

데이터베이스론

문 1. 파일처리 시스템에서 데이터 중복의 단점에 대한 설명으로 옳지 않은 것은?

- ① 같은 데이터가 여러 곳에 중복되어 있어 동일 수준의 보안이 가능하다.
- ② 데이터 저장공간에 대한 추가 비용이 소요된다.
- ③ 데이터 간의 불일치로 인해 데이터 일관성이 결여된다.
- ④ 데이터 갱신 시 중복된 모든 데이터를 찾아내어 갱신해야 하므로 갱신 비용이 추가된다.

문 2. 다음 데이터베이스 설계의 주요 단계를 순서대로 바르게 나열한 것은?

ㄱ. 정규화	ㄴ. 물리적 설계
ㄷ. 요구사항 수집 및 분석	ㄹ. 개념적 설계
ㅁ. 데이터베이스 튜닝	ㅂ. 논리적 설계

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

문 3. 함수적 종속성과 정규화에 대한 설명으로 옳지 않은 것은?

- ① 함수적 종속성 $X \rightarrow Y$ 는 릴레이션 내 임의의 두 튜플에서 속성 X의 값이 같을 경우, 속성 Y의 값도 항상 같음을 의미한다.
- ② 함수적 종속성은 릴레이션 상태(외연)의 특성이다.
- ③ 제2정규형에서 릴레이션의 모든 비주요 속성들은 릴레이션의 기본키에 완전하게 함수적으로 종속한다.
- ④ 정규화는 함수적 종속성과 기본키를 기반으로 주어진 릴레이션 스키마를 분석하는 과정이다.

문 4. 위치 기반 서비스(Location-Based Services)를 제공하는 시스템의 요소로 옳지 않은 것은?

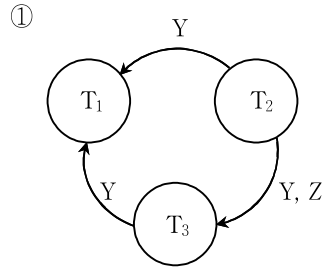
- ① 푸시(push) 기반 서비스
- ② 최인접 주변 질의
- ③ TSQL2 언어
- ④ 윈도우 질의

문 5. SQL 질의문은 동치인 관계대수식으로 변환할 수 있다. 다음 SQL 질의문을 관계대수식으로 변환한 것 중 옳은 것은? (단, 제품 테이블 P, 판매 테이블 S, 집계함수 표현을 위한 관계대수 연산자 기호는 $\bowtie$ 로 한다)

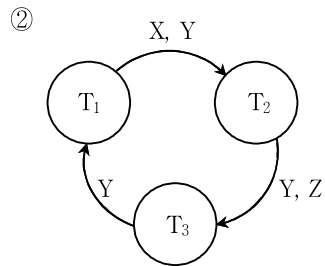
Schema: S(sId, sDate, pId, sAmount)
P(pId, pName)
SQL: SELECT SUM(sAmount)
FROM P, S
WHERE S.pId = P.pId AND P.pName='P1'
GROUP BY sDate;

- ① $\bowtie_{SUM(sAmount)} (\pi_{P.pName} (\sigma_{P.pName='P1'} \bowtie_{S.pId=P.pId} P))$
- ② $sDate \bowtie_{SUM(sAmount)} (\pi_{S.sAmount} (\sigma_{P.pName='P1'} (S \bowtie_{S.pId=P.pId} P)))$
- ③ $\bowtie_{SUM(sAmount)} (S \bowtie_{S.pId=P.pId} (\sigma_{P.pName='P1'} (P)))$
- ④ $sDate \bowtie_{SUM(sAmount)} (\sigma_{P.pName='P1'} (P) \bowtie_{P.pId=S.pId} S)$

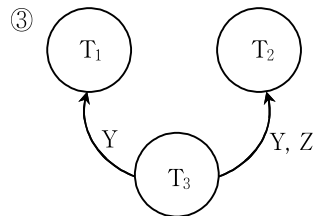
문 6. 다음 트랜잭션의 실행그래프 및 그와 동치인 직렬 가능한 스케줄의 쌍들이다. 이 쌍들 중 옳지 않은 것은?



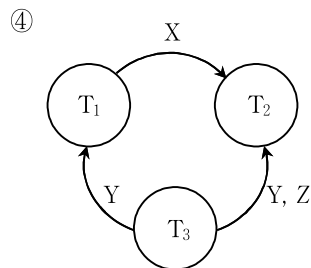
동치 직렬 스케줄: $T2 \rightarrow T3 \rightarrow T1$



동치 직렬 스케줄: $T3 \rightarrow T1 \rightarrow T2$



동치 직렬 스케줄: $T3 \rightarrow T2 \rightarrow T1$



동치 직렬 스케줄: $T3 \rightarrow T1 \rightarrow T2$

문 7. 다음 '사원' 테이블을 생성하는 SQL문에서 부서 속성값을 '인사', '연구', '영업'으로 제한하고자 한다. ㉠, ㉡에 들어갈 내용으로 옳은 것은?

CREATE TABLE 사원
(사번 NUMBER NOT NULL,
이름 CHAR(10),
직급 CHAR(10),
부서 CHAR(10) ㉠ (부서 ㉡ ('인사', '연구', '영업'));

㉠ ㉡

- ① UNIQUE ON
- ② CHECK ON
- ③ UNIQUE IN
- ④ CHECK IN

문 8. 2개의 계좌 A1, A2의 잔액이 각각 2,000,000원, 3,000,000원이라고 하자. 이 상태에서 각 계좌의 관리를 수행하는 트랜잭션 T1 및 T2의 실행이 다음과 같을 때, 발생한 현상으로 옳은 것은?

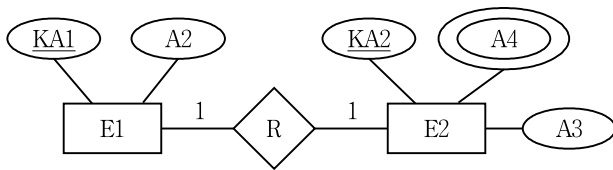
- 1: T2가 계좌 A2에 2,500,000원을 더해 5,500,000원이 됨
- 2: T1이 계좌 A1에 1,500,000원을 더해 3,500,000원이 됨
- 3: T2가 계좌 A1의 잔액(3,500,000원)이 충분하다고 판단하여 A1에서 A2로 2,500,000원 이체를 실행함
- 4: T1이 알 수 없는 이유로 트랜잭션의 수행이 실패함

- ① 오손 판독(dirty read)
- ② 반복할 수 없는 판독(non-repeatable read)
- ③ 부정확한 요약(incorrect summary)
- ④ 유령 판독(phantom read)

문 9. 로킹(locking)을 이용한 DBMS 동시성 제어 기법의 설명으로 옳은 것은?

- ① 2단계 로킹 프로토콜(two-phase locking protocol)의 확대 단계에서는 로크를 해제할 수 없지만, 축소 단계에는 로크를 해제한 후 같은 항목에 대하여 다시 로크를 획득할 수 있다.
- ② 로킹 기법을 사용하여 트랜잭션의 동시성을 제어하면 항상 직렬 가능(serializable)한 스케줄을 보장한다.
- ③ 다중 단위 로킹(multiple granularity locking) 프로토콜에서 SIX(intention-shared-exclusive) 로크가 걸려있는 노드에 IS(intention-shared) 로크를 추가로 걸 수 있다.
- ④ 2단계 로킹 프로토콜을 활용하면 불필요한 교착상태(deadlock)를 회피할 수 있어서 트랜잭션의 철회(rollback)를 막을 수 있다.

문 10. 다음 ERD(Entity-Relationship Diagram)를 관계형 데이터베이스 스키마로 변환한 것으로 옳지 않은 것은?(단, 밑줄은 기본키이다)



- ① E1(KA1, A2)
E2(KA2, A3, KA1)
E21(KA2, A4)
- ② E1(KA1, A2, KA2)
E2(KA2, A3)
E21(KA2, A4)
- ③ E1(KA1, A2)
E2(KA2, A3)
R(KA1, KA2)
E21(KA2, A4)
- ④ E1(KA1, A2, KA2)
E2(KA2, A3, A4)

문 11. (가), (나), (다)는 데이터베이스의 보안 위협에서 보호되어야 하는 성질을 설명하고 있다. ㉠, ㉡, ㉢에 들어갈 용어로 옳은 것은?

- (가) (㉠)은 정보가 부적절한 변경으로부터 보호되어야 하는 성질을 의미한다.
- (나) (㉡)은 적법한 권리가 있는 사용자 또는 프로그램이 객체를 사용할 수 있어야 하는 성질을 의미한다.
- (다) (㉢)은 권한을 갖지 않은 노출로부터 데이터를 보호해야 하는 성질을 의미한다.

㉠

㉡

㉢

- | | | |
|-------|-----|-----|
| ① 무결성 | 가용성 | 조직성 |
| ② 무결성 | 가용성 | 기밀성 |
| ③ 기밀성 | 무결성 | 가용성 |
| ④ 기밀성 | 무결성 | 조직성 |

문 12. 다음 관계형 데이터베이스 릴레이션의 특성으로 옳은 것만을 모두 고르면?

- ㄱ. 한 릴레이션에는 동일한 값을 가지는 튜플이 두 개 이상 존재할 수 없다.
- ㄴ. 한 릴레이션에서 한 속성의 값은 모두 같은 도메인에 속해야 한다.
- ㄷ. 한 릴레이션에서 튜플의 순서는 중요하지 않다.
- ㄹ. 한 릴레이션에서 속성의 순서는 중요하지 않다.
- ㅁ. 한 릴레이션에서 속성은 다중값(multiple value)을 포함할 수 있다.

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

문 13. 다음은 장비구니 트랜잭션의 예이다. 연관규칙(Association Rule)을 발견하기 위해 Apriori 알고리즘을 적용한다고 가정할 때, 다음 설명으로 옳지 않은 것은?

TID	Items
1	A, F
2	A, B, D, E
3	B, C, D
4	A, B, D, F
5	A, C, D, F

- ① 항목집합 {A}의 지지도(support)는 80 %이다.
- ② 항목집합 {A, B, D}가 빈발하면 항목집합 {A, D}도 빈발한다.
- ③ 규칙 $A \rightarrow \{B, D\}$ 의 신뢰도(confidence)가 $\{A, B\} \rightarrow D$ 의 신뢰도보다 크거나 같다.
- ④ 규칙 $A \rightarrow B$ 의 지지도는 40 %이다.

문 14. 다음 EMPLOYEE, PROJECT, WORKSON 테이블로 구성된 데이터베이스에서, “모든 프로젝트에 참가하는 직원의 이름을 검색하라.”를 수행하기 위한 SQL문으로 옳은 것은?

```
EMPLOYEE( eNo, eName )
PROJECT( pNo, pName )
WORKSON( eNo, pNo )
```

- ① SELECT eName
FROM EMPLOYEE E, WORKSON W, PROJECT P
WHERE E.eNo = W.eNo
AND W.pNo = P.pNo;
- ② SELECT eName
FROM EMPLOYEE E
WHERE NOT EXISTS (SELECT P.pNo
FROM PROJECT P
WHERE NOT EXISTS (SELECT W.pNo
FROM WORKSON W
WHERE P.pNo = W.pNo AND E.eNo = W.eNo));
- ③ SELECT eName
FROM EMPLOYEE E
WHERE EXISTS ((SELECT pNo FROM PROJECT)
NOT EXCEPT
(SELECT pNo
FROM WORKSON W
WHERE E.eNo=W.eNo));
- ④ SELECT eName
FROM EMPLOYEE E
WHERE EXISTS ((SELECT pNo FROM PROJECT)
EXCEPT
(SELECT pNo
FROM WORKSON W
WHERE E.eNo = W.eNo));

문 15. 하루 수천 건의 거래가 발생하는 온라인 쇼핑몰에서 ‘주문정보’ 테이블은 대용량의 데이터를 저장해야 하고 트랜잭션이 자주 발생하기 때문에 주문정보 시스템의 성능에 크게 영향을 준다. 다음 질의가 빈번하게 발생하는 상태에서 성능향상을 위하여 ‘주문정보’ 테이블의 ‘주문일자’ + ‘순번’ 속성으로 구성된 키(key)에 대해 인덱스를 생성하고자 할 때, 성능향상에 적합한 인덱스 구조로 옳은 것만을 모두 고르면?

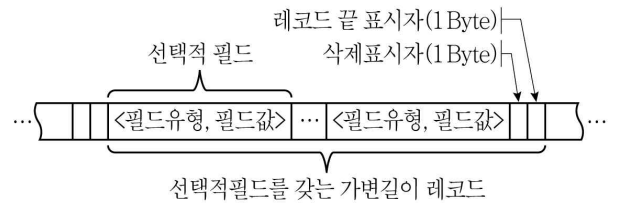
```
SELECT *
FROM 주문정보
WHERE 주문일자 BETWEEN '03-01-2016' AND '03-31-2016';
```

- ㄱ. B⁺-트리 인덱스
- ㄴ. B*-트리 인덱스
- ㄷ. 해시 인덱스
- ㄹ. 비트맵(bitmap) 인덱스

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

문 16. 다음의 ‘선택적필드’를 갖는 가변길이 레코드 10,000개가 저장된 PLAYER 파일이 있다. PLAYER 파일 내의 레코드는 <필드유형, 필드값>으로 표현되는 ‘선택적필드’들로 구성되어 있으며, 레코드 끝 부분에 1바이트의 ‘삭제표시자’와 1바이트의 ‘레코드 끝 표시자’를 포함한다. PLAYER 파일에 저장되어 있는 레코드들의 상태는 <선택적필드를 갖는 가변길이 레코드 저장 상태> 표에 제시되어 있으며, 레코드를 저장하기 위한 블록 구조 역시 <PLAYER 파일 구성을 위한 블록 구조> 표에 제시되어 있다. 이와 같은 상태에서 PLAYER 파일에 저장되어 있는 레코드들의 평균 크기와 필요한 블록 수를 계산한 결과는?

<PLAYER 파일의 구조>



<선택적필드를 갖는 가변길이 레코드 저장 상태>

필드이름	필드유형 크기	필드값 크기	선택적필드 참여율
TEAM	1바이트	39바이트	90 %
SALARY	1바이트	29바이트	80 %
INJURY	1바이트	9바이트	10 %
UNIVERSITY	1바이트	9바이트	70 %

- 필드이름은 ‘선택적필드’를 식별하기 위한 것으로 ‘선택적 필드’의 ‘필드유형’ 값으로 코드화되어 저장된다.
- 필드유형 크기는 ‘필드유형’을 저장하기 위한 항목으로 파일 내에 있는 모든 ‘선택적필드’에서 1바이트의 크기를 갖는다.
- 필드값 크기는 ‘선택적필드’에서 ‘필드유형’을 제외한 ‘필드값’을 저장하기 위한 크기로 ‘선택적필드’에 따라 적합한 크기를 갖는다.
- 선택적필드 참여율은 파일 내 전체 레코드들에서 해당 ‘선택적필드’가 나타나는 비율을 의미한다.

<PLAYER 파일 구성을 위한 블록 구조>

블록조직의 형태	블록 크기	기타
신장(spanned) 조직	2400바이트	파일 내 다음 번 블록을 가리키기 위한 4바이트 크기 포인터 필드 포함

레코드들의 평균 크기

필요한 블록의 수

- ① 70바이트 293개
- ② 71바이트 296개
- ③ 70바이트 291개
- ④ 71바이트 294개

문 17. 릴레이션 스키마 R(A, B, C, D, E, F, G, H, I)에 대하여 다음과 같은 함수적 종속성의 집합 F가 주어졌을 때, 스키마 R의 속성들의 부분집합 {A, G}에 대하여, F 하의 {A, G}의 폐포(closure) {A, G}⁺로 옳은 것은?

F = {A → B, BH → D, C → EF, BG → IE, AE → C}

- ① {A, B, C, E, G, H, I}
- ② {A, C, D, E, G, I, F}
- ③ {A, C, D, E, G, H, I}
- ④ {A, B, C, E, G, I, F}

문 18. 다음은 장애 발생 시 지연 갱신 회복 기법으로 복구하는 데이터베이스 시스템의 로그 기록이다. T3를 수행하던 도중 시스템 장애가 발생했을 때 수행되는 복구 절차로 옳은 것은?

1: (T1 Start)
2: (T1, X, 1000)
3: (T1, Y, 1500)
4: (T1 Commit)
5: (T2 Start)
6: (T2, Z, 1300)
7: (T2 Commit)
8: (T3 Start)
9: (T3, X, 2000)

← Z 장애 발생

- ① T1과 T2를 redo 연산을 수행하여 복구하고, T3에 대해서는 undo 연산을 수행한다.
- ② T1과 T2를 undo 연산을 수행하여 복구하고, T3에 대해서는 redo 연산을 수행한다.
- ③ T1과 T2를 redo 연산을 수행하여 복구하고, T3에 대해서는 별도의 연산을 수행하지 않는다.
- ④ T1과 T2는 이미 데이터베이스에 반영되어 있으므로 별도의 연산을 수행하지 않고, T3를 redo 연산을 수행하여 복구한다.

문 19. 다음은 트랜잭션 격리수준(Isolation Level) 설정을 위한 SQL 표준 형식이다. 이를 이용하여 설정할 수 있는 트랜잭션 격리수준에 대한 설명으로 옳지 않은 것은?

SET TRANSACTION {READ ONLY | READ WRITE}
ISOLATION LEVEL {READ UNCOMMITTED
| READ COMMITTED
| REPEATABLE READ
| SERIALIZABLE }

- ① READ UNCOMMITTED로 설정된 트랜잭션은 완료되지 않은 트랜잭션이 생성한 데이터 값을 읽어가는 것을 허용하는 것이며, READ WRITE와 함께 설정될 수 없다.
- ② READ COMMITTED로 설정된 트랜잭션은 동일 조건의 데이터 집합을 반복하여 읽을 경우, 먼저 읽었던 데이터 집합에 있던 데이터 값과 새로이 읽은 데이터 집합에 있는 데이터 값이 달라져 있을 수 있다.
- ③ REPEATABLE READ로 설정된 트랜잭션은 동일 조건의 데이터 집합을 반복하여 읽을 경우, 읽혀지는 집합을 구성하는 로우들의 수가 달라질 수 있다.
- ④ SERIALIZABLE로 설정된 트랜잭션은 격리성을 완전하게 유지함으로써 트랜잭션의 일관성을 보존하고 동시에 수행 중인 트랜잭션들의 처리율을 향상시킨다.

문 20. <보기 1>은 관계형 데이터베이스 스키마와 이 스키마에 대하여 “1980년 이후에 출생하고 프로젝트명이 ‘DBMS개발’인 프로젝트를 수행하는 연구원 이름을 검색하라.”를 SQL로 표현한 것이다. <보기 2>는 경험적 최적화 기법을 적용하여 최적화를 수행하기 위해, <보기 1>의 SQL 질의문에서 생성되는 초기 질의트리를 최적화된 질의트리로 변환하는 과정에서 유도되는 관계대수식의 목록이다. 최적화 순서에 맞도록 바르게 나열한 것은? (단, 연구원 테이블 R, 프로젝트 테이블 P, 프로젝트 팀 테이블 T라 한다)

<보기 1>

Schema: R(rNo, rName, bDate, address)

P(pNo, pName, pPeriod)

T(rNo, pNo, pBudget)

SQL 질의문:

SELECT rName

FROM R, T, P

WHERE pName = 'DBMS개발'

AND P.pNo = T.pNo AND R.rNo = T.rNo

AND bDate > '12-31-1979';

<보기 2>

ㄱ. $\pi_{rName} (\pi_{rNo} (\pi_{pNo} (\sigma_{pName = 'DBMS개발'} (P)) \bowtie_N \pi_{rNo, pNo} (T)) \bowtie_N \pi_{rNo, rName} (\sigma_{bDate > '12-31-1979'} (R))))$

ㄴ. $\pi_{rName} (\sigma_{P.pNo = T.pNo \wedge T.rNo = R.rNo \wedge pName = 'DBMS개발' \wedge bDate > '12-31-1979'} ((R \times T) \times P))$

ㄷ. $\pi_{rName} (\sigma_{P.pNo = T.pNo} (\sigma_{T.rNo = R.rNo} (\sigma_{bDate > '12-31-1979'} (R) \times T) \times \sigma_{pName = 'DBMS개발'} (P)))$

ㄹ. $\pi_{rName} (\sigma_{pName = 'DBMS개발'} (P) \bowtie_N T) \bowtie_N \sigma_{bDate > '12-31-1979'} (R)$

ㅁ. $\pi_{rName} (\sigma_{R.rNo = T.rNo} (\sigma_{P.pNo = T.pNo} (\sigma_{pName = 'DBMS개발'} (P) \times T) \times \sigma_{bDate > '12-31-1979'} (R)))$

- ① ㄱ → ㄴ → ㄷ → ㄹ → ㅁ
- ② ㄴ → ㄱ → ㄷ → ㄹ → ㅁ
- ③ ㄴ → ㄷ → ㄱ → ㄹ → ㅁ
- ④ ㄱ → ㄴ → ㄷ → ㄹ → ㅁ