데이터베이스론

문 1. DBMS를 사용하는 것이 파일 시스템(file system)을 사용하는 것보다 더 적합한 경우는?

① 데이터와 응용이 단순하고 변경이 거의 일어나지 않는 경우
② 예약 시스템과 같이 최신 정보를 다수의 사용자가 공유해야 하는 경우
③ 응용프로그램의 실시간 요구사항이 엄격한 경우
④ 내장형 시스템과 같이 저장 용량이 제한된 경우

문 2. 동시성 제어(concurrency control) 방법에서 로킹(locking) 단위가 커지는 경우에 대한 설명으로 옳은 것은?

① 로킹 오버헤드 감소, 동시성 정도 증가
② 로킹 오버헤드 감소, 동시성 정도 감소
③ 로킹 오버헤드 증가, 동시성 정도 증가
④ 로킹 오버헤드 증가, 동시성 정도 감소

문 3. 뷰(view)에 대한 설명으로 옳지 않은 것은?

① 뷰는 기본 테이블(base table)에 대한 질의로 정의되는 가상 테이블(virtual table)로 질의 처리 성능을 향상시킬 수 있다.
② 뷰를 통해 기본 테이블에 대한 사용자의 접근을 제한함으로써 보안성을 높일 수 있다.
③ WITH CHECK OPTION을 사용하여 뷰를 정의하면, 뷰를 통해 삽입 또는 갱신되는 투플(tuple)에 대해 제한을 둘 수 있다.
④ 집계 함수의 결과를 애트리뷰트(attribute)로 사용하는 뷰에 투플의 삽입이나 갱신이 불가능하다.

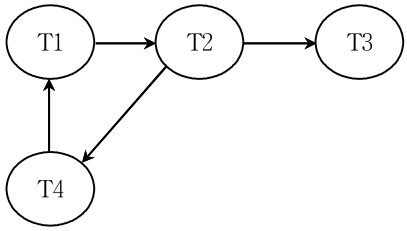
문 4. 다음과 같이 동시성 제어 없이 두 트랜잭션 T1과 T2가 수행되는 경우 X의 최종값은? (단, X의 초기값은 100이다)

T1	T2
Read_item(X)	
X = X + 30	
Write_item(X)	
	Read_item(X)
Rollback	
	X = X - 50
	Write_item(X)

시간 ↓

① 30
② 50
③ 80
④ 100

문 5. 어떤 DBMS에서 실행 중인 트랜잭션의 대기 그래프(wait-for graph)가 다음과 같을 때, 이 상태로부터의 회복에 대한 설명으로 옳지 않은 것은?



① 대기-롤백(wait-die) 또는 롤백-대기(wound-wait) 기법을 사용하여 회복한다.
② 회복 비용에 따라 희생자(victim)를 선택하고 롤백한다.
③ 회복 과정 중 전체 롤백 또는 부분 롤백을 선택할 수 있다.
④ 회복 과정 중 기아현상(starvation)이 발생할 수 있다.

문 6. 다음은 파일 내의 레코드들에 대한 인덱스 생성 방법을 설명하고 있다. 괄호 안에 들어갈 말로 옳은 것은?

○ (㉠)는 인덱스의 엔트리 순서가 레코드의 물리적 순서와 동일하게 유지되는 인덱스이다.
○ (㉡)는 탐색키 값에 따라 정렬되지 않은 데이터 파일에 대하여 정의되는 인덱스이다.
○ (㉢)는 각 레코드마다 하나의 인덱스 엔트리를 갖도록 만드는 인덱스이다.

	㉠	㉡	㉢
①	기본 인덱스	희소 인덱스	클러스터링 인덱스
②	보조 인덱스	밀집 인덱스	희소 인덱스
③	희소 인덱스	클러스터링 인덱스	기본 인덱스
④	클러스터링 인덱스	보조 인덱스	밀집 인덱스

문 7. 다음 두 릴레이션 R(A, B, C)와 S(A, D, E)가 있을 때, SQL 문을 수행한 후 생성되는 투플(tuple)의 개수는?

R

A	B	C
1	a	10
1	a	11
1	a	25
2	b	22
3	b	21
5	c	17

S

A	D	E
1	p	x
1	p	y
2	q	y
4	r	w
6	s	z

```
(SELECT DISTINCT A FROM R) UNION ALL (SELECT A FROM S)
```

① 6
② 7
③ 8
④ 9

문 8. BCNF(Boyce-Codd Normal Form)를 만족하기 위한 조건만을 모두 고른 것은?

ㄱ. 모든 결정자(determinant)가 후보키(candidate key)여야 한다.
ㄴ. 후보키에 속하지 않는 모든 애트리뷰트가 기본키에 이행 함수 종속(transitive functional dependency)되어 있지 않다.
ㄷ. 릴레이션의 모든 애트리뷰트가 원자값을 갖는다.
ㄹ. 후보키에 속하지 않는 모든 애트리뷰트가 기본키에 부분 함수 종속(partial functional dependency)되어 있지 않다.

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

문 9. 사원 데이터베이스에서 '부양가족이 없는 사원들의 성과 이름을 검색하라'라는 질의에 대한 관계대수식은? (단, ⋈는 자연조인이고, 조인 애트리뷰트는 주민번호이다)

사원	주민번호(PK)	성	이름	주소	성별	연봉	부서번호(FK)
부서	부서번호(PK)	부서명	관리자주민번호(FK)	관리시작일			
부양가족	주민번호(FK)	가족이름	관계	생년월일			

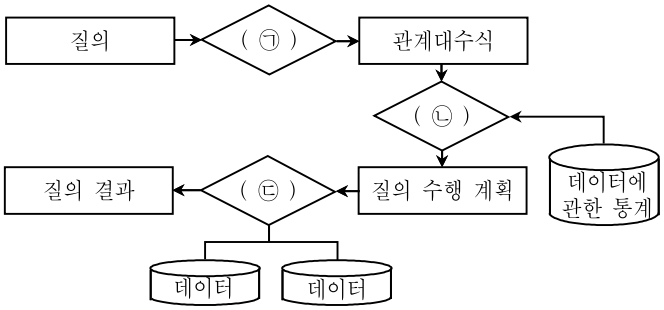
① $\Pi_{성,\ 이름}((\Pi_{주민번호}(사원) - \Pi_{주민번호}(부양가족)) \bowtie 사원)$
② $\Pi_{성,\ 이름}((\Pi_{주민번호}(사원) \bowtie \Pi_{주민번호}(부양가족)) \times 사원)$
③ $\Pi_{성,\ 이름}((\Pi_{주민번호}(사원) \times \Pi_{주민번호}(부양가족)) \bowtie 사원)$
④ $\Pi_{성,\ 이름}((\Pi_{주민번호}(사원) \bowtie \Pi_{주민번호}(부양가족)) - 사원)$

문 10. 다음은 즉시 갱신 기법을 사용하는 DBMS의 로그(log)이다. 시스템 고장 후, 회복(recovery)에 대한 설명으로 옳지 않은 것은?

<T1, start>
<T1, A, 100, 200>
<T1, C, 200, 300>
<T2, start>
<T1, commit>
<checkpoint>
<T2, B, 100, 10>
<T3, start>
<T3, D, 1000, 500>
<T4, start>
<T3, commit>
<T4, A, 200, 500>
<T2, commit>
---------- 시스템 고장

① T1은 회복 작업을 수행할 필요가 없다.
② T2는 트랜잭션 전체 연산에 대해 undo한다.
③ T3는 트랜잭션 전체 연산에 대해 redo한다.
④ T4는 트랜잭션 전체 연산에 대해 undo한다.

문 11. 일반적으로 DBMS에서 질의를 처리하는 단계는 그림과 같다. 괄호 안에 들어갈 기능으로 옳은 것은?



	㉠	㉡	㉢
①	최적기(optimizer)	질의 수행 엔진	파서와 변환기
②	최적기(optimizer)	파서와 변환기	질의 수행 엔진
③	파서와 변환기	질의 수행 엔진	최적기(optimizer)
④	파서와 변환기	최적기(optimizer)	질의 수행 엔진

문 12. 데이터 웨어하우스(data warehouse)의 특징으로 옳지 않은 것은?

① 여러 데이터 소스(근원지)로부터 수집된 정보를 하나의 통일된 스키마에 저장한다.
② 저장된 데이터의 추가, 삭제, 갱신 작업이 자주 발생한다.
③ 의사 결정에 필요한 주제와 관련된 데이터를 유지한다.
④ 과거와 현재의 데이터를 동시에 유지하여 데이터 간의 시간적 관계나 동향을 분석해 의사 결정에 반영할 수 있도록 한다.

문 13. DBMS 아키텍처에 대한 설명으로 옳지 않은 것은?

① 3-층(tier) 아키텍처는 데이터베이스 서버에 비즈니스 규칙들을 저장한다.
② 3-층 아키텍처는 많은 웹 응용에 적합한 구조이다.
③ 2-층 아키텍처는 클라이언트 프로그램이 서버 측의 DBMS와 통신할 수 있도록 표준 API를 제공한다.
④ 2-층 아키텍처는 질의처리와 트랜잭션 기능을 모두 서버에서 수행한다.

문 14. 해싱(hashing) 기법에서 버킷 오버플로우(bucket overflow)가 발생하는 주된 이유가 아닌 것은?

① 해시 함수가 랜덤한 해시키(hash key)를 생성할 경우
② 대부분의 레코드가 동일한 탐색키를 가질 경우
③ 해시 함수가 탐색키를 균등하게 분배하지 않을 경우
④ 버킷에 저장 가능한 최대 레코드 개수보다 저장할 레코드 개수가 많을 경우

문 15. 릴레이션 스키마(relation schema)와 무결성 제약조건에 대한 설명으로 옳은 것만을 모두 고른 것은?

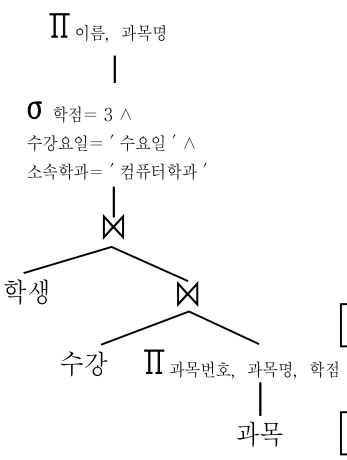
ㄱ. 스키마에는 무결성 제약조건이 포함된다.
ㄴ. 스키마는 데이터베이스 상태(state)와 마찬가지로 변경될 수 있다.
ㄷ. 참조 무결성 제약조건(referential integrity constraint)은 두 릴레이션의 연관된 투플(tuple)들 사이의 무결성 유지와 관련이 있다.
ㄹ. 한 릴레이션에 외래키(foreign key)가 여러 개 존재할 수 있다.
ㅁ. 외래키도 기본키(primary key)의 구성요소가 될 수 있다.

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

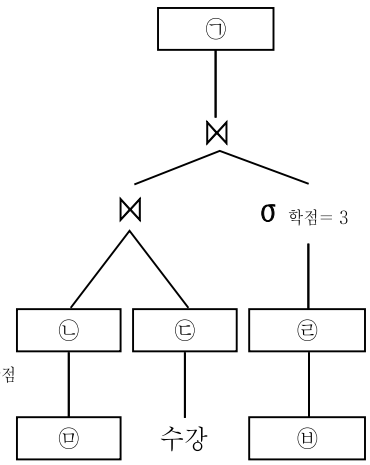
문 16. 주어진 학교 데이터베이스 스키마에서 <질의 트리>를 <변환 트리>와 같이 최적화할 때, '$\Pi_{\text{과목번호, 과목명, 학점}}$'과 '$\sigma_{\text{소속학과='컴퓨터학과'}}$'가 들어갈 위치로 바르게 나열한 것은?

학생	학번(PK)	이름	소속학과	
수강	학번(PK, FK)	과목번호(PK, FK)	수강요일	
과목	과목번호(PK)	과목명	개설학과	학점

<질의 트리>

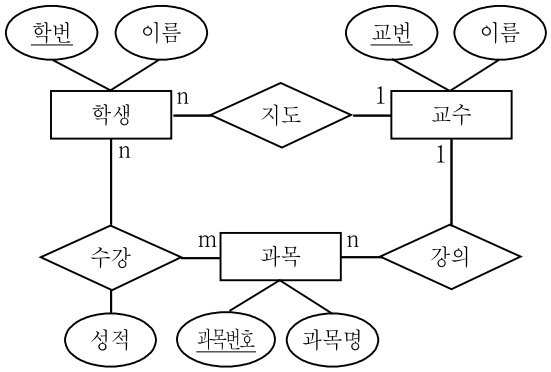


<변환 트리>



	$\Pi_{\text{과목번호, 과목명, 학점}}$	$\sigma_{\text{소속학과='컴퓨터학과'}}$
①	㉥	㉤
②	㉣	㉠
③	㉣	㉡
④	㉡	㉢

문 17. 개체-관계(ER) 다이어그램을 관계형 데이터베이스 스키마로 바르게 변환한 것은?



① 학생 | 학번(PK) | 이름 | 교번(FK) |

② 교수 | 교번(PK) | 이름 | 과목번호(FK) |

③ 과목 | 과목번호(PK) | 과목명 | 학번(FK) |

④ 수강 | 학번(PK, FK) | 과목번호(PK, FK) |

문 18. 다음 릴레이션 R과 함수 종속성(FD)에 대하여 R의 후보키(candidate key)가 될 수 없는 것은?

R(A, B, C, D, E)
FD : AB → C, CD → E, C → A, C → D, D → B

① AB
② AD
③ BD
④ C

문 19. 로그 버퍼에 대한 설명으로 옳지 않은 것은?

① 로그 파일은 안정 저장장치(stable storage)에서 운영되며 로그 버퍼는 주기억장치에서 운영된다. 따라서 시스템 고장 발생 시 로그 버퍼의 내용을 잃을 수 있다.
② 로그 레코드 <Ti, commit>가 로그 파일에 기록되기 전에 로그 버퍼내의 Ti와 관련된 모든 로그 레코드들은 로그 파일에 기록되어야 한다.
③ 데이터베이스 버퍼에 있는 블록을 데이터베이스 파일에 기록하는 것과 로그 버퍼에 있는 블록을 로그 파일에 기록하는 것은 순서적으로 독립적이다.
④ 로그 버퍼에 기록된 로그 레코드들의 순서와 로그 파일에서의 이들의 순서는 동일하여야 한다.

문 20. 회사 데이터베이스에서 직원이 6명 이상인 부서의 부서명과 그 부서 소속 직원 중 급여가 40,000 이상인 직원의 수를 검색하는 SQL 질의로 옳은 것은? (단, 모든 부서에서 급여가 40,000 이상인 직원이 1명 이상 있다고 가정한다)

직원	주민번호(PK)	이름	주소	성별	급여	소속부서번호(FK)
부서	부서번호(PK)	부서명	부서장주민번호(FK)	주소		

①
```
SELECT B.부서명, COUNT(*)
FROM 직원 as A, 부서 as B
WHERE B.부서번호 = A.소속부서번호 AND A.급여 >= 40000
GROUP BY B.부서번호
HAVING COUNT(*) > 5
```

②
```
SELECT B.부서명, COUNT(*)
FROM 직원 as A, 부서 as B
WHERE B.부서번호 = A.소속부서번호 AND A.급여 >= 40000 AND
        (SELECT COUNT(*)
        FROM 직원 as C
        GROUP BY C.소속부서번호 > 5)
GROUP BY B.부서번호
HAVING COUNT(*) > 5
```

③
```
SELECT A.부서명, COUNT(*)
FROM 직원 as A
WHERE A.급여 >= 40000 AND
        A.소속부서번호 IN (SELECT B.소속부서번호
                           FROM 직원 as B
                           GROUP BY B.소속부서번호
                           HAVING COUNT(*) > 5)
GROUP BY A.부서번호
```

④
```
SELECT B.부서명, COUNT(*)
FROM 직원 as A, 부서 as B
WHERE B.부서번호 = A.소속부서번호 AND A.급여 >= 40000 AND
        A.소속부서번호 IN (SELECT C.소속부서번호
                           FROM 직원 as C
                           GROUP BY C.소속부서번호
                           HAVING COUNT(*) > 5 )
GROUP BY B.부서번호
```