

# 데이터베이스론

문 1. 데이터 웨어하우스(data warehouse)에 대한 설명으로 옳은 것은?

- ① 스타 스키마는 다차원 데이터를 사실 테이블(fact table)과 차원 테이블(dimension table)을 이용하여 저장하며, 차원 테이블은 질의 처리 성능의 향상을 위해 일반적으로 정규화하지 않는다.
- ② 데이터 웨어하우스는 온라인 트랜잭션 작업과 데이터 분석 작업을 모두 수행하기 위해 사용된다.
- ③ 데이터 웨어하우스를 구축하기 위해 운영 시스템의 데이터를 추출, 변환, 정제하며, 구축된 이후에는 운영 시스템과 분리하여 독립적으로 갱신 및 관리한다.
- ④ 드릴 다운(drill-down)은 데이터 웨어하우스에서 하나의 차원 계층 구조로부터 다른 차원의 계층 구조로 전환시키는 기법이다.

문 2. XML 데이터 처리를 위한 질의어에 대한 설명으로 옳은 것은?

- ① SQL/XML은 XMLELEMENT() 함수나 XMLGEN() 함수를 이용해 테이블에서 XML 문서를 생성한다.
- ② XPath는 XML 문서를 다른 XML 문서로 변환하거나, XML 문서로부터 어떤 원소들을 추출하기 위해 사용한다.
- ③ XSLT는 XML 경로식에 탐색 조건을 수용할 수 있도록 확장하여 질의를 표현한다.
- ④ XQuery는 SQL 언어의 골격인 SELECT-FROM-WHERE 구조를 유지하고 있다.

문 3. 데이터베이스 관리시스템에서 하드디스크 장애가 발생한 경우 데이터베이스의 무결성 보장을 위해 가장 먼저 사용하는 파일은?

- ① 백업 파일
- ② 트랜잭션 로그 파일
- ③ 데이터 사전 파일
- ④ 인덱스 파일

문 4. 다음 두 질의에 대한 설명으로 옳은 것은? (단, 테이블 R(a, b)과 S(b, c)가 있고, 각 애트리뷰트는 NULL을 포함하지 않지만 중복 값을 가지고 있다고 가정한다)

|    |                                                           |
|----|-----------------------------------------------------------|
| q1 | SELECT R.a<br>FROM R, S<br>WHERE R.b = S.b;               |
| q2 | SELECT R.a<br>FROM R<br>WHERE R.b IN (SELECT S.b FROM S); |

- ① q1과 q2의 결과는 같다.
- ② q1의 결과가 항상 q2의 결과를 포함한다.
- ③ q2의 결과가 항상 q1의 결과를 포함한다.
- ④ q1과 q2는 실행할 수 없다.

문 5. 버킷 크기는 2이고, 모조키의 비교를 상위 비트에서 하위 비트 순서로 하는 확장성 해싱(extendible hashing)으로 <사원 테이블>을 저장하려고 한다. <부서별 모조키 값>을 이용하여 부서명을 검색키로 저장할 때, 다음 설명으로 옳은 것은?

| <사원 테이블> |     |     | <부서별 모조키 값> |             |
|----------|-----|-----|-------------|-------------|
| 사번       | 이름  | 부서명 | 부서명         | h(부서명): 모조키 |
| 100      | 홍길동 | 경리부 | 경리부         | 11110001    |
| 200      | 김구  | 총무부 | 총무부         | 10100011    |
| 300      | 이순신 | 인사부 | 영업부         | 11001101    |
| 400      | 안창호 | 기획부 | 인사부         | 001110101   |
| 500      | 심청  | 영업부 | 홍보부         | 01001101    |
| 600      | 강감찬 | 기획부 | 기획부         | 10011000    |
| 700      | 안중근 | 경리부 |             |             |

- ① 사번 100과 사번 500 레코드는 같은 버킷에 저장된다.
- ② 사번 200의 레코드가 저장된 버킷의 지역깊이(local depth)는 2이다.
- ③ 레코드를 저장하기 위해 모두 5개의 버킷이 생성되고, 버킷에 2개의 레코드가 저장된 버킷의 개수는 3개이다.
- ④ 전역깊이(global depth) 값과 사번 300인 레코드가 저장된 버킷의 지역깊이 값을 더하면 4가 된다.

문 6. 다음 SQL문을 실행하였을 때, 이에 대한 설명으로 옳은 것은?

```
CREATE TABLE R (A INTEGER,
                  B INTEGER,
                  PRIMARY KEY(A),
                  FOREIGN KEY(B) REFERENCES S(B)
                  ON DELETE CASCADE);
```

- ① S.B의 값이 100인 것을 삭제하려고 할 때, R.B의 값이 100인 것이 하나라도 있으면 삭제는 수행되지 못한다.
- ② R.B의 값이 100인 것을 삭제하려고 할 때, S.B의 값이 100인 것이 하나라도 있으면 삭제는 수행되지 못한다.
- ③ R.B의 값이 100인 것을 삭제하면 S.B의 값이 100인 것은 모두 삭제된다.
- ④ S.B의 값이 100인 것을 삭제하면 R.B의 값이 100인 것은 모두 삭제된다.

문 7. Kim이 테이블 R(A, B, C)을 생성하고, <보기>와 같은 권한 SQL 명령문을 수행하였다. 실행할 수 없는 SQL문은? (단, A, B, C는 정수형이다)

— <보 기> —

```
Kim : GRANT SELECT, INSERT, DELETE ON R
      TO Han, Lee WITH GRANT OPTION;
Han : GRANT SELECT, INSERT ON R TO Park;
Kim : REVOKE GRANT OPTION FOR DELETE
      ON R FROM Lee;
Kim : REVOKE INSERT, DELETE ON R
      FROM Han CASCADE;
```

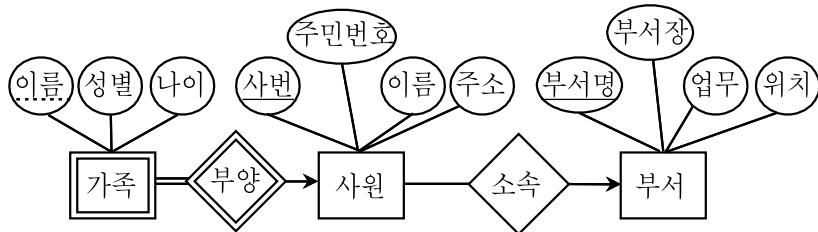
- ① Park : SELECT \* FROM R WHERE A>100 AND B=10;
- ② Han : INSERT INTO R VALUES (100, 200, 300);
- ③ Kim : DELETE FROM R WHERE C<100;
- ④ Lee : GRANT INSERT ON R TO Park;

문 8. 다음 데이터베이스에서 <부서 릴레이션>의 관리자번호가 <직원 릴레이션>의 사번을 참조하도록 외래키(foreign key)를 설정하고 '관리' 관계(relationship)를 표현하였다. 다음과 같이 데이터가 주어졌을 때 '관리' 관계에 대한 설명으로 옳지 않은 것은?

| <직원 릴레이션> |     |      | <부서 릴레이션> |     |       |
|-----------|-----|------|-----------|-----|-------|
| 사번        | 이름  | 부서번호 | 번호        | 부서명 | 관리자번호 |
| 111       | 박영수 | 2    | 1         | 총무부 | 222   |
| 222       | 김창식 | 1    | 2         | 개발실 | 111   |
| 333       | 이민수 | 2    | 3         | 영업부 | 222   |
| 444       | 최종호 | 2    |           |     |       |
| 555       | 민지혜 | 3    |           |     |       |

- ① '관리' 관계는 직원에서 부서쪽으로 1:N 관계이다.
- ② 직원은 '관리' 관계에 부분 참여(partial participation)한다.
- ③ '관리' 관계는 부서에서 직원쪽으로 1:N 관계이다.
- ④ 부서는 '관리' 관계에 전부 참여(total participation)한다.

문 9. 다음과 같은 객체-관계(ER) 다이어그램을 관계형 데이터베이스 테이블로 변환하였을 때 바르게 표현한 것은? (단, 밑줄은 기본키이며, 점선 밑줄은 부분키이다)



- ① 가족(이름, 사번, 주민번호, 성별, 나이),  
사원(사번, 부서명, 주민번호, 이름, 주소),  
부서(부서명, 부서장, 업무, 위치)
- ② 가족(사번, 이름, 성별, 나이),  
사원(부서명, 사번, 주민번호, 이름, 주소),  
부서(부서명, 부서장, 업무, 위치)
- ③ 가족(주민번호, 이름, 사번, 성별, 나이),  
사원(사번, 부서명, 주민번호, 이름, 주소),  
부서(부서명, 부서장, 업무, 위치)
- ④ 가족(사번, 이름, 성별, 나이),  
사원(사번, 부서명, 주민번호, 이름, 주소),  
부서(부서명, 부서장, 업무, 위치)

문 10. 로킹(locking)에 대한 설명으로 옳지 않은 것은?

- ① 로킹 단위가 크면 병행성 수준이 낮아진다.
- ② 2단계 로킹 규약은 2개 이상의 트랜잭션이 동시에 수행될 때 직렬성을 보장한다.
- ③ 2단계 로킹 규약에서는 일단 언로크(unlock) 연산을 실행하면 그 시점 이후에는 더 이상 새로운 로크(lock) 연산을 실행할 수 없다.
- ④ 2단계 로킹 규약은 트랜잭션들이 동시에 수행될 때 교착상태가 발생하지 않도록 보장한다.

문 11. RAID에 대한 설명으로 옳지 않은 것은?

- ① RAID는 디스크 시스템의 성능을 향상시키기 위해 여러 개의 디스크를 병렬로 연결하여 하나의 고성능 논리적 디스크처럼 동작한다.
- ② RAID에서 디스크 시스템의 성능 향상은 데이터를 다중의 디스크에 투명하게 분산시키는 미러링(mirroring) 기법을 통해 구현한다.
- ③ RAID를 구성하는 디스크 수가 증가할수록 신뢰성이 떨어지는데, 이에 대한 해결책으로 데이터를 중복해서 저장한다.
- ④ RAID 레벨 0은 여분의 데이터를 갖지 않고, 데이터의 갱신이 중복해서 일어나지 않으므로 가장 좋은 쓰기 성능을 가진다.

문 12. 다음은 민원인이 관공서에 발급 신청한 서류의 접수대장을 저장하기 위한 테이블이다. 이 테이블이 충족하는 가장 높은 정규형은? (단, 밑줄은 기본키이다)

접수대장(접수일련번호, 민원인주민등록번호, 민원인성명, 신청서류, 수량, 접수일자)

- ① 제1정규형
- ② 제2정규형
- ③ 제3정규형
- ④ BCNF

문 13. SQL 주입(injection)에 대한 설명으로 옳은 것은?

- ① 응용 프로그램에 SQL문을 삽입하여 DB 연동작업을 수행하게 하는 일종의 프로그래밍 기법이다.
- ② DBMS의 저장 프로시저에서 오류를 일으키는 부분의 SQL문을 탐지하는 일종의 진단기법이다.
- ③ 응용에 들어있는 SQL문의 일부를 변경하여 불법적인 작업을 수행하는 일종의 보안 공격이다.
- ④ 응용을 개발할 때 사용자가 지정한 부분에 자동으로 SQL문을 생성하는 일종의 개발도구 기능이다.

문 14. 검사시점(checkpoint) 회복 기법에서 장애 발생 시점의 로그의 내용이 다음과 같다고 할 때, Undo-list와 Redo-list를 바르게 나열한 것은?

<T1, Start>  
<T2, Start>  
<T3, Start>  
<T1, Commit>  
<Checkpoint, {T2, T3}>  
<T2, Commit>  
<T4, Start>  
<T4, Commit>

|   | Undo-list | Redo-list    |
|---|-----------|--------------|
| ① | {T3}      | {T4}         |
| ② | {T3}      | {T2, T4}     |
| ③ | {T3}      | {T1, T2, T4} |
| ④ | {T3, T4}  | {T1, T2}     |

문 15. 다음 릴레이션 R과 함수 종속 집합 FD를 참고하여 R의 후보키(candidate key)를 모두 고른 것은?

R(A, B, C, D, E)  
FD:  $A \rightarrow B$ ,  $B \rightarrow A$ ,  $AC \rightarrow DE$

- ① AC                                      ② A, B, AB  
③ AB                                      ④ AC, BC

문 16. 조인 연산을 구현하는 방법에 대한 설명으로 옳지 않은 것은?

- ① 정렬-합병 조인방법은 두 릴레이션이 각각 조인 애트리뷰트 값에 따라 정렬되어 있는 경우에 유용하다.  
② 중첩 루프(nested loop) 조인방법은 모든 조인 비교연산자를 처리할 수 있다.  
③ 세타 조인(theta-join)의 경우 메모리가 충분하면 해시 조인 방법이 유리하다.  
④ 중첩 루프 조인방법에서 내부 릴레이션의 조인 애트리뷰트에 인덱스가 존재할 경우 조인 연산을 효율적으로 처리할 수 있다.

문 17. 다음 세 트랜잭션에 대한 설명으로 옳지 않은 것은? (단, lock-X는 배타 로크(exclusive lock), lock-S는 공유 로크(shared lock)를 의미하며, 최초 A의 값이 1,000이고 B의 값이 500이라고 가정한다)

| T1                                                                                                                                       | T2                                                                                                                                       | T3                                                                                            |
|------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------|
| lock-X(B);<br>read(B);<br>B := B + 100;<br>write(B);<br>unlock(B);<br>lock-X(A);<br>read(A);<br>A := A - 100;<br>write(A);<br>unlock(A); | lock-X(B);<br>read(B);<br>B := B - 100;<br>write(B);<br>lock-X(A);<br>read(A);<br>A := A + 100;<br>write(A);<br>unlock(B);<br>unlock(A); | lock-S(A);<br>read(A);<br>lock-S(B);<br>read(B);<br>display(A+B);<br>unlock(A);<br>unlock(B); |

- ① T1과 T3이 동시에 수행될 경우 트랜잭션의 결과 값이 항상 옳게 나온다는 보장은 할 수 없다.  
② T2와 T3이 동시에 수행될 경우 교착상태에 빠질 수 있다.  
③ T2와 T3이 동시에 수행될 경우 출력(display) 결과는 두 트랜잭션의 수행 순서와 상관없이 1,500이다.  
④ T1, T2, T3은 모두 2단계 로킹 규약을 따르고 있다.

문 18. 트랜잭션의 격리 수준(isolation level)에 대한 설명으로 옳지 않은 것은?

- ① 읽기 완료(read committed) 수준에서는 팬텀 읽기(phantom read)가 발생하지 않는다.  
② 반복가능 읽기(repeatable read) 수준에서는 오손 읽기(dirty read)와 반복불능 읽기(non-repeatable read)가 발생하지 않는다.  
③ 직렬가능(serializable) 수준은 가장 강하고 제한적인 격리 수준이다.  
④ 격리 수준이 더 제한적일수록 트랜잭션 처리율이 떨어진다.

문 19. <보기>와 같은 명령어들을 차례로 실행하였다. 다음 SQL문을 실행할 때, 오류가 발생하지 않는 것은?

— <보 기> —  
CREATE TABLE 부서  
(부서번호 NUMBER(2) PRIMARY KEY,  
부서명 CHAR(15),  
위치 CHAR(15));  
CREATE TABLE 사원  
(사원번호 NUMBER(4) PRIMARY KEY,  
이름 CHAR(10) NOT NULL,  
연령 NUMBER(2),  
급여 NUMBER(7, 2) NOT NULL,  
부서번호 NUMBER(2) REFERENCES 부서);  
INSERT INTO 부서 VALUES (1, 'ACCOUNTING', 'SEOUL');  
INSERT INTO 사원 VALUES (10, 'Jain', 20, 1000, 1);  
INSERT INTO 사원 VALUES (20, 'Tom', 21, 2000, 1);

- ① SELECT 이름, MAX(급여)  
FROM 사원, 부서  
WHERE 연령 < 40 AND 사원.부서번호 = 부서.부서번호  
GROUP BY 부서명;  
② INSERT INTO 사원  
SELECT \* FROM 사원 WHERE 이름 LIKE 'J%' AND 1 = 0;  
③ INSERT INTO 사원 VALUES (30, 'Smith', 3000, 1);  
④ SELECT 부서명  
FROM 부서  
WHERE 부서번호 = (SELECT 부서번호  
FROM 사원  
WHERE 급여 >= (SELECT MIN(급여)  
FROM 사원));

문 20. B+ 트리에 대한 설명으로 옳지 않은 것은?

- ① B+ 트리는 루트노드(root node)에서 모든 단말노드(leaf node)까지의 높이가 동일하다.  
② B+ 트리의 단말노드들은 순차 접근이 가능하도록 링크로 연결되어 있다.  
③ 비클러스터링(nonclustering) B+ 트리 인덱스에서는 데이터 레코드들이 정렬되어 있다.  
④ B+ 트리는 팬 아웃(fan-out) 값이 커서 트리의 높이가 낮기 때문에 이진 트리보다 데이터베이스 인덱스로서 훨씬 효율적이다.