

3과목-데이터베이스 구축

(Part 1. 논리 데이터베이스 설계 - Ⅱ)

논리 데이터베이스 설계-SEC_09(관계형 데이터베이스의 구조) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(관계형 데이터베이스의 구조)

1. 하나의 애트리뷰트가 가질 수 있는 원자 값들의 집합을 의미 하는 것은?

- ① 도메인 ② 튜플
- ③ 엔티티 ④ 다형성

2. 관계 데이터 모델에서 릴레이션(relation)에 관한 설명으로 옳은 것은?

- ① 릴레이션의 각 행을 스키마(schema)라 하며, 예로 도서 릴레이션을 구성하는 스키마에는 도서번호, 도서명, 저자, 가격 등이 있다.
- ② 릴레이션의 각 열을 튜플(tuple)이라 하며, 하나의 튜플은 각 속성에서 정의된 값을 이용하여 구성된다.
- ③ 도메인(domain)은 하나의 속성이 가질 수 있는 같은 타입의 모든 값의 집합으로 각 속성의 도메인은 원자 값을 갖는다.
- ④ 속성(attribute)은 한 개의 릴레이션의 논리적인 구조를 정의 한 것으로, 릴레이션의 이름과 릴레이션에 포함된 속성들의 집합 을 의미한다.

3. 다음 관계형 데이터 모델에 대한 설명으로 옳은 것은?

고객ID	고객이름	거주도시
S1	홍길동	서울
S2	이정재	인천
S3	신보라	인천
S4	김홍국	서울
S5	도요새	용인

- ① Relation 3개, Attribute 3개, Tuple 5개
- ② Relation 3개, Attribute 5개, Tuple 3개
- ③ Relation 1개, Attribute 5개, Tuple 3개
- ④ Relation 1개, Attribute 3개, Tuple 5개

4. 관계 데이터 모델에서 릴레이션(Relation)에 포함되어 있는 튜플 (Tuple)의 수를 무엇이라고 하는가?

- ① Degree ② Cardinality
- ③ Attribute ④ Cartesian product

논리 데이터베이스 설계-SEC_09(관계형 데이터베이스의 구조) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(관계형 데이터베이스의 구조)

5. 속성(Attribute)에 대한 설명으로 틀린 것은?

- ① 속성은 개체의 특성을 기술한다.
- ② 속성은 데이터베이스를 구성하는 가장 작은 논리적 단위이다.
- ③ 속성은 파일 구조상 데이터 항목 또는 데이터 필드에 해당된다.
- ④ 속성의 수를 "Cardinality"라고 한다.

6. A1, A2, A3 3개 속성을 갖는 한 릴레이션에서 A1의 도메인은 3개 값, A2의 도메인은 2개 값, A3의 도메인은 4개 값을 갖는다. 이 릴레이션에 존재할 수 있는 가능한 튜플(Tuple)의 최대 수는?

- ① 24 ② 12
- ③ 8 ④ 9

7. 관계형 데이터베이스의 구성 요소에 대한 설명으로 틀린 것은?

- ① 속성을 구성하는 값에는 동일한 값이 있을 수 있다.
- ② 한 릴레이션에 포함된 튜플은 모두 상이하다.
- ③ 한 릴레이션에는 동일한 이름의 속성이 있을 수 있다.
- ④ 한 릴레이션을 구성하는 속성 사이에는 순서가 없다.

8. 관계형 데이터 모델의 릴레이션에 대한 설명으로 틀린 것은?

- ① 모든 속성 값은 원자 값을 갖는다.
- ② 한 릴레이션에 포함된 튜플은 모두 상이하다.
- ③ 한 릴레이션에 포함된 튜플 사이에는 순서가 없다.
- ④ 한 릴레이션을 구성하는 속성 사이에는 순서가 존재한다.

논리 데이터베이스 설계-SEC_09(관계형 데이터베이스의 구조) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(관계형 데이터베이스의 구조)

9. 괄호 안 내용으로 옳게 짝지어진 것은?

In relational database terminology, a row is called
a(an) (㉠), a column is called a(an) (㉡), and
the (㉢) is called a relation.

- ① ㉠ tuple - ㉡ table - ㉢ attribute
- ② ㉠ table - ㉡ attribute - ㉢ tuple
- ③ ㉠ tuple - ㉡ attribute - ㉢ table
- ④ ㉠ attribute - ㉡ tuple - ㉢ table

10. 한 릴레이션 스키마가 4개 속성, 2개 후보키 그리고 그 스키마의
대응 릴레이션 인스턴스가 7개 튜플을 갖는다면 그 릴레이션의
차수(Degree)는?

- ① 1 ② 2
- ③ 4 ④ 7

논리 데이터베이스 설계-SEC_10(관계형 데이터베이스의 제약 조건 - 키(Key)) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(관계형 데이터베이스의 제약 조건-키(Key))

1. 다른 릴레이션의 기본키를 참조하는 키를 의미하는 것은?

- ① 필드키 ② 슈퍼키
- ③ 외래키 ④ 후보키

2. 다음 두 릴레이션에서 외래키로 사용된 것은? (단, 밑줄 친 속성은 기본키이다.)

과목(과목번호, 과목명)

수강(수강번호, 학번, 과목번호, 학기)

- ① 수강번호 ② 과목번호
- ③ 학번 ④ 과목명

3. 테이블의 기본키(Primary Key)로 지정된 속성에 관한 설명으로 가장 거리가 먼 것은?

- ① NOT NULL로 널 값을 가지지 않는다.
- ② 릴레이션에서 튜플을 구별할 수 있다.
- ③ 외래키로 참조될 수 있다.
- ④ 검색할 때 반드시 필요하다.

4. 키는 개체 집합에서 고유하게 개체를 식별할 수 있는 속성이다. 데이터베이스에서 사용되는 키의 종류에 대한 설명 중 옳지 않은 것은?

- ① 후보키는 개체들을 고유하게 식별할 수 있는 속성이다.
- ② 슈퍼키는 한 개 이상의 속성들의 집합으로 구성된 키이다.
- ③ 외래키는 다른 테이블의 기본키로 사용되는 속성이다.
- ④ 대체키는 슈퍼키 중에서 기본키를 제외한 나머지 키를 의미한다.

논리 데이터베이스 설계-SEC_10(관계형 데이터베이스의 제약 조건 - 키(Key)) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(관계형 데이터베이스의 제약 조건-키(Key))

5. 다음은 학생이라는 개체의 속성을 나타내고 있다. 여기서 '학과'를 기본키로 사용하기 곤란한 이유로 가장 타당한 것은?

학생(학과, 성명, 학번, 세부전공, 주소, 우편번호)

- ① 학과는 기억하기 어렵다.
- ② 학과는 정렬하는데 많은 시간이 소요된다.
- ③ 학과는 기억 공간을 많이 필요로 한다.
- ④ 동일한 학과 명을 가진 학생이 두 명 이상 존재할 수 있다.

6. 다음 설명의 ()안에 들어갈 내용으로 적합한 것은?

후보키는 릴레이션에 있는 모든 튜플에 대해 유일성과 ()
을 모두 만족시켜야 한다.

- ① 중복성 ② 최소성
- ③ 참조성 ④ 동일성

7. 릴레이션에 있는 모든 튜플에 대해 유일성은 만족시키지만 최소성은 만족시키지 못하는 키는?

- ① 후보키 ② 슈퍼키
- ③ 기본키 ④ 외래키

8. 다음은 관계형 데이터베이스의 키(Key)를 설명하고 있다. 해당 되는 키는?

한 릴레이션 내의 속성들의 집합으로 구성된 키로서, 릴레이션을 구성하는 모든 튜플에 대한 유일성은 만족시키지만 최소성은 만족시키지 못한다.

- ① 후보키 ② 대체키
- ③ 슈퍼키 ④ 외래키

논리 데이터베이스 설계-SEC_10(관계형 데이터베이스의 제약 조건 - 키(Key)) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(관계형 데이터베이스의 제약 조건-키(Key))

9. 다음 두 릴레이션 간의 관계에서 교수 릴레이션에 존재하는
외래키는? (단, 교수 릴레이션의 기본키는 교수번호이고 학과
릴레이션의 기본키는 학과번호이다)

교수(교수번호, 교수이름, 학과번호, 직급)

학과(학과번호, 학과이름, 학과장교수번호, 학생수)

- ① 교수이름 ② 학과번호
- ③ 학과장교수번호 ④ 학과이름

논리 데이터베이스 설계-SEC_11(관계형 데이터베이스의 제약 조건 - 무결성) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(관계형 데이터베이스의 제약 조건-무결성)

1. 데이터 무결성 제약 조건 중 "개체 무결성 제약" 조건에 대한 설명으로 맞는 것은?

- ① 릴레이션 내의 튜플들이 각 속성의 도메인에 지정된 값만을 가져야 한다.
- ② 기본키에 속해 있는 애트리뷰트는 널 값이나 중복 값을 가질 수 없다.
- ③ 릴레이션은 참조할 수 없는 외래키 값을 가질 수 없다.
- ④ 외래키 값은 참조 릴레이션의 기본키 값과 동일해야 한다.

2. 릴레이션 R1에 속한 애트리뷰트의 조합인 외래키를 변경하려면 이를 참조하고 있는 릴레이션 R2의 기본키도 변경해야 하는데 이를 무엇이라 하는가?

- ① 정보 무결성 ② 고유 무결성
- ③ 널 제약성 ④ 참조 무결성

3. 데이터 무결성과 가장 관계가 깊은 것은?

- ① 데이터의 안전성 ② 데이터의 공유성
- ③ 데이터의 중복성 ④ 데이터의 정확성

4. 데이터베이스 무결성에 관한 설명으로 옳지 않은 것은?

- ① 개체 무결성 규정은 한 릴레이션의 기본키를 구성하는 어떠한 속성 값도 널(NULL) 값이나 중복 값을 가질 수 없음을 규정하는 것이다.
- ② 무결성 규정에는 규정 이름, 검사 시기, 제약 조건 등을 명시한다.
- ③ 도메인 무결성 규정은 주어진 튜플의 값이 그 튜플이 정의된 도메인에 속한 값이어야 한다는 것을 규정하는 것이다.
- ④ 트리거는 트리거 조건이 만족되는 경우에 취해야 하는 조치를 명세한다.

논리 데이터베이스 설계- SEC_11(관계형 데이터베이스의 제약 조건 - 무결성) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(관계형 데이터베이스의 제약 조건-무결성)

5. 외래키(Foreign Key)와 가장 직접적으로 관련된 제약조건은 어느 것인가?

- ① 개체 무결성 ② 보안 무결성
- ③ 참조 무결성 ④ 정보 무결성

6. 다음 중 속성 값들은 사용자가 정의한 제약 조건을 만족해야 한다는 규정을 의미하는 것은?

- ① 도메인 무결성 ② 사용자 정의 무결성
- ③ 참조 무결성 ④ 개체 무결성

7. 다음 중 무결성 강화 방법과 가장 거리가 먼 것은?

- ① 제약 조건
- ② 애플리케이션
- ③ 데이터베이스 트리거
- ④ 인덱스

8. 데이터베이스의 무결성 규정(Integrity Rule)과 관련한 설명으로 틀린 것은?

- ① 무결성 규정에는 데이터가 만족해야 될 제약 조건, 규정을 참조할 때 사용하는 식별자 등의 요소가 포함될 수 있다.
- ② 무결성 규정의 대상으로는 도메인, 키, 종속성 등이 있다.
- ③ 정식으로 허가 받은 사용자가 아닌 불법적인 사용자에게 의한 갱신으로부터 데이터베이스를 보호하기 위한 규정이다.
- ④ 릴레이션 무결성 규정(Relation Integrity Rules)은 릴레이션을 조작하는 과정에서의 의미적 관계(Semantic Relationship)를 명세한 것이다.

논리 데이터베이스 설계- SEC_11(관계형 데이터베이스의 제약 조건 - 무결성) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(관계형 데이터베이스의 제약 조건-무결성)

9. 관계 데이터 모델의 무결성 제약 중 기본키 값의 속성 값이 널(Null) 값이 아닌 원자 값을 갖는 성질은?

- ① 개체 무결성
- ② 참조 무결성
- ③ 도메인 무결성
- ④ 튜플의 유일성

논리 데이터베이스 설계-SEC_12(관계대수 및 관계해석) 기출 문제

기출 문제(관계대수 및 관계해석)

1. 관계대수의 순수관계 연산자가 아닌 것은?

- ① Select ② Cartesian Product
- ③ Division ④ Project

2. 관계대수 연산에서 두 릴레이션이 공통으로 가지고 있는 속성을 이용하여 두 개의 릴레이션을 하나로 합쳐서 새로운 릴레이션을 만드는 연산은?

- ① \bowtie ② \supset
- ③ π ④ σ

3. 테이블에서 특정 속성에 해당하는 열을 선택하는 데 사용되며 결과로는 릴레이션의 수직적 부분집합에 해당하는 관계대수 연산자는?

- ① Project 연산자 ② Join 연산자
- ③ Division 연산자 ④ Select 연산자

4. 릴레이션 R의 차수가 4이고 카디널리티가 5이며, 릴레이션 S의 차수가 6이고 카디널리티가 7일 때, 두 개의 릴레이션을 카티션 프로덕트한 결과의 새로운 릴레이션의 차수와 카디널리티는 얼마인가?

- ① 24, 35 ② 24, 12
- ③ 10, 35 ④ 10, 12

논리 데이터베이스 설계-SEC_12(관계대수 및 관계해석) 기출 예상 문제

기출 문제(관계대수 및 관계해석)

5. 관계대수에 대한 설명으로 틀린 것은?

- ① 주어진 릴레이션 조작을 위한 연산의 집합이다.
- ② 일반 집합 연산과 순수관계 연산으로 구분된다.
- ③ 질의에 대한 해를 구하기 위해 수행해야 할 연산의 순서를 명시한다.
- ④ 원하는 정보와 그 정보를 어떻게 유도하는가를 기술하는 비절차적 방법이다.

6. 다음 R과 S 두 릴레이션에 대한 Division 연산의 수행 결과는?

R		
D1	D2	D3
a	1	A
b	1	A
c	2	A
d	2	B

S	
D2	D3
1	A

- ①

D3
A
B
- ②

D2
2
2
- ③

D3
A
- ④

D1
a
b

7. 관계대수 식을 SQL 질의로 옳게 표현한 것은?

$$\pi_{이름}(\sigma_{학과 = '교육'}(학생))$$

- ① SELECT 학생 FROM 이름 WHERE 학과 = '교육';
- ② SELECT 이름 FROM 학생 WHERE 학과 = '교육';
- ③ SELECT 교육 FROM 학과 WHERE 이름 = '학생';
- ④ SELECT 학과 FROM 학생 WHERE 이름 = '교육';

8. 관계해석(Relational Calculus)에 대한 설명으로 잘못된 것은?

- ① 튜플 관계해석과 도메인 관계해석이 있다.
- ② 원하는 정보와 그 정보를 어떻게 유도하는가를 기술하는 절차적인 특성을 가진다.
- ③ 기본적으로 관계해석과 관계대수는 관계 데이터베이스를 처리하는 기능과 능력 면에서 동등하다.
- ④ 수학의 Predicate Calculus에 기반을 두고 있다.

논리 데이터베이스 설계-SEC_12(관계대수 및 관계해석) 기출 문제

기출 문제(관계대수 및 관계해석)

9. 관계 대수와 관계 해석에 대한 설명으로 옳지 않는 것은?

- ① 관계대수는 원래 수학의 프레디킷 해석에 기반을 두고 있다.
- ② 관계대수로 표현한 식은 관계해석으로 표현할 수 있다.
- ③ 관계해석은 관계 데이터의 연산을 표현하는 방법이다.
- ④ 관계해석은 원하는 정보가 무엇이라는 것만 정의하는 비절차적인 특징을 가지고 있다.

프레디킷 해석이라고 말이 나오면 무조건 관계해석이다 란 것을 기억하자.

예) 평상문 : 철수는 남자이다. -> 술어적 표현 : $\text{man}(\text{철수})$

10. 관계해석에서 '모든 것에 대하여'의 의미를 나타내는 논리 기호는?

- ① \exists ② \in
- ③ \forall ④ \subset

\forall : 전칭 정량자 -> 가능한 모든 튜플에 대하여(For All)

\exists : 존재 정량자 -> 하나라도 일치하는 튜플이 있음(There Exists)

11. 조건을 만족하는 릴레이션의 수평적 부분집합으로 구성하며, 연산자의 기호는 그리스 문자 시그마를 사용하는 관계대수 연산은?

- ① Select
- ② Project
- ③ Join
- ④ Division

1. 논리 데이터베이스 설계-SEC_13(정규화(Normalization))

1) 정규화의 개요

; 정규화란 함수적 종속성 등의 종속성 이론을 이용하여 잘못 설계된 관계형 스키마를 더 작은 속성의 세트로 쪼개어 바람직한 스키마로 만들어 가는 과정이다.

- 하나의 종속성이 하나의 릴레이션에 표현될 수 있도록 분해해가는 과정이라 할 수 있다.
- 정규형에는 제1정규형, 제2정규형, 제3정규형, BCNF형, 제4정규형, 제5정규형이 있으며, 차수가 높아질수록 만족시켜야 할 제약 조건이 늘어난다.
- 정규화는 데이터베이스의 논리적 설계 단계에서 수행한다.
- 정규화는 논리적 처리 및 품질에 큰 영향을 미친다.
- 정규화된 데이터 모델은 일관성, 정확성, 단순성, 비중복성, 안정성 등을 보장한다.
- 정규화 수준이 높을수록 유연한 데이터 구축이 가능하고 데이터의 정확성이 높아지는 반면 물리적 접근이 복잡하고 너무 많은 조인으로 인해 조회 성능이 저하된다.

1. 논리 데이터베이스 설계-SEC_13(정규화(Normalization))

2) 정규화의 목적

- 데이터 구조의 안정성 및 무결성을 유지한다.
- 어떠한 릴레이션이라도 데이터베이스 내에서 표현 가능하게 만든다.
- 효과적인 검색 알고리즘을 생성할 수 있다.
- 데이터 중복을 배제하여 이상(Anomaly)의 발생 방지 및 자료 저장 공간의 최소화가 가능하다.
- 데이터 삽입 시 릴레이션을 재구성할 필요성을 줄인다.
- 데이터 모형의 단순화가 가능하다.
- 속성의 배열 상태 검증이 가능하다.
- 개체와 속성의 누락 여부 확인이 가능하다.
- 자료 검색과 추출의 효율성을 추구한다.

1. 논리 데이터베이스 설계-SEC_13(정규화(Normalization))

3) 이상(Anomaly)의 개념 및 종류

; 정규화를 거치지 않으면 데이터베이스 내에 데이터들이 불필요하게 중복되어 릴레이션 조작 시 예기치 못한 곤란한 현상이 발생하는데, 이를 이상(Anomaly)이라 하며 삽입 이상, 삭제 이상, 갱신 이상이 있다.

- **삽입 이상(Insertion Anomaly)** : 릴레이션에 데이터를 삽입할 때 의도와는 상관없이 원하지 않은 값들도 함께 삽입되는 현상이다.
- **삭제 이상(Deletion Anomaly)** : 릴레이션에서 한 튜플을 삭제할 때 의도와는 상관없는 값들도 함께 삭제되는 연쇄가 일어나는 현상이다.
- **갱신 이상(Update Anomaly)** : 릴레이션에서 튜플에 있는 속성값을 갱신할 때 일부 튜플의 정보만 갱신되어 정보에 모순이 생기는 현상이다.

1. 논리 데이터베이스 설계-SEC_13(정규화(Normalization))

4) 정규화의 원칙

- 정보의 무손실 표현, 즉 하나의 스키마를 다른 스키마로 변환할 때 정보의 손실이 있어서는 안 된다.
- 분리의 원칙, 즉 하나의 독립된 관계성은 하나의 독립된 릴레이션으로 분리시켜 표현해야 한다.
- 데이터의 중복성이 감소되어야 한다.

5) 정규화 과정

- 1NF(제1정규형)
 - 1NF는 릴레이션에 속한 모든 도메인(Domain)이 원자 값(Atomic Value)만으로 되어 있는 정규형이다.
즉, 릴레이션의 모든 속성 값이 원자 값으로만 되어 있는 정규형이다.
-> 릴레이션의 모든 속성이 단순 영역에서 정의된다.
- 2NF(제2정규형)
 - 2NF는 릴레이션 R이 1NF이고, 기본키가 아닌 모든 속성이 기본키에 대하여 완전 함수적 종속을 만족하는 정규형이다.

1. 논리 데이터베이스 설계-SEC_13(정규화(Normalization))

5) 정규화 과정

● 함수적 종속 / 완전 / 부분 함수적 종속 및 이해

- 함수적 종속(Functional Dependency)

-> 함수적 종속은 데이터들이 어떤 기준값에 의해 종속되는 것을 의미한다.

예를 들어, <수강> 릴레이션이 (학번, 이름, 과목명)으로 되어 있을 때, '학번'이 결정되면 '과목명'에 상관없이 '학번'에는 항상 같은 '이름'이 대응된다. '학번'에 따라 '이름'이 결정될 때 '이름'을 '학번'에 함수 종속적이라고 하며 '학번 \rightarrow 이름'과 같이 쓴다.

- 완전 함수적 종속

-> 어떤 테이블 R에서 속성 A가 다른 속성 집합 B 전체에 대해 함수적 종속이지만 속성 집합 B의 어떠한 진 부분 집합 C(즉, $C \subset B$)에는 함수적 종속이 아닐 때 속성 A는 속성 집합 B에 완전 함수적 종속이라고 한다.

- 부분 함수적 종속

-> 어떤 테이블 R에서 속성 A가 다른 속성 집합 B 전체에 대해 함수적 종속이면서 속성 집합 B의 어떠한 진 부분 집합에도 함수적 종속일 때, 속성 A는 속성 집합 B에 부분 함수적 종속이라고 한다.

1. 논리 데이터베이스 설계-SEC_13(정규화(Normalization))

5) 정규화 과정

● 함수적 종속 / 완전 / 부분 함수적 종속 및 이해

- 완전 / 부분 함수적 종속의 이해

-> 완전 함수적 종속은 어떤 속성이 기본키에 대해 완전히 종속적일 때를 말한다.

예를 들어 <수강> 릴레이션이 (학번, 과목명, 성적, 학년)으로 되어 있고 (학번, 과목명)이 기본키일 때, '성적'은 '학번'과 '과목명'이 같을 경우에는 항상 같은 '성적'이 온다.

즉 '성적'은 '학번'과 '과목명'에 의해서만 결정되므로 '성적'은 기본키(학번, 과목명)에 완전 함수적 종속이 되는 것이다.

-> 반면에 '학년'은 '과목명'에 관계없이 '학번'이 같으면 항상 같은 '학년'이 온다. 즉 기본키의 일부인 '학번'에 의해서 '학년'이 결정되므로 '학년'은 부분 함수적 종속이라고 한다.

● 3NF(제3정규형)

- 3NF는 릴레이션 R이 2NF이고, 기본키가 아닌 모든 속성이 기본키에 대해 이행적 종속을 만족하지 않는 정규형이다.

-> 무손실 조인 또는 종속성 보존을 저해하지 않고도 항상 3NF 설계를 얻을 수 있다.

이행적 종속(Transitive Dependency) 관계 : $A \rightarrow B$ 이고 $B \rightarrow C$ 일 때 $A \rightarrow C$ 를 만족하는 관계를 의미한다.

1. 논리 데이터베이스 설계-SEC_13(정규화(Normalization))

5) 정규화 과정

● BCNF(Boyce-Codd 정규형)

- BCNF는 릴레이션 R에서 결정자가 모두 후보키(Candidate Key)인 정규형이다.
- > 3NF에서 후보키가 여러 개 존재하고 서로 중첩되는 경우에 적용하는, **강한 제3정규형**이라고도 한다.
- > **모든 BCNF(Boyce-Codd Normal Form)가 종속성을 보존하는 것은 아니다.**
- > BCNF의 제약 조건
 - ▶ 키가 아닌 모든 속성은 각 키에 대하여 완전 종속해야 한다.
 - ▶ 키가 아닌 모든 속성은 그 자신이 부분적으로 들어가 있지 않은 모든 키에 대하여 완전 종속해야 한다.
 - ▶ 어떤 속성도 키가 아닌 속성에 대해서는 완전 종속할 수 없다.

결정자/종속자 : 결정자(Determinant)는 속성 간의 종속성을 규명할 때 기준이 되는 값이고, 종속자(Dependent)는 결정자의 값에 의해 정해지는 값을 의미한다.

예) '학번'에 따라 '이름'이 결정되는 '학번→이름'일 때 '학번'을 결정자라고, '이름'을 종속자라 한다.

1. 논리 데이터베이스 설계-SEC_13(정규화(Normalization))

5) 정규화 과정

● 4NF(제4정규형)

- 4NF는 릴레이션 R에 다치 종속 $A \twoheadrightarrow B$ 가 성립하는 경우 R의 모든 속성이 A에 함수적 종속 관계를 만족하는 정규형이다.

● 5NF(제5정규형, PJ/NF)

- 5NF는 릴레이션 R의 모든 조인 종속이 R의 후보키를 통해서만 성립되는 정규형이다.

다치 종속(Multi Valued Dependency, 다가 종속) : A, B, C 3개의 속성을 가진 릴레이션 R에서 어떤 복합 속성 (A, C)에 대응하는 B 값의 집합이 A 값에만 종속되고 C 값에는 무관하면, B는 A에 다치 종속이라 하고 ' $A \twoheadrightarrow B$ '로 표기한다.

조인 종속(Join Dependency) : 어떤 릴레이션 R의 속성에 대한 부분집합 A, B, ..., C가 있다고 해보자. 이때 만일 릴레이션 R이 자신의 프로젝션(Projection) A, B, ..., C를 모두 조인한 결과가 자신과 동일한 경우 릴레이션 R은 조인 종속(A, B, ..., C)을 만족한다고 한다.

1. 논리 데이터베이스 설계-SEC_13(정규화(Normalization))

6) 정규화 과정 정리



정규화 단계 암기 요령

두부를 좋아하는 정규화가 두부가게에
가서 가게에 있는 두부를 다 달라고 말
하니 주인이 깜짝 놀라며 말했다.

두부이걸다줘? - 도부이걸다조

도메인이 원자값

부분적 함수 종속 제거

이행적 함수 종속 제거

결정자이면서 후보키가 아닌 것 제거

다치 종속 제거

조인 종속성 이용

논리 데이터베이스 설계-SEC_13(정규화(Normalization)) 기출 문제

기출 문제(정규화(Normalization))

1. 정규화의 목적으로 틀린 것은?

- ① 어떠한 릴레이션이라도 데이터베이스 내에서 표현 가능하게 만든다.
- ② 데이터 삽입 시 릴레이션을 재구성할 필요성을 줄인다.
- ③ 중복을 배제하여 삽입, 삭제, 갱신 이상의 발생을 야기한다.
- ④ 효과적인 검색 알고리즘을 생성할 수 있다.

정규화의 목적

- 1. 데이터 구조의 안정성 및 무결성을 유지한다.
- 2. 어떠한 릴레이션이라도 데이터베이스 내에서 표현 가능하게 만든다.
- 3. 효과적인 검색 알고리즘을 생성할 수 있다.
- 4. 데이터 중복을 배제하여 이상(Anomaly)의 발생 방지 및 자료 저장 공간의 최소화가 가능하다.
- 5. 데이터 삽입 시 릴레이션을 재구성할 필요성을 줄인다.
- 6. 데이터 모형의 단순화가 가능하다.
- 7. 속성의 배열 상태 검증이 가능하다.
- 8. 개체와 속성의 누락 여부 확인이 가능하다.

3. 다음과 같이 왼쪽 릴레이션을 오른쪽 릴레이션으로 정규화를 하였을 때 어떤 정규화 작업을 한 것인가?

국가	도시		국가	도시
대한민국	서울, 부산	⇒	대한민국	서울
미국	워싱턴, 뉴욕		대한민국	부산
중국	베이징		미국	워싱턴
			미국	뉴욕
			중국	베이징

- ① 제1정규형 ② 제2정규형
- ③ 제3정규형 ④ 제4정규형

테이블 '도시'속성이 다중값을 갖고 있었으나 정규화를 수행한 후에는 한 개의 값 즉 원자값(Atomic Value)만을 가진 것으로 보아 제1정규화 작업을 수행한 것임을 알 수 있다.

제1정규화

1NF는 릴레이션에 속한 모든 도메인(Domain)이 원자값만으로 되어 있는 정규형이다. 즉, 릴레이션의 모든 속성 값이 원자 값으로만 되어 있는 정규형이다.

-> 릴레이션의 모든 속성이 단순 영역에서 정의된다.

4. 정규화를 거치지 않아 발생하게 되는 이상(anomaly) 현상의 종류에 대한 설명으로 옳지 않은 것은?

논리 데이터베이스 설계-SEC_13(정규화(Normalization)) 기출 문제

기출 문제(정규화(Normalization))

5. 정규화 과정 1NF에서 2NF가 되기 위한 조건은?

- ① 1NF를 만족하고 모든 도메인이 원자값 이어야 한다.
- ② 1NF를 만족하고, 키가 아닌 모든 애트리뷰트들이 기본키에 이행적으로 함수 종속되지 않아야 한다.
- ③ 1NF를 만족하고 다치 종속이 제거되어야 한다.
- ④ 1NF를 만족하고 키가 아닌 모든 속성이 기본키에 대하여 완전 함수적 종속관계를 만족해야 한다.

두부이절다조 = 도부이절다조 에서 부에 해당하는 것으로 부분적 함수 종속을 제거함으로써 완전 함수적 종속을 만족한다.

제2정규형

- 2NF는 릴레이션 R이 1NF이고, 기본키가 아닌 모든 속성이 기본키에 대하여 완전 함수적 종속을 만족하는 정규형이다.

6. 제3정규형에서 보이스코드 정규형(BCNF)으로 정규화하기 위한 작업은?

- ① 원자 값이 아닌 도메인을 분해
- ② 부분 함수 종속 제거
- ③ 이행 함수 종속 제거

7. 어떤 릴레이션 R에서 X와 Y를 각각 R의 애트리뷰트 집합의 부분 집합이라고 할 경우 애트리뷰트 X의 값 각각에 대해 시간에 관계없이 항상 애트리뷰트 Y의 값이 오직 하나만 연관되어 있을 때 Y는 X에 함수종속이라고 한다. 이 함수 종속의 표기로 옳은 것은?

- ① $Y \rightarrow X$ ② $Y \subset X$
- ③ $X \rightarrow Y$ ④ $X \subset Y$

항상 X에 따라 Y가 결정될 때 Y를 X에 함수 종속적이라고 하며, $X \rightarrow Y$ 와 같이 기재한다. 중요.

8. 다음 조건을 모두 만족하는 정규형은?

테이블 R에 속한 모든 도메인이 원자값만으로 구성되어 있다.

테이블 R에서 키가 아닌 모든 필드가 키에 대해 함수적으로 종속되며, 키의 부분집합이 결정자가 되는 부분 종속이 존재하지 않는다.

테이블 R에 존재하는 모든 함수적 종속에서 결정자가 후보키이다.

- ① BCNF ② 제1정규형
- ③ 제2정규형 ④ 제4정규형

지문에 제시된 조건은 '도메인 원자값', '부분 종속이 제거', '결정자가 후보'이며, 이는 도부이절다조에서 결까지 만족한 것이다.

논리 데이터베이스 설계-SEC_13(정규화(Normalization)) 기출 문제

기출 문제(정규화(Normalization))

9. 다음에 해당하는 함수 종속의 추론 규칙은?

$X \rightarrow Y$ 이고 $Y \rightarrow Z$ 이면 $X \rightarrow Z$ 이다.

- ① 분해 규칙 ② 이행 규칙
- ③ 반사 규칙 ④ 결합 규칙

이행적 종속(Transitive Dependency) 관계 : $A \rightarrow B$ 이고 $B \rightarrow C$ 일 때 $A \rightarrow C$ 를 만족하는 관계를 의미한다.

제3정규형

- 3NF는 릴레이션 R이 2NF이고, 기본키가 아닌 모든 속성이 기본키에 대해 이행적 종속을 만족하지 않는 정규형이다.

-> 무손실 조인 또는 종속성 보존을 저해하지 않고도 항상 3NF설계를 얻을 수 있다.

10. 릴레이션 조작 시 데이터들이 불필요하게 중복되어 예기치 않게 발생하는 곤란한 현상을 의미하는 것은?

- ① normalization ② rollback
- ③ cardinality ④ anomaly

정규화를 거치지 않으면 데이터베이스 내에 데이터들이 불필요하게 중복되어 릴레이션 조작 시 예기치 못한 곤란한

11. 정규화에 대한 설명으로 적절하지 않은 것은?

- ① 데이터베이스의 개념적 설계 단계 이전에 수행한다.
- ② 데이터 구조의 안정성을 최대화한다.
- ③ 중복을 배제하여 삽입, 삭제, 갱신 이상의 발생을 방지한다.
- ④ 데이터 삽입 시 릴레이션을 재구성할 필요성을 줄인다.

정규화는 개념적 설계의 다음 단계인 논리적 설계 단계에서 수행하는 작업이다.

정규화란 함수적 종속성 등의 종속성 이론을 이용하여 잘못 설계된 관계형 스키마를 더 작은 속성의 세트로 쪼개어 바람직한 스키마로 만들어 가는 과정이다.

- 1. 하나의 종속성이 하나의 릴레이션에 표현될 수 있도록 분해해가는 과정이라 할 수 있다.
- 2. 정규형에는 제1정규형, 제2정규형, 제3정규형, BCNF형, 제4정규형, 제5정규형이 있으며, 차수가 높아질수록 만족시켜야 할 제약 조건이 늘어난다.
- 3. 정규화는 개념적 설계의 다음 단계인 논리적 설계 단계에서 수행하는 작업이다.

논리 데이터베이스 설계-SEC_13(정규화(Normalization)) 기출 문제

기출 문제(정규화(Normalization))

13. 정규화 과정에서 함수 종속이 $A \rightarrow B$ 이고 $B \rightarrow C$ 일 때 $A \rightarrow C$ 인 관계를 제거하는 단계는?

- ① 1NF \rightarrow 2NF ② 2NF \rightarrow 3NF
- ③ 3NF \rightarrow BCNF ④ BCNF \rightarrow 4NF

3NF은 릴레이션 R이 2NF이고, 기본키가 아니 모든 속성이 기본키에 대해 이행적 종속을 만족하지 않는 정규형이다.
도부이결다조에서 이에 해당하는 것이다.

14. 데이터 속성 간의 종속성에 대한 엄밀한 고려 없이 잘못 설계된 데이터베이스에서는 데이터 처리 연산수행 시 각종 이상 현상이 발생할 수 있는데, 이러한 이상 현상이 아닌 것은?

- ① 검색 이상 ② 삽입 이상
- ③ 삭제 이상 ④ 갱신 이상

이상(Anomaly)에는 삽입, 삭제, 갱신 이상이 있다.

15. 다음 정의에서 말하는 기본 정규형은?

어떤 릴레이션 R에 속한 모든 도메인이 원자값(Atomic Value)만으로 되어 있다.

- ① 제1정규형(NF) ② 제2정규형(NF)
- ③ 제3정규형(NF) ④ 보이스/코드 정규형(BCNF)

도부이결다조에서 도에 해당한다.

제1정규형

1NF는 릴레이션에 속한 모든 도메인이 원자 값만으로 되어 있는 정규형이다.

16. 이전 단계의 정규형을 만족하면서 후보키를 통하지 않는 조인 종속(JD : Join Dependency)을 제거해야 만족하는 정규형은?

- ① 제3정규형(NF) ② 제4정규형(NF)
- ③ 제5정규형(NF) ④ 제6정규형(NF)

도부이결다조에서 조에 해당한다.

5NF(제5정규형, PJ/NF)

5NF는 릴레이션 R의 모든 조인 종속이 R의 후보키를 통해서만 성립되는 정규형이다.

조인 종속(Join Dependency) : 어떤 릴레이션 R의 속성에 대한

1. 논리 데이터베이스 설계-SEC_14(반정규화(Denormalization))

1) 반정규화의 개념

; 반정규화란 시스템의 성능 향상, 개발 및 운영의 편의성 등을 위해 정규화된 데이터 모델을 통합, 중복, 분리하는 과정으로, 의도적으로 정규화 원칙을 위배하는 행위이다.

- 반정규화를 수행하면 시스템의 성능이 향상되고 관리 효율성은 증가하지만 데이터의 일관성 및 정합성이 저하될 수 있다.
- 과도한 반정규화는 오히려 성능을 저하시킬 수 있다.
- 반정규화를 위해서는 사전에 데이터의 일관성과 무결성을 우선으로 할지, 데이터베이스의 성능과 단순화를 우선으로 할지를 결정해야 한다.
- 반정규화 방법에는 테이블 통합, 테이블 분할, 중복 테이블 추가, 중복 속성 추가 등이 있다.

정규화의 원칙

정보의 무손실 표현, 즉 하나의 스키마를 다른 스키마로 변환할 때 정보의 손실이 있어서는 안 된다.
분리의 원칙, 즉 하나의 독립된 관계성은 하나의 독립된 릴레이션으로 분리시켜 표현해야 한다.
데이터의 중복성이 감소되어야 한다.

1. 논리 데이터베이스 설계-SEC_14(반정규화(Denormalization))

2) 테이블 통합

; 테이블 통합은 두 개의 테이블이 조인(Join)되는 경우가 많아 하나의 테이블로 합쳐 사용하는 것이 성능 향상에 도움이 될 경우 수행한다.

- 두 개의 테이블에서 발생하는 프로세스가 동일하게 자주 처리되는 경우, 두 개의 테이블을 이용하여 항상 조회를 수행하는 경우 테이블 통합을 고려한다.

학번	담당교수	담당교수	과목명
201001	홍길동	홍길동	정보처리
201002	유관순	이순신	정보처리
201003	윤봉길	윤봉길	인공지능
201004	홍길동	유관순	네트워크
201005	이순신		
201006	유관순		

학번	담당교수	과목명
201001	홍길동	정보처리
201002	유관순	네트워크
201003	윤봉길	인공지능
201004	홍길동	정보처리
201005	이순신	정보처리
201006	유관순	네트워크

<테이블 통합>

1. 논리 데이터베이스 설계-SEC_14(반정규화(Denormalization))

2) 테이블 통합

- 테이블 통합의 종류에는 1:1 관계 테이블 통합, 1:N 관계 테이블 통합, 슈퍼타입/서브타입 테이블 통합이 있다.
- 테이블 통합 시 고려 사항
 - 데이터 검색은 간편하지만 레코드 증가로 인해 처리량이 증가한다.
 - 테이블 통합으로 인해 입력, 수정, 삭제 규칙이 복잡해질 수 있다.
 - Not Null, Default, Check 등의 제약조건(Constraint)을 설계하기 어렵다.

슈퍼타입/서브타입 : 슈퍼타입은 상위 개체를 서브타입은 하위 개체를 의미한다.

Not Null : 속성 값이 Null이 될 수 없다.

Default : 속성 값이 생략되면 기본값 설정

Check : 속성 값의 범위나 조건을 설정하여 설정한 값만 허용

1. 논리 데이터베이스 설계-SEC_14(반정규화(Denormalization))

3) 테이블 분할

- 테이블 분할은 테이블을 수직 또는 수평으로 분할하는 것이다.

〈수평 분할〉			〈수직 분할〉		
학번	담당교수	과목명	학번	담당교수	과목명
201001	홍길동	정보처리	201001	홍길동	정보처리
201002	유관순	네트워크	201002	유관순	네트워크
201003	윤봉길	인공지능	201003	윤봉길	인공지능
201004	홍길동	정보처리	201004	홍길동	정보처리
201005	이순신	정보처리	201005	이순신	정보처리
201006	유관순	네트워크	201006	유관순	네트워크

↓

학번	담당교수	과목명
201001	홍길동	정보처리
201002	유관순	네트워크
201003	윤봉길	인공지능

↓

학번	담당교수	과목명
201004	홍길동	정보처리
201005	이순신	정보처리
201006	유관순	네트워크

↓

학번	담당교수
201001	홍길동
201002	유관순
201003	윤봉길
201004	홍길동
201005	이순신
201006	유관순

↓

학번	과목명
201001	정보처리
201002	네트워크
201003	인공지능
201004	정보처리
201005	정보처리
201006	네트워크

1. 논리 데이터베이스 설계-SEC_14(반정규화(Denormalization))

3) 테이블 분할

● 수평 분할(Horizontal Partitioning)

- 수평 분할은 레코드(Record)를 기준으로 테이블을 분할하는 것이다.
- 레코드 별로 사용 빈도의 차이가 큰 경우 사용 빈도에 따라 테이블을 분할한다.

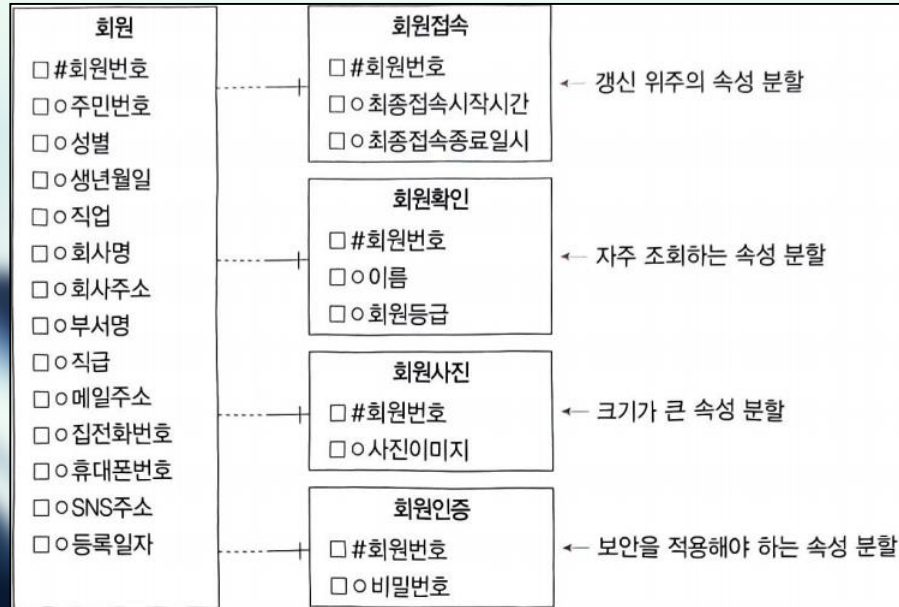
● 수직 분할(Vertical Partitioning)

- 수직 분할은 하나의 테이블에 속성이 너무 많을 경우 속성을 기준으로 테이블을 분할하는 것이다.
- 갱신 위주의 속성 분할 : 데이터 갱신 시 레코드 잠금으로 인해 다른 작업을 수행할 수 없으므로 갱신이 자주 일어나는 속성들을 수직 분할하여 사용한다.
- 자주 조회되는 속성 분할 : 테이블에서 자주 조회되는 속성이 극히 일부일 경우 자주 사용되는 속성들을 수직 분할하여 사용한다.
- 크기가 큰 속성 분할 : 이미지나 2GB 이상 저장될 수 있는 텍스트 형식 등으로 된 속성들을 수직 분할하여 사용한다.
- 보안을 적용해야 하는 속성 분할 : 테이블 내의 특정 속성에 대해 보안을 적용할 수 없으므로 보안을 적용해야 하는 속성들을 수직 분할하여 사용한다.

레코드 잠금 : 데이터의 무결성을 유지하기 위해 어떤 프로세스가 데이터 값을 변경하려고 하면 변경 작업이 완료될 때까지 다른 프로세스가 해당 데이터 값을 변경하지 못하도록 하는 것을 의미한다.

1. 논리 데이터베이스 설계-SEC_14(반정규화(Denormalization))

3) 테이블 분할



<수직 분할 수행의 예>

● 테이블 분할 시 고려 사항

- 기본키의 유일성 관리가 어려워진다.
- 데이터 양이 적거나 사용 빈도가 낮은 경우 테이블 분할이 필요한지를 고려해야 한다.
- 분할된 테이블로 인해 수행 속도가 느려질 수 있다.
- 데이터 검색에 중점을 두어 테이블 분할 여부를 결정해야 한다.

1. 논리 데이터베이스 설계-SEC_14(반정규화(Denormalization))

4) 중복 테이블 추가

; 여러 테이블에서 데이터를 추출해서 사용해야 하거나 다른 서버에 저장된 테이블을 이용해야 하는 경우 중복 테이블을 추가하여 작업의 효율성을 향상시킬 수 있다.

● 중복 테이블을 추가하는 경우

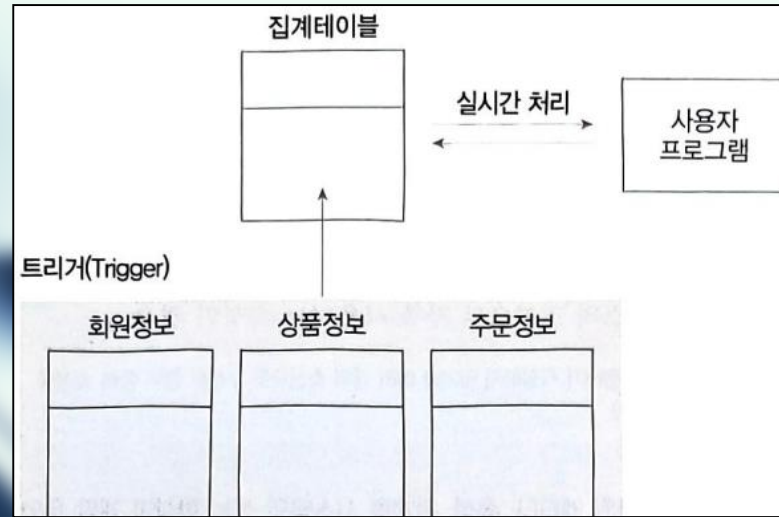
- 정규화로 인해 수행 속도가 느려지는 경우
- 많은 범위의 데이터를 자주 처리해야 하는 경우
- 특정 범위의 데이터만 자주 처리해야 하는 경우
- 처리 범위를 줄이지 않고는 수행 속도를 개선할 수 없는 경우

● 중복 테이블을 추가하는 방법은 다음과 같다.

- **집계 테이블의 추가** : 집계 데이터를 위한 테이블을 생성하고, 각 원본 테이블에 트리거(Trigger)를 설정하여 사용하는 것으로, 트리거의 오버헤드(Overhead)에 유의해야 한다.
- **진행 테이블의 추가** : 이력 관리 등의 목적으로 추가하는 테이블로, 적절한 데이터 양의 유지와 활용도를 높이기 위해 기본키를 적절히 설정한다.
- **특정 부분만을 포함하는 테이블의 추가** : 데이터가 많은 테이블의 특정 부분만을 사용하는 경우 해당 부분만으로 새로운 테이블을 생성한다.

1. 논리 데이터베이스 설계-SEC_14(반정규화(Denormalization))

4) 중복 테이블 추가



<집계 테이블 추가>

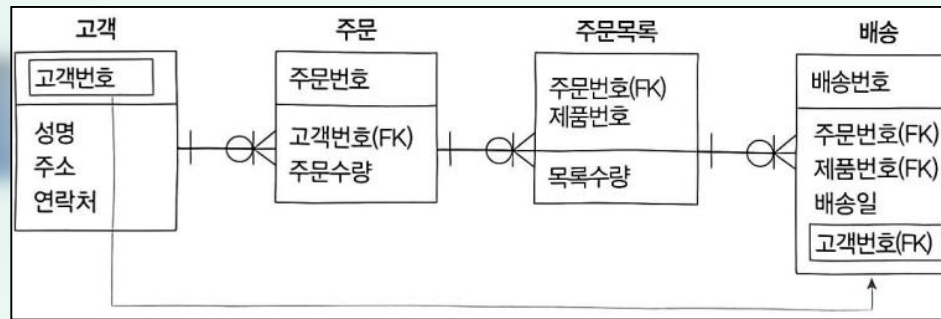
위의 이미지를 보면 <회원정보>, <상품정보>, <주문정보> 테이블의 데이터를 집계하는 집계 테이블을 추가하여 시스템 사용이 적은 시간에 배치 작업에 의해 원하는 데이터를 생성하여 사용하는 것이다.

1. 논리 데이터베이스 설계-SEC_14(반정규화(Denormalization))

5) 중복 속성 추가

; 중복 속성 추가는 조인해서 데이터를 처리할 때 데이터를 조회하는 경로를 단축하기 위해 자주 사용하는 속성을 하나 더 추가하는 것이다.

- 중복 속성을 추가하면 데이터의 무결성 확보가 어렵고, 디스크 공간이 추가로 필요하다.



<중복 속성 추가>

1. 논리 데이터베이스 설계-SEC_14(반정규화(Denormalization))

5) 중복 속성 추가

● 중복 속성을 추가하는 경우

- 조인이 자주 발생하는 속성인 경우
- 접근 경로가 복잡한 속성인 경우
- 액세스의 조건으로 자주 사용되는 속성인 경우
- 기본키의 형태가 적절하지 않거나 여러 개의 속성으로 구성된 경우

● 중복 속성 추가 시 고려 사항

- 테이블 중복과 속성의 중복을 고려한다.
- 데이터 일관성 및 무결성에 유의해야 한다.
- SQL 그룹 함수를 이용하여 처리할 수 있어야 한다.
- 저장 공간의 지나친 낭비를 고려한다.

논리 데이터베이스 설계-SEC_14(반정규화(Denormalization)) 기출 및 출제 예상 문제

기출 및 출제 예상문제(반정규화(Denormalization))

1. 정규화된 엔티티, 속성, 관계를 시스템의 성능 향상과 개발 운영의 단순화를 위해 중복, 통합, 분리 등을 수행하는 데이터 모델링 기법은?

- ① 인덱스 정규화 ② 반정규화
- ③ 집단화 ④ 머징

반정규화란 시스템의 성능 향상, 개발 및 운영의 편리성 등을 위해 정규화된 데이터 모델을 통합, 중복, 분리하는 과정으로 의도적으로 정규화 원칙을 위배하는 행위이다.

- 반정규화를 수행하면 시스템의 성능이 향상되고 관리 효율성은 증가하지만 데이터의 일관성 및 정합성이 저하될 수 있다.

- 과도한 반정규화는 오히려 성능을 저하시킬 수 있다.

- 반정규화를 위해서는 사전에 데이터의 일관성과 무결성을 우선으로 해야 할지, 데이터베이스의 성능과 단순화를 우선으로 할지를 결정해야 한다.

- 반정규화 방법에는 테이블 통합, 테이블 분할, 중복 테이블 추가, 중복 속성 추가 등이 있다.

집단화 : 클래스들 사이의 부분-전체 또는 부분관계로 설명되는 연관성을 나타낸다.

3. 다음 중 반정규화에 대한 설명으로 가장 옳지 않은 것은?

① 반정규화는 의도적으로 정규화 원칙을 위배하는 행위이다.

② 반정규화를 수행하면 시스템의 성능이 저하된다.

③ 반정규화로 인해 데이터의 일관성 및 정합성이 저하될 수 있다.

④ 반정규화 방법에는 테이블 통합, 테이블 분할, 중복 테이블 및 중복 속성 추가 등이 있다.

반정규화를 수행하는 이유는 시스템 성능 향상과 관리 효율성의 증가이다.

4. 다음 중 반정규화 방법 중 테이블 통합의 종류가 아닌 것은?

① 1:1 관계 테이블 통합

② 1:N 관계 테이블 통합

③ 슈퍼/서브타입 테이블 통합

④ N:M 관계 테이블 통합

테이블 통합의 종류에는 1:1 관계 테이블 통합, 1:N 관계 테이블 통합, 슈퍼/서브타입 테이블 통합이 있다.

테이블 통합 시 고려사항

- 데이터 검색은 간편하지만 레코드 증가로 인해 처리량이 증가한다.

- 테이블 통합으로 인해 입력, 수정, 삭제 규칙이 복잡해질 수 있다.

논리 데이터베이스 설계-SEC_14(반정규화(Denormalization)) 기출 및 출제 예상 문제

기출 및 출제 예상문제(반정규화(Denormalization))

5. 다음 중 반정규화 시 중복 테이블을 추가해야 하는 경우가 아닌 것은?

- ① 많은 양의 범위를 자주 처리해야 하는 경우
- ② 특정 범위의 데이터만 자주 처리해야 하는 경우
- ③ 처리 범위를 늘리지 않고는 수행 속도를 개선할 수 없는 경우
- ④ 정규화로 인해 수행 속도가 느려지는 경우

수행 속도를 개선하려면 처리 범위를 줄여야 한다.

중복 테이블을 추가하는 경우

- 정규화로 인해 수행 속도가 느려지는 경우
- 많은 범위의 데이터를 자주 처리하는 경우
- 특정 범위의 데이터만 자주 처리해야 하는 경우
- 처리 범위를 줄이지 않고는 수행 속도를 개선할 수 없는 경우

6. 다음의 내용은 반정규화의 어떤 유형에 대한 고려사항인가?

데이터 검색은 간편하지만 Row 수가 증가하여 처리량이 증가할 수 있음을 고려해야 한다.

입력, 수정, 삭제 규칙이 복잡해질 수 있음을 고려해야 한다.

7. 다음 중 반정규화 시 중복 속성을 추가해야 하는 경우가 아닌 것은?

- ① 기본키가 하나의 속성으로 구성된 경우
- ② 조인이 자주 발생하는 속성인 경우
- ③ 접근 경로가 복잡한 속성인 경우
- ④ 액세스의 조건으로 자주 사용되는 속성인 경우

중복 속성을 추가하는 경우

- 조인이 자주 발생하는 속성인 경우
- 접근 경로가 복잡한 속성인 경우
- 액세스의 조건으로 자주 사용되는 속성인 경우
- 기본키의 형태가 적절하지 않거나 여러 개의 속성으로 구성된 경우

8. 다음 반정규화 방법 중 중복 속성 추가 시 고려할 사항으로 가장 거리가 먼 것은?

- ① 테이블 중복과 속성의 중복을 고려한다.
- ② 데이터 일관성 및 무결성에 유의해야 한다.
- ③ SQL Group Function을 이용하여 해결 가능한지 검토한다.
- ④ 기본키의 유일성 관리가 어려워짐을 고려한다.

중복 속성 추가 시 고려사항

- 테이블 중복과 속성의 중복을 고려한다.

1. 논리 데이터베이스 설계-SEC_15(시스템 카탈로그)

1) 시스템 카탈로그(System Catalog)의 의미

; 시스템 카탈로그는 시스템 그 자체에 관련이 있는 다양한 객체에 관한 정보를 포함하는 시스템 데이터베이스이다.

- 시스템 카탈로그 내의 각 테이블은 사용자를 포함하여 DBMS에서 지원하는 모든 데이터 객체에 대한 정의나 명세에 관한 정보를 유지 관리하는 시스템 테이블이다.
- 카탈로그들이 생성되면 데이터 사전(Data Dictionary)에 저장되기 때문에 좁은 의미로는 카탈로그를 데이터 사전이라고도 한다.

2) 시스템 카탈로그 저장 정보

; 시스템 카탈로그에 저장된 정보를 메타 데이터(Meta-Data)라고 한다.

- 메타 데이터의 유형
 - 데이터베이스 객체 정보 : 테이블(Table), 인덱스(Index), 뷰(View) 등의 구조 및 통계 정보
 - 사용자 정보: 아이디, 비밀번호, 접근 권한 등
 - 테이블의 무결성 제약 조건 정보 : 기본키, 외래키, NULL 값 허용 여부 등
 - 함수, 프로시저, 트리거 등에 대한 정보

Catalog(카탈로그)의 사전적 의미는 상품 목록, 상품 안내서를 말한다. 즉, 어떤 상품을 소개하기 위해 기업에서 발행하는 책자다.
메타 데이터 : '속성정보'라고도 불리는 메타 데이터는 '데이터에 관한 구조화된 데이터', '다른 데이터를 설명해 주는 데이터' 이다.

1. 논리 데이터베이스 설계-SEC_15(시스템 카탈로그)

3) 카탈로그의 특징

- 카탈로그 자체도 시스템 테이블로 구성되어 있어 일반 이용자도 SQL을 이용하여 내용을 검색해 볼 수 있다.
- INSERT, DELETE, UPDATE문으로 카탈로그를 갱신하는 것은 허용되지 않는다.
- 데이터베이스 시스템에 따라 상이한 구조를 갖는다.
- 카탈로그는 DBMS가 스스로 생성하고 유지한다.
- 카탈로그의 갱신 : 사용자가 SQL문을 실행시켜 기본 테이블, 뷰, 인덱스 등에 변화를 주면 시스템이 자동으로 갱신한다.
- 분산 시스템에서의 카탈로그 : 보통의 릴레이션, 인덱스, 사용자 등의 정보를 포함할 뿐 아니라 위치 투명성 및 중복 투명성을 제공하기 위해 필요한 모든 제어 정보를 가져야 한다.

인덱스(Index)는 데이터베이스 테이블에 대한 검색 성능의 속도를 높여주는 자료구조이다. 특정 컬럼에 인덱스를 생성하면, 해당 컬럼의 데이터들을 정렬하여 별도의 메모리 공간에 데이터의 물리적 주소와 함께 저장된다.

뷰(View)는 사용자에게 접근이 허용된 자료만을 제한적으로 보여주기 위해 하나 이상의 기본 테이블로부터 유도된, 이름을 가지는 가상 테이블이다.

1. 논리 데이터베이스 설계-SEC_15(시스템 카탈로그)

4) 카탈로그/데이터 사전을 참조하기 위한 DBMS 내의 모듈 시스템

- 데이터 정의를 번역기(DDL Compiler) : DDL을 메타 데이터를 갖는 테이블(카탈로그)로 변환하여 데이터 사전에 저장시킨다.
- 데이터 조작어 번역기(DML Compiler) : 응용 프로그램에 삽입된 DML문을 주 언어로 표현한 프로시저 호출로 변환하여 질의 처리기와 상호 통신한다.
- Data Directory
 - 데이터 사전에 수록된 데이터를 실제로 접근하는 데 필요한 정보를 관리 유지하는 시스템이다.
 - 시스템 카탈로그는 사용자와 시스템 모두 접근할 수 있지만 데이터 디렉터리는 시스템만 접근할 수 있다.
- 질의 최적화기 : 사용자의 요구를 효율적인 형태로 변환하고 질의를 처리하는 좋은 전략을 모색한다.
- 트랜잭션 처리기 : 복수 사용자 환경에서 병행으로 동시에 일어나는 트랜잭션 문제를 해결하여, 각각의 사용자가 데이터베이스 자원을 배타적으로 이용할 수 있도록 한다.

논리 데이터베이스 설계-SEC_15(시스템 카탈로그) 기출 및 출제 예상 문제

기출 및 출제 예상문제(시스템 카탈로그)

1. 시스템 자신이 필요로 하는 여러 가지 객체에 관한 정보를 포함하고 있는 시스템 데이터베이스로서, 포함하고 있는 객체로는 테이블, 데이터베이스, 뷰, 접근 권한 등이 있는 것은?

- ① 인덱스(Index)
- ② 카탈로그(Catalog)
- ③ QBE(Query By Example)
- ④ SQL(Structure Query Language)

시스템 카탈로그는 시스템 그 자체에 관련이 있는 다양한 객체에 관한 정보를 포함하는 시스템 데이터베이스이다.

- 시스템 카탈로그 내의 각 테이블은 사용자를 포함하여 DBMS에서 지원하는 모든 데이터 객체에 대한 정의나 명세에 관한 정보를 유지 관리하는 시스템 테이블이다.
- 카탈로그들이 생성되면 데이터 사전(Data Dictionary)에 저장되기 때문에 좁은 의미로는 카탈로그를 데이터 사전이라고도 한다.

QBE : IBM사의 데이터베이스 시스템. 화면에 나타나는 예를 통하여 관계 데이터베이스 시스템에서 자료를 검색하거나 자료의 내용을

3. 시스템 카탈로그에 대한 설명으로 옳지 않은 것은?

- ① 시스템 자신이 필요로 하는 여러 가지 개체에 대한 정보를 포함한 시스템 데이터베이스이다.
- ② 개체들로서는 기본 테이블, 뷰, 인덱스, 데이터베이스, 패키지, 접근 권한 등이 있다.
- ③ 카탈로그 자체도 시스템 테이블로 구성되어 있어 일반 이용자도 SQL을 이용하여 내용을 검색해 볼 수 있다.
- ④ 모든 데이터베이스 시스템에서 요구하는 정보는 동일하므로 데이터베이스 시스템의 종류에 관계없이 동일한 구조로 필요한 정보를 제공한다.

카탈로그의 특징

- 1. 카탈로그 자체도 시스템 테이블로 구성되어 있어 일반 사용자도 SQL을 이용하여 내용을 검색해 볼 수 있다.
- 2. Insert, Delete, Update 구문으로 카탈로그를 갱신하는 것은 허용되지 않는다.
- 3. 데이터베이스 시스템에 따라 상이한 구조를 갖는다.
- 4. 카탈로그는 DBMS가 스스로 생성하고 유지한다.
- 5. 카탈로그의 갱신 : 사용자가 SQL문을 실행시켜 기본 테이블, 뷰,

논리 데이터베이스 설계-SEC_15(시스템 카탈로그) 기출 및 출제 예상 문제

기출 및 출제 예상문제(시스템 카탈로그)

5. 시스템 카탈로그에 대한 설명으로 옳지 않은 것은?

- ① 시스템 자체에 관련 있는 다양한 객체에 관한 정보를 포함하는 시스템 데이터베이스이다.
- ② 데이터 사전이라고도 한다.
- ③ 기본 테이블, 뷰, 인덱스, 패키지, 접근 권한 등의 정보를 저장한다.
- ④ 시스템을 위한 정보를 포함하는 시스템 데이터베이스이므로 일반 사용자는 SQL을 이용하여 내용을 검색해 볼 수 없다.

시스템 카탈로그 자체도 시스템 테이블로 구성되어 있기 때문에 일반 사용자도 SQL문을 이용하여 내용을 검색해 볼 수 있다. 단, 갱신은 불가능 하다.

6. 다음 영문의 () 안 내용으로 가장 적절한 것은?

A(n) () is a file that contains meta data that is, data about data. This file is consulted before actual data are read or modified in the database system.

- ① VIEW ② Index
- ③ ISAM File ④ Data Dictionary

7. 데이터 사전에 대한 설명으로 틀린 것은?

- ① 시스템 카탈로그 또는 시스템 데이터베이스라고도 한다.
 - ② 데이터 사전 역시 데이터베이스의 일종이므로 일반 사용자가 생성, 유지 및 수정할 수 있다.
 - ③ 데이터베이스에 대한 데이터인 메타 데이터(Metadata)를 저장하고 있다.
 - ④ 데이터 사전에 있는 데이터에 실제로 접근하는 데 필요한 위치 정보는 데이터 디렉터리(Data Directory)라는 곳에서 관리한다.
- 데이터 사전은 DBMS가 스스로 생성하고 유지한다. 일반 사용자는 SQL문을 이용하여 내용을 검색할 수 있지만, 내용을 생성하거나 수정할 수는 없다.

8. 시스템 카탈로그에 대한 설명으로 옳지 않은 것은?

- ① 시스템 카탈로그는 DBMS가 생성하고 유지하는 데이터베이스 내의 특별한 테이블들의 집합체이다.
- ② 일반 사용자도 시스템 카탈로그의 내용을 검색할 수 있다.
- ③ 시스템 카탈로그 내의 각 테이블은 DBMS에서 지원하는 개체들에 관한 정보를 포함한다.
- ④ 시스템 카탈로그에 대한 갱신은 데이터베이스의 무결성 유지를 위하여



감사합니다.