



데이터베이스

관계형 데이터 모델





학습목표

- ➔ 데이터 모델의 구성 요소를 설명할 수 있다.
- ➔ 관계형 데이터 모델의 개념을 설명할 수 있다.



학습내용

- ➔ 데이터 모델의 개념
- ➔ 관계형 데이터 모델의 개념



데이터 모델의 개념



데이터 모델

데이터 모델 $D = S, O, C$

구조
(Structure: S)

연산
(Operation: O)

제약조건
(Constraint: C)

01 구조(Structure)

- 데이터의 **정적**인 성질
- 개체 타입과 이들 간의 **관계**를 표현

02 연산(Operation)

- 데이터의 **동적**인 성질
- 개체를 처리하는 작업에 대한 **명세**, 데이터 **조작 기법**

03 제약조건(Constraint)

- 데이터의 **논리적 제약**
- 데이터 조작의 **한계**를 표현한 규정



데이터 모델의 개념



데이터 모델

04

예) 정수(Integer)

- | 구조: ..., -2, -1, 0, 1, 2, ...
- | 연산: 사칙연산(+, -, x, /)
- | 제약조건

정수/정수	정의할 수 없는 경우가 있음
3/2	1.33333: 정수가 아님



데이터 모델의 개념



데이터베이스의 구성 요소

개념적 구조

사용자 입장에서의
구성 요소

데이터베이스
= {개체, 관계}

논리적 구조

컴퓨터가 처리할 수
있는 데이터 모델

물리적 구조

저장 미디어에서 본
데이터의 물리적
배치 및 구성

데이터베이스는 데이터들의 집합

01 개념적 구조

개체(Entity)

- 표현하고자 하는 **현실 세계에 존재하는 유·무형의** 객체
 - ▶ 반드시 물질적으로 존재할 필요는 없음
 - » 예 | 수강신청, 계약
- 데이터베이스에서 표현되는 **정보의 단위**
- 개체는 **하나 이상의 속성(Attribute)**로 구성
 - ▶ 속성: 데이터의 **가장 작은** 논리적인 단위
 - ▶ 물리적인 단위: **Bit**



데이터 모델의 개념



데이터베이스의 구성 요소

01

개념적 구조

개체(Entity)

- 파일 시스템의 **레코드(Record)**에 대응되는 개념



- 개체 타입(Entity Type)

- ▶ 개체 집합에 속한 **개체 인스턴스**들이 공통으로 가지는 특징(구조)

관계(Relationship)

- 개체들 사이에 존재하는 **연관성**

» 예 | 교수와 학생 사이에는 지도 관계가 있다.

| 신랑과 신부 사이에는 결혼 관계가 있다.



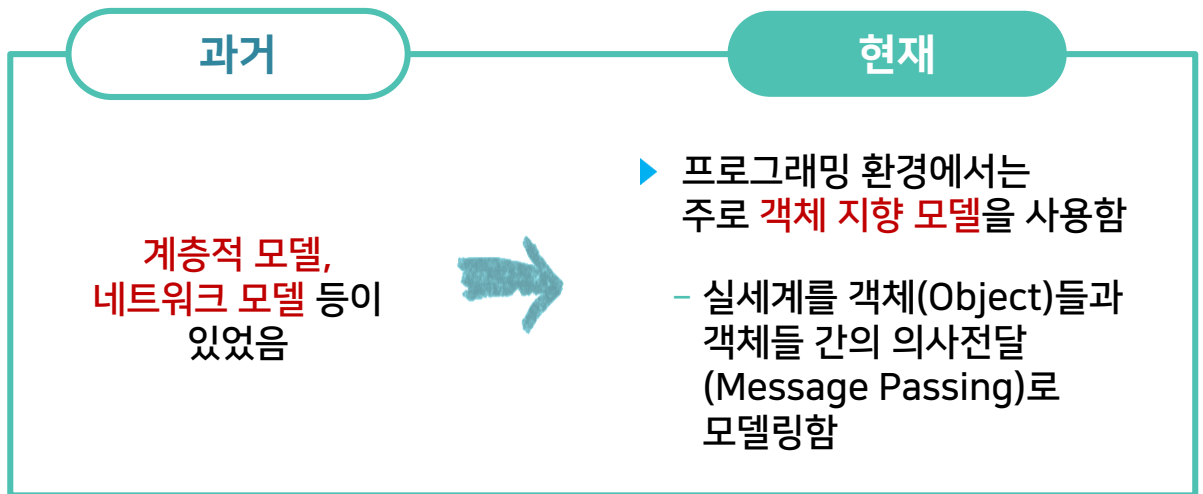
데이터 모델의 개념



데이터베이스의 구성 요소

02 논리적 구조

- 컴퓨터가 **인식**하고 **처리**할 수 있는 데이터의 논리적 구조
- 데이터베이스 관리 시스템과 사용자가 **공통으로 인식**하는 구조
 - 데이터의 **논리적 배치와 구성**을 나타냄
- 데이터베이스는 주로 관계형 데이터 모델에 기반함



- 최근 데이터베이스 관리 시스템들은 **객체 관계형 모델**을 사용





데이터 모델의 개념



데이터베이스의 구성 요소

03 물리적 구조

- 하드 디스크, SSD, DVD와 같은 저장 미디어에서 본 데이터의 **물리적인 배치 및 구성**
- 컴퓨터 위주의 표현 방식으로 저장 장치에 저장된 데이터의 **실제 구조 및 접근 방법**
- 일반 사용자나 응용 프로그래머들은 물리적 구조를 **인지하거나 파악하지 못함**

04 데이터베이스 구성 시 유의사항

- 데이터베이스의 논리적 구조는 **사용자가 숙지**하고 있어야 함
- 물리적 구조는 일반 사용자가 숙지할 필요는 없음
- 논리적 구조와 물리적 구조 사이의 **도표화(Mapping)**는 데이터베이스 관리 시스템이 지원함
 - 데이터 종속성의 해결 → 데이터 독립성



관계형 데이터 모델의 개념



관계형 데이터베이스

01 관계형 데이터 모델(Relational Data Model)

데이터베이스에서 **가장 많이** 쓰이고 있는 논리적 모델

02 관계형 데이터 모델의 특징

01 단순한 모델 구조

02 수학적 이론(집합론(Set Theory))에 기반한 모델의 강건(Sound)함

03 SQL이라는 간단한 비절차적 언어로 쉬운 데이터 처리 가능

03 집합론

창시자: 독일 수학자 칸토어(Georg Cantor)

집합의 특성

- 우리의 직관 또는 사고의 대상 중에서 **확정**되어 있고 **서로 명확히 구별**되는 것들의 모임
- **구체적·객관적 기준**에 의해 원소를 명확히 구분 가능해야 함
 - ▶ 집합을 구성하는 원소: 집합 내에서 유일하게 식별될 수 있어야 함
- 논리적인 개념도 집합의 구성 요소가 될 수 있음
 - ▶ 눈으로 보거나 손으로 만질 수 없더라도 첫 번째 조건을 충족하는 개념이라면 이 역시 집합이 될 수 있음



관계형 데이터 모델의 개념



관계형 데이터베이스

04

집합

군집(Collection)



- 동일한 유형의 원소들이 모여 있는 군집
- 하위 타입: Set, Bag, List, Array

Set



- 원소의 중복 불가
- 원소들 간의 순서 없음

Bag
(또는 Multi-Set)



- 원소의 중복 허용
- 원소들 간의 순서 없음

List



- 원소의 중복 불가
- 원소들 간의 순서 있음

Array



- 원소의 중복 허용
- 원소들 간의 순서 있음

예 | Array

```
{a, b, c}  
: set /list(o), bag/array(o), {a, b, b, c} : set/list(x),  
bag/array(o)
```

```
{a, b, c} == {b, a, c} ? set/bag(true), list/array(false)
```



관계형 데이터 모델의 개념



관계형 데이터베이스



05 관계형 데이터 모델의 구성요소

구조

- ▶ 릴레이션(또는 테이블)

연산

- ▶ 관계 대수(Relational Algebra)
- ▶ 관계 해석(Relational Calculus)

제약조건

- ▶ 무결성(Integrity) 제약조건



관계형 데이터 모델의 개념



관계형 데이터베이스

05 관계형 데이터 모델의 구성요소

- | 구조
 - 릴레이션(또는 테이블)
- | 연산
 - 관계 대수(Relational Algebra)
 - 관계 해석(Relational Calculus)
- | 제약조건
 - 무결성(Integrity) 제약조건



관계형 데이터 모델의 개념



릴레이션(Relation)

01

관계형 데이터 모델의 구조(Structure)

- 2차원 테이블 구조
- 테이블의 행(Row): 튜플(Tuple)
 - 릴레이션은 튜플들의 집합
 - 개념적 모델의 개체(Entity)가 관계형 모델에서 한 튜플에 대응됨
 - 개념적 모델의 관계(Relationship)도 관계형 모델에서 한 튜플에 대응됨
- 테이블의 열(Column): 속성(Attribute)
 - 도메인(Domain)
 - 한 속성이 가질 수 있는 값의 범위

학생

학번	이름	주소	학년
100	김진현	서울	4
101	박재영	대전	1
102	김홍연	제주	3

스키마

튜플

속성



관계형 데이터 모델의 개념



릴레이션(Relation)

02

속성(Attribute)

[관계형 데이터 모델에서 데이터의 **가장 작은** 논리적 단위]

- 관계형 모델에서는 이 데이터 값들을 더 분해하려고 해도 더 분해할 수 없는 **원자 값(Atomic Value)**만을 허용

➡ 원자 값은 아무런 내부적 구조를 지니고 있지 않음

03

도메인(Domain)

[하나의 속성이 취할 수 있는 값들의 집합]

- 도메인과 속성의 관계는 프로그래밍 언어에서 **타입과 변수에 비유**

- 어떤 데이터 타입으로 선언된 변수는 언제 어느 때고 그 선언된 데이터 타입의 값만을 가짐

➡ 예 | `int x` → 변수 `x`는 정수형 값만을 가진다.

- 속성 정의시 도메인도 함께 명세함으로써 실제 속성이 지니는 값이 **합법적인지 아닌지를 데이터베이스 관리 시스템(DBMS)을 통해 검사 가능**

- 어떤 릴레이션 `R`이 n 개의 도메인 D_1, D_2, \dots, D_n 으로 구성될 때

➡ 릴레이션 `R`은 릴레이션 스키마와 릴레이션 인스턴스로 구성

- 릴레이션 스키마: **릴레이션 이름 `R`**과 n 개의 속성 A_1, \dots, A_n 으로 구성

$R(A_1, \dots, A_n)$

▶ 여기서 속성 A_i 는 한 도메인 D_i 와 대응됨

- 릴레이션 인스턴스: 어느 한 시점에 릴레이션 `R`에 포함된 **튜플들의 집합**



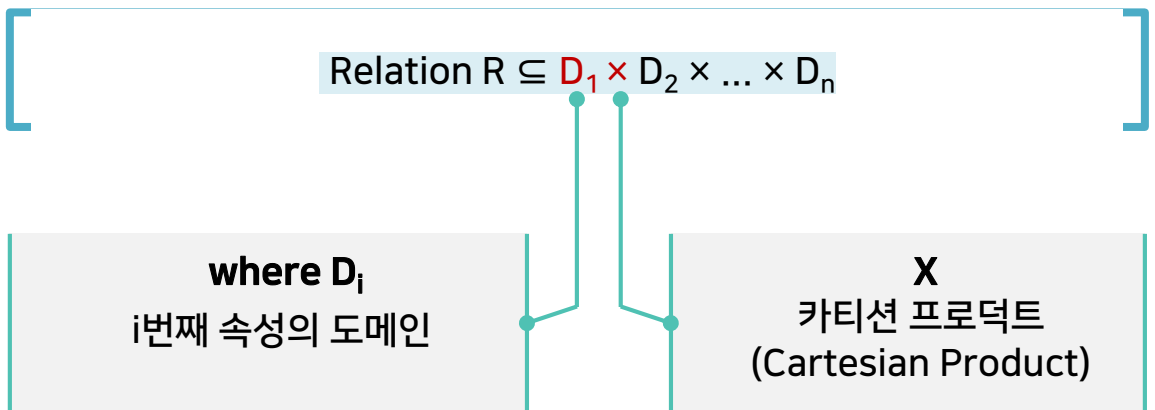
관계형 데이터 모델의 개념



릴레이션(Relation)

04

수학적 정의



■ X: 카티션 프로덕트(Cartesian Product)

- 집합론에서 사용되는 곱셈 연산
- 두 집합에 속한 원소들을 이용하여 모든 가능한 쌍 만들기

» 예 | 학년 = {1,2,3,4}
 학과 = {컴공, 전자, 기계}

학년 X 학과 =
 {(1,컴공), (2,컴공), (3,컴공), (4,컴공),
 (1,전자), (2,전자), (3,전자), (4,전자),
 (1,기계), (2,기계), (3,전자), (4,전자)}



관계형 데이터 모델의 개념



릴레이션(Relation)

04 수학적 정의

- $t \in R$: t 를 릴레이션 R 에 속한 한 튜플이라고 가정한 경우
 - t 는 n 개의 속성들로 이루어진 n -튜플 $\langle d_1, d_2, \dots, d_n \rangle$
 - 여기서, $d_i \in D_i$ ($1 \leq i \leq n$) 즉 d_i 는 도메인 D_i 에 속한 한 값
 - ▶ n : 릴레이션 R 의 차수(Degree)
 - R 의 Column 수(속성의 수)
 - ▶ $|R|$: 릴레이션 R 의 카디널리티(Cardinality)
 - R 의 Row 수(튜플의 수)

학생(학번, 학년)

(100, 3) \subseteq

(200, 4)

(300, 1)

$\text{dom}(\text{학번}) \times \text{dom}(\text{학년})$

$\{100, \dots, 999\} \times \{1, 2, 3, 4\}$

$= \{(100, 1), (100, 2), (100, 3), \dots, (999, 4)\}$

degree: 2

cardinality: 3



관계형 데이터 모델의 개념



릴레이션(Relation)

05

릴레이션의 특성



| 튜플의 유일성

- 릴레이션은 튜플의 집합
- 집합은 **중복을 허용하지 않음**

| 튜플의 무순서성

- 릴레이션은 튜플의 집합
- 집합에서 **원소들 간의 순서는 없음**

| 속성의 무순서성

- **스키마**는 속성들의 집합

| 속성의 원자성(Atomicity)

- 속성 값은 원자 값
 - ▶ 원자(Atom): 더 이상 쪼갤 수 없는 **물질의 최소 단위**
- 속성은 개체를 구성하는 가장 작은 논리적 단위
 - ▶ 논리적으로 더 분해할 수 없음



관계형 데이터 모델의 개념



릴레이션(Relation)

05

릴레이션의 특성

속성의 원자성(Atomicity)

- 한 릴레이션에 나타난 속성값은 논리적으로 더 이상 분해될 수 없는 원자 값
 - ▶ 처리상의 단위 값: **Unit Value**
- 이 성질의 근본적인 의미: 튜플의 속성값은 하나의 값만을 허용
 - ▶ 반복 그룹, 즉 **값의 집합은 허용하지 않음**
 - ▶ 반복 그룹을 허용하지 않는 릴레이션
: **정규화 릴레이션(Normalized Relation)**
 - ▶ 관계형 데이터 모델은 이 정규화 릴레이션만을 취급함



1 데이터 모델의 개념

- ✓ 데이터 모델의 구성
- ✓ 구조(Structure):
데이터의 정적인 성질로 개체 타입과 이들 간의 관계를 표현
- ✓ 연산(Operation):
데이터의 동적인 성질로 개체를 처리하는 작업에 대한 명세
- ✓ 제약조건(Constraint):
데이터의 논리적 제약으로 데이터 조작의 한계를 표현한 규정

2 관계형 데이터 모델의 개념

- ✓ 구조: 릴레이션
- ✓ 연산: 관계대수/관계 해석
- ✓ 제약조건: 무결성 제약조건
- ✓ 릴레이션은 2차원 테이블 구조로써 다음과 같은 특징을 지님
 - 튜플의 유일성, 튜플의 무순서성, 속성의 무순서성, 속성의 원자성