데이터베이스의 기본

데이터베이스 (DB, DataBase)

일정한 규칙 혹은 규약을 통해 구조화되어 저장되는 데이터의 모음

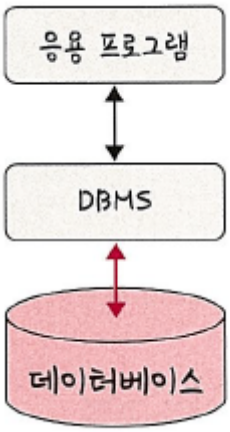
DBMS (DataBase Management System)

DB를 제어, 관리하는 통합 시스템

DB 내의 데이터들은 특정 DBMS 마다 정의된 쿼리 언어를 통해 CRUD 등을 수행할 수 있음

또한, 실시간 접근과 동시 공유가 가능

아래와 같은 구조를 기반으로 데이터를 주고 받음



엔터티 (Entity)

사람, 장소, 물건, 사건, 개념 등 여러개의 속성을 지닌 명사

속성은 서비스의 요구사항에 맞춰 정해짐

엔터티는 약한 엔터티와 강한 엔터티로 나뉨

ex) A 가 혼자서는 존재하지 못하고, B의 존재 여부에 따라 종속적인 경우

A 는 약한 엔터티, B 는 강한 엔터티

A 가 방, B 를 건물이라고 생각

릴레이션 (Relation)

DB에서 정보를 구분하여 저장하는 기본 단위

엔터티에 관한 데이터를 DB에서는 릴레이션 하나에 담아 관리함

RDB 에서는 이를 테이블이라함

NoSQL 에서는 컬렉션이라함

속성 (Attribute)

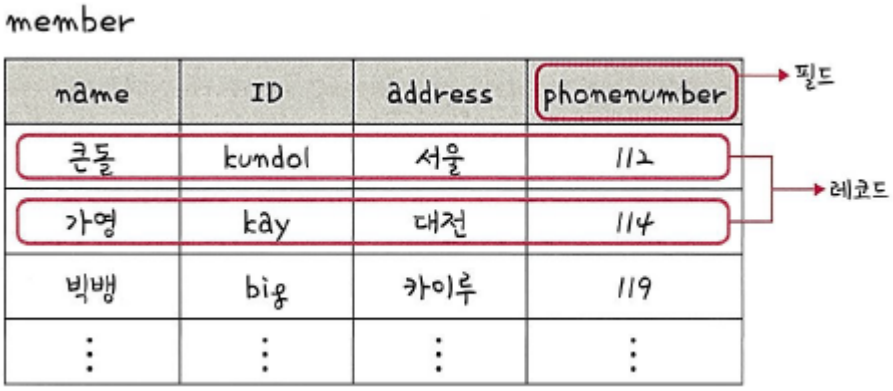
릴레이션에서 관리하는 구체적이며 고유한 이름을 갖는 정보

서비스의 요구 사항을 기반으로 관리해야할 필요가 있는 속성들만 엔터티의 속성이 됨

도메인 (Domain)

릴레이션에 포함된 각각의 속성들이 가질 수 있는 값의 집합

필드와 레코드



회원이라는 엔터티는 member 라는 테이블로 속성인 이름, 아이디 등을 가짐

name, ID, address 등의 필드를 가짐

레코드(Record) : 이 테이블에 쌓이는 행(row) 단위의 데이터. **튜플(tuple)** 이라고도함

필드 타입

MySQL 기준

숫자 타입

TINYINT, SMALLINT, MEDIUMINT, INT, BIGINT 등이 있음

타입	용량(바이트)	최솟값(부호 있음)	최솟값(부호 없음)	최댓값(부호 없음)	최댓값(부호 있음)
TINYINT	1	-128	0	127	255
SMALLINT	2	-32768	0	32767	65535
MEDIUMINT	3	-8388608	0	8388607	16777215
INT	4	-2147483648	0	2147483647	4294967295
BIGINT	8	-263	0	263-1	264-1

날짜타입

DATE, DATETIME, TIMESTAMP 등이 있음

- **DATE**

날짜 부분만 있는 값에 사용

3바이트의 용량을 가짐

- **DATETIME**

  날짜 및 시간 부분을 모두 포함하는 값에 사용됨

  8바이트의 용량을 가짐

- **TIMESTAMP**

  날짜 및 시간 부분을 모두 포함하는 값에 사용됨

  4바이트의 용량을 가짐

문자 타입

CHAR, VARCHAR, TEXT, BLOB, ENUM, SET 이 있음

- **CHAR 와 VARCHAR**
  - **CHAR**

    고정 길이 문자열

    테이블 생성 시 선언한 길이로 고정

    레코드 저장 시 무조건 선언한 길이 값으로 고정해서 저장됨
  - **VARCHAR**

    가변 길이 문자열

    입력된 데이터에 따라 용량을 가변시켜 저장함

  지정된 형태에 따라 저장된 CHAR 의 경우, 검색에 유리

  검색이 잦지 않고 유동적인 길이를 가진 데이터는 VARCHAR 로 저장하는 것이 좋음

- **TEXT 와 BLOB**

  두 타입 모두 큰 데이터 저장 시 사용

- **TEXT** : 큰 문자열 저장에 사용 (주로 게시판의 본문)
- **BLOB** : 이미지, 동영상 등 큰 데이터 저장에 사용 (일반적으로 파일은 VARCHAR 로 경로만 저장함)

- **ENUM 과 SET**
  - **ENUM**

    ENUM('x-small', 'small', 'medium', 'large', 'x-large') 형태로 사용

    이중에서 하나를 선택하는 단일 선택만 가능

    ENUM 리스트에 없는 잘못된 값 삽입 시 빈 문자열이 삽입

    메모리를 적게 사용하는 이점을 얻음

    최대 65,535 개의 요소들을 넣을 수 있음

  - **SET**

    ENUM 과 비슷하나, 여러 개의 데이터 선택이 가능하고 비트 단위 연산이 가능함

    최대 64개의 요소를 집어넣을 수 있음

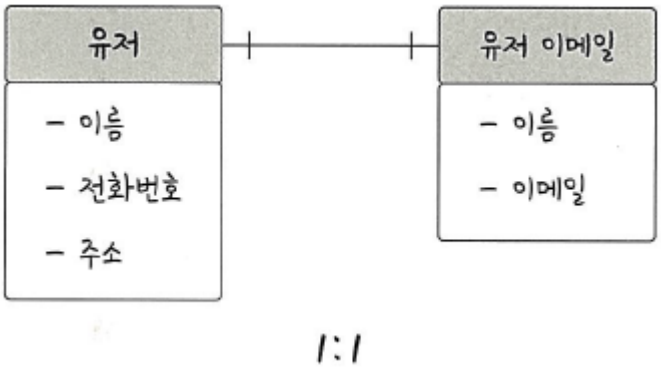
  공간적으로 이점을 볼 수 있으나, 애플리케이션의 수정에 따라 DB 의 ENUM 이나 SET 에서 정의한 목록을 수정해야한다는 단점이 있음

관계

DB 에는 여러개의 테이블이 있고, 이러한 테이블은 서로의 관계가 정의되어 있음

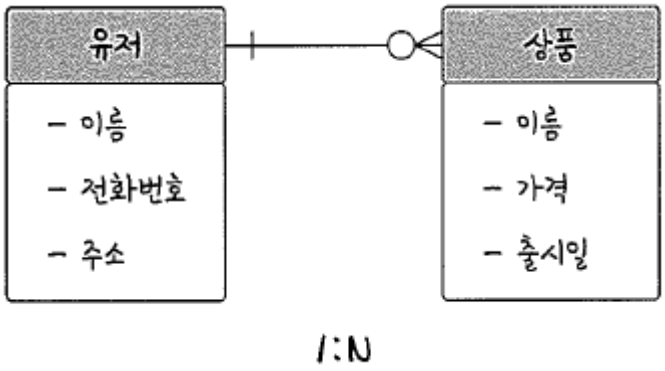
관계화살표로 나타냄

1:1 관계



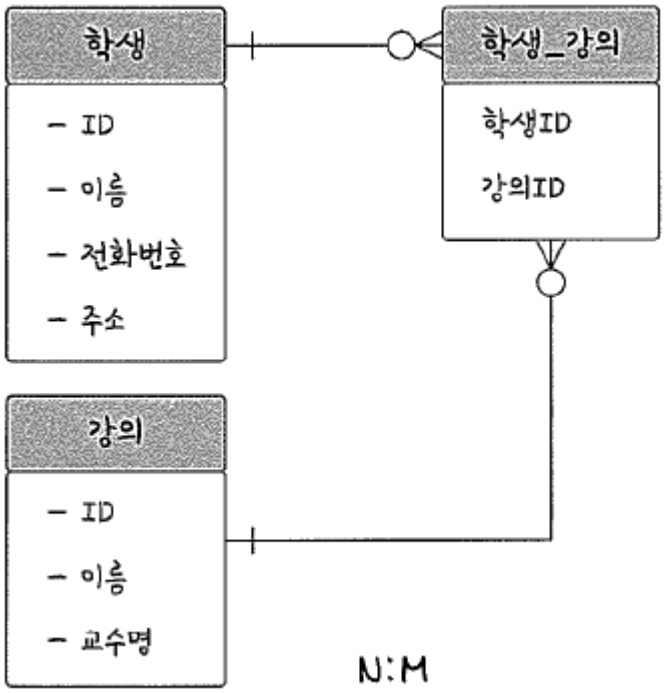
테이블을 두개의 테이블로 나눠 테이블의 구조를 더 이해하기 쉽게 만들어 줌

1:N 관계



한 개체가 다른 많은 개체를 포함하는 관계

N:M 관계



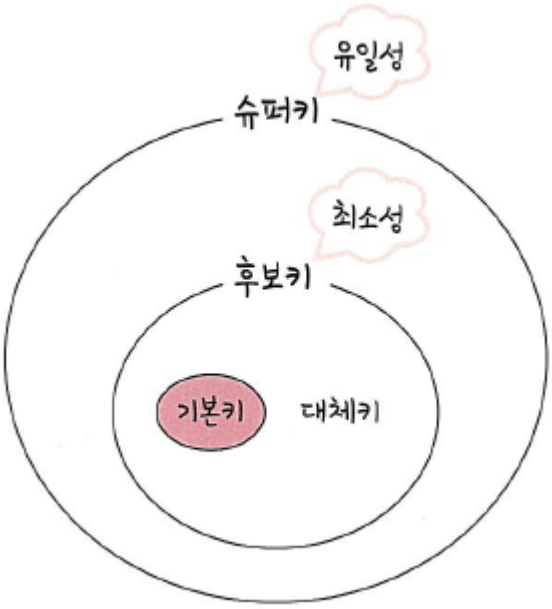
N:M 은 테이블 두개를 직접적으로 연결해서 구축하기 보다는 1:N, 1:M 이라는 관계를 갖는 두 개의 테이블로 나눠서 설정

키

테이블 간의 관계를 좀 더 명확하게 하고 테이블 자체의 인덱스를 위해 설정된 장치

기본키, 외래키, 후보키, 수퍼키, 대체키가 있음

키 간의 관계



슈퍼키 : 유일성

후보키 : 최소상까지 갖춤

대체키 : 후보키 중 슈퍼키가 되지 못한 키

기본키 (Primary Key)

유일성과 최소성을 만족하는 키

자연키 또는 인조키 중에 골라 설정

자연키

중복된 값들을 제외하며 중복되지 않는 것을 자연스레 뽑다가 나오는 키

언젠가는 변하는 속성을 가짐

ex) 주민등록번호

인조키

인위적으로 생성한 키

자연키와 달리 변하지 않음

일반적으로 기본키는 인조키로 설정함

ex) ID

외래키 (Foriegn Key)

다른 테이블의 기본키를 그대로 참조하는 값

개체와의 관계 식별에 사용됨

중복되어도 상관없음

후보키 (Candidate Key)

기본키가 될 수 있는 후보들

유일성과 최소성을 동시에 만족하는 키

대체키 (Alternate Key)

후보키가 두 개 이상일 경우, 어느 하나를 기본키로 지정하고 남은 후보키들

슈퍼키 (Super Key)

각 레코드를 유일하게 식별할 수 있는 유일성을 갖춘 키