

|                |                 |                     |
|----------------|-----------------|---------------------|
| <b>데이터베이스론</b> | <b>국가 전산 7급</b> | <b>2023년 9월 23일</b> |
|----------------|-----------------|---------------------|

◆ 필기합격인원/합격선(52명/79점) - 선발예정인원 38명 ◆

1. 데이터 독립성에 대한 설명으로 옳은 것만을 모두 고르면? [2023년 국가 7급]

- ㄱ. 물리적 데이터 독립성은 데이터의 물리적구조가 변경될 때, 데이터의 논리적구조도 이에 따라 자동으로 변경되게 한다.
- ㄴ. 응용프로그램이나 데이터베이스의 논리적구조에 독립적으로 데이터의 물리적구조를 변경할 수 있다.
- ㄷ. 논리적 데이터 독립성은 개념스키마가 변경되어도 외부스키마나 응용프로그램에 영향을 끼치지 않는 것을 의미한다.
- ㄹ. 3단계 데이터베이스 구조의 내부단계에서는 파일에 데이터를 효율적으로 저장하고 관리하기 위하여 여러 개의 내부스키마를 가지고 있다.

- ① ㄱ, ㄴ
- ② ㄱ, ㄷ
- ③ ㄴ, ㄷ
- ④ ㄷ, ㄹ

☞ 데이터 독립성

- ㄱ. 물리적 데이터 독립성은 데이터의 물리적구조가 변경될 때, 데이터의 논리적구조도 이에 따라 자동으로 변경되게 한다.(x)
    - 물리적 데이터 독립성은 **기존 응용프로그램**에 영향을 주지 않고 데이터베이스의 물리적구조를 수정할 수 있는 능력이다.
    - 물리적 데이터 독립성 : **내부스키마**가 변경되어도 **개념스키마/외부스키마**에 영향주지 않는다.
    - 물리적구조 변경이 논리적구조에 영향을 주어서는 안 된다는 것이다.
  - ㄹ. 3단계 데이터베이스 구조의 **내부단계**에서는 파일에 데이터를 효율적으로 저장하고 관리하기 위하여 **여러 개의 내부스키마**를 가지고 있다.(x)
    - 내부단계는 **하나의 내부스키마**를 정의한다.
- ① 내부스키마는 기억장치(디스크) 관점에서 데이터베이스를 정의한 것이다.
  - ② 내부스키마는 '개념스키마가 어떻게 저장될 것인가?'를 기술한 것이다.
  - ③ 레코드 형식, 각 객체의 바이트 수, 인덱스 유무, 레코드의 물리적인 순서 등을 기술한다.
    - 내부스키마 아래에 '물리적 단계'가 존재한다.
  - ④ 내부스키마는 **시스템 프로그래머나 시스템 설계자**가 보는 관점의 스키마이다.

정답 : ③

2. 다음에서 설명하는 데이터베이스 용어는? [2023년 국가 7급]

- 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

♣ trigger

- 트리거는 데이터베이스에 어떠한 일이 일어나면 **자동으로 실행**되는 개체이다.
- 트리거는 데이터베이스 갱신이 발생할 때 DBMS가 자동으로 수행하는 사용자 프로시저이다.
- SQL 서버나 데이터베이스에 어떤 **이벤트**가 발생했을 때, 정의된 **트리거**가 **수행**될 수 있다.
- 트리거는 제약조건과 더불어 데이터 **무결성**을 위해서 사용할 수 있는 기능이다.
- 트리거는 잘못된 갱신을 예방할 수 있는 특수 제약조건으로 사용될 수 있다.
- 트리거는 사건(삽입, 삭제, 갱신) 발생 **후**가 아닌 사건 발생 **전**에도 활성화 될 수 있다.

사원

| 사번 | 이름  | 소속 | 상사   | 급여  |
|----|-----|----|------|-----|
| 1  | 홍길동 | 10 | NULL | 500 |
| 2  | 이몽룡 | 20 | 100  | 400 |
| 3  | 성춘향 | 20 | 200  | 300 |
| 4  | 박대감 | 30 | 100  | 350 |

```

Create Trigger 급여_인상                                     //트리거 생성(트리거명 : 급여_인상)
After Update Of 급여 On 사원                               //사원 테이블에서 급여 필드를 갱신하면 실행
Referencing Old Row As O, New Row As N                   //새로운 값은 N, 이전 값은 O
For Each Row
When (N.급여 > O.급여 × 1.1)                               //새로운 급여가 기존 급여보다 110% 초과하면
    Update 사원 Set 급여 = O.급여 × 1.1                 //급여를 강제로 110% 인상 갱신하라.
    Where 사원.사번 = O.사번;
    
```

- 사원 급여를 최대 10%까지만 인상하는데 잘못 처리해서 인상률이 10% 초과하게 되면
- 급여\_인상 트리거가 자동으로 실행되어 급여를 최대 10%까지만 인상시킨다.

3. 릴레이션 R(A, B, C, D, E, F)에 대한 함수적 종속성(functional dependency)의 집합 F가 다음과 같이 주어졌을 때, 기본키는? [2023년 국가 7급]

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

- ① A                      ② AB                      ③ AD                      ④ AE

☞ 기본키 찾기

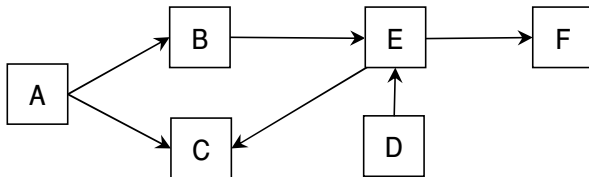
// 릴레이션의 기본키

- 기본키는 유일성과 최소성을 만족하는 속성이다.
- 기본키는 다른 모든 속성을 결정하게 된다.
- 해서, 릴레이션 R의 모든 속성과 함수종속성을 가지는 관계를 찾으면 기본키이다.
- 릴레이션의 기본키는 속성의 **폐포** 또는 **추론규칙**을 이용해서 찾을 수 있다.

|                  |   |
|------------------|---|
| 추론규칙을 이용한 기본키 찾기 | <ul style="list-style-type: none"> <li>· 추론규칙을 이용한 기본키 찾기는 6가지 규칙을 적절하게 적용해야 한다.</li> <li>· 실수할 확률이 높다.</li> </ul>  |
| 폐포를 이용한 기본키 찾기   | <ul style="list-style-type: none"> <li>· 폐포를 이용한 기본키 찾기는 함수종속 관계를 그림으로 그려서 찾는다.</li> <li>· 그림을 보고 함수종속 관계를 나타내는 화살표를 추적하면 된다.</li> <li>· 그림을 보고 기본키 찾는 것도 어느 정도 정리해야 한다.</li> <li>· 아무튼, 폐포를 이용한 기본키 찾기를 권장한다.</li> <li>· 당연, 기본키를 찾기 위해서는 폐포가 무엇인지? 알아야 한다.</li> <li>· 함수종속집합에서 추론될 수 있는 모든 함수종속집합을 폐포라 한다.</li> </ul> |

· 기본키 찾기와 후보키 찾는 것은 같은 개념이다.

// 그림으로 릴레이션의 기본키 찾기 - 폐포를 이용한 기본키 찾기



- A의 폐포 =  $A^+ = \{A, B, C, E, F\}$  → A는 기본키 불가(모든 속성을 가지지 않으므로)
- B의 폐포 =  $B^+ = \{B, C, E, F\}$  → B는 기본키 불가(모든 속성을 가지지 않으므로)
- {A, B}의 폐포 =  $\{A, B\}^+ = \{A, B, C, E, F\}$  → 기본키 불가(모든 속성을 가지지 않으므로)
- {A, E}의 폐포 =  $\{A, E\}^+ = \{A, B, C, E, F\}$  → 기본키 불가(모든 속성을 가지지 않으므로)
- {A, D}의 폐포 =  $\{A, D\}^+ = \{A, B, C, D, E, F\}$  → AD는 **기본키**(모든 속성을 가지므로)

4. 다음 릴레이션에서 키에 대한 설명으로 옳지 않은 것은? (단, 밑줄 친 속성은 기본키이고, 고객 릴레이션에서 주민등록번호 속성의 값은 중복되지 않는다) [2023년 국가 7급]

| 고객   |      |                   |        |      |      |
|------|------|-------------------|--------|------|------|
| 고객ID | 고객이름 | 주민등록번호            | 등급     | 적립금  | 추천고객 |
| K123 | 김탁구  | 96xxxx - xxxxxxxx | Gold   | 2000 | P492 |
| L243 | 이승룡  | 97xxxx - xxxxxxxx | Vip    | 3000 | K123 |
| C524 | 최무선  | 93xxxx - xxxxxxxx | Gold   | 2000 | K123 |
| P492 | 박하얀  | 95xxxx - xxxxxxxx | 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 슈퍼키가 될 수 있다.
- ② 고객 릴레이션에서 추천고객 속성은 자신의 고객ID 속성을 참조하는 외래키로 정의가 가능하며, NULL 값을 가질 수 있다.
- ③ 주문 릴레이션의 주문고객 속성을 고객 릴레이션의 고객ID에 대한 외래키로 정의할 때, 주문고객의 값으로 'S254'를 가질 수 있다.
- ④ 고객 릴레이션에서 고객ID, 주민등록번호는 후보키가 될 수 있으나, (고객ID, 고객이름)의 쌍은 후보키가 될 수 없다.

☞ 키

- 주문 릴레이션의 주문고객 속성을 고객 릴레이션의 고객ID에 대한 외래키로 정의할 때, 주문고객의 값으로 'S254'를 가질 수 있다.(x)
  - 고객 릴레이션의 고객ID에 'S254'라는 값이 없다.
  - 주문 릴레이션의 주문고객 속성은 'S254'라는 값을 가질 수 없다.

// 무결성 제약(integrity constraint)

|                                   |  |
|-----------------------------------|--|
| 참조 무결성<br>(referential integrity) | <ul style="list-style-type: none"> <li>• 외래키는 널 또는 참조 릴레이션의 기본키가 가지는 값이어야 한다.</li> <li>• 이를 다시 설명하면 '릴레이션은 대응이 되지 않는 외래키 값을 가질 수 없다'는 제약이다.</li> <li>• 외래키와 참조 릴레이션의 기본키 사이의 관계가 정확하게 유지됨을 보장</li> </ul> |
|-----------------------------------|--|

5. 다음 릴레이션 R과 S에 대한 관계대수식의 결과로 옳지 않은 것은? (단, 기호  $\bowtie_N$ 은 자연조인을 의미하고, 릴레이션에서 A, B, C, D는 속성 이름이다) [2023년 국가 7급]

| R  |    |    |
|----|----|----|
| A  | B  | C  |
| a1 | b1 | c1 |
| a2 | b2 | c2 |
| a3 | b3 | c3 |
| a4 | b4 | c5 |

| S  |    |    |
|----|----|----|
| B  | C  | D  |
| b1 | c1 | d1 |
| b3 | c2 | d2 |
| b3 | c3 | d3 |
| b5 | c5 | d5 |

①  $\sigma_{B='b3'}(S)$

| B  | C  | D  |
|----|----|----|
| b3 | c2 | d2 |
| b3 | c3 | d3 |

②  $\pi_{B,C}(R \bowtie_N S)$

| B  | C  |
|----|----|
| b1 | c1 |
| b3 | c3 |

③  $\pi_{A,B}(\sigma_{C='c3'}(R))$

| A  | B  |
|----|----|
| a3 | b3 |

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

| A  | C  |
|----|----|
| a2 | c2 |

☞ 자연조인

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

↓  $R \bowtie_N S$

↓ Select \* From R Natural Join S;

| A  | B  | C  | D  |
|----|----|----|----|
| a1 | b1 | c1 | d1 |
| a3 | b3 | c3 | d3 |

↓  $\sigma_{B='b3'}(R \bowtie_N S)$

↓ Select \* From R Natural Join S Where B = 'b3';

| A  | B  | C  | D  |
|----|----|----|----|
| a3 | b3 | c3 | d3 |

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

↓ Select A, C From R Natural Join S Where B = 'b3';

| A  | C  |
|----|----|
| a3 | c3 |

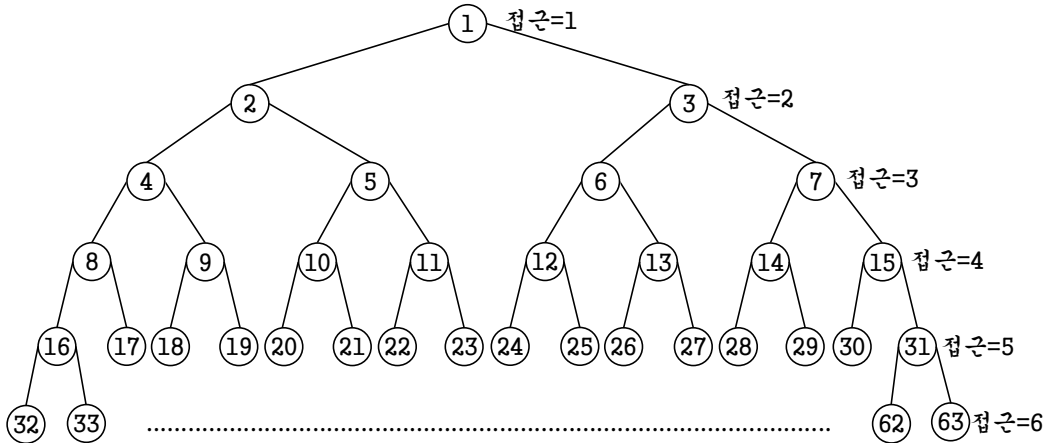
6. 크기가 100바이트인 고정길이 레코드 630개로 구성된 정렬된 순차파일(sequential file)에서 이진검색으로 레코드를 검색할 때, 최대로 접근하는 블록의 수는? (단, 블록의 크기는 1,024바이트이며, 하나의 레코드가 두 개 이상의 블록에 나누어 저장되지 않는다고 가정한다) [2023년 국가 7급]

- ① 6                                      ② 10
- ③ 64                                     ④ 320

☞ 이진검색으로 레코드를 검색

- 하나의 블록에 저장될 수 있는 레코드 수 =  $1,024 / 100 = 10.24 = 10$ (개)
- 블록 수 =  $630 / 10 = 63$  ← 63개의 블록에 대해 이진검색으로 레코드를 검색하는 것

↓  
↓ 63개의 블록에 대한 이진검색 과정을 트리로 그리면  
↓



- 하드디스크는 블록 단위로 읽고 쓰기를 한다.
- 최대로 접근하는 블록의 수 = 32(개) ← 트리에서 단말노드 수  
→ 주어진 의미를 "탐색을 위한 최대 접근횟수를 가지는 블록의 수"로 해석한 경우
- 레코드 탐색을 위한 블록 최대 접근횟수 = 6(번) ← 트리에서 단말노드에 있는 블록

// 이진탐색에 대해 공식으로 풀면 - 블록 수 63(개)

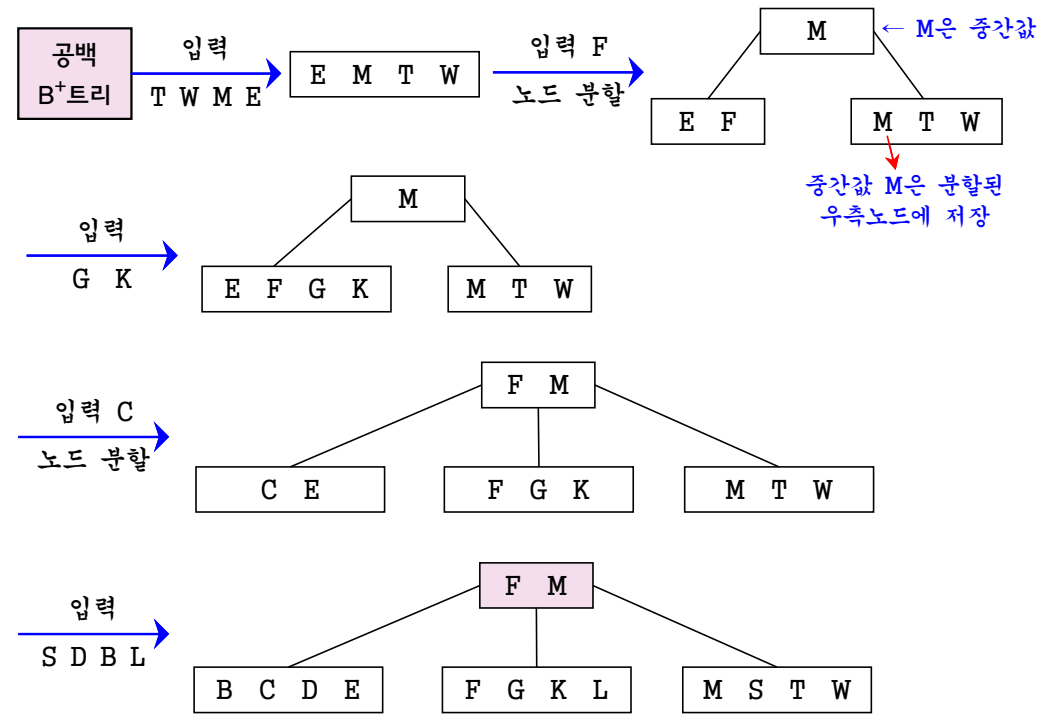
- $\lceil \log_2(n+1) \rceil = \lceil \log_2(63+1) \rceil = \lceil 6 \rceil = 6$ (번) ← 천장함수(올림함수) 사용
- 최악의 경우, 최대 6번 블록에 접근해야 원하는 record를 찾을 수 있다.

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 값을 의미하고, 트리는 비어 있는(empty) 상태이며, 기호  $\equiv$ 는 동치를 의미한다) [2023년 국가 7급]

- 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 상위노드로 올리는 값은 분할대상 키 값들 중 중간 값을 올리는 것으로 가정한다.

- ① C, K                      ② D, L
- ③ F, M                      ④ G, S

☞ 차수가 5인  $B^+$ -트리 : 한 노드에 최대 4개의 키 값을 가짐



정답 : ③

8. 데이터베이스 관리시스템(DBMS)에 대한 설명으로 옳은 것만을 모두 고르면? [2023년 국가 7급]

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

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

☞ 데이터베이스 관리시스템

- ㄴ. DBMS의 필수 기능으로 정의의 기능, 조작 기능, 제어 기능, 처리 기능이 있다.(x)  
→ DBMS의 필수 기능으로 정의의 기능, 조작 기능, 제어 기능이 있다.
- ㄷ. DBMS의 정의의 기능은 데이터를 정확하고 안전하게 유지하기 위한 데이터 무결성 유지, 보안 유지, 병행제어 기능을 포함한다.(x)  
→ 데이터 무결성 유지, 보안 유지, 병행제어 기능은 제어 기능이다.

// DBMS 기능

|                   |  |
|-------------------|--|
| 정의<br>기능<br>(DDL) | <ul style="list-style-type: none"> <li>• 데이터베이스 구조를 정의하는 기능이다.</li> <li>• 데이터베이스와 다양한 응용프로그램이 서로 인터페이스할 수 있는 방법을 제공한다.</li> <li>• 스키마, 도메인, 테이블, 뷰, 인덱스 등을 정의하거나 변경, 삭제할 수 있다.</li> <li>• 데이터의 논리적 구조와 물리적 구조 사이에 상호 변환이 가능하도록 두 구조 사이의 사상(mapping)을 명세해야 한다.</li> <li>• 데이터베이스 관리자나 설계자가 사용한다.</li> </ul> |
| 조작<br>기능<br>(DML) | <ul style="list-style-type: none"> <li>• 데이터베이스와 사용자 사이의 인터페이스를 위한 수단을 제공한다.</li> <li>• DBMS는 사용자 요구에 따라 데이터베이스를 접근하고, 처리할 수 있어야 한다.</li> <li>• DML은 사용하기 쉽고, 명확하고, 효율적이어야 한다.</li> <li>• Select(검색), Insert(삽입), Update(갱신), Delete(삭제) 연산이 있다.</li> </ul>  |
| 제어<br>기능<br>(DCL) | <ul style="list-style-type: none"> <li>• 데이터베이스를 정확하고, 안전하게 유지하는 기능이다.(무결성 유지)</li> <li>• 정확성과 안전성은 공유를 목적으로 하는 데이터베이스의 기본 기능이다.</li> <li>• 데이터 무결성, 보안, 권한, 회복, 병행제어 등을 정의하는 데 사용하는 언어</li> <li>• 데이터베이스 관리자가 데이터 관리를 목적으로 사용한다.</li> </ul>   |



9. 음식점 관리시스템에서 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 기본키이고, 관계대수 연산자 중에서 ρ는 재명명(rename) 연산자이며, ∫은 집계함수(aggregate function) 연산을 정의하는 연산자이다) [2023년 국가 7급]

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

|      |  |  |
|------|--|--|
| ①    | SQL  | SELECT name<br>FROM food AS f<br>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<br>FROM customer AS c, order AS o, food AS f<br>WHERE o.fname = f.name AND c.name = o.cname<br>AND f.price > 10000; |
| 관계대수 | $\pi_{c.name, c.phone} \left( \begin{matrix} \sigma_{o.fname = f.name \\ \text{AND } c.name = o.cname} \left( \begin{matrix} \rho_c(customer) \\ \times \rho_o(order) \\ \times \rho_f(food) \end{matrix} \right) \right)$ |  |
| ③    | SQL  | SELECT name, SUM(price) AS orderPrice<br>FROM customer, order<br>WHERE name = cname<br>GROUP BY name;                                      |
| 관계대수 | $\rho_{(name, orderPrice)}(name \int_{SUM(price)}(customer \bowtie_{name = cname} order))$   |  |
| ④    | SQL  | SELECT name, AVG(food.price) AS avgPrice<br>FROM food, order<br>GROUP BY name;   |
| 관계대수 | $\rho_{(name, avgPrice)}(name \int_{AVG(food.price)}(food \times order))$  |  |

☞ SQL문에 대응하는 관계대수식

- 먼저, 항목 ②는 SQL문과 관계대수식의 결과는 다르게 검색된다.
- SQL문에는 조건 f.price>10000이 있고, 관계대수식에는 조건 f.price>10000이 없다.

|                            |   |
|----------------------------|---|
| 집계함수 연산<br>$\gamma$        | <ul style="list-style-type: none"> <li>• 예 : 이름 <math>\gamma_{Avg(점수)}</math></li> <li style="margin-left: 20px;">↓</li> <li>• 속성 이름으로 그룹화하고(Group By 이름)</li> <li>• 집계함수 Avg(점수) 연산한다는 의미이다.</li> </ul>  |
| 재명명 연산<br>rename operation | <ul style="list-style-type: none"> <li>• 재명명은 테이블명이나 속성명을 변경하는 연산이다.</li> <li>• 수학 기호 <math>\rho</math>(rho)를 사용한다.</li> <li>• 예1 : <math>\rho_{(이름, 평균)} \leftarrow (\pi_{이름, 평균}) \leftarrow \text{Select 이름, Avg(점수) As 평균}</math></li> <li style="margin-left: 20px;">↓</li> <li>• 추출된 필드인 이름, Avg(점수)는 이름, 평균으로 바꾸어 출력</li> <li>• <math>\rho</math>(rho)를 사용하면 <math>\pi</math>에 필드 기술하는 것은 생략한다.</li> <li>• 예2 : <math>\rho_f(\text{food}) \leftarrow</math> 테이블명 food을 f로 변경</li> <li>• SQL문 : FROM food AS f</li> </ul> |

| R  |    |    |
|----|----|----|
| a  | b  | c  |
| a1 | b1 | c1 |
| a2 | b2 | c2 |

 $\rho_{S(d,e,f)}(R)$ 

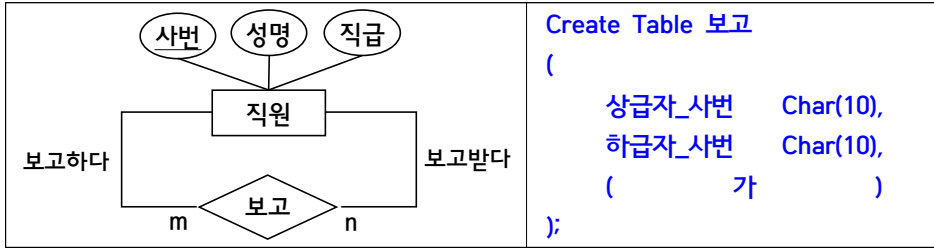
| S  |    |    |
|----|----|----|
| d  | e  | f  |
| a1 | b1 | c1 |
| a2 | b2 | c2 |

• 테이블명을 R에서 S로 변경하고, 속성 a, b, c를 d, e, f로 재명명 - 속성 (d, e, f)는 생략 가능

|      |  |
|------|--|
| SQL  | SELECT c.name, c.phone<br>FROM customer AS c, order AS o, food AS f<br>WHERE o.fname = f.name AND c.name = o.cname<br>AND f.price > 10000;   |
| 결과   | 조인 결과에서 o.fname = f.name AND c.name = o.cname이면서<br>f.price > 10000인 레코드의 c.name, c.phone이 검색된다.   |
| 관계대수 | $\pi_{c.name, c.phone} \left( \begin{matrix} \sigma_{o.fname = f.name} \\ AND\ c.name = o.cname} \left( \begin{matrix} \rho_c(customer) \\ \times \rho_o(order) \\ \times \rho_f(food) \end{matrix} \right) \right)$ |
| ② 결과 | 카티션곱(×) 결과에서 o.fname = f.name AND c.name = o.cname인<br>레코드의 c.name, c.phone이 검색된다. - SQL문 결과와 다름   |

- 카티션곱은 where절에 기술하는 조인(join) 조건이 없는 경우이다.
- 조건이 없으므로 각 테이블에 대한 모든 데이터를 조인하게 된다.
- 카티션곱은 where절에 조인조건이 잘못 기술되었을 때도 발생한다.
- 카티션곱은 조인에 참조되는 테이블 행수를 모두 곱한 값의 결과가 만들어진다.

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 기본키이고, 직원 개체(entity)는 테이블로 생성되어 있다고 가정한다) [2023년 국가 7급]



- 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)

☞ ERD - 관계형 데이터베이스 스키마 모습

| 직원 |     |    | 보고     |        |
|----|-----|----|--------|--------|
| 사번 | 성명  | 직급 | 상급자_사번 | 하급자_사번 |
| k1 | 홍재연 | 3  | k1     | k2     |
| k2 | 갑돌이 | 4  | k1     | k3     |
| k3 | 갑순이 | 3  | k1     | k4     |
| k4 | 심순애 | 4  | k2     | k1     |
| k5 | 이순신 | 4  | k2     | k3     |
| k6 | 이하늘 | 2  | k2     | k5     |

Create Table 보고(

```
상급자_사번 Char(10),
하급자_사번 Char(10),
Primary Key (상급자_사번, 하급자_사번),
Foreign Key (상급자_사번) References 직원(사번),
Foreign Key (하급자_사번) References 직원(사번)
);
```

↓ 상급자와 하급자 관계 분석

| 상급자 | 하급자           |
|-----|---------------|
| 홍재연 | 갑돌이, 갑순이, 심순애 |
| 갑돌이 | 홍재연, 갑순이, 이순신 |

- 직원들은 여러 명에게 보고하고, 여러 명에게 보고받는다.
- 다대다(m:n) 관계