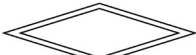
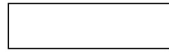
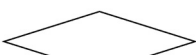



1. 데이터베이스 설계에서 물리적 설계에 해당하지 않는 것은?

- ① 접근 경로(access path) 설계
- ② 저장 레코드(stored record)의 양식(format) 설계
- ③ 스키마의 평가와 정제(evaluation & refinement)
- ④ 레코드 집중(clustering)의 분석 및 설계

2. ER 다이어그램에서 주민등록번호와 같은 기본키가 존재하지 않는 피부양자와 부양자의 관계를 표현하는 관계 기호에 해당하는 것은?

- ① 
- ② 
- ③ 
- ④ 

3. <보기>의 관계 데이터베이스 설계의 함수적 종속성과 정규형에 대한 설명 중 괄호에 들어갈 용어는?

<보기>

릴레이션 R이 BCNF에 속하고 모든 ()이 함수 종속(functionally dependent)이면 릴레이션 R은 4NF에 속한다.

- ① 다치 종속성(multivalued dependency)
- ② 이행 종속성(transitive dependency)
- ③ 부분 종속성(partial dependency)
- ④ 조인 종속성(join dependency)

4. 2단계 로킹(two-phase locking)에 대한 설명으로 가장 옳지 않은 것은?

- ① 모든 트랜잭션이 이를 준수하면 직렬 가능(conflict serializable)하다.
- ② 모든 트랜잭션이 이를 준수하면 교착 상태(deadlock)에 빠지지 않는다.
- ③ 확장(growing) 단계에서는 lock만 수행할 수 있고 unlock은 수행할 수 없다.
- ④ 축소(shrinking) 단계에서는 unlock만 수행할 수 있고 lock은 수행할 수 없다.

5. 릴레이션 R(A, B, C, D)에 대한 함수적 종속성이 <보기>와 같이 주어졌을 때 슈퍼키(super key)가 아닌 것은?

- <보기>
- | | |
|---------|-------|
| ABC → D | D → A |
|---------|-------|
- ① ABC
 - ② BCD
 - ③ D
 - ④ ABCD

6. 일반적인 DBMS에서 문자열을 저장하기에 가장 적합한 속성의 데이터 타입은?

- ① SMALLINT
- ② VARCHAR(n)
- ③ REAL
- ④ TIME

7. <보기 1>의 내용에 따라 <보기 2>의 로크 호환성 테이블을 옳게 채운 것은?

<보기 1>

공유 로크(shared lock)와 배타 로크(exclusive lock)만 가지는 표준 로킹 기법에서, 공유 로크와 배타 로크의 관계는 로크 호환성 테이블로 나타낼 수 있다. <보기 2>는 어떤 트랜잭션 T가 항목 X에 대해 열 제목에 명시된 로크를 보유하고 있고, 다른 트랜잭션 T'이 같은 항목 X에 대해 행 제목에 명시된 로크를 요청할 때, T'이 로크를 획득할 수 있는지 여부를 나타낸다.

<보기 2>

		T 보유	
		공유 로크	배타 로크
T'	공유 로크	㉠	㉡
	배타 로크	㉢	㉣

- | | ㉠ | ㉡ | ㉢ | ㉣ |
|---|-----|-----|-----|-----|
| ① | Yes | Yes | Yes | Yes |
| ② | Yes | Yes | Yes | No |
| ③ | Yes | Yes | No | No |
| ④ | Yes | No | No | No |

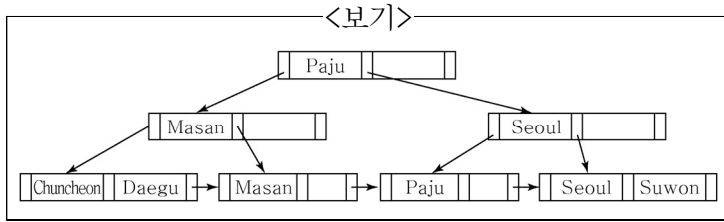
8. <보기>의 분산 데이터베이스의 단편화 투명성(fragmentation transparency)에 대한 설명에서, (가)와 (나) 각각에 들어갈 용어로 가장 옳은 것은?

<보기>

- (가)는 한 릴레이션을 서브 릴레이션으로 분산하며, 서브 릴레이션은 원래 릴레이션의 튜플(행)들의 부분집합이다.
- (나)는 한 릴레이션을 애트리뷰트들의 부분집합으로 이루어진 두 개 이상의 릴레이션들로 나눈다.

- | | (가) | (나) |
|---|-----------|-----------|
| ① | 수평 단편화 | 수직 단편화 |
| ② | 수직 단편화 | 수평 단편화 |
| ③ | 튜플 단편화 | 애트리뷰트 단편화 |
| ④ | 애트리뷰트 단편화 | 튜플 단편화 |

9. <보기>의 트리 구조에 대한 설명으로 가장 옳은 것은?



- ① B⁺-트리로서, 삭제 과정에서 재분배가 필요 없는 장점이 있다.
- ② B-트리로서, 삭제 과정에서 재분배가 필요 없는 장점이 있다.
- ③ B⁺-트리로서, 루트 노드를 제외한 모든 노드는 절반 이상의 사용률(utilization)을 보장한다.
- ④ B-트리로서, 루트 노드를 제외한 모든 노드는 절반 이상의 사용률(utilization)을 보장한다.

10. <보기>의 개체-관계(Entity-Relationship) 모델링에서 EMPLOYEE와 SKILL 간의 관계는? (단, EMPLOYEE는 종업원에 대한 정보가, SKILL은 기술에 대한 정보가 저장되어 있다.)

<보기>

한 개의 EMPLOYEE 튜플은 많은 SKILL 튜플들과 연관될 수 있으며, 한 개의 SKILL 튜플은 많은 EMPLOYEE 튜플들과 연관될 수 있다.

- ① 1 대 1 관계 ② 1 대 다 관계
- ③ 다 대 1 관계 ④ 다 대 다 관계

11. 무결성 규정(integrity rule)에 대한 설명으로 가장 옳지 않은 것은?

- ① 널(NULL) 무결성 제약조건이란 릴레이션의 특정 속성값이 널(NULL)이 될 수 없다는 규정이다.
- ② 엔티티(entity) 무결성 제약조건이란 어떠한 기본키(primary key) 값도 널(NULL)이 될 수 없다는 규정이다.
- ③ 참조 무결성 제약조건이란 외래키 값은 널(NULL)이거나 참조 릴레이션의 기본키(primary key) 값과 달라야 한다는 규정이다.
- ④ 키 무결성 제약조건이란 하나의 테이블에는 적어도 하나의 키가 존재해야 한다는 규정이다.

12. 질의 최적화 방법으로 가장 옳지 않은 것은?

- ① σ (select)는 최대한 빨리 수행한다.
- ② Π (project)는 최대한 빨리 수행한다.
- ③ 수행 비용이 같다면 결과 테이블의 크기가 작은 연산을 먼저 수행한다.
- ④ \times (cartesian product)는 최대한 빨리 수행한다.

13. <보기>에서 사원 테이블의 소속부서 속성은 부서 테이블의 부서번호 속성을 참조하는 외래키이다. 사원 테이블을 정의하는 CREATE TABLE을 작성할 때 외래키에 ON DELETE NO ACTION을 지정하였으면, 부서 테이블에서 홍보부 튜플을 삭제하려고 할 때 발생하는 일로 가장 옳은 것은?

<보기>

부서 테이블		사원 테이블		외래키
부서번호	부서이름	사원번호	사원이름	소속부서
1	연구부	1001	김철수	3
2	홍보부	1002	박수영	1
3	인사부	1003	이영희	2

- ① 부서 테이블의 홍보부 튜플이 삭제되지 않는다.
- ② 사원 테이블에서 이영희 사원에 대한 튜플도 함께 삭제된다.
- ③ 사원 테이블에서 이영희 사원 튜플의 소속부서 속성 값이 널(NULL)로 변경된다.
- ④ 사원 테이블에서 이영희 사원 튜플의 소속부서 속성 값이 미리 지정한 기본값으로 변경된다.

14. 시스템이 교착 상태(deadlock)에 빠졌으면 교착 상태를 야기한 트랜잭션들 중의 일부를 철회해야 한다. 철회시킬 트랜잭션을 선택하는 것을 희생자 선택(victim selection)이라 하며, 희생자 선택을 위한 올바른 방안을 <보기>에서 모두 고른 것은?

<보기>

ㄱ. 먼저 시작한 트랜잭션을 선택
 ㄴ. 나중에 시작한 트랜잭션을 선택
 ㄷ. 많은 갱신을 수행한 트랜잭션을 선택
 ㄹ. 적은 갱신을 수행한 트랜잭션을 선택

- ① ㄱ, ㄷ ② ㄱ, ㄹ
- ③ ㄴ, ㄷ ④ ㄴ, ㄹ

15. <보기>와 같은 함수적 종속성 집합 F를 가지는 릴레이션 R(A, B, C, D, E)을 여러 릴레이션으로 분해하고자 할 때 이에 대한 설명으로 가장 옳지 않은 것은?

<보기>

$F = \{A \rightarrow B, C \rightarrow D, D \rightarrow AE\}$

- ① R에서 모든 속성은 C에 함수적으로 종속될 수 있다.
- ② $R_1(A, B)$, $R_2(C, D, E)$, $R_3(D, A)$ 로 분해할 경우, R_1 , R_2 , R_3 는 제3정규형을 만족한다.
- ③ $R_1(D, A, B)$, $R_2(C, D, E)$ 로 분해할 경우, R_1 , R_2 는 제2정규형을 만족한다.
- ④ $R_1(C, D)$, $R_2(D, A, E)$, $R_3(A, B)$ 로 분해할 경우, R_1 , R_2 , R_3 는 BCNF를 만족한다.

16. <보기>의 두 트랜잭션 T_1 , T_2 를 적절한 제어 없이 동시에 수행할 때 발생할 수 있는 상황에 대한 설명으로 가장 옳지 않은 것은?

<보기>	
T_1	T_2
read_item(X);	read_item(X);
$X = X - 5$;	$X = X * 2$;
read_item(Y);	write_item(X);
write_item(X);	read_item(Y);
$Y = Y + 5$;	read_item(X);
write_item(Y);	

- ① 두 트랜잭션이 끝난 후 Y의 값에 오류가 발생할 가능성은 없다.
- ② 초기 주어진 X의 값이 10이라면 T_1 의 read_item(X)에서 읽어 들인 X의 값은 10일 수도 있고 20일 수도 있다.
- ③ 초기 주어진 X의 값이 10이라면 두 트랜잭션이 끝난 후 X의 값은 10 또는 15가 될 것이다.
- ④ 초기 주어진 Y의 값이 10이라면 T_2 의 read_item(Y)에서 읽어 들인 Y의 값은 10일 수도 있고 15일 수도 있다.

17. NoSQL 개발 동기가 된 기존의 관계형 데이터베이스 관리시스템(RDBMS)의 한계에 대한 설명으로 가장 옳지 않은 것은?

- ① 애초에 테이블을 분할(partition)하여 저장하도록 설계되지 않아 분산처리에 어려움이 있을 수 있다.
- ② 원본 데이터 구조가 테이블 형태가 아닐 경우 변환에 어려움이 있을 수 있다.
- ③ SQL 언어는 테이블 형태가 아닌 데이터에 적용하기에 어려움이 있을 수 있다.
- ④ 일반적으로 RDBMS는 일관성(consistency) 유지 기능이 부족하다.

18. (맥주 → 기저귀)의 관계에서 장바구니 분석(연관 규칙)의 지지도와 신뢰도를 구하는 데 필요한 정보가 아닌 것은?

- ① 맥주가 포함된 거래 횟수
- ② 맥주와 기저귀가 함께 거래된 횟수
- ③ 기저귀가 포함된 거래 횟수
- ④ 전체 거래 횟수

19. <보기>의 릴레이션 R과 S에 대하여 주어진 관계 대수식을 수행하려고 한다. 관계 대수식의 수행 결과로 가장 옳지 않은 것은?

<보기>

R	sid	name	address	year
	1	문재훈	광양	7
	2	김형진	대구	6
	3	유제현	서울	9
	4	박진우	서울	8
	5	김현호	서울	2
	6	정승완	서울	5
	7	박성욱	부산	1

S	sid	score	dept
	1	80	100
	2	80	200
	3	90	300
	4	70	400
	6	100	500

- ① $\sigma_{\text{address}='서울'}(R) - \sigma_{\text{year} \geq 6}(R)$

sid	name	address	year
5	김현호	서울	2
6	정승완	서울	5

- ② $\pi_{\text{name, score}}(\sigma_{\text{score} \geq 80 \text{ and year} > 5}(R \bowtie S))$

name	score
문재훈	80
김형진	80
유제현	90
정승완	100

- ③ $\pi_{\text{name, address, year, score}}(\sigma_{\text{year} \geq 5}(R \bowtie S))$

name	address	year	score
문재훈	광양	7	80
김형진	대구	6	80
유제현	서울	9	90
박진우	서울	8	70
정승완	서울	5	100

- ④ $\pi_{\text{score, dept}}(\sigma_{\text{score} \geq 90}(S))$

score	dept
90	300
100	500

20. <보기 1>은 학생들에 대한 정보를 담고 있는 WORK_INFO 테이블이다. <보기 2>의 SQL 질의를 수행할 때 예상되는 출력 결과는?

<보기 1>

Name	D_name	Salary
심종화	engineering_building	250
이준석	information_building	240
김성희	engineering_building	260
한상유	engineering_building	210
김태의	science_building	310
유상민	computer_building	290
김소희	information_building	220
김은빈	science_building	180
조성국	computer_building	300

<보기 2>

```
CREATE VIEW DEPT_INFO
    (Dept_name, No_of_emps, Total_sal)
AS SELECT D_name, COUNT(*), SUM(Salary)
FROM WORK_INFO
GROUP BY D_name
```

① DEPT_INFO

COUNT(*)	SUM(Salary)
3	720
2	460
2	490
2	590

② DEPT_INFO

D_name	No_of_emps	Total_sal
engineering_building	3	720
information_building	2	460
science_building	2	490
computer_building	2	590

③ DEPT_INFO

Dept_name	COUNT(*)
engineering_building	3
information_building	2
science_building	2
computer_building	2

④ DEPT_INFO

Dept_name	No_of_emps	Total_sal
engineering_building	3	720
information_building	2	460
science_building	2	490
computer_building	2	590