 문서에 사용된 태그 이름을 명확하게 구별하기 위한 것이다. • **XML 스키마 단점**은 DTD에 비해 문서구조가 매우 **복잡**하다. • XML 스키마는 원소와 속성에 다양한 데이터 타입을 명세할 수 있어서

정답 : ③

10. 즉시갱신(immediate update) 전략을 이용하는 회복 시스템에서 〈보기 1〉의 로그레코드 형식으로 〈보기 2〉의 로그레코드가 형성되어 있다. 복구는 redo 단계 실행 후 undo 단계를 실행한다. 이 시스템에서 복구 절차에 대한 설명으로 옳지 않은 것으로만 묶은 것은?

<보기 1>

○ Ti: 트랜잭션 번호가 i인 트랜잭션
○ <Ti, start>, <Ti, commit>, <Ti, abort>: Ti에 대한 시작, 완료, 철회를 기록하는 형식
○ <Ti, rollback>: Ti가 실행 중 rollback이 발생함을 기록하는 형식
○ <Ti, 데이터항목, 이전값, 이후값>: Ti가 데이터 항목을 변경할 때 기록하는 형식
○ <Ti, 데이터항목, 이전값>: Ti가 철회되기 위하여 undo를 실행할 때 기록하는 형식
○ <checkpoint, {트랜잭션리스트}>: 검사점 실행을 기록하는 형식
○ {트랜잭션리스트}: 검사점 실행시점에 실행 중에 있던 트랜잭션들의 목록을 포함하는 트랜잭션 리스트

<보기 2>

로그번호	로그레코드
1	<T0, start>
2	<T0, A, 100, 200>
3	<T1, start>
4	<T1, B, 200, 300>
5	<T0, C, 300, 400>
6	<checkpoint, {T0, T1}>
7	<T1, D, 500, 600>
8	<T3, start>
9	<T0, rollback>
10	<T4, start>
11	<T0, C, 300>
12	<T3, E, 600, 700>
13	<T0, A, 100>
14	<T4, F, 700, 800>
15	<T0, abort>
16	<T4, G, 800, 900>
17	<T3, H, 900, 950>
18	<T4, commit>

-고장 발생-

ㄱ. 복구가 시작되면 checkpoint 로그레코드를 찾아 undo 대상 트랜잭션 리스트를 {T0, T1}으로 초기화한다.
ㄴ. T0의 rollback을 처리하기 위하여 생성된 <T0, C, 300>, <T0, A, 100>은 복구과정에서 undo 단계 연산이 된다.
ㄷ. redo 단계를 완료하면 undo 대상 트랜잭션 리스트는 {T1, T3}로 구성된다.
ㄹ. redo 단계에서 로그레코드 <T1, D, 500, 600>, <T0, C, 300>, <T3, E, 600, 700>, <T0, A, 100>, <T4, F, 700, 800>, <T4, G, 800, 900>, <T3, H, 900, 950>은 redo 연산의 대상이 된다.
ㅁ. redo 단계에서는 redo 연산에 따른 로그레코드를 로그에 기록한다.
ㅂ. undo 단계에서는 <T3, H, 900>, <T3, E, 600>, <T3, abort>, <T1, D, 500>, <T1, B, 200>, <T1, abort>를 로그에 기록한다.

---

① ㄱ, ㅂ　　② ㄴ, ㅁ　　③ ㄷ, ㄹ　　④ ㄹ, ㅂ

♤ 즉시갱신

// 트랜잭션 실행 내용은 다음과 같다.

시간	T0	T1	T3	T4	비고
↓	<T0, start> <T0, A, 100, 200>				
		<T1, start> <T1, B, 200, 300>			
	<T0, C, 300, 400>				
	<checkpoint, {T0, T1}>		← undo 리스트에 {T0, T1}을 삽입		
		<T1, D, 500, 600>			
			<T3, start>		
	<T0, rollback>				T0 : 복귀
				<T4, start>	
	<T0, C, 300>				
			<T3, E, 600, 700>		
	<T0, A, 100>				
				<T4, F, 700, 800>	
	<T0, abort>				T0 : 철회
				<T4, G, 800, 900>	
			<T3, H, 900, 950>		
				<T4, commit>	T4 : 완료
	-고장 발생-				

・〈T0, abort〉 : T0는 복귀 연산을 실행한 상태 - undo 대상이 아님

・이유 : 복귀 연산은 트랜잭션 수행 중에 진행/완료되므로 회복할 필요가 없다.

redo 단계	① **redo** 복구는 **실패**한 트랜잭션에 대해서도 **redo** 복구를 수행한다.**(주의!)** ② **redo** 연산에서는 로그레코드를 **로그에 기록하지 않는다.** ③ **redo** 연산 대상 : **〈checkpoint〉** 이후에 실행된 내용 ・〈T1, D, 500, 600〉 〈T0, C, 300〉, 〈T3, E, 600, 700〉, 〈T0, A, 100〉 ・〈T4, F, 700, 800〉, 〈T4, G, 800, 900〉, 〈T3, H, 900, 950〉
undo 단계	① **undo** 단계에서는 로그레코드를 로그에 기록한다. ② **undo** 연산 대상 : **T3, T1 - 미완료** ・〈T3, H, 900〉, 〈T3, E, 600〉, 〈T3, abort〉 ・〈T1, D, 500〉, 〈T1, B, 200〉, 〈T1, abort〉

정답 : ②