

3과목-데이터베이스 구축

(Part 1. 논리 데이터베이스 설계 - I)

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_01(데이터베이스 설계)

- 이 장을 공부하면서 반드시 알아두어야 할 키워드

; 개체, 속성, 관계, 튜플, 도메인, 키, 무결성, 관계대수, 정규화, 시스템 카탈로그

1) 데이터베이스 설계의 개념

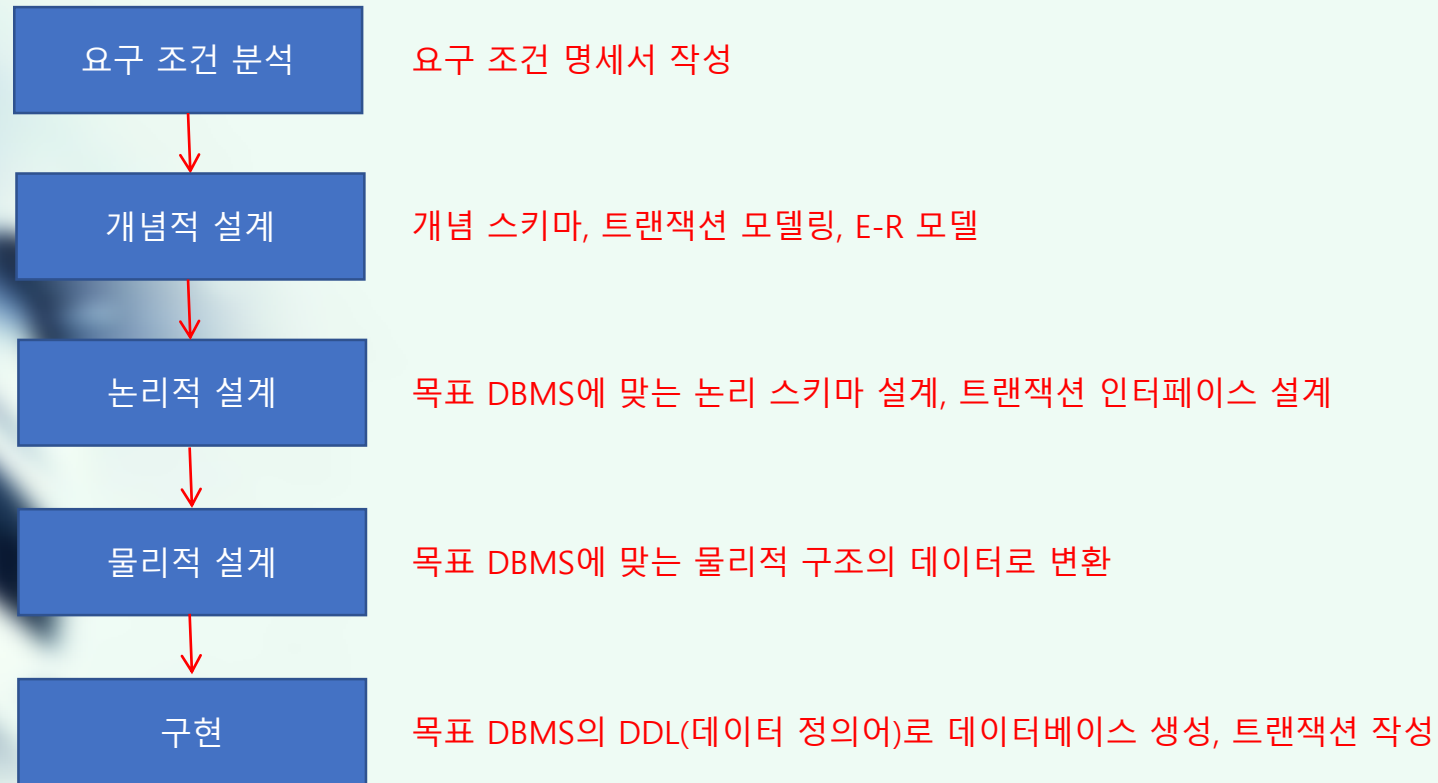
; 데이터베이스 설계란 사용자의 요구를 분석하여 그것들을 컴퓨터에 저장할 수 있는 데이터베이스의 구조에 맞게 변형한 후 특정 DBMS로 데이터베이스를 구현하여 일반 사용자들이 사용하게 하는 것이다.

2) 데이터베이스 설계 시 고려사항

- **무결성** : 삽입, 삭제, 갱신 등의 연산 후에도 데이터베이스에 저장된 데이터가 정해진 제약 조건을 항상 만족해야 한다.
- **일관성** : 데이터베이스에 저장된 데이터들 사이나, 특정 질의에 대한 응답이 처음부터 끝까지 변함 없이 일정해야 한다.
- **회복** : 시스템에 장애가 발생했을 때 장애 발생 직전의 상태로 복구할 수 있어야 한다.
- **보안** : 불법적인 데이터의 노출 또는 변경이나 손실로부터 보호할 수 있어야 한다.
- **효율성** : 응답시간의 단축, 시스템의 생산성, 저장 공간의 최적화 등이 가능해야 한다.
- **데이터베이스 확장** : 데이터베이스 운영에 영향을 주지 않으면서 지속적으로 데이터를 추가할 수 있어야 한다.

1. 논리 데이터베이스 설계-SEC_01(데이터베이스 설계)

3) 데이터베이스 설계 순서



스키마 : 데이터베이스의 구조와 제약조건에 관해 전반적인 명세를 기술한 것이다.
상세하게 말하면, 개체의 특성을 나타내는 속성(Attribute)과 속성들의 집합으로 이루어진 개체(Entity), 개체 사이에 존재하는 관계(Relation)에 대한 정의와 이들이 유지해야 할 제약조건들을 기술한 것이다.

E-R 모델(Entity-Relationship model) : 개념적 데이터 모델의 대표적인 모델로 개체와 개체 간의 관계를 이용하여 현실 세계를 개념적 구조로 표현하는 방법이다.

1. 논리 데이터베이스 설계-SEC_01(데이터베이스 설계)

4) 요구 조건 분석

; 요구 조건 분석은 데이터베이스를 사용할 사람으로부터 필요한 용도를 파악하는 것이다.

- 데이터베이스 사용자에게 따른 수행 업무와 필요한 데이터의 종류, 용도, 처리 형태, 흐름, 제약 조건 등을 수집한다.
- 수집된 정보를 바탕으로 요구 조건 명세를 작성한다.

5) 개념적 설계(정보 모델링, 개념화)

; 개념적 설계란 정보의 구조를 얻기 위하여 현실 세계의 무한성과 계속성을 이해하고, 다른 사람과 통신하기 위하여 현실 세계에 대한 인식을 추상적 개념으로 표현하는 과정이다.

- 개념적 설계 단계에서는 개념 스키마 모델링과 트랜잭션 모델링을 병행 수행한다.
- 개념적 설계 단계에서는 요구 분석 단계에서 나온 결과인 요구 조건 명세를 DBMS에 독립적인 E-R 다이어그램으로 작성한다.
- DBMS에 독립적인 개념 스키마를 설계한다.

개념 스키마(Conceptual Schema) : 데이터베이스의 전체 조직에 대한 논리적인 구조로, 물리적인 구현은 고려하지 않는다. 각 데이터베이스당 한 개의 개념 스키마만 존재한다. 개체 간의 관계 및 무결성 제약 조건에 대한 명세를 정의한다. 단순한 스키마라고 하면 개념스키마를 의미한다.

무결성 제약 조건 : 데이터베이스의 정확성, 일관성을 보장하기 위해 저장, 삭제, 수정 등을 제약하기 위한 조건
데이터베이스 파일에 저장되는 데이터의 형태를 나타낸다. (DBA에 의해 구성)

트랜잭션 모델링 : 처리 중심 설계

1. 논리 데이터베이스 설계-SEC_01(데이터베이스 설계)

6) 논리적 설계(데이터 모델링)

; 논리적 설계 단계란 현실 세계에서 발생하는 자료를 컴퓨터가 이해하고 처리할 수 있는 물리적 저장 장치에 저장할 수 있도록 변환하기 위해 특정 DBMS가 지원하는 논리적 자료 구조로 변환(mapping)시키는 과정이다.

- 개념 세계의 데이터를 필드로 기술된 데이터 타입과 이 데이터 타입들 간의 관계로 표현되는 논리적 구조의 데이터로 모델화한다.
- 개념적 설계가 개념 스키마를 설계하는 단계라면 논리적 설계에서는 개념 스키마를 평가 및 정제하고 DBMS에 따라 서로 다른 논리적 스키마를 설계하는 단계이다.
- 트랜잭션의 인터페이스를 설계한다.
- 관계형 데이터베이스라면 테이블을 설계하는 단계이다.

트랜잭션(transaction)이란 "쪼갤 수 없는 업무 처리의 최소 단위"를 말한다. 거래내역이라고도 한다. 하지만, 데이터베이스에서는 하나의 논리적인 작업 단위를 구성하는 연산들의 집합을 트랜잭션이라 한다.

1. 논리 데이터베이스 설계-SEC_01(데이터베이스 설계)

7) 물리적 설계(데이터 구조화)

; 물리적 설계란 논리적 설계 단계에서 논리적 구조로 표현된 데이터를 디스크 등의 물리적 저장장치에 저장할 수 있는 물리적 구조의 데이터로 변환하는 과정이다.

- 물리적 설계 단계에서는 다양한 데이터베이스 응용에 대해 처리 성능을 얻기 위해 데이터베이스 파일의 저장 구조 및 액세스 경로를 결정한다.
- 저장 레코드의 양식, 순서, 접근 경로, 조회가 집중되는 레코드와 같은 정보를 사용하여 데이터가 컴퓨터에 저장되는 방법을 묘사한다.
- 물리적 설계 시 고려할 사항 : 트랜잭션 처리량, 응답 시간, 디스크 용량, 저장 공간의 효율화 등

저장 레코드 양식 : 물리적 데이터베이스 구조의 기본 데이터 단위인 저장 레코드의 양식을 설계할 때는 데이터 타입, 데이터 값의 분포, 접근 빈도 등을 고려해야 한다.

1. 논리 데이터베이스 설계-SEC_01(데이터베이스 설계)

8) 데이터베이스 구현

; 데이터베이스 구현 단계란 논리적 설계 단계와 물리적 설계 단계에서 도출된 데이터베이스 스키마를 파일로 생성하는 과정이다.

- 사용하려는 특정 DBMS의 DDL(데이터 정의어)을 이용하여 데이터베이스 스키마를 기술한 후 컴파일하여 빈 데이터베이스 파일을 생성한다.
- 생성된 빈 데이터베이스 파일에 데이터를 입력한다.
- 응용 프로그램을 위한 트랜잭션을 작성한다.
- 데이터베이스 접근을 위한 응용 프로그램을 작성한다.

DDL(Data Definition Language)

데이터 정의어란? 데이터베이스를 정의하는 언어이며, 데이터를 생성, 수정, 삭제하는 등의 데이터의 전체의 골격을 결정하는 역할을 하는 언어이다.

create : 데이터베이스, 테이블 등을 생성

alter : 테이블을 수정

drop : 데이터베이스, 테이블을 삭제

truncate : 테이블을 초기화

논리 데이터베이스 설계-SEC_01(데이터베이스 설계) 기출 문제

기출 문제(데이터베이스 설계)

1. 데이터베이스에서 개념적 설계 단계에 대한 설명으로 틀린 것은?

- ① 산출물로 E-R Diagram을 만들 수 있다.
- ② DBMS에 독립적인 개념 스키마를 설계한다.
- ③ 트랜잭션 인터페이스를 설계 및 작성한다.
- ④ 논리적 설계 단계의 앞 단계에서 수행된다.

트랜잭션 인터페이스를 설계 및 작성하는 단계는 논리적 설계에 해당한다.

개념적 설계란 정보의 구조를 얻기 위하여 현실 세계의 무한성과 계속성을 이해하고, 다른 사람과 통신하기 위하여 현실 세계에 대한 인식을 추상적 개념으로 표현하는 과정이다.

- 개념적 설계 단계에서는 개념 스키마 모델링과 트랜잭션 모델링 병행 수행한다.

- 개념적 설계 단계에서는 요구 분석 단계에서 나온 결과인 요구 조건 명세를 DBMS에 독립적인 E-R 다이어그램으로 작성한다.

- DBMS에 독립적인 개념 스키마를 설계한다.

2. 물리적 데이터베이스를 설계하는 전 단계로서, 데이터 모델링 이라 불리는 데이터베이스 설계 단계는?

3. 데이터베이스 설계 단계와 그 단계에서 수행되는 결과의 연결이 잘못된 것은?

- ① 개념적 설계 단계 - 트랜잭션 모델링
- ② 물리적 설계 단계 - 목표 DBMS에 맞는 물리적 구조설계
- ③ 논리적 설계 단계 - 목표 DBMS에 독립적인 논리 스키마 설계
- ④ 구현 단계 - 목표 DBMS DDL로 스키마 작성

논리적 설계 단계는 목표 DBMS에 맞는, 즉 독립적인이 아니라 종속적인 논리 스키마를 설계하는 단계이다.

4. 물리적 데이터베이스 설계에 대한 설명으로 거리가 먼 것은?

- ① 물리적 설계의 목적은 효율적인 방법으로 데이터를 저장하는 것이다.
- ② 트랜잭션 처리량과 응답시간, 디스크 용량 등을 고려해야 한다.
- ③ 저장 레코드의 형식, 순서, 접근 경로와 같은 정보를 사용하여 설계한다.
- ④ 트랜잭션의 인터페이스를 설계하며, 데이터 타입 및 데이터 타입들 간의 관계로 표현한다.

트랜잭션의 인터페이스를 설계하며, 데이터 타입 및 데이터 타입들 간의 관계로 표현하는 것은 논리적 설계 단계에서 수행하는 것이다.

물리적 설계란 논리적 단계에서 논리적 구조로 표현된 데이터를 디스크 등의 물리적 저장장치에 저장할 수 있는 물리적 구조의

논리 데이터베이스설계-SEC_01(데이터베이스 설계) 기출 문제

기출 문제(데이터베이스 설계)

5. 데이터베이스의 논리적 설계(Logical Design) 단계에서 수행 하는 작업이 아닌 것은?

- ① 레코드 집종의 분석 및 설계
- ② 논리적 데이터베이스 구조로 매핑(mapping)
- ③ 트랜잭션 인터페이스 설계
- ④ 스키마의 평가 및 정제

레코드 집종의 분석 및 설계는 물리적 단계에서 수행하는 작업이다.
논리적 설계 단계란 현실 세계에서 발생하는 자료를 컴퓨터가 이해하고 처리할 수 있는 물리적 저장 장치에 저장할 수 있도록 변화하기 위해서 특정 DBMS가 지원하는 논리적 자료 구조로 변환(mapping)시키는 과정이다.

- 개념 세계의 데이터를 필드로 기술된 데이터 타입과 이 데이터 타입들 간의 관계로 표현되는 논리적 구조의 데이터로 모델화한다.
- 개념적 설계가 개념 스키마를 설계하는 단계라면 논리적 설계에서는 개념 스키마를 평가 및 정제하고 DBMS에 따라 서로 다른 논리적 스키마를 설계하는 단계이다.
- 트랜잭션 인터페이스를 설계한다.

7. 데이터베이스 설계 단계 중 물리적 설계 시 고려 사항으로 적절 하지 않은 것은?

- ① 스키마의 평가 및 정제
- ② 응답 시간
- ③ 저장 공간의 효율화
- ④ 트랜잭션 처리량

스키마의 평가 및 정제는 논리적 설계 단계에서 수행한다.

8. 데이터베이스 설계 단계 중 저장 레코드 양식 설계, 레코드 집종의 분석 및 설계, 접근 경로 설계와 관계되는 것은?

- ① 논리적 설계
- ② 요구 조건 분석
- ③ 개념적 설계
- ④ 물리적 설계

‘레코드 양식, 집중 분석, 접근 경로 설계’하면 물리적 설계이다.

논리 데이터베이스설계-SEC_01(데이터베이스 설계) 기출 문제

기출 문제(데이터베이스 설계)

9. 물리적 데이터베이스 구조의 기본 데이터 단위인 저장 레코드 의 양식을 설계할 때 고려 사항이 아닌 것은?

- ① 데이터 타입
- ② 데이터 값의 분포
- ③ 트랜잭션 모델링
- ④ 접근 빈도

트랜잭션 모델링은 개념적 설계 단계에서 수행한다.

저장 레코드 양식 : 물리적 데이터베이스 구조의 기본 데이터 단위인 저장 레코드의 양식을 설계할 때는 데이터 타입, 데이터 값의 분포, 접근 빈도 등을 고려해야 한다.

1. 논리 데이터베이스 설계-SEC_02(데이터 모델의 개념)

1) 데이터 모델의 정의

; 데이터 모델은 현실 세계의 정보들을 컴퓨터에 표현하기 위해서 단순화, 추상화하여 체계적으로 표현한 개념적 모형이다.

- 데이터 모델은 데이터, 데이터의 관계, 데이터의 의미 및 일관성, 제약 조건 등을 기술하기 위한 개념적 도구들의 모임이다.
- 현실 세계를 데이터베이스에 표현하는 중간 과정, 즉 데이터베이스 설계 과정에서 데이터의 구조 (Schema)를 논리적으로 표현하기 위해 사용되는 지능적 도구이다.
- 데이터 모델 구성 요소 : 개체, 속성, 관계
- 데이터 모델 종류 : 개념적 데이터 모델, 논리적 데이터 모델, 물리적 데이터 모델
- 데이터 모델에 표시할 요소 : 구조, 연산, 제약 조건

물리적 데이터 모델은 실제 컴퓨터에 데이터가 저장되는 방법을 정의하는 물리 데이터베이스 설계 과정을 말한다.

1. 논리 데이터베이스 설계-SEC_02(데이터 모델의 개념)

2) 데이터 모델의 구성 요소

- 개체(Entity) : 데이터베이스에 표현하려는 것으로, 사람이 생각하는 개념이나 정보 단위 같은 현실 세계의 대상체이다.
- 속성(Attribute) : 데이터의 가장 작은 논리적 단위로서 파일 구조상의 데이터 항목 또는 데이터 필드에 해당한다.
- 관계(Relationship) : 개체 간의 관계 또는 속성 간의 논리적인 연결을 의미한다.

3) 개념적 데이터 모델

; 개념적 데이터 모델은 현실 세계에 대한 인간의 이해를 돕기 위해 현실 세계에 대한 인식을 추상적 개념으로 표현하는 과정이다.

- 개념적 데이터 모델은 속성들로 기술된 개체 타입과 이 개체 타입들 간의 관계를 이용하여 현실 세계를 표현한다.
- 개념적 데이터 모델은 현실 세계에 존재하는 개체를 인간이 이해할 수 있는 정보 구조로 표현하기 때문에 정보 모델이라고도 한다.
- 대표적인 개념적 데이터 모델로는 E-R 모델이 있다.

1. 논리 데이터베이스 설계-SEC_02(데이터 모델의 개념)

4) 논리적 데이터 모델

; 논리적 데이터 모델은 개념적 모델링 과정에서 얻은 개념적 구조를 컴퓨터가 이해하고 처리할 수 있는 컴퓨터 세계의 환경에 맞도록 변환하는 과정이다.

- 논리적 데이터 모델은 필드로 기술된 데이터 타입과 이 데이터 타입들 간의 관계를 이용하여 현실 세계를 표현한다.
- 단순히 데이터 모델이라고 하면 논리적 데이터 모델을 의미한다.
- 특정 DBMS는 특정 논리적 데이터 모델 하나만 선정하여 사용한다.
- 논리적 데이터 모델은 데이터 간의 관계를 어떻게 표현하느냐에 따라 관계 모델, 계층 모델, 네트워크 모델로 구분한다.

1. 논리 데이터베이스 설계-SEC_02(데이터 모델의 개념)

5) 논리적 데이터 모델의 품질 검증

; 논리 데이터 모델 품질 검증은 완성된 논리 데이터 모델이 기업에 적합한지를 확인하기 위해 품질을 검증하는 것이다.

- 논리 데이터 모델의 품질은 논리 데이터 모델 품질 기준에 따라 개체, 속성, 관계, 식별자, 모델 전반 등에 대하여 검토 체크리스트를 작성하고 체크리스트의 각 항목을 확인하는 방식으로 검증한다.
- 개체 품질 검증 항목 : 단수 명사 여부, 개체의 주 식별자, 개체 간 상호 배타성, 개체의 정규화 여부, 개체 상세 정의, 개체 관리 업무 기능, 개체에 2개 이상의 속성 존재 여부, 개체의 총 길이, 개체 동의어 여부, 개체 분산 요구 등
- 속성 품질 검증 항목 : 단수 명사 여부, 속성의 값 존재 여부 및 개수, 도메인 정의, 반복되는 속성, 그룹화 가능 속성, 주 식별자 및 비 식별자에 의존하는 속성, 다치 종속 속성 등
- 관계 품질 검증 항목 : 관계의 명칭, 2개 이상의 노드와 관계 존재 여부, 노드의 기수성과 선택성, 필수적 관계, 유효한 관계, 중복된 관계, 외부 식별자 존재 여부, 참조 무결성 여부 등

논리 데이터 모델 품질 기준에는 정확성, 완전성, 준거성, 최신성, 일관성, 활용성 등이 있다.

다치 종속 : A, B, C 3개의 속성을 가진 테이블 R에서 어떤 복합 속성(A, C)에 대응하는 B값의 집합이 A값에만 종속 되고 C값에는 무관하면 B는 A에 다치 종속이라 한다.

기수성(Cardinality) : 관계에 참여하는 각 개체에 대하여 가능한 관계의 수를 의미하며, 관계 차수라고도 한다.

예) 1:1, 1:M, M:N

선택성(Optionality) : 관계가 필수인지 선택인지를 표현하는 것을 의미하며, 관계 선택 사양이라고도 한다.

1. 논리 데이터베이스 설계-SEC_02(데이터 모델의 개념)

5) 논리적 데이터 모델의 품질 검증

- 식별자 품질 검증 항목 : 식별자의 명칭, 정의, 구성, 정합성, 크기, 순서 등
- 전반적인 품질 검증 항목 : 주제 영역 구성의 적절성, 데이터 모델 상에 정규화 여부, 다대다 관계 해소 여부, 이력 관리 대상 선정 확인, 이력 관리 방법의 적절성 확인

6) 데이터 모델에 표시할 요소

- 구조(Structure) : 논리적으로 표현된 개체 타입들 간의 관계로서 데이터 구조 및 정적 성질을 표현한다.
- 연산(Operation) : 데이터베이스에 저장된 실제 데이터를 처리하는 작업에 대한 명세로서 데이터베이스를 조작하는 기본 도구이다.
- 제약조건(Constraint): 데이터베이스에 저장될 수 있는 실제 데이터의 논리적인 제약 조건이다.

논리 데이터베이스 설계-SEC_02(데이터 모델의 개념) 기출 문제

기출 문제(데이터 모델의 개념)

1. 데이터 모델(Data Model)의 개념으로 가장 적절한 것은?

- ① 현실 세계의 데이터 구조를 컴퓨터 세계의 데이터 구조로 기술하는 개념적인 도구이다.
- ② 컴퓨터 세계의 데이터 구조를 현실 세계의 데이터 구조로 기술하는 개념적인 도구이다.
- ③ 현실 세계의 특정한 한 부분의 표현이다.
- ④ 가상 세계의 데이터 구조를 현실 세계의 데이터 구조로 기술하는 개념적인 도구이다.

데이터 모델은 현실 세계의 정보들을 컴퓨터에 표현하기 위해서 단순화, 추상화하여 체계적으로 표현한 개념적 모형이다.

- 데이터 모델은 데이터, 데이터의 관계, 데이터의 의미 및 일관성, 제약 조건 등을 기술하기 위한 개념적 도구들의 모임이다.
- 현실 세계를 데이터베이스에 표현하는 중간 과정, 즉 데이터베이스 설계 과정에서 데이터의 구조(Schema)를 논리적으로 표현하기 위해 사용되는 지능적 도구이다.
- 데이터 모델의 구성 요소 : 개체, 속성, 관계
- 데이터 모델의 종류 : 개념적 데이터 모델, 논리적 데이터 모델

3. 데이터의 가장 작은 논리적 단위로서 파일 구조상의 데이터 항목 또는 데이터 필드에 해당하는 것은?

- ① Tuple ② Relation
- ③ Domain ④ Attribute

개체와 속성의 구분을 명확히 이해해야 한다. 개체는 한 개 이상의 속성(Attribute)으로 이루어진다. 그리고 속성은 데이터베이스 구성의 가장 작은 논리적 단위이다.

테이블을 구성하는 데이터들 중 가로(행)로 묶은 데이터 셋을 의미한다. 일반적으로 행(row) 한 객체에 대한 정보를 가지고 있다. 이 또한 관계형 데이터베이스에서는 튜플(Tuple) 또는 레코드(Record)라는 이름으로 칭해진다.

도메인(Domain)은 데이터베이스에서 필드(Field)에 채워질 수 있는 값의 집합을 의미한다. 예를 들어 도메인이 1에서 10사이의 정수인 속성의 필드에 11이나 -1처럼 도메인을 벗어나는 값 또는 "강아지"처럼 아예 자료형이 다른 값이 들어갈 수가 없다.

행, row, tuple, record -> 같은 말

열, column, 속성, attribute -> 같은 말

4. 데이터 모델에 표시할 요소 중 데이터베이스에 표현될 대상으로

논리 데이터베이스 설계-SEC_02(데이터 모델의 개념) 기출 문제

기출 문제(데이터 모델의 개념)

5. 데이터 모델의 구성 요소 중 데이터 구조에 따라 개념 세계나 컴퓨터 세계에서 실제로 표현된 값들을 처리하는 작업을 의미하는 것은?

- ① Relation ② Data Structure
- ③ Constraint ④ Operation

6. 데이터 모델에 대한 다음 설명 중 ()안에 공통으로 들어갈 내용으로 가장 타당한 것은?

데이터 모델은 일반적으로 3가지 구성 요소를 포함하고 있다.
첫째, 논리적으로 표현된 데이터 구조, 둘째, 이 구조에서 허용될 수 있는 (), 셋째, 이 구조와 ()에서의 제약 조건에 대한 명세를 기술한 것이다.

- ① 개체 ② 연산
- ③ 속성 ④ 도메인

1. 논리 데이터베이스 설계-SEC_03(데이터 모델의 구성 요소-개체(Entity))

1) 개체의 정의 및 특징

; 개체(Entity)는 데이터베이스에 표현하려는 것으로, 사람이 생각하는 개념이나 정보 단위 같은 현실 세계의 대상체이다.

- 개체는 실세계에 독립적으로 존재하는 유형, 무형의 정보로서 서로 연관된 몇 개의 속성으로 구성된다.
- 파일 시스템의 레코드에 대응하는 것으로 어떤 정보를 제공하는 역할을 수행한다.
- 영속적(Persistence)으로 존재하는 개체의 집합이다.
- 독립적으로 존재하거나 그 자체로서도 구별이 가능하다.
- 유일한 식별자(Unique Identifier)에 의해 식별이 가능하다.
- 개체는 업무 프로세스에 의해 이용된다.
- 다른 개체와 하나 이상의 관계(Relationship)가 있다.

영속성(persistence)은 데이터를 생성한 프로그램의 실행이 종료되더라도 사라지지 않는 데이터의 특성을 의미한다. 영속성은 파일 시스템, 관계형 데이터베이스 혹은 객체 데이터베이스 등을 활용하여 구현한다.

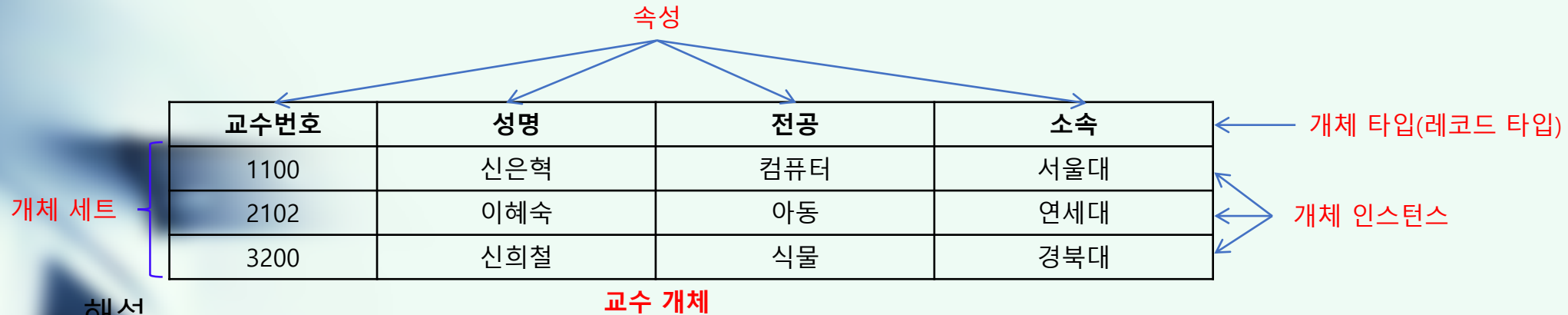
식별자(Identifier)는 하나의 엔터티에 구성되어 있는 여러 개의 속성 중에 엔터티를 대표할 수 있는 속성을 의미한다. 하나의 엔터티 안에는 반드시 유일한 식별자가 존재해야 한다.

유형, 무형의 정보 : 유형의 정보는 물리적으로 존재하는 사람, 자동차, 집 등을 말하고, 무형의 정보는 개념적으로 존재하는 여행, 음악, 취미 등을 말합니다.

1. 논리 데이터베이스 설계-SEC_03(데이터 모델의 구성 요소-개체(Entity))

1) 개체의 정의 및 특징

예제) 다음은 교수번호, 성명, 전공, 소속으로 구성된 교수 개체이다.



해설

교수 개체의 구성 요소

- 속성 : 개체가 가지고 있는 특성, 교수번호, 성명, 전공, 소속
- 개체 타입 : 속성으로만 기술된 개체의 정의
- 개체 인스턴스 : 개체를 구성하고 있는 각 속성들이 값을 가져 하나의 개체를 나타내는 것으로 개체 어커런스(Occurrence)라고도 한다. 어커런스는 정의된 레코드의 구조에 따라 데이터베이스에 구체적이고 실제적인 정보를 저장하고 있는 데이터를 의미한다.
- 개체 세트 : 개체 인스턴스의 집합

1. 논리 데이터베이스 설계-SEC_03(데이터 모델의 구성 요소-개체(Entity))

2) 개체 선정 방법

- 업무 분석에 관한 내용을 구체적으로 설명한 업무 기술서를 이용한다.
- 실제 업무를 담당하고 있는 담당자와 인터뷰를 한다.
- 업무 기술서와 인터뷰에서 확인하지 못한 정보가 있는지 실제 업무를 직접 견학하여 확인한다.
- 실제 업무에 사용되고 있는 장부와 전표를 이용한다.
- 이미 구축된 시스템이 있는 경우 해당 시스템의 산출물을 검토한다.
- 자료 흐름도(DFD)를 통해 업무 분석을 수행했을 경우 자료 흐름도의 자료 저장소(Data Store)를 이용한다.
- BPR(업무 프로세스 재설계)에 의해 업무를 재정의한 경우 관련 개체를 찾는다.

자료흐름도(DFD : Data Flow Diagram) : 자료 흐름도는 요구사항 분석에서 자료의 흐름 및 변환 과정과 기능을 도형을 중심으로 기술하는 방법으로 자료 흐름 그래프 또는 버블 차트라고도 한다.

자료 저장소(Data Store) : 자료 저장소는 시스템에서의 자료 저장소(파일, 데이터베이스)를 의미한다.

BPR(Business Process Reengineering, 업무 프로세스 재설계) : BPR 기업이 경쟁에서 우위를 확보하기 위해 기업의 핵심 부문인 비용, 품질 서비스 스피드와 같은 요인들이 획기적인 향상을 이룰 수 있도록, 프로세스를 기반으로 비즈니스 시스템을 근본적으로 재설계 하여 혁신적인 성과를 추구하는 것을 의미한다.

1. 논리 데이터베이스 설계-SEC_03(데이터 모델의 구성 요소-개체(Entity))

3) 개체명 지정 방법

- 일반적으로 해당 업무에서 사용하는 용어로 지정한다.
- 약어 사용은 되도록 제한한다.
- 가능하면 단수 명사를 사용한다.
- 모든 개체명은 유일해야 한다.
- 가능하면 개체가 생성되는 의미에 따라 이름을 부여한다.

예) 교수, 고객, 주문, 도시 등

논리 데이터베이스 설계-SEC_03(데이터 모델의 구성 요소-개체(Entity)) 기출 문제 및 출제 예상 문제

기출 및 출제 예상 문제(데이터 모델의 구성 요소-개체(Entity))

1. 데이터베이스에 관한 사항으로 다음에서 설명하는 것은?

This is a "thing" in the real world an independent existence, It may be an object with a physical existence(a particular person, car, house, or employee) or an object with a conceptual existence(a company, a job, or a university course).

이것은 현실 세계에서 독립적인 존재인 "사물"입니다. 그것은 물리적 존재(특정한 사람, 자동차, 집 또는 직원)를 가지 물체이거나 개념적 존재(회사, 직업 또는 대학 과정)를 가진 물체일 수 있습니다.

- ① entity ② view
- ③ value ④ relationship

2. 다음 중 개체에 대한 설명으로 가장 옳지 않은 것은?

- ① 개체는 사람이 생각하는 개념이나 정보 단위 같은 현실 세계의 대상체를 말한다.
- ② 개체는 유일한 식별자에 의해 식별이 가능해야 한다.
- ③ 개체는 다른 개체와 하나 이상의 관계(Relationship)가

3. 다음 중 개체 선정 방법에 대한 설명으로 옳지 않은 것은?

- ① 업무와 관련해서 설명한 업무 기술서를 이용한다.
- ② DFD에 의해 업무를 재정의한 경우 관련 개체를 찾는다.
- ③ 현업의 장표를 활용한다.
- ④ 현업 담당자와의 인터뷰를 활용한다.

개체 선정 방법

- 업무 분석에 관한 내용을 구체적으로 설명한 업무 기술서를 이용한다.
- 실제 업무를 담당하고 있는 담당자와 인터뷰를 한다.
- 업무 기술서와 인터뷰에서 확인하지 못한 정보가 있는지 실제 업무를 직접 견학하여 확인한다.
- 실제 업무에서 사용되고 있는 장부와 전표를 이용한다.
- 이미 구축된 시스템이 있는 경우 해당 시스템의 산출물을 검토한다.
- 자료 흐름도(DFD)를 통해 업무 분석을 수행했을 경우 자료 흐름도의 자료 저장소(Data Store)를 이용한다.
- BPR(업무 프로세스 재설계)에 의해 업무를 재정의한 경우 관련된 개체를 찾는다.

4. 다음 중 개체명 지정 방법에 대한 설명으로 가장 옳지 않은 것은?

- ① 일반적으로 현업에서 사용하는 용어를 사용한다.

논리 데이터베이스 설계-SEC_03(데이터 모델의 구성 요소-개체(Entity)) 기출 문제 및 출제 예상 문제

기출 및 출제 예상 문제(데이터 모델의 구성 요소-개체(Entity))

5. 데이터베이스의 구성 요소 중 개체(Entity)에 대한 설명으로 적합하지 않은 것은?

- ① 속성들이 가질 수 있는 모든 값들의 집합이다.
- ② 데이터베이스에 표현하려고 하는 현실 세계의 대상체이다.
- ③ 유형, 무형의 정보로서 서로 연관된 몇 개의 속성으로 구성된다.
- ④ 파일의 레코드에 대응하는 것으로 어떤 정보를 제공하는 역할을 수행한다.

속성들이 가질 수 있는 모든 값들의 집합은 도메인(Domain)이라고 칭한다.

개체의 구성 요소

- 1. 속성 : 개체가 가지고 있는 특성
- 2. 개체 타입 : 속성만으로 기술된 개체의 정의
- 3. 개체 인스턴스 : 개체를 구성하고 있는 각 속성들이 값을 가져 하나의 개체를 나타내는 것으로 개체 어커런스(Occurrence)라고도 한다. 어커런스는 정의된 레코드의 구조에 따라 데이터베이스에 구체적이고 실제적인 정보를 저장하고 있는 데이터를 의미한다.

1. 논리 데이터베이스 설계-SEC_04(데이터 모델의 구성 요소-속성(Attribute))

1) 속성의 정의 및 특징

- 속성(Attribute)은 데이터베이스를 구성하는 가장 작은 논리적 단위이다.
- 파일 구조상의 데이터 항목 또는 데이터 필드에 해당한다.
- 속성은 개체를 구성하는 항목이다.
- 속성은 개체의 특성을 기술한다.
- 속성의 수를 디그리(Degree) 또는 차수라고 한다.

2) 속성의 종류

; 속성은 속성의 특성과 개체 구성 방식에 따라 분류할 수 있다.

- 속성의 특성에 따른 분류

기본 속성 (Basic Attribute)	<ul style="list-style-type: none">•업무 분석을 통해 정의한 속성이다.•속성 중 가장 많고 일반적이다.•업무로부터 분석한 속성이라도 업무상 코드로 정의한 속성은 기본 속성에서 제외된다.
설계 속성 (Designed Attribute)	<ul style="list-style-type: none">•원래 업무상 존재하지 않고 설계 과정에서 도출해내는 속성이다.•업무에 필요한 데이터 외에 데이터 모델링을 위해 업무를 규칙화하려고 속성을 새로 만들거나 변형하여 정의하는 속성이다.
파생 속성 (Derived Attribute)	<ul style="list-style-type: none">•다른 속성으로부터 계산이나 변형 등의 영향을 받아 발생하는 속성이다.•파생 속성은 되도록 적은 수를 정의하는 것이 좋다.

1. 논리 데이터베이스 설계-SEC_04(데이터 모델의 구성 요소-속성(Attribute))

2) 속성의 종류

; 속성은 속성의 특성과 개체 구성 방식에 따라 분류할 수 있다.

● 속성의 특성에 따른 분류

예제) 속성의 특성에 따른 분류 - 자동차

기본 속성	자동차명, 제조일, 연비
설계 속성 (자동차 코드)	A01-세단, A02-SUV, A03-트럭
파생 속성 (계산값)	총판매수량, 총판매금액 등

해설

파생 속성은 다른 속성의 영향을 받는 만큼 프로세스 설계 시 정합성 유지를 위해 유의해야 할 점이 많으므로 되도록 적게 정의하는 것이 좋다. 기본 속성인 '자동차명', '제조일', '연비'는 업무 분석을 통해 정의한 가장 일반적인 속성이고, 설계 속성인 '자동차 코드'는 판매되는 자동차를 종류별로 구분하기 위해 업무에는 없지만 새롭게 정의한 속성이며, 파생 속성인 '총판매수량'과 '총판매금액'은 특정 기간 동안 판매된 자동차의 수량과 금액의 합계 계산을 위해 정의한 속성이다.

1. 논리 데이터베이스 설계-SEC_04(데이터 모델의 구성 요소-속성(Attribute))

2) 속성의 종류

; 속성은 속성의 특성과 개체 구성 방식에 따라 분류할 수 있다.

● 개체 구성 방식에 따른 분류

기본키 속성 (Primary Key Attribute)	개체를 식별할 수 있는 속성이다.
외래키 속성 (Foreign Key Attribute)	다른 개체와의 관계에서 포함된 속성이다.
일반 속성	개체에 포함되어 있고 기본키, 외래키에 포함되지 않은 속성이다.

기본키(Primary Key) : 기본키는 개체 인스턴스들을 서로 구분할 수 있는 유일한 속성을 말한다.
외래키(Foreign Key) : 다른 릴레이션의 기본키를 참조하는 속성 또는 속성들의 집합을 의미한다.

1. 논리 데이터베이스 설계-SEC_04(데이터 모델의 구성 요소-속성(Attribute))

3) 속성 후보 선정 원칙

; 속성으로 지정할 후보는 최대한 많이 선택하는 것이 좋으며 선정 원칙은 다음과 같다.

- 원시(Source)속성으로 판단되는 속성 후보는 버리지 않는다.
- 소그룹 별로 속성 후보군을 만들고 가장 근접한 개체에 할당한다.

원시 속성 : 다른 속성을 통해 다시 재현할 수 없는 속성을 의미한다. 원시 속성을 버리면 해당 속성 정보가 완전히 소실되므로 버리면 안된다.

1. 논리 데이터베이스 설계-SEC_04(데이터 모델의 구성 요소-속성(Attribute))

4) 속성명 지정 원칙

- 속성명은 웹이나 클라이언트/서버(Client/Server) 등 어떠한 환경에서든 사용자 인터페이스에 나타나기 때문에 정확하고 혼란이 없도록 지정해야 한다.
- 속성명 지정 원칙
 - 해당 업무에서 사용하는 용어로 지정한다.
 - 서술형으로 지정하지 않는다.
 - 가급적이면 약어의 사용은 제한한다.
 - 개체명은 속성명으로 사용할 수 없다.
 - 개체에서 유일하게 식별 가능하도록 지정한다.

속성명을 서술형으로 지정하지 않는 이유는 속성명을 수식어가 많은 서술형으로 지정하면 해당 속성명이 무엇을 의미하는지 파악하기 어렵고 다루기 힘들다. 때문에 의미 파악이 쉽고 다루기 쉬운 명사형으로 지정해야 한다.

논리 데이터베이스 설계-SEC_04(데이터 모델의 구성 요소-속성(Attribute)) 출제 예상 문제

출제 예상 문제(데이터 모델의 구성 요소-속성(Attribute))

1. 다음 중 속성에 대한 설명으로 가장 옳지 않은 것은?

- ① 속성은 데이터베이스를 구성하는 가장 큰 논리적 단위이다.
- ② 속성은 개체를 구성하는 항목이다.
- ③ 파일 구조상의 데이터 필드에 해당한다.
- ④ 속성의 수를 디그리(Degree) 또는 차수라고 한다.

속성의 정의 및 특징

- 속성(Attribute)은 데이터베이스를 구성하는 가장 작은 논리적 단위이다.
- 파일 구조상의 데이터 항목 또는 데이터 필드에 해당한다.
- 속성은 개체를 구성하는 항목이다.
- 속성은 개체의 특성을 기술한다.
- 속성의 수를 디그리(Degree) 또는 차수라고 한다.

2. 다음 중 속성의 특성에 따른 분류에 대한 설명으로 옳지 않은 것은?

- ① 기본 속성은 업무로부터 추출한 모든 속성이다.
- ② 설계 속성은 데이터 모델링을 위해 업무를 규칙화하려고 속성을 새로 만들거나 변형하여 정의하는 속성이다.
- ③ 파생 속성은 다른 속성에 영향을 받아 발생하는 속성으로, 보통

3. 속성을 개체 구성 방식에 따라 분류할 경우 포함되지 않는 것은?

- ① 기본키 속성 ② 외래키 속성
- ③ 일반 속성 ④ 파생 속성

4. 다음 중 속성명 지정 원칙에 대한 설명으로 가장 옳지 않은 것은?

- ① 해당 업무에서 사용하는 이름을 부여한다.
- ② 속성명은 서술식으로 지정해도 된다.
- ③ 약어 사용은 가급적 자제한다.
- ④ 개체명은 속성명으로 사용할 수 없다.

속성명은 웹이나 클라이언트/서버 등 어떠한 환경에서든 사용자 인터페이스에 나타나기 때문에 정확하고 혼란이 없도록 지정해야 한다.

속성명 지정 원칙

- 해당 업무에서 사용하는 용어로 지정한다.
- 서술형으로 지정하지 않는다.
- 가급적이면 약어의 사용은 제한한다.
- 개체명은 속성명으로 사용할 수 없다.

논리 데이터베이스 설계-SEC_04(데이터 모델의 구성 요소-속성(Attribute)) 출제 예상 문제

출제 예상 문제(데이터 모델의 구성 요소-속성(Attribute))

5. 다음 중 자동차를 개체로 나타낼 때 일반 속성에 해당하지 않는 것은?

- ① 총 판매액
- ② 제조일
- ③ 자동차명
- ④ 연비

속성의 특성에 따른 분류 – 자동차

기본 속성 – 자동차명, 제조일, 연비

설계 속성(자동차 코드) – A01-세단, A02-SUV, A03-트럭

파생 속성 – 총판매수량, 총판매금액

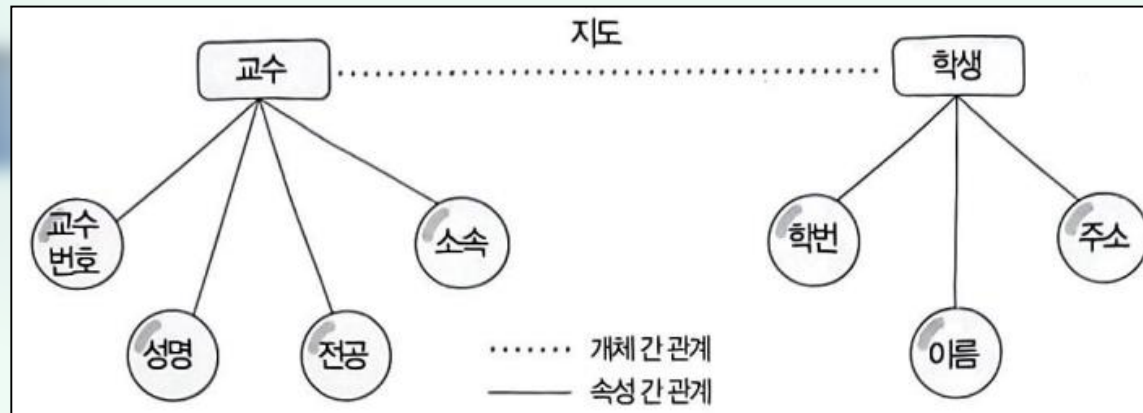
1. 논리 데이터베이스 설계-SEC_05(데이터 모델의 구성 요소-관계(Relationship))

1) 관계의 정의

; 관계는 개체와 개체 사이의 논리적인 연결을 의미한다.

- 관계에는 개체 간의 관계와 속성 간의 관계가 있다.

예) 다음은 교수가 학생을 지도하는 관계이다.



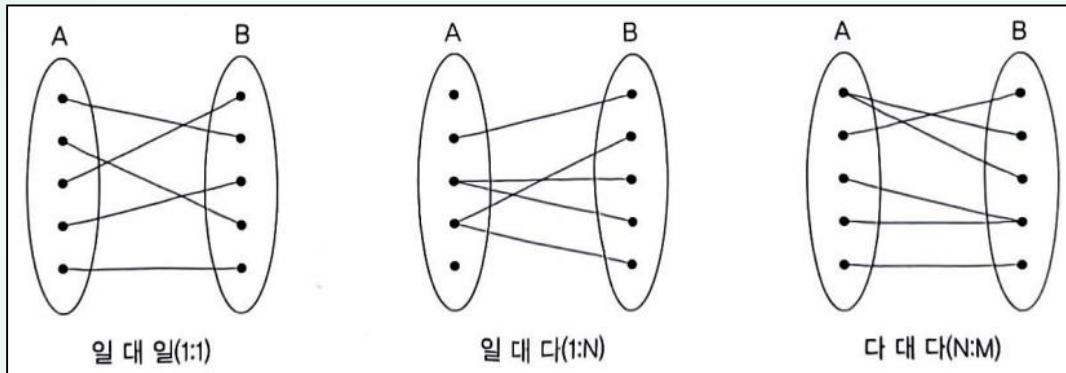
관계(Relationship)의 이용 : 이미속 교수의 전공을 알기 위해서는 속성 간의 관계를 이용하고, 이미속 교수가 가르치는 학생의 이름을 알기 위해서는 개체 간 관계를 이용한다.

1. 논리 데이터베이스 설계-SEC_05(데이터 모델의 구성 요소-관계(Relationship))

2) 관계의 형태

; 관계의 형태에는 일 대 일(1:1), 일 대 다(1:N), 다 대 다(N:M) 관계가 있다.

- **일 대 일(1:1)** : 개체 집합 A의 각 원소가 개체 집합 B의 원소 한 개와 대응하는 관계이다.
- **일 대 다(1:N)** : 개체 집합 A의 각 원소는 개체 집합 B의 원소 여러 개와 대응하고 있지만, 개체 집합 B의 각 원소는 개체 집합 A의 원소 한 개와 대응하는 관계이다.
- **다 대 다(N:M)** : 개체 집합 A의 각 원소는 개체 집합 B의 원소 여러 개와 대응하고, 개체 집합 B의 각 원소도 개체 집합 A의 원소 여러 개와 대응하는 관계이다.



1. 논리 데이터베이스 설계-SEC_05(데이터 모델의 구성 요소-관계(Relationship))

3) 관계의 종류

; 관계는 처리하는 업무 형태에 따라 종속 관계, 중복 관계, 재귀 관계, 배타 관계 등이 있다.

- 종속 관계(Dependent Relationship) : 두 개체 사이의 주,종 관계를 표현한 것으로, 식별 관계와 비식별 관계가 있다.
- 중복 관계(Redundant Relationship) : 두 개체 사이에 2번 이상의 종속 관계가 발생하는 관계이다.
- 재귀 관계(Recursive Relationship) : 개체가 자기 자신과 관계를 갖는 것으로, 순환 관계 (Recursive Relationship)라고도 한다.
- 배타 관계(Exclusive Relationship)
 - 개체의 속성이나 구분자를 기준으로 개체의 특성을 분할하는 관계로, 배타 AND 관계와 배타 OR 관계로 구분한다.
 - 배타 AND 관계는 하위 개체들 중 속성이나 구분자 조건에 따라 하나의 개체만을 선택할 수 있고, 배타 OR 관계는 하나 이상의 개체를 선택할 수 있다.

관계 데이터베이스 특성상 관계가 복잡하면 최적화 작업이 어려우므로 중복 관계는 되도록 사용하지 않는 것이 좋다.

1. 논리 데이터베이스 설계-SEC_05(데이터 모델의 구성 요소-관계(Relationship))

4) 관계의 표현

; 관계는 관계 표기 기호를 사용하여 표시한다.

● 관계 표기 기호

기호	의미
	필수(Mandatory)
○	선택적(Optional)
<	다중(Multiple)

● 관계의 표현

관계	표현		의미
	정보공학 표기법	바커 표기법	
1:1			양쪽에 반드시 1개씩 존재
1:0 또는 1:1			왼쪽에는 반드시 1개, 오른쪽에는 없거나 1개 존재
1:N			왼쪽에는 반드시 1개, 오른쪽에는 반드시 여러 개 존재
1:1 또는 1:N			왼쪽에는 반드시 1개, 오른쪽에는 1개 또는 여러 개 존재
1:0 또는 1:1 또는 1:N			왼쪽에는 반드시 1개, 오른쪽에는 0개 또는 1개 또는 여러 개 존재

1. 논리 데이터베이스 설계-SEC_05(데이터 모델의 구성 요소-관계(Relationship))

5) 식별 관계 / 비식별 관계

● 식별 관계(Identifying Relationship)

- 개체 A, B 사이의 관계에서 A 개체의 기본키가 B 개체의 외래키이면서 동시에 기본키가 되는 관계를 말한다.
- B 개체의 존재 여부가 A 개체의 존재 여부에 의존적인 경우에 발생한다.
- ER 도형에서 식별 관계는 실선으로 표시한다.

● 비식별 관계(Non-Identifying Relationship)

- 개체 A, B 사이의 관계에서 A 개체의 기본키가 B 개체의 비기본키 영역에서 외래키가 되는 관계를 말한다.
- B 개체의 존재 여부는 A 개체의 존재 여부와 관계없이 존재한다.
- 일반적으로 두 개체는 비식별 관계로 존재하는 경우가 많다.
- ER 도형에서 비식별 관계는 점선으로 표시한다

기본키(Primary Key) : 기본키는 후보키 중에서 선택한 주키(Main Key)를 의미한다.

※후보키(Candidate Key) : 릴레이션을 구성하는 속성들 중에서 튜플을 유일하게 식별할 수 있는 속성들의 부분 집합

외래키(Foreign Key) : 외래키는 다른 릴레이션을 기본키를 참조하는 속성 또는 속성들의 집합을 의미한다.

논리 데이터베이스 설계-SEC_05(데이터 모델의 구성 요소-관계(Relationship)) 출제 예상 문제

출제 예상 문제(데이터 모델의 구성 요소-관계(Relationship))

1. 개체 집합 A의 각 원소는 개체 집합 B의 원소 여러 개와 대응하고 있지만, 개체 집합 B의 각 원소는 개체 집합 A의 원소 한 개와 대응하는 관계를 의미하는 것은?

- ① 1:1 ② 1:N
- ③ N:M ④ 1:0

관계의 형태에는 일 대 일(1:1), 일 대 다(1:N), 다 대 다(N:M) 관계가 있다.

- 일 대 일(1:1) : 개체 집합 A의 각 원소가 개체 집합 B의 원소 한 개와 대응하는 관계이다.

- 일 대 다(1:N) : 개체 집합 A의 각 원소는 개체 집합 B의 원소 여러 개와 대응하고 있지만, 개체 집합 B의 각 원소는 개체 집합 A의 원소 한 개와 대응하는 관계이다.

- 다 대 다(N:M) : 개체 집합 A의 각 원소는 개체 집합 B의 원소 여러 개와 대응하고, 개체 집합 B의 각 원소도 개체 집합 A의 원소 여러 개와 대응하는 관계이다.

2. 다음 중 관계의 종류에 대한 설명으로 가장 옳지 않은 것은?

- ① 종속 관계는 두 개체 간의 주/종 관계를 표현하는 것

3. 개체 집합 A의 각 원소는 개체 집합 B의 원소 여러 개와 대응 하고, 개체 집합 B의 각 원소도 개체 집합 A의 원소 여러 개와 대응하는 관계를 의미하는 것은?

- ① 1:0 ② 1:1
- ③ 1:N ④ N:M

4. 다음 중 관계의 종류에 해당하지 않는 것은?

- ① 종속 관계 ② 재귀 관계
- ③ 단일 관계 ④ 중복 관계

관계의 종류에는 종속, 중복, 재귀, 배타 관계가 있다는 것을 기억하자.

논리 데이터베이스 설계-SEC_05(데이터 모델의 구성 요소-관계(Relationship)) 출제 예상 문제

출제 예상 문제(데이터 모델의 구성 요소-관계(Relationship))

5. 다음 설명이 의미하는 A와 B의 관계는?

An entity in an entity set A is associated with any number of entities in an entity set B, and an entity in B is associated with any number of entities in A.

- ① one to one ② one to many
- ③ many to one ④ many to many

6. 관계 표기 기호 중 필수를 나타내는 기호는?

- ① O ② <
- ③ > ④ |

관계 표기 기호

| : 필수(Mandatory)

O : 선택적(Optional)

< : 다중(Multiple)

7. 배타 관계(Exclusive Relationship)에 대한 설명 중 옳지 않은 것은?

- ① 개체의 속성이나 구분자를 기준으로 개체의 특성을 분할하는 관계이다.
 - ② 배타 AND 관계와 배타 OR 관계로 구분한다.
 - ③ 배타 AND 관계는 하위 개체들 중 속성이나 구분자 조건에 따라 여러 개의 개체를 선택할 수 있다.
 - ④ 배타 OR 관계는 하나 이상의 개체를 선택할 수 있다.
- 배타 AND 관계는 하위 개체들 중 속성이나 구분자 조건에 따라 한 개의 개체를 선택할 수 있다.

1. 논리 데이터베이스 설계-SEC_06(식별자(Identifier))

1) 식별자의 정의 및 분류

; 식별자(Identifier)는 하나의 개체 내에서 각각의 인스턴스를 유일(Unique)하게 구분할 수 있는 구분자로, 모든 개체는 한 개 이상의 식별자를 반드시 가져야 한다.

- 식별자는 개체 내에서 대표성 여부, 스스로 생성 여부, 단일 속성 여부, 대체 여부에 따라 다음과 같이 분류한다.

분류	식별자
대표성 여부	주 식별자(Primary Identifier), 보조 식별자(Alternate Identifier)
스스로 생성 여부	내부 식별자(Internal Identifier), 외부 식별자(Foreign Identifier)
단일 속성 여부	단일 식별자(Single Identifier), 복합 식별자(Composite Identifier)
대체 여부	원조 식별자(Original Identifier), 대리 식별자(Surrogate Identifier)

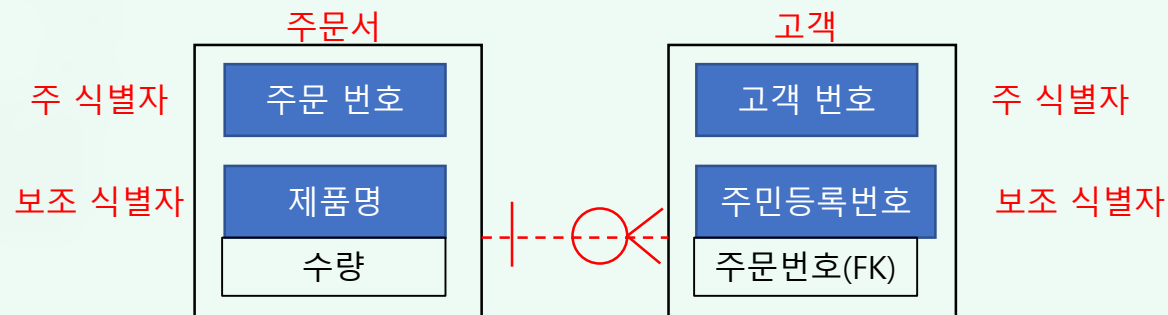
식별자와 키(Key)는 같은 의미로 사용된다. 식별자는 논리 데이터 모델링 단계에서 사용되고, 키는 물리 데이터 모델링 단계에서 사용된다. 따라서 개체는 식별자를 갖고, 릴레이션은 키를 갖는다.

인스턴스(Instance) : 데이터 개체를 구성하고 있는 속성들에 데이터 타입이 정의되어 구체적인 데이터 값을 갖고 있는 것을 말한다.

1. 논리 데이터베이스 설계-SEC_06(식별자(Identifier))

2) 주 식별자 / 보조 식별자

- 주 식별자(Primary Identifier)는 개체를 대표하는 유일한 식별자이다.
- 보조 식별자(Alternate Identifier)는 주 식별자를 대신하여 개체를 식별할 수 있는 속성이다.
- 두 식별자 모두 개체를 유일하게 식별할 수 있어야 한다.
- 하나의 개체에 주 식별자는 한 개만 존재하지만 보조 식별자는 한 개 이상 존재한다.
- 개체를 유일하게 식별할 수 있는 속성이 두 개 이상인 경우 업무에 가장 적합한 속성을 주 식별자로 지정하고, 나머지는 보조 식별자로 지정한다.
- 물리적 테이블에서 주 식별자는 기본키(Primary Key)로, 보조 식별자는 유니크 인덱스(Unique Index)로 지정되어 사용된다.



유니크 인덱스(Unique Index) : 속성값에서 중복 값이 없도록 하는 인덱스이다.

1. 논리 데이터베이스 설계-SEC_06(식별자(Identifier))

2) 주 식별자 / 보조 식별자

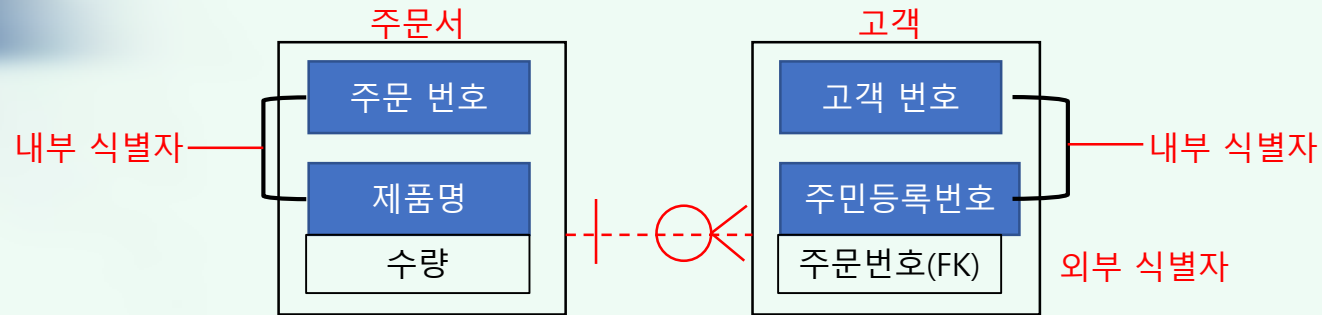
● 주 식별자의 4가지 특징

특징	설명
유일성	주 식별자에 의해 개체 내에 모든 인스턴스들이 유일하게 구분되어야 한다.
최소성	주 식별자를 구성하는 속성의 수는 유일성을 만족하는 최소 수가 되어야 한다.
불변성	주 식별자가 한 번 특정 개체에 지정되면 그 식별자는 변하지 않아야 한다.
존재성	주 식별자가 지정되면 식별자 속성에 반드시 데이터 값이 존재해야 한다.

1. 논리 데이터베이스 설계-SEC_06(식별자(Identifier))

3) 내부 식별자/외부 식별자

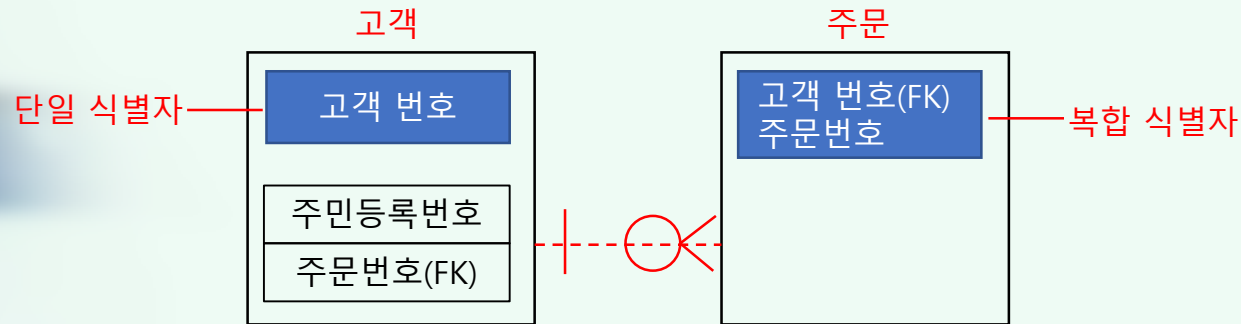
- 내부 식별자(Internal Identifier)는 개체 내에서 스스로 만들어지는 식별자이다.
- 외부 식별자(Foreign Identifier)는 다른 개체와의 관계(Relationship)에 의해 외부 개체의 식별자를 가져와 사용하는 식별자이다.
- 외부 식별자는 자신의 개체에서 다른 개체를 찾아가는 연결자 역할을 한다.



1. 논리 데이터베이스 설계-SEC_06(식별자(Identifier))

4) 단일 식별자/복합 식별자

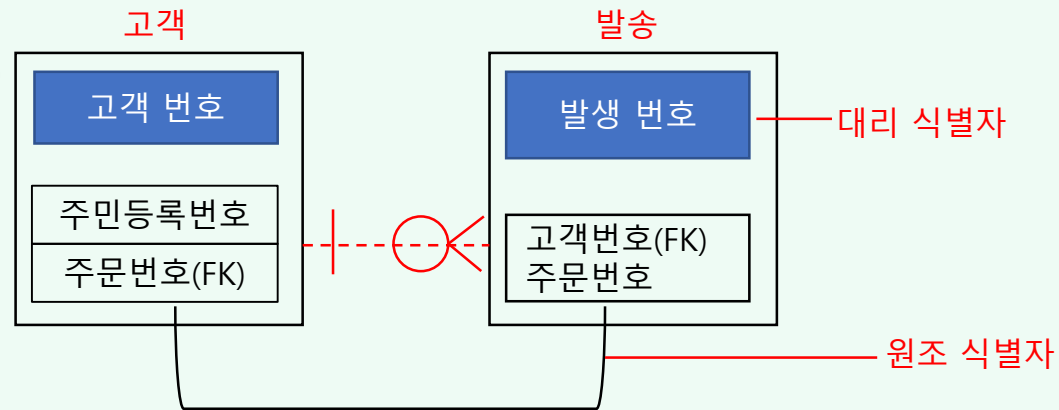
- 단일 식별자(Single Identifier)는 주 식별자가 한 가지 속성으로만 구성된 식별자이다.
- 복합 식별자(Composite Identifier)는 주 식별자가 두 개 이상의 속성으로 구성된 식별자이다.



1. 논리 데이터베이스 설계-SEC_06(식별자(Identifier))

5) 원조 식별자/대리 식별자

- 원조 식별자(Original Identifier)는 업무에 의해 만들어지는 가공되지 않은 원래의 식별자로, 본질 식별자라고도 한다.
- 대리 식별자(Surrogate Identifier)는 주 식별자의 속성이 두 개 이상인 경우 속성들을 하나의 속성으로 묶어 사용하는 식별자로, 인조 식별자라고도 한다.



1. 논리 데이터베이스 설계-SEC_06(식별자(Identifier))

6) 대리 식별자의 조건

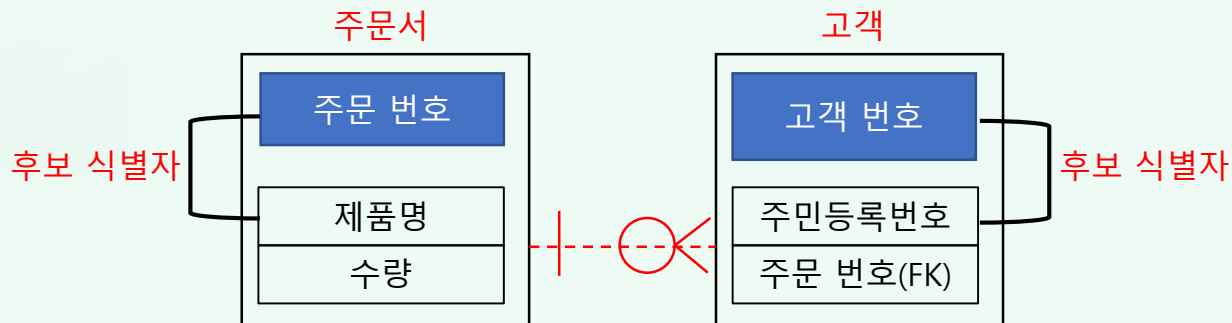
- 최대한 범용적인 값을 사용한다.
- 유일한 값을 만들기 위한 대리 식별자를 사용한다.
- 하나의 대리 식별자 속성으로 대체할 수 없는 경우를 주의한다.
- 편의성과 단순성, 의미의 체계화를 위한 대리 식별자를 사용할 수 있다.
- 시스템적인 필요성에 의해 내부적으로만 사용하는 대리 식별자를 사용할 수 있다.

1. 논리 데이터베이스 설계-SEC_06(식별자(Identifier))

7) 후보 식별자

; 후보 식별자는 개체에서 각 인스턴스를 유일하게 식별할 수 있는 속성 또는 속성 집합을 의미한다.

- 하나의 개체에는 한 개 이상의 후보 식별자가 있고, 이 중 개체의 대표성을 나타내는 식별자를 주 식별자로, 나머지는 보조 식별자로 지정한다.
- 후보 식별자 조건
 - 각 인스턴스를 유일하게 식별할 수 있어야 한다.
 - 속성들을 직접 식별할 수 있어야 한다.
 - 널 값(NULL Value)이 될 수 없다.
 - 속성 집합은 후보 식별자로 지정한 경우 개념적으로 유일해야 한다.
 - 후보 식별자의 데이터는 자주 변경되지 않아야 한다.



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.

논리 데이터베이스 설계-SEC_06(식별자(Identifier)) 출제 예상 문제

출제 예상 문제(식별자(Identifier))

1. 다음 중 식별자를 대표성 여부로 분류했을 경우 해당하는 것은?

- ① 내부 식별자, 외부 식별자
- ② 주 식별자, 보조 식별자
- ③ 단일 식별자, 복합 식별자
- ④ 원조 식별자, 대리 식별자

1. 대표성 여부

- 주 식별자(Primary Identifier), 보조 식별자(Alternate Identifier)

2. 스스로 생성 여부

- 내부 식별자(Internal Identifier), 외부 식별자(Foreign Identifier)

3. 단일 속성 여부

- 단일 식별자(Single Identifier), 복합 식별자(Composite Identifier)

4. 대체 여부

- 원조 식별자(Original Identifier), 대리 식별자(Surrogate Identifier)

2. 다음 중 주 식별자와 보조 식별자에 대한 설명으로 가장 옳지 않은 것은?

- ① 주 식별자는 개체의 대표성을 나타내는 유일한 식별자이다.
- ② 보조 식별자는 주 식별자를 대신하여 보조적으로 개체를 식별 할

3. 다음 중 주 식별자의 특징이 아닌 것은?

- ① 존재성 ② 최대성
- ③ 불변성 ④ 유일성

주 식별자의 4가지 특징

1. 유일성 : 주 식별자에 의해 개체 내에 모든 인스턴스들이 유일하게 구분되어야 한다.

2. 최소성 : 주 식별자를 구성하는 속성의 수는 유일성을 만족하는 최소 수가 되어야 한다.

3. 불변성 : 주 식별자가 한 번 특정 개체에 지정되면 그 식별자는 변하지 않아야 한다.

4. 존재성 : 주 식별자가 지정되면 식별자 속성에 반드시 데이터 값이 존재해야 한다. 즉 null값을 허용하지 않는다.

4. 다음 중 식별자에 대한 설명으로 가장 옳지 않은 것은?

- ① 내부 식별자는 자신의 개체 타입 내에서 스스로 생성되어 존재하는 식별자이다.
- ② 외부 식별자는 다른 개체와의 관계에 의해 주 식별자 속성을 상속 받아 자신의 속성에 포함되는 식별자이다.
- ③ 단일 식별자는 주 식별자의 구성이 한 가지 속성으로만

논리 데이터베이스 설계-SEC_06(식별자(Identifier)) 출제 예상 문제

출제 예상 문제(식별자(Identifier))

5. 다음 중 후보 식별자에 대한 설명으로 가장 옳지 않은 것은?

- ① 후보 식별자는 개체에서 각 인스턴스를 유일하게 식별할 수 있는 속성 또는 속성 집합이다.
- ② 하나의 개체에는 한 개 이상의 후보 식별자가 있다.
- ③ 후보 식별자는 널 값(NULL Value)이 될 수 있다.
- ④ 후보 식별자의 데이터는 자주 변경되지 않아야 한다.

후보 식별자

후보 식별자는 개체에서 각 개체 인스턴스(튜플, 레코드, ROW)를 유일하게 식별할 수 있는 속성 또는 속성 집합을 의미한다.

- 하나의 개체에는 한 개 이상의 후보 식별자가 있고, 이중 개체의 대표성을 나타내는 식별자를 주 식별자로, 나머지는 보조 식별자로 지정한다.

후보 식별자의 조건

- 각 개체 인스턴스를 유일하게 식별할 수 있어야 한다.
- 속성들을 직접 식별할 수 있어야 한다.
- 널 값(NULL Value)이 될 수 없다.
- 속성 집합은 후보 식별자로 지정한 경우 개념적으로 유일해야

1. 논리 데이터베이스 설계-SEC_07(E-R(개체-관계) 모델)

1) E-R(Entity-Relationship, 개체-관계) 모델의 개요

; E-R 모델은 개념적 데이터 모델의 가장 대표적인 것으로, 1976년 피터 첸(Peter Chen)에 의해 제안되고 기본적인 구성 요소가 정립되었다.

- E-R 모델은 개체와 개체 간의 관계를 기본 요소로 이용하여 현실 세계의 무질서한 데이터를 개념적인 논리 데이터로 표현하기 위한 방법으로 많이 사용되고 있다.
- E-R 모델은 개체 타입(Entity Type)과 이들 간의 관계 타입(Relationship Type)을 이용해 현실 세계를 개념적으로 표현한다.
- E-R 모델에서는 데이터를 개체(Entity), 관계(Relationship), 속성(Attribute)으로 묘사한다.
- E-R 모델은 특정 DBMS를 고려한 것은 아니다.
- E-R 다이어그램으로 표현하며, 1:1, 1:N, N:M 등의 관계 유형을 제한 없이 나타낼 수 있다.
- 최초에는 개체, 관계, 속성과 같은 개념들로 구성되었으나 나중에는 일반화 계층 같은 복잡한 개념들이 첨가되어 확장된 모델로 발전했다.

개체(Entity) : 학생, 교수, 자동차 등과 같이 실세계에서 개념적 또는 물리적으로 존재하는 실제 사용을 의미한다.


관계(Relationship) : 교수 개체는 학생 개체를 지도하는 관계인 것처럼 다른 개체 타입에 속한 개체 사이의 관계를 표현한다.

속성(Attribute) : 학생의 이름, 주소 등과 같이 개체를 묘사하는데 사용될 수 있는 특성을 의미한다.

1. 논리 데이터베이스 설계-SEC_07(E-R(개체-관계) 모델)

2) E-R 다이어그램(Entity-Relationship Diagram)

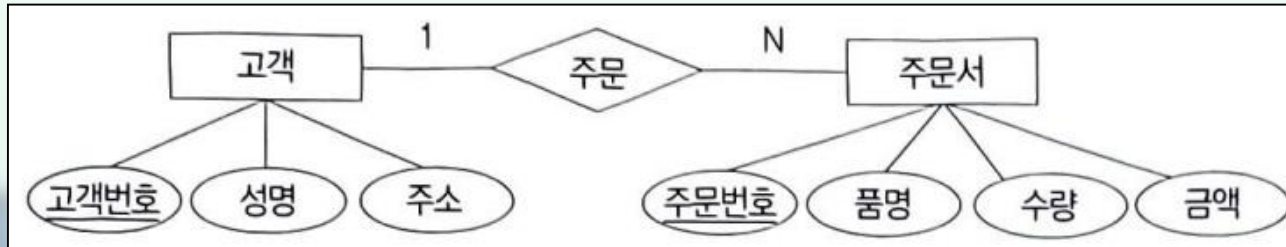
; E-R 모델의 기본 아이디어를 시각적으로 표현하기 위한 그림으로, 실체 간의 관계는 물론 조직, 사용자, 프로그램, 데이터 등 시스템 내에서 역할을 가진 모든 실체들을 표현한다.

기호	기호이름	의미
	사각형	· 개체(Entity) 타입
	마름모	· 관계(Relationship) 타입
	타원	· 속성(Attribute)
	이중 타원	· 다중값 속성(복합 속성)
	밑줄 타원	· 기본키 속성
	복수 타원	· 복합 속성 예) 성명과 성과 이름으로 구성
	관계	· 1:1, 1:N, N:M 등의 개체 간 관계에 대한 대응수를 선 위에 기술함
	선 링크	· 개체 타입과 속성을 연결

1. 논리 데이터베이스 설계-SEC_07(E-R(개체-관계) 모델)

2) E-R 다이어그램(Entity-Relationship Diagram)

예제) 다음은 고객과 주문서 간의 관계를 나타낸 E-R 다이어그램이다.



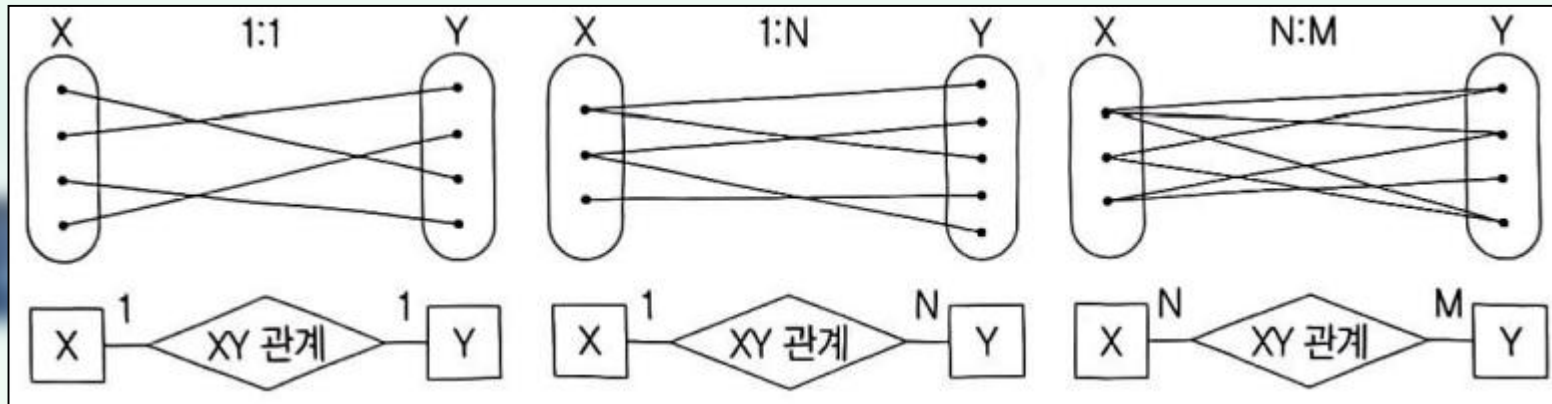
설명

- 개체 : 고객, 주문서
- 속성
 - 고객의 속성 : 고객번호, 성명, 주소
 - 주문서의 속성 : 주문번호, 품명, 수량, 금액
- 관계 : '고객'과 '주문서'의 '주문' 관계는 일 대 다의 관계, 즉 한 사람의 고객이 다수의 주문을 할 수 있고 주문서 1개는 특정인의 주문서로 되어 있다.
- 밑줄 친 속성은 기본키를 나타낸다.

1. 논리 데이터베이스 설계-SEC_07(E-R(개체-관계) 모델)

3) E-R 다이어그램(Entity-Relationship Diagram) 대응 수

; E-R 다이어그램 표현에서 대응 수는 함수 관계도를 이용하여 쉽게 구할 수 있다.



논리 데이터베이스 설계-SEC_07(E-R(개체-관계) 모델)) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(E-R(개체-관계) 모델)

1. 개체-관계(ER) 모델에 대한 설명으로 잘못된 것은?

- ① E-R 다이어그램으로 표현하며 P. Chen이 제안했다.
- ② 일대일(1:1) 관계 유형만을 표현할 수 있다.
- ③ 개체 타입과 이들 간의 관계 타입을 이용해 현실 세계를 개념적으로 표현한 방법이다.
- ④ E-R 다이어그램은 E-R 모델을 그래프 방식으로 표현한 것이다.

E-R 모델은 개념적 데이터 모델의 가장 대표적인 것으로, 1976년 피터 첸(P. Chen)에 의해 제안되고 기본적인 구성 요소가 정립되었다.

- 1. E-R 모델은 개체와 개체 간의 관계를 기본 요소로 이용하여 현실 세계의 무질서한 데이터를 개념적인 논리 데이터로 표현하기 위한 방법으로 많이 사용되고 있다.
- 2. E-R 모델은 개체 타입(Entity Type)과 이들 간의 관계 타입을 이용해 현실 세계를 개념적으로 표현한다.
- 3. E-R 모델에서는 데이터를 개체, 관계, 속성으로 묘사된다.
- 4. E-R 모델은 특정 DBMS를 고려한 것이 아니다.
- 5. E-R 다이어그램으로 표현하며, 1:1, 1:N, N:M 등의 관계 유형

3. 개체 관계(E-R) 모델에 대한 설명으로 잘못된 것은?

- ① 특정 DBMS를 고려하여 제작하지 않는다.
- ② 개체는 마름모, 속성은 사각형을 이용하여 표현한다.
- ③ 개념적 데이터베이스 단계에서 제작된다.
- ④ E-R 모델의 기본적인 아이디어를 시각적으로 가장 잘 나타낸 것이 E-R 다이어그램이다.

사각형 : 개체(Entity)타입

마름모(다이아몬드) : 관계(Relationship) 타입

타원 : 속성(Attribute)

이중 타원 : 다중값 속성(복합 속성)

밑줄 타원 : 기본키 속성

복수 타원 : 복합 속성

예) 성명과 성과 이름으로 구성

관계 : 1:1, 1:N, N:M 등의 개체 간 관계에 대한 대응 수를 선 위에 기술함

선 링크 : 개체 타입과 속성을 연결

4. E-R 모델에서 다중값 속성의 표기법은?

- ① 마름모 ② 사각형

논리 데이터베이스 설계-SEC_07(E-R(개체-관계) 모델)) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(E-R(개체-관계) 모델)

5. 개체-관계 모델의 E-R 다이어그램에서 사용되는 기호와 그 의미의 연결이 틀린 것은?

- ① 사각형 - 개체 타입
- ② 삼각형 - 속성
- ③ 선 - 개체 타입과 속성을 연결
- ④ 마름모 - 관계 타입

6. 학생과 학교 개체 간의 학적 관계를 E-R 다이어그램으로 옳게 표현한 것은?

- ①
- ②
- ③
- ④

7. 사람과 도시 사이의 거주 관계에서 사람은 반드시 하나의 도시 에 거주해야만 하며, 하나의 도시에는 다수의 사람이 거주한다고 할 때 이를 E-R 다이어그램으로 정확히 표현한 것은?

- ①
- ②
- ③
- ④

사람은 하나의 도시에, 도시에 사는 사람은 여럿, 그러면 도시와 사람의 관계는 1:N의 관계가 된다.

8. E-R 모델의 표현 방법으로 옳지 않은 것은?

- ① 개체 타입 : 사각형
- ② 관계 타입 : 마름모
- ③ 속성 : 오각형
- ④ 연결 : 선

논리 데이터베이스 설계-SEC_07(E-R(개체-관계) 모델)) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(E-R(개체-관계) 모델)

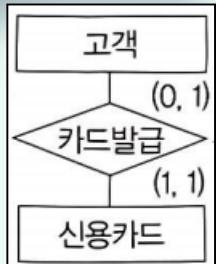
9. 개체-관계(E-R) 모델의 설명으로 옳지 않은 것은?

- ① 데이터를 엔티티, 관계, 속성으로 묘사한다.
- ② E-R 모델에서 엔티티는 실세계에서 개념적 또는 물리적으로 존재하는 실제 사용을 의미한다.
- ③ E-R 모델에서 속성은 엔티티를 묘사하는 데 사용될 수 있는 특성을 의미한다.

④ E-R 모델에서 관계는 속성들에 대한 관계 표현이다.

E-R 모델에서 관계는 속성들에 대한 관계 표현이 아니고 개체 간의 관계를 표현하는 것이다.

10. 다음은 어느 신용카드 회사에서 고객과 고객이 발급받은 신용카드 간의 관계를 나타내는 E-R 다이어그램이다. 이에 대한 설명 중 부적합한 것은?



① 신용카드를 발급 받지 않은 고객은 존재할 수 없다.

1. 논리 데이터베이스 설계-SEC_08(관계형 데이터 모델)

1) 관계형 데이터 모델(Relational Data Model)의 개요

; 관계형 데이터 모델은 가장 널리 사용되는 데이터 모델로, 2차원적인 표(Table)를 이용해서 데이터 상호 관계를 정의하는 DB 구조를 말한다.

- 파일 구조처럼 구성한 테이블들을 하나의 DB로 묶어서 테이블 내에 있는 속성들 간의 관계(Relationship)를 설정하거나 테이블 간의 관계를 설정하여 이용한다.
- 기본키(Primary Key)와 이를 참조하는 외래키(Foreign Key)로 데이터 간의 관계를 표현한다.
- 계층 모델과 망 모델의 복잡한 구조를 단순화시킨 모델이다.
- 관계형 모델의 대표적인 언어는 SQL이다.
- 1:1, 1:N, N:M 관계를 자유롭게 표현할 수 있다.

관계(Relation)와 관계(Relationship) : 관계 모델에서는 테이블을 릴레이션(Relation)이라고 부르는데, 릴레이션을 우리말로 관계라고 해석하는 경우가 종종 있어 개체와 개체 간의 관계를 나타내는 Relationship과 혼동되는 경우가 있다. 대부분 Relation 은 테이블 혹은 릴레이션이라 표기하고 Relationship은 관계라고 표시하지만 간혹 그렇지 않은 경우도 있으니 주의하기 바란다.

1. 논리 데이터베이스 설계-SEC_08(관계형 데이터 모델)

2) 관계형 데이터 모델의 구성

예제) E-R 모델에서 설명한 고객과 주문서 그리고 주문관계를 관계형 데이터 모델로 표시하면 다음과 같다.



- '고객'과 '주문서'는 개체 테이블이고, '주문'은 관계(Relationship)로 만들어진 테이블이다.
- 위의 테이블에 데이터를 입력해 보면 좀 더 명확하게 이해할 수 있다.

1. 논리 데이터베이스 설계-SEC_08(관계형 데이터 모델)

2) 관계형 데이터 모델의 구성

고객			주문		주문서			
고객번호	성명	주소	고객번호	주문번호	주문번호	품명	수량	금액
001	박인식	수원시	001	101	101	01	2	300
002	김규정	안산시	001	102	102	02	3	200
003	이정훈	서울시	002	103	103	03	4	100
004	강현준	안양시	002	104	104	01	7	200
			003	105	105	02	5	300
			003	106	106	05	2	200
			003	107	107	01	3	100
			003	108	108	02	1	200
			004	109	109	03	5	200

- <고객> 테이블과 <주문서> 테이블은 <주문> 테이블을 관계로 하여 연결되어 있다. 이때 <고객> 테이블에 있는 '고객번호'와 <주문서> 테이블에 있는 '주문번호'는 기본키이고, <주문> 테이블에 있는 '고객번호'와 '주문번호'는 외래키이다.
- <고객>과 <주문>의 관계는 1:N, 즉 한 명의 고객은 여러 개의 주문을 신청할 수 있다.
- <주문서>와 <주문>의 관계는 1:1 즉 '주문서'는 '주문번호' 1개에 대한 정보만을 가지고 있다.
- <주문> 테이블의 내용을 가지고 주문 고객의 '성명'을 찾으려면 <주문> 테이블의 '고객번호'와 같은 고객번호'를 <고객> 테이블에서 찾으면 되고, 주문 내역을 알려면 주문번호와 같은 '주문번호'를 <주문서> 테이블에서 찾으면 된다.

논리 데이터베이스 설계-SEC_08(관계형 데이터 모델) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(관계형 데이터 모델)

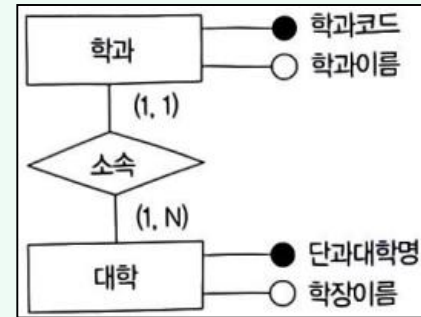
1. 다음의 논리적인 데이터 모델에서 데이터 간의 관계를 기본키(Primary Key)와 이를 참조하는 외래키(Foreign Key)로 표현하는 데이터 모델은?

- ① 관계형 데이터 모델
- ② 네트워크 데이터 모델
- ③ 계층적 모델
- ④ 객체지향 데이터 모델

관계형 데이터 모델은 가장 널리 사용되는 데이터 모델로써, 2차원적인 표(Table)를 이용해서 데이터 상호 관계를 정의하는 DB 구조를 말한다.

- ① 파일 구조처럼 구성한 테이블들을 하나의 DB로 묶어서 테이블 내에 있는 속성들 간의 관계(Relationship)를 설정하거나 테이블 간의 관계를 설정하여 이용한다.
- ② 기본키(PK)와 이를 참조하는 외래키(FK)로 데이터 간의 관계를 표현한다.
- ③ 계층 모델과 망 모델의 복잡한 구조를 단순화시킨 모델이다.
- ④ 관계형 모델의 대표적인 언어는 SQL이다.

3. 다음의 E-R 다이어그램은 대학교에서 어떤 학과가 어느 단과 대학에 소속되는지의 관계를 모델링 한 것이다. 아래 다이어그램을 관계 모델의 릴레이션으로 표현한 것들 중 가장 적합한 것은?
(단, 릴레이션 표현에서 밑줄 친 애트리뷰트는 기본키를 의미한다)



- ① 학과(학과코드, 학과이름, 단과대학명)
대학(단과대학명, 학장이름)
- ② 학과(학과코드, 학과이름)
대학(단과대학명, 학장이름, 소속학과)
- ③ 학과(학과코드, 학과이름, 단과대학명, 학장이름)
- ④ 학과(학과코드, 학과)
대학(단과대학명, 학장이름)
소속(학과코드, 단과대학명)

4. 다음 E-R 다이어그램에 대한 설명 중 거리가 먼 것은?

논리 데이터베이스 설계-SEC_08(관계형 데이터 모델) 기출 및 출제 예상 문제

기출 및 출제 예상 문제(관계형 데이터 모델)

5. 다음 영문의 괄호에 적합한 Database System은?

Database management systems that process data from the perspective of a () structure use a set of two dimensional tables to represent the logical relationships among the records of a user's files.

해석 : 관계 데이터 베이스 구조의 관점에서 데이터를 처리하는 데이터베이스 관리 시스템은 2차원 테이블 집합을 사용하여 사용자 파일의 레코드 간의 논리적 관계를 나타낸다.

- ① Hierarchical Database
- ② Network Database
- ③ Relational Database
- ④ Object-oriented Database



감사합니다.