

데이터 베이스



MySQL™

목차

1.데이터베이스

1. 데이터란?
2. 데이터베이스란?
3. 데이터베이스 시스템이란?
4. 관계형데이터베이스란?

2.데이터 모델링

1. 데이터 모델링이란?
2. 개념적 데이터 모델링
3. 논리적 데이터 모델링

3.MySQL

1. 설치방법
2. 스키마, 테이블 생성
3. CRUD
4. JOIN
5. 기타 개념 (view, 트랜잭션 등)





데이터베이스란?

단순히 관찰, 측정하여
수집한 사실

자료, 데이터

월간 아이스크림 판매량

데이터를 유용한 형태로
처리한 결과

**정보,
인포메이션**

아이스크림은 여름에 더
잘 팔리네!



데이터베이스란?

데이터베이스

“ 여러 사람들이 공유하고 사용할 목적으로 통합 관리되는 데이터들의 모임 ”

데이터베이스의 목적

“ 데이터를 수집하고 저장한 후 필요할 때 제공하여 정보를 만드는 것 ”



데이터베이스란?

스키마(schema)

: 데이터베이스에 저장되는 데이터 구조와 제약조건(중복 값, not null 등)을 정의한 것

고객 번호 INT	이름 CHAR(10)	나이 INT	주소 CHAR(20)
--------------	----------------	-----------	----------------

인스턴스(instance)

: 스키마에 따라 데이터베이스에 실제로 저장된 값



데이터베이스란?

데이터 언어

- **데이터 정의어**(DDL, Data Definition Language) : **스키마**를 정의하거나 변경, 삭제하기 위해 사용.
- **데이터 조작어**(DML, Data Manipulation Language) : **실제 데이터**의 삽입, 삭제, 수정, 검색 등의 처리를 요구하기 위해 사용
- **데이터 제어어**(DCL, Data Control Language) : 내부적으로 필요한 **규칙**이나 **기법** 정의하기 위해 사용. 데이터의 무결성, 보안, 회복 등을 위해 사용. DCL을 이용해 정한 규칙과 기법에 따라 DBMS가 DB를 제어하고 보호.

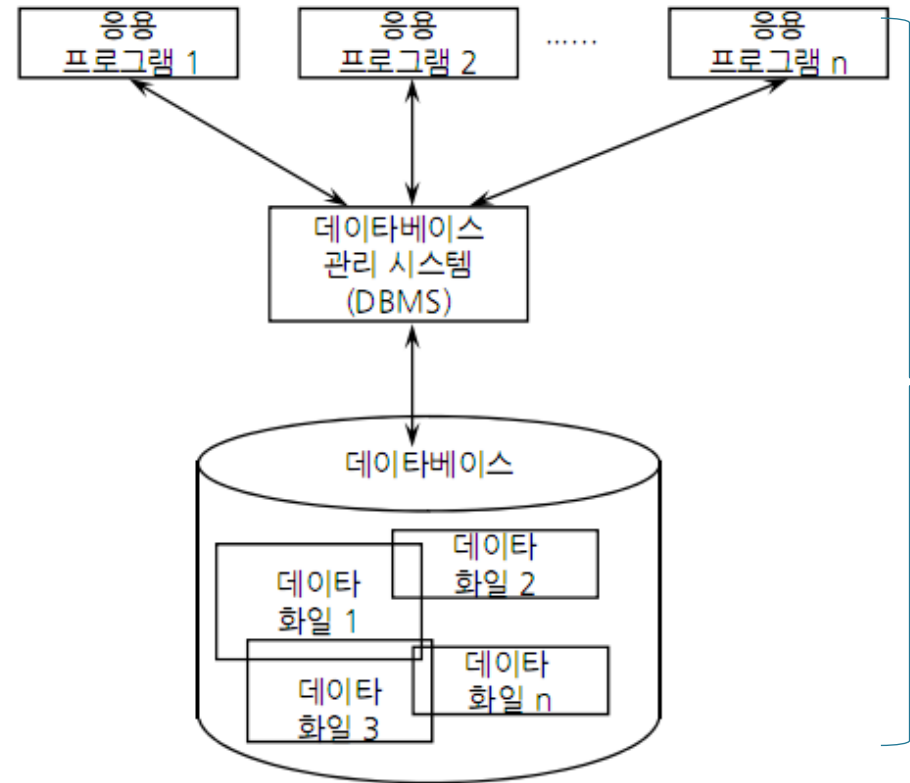


데이터베이스란?

데이터베이스 관리 시스템(DataBaseManagementSystem)

응용프로그램을 대신하여 데이터베이스에 존재하는 데이터를 저장하고 관리.

모든 응용 프로그램이 데이터 베이스를 공유할 수 있게 함.



데이터
베이스
시스템



데이터베이스란?

데이터베이스 관리 시스템(DataBaseManagementSystem)

DBMS의 기능

- “ 적합한 DB 구조를 정의, 이미 정의된 구조를 수정 ”
- “ 데이터의 삽입, 삭제, 수정, 검색연산 ”
- “ 데이터를 항상 정확하고 안전하게 유지 ”



데이터베이스란?

관계형 데이터베이스(Relational DataBase)

“데이터를 테이블 형태로 저장하는 데이터베이스.”

- 테이블(Table) : 개체(Entity), 릴레이션(Relation)이라고도 하며 행과 열로 구성.
- 행(row) : 레코드(record), 튜플(tuple)이라고도 하며 각 데이터항목을 저장. 튜플의 수는 Cardinality라고 함.
- 열(column) : 필드(field), 데이터 항목(Data item), 속성(Attribute)이라고도 하며 항목의 명칭을 나타냄. 개체를 구성하는 논리적 단위. 속성의 수는 Degree 또는 차수라고 함.



데이터베이스란?

관계형 데이터베이스(Relational DataBase)

< 주문 테이블 >

주문 번호	고객 이름	고객 지역	주문 상품
1	노아	부산	Hp 노트북
2	두루미	서울	맥북
3	노아	부산	LG 그램
4	노아	부산	삼성노트

불필요한 중복 발생



데이터베이스란?

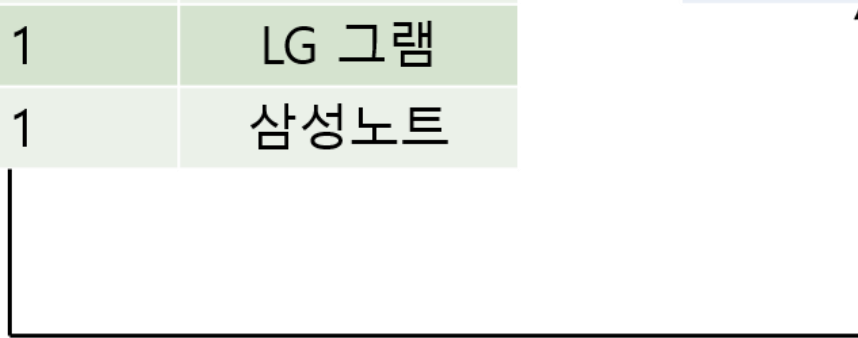
관계형 데이터베이스(Relational DataBase)

< 주문 테이블 >

주문 번호	고객 번호	주문 상품
1	11	Hp 노트북
2	22	맥북
3	11	LG 그램
4	11	삼성노트

< 고객 테이블 >

고객 번호	고객 이름	고객 지역
11	노아	부산
22	두루미	서울





데이터베이스란?

관계형 데이터베이스(Relational DataBase)

- Key : 릴레이션 내에서 튜플을 구별할 수 있는 속성
- PK(Primary Key), 주식별자 : 모든 데이터를 유일하게 식별해주는 속성. 중복되지 않으면서 NULL일 수 없는 컬럼이 선택됨. Ex) 주민번호
- FK(Foreign Key), 외래키, 외부 식별자 : 참조하는 테이블과 참조되는 테이블의 관계를 나타내는 속성.



데이터베이스란?

관계형 데이터베이스(Relational DataBase)

< 주문 테이블 >

주문 번호	고객 번호	주문 상품
1	11	Hp 노트북
2	22	맥북
3	11	LG 그램
4	11	삼성노트

PK

< 고객 테이블 >

고객 번호	고객 이름	고객 지역
11	노아	부산
22	두루미	서울

FK, 주문테이블은 고객 테이블을 참조하는 테이블이라는 관계 표현.



데이터베이스란?

관계형 데이터베이스 관리 시스템 (Relational DataBase Management System)

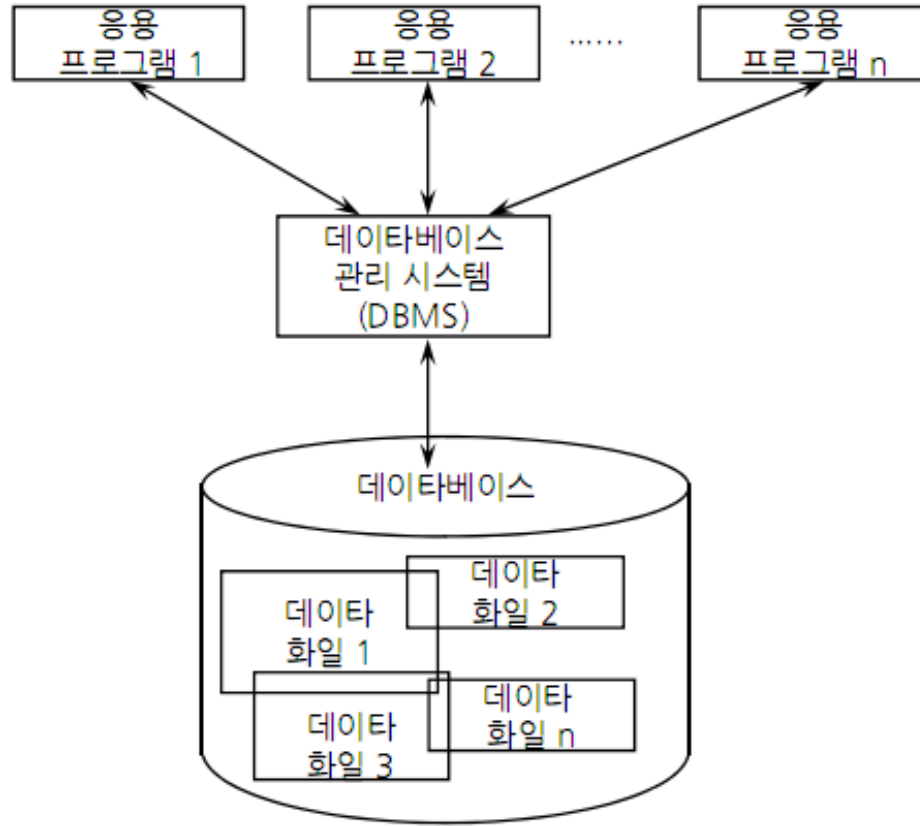
현재 가장 널리 쓰이는 관계형 데이터베이스를 생성, 수정, 관리하는 프로그램.

Ex) 오라클(Oracle), MS SQL 서버, MySQL, PostgreSQL, Maria DB 등

SQL(Structured Query Language) : 관계형 데이터베이스에서 사용하는 구조화 질의어. RDBMS에서 사용하는 프로그래밍 언어. 데이터 정의 언어, 데이터 조작 언어, 데이터 제어 언어로 나뉜다.



데이터베이스란?



현실 세계에 존재하는 데이터를 어떻게 컴퓨터에 옮길까?

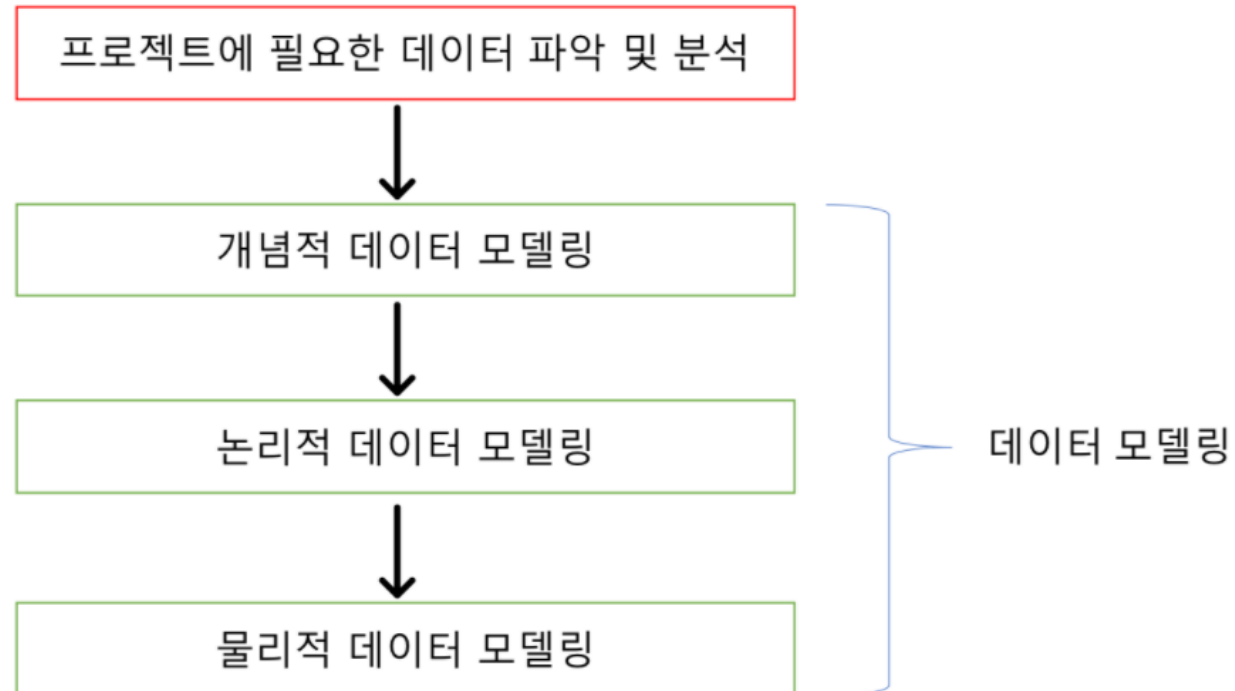
- > 데이터 모델링



데이터 모델링이란?

데이터 모델링(Data Modeling)

: 현실 세계에 존재하는 데이터를 컴퓨터의 데이터베이스로 옮기는 변환 과정





데이터 모델링이란?

개념적 데이터 모델링(Conceptual Data Modeling)

: 데이터들의 구조도를 그리는 과정. 어떤 데이터가 중요하고, 유지되어야 하는지를 결정하는 과정을 포함. **핵심 엔터티**와 그들 간의 관계를 발견하고, **개체-관계 다이어그램 (ERD, Entity-Relation Diagram)**을 생성.

- **엔터티** : 실체, 객체. 업무에 필요하고 유용한 정보를 저장, 관리하기 위한 집합적인 것(Thing). 업무 활동상 지속적인 관심을 가지고 있어야 하는 대상. 인스턴스의 집합. 반드시 속성을 지녀야 하고, 다른 엔터티와 한 개 이상의 관계가 있어야 함.

Ex) 과목이라는 엔터티는 과목명, 담당선생님, 시험일 등을 속성으로 가지며 국어, 수학 등의 인스턴스를 가지고 담당선생님이라는 속성으로 선생님 엔터티와 관계를 가짐.



데이터 모델링이란?

ERD(Entity-Relation Diagram)



개체



속성



관계



데이터 모델링이란?

ERD(Entity-Relation Diagram)

키 속성 : pk



복합 속성 : 의미 분해 가능



다중 값 속성 : 값 여러 개



유도 속성 : 기존의 다른 속성 값에서 유도

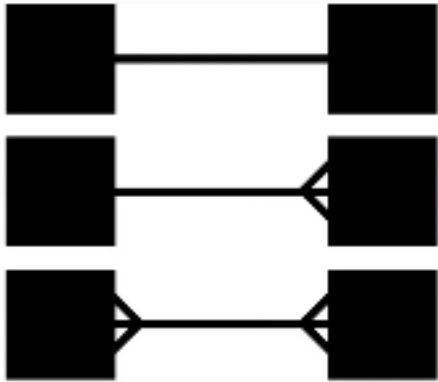




데이터 모델링이란?

ERD(Entity-Relation Diagram)

Cardinality



1:1 (반&담임쌤)

1: N (과목&담당쌤)

N : M (과목&학생)

Optionality

필수와 선택



선생님

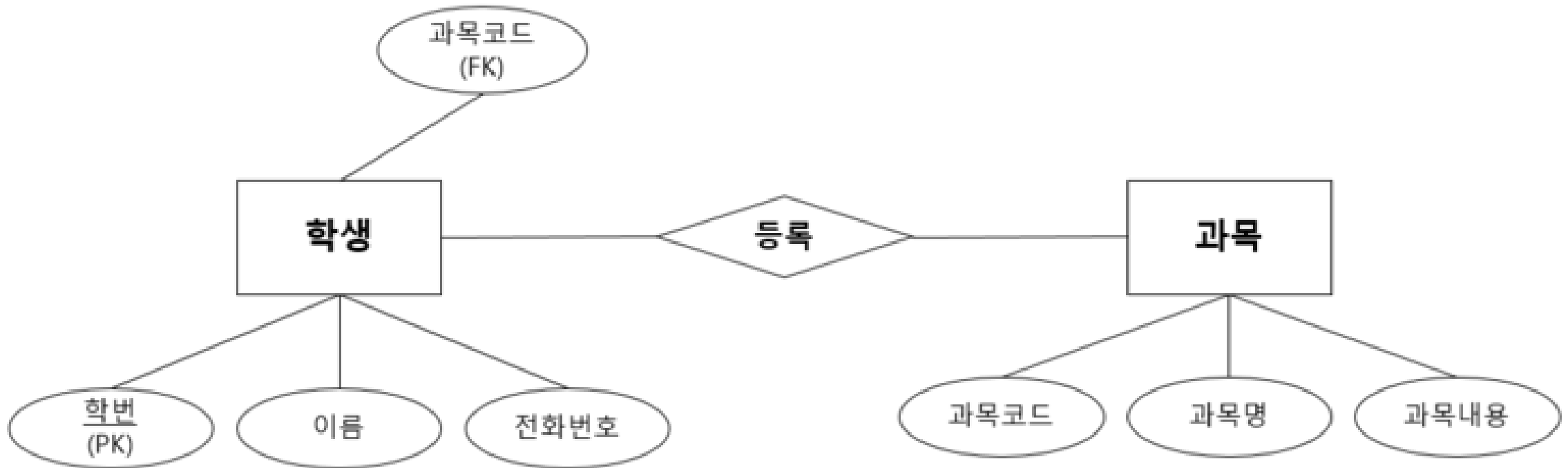
반





데이터 모델링이란?

ERD(Entity-Relation Diagram)





데이터 모델링이란?

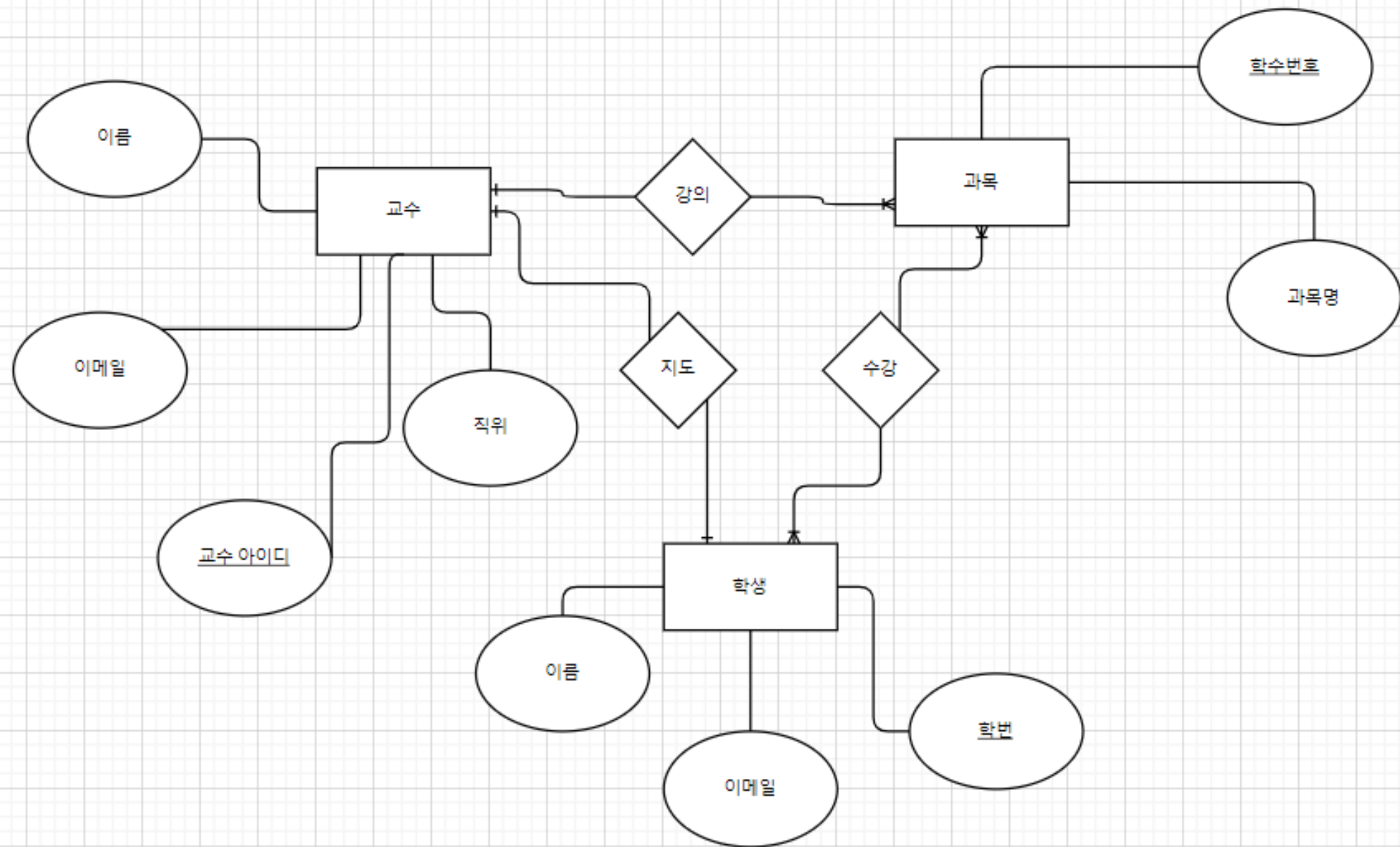
ERD(Entity-Relation Diagram) 실습

draw.io

학사 정보 관리 시스템을 위한 DB를 모델링해보자!

erd.yah.ac

1. 한 교수는 한 과목 이상을 강의해야 하며 여러 과목을 강의할 수 있고, 한 과목당 담당교수는 한 명이다.
2. 한 학생은 반드시 한 과목 이상을 수강해야 하며 여러 과목을 수강할 수 있고, 한 과목은 여러 명의 학생이 수강할 수 있다.
3. 교수는 이름, 이메일, 직위를 가지고 학생은 이름, 이메일, 학번을 가지며 과목은 과목명과 학수번호를 가진다.
4. 한 교수는 반드시 한 명의 학생을 지도하고 한 학생은 반드시 한 명의 지도교수를 가진다.

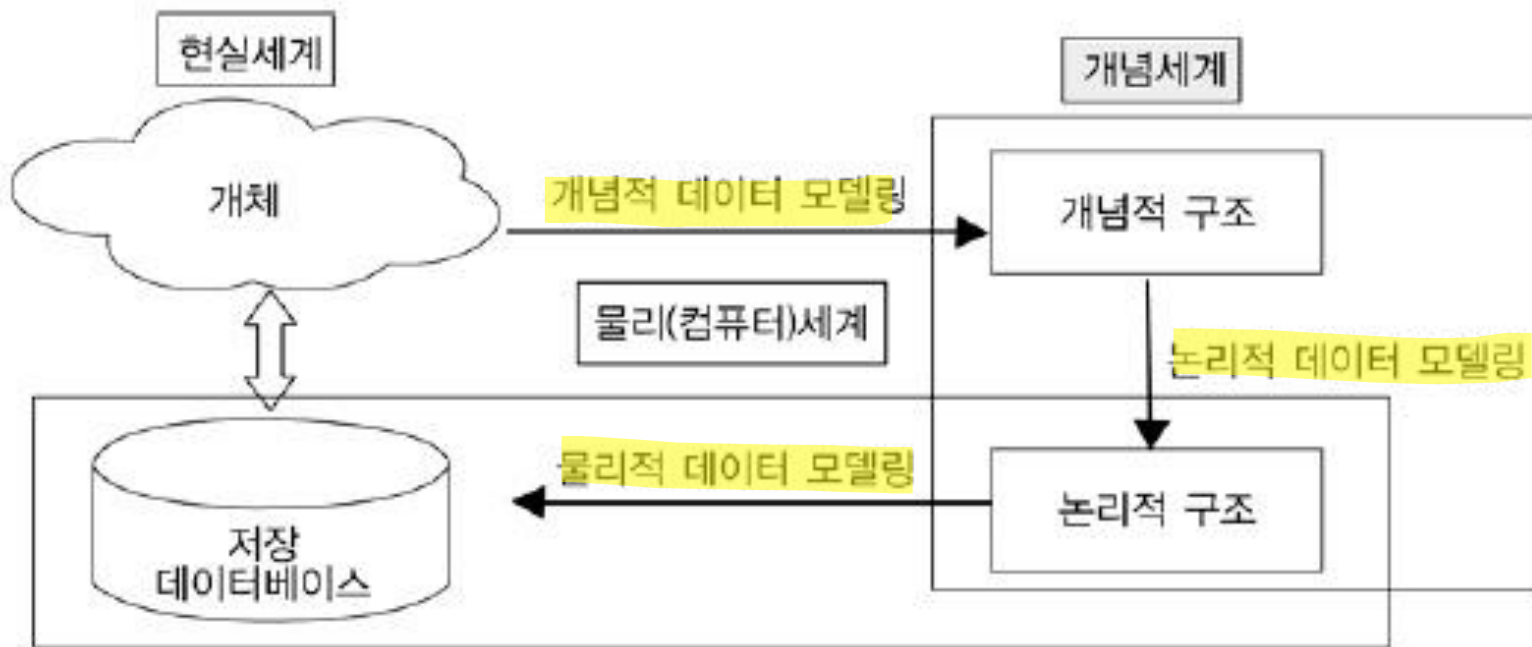




데이터 모델링이란?

논리적 데이터 모델링(Logical Data Modeling)

: 데이터 모델을 선택하고 데이터 스키마를 결정하는 과정. ERD로 표현한 개념적 데이터 모델을 관계형 데이터 베이스 체계에 어울리도록 데이터 형식을 잘 정리.





데이터 모델링이란?

논리적 데이터 모델링(Logical Data Modeling)

Entity



테이블

Attribute



컬럼

Relationship



PK, FK



데이터 모델링이란?

1:1 관계

1) 교수 아이디를 외래키로

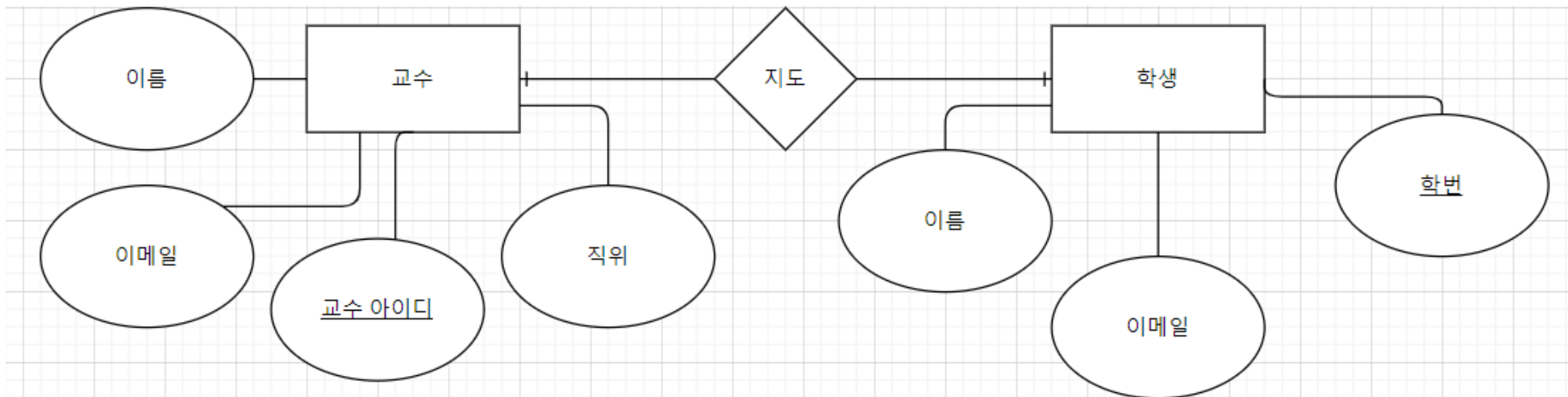
교수 : 교수 아이디, 이름, 이메일, 직위
학생 : 학번, 이름, 이메일, *교수 아이디*

2) 학번 외래키로

교수 : 교수 아이디, 이름, 이메일, 직위, *학번*
학생 : 학번, 이름, 이메일

3) 한 개의 테이블로

학생 : 학번, 이름, 이메일, 교수 아이디, 이름, 이메일, 직위





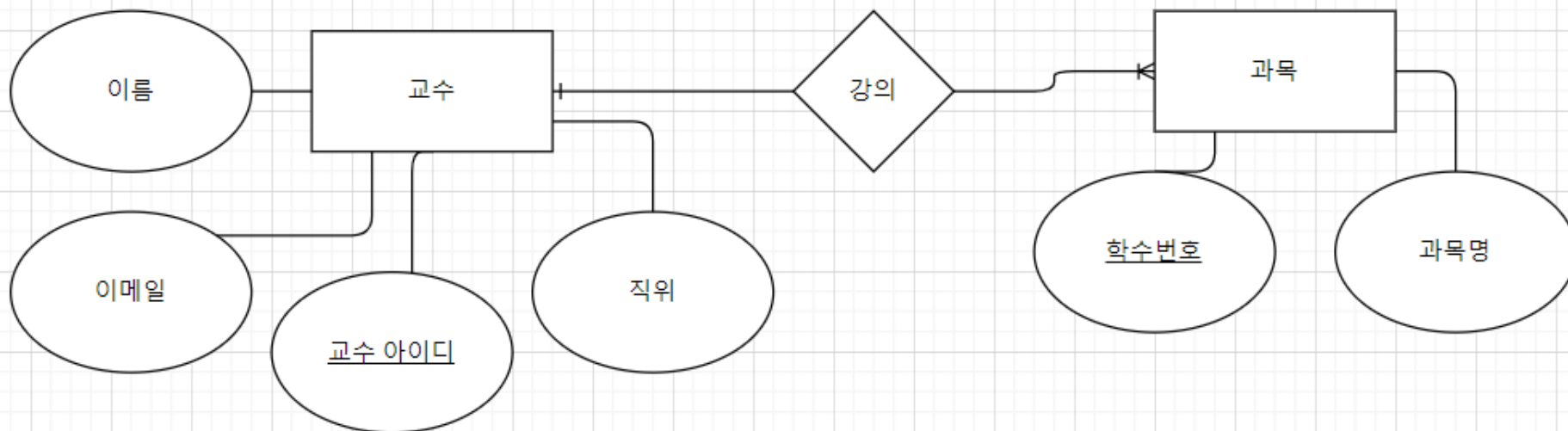
데이터 모델링이란?

1:N 관계

N(과목)측이 1(교수)의 기본키를 외래키로 포함

교수 : 교수 아이디, 이름, 이메일, 직위

과목 : 학수번호, 과목명, 교수 아이디





데이터 모델링이란?

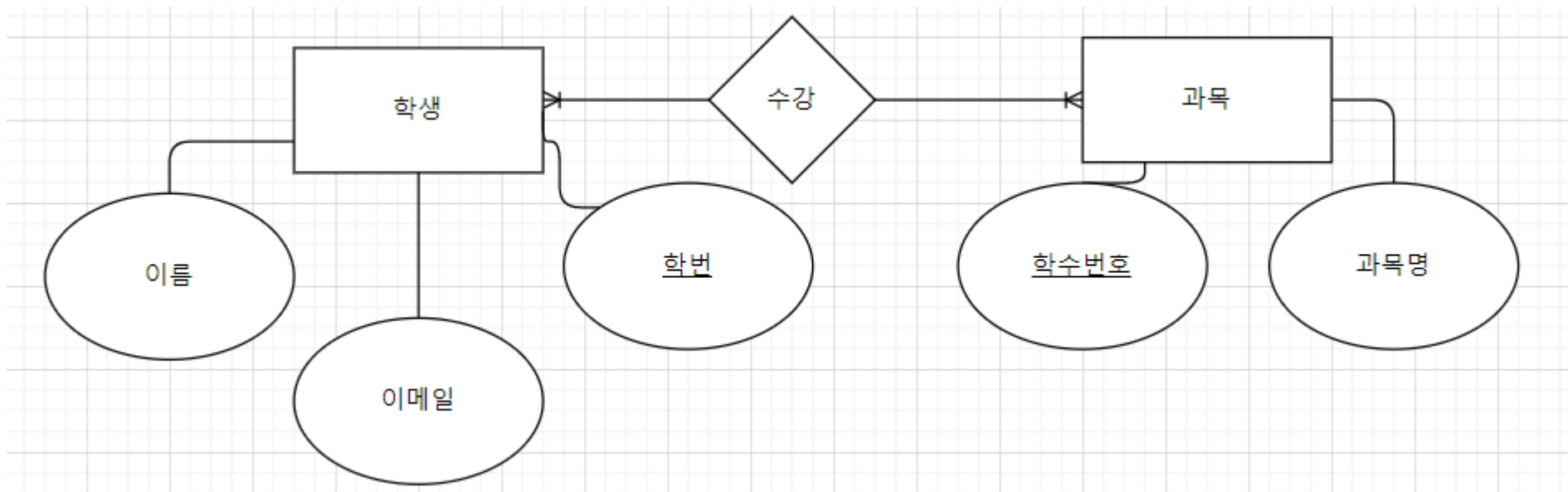
N:M 관계

관계를 하나의 릴레이션으로 생성하여 양쪽의 기본키를 외래키로 포함

학생(학번, 이름, 이메일)

과목(학수번호, 과목명)

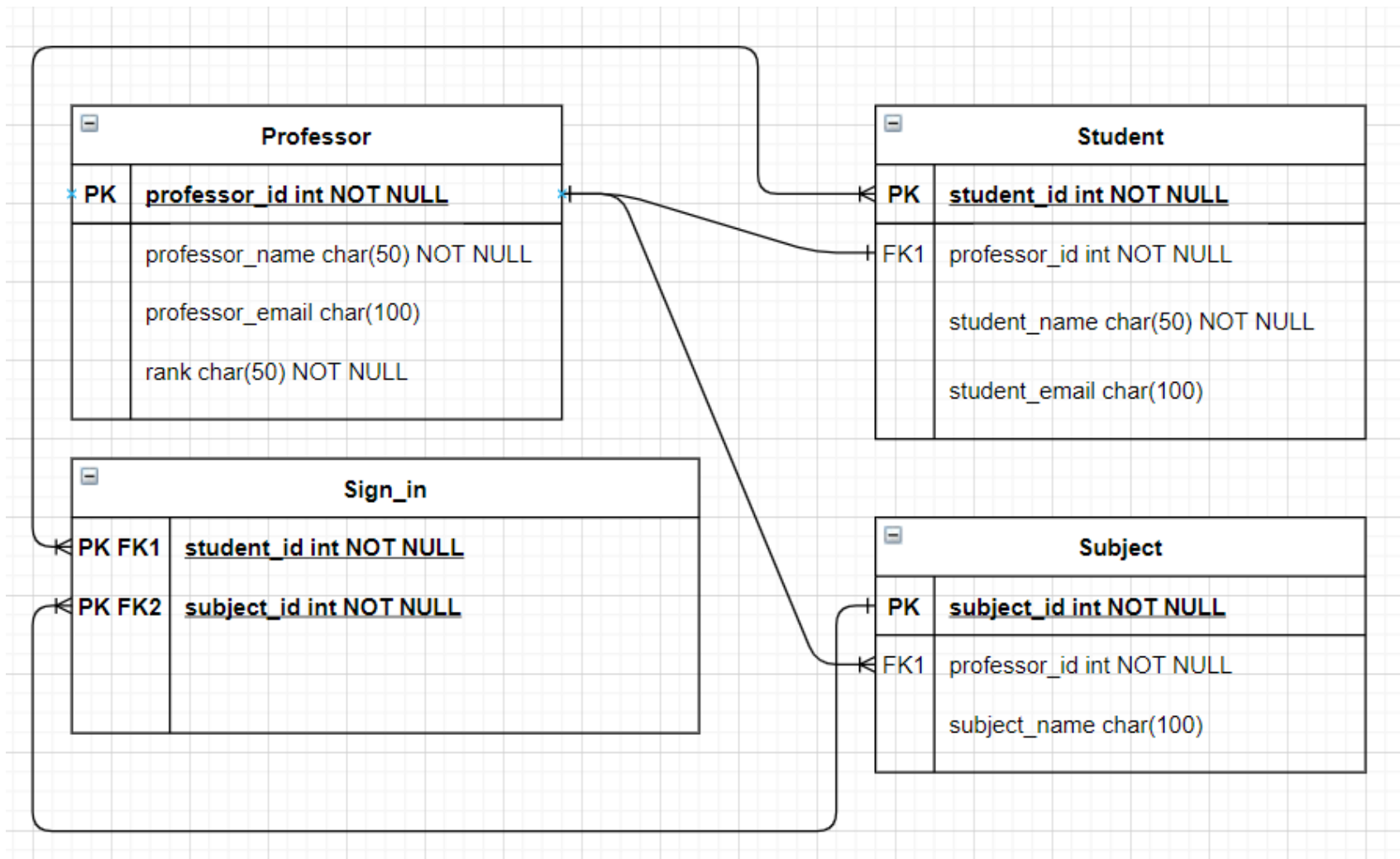
수강(학번, 학수번호)





데이터 모델링이란?

논리적 데이터 모델링(Logical Data Modeling)





데이터 모델링이란?

논리적 데이터 모델링(Logical Data Modeling)

정규화 (Normalization)

이상현상이 발생하는 릴레이션을 분해해 이상현상을 없애고, 관계형 데이터베이스에 어울리는 릴레이션으로 만들어주는 방법. 정규형이 높을수록 이상현상이 줄어든다. 제3정규형까지 산업적으로 많이 사용된다.



데이터 모델링이란?

제1 정규화

각 튜플의 속성값은 최소로 분리된 형태의 값만이 들어가야 한다. 한 컬럼에 쪼갤 수 있는 유의미한 것들의 합이 저장되어 있다면, 분리해주어야 한다.

고객취미들(이름, 취미들)		고객취미(이름, 취미)	
이름	취미들	이름	취미
김연아	인터넷	김연아	인터넷
추신수	영화, 음악	추신수	영화
추신수	음악	추신수	음악
박세리	음악, 쇼핑	박세리	음악
박세리	쇼핑	박세리	쇼핑
장미란	음악	장미란	음악
박지성	게임	박지성	게임

그림 7-17 속성 값이 원자값을 갖도록 분해

1. 왼쪽 그림처럼 여러 인스턴스로 나누기
2. 취미1, 취미2처럼 여러 컬럼으로 나누기
- 테이블 구조 바꾸어야 함, NULL값 생김.
3. 새로운 테이블 생성 (고객, 취미가 N:M 관계)



데이터 모델링이란?

제2정규화

* 복합키(composite key) :
여러 속성을 묶어서 기본키처럼 사용하는 것

기본키(Primary key) 중에 특정 컬럼에만 종속된 컬럼(부분 종속성)이 없어야 한다. 복합키를 사용한다면 살펴보아야 하고, 사용하지 않는다면 자연히 만족하게 된다.

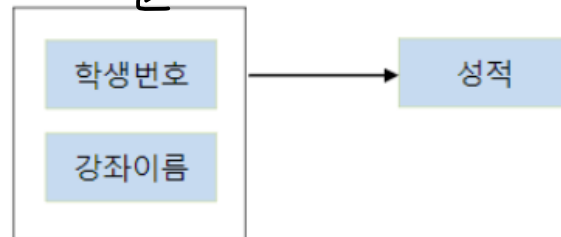
수강강좌

학생번호	강좌이름	강의실	성적
501	데이터베이스	공학관 110	3.5
401	데이터베이스	공학관 110	4.0
402	스포츠경영학	체육관 103	3.5
502	자료구조	공학관 111	4.0
501	자료구조	공학관 111	3.5



수강

학생번호	강좌이름	성적
501	데이터베이스	3.5
401	데이터베이스	4.0
402	스포츠경영학	3.5
502	자료구조	4.0
501	자료구조	3.5



강의실

강좌이름	강의실
데이터베이스	공학관 110
스포츠경영학	체육관 103
자료구조	공학관 111





데이터 모델링이란?

제3정규화

기본키가 아닌 컬럼은 기본키 이외의 다른 컬럼에게 종속성을 가져서는 안 된다.

topic TABLE

<u>title</u>	description	created	author_id	author_name	author_profile
MySQL	MySQL is ...	2011	1	kim	developer
ORACLE	ORACLE is ...	2012	1	kim	developer

topic TABLE

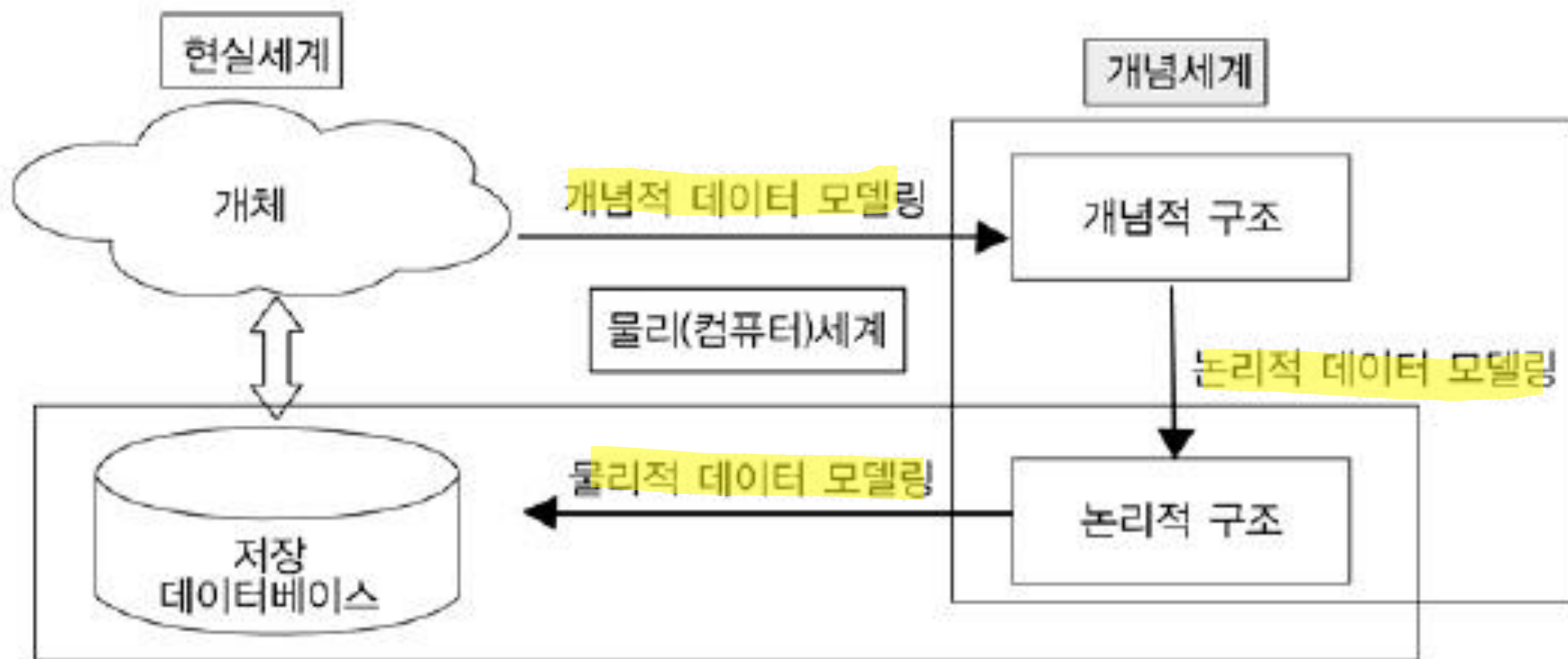
<u>title</u>	description	created	author_id
MySQL	MySQL is ...	2011	1
ORACLE	ORACLE is ...	2012	1

author TABLE

<u>id</u>	author_name	author_profile
1	kim	developer



데이터 모델링이란?





MySQL

전세계적으로 가장 널리 사용되고 있는 오픈 소스 무료 데이터베이스. 매우 빠르고 사용하기 쉽고 유연하여 웹 개발에 널리 사용되고 있다.

370 systems in ranking, April 2021

Rank			DBMS	Database Model	Score		
Apr 2021	Mar 2021	Apr 2020			Apr 2021	Mar 2021	Apr 2020
1.	1.	1.	Oracle +	Relational, Multi-model i	1274.92	-46.82	-70.51
2.	2.	2.	MySQL +	Relational, Multi-model i	1220.69	-34.14	-47.66
3.	3.	3.	Microsoft SQL Server +	Relational, Multi-model i	1007.97	-7.33	-75.46
4.	4.	4.	PostgreSQL +	Relational, Multi-model i	553.52	+4.23	+43.66
5.	5.	5.	MongoDB +	Document, Multi-model i	469.97	+7.58	+31.54



MySQL 설치

MySQL 설치 사이트

<https://dev.mysql.com/downloads/mysql/>

MySQL Workbench 설치 가이드 블로그

<https://m.blog.naver.com/bjh7007/221829548634>



MySQL

MySQL 구조

표 (table)

데이터베이스(database) = 스키마(Schema)

데이터베이스 서버(database server) : MySQL



MySQL 데이터타입

[MySQL 데이터타입 참고](#)

숫자형

TINYINT(size)

INT(size)

BIGINT(size)

FLOAT(size,d)

DOUBLE(size,d)

문자형

CHAR(size)

VARCHAR(size)

TINYTEXT(size)

TEXT(size)

LONGTEXT(size)

날짜형

DATE()

DATETIME()

TIME()

YEAR()



스키마의 사용

생성

```
CREATE DATABASE [NAME];  
CREATE SCHEMA `efub` ;
```

삭제

```
DROP DATABASE [NAME];  
DROP SCHEMA `efub` ;
```

확인

```
SHOW DATABASES;
```

사용

```
USE [NAME];  
USE 'efub';
```

이 이후에 내리는 명령은
[NAME] DB에 있는 표를 대상
으로 실행



테이블 생성

```
CREATE TABLE tbl(  
  c1 datatype(length)  
  c2 datatype(length)  
  ....  
  PRIMARY KEY(c1));
```

USE [NAME]; 으로 데이터베이스 설정 후 실행
or 스키마.테이블 (efub.member) 으로 실행

```
CREATE TABLE `member` (  
  `member_id` INT NOT NULL AUTO_INCREMENT,  
  `name` VARCHAR(45) NOT NULL,  
  `position` VARCHAR(45) NOT NULL,  
  `email` VARCHAR(100) NULL,  
  `birth_date` DATE NULL,  
  PRIMARY KEY (`member_id`));
```

NOT NULL(NN) : NULL값을 가질 수 없음

AUTO_INCREMENT(AI) : 중복없이 자동으로 증가



테이블 확인 및 수정, 삭제

`desc efub.member;` 테이블 구조 확인

`alter table member drop birth_date;`

컬럼 삭제

`alter table member add team VARCHAR(45) NOT NULL;`

컬럼 추가

`alter table member add MBTI VARCHAR(10);`

`alter table member change MBTI mbti VARCHAR(10);`

컬럼명 변경 및 타입 변경

`alter table member modify team TINYINT(10);`

컬럼 타입 수정

`alter table tbl rename tbl2;` 테이블명 수정

`drop table tbl2;` 테이블 삭제



MySQL의 CRUD(Create, Read, Update, Delete)

Create `INSERT INTO tbl(c1,c2....) VALUES(v1,v2....);`

Update `UPDATE tbl SET c1=v1 WHERE (조건);`

Delete `DELETE FROM tbl WHERE (조건)`

} 조건에 해당하는 row에
한해서 시행.

WHERE 꼭 붙이기!!

```
INSERT INTO tbl(c1,c2,c3)
VALUES(v1,v2,v3)
ON DUPLICATE KEY UPDATE c2=v2, c3=v3;
```

* c1이 기본키라고 가정.

키 값이 없으면 Insert, 있으면 Update.

Update 뒤에는 키를 제외한 컬럼들을 나열.



MySQL의 CRUD(Create, Read, Update, Delete)

Read

```
SELECT c1,c2,c3 FROM tbl ;
```

```
SELECT * FROM efub.member;
```

```
SELECT c1,c2,c3 FROM tbl WHERE (조건) LIMIT (최대 조회 레코드 수);
```

```
SELECT *FROM efub.member WHERE position='back' LIMIT 5;
```

[SELECT 참고 링크](#)

```
SELECT 컬럼 FROM 테이블 [WHERE 조건식]
```

```
GROUP BY 그룹화할 컬럼 [HAVING 조건식] ORDER BY 컬럼1 [, 컬럼2, 컬럼3 ...];
```

While : 그룹화 하기 전 조건 추가

Having : 그룹화 후에 조건 추가

Group by : 컬럼값을 기준으로 그룹화

Order by : 정렬



MySQL의 CRUD(Create, Read, Update, Delete)

Read

```
select team, count(team) as cnt from efub.member group by team having cnt >=1 order by cnt desc;
```

*팀 컬럼과 팀원 수 컬럼을 선택하여 팀별로 그룹핑한 다음 팀원 수가 1명 이상인 팀을 팀원 수 컬럼을 기준으로 내림차순으로 정렬하여 출력

	member_id	name	position	team
▶	1	박현아	back	3
	2	김정은	back	2
	3	정아연	front	4
	4	김혜린	front	1
	5	이윤지	front	3
	6	윤희정	back	2
*	NULL	NULL	NULL	NULL

	team	cnt
▶	3	2
	2	2
	4	1
	1	1



JOIN

각각 독립적인, 분리된 테이블을 읽을 때, 그 테이블들이 하나의 테이블로 저장되어 있는 것처럼 출력되도록 관계를 맺어주는 것.

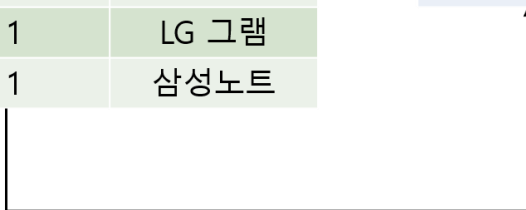
실제 데이터 저장된 테이블 모습

< 주문 테이블 >

주문 번호	고객 번호	주문 상품
1	11	Hp 노트북
2	22	맥북
3	11	LG 그램
4	11	삼성노트

< 고객 테이블 >

고객 번호	고객 이름	고객 지역
11	노아	부산
22	두루미	서울



출력 결과

< 주문 테이블 >

주문 번호	고객 이름	고객 지역	주문 상품
1	노아	부산	Hp 노트북
2	두루미	서울	맥북
3	노아	부산	LG 그램
4	노아	부산	삼성노트



테이블 저장 상태

Member table

	member_id	name	position	team
	1	박현아	back	3
	2	김정은	back	2
	3	정아연	front	4
	4	김혜린	front	1
	5	이윤지	front	3
	6	윤효정	back	2
▶	7	김이화	back	

Team table

	team_id	team_name	project_title	project_description
▶	1	코딩하는 칠공주	코딕	코딩하는 벳들을 위한 이화 개발 커뮤니티
	2	시계는영어로와치	Studing In Ewha Watch	번개 혈상의 공부 스텐디, 장기 스텐디 팟을 꾸...
	3	돼지감자	신촌대 맛집전공	대학가 맛집을 공유하고 싶은 신촌 부근 대학생...
	4	웨더레인저	See Me	대한민국 국민을 위한 날씨 통합 웹 어플리케이션...
★	NULL	NULL	NULL	NULL



LEFT JOIN

기준이 되는 표를 왼쪽에 두고, 왼쪽에 존재하는 행을 기준으로 오른쪽의 표를 합성하여 하나의 표를 만드는 방법

- 1 `SELECT member_id, name, team_name, project_title FROM efub.member LEFT JOIN efub.team ON team = team_id;`
- 2 `SELECT team_name, project_title, name, member_id FROM efub.team LEFT JOIN efub.member ON team = team_id;`

	member_id	name	team_name	project_title
▶	1	박현아	돼지감자	신촌대 맛집전공
	2	김정은	시계는영어로와치	Studing In Ewha Watch
	3	정아연	웨더레인저	See Me
	4	김혜린	코딩하는 칠공주	코딕
	5	이윤지	돼지감자	신촌대 맛집전공
	6	윤효정	시계는영어로와치	Studing In Ewha Watch
	7	김이화	NULL	NULL

team_name	project_title	name	member_id
코딩하는 칠공주	코딕	김혜린	4
시계는영어로와치	Studing In Ewha Watch	김정은	2
시계는영어로와치	Studing In Ewha Watch	윤효정	6
돼지감자	신촌대 맛집전공	박현아	1
돼지감자	신촌대 맛집전공	이윤지	5
웨더레인저	See Me	정아연	3



INNER JOIN

왼쪽과 오른쪽 표 모두에서 존재하는 행을 모아서 하나의 표를 만드는 방법. JOIN이라고만 말하면 INNER JOIN을 의미하는 것.

```
SELECT team_name, project_title, name, member_id FROM efub.team INNER JOIN efub.member ON team = team_id;  
SELECT team_name, project_title, name, member_id FROM efub.member INNER JOIN efub.team ON team = team_id;
```

	team_name	project_title	name	member_id
▶	돼지감자	신촌대 맛집전공	박현아	1
	시계는영어로와치	Studing In Ewha Watch	김정은	2
	워더레인저	See Me	정아연	3
	코딩하는 칠공주	코딩	김혜린	4
	돼지감자	신촌대 맛집전공	이윤지	5
	시계는영어로와치	Studing In Ewha Watch	윤효정	6

실행결과 동일



FULL OUTER JOIN

왼쪽과 오른쪽에 있는 행 모두를 합성해서 하나의 표를 만드는 방법. MySQL에는 FULL OUTER JOIN이 없어서 (LEFT JOIN) UNION (RIGHT JOIN)을 사용.

team_id	team_name	project_title	project_description
1	코딩하는 칠공주	코딩	코딩하는 벳들을 위한 이화 개발 커뮤니티
2	시계는영어로와치	Studing In Ewha Watch	번개 형상의 공부 스터디, 장기 스터디 짖을 꾸...
3	돼지감자	신촌대 맛집전공	대학가 맛집을 공유하고 싶은 신촌 부근 대학생...
4	웨더레인저	See Me	대한민국 국민을 위한 날씨 통합 웹 어플리케이...
5	이럽	상상프로젝트	이화인의 상상을 더해주기 위한 상상력 촉진 서...
NULL	NULL	NULL	

<- team 테이블에만
존재하는 행 추가



FULL OUTER JOIN

```
(SELECT team_name, project_title, name, member_id FROM efub.member LEFT JOIN efub.team ON team = team_id) UNION  
(SELECT team_name, project_title, name, member_id FROM efub.member RIGHT JOIN efub.team ON team = team_id);
```

	team_name	project_title	name	member_id
▶	돼지감자	신촌대 맛집전공	박현아	1
	시계는영어로와치	Studing In Ewha Watch	김정은	2
	웨더레인저	See Me	정아연	3
	코딩하는 칠공주	코딩	김혜린	4
	돼지감자	신촌대 맛집전공	이윤지	5
	시계는영어로와치	Studing In Ewha Watch	윤효정	6
	NULL	NULL	김이화	7
	이팝	상상프로젝트	NULL	NULL



뷰(View)란?

하나 이상의 기본 테이블로부터 유도된, 이름을 가지는 가상 테이블. **사용자에게 접근이 허용된 자료만 제한적으로 보여주기 위해 사용**하며 데이터 보정 작업, 처리과정 테스트 등 임시 작업에 활용된다. 물리적으로 존재하지 않지만 사용자에게 있는 것처럼 간주된다.

- 필요한 데이터만 뷰로 정의해 처리할 수 있기 때문에 **관리가 용이하고 명령문이 간단해짐.**
- 정의된 뷰는 다른 뷰의 정의에 기초가 될 수 있으며, 특정 뷰가 정의된 기본 테이블이나 뷰를 삭제하면 해당 뷰도 자동으로 삭제됨.
- 논리적 데이터 독립성을 제공하나 한 번 뷰를 정의하면 변경할 수 없고, 삽입과 삭제 등의 작업에 많은 제한 사항을 가짐.



트랜잭션이란?

데이터베이스의 상태를 변화시키기 위해 수행하는 작업의 단위. SELECT, INSERT 등의 질의어 한 문장이 아니라, 하나의 논리적 기능을 수행하기 위한 일련의 연산들을 의미.

Ex) a계좌에서 b계좌로 송금할 때는 a계좌에서 돈을 빼낸 후에 b계좌에 돈을 입금하는 것까지가 하나의 작업.

- 원자성 : 트랜잭션의 연산은 DB에 모두 반영되든가 아니면 전혀 반영되지 않아야 함.
- 일관성 : 트랜잭션의 작업 처리 결과는 항상 일관성 있어야 함.
- 독립성 : 둘 이상의 트랜잭션이 동시에 병행 실행되고 있을 때, 어떤 트랜잭션도 다른 트랜잭션의 연산에 끼어들 수 없음.
- 지속성 : 트랜잭션이 성공적으로 완료됐을 경우, 결과는 영구적으로 반영되어야 함.



트랜잭션이란?

- 커밋(Commit) : 모든 부분작업이 정상적으로 완료되면 이 변경사항을 한꺼번에 DB에 반영.
- 롤백(Rollback) : 부분 작업이 실패하면 트랜잭션 실행 전으로 되돌림.
- SAVEPOINT : 부분 작업이 실패하였을 때 전체 작업을 취소하는 것이 아니라, 특정 부분에서 트랜잭션을 취소시킬 수 있게해줌.

SAVEPOINT 세이프포인트명

ROLLBACK TO 세이프포인트명

<- 취소하려는 시점에 명시

<- SAVEPOINT 지점까지 처리한 작업이 롤백됨.



표 (table)

데이터베이스(database) = 스키마(Schema)

데이터베이스 서버(database server) : MySQL



MySQL 실습

MySQL Connections  

pw 저장

Setup New Connection

Connection Name: Type a name for the connection

Connection Method: Method to use to connect to the RDBMS

Parameters SSL Advanced

Hostname: Port: Name or IP address of the server host - and TCP/IP port.

Username: Name of the user to connect with.

Password: The user's password. Will be requested later if it's not set.

Default Schema: The schema to use as default schema. Leave blank to select it later.



Navigator

SCHEMAS

Filter objects

- efuh
 - Tables**
 - Views
 - Stored Procedures
 - Functions
- opentutorials
- sys
- workbench

Administration **Schemas**

my server x

File Edit View Query Database Server Tools Scripting Help

Navigator: Query 1 topic new_schema - Schema x

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

PERFORMANCE

- Dashboard
- Performance Reports
- Performance Schema Setup

Administration Schemas

Information Schema

No object selected

Name: new_schema Specify the name of the schema

Rename References Refactor model, changing all references

Charset/Collation: Default Charset Default Collation The character set and its collation

Apply Revert



이팝 멤버 데이터를 저장하는
테이블을 생성해봐요!

1. 임원진 정보
2. 프로젝트 팀 정보
3. 포지션 정보

efub - Schema member - Table

Table Name: member Schema: efub

Charset/Collation: Default Default Engine: InnoDB

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G
member_id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
name	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
position	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
email	VARCHAR(100)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
birth_date	DATE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Column Name: Data Type: Charset/Collation: Default: Storage: ☐ Virtual ☐ Stored ☐ Primary Key ☐ Not Null ☐ Unique ☐ Binary ☐ Unsigned ☐ Zero Fill ☐ Auto Increment ☐ Generated

Columns Indexes Foreign Keys Triggers Partitioning Options

Apply Revert



member 테이블 맨 오른쪽
표 모양 아이콘 클릭 !

member 인스턴스 insert

member x

Limit to 1000 rows

1 여기에 쿼리 작성 후 번개모양 클릭하여 쿼리 실행

Result Grid

	member_id	name	position	email	birth_date
	0	park...	back	pha...	2000-11-21
▶	NULL	kim j...	back	sva...	2000-01-24
*	NULL	NULL	NULL	NULL	NULL

Form Editor

Field Types

Query Stats

member 1 x

Apply Revert



실습 과제 : 한 번에 두 번 이상의 JOIN을 사용하세요.

hint : full outer join 혹은 3개 이상의 테이블