

데이터베이스

# DBMS 기초

## DataBase Management System

- 데이터베이스(DB)를 관리하는(Management) 시스템(System)
  - DB : 테이블들이 모여 이루는 데이터 단위
- 데이터를 저장하고 유지보수(수정, 삭제, 추가)하고 이를 검색하는 시스템
  - CRUD(Create, Retrieve, Update, Delete)
- 대량의 데이터를 처리하는 시스템
- 다양한 자료구조와 검색구조(소팅, 인덱싱..등)사용해 '빠른'검색가능
- 대부분의 시스템은 CUD(업데이트) 보다 R(검색)의 빈도수가 많은
- 검색에 최적화

# DBMS 기초

## DataBase Management System

### 정렬

- 빠른 검색을 위해서는 반드시 정렬(Sorting)되어 있어야 함
- 정렬 되어 있지 않다면 평균적으로 전체 데이터의 절반 필요(최선:1, 최악:N, 평균:N/2)
- 정렬되어 있을 경우 데이터를 빠른 시간 안에 찾을 수 있음
- 퀵정렬/힙정렬 계열이 주로 사용됨

### 인덱스

RDBMS에서 검색속도를 높이기 사용하는 하나의 기술입니다.

INDEX는 색인입니다. 해당 TABLE의 컬럼을 색인화(따로 파일로 저장)하여 검색시 해당 TABLE의 레코드를 full scan 하는게 아니라 색인화 되어있는 INDEX 파일을 검색하여 검색속도를 빠르게 합니다.

# DBMS 기초

## 데이터베이스 종류

- 계층형 데이터베이스
- 네트워크형 데이터베이스
- 관계형 데이터베이스(RDBMS)
- 객체지향 데이터베이스
- 객체관계형 데이터베이스(ORDBMS)
- NoSQL(Not Only SQL);

# DBMS 기초

## RDBMS란?

- 관계형 데이터베이스 시스템
- 테이블(**Table based**)기반의 DBMS
  - 테이블-컬럼형태의 데이터 저장 방식
  - 테이블과 테이블 간의 연관관계(주로 외래키형태)를 이용해 필요한 정보를 구하는 방식
- 모델링은 E-R(**Entity Relationship**)모델을 사용
- 테이블을 엔티티(기본)와 릴레이션쉽(유도) 테이블로 구분하는 방식

# DBMS 기초

## 기본용어

- 스키마(Schema) - DB, 테이블 정의 내역
- SQL쿼리(SQL Query)
  - 관계형 DBMS를 사용하는 전용 질의언어
  - 대소문자 가리지 않음
- 기본키(Primary Key:PK)
  - 테이블에서 하나의 레코드를 지정할 수 있는 하나 이상의 컬럼집합
  - 예) 주민등록번호 , SSN(Social Security Number)
- 외래키(Foreign Key:FK)
  - 어떤 테이블의 기본키가 다른 테이블의 컬럼에 들어 있을 경우

# DBMS 기초

## 기본용어

- 테이블(Table)
  - 정보들의 묶음단위
  - 예) 학교, 학생, 교수...
- 레코드(Record)
  - 테이블에 들어 있는 여러가지 인스턴스 하나하나를 지정
  - 대학교의 학과테이블
  - 예) 경영학과, 미술학과, 수학과, 컴퓨터공학과..
  - 기본키(PK)로 구별가능

# MySQL

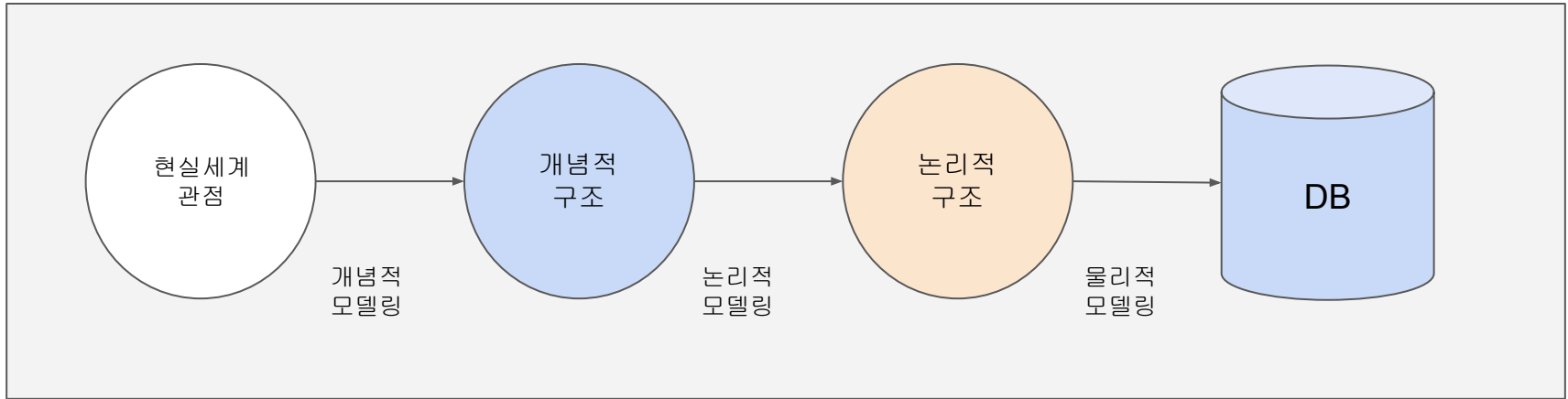
오라클에서 만든 무료(부분 유료) 오픈소스 RDBMS(관계형 데이터베이스 시스템)

- Sun Microsystems가 MySQL 인수
- 오라클이 Sun Microsystems 같이 인수
- 쌍둥이 형제인 MariaDB도 있음
- Windows/Linux/Mac용이 모두 있음
  
- MySQL 다운로드 (URL)
  - [www.mysql.com](http://www.mysql.com)
  - [dev.mysql.com/download/installer/](http://dev.mysql.com/download/installer/)



데이터 모델링

# 데이터 모델링 과정



- 개념적설계
  - 현실 세계에 대한 인식을 추상적 개념으로 표현하는 과정
- 논리적설계
  - 현실 세계에서 발생하는 데이터 형태를 컴퓨터가 이해하고 처리할 수 있는 논리적 데이터 구조로 변환시키는 과정
- 물리적구조
  - 데이터베이스 파일에 대한 특정한 저장 구조와 접근 경로를 결정하는 과정

# 데이터베이스 테이블 구조


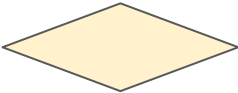
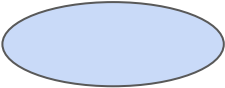

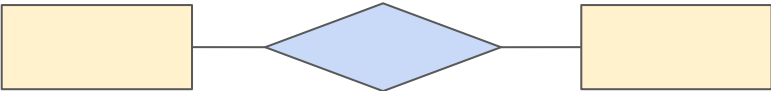

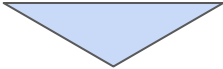
1. 스키마(Schema)
2. 레코드(Record)
3. 인스턴스(Instance)
4. 튜플(Tuple)
5. 기본키(Primary Key)
6. 도메인(Domain) - 한 속성에 나타날 수 있는 값의 범위(1~5)
7. 차수(Degree) - 레코드의 개수(6개)
8. 카디널리티(cardinality) - 인스턴스의 개수(5개)

The diagram illustrates database concepts using a table and annotations:

- Annotation ①:** Points to the entire table structure, representing the **Schema**.
- Annotation ②:** Points to the header row (num(index), name, pw, tel, email, userid), representing the **Schema**.
- Annotation ③:** Points to the data rows, representing the **Instance**.
- Annotation ④:** Points to the first column (num(index)), representing the **Primary Key**.

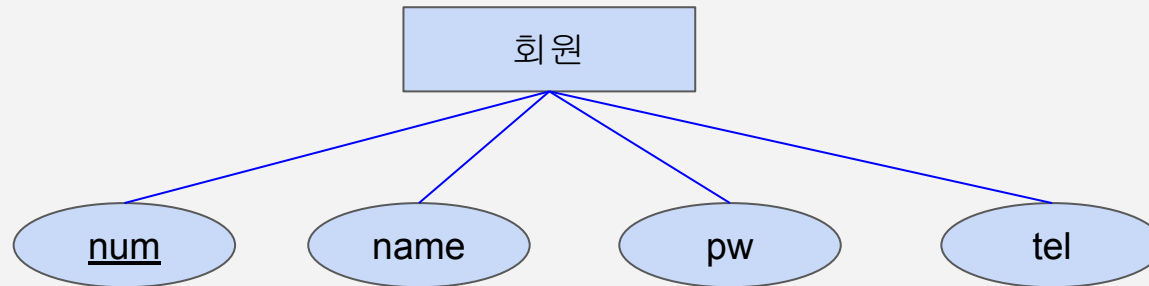
num(index)	name	pw	tel	email	userid
1	홍길동	*****	1039442612	cha@naver.com	홍길동
2	홍길동	*****	1039442612	cha@naver.com	홍길동
3	홍길동	*****	1039442612	cha@naver.com	홍길동
4	홍길동	*****	1039442612	cha@naver.com	홍길동
5	홍길동	*****	1039442612	cha@naver.com	홍길동

# ER diagram

도형	의미
	개체(Entity)
	관계(Relationship)
	속성(Attribute)
	기본키(Primary Key) 속성
	개체간의 관계(연관성)
	개체와 속성을 연결
	ISA관계(상속 관계)

# ER diagram

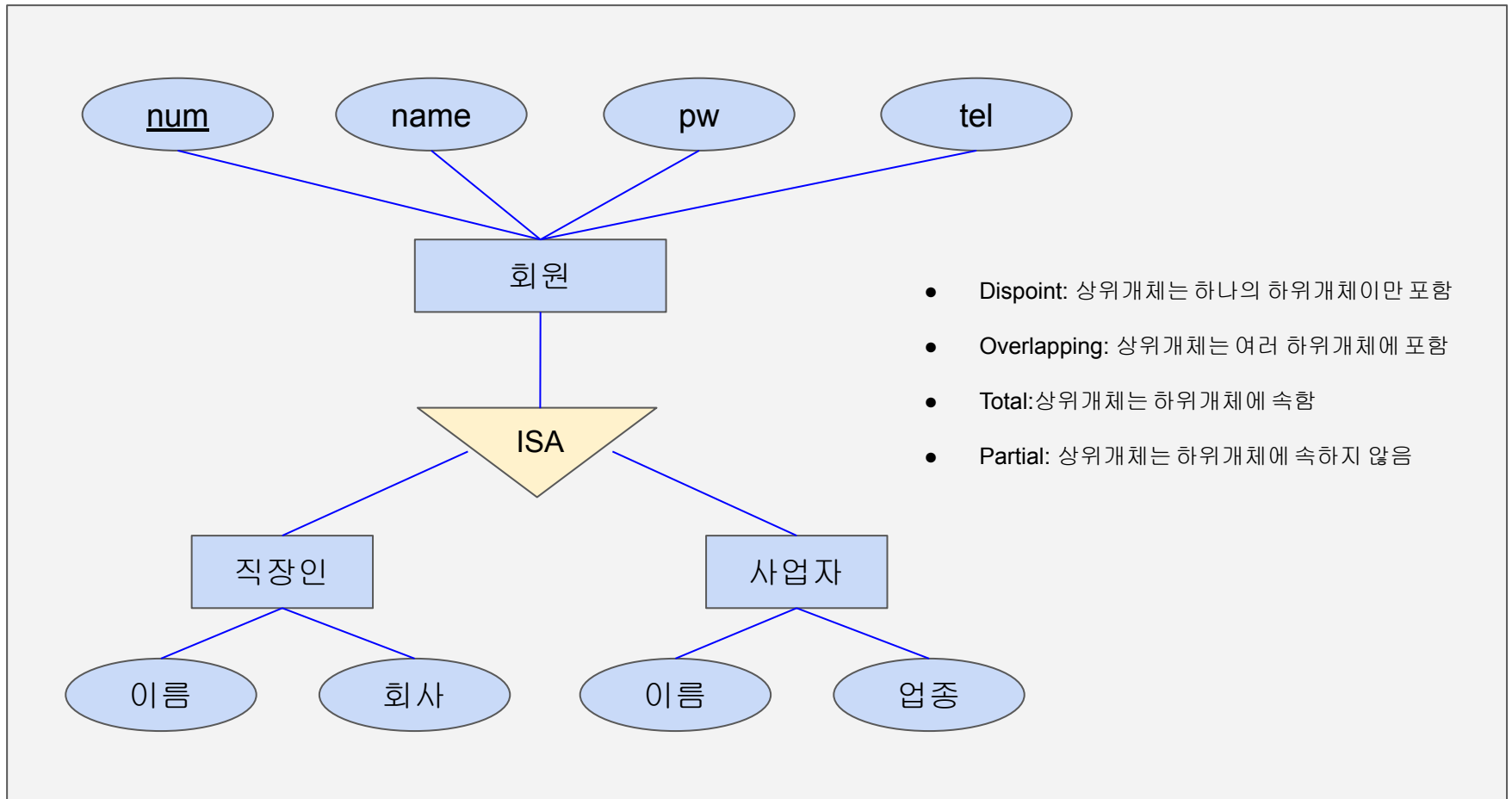
## Entity(개체) Relationship Diagram



num(index)	name	pw	tel
1	홍길동	*****	1039442612
2	홍길동	*****	1039442612
3	홍길동	*****	1039442612

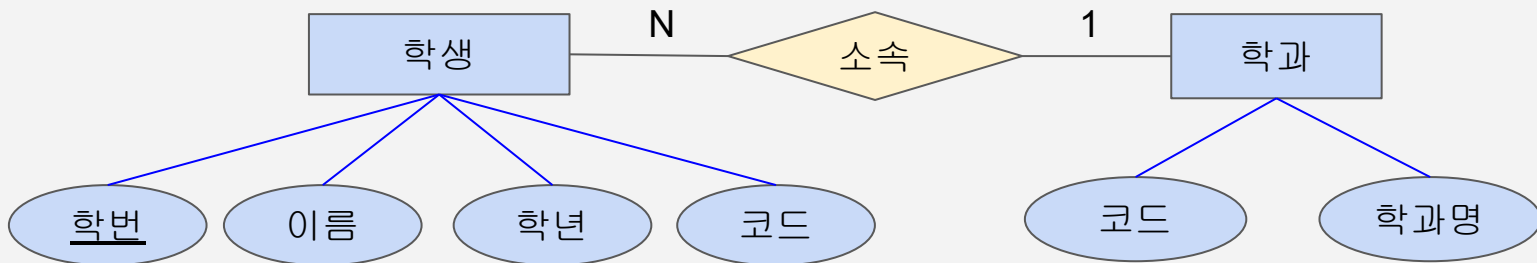
# ER diagram

## ISA 관계 (상속 관계)



# ER diagram

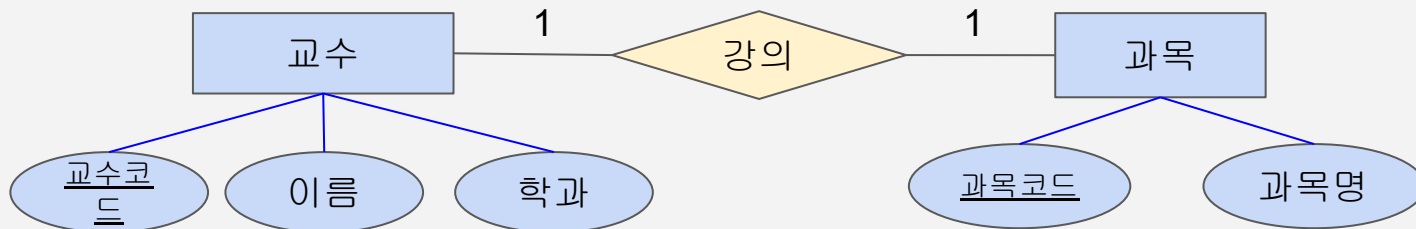
## 관계 타입 유형



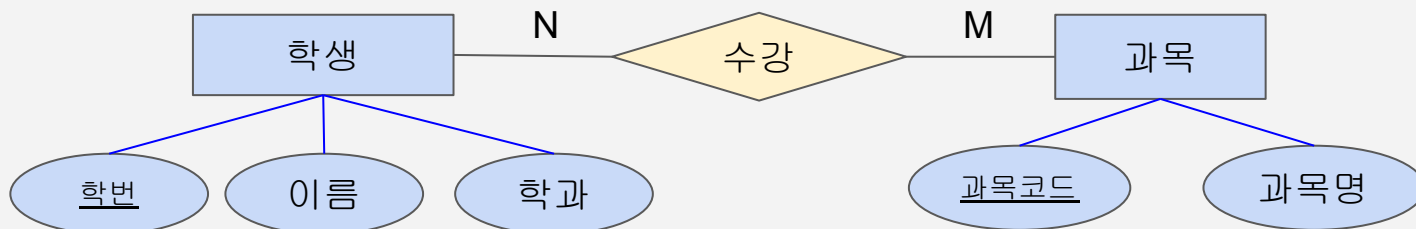
- 가정) 여러명의 학생이 하나의 학과에 소속될 수 있으며 한명의 학생이 여러개의 학과에 소속 될 수 없다.
- 1:1, 1:N, N:M

# ER diagram

## 관계 타입 유형



- 가정) 한 명의 교수는 한 과목만 강의, 한 개의 과목은 한 교수에 의해서만 강의 되어진다.

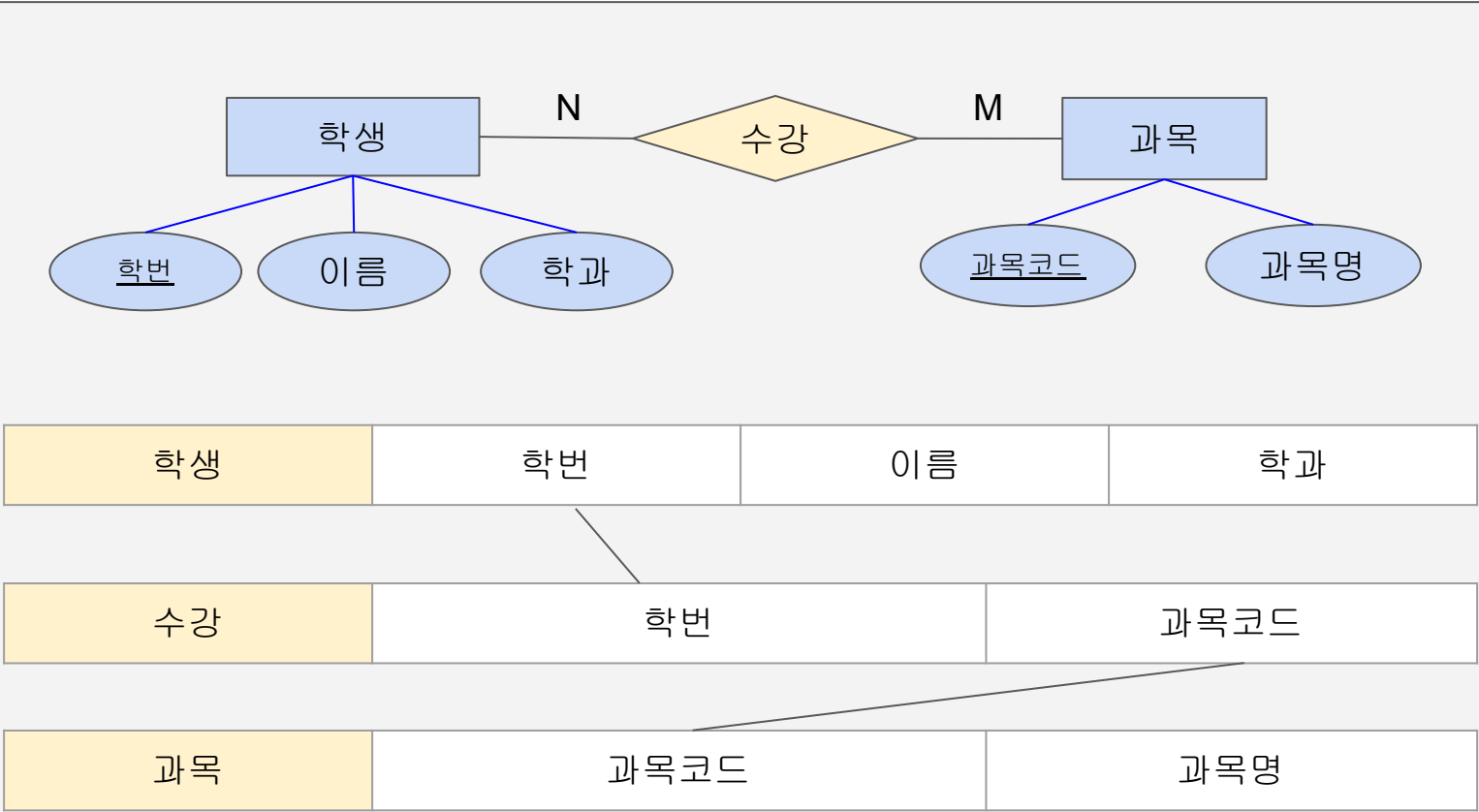


- 한 명의 학생은 여러 과목을 수강할 수 있고, 한 개의 과목은 여러 학생에 의해 수강되어질 수 있다.



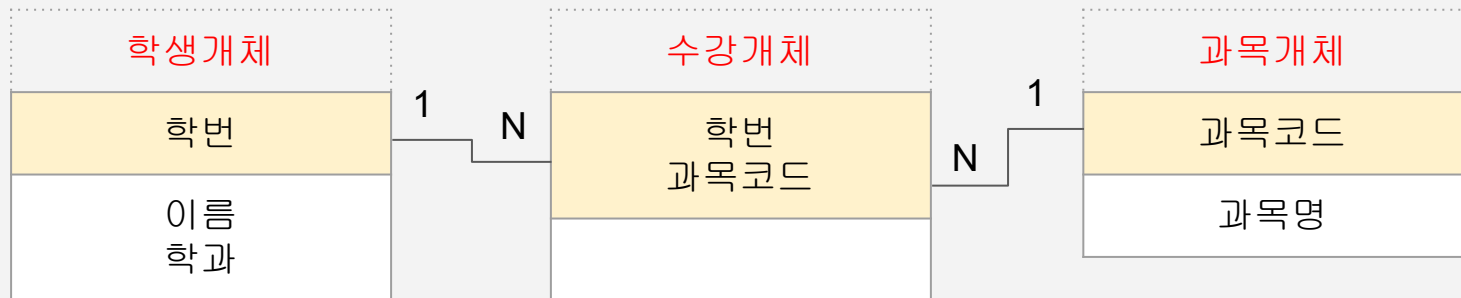
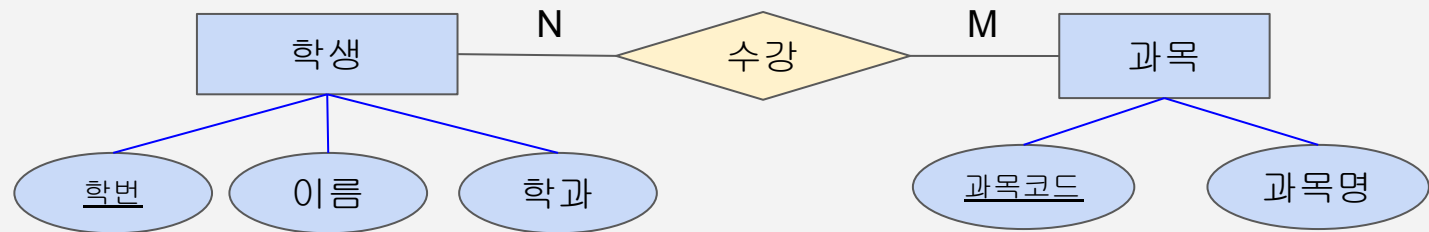
# ER 모델 정리

## 릴레이션 스키마



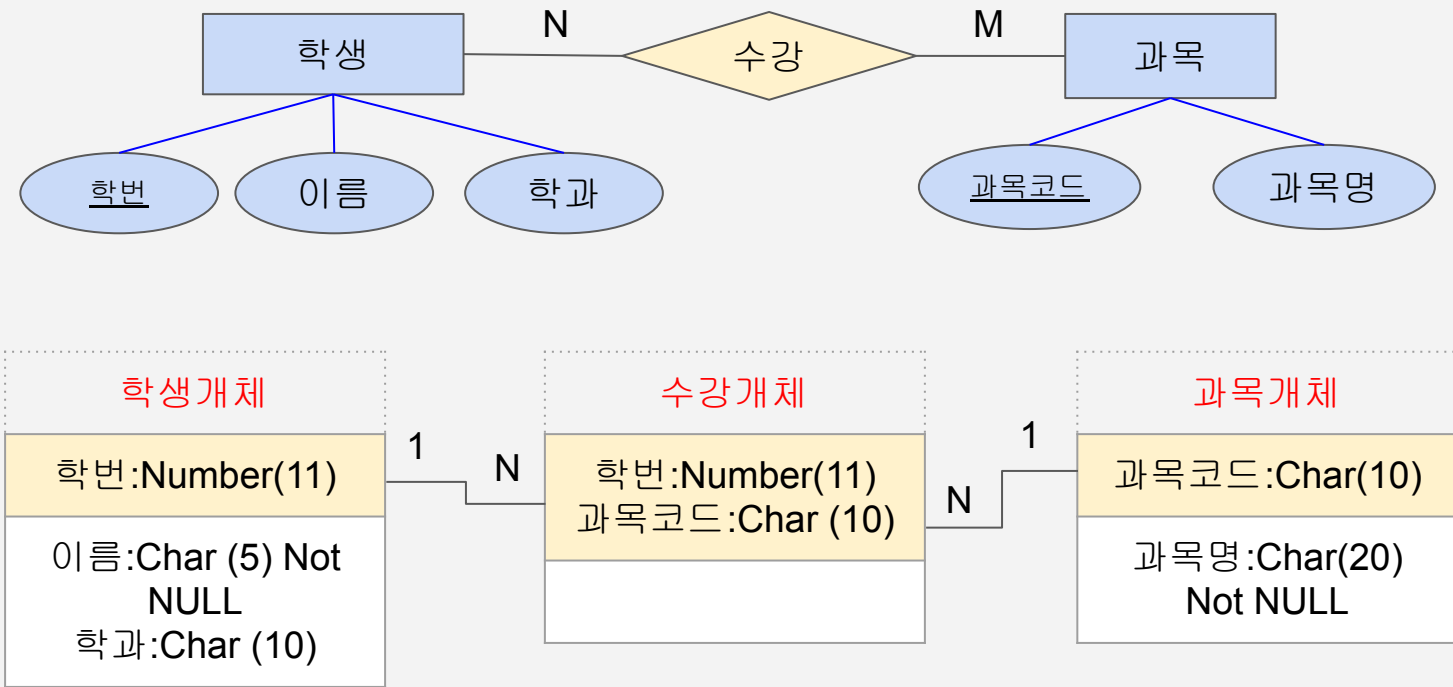
# ER 모델 정리

## 논리적설계(ERD)



# ER 모델 정리

## 물리적설계(ERD)



# 테이블 정의서

테이블 정의서						
시스템명	수강생관리				작성일	2020-05-03
테이블ID	Students				작성자	홍길동
테이블명	수강생 정보					
NO	컬럼ID	컬럼명	데이터형식	길이	KEY	NULL
1	S_Num	학번	Varchar	10	PK	N
2	S_NAME	이름	Varchar	10		N
3	S_LESS	학과	Varchar	50		Y

SQL

# SQL의 이해와 종류

## SQL의 이해

- SQL(Structured Query Language)
  - 데이터베이스에 있는 필요한 정보를 사용할 수 있도록 도와주는 언어
  - 사용방법이나 문법이 다른 언어(Java, C, C\$, VB등)보다 단순함
  - 하나를 배워두면 모든 **DBMS**에서 사용가능함
  - 인터프리터
  - 대소문자 구별하지 않음(데이터 내용은 구별함)

# SQL의 이해와 종류

## DML(Data Manipulation Language)

- 테이블의 데이터를 조작하는 기능
- 테이블의 레코드를 CRUD(Create, Retrieve, Update, Delete)

INSERT	데이터베이스 객체에 데이터를 입력 한다.
DELETE	데이터베이스 객체에 데이터를 삭제 한다.
UPDATE	데이터베이스 객체 안의 데이터를 수정 한다.
SELECT	데이터베이스 객체 안의 데이터를 조회 한다.

# SQL의 이해와 종류

## DDL(Data Definition Language)

- DB, 테이블의 스키마를 정의, 수정하는 기능
- 테이블 생성, 컬럼 추가, 타입변경, 각종 제약조건 지정, 수정 등

Create	데이터베이스 객체를 생성 한다.
Drop	데이터베이스 객체를 삭제 한다.
Alter	기존에 존재하는 데이터베이스 객체를 다시 정의한다.



# SQL의 이해와 종류

## DCL(Data Control Language)

- DB나 테이블의 접근권한이나 **CRUD**권한을 정의하는 기능
- 특정 사용자에게 테이블의 조회권한 허가/금지 등

Grant	데이터베이스 객체에 권한을 부여 한다.
Revoke	이미 부여된 데이터베이스 객체 권한을 취소한다.

# SQL의 실습 예제

## SELECT문

- SELECT 컬럼명 FROM 테이블명 WHERE 조건절;
- select \* from city;

질의	1. 국가 코드가 'KOR'으로 되어 있는 도시의 '이름'을 구하시오. 2. 인구가 500만 이상인 도시의 '이름'을 구하시오.
SQL	1. select Name from city where CountryCode='KOR'; 2. select Name from city where Population > 50000;

# SQL의 실습 예제

## INSERT문

- INSERT INTO 테이블명(컬럼명) VALUES (값);

질의	값을 추가하시오. ID => 10000, Name => 'Sample', CountryCode => 'KOR', District => 'Test', Population => 10000
SQL문	1. insert into city(ID, Name, CountryCode, District, Population) values (10000, 'Sample', 'KOR', 'Test', 10000);  2. insert into city values(10000, 'SampleTest', 'KOR', 'Test', '10000');
확인	select * from city where ID=10000;

# SQL의 실습 예제

## UPDATE문

- UPDATE 테이블명 SET 컬럼명=값 ... WHERE 조건절;

질의	Name이 'Sample'인 도시를 찾아 'SampleRevised'로 업데이트 하시오. 조건 - id=10000
SQL문	update city set name='SampleRevised' where id = 10000;
확인	select * from city where ID=10000;

# SQL의 실습 예제

## DELETE 문

- DELETE FROM 테이블명 WHERE 조건절;

질의	city테이블의 id가 10000인 데이터를 삭제하시오.
SQL문	delete from city where id=10000;
확인	select * from city where ID=10000;