

데이터베이스론

문 1. 데이터베이스 관리 시스템(DBMS)의 역할에 대한 설명으로 옳지 않은 것은?

- ① 데이터 조작용(DML)로 스키마의 구조를 기술하여 시스템 카탈로그(혹은 데이터 사전)에 저장한 후 필요할 때 활용한다.
- ② 질의어 처리기는 질의문을 파싱하고 분석해서 효율적인 데이터 베이스 접근코드를 생성한다.
- ③ 트랜잭션 관리자는 무결성 제약조건 검사, 사용자의 권한 검사, 병행제어, 회복 등의 작업을 수행한다.
- ④ 저장 데이터 관리자는 디스크에 저장되어 있는 사용자 데이터 베이스와 시스템 카탈로그의 접근을 책임진다.

문 2. 관계형 데이터베이스에 대한 설명으로 옳지 않은 것만을 모두 고른 것은?

- ㄱ. 기본키 속성이 복합 속성인 경우 그 속성의 일부 요소 속성에서 널(NULL) 값을 가질 수 있다.
- ㄴ. 슈퍼키는 후보키가 되기 위한 필요충분조건이다.
- ㄷ. 릴레이션 R이 릴레이션 S를 참조하는 경우 R의 외래키가 S의 기본키가 아닌 후보키 중 하나를 참조해야 한다.
- ㄹ. 테이블에 튜플 삽입 시 엔터티 무결성 혹은 키 제약조건, 도메인 제약조건, 참조 무결성 제약조건이 위배될 수 있다.

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

문 3. 테이블의 튜플 데이터를 파일 내에 저장하기 위한 구조는 힙(heap) 파일구조, 순차(sequential) 파일구조, 해시(hash) 파일 구조로 구분될 수 있다. 일반적으로 테이블에 가장 빈번하게 이루어지는 연산에 의해 사용될 파일 구조가 결정된다. 다음에 주어진 각각의 연산에 대하여 가장 효율적인 파일구조를 바르게 연결한 것은?

- ㄱ. 속성 값의 범위 구간을 주고 이에 해당하는 튜플 검색(range search)
- ㄴ. 빈번한 튜플 삽입
- ㄷ. 특정 속성 값과 일치하는 튜플 검색

- | | | |
|------|----|----|
| ㉠ | ㉡ | ㉢ |
| ① 해시 | 힙 | 순차 |
| ② 힙 | 순차 | 해시 |
| ③ 순차 | 힙 | 해시 |
| ④ 순차 | 해시 | 힙 |

문 4. 부서와 사원 테이블을 생성하는 SQL 문장을 수행한 후 튜플 삽입으로 두 테이블의 상태가 다음과 같을 때, 테이블 연산 수행에 대한 설명으로 옳지 않은 것은?

CREATE TABLE 부서 (

부서번호 INT NOT NULL,
부서명 VARCHAR(20),
PRIMARY KEY(부서번호));

CREATE TABLE 사원 (

사번 INT NOT NULL,
이름 VARCHAR(20),
부서번호 INT,
PRIMARY KEY(사번),
FOREIGN KEY(부서번호) REFERENCES 부서(부서번호));

부서	
부서번호	부서명
1	자재부
2	영업부

사원		
사번	이름	부서번호
11	홍길동	1
12	이순신	2

- ① 부서 테이블에서 (2, '영업부') 튜플을 삭제한다면 참조 무결성 제약조건을 위배한다.
- ② 사원 테이블에 (13, '강감찬', 'A1') 튜플을 삽입한다면 도메인 무결성 제약조건을 위배한다.
- ③ 사원 테이블에 (14, '김유신', 0) 튜플을 삽입한다면 참조 무결성 제약조건을 위배한다.
- ④ 부서 테이블에 (1, '연구부') 튜플을 삽입한다면 참조 무결성 제약조건을 위배한다.

문 5. 인덱싱(indexing) 기법에 대한 설명으로 옳은 것은?

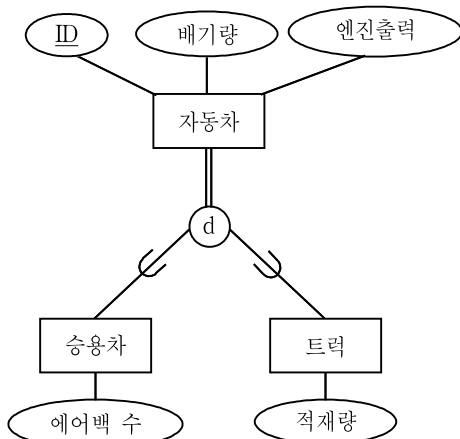
- ① 순서(ordered) 인덱싱은 키 값을 이용하여 찾고자 하는 튜플이 저장된 물리적 주소를 계산해내는 기법이다.
- ② 차수가 p인 B트리의 모든 노드는 최소한 $\left\lceil \frac{p}{2} \right\rceil$ 개의 트리 포인터를 갖는다.
- ③ B트리와 B+트리의 노드 크기가 동일한 경우, B+트리의 내부 노드 차수(degree)가 B트리 노드 차수보다 더 크다.
- ④ 100명이 저장된 학생 테이블의 '성별' 속성에 대하여 비트맵 인덱스를 만들 때, 남학생의 비트맵 인덱스 크기는 남학생 수에 영향을 받는다.

문 6. 두 트랜잭션 T_1 과 T_2 가 다음과 같은 트랜잭션 스케줄로 실행될 때 발생하는 문제는? (단, 데이터 항목 X와 Y의 초기값은 각각 200과 300이고, read(X)와 write(X)는 각각 트랜잭션이 데이터 항목 X를 읽고 쓰는 연산이다)

T_1	T_2	시간 ↓
read(X); $X = X - 20$;		
	read(X); $X = X + 50$;	
write(X); read(Y);		
	write(X);	
$Y = Y - 20$; write(Y);		
	read(Y); $Y = Y + 30$; write(Y);	

- ① 반복할 수 없는 읽기 문제(unrepeatable read problem)
- ② 부정확한 요약 문제(incorrect summary problem)
- ③ 오손 읽기 문제(dirty read problem)
- ④ 갱신 손실 문제(lost update problem)

문 7. ISA 관계를 슈퍼타입(혹은 상위클래스)과 서브타입(혹은 하위 클래스)으로 표현한 EER(Enhanced E-R) 다이어그램의 한 예를 다음 그림에서 보여주고 있다. 이 그림에서 슈퍼타입은 전체 세분화(total specialization)되어 있고, 서브타입들은 서로 분리(disjoint)되어 있다. 이 EER 다이어그램에 대한 설명으로 옳지 않은 것은? (단, 밑줄은 기본키를 의미한다)

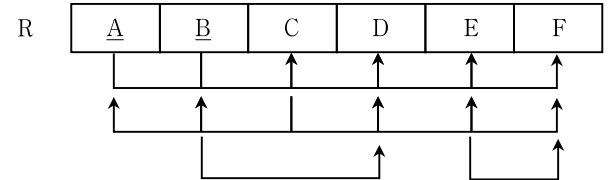


- ① 트럭 엔터티의 속성의 개수는 4개이다.
- ② 위 EER 다이어그램에서는 승용차도 아니고 트럭도 아닌 자동차(예를 들면 버스)는 존재할 수 없다.
- ③ 자동차 엔터티를 테이블로 표현하지 않고, ID, 배기량, 엔진출력 속성을 포함한 승용차와 트럭 테이블로 표현할 수 있다.
- ④ 승용차와 트럭이 중복(overlap)되었다면 자동차 엔터티는 테이블로 변환할 수 없다.

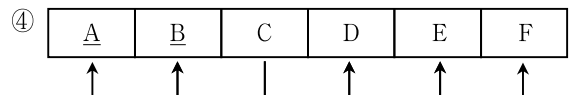
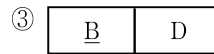
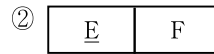
문 8. 데이터베이스 관리 시스템의 캐시 관리방식에 대한 설명으로 옳지 않은 것은?

- ① no-steal 방식에서는 회복과정 중 UNDO와 REDO 연산의 수행이 모두 필요하다.
- ② 갱신된 페이지의 수가 많고 크기가 큰 경우 no-steal 방식에 비해 steal 방식이 필요한 캐시 버퍼 크기를 줄일 수 있다.
- ③ 완료된(committed) 트랜잭션에서 갱신된 페이지가 다른 트랜잭션들에서도 빈번히 갱신되는 경우, force 방식에 비해 no-force 방식이 디스크로부터 캐시로 그 페이지를 다시 읽는 횟수를 줄일 수 있다.
- ④ force 방식은 트랜잭션이 완료되기 전에 그 트랜잭션이 갱신한 모든 페이지들을 디스크에 저장하게 한다.

문 9. 다음 그림은 릴레이션 R과 그 함수적 종속성을 표현하고 있다. 속성 C는 릴레이션 R의 후보키이며, 이 릴레이션은 이미 제1정규화를 수행하였다. 이 릴레이션을 후보키까지 고려하여 제2정규화하였을 때, 분해된 릴레이션 중 기본키가 A, B가 아닌 릴레이션은? (단, 밑줄은 기본키를 의미한다)



- ① 제2정규화하더라도 변화 없음



문 10. 병렬 데이터베이스 구조에 대한 설명으로 옳지 않은 것은?

- ① 공유 메모리(shared memory) 구조는 비공유(shared nothing) 구조에 비해 프로세서 간 통신 속도가 빠르지만 프로세서 수의 확장성이 낮다.
- ② 공유 디스크(shared disk) 구조는 공유 메모리 구조에 비해 메모리 버스의 병목현상과 프로세서 간 통신 속도를 모두 줄인다.
- ③ 분산 가상 메모리(distributed virtual memory) 구조는 논리적으로는 단일 공유 메모리 구조와 동일하게 사용할 수 있지만 물리적으로는 다중 분리 메모리(multiple disjoint memory) 구조로 구축될 수 있다.
- ④ 비공유 구조는 공유 디스크 구조에 비해 한 프로세서에서 다른 프로세서가 관리하는 비지역(non-local) 디스크 접근에 대한 비용이 높을 수 있다.

문 11. RAID에 대한 설명으로 옳지 않은 것은?

- ① 장애가 발생했을 때 손실된 데이터를 복구하기 위해 정상 수행 시 여분의 정보를 별도의 저장 공간에 저장해둠으로써 신뢰성을 높일 수 있다.
- ② Level 0은 동일한 데이터를 여러 디스크에 중복하여 저장함으로써 읽기 성능을 향상시킨다.
- ③ Level 3은 별도의 한 개의 디스크에 패리티 비트만을 저장한다.
- ④ Level 5는 모든 디스크가 읽기 작업에 참여한다.

문 12. 다음 과일 테이블에 대한 SQL 문장 내의 비교조건을 해석한 것으로 옳지 않은 것은? (단, 밑줄은 기본키를 의미한다)

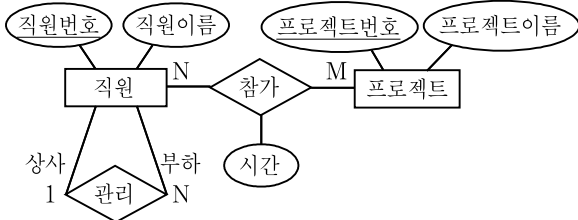
과일	
<u>과일코드</u>	과일명
10	오렌지
15	키위
19	파인애플

- ① “21 NOT IN (SELECT 과일코드 FROM 과일)”은 참이다.
- ② “19 < ANY (SELECT 과일코드 FROM 과일)”은 거짓이다.
- ③ “15 < ALL (SELECT 과일코드 FROM 과일)”은 참이다.
- ④ “19 = ALL (SELECT 과일코드 FROM 과일)”은 거짓이다.

문 13. 데이터베이스의 위협 요소와 보안에 대한 설명으로 옳지 않은 것은?

- ① 통계 데이터베이스는 원본 데이터로부터 통계 정보를 추출한 것이므로 보안의 대상이 아니다.
- ② GRANT와 REVOKE는 데이터베이스에 대한 접근 권한을 부여하거나 취소하기 위한 SQL 명령어이다.
- ③ 데이터베이스에 대한 접근 제어는 권한이 없는 사용자나 프로그램이 데이터를 검색하거나 변경하는 것을 방지하기 위한 것이다.
- ④ 데이터베이스에 대한 무결성(integrity) 제약조건이란 부적절한 갱신으로부터 데이터를 보호하기 위한 조건이다.

문 14. 다음 ERD(Entity-Relationship Diagram)를 관계형 데이터베이스 스키마로 가장 적절하게 변환한 것은? (단, 밑줄은 기본키, 이탤릭체는 외래키를 의미한다)



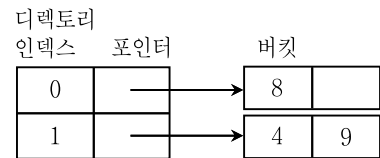
- ① 직원(직원번호, 직원이름), 관리(상사직원번호, 부하직원번호), 프로젝트(프로젝트번호, 프로젝트이름), 참가(시간)
- ② 직원(직원번호, 직원이름), 프로젝트(프로젝트번호, 프로젝트이름), 참가(직원번호, 프로젝트번호, 시간)
- ③ 직원(직원번호, 직원이름, 상사직원번호), 프로젝트(프로젝트번호, 프로젝트이름), 참가(직원번호, 프로젝트번호, 시간)
- ④ 직원(직원번호, 직원이름, 부하직원번호), 프로젝트(프로젝트번호, 프로젝트이름), 참가(직원번호, 프로젝트번호, 시간)

문 15. 메인 메모리 데이터베이스(main memory database) 시스템의 동작에 대한 설명으로 옳지 않은 것은?

- ① 메인 메모리 데이터베이스와 디스크 기반 데이터베이스가 동일한 데이터를 저장하고 관리한다면 내부 인덱스 구조도 동일해야 한다.
- ② 메인 메모리의 데이터를 이용하여 연산을 수행하지만 로깅(logging)은 상대적으로 저속의 비휘발성 저장공간(예를 들면 디스크)에 수행해야 하므로 성능 향상에 방해요소가 될 수 있다.
- ③ 디스크 기반 데이터베이스보다 메인 메모리 데이터베이스가 트랜잭션 처리 속도 면에서 더 좋은 성능을 보인다.
- ④ 그룹 완료(group commit) 기법은 트랜잭션이 완료를 요청 하더라도 즉시 처리하는 대신, 완료를 기다리는 트랜잭션의 개수가 일정 수 이상이 되거나 일정 시간이 경과한 후 완료 연산을 수행하는 것이다.

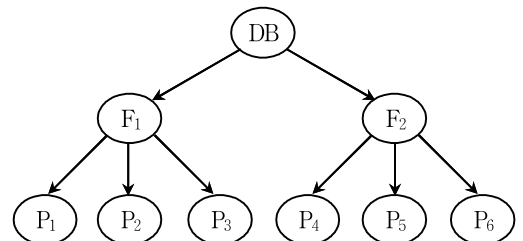
문 16. 다음 그림은 키 값에 대한 해시함수 h의 결과와, 확장 가능 해싱(extendible hashing) 기법에 의해 최초로 키 값 4, 8, 9를 순서대로 입력하여 수행한 결과이다. 이후 키 값 11, 7, 2, 5, 13을 순서대로 입력하는 경우에 대한 설명으로 옳은 것은? (단, 버킷(bucket)의 크기는 2이고 입력된 키 값은 삽입한 순서대로 버킷 내의 왼쪽부터 저장된다)

키 값	해시함수 h(키 값)
4	1010
8	0001
9	1101
11	1110
7	0110
2	1111
5	0111
13	0101



- ① 모든 키 값이 삽입된 후 디렉토리 인덱스 111이 가리키는 버킷의 첫 번째 키 값은 11이다.
- ② 모든 키 값이 삽입된 후 디렉토리 인덱스 010이 가리키는 버킷에는 2개의 키 값이 존재한다.
- ③ 모든 키 값이 삽입된 후 버킷의 개수는 5이다.
- ④ 키 값 7이 삽입될 때 새 버킷이 생성된다.

문 17. 다음 그림은 다중 단위크기 로킹(multiple granularity locking) 기법에서 3계층 단위크기의 예를 보여주고 있다. 다중 단위크기 2단계 로킹 기법의 동작에 대한 설명으로 옳지 않은 것은? (단, 그림에서 DB는 F₁과 F₂로 구성되어 있고 F₁은 P₁, P₂, P₃로 구성되어 있음을 의미하며, 로크 S는 공유(Shared), X는 배타(eXclusive), I는 의도(Intention)를 의미한다)



- ① P₃에 읽기 연산을 수행하기 위해서는 DB와 F₁에 순서대로 IS 로크를 건 후 P₃에 S 로크를 걸어야 한다.
- ② 트랜잭션 T₁이 F₁에 올바르게 S 로크를 걸고 읽기 작업을 진행하던 중 트랜잭션 T₂가 P₃에 쓰기 작업을 진행하려고 하면 로크를 걸지 못하여 대기해야 한다.
- ③ 트랜잭션 T₁이 F₁에 읽기 작업을 위해 로크를 거는 것과 트랜잭션 T₂가 F₂에 쓰기 작업을 위해 로크를 거는 것은 충돌이 발생하지 않는다.
- ④ 트랜잭션 T₁이 P₁, P₄에 순서대로 작업을 해야할 때, P₄와 관련된 로크를 걸기 전이라도 P₁에 대한 작업을 완료하자마자 로크를 해제해주어 다른 트랜잭션들이 로크를 걸기 위해 대기하는 시간을 줄일 수 있다.

문 18. 데이터베이스 관리자 DBA가 생성한 테이블 T에 대해 사용자 USR1이 T에 대한 뷰 V를 생성한 후 사용자 USR2가 V에 대한 검색을 실행할 수 있도록 할 때, 반드시 필요한 권한부여 작업만을 모두 나열한 것은?

- ㄱ. DBA가 “GRANT CREATE VIEW TO USR1”을 실행
 ㄴ. DBA가 “GRANT SELECT ON T TO USR1”을 실행
 ㄷ. DBA가 “GRANT SELECT ON T TO USR1 WITH GRANT OPTION”을 실행
 ㄹ. USR1이 “GRANT SELECT ON T TO USR2”를 실행
 ㅁ. USR1이 “GRANT SELECT ON V TO USR2”를 실행

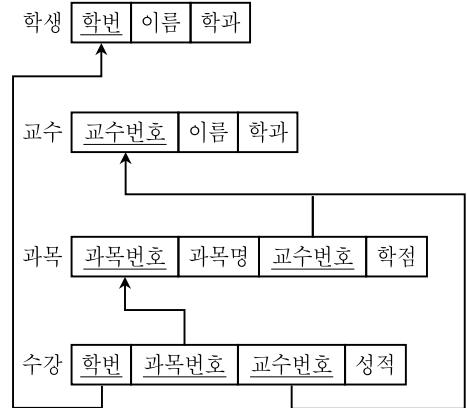
- ① ㄴ, ㅁ
 ② ㄱ, ㄷ, ㄹ
 ③ ㄱ, ㄷ, ㅁ
 ④ ㄴ, ㄷ, ㄹ

문 19. 다음 그림은 ARIES 회복 기법에 의해 시스템 고장 전에 마지막으로 기록된 DBMS 로그이다. 이 로그를 사용하여 수행하는 회복 과정에 대한 설명으로 옳지 않은 것은? (단, LSN은 로그 순차번호(Log Sequence Number)이며, 표현되지 않은 로그의 다른 속성은 고려하지 않는다)

LSN	최근 이전 LSN	트랜잭션 번호	연산 종류	페이지 번호
1	0	T ₁	갱신	P ₁
2	0	T ₂	갱신	P ₂
3	1	T ₁	완료	
4	begin checkpoint			
5	end checkpoint			
6	0	T ₃	갱신	P ₁

- ① UNDO 단계에서는 로그의 끝에서부터 반대 방향으로 완료되지 않은 트랜잭션에서 수행된 갱신 연산을 UNDO한다.
 ② UNDO 단계에서 완료되지 않은 트랜잭션의 복귀(rollback) 시 수행한 갱신연산에 해당하는 UNDO에 대해 보상 로그 레코드(compensation log record)를 기록하지 않는다.
 ③ REDO 단계에서 오손 페이지 테이블(dirty page table)을 사용하여 불필요한 REDO 연산의 수행을 줄여 회복시간을 단축할 수 있다.
 ④ 분석 단계에서 트랜잭션 T₂와 T₃은 처리 중(in progress)인 트랜잭션으로 분류된다.

문 20. 다음 그림은 학생, 교수, 과목, 수강 4개의 테이블 스키마와 그 인스턴스를 보여주고 있다. 이 데이터베이스에 SQL문을 수행하였을 때, 오류가 발생하지 않는 것은? (단, 밑줄은 기본키를 의미한다)



학생	교수
학번	교수번호
100	1000
200	2000
300	3000
400	4000
	5000
이름	이름
강감찬	손기훈
안중근	남승민
이순신	양정상
김구	장은상
	김연호
학과	학과
컴퓨터	컴퓨터
컴퓨터	컴퓨터
수학	수학
전자	전자
전자	경영

과목	수강
과목번호	학번
C001	100
C001	200
C002	200
M001	300
E001	400
과목명	과목번호
데이터베이스	C001
데이터베이스	C001
자료구조	C002
이산수학	M001
회로이론	E001
교수번호	교수번호
1000	1000
2000	2000
1000	1000
3000	3000
4000	4000
학점	성적
3	A+
3	B+
3	A
3	C+
3	B

- ① UPDATE 교수 SET 교수번호 = 2000
 WHERE 이름 = '김연호';
- ② CREATE VIEW 컴퓨터학과
 AS SELECT 이름, 학과
 FROM 학생
 WHERE 학과 = '컴퓨터';
 INSERT INTO 컴퓨터학과
 VALUES ('윤봉길', '컴퓨터');
- ③ CREATE VIEW 교수별수강학생(교수번호, 학번, 과목번호, 성적)
 AS SELECT 교수.교수번호, 학번, 과목번호, 성적
 FROM 교수, 수강
 WHERE 교수.교수번호 = 수강.교수번호;
 INSERT INTO 교수별수강학생(학번, 과목번호, 성적)
 VALUES ('300', 'C002', 'B');
- ④ CREATE VIEW 교수명단
 AS SELECT 교수번호, 이름
 FROM 교수;
 INSERT INTO 교수명단
 VALUES (6000, '진상오');