

1. DBMS 언어에 대한 설명으로 가장 옳지 않은 것은?

- ① 데이터 조작어(DML)는 고수준 데이터 조작어와 저수준 데이터 조작어로 구분할 수 있다.
- ② 저장구조 정의어(SDL)를 사용하여 내부 스키마를 표현(지정)할 수 있다.
- ③ SQL은 데이터 정의 기능, 데이터 조작 기능, 데이터 제어 기능을 모두 지원한다.
- ④ SQL은 한 번에 한 레코드(one-record-at-a-time) 조작어에 해당한다.

2. <보기>에서 설명하는 데이터베이스 시스템을 위한 아키텍처로 가장 옳은 것은?

<보기>

- 많은 웹 응용에서 활용
- 클라이언트-웹 서버-데이터베이스 서버 구조
- 웹 서버는 클라이언트의 요청을 데이터베이스 서버로 전달하기 전에 보안과 관련된 기능이나 기타 역할 수행

- ① 중앙집중식
- ② 2-층(2-tier) 클라이언트-서버
- ③ 3-층(3-tier) 클라이언트-서버
- ④ P2P(Peer to Peer)

3. RAID(Redundant Array of Independent Disks) 구조에 대한 설명으로 가장 옳지 않은 것은?

- ① 디스크에 대한 접근 성능과 신뢰성을 높일 수 있다.
- ② RAID 내의 각 디스크는 각각 독립적으로 동작할 수 있다.
- ③ 더 빠른 전송속도를 제공하기 위해 데이터 스트라이핑(data striping) 기법을 사용한다.
- ④ RAID 레벨 0은 장애 발생에 대비하여 별도의 디스크에 중복 쓰기를 수행한다.

4. <보기>와 같은 순서로 키(key)값을 입력하여, B-트리(B-tree)를 구성하고자 한다. (가) 차수(degree)가 3인 B-트리와 (나) 차수가 5인 B-트리의 루트 노드(root node)에 있는 키값을 옳게 짝지은 것은?

<보기>

키 입력순서: 5, 7, 3, 6, 1, 8, 2

- | | (가) | (나) | | (가) | (나) |
|---|-----|-----|---|-----|-----|
| ① | 3 | 5 | ② | 3 | 6 |
| ③ | 5 | 5 | ④ | 5 | 6 |

5. <보기>의 데이터베이스 설계 단계를 순서대로 가장 바르게 나열한 것은?

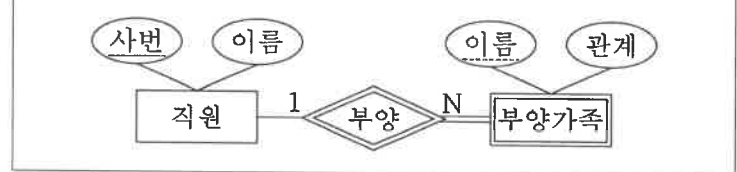
<보기>

- ㄱ. 핵심 정보 개체, 제한 조건 등 특정 데이터베이스 관리 시스템에 독립적이고 추상화된 개체관계 모델 생성
- ㄴ. 데이터베이스를 활용하게 되는 도메인, 운영 환경, 처리 조건 등을 분석하여 데이터베이스가 제공해야 하는 핵심적인 정보 대상 및 제한 조건 등 도출
- ㄷ. 관계형 데이터 모델과 같이 논리적 데이터베이스 구조에 맞는 테이블 스키마 생성
- ㄹ. 논리적 데이터베이스 구조로부터 효율적이고 구현 가능한 물리적 데이터베이스 구조 설계

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

6. <보기>의 개체-관계 다이어그램(E-R Diagram)에 대한 설명으로 가장 옳지 않은 것은?

<보기>



- ① 모든 부양가족은 자신을 부양하는 직원이 반드시 한 명 있어야 한다.
- ② 부양가족 개체의 이름 속성은 식별키에 해당한다.
- ③ 부양가족이 없는 직원도 있다.
- ④ 직원 개체는 자신과 관련된 부양가족 개체를 소유한다고 말할 수 있다.

7. <보기>에서 설명하는 K에 해당하는 키(key)의 명칭으로 가장 옳은 것은?

<보기>

K는 릴레이션 R의 속성들의 부분집합이다. R의 모든 속성들은 K에 대해 함수적 종속성(FD, Functional Dependency)을 가지나, K의 모든 부분집합 K' (단, K' ≠ K)에 대해서는 함수적 종속성을 갖지 않는다.

- ① 외래키(foreign key)
- ② 대체키(alternate key)
- ③ 후보키(candidate key)
- ④ 슈퍼키(super key)

8. <보기>의 릴레이션 R1, R2, R3에 대해서 주어진 관계 대수식(relational algebra) 쌍들 중, (가)와 (나)의 결과가 동일하지 않은 것은?

<보기>

R1=(A, B, C) R2=(D, E) R3=(F, G, H)

(가)

(나)

- ① $\Pi_{A,B,H}(R1 \bowtie_{A=F} R3)$ $\sigma_{A=F}(\Pi_{A,B}(R1) \times \Pi_{H}(R3))$
 ② $\Pi_{B,E}(R1 \bowtie_{A=D} R2)$ $\Pi_{B,E}(\sigma_{A=D}(R1 \times R2))$
 ③ $(\sigma_{B=5}(R1)) \bowtie_{A=D} R2$ $\sigma_{B=5}(R1 \bowtie_{A=D} R2)$
 ④ $(R1 \bowtie_{A=D} R2) \bowtie_{D=F} R3$ $(R2 \bowtie_{D=F} R3) \bowtie_{D=A} R1$

9. <보기>의 두 릴레이션 '부서'와 '프로젝트'에 대해서 조인(JOIN) 연산을 수행한 결과로 옳지 않은 것은?

<보기>

부서		프로젝트	
ID	부서명	ID	프로젝트명
1	인사	1	채용
2	재무	2	회계
3	총무	4	세무
		7	총무

- ① 내부 조인

ID	부서명	프로젝트명
1	인사	채용
2	재무	회계
3	총무	세무
7	NULL	총무

- ② 왼쪽 외부 조인

ID	부서명	프로젝트명
1	인사	채용
2	재무	회계
3	총무	NULL

- ③ 오른쪽 외부 조인

ID	부서명	프로젝트명
1	인사	채용
2	재무	회계
4	NULL	세무
7	NULL	총무

- ④ 완전 외부 조인

ID	부서명	프로젝트명
1	인사	채용
2	재무	회계
3	총무	NULL
4	NULL	세무
7	NULL	총무

10. 릴레이션 처리에서 데이터 중복으로 인해 발생하는 이상(anomaly) 현상으로 가장 옳지 않은 것은?

- ① 삭제 이상(deletion anomaly)
 ② 검색 이상(search anomaly)
 ③ 삽입 이상(insertion anomaly)
 ④ 갱신 이상(update anomaly)

11. <보기 1>의 테이블 스키마는 저자, 책, 저자-책 관계 정보를 나타내고, 밑줄 친 속성은 기본키이다. <보기 2>는 동일한 책을 공동으로 저술한 모든 저자들의 이름과 그 책의 book_id를 함께 출력하기 위한 질의문이다. <보기 2> 질의문의 빈칸에 들어갈 것을 가장 옳게 짝지은 것은? (단, 동일한 책에 대한 공동저자들의 이름은 순서에 관계없이 한 번씩만 출력되어야 하고, 3인 이상 공동 저술한 책은 없다.)

<보기 1>

테이블 스키마	설명
author(author_id INT, name VARCHAR(10))	저자 정보 테이블
book(book_id INT, title VARCHAR(50))	책 정보 테이블
writes(author_id INT, book_id INT)	저자와 책 간의 관계 정보 테이블

<보기 2>

```
SELECT a1.name, a2.name, w1.book_id
FROM   writes w1, writes w2, author a1, author a2
WHERE  w1.book_id (가) and w1.author_id (나)
       and w1.author_id = a1.author_id and (다) a2.author_id;
```

(가)

(나)

(다)

- ① $\neq w2.book_id$ $> w2.author_id$ $w2.author_id =$
 ② $= w2.book_id$ $\neq w2.author_id$ $w2.author_id =$
 ③ $= w2.book_id$ $\neq w2.author_id$ $a1.author_id =$
 ④ $= w2.book_id$ $< w2.author_id$ $w2.author_id =$

12. DBMS는 사용자가 제시한 SQL 질의를 동등한 관계 대수식으로 번역한 후 다양한 방식으로 최적화를 수행한다. SQL 문장 혹은 이를 번역한 관계대수식에 대한 최적화와 관련된 설명으로 가장 옳지 않은 것은?

- ① 주기억장치보다 더 큰 데이터에 대해 order by 구문은 외부 정렬 알고리즘에 의해 정렬(sort)-합병(merge) 전략으로 수행시킨다.
 ② where SSN='1234' 구문에서, SSN이 기본키이고 그 기본키에 기본 인덱스가 생성되어 있다면 이진 탐색(Binary Search) 방식으로 동작한다.
 ③ where 절에 논리합 조건이 있는 경우는 논리곱 조건보다 실행하거나 최적화하기가 더 어려운 경우가 많다.
 ④ $R \bowtie_{A=B} S$ 와 같은 동등 조인 연산은 중첩 루프 조인(nested loop join) 방식으로 처리 가능하다.

13. <보기>는 부서와 그 부서에서 수행하는 프로젝트에 대한 데이터를 저장하는 테이블 스키마다. 이 스키마에서 (가)출력하고자 하는 내용과 (나)이를 출력하기 위한 SQL문을 짰지은 것으로 가장 옳지 않은 것은? (단, 밑줄 친 속성이 기본키며, 목표 외의 다른 내용은 출력되지 않아야 한다.)

<보기>

부서(ID, 부서이름, 부서위치, 설립일자, 부서장)
 프로젝트(ID, 프로젝트명, 담당부서ID, 예산, 프로젝트타입, 프로젝트시작일)

①	(가) 예산이 10,000 이상인 프로젝트의 이름과 예산 출력 (나) SELECT 프로젝트명, 예산 FROM 프로젝트 WHERE 예산 >= 10000;
②	(가) 부서별로 수행 중인 모든 소관 프로젝트의 평균 예산을 부서이름과 함께 출력하되, 프로젝트가 없는 부서는 제외 (나) SELECT 부서.부서이름, AVG(프로젝트.예산) AS avg_budget FROM 부서 JOIN 프로젝트 ON 부서.ID = 프로젝트.담당부서ID GROUP BY 부서.부서이름;
③	(가) 모든 부서의 이름과 해당 부서에서 수행 중인 프로젝트의 수를 출력하되, 프로젝트가 없는 부서도 포함 (나) SELECT 부서.부서이름, COUNT(프로젝트.ID) AS project_count FROM 부서 LEFT JOIN 프로젝트 ON 부서.ID = 프로젝트.담당부서ID GROUP BY 부서.부서이름;
④	(가) 특정 부서(ID=2)에서 수행 중인 프로젝트 중 예산이 가장 높은 프로젝트의 이름을 출력 (나) SELECT 프로젝트명 FROM 프로젝트 WHERE 예산 = (SELECT MAX(예산) FROM 프로젝트 WHERE 담당부서ID = 2);

14. 트랜잭션의 동시성 제어(concurrency control)를 위한 로킹(locking)에 대한 설명으로 가장 옳지 않은 것은? (단, lock은 배타로크이고 shared-lock은 공유로크이다.)
- 트랜잭션 T는 다른 트랜잭션에 의해 이미 lock이 걸려 있는 데이터 아이템 x에 대해 다시 lock(x)를 실행시키지 못한다.
 - 트랜잭션 T가 데이터 아이템 x에 대해 실행한 lock(x)에 대해서는 T가 모든 실행을 종료하기 전에 반드시 unlock(x)를 실행해야 한다.
 - 트랜잭션 T가 데이터 아이템 x에 대해 read(x)나 write(x)를 하려면 반드시 먼저 lock(x)를 실행해야 한다.
 - 트랜잭션 T가 데이터 아이템 x에 대해 shared-lock을 걸고 난 후 T는 x에 대해 판독과 기록을 할 수 있다.

15. 낙관적 동시성 제어(optimistic concurrency control)에 대한 설명으로 가장 옳지 않은 것은?

- 기록 단계(write phase) - 판독 단계(read phase) - 확인 단계(validation phase)를 거쳐 처리된다.
- 트랜잭션을 실행하는 동안 모든 갱신은 트랜잭션을 위해 유지되는 작업 구역의 지역 사본에만 반영된다.
- 지역 사본에 반영된 트랜잭션의 실행 결과를 데이터 베이스에 그대로 반영할 때 직렬 가능성 위반 여부를 확인한다.
- 아무런 검사를 시행하지 않고 트랜잭션 실행을 시작하는 것으로 트랜잭션이 수행하는 갱신은 다 종료될 때까지 데이터베이스에 직접 반영되지 않는다.

16. <보기>는 트랜잭션 처리 과정에서 남은 로그 기록이며, 마지막 줄까지 로그를 기록한 후 시스템 장애가 발생하였다. <보기>의 로그 기록을 사용하여 복구(recovery) 과정을 거친 후의 A, B, C, D 값을 옳게 짰지은 것은?

<보기>

로그 기록			
<T1, start>			
<T2, start>			
<T1, B, 6, 12>			
<T3, start>			
<T1, commit>			
<T3, C, 21, 8>			
<checkpoint, active (T2, T3)>			
<T4, start>			
<T3, commit>			
<T4, B, 12, 17>			
<T5, start>			
<T2, A, 45, 16>			
<T4, abort>			
<T5, D, 16, 21>			
<T5, commit>			
-- 시스템 장애 발생 --			

	A	B	C	D
①	16	12	8	21
②	16	12	21	21
③	45	12	8	21
④	45	12	21	21

17. <보기>는 SQL에 정의된 고립도(Isolation Level)에 따라 발생 가능한 위반 유형을 나타내고 있다. (가)~(라)에 들어갈 것을 옳게 짝지은 것은? (단, Yes는 위반이 가능함을 나타내고, No는 위반이 가능하지 않음을 나타낸다.)

<보기>

고립도 (Isolation Level)	위반 유형		
	오손 읽기 (Dirty Read)	반복할 수 없는 읽기 (Nonrepeatable Read)	팬텀 (Phantom)
(가)	Yes	Yes	Yes
(나)	No	Yes	Yes
(다)	No	No	Yes
(라)	No	No	No

- ① (가) READ UNCOMMITTED (나) READ COMMITTED
 (다) REPEATABLE READ (라) SERIALIZABLE
- ② (가) READ UNCOMMITTED (나) NO DIRTY READ
 (다) REPEATABLE READ (라) SERIALIZABLE
- ③ (가) READ UNCOMMITTED (나) NO DIRTY READ
 (다) REPEATABLE READ (라) NO PHANTOM READ
- ④ (가) DIRTY READ (나) NO DIRTY READ
 (다) REPEATABLE READ (라) NO PHANTOM READ

18. 분산 데이터베이스의 특징에 대한 설명으로 가장 옳지 않은 것은?

- ① 분산 데이터베이스는 컴퓨터 네트워크를 통해 분산된, 논리적으로 상호연관된 다수의 데이터베이스의 모임이다.
- ② 수평 단편화란 한 릴레이션의 튜플들의 부분집합을 여러 사이트에 할당하는 방식이다.
- ③ 여러 사이트에 데이터가 완전히 중복된 분산 데이터베이스를 구축할 경우, 사용자 검색 질의나 갱신 요구를 효율적으로 처리할 수 있다.
- ④ 2단계 완료 프로토콜(2PC)은 봉쇄형 프로토콜이라는 단점이 있어서, 3단계 완료 프로토콜(3PC)을 사용하기도 한다.

19. 분산 데이터베이스에서 <보기 1>의 테이블 스키마에 대해 <보기 2>의 질의를 처리한 결과(RESULT)를 사이트(site) 3으로 전송하고자 한다. 데이터 전송량을 줄이기 위해 세미조인을 이용하여 분산 질의 처리를 수행한다고 할 때, 이에 대한 설명으로 가장 옳지 않은 것은?

<보기 1>

테이블 스키마	설명
EMP (ID, Ename, Dno)	<ul style="list-style-type: none"> 직원 정보 테이블로 사이트 1에 존재함 직원 1,000명, ID와 Dno 속성의 크기는 각 4바이트, Ename 속성의 크기는 10바이트임
DEPT (Dno, Dname, MgrID)	<ul style="list-style-type: none"> 부서 정보 테이블로 사이트 2에 존재함 MgrID는 EMP 테이블의 직원번호 속성인 ID를 참조하는 외래키로 부서 관리자의 ID를 나타내며, 중복된 값이나 NULL 값을 가질 수 없음 부서 10개, Dno와 MgrID 속성의 크기는 각 4바이트, Dname 속성의 크기는 10바이트임

<보기 2>

RESULT ← $\pi_{MgrID, Ename, Dname}(EMP \bowtie_{ID=MgrID} DEPT)$

- ① 사이트 1에서 사이트 2로 $\pi_{ID}(EMP)$ 을 전송해야 한다.
- ② 사이트 2에서 사이트 1로 $\pi_{MgrID}(DEPT)$ 을 전송해야 한다.
- ③ 사이트 2에서 사이트 3으로 240바이트를 전송해야 한다.
- ④ 사이트 1에서 사이트 2로 140바이트를 전송해야 한다.

20. <보기>에서 설명하는 데이터 마이닝(data mining) 분석 기법으로 가장 옳은 것은?

<보기>

- 비지도학습(unsupervised learning)의 일종이다.
- 유사한 구매 패턴(pattern)을 가진 고객들의 집단을 알아내는 데 사용할 수 있다.

- ① 연관 분석 ② 군집 분석
 ③ 분류 분석 ④ 회귀 분석