

데이터베이스론

1. 디스크에 저장된 사용자 데이터베이스나 시스템 카탈로그 접근에 대하여 책임지는 역할을 하는 데이터베이스 관리시스템의 구성 요소는?

- ① 질의 처리기
- ② 트랜잭션 관리자
- ③ 저장 데이터 관리자
- ④ 런타임 데이터베이스 처리기

2. 다음은 데이터베이스 설계 단계이다. (가) ~ (다)에 들어갈 내용을 바르게 연결한 것은?

요구 조건 분석 → (가) → (나) → (다) → 구현

(가)

(나)

(다)

- | | | |
|----------|--------|--------|
| ① 개념적 설계 | 논리적 설계 | 물리적 설계 |
| ② 논리적 설계 | 개념적 설계 | 물리적 설계 |
| ③ 물리적 설계 | 개념적 설계 | 논리적 설계 |
| ④ 물리적 설계 | 논리적 설계 | 개념적 설계 |

3. 직원(사번, 이름, 부서, 전화번호, 월급, 고용일) 테이블에 대한 유효한 표준 SQL문으로 옳지 않은 것은? (단, 밑줄 속성은 기본키이다)

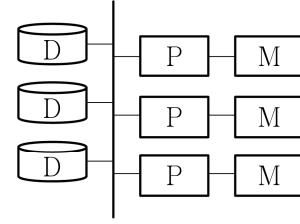
- ① SELECT 사번, 월급
FROM 직원
ORDER BY 월급;
- ② SELECT *
FROM 직원
WHERE 이름 LIKE '김%';
- ③ SELECT AVG(월급)
FROM 직원
WHERE 부서 = '인사팀';
- ④ SELECT SUM(부서, 월급)
FROM 직원
GROUP BY 부서;

4. 트랜잭션의 고립성 수준(isolation level) 중 반복할 수 없는 읽기(non-repeatable read) 문제가 있는 것만을 모두 고르면?

- ㄱ. read committed
ㄴ. read uncommitted
ㄷ. repeatable read
ㄹ. serializable

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

5. 병렬 데이터베이스에서 다음과 같이 구성된 병렬 장치를 위한 구조 모델은? (단, D는 디스크, M은 메모리, P는 프로세서이다)



- ① 비공유(shared nothing) 모델
- ② 공유 디스크(shared disk) 모델
- ③ 공유 메모리(shared memory) 모델
- ④ 공유 프로세서(shared processor) 모델

6. 반정형 데이터(semi-structured data)에 대한 설명으로 옳은 것만을 모두 고르면?

- ㄱ. 동영상, 이미지, 음성 파일이 해당한다.
ㄴ. 데이터 구조를 파악하기 위한 과정이 필요하다.
ㄷ. 데이터 내용 안에 구조 정보가 함께 저장된다.

- ① ㄱ, ㄴ
- ② ㄱ, ㄷ
- ③ ㄴ, ㄷ
- ④ ㄱ, ㄴ, ㄷ

7. 데이터베이스의 시스템 카탈로그에 저장되는 데이터로 옳지 않은 것은?

- ① 속성의 데이터 타입
- ② 테이블에 저장된 레코드 값
- ③ 데이터에 대한 다양한 제약 조건
- ④ 테이블 이름, 인덱스(index) 이름, 뷰(view) 이름

8. 빅데이터 분석 및 처리를 위한 기술인 하둡(Hadoop)에 대한 설명으로 옳은 것은?

- ① 맵리듀스(MapReduce) 기법을 사용하는 파일 시스템은 병렬 처리를 지원하지 않는다.
- ② 맵리듀스 기법에서 reduce() 함수의 입력 키 타입은 map() 함수의 출력 키 타입과 다르다.
- ③ 맵리듀스 기법의 map() 함수는 각 입력 레코드에 대해 출력이 없거나, 한 개 이상의 (키, 값) 쌍들을 출력한다.
- ④ 하둡 분산 파일 시스템(HDFS)에서 각 파일의 블록을 식별하기 위한 메타데이터를 저장하는 서버는 데이터노드(DataNode)이다.

9. 다음 트랜잭션 T₁, T₂, T₃ 스케줄의 충돌 직렬 가능성 검사 시 충돌하는 연산으로 옳지 않은 것은?

시간 ↓	T ₁	T ₂	T ₃
		read(A)	
	read(B)		
		write(A)	
		read(B)	
			read(A)
	write(B)		
			write(A)
		write(B)	

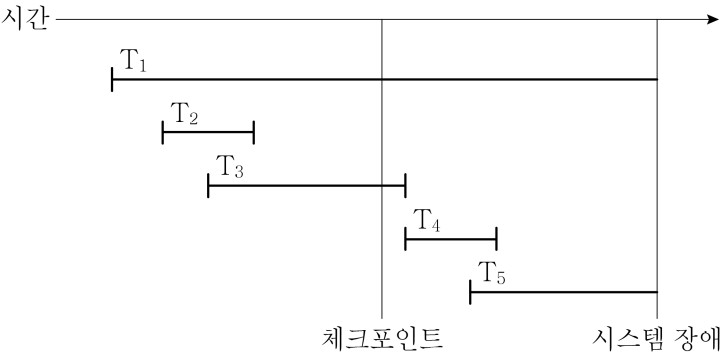
- ① T₁의 read(B)와 T₂의 read(B)
- ② T₁의 write(B)와 T₂의 write(B)
- ③ T₂의 read(B)와 T₁의 write(B)
- ④ T₂의 write(A)와 T₃의 read(A)

10. 관계형 데이터베이스의 질의 처리 방법에 대한 설명으로 옳은 것만을 모두 고르면?

- ㄱ. 질의 최적화 모듈은 질의 처리 연산을 실행할 때 소요되는 정확한 비용을 계산한다.
- ㄴ. 조인 연산 최적화를 위한 해시 조인 알고리즘은 동등 조인(equijoin)에 적용할 수 없다.
- ㄷ. 질의 처리는 관계 대수 표현식으로의 변환, 질의 최적화, 실행 계획 순으로 진행한다.
- ㄹ. 두 테이블 중 한 개 또는 두 개가 정렬되어 있는 경우에는 정렬 합병 조인 알고리즘이 유용하다.

- ① ㄱ, ㄴ
- ② ㄱ, ㄹ
- ③ ㄴ, ㄷ
- ④ ㄷ, ㄹ

11. 체크포인트 회복 기법을 사용하는 시스템에서 트랜잭션이 수행되는 중 다음과 같이 장애가 발생하였을 때, 취소(UNDO) 연산이 필요한 트랜잭션만을 모두 고르면?



- ① T₅
- ② T₁, T₅
- ③ T₄, T₅
- ④ T₁, T₃, T₅

12. ‘선수’ 테이블에서 검색 결과가 다음과 같이 생성되는 SQL문의 (가)에 들어갈 내용은?

선수				
선수번호	이름	포지션	점수	팀번호
2123	장영실	CF	30	2
3321	이순신	RB	12	1
2276	정약용	LW	6	1
1925	허준	RW	24	1
9842	김유신	GK	10	2

검색 결과		
팀번호	합계점수	최대점수
2	40	30

```
SELECT 팀번호, SUM(점수) AS 합계점수, MAX(점수) AS
      최대점수
FROM 선수
      (가)
```

- ① GROUP BY 팀번호;
- ② GROUP BY 선수번호;
- ③ GROUP BY 팀번호 HAVING AVG(점수) >= 15;
- ④ GROUP BY 팀번호 HAVING MAX(점수) >= 20;

13. 트랜잭션 T₁, T₂ 스케줄이 다음과 같을 때 발생하는 문제는?

시간 ↓	T ₁	T ₂
		sum = sum + 0
		read(A)
		sum = sum + A
	read(X)	
	X = X - 100	
	write(X)	
		read(X)
		sum = sum + X
		read(Y)
		sum = sum + Y
	read(Y)	
	Y = Y + 100	
	write(Y)	

- ① 오손 읽기(dirty read)
- ② 갱신 손실(lost update)
- ③ 팬텀 문제(phantom problem)
- ④ 부정확한 요약(incorrect summary)

14. 다음의 요구사항을 반영하여 도서 대출 관리용 관계 데이터 모델을 생성할 때, 이에 대한 설명으로 옳지 않은 것은?

- 회원은 회원번호, 이름, 생년월일, 주소를 가지며, 회원번호를 기본키로 한다.
- 도서는 도서번호, 제목, 저자, 출판연도를 가지며, 도서번호를 기본키로 한다.
- 대출은 대출번호, 회원번호, 도서번호, 대출일을 가지며, 대출번호를 기본키로 한다.
- 반납은 반납번호, 대출번호, 반납일을 가지며, 반납번호를 기본키로 한다.

- ① ‘회원’, ‘도서’, ‘대출’, ‘반납’이 주요 릴레이션이다.
- ② ‘도서’와 ‘반납’ 릴레이션 간의 관계는 일대다(1:n) 관계이다.
- ③ ‘대출’과 ‘반납’ 릴레이션 간의 관계는 다대다(m:n) 관계이다.
- ④ ‘대출’ 릴레이션의 회원번호는 ‘회원’ 릴레이션의 회원번호를 참조하는 외래키이다.

15. 표준 RAID(Redundant Array of Independent Disks) 1에 대한 설명으로 옳은 것은?

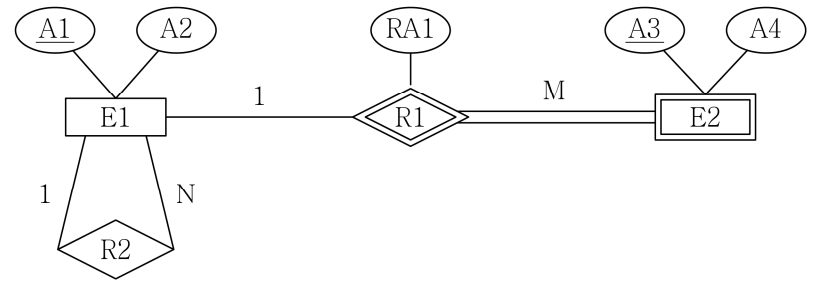
- ① 데이터를 디스크에 분산 저장하고 오류 검출 및 정정이 가능한 해밍코드를 사용한다.
- ② 미러링(mirroring) 기법을 활용하여 한 디스크 장애 시 미러 디스크에서 읽을 수 있다.
- ③ 데이터를 디스크에 교차 배치하여 읽기와 쓰기 성능이 향상되나, 장애 복구 기능은 제공되지 않는다.
- ④ 블록 단위로 데이터를 분할하여 여러 디스크에 분산 저장하고, 장애 시 패리티 비트를 통해 복구한다.

16. 분산 데이터베이스 시스템에 대한 설명으로 옳은 것만을 모두 고르면?

- ㄱ. 지역 트랜잭션과 전역 트랜잭션으로 구분할 수 있다.
- ㄴ. 분산 투명성에는 위치 투명성, 중복 투명성, 단편화 투명성이 포함된다.
- ㄷ. 사용자는 원하는 데이터가 어느 지역에 저장되어 있는지를 명확하게 알 필요가 있다.
- ㄹ. 한 릴레이션의 속성을 분할하여 두 개 이상의 서로 다른 장소에 저장하는 것은 수평적 단편화이다.

- ① ㄱ, ㄴ
- ② ㄱ, ㄹ
- ③ ㄴ, ㄷ
- ④ ㄷ, ㄹ

17. 다음 개체-관계 다이어그램(E-R diagram)을 관계형 데이터베이스 스키마로 변환하면? (단, 밑줄 속성은 기본키이고, RA2와 ‘E2’ 릴레이션의 A1은 ‘E1’ 릴레이션의 A1을 참조하는 외래키이다)



- ① E1(A1, A2, RA2)
E2(A3, A4, RA1)
- ② E1(A1, A2, RA1, RA2)
E2(A3, A4)
- ③ E1(A1, A2, RA1, RA2)
E2(A1, A3, A4)
- ④ E1(A1, A2, RA2)
E2(A1, A3, A4, RA1)

18. 다음에서 설명하는 테이블에서 모든 회원의 이름과 추천인의 이름을 함께 출력하기 위한 SQL문은? (단, 추천인이 없는 경우 추천인의 이름은 NULL 값으로 출력한다)

- ‘회원’ 테이블의 속성은 (아이디, 이름, 등급, 추천인아이디)이다.
- 밑줄 속성은 기본키이고 추천인아이디는 ‘회원’ 테이블의 아이디를 참조하는 외래키이다.
- 추천인아이디를 제외한 나머지 속성은 NOT NULL로 정의된다.
- 추천인이 없는 회원인 경우 추천인아이디 속성은 NULL로 저장된다.

- ① SELECT e.이름, s.이름 AS 추천인이름
FROM 회원 e LEFT OUTER JOIN 회원 s
ON e.추천인아이디 = s.아이디;
- ② SELECT e.이름, s.이름 AS 추천인이름
FROM 회원 e, 회원 s
WHERE e.추천인아이디 = s.아이디;
- ③ SELECT 이름, 추천인이름
FROM 회원 e
WHERE EXISTS
(SELECT 이름 AS 추천인이름
FROM 회원
WHERE e.아이디 = 추천인아이디);
- ④ SELECT e.이름, s.이름 AS 추천인이름
FROM 회원 e, 회원 s
WHERE e.추천인아이디 = s.아이디
AND e.추천인아이디 IS NOT NULL;

19. ‘R1’과 ‘R2’ 릴레이션이 다음과 같을 때 관계 대수 연산 적용에 대한 설명으로 옳지 않은 것은? (단, ‘R1’의 B와 ‘R2’의 B의 도메인은 동일하다)

R1		R2		
A	B	B	C	D
a1	b1	b1	c1	d1
a2	b2	b3	c2	d3
a3	b3	b4	c1	d5

- ① $R1 \cup R2$ 연산은 수행될 수 없다.
- ② $R2 \div R1$ 연산 결과의 차수는 2이고 카디널리티는 2이다.
- ③ $\sigma_{R1.B = 'b2'}(R1 \times \Pi_{B,D}(R2))$ 연산 결과의 카디널리티는 3이다.
- ④ $\sigma_{A = 'a1'}(R1 \times R2)$ 와 $\sigma_{A = 'a1'}(R1 \bowtie_{R1.B = R2.B} R2)$ 연산 결과의 차수는 동일하다.

20. 다음 <정보>를 이용하여 아래에 주어진 <관계 대수>를 세미조인(semijoin) 방식으로 처리하고자 할 때, 가장 먼저 수행되는 연산은? (단, 속성 이름의 괄호 안 숫자는 바이트 크기를 나타낸다)

- <정 보> —
- ‘EMP’와 ‘DEPT’ 테이블이 분산 저장되어 있다.
 - 사이트 1에 ‘EMP’ 테이블이 저장되며, SSN(10), NAME(30), DNO(5), ADDR(55)의 속성값으로 구성된 100 바이트 크기의 튜플 10,000 개를 포함한다.
 - 사이트 2에 ‘DEPT’ 테이블이 저장되며, DNUM(5), DNAME(25), MGRSSN(10), DATE(10)의 속성값으로 구성된 50 바이트 크기의 튜플 100 개를 포함한다.

— <관계 대수> —

$\Pi_{NAME,DNAME}(EMP \bowtie_{DNO = DNUM} DEPT)$

- ① $\Pi_{DNO}(EMP)$
 - ② $\Pi_{DNUM}(DEPT)$
 - ③ $\Pi_{DNO,NAME}(EMP)$
 - ④ $\Pi_{DNUM,DNAME}(DEPT)$
21. 인덱스에 대한 설명으로 옳은 것은?
- ① SQL문에서 조인 연산에 자주 사용되는 속성은 인덱스로 적합하다.
 - ② 클러스터링(clustering) 인덱스는 검색 효율을 위해 테이블당 2개 이상 활용된다.
 - ③ 밀집(dense) 인덱스는 데이터 파일의 레코드 그룹에 대해 생성되는 하나의 엔트리이다.
 - ④ 인덱스를 구성하기 위한 자료구조로는 이진 탐색 트리(binary search tree)가 가장 많이 활용된다.

22. 데이터베이스 관리시스템의 로그 레코드 버퍼링(log record buffering)에 대한 설명으로 옳지 않은 것은?

- ① 로그 레코드가 상당 기간 메인 메모리에 있으면 시스템 장애 시 해당 로그 레코드가 손실될 수 있다.
- ② 트랜잭션의 원자성을 보장하기 위한 회복 기법으로 로그 우선 기록 규약(write-ahead log protocol)을 지원해야 한다.
- ③ 시스템 장애 시 복구를 위해 변경 사항을 즉시 디스크에 기록하며, 이는 시스템의 성능을 최적화하는 데 중요한 역할을 한다.
- ④ 트랜잭션이 수행되는 동안 발생하는 변경 사항을 일시적으로 메모리 버퍼에 저장하여, 디스크 출력 횟수를 줄이고 시스템 성능을 향상시킨다.

23. 고객(고객번호, 고객이름, 나이, 등급, 직업, 적립금) 테이블에 대하여 다음과 같이 4개의 뷰를 생성하였을 때, 데이터 삽입이 가능한 뷰는? (단, 밑줄 속성은 기본키이며, 기본키 이외에 NOT NULL 속성은 없다)

```
CREATE VIEW 고객1(번호, 이름, 등급)
AS SELECT DISTINCT 고객번호, 고객이름, 등급
   FROM 고객
   WHERE 적립금 > 20000;

CREATE VIEW 고객2(고객이름, 등급, 적립금)
AS SELECT 고객이름, 등급, 적립금
   FROM 고객
   WHERE 직업 = '학생';

CREATE VIEW 고객3(등급, 평균나이)
AS SELECT 등급, AVG(나이) AS 평균나이
   FROM 고객
   GROUP BY 등급;

CREATE VIEW 고객4
AS SELECT 고객번호, 고객이름, 직업, 적립금
   FROM 고객
   WHERE 나이 >= 30 AND 등급 = 'GOLD';
```

- ① 고객1
- ② 고객2
- ③ 고객3
- ④ 고객4

24. 다음 ‘고객’ 테이블과 ‘주문’ 테이블에서 아래에 주어진 SQL문과 동일한 결과를 출력하는 SQL문은? (단, 밑줄 속성은 기본키이며, ‘주문’ 테이블의 고객번호는 ‘고객’ 테이블의 번호를 참조하는 외래키이다)

고객		
번호	이름	성별
1001	김홍도	M
1002	허균	M
1003	유관순	F

주문		
고객번호	제품번호	개수
1001	AA	1
1003	AA	1
1001	BA	2

SELECT 이름, 제품번호, 개수
FROM 고객, 주문
WHERE 번호 = 고객번호
ORDER BY 이름, 제품번호;

- ① SELECT 이름, 제품번호, 개수
FROM 고객 OUTER JOIN 주문
ON 번호 = 고객번호
ORDER BY 이름, 제품번호;
- ② SELECT 이름, 제품번호, 개수
FROM 고객 RIGHT OUTER JOIN 주문
ON 번호 = 고객번호
ORDER BY 이름, 제품번호;
- ③ SELECT 이름, 제품번호, 개수
FROM 고객 LEFT OUTER JOIN 주문
ON 번호 = 고객번호
ORDER BY 이름, 제품번호;
- ④ SELECT 이름, 제품번호, 개수
FROM 고객, 주문
WHERE 번호 IN (SELECT 고객번호 FROM 주문)
ORDER BY 이름, 제품번호;

25. 다음 ‘부품상태’ 테이블에서 도출할 수 있는 함수 종속을 화살표(→)로 표현할 때 옳은 것만을 표현한 것은? (단, 밑줄 속성은 기본키이며, 아래 튜플을 기준으로 판단한다)

부품상태				
상태번호	상태	재고위치	부품번호	수량
S1	20	서울	P1	300
S1	20	서울	P2	200
S1	20	서울	P3	400
S1	20	서울	P4	200
S1	20	서울	P5	100
S1	20	서울	P6	100
S2	10	부산	P1	300
S2	10	부산	P2	400
S3	10	부산	P2	200
S4	20	서울	P2	200
S4	20	서울	P4	300
S4	20	서울	P5	400

- ① {상태번호} → {재고위치}
{부품번호, 수량} → {상태}
- ② {부품번호} → {상태번호}
{상태번호, 부품번호} → {상태}
- ③ {부품번호} → {수량}
{상태번호, 상태} → {부품번호}
- ④ {상태번호} → {상태, 재고위치}
{상태번호, 부품번호} → {수량}