

데이터베이스론

1. 데이터 독립성에 대한 설명으로 옳은 것만을 모두 고르면?

- ㄱ. 물리적 데이터 독립성은 데이터의 물리적 구조가 변경될 때, 데이터의 논리적 구조도 이에 따라 자동으로 변경되게 한다.
ㄴ. 응용프로그램이나 데이터베이스의 논리적 구조에 독립적으로 데이터의 물리적 구조를 변경할 수 있다.
ㄷ. 논리적 데이터 독립성은 개념 스키마가 변경되어도 외부 스키마나 응용프로그램에 영향을 끼치지 않는 것을 의미한다.
ㄹ. 3단계 데이터베이스 구조의 내부 단계에서는 파일에 데이터를 효율적으로 저장하고 관리하기 위하여 여러 개의 내부 스키마를 가지고 있다.

- ① ㄱ, ㄴ
② ㄱ, ㄹ
③ ㄴ, ㄷ
④ ㄷ, ㄹ

2. 다음에서 설명하는 데이터베이스 용어는?

- 데이터베이스의 갱신이 발생할 때 DBMS가 자동으로 수행하는 사용자 정의 프로시저이다.
○ 무한 호출에 따라 실행의 종료를 보장할 수 없는 문제가 있다.
○ 무결성 제약조건을 유지하기 위하여 데이터베이스 갱신을 모니터링하고, 데이터베이스 갱신을 전파한다.

- ① assertion
② catalog
③ check
④ trigger

3. 릴레이션 $R(A, B, C, D, E, F)$ 에 대한 함수적 종속성(functional dependency)의 집합 F 가 다음과 같이 주어졌을 때, 기본 키는?

$$F = \{ A \rightarrow BC, \quad E \rightarrow CF, \quad B \rightarrow E, \quad D \rightarrow E \}$$

- ① A
② AB
③ AD
④ AE

4. 다음 릴레이션에서 키에 대한 설명으로 옳지 않은 것은? (단, 밑줄 친 속성은 기본 키이고, 고객 릴레이션에서 주민등록번호 속성의 값은 중복되지 않는다)

고객					
고객ID	고객이름	주민등록번호	등급	적립금	추천고객
K123	김탁구	96XXXX-XXXXXXX	Gold	2000	P492
L243	이승룡	97XXXX-XXXXXXX	Vip	3000	K123
C524	최무선	93XXXX-XXXXXXX	Gold	2000	K123
P492	박하얀	95XXXX-XXXXXXX	Silver	1000	L243

주문					
주문번호	주문고객	주문제품	수량	단가	주문일자
1	K123	우동	10	2000	2023-01-01
2	C524	라면	5	1000	2023-01-02
3	P492	만두	7	3000	2023-01-03

- ① 고객 릴레이션에서 고객ID, 주민등록번호, (고객ID, 고객이름)의 쌍, (고객이름, 주민등록번호)의 쌍은 슈퍼 키가 될 수 있다.
② 고객 릴레이션에서 추천고객 속성은 자신의 고객ID 속성을 참조하는 외래 키로 정의가 가능하며, NULL 값을 가질 수 있다.
③ 주문 릴레이션의 주문고객 속성을 고객 릴레이션의 고객ID에 대한 외래 키로 정의할 때, 주문고객의 값으로 'S254'를 가질 수 있다.
④ 고객 릴레이션에서 고객ID, 주민등록번호는 후보 키가 될 수 있으나, (고객ID, 고객이름)의 쌍은 후보 키가 될 수 없다.

5. 다음 릴레이션 R 과 S 에 대한 관계대수식의 결과로 옳지 않은 것은? (단, 기호 $\bowtie_N$ 은 자연조인을 의미하고, 릴레이션에서 A, B, C, D는 속성 이름이다)

R			S		
A	B	C	B	C	D
a1	b1	c1	b1	c1	d1
a2	b2	c2	b3	c2	d2
a3	b3	c3	b3	c3	d3
a4	b4	c5	b5	c5	d5

① $\sigma_{B='b3'}(S)$

② $\pi_{B,C}(R \bowtie_N S)$

B	C	D
b3	c2	d2
b3	c3	d3

B	C
b1	c1
b3	c3

③ $\pi_{A,B}(\sigma_{C='c3'}(R))$

④ $\pi_{A,C}(\sigma_{B='b3'}(R \bowtie_N S))$

A	B
a3	b3

A	C
a2	c2

6. 크기가 100바이트인 고정길이 레코드 630개로 구성된 정렬된 순차 파일(sequential file)에서 이진검색으로 레코드를 검색할 때, 최대 접근하는 블록의 수는? (단, 블록의 크기는 1,024바이트이며, 하나의 레코드가 두 개 이상의 블록에 나누어 저장되지 않는다고 가정한다)

- ① 6
② 10
③ 64
④ 320

7. 다음과 같은 조건을 모두 만족하는 B^+ -트리에 T, W, M, E, F, G, K, C, S, D, B, L 순으로 키 값이 입력되었을 때, 루트(root)에 있는 키 값을 바르게 나열한 것은? (단, $[X_i]$ 는 포인터 P_i 가 가리키는 서브트리의 모든 키 값들을 의미하고, 트리는 비어 있는(empty) 상태이며, 기호 $\equiv$ 는 동치를 의미한다)

- 한 노드에 최대 네 개의 키 값을 가질 수 있다.
- 내부노드의 구조는 $\langle P_1, K_1, \dots, P_i, K_i, \dots, K_{q-1}, P_q \rangle$ 이며, K_i 는 i 번째 키, P_i 는 i 번째 서브트리를 가리키는 포인터이다.
- $i \equiv 1$ 이면 $[X_1] < K_1$ 이고, $i \equiv q$ 이면 $K_{q-1} \leq [X_q]$ 를 만족한다.
- $1 < i < q$ 이면 $K_{i-1} \leq [X_i] < K_i$ 를 만족한다.
- 노드 분할 시 상위 노드로 올리는 값은 분할대상 키 값들 중 중간 값을 올리는 것으로 가정한다.

- ① C, K
② D, L
③ F, M
④ G, S

8. 데이터베이스 관리시스템(DBMS)에 대한 설명으로 옳은 것만을 모두 고르면?

- ㄱ. 일반적으로 데이터베이스는 저장된 데이터가 많아 구조가 복잡하고, 동시 공유를 지원하므로 장애 발생에 대응해서 데이터를 일관된 상태로 회복하는 과정이 복잡하다.
- ㄴ. DBMS의 필수 기능으로 정의 기능, 조작 기능, 제어 기능, 처리 기능이 있다.
- ㄷ. DBMS의 정의 기능은 데이터를 정확하고 안전하게 유지하기 위한 데이터 무결성 유지, 보안 유지, 병행 제어 기능을 포함한다.
- ㄹ. DBMS는 데이터의 중복성과 응용프로그램에 대한 종속성 문제 해결에서 파일 시스템 대비 우수한 장점을 가지고 있다.

- ① ㄱ, ㄴ
② ㄱ, ㄷ
③ ㄴ, ㄷ
④ ㄷ, ㄹ

9. 음식점 관리 시스템에서 고객(customer), 주문(order), 음식(food)을 관리하기 위해 릴레이션을 다음과 같이 설계하였다. 이 릴레이션에 대하여, SQL문에 대응하는 관계대수식으로 옳지 않은 것은? (단, 밑줄 친 속성은 기본 키이고, 관계대수 연산자 중에서 ρ 는 재명명(rename) 연산자이며, $\mathfrak{J}$ 은 집계함수(aggregate function) 연산을 정의하는 연산자이다)

```
customer(name, gender, phone, address, year)
order(cname, fname, price, date)
food(name, price, stock)
```

① SQL:

```
SELECT name
FROM food AS f
WHERE f.stock < 10 OR f.price > 20000;
관계대수:
 $\pi_{name}(\sigma_{f.stock < 10 \text{ OR } f.price > 20000}(\rho_f(food)))$ 
```

② SQL:

```
SELECT c.name, c.phone
FROM customer AS c, order AS o, food AS f
WHERE o.fname = f.name AND c.name = o.cname
AND f.price > 10000;
관계대수:
```

$$\pi_{c.name, c.phone} \left(\sigma_{\substack{o.fname = f.name \\ AND \ c.name = o.cname}} \left(\rho_c(customer) \times \rho_o(order) \times \rho_f(food) \right) \right)$$

③ SQL:

```
SELECT name, SUM(price) AS orderPrice
FROM customer, order
WHERE name = cname
GROUP BY name;
관계대수:
```

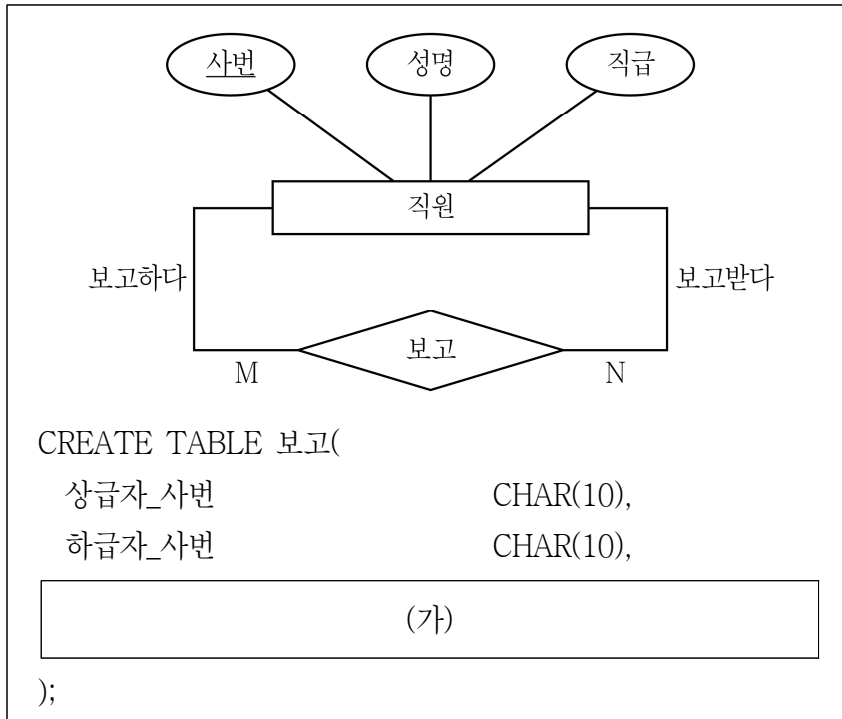
$$\rho_{(name, orderPrice)}(name \mathfrak{J}_{SUM(price)}(customer \bowtie_{name=cname} order))$$

④ SQL:

```
SELECT name, AVG(food.price) AS avgPrice
FROM food, order
GROUP BY name;
관계대수:
```

$$\rho_{(name, avgPrice)}(name \mathfrak{J}_{AVG(food.price)}(food \times order))$$

10. 다음 ERD(entity-relationship diagram)에 대응하는 보고 테이블을 생성하고자 한다. 보고 테이블을 생성하는 SQL문의 (가)에 들어갈 내용으로 옳은 것은? (단, 밑줄 친 속성은 기본 키이고, 직원 개체(entity)는 테이블로 생성되어 있다고 가정한다)



- ① PRIMARY KEY (상급자_사번, 하급자_사번),
FOREIGN KEY (상급자_사번) REFERENCES 직원(사번),
FOREIGN KEY (하급자_사번) REFERENCES 직원(사번)
- ② PRIMARY KEY (상급자_사번),
FOREIGN KEY (상급자_사번) REFERENCES 직원(사번),
FOREIGN KEY (하급자_사번) REFERENCES 직원(사번)
- ③ PRIMARY KEY (상급자_사번, 하급자_사번),
FOREIGN KEY (상급자_사번, 하급자_사번)
REFERENCES 직원(사번)
- ④ PRIMARY KEY (하급자_사번),
FOREIGN KEY (하급자_사번) REFERENCES 직원(사번)
11. 다음은 사원 정보를 나타내기 위한 EMP 테이블이다. EMP 테이블에서 상사가 없는 사원의 SUPERID가 NULL 처리될 때, 아래 SQL문 실행 결과의 차수(degree)와 카디널리티(cardinality)는? (단, 밑줄 친 속성은 기본 키이다)

EMP			
<u>ID</u>	NAME	SALARY	SUPERID
1001	Kim	8000	NULL
1002	Lee	5000	1003
1003	Park	7000	NULL
1004	Cho	3000	1001

```

SELECT *
FROM EMP S RIGHT OUTER JOIN EMP E
ON S.ID = E.SUPERID;

```

차수 카디널리티

- ① 4 2
- ② 4 8
- ③ 8 2
- ④ 8 4

12. 다음 고객 테이블에 대한 SQL 구문에서 실행 결과가 다른 하나는? (단, 밑줄 친 속성은 기본 키이다)

고객			
<u>고객ID</u>	고객이름	등급	적립금
k123	김상중	Gold	3000
l018	이영우	Vip	4000
c389	최영	Gold	3000
p273	박구연	Silver	2000
h328	홍연수	Vip	5000
j451	정영상	Silver	3000
y524	윤성진	Gold	2000

- ① SELECT 고객이름
FROM 고객
WHERE 적립금 = (SELECT MAX(적립금) FROM 고객);
- ② SELECT 고객이름
FROM 고객
WHERE 적립금 >= ALL (SELECT 적립금 FROM 고객);
- ③ SELECT 고객이름
FROM 고객
WHERE 적립금 > ANY (SELECT 적립금 FROM 고객
WHERE 등급 = 'Vip');
- ④ SELECT 고객이름
FROM 고객
WHERE 적립금 IN (SELECT 적립금 FROM 고객
WHERE 등급 = 'Vip');
13. 데이터베이스 시스템의 데이터 언어에 대한 설명으로 옳지 않은 것은?
- ① 데이터 언어는 사용자가 데이터베이스를 구축하고, 이에 접근하기 위해 데이터베이스 관리시스템과 통신하는 수단이다.
- ② 비절차적 데이터 조작어는 사용자가 어떤 데이터를 원하고 해당 데이터를 얻으려면 어떻게 처리해야 하는지에 대한 절차를 구체적으로 명세한다.
- ③ 데이터 제어어는 데이터베이스에 저장된 데이터를 여러 사용자가 무결성과 일관성을 유지하며 문제없이 공유할 수 있도록, 내부적으로 필요한 규칙이나 기법을 정의하는 데 사용하는 데이터 언어이다.
- ④ 데이터 정의어는 새로운 데이터베이스를 구축하기 위해 스키마를 정의하거나 기존 스키마의 정의를 삭제하거나 수정하기 위해 사용하는 데이터 언어이다.

14. 다음은 분산 데이터베이스에 대한 설명이다. (가), (나)에 들어갈 내용을 바르게 연결한 것은?

- (가)는 각 조각 스키마의 인스턴스가 물리적으로 저장되어야 하는 지역을 정의한다.
- (나)는 분산 데이터베이스에 저장할 모든 데이터 구조와 제약조건을 정의하는 것으로서, 관계 데이터 모델의 관점에서 보면 데이터베이스 안에 존재하는 모든 릴레이션 스키마의 집합이다.

(가)

(나)

- | | |
|-----------|-----------|
| ① 지역 스키마 | 전역 개념 스키마 |
| ② 단편화 스키마 | 지역 스키마 |
| ③ 단편화 스키마 | 할당 스키마 |
| ④ 할당 스키마 | 전역 개념 스키마 |

15. NoSQL에 대한 설명으로 옳지 않은 것은?

- ① NoSQL 시스템으로 Ingres 등이 있다.
- ② 가용성을 높이기 위해 일치성을 약하게 보장하는 궁극적 일치성(eventual consistency)을 제공하는 경우가 많다.
- ③ 미리 정의된 스키마를 사용하지 않아도 된다.
- ④ 다수의 컴퓨터에 데이터를 분산, 저장, 처리하는 것이 가능한 데이터베이스 시스템이다.

16. 다음 두 종류의 로그 및 데이터베이스 기록 내용에서 (가), (나), (다), (라) 시점에 장애가 발생하였을 때, 수행되어야 할 회복 연산이 바르게 연결된 것은? (단, A, C, E, G의 초깃값은 100이며, B, D, F, H의 초깃값은 50이고, (가) ~ (라)의 각 시점에 발생하는 장애는 상호 영향을 주지 않는다)

시간	로그	데이터베이스	시간	로그	데이터베이스
	<T ₁ , start>			<T ₃ , start>	
	<T ₁ , A, 100, 90>			<T ₃ , E, 90>	
	<T ₁ , B, 50, 60>	A=90		<T ₃ , F, 60>	
		B=60	(가)→	<T ₃ , commit>	
	<T ₁ , commit>				E=90, F=60
				<T ₄ , start>	
	<T ₂ , start>			<T ₄ , G, 90>	
	<T ₂ , C, 100, 90>			<T ₄ , H, 60>	
	<T ₂ , D, 50, 60>	C=90	(다)→	<T ₄ , commit>	
		D=60			G=90, H=60
(나)→	<T ₂ , commit>				

(가)

(나)

(다)

(라)

- | | | | |
|-------------------------|--|-----------------------|--|
| ① 회복작업 없음 | Redo(T ₁) | Undo(T ₃) | Redo(T ₃), Undo(T ₄) |
| ② 회복작업 없음 | Redo(T ₁) | 회복작업 없음 | Redo(T ₃) |
| ③ Undo(T ₁) | Redo(T ₁), Undo(T ₂) | 회복작업 없음 | Redo(T ₃) |
| ④ Undo(T ₁) | Redo(T ₁), Undo(T ₂) | Undo(T ₃) | Redo(T ₃), Undo(T ₄) |

17. 관계형 데이터베이스의 상태에 대한 제약조건으로 옳지 않은 것은?

- ① 참조되는 릴레이션에 새로운 튜플이 삽입되는 과정에서 참조무결성 제약조건 위반이 발생하면, 제약조건 위반의 원인이 되는 연산을 거부하도록 해야 한다.
- ② 릴레이션 스키마 R의 임의의 두 튜플 t_1, t_2 에 대하여 $t_1[SK] \neq t_2[SK]$ 이 성립한다면, 속성 집합 SK를 R의 슈퍼키라고 한다.
- ③ 참조하는 릴레이션 R1의 외래 키는 참조되는 릴레이션 R2의 기본 키와 동일한 도메인이어야 하며, 기본 키 값 중 하나이거나 NULL 값이어야 한다.
- ④ 데이터베이스는 개체무결성과 참조무결성 제약조건을 만족해야 한다.

18. SQL의 부속질의(subquery)에 대한 설명으로 옳지 않은 것은?

- ① 부속질의는 하나의 SQL문 안에 다른 SQL문이 중첩된 질의를 말한다.
- ② 연산자 ANY는 부속질의가 단일 행을 반환할 때만 적용할 수 있다.
- ③ 스칼라(scalar) 부속질의는 결과 값을 단일한 스칼라 값으로 반환한다.
- ④ 인라인 뷰(inline view)는 FROM 절에서 사용되는 부속질의이다.

19. 다음 릴레이션 R_1, R_2, R_3 에 대한 관계대수식의 의미가 다른 것은?

$$R_1(A, B, C) \quad R_2(D, E) \quad R_3(F, G, H)$$

- ① $\sigma_{B=5}(R_1 \bowtie_{A=D} R_2) \bowtie_{E=F} R_3$
- ② $\sigma_{B=5 \text{ AND } A=D}(\pi_{A,B}(R_1) \times R_2) \bowtie_{E=F} R_3$
- ③ $\sigma_{A=D}(\sigma_{B=5}(R_1) \times R_2) \bowtie_{E=F} R_3$
- ④ $\sigma_{B=5}((R_2 \bowtie_{E=F} R_3) \bowtie_{A=D} R_1)$

20. 두 개의 트랜잭션 T1과 T2를 수행하는 과정에서 다음과 같은 현상이 발생하였다. 이 현상이 발생하는 격리 수준은?

- T1에서 1번 상품의 가격을 조회하였을 때 10만 원으로 나타났고, T2에서 1번 상품의 가격을 15만 원으로 변경하고 커밋(commit)을 수행하였다. 그 후에 T1에서 1번 상품의 가격을 조회하니 15만 원으로 나타났다.
- T2에서 1번 상품의 가격을 10만 원에서 15만 원으로 변경하다가 커밋을 수행하지 않고 롤백(rollback)하는 과정에서 T1에서 1번 상품의 가격을 조회하였으나 허용되지 않았다.
- T1에서 등록된 상품을 조회하였는데 0건이 조회되었고, T2에서 1번 상품을 등록하고 커밋을 수행하였다. T1에서 등록된 상품을 다시 조회하였더니 1개 상품이 나타났다.

- ① Read Committed
- ② Serializable
- ③ Repeatable Read
- ④ Read Uncommitted

21. 릴레이션 R과 S에 대한 조인 연산의 구현 방법을 설명한 것으로 옳지 않은 것은?

- ① 중첩 루프 조인(nested loop join): R과 S의 레코드들에 대한 별도의 인덱스나 정렬을 필요로 하지 않는다.
- ② 단일 루프 조인(single loop join): R과 S 중 하나의 릴레이션에 있는 레코드들은 조인 속성 값에 따라 물리적으로 정렬되어 있어야 한다.
- ③ 정렬-합병 조인(sort-merge join): R과 S의 레코드들이 조인 속성 값에 따라 물리적으로 정렬되어 있을 때 가장 효율적이다.
- ④ 해시 조인(hash join): R과 S의 레코드들은 조인 속성을 해시 키로 하고 동일한 해시 함수를 이용해 해시되어 있어야 한다.

22. 회복 기법과 그 특징이 바르게 짝지어진 것은?

- ① 지연 갱신에 기반을 둔 기법: UNDO/REDO
- ② 즉시 갱신에 기반을 둔 기법: NO-UNDO/NO-REDO
- ③ 그림자 페이지 기법: UNDO/NO-REDO
- ④ ARIES 기법: 분석 → REDO → UNDO

23. 데이터베이스 보안에 대한 설명으로 옳지 않은 것은?

- ① 권한 관리를 이용하여 접근 권한이 없는 사용자의 접근으로부터 데이터베이스를 보호한다.
- ② 사용자가 소유한 데이터베이스 객체에 대한 권한 부여는 반드시 데이터 관리자에 의해 수행되어야 한다.
- ③ 접근이 허락된 사용자가 실수 등의 이유로 데이터 무결성을 위반하지 않도록 정의된 제약조건 등을 이용하여 통제할 수 있어야 한다.
- ④ 보안을 위한 데이터 단위는 데이터베이스 전체로부터 특정 테이블의 특정 행과 열 수준까지 정의될 수 있다.

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

- ① 데이터 웨어하우스는 의사결정에 필요한 주제에 따라 데이터를 관리하는 주제 지향성(subject orientation) 특징을 가진다.
- ② 데이터 웨어하우스는 데이터가 일관된 형식으로 정의되는 통합성(integration) 특징을 가진다.
- ③ 데이터 웨어하우스는 데이터에 접근하는 현재 시간을 기준으로 최신의 값을 유지하는 시불변성(time invariance) 특징을 가진다.
- ④ 데이터 웨어하우스는 읽기 전용의 데이터를 유지하는 비휘발성(nonvolatilization) 특징을 가진다.

25. 데이터베이스에서 동시성 제어(concurrency control)를 위한 로킹(locking) 기법과 이를 이용하여 제어할 수 있는 상황이 바르게 짝지어진 것은?

- ㄱ. 직렬가능(serializable) 스케줄 보장
- ㄴ. 교착상태(deadlock) 방지
- ㄷ. 복구가능한(recoverable) 스케줄 보장

- ① 2PL(two-phase locking) - ㄷ
- ② 엄격한(strict) 2PL - ㄱ, ㄷ
- ③ 엄중한(rigorous) 2PL - ㄱ, ㄴ
- ④ 엄중한(rigorous) 2PL - ㄱ, ㄴ, ㄷ