

1. 데이터 웨어하우스(data warehouse)에 대한 설명으로 가장 옳지 않은 것은?

- ① 데이터가 빈번하게 갱신되는 환경에 최적화된 시스템이다.
- ② ETL(extract, transform, load)은 데이터 수집 및 저장 과정에 사용된다.
- ③ 데이터를 데이터 큐브(data cube)의 형태로 저장할 수 있다.
- ④ 스타 스키마(star schema)는 1개의 사실 테이블(fact table)과 여러 개의 차원 테이블(dimension table)들로 구성된다.

2. <보기>는 데이터베이스 관리 시스템(DBMS)의 주요 구성 요소들에 대한 설명이다. ㉠~㉣에 들어갈 요소를 순서대로 옳게 짝지은 것은?

〈보기〉

- ㉠ 는 디스크에 저장되어 있는 데이터에 대한 접근을 제어한다.
- ㉡ 는 데이터 정의어(DDL)로 작성된 데이터베이스 스키마를 메타데이터로 처리하여 시스템 카탈로그에 저장한다.
- ㉢ 는 대화식으로 입력된 데이터 조작어(DML)를 파싱하여 컴파일하고, 데이터베이스에 접근하는 최적의 코드를 생성한다.
- ㉣ 는 호스트 프로그램 내에 삽입된 데이터 조작어(DML)를 추출하고 프로시저 호출로 대체시킨다.

	㉠	㉡	㉢	㉣
①	예비컴파일러 (precompiler)	질의처리기 (query processor)	데이터 정의어(DDL) 컴파일러	트랜잭션관리자
②	예비컴파일러 (precompiler)	질의처리기 (query processor)	트랜잭션관리자	런타임 데이터베이스처리기
③	저장데이터관리자	데이터 정의어(DDL) 컴파일러	질의처리기 (query processor)	예비컴파일러 (precompiler)
④	저장데이터관리자	트랜잭션관리자	데이터 정의어(DDL) 컴파일러	예비컴파일러 (precompiler)

3. 릴레이션 R(A, B, C, D)에 대해 <보기 1>이 성립하고, 이를 <보기 2>와 같이 분해하였을 때, (가) 릴레이션 R의 정규형과 분해 시 (나) 정보 손실 여부를 옳게 짝지은 것은?

〈보기 1〉

함수적 종속성:  $AB \rightarrow C, C \rightarrow A, C \rightarrow D$

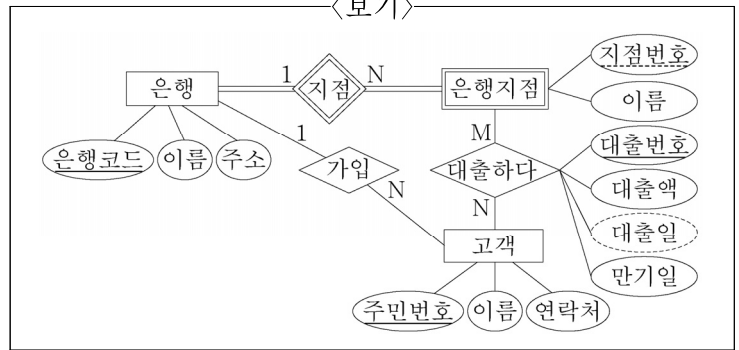
〈보기 2〉

분해:  $R1(A, B, C), R2(C, D)$

- |   | (가)  | (나)    |
|---|------|--------|
| ① | 2정규형 | 무손실 분해 |
| ② | 2정규형 | 손실 분해  |
| ③ | 3정규형 | 무손실 분해 |
| ④ | 3정규형 | 손실 분해  |

4. <보기>의 대출과 관련하여 작성된 E-R 다이어그램에 대한 설명으로 가장 옳지 않은 것은?

〈보기〉



- ① 모든 '고객'은 적어도 하나의 '은행'에 가입되어야 한다.
- ② E-R 다이어그램에 표현된 관계들의 차수는 모두 2이다.
- ③ '은행지점' 엔티티를 릴레이션으로 변환하면 속성은 3개 이상이 된다.
- ④ 모든 '대출'은 1년만 가능하다고 설계되었다면, '대출일'은 '만기일'로부터 유도된 속성이 될 수 있다.

5. 테이블 T1(A, B, C)에 대하여 사용자들의 접근 권한을 설정하기 위해 <보기>와 같은 SQL 명령문을 수행하였을 때, 실행할 수 있는 SQL문은? (단, A, B, C는 정수형이다.)

〈보기〉

Allen: GRANT SELECT, INSERT ON T1  
TO Brown, Clark WITH GRANT OPTION;  
Brown: GRANT SELECT, INSERT ON T1 TO Davis;  
Allen: GRANT DELETE ON T1 TO Clark;  
Allen: REVOKE INSERT ON T1 FROM Brown  
CASCADE;  
Allen: REVOKE GRANT OPTION FOR SELECT  
ON T1 FROM Clark;

- ① Clark: GRANT DELETE ON T1 TO Brown;
- ② Brown: INSERT INTO T1 VALUES (10, 20, 30);
- ③ Davis: INSERT INTO T1 VALUES (10, 20, 30);
- ④ Clark: SELECT \* FROM T1 WHERE C < 10;

6. 로킹(locking)이란 동일한 데이터 항목에 대한 여러 트랜잭션의 임의적인 병행 접근을 방지하는 방법을 말한다. 로킹에 대한 설명으로 가장 옳지 않은 것은?

- ① 로킹 단위는 레코드, 파일, 데이터베이스 등이 될 수 있다.
- ② 로킹 단위가 작아질수록 로크 수가 적어 로킹 오버헤드가 감소한다.
- ③ 교착상태(deadlock)란 두 개 이상의 트랜잭션들이 실행되지 못한 채 다른 트랜잭션이 로크한 항목을 무한정 기다리는 것을 말하며, 교착 상태가 발생하면 일부 트랜잭션을 롤백(rollback)해서 해결할 수 있다.
- ④ 하나의 스케줄에 있는 모든 트랜잭션들이 2단계 로킹 규약(protocol)을 준수한다면 그 스케줄은 직렬 가능성을 보장한다.

7. <보기>는 고객 데이터를 관리하는 프로그램의 일부분을 보여주고 있다. <보기>와 같이 파일에 데이터를 저장하는 방식으로 고객의 데이터를 활용하는 프로그램을 작성할 경우에 대한 설명으로 가장 옳지 않은 것은?

<보기>

```
typedef struct {
    int id; // 고객 번호
    char name[20]; // 고객 이름
    char phone[15]; // 고객 휴대폰 번호
    int point; // 고객이 보유한 포인트
} Client; // 고객 데이터는 data.txt에 저장되어 있음

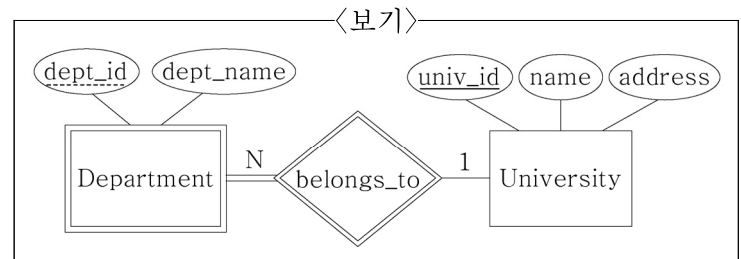
int main() {
    Client client[100];
    FILE *fp;
    // 고객 데이터 파일을 열고 데이터 읽어 들이기
    fp = fopen("data.txt", "rw");
    ...
}
```

- ① 동일한 정보가 여러 파일에 저장되어 있을 경우, 데이터 중복성으로 값이 일관되지 못할 수 있다.
- ② 고객으로부터 수집하는 데이터의 종류가 변경될 경우, 프로그램 코드를 재컴파일해야 한다.
- ③ 동일한 고객 데이터 파일을 둘 이상의 프로그램에서 접근할 경우, 오류가 발생할 수 있다.
- ④ DBMS를 사용하더라도, 프로그램 내에 데이터 정의를 저장해야 한다.

8. 데이터베이스에서 해싱(hashing)은 키 값에 직접 산술적인 연산을 적용하여 계산된 레코드의 저장 주소에 따라 레코드를 삽입하거나 접근하는 방식이다. 해싱기법들 중 확장성 해싱(extendible hashing)에 대한 설명으로 가장 옳은 것은?

- ① 각 버킷(bucket)의 지역 깊이는 전역 깊이보다 크거나 같다.
- ② 지역 깊이와 전역 깊이가 같은 버킷에서 오버플로가 발생하면 디렉터리 배열 내의 엔트리 수는 두 배가 된다.
- ③ 확장성 해싱은 앞으로의 파일 크기 증가를 대비해서 공간을 미리 할당해둔다.
- ④ 레코드 검색은 한 번의 디스크 블록 접근을 필요로 한다.

9. <보기>의 E-R 다이어그램에 대한 설명으로 가장 옳은 것은?



- ① 관계 belongs\_to에서 하나의 Department 개체(entity)는 여러 개의 University 개체에 대응한다.
- ② Department는 독립적으로 존재할 수 있는 개체들의 타입이다.
- ③ belongs\_to는 식별 관계(identifying relationship)이다.
- ④ Department의 기본키(primary-key)는 {dept\_id}이다.

10. 3단계 데이터베이스 구조에 대한 설명으로 가장 옳은 것은?

- ① 매핑 개념을 활용하여 각 단계의 스키마 간의 대응 관계를 정의할 수 있다.
- ② 데이터베이스의 각 단계마다 개념 단계 스키마를 저장한다.
- ③ 외부 단계의 사용자마다 필요한 데이터가 다를 수 있지만 하나의 스키마만 존재한다.
- ④ 물리적 저장장치에 데이터베이스를 저장하는 방식이 변경되면 상위 계층의 스키마도 적절하게 수정해야 한다.

11. B-트리는 탐색 성능을 높이기 위해 균형 있게 높이를 유지하는 균형 트리(balanced tree)의 일종이다. <보기>와 같이 나열된 순서로 키가 삽입되어 B-트리가 구성되었을 때, 루트 노드에 존재하는 키 값은? (단, 이 트리의 차수는 3이다.)

<보기>

키 삽입순서: 7, 19, 14, 15, 3, 5, 10, 20, 6

- ① 7
- ② 10
- ③ 14
- ④ 15

12. 관계 데이터 모델 및 그 제약조건에 대한 설명으로 가장 옳은 것은?

- ① 한 릴레이션(relation) 내의 튜플(tuple)들은 수학적으로 집합이므로 튜플들 간의 순서가 중요하지 않지만, 각 튜플은 (<에트리뷰트(attribute)>, <값>)의 쌍들의 집합으로 해석되므로 순서가 중요하다.
- ② 릴레이션 R의 에트리뷰트들의 부분집합 A에 대하여 현재 R의 어떤 임의의 두 튜플도 동일한 값을 가지지 않는다면 이 집합 A를 릴레이션 R의 후보키라고 한다.
- ③ 대학교의 '학생' 릴레이션의 '학년' 에트리뷰트 값이 1~4 사이의 정수 값을 가져야 한다는 조건은 전이 제약조건(transition constraint)이라고 하며, 트리거(trigger)와 주장(assertion)이라는 기법을 이용하여 보장할 수 있다.
- ④ 릴레이션 R 내의 에트리뷰트 집합 X와 Y 간의 함수 관계는 정규화(normalization)과정에서 활용될 수 있다.

13. 데이터베이스 스키마가 <보기>와 같을 때, 결과가 다른 질의는?

<보기>

```

Student(id, name, dept_id)
Dept(id, dept_name, building)
    
```

- ①  $(\sigma_{name="Kim"}(Student)) \bowtie_{student.dept\_id=dept.id} (\sigma_{building="F2"}(Dept))$
- ②  $\sigma_{building="F2"}((\sigma_{name="Kim"}(Student)) \bowtie_{student.dept\_id=dept.id} Dept)$
- ③  $Student \bowtie_{student.dept\_id=dept.id} (\sigma_{name="Kim" \wedge building="F2"}(Dept))$
- ④  $\sigma_{name="Kim" \wedge building="F2"}(Student \bowtie_{student.dept\_id=dept.id} Dept)$

14. <보기>의 SQL 질의의 의미로 가장 옳은 것은?  
(단, /\* \*/는 주석이다.)

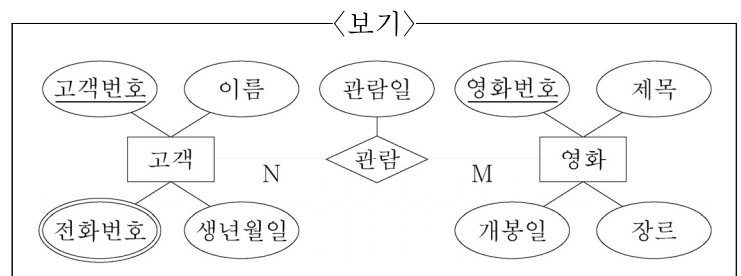
<보기>

```

SELECT Dnumber, COUNT(*)
FROM DEPARTMENT, EMPLOYEE
WHERE Dnumber = Dno AND Salary >= 40000 AND
      Dno IN ( SELECT      Dno
                FROM        EMPLOYEE
                GROUP BY    Dno
                HAVING      COUNT(*) >= 5 )
GROUP BY Dnumber;
/* 릴레이션 스키마: DEPARTMENT(Dnumber, Dname, Mgr_ssn)
   EMPLOYEE(ssn, Ename, Address, Salary, Dno) */
    
```

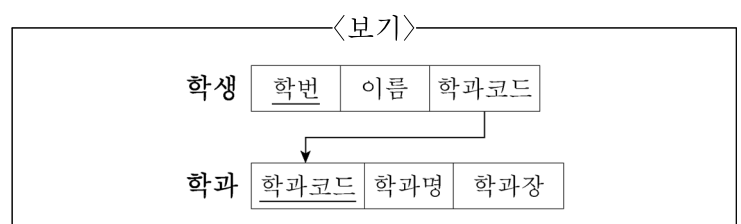
- ① 40,000 이상의 급여를 받는 직원들의 부서번호와 부서별 사원 수를 검색하라.
- ② 40,000 이상의 급여를 받는 직원들이 소속되거나 소속 직원이 5명 이상인 부서의 부서번호와 그 부서에서 40,000 이상의 급여를 받는 사원 수를 검색하라.
- ③ 5명 이상의 사원이 근무하고 40,000 이상의 급여를 받는 직원들이 소속되어 있는 부서의 부서번호와 그 부서에서 40,000 이상의 급여를 받는 사원 수를 검색하라.
- ④ 5명 이상의 사원이 근무하는 부서의 부서번호와 부서별 사원 수를 검색하라.

15. <보기>의 E-R 다이어그램을 관계형 스키마로 변환한 결과로 가장 옳은 것은?



- ① 고객(고객번호, 이름, 생년월일, 관람일),  
고객전화번호(고객번호, 전화번호),  
영화(영화번호, 제목, 장르, 개봉일)
- ② 고객(고객번호, 이름, 생년월일),  
고객전화번호(고객번호, 전화번호),  
영화(영화번호, 제목, 장르, 개봉일),  
관람(고객번호, 영화번호, 관람일)
- ③ 고객(고객번호, 이름, 생년월일, 전화번호),  
영화(영화번호, 제목, 장르, 개봉일),  
관람(고객번호, 영화번호, 관람일)
- ④ 고객(고객번호, 이름, 생년월일),  
영화(영화번호, 제목, 장르, 개봉일),  
관람(고객번호, 관람일)

16. <보기>와 같은 분산 데이터베이스를 단편화하여, 사이트 1에는 학생 테이블, 사이트 2에는 학과 테이블을 저장한다. 제3의 사이트에서 "select 학생.이름, 학과.학과명 from 학생, 학과 where 학생.학과코드 = 학과.학과코드"를 수행하려고 할 때, 사이트 간에 전송되는 데이터의 양을 최소로 한다면, 전송되는 데이터 양[바이트]의 범위는?  
(단, 학과가 없는 학생은 없으며, 학생과 학과 테이블의 모든 속성들의 크기는 10바이트이고, 학생은 10,000명, 학과는 60개이다.)



- ① 0 이상 1,000 미만
- ② 1,000 이상 10,000 미만
- ③ 10,000 이상 100,000 미만
- ④ 100,000 이상

17. <보기>의 부서와 사원 테이블에 대해, 결과 테이블의 카디널리티(cardinality)가 가장 큰 관계 대수식은?

<보기>

부서		사원		
이름	달성률	이름	소속부서	봉급
생산	60	박찬형	생산	350
기획	70	홍길동	생산	250
개발	80	김철수	기획	200
영업	90	김영희	기획	400
		황이수	개발	500
		정순희	영업	300

- ①  $\sigma_{\text{달성률}=70}(\text{부서}) \bowtie_{\text{부서.이름}=\text{사원.소속부서}}(\text{사원})$   
 ②  $\sigma_{\text{사원.봉급}<300}(\text{부서} \times \text{사원})$   
 ③  $\sigma_{\text{봉급}=200 \text{ and } \text{봉급}<500}(\text{사원})$   
 ④  $\sigma_{\text{달성률}<80}(\text{부서}) \bowtie_{\text{부서.이름}=\text{사원.소속부서}}(\text{사원})$

18. <보기>의 두 트랜잭션 T1, T2의 수행에 대한 설명으로 가장 옳지 않은 것은? (단, write(a, b)는 b를 a에 대입한다는 의미이며, 데이터 X, Y, Z의 초깃값은 각각 10, 30, 100이다.)

<보기>

트랜잭션 T1	트랜잭션 T2
read(X) $A \leftarrow X + 10$ write(Y, A)	read(Y) read(Z) $B \leftarrow Y + Z$ write(X, B)

- ① T1과 T2가 직렬적으로(serially) 수행되고 완료되었다면 X의 값은 120이거나 130이 된다.  
 ② 교착상태 방지(deadlock prevention) 프로토콜의 wait-die 규칙은 T1이 먼저 시작됐다면 나중에 시작된 T2를 기다리는 것이 허용된다.  
 ③ 엄격한 2단계 로킹(strict 2PL) 기법에 의해 T1의 read(X) 연산이 먼저 수행되면 T2의 write(X, B) 연산은 T1이 완료되거나 철회된 이후에만 가능하다.  
 ④ 데이터 X, Y, Z가 각각 다른 사이트에 분산 저장된 경우, 3단계 완료 프로토콜(3PC)을 사용하기도 한다.

19. 데이터베이스 시스템이 다운(down)된 시점에 시스템에 <보기>와 같은 데이터베이스 로그가 남아있다고 한다. 데이터베이스 복구를 위해 수행되어야 할 작업에 대한 설명으로 가장 옳지 않은 것은? (단,  $\langle T_i, X, V_1, V_2 \rangle$ 는 트랜잭션  $T_i$ 가 데이터 항목 X의 값을  $V_1$ 에서  $V_2$ 로 변경했음을 나타내며, <checkpoint L>은 체크포인트가 수행된 시점에 L에 포함된 트랜잭션들은 활동 중(active)이었음을 나타낸다.)

<보기>

$\langle T_1, \text{start} \rangle$
$\langle T_1, A, 1, 2 \rangle$
$\langle T_2, \text{start} \rangle$
$\langle T_2, B, 3, 4 \rangle$
$\langle T_1, \text{commit} \rangle$
$\langle \text{checkpoint } \{T_2\} \rangle$
$\langle T_3, \text{start} \rangle$
$\langle T_3, C, 5, 6 \rangle$
$\langle T_3, \text{commit} \rangle$
$\langle T_2, B, 4, 5 \rangle$

- ① A의 값은 수정할 필요 없다.  
 ② B의 값은 최종적으로 4로 복구되어야 한다.  
 ③ T2는 undo되어야 한다.  
 ④ T3는 redo되어야 한다.

20. 릴레이션  $R(a, b, c, d, e)$ 에서  $\{a, b\}$ 가 후보키(candidate-key)일 때, 이에 대한 설명으로 가장 옳지 않은 것은?

- ①  $\{a\}$ 는 후보키(candidate-key)가 될 수 없다.  
 ②  $\{a, b\}$ 는 기본키(primary-key)가 될 수 있다.  
 ③  $\{a, b, c\}$ 는 슈퍼키(superkey)이다.  
 ④  $\{a, b, c, d, e\}$ 는 슈퍼키(superkey)가 아니다.