

1. SQL을 실행할 때 디스크 접근 횟수를 줄여 검색 속도를 높이기 위해 사용되고, 필요한 데이터를 빠르게 검색하기 위해 사용되는 것으로 가장 옳은 것은?

- ① 인덱스(index)
- ② 트랜잭션(transaction)
- ③ 뷰(view)
- ④ 트리거(trigger)

2. 트랜잭션(transaction)의 특성 중 지속성(durability)에 대한 설명으로 가장 옳은 것은?

- ① 트랜잭션 내의 모든 연산은 완전히 수행되거나 전혀 수행되지 않은 상태로 유지되어야 한다.
- ② 트랜잭션이 수행되는 동안에는 다른 트랜잭션이 해당 데이터에 접근해서는 안 된다.
- ③ 트랜잭션이 성공적으로 완료된 후에는 시스템 장애가 발생하더라도 그 결과가 영구적으로 반영되어야 한다.
- ④ 트랜잭션 수행 전후 데이터베이스는 일관성 있는 상태를 유지해야 한다.

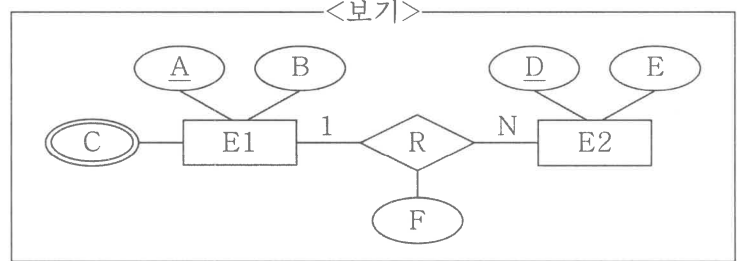
3. 데이터 제어어(DCL, data control language)의 특성으로 가장 옳지 않은 것은?

- ① 무결성 유지
- ② 데이터베이스 관리자(DBA)가 주로 사용
- ③ 접근 제어 및 권한 부여
- ④ 스키마 정의

4. 릴레이션 R의 차수(degree)가 3이고 카디널리티(cardinality)가 5이며, 릴레이션 S의 차수는 5이고 카디널리티가 7이다. 이 두 릴레이션을 카티션 프로덕트(cartesian product)했을 때 생성된 릴레이션의 차수와 카디널리티로 옳은 것은?

- ① 8, 12
- ② 8, 35
- ③ 15, 12
- ④ 15, 35

5. <보기>의 E-R 다이어그램(Entity-Relationship diagram)을 관계 데이터베이스 스키마로 변환했을 때, 가장 옳은 것은? (단, 밑줄 친 속성은 기본키(primary key)이다.)

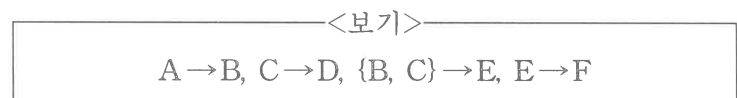


- ① E1(A, B, C)
E2(D, E, A, F)
- ② E1(A, B, C, D, F)
E2(D, E)
- ③ E1(A, B, D, F)
E2(D, E)
E3(A, C)
- ④ E1(A, B)
E2(D, E, A, F)
E3(A, C)

6. 관계 대수의 연산에 대한 설명으로 가장 옳지 않은 것은?

- ① 선택(select) 연산은 릴레이션에서 주어진 조건을 만족하는 튜플(tuple)을 추출한다.
- ② 프로젝트(project) 연산은 릴레이션에서 원하는 속성(attribute)만을 추출하며, 결과에서 중복 튜플은 제거된다.
- ③ 자연 조인(natural join)은 두 릴레이션에서 같은 이름을 가진 속성에 대해 동등 조인을 수행하며, 결과에서 조인 속성은 한 번만 나타난다.
- ④ 디비전(division) 연산 $R \div S$ 의 결과는 S에 있는 모든 값에 대응되지 않는 R의 튜플들을 반환한다.

7. 릴레이션 R(A, B, C, D, E, F)에서 함수적 종속성(functional dependency)이 <보기>와 같이 주어졌을 때, 릴레이션 R의 후보키(candidate key)로 가장 옳은 것은?



- ① {A, C}
- ② {A, D}
- ③ {A, B, C}
- ④ {A, C, E}

8. <보기>의 정규화 과정을 순서대로 바르게 나열한 것은?

<보기>

ㄱ. 후보키(candidate key)가 아닌 결정자 제거
 ㄴ. 부분 함수 종속성을 제거
 ㄷ. 속성을 원자값만 갖도록 분해
 ㄹ. 이행적 함수 종속성을 제거

- ① ㄱ → ㄴ → ㄷ → ㄹ
- ② ㄱ → ㄷ → ㄴ → ㄹ
- ③ ㄷ → ㄱ → ㄴ → ㄹ
- ④ ㄷ → ㄴ → ㄹ → ㄱ

9. 관계 데이터 모델에서 외래키(foreign key)에 대한 설명으로 가장 옳지 않은 것은?

- ① 하나의 릴레이션에는 여러 개의 외래키가 존재할 수 있다.
- ② 외래키가 되는 속성(attribute)과 그 외래키의 참조 대상이 되는 릴레이션 내 기본키(primary key) 속성의 도메인은 동일해야 한다.
- ③ 외래키는 다른 릴레이션을 참조하기 때문에 널(null) 값을 가질 수 없다.
- ④ 외래키는 자신이 속한 릴레이션의 기본키를 참조하는 것도 가능하다.

10. 한 번에 로크(lock) 되어야 할 데이터의 크기를 '로크 단위'라고 한다. 이 단위의 크기에 대한 설명으로 가장 옳지 않은 것은?

- ① 로크 단위가 작아지면 로크를 설정하는 오버헤드가 증가한다.
- ② 로크 단위가 커지면 트랜잭션(transaction)들의 동시 수행 가능성이 증가한다.
- ③ 릴레이션(테이블), 블록, 레코드 등이 로크 단위가 될 수 있다.
- ④ 로크 단위가 커지면 로크 단위가 작을 때보다 설정된 로크의 수가 적다.

11. <보기 1>의 테이블 R1과 R2에 대해 <보기 2>의 SQL문을 수행한 결과에서 튜플(tuple)의 개수는?

(단, 밑줄 친 속성은 기본키(primary key)이고 R1 테이블의 C는 R2 테이블을 참조하기 위한 외래키(foreign key)이다.)

<보기 1>

R1			R2		
<u>A</u>	B	C	<u>C</u>	D	E
A1	B1	1	1	고급	E1
A2	B2	2	2	특급	E2
A3	B3	3	3	중급	E2
A4	B4	1	4	초급	E1
A5	B5	2			
A6	B6	3			
A7	B7	2			

<보기 2>

```
SELECT A, C
FROM R1
WHERE C NOT IN (SELECT C
                FROM R2
                WHERE D = '고급' OR D = '중급');
```

- ① 2
- ② 3
- ③ 4
- ④ 6

12. 데이터베이스 질의 최적화에 대한 설명으로 가장 옳지 않은 것은?

- ① 주어진 질의를 처리할 수 있는 동등한 관계 대수들을 찾아, 가장 효율적인 질의 실행 계획을 선택하는 과정이다.
- ② 비용 기반 최적화는 다양한 실행 계획의 비용을 추정하여 가장 낮은 비용의 실행 계획을 선택하는 것이다.
- ③ 조인(join) 순서는 질의 처리 성능에 영향을 미치지 않으므로 어떤 순서로 조인을 수행해도 최적화에 무관하다.
- ④ 경험적(heuristic) 최적화의 예는 셀렉트(select) 연산을 가능한 이른 단계에 수행하여 중간 결과의 크기를 줄이는 것이다.

17. 전통적인 파일 처리 시스템(file-processing system)의 문제점을 극복하기 위해 도입된 데이터베이스 관리 시스템(DBMS, database management system)의 특징으로 가장 옳지 않은 것은?

- ① 응용 프로그램과 데이터 간의 상호 의존성을 높여 데이터 처리의 효율성을 극대화한다.
- ② 다수의 사용자가 동시에 동일한 데이터에 접근하고 갱신할 수 있도록 동시성 제어(concurrency control) 기능을 제공한다.
- ③ 데이터의 중복을 제어하여 저장 공간의 낭비를 막고 데이터의 일관성을 유지한다.
- ④ 물리적 데이터 구조가 변경되더라도 응용 프로그램에 영향을 주지 않는 물리적 데이터 독립성을 제공한다.

18. <보기 1>의 시험(EXAM) 테이블에는 6건의 시험성적 데이터가 저장되어 있다. <보기 2>의 SQL문 실행 결과에 포함되지 않는 것은?

<보기 1>

EXAM			
no	name	course	score
1	Kim	DataBases	20
2	Lee	DataBases	35
3	Kang	Algorithms	25
4	Jeon	Networks	30
5	Seo	Systems	25
6	Shin	Systems	20

<보기 2>

```

SELECT name
FROM EXAM AS E1
WHERE score >= (SELECT AVG(score)
                FROM EXAM AS E2
                WHERE E2.course = E1.course);
    
```

- ① Lee
- ② Seo
- ③ Shin
- ④ Jeon

19. <보기>는 즉시 갱신(immediate update) 기법을 사용하는 데이터베이스의 로그 파일 기록이다. 시스템 장애가 발생한 직후 회복 관리자가 수행해야 할 조치로 가장 옳은 것은? (단, 체크포인트 수행 시 메모리 버퍼의 갱신 내용은 모두 디스크에 안전하게 기록된다고 가정한다.)

<보기>

로그 기록
<T1, start>
<T1, A, 10, 20>
<T2, start>
<T2, B, 15, 30>
<T1, commit>
<checkpoint, active {T2}>
<T3, start>
<T3, C, 40, 50>
<T2, commit>
-- 시스템 장애 발생 --

- ① T1과 T2에 대해 Redo 연산을 수행하고, T3에 대해서는 아무 작업도 수행하지 않는다.
- ② T2에 대해 Redo 연산을 수행하고, T3에 대해서는 Undo 연산을 역순으로 수행한다.
- ③ T2와 T3 모두 장애 발생 시점에 완전하게 반영되지 않았으므로 Undo 연산을 역순으로 수행한다.
- ④ T1, T2, T3 모든 트랜잭션에 대해 순차적으로 Redo 연산을 수행하여 시스템을 복구한다.

20. 대규모 의사결정 지원을 위한 데이터 웨어하우스(data warehouse)의 다차원 모델링 기법 중, <보기>의 설명에 해당하는 스키마 구조로 가장 옳은 것은?

- <보기>
- 중앙에 매출, 수량 등의 트랜잭션 수치 데이터를 담고 있는 하나의 거대한 '사실(fact) 테이블'이 존재한다.
 - 사실 테이블을 둘러싼 '차원(dimension) 테이블'들이 존재하는데, 이 차원 테이블들이 제3정규형(3NF) 등으로 정규화되어 있고 사실 테이블을 중심으로 뻗어 나가는 다단계 계층 구조를 이룬다.
 - 차원 테이블 내의 데이터 중복을 최소화하여 저장 공간을 절약할 수 있으나, 사용자의 질의 수행 시 여러 테이블을 조인(join)해야 하므로 모델 구조가 복잡해지고 검색 성능이 저하될 수 있다.

- ① 스노우플레이크 스키마(snowflake schema)
- ② 스타 스키마(star schema)
- ③ 팩트 컨스텔레이션(fact constellation)
- ④ 데이터 큐브(data cube)