데이터베이스론

문 1. DBMS의 3단계 스키마 아키텍처와 데이터 독립성에 대한 설명으로 옳지 않은 것은?

① 개념 단계는 개념 스키마를 가지며, 개념 스키마는 표현 데이터 모델을 이용해서 나타낸다.

② 외부 단계는 외부 스키마나 사용자 뷰를 포함하며, 외부 스키마를 기술할 때 표현 데이터 모델을 사용하여 구현된다.

③ 논리적 데이터 독립성은 외부 스키마나 응용 프로그램을 변경하지 않으면서 개념 스키마를 변경할 수 있는 것을 말한다.

④ 3단계 스키마 아키텍처에서 실제로 존재하는 데이터는 내부 단계와 개념 단계에 존재한다.

문 2. 데이터베이스에서 널(null) 값이 적용되는 예로 적절하지 않은 것은?

① A 남자 환자의 경우, 출산여부 항목

② B 신입생의 주소가 아직 파악되지 않은 경우, 주소 항목

③ 연봉이 동결된 C 사원의 경우, 연봉인상률 항목

④ D 제품에 대한 제품가격이 담당부서에서 아직 넘어오지 않은 경우, 제품가격 항목

문 3. 다음과 같은 로그에서 지연 갱신(deferred update) 회복기법을 이용할 때, 시스템 붕괴 후 복구 과정에 대한 설명으로 옳은 것은?

begin_transaction, T1
read, T1, A
read, T1, D
write, T1, D, 20
commit, T1
checkpoint
begin_transaction, T2
read, T2, B
write, T2, B, 18
begin_transaction, T4
read, T4, D
write, T4, D, 35
begin_transaction, T3
write, T3, C, 30
read, T4, A
write, T4, A, 30
commit, T4
read, T2, D
write, T2, D, 25

← 시스템 붕괴

① 트랜잭션 T1은 검사점(checkpoint)을 기록하기 전에 트랜잭션이 완료되었으므로 redo 되어야 한다.

② 트랜잭션 T2는 read 및 write 연산이 직렬 가능하게 수행되었으므로 redo 되어야 한다.

③ 트랜잭션 T3은 read 연산 없이 write 연산만을 수행하였으므로 undo 되어야 한다.

④ 트랜잭션 T4는 시스템 붕괴 이전에 완료되었기 때문에 redo 되어야 한다.

문 4. 릴레이션 R(A, B, C, D, E, F)에 대한 함수적 종속성 집합이 {A→BC, E→CF, B→E, CD→EF} 일 때, 속성 집합 {A, C}의 폐포(closure)를 구한 것은?

① {A, B, C, F}

② {A, B, C, D, E}

③ {A, B, C, D, F}

④ {A, B, C, E, F}

문 5. 릴레이션 R(A, B, C)의 기본키(primary key)인 속성 A에 대하여 생성된 B^+-트리 인덱스에 대한 설명으로 옳은 것은? (단, 속성 A의 자료형은 integer이다)

① R의 투플들을 속성 A값 순서로 검색하는 것을 효율적으로 지원할 수 없다.

② R의 투플 삭제 시 B^+-트리에서 노드에 언더플로우가 발생하면 항상 형제 노드와의 합병으로 해결해야 한다.

③ 리프 노드는 모두 동일한 레벨에 위치해 있으며, 한 노드 안의 키 값들은 정렬되어 있다.

④ "SELECT * FROM R WHERE A=100"의 수행에 이용할 수 있으며, 결과 투플의 검색은 리프 노드에 도달하기 전에 완료될 수 있다.

문 6. 주어진 트랜잭션과 스케줄에 대하여 직렬 가능한(serializable) 스케줄을 모두 고른 것은? (단, r은 read 연산을 의미하고, w는 write 연산을 의미한다)

○ 트랜잭션

T1 : r1(X); r1(Z); w1(X);

T2 : r2(Z); r2(Y); w2(Z); w2(Y);

T3 : r3(X); r3(Y); w3(Y);

○ 스케줄

S1 : r1(X); r2(Z); r1(Z); r3(X); r3(Y); w1(X); w3(Y); r2(Y); w2(Z); w2(Y);

S2 : r1(X); r2(Z); r3(X); r1(Z); r2(Y); r3(Y); w1(X); w2(Z); w3(Y); w2(Y);

S3 : r1(X); r2(Z); r2(Y); w2(Z); r3(X); r3(Y); w3(Y); w2(Y); r1(Z); w1(X);

① S1 ② S2

③ S1, S2 ④ S3

문 7. 다음 장바구니 모델의 트랜잭션에서 최소 지지도(support)를 70%라고 할 때, 빈발 항목집합(frequent itemset)을 모두 고른 것은?

트랜잭션 ID	구입물품
1	기저귀, 맥주, 분유
2	기저귀, 분유
3	기저귀, 두부, 맥주
4	기저귀, 두부, 맥주, 양파

① {기저귀}, {맥주}

② {기저귀}, {맥주}, {기저귀, 맥주}

③ {기저귀}, {맥주}, {기저귀, 두부}

④ {기저귀}, {맥주}, {기저귀, 맥주}, {기저귀, 맥주, 분유}

문 8. 축구팀과 선수들에 대한 릴레이션 스키마와 관계 대수식이 다음과 같을 때, 동등한 관계 대수식을 모두 고른 것은? (단, 밑줄은 기본키이다)

○ 릴레이션 스키마
축구팀(<u>이름</u>, 연고지, 경기수, 승률)
선수(<u>이름</u>, 소속팀, 연봉)

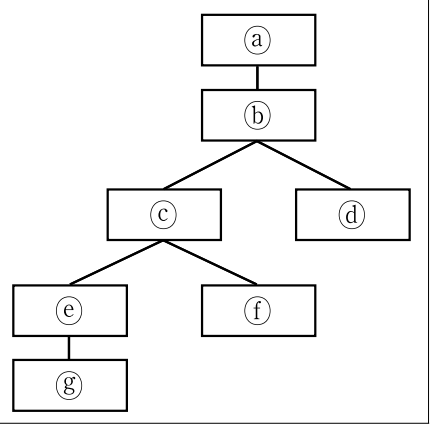
○ 관계 대수식
E1 : $\Pi_{\text{축구팀.승률,선수.이름}}(\sigma_{\text{축구팀.이름=선수.소속팀 and 선수.연봉>20000}}(\text{축구팀} \times \text{선수}))$
E2 : $\Pi_{\text{축구팀.승률,선수.이름}}(\Pi_{\text{이름,승률}}(\text{축구팀}) \bowtie_{\text{축구팀.이름=선수.소속팀}} (\sigma_{\text{연봉>20000}}(\text{선수})))$
E3 : $\Pi_{\text{축구팀.승률,선수.이름}}(\Pi_{\text{이름,승률}}(\text{축구팀}) \bowtie_{\text{축구팀.이름=선수.소속팀}} (\sigma_{\text{연봉>20000}}(\Pi_{\text{연봉,소속팀}}(\text{선수}))))$
E4 : $\Pi_{\text{축구팀.승률,선수.이름}}(\text{축구팀} \bowtie_{\text{축구팀.이름=선수.소속팀}} (\sigma_{\text{선수.연봉>20000}}(\text{선수})))$

① E1, E2 ② E2, E3
③ E1, E2, E4 ④ E1, E2, E3, E4

문 9. 테이블 P(p1, p2), S(s1, s2, s3), T(t1, t2)에 대하여 <SQL 질의문>을 주어진 <보기>를 이용하여 <질의문 트리>를 구성할 때, ⓐ ~ ⓖ에 들어갈 요소가 바르게 짝지어진 것은?

<SQL 질의문>
```
SELECT P.p1, S.s1, T.t1, T.t2
FROM   P, S, T
WHERE P.p1 = S.s2 and S.s3 = T.t1 and P.p2 = 'KOREA'
```

<보 기>
ㄱ. P
ㄴ. S
ㄷ. T
ㄹ. $\Pi_{\text{P.p1,S.s1,T.t1,T.t2}}$
ㅁ. $\sigma_{\text{P.p2 = 'KOREA'}}$
ㅂ. $\bowtie_{\text{P.p1 = S.s2}}$
ㅅ. $\bowtie_{\text{S.s3 = T.t1}}$

<질의문 트리>


	ⓐ	ⓑ	ⓒ	ⓓ	ⓔ	ⓕ	ⓖ
①	ㄹ	ㄴ	ㅂ	ㄷ	ㅁ	ㅅ	ㄱ
②	ㄹ	ㅅ	ㅂ	ㄷ	ㅁ	ㄴ	ㄱ
③	ㄹ	ㅅ	ㅂ	ㅁ	ㄷ	ㄴ	ㄱ
④	ㅁ	ㅅ	ㅂ	ㄷ	ㄹ	ㄴ	ㄱ

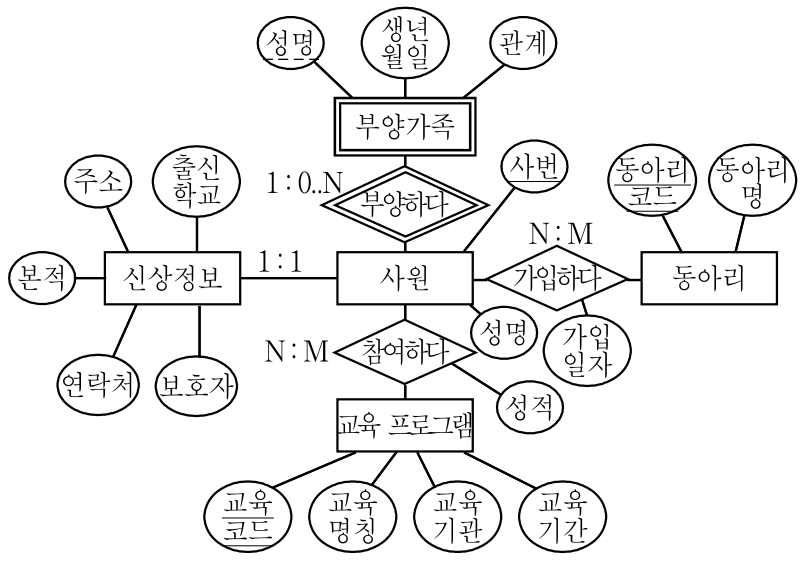
문 10. 로크(lock) 기반의 동시성 제어 기법에 대한 설명으로 옳지 않은 것은?
① 2단계 로킹(two-phase locking) 규약을 이용하면 직렬 가능 스케줄을 보장한다.
② 교착상태(deadlock)가 발생할 수 있다.
③ 2단계 로킹 규약에서는 트랜잭션이 어떠한 로크도 해제하지 않은 경우에만 새로운 로크를 요청할 수 있다.
④ 2단계 로킹 규약을 따르면 트랜잭션은 항상 트랜잭션이 시작한 순서대로 완료된다.

문 11. 최근 들어 데이터베이스 시스템의 성능 향상을 위해서 메인 메모리, 플래쉬 메모리, RAID, 클라우드 시스템 상에서 서비스 되는 데이터베이스 시스템들이 개발되고 있다. 이러한 데이터베이스 시스템에 대한 설명으로 적절하지 않은 것은?
① 메인 메모리 데이터베이스에서는 로그(log)를 메인 메모리에만 저장하므로 속도가 빠르다.
② 플래쉬 메모리는 쓰기 속도가 읽기 속도에 비해 느리기 때문에 이를 고려하여 데이터베이스 시스템이 구현된다.
③ RAID를 통하여 데이터 안전성의 향상과 디스크 접근 속도의 향상을 동시에 얻을 수 있다.
④ 클라우드 시스템을 통해서 시스템 부하에 따른 유연한 확장성을 얻을 수 있다.

문 12. 트랜잭션들의 수행에 있어 교착 상태 방지(deadlock prevention) 프로토콜에서 일어나는 동작으로 옳지 않은 것은?
① 트랜잭션이 필요로 하는 모든 항목에 대해 로크를 미리 획득하도록 한다.
② 데이터베이스의 모든 항목들에 대해 정해진 순서에 따라 로크를 획득하도록 한다.
③ 대기 그래프를 유지하며 사이클의 존재 여부를 결정한다.
④ 각 항목에 대해 트랜잭션 타임스탬프들의 값을 이용하여 대기 여부를 결정한다.

문 13. 다음 <요구 조건>을 표현한 회사의 개체-관계(Entity-Relationship) 다이어그램을 관계 데이터베이스 스키마로 변환할 때, 적절하지 않은 것은? (단, 밑줄은 기본키이고, 점선 밑줄은 부분키이다)

<요구 조건>
○ 모든 사원은 부양가족이 없거나 여러 명을 가질 수 있다.
○ 모든 사원은 여러 개의 동아리에 가입할 수 있다.
○ 한 사원에 대한 신상정보는 별도로 분리하여 관리한다.
○ 한 사원은 필요에 따라 여러 개의 교육 프로그램에 참여할 수 있다.



① 부양가족(<u>성명</u>, 생년월일, 관계)
② 신상정보(<u>사번</u>, 주소, 본적, 출신학교, 보호자, 연락처)
③ 가입하다(<u>사번</u>, <u>동아리코드</u>, 가입일자)
④ 참여하다(<u>사번</u>, <u>교육코드</u>, 성적)

문 14. 함수 종속성에 대한 추론 규칙으로 옳지 않은 것은?
① 재귀성(reflexive) 규칙 : $X \supseteq Y$이면 $X \rightarrow Y$이다.
② 부가성(augmentation) 규칙 : $X \rightarrow Y$이면 $XZ \rightarrow YZ$이다.
③ 이행성(transitive) 규칙 : $X \rightarrow Y$이고 $Y \rightarrow Z$이면 $X \rightarrow Z$이다.
④ 분해성(decomposition) 규칙 : $X \rightarrow Y$이고 $WX \rightarrow Z$이면 $WX \rightarrow Y$이다.

문 15. 다음과 같은 7개의 모조키(pseudo key) 값을 갖는 키가 순서대로 삽입되어 확장성 해싱(extendible hashing)이 구축되었을 때, 전역 깊이(global depth) 값과 R7이 저장되는 버킷(bucket)에 저장된 키의 개수를 합한 값은? (단, 버킷의 크기는 2이고, 삽입을 위한 모조키의 비교는 상위 비트에서 하위 비트 순서로 한다)

<삽입순서>

h(R1)=01101, h(R2)=10011, h(R3)=01000, h(R4)=10010,
h(R5)=00110, h(R6)=00011, h(R7)=11101

① 2 ② 3
③ 4 ④ 5

문 16. 다음은 회원과 동호회 관리를 위해 모델링된 릴레이션 스키마와 SQL 질의문이다. 이 SQL 질의문을 관계대수로 변환한 것으로 옳지 않은 것은? (단, 밑줄은 기본키이다)

○ 릴레이션 스키마
회원(<u>회원번호</u>, 주민등록번호, 회원명, 관심분야)
동호회(<u>동호회ID</u>, 동호회이름, 개설일, 관리자회원번호)
가입(<u>회원번호</u>, <u>동호회ID</u>)

○ SQL 질의문
```
SELECT a.회원명, a.관심분야
FROM   회원 a, 가입 b, 동호회 c
WHERE c.동호회이름='자연사랑'
  AND c.동호회ID=b.동호회ID
  AND b.회원번호=a.회원번호
```

① $\Pi_{\text{회원명,관심분야}}(\sigma_{\text{동호회이름='자연사랑'}}((\Pi_{\text{회원번호,회원명,관심분야}}$ 회원) $\bowtie_{\text{회원번호=회원번호}}$(가입$\bowtie_{\text{동호회ID=동호회ID}}$ 동호회)))

② $\Pi_{\text{회원명,관심분야}}(\sigma_{\text{동호회이름='자연사랑'}}(\Pi_{\text{회원명,관심분야}}$(회원$\bowtie_{\text{회원번호=회원번호}}$(가입$\bowtie_{\text{동호회ID=동호회ID}}$ 동호회))))

③ $\Pi_{\text{회원명,관심분야}}$(회원$\bowtie_{\text{회원번호=회원번호}}$(가입$\bowtie_{\text{동호회ID=동호회ID}}$ ($\sigma_{\text{동호회이름='자연사랑'}}$ 동호회)))

④ $\Pi_{\text{회원명, 관심분야}}(\sigma_{\text{동호회이름='자연사랑'}}$(회원$\bowtie_{\text{회원번호=회원번호}}$(가입$\bowtie_{\text{동호회ID=동호회ID}}$ 동호회)))

문 17. 다음을 관계 데이터베이스 스키마로 설계했을 때, 가장 적절한 것은? (단, 밑줄은 기본키이다)

○ 대학에서 3, 4학년 학생은 교수로부터 진로 지도를 받는다(교수 : 학생 = 1 : N 관계).
○ 학생들 사이에도 멘토링을 실시하며, 멘티(mentee)인 1 ~ 3학년 학생이 멘토(mentor)인 4학년 학생으로부터 조언을 받는다(멘토 : 멘티 = 1 : N 관계).

① 학생(<u>학번</u>, 이름, 학과, 멘토, 멘토링내용)
교수(<u>교수ID</u>, 이름, 학과, 지도학생학번, 지도내용)
② 학생(<u>학번</u>, 이름, 학과, 멘티, 멘토링내용)
교수(<u>교수ID</u>, 이름, 학과, 지도학생학번, 지도내용)
③ 학생(<u>학번</u>, 이름, 학과, 지도교수ID, 지도내용, 멘토, 멘토링내용)
교수(<u>교수ID</u>, 이름, 학과)
④ 학생(<u>학번</u>, 이름, 학과, 지도교수ID, 지도내용, 멘티, 멘토링내용)
교수(<u>교수ID</u>, 이름, 학과)

문 18. 다음의 트랜잭션 T1과 T2는 동시에 수행될 때, 트랜잭션 간의 직렬성을 보장하지 못한다. 트랜잭션 T1과 T2의 직렬성을 보장하기 위해 2단계 로킹(two-phase locking) 규약을 적용하여 각각 트랜잭션 T3과 T4로 변환할 때, 단계 ㉠, ㉡, ㉢, ㉣에서 수행해야 할 문장이 바르게 짝지어진 것은?

T1	T2	T3	T4
read_lock(B);	read_lock(A);	read_lock(B);	read_lock(A);
read_item(B);	read_item(A);	read_item(B);	read_item(A);
unlock(B);	unlock(A);	㉠ ;	㉢ ;
write_lock(A);	write_lock(B);	㉡ ;	㉣ ;
read_item(A);	read_item(B);	read_item(A);	read_item(B);
A = A + B;	B = A + B;	A = A + B;	B = A + B;
write_item(A);	write_item(B);	write_item(A);	write_item(B);
unlock(A);	unlock(B);	unlock(A);	unlock(B);

	㉠	㉡	㉢	㉣
①	unlock(A)	write_lock(B)	unlock(B)	write_lock(A)
②	unlock(B)	write_lock(A)	write_lock(B)	unlock(A)
③	write_lock(A)	unlock(B)	unlock(A)	write_lock(B)
④	write_lock(A)	unlock(B)	write_lock(B)	unlock(A)

문 19. 다음과 같은 일련의 권한 부여 SQL 명령어를 실행하였을 때, 이에 대한 설명으로 옳지 않은 것은?

U1 : GRANT SELECT, INSERT ON STUDENT TO U2 WITH GRANT OPTION
U2 : GRANT SELECT, INSERT ON STUDENT TO U3
U1 : REVOKE GRANT OPTION FOR SELECT ON STUDENT FROM U2
U1 : REVOKE INSERT ON STUDENT FROM U2 CASCADE

① U2는 STUDENT에 대한 삽입 권한을 다른 사용자에게 부여할 수 없다.
② U2는 STUDENT에 대한 검색 권한을 다른 사용자에게 부여할 수 있다.
③ U2, U3은 STUDENT에 대한 삽입 권한이 없다.
④ U3은 STUDENT에 대한 검색 권한이 있다.

문 20. XML 문서와 데이터베이스에 대한 다음의 설명 중 옳지 않은 것은?

① DBMS를 이용하여 XML 문서 전체를 하나의 애트리뷰트 값으로 저장하는 방식은 데이터 중심(data-centric) XML 문서를 효율적으로 저장할 수 있다.
② DBMS를 이용하여 XML 문서 전체를 하나의 애트리뷰트 값으로 저장하는 방식은 문서의 검색 성능을 높이기 위해 키워드 인덱싱 기술이 활용될 수 있다.
③ DBMS를 이용하여 XML 문서의 내용을 여러 데이터 요소로 나누어 저장하는 방식은 XML 스키마 또는 DTD(Document Type Definition)에 명시된 구조에 대응하는 데이터베이스 스키마를 생성할 필요가 있다.
④ DBMS를 이용하여 XML 문서의 내용을 여러 데이터 요소로 나누어 저장하는 방식은 데이터베이스에 저장되어 있는 데이터로부터 XML 문서를 생성해 내는 매핑 알고리즘이 필요하다.