

4.1 데이터베이스의 기본

데이터베이스

데이터베이스(DB, DataBase)는 일정한 규칙, 혹은 규약을 통해 구조화되어 저장되는 데이터의 모음입니다. 해당 데이터베이스를 제어, 관리하는 통합 시스템을 DBMS(DataBase Management System)라고 하며, 데이터베이스 안에 있는 DBMS마다 정의된 쿼리 언어(query language)를 통해 삽입, 삭제, 수정, 조회등을 할 수 있습니다.

4.1.1 엔터티

엔터티(entity)는 사람, 장소, 물건, 사건, 개념 등 여러 개의 속성을 지닌 명사를 의미합니다.

약한 엔터티와 강한 엔터티

A가 혼자서는 존재하지 못하고 B의 존재 여부에 따라 종속적이라면 A는 약한 엔터티이고 B는 강한 엔터티가 된다.

4.1.2 릴레이션

릴레이션(relation)은 데이터베이스에서 정보를 구분하여 저장하는 기본 단위입니다. 엔터티에 관한 데이터를 데이터베이스는 릴레이션 하나에 담아서 관리합니다.

릴레이션은 관계형 데이터베이스에서는 ‘테이블’이라고 하며, NoSQL 데이터베이스에서는 ‘컬렉션’이라고 합니다.

관계형 데이터베이스 예) MySQL

NoSQL 데이터베이스 예) MongoDB

MySQL의 구조 : 레코드-테이블-데이터베이스

NoSQL의 구조: 도큐먼트-컬렉션-데이터베이스

4.1.3 속성

속성(attribute)은 릴레이션에서 관리하는 구체적이며 고유한 이름을 갖는 정보입니다.

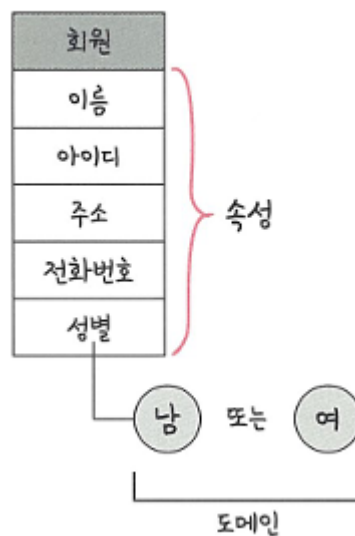
ex) 차의 속성 - 차 넘버, 바퀴 수, 차 색깔, 차종 등

4.1.4 도메인

도메인(domain)이란 릴레이션에 포함된 각각의 속성들이 가질 수 있는 값의 집합을 말합니다.

ex) 성별이란 속성의 도메인 - 남, 여

▼그림 4-5 속성과 도메인



4.1.5 필드와 레코드

▼그림 4-6 필드와 레코드

member

name	ID	address	phonenumber
큰돌	kundol	서울	112
가영	kay	대전	114
빅뱅	big	카이루	119
⋮	⋮	⋮	⋮

→ 필드 (points to the header row)

→ 레코드 (points to the data rows)

회원이란 엔터티는 member라는 테이블로 속성인 이름, 아이디 등을 가지고 있으며 name, ID, address 등의 필드를 가집니다. 그리고 이 테이블에 쌓이는 행 단위의 데이터를 레코드라고 합니다. 또한 레코드를 튜플이라고도 합니다.

필드 타입

- 숫자 타입
 - TINYINT, SMALLINT, MEDIUMINT, INT, BIGINT 등이 있다.

▼ 표 4-1 MySQL 숫자 타입

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

- 날짜 타입
 - DATE - 날짜 부분은 있지만 시간 부분은 없는 값에 사용됨. 3바이트의 용량 1000-01-01 ~ 9999-12-31
 - DATETIME - 날짜 및 시간 부분을 모두 포함하는 값에 사용됨 8바이트의 용량 DATE + 00:00:00~23:59:59
 - TIMESTAMP - 날짜 및 시간 부분을 모두 포함하는 값에 사용됨 4바이트의 용량 1970-01-01 00:00:00 ~ 2038-01-19 03:14:07

- 문자 타입

문자는 CHAR 또는 VARCHAR로 구분 둘다 숫자를 입력해서 범위를 정한다

하지만 CHAR는 범위 지정시 지정한 만큼 데이터가 저장되며 VARCHAR은 범위를 넓게 지정해도 사용을 적게 하면 적게 한만큼 +1바이트의 데이터가 저장된다.

TEXT - 큰 문자열 저장에 쓰이며 주로 게시판의 본문을 저장할 때 사용

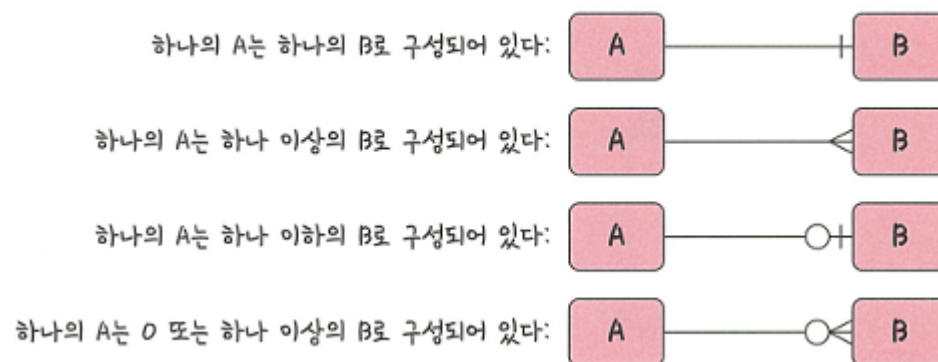
BLOB - 이미지, 동영상 등 큰 데이터 저장에 사용, 그러나 보통은 아마존의 이미지 호스팅 서비스인 S3을 이용하는 등 서버에 파일을 올리고 파일에 관한 경로를 VARCHAR로 저장함.

ENUM - ENUM('x-small', 'small', 'medium', 'large', 'x-large') 형태로 쓰이며, 이 중에서 하나만 선택하는 단일 선택만 가능하고 ENUM 리스트에 없는 잘못된 값을 삽입하면 빈 문자열이 대신 삽입됩니다. ENUM을 이용하면 x-small등이 0, 1 등으로 매핑되어 메모리를 적게 사용하는 이점을 얻습니다. ENUM은 최대 65535개의 요소들을 넣을 수 있습니다.

SET - ENUM과 비슷하지만 여러 개의 데이터를 선택할 수 있고 비트 단위의 연산을 할 수 있으며 최대 64개의 요소를 집어넣을 수 있다는 점이 다릅니다.

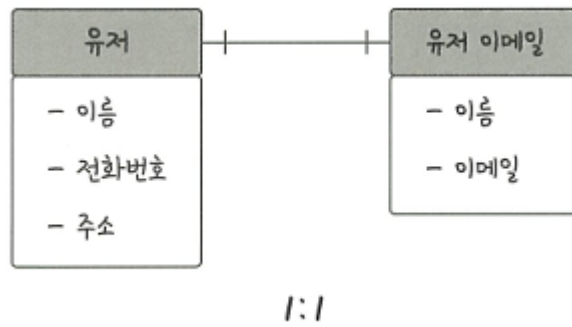
4.1.6 관계

▼ 그림 4-9 관계화살표



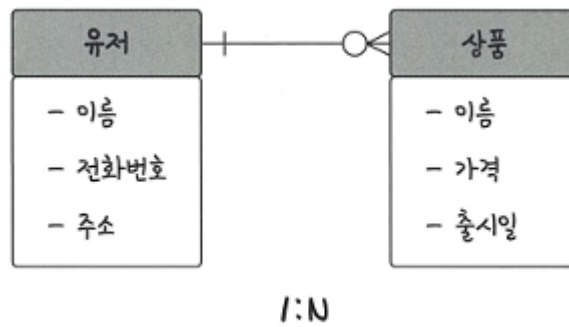
1:1 관계

ex) 유저와 이메일 - 유저당 유저 이메일은 하나씩만 있다



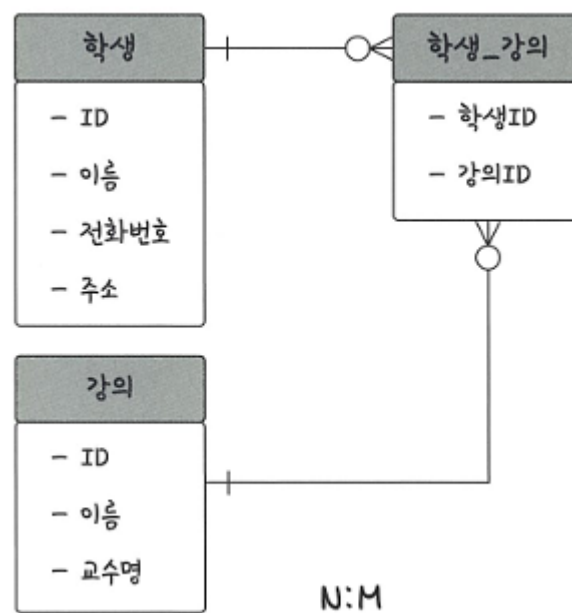
1:N 관계

ex) 쇼핑몰의 유저와 장바구니 - 유저당 장바구니에 상품을 여러가지를 넣을 수 있다.



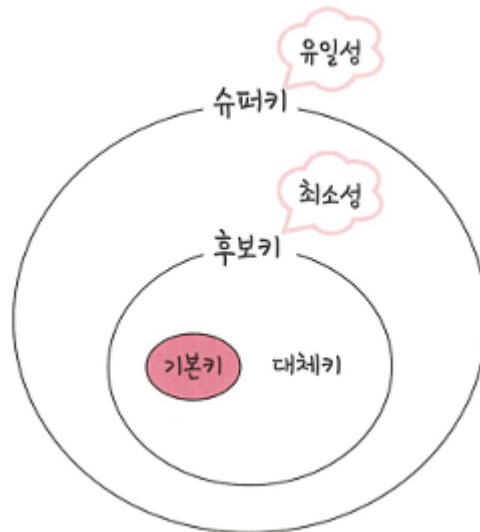
N:M 관계

ex) 학생과 강의의 관계 학생도 강의를 여러개 들을 수 있고 강의도 여러명의 학생에게 보여 줄 수 있다.



N:M은 테이블 두 개를 직접적으로 연결해서 구축하지는 않고 1:N, 1:M이라는 관계를 갖는 테이블 두 개로 나눠서 설정합니다.

4.1.7 키



기본키(Primary Key) - 줄여서 PK 또는 프라이머리키라고 많이 부르며, 유일성과 최소성을 만족하는 키

ID	name
PDT-0001	홍철이의 따스한 점퍼
PDT-0002	제호의 BMW
PDT-0002	제호의 BMW
PDT-0003	종선이의 벤츠

기본키에 해당하는 데이터는 위의 그림의 ID처럼 중복되어서는 안됩니다.

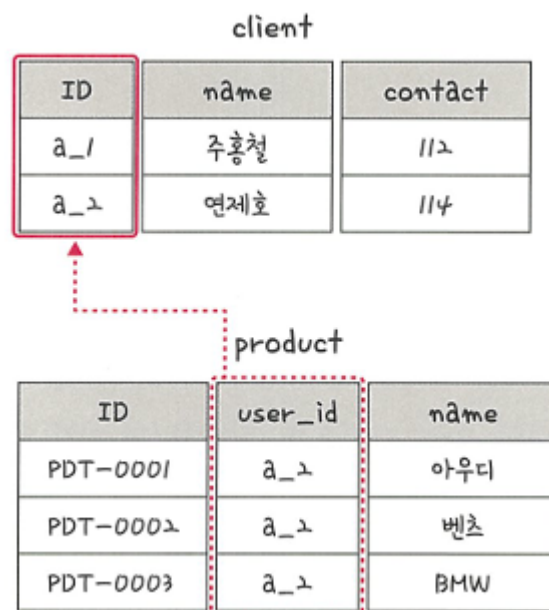
자연키 - 중복된 값들을 제외하며 중복되지 않는 것을 '자연스레' 뽑다가 나오는 키를 자연키라고 한다. 자연키는 언젠가는 변하는 속성을 가진다.

인조키 - 유저 테이블을 만든다고 할 때 회원 테이블을 생성한다고 가정하면 주민등록번호, 이름, 성별 등의 속성이 있다. 여기에 인위적으로 유저 아이디를 부여한다. 이를 통해 고유 식별자가 생긴다

이렇게 인위적으로 생성한 키를 인조키라고 한다. 자연키와는 대조적으로 변하지 않는다. 보통 기본키는 인조키로 설정한다.

외래키

외래키(Foreign Key)는 FK라고도 하며, 다른 테이블의 기본키를 그대로 참조하는 값으로 개체와의 관계를 식별하는 데 사용합니다.



외래키는 중복되어도 괜찮다.

후보키 - 기본키가 될 수 있는 후보들이며 유일성과 최소성을 동시에 만족하는 키이다.

대체키 - 후보키가 두 개 이상일 경우 어느 하나를 기본키로 지정하고 남은 후보키들을 말한다.

슈퍼키 - 각 레코드를 유일하게 식별할 수 있는 유일성을 갖춘 키