

3과목-데이터베이스 구축

(Part 1. 논리 데이터베이스 설계 - Ⅱ)

데이터베이스 구축 총 파트

데이터베이스 구축 3과목은 총 5Part로 이루어져 있다.

1장 논리 데이터베이스 설계(49.72%)

2장 물리 데이터베이스 설계(22.91%)

3장 SQL 응용(26.26%)

4장 SQL 활용(1.12%)

5장 데이터 전환(0.00%)

논리 데이터베이스 설계

논리 데이터베이스 설계 Part는 15개의 섹션으로 구성되어 있다.

01 데이터베이스 설계

02 데이터 모델의 개념

03 데이터 모델의 구성요소 - 개체(Entity)

04 데이터 모델의 구성 요소- 속성(Attribute)

05 데이터 모델의 구성 요소 - 관계(Relationship)

06 식별자(Identifier)

07 E-R(개체-관계) 모델

08 관계형 데이터 모델

09 관계형 데이터베이스의 구조

10 관계형 데이터베이스의 제약 조건 - 키(Key)

11 관계형 데이터베이스의 제약 조건 - 무결성

12 관계대수 및 관계해석

13 정규화(Normalization)

14 반정규화(Denormalization)

15 시스템 카탈로그

1. 논리 데이터베이스 설계-SEC_09(관계형 데이터베이스의 구조)

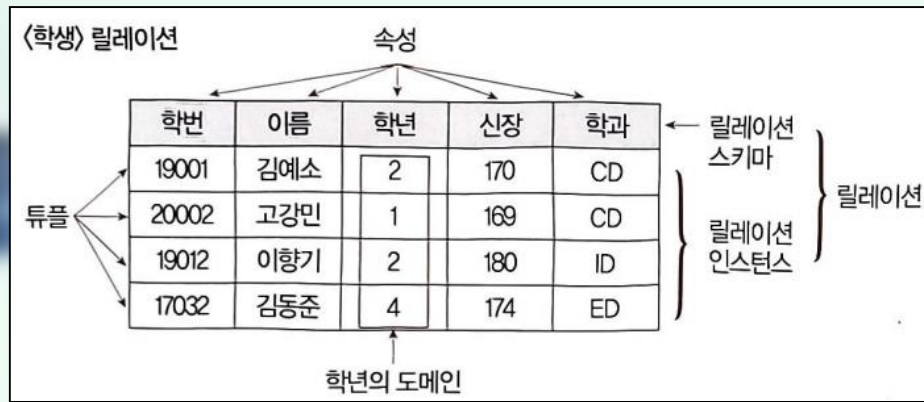
1) 관계형 데이터베이스의 개요

- 1970년 IBM에 근무하던 코드(E. F. Codd)에 의해 처음 제안되었다.
- 관계형 데이터베이스를 구성하는 개체(Entity)나 관계(Relationship)를 모두 릴레이션(Relation)이라는 표(Table)로 표현한다.
- 릴레이션은 개체를 표현하는 개체 릴레이션, 관계를 나타내는 관계 릴레이션으로 구분할 수 있다.
- 장점 : 간결하고 보기 편리하며, 다른 데이터베이스로의 변환이 용이하다.
- 단점: 성능이 다소 떨어진다.

1. 논리 데이터베이스 설계-SEC_09(관계형 데이터베이스의 구조)

2) 관계형 데이터베이스의 Relation 구조

; 릴레이션은 데이터들을 표(Table)의 형태로 표현한 것으로 구조를 나타내는 릴레이션 스키마와 실제 값들인 릴레이션 인스턴스로 구성된다.



; 튜플(Tuple)

- 튜플은 릴레이션을 구성하는 각각의 행을 말한다.
- 튜플은 속성의 모임으로 구성된다.
- 파일 구조에서 레코드와 같은 의미이다.
- 튜플의 수를 카디널리티(Cardinality) 또는 기수, 대응수라고 한다.

관계형 데이터베이스에서 릴레이션을 구성하는 용어들은 매우 중요하다.

용어 : 동일한 요소에 대해서 앞서 사용했던 개체(Entity)에서 사용했던 것과 다르게 부른다.

릴레이션 인스턴스 : 데이터 개체를 구성하고 있는 속성들에 데이터 타입이 정의되어 구체적인 데이터 값을 갖고 있는 것을 말한다. 예) <학생> 릴레이션의 인스턴스

튜플 : <학생> 릴레이션에서 카디널리티는 4이다. 카디널리티 = 튜플의 수 = 기수 = 대응수

1. 논리 데이터베이스 설계-SEC_09(관계형 데이터베이스의 구조)

2) 관계형 데이터베이스의 Relation 구조

; 속성(Attribute)

- 속성은 데이터베이스를 구성하는 가장 작은 논리적 단위이다.
- 파일 구조상의 데이터 항목 또는 데이터 필드에 해당된다.
- 속성은 개체의 특성을 기술한다.
- 속성의 수를 디그리(Degree) 또는 차수라고 한다.

; 도메인(Domain)

- 도메인은 하나의 애트리뷰트가 취할 수 있는 같은 타입의 원자(Atomic)값들의 집합이다.
- 도메인은 실제 애트리뷰트 값이 나타날 때 그 값의 합법 여부를 시스템이 검사하는 데에도 이용된다.
예) 성별 애트리뷰트의 도메인은 '남'과 '여'로 그 외의 값은 입력될 수 없다.

<학생> 릴레이션에서 디그리는 5이다.

디그리 속성의 수 = 차수

<학생> 릴레이션에서 '학년'의 도메인은 1~4이다.

1. 논리 데이터베이스 설계-SEC_09(관계형 데이터베이스의 구조)

3) 릴레이션의 특징

- 한 릴레이션에는 똑같은 튜플이 포함될 수 없으므로 릴레이션에 포함된 튜플들은 모두 상이하다.
예) <학생> 릴레이션을 구성하는 김예소 레코드는 김예소에 대한 학적 사항을 나타내는 것으로 <학생> 릴레이션 내에서는 유일하다.
- 한 릴레이션에 포함된 튜플 사이에는 순서가 없다.
예) <학생> 릴레이션에서 김예소 레코드와 고강민 레코드의 위치가 바뀌어도 상관없다.
- 튜플들의 삽입, 삭제 등의 작업으로 인해 릴레이션은 시간에 따라 변한다.
예) <학생> 릴레이션에 새로운 학생의 레코드를 삽입하거나 기존 학생에 대한 레코드를 삭제함으로써 테이블은 내용 면에서나 크기 면에서 변하게 된다.
- 릴레이션 스키마를 구성하는 속성들 간의 순서는 중요하지 않다.
예) 학번, 이름 등의 속성을 나열하는 순서가 이름, 학번 순으로 바뀌어도 데이터 처리에는 아무런 영향을 미치지 않는다.
- 속성의 유일한 식별을 위해 속성의 명칭은 유일해야 하지만, 속성을 구성하는 값은 동일한 값이 있을 수 있다.
예) 각 학생의 학년을 기술하는 속성인 '학년'은 다른 속성명들과 구분되어 유일해야 하지만 '학년' 속성에는 2, 1, 2, 4 등이 입력된 것처럼 동일한 값이 있을 수 있다.

1. 논리 데이터베이스 설계-SEC_09(관계형 데이터베이스의 구조)

3) 릴레이션의 특징

- 릴레이션을 구성하는 튜플을 유일하게 식별하기 위해 속성들의 부분집합을 키(Key)로 설정한다.
예) <학생> 릴레이션에서는 '학번'이나 '이름'이 튜플들을 구분하는 유일한 값인 키가 될 수 있다.
- 속성의 값은 논리적으로 더 이상 쪼갤 수 없는 원자 값만을 저장한다.
예) 학년'에 저장된 1, 2, 4 등은 더 이상 세분화할 수 없다.

교수번호	성명	전공	소속
1100	신은혁	컴퓨터	서울대
2102	이혜숙	아동	연세대
3200	신희철	식물	경북대

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

도메인(Domain)

- 1. 도메인은 하나의 애트리뷰트가 취할 수 있는 같은 타입의 원자(Atomic) 값들의 집합(범위)이다.
- 2. 도메인은 실제 애트리뷰트 값이 나타날 때 그 값의 합법 여부를 시스템이 검사하는 데에도 이용된다.

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

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

Relation은 테이블(개체, Entity), Attribute는 테이블의 열, Tuple은 테이블의 행(Row)이다.

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

- ① Degree ② Cardinality
- ③ Attribute ④ Cartesian product

Degree는 속성의 수를 의미하고, 튜플의 수를 카디널리티 또는 기수, 대응 수라고 한다.

Cartesian product(카테시안 곱) : From절에 2개 이상의 테이블이 존재

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.

속성의 수를 디그리(Degree) 또는 차수라고 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

한 릴레이션에 속한 튜플들은 모두 서로 다른 값을 가져야 한다. 즉 튜플에 속한 속성 A1, A2, A3 한 개는 다른 튜플들과 다른 값을 가져야 하기 때문에 존재할 수 있는 튜플의 최대 수는 각 도메인이 가지고 있는 값의 종류를 모두 곱한 값이 된다.

$3 * 2 * 4 = 24$ 개가 최대 튜플의 수이다.

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.

릴레이션의 특징

- 1. 한 릴레이션에는 똑같은 튜플이 포함될 수 없으므로 릴레이션에 포함된 튜플들의 모두 상이하다.
- 2. 한 릴레이션에 포함된 튜플 사이에는 순서가 없다.
- 3. 튜플들의 삽입, 삭제 등의 작업으로 인해 릴레이션은 시간에 따라 변한다.
- 4. 릴레이션 스키마를 구성하는 속성들 간의 순서는 중요하지 않다.
- 5. 속성의 유일한 식별을 위해 속성의 명칭은 유일해야 하지만, 속성을 구성하는 값은 동일한 값이 있을 수 있다.
- 6. 릴레이션을 구성하는 튜플을 유일하게 식별하기 위해 속성들의 부분집합을 키(Key)로 설정한다.
- 7. 속성의 값은 논리적으로 더 이상 쪼갤 수 없는 원자 값만을 저장한다.
- 8. 관계형 데이터 모델의 릴레이션에 대한 설명으로 틀린 것은?

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

관계형 데이터베이스 관련 용어 중 행은 튜플(Tuple)이라고 부르며,
열은 속성(Attribute)라고 부른다. 그리고 테이블(Table)
은 릴레이션(Relation)이라고 부른다.

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

속성(Attribute)의 수를 디그리(Degree) 또는 차수라고 한다.

1. 논리 데이터베이스 설계-SEC_10(관계형 데이터베이스의 제약 조건 - 키(Key))

제약 조건이란 데이터베이스에 저장되는 데이터의 정확성을 보장하기 위하여 키(Key)를 이용하여 입력되는 데이터에 제한을 주는 것으로 개체 무결성 제약, 참조 무결성 제약 등이 해당된다.

1) 키(Key)의 개념 및 종류

; 키는 데이터베이스에서 조건에 만족하는 튜플을 찾거나 순서대로 정렬할 때 튜플들을 서로 구분할 수 있는 기준이 되는 애트리뷰트를 말한다.

〈학생〉 릴레이션			〈수강〉 릴레이션	
학번	주민번호	성명	학번	과목명
1001	010429-3*****	김상욱	1001	영어
1002	000504-3*****	임선호	1001	전산
1003	011215-3*****	김한순	1002	영어
1004	001225-4*****	이다해	1003	수학
			1004	영어
			1004	전산

- 키의 종류에는 후보키, 기본키, 대체키, 슈퍼키, 외래키 등이 있다.

1. 논리 데이터베이스 설계-SEC_10(관계형 데이터베이스의 제약 조건 - 키(Key))

2) 후보키(Candidate Key)

; 후보키는 릴레이션을 구성하는 속성들 중에서 튜플을 유일하게 식별하기 위해 사용하는 속성들의 부분집합, 즉 기본키로 사용할 수 있는 속성들을 말한다.

- 하나의 릴레이션 내에서는 중복된 튜플들이 있을 수 없으므로 모든 릴레이션에는 반드시 하나 이상의 후보키가 존재한다.
 - 후보키는 릴레이션에 있는 모든 튜플에 대해서 유일성과 최소성을 만족시켜야 한다.
 - 유일성(Unique) : 하나의 키 값으로 하나의 튜플만을 유일하게 식별할 수 있어야 한다.
 - 최소성(Minimality) : 모든 레코드들을 유일하게 식별하는 데 꼭 필요한 속성으로만 구성되어야 한다.
- <학생> 릴레이션에서 '학번'이나 '주민번호'는 다른 레코드를 유일하게 구별할 수 있는 기본키로 사용할 수 있으므로 후보키이다.

1. 논리 데이터베이스 설계-SEC_10(관계형 데이터베이스의 제약 조건 - 키(Key))

3) 기본키(Primary Key)

; 기본키는 후보키 중에서 특별히 선정된 주키(Main Key)로 중복된 값을 가질 수 없다.

- 한 릴레이션에서 특정 튜플을 유일하게 구별할 수 있는 속성이다.
- 기본키는 후보키의 성질을 갖는다. 즉, 유일성과 최소성을 가지며 튜플을 식별하기 위해 반드시 필요한 키이다.
- 기본키는 NULL 값을 가질 수 없다. 즉 튜플에서 기본키로 설정된 속성에는 NULL 값이 있어서는 안 된다.

예) <학생> 릴레이션에서는 '학번'이나'주민번호'가 기본키가 될 수 있고, <수강> 릴레이션에서는 '학번'+'과목'으로 조합해야 기본키가 만들어진다.

예) '학번'이 <학생> 릴레이션의 기본키로 정의되면 이미 입력된 '1001'은 다른 튜플의 '학번' 속성의 값으로 입력할 수 없다.

널 값(NULL Value) : 데이터베이스에서 아직 알려지지 않거나 모르는 값으로서 '해당 없음' 등의 이유로 정보 부재를 나타내기 위해 사용하는 이론적으로 아무것도 없는 특수한 데이터를 말한다.

1. 논리 데이터베이스 설계-SEC_10(관계형 데이터베이스의 제약 조건 - 키(Key))

4) 대체키(Alternate Key)

; 대체키는 후보키가 둘 이상일 때 기본키를 제외한 나머지 후보키를 의미한다.

- 보조키 라고도 한다.

예) <학생> 릴레이션에서 '학번'을 기본키로 정의하면 '주민번호'는 대체키가 된다.

5) 슈퍼키(Super Key)

; 슈퍼키는 한 릴레이션 내에 있는 속성들의 집합으로 구성된 키로서 릴레이션을 구성하는 모든 튜플들 중 슈퍼키로 구성된 속성의 집합과 동일한 값은 나타나지 않는다.

- 슈퍼키는 릴레이션을 구성하는 모든 튜플에 대해 유일성은 만족시키지만, 최소성은 만족시키지 못한다.

예) <학생> 릴레이션에서는 '학번', '주민번호', '학번'+ '주민번호', '주민번호'+ '성명', '학번'+ '주민번호'+ '성명' 등으로 슈퍼키를 구성할 수 있다.

최소성 : '학번'+ '주민번호'를 사용하여 슈퍼키를 만들면 다른 튜플들과 구분할 수 있는 유일성은 만족하지만, '학번'이나 '주민번호' 하나만 가지고도 다른 튜플들을 구분할 수 있으므로 최소성은 만족시키지 못한다.

1. 논리 데이터베이스 설계-SEC_10(관계형 데이터베이스의 제약 조건 - 키(Key))

6) 외래키(Foreign Key)

; 외래키는 다른 릴레이션의 기본키를 참조하는 속성 또는 속성들의 집합을 의미한다.

- 외래키는 참조되는 릴레이션의 기본키와 대응되어 릴레이션 간에 참조 관계를 표현하는데 중요한 도구이다.
- 한 릴레이션에 속한 속성 A와 참조 릴레이션의 기본키인 B가 동일한 도메인 상에서 정의되었을 때의 속성 A를 외래키라고 한다.
- 외래키로 지정되면 참조 릴레이션의 기본키에 없는 값은 입력할 수 없다.
예) <수강> 릴레이션이 <학생> 릴레이션을 참조하고 있으므로 <학생> 릴레이션의 '학번'은 기본키이고, <수강> 릴레이션의 '학번'은 외래키이다.
예) <수강> 릴레이션의 '학번'에는 <학생> 릴레이션의 '학번'에 없는 값은 입력할 수 없다.

참조 릴레이션 : 외래키를 포함하는 릴레이션이 참조하는 릴레이션이고 대응되는 기본키를 포함하는 릴레이션이 참조 릴레이션이다. 여기서는 <수강> 릴레이션이 참조하는 릴레이션이고, <학생> 릴레이션이 참조 릴레이션이다.

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

슈퍼키(Super Key)

슈퍼키는 한 릴레이션 내에 있는 속성들의 집합으로 구성된 키로써 릴레이션을 구성하는 모든 튜플들 중 슈퍼키로 구성된 속성의 집합과 동일한 값은 나타나지 않는다.

- 슈퍼키는 릴레이션을 구성하는 모든 튜플에 대해 유일성은 만족시키지만, 최소성은 만족시키지 못한다.

외래키(Foreign Key)

외래키는 다른 릴레이션의 기본키를 참조하는 속성 또는 속성들의 집합을 의미한다.

- 외래키는 참조되는 릴레이션의 기본키와 대응되어 릴레이션 간에 참조 관계를 표현하는데 중요한 키이다.

- 한 릴레이션에 속한 속성 A와 참조 릴레이션의 기본키인 B가 동일한 도메인 상에서 정의되었을 때 속성 A를 외래키라고 한다.

- 외래키로 지정되면 참조 릴레이션의 기본키에 없는 값은 입력할

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.

기본키가 지정되어 있지 않아도 검색은 가능하다. 설사 기본키가 있더라도 일반속성으로 조건을 주어 검색도 역시 가능하다.

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.

대체키(Alternate Key)

대체키는 후보키가 둘 이상일 때 기본키를 제외한 나머지 후보키를 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.

기본키는 한 릴레이션에서 특정 튜플을 유일하게 구별하기 위한 속성이므로 기본키로 지정된 속성에는 동일한 값이 있어서는 안 된다. 여기서는 '학번'이 기본키가 되어야 한다.

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

후보키는 릴레이션에 있는 모든 튜플에 대해서 유일성과 최소성을 만족시켜야 한다.

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

- ① 후보키 ② 슈퍼키
- ③ 기본키 ④ 외래키

유일성은 만족시키지만 최소성은 만족시키지 못하는 키는 슈퍼키이다.

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

한 학과에 여러 명의 교수가 있을 수 있지만 한 교수가 여러 학과에 소속될 수는 없다. 그러므로 문제의 <교수>와 <학과> 릴레이션은 <교수>가 <학과>를 참조하는 N:1의 관계이다.

<학과> 릴레이션의 '학과번호'는 기본키이고, <학과>를 참조하는 <교수> 릴레이션의 '학과번호'는 외래키가 된다.

1. 논리 데이터베이스 설계-SEC_11(관계형 데이터베이스의 제약 조건 - 무결성)

1) 무결성(Integrity)의 개념 및 종류

; 무결성이란 데이터베이스에 저장된 데이터 값과 그것이 표현하는 현실 세계의 실제 값이 일치하는 정확성을 의미한다.

- 무결성 제약 조건은 데이터베이스에 들어 있는 데이터의 정확성을 보장하기 위해 부정확한 자료가 데이터베이스 내에 저장되는 것을 방지하기 위한 제약 조건을 말한다.
- 무결성의 종류에는 개체 무결성, 도메인 무결성, 참조 무결성, 사용자 정의 무결성 등이 있다.

〈학생〉 릴레이션			〈수강〉 릴레이션	
학번	주민번호	성명	학번	과목명
1001	010429-3*****	김상욱	1001	영어
1002	000504-3*****	임선호	1001	전산
1003	011215-3*****	김한순	1002	영어
1004	001225-4*****	이다해	1003	수학
			1004	영어
			1004	전산

무결성이란 쉽게 말해 저장된 데이터베이스에 잘못된 데이터가 없다는 것을 의미한다. 보기로 주어진 <학생> 릴레이션과 <수강> 릴레이션은 '학번'을 기준으로 일대다(1:N)의 관계를 맺고 있다.
<학생> 릴레이션과 <수강> 릴레이션을 통해 무결성 종류를 확실하게 이해하자.

1. 논리 데이터베이스 설계-SEC_11(관계형 데이터베이스의 제약 조건 - 무결성)

2) 개체 무결성(Entity Integrity, 실체 무결성)

; 개체 무결성은 기본 테이블의 기본키를 구성하는 어떤 속성도 Null 값이나 중복 값을 가질 수 없다는 규정이다.

예) <학생> 릴레이션에서 '학번'이 기본키로 정의되면 튜플을 추가할 때 '주민번호'나 '성명' 필드에는 값을 입력하지 않아도 되지만 '학번' 속성에는 반드시 값을 입력해야 한다. 또한 '학번' 속성에는 이미 한 번 입력한 속성 값을 중복하여 입력할 수 없다.

3) 도메인 무결성(Domain Integrity, 영역 무결성)

; 도메인 무결성은 주어진 속성 값이 정의된 도메인에 속한 값이어야 한다는 규정이다.

예) <수강> 릴레이션의 '과목명' 속성에는 영어, 수학, 전산 세 가지만 입력되도록 유효값이 지정된 경우 반드시 해당 값만 입력해야 한다.

기본키(Primary Key) : 기본키는 한 릴레이션에서 특정 튜플을 유일하게 구별할 수 있는 속성이다.
도메인 : 하나의 애트리뷰트가 취할 수 있는 같은 타입의 원자(Atomic)값들의 집합이다.

1. 논리 데이터베이스 설계-SEC_11(관계형 데이터베이스의 제약 조건 - 무결성)

4) 참조 무결성(Referential Integrity)

; 참조 무결성은 외래키 값은 Null이거나 참조 릴레이션의 기본키 값과 동일해야 한다. 즉 릴레이션은 참조할 수 없는 외래키 값을 가질 수 없다는 규정이다.

● 외래키와 참조하려는 테이블의 기본키는 도메인과 속성 개수가 같아야 한다.

예) <수강> 릴레이션의 '학번'속성에는 <학생> 릴레이션의 '학번' 속성에 없는 값은 입력할 수 없다.

예) <수강> 릴레이션의 '학번'과 <학생> 릴레이션의 '학번'속성에는 같은 종류의 데이터가 입력되어 있어야 하며, <학생> 릴레이션의 기본키가 '학번'+'이름'이었다면 <수강> 릴레이션의 외래키도 '학번'+'이름'으로 구성되어야 한다.

5) 사용자 정의 무결성

; 사용자 정의 무결성(User-Defined Integrity)은 속성 값들이 사용자가 정의한 제약 조건에 만족해야 한다는 규정이다.

외래키 : 다른 릴레이션의 기본키를 참조하는 속성 또는 속성들의 집합을 의미한다.

1. 논리 데이터베이스 설계-SEC_11(관계형 데이터베이스의 제약 조건 - 무결성)

6) 데이터 무결성 강화

; 데이터 무결성은 데이터 품질에 직접적인 영향을 미치므로 데이터 특성에 맞는 적절한 무결성을 정의하고 강화해야 한다.

- 프로그램이 완성되고 데이터가 저장된 상태에서 무결성을 정의할 경우 많은 비용이 발생하므로 데이터베이스 구축 과정에서 정의한다.
- 데이터 무결성은 애플리케이션, 데이터베이스 트리거, 제약 조건을 이용하여 강화할 수 있다.

① 애플리케이션

- 데이터 생성, 수정, 삭제 시 무결성 조건을 검증하는 코드를 데이터를 조작하는 프로그램 내에 추가한다.
- 코드를 이용한 복잡한 규칙 등을 검토하는 무결성 검사는 데이터베이스에서 수행하기 어려우므로 애플리케이션 내에서 처리한다.
 - 장점 : 사용자 정의 같은 복잡한 무결성 조건의 구현이 가능하다.
 - 단점 : 소스 코드에 분산되어 있어 관리가 힘들고, 개별적인 시행으로 인해 적정성 검토가 어렵다.

1. 논리 데이터베이스 설계-SEC_11(관계형 데이터베이스의 제약 조건 - 무결성)

6) 데이터 무결성 강화

② 데이터베이스 트리거

- 트리거 이벤트에 무결성 조건을 실행하는 절차형 SQL을 추가한다.
- 장점 : 통합 관리가 가능하고, 복잡한 요구 조건의 구현이 가능하다.
- 단점 : 운영 중 변경이 어렵고, 사용상 주의가 필요하다.

③ 제약 조건

- 데이터베이스에 제약 조건을 설정하여 무결성을 유지한다.
- 장점 : 통합 관리 가능, 간단한 선언으로 구현 가능, 변경 용이, 오류 데이터 발생 방지 등이 있다.
- 단점 : 복잡한 제약 조건의 구현과 예외적인 처리가 불가능하다.

트리거(Trigger) : 데이터베이스 시스템에서 데이터의 입력, 갱신, 삭제 등의 이벤트(Event)가 발생할 때마다 자동적으로 수행되는 절차형 SQL이다.

이벤트(Event) : 이벤트는 시스템에 어떤 일이 발생한 것을 말하며, 트리거에서 이벤트는 데이터의 입력, 갱신, 삭제와 같이 데이터를 조작하는 작업이 발생했음을 의미한다.

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.

개체 무결성(Entity Integrity, 실체 무결성)

개체 무결성은 기본 테이블의 기본키를 구성하는 어떤 속성도 Null값이나 중복 값을 가질 수 없다는 구정이다.

도메인 무결성(Domain Integrity, 영역 무결성)

도메인 무결성은 주어진 속성 값이 정의된 도메인에 속한 값이어야 한다.

참조 무결성(Referential Integrity)

참조 무결성은 외래키 값은 Null이거나 참조 릴레이션의 기본키 값과 동일해야 한다. 즉, 릴레이션은 참조할 수 없는 외래키 값을

3. 데이터 무결성과 가장 관계가 깊은 것은?

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

무결성은 결점이 없다는 뜻이다. 즉, 데이터가 정확하다라는 뜻이다.

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.

도메인 무결성 규정은 주어진 튜플의 값에 대한 규정이 아니라 튜플을 구성하는 속성의 값이 그 속성의 정의된 도메인에 속한 값이어야 한다는 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)

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

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

외래키는 참조되는 릴레이션의 기본키와 대응되는 키라는 것을 다시 기억하자.

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

인덱스는 검색속도나 성능을 향상시키는 도구이다.

데이터 무결성은 애플리케이션, 데이터베이스 트리거, 제약 조건을 이용하여 강화할 수 있다.

① 애플리케이션

- 데이터 생성, 수정, 삭제 시 무결성 조건을 검증하는 코드를 데이터를 조작하는 프로그램 내에 추가한다.

- 코드를 이용한 복잡한 규칙 등을 검토하는 무결성 검사는 데이터베이스에서 수행하기 어려우므로 애플리케이션 내에서 처리한다.

- 장점 : 사용자 정의 같은 복잡한 무결성 조건의 구현이 가능하다.

- 단점 : 소스 코드 분산되어 있어 관리가 힘들고, 개별적인 시행으로 인해 적정성 검토가 어렵다.

② 데이터베이스 트리거

- 트리거 이벤트에 무결성 조건을 실행하는 절차형 SQL을 추가한다.

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

개체 무결성은 기본 테이블의 기본키를 구성하는 어떤 속성도 Null 값이나 중복된 값을 가질 수 없다는 규정이다.

1. 논리 데이터베이스 설계-SEC_12(관계대수 및 관계해석)

1) 관계대수의 개요

- ; 관계대수는 관계형 데이터베이스에서 원하는 정보와 그 정보를 검색하기 위해서 어떻게 유도하는가를 기술하는 절차적인 언어이다.
- 관계대수는 릴레이션을 처리하기 위해 연산자와 연산규칙을 제공하는 언어로 피연산자가 릴레이션이고, 결과도 릴레이션이다.
- 질의에 대한 해를 구하기 위해 수행해야 할 연산의 순서를 명시한다.
- 관계대수에는 관계 데이터베이스에 적용하기 위해 특별히 개발한 순수 관계 연산자와 수학적 집합 이론에서 사용하는 일반 집합 연산자가 있다.
- 순수 관계 연산자 : Select, Project, Join, Division
- 일반 집합 연산자 : UNION(합집합), INTERSECTION(교집합), DIFFERENCE(차집합), CARTESIAN PRODUCT(교차곱)

1. 논리 데이터베이스 설계-SEC_12(관계대수 및 관계해석)

2) Select

; Select는 릴레이션에 존재하는 튜플 중에서 선택 조건을 만족하는 튜플의 부분집합을 구하여 새로운 릴레이션을 만드는 연산이다.

- 릴레이션의 행(가로)에 해당하는 튜플을 구하는 것이므로 수평 연산이라고도 한다.
- 연산자의 기호는 그리스 문자 **시그마(σ)**를 사용한다.
- 표기 형식 : $\sigma_{\langle \text{조건} \rangle}(R)$

- R은 릴레이션 이름이다.

- 조건에서는 =, \neq , <, \leq , >, \geq 그 등의 기호를 사용한 비교 연산이 허용되며, AND(\wedge), OR(\vee), NOT(\neg) 등의 논리 연산자를 사용하여 여러 개의 조건들을 하나의 조건으로 결합시킬 수도 있다.

예제) $\sigma_{\text{Avg} \geq 90}$ (성적) : <성적> 릴레이션에서 '평균'(Avg)이 90점 이상인 튜플들을 추출하시오.

〈성적〉					
Name	Kor	Eng	Mat	Tot	Avg
고희식	100	90	100	290	96.6
김은소	80	80	100	260	86.6
최미경	100	70	80	250	83.3
김준용	90	100	90	280	93.3
윤정희	85	95	90	270	90.0



Name	Kor	Eng	Mat	Tot	Avg
고희식	100	90	100	290	96.6
김준용	90	100	90	280	93.3
윤정희	85	95	90	270	90.0

1. 논리 데이터베이스 설계-SEC_12(관계대수 및 관계해석)

3) Project

; Project는 주어진 릴레이션에서 속성 리스트(Attribute List)에 제시된 속성 값만을 추출하여 새로운 릴레이션을 만드는 연산이다. 단 연산 결과에 중복이 발생하면 중복이 제거된다.

- 릴레이션의 열(세로)에 해당하는 Attribute를 추출하는 것이므로 수직 연산자라고도 한다.
- 연산자의 기호는 그리스 문자 파이(π)를 사용한다.
- 표기 형식 : $\pi_{\langle \text{속성리스트} \rangle}(R)$
 - R은 릴레이션 이름이다.

예제) $\pi_{\text{Name, Avg}}$ (성적) : <성적> 릴레이션에서 'Name'과 'Avg' 속성을 추출하시오.

〈성적〉					
Name	Kor	Eng	Mat	Tot	Avg
고희식	100	90	100	290	96.6
김은소	80	80	100	260	86.6
최미경	100	70	80	250	83.3
김준용	90	100	90	280	93.3
윤정희	85	95	90	270	90.0



Name	Avg
고희식	96.6
김은소	86.6
최미경	83.3
김준용	93.3
윤정희	90.0

1. 논리 데이터베이스 설계-SEC_12(관계대수 및 관계해석)

4) Join

; Join은 공통 속성을 중심으로 두 개의 릴레이션을 하나로 합쳐서 새로운 릴레이션을 만드는 연산이다.

- Join의 결과로 만들어진 릴레이션의 차수는 조인된 두 릴레이션의 차수를 합한 것과 같다.
- Join의 결과는 Cartesian Product(교차곱)를 수행한 다음 Select를 수행한 것과 같다.
- 연산자의 기호는 나비 넥타이(\bowtie)를 사용한다.
- 표기 형식 : $R \bowtie_{\text{키속성}_r = \text{키속성}_s} S$
 - 키 속성 r 은 릴레이션 R 의 속성이고, 키 속성 s 는 릴레이션 S 의 속성이다 .

Cartesian Product(교차곱)의 연산은 두 릴레이션에 존재하는 모든 튜플들을 대응시켜 새로운 릴레이션을 만드는 연산으로 연산의 결과 차수는 두 릴레이션의 차수를 합한 것과 같고 튜플은 두 릴레이션의 튜플 수를 곱한 것과 같다. 즉, Cartesian Product의 결과는 두 릴레이션을 연결하여 나타낼 수 있는 모든 튜플들을 표현할 수 있으므로 여기 에서 필요한 튜플만 선별하는 Select 연산을 수행하면 Join 연산의 결과와 같아지는 것이다.

1. 논리 데이터베이스 설계-SEC_12(관계대수 및 관계해석)

4) Join

예제) 성적 $\bowtie_{No = No}$ 학적부 : <성적> 릴레이션과 <학적부> 릴레이션을 'No' 속성을 기준으로 합치시오.

〈성적〉							〈학적부〉	
No	Name	Kor	Eng	Mat	Tot	Avg	No	Addr
9801	고희식	100	90	100	290	96.6	9801	망원동
9802	김은소	80	80	100	260	86.6	9802	서교동
9803	최미경	100	70	80	250	83.39	9803	성산동
9804	김준용	90	100	90	280	93.3	9804	합정동
9805	윤정희	85	95	90	270	90.0	9805	공덕동

성적.No	Name	Kor	Eng	Mat	Tot	Avg	학적부.No	Addr
9801	고희식	100	90	100	290	96.6	9801	망원동
9802	김은소	80	80	100	260	86.6	9802	서교동
9803	최미경	100	70	80	250	83.39	9803	성산동
9804	김준용	90	100	90	280	93.3	9804	합정동
9805	윤정희	85	95	90	270	90.0	9805	공덕동

위 그림에서와 같이 Join 조건이 '='일 때 동일한 속성이 2번 나타나게 된다.

1. 논리 데이터베이스 설계-SEC_12(관계대수 및 관계해석)

5) 자연 조인(Natural Join)

- 조인 조건이 '='일 때 동일한 속성이 두 번 나타나게 되는데, 이중 중복된 속성을 제거하여 같은 속성을 한 번만 표기하는 방법을 자연(Natural) 조인이라고 한다.
- 예를 들어, 앞의 예제의 Join 연산 결과에는 조인에 사용된 'No' 속성이 두 번 표기되었는데 이것은 의미상 아무런 도움을 주지 않는다. 이런 경우 다음과 같은 자연 조인 연산을 통해 'No' 속성이 한 번만 표기되게 할 수 있다.

성적 \bowtie_{No} 학적부

- 자연 조인이 성립하려면 두 릴레이션의 속성명과 도메인이 같아야 한다.

1. 논리 데이터베이스 설계-SEC_12(관계대수 및 관계해석)

6) Division

; Division은 $X \supset Y$ 인 두 개의 릴레이션 $R(X)$ 와 $S(Y)$ 가 있을 때, R 의 속성이 S 의 속성 값을 모두 가진 튜플에서 S 가 가진 속성을 제외한 속성만을 구하는 연산이다.

- 연산자의 기호는 나누기 기호 \div 를 사용한다.

- 표기형식: $R [\text{속성}r \div \text{속성}s] S$

- 속성 r 은 릴레이션 R 의 속성, 속성 s 는 릴레이션 S 의 속성, 속성 r 과 속성 s 는 **동일 속성값**을 가지는 속성이어야 한다.

1. 논리 데이터베이스 설계-SEC_12(관계대수 및 관계해석)

6) Division

〈구입자〉			〈생산품1〉		〈생산품2〉	
번호	이름	구입품코드	생산품코드	이름	생산품코드	
1	고희식	A	A	김준용	A	
1	고희식	B	B	김준용	B	
2	김준용	A				
2	김준용	B				
2	김준용	C				
4	윤정희	C				

예제1) 구입자[구입품코드 ÷ 생산품코드] 생산품1의 결과는 다음과 같다.

번호	이름
1	고희식
2	김준용

설명 : <구입자> 릴레이션에서 '구입품코드' 속성을 제외하고 '번호'와 '이름'을 추출한다.

단, <생산품1> 릴레이션의 모든 튜플과 연관되어 있는 튜플만 추출한다.

예제2) 구입자[이름, 구입품코드 ÷ 이름, 생산품코드] 생산품2의 결과는 다음과 같다.

번호
2

설명 : <구입자> 릴레이션에서 '이름', '구입품코드' 속성을 제외하고 번호만 추출한다.

단, <생산품2> 릴레이션의 모든 튜플과 연관되어 있는 튜플만 추출한다.

1. 논리 데이터베이스 설계-SEC_12(관계대수 및 관계해석)

7) 일반 집합 연산자

; 일반 집합 연산자는 수학적 집합 이론에서 사용하는 연산자로서 릴레이션 연산에도 그대로 적용할 수 있다.

- 일반 집합 연산자 중 합집합(UNION), 교집합(INTERSECTION), 차집합(DIFFERENCE)을 처리하기 위해서는 합병 조건을 만족해야 한다.
- 합병 가능한 두 릴레이션 R과 S가 있을 때 각 연산의 특징을 요약하면 다음과 같다.

합병 조건
합병 조건은 합병하려는 두 릴레이션 간에 속성의 수가 같고 대응되는 속성별로 도메인이 같아야 한다. 즉, 릴레이션 R과 S가 합병이 가능하다면, 릴레이션 R의 1번째 속성과 릴레이션 S의 1번째 속성의 도메인이 서로 같아야 한다. 그러나 속성의 이름이 같아야 되는 것은 아니다.

교차곱은 두 릴레이션의 **차수(Degree, 속성의 수)**는 더하고, **카디널리티(튜플의 수)**는 곱하면 된다는 것을 꼭 기억하자.

연산자	기능 및 수학적 표현	카디널리티
합집합 UNION \cup	<ul style="list-style-type: none">• 두 릴레이션에 존재하는 튜플의 합집합을 구하되, 결과로 생성된 릴레이션에서 중복되는 튜플은 제거되는 연산이다.• $R \cup S = \{t \mid t \in R \vee t \in S\}$※ t는 릴레이션 R 또는 S에 존재하는 튜플이다.	<ul style="list-style-type: none">• $R \cup S \leq R + S$• 합집합의 카디널리티는 두 릴레이션 카디널리티의 합보다 크지 않다.
교집합 INTERSECTION \cap	<ul style="list-style-type: none">• 두 릴레이션에 존재하는 튜플의 교집합을 구하는 연산이다.• $R \cap S = \{t \mid t \in R \wedge t \in S\}$※ t는 릴레이션 R 또는 S에 존재하는 튜플이다.	<ul style="list-style-type: none">• $R \cap S \leq \min\{ R , S \}$• 교집합의 카디널리티는 두 릴레이션 중 카디널리티가 적은 릴레이션의 카디널리티보다 크지 않다.
차집합 DIFFERENCE $-$	<ul style="list-style-type: none">• 두 릴레이션에 존재하는 튜플의 차집합을 구하는 연산이다.• $R - S = \{t \mid t \in R \wedge t \notin S\}$※ t는 릴레이션 R 또는 S에 존재하는 튜플이다.	<ul style="list-style-type: none">• $R - S \leq R$• 차집합의 카디널리티의 릴레이션 R의 카디널리티보다 크지 않다.
교차곱 CARTESIAN PRODUCT \times	<ul style="list-style-type: none">• 두 릴레이션에 있는 튜플의 순서쌍을 구하는 연산이다.• $R \times S = \{r, s \mid r \in R \wedge s \in S\}$※ r은 R에 존재하는 튜플이고, s는 S에 존재하는 튜플이다.	<ul style="list-style-type: none">• $R \times S = R \times S$• 교차곱의 디그리는 두 릴레이션의 디그리를 더한 것과 같고, 카디널리티는 곱한 것과 같다.

1. 논리 데이터베이스 설계-SEC_12(관계대수 및 관계해석)

7) 일반 집합 연산자

〈사원〉		〈직원〉	
사번	이름	사번	이름
1	고희식	2	김준용
2	김준용	3	윤정희

예제 1) $\pi_{\text{이름}}(\text{사원}) \cup \pi_{\text{이름}}(\text{직원})$: 〈사원〉 릴레이션과 〈직원〉 릴레이션에서 '이름'을 추출한 것의 **합집합**을 구한다.

이름
고희식
김준용
윤정희

예제 2) $\pi_{\text{이름}}(\text{사원}) \cap \pi_{\text{이름}}(\text{직원})$: 〈사원〉 릴레이션과 〈직원〉 릴레이션에서 '이름'을 추출한 것의 **교집합**을 구한다.

이름
김준용

1. 논리 데이터베이스 설계-SEC_12(관계대수 및 관계해석)

7) 일반 집합 연산자

〈사원〉		〈직원〉	
사번	이름	사번	이름
1	고희식	2	김준용
2	김준용	3	윤정희

예제3) $\pi_{\text{이름}}(\text{사원}) - \pi_{\text{이름}}(\text{직원})$: <사원> 릴레이션과 <직원> 릴레이션에서 '이름'을 추출한 것의 **차집합**을 구한다.

이름
고희식

예제4) $\pi_{\text{이름}}(\text{사원}) \times \pi_{\text{이름}}(\text{직원})$: <사원> 릴레이션과 <직원> 릴레이션에서 '이름'을 추출한 것의 **교차곱**을 구한다.

사원.이름	직원.이름
고희식	김준용
고희식	윤정희
김준용	김준용
김준용	윤정희

1. 논리 데이터베이스 설계-SEC_12(관계대수 및 관계해석)

8) 관계해석(Relational Calculus)

; 관계해석은 관계 데이터 모델의 제안자인 코드(E. F. Codd)가 수학의 Predicate Calculus(술어 해석)에 기반을 두고 관계 데이터베이스를 위해 제안했다.

- 관계해석은 관계 데이터의 연산을 표현하는 방법으로, 원하는 정보를 정의할 때는 계산 수식을 사용한다.
- 관계해석은 원하는 정보가 무엇이라는 것만 정의하는 비절차적 특성을 지닌다.
- 튜플 관계해석과 도메인 관계해석이 있다.
- 기본적으로 관계해석과 관계대수는 관계 데이터베이스를 처리하는 기능과 능력면에서 동등하며, 관계대수로 표현한 식은 관계해석으로 표현할 수 있다.
- 질의어로 표현한다
- 주요 논리기호

기호	구성 요소	설명
\forall	전칭 정량자	가능한 모든 튜플에 대하여(For All)
\exists	존재 정량자	하나라도 일치하는 튜플이 있음(There Exists)

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

순수 관계 연산자 : Select, Project, Join, Division

일반 집합 연산자 : UNION(합집합), INTERSECTION(교집합),
DIFFERENCE(차집합), CATESIAN PRODUCT(교차곱)

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

각 관계대수 연산자의 특징을 확실하게 구분할 수 있어야 한다.

Join은 공통 속성을 중심으로 두 개의 릴레이션을 하나로 합쳐서 새로운 릴레이션을 만드는 것이다.

- Join의 결과로 만들어진 릴레이션의 차수는 조인된 두 릴레이션의 차수를 합한 것과 같다.

- Join의 결과는 카테시안 곱을 수행한 다음 Select를 수행한 것과

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

수평은 Select, 수직은 Project 기억하자.

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

카티션 프로덕트, 즉 교차곱은 두 릴레이션의 차수(Degree, 속성의 수)는 더하고, 카디널리티(튜플의 수)는 곱하면 된다.

즉 차수는 $4 + 6 = 10$ 이고, 카디널리티는 $5 * 7 = 35$ 이다.

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.

관계대수는 관계형 데이터베이스에서 원하는 정보와 그 정보를 검색하기 위해서 어떻게 유도하는가를 기술하는 절차적인 언어이다.

- 관계대수는 릴레이션을 처리하기 위해 연산자와 연산규칙을 제공하는 언어로 피연산자가 릴레이션이고, 결과도 릴레이션이다.
- 질의에 대한 해를 구하기 위해 수행해야 할 연산의 순서를 명시한다.
- 관계대수에는 관계 데이터베이스에 적용하기 위해 특별히 개발한 순수 관계 연산자와 수학적 집합 이론에서 사용하는 일반 집합 연산자가 있다.

7. 관계대수 식을 SQL 질의로 옳게 표현한 것은?

$$\pi_{\text{이름}}(\sigma_{\text{학과} = \text{'교육'}}(\text{학생}))$$

- 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';

파이(이름) : '이름' 필드를 표시하므로 Select 이름 이 되어야 한다.
시그마(학과 = '교육') : '학과'가 '교육'인 자료만을 대상으로 검색하므로 Where 학과 = 교육; 이다.
(학생) : <학생>테이블의 자료를 검색하기에 From 학생이 된다.

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.

관계해석은 관계 데이터 모델의 제안자인 코드가 수학의 Predicate

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.

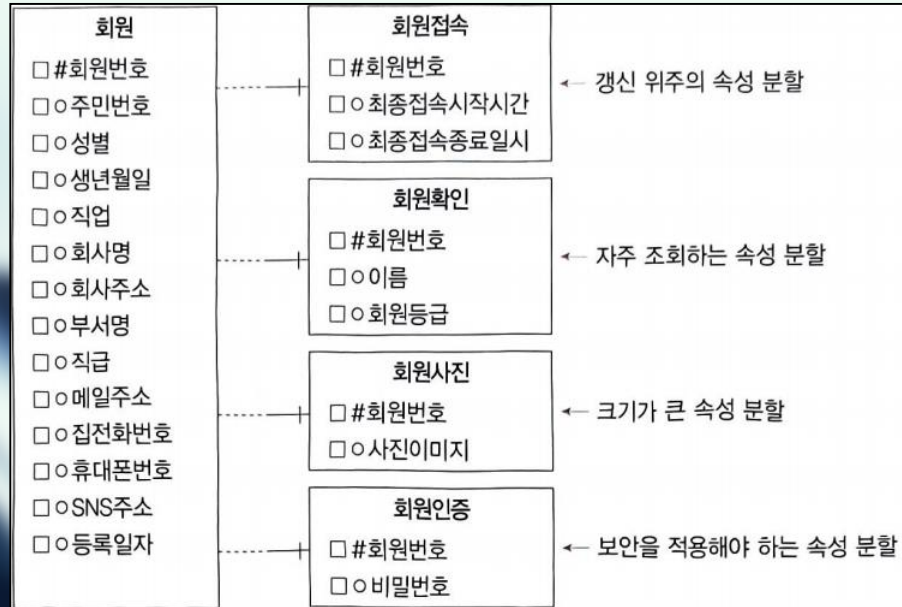
● 수직 분할(Vertical Partitioning)

- 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

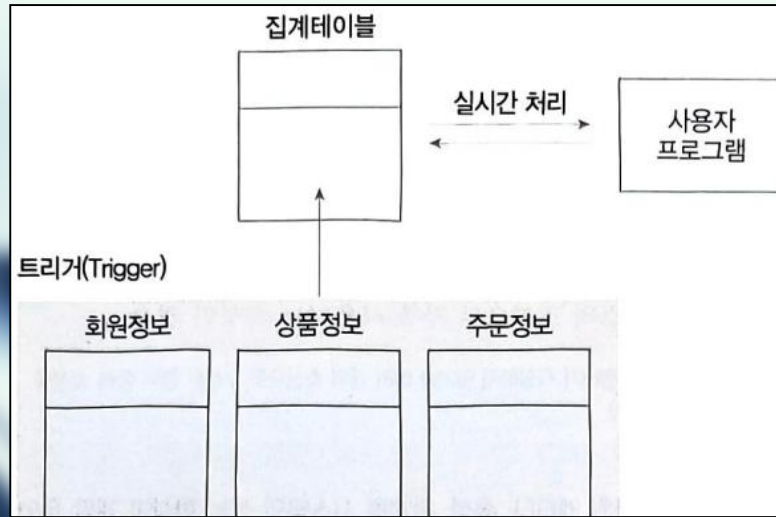
- 정규화로 인해 수행 속도가 느려지는 경우
- 많은 범위의 데이터를 자주 처리해야 하는 경우
- 특정 범위의 데이터만 자주 처리해야 하는 경우
- 처리 범위를 줄이지 않고는 수행 속도를 개선할 수 없는 경우

● 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>

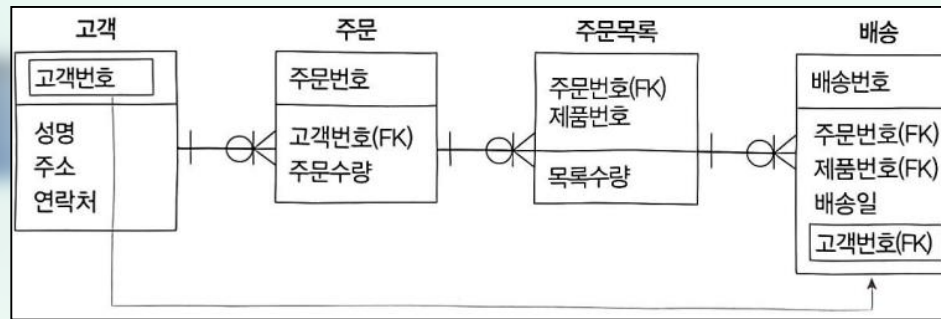
위의 이미지를 보면 <회원정보>, <상품정보>, <주문정보> 테이블의 데이터를 집계하는 집계 테이블을 추가하여 시스템 사용이 적은 시간에 배치 작업에 의해 원하는 데이터를 생성하여 사용하는 것이다.

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 정의어 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

A close-up, low-angle shot of a white car's side mirror and door handle against a light blue sky. The car is on the left side of the frame, with the mirror and handle clearly visible. The sky is a uniform light blue, occupying the right side of the frame. A semi-transparent dark blue horizontal bar is positioned across the middle of the image, containing the Korean text '감사합니다.' in white.

감사합니다.