



# Chap 04 데이터베이스\_4.1 데이터베이스의 기본

## 4.1 데이터베이스의 기본

### ✓ 개념 정리

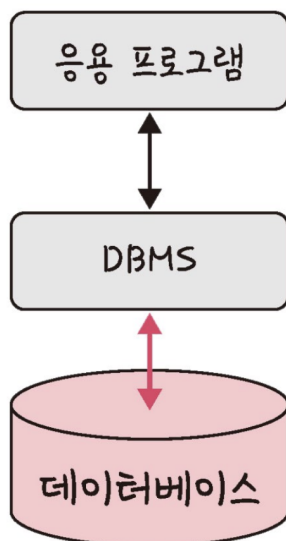
#### •데이터베이스란?

일정한 규칙, 혹은 규약을 통해 구조화되어 저장되는 데이터의 모음

#### •DBMS란?

데이터베이스를 제어, 관리하는 통합 시스템

DB안에 있는 데이터들은 특정 DBMS마다 정의된 쿼리언어를 통해 삽입, 수정, 조회 등을 수행할 수 있다.



▲ 그림 4-1 데이터베이스와 DBMS

ex)

mysql(DBMS)이 있고 그 위에 응용 프로그램에 속하는 Node.js나 php에 해당 데이터베이스 안에 있는 데이터를 끄집어내 해당 데이터 관련 로직 구축 가능

### 4.1.1 Entity

## 엔티티란?

사람, 장소, 물건, 사건, 개념 등 여러 개의 속성을 지닌 명사를 의미

ex) 회원 엔티티 : 이름, 아이디, 주소, 전화번호의 속성을 가짐

## 약한 엔티티와 강한 엔티티

A가 혼자서는 존재하지 못하고 B의 존재 여부에 따라 종속적이다.

→ A = 약한 엔티티, B = 강한 엔티티

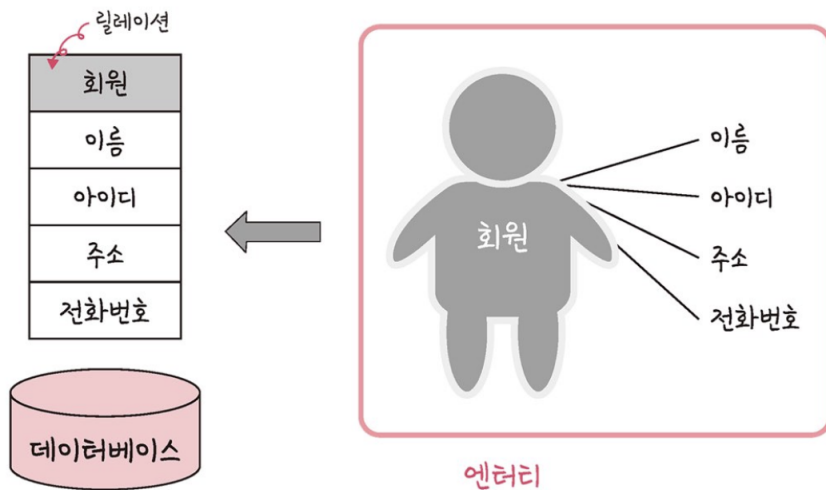
ex) 방과 건물 : 방 = 약한 엔티티, 건물 = 강한 엔티티

## 4.1.2 Relation

### 릴레이션이란?

DB에서 정보를 구분하여 저장하는 기본 단위

→ 엔티티에 관한 데이터를 DB는 릴레이션 하나에 담아서 관리



▲ 그림 4-3 릴레이션

-회원이라는 entity가 DB에서 관리될때 릴레이션으로 변화

-릴레이션은 관계형 DB에서는 '테이블'이라고 하며 NoSQL DB에서는 '컬렉션'이라고 함

## 테이블과 컬렉션

DB종류 = 관계형 DB(MySQL) / NoSQL DB (MongoDB)

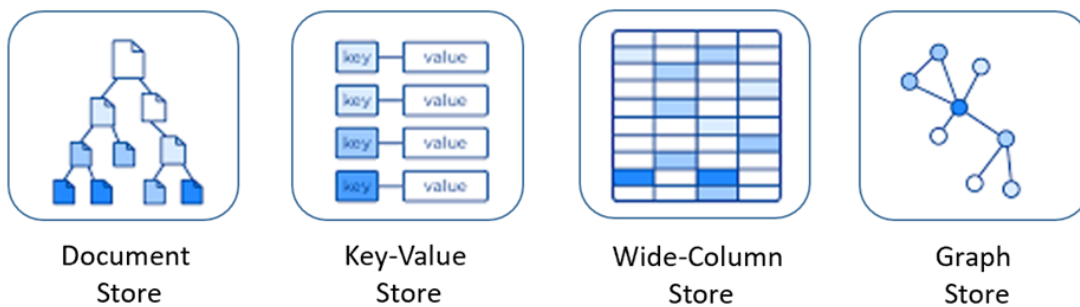
- MySQL = 레코드-테이블-데이터베이스
- MongoDB = 도큐먼트-컬렉션-데이터베이스

## •NoSQL이란?

### [특징]

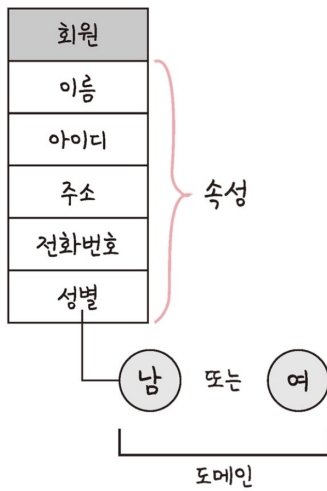
- 비관계형 데이터베이스
- 대량의 분산된 비정형 데이터를 저장하고 조회하는데 특화
- 관계를 정의하지 않음 (key-value 형태)
- 분산형 구조로 설계
- 저렴한 비용으로 분산처리와 병렬처리 가능
- 비정형 데이터 구조 설계로 설계 비용 감소
- 데이터 업데이트 중 장애 발생시 데이터 손실 발생 가능
- 많은 인덱스를 사용하려면 충분한 메모리가 필요. 인덱스 구조가 메모리에 저장
- 데이터 일관성이 항상 보장되지 않음

### [종류]



- key-value DB (Redis, oracle nosql)
- wide-column DB (hbase, googlebig talbe, cassandra)
- Document DB (mongoDB,)
- Graph DB ()

## 속성, 도메인



▲ 그림 4-5 속성과 도메인

## 필드, 레코드

member

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

### ■ 필드 타입

[숫자 타입] TINYINT, SMALLINT, MEDIUMINT, INT, BIGINT 등

▼ 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, DATETIME, TIMESTAMP 등

타입	날짜	시간	지원 범위	용량
DATE	o	x	1000-01-01 9999-12-31	3바이트
DATETIME	o	o	1000-01-01 00:00:00 9999-12-31 23:59:59	8바이트
TIMESTAMP	o	o	1000-01-01 00:00:01 1970~2038-01-19 03:14:07	4바이트

[문자타입] CHAR, VARCHAR, TEXT, BLOB, ENUM, SET

문자열

타입	특징	특징	길이
CHAR	고정 길이 문자열	선언한 길이 값으로 '고정'하여 저장됨	0-255
VARCHAR	가변 길이 문자열	입력된 데이터에 따라 용량 가변시켜 저장	0~65,535

큰 데이터 저장

타입	특징	예시
TEXT	큰 문자열 저장	게시판 본문 저장
BLOB	이미지, 동영상	보통 아마존 S3(이미지 호스팅 서비스) 서버에 파일 올린 후 파일에 관련 경로를 VARCHAR로 저장

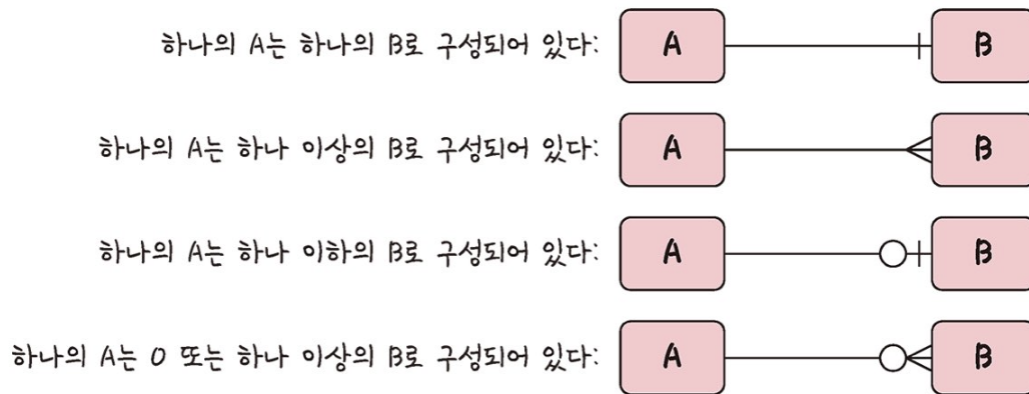
문자열 열거

타입	형태	최대 요소	특징
ENUM	'x-small', 'small', 'medium', 'large', 'x-large'	65,535	1개의 형태만 선택 가능 (단일 선택)
SET	'x-small', 'small', 'medium', 'large', 'x-large'	64	형태 중 다중 형태 가능

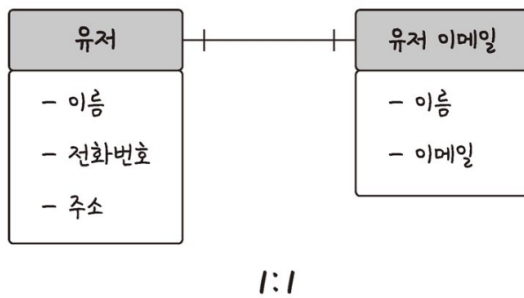
→ ENUM을 이용하면 x-small 등이 0,1 등으로 매핑되어 메모리를 적게 사용함

→ SET은 비트 단위의 연산 가능

## 4.1.6 관계

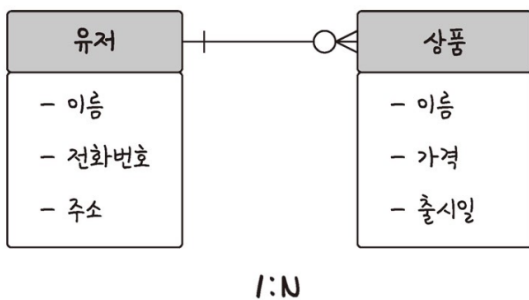


### 1:1 관계



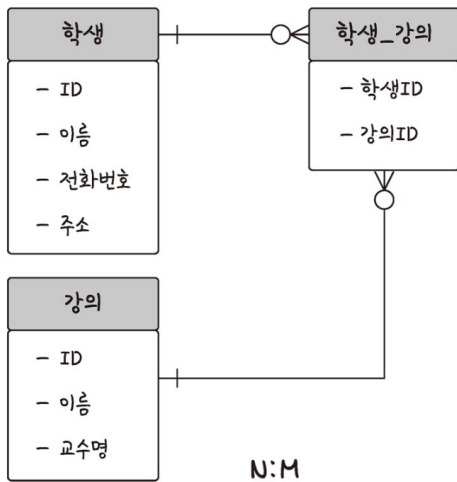
▲ 그림 4-10 1:1 관계

### 1:n 관계



▲ 그림 4-11 1:N 관계

### m:n 관계

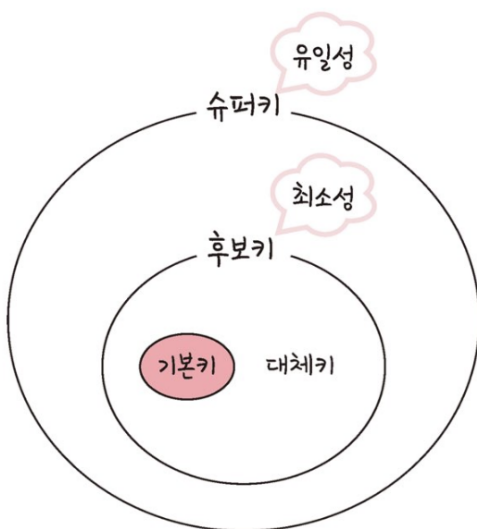


▲ 그림 4-12 N:M 관계

n:m은 테이블 두 갯의 직접적으로 연결하지 않고 1:n, 1:m 관계를 갖는 테이블 2개로 나눔

## 4.1.6 키

테이블 간의 관계를 명확하게 하고 테이블 자체의 인덱스를 위해 설정된 장치로 기본키, 외래키, 후보키, 슈퍼키, 대체키가 있음



▲ 그림 4-13 키 간의 관계

### ✓ 기본키(Primary key; PK)

유일성 o, 최소성 o

key	ex	변경 유무
자연키	주민등록번호	변화o

key	ex	변경 유무
인조키	유저 아이디	변화x

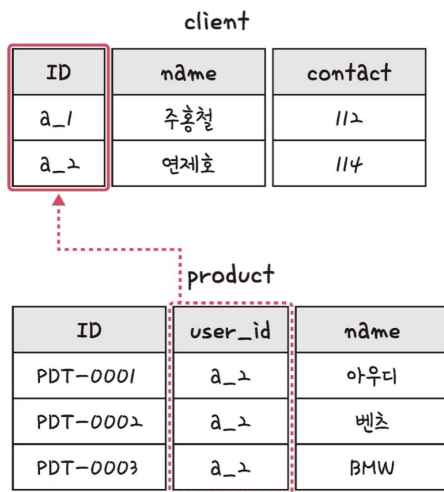
\*보통 기본키는 인조키로 설정

## ✅ 외래키 (Foreign Key; FK)

다른 테이블의 기본키를 그대로 참조하는 값

→ 개체와의 관계를 식별하는데 사용

→ 중복 가능



▲ 그림 4-16 외래키

## ✅ 후보키(candidate key)

- 기본키가 될수 있는 후보
- 유일성o, 최소성o

## ✅ 대체키(alternate key)

- 후보키 2개일 때 기본키 하고 남은거

## ✅ 슈퍼키(Super key)

- 각 레코드를 유일하게 식별할 수 있는 유일성을 갖춘 키
- 유일성o, 최소성x