

# 데이터베이스

인공지능소프트웨어학과

담당교수: 김희숙  
(jasmin11@hanmail.net)

# 데이터베이스

3주차

담당교수: 김희숙  
(jasmin11@hanmail.net)

# 실습

## 3주차

담당교수: 김희숙  
(jasmin11@hanmail.net)

# [실습] 테이블 생성

데이터베이스: testdb

테이블: 고객

테이블명세서

테이블명	열 이름	데이터 형식	NULL 유무	기본키	외래키	FK 테이블명	FK 열 이름	비고
고객	고객번호	char(4)	NOT NULL	PK				cno
	고객명	varchar(10)	NOT NULL					cname
	거주지	varchar(20)						city
	포인트	int						point

고객

고객번호	고객명	거주지	포인트
c101	홍길동	서울	500
c102	임꺽정	인천	300
c103	박찬호	안양	800
c104	신동엽	과천	350
c105	정진우	고양	400

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

sys

testdb

Tables

학기

Views

Stored Procedures

Functions

Query 1 SQL File 6 고객 - Table x

Table Name: 고객 Schema: testdb

Charset/Collation: Default Charset Default Collation Engine: InnoDB

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
고객번호	CHAR(4)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
고객명	VARCHAR(10)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
거주지	VARCHAR(20)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
포인트	INT	<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: Default: Storage: ☐ Virtual ☐ Stored ☐ Primary Key ☐ Not Null ☐ Unique ☐ Binary ☐ Unsigned ☐ Zero Fill ☐ Auto Increment ☐ Generated

Administration Schemas Information

Schema: testdb

Columns Indexes Foreign Keys Triggers Partitioning Options

Apply Revert

MySQL Workbench

Local instance MySQL80 x

File Edit View Query Database Server Tools Scripting Help

Navigator: SCHEMAS

Filter objects

- sys
- testdb
  - Tables
    - 고객
    - 학과
  - Views
  - Stored Procedures
  - Functions

Query 1 SQL File 6 고객 x

Limit to 1000 rows

```
1 • SELECT * FROM testdb.고객;
```

고객번호	고객명	거주지	포인트
c101	홍길동	서울	500
c102	임객정	인천	300
c103	박찬호	안양	800
c104	신동엽	과천	350
c105	정진우	고양	400

Result Grid

	고객 번호	고객 명	거 주 지	포 인 트
*	NULL	NULL	NULL	NULL

[실습] 고객 테이블

- 1) 데이터 입력한다
- 2) 오른쪽 하단 <Apple> 눌러야 데이터 저장된다

# [실습] SELECT (데이터 조회)

[실습] 고객

[실습 1]

- 고객(고객번호, 고객명, 거주지, 포인트)
- 1-1) 테이블의 모든 열을 검색하라

```
SELECT *  
FROM   고객 ;
```

# [실습] SELECT (데이터 조회)

## [실습 1]

- 고객(고객번호, 고객명, 거주지, 포인트)
- 1-2) 고객의 고객명, 거주지를 검색하라(테이블의 특정 열을 검색)

```
SELECT  고객명, 거주지  
FROM    고객 ;
```



# [실습] SELECT (데이터 조회)

[실습 1]

- 고객(고객번호, 고객명, 거주지, 포인트)
- 2-1) 고객번호가 c101 인 고객의 모든 정보를 검색하라

```
SELECT *  
FROM   고객  
WHERE  고객번호 = 'c101';
```

# [실습] SELECT (데이터 조회)

[실습 1]

- 고객(고객번호, 고객명, 거주지, 포인트)
- 2-2) 포인트가 400 이하인 고객의 모든 정보를 검색하라

```
SELECT *  
FROM   고객  
WHERE  포인트 <= 400;
```

# [Quiz] MySQL 실습



제품

제품번호	제품명	재고량	단가	제조업체
p01	그냥만두	5000	4500	대한식품
p02	매운짬면	2500	5500	민국푸드
p03	쿵떡파이	3600	2600	한빛제과
p04	맛난초콜렛	1250	2500	한빛제과
p05	얼큰라면	2200	1200	대한식품

```
-- 데이터 조회  
select * from 제품;
```

```
-- 화면캡처(win)  
<윈+shift+s>
```

```
-- 제품 테이블에서 제품명, 재고량을 검색하라  
select 제품명, 재고량  
from 제품;
```

```
-- 제품 테이블에서 제조업체가 대한식품인  
-- 제품명, 단가, 제조업체를 검색하라  
select 제품명, 단가, 제조업체  
from 제품  
where 제조업체 = '대한식품';
```

# 데이터베이스 개념

3주차

담당교수: 김희숙  
(jasmin11@hanmail.net)

# 데이터베이스 개념



## • 데이터베이스:

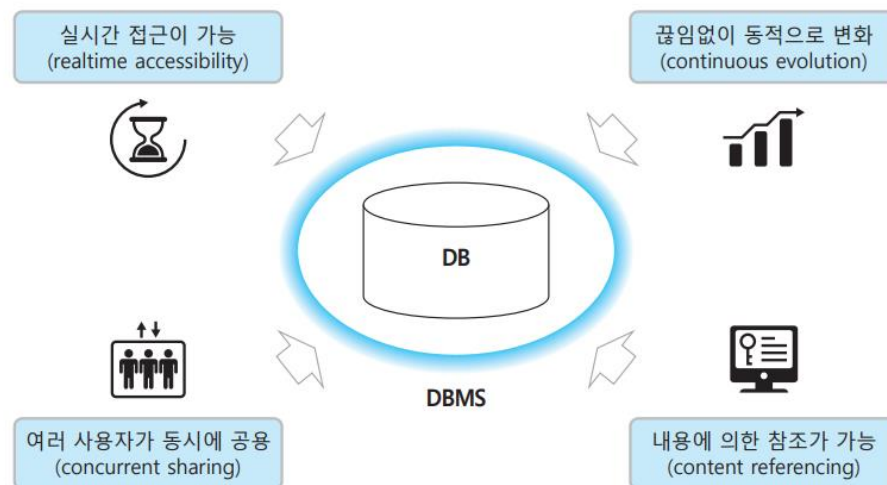
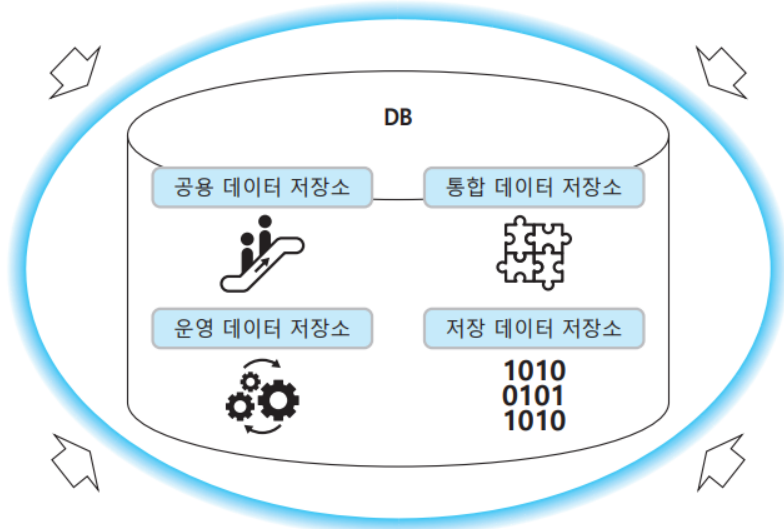
- 한 조직의 여러 응용시스템들이 데이터를 **공유**할 수 있도록 **통합**, **저장**된 **운영**데이터의 집합체

### \* 데이터베이스 정의:

- 1) 공유 데이터(shared data)
- 2) 통합 데이터(integrated data)
- 3) 저장 데이터(stored data)
- 4) 운영 데이터(operational data)

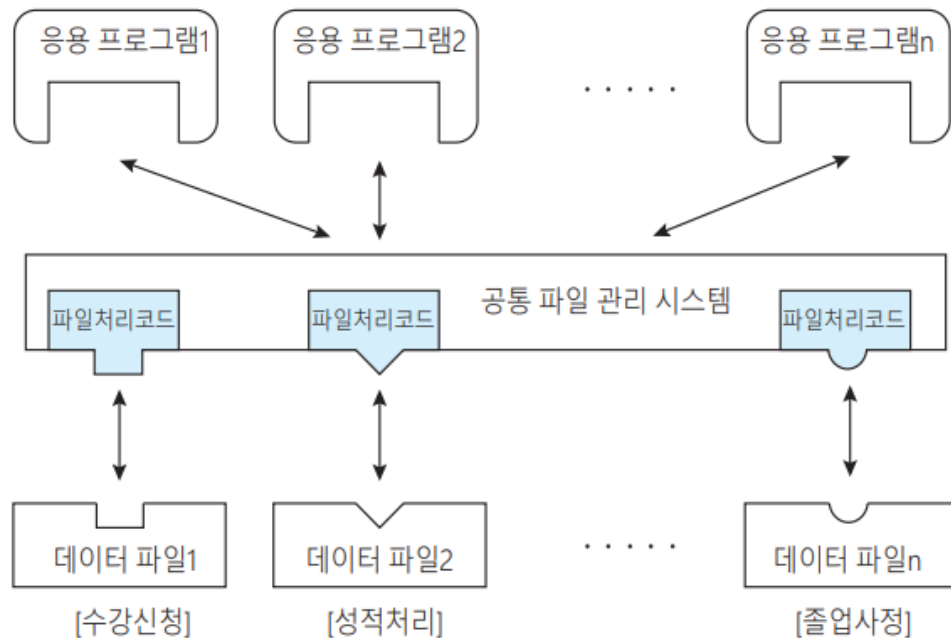
### \* 데이터베이스 특성:

- 1) 실시간 접근이 가능(real time accessibility)
- 2) 계속 변화(continuous evolution)
- 3) 동시 공유 가능(concurrent sharing)
- 4) 내용으로 참조 가능(contents references)



# 파일 시스템

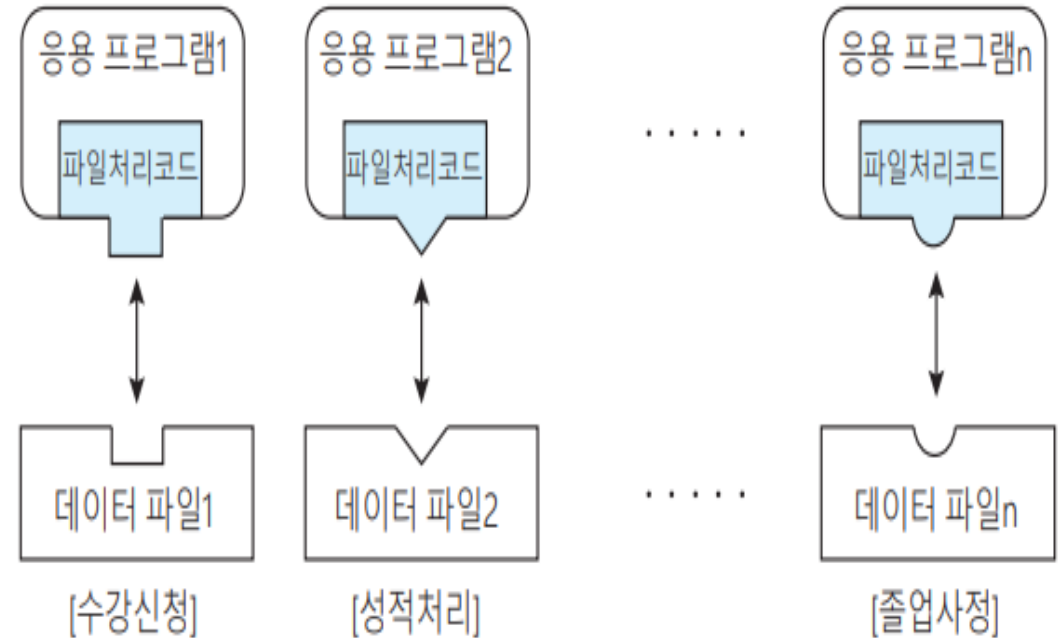
\* 데이터 종속성:  
프로그램이 변경되면 데이터도 변경하여야 한다



데이터 중복성

\* 파일시스템 단점:

- 1) 데이터 중복성
- 2) 데이터 종속성

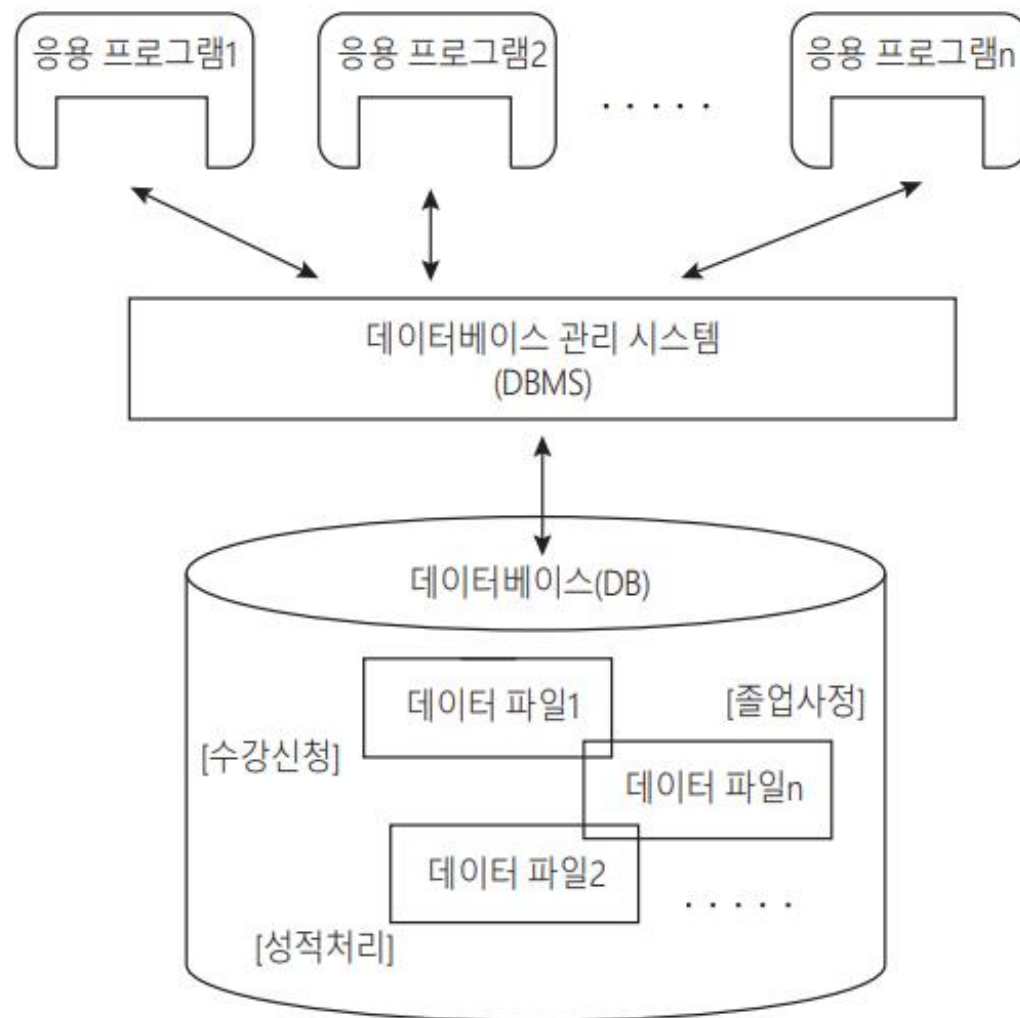


데이터 종속성

# 데이터베이스 관리 시스템(DBMS)

\* DBMS 장점:

- 1) 중복 최소화
- 2) 데이터 독립성



# 파일시스템 vs. DBMS



- **데이터베이스:**

- 한 조직의 여러 응용시스템들이 데이터를 **공유**할 수 있도록 **통합**, **저장**된 **운영**데이터의 집합체

- \* **DBMS:**

- 1) 데이터베이스를 관리하는 시스템 소프트웨어
- 2) 사용자 또는 응용프로그램과 데이터베이스 간의 인터페이스 역할을 하는 논리적인 프로그램

- \* **DBMS 장점:**

- 1) 중복 최소화
- 2) 데이터 **독립성**



- \* **파일시스템 단점:**

- 1) 데이터 **중복성**
- 2) 데이터 **종속성**



# 데이터베이스 시스템

## 3주차

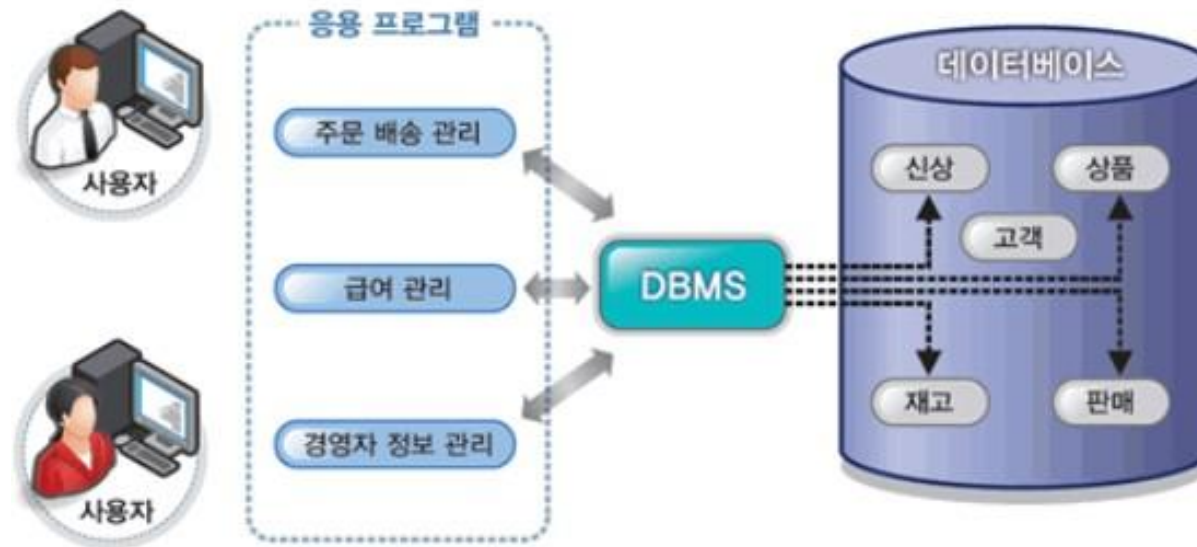
담당교수: 김희숙  
(jasmin11@hanmail.net)

# 데이터베이스 시스템: 구성요소



## □ 데이터베이스 시스템

✓ 데이터(데이터베이스), 하드웨어(HW), DBMS, 사용자, (데이터언어)



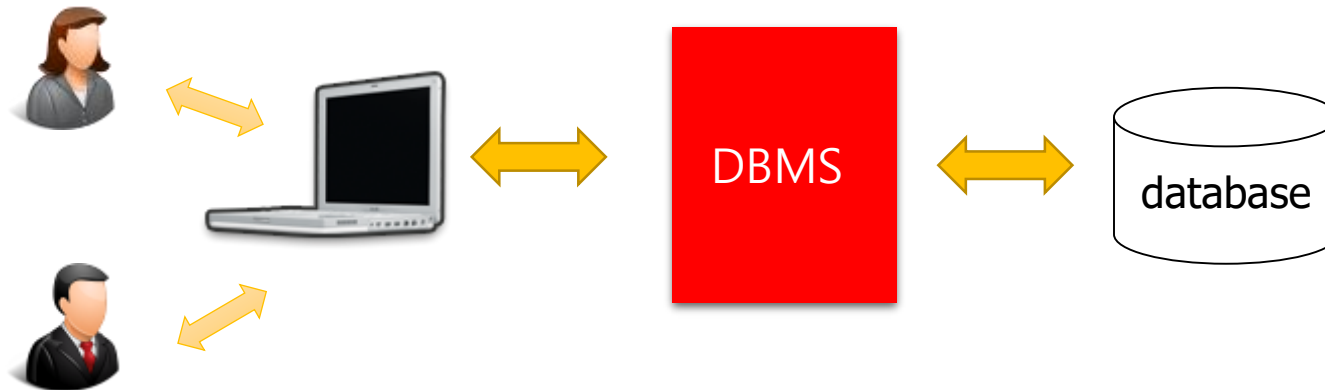
(그림 출처: "Understanding of Database", 이상구외 공저, 이한, 2012)

# 데이터베이스 관리 시스템(DBMS)



## □ 데이터베이스 관리 시스템(DBMS)

- ✓ 데이터베이스를 관리하는 시스템 소프트웨어
  - ✓ MS SQL, MySQL, Oracle, DB2
  - ✓ MariaDB, CUBRID, PostgreSQL



# 데이터베이스 사용자



## □ 데이터베이스 사용자

- ✓ 데이터베이스 관리자: DBA
- ✓ 응용 프로그래머
- ✓ 일반 사용자

# DBA: 데이터베이스 관리자



## □ 데이터베이스 관리자 역할: DBA

- ✓ 데이터베이스 시스템의 구성 요소 선정
- ✓ 데이터베이스의 구조 정의
- ✓ 물리적 저장 구조와 접근 방법 결정
- ✓ 무결성 유지를 위한 제약 조건 정의
- ✓ 보안 및 접근 권한 정책 결정
- ✓ 백업 및 회복 기법 정의
- ✓ 시스템 데이터베이스 관리
- ✓ 시스템 성능 감시 및 성능 분석
- ✓ 데이터베이스의 재구성
- ✓ 데이터베이스 관련 의견 조정과 분쟁 해결

\* DBA: 슈퍼 유저(Super user)  
데이터베이스를 구축하고  
데이터베이스 시스템을 자체적으로 운영·통제하는  
사용자

# 데이터 언어(SQL)



## □ 데이터 언어(SQL)

- ✓ 데이터 정의어(DDL)
- ✓ 데이터 조작어(DML)
- ✓ 데이터 제어어(DCL)

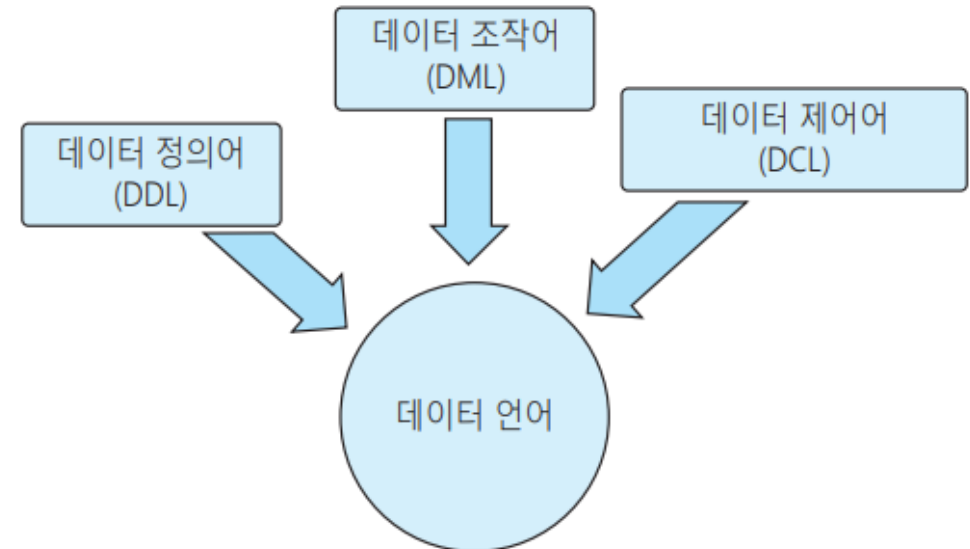
SQL	명령어
DDL	CREATE ALTER DROP
DML	INSERT UPDATE DELETE SELECT
DCL	GRANT REVOKE

# 데이터 언어



## □ 데이터 언어

- ✓ 데이터 정의어(DDL): Data Definition Language
- ✓ 데이터 조작어(DML): Data Manipulation Language
- ✓ 데이터 제어어(DCL): Data Control Language



# SQL





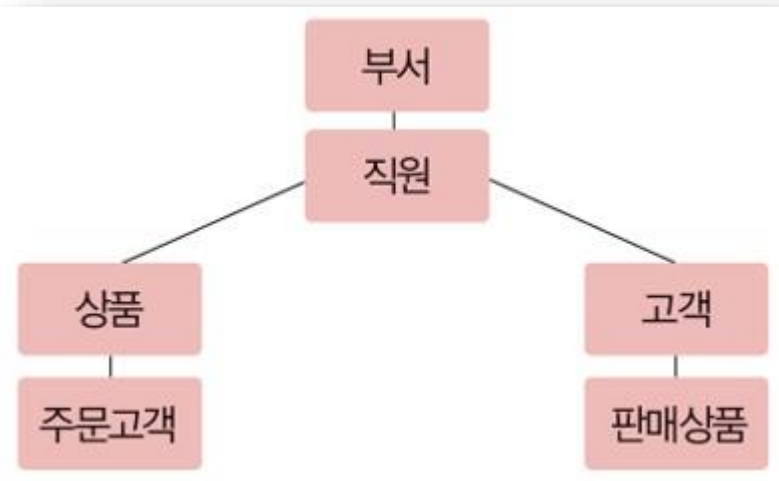
# 데이터베이스 시스템 발전 과정

## • 데이터베이스 관리 시스템

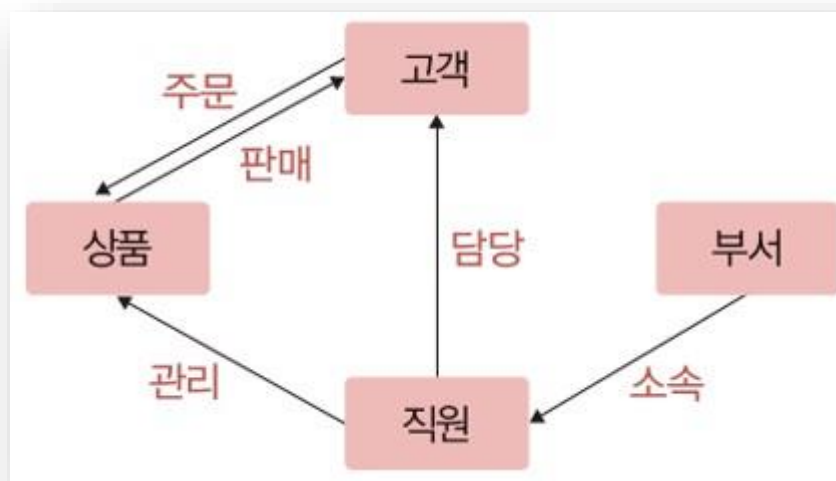
- 계층형 데이터 모델: HDBMS
- 네트워크형 데이터 모델: NDBMS
- 관계형 데이터 모델: RDBMS
- 객체지향형 데이터 모델: OODBMS
- 객체관계형 데이터 모델: ORDBMS

아이디	비밀번호	이름	연락처	주소	적립금
apple	1234	정소화	02-111-1111	서울시 마포구	1000
banana	9876	김선우	02-222-2222	경기도 부천시	500

관계형 데이터 모델



계층형 데이터 모델

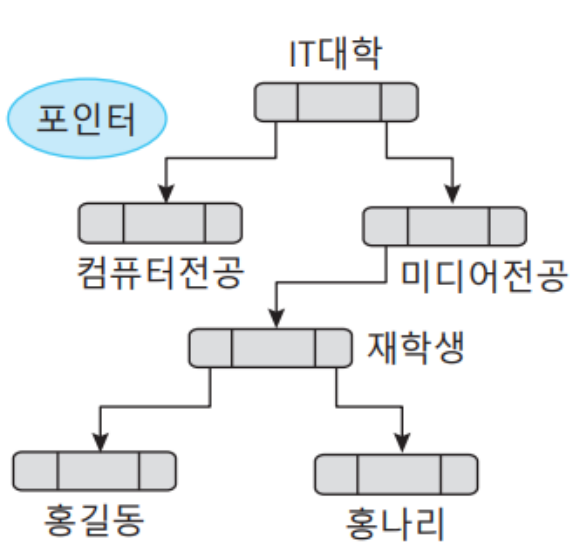


네트워크형 데이터 모델

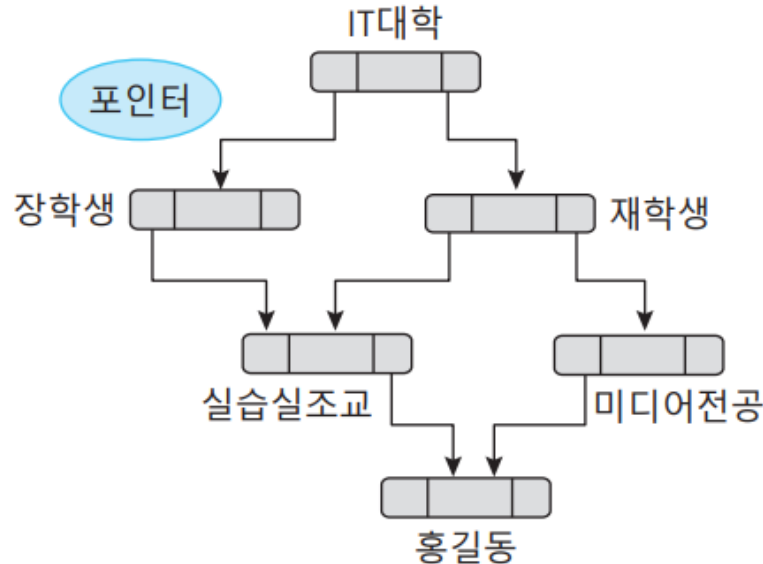
# 데이터 모델

## \* 데이터 모델(Data Model):

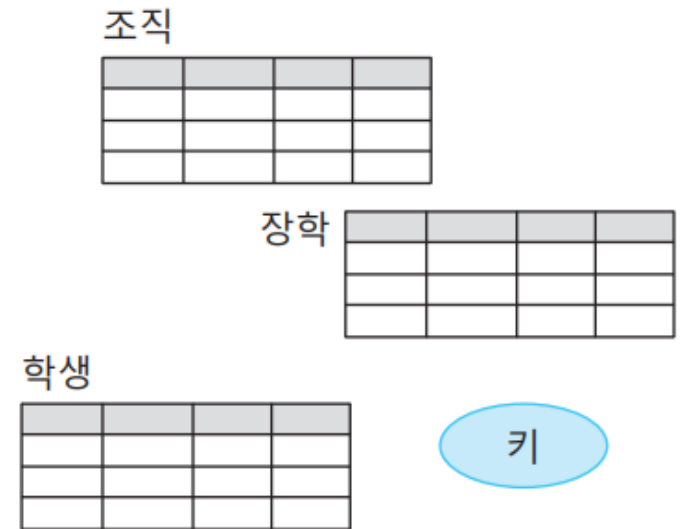
데이터베이스 종류에 따라서 다양한 저장 구조를 갖는다  
데이터베이스 구조를 명세하기 위한 개념



계층형 데이터 모델



네트워크형 데이터 모델

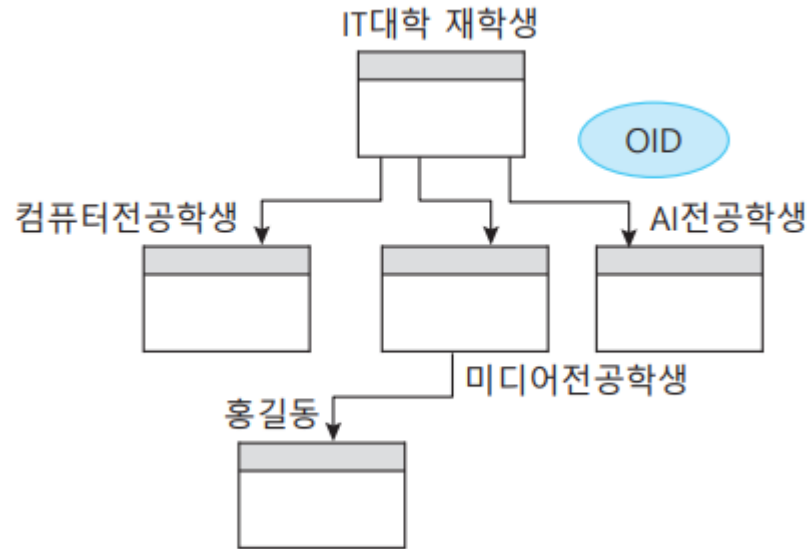


관계형 데이터 모델

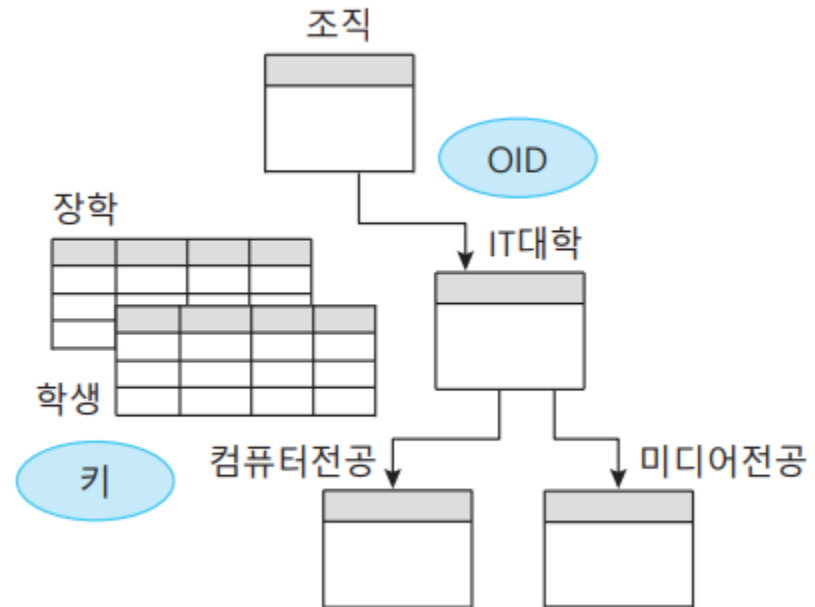
# 데이터 모델

## \* 데이터 모델(Data Model):

데이터베이스 종류에 따라서 다양한 저장 구조를 갖는다  
데이터베이스 구조를 명세하기 위한 개념



객체 지향형 데이터 모델



객체 관계형 데이터 모델

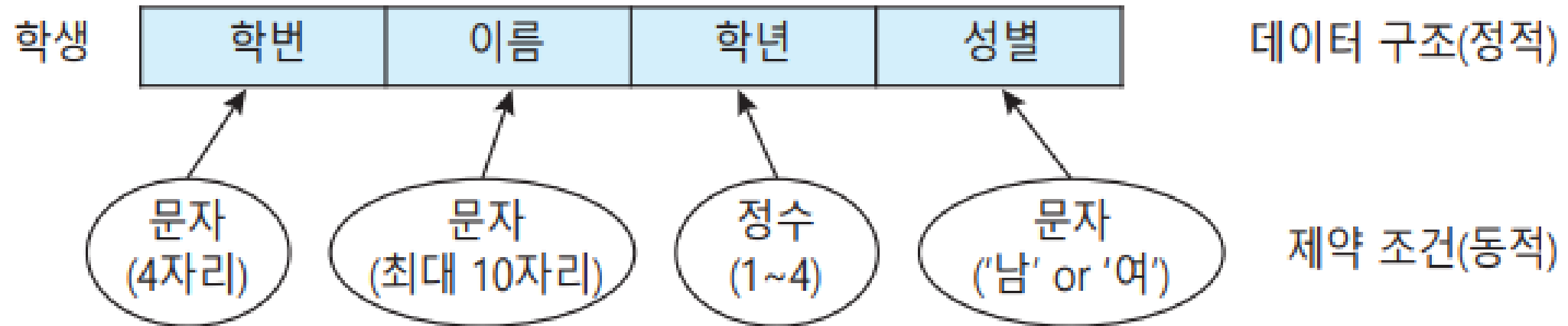


# 스키마

\* 스키마(Schema):

데이터베이스를 구성하는 구조와 제약조건에 대한 명세를 기술한 것

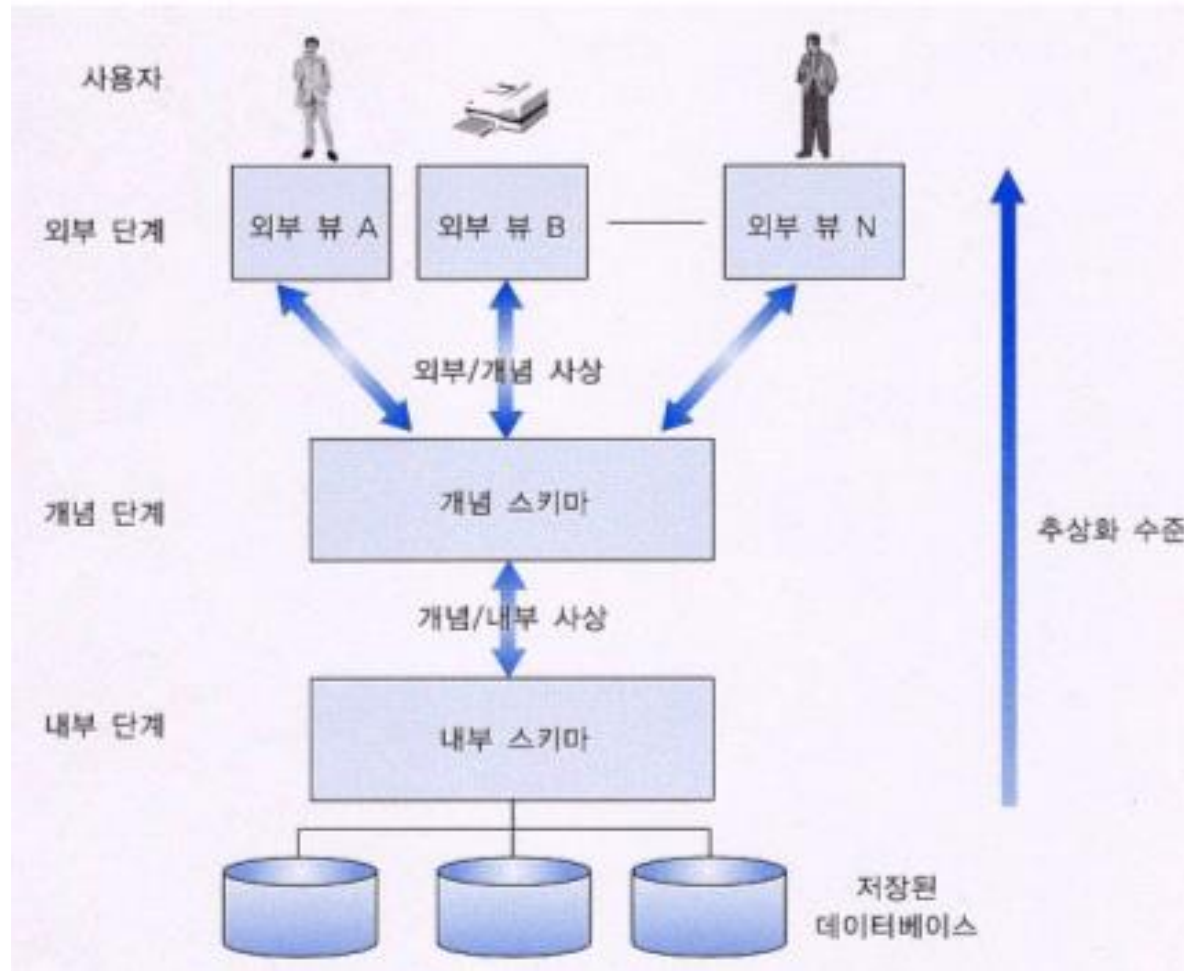
예) 학생 스키마



학생(학번, 이름, 학년, 성별)

# [요약] 3단계 구조

ANSI/SPARC 아키텍처 (스키마 3단계 구조)



(그림 출처: "데이터베이스배움터", 홍의경 저, 생능, 2012)





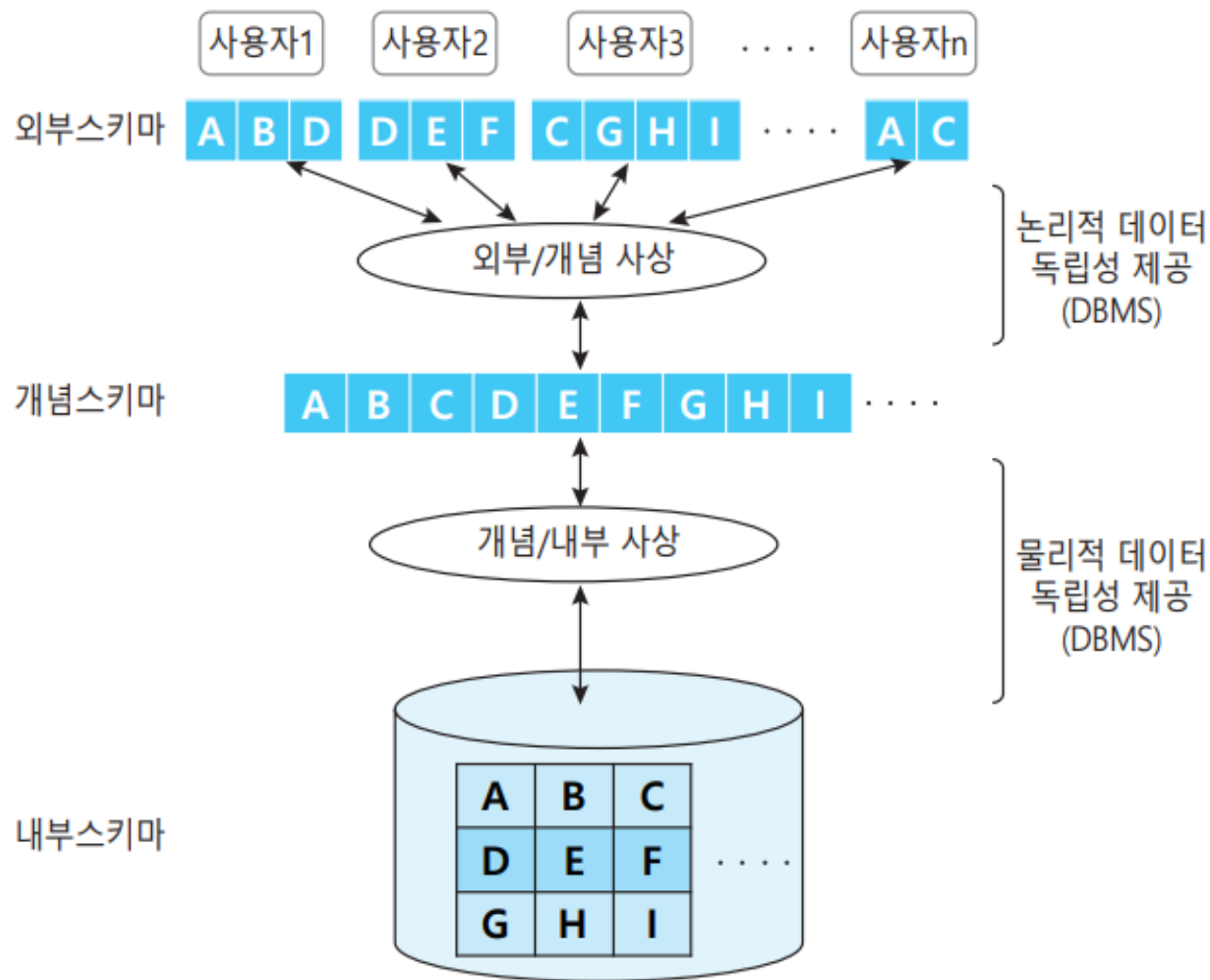
# 데이터 독립성

## \* 3단계 스키마 구조:

- 1) 외부 스키마
- 2) 개념 스키마
- 3) 내부 스키마

## \* 데이터 독립성:

논리적 데이터 독립성  
물리적 데이터 독립성



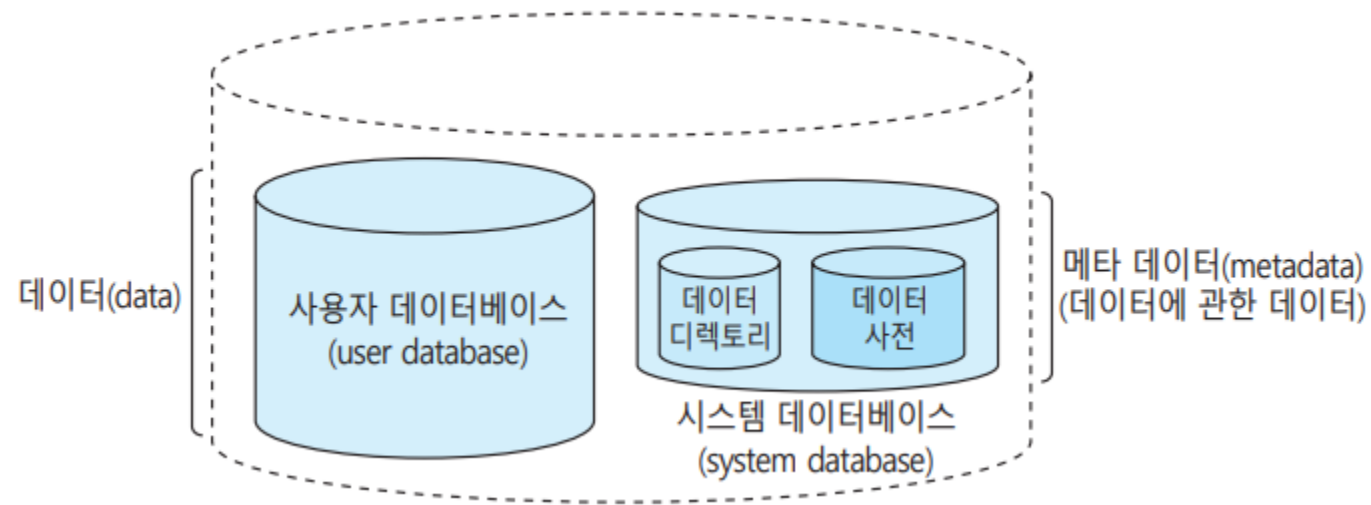
(그림 출처: "SQL과 NoSQL 기반의데이터베이스 입문", 박성진, 생능, 2023)

# 데이터 사전

## 데이터 사전

- \* 데이터 사전(data dictionary)
- \* 데이터 디렉토리(data directory)
- \* 메타 데이터: 데이터에 관한 데이터

## 데이터 사전



(그림 출처: "'SQL과 NoSQL 기반의 데이터베이스 입문', 박성진, 생능, 2023)

# 실습

## 3주차

담당교수: 김희숙  
(jasmin11@hanmail.net)



# [실습] MySQL

- 1단계: 데이터베이스 생성(스키마 생성)
- 사용할 데이터베이스 선택
- 2단계: 테이블 생성
- 3단계: 데이터 입력

[테이블 명세서] ↗

테이블이름: 학과

테이블 이름	열 이름	데이터 형식	제약조건	기본키	외래키	FK테이블	FK열
학과	학과번호	int	NOT NULL	PK			
	학과명	varchar(50)					

학과(학과번호, 학과명)

학과 테이블 <<출력 형태>>

학과

학과번호	학과명
1	컴퓨터소프트웨어공학과
2	컴퓨터정보공학과
3	인공지능소프트웨어과

# [실습] MySQL



-- MySQL Workbench 실행

-- (MySQL)

```
drop database studydb;
```

```
CREATE DATABASE `studydb`  
  DEFAULT CHARACTER SET utf8mb4;
```

```
use studydb;
```

1단계: 스키마/데이터베이스 생성 **studydb**

2단계: 테이블 생성

학과

3단계: 데이터 입력

-- 1단계: 스키마 생성

```
CREATE SCHEMA `studydb` DEFAULT CHARACTER SET utf8mb4 ;
```

-- 사용할 데이터베이스 선택

```
USE studydb;
```

-- 데이터베이스 삭제

-- (MySQL)

```
drop database studydb;
```

# [실습] MySQL



[테이블 명세서] ↗

테이블이름: 학과

테이블 이름	열 이름	데이터 형식	제약조건	기본키	외래키	FK테이블	FK열
학과	학과번호	int	NOT NULL	PK			
	학과명	varchar(50)					

-- 학과 테이블 생성

```
CREATE TABLE 학과 (  
    학과번호 int NOT NULL ,  
    학과명 varchar(50) ,  
    PRIMARY KEY(학과번호)  
);
```

-- 학과(학과번호,학과명)

```
INSERT INTO 학과 VALUES(1,'컴퓨터소프트웨어공학과');  
INSERT INTO 학과 VALUES(2,'컴퓨터정보공학과');  
INSERT INTO 학과 VALUES(3,'인공지능소프트웨어과');
```

학과

학과번호	학과명
1	컴퓨터소프트웨어공학과
2	컴퓨터정보공학과
3	인공지능소프트웨어과

```
select * from 학과;
```

# [실습] 테이블 생성

[Quiz 1-1] SQL 문법을 사용하여 테이블 생성하고 데이터 입력하시오 (studept-k.sql)

--학과(학과번호, 학과명)

테이블이름: 학과

테이블 이름	열 이름	데이터 형식	제약조건	기본키	외 레키	FK테이블	FK열
학과	학과번호	int	NOT NULL	PK			
	학과명	varchar(50)					

drop table 학과;

--학과(학과번호, 학과명)

```

[ ] 학과 (
    학과번호 int [ ],
    학과명 varchar(50),
    [ ] KEY(학과번호)
);
    
```

select \* from 학과;

## [Quiz01]

### [Quiz01] MySQL에서 실습하시오

아이디	비밀번호	이름	연락처	주소	적립금
apple	1234	정소화	02-111-1111	서울시 마포구	1000
banana	9876	김선우	02-222-2222	경기도 부천시	500

테이블 이름	열 이름	데이터 형식	NULL 유무	기본키
	아이디			
	비밀번호			
	이름			
	연락처			
	주소			
	적립금			

# [실습] 테이블 생성

데이터베이스: testdb

테이블: 제품

제품

제품번호	제품명	재고량	단가	제조업체
p01	그냥만두	5000	4500	대한식품
p02	매운짬면	2500	5500	민국푸드
p03	쿵떡파이	3600	2600	한빛제과
p04	맛난초콜렛	1250	2500	한빛제과
p05	얼큰라면	2200	1200	대한식품

```
USE testdb;
```

```
DROP TABLE if exists 제품;
```

```
CREATE TABLE 제품 (  
    제품번호    char(3)          NOT NULL ,  
    제품명      varchar(20)      ,  
    재고량      int ,  
    단가        int ,  
    제조업체    varchar(20)      ,  
    PRIMARY KEY(제품번호)  
);
```