

데이터베이스

강사 : 강병준

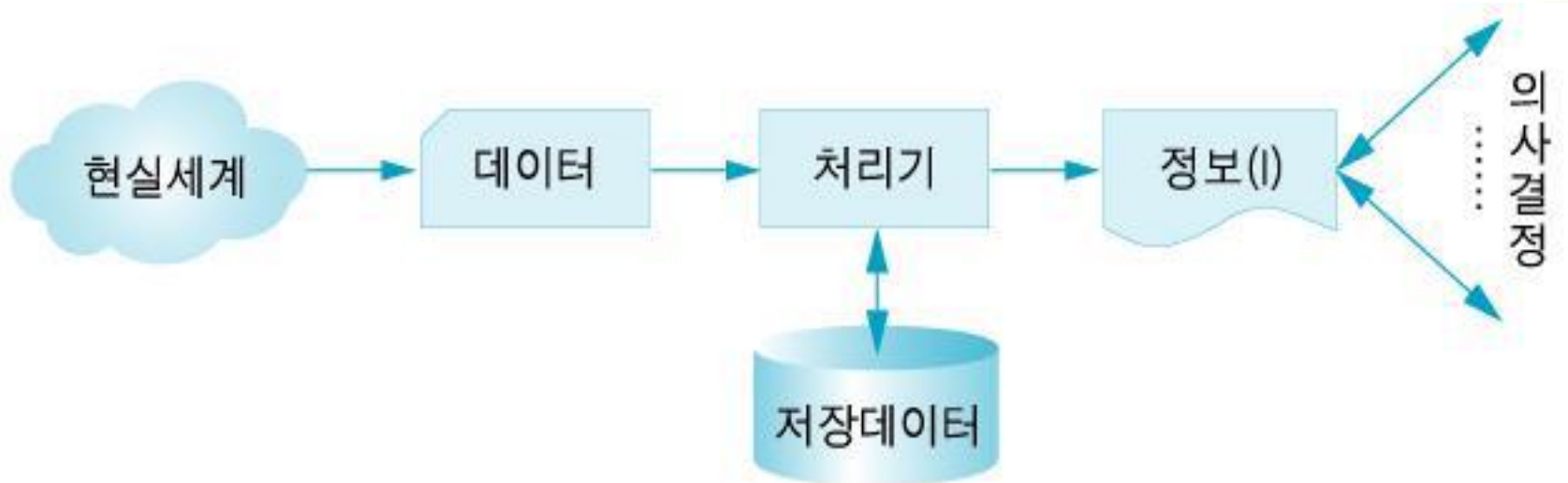
데이터란 ?

- 데이터 (**data**)
 - 관찰이나 측정을 통해 현실 세계에서 수집한 사실(**facts**)이나 값(**value**)
- 정보 (**information**)
 - 어떤 상황에서 의사 결정을 돕는
데이터의 유효한 해석이나 상호 관계
 - 데이터를 처리(**data processing**)해서 얻을 수 있는 경과



데이터란 ?

- 정보 시스템(**information system**)
 - 하나의 기관을 위해 데이터를 수집, 조직, 저장하고 정보를 생성, 분배하는 수단
 - 경영 정보 시스템, 군사 정보 시스템 등



파일시스템

데이터를 가공, 처리하여 유용한 정보를
얻기 위한 기본적인 데이터 저장 도구로
초기에 사용된 것

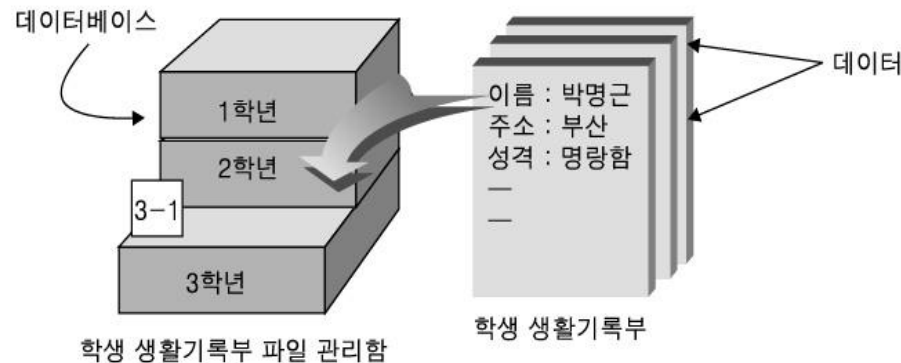
- SAM
- DAM
- ISAM
- VSAM

파일시스템의 문제점

- 데이터 종속
- 데이터 중복
 - 일관성 - 동일성을 유지하기 위해 데이터 중복을 피함
 - 보안성- 동일한 수준에서 보안이 유지
 - 경제성- 저장되는 공간에 대한 비용 절감
 - 무결성- 데이터가 정확성을 유지

데이터베이스 정의

- 데이터베이스란?
 - 논리적으로 연관된 하나 이상의 자료의 모음
 - 특정 조직의 응용 업무에 공동으로 사용하기 위하여 운영상 필요한 데이터를 중복을 최소화하여 컴퓨터 기억 장치 내에 모아 놓은 집합체
 - 데이터의 중복 없이 서로 관련되어 있어 관련된 모든 응용환경에서 사용될 수 있는 데이터들의 집합
 - 자료를 획득하여 체계적으로 분류하고 정리한 다음, 컴퓨터에서 처리가 가능하도록 전자적 형태로 저장한 것
 - 하나의 주제와 관련된 의미있는 데이터(Data)들의 모음



데이터베이스의 정의

- 통합된 데이터(integrated data)
 - 데이터베이스는 똑같은 데이터가 원칙적으로 중복되어 있지 않다는 것을 말하며, 데이터의 중복은 일반적으로 관리상의 복잡한 부작용을 초래합니다.
- 저장된 데이터(stored data)
 - 컴퓨터가 접근할 수 있는 기억장치에 저장된 데이터를 말합니다. 주로 하드디스크에 저장되어 관리됩니다.
- 운영 데이터(operational data)
 - 존재 목적이 명확하고 유용성을 지니고 있는 데이터를 말합니다. 즉 단순히 데이터를 모아둔 개념이 아닌 병원 관리를 위한 데이터 구축과 같은 목적이 분명한 데이터여야만 합니다.
- 공유 데이터(shared data)
 - 여러 사용자들이 서로 다른 목적으로 사용하는 공유 가능한 데이터를 말합니다.

데이터베이스의 특징

(1) 실시간 접근성(Real-time accessibility)

다수의 사용자의 요구에 대해서 처리 시간이 몇 초를 넘기지 말아야 한다는 의미입니다.

(2) 지속적인 변화(Continuos evolution)

데이터베이스에 저장된 데이터는 최신의 정보가 정확하게 저장되어 처리되어야 합니다.

(3) 동시 공유(Concurrent sharing)

동일 데이터를 동시에 서로 다른 목적으로 사용할 수 있어야 합니다.

(4) 내용에 대한 참조

데이터베이스 내에 있는 데이터 레코드들은 주소나 위치에 의해 참조되는 것이 아니라 가지고 있는 값에 따라 참조해야 합니다.

데이터베이스의 목적

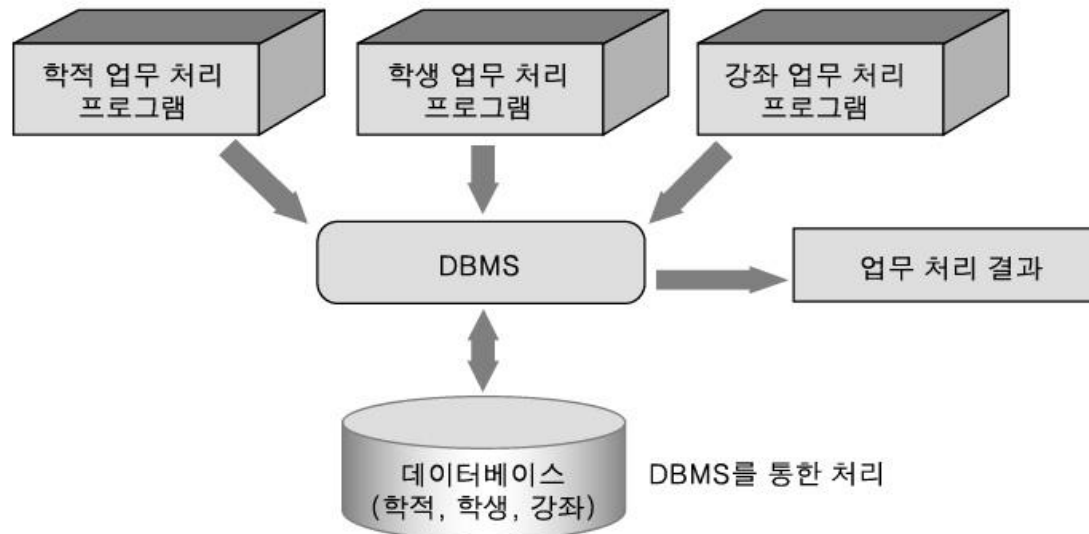
1. 데이터 독립 유지-데이터와 응용 프로그램의 독립
 - 1) 물리적 데이터 독립(physical data independence)
 - 2) 논리적 데이터 독립(logical data independence)
2. 데이터 중복의 최소화
3. 데이터의 공유
4. 데이터의 보안 유지
5. 데이터 무결성(Integrity)의 유지
6. 데이터베이스를 이용한 데이터베이스 관리 시스템은 파일 관리 시스템이 갖는 구조적인 한계를 극복하기 위해 고안
7. 데이터와 응용 프로그램의 종속성을 최소화
 - 1) 파일 관리 시스템의 가장 큰 문제인 데이터와 프로그램의 밀접한 관계를 제거하기 위해 고안
 - 2) 데이터를 저장하거나 검색하는 등의 기능을 데이터베이스 관리 시스템이 전담하도록 함으로써 파일 관리 시스템이 지닌 데이터와 응용 프로그램의 종속성을 최소화, 데이터 변경으로 인해 생기는 프로그램의 수정, 관리를 최소화.
8. 데이터의 중복으로 인한 비 일관성 최소화
 - 1) 공통된 데이터 저장 공간을 사용하기 때문에 데이터의 중복에서 발생하는 비 일관성을 최소화

Database의 역사

연도	역사
1963	데이터베이스라는 용어가 'Development and Management of Computer - Center Data Bases'라는 심포지움에서 처음 사용
1963	최초의 범용 DBMS인 설계 : GE에서 개발한 'Integrated DataStore
1970	E. F. Codd는 관계형 데이터베이스 모델 제안
1976	Chen이 개체 관계(ER) 모델 제안
1979	IBM의 System/R 프로젝트의 엔지니어들이 오라클의 전신인 Relational Software 설립함 이들이 최초의 상용 관계형 DBMS인 오라클을 출시함
1983	상용 관계 DBMS 등장(DB2, ORACLE, SYBASE 등)
1986	데이터베이스를 다루는 언어인 SQL이 관계형 데이터베이스 관리 시스템의 표준 언어로 채택
1990 ~	상용 객체 지향 DBMS 등장

DBMS

- 기업이 지속적으로 유지관리 해야 할 데이터의 집합(Database)
- 방대한 양의 데이터를 편리하게 관리하고 효율적으로 저장하고 검색할 수 있는 환경을 제공해 주는 시스템 소프트웨어(DBMS)
- 데이터베이스에 대한 사용자의 모든 요구를 수행할 수 있는 기능을 갖도록 하는 각 단계별 구조와 이들 사이의 인터페이스 및 데이터베이스 언어로 구성된 소프트웨어



데이터베이스 관리 시스템의 기능

정의 기능 (Definition)

1. 데이터의 형태, 구조, 데이터베이스의 저장에 관한 내용을 정의하며, 다양한 응용 프로그램과 데이터베이스가 서로 인터페이스 할 수 있는 방법 제공
2. 구현된 하나의 물리적 구조의 데이터베이스로 여러 사용자가 다양한 형태의 데이터를 요구해도 이를 지원할 수 있도록 가장 적절한 데이터베이스 구조를 정의할 수 있는 기능
3. 특징
 - 모든 응용 프로그램이 요구하는 데이터 구조를 지원할 수 있게끔 데이터베이스의 논리적 구조와 그 특성을 목표 DBMS가 지원하는 데이터 모델에 맞게 기술.
 - 데이터베이스를 물리적 저장장치에 저장하는 데 필요한 명세 포함.
 - 데이터의 논리적 구조와 물리적 구조 사이에 변환이 가능하도록 명세.

데이터베이스 관리 시스템의 기능

조작 기능 (Manipulation)

1. 사용자의 요구에 따라 검색, 갱신, 삽입, 삭제 등을 지원하는 기능
2. 사용자와 데이터베이스 사이의 인터페이스를 위한 수단 제공.
3. 특징
 - 1) 사용하기 쉽고 자연스러워야 한다.
 - 2) 명확하고 완전해야 한다.
 - 3) 효율적이어야 한다.

조작 기능 (Manipulation)

1. 데이터베이스의 내용에 대해 정확성과 안전성을 유지하는 기능(무결성 유지, 보안, 병행 수행 제어)
2. 특징
 - 1) 데이터베이스를 접근하는 갱신, 삽입, 삭제 작업의 정확하게 수행되게 하여 데이터의 무결성이 파괴되지 않도록 제어할 수 있어야 한다.
 - 2) 정당한 사용자가 허가된 데이터가 접근할 수 있게끔 보안을 유지하고 권한을 검사할 수 있어야 한다.
 - 3) 여러 사용자가 데이터베이스를 동시에 접근하여 데이터를 처리할 때 데이터베이스와 처리 결과가 항상 정확성을 유지하도록 병행 제어를 할 수 있어야 한다.

데이터베이스 관리 시스템의 기능

데이터베이스 관리 시스템의 장점

1. 데이터 중복의 최소화
2. 데이터 공유
3. 데이터의 일관성 유지
4. 데이터의 무결성 유지
5. 데이터의 보안 보장
6. 데이터 관리 표준화
7. 데이터 관리의 유연성
8. 자료에 대한 접근성 및 응답성 향상

데이터베이스 관리 시스템의 단점

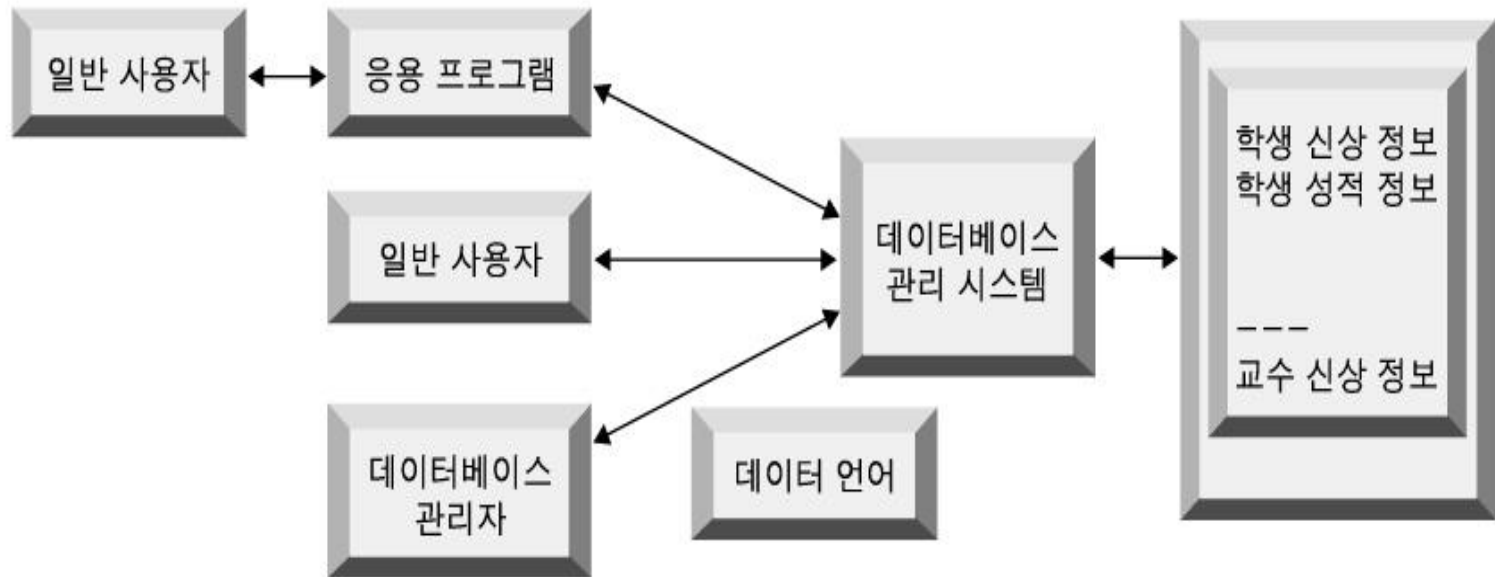
1. 추가적인 운영비 증대
2. 자료 처리의 복잡화
3. 복잡한 백업과 복구

데이터 독립성

1. 데이터베이스 구조 변화로 인한 영향을 응용 프로그램 (또는 논리적 구조)에 미치지 않도록 하는 것.
2. 종류
 - 1) 논리적 데이터 독립성
 - 2) 물리적 데이터 독립성
3. 논리적 데이터 독립성
 - 1) 응용 프로그램에 영향을 주지 않고 논리적 데이터 구조를 변경할 수 있게 하는 것
 - 2) 외부/개념 스키마간의 사상에 의해 제공
 - 3) 데이터베이스에 새로운 데이터 항목이나 레코드를 추가해도, 현재 정의된 사용자의 관점이나 사용되고 있는 응용 프로그램 중에서 직접 관련되지 않는 사용자 관점과 응용 프로그램은 영향을 받지 않는 성질
4. 물리적 데이터 독립성
 - 1) 응용 프로그램과 데이터베이스의 논리적 구조에 영향을 주지 않고 물리적 구조를 변경할 수 있게 하는 것
 - 2) 개념/내부 스키마 간의 사상에 의해 제공된다.

데이터베이스 관리 시스템 구성 요소

1. 데이터베이스
2. 데이터베이스 관리 시스템
3. 데이터 언어(Data Language)
4. 데이터베이스 관리자
5. 응용 프로그램(어)과 사용자



관계형 DBMS

- Relational Data Base Management System
- RDBMS의 구성
 - 기본적인 데이터 저장 단위는 테이블이다.
 - 로우(ROW)와 칼럼(COLUMN)으로 구성
 - 로우는 하나의 레코드이다.
 - 칼럼은 속성을 나타낸다.

데이터 모델링

개발 단계

사용자의 요구

정보에 대한 요구
(DATA)

프로그램에 대한 요구
(PROCESS)

계획

개념적 데이터 모델링
(Entity Relational Diagram)

기능 모델링
(Function Definition Diagram)

분석

논리적 데이터베이스 설계
(TABLE, INDEX, VIEW 등)

프로그램 설계
(Highrical Diagram)

설계

물리적 데이터베이스 구축

프로그램 구축

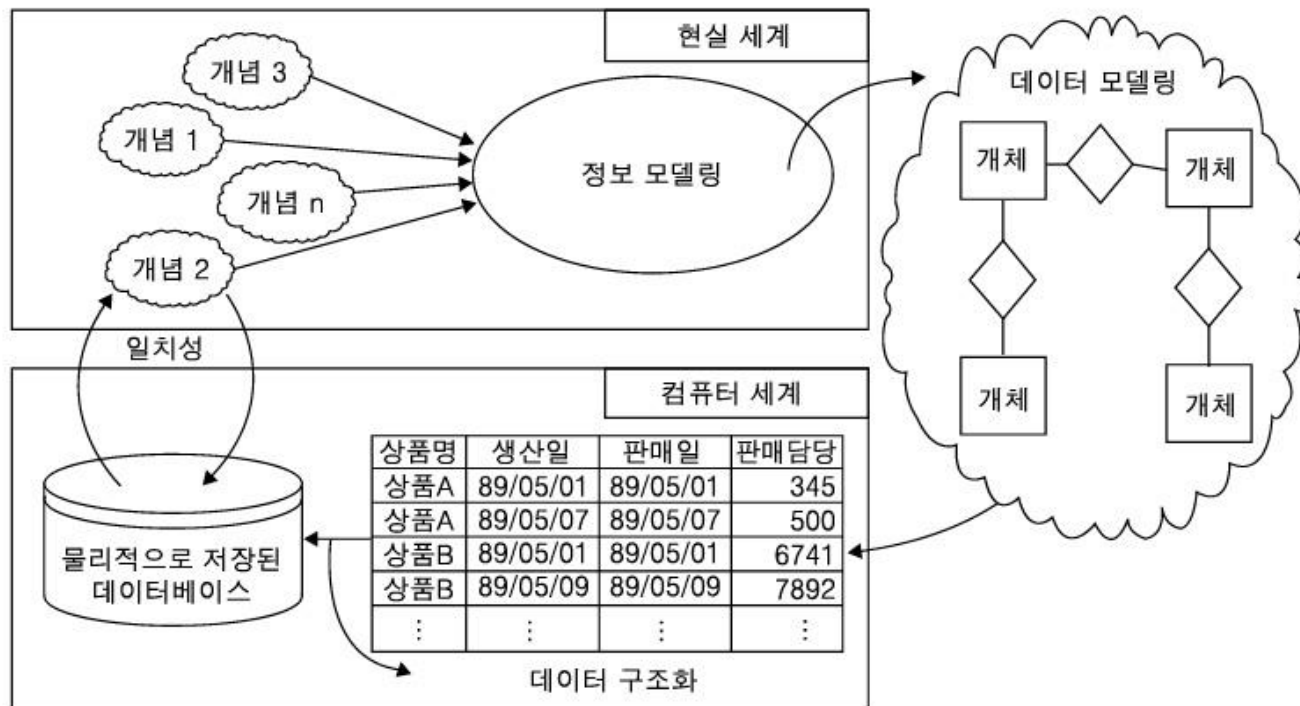
구축

정보 시스템

데이터 모델링

데이터 모델링(Data Modelling)

- 복잡한 실세계를 단순화하여 실세계에 존재하는 개체들을 식별하여 이들 객체와 객체 사이의 관계를 정의함으로써 컴퓨터상의 데이터베이스를 추상화된 개념으로 이해하기 쉽게 할 뿐만 아니라 사용자들 사이의 의사소통을 원활히 할 수 있도록 도와주는 도구



데이터 모델링

개체-관계(Entity-Relationship) 모델

- 현실 세계의 많은 데이터 중에서 관심의 대상이 되는 데이터를 언어보다 좀더 형식화된 다이어그램을 사용하여 표현한 것
- 1976년 Chen에 의해 제안
 - 데이터에 대해 관리자, 사용자, 프로그래머들이 서로 다르게 인식되고 있는 뷰들을 하나로 통합할 수 있는 단일화된 설계안을 만들기 위해서 사용.
 - 서로 다른 뷰들을 충족시킬 수 있는 데이터 처리와 제약조건 등의 요구사항들을 정의하기 위해 사용.



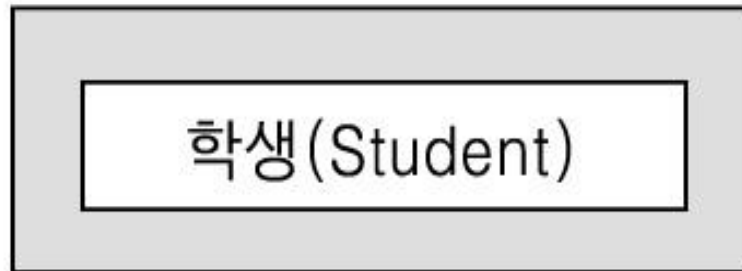
사용자의 요구로부터 개체-관계 모델의 설계

데이터 모델링

구성 요소

– 개체(엔티티, Entity)

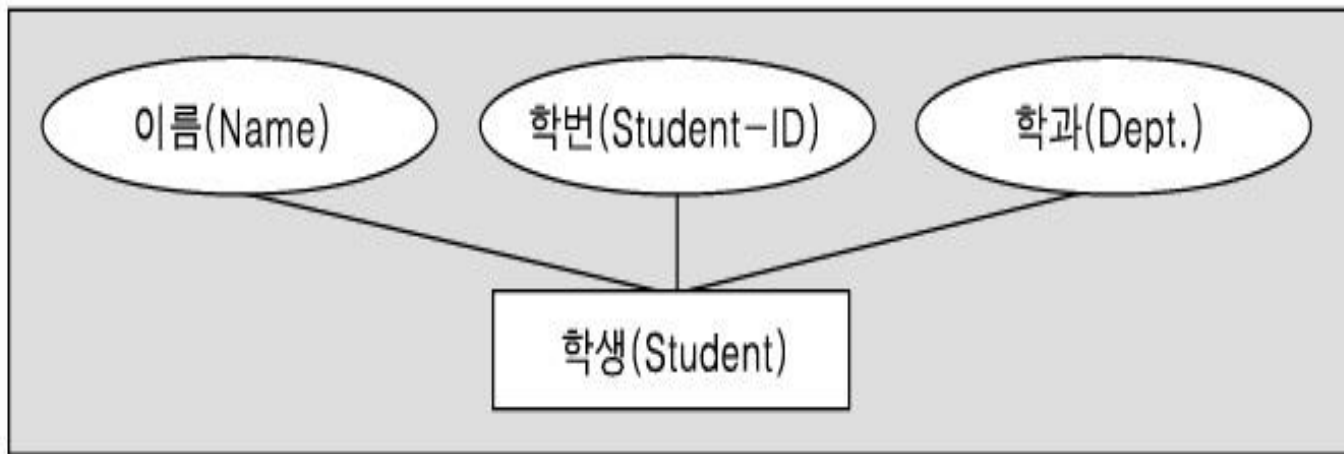
- 데이터 수집의 대상이 되는 정보 세계에 존재하는 사물(thing)
- 종류
 - 개념적 개체 : 장소, 사건 등과 같은 눈에 보이지 않는 것
 - 물리적 개체 : 물건 등과 같은 눈에 보이는 것. 즉 현실 세계에 존재하는 사물.
- 각 개체는 속성(attribute)으로 알려진 특성들로 정의.
- 개체 관계 다이어그램(ERD)에서 개체 집합은 직사각형으로 표시



데이터 모델링

속성(Attribute)

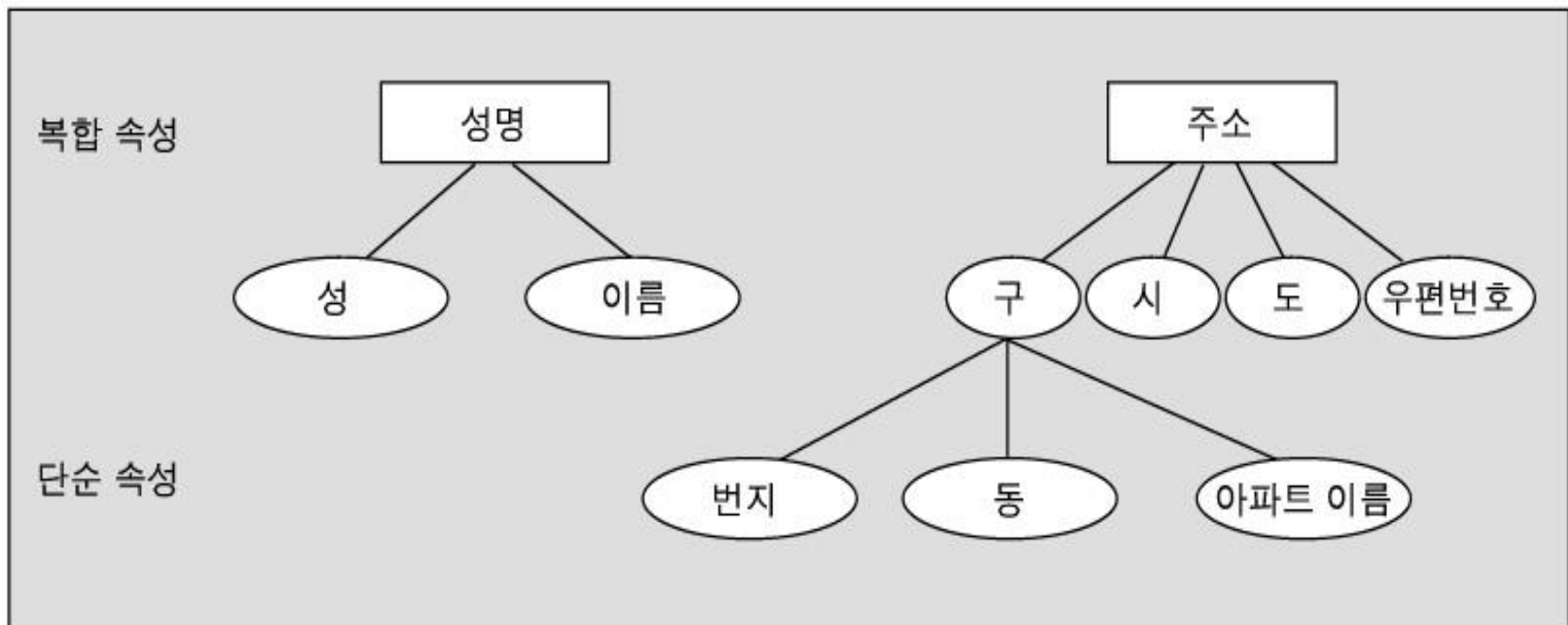
- 개체를 나타내는 특성
- ERD에서 속성은 개체 집합을 나타내는 직사각형에 실선으로 연결된 타원형으로 표현
- 도메인(domain)
 - 각 속성마다 가질 수 있는 값들의 범위



데이터 모델링

복합 속성(Composite Attribute)과 단순 속성(Simple Attribute)

- 단순 속성
더 이상 작은 구성 요소로 분해할 수 없는 속성
- 복합 속성
독립적인 의미를 좀더 기본적인 속성들로 분해할 수 있는 속성



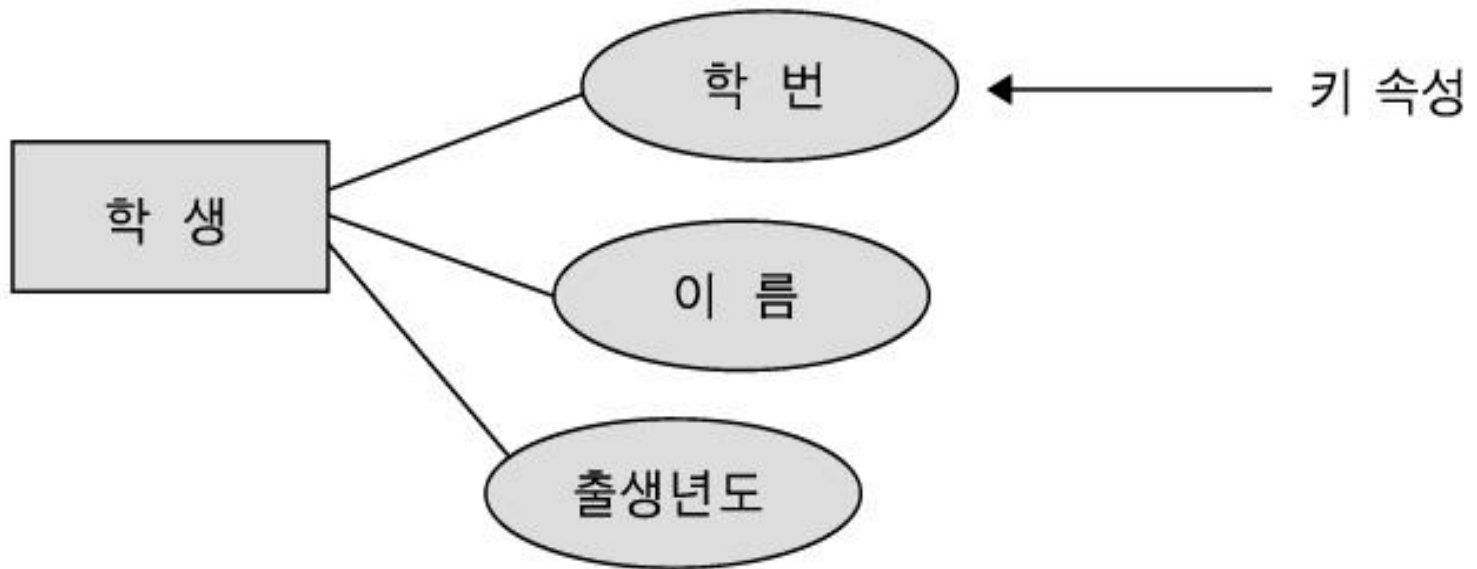
데이터 모델링

- 단일치 속성(single-valued attribute)과 다중치 속성(multi-valued attribute)
 - 단일치 속성(single-valued attribute): 개체의 속성 중 주민등록번호, 또는 학번과 같이 반드시 하나의 값만 존재하는 속성
 - 다중치 속성(multi-valued attribute): 전화번호와 같이 집, 핸드폰, 회사 전화번호와 같이 여러 개의 값을 가질 수 있는 속성.
- 다중치 속성 변경 방법
 1. 개체형 내에서 다중치 속성을 여러 개의 새로운 속성들로 분리
 2. 다중치 속성을 구성하는 속성들로 구성된 새로운 개체형 생성하고 새로운 개체 형과는 다대일 관계 설정.

데이터 모델링

- 개체의 키(Key)

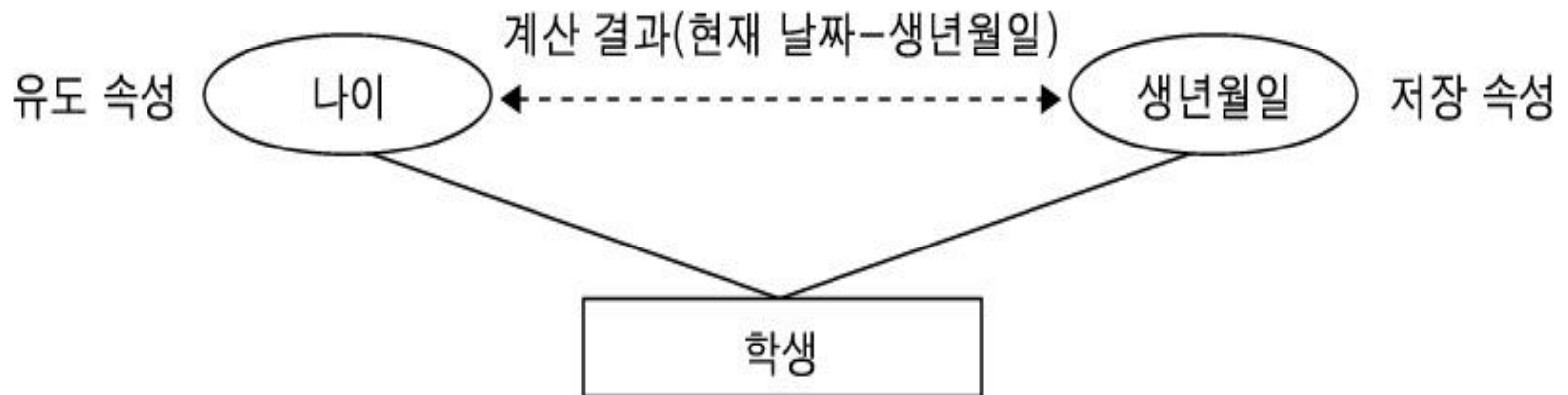
- 개체의 속성 중 하나 또는 그 이상의 속성이 개체를 다른 개체와 구별할 수 있는 속성



데이터 모델링

- 유도 속성(Drived Attribute)과 저장 속성(Stored Attribute)
 - 유도 속성: 속성의 값이 다른 관련된 속성이나 개체가 가지고 있는 값으로부터 유도되어 결정되는 속성
 - 저장 속성: 유도 속성을 결정하기 위해 사용된 속성

ERD에서 유도 속성은 점선으로 표시



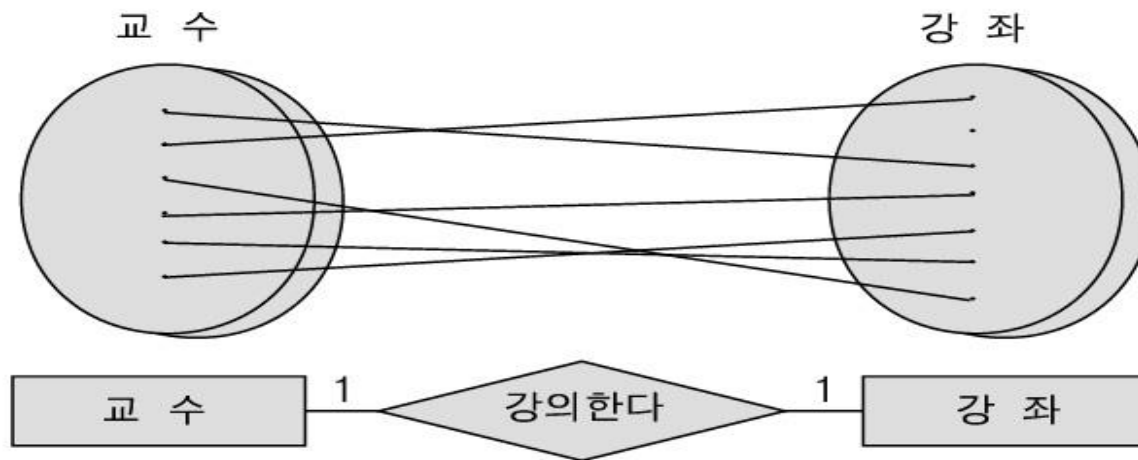
데이터 모델링

관계(Relationship)

- 개체-관계 모델에서 개체 사이의 연관성을 표현하는 개념
- ERD에서 개체들 사이의 관계는 마름모를 사용하여 표현

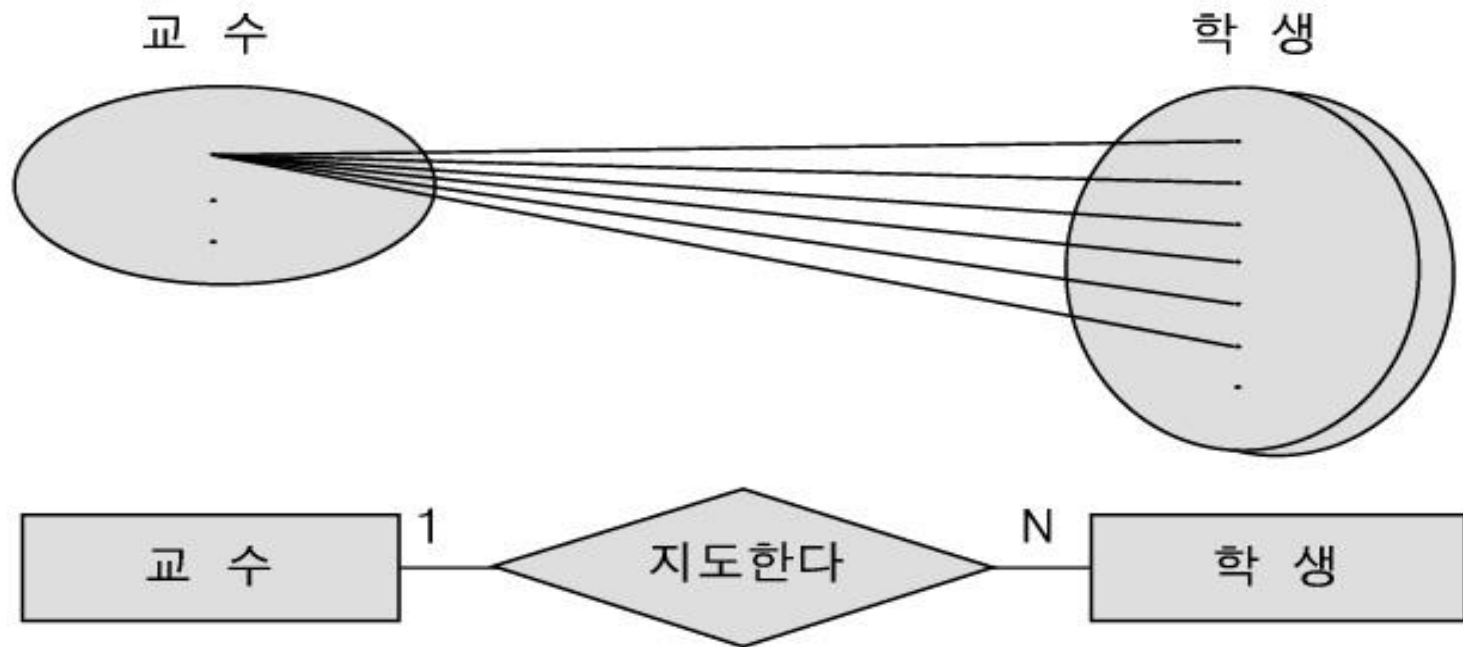


- 관계의 유형
 - 일대일 관계



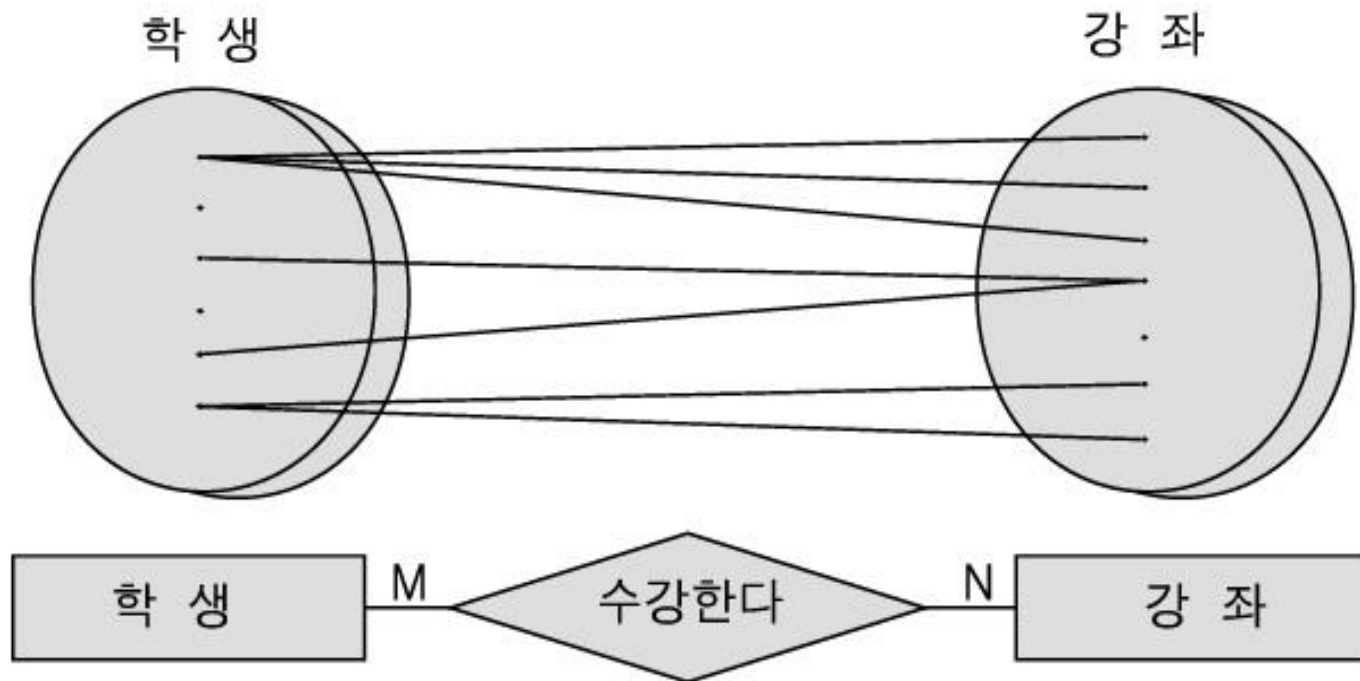
데이터 모델링

일대다 관계



데이터 모델링

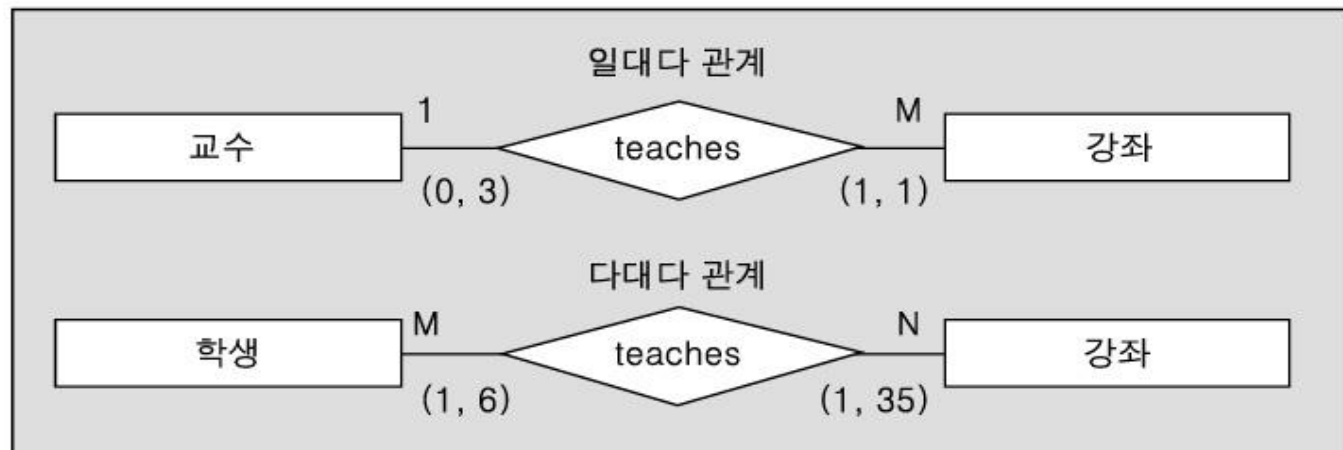
다대다 관계



데이터 모델링

관계의 대응 개체 수(Cardinality, 카디날리티)

- 관계에 참여하는 하나의 개체에 대해 다른 개체 형에서 몇 개의 개체가 참여하는지를 나타내는 것
- 예
 - 한 명의 학생이 1개 이상 6개 이하의 과목에 등록할 수 있다면 대응 개체 수는 (1, 6)
 - 대응 개체 수는 (min, max)의 한 쌍의 값으로 표현하는데 여기서 min은 관계에 참여하는 개체의 최소 개수, max는 관계에 참여하는 최대 개수 의미



데이터 모델링

존재 종속(existence-dependent)

- 한 개체의 존재가 다른 개체(들)의 존재에 영향을 받는 경우 이를 존재 종속 (existence-dependent)이라 한다.

- 예

보험 회사가 “이몽룡”이라는 사원과 그의 부양가족에게 보험 혜택을 준다고 할 때 “사원”, “부양_정보”라는 개체형들을 정의한다고 하자. “이몽룡”에게 “박하늘, 이그름, 이단비”라는 3명의 부양가족이 있다면 부양가족 3명은 “이몽룡”없이 보험 혜택을 받을 수 없다. 다시 말해 부양가족 3명의 정보는 “부양_정보”에 존재하지만 “사원”과 연관지어 지는 경우에만 존재하게 되는데, 이를 존재 종속이라 한다. 만약 “이몽룡”이 직장을 그만 두어 “사원” 테이블에서 삭제되면 부양가족 3명도 함께 “부양_정보”로부터 삭제된다.

사원		부양_정보		
사원_번호	사원_이름	사원_번호	부양가족	부양가족_이름
10111	홍길동	10111	1	최성실
10258	이몽룡	10258	1	박하늘
19658	성춘향	10258	2	이구름
		10258	3	이단비
		19685	1	김가은

데이터 모델링

관계 참여

- 선택적(optional) 관계
 - 개체형의 개체는 관계를 이루는 다른 개체형에서의 개체와 연관이 없어도 되는 관계.
 - 대응 개체 수(min, max)로도 표현하는 경우 min의 값이 0
- 의무적(mandatory) 관계
 - 그 개체형의 모든 개체는 반드시 관계를 이루는 다른 개체형에서의 개체와 연관이 있어야 하는 것 의미
 - 대응 개체 수(min, max)로도 표현하는 경우 min의 값이 1



데이터 모델링

관계 데이터 구조

- 1970년 Codd에 의해 개발
- 관계형 데이터 구조를 구성하는 용어
 - 엔티티(Entity) : 정보 저장의 기본 형태가 2차원 구조의 테이블
 - 속성(attribute) : 테이블의 각 열을 의미
 - 도메인(Domain) : 속성이 가질 수 있는 값들의 집합
 - 튜플(Tuple) : 테이블이 한 행을 구성하는 속성들의 집합

학생				튜플
학번	이름	학년	학과	
100	홍길동	4	컴퓨터	
200	이하늘	3	전기	
300	임꺽정	1	컴퓨터	
400	송호준	4	컴퓨터	
500	박현진	2	음악	
학번 도메인	이름 도메인	학년 도메인	학과 도메인	

데이터 모델링

엔티티(Entity)

- 정의

- 엔티티는 행(row)과 열(column)로 구성되는 2차원 구조
- 엔티티의 각 행은 하나의 개체를 나타내고, 릴레이션의 각 열은 개체의 각 속성 의미
- 관계형 모델에서는 행은 튜플(tuple), 열은 속성(attribute)이라는 이름 사용

- 특징

- 하나의 엔티티에 있는 튜플들은 모두 상이(distinct)해야 한다.
- 하나의 엔티티에서 튜플들의 순서와 속성들의 순서는 아무런 의미가 없다.
- 하나의 엔티티에서 같은 이름을 가진 속성들이 있을 수 없다
- 각 속성이 가질 수 있는 값들의 범위(도메인,domain)를 벗어나는 값을 가진 튜플들이 존재할 수 없다.
- 행과 열이 교차되는 곳은 원자값(atomic value)으로만 표현된다. 원자값은 논리적으로 더 이상 쪼개질 수 없는 값으로서, 여러 개의 값을 갖는 속성을 직접 표현하는 것이 불가능함을 의미한다.
- 수치형 (Numeric Type), 문자형 (Character Type) , 날짜형 (Date Type), 논리형 (Logical Type)

데이터 모델링

키 (key)

- 엔티티를 구성하는 각 튜플들을 데이터 값들에 의해 유일하게 식별할 수 있는 속성
- 후보키(Candidate Key)
 - 엔티티의 한 속성 집합(K)이 릴레이션이 전체 속성 집합 A의 부분 집합이면서 유일성(uniqueness)과 최소성(minimality)을 만족하는 경우 속성 집합(K)
 - 학번 속성
 - {이름, 학과} 속성

학생

학번	이름	학년	학과
100	홍길동	4	컴퓨터
200	이하늘	3	전기
300	임꺽정	1	컴퓨터
400	송호준	4	컴퓨터
500	박현진	2	음악

데이터 모델링

- 기본 키(Pimary Key)
 - 후보 키 중 하나의 속성만으로 엔티티의 튜플들을 유일하게 식별할 수 있는 키
- 대체키(Alternate Key)
 - 기본 키를 제외한 나머지 후보키(Candidate Key)
- 외래키(Foreign Key)
 - 하나 이상의 테이블을 연결하여 사용하는 경우 필요한 키
 - 한 테이블의 속성들의 집합으로 그 값이 다른 테이블의 주키와 일치하거나 null 값인 키를 의미

데이터 모델링

엔티티의 연결 : 릴레이션

공통된 속성을 공유하여 관계형 데이터베이스 내의 엔티티들을 연결

수강과목

학번	이름	학점	학과	과목번호
100	홍길동	4	컴퓨터	C123
200	이하늘	3	전기	C234
300	임걱정	1	컴퓨터	C236
400	송호준	4	컴퓨터	C156
500	박현진	2	음악	C234

외래 키

과목

과목번호	과목이름	학점	학과	과목이름
C123	컴퓨터개론	3	컴퓨터	임사랑
C234	프로그래밍	3	전기	정파란
C236	자료구조	3	컴퓨터	최하늘
C156	파일처리	3	컴퓨터	최정상
C198	데이터베이스	3	컴퓨터	김홍두

데이터 모델링

도메인(Domain)과 속성(Attribute)

- 속성 값(Attribute Value)
 - 개개의 속성이 가지는 데이터 값
 - 관계형 모델에서 이러한 데이터 값들은 더 이상 분해할 수 없는 원자 값만을 허용
- 도메인 (Domain)
 - 하나의 속성이 가질 수 있는 같은 타입의 모든 원자 값의 각 속성은 하나의 도메인에 대해서만 값을 사용할 수 있다.

제약 조건

- 개체 무결성(Entity Integrity)
 - 엔티티 개체들을 식별할 기본 키로 사용되는 속성이 튜플들을 유일하게 식별할 수 있도록 널 값(null)을 가질 수 없는 성질
- 참조 무결성(Referential Integrity)
 - 엔티티는 참조할 수 없는 외래키의 값을 가져서는 안 된다는 것을 의미.
 - 참조할 수 없는 외래키 값이란 널이 아니면서 참조된 엔티티의 어떤 기본키의 값과도 일치하지 않는 값 의미.

데이터 모델링

카티션 프로덕트(CARTESIAN PRODUCT, X)

- 두 엔티티의 조합 가능한 모든 릴레이션

A
A#
A1
A2
A3

B
B#
B1
B4
B5

(A TIMES B)	
A#	B#
A1	B1
A1	B4
A1	B5
A2	B1
A2	B4
A#	B#
A2	B5
A3	B1
A3	B4
A3	B5

데이터 모델링

선택트(SELECT, σ)

- 엔티티로부터 조건에 만족된 튜플들을 선택하는 연산자
- 선택트 연산 결과 구성되는 엔티티의 수평 부분 집합(horizontal subset)으로 데이터베이스 조작어의 조건절에 지정된 조건식(predicate)을 만족하는 엔티티 내의 튜플의 집합이 된다.
- $\sigma_{\langle \text{선택조건} \rangle}$ (테이블이름)

학생

학번	이름	학번	학과	점수
100	홍길동	4	컴퓨터	80
200	이하늘	3	전기	90
300	임꺽정	1	컴퓨터	85
400	송호준	4	컴퓨터	70
500	박현진	2	음악	79

$\sigma_{\text{점수} \geq 80}$ (학생)

학번	이름	학번	학과	점수
100	홍길동	4	컴퓨터	80
200	이하늘	3	전기	90
300	임꺽정	1	컴퓨터	85

데이터 모델링

프로젝트(PROJECT, π)

- 프로젝트 연산은 엔티티의 특정 속성만으로 구성된 새로운 릴레이션을 구하기 위한 연산
- 결과 릴레이션은 엔티티를 수직으로 절단한 열(column)의 집합
- 프로젝트는 엔티티의 수직적 부분 집합(vertical subset)
- 프로젝트 연산 결과로 만들어진 엔티티에 똑같은 튜플이 중복되어 존재하는 경우 시스템은 그 중 하나만 제외하고 나머지는 모두 삭제
- π 속성리스트(테이블이름)

학생

학번	이름	학년	학과	점수
100	홍길동	4	컴퓨터	80
200	이하늘	3	전기	90
300	임걱정	1	컴퓨터	85
400	송호준	4	컴퓨터	70
500	박현진	2	음악	79

π 이름, 학과 (학생)

이름	학과
홍길동	컴퓨터
이하늘	전기
임걱정	컴퓨터
송호준	컴퓨터
박현진	음악

데이터 모델링

조인(JOIN, \bowtie)

- 두 엔티티와 관련된 튜플을 하나의 튜플로 결합하는 연산
- 카티션 프로덕트(cartisian product) 연산의 결과에서 얻어진 엔티티로부터 조건에 맞는 튜플의 집합을 구하기 위한 연산
- 조인 연산은 연산자를 θ 로 표현하여 일반화하므로 θ 로 표현될 수 있는 조인을 세타 조인(θ -join)이라 한다.
- θ 가 "="인 조인 : 동일 조인 또는 이퀴 조인(equijoin)

학생

학번	이름	학점	학과
100	홍길동	4	컴퓨터
200	이하늘	3	전기
300	임꺽정	1	컴퓨터

성적

학번	과목번호	등급
100	C123	B
100	C234	A
200	C236	B
300	C156	A
300	C234	A

학생 \bowtie 학번=학번 성적

학생 · 학번	학생 · 이름	학생 · 학년	학생 · 학과	수강과목 · 과목 번호	수강과목 · 등급
100	홍길동	4	컴퓨터	C123	B
100	홍길동	4	컴퓨터	C234	A
200	이하늘	3	전기	C236	B
300	임꺽정	1	컴퓨터	C156	A
300	임꺽정	1	컴퓨터	C234	A

데이터 모델링

– θ 가 "N"인 조인 : 자연 조인(Natural Join)

학생 \bowtie_N 등록

학생 · 학번	학생 · 이름	학생 · 학년	학생 · 학과	수강과목 · 과목 번호	수강과목 · 등급
100	홍길동	4	컴퓨터	C123	B
100	홍길동	4	컴퓨터	C234	A
200	이하늘	3	전기	C236	B
300	임꺽정	1	컴퓨터	C156	A
300	임꺽정	1	컴퓨터	C234	A

데이터 모델링

디비전(DIVISION, ÷)

- 두 엔티티에 대해 A, B에 대해 B 엔티티의 튜플에 관련된 모든 튜플들을 A 엔티티로부터 구하는 것

학과목(SC)

학번	과목 번호
100	C123
100	C234
200	C236
200	C234
200	C123
300	C156
300	C234
300	C236
300	C123
400	C156
400	C234

과목_1(C1)

과목 번호
C123

과목_2(C2)

과목 번호
C234

과목_31(C3)

과목 번호
C236
C156

(디비전의 결과)

SC ÷ C1

학번
100
200
300

SC ÷ C2

학번
100
200
400

SC ÷ C3

학번
300

함수적 종속성 (Functional Dependency: FD)

1. 의미

- 1) 속성들 사이의 관계(relationship)에 대한 제약 조건
- 2) 속성 X의 값이 속성 Y의 값을 결정지으면, "속성 Y는 속성 X에 함수적으로 종속된다"고 함

2. 표기

학생

학번	이름	학년	학과
100	홍길동	4	컴퓨터
101	이하늘	3	전기
102	임꺽정	1	컴퓨터
103	이하늘	2	음악

함수적 종속성

함수적 종속성(FD, Functional Dependency)

- 애트리뷰트들 간에 존재하는 제약조건을 표현한 것

“X에서 Y로 함수적 종속성이 있다.”

$$X \rightarrow Y$$

X : 결정자(Determinant) Y : 종속자(Dependent)

함수적 종속성(FD, Functional Dependency)

- 임의의 관계형 스키마 R에서 임의의 두 튜플 t1, t2을 선택했을 때 $t1[X] = t2[X]$ 이면 반드시 $t1[Y] = t2[Y]$ 인 것이다.
- X, Y 는 R의 애트리뷰트 또는 복합 애트리뷰트이다

데이터 모델링

학생				
학번	이름	생년월일	주소	학과명
200415001	김영구	1985년 6월 1일	서울	컴퓨터공학과
200415002	홍길동	1985년 4월 12일	서울	컴퓨터공학과
200415003	임겨절	1985년 12월 25일	인천	컴퓨터공학과

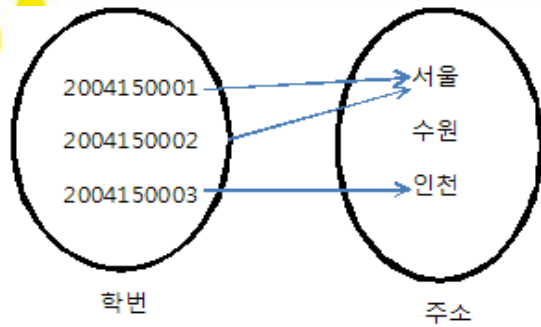
기본키 : 학번

결정자 : 학번

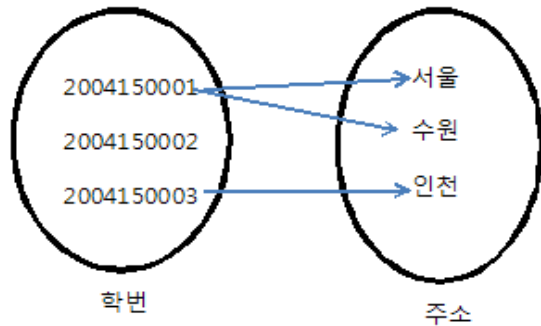
종속자 : 이름, 생년월일, 주소, 학과명

- 특징 1 : 만약 X 가 관계형 스키마 R 의 기본키 또는 후보키이면 해당 투플의 어떤 부분집합 Y 에 대해서도 $X \rightarrow Y$ 가 성립한다.
- 특징 2 : 관계형 스키마 R 에 $X \rightarrow Y$ 가 성립하더라도 $Y \rightarrow X$ 가 성립하는 것은 아니다.

함수



$X \rightarrow Y$



$X \nrightarrow Y$

• 추론 규칙

- 다른 함수적 종속성을 추론하는 과정은 이미 아는 함수적 종속성에 추론 규칙을 차례로 적용함으로써 이루어진다.
- 암스트롱의 추론 규칙 (Armstrong's Inference Rule)을 사용하면 함수적 종속성이 같은 것인지를 알 수 있다

추론 규칙(암스트롱의 추론 규칙)

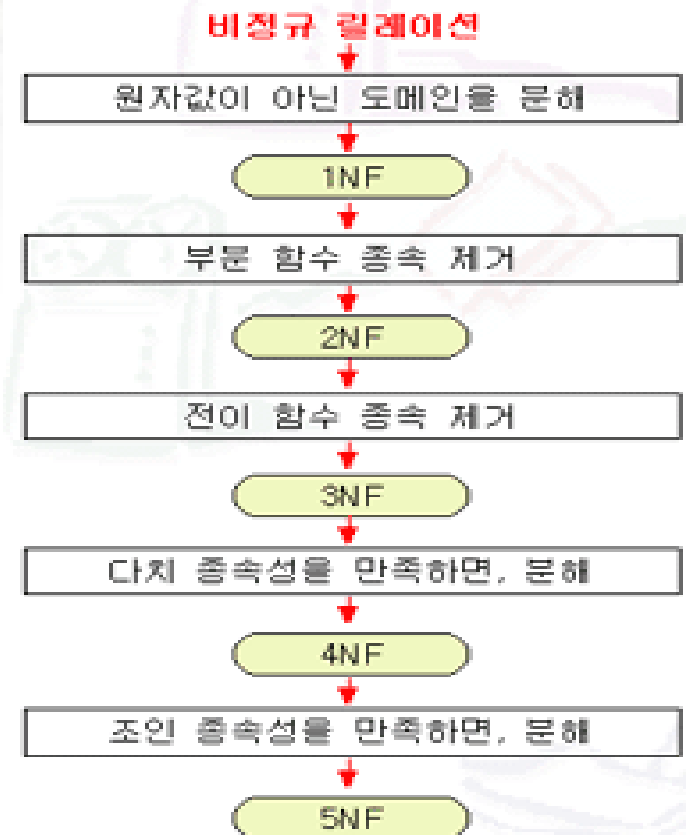
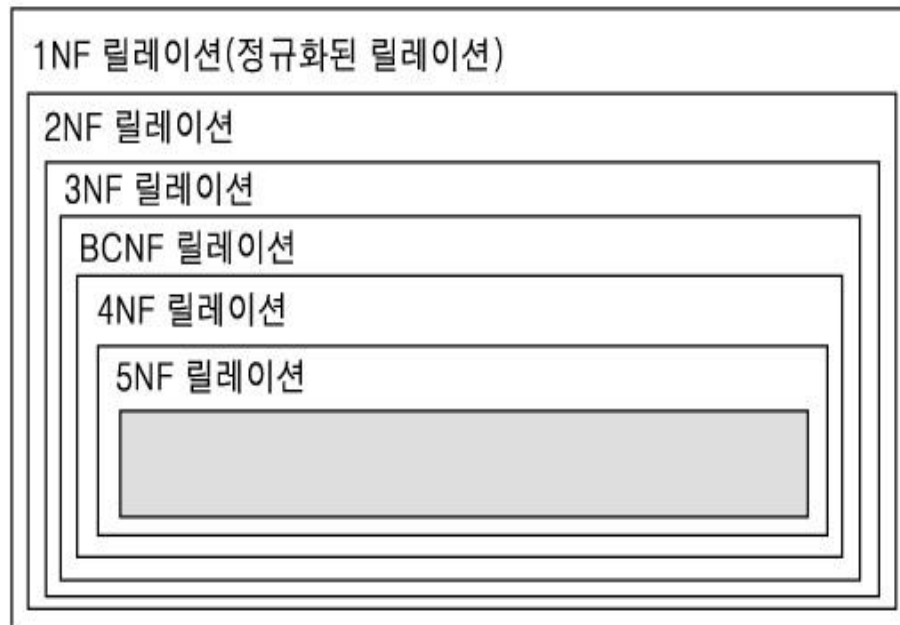
- 부가성(Augmentation) 규칙 $X \rightarrow Y$ 이면 $XZ \rightarrow YZ$ 이다.
- 이행성(Transitive) 규칙 $X \rightarrow Y$ 이고, $Y \rightarrow Z$ 이면 $X \rightarrow Z$ 이다.
- 분해(Decomposition) 규칙 $X \rightarrow YZ$ 이면 $X \rightarrow Y$ 이다.
- 합집합(Union) 규칙 $X \rightarrow Y$ 이고 $X \rightarrow Z$ 이면 $X \rightarrow YZ$ 이다.
- 의사 이행성(Pseudo-transitive) 규칙 $X \rightarrow Y$ 이고 $WY \rightarrow Z$ 이면 $WX \rightarrow Z$ 이다

정규화

정규화 형태

- 엔티티의 정규화는 실제 데이터 값이 아니라 개념적인 측면에서 다루어져야 함
- 실제 정규화 과정은 정규형의 순서와 다를 수 있음

전체 릴레이션(정규화 또는 비정규화된)



정규화

제1정규형(1NF) : 반복그룹 제거

- 엔티티에서 속성의 값은 속성의 도메인에 속하는 단일 값(원자 값, atomic Value)이어야 한다는 제약

학번	이름	학과	동아리
20013426	박하늘	컴퓨터학과	{영어회화반, 검도부}
20025914	홍길동	영문학과	{수화반, 합창반}
20038540	홍길순	음악학과	미술반
99590264	이몽룡	사회복지학과	검도부
97456123	최푸름	국어국문학과	축구부

- 반복되는 값이나 여러 값을 갖는 다치 애트리뷰트를 제거해서 각 애트리뷰트가 반드시 하나의 값만을 갖도록 하는 과정



학번	이름	학과	동아리
20013426	박하늘	컴퓨터학과	영어회화반
20013426	박하늘	컴퓨터학과	검도부
20025914	홍길동	영문학과	수화반
20025914	홍길동	영문학과	합창반
20038540	홍길순	음악학과	미술반
99590264	이몽룡	사회복지학과	검도부
97456123	최푸름	국어국문학과	축구부

정규화

• 키가 되는 속성위주로 정리

학번	이름	학과	동아리
20013426	박하늘	컴퓨터학과	{영어회화반, 검도부}
20025914	홍길동	영문학과	{수화반, 합창반}
20038540	홍길순	음악학과	미술반
99590264	이몽룡	사회복지학과	검도부
97456123	최푸름	국어국문과	축구부

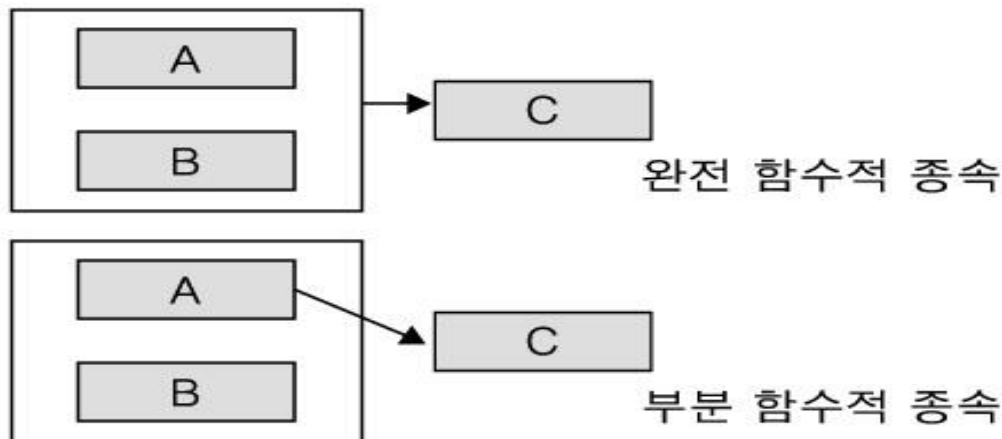


학번	동아리	이름	학과
20013426	영어회화반	박하늘	컴퓨터학과
20013426	검도부	박하늘	컴퓨터학과
20025914	수화반	홍길동	영문학과
20025914	합창반	홍길동	영문학과
20038540	미술반	홍길순	음악학과
99590264	검도부	이몽룡	사회복지학과
97456123	축구부	최푸름	국어국문과

정규화

제 2정규형(2NF) ; 완전 기능 종속

- 완전 함수적 종속성(full functional dependency)의 개념에 기반을 둔 것으로, 엔티티 R이 1NF이고 키가 아닌 모든 속성이 기본키에 완전 함수적 종속이면 엔티티 R은 2NF임.
- 완전 함수적 종속성
 - 두 속성 A와 B 사이에 $A \rightarrow B$ 의 함수적 종속성이 존재할 때, B가 A의 부분집합 A'에 함수적으로 종속되지 않는 것 의미
 - 즉, $A' \rightarrow B$ 가 성립되지 않아야 하며, 만약 $A' \rightarrow B$ 가 성립하면 부분종속(partial dependency)이라고 부른다.



정규화

공장	라인	생산품	전화		공장	전화		공장	라인	생산품
제1	A	라면	111-1111		제1	111-1111		제1	A	라면
제1	B	소주	111-1111		제2	222-2222		제1	B	소주
제1	C	맥주	111-1111					제1	C	맥주
제2	A	건포도	222-2222					제2	A	건포도
제2	B	오징어	222-2222					제2	B	오징어
키를 공장 + 라인										

정규화

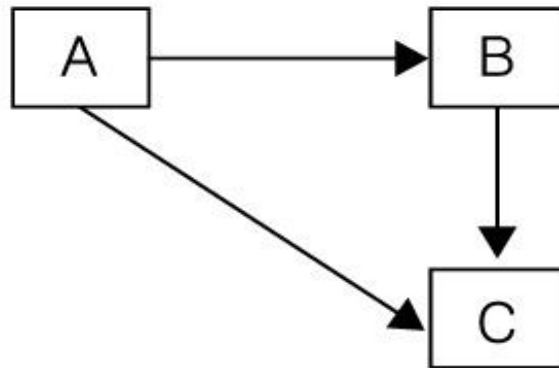
제 3정규형(3NF) : 전이 종속성

- 특징

- 추이적 종속성(transitive dependency)의 개념에 기반을 둔 정규형
- 키가 아닌 속성 값을 갱신하는 경우 불필요한 부작용(이상) 발생 없음
- 모든 이진 엔티티(2NF)는 3NF에 속함

- 추이적 종속성

- 세 속성 사이에 존재하는 함수적 종속성
- A, B, C가 한 테이블 내의 세 속성이고 $A \rightarrow B$ 와 $B \rightarrow C$ 의 함수적 종속성이 존재하면, 함수적 종속성 $A \rightarrow C$ 가 성립하는 것 의미



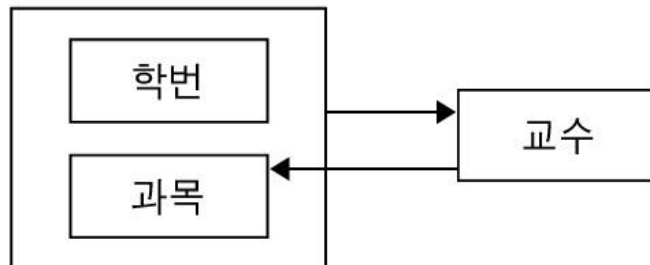
정규화

학번	학과명	전화번호		학번	학과명		컴공	1111-1111
111	컴공과	1111-1111		111	컴공과		수학	2222-2222
222	컴공과	1111-1111		222	컴공과			
333	수학과	2222-2222		333	수학과			
444	수학과	2222-2222		444	수학과			

정규화

BCNF

- 기본 키가 둘 이상의 속성으로 구성된 합성키이고, 합성키가 둘 이상 존재할 경우에 발생하는 이상현상을 방지하기 위해서 고안
- 모든 결정자가 후보키일 경우 엔티티는 BCNF형
- 엔티티 R이 BCNF에 속하면 R은 제1, 제2, 제3 정규형에 속함
- 강한 제3정규형(strong 3NF)
- 예(3NF) : 과목 엔티티
 - 과목 (학번,과목,교수)
 - 후보키 : (학번,과목), (학번,교수)
 - 기본키 : (학번,과목)
 - 함수종속 : (학번,과목) \rightarrow 교수 , 교수 \rightarrow 과목



학번	과목	교수
100	프로그래밍	P1
100	자료구조	P2
200	프로그래밍	P1
200	자료구조	P3
300	자료구조	P3
300	프로그래밍	P4

정규화

3NF(수강과목 릴레이션)에서의 이상

삽입이상: 교수 P5가 자료구조를 담당한다는 사실의 삽입은 수강 학생이 있어야 가능

삭제이상: 100번 학생이 자료구조를 취소하여 튜플을 삭제하면 P2가 담당 교수라는 정보도 삭제됨

갱신이상: P1이 프로그래밍 대신 자료구조를 담당하게 되면 P1이 나타난 모든 튜플을 변경하여야 함

⇒ 원인 : 교수가 결정자이나 후보키가 아님

⇒ 해결 : 과목 ⇒ 교수, 지도과목 엔티티로 분해

수강

학번	교수
100	P1
100	P2
200	P1
200	P3
300	P3
300	P4

지도과목

교수	과목
P1	프로그래밍
P2	자료구조
P3	자료구조
P4	프로그래밍

정규화

제 4정규형 : 조건 의존성

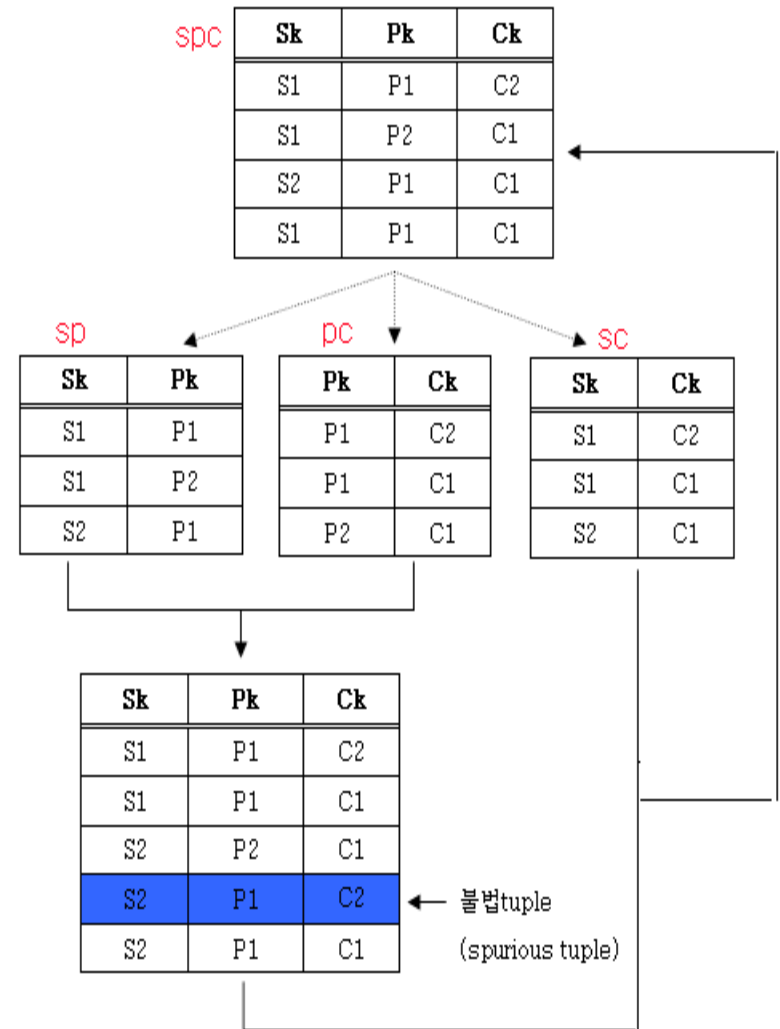
- 의미

- 엔티티 R에서 A B가 존재할 때 R의 모든 속성들이 A에 함수적 종속 (FD)이면 R은 4NF
(즉, R의 모든 애트리뷰트 X에 대해 $A \rightarrow X$ 이고 A가 후보키)
- 엔티티 R이 BCNF에 속하고 모든 MVD(다치종속성)가 FD이면 R은 4NF
- 엔티티 R이 4NF이라면 MVD가 없거나, MDV A B|C가 있을 경우 A에 대응되는 B와 C의 값은 하나씩 이어야 하며 이때 A는 후보키라는 것 의미

사번	이름	전화	자격증	사번	이름	전화	사번	자격증
1	김갑돌	111-1111		1	김갑돌	111-1111	2	정보처리가사
2	이갑돌	1111-222	정보처리가사	2	이갑돌	1111-222	4	운전면허
3	최갑돌	222-3333		3	최갑돌	222-3333		
3	전갑동	333-444	운전면허	3	전갑동	333-444		

제5정규형 : 조인 종속성

- 제 5 정규형은 주어진 테이블이 그보다 더 작은 테이블의 join으로 재구성할 수 없을 때, 즉 테이블을 분해하면 원래 가지고 있던 정보가 손실되어서, 더 이상 테이블을 분해할 수 없는 경우일 때, 그 테이블을 제 5 정규형이라고 합니다. 테이블을 칼럼들로 분해하는 것이 Projection, 다시 합치는 것이 Join, 그래서 이 둘을 합쳐 제 5 정규형을 PJNF라고도 합니다. 정보의 손실이 없이 더 작은 테이블로 분해할 수 있는가, 분해된 테이블들을 Join하면 원래의 테이블이 구성되는가에 있습니다. 원래의 테이블보다 더 작은 테이블로 분해될 수 있고, 이들을 다시 Join해서 원래의 테이블이 구성된다면, 이 테이블은 Join Dependency를 갖게 되는 것이지요.
- "릴레이션(테이블) R은 Join Dependency $*(R_1, R_2, \dots, R_n)$ 을 만족한다"
- \leftrightarrow "R의 부분집합인 각 R_i 에 대하여 R은 R_1, R_2, \dots, R_n 의 Join과 같다"



정규화

학생	수강과목	동아리
홍길동	심리학	타임반
홍길동	심리학	영화감상반
홍길동	컴퓨터	타임반
홍길동	컴퓨터	영화감상반
홍길동	교양영어	타임반
홍길동	교양영어	영화감상반
이몽룡	교양영어	영어회화반
이몽룡	교양영어	유도반
이몽룡	프로그래밍 기초	영어회화반
이몽룡	프로그래밍 기초	유도반

← BCNF

∴ (키에 속하지 않는 결정자 속성이 없음)

기본키: (학생, 수강과목, 동아리)



학생등록

수강등록

동아리등록

학생	수강과목
홍길동	심리학
홍길동	컴퓨터
홍길동	교양영어
이몽룡	교양영어
이몽룡	프로그램의기초

학생	동아리
홍길동	타임반
홍길동	영화감상반
이몽룡	타임반
이몽룡	영화감상반

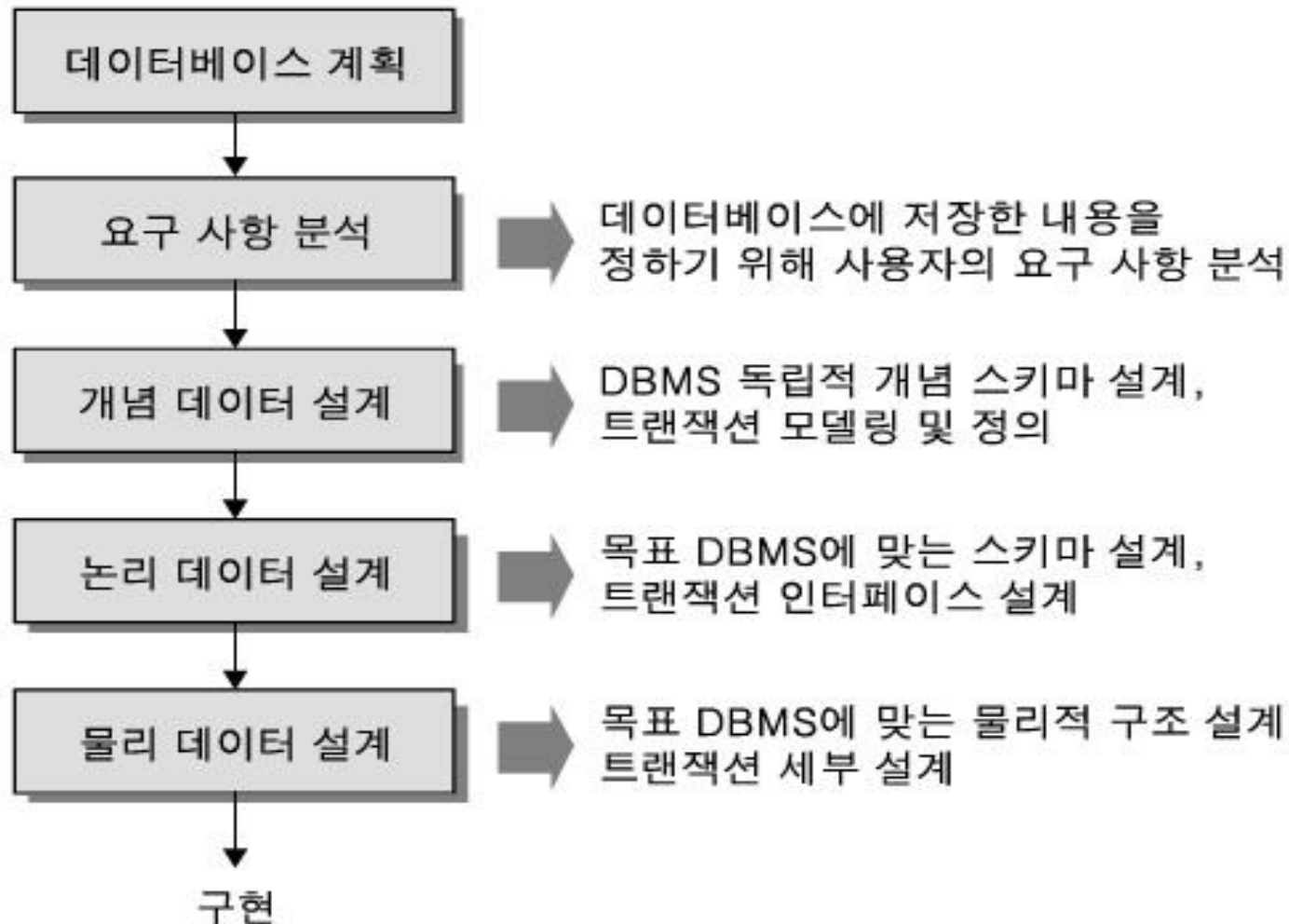
데이터베이스 설계

- 데이터베이스 설계 시 고려 사항

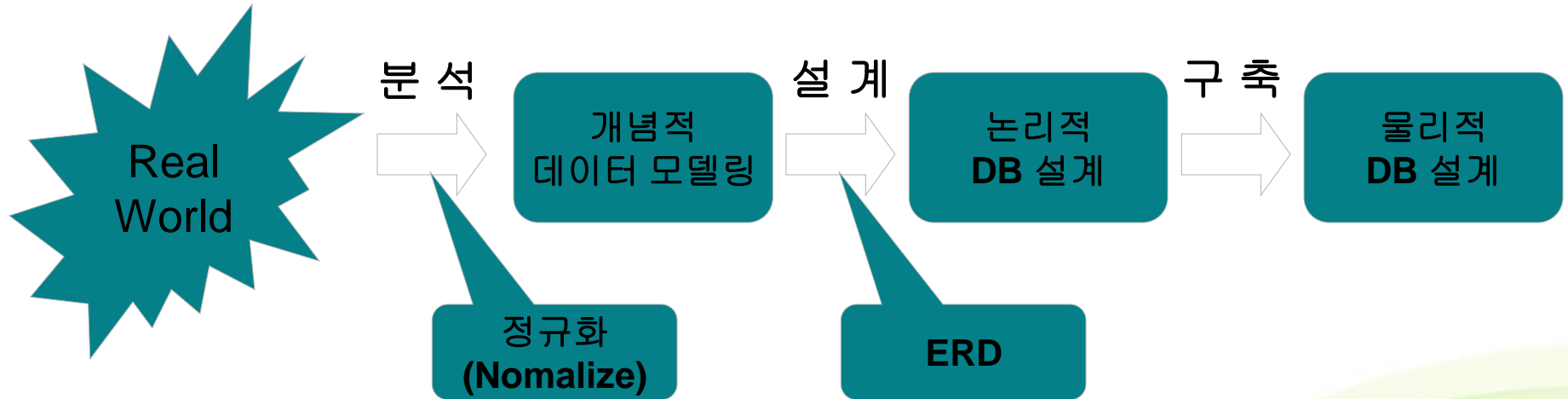
항목	설명
무결성	데이터가 만족해야 할 제약 조건 만족 (갱신, 삽입, 삭제 등의 연산 후에도 데이터 값이 정확)
일관성	저장된 데이터와 질의응답 일치하여 모순성이 없어야 함
회복	시스템에 장애 발생 시 장애 발생 전의 일관된 상태의 데이터베이스 상태로 복구 가능
보안	불법 접근(데이터의 변경, 손실, 노출)에 대한 보호 가능
효율성	응답시간 단축, 저장 공간 최적화, 시스템의 생산성(처리도) 등을 고 려
데이터베이스 확장	응용과 데이터의 확대(시스템에 영향을 주지 않고, 새로운 데이터 추 가 가능)

데이터베이스 설계

- 데이터베이스 설계 과정



데이터베이스 설계



- 1) 사람 : 사원, 회원
- 2) 사건 : 계약, 작업
- 3) 장소 : 창고, 지역
- 4) 개념 : 목표, 계획

- 1) Entity
 - 관리대상
- 2) Attribute
 - 관리대상의 구체적 내용
- 3) Identifier
 - Entity를 대표하는 Att.
- 4) Tuple
 - 여러 개의 Att.의 집합
- 5) Relation
 - Entity와 Entity의 관계

- 1) TABLE
- 2) COLUMN
- 3) PRIMARY-KEY
- 4) ROWS
- 5) FOREIGN-KEY

SQL

- SQL(Structured Query Language)
 - 1974년 IBM 연구소에서 발표된 SEQUEL(Structured English QUery Language)에서 유래
 - 특징
 - SQL은 비절차적인
 - 대화식 언어로 사용 가능
 - 다른 종류의 범용 프로그래밍 언어로 작성된 프로그램에 내장(embed)시킨 형태로도 사용 가능
 - 각각의 튜플 단위가 아니라 튜플들을 집합 단위로 처리

SQL의 특징

관계대수와 관계해석을 기초로 한 고급 데이터 언어
이해하기 쉬운 형태로 표현
대화식 질의어로 사용 가능
데이터 정의, 데이터 조작, 제어 기능 제공
COBOL, C, PASCAL 등이 언어에 삽입
레코드 집합 단위로 처리
비절차적 언어

SQL

- SQL의 명령어

- DDL (Data Definition Language) : 데이터베이스 및 테이블의 구조를 정의하거나 변경

SQL문	내 용
CREATE	데이터베이스 및 객체 생성
DROP	데이터베이스 및 객체 삭제
ALTER	기존에 존재하는 데이터베이스 객체를 변경

- DML (Data Manipulation Language) : 데이터의 삽입, 삭제, 검색과 수정 등을 처리

SQL문	내 용
INSERT	데이터베이스 객체에 데이터를 입력
DELETE	데이터베이스 객체에 데이터를 삭제
UPDATE	기존에 존재하는 데이터베이스 객체안의 데이터 수정
SELECT	데이터베이스 객체로부터 데이터를 검색

SQL

- **DCL (Data Control Language) : 데이터베이스 사용자의 권한을 제어**

SQL문	내 용
GRANT	데이터베이스 객체에 권한 부여
REVOKE	이미 부여된 데이터베이스 객체의 권한 취소

ERD작성 툴

Amateras ERD

<http://takezoe.github.io/amateras-update-site/>

eXERD

http://www.exerd.com/ko_KR/

er win

<https://erwin.com/>