

RDB 기초

DB 설계

DB 란?

구조화된 **데이터의 집합**

-> 데이터를 **효율적**으로 저장, 검색, 수정 및 **관리**할 수 있음

DBMS 란?

데이터베이스를 효율적으로 조작하기 위한
별도의 소프트웨어

-> DBMS(**D**ata**B**ase **M**anagement **S**ystem)



DB 종류

1세대
네트워크 DB,
계층 DB
IDS, IMS

2세대
RDB
MySql,
MariaDB

3세대
ODB
O2,
ONTOS

4세대
NOSQL
MongoDB,
Cassandra

RDB 란?

R elational D ata B ase

= **관계형** 데이터베이스

RDB 특징

- * 데이터들을 **2차원 테이블**로 표현(구성)

- * 테이블 간의 **관계**를 통해 데이터 관리

-> **2차원 테이블들이 관계를 맺은 집합체**

2차원 테이블

- * 열: 각 객체가 가지는 속성 한 개(= **column**)
- * 행: 하나의 데이터에 대한 모든 속성들의 묶음
(하나의 객체 = **tuple**)

유저

소속	이름		id
떡잎마을	짱구		Jjang_gu0505
떡잎마을	맹구		Mang_gu0910

-> **테이블(Entity)**

Key

테이블 내에서 특정 행을 유일하게 식별하거나,
테이블 간의 관계를 정의하는 데 사용되는 열

하나의 객체
(tuple)

속성
(column)

Key

하나의 객체를 **식별**하기 위한 **속성**
OR
테이블 간의 **관계를 정의**하는데 사용되는 **속성**

Key

식별하기 위한 속성 = 기본 키(Primary Key)

관계를 정의하는 속성 = 외래 키(Foreign Key)

기본 키(Primary Key)

: 하나의 객체를 **식별**하기 위한 **속성(column;열)**

-> **유일**해야함. **NULL**을 가질 수 없음

null [nvl]

형용사

1. (법률상) 무효의(useless); 무익한; 중요하지 않은, 무의미의
2. 빈 ((정보의 부재))



유저

유저 id	이름	소속
Jjang_gu0505	짱구	떡잎마을
Mang_gu0910	맹구	떡잎마을

외래 키(ForeignKey)

: 테이블 간의 **관계를 정의**하는 데 사용되는 **속성(column; 열)**

-> 다른 테이블의 기본 키를 **참조**

유저 id	이름	소속
Jjang_gu0505	짱구	떡잎마을
Mang_gu0910	맹구	떡잎마을

PK

참조된 테이블의
pk만 가짐

게시글

유저 id	게시글 번호	내용
Jjang_gu0505	1	난 짱구야
Mang_gu0910	2	난 짱구야
Mang_gu0910	3	사실 난 맹구야

FK

PK

외래 키(ForeignKey)

예제) 게시물, 유저, 댓글이라는 테이블이 존재할 때 사용자가 댓글을 단 게시물들을 조회하는 방법?

게시글 **FK**: 유저 id

댓글 **FK**: 유저 id, 게시물 번호

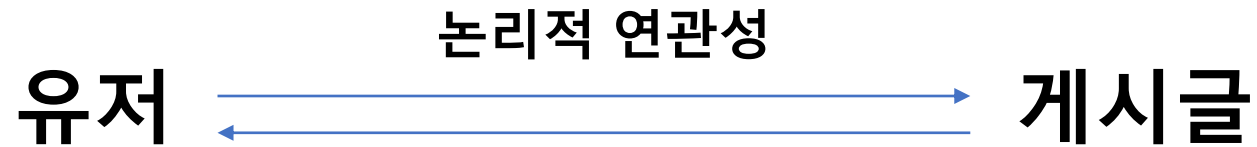
외래 키(ForeignKey)

예제) 게시물, 유저, 댓글이라는 테이블이 존재할 때 사용자가 댓글을 단 게시물들을 조회하는 방법?

1. 댓글 테이블에서 해당 사용자를 FK로 가지고 있는 행 조회
2. 해당 행들에 있는 게시물 번호를 중복을 제거하고 게시물 테이블에서 조회

관계

-> 데이터 테이블(**Entity**)들 간의 **연결**



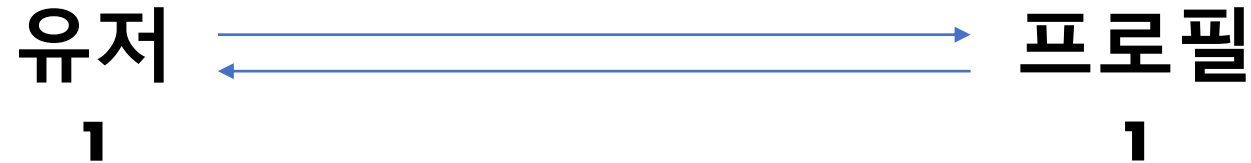
1:1관계

1:N관계

M:N관계

1:1관계

: 두 엔티티간의 관계에서 한 행이 정확히 한 행이랑 연결되는 관계



PK

유저 id	이름	소속
Jjang_gu0505	짱구	떡잎마을
Mang_gu0910	맹구	떡잎마을

FK

유저 id	닉네임	상태메세지
Jjang_gu0505	초코비 수집가	초코비 삽니다
Mang_gu0910	돌 수집가	돌 삽니다

Unique 조건

1:N관계

: 두 엔티티 간의 관계에서 한 행이 여러 행이랑 연결 되는 관계



PK

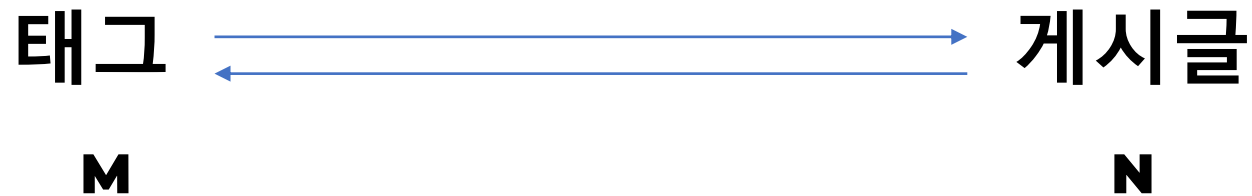
유저 id	이름	소속
Jjang_gu0505	짱구	떡잎마을
Mang_gu0910	맹구	떡잎마을

FK

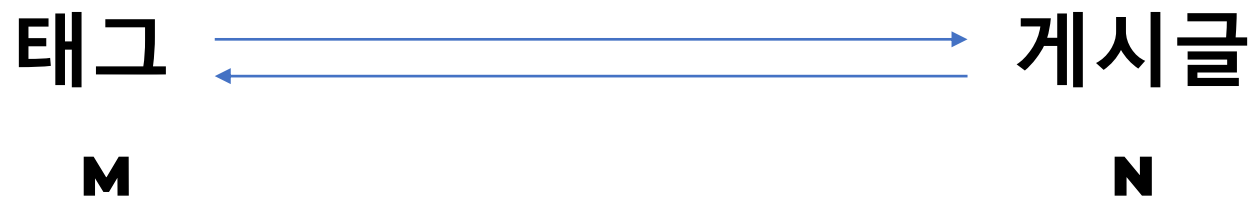
유저 id	게시글 번호	내용
Jjang_gu0505	1	난 짱구야
Mang_gu0910	2	난 짱구야
Mang_gu0910	3	사실 난 맹구야

M:N관계

: 두 엔티티 간의 관계에서 서로 다수의 행과 연결되는 관계



M:N관계

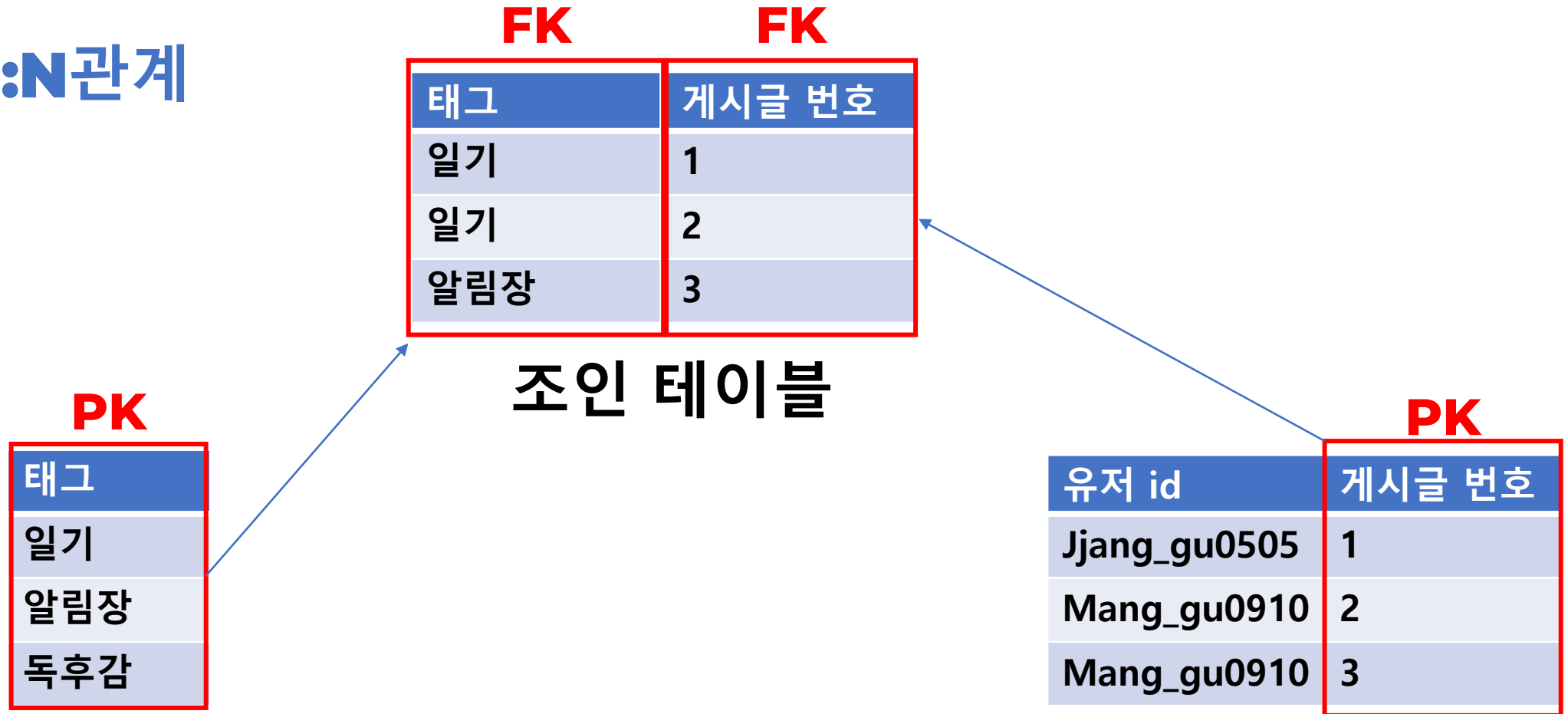


PK	FK
태그	게시글 번호
일기	1
일기 ???	2
알림장	3
독후감	

	PK	FK
유저 id	게시글 번호	태그
Jjang_gu0505	1	일기
Mang_gu0910	2	일기
Mang_gu0910	3	알림장



M:N관계



On Delete

외래 키 제약 조건의 일부로 사용되며, 참조하는 테이블의 행이 삭제될 때 참조된 테이블의 관련 행에 적용할 작업을 정의하는 것

데이터베이스의 참조 무결성을 유지하고 연관된 데이터 간의 일관성을 보장하기 위해 중요!

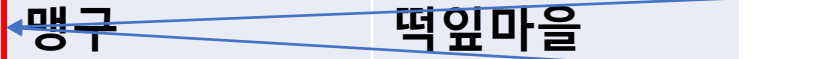
On Delete

PK

유저 id	이름	소속
Jjang_gu0505	짱구	떡잎마을
Mang_gu0910	맹구	떡잎마을

FK

유저 id	게시글 번호	내용
Jjang_gu0505	1	난 짱구야
Mang_gu0910	2	난 짱구야
Mang_gu0910	3	사실 난 맹구야





1. No Action

참조된 테이블의 행(**tuple**)이 삭제될 때
아무런 작업을 수행하지 않는 것

만약 삭제로 인해 데이터의 무결성이 깨진다면 삭제하지 않음

PK

유저 id	이름	소속
Jjang_gu0505	짱구	떡잎마을
Mang_gu0910	맹구	떡잎마을

FK

유저 id	게시글 번호	내용
Jjang_gu0505	1	난 짱구야
Mang_gu0910	2	난 짱구야
Mang_gu0910	3	사실 난 맹구야

변화 X

2. Cascade

참조된 테이블의 행(**tuple**)이 삭제될 때
해당 행을 참조하는 모든 행도 함께 삭제
연관된 모든 데이터들을 자동으로 정리하고 싶을 때 유용

PK

유저 id	이름	소속
Jjang_gu0505	짱구	떡잎마을
Mang_gu0910	맹구	떡잎마을

FK

유저 id	게시글 번호	내용
Jjang_gu0505	1	난 짱구야
Mang_gu0910	2	난 짱구야
Mang_gu0910	3	사실 난 맹구야



3. Set Null

참조된 테이블의 행(**tuple**)이 삭제될 때
외래 키 속성(column)을 Null로 설정
외래 키 속성이 **Nullable**이어야 함

PK

유저 id	이름	소속
Jjang_gu0505	짱구	떡잎마을
Mang_gu0910	맹구	떡잎마을

FK

유저 id	게시글 번호	내용
Jjang_gu0505	1	난 짱구야
NULL	2	난 짱구야
NULL	3	사실 난 맹구야



SKU × LIKELION

Official Team
Official LIKELION at SKU

RDB 기초

테이블
Column
tuple

Key
관계

On Delete