

# 리눅스

이게 다 구성되면 서비스를 제공할 수 있다.

- 순서 : 제14장 ( Linux AMP LAMP )  
Webserver(Apache), Database(MySQL), 웹 프로그래밍 PHP)  
MySQL에서 MariaDB로 바뀜
- 학기 : 2019학년도 1학기
- 학과 : 가천대학교 컴퓨터공학과 2학년
- 교수 : 박 양 재

# 목차 (LAMP)

---

1. LAMP(Linux, Apache, MySQL, PHP)
2. 데이터베이스 개념
3. SQL 구문
4. SQL 실습
5. MariaDB 설치 및 운영
6. Apache web server 설치 및 운영
7. Web program PHP 설치 및 운영
8. 보고서

# 1. LAMP(Linux, Apache, MySQL, PHP)

- CentOS7 운영체제에 Apache(웹서버)와 MariaDB(MySQL) (데이터베이스), PHP( 웹프로그래밍) 이 연동하여 인터넷 상에서 웹서비스가 가능한 시스템을 구축

- 구축 순서

- 1) 서버 IP 주소 확인

```
# ifconfig
```

- 1) 설치 확인

```
# rpm -qa httpd php mariadb-server
```

- 1) 설치

```
# yum -y install mariadb-server mariadb php php-mysqlnd php-gd
```

- 1) 재시작 및 상시 가동 :

```
# systemctl restart httpd
```

```
# systemctl restart mariadb
```

- 5) 방화벽 포트(http(80), https(443) 포트 오픈)

- 6.외부 접속

# 1. LAMP(Linux, Apache, MySQL, PHP)

■ **httpd(웹서버: Apache)와 MariaDB (데이터베이스), PHP(웹프로그램)설치**

■ **Apache 웹서버 설치**

```
[root@localhost ~] # yum install httpd
```

■ **Apache 웹서버 버전 체크**

```
[root@localhost ~] # httpd -v
Server version: Apache/2.4.6 (CentOS)
Server built: Apr 24 2019 13:45:48
[root@localhost ~] #
```

■ **Apache 웹서버 실행 및 확인**

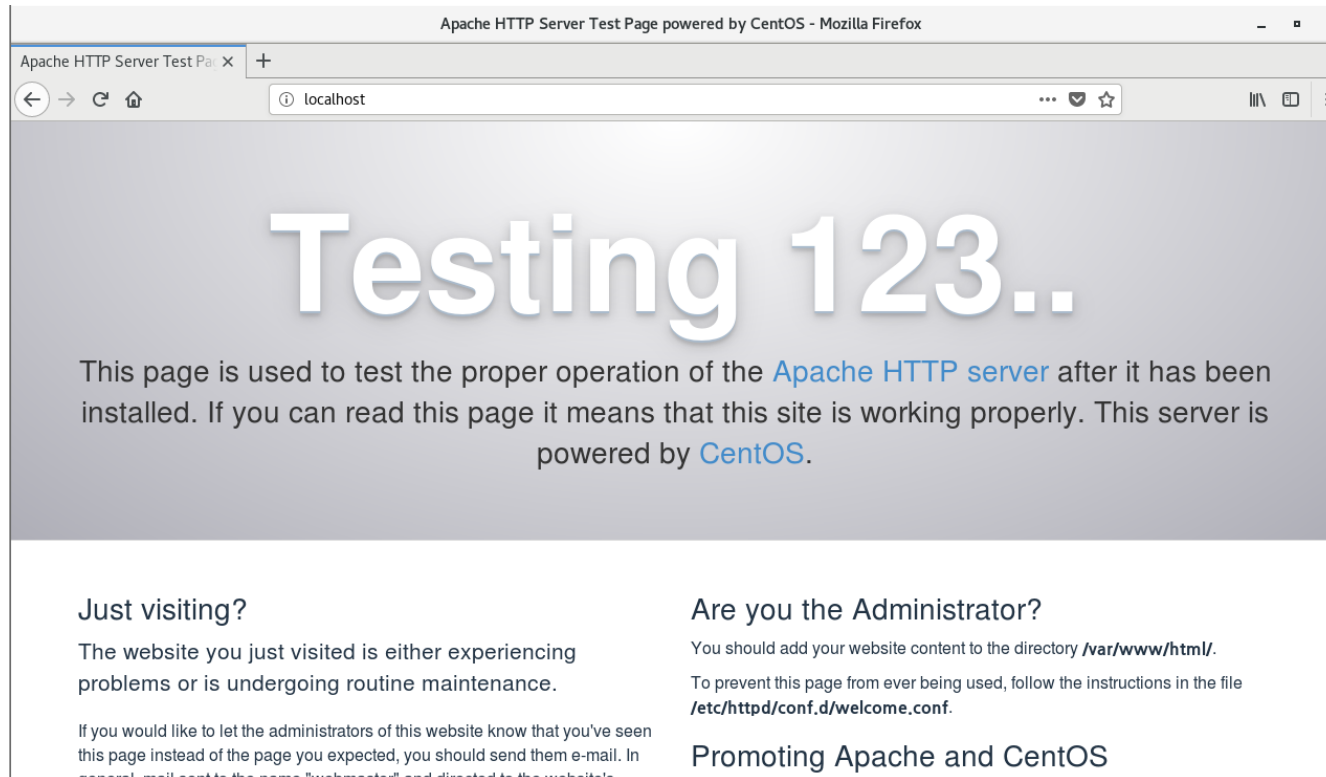
```
[root@localhost ~] # systemctl start httpd
[root@localhost ~] #
[root@localhost ~] # systemctl status httpd
● httpd.service - The Apache HTTP Server
   Loaded: loaded (/usr/lib/systemd/system/httpd.service; enabled; vendor preset: disabled)
   Active: active (running) since 목 2019-05-23 14:20:04 KST; 20min ago
     Docs: man: httpd(8)
           man: apachectl(8)
   Main PID: 17481 (httpd)
```

# 1. LAMP(Linux, Apache, MySQL, PHP)

■ httpd(웹 서버: Apache)와 MariaDB (데이터베이스), PHP(웹 프로그램)설치

■ Firefox 웹 브라우저에서 접속 테스트(**http://localhost**)

http://127.0.0.1  
-> 똑같은 결과



인트로 페이지라고도 함  
php 사용하면 /index.php  
java면 /index.jsp

# 1. LAMP(Linux, Apache, MySQL, PHP)

- httpd(웹 서버: Apache)와 MariaDB (데이터베이스), PHP(웹 프로그램)설치

- PHP 5.x 설치

```
[root@localhost ~]# yum install php
```

- Apache 웹서버를 중단 시킨다.

```
[root@localhost ~]# systemctl stop httpd
```

- PHP에 이미지 처리 모듈 추가

```
[root@localhost ~]# yum install php-gd
```

- PHP 다국어 처리 모듈 추가

```
[root@localhost ~]# yum install php-mbstring
```

- Apache 웹 서버를 설정 파일에 php 등록 후 restart 한다.

```
[root@localhost ~]# systemctl restart httpd
```

# 1. LAMP(Linux, Apache, MySQL, PHP)

■ **httpd(웹 서버: Apache)와 MariaDB (데이터베이스), PHP(웹 프로그램)설치**

■ **Mariadb 설치(버전 5,5,41)**

```
[root@localhost ~]# yum install mariadb
[root@localhost ~]# yum install mariadb-server
```

■ **Mariadb 시작**

```
[root@localhost ~]# systemctl start mariadb
[root@localhost ~]#
[root@localhost ~]# systemctl status mariadb
```

enable 설정도 해주면 좋음

```
● mariadb.service - MariaDB database server
   Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; vendor preset: disabled)
   Active: active (running) since 목 2019-05-23 14:19:56 KST; 29min ago
 Main PID: 17251 (mysqld_safe)
    Tasks: 20
   CGroup: /system.slice/mariadb.service
           └─17251 /bin/sh /usr/bin/mysqld_safe --basedir=/usr
           └─17413 /usr/libexec/mysqld --basedir=/usr --datadir=/var/lib/mysql --plu...
```

# 1. LAMP(Linux, Apache, MySQL, PHP)

- httpd(웹서버: Apache)와 MariaDB (데이터베이스), PHP(웹프로그램)설치
- Mariadb 기본 환경 설정[Y/n] 대문자 기본값 표시, root 암호만 설정

```
[root@localhost ~] # /usr/bin/mysql_secure_installation
```

수업 때 지장이 있어서 mysql 입력 후 진입  
maria> 가 나오면 정상적으로 설치가 된 것  
(대신 비밀번호없이 들어갈 수 있음)

```
NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MariaDB  
SERVERS IN PRODUCTION USE! PLEASE READ EACH STEP CAREFULLY!
```

```
In order to log into MariaDB to secure it, we'll need the current  
password for the root user. If you've just installed MariaDB, and  
you haven't set the root password yet, the password will be blank,  
so you should just press enter here.
```

컨트롤 D 눌러서 진행

```
Enter current password for root (enter for none):
```

```
OK, successfully used password, moving on...
```

비밀번호 가천1234로 설정할 것(수업 공통)

```
Setting the root password ensures that nobody can log into the MariaDB  
root user without the proper authorisation.
```

```
Set root password? [Y/n] Y
```

```
New password:
```

```
Re-enter new password:
```

```
Password updated successfully!
```

```
Reloading privilege tables..
```

```
... Success!
```



# 1. LAMP(Linux, Apache, MySQL, PHP)

- **httpd(웹서버: Apache)와 MariaDB (데이터베이스), PHP(웹프로그램)설치**
- **Mariadb 기본 환경 설정**

```
By default, a MariaDB installation has an anonymous user, allowing anyone
to log into MariaDB without having to have a user account created for
them. This is intended only for testing, and to make the installation
go a bit smoother. You should remove them before moving into a
production environment.
```

```
Remove anonymous users? [Y/n] Y
... Success!
```

```
Normally, root should only be allowed to connect from 'localhost'. This
ensures that someone cannot guess at the root password from the network.
```

```
Disallow root login remotely? [Y/n] Y
... Success!
```

```
By default, MariaDB comes with a database named 'test' that anyone can
access. This is also intended only for testing, and should be removed
before moving into a production environment.
```

```
Remove test database and access to it? [Y/n] Y
- Dropping test database...
... Success!
- Removing privileges on test database...
... Success!
```

# 1. LAMP(Linux, Apache, MySQL, PHP)

- httpd(웹 서버: Apache)와 MariaDB (데이터베이스), PHP(웹 프로그램)설치
- Mariadb 기본 환경 설정

```
Reloading the privilege tables will ensure that all changes made so far  
will take effect immediately.
```

```
Reload privilege tables now? [Y/n] ☒ Y  
... Success!
```

```
Cleaning up...
```

```
All done!  If you've completed all of the above steps, your MariaDB  
installation should now be secure.
```

```
Thanks for using MariaDB!  
[root@localhost ~]# _
```

# 1. LAMP(Linux, Apache, MySQL, PHP)

■ httpd(웹 서버: Apache)와 MariaDB (데이터베이스), PHP(웹 프로그램)설치

■ Php와 mysql 연동 모듈 설치

```
[root@localhost ~]# yum install php-mysql
```

■ Mariadb 기본 언어 set 설정(추가 2줄)

```
[root@localhost ~]# vi /etc/my.cnf
```

이거 안하면 한글이 깨질 수 있다?

```
1 [mysqld]
2 datadir=/var/lib/mysql
3 socket=/var/lib/mysql/mysql.sock
4
5 character-set-server = utf8mb4
6 collation-server = utf8mb4_unicode_ci
7
8 # Disabling symbolic-links is recommended to prevent assorted security risks
9 symbolic-links=0
10 # Settings user and group are ignored when systemd is used.
11 # If you need to run mysqld under a different user or group,
12 # customize your systemd unit file for mariadb according to the
13 # instructions in http://fedoraproject.org/wiki/Systemd
```

# 1. LAMP(Linux, Apache, MySQL, PHP)

■ **httpd(웹 서버: Apache)와 MariaDB (데이터베이스), PHP(웹 프로그램)설치**

■ **Mariadb 변경사항 적용 및 확인**

```
[root@localhost ~]# systemctl httpd restart
```

systemctl restart httpd

```
[root@localhost ~]# systemctl status mariadb
```

● mariadb.service - MariaDB database server

Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; vendor preset: disabled)

Active: **active (running)** since 목 2019-05-23 15:08:16 KST; 1min 39s ago

Process: 20982 ExecStartPost=/usr/libexec/mariadb-wait-ready \$MAINPID (code=exited, status=0/SUCCESS)

Process: 20948 ExecStartPre=/usr/libexec/mariadb-prepare-db-dir %n (code=exited, status=0/SUCCESS)

Main PID: 20981 (mysqld\_safe)

Tasks: 20

CGroup: /system.slice/mariadb.service

└─20981 /bin/sh /usr/bin/mysqld\_safe --basedir=/usr

└─21167 /usr/libexec/mysqld --basedir=/usr --datadir=/var/lib/mysql --plugin-dir=/usr/...

# 1. LAMP(Linux, Apache, MySQL, PHP)

■ httpd(웹 서버: Apache)와 MariaDB (데이터베이스), PHP(웹 프로그램)설치

■ 웹사이트 Apache 환경설정 파일 수정 (포트 확인)

```
[root@localhost ~]# vi /etc/httpd/conf/httpd.conf
```

수정할 건 일단 없고 80번 포트인지 확인만 할 것

■ 웹사이트 Apache 설정 구문 및 동작 테스트

```
[root@localhost ~]# service httpd configtest
AH00558: httpd: Could not reliably determine the server's fully qualified domain name,
st.localdomain. Set the 'ServerName' directive globally to suppress this message
Syntax OK
```

# 1. LAMP(Linux, Apache, MySQL, PHP)

- **httpd(웹 서버: Apache)와 MariaDB (데이터베이스), PHP(웹 프로그램)설치**

- **부팅시 자동으로 httpd와 mariadb 자동 실행**

```
[root@localhost ~]# systemctl enable httpd
[root@localhost ~]# systemctl enable mariadb
```

- **확인하기**

```
[root@localhost ~]# systemctl list-unit-files | grep httpd
httpd.service                                enabled
[root@localhost ~]#
[root@localhost ~]# systemctl list-unit-files | grep mariadb
mariadb.service                             enabled
[root@localhost ~]#
```

- **보안강화 모듈 해제하기(두 명령 중 하나만)**

```
[root@localhost ~]# setenforce 0
setenforce: SELinux is disabled
[root@localhost ~]# _
[root@localhost ~]# vi /etc/sysconfig/selinux
```

# 1. LAMP(Linux, Apache, MySQL, PHP)

■ 웹 서버에 root directory 인 /var/www/html 에 웹페이지 작성 후 테스트

■ # vi /var/www/html/ index.html

```
<html>
<head>
<title> HTML TEST </title>
</head>
<body>
<h1> Welcome to GACHON UNIVERSITY WEB SERVER </h1>
<h2> Dept. of Computer Engineering </h2>
</body>
</html>
```

~

# 1. LAMP(Linux, Apache, MySQL, PHP)

- 웹 서버에 root directory 인 /var/www/html 에 웹페이지 작성 후 테스트



**Welcome to GACHON UNIVERSITY WEB SERVER**

Dept.of Computer Engineering

방화벽

```
firewall-cmd - -permanent --add-port = 80/tcp  
firewall-cmd - -reload
```



# 1. LAMP(Linux, Apache, MySQL, PHP)

## ■ 일반 사용자 홈페이지 게시판 환경 설정

```
[root@localhost ~]# vi /etc/httpd/conf.d/userdir.conf

17      # UserDir disabled
18
19      #
20      # To enable requests to /~user/ to serve the user's public_html
21      # directory, remove the "UserDir disabled" line above, and uncomment
22      # the following line instead:
23      #
24      UserDir public_html
25 </IfModule>
26
27 #
28 # Control access to UserDir directories.  The following is an example
29 # for a site where these directories are restricted to read-only.
30 #
31 <Directory "/home/*/public_html">
32     AllowOverride All
33     Options None
34     # AllowOverride FileInfo AuthConfig Limit Indexes
35     # Options MultiViews Indexes SymLinksIfOwnerMatch IncludesNoExec
36     Require method GET POST OPTIONS
37 </Directory>
```

# 1. LAMP(Linux, Apache, MySQL, PHP)

- httpd(웹서버: Apache)와 MariaDB (데이터베이스), PHP(웹프로그램)설치
- 일반 사용자 계정 홈 디렉토리 설정하기(예: 일반 사용자 guser)

```
[root@localhost ~]# su - guser
[guser@localhost ~]$
[guser@localhost ~]$ pwd
/home/guser
[guser@localhost ~]$
[guser@localhost ~]$ mkdir public_html
[guser@localhost ~]$ chmod 711 /home/guser
[guser@localhost ~]$ chmod 755 /home/guser/public_html
[guser@localhost ~]$ vi ./public_html/index.html
```

```
[guser@localhost ~]$ systemctl httpd restart
```

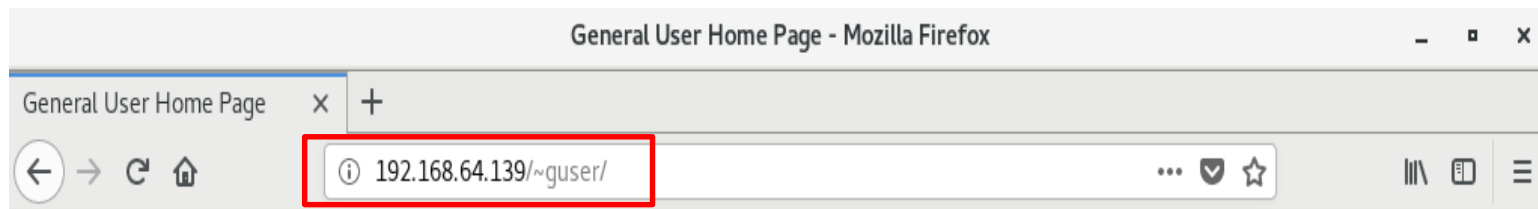
root pw 입력

```
[guser@localhost ~]$ vi ./public_html/index.html
```

```
1 <html>
2 <head>
3 <title> General User Home Page </title>
4 </head>
5 <body>
6 <h1> General User Home Page </h1>
7 </body>
8 </html>
```

# 1. LAMP(Linux, Apache, MySQL, PHP)

- httpd(웹서버: Apache)와 MariaDB (데이터베이스), PHP(웹프로그램)설치
- 일반 사용자 계정 홈 디렉토리 설정하기(예: 일반 사용자 guser)
- 접속하기

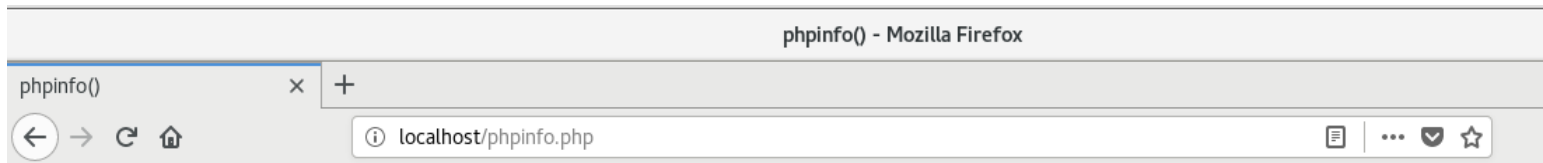


# 1. LAMP(Linux, Apache, MySQL, PHP)

## ■ PHP test page 작성(/var/www/html)

```
root@localhost html # vi phpinfo.php
<?php  phpinfo();  ?>
```

## ■ PHP page 접속



PHP Version 5.4.16



System	Linux localhost.localdomain 3.10.0-957.10.1.el7.x86_64 #1 SMP Mon Mar 18 15:06:45 UTC 2019 x86_64
Build Date	Oct 30 2018 19:31:42
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc
Loaded Configuration File	/etc/php.ini
Scan this dir for additional .ini files	/etc/php.d
Additional .ini files parsed	/etc/php.d/curl.ini, /etc/php.d/fileinfo.ini, /etc/php.d/json.ini, /etc/php.d/mbstring.ini, /etc/php.d/mysql.ini, /etc/php.d/mysqli.ini, /etc/php.d/pdo.ini, /etc/php.d/pdo_mysql.ini, /etc/php.d/pdo_sqlite.ini, /etc/php.d/phar.ini, /etc/php.d/sqlite3.ini, /etc/php.d/sockets.ini

# 1. LAMP(Linux, Apache, MySQL, PHP)

## ■ MariaDB 접속

```
[root@localhost ~]# mysql
ERROR 1045 (28000): Access denied for user 'root'@localhost' (using password: NO)
[root@localhost ~]# mysql -u root -p
Enter password:
Welcome to the MariaDB monitor.  Commands end with ; or \q.
Your MariaDB connection id is 3
Server version: 5.5.60-MariaDB MariaDB Server

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> exit
Bye
[root@localhost ~]#
```

## ■ Mariadb 환경 설정 시 root 암호로 접속해야 함.(pw : gachon1234)

## 2. 데이터베이스 개념

### ■ 관계형 데이터베이스의 기본 개념

- 데이터베이스: 서로 관련성을 가진 데이터들을 데이터 간의 중복을 최소화해서 체계적으로 모아놓은 것
- 관계형 데이터베이스는 데이터를 테이블(표)로 표현

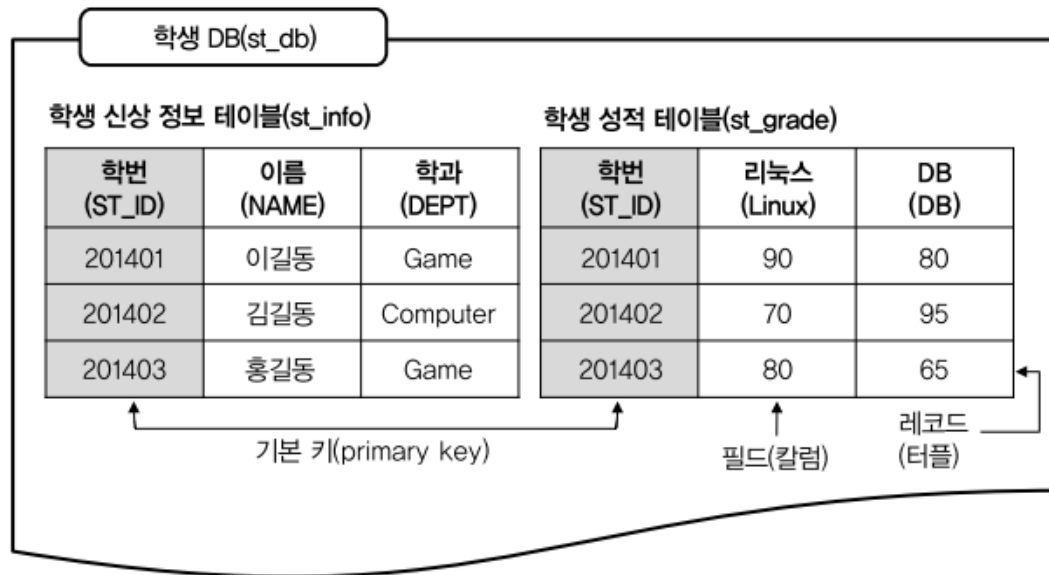
### ■ 관계형 데이터베이스 관련 용어

- 데이터 : 각 항목에 저장되는 값이다.
- 테이블 : 데이터를 체계화하여 행과 열의 형태로 구성한 것으로 테이블 이름을 붙인다.
- 데이터베이스 : 관련 있는 데이터를 체계적으로 저장한 것으로 데이터는 테이블 형태로 저장된다. 데이터베이스에는 하나 이상의 테이블이 있을 수 있다. 데이터베이스 이름을 붙인다.
- 필드 : 테이블의 열을 의미하며 칼럼이라고도 한다.
- 레코드 : 테이블에 저장한 한 행의 정보를 의미하며 튜플(tuple)이라고도 한다. 한 행에는 여러 필드의 값이 저장될 수 있다.
- 키 : 레코드를 구성하는 필드에서 각 레코드를 구분할 수 있는 필드 값을 뜻한다. 예를 들어 학생 데이터를 구성하는 레코드라면 학번 필드가 키가 될 수 있다. 키로 선택된 필드의 값은 중복될 수 없다. 키는 기본 키(primary key)와 그 외 다른 키들이 있을 수 있다.

## 2. 데이터베이스 개념

### ■ DBMS(Database Management System)이란?

데이터베이스를 관리하기 위한 소프트웨어 또는 DB 서버(데이터베이스 서버)를 말한다.  
Mysql과 test DB는 시스템이 제공하는 DB, st\_db는 사용자가 생성한 DB.



[그림 13-2] 학생 데이터베이스의 예

# 3. SQL 구문

## 1. MySQL의 규칙

- ① SQL 문은 대소문자를 구별하지 않는다.(단, 테이블명은 예외) -예약어는 대문자, 사용자 입력 명령어는 소문자로 표시-가독성을 높이기 위함
- ② 명령의 끝에는 반듯이 ; 넣어준다.

## 2. DB와 관련된 SQL문

- ① DB이름 조회 > SHOW DATABASES ;
- ② 사용할 DB 지정 > USE db명 ;
- ③ DB생성 > CREATE DATABASE db명 ;
- ④ DB 삭제 > DROP DATABASE db명 ;

## 3. 테이블 관련 SQL문

- ① 테이블 이름 조회 > SHOW TABLES ;
- ② 테이블 구조(형태)조회 > EXPLAIN 테이블명 ; 또는 DESC 테이블명 ;
- ③ 테이블 생성 > CREATE TABLE 테이블명(필드명1 필드타입1, ...)
- ④ 테이블 삭제 > DROP TABLE 테이블명 ;



## 3. SQL 구문

---

### 4. 사용자 생성과 권한 설정

```
MariaDB[None]> CREATE USER KIM IDENTIFIED BY 'gachon1234'  
                DEFAULT TABLESPACE users  
                TEMPORARY TABLESPACE temp;
```

```
MariaDB[None]> GRANT CONNECT, RESOYRCE, CREATE SESSION,  
                CREATE VIEW TO KIM;
```

```
MariaDB[None]> FLUSH PRIVILEGES;
```

# 3. SQL 구문

---

## 5. 현재 사용자 확인

```
MariaDB[None]> use mysql;
```

```
MariaDB[mysql]> SELECT user, host FROM user ;
```

## 6. 사용자 삭제 (kim)

```
MariaDB[None]> DELETE FROM user WHERE user='kim' ;
```

```
MariaDB[None]> FLUSH PRIVILEGES;
```

```
MariaDB[None]> FLUSH PRIVILEGES;
```

# 3. SQL 구문

## 6. 테이블 관련 SQL문

테이블 수정 <sup>alternate</sup> >ALTER TABLE 테이블명 옵션 ;

ALTER TABLE customer MODIFY name CHAR(20) ;

ALTER TABLE customer CHANGE name fullname CHAR(20) ;

ALTER TABLE customer ADD phone VARCHAR(20) after name ;

ALTER TABLE customer DROP age ;

## 7. 테이블 각 필드의 자료형

- ① VARCHAR(n) – 최대 n개의 크기를 갖는 가변길이 문자열-주소 등
- ② CHAR(n) – n개의 문자가 저장되는 고정길이 문자열-주민번호, 학번 등
- ③ INT – 정수형 숫자
- ④ FLOAT – 실수형 숫자
- ⑤ DATE – 날짜형 데이터
- ⑥ TIME – 시간형 데이터

# 3. SQL 구문

## 8. 레코드 삽입/삭제/수정 관련 SQL문 DML(Data Manipulation Language)

- ① 레코드 삽입 > INSERT INTO customer VALUES('hong','홍길동',22, '경기');
- ② 레코드 삭제 > DELETE FROM customer WHERE id='hong';
- ③ 레코드 수정 > UPDATE customer age=25 WHERE id='hong';

## 9.테이블 조회

- ① SELECT \* FROM customer ;
- ② SELECT id, name FROM customer ;
- ③ SELECT id, name FROM customer WHERE id='park'
- ④ SELECT id, name FROM customer WHERE age > 21 ;

- 1 DB만들기
- 2 Table 생성(업무분석) - 중요
- 3 튜플 삽입
- 4 조회, 검색, 연산 등

## 4. SQL 실습

### ■ SQL의 기초

- **SQL(structured query language)** 구조적 질의어
- 관계형 데이터베이스를 생성, 테이블 생성, 데이터 입력/삭제/수정 등

### ■ 데이터베이스 관련 SQL

- ① 데이터베이스 보기 : 기존에 있는 데이터베이스의 목록을 출력

```
show databases;
```

- ② 데이터베이스 생성 : 새로운 데이터베이스를 생성

```
create database 데이터베이스 이름;
```

```
create database st_db;
```

- ③ 데이터베이스 삭제 : 지정한 데이터베이스를 삭제

```
drop database 데이터베이스 이름;
```

- ④ 사용할 데이터베이스 지정

```
use 데이터베이스 이름;
```

## 4. SQL 실습

### ■ 테이블 관련 SQL

- ① 테이블 보기 : 현재 데이터베이스에 있는 테이블의 목록을 출력

```
show tables;
```

- ② 테이블 생성 : 새로운 테이블을 생성

[표 13-1] 테이블 필드의 자료형

자료형	설명	예
varchar(n)	최대 n개의 크기를 가진 가변 문자열	varchar(10)
char(n)	n개의 크기를 가진 고정 문자열	char(5)
int	정수형 숫자	
float	실수형 숫자	
date	날짜	
time	시간	

```
create table 테이블 이름(필드명 1 필드 자료형 1, 필드명 2 필드 자료형 2, ...);
```

```
create table st_info (ST_ID int, NAME varchar(20), DEPT varchar(25));
```

## 4. SQL 실습

### ■ 테이블 관련 SQL

#### ③ 테이블 구조 보기 : 지정한 테이블의 구조를 출력

```
explain 테이블 이름;
```

#### ④ explain 테이블 이름;

```
alter table 테이블 이름 수정 명령;
```

- 테이블에 필드 추가 : **alter table** 테이블 이름 **add** 필드명 자료형
  - 예 : alter table st\_info add AGE int;
- 필드의 자료형 변경 : **alter table** 테이블 이름 **modify** 필드명 자료형
  - 예 : alter table st\_info modify AGE float;
- 필드 삭제 : **alter table** 테이블 이름 **drop column** 필드명
  - 예 : alter table st\_info drop column AGE;
- 기본 키 추가 : **alter table** 테이블 이름 **add constraint** 제약 조건명 **primary key** (필드명)
  - 예 : alter table st\_info add constraint pk\_stinfo primary key (ST\_ID);
- 제약 조건명은 사용자가 정하면 된다. 처음 테이블을 생성하면서 기본 키를 지정할 때 다음과 같이 한다.
  - 예 : create table st\_info (ST\_ID int Not NULL primary key, NAME varchar(20), DEPT varchar(25));

#### ⑤ 테이블 삭제 : 지정한 테이블을 삭제

```
drop table 테이블 이름;
```

## 4. SQL 실습

### ■ 레코드 삽입·삭제·수정 관련 SQL

#### ① 레코드 삽입 : 테이블에 새로운 레코드를 추가

```
insert into 테이블 이름 values (값 1, 값 2, ...);
```

- 예 : `insert into st_info values (2010401, '이길동', 'Game');`

#### ② 레코드 수정 : 기존 레코드를 수정

```
update 테이블 이름 set 필드명 1=수정할 값 1, 필드명 2=수정할 값 2, ... where 조건;
```

- 예 : `update st_info set DEPT='Computer' where ST_ID=201403;`

#### ③ 레코드 삭제 : 테이블에서 해당 레코드를 삭제

```
delete from 테이블 이름 where 조건;
```

- 예 : `delete from st_info where ST_ID=201403;`



## 4. SQL 실습

### ■ 레코드 검색하기

```
select 필드명 1, 필드명 2, ... from 테이블 이름 where 조건;
```

- `select * from st_info;`
- `select NAME from st_info where ST_ID=201401;`
- `select Linux, DB from st_grade where ST_ID=201401;`
- `select st_info.NAME, st_grade, Linux`

JOIN 연산

```
from st_info, st_grade  
where st_info.ST_ID=201401 and st_grade.ST_ID=201401;
```

### ■ 접근 권한 부여하기

- 시스템 관리자가 데이터베이스를 만들고 일반 사용자들이 사용하도록 설정해야 할 때 `grant` 문을 사용

```
grant all privileges on DB명.* to 사용자 계정@localhost identified by '패스워드';
```

- 예 : `grant all privileges on st_db.* to user1@localhost identified by '123456';`

## 5. MariaDB 설치 및 운영

- 오픈 소스 기반의 데이터베이스로 MySQL이 오랫동안 사용되어 왔다.

그러나 MySQL이 Sun사에 인수되고, 다시 Sun사가 Oracle 사에 인수되자 오라클 사의 라이선스 정책에 동조하지 않는 주요 핵심 개발자들이 2009년도에 퇴사해서 별도의 데이터베이스 도구를 개발했는데, 이것이 MariaDB이다.

그러므로 MariaDB는 MySQL과 거의 동일한 제품으로 보면 된다.

- 현재 CentOS를 비롯한 많은 리눅스 진영에서 MySQL 대신에 MariaDB를 기본 DBMS로 제공한다.

## 5. MariaDB 설치 및 운영

- 데이터베이스 목록 확인 : SHOW DATABASES;

```
MariaDB [(none)]> SHOW DATABASES;
```

```
+-----+  
| Database |  
+-----+  
| information_schema |  
| mysql |  
| performance_schema |  
+-----+  
3 rows in set (0.01 sec)
```

```
MariaDB [(none)]>
```

## 5. MariaDB 설치 및 운영

### ■ 데이터베이스 생성하기 : CREATE DATABASE st\_db ;

```
MariaDB [(none)]> CREATE DATABASE st_db;
Query OK, 1 row affected (0.00 sec)

MariaDB [(none)]> SHOW DATABASES;
+-----+
| Database |
+-----+
| information_schema |
| mysql |
| performance_schema |
| st_db |
+-----+
4 rows in set (0.00 sec)

MariaDB [(none)]>
```

### ■ 데이터베이스 사용하기 : USE db명;

```
MariaDB [(none)]> USE st_db;
Database changed
MariaDB [st_db]>
```

### ■ 테이블 확인하기 : SHOW TABLES ;

```
MariaDB [st_db]> SHOW TABLES;
Empty set (0.00 sec)
```

## 5. MariaDB 설치 및 운영

- 테이블 생성하기 : CREATE TABLE 테이블명( 필드명1 데이터형, ...;

```
MariaDB [st_db] > CREATE TABLE st_info(ST_ID int, NAME varchar(20), DEPT varchar(25)) DEFAULT charset=utf8;  
Query OK, 0 rows affected (0.01 sec)
```

```
MariaDB [st_db] > CREATE TABLE st_grad(ST_ID int, LINUX int, DB int);  
Query OK, 0 rows affected (0.01 sec)
```

```
MariaDB [st_db] > SHOW TABLES;
```

```
+-----+  
| Tables_in_st_db |  
+-----+  
| st_grad          |  
| st_info          |  
+-----+  
2 rows in set (0.00 sec)
```

```
MariaDB [st_db] > █
```

## 5. MariaDB 설치 및 운영

- 테이블 구조 확인하기 : EXPLAIN st\_info ;
- 테이블 구조 확인하기 : EXPLAIN st\_grad ;

```
MariaDB [st_db] > EXPLAIN st_info;
```

Field	Type	Null	Key	Default	Extra
ST_ID	int(11)	YES		NULL	
NAME	varchar(20)	YES		NULL	
DEPT	varchar(25)	YES		NULL	

3 rows in set (0.03 sec)

```
MariaDB [st_db] > EXPLAIN st_grad;
```

Field	Type	Null	Key	Default	Extra
ST_ID	int(11)	YES		NULL	
LINUX	int(11)	YES		NULL	
DB	int(11)	YES		NULL	

3 rows in set (0.00 sec)

## 5. MariaDB 설치 및 운영

- 기본 키 Not Null 설정하기 : ST\_ID int Not Null
- 기본 키 지정하기 : ST\_ID 기본 키 (Primary key)지정

```
MariaDB [st_db]> ALTER TABLE st_info MODIFY st_id INT NOT NULL;  
Query OK, 0 rows affected (0.03 sec)  
Records: 0 Duplicates: 0 Warnings: 0
```

```
MariaDB [st_db]> ALTER TABLE st_info MODIFY COLUMN ST_ID int PRIMARY KEY;  
Query OK, 0 rows affected (0.16 sec)  
Records: 0 Duplicates: 0 Warnings: 0
```

## 5. MariaDB 설치 및 운영

- 기본 키 Not Null , 기본 키 지정 확인 하기 : EXPLAIN st\_info;

```
MariaDB [st_db]> EXPLAIN st_info;
```

Field	Type	Null	Key	Default	Extra
ST_ID	int(11)	NO	PRI	NULL	
NAME	varchar(20)	YES		NULL	
DEPT	varchar(25)	YES		NULL	

```
3 rows in set (0.01 sec)
```

```
MariaDB [st_db]>
```



## 5. MariaDB 설치 및 운영

■ 레코드 입력하기 ; INSERT INTO 테이블명 VALUES ( 필드값1, ... )

```
MariaDB [st_db]>  
MariaDB [st_db]> INSERT INTO st_info VALUES(201901, '김을순', 'Game');  
Query OK, 1 row affected (0.00 sec)
```

```
MariaDB [st_db]> INSERT INTO st_info VALUES(201904, '강성구', 'Computer');  
Query OK, 1 row affected (0.04 sec)
```

```
MariaDB [st_db]> INSERT INTO st_info VALUES(201902, '김관용', 'Game');  
Query OK, 1 row affected (0.00 sec)
```

```
MariaDB [st_db]> INSERT INTO st_grad VALUES(201901, 90, 85);  
Query OK, 1 row affected (0.00 sec)
```

```
MariaDB [st_db]> INSERT INTO st_grad VALUES(201904, 70, 95);  
Query OK, 1 row affected (0.00 sec)
```

```
MariaDB [st_db]> INSERT INTO st_grad VALUES(201902, 80, 65);  
Query OK, 1 row affected (0.00 sec)
```

## 5. MariaDB 설치 및 운영

■ 레코드 검색하기 ; SELECT 필드명 FROM 테이블명 WHERE 조건절

```
MariaDB [st_db] > SELECT * FROM st_info;
```

ST_ID	NAME	DEPT
201901	김을순	Game
201902	김관용	Game
201904	강성구	Computer

3 rows in set (0.00 sec)

```
MariaDB [st_db] > SELECT NAME, DEPT FROM st_info WHERE ST_ID=201901;
```

NAME	DEPT
김을순	Game

1 row in set (0.00 sec)

## 5. MariaDB 설치 및 운영

■ 레코드 검색하기 ; SELECT 필드명 FROM 테이블명 WHERE 조건절

■ 학번이 201901인 학생의 리눅스 성적을 검색하기

```
MariaDB [st_db] > SELECT LINUX FROM st_grad WHERE ST_ID=201901;
+-----+
| LINUX |
+-----+
|    90 |
+-----+
1 row in set (0.00 sec)
```

■ 학번이 201901인 학생의 이름과 학과, DB성적을 한번에 검색하기

```
MariaDB [st_db] > SELECT st_info.DEPT, st_info.DEPT, st_grad.DB
-> FROM st_info, st_grad
-> WHERE st_info.ST_ID=201901 AND st_grad.ST_ID=201901;
+-----+-----+-----+
| DEPT | DEPT | DB   |
+-----+-----+-----+
| Game | Game | 85   |
+-----+-----+-----+
1 row in set (0.00 sec)

MariaDB [st_db] >
```

## 5. MariaDB 설치 및 운영

- 레코드 수정하기 ; UPDATE 테이블명 SET 필드명=값 WHERE 조건절 ;
- 학번이 201901인 학생의 DB 성적을 85점에서 90점으로 수정

```
MariaDB [st_db] > SELECT * FROM st_grad;
```

ST_ID	LINUX	DB
201901	90	85
201904	70	95
201902	80	65

3 rows in set (0.01 sec)

```
MariaDB [st_db] > UPDATE st_grad SET DB=90 WHERE ST_ID=201901;  
Query OK, 1 row affected (0.01 sec)  
Rows matched: 1 Changed: 1 Warnings: 0
```

```
MariaDB [st_db] > SELECT * FROM st_grad WHERE ST_ID=201901;
```

ST_ID	LINUX	DB
201901	90	90

1 row in set (0.00 sec)

## 5. MariaDB 설치 및 운영

### ■ `mysqladmin status`

#### ■ 상태 정보로 출력되는 주요 항목

- Uptime : 서버가 동작한 시간을 초 단위로 표시
- Threads : 현재 동작 중인 MariaDB 서버 스레드 수
- Questions : 서버가 동작한 이후 처리한 질의 수
- Slow queries : 일정 시간보다 처리 시간이 길어진 질의 수
- Opens : 서버가 열었던 테이블 수
- Flush : flush, refresh, reload 명령을 수행한 횟수
- Open tables : 현재 열려 있는 테이블 수

```
[root@localhost ~]# mysqladmin -u root -p status
```

```
Enter password:
```

```
Uptime: 76430 Threads: 1 Questions: 41 Slow queries: 0 Opens: 5 Flush tables: 2 Open tables: 2
```

```
8 Queries per second avg: 0.000
```

```
[root@localhost ~]#
```

## 5. MariaDB 설치 및 운영

- mysqladmin version (root pw 설정 했음)

```
[root@localhost ~]# mysqladmin -u root -p version
```

```
Enter password:
```

```
mysqladmin Ver 9.0 Distrib 5.5.60-MariaDB, for Linux on x86_64  
Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.
```

```
Server version      5.5.60-MariaDB  
Protocol version    10  
Connection          Localhost via UNIX socket  
UNIX socket         /var/lib/mysql/mysql.sock  
Uptime:             21 hours 14 min 25 sec
```

```
Threads: 1 Questions: 42 Slow queries: 0 Opens: 5 Flush tables: 2 Open tables: 28 Queries per  
second avg: 0.000
```

## 6. Apache web server 설치 및 운영

- Apache라는 이름은 패치(patch)를 포함하고 있다는 뜻 또는 미국의 인디언의 생명력을 닮은 웹 서버라는 뜻이다.
- 최초의 웹 서버는 스위스 제네바에 있는 CERN(European Center for Nuclear Research)에 근무하던 Tim Berners-Lee가 만들었으며 아파치의 조상이라 할 수 있는 웹 서버는 미국정부 산하 기관인 NCSA(National Center for Supercomputing Agencies)에서 만들어졌으며 미국민의 세금으로 운영되는 기관이기 때문에 소스코드(C언어)가 공개되어 배포되었다.
- NCSA발표 후 NCSA보다 훨씬 좋은 기능을 지닌 웹 서버를 만들 수 있다고 생각하는 사람들이 NCSA 1.3버전(1995년 초)의 소스코드와 아이디어를 바탕으로 만들었다.
- Web programming-web server programming(PHP, ASP,JSP 등)-Server side Script 기능 제공하는 서버

## 6. Apache web server 설치 및 운영

- Apache 웹서버는 HTTP(Hypertext Transfer Protocol), Hypertext, Hypermedia를 전송하기 위한 프로토콜로 FTP, SMTP, NNTP 등과 비슷한 종류의 프로토콜이다.
- HTTP는 일반적으로 Request(요청)과 Response(응답)형식으로 처리된다.
- Request처리는 server 쪽의 daemon이 하고, 이 daemon이 httpd(http daemon)이다. 반대로 Apache의 response(응답)을 처리하는 것은 Internet Explorer 또는 firefox 와 같은 web browser이다.
- 아파치 주요 디렉토리
  - conf : 아파치 환경 설정(httpd.conf)
  - Htdocs : 클라이언트들이 실제 보게 되는 웹사이트 HTML 소스
  - logs : 접근기록 및 에러 로그 파일을 보관
  - cgi-bin : CGI 스크립트(클라이언트 측의 요청에 따라 아파치가 실행하는 프로그램 또는 스크립트들)



## 7. Web program PHP 설치 및 운영

- PHP(Personal Home Page)는 HTML에 내장되어 동작하는 스크립트 언어 (HTML-embedded scripting language)이다
- PHP는 C와 Java, Perl 등으로부터 많은 문장형식을 빌려 쓰고 있으며 고유한 문장을 탑재하고 있다.
- PHP의 목적은 웹 개발자들이 동적인 웹 문서를 빠르고 쉽게 작성할 수 있도록 하는데 초점을 맞추어 개발되었다.
- 1994년 가을 라스머스 레도르프가 고안하여 초기버전은 자신의 홈페이지에서 온라인 이력서를 누가 보았는지 기록하는데 사용하였으며 현재는 전세계적으로 4천만 개 이상의 사이트에서 사용되고 있다.
- 현재 버전은 PHP4, 5에서는 Zend 스크립트 엔진을 사용하고 있으며, Apache 이외의 웹 서버에서도 모두 실행될 수 있는 모듈로 개발되어 있다.

# 7. Web program PHP 설치 및 운영

## ➤ PHP의 특징

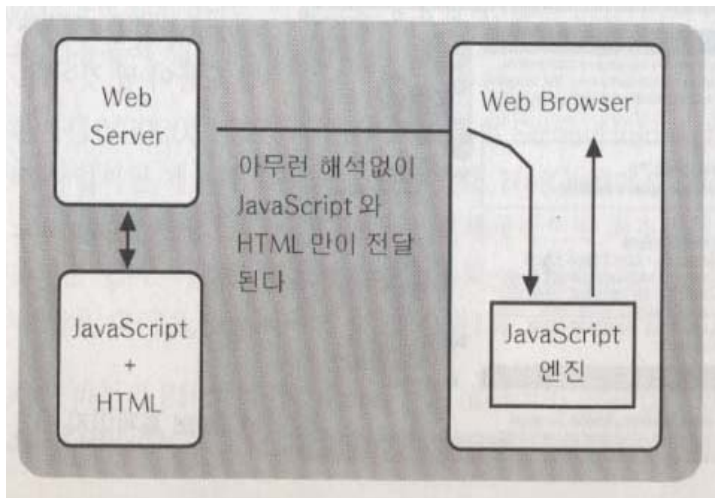
1. Perl이나 C와 같은 CGI 스크립트와 달리 PHP는 HTML을 출력하는데 많은 명령어가 필요 없다. 아래 예와 같이 HTML 내에 특정코드를 적어주면 된다. PHP코드는 특정한 시작과 끝 태그 사이에 들어가게 되는데 이 태그를 통해 HTML모드에서 PHP모드로 들어가거나 나오는 것이다.

```
<html>
  <head>
    <title>예제</title>
  </head>
  <body>
    <?php
    echo "Hi, I'm a PHP script!"; ?>
  </body>
</html>
```

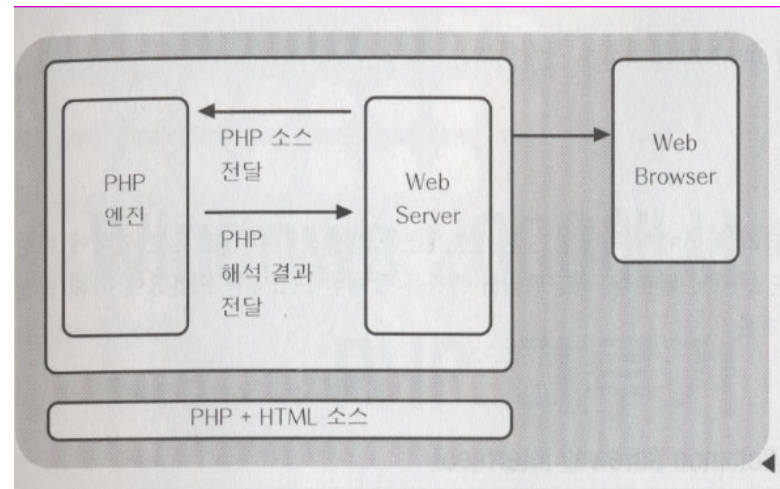
# 7. Web program PHP 설치 및 운영

## ➤ PHP의 특징

2. PHP는 서버에서 해석되는 스크립트 언어이다. PHP는 자바스크립트와 유사한 형태로 자바스크립트가 웹 브라우저를 통해서 해석되는 반면 PHP는 ASP와 같이 Server에서 해석되어 HTML형식으로 보여지는 차이점이 있다.
3. 이와 같이 자바스크립트는 Client Side Program이라고 하며 PHP, ASP는 Sever Side program이라고 한다.



Client Side Program



Sever Side program

# 7. Web program PHP 설치 및 운영

## ➤ PHP의 특징

### 4. PHP는 많은 데이터베이스를 지원한다.

adabas D	Ingres	Oracle
Dbase	InterBase	Ovrimos
Empress	FrontBase	PostgreSQL
Filepro	mSQL	Solid
Hyperwave	MS-SQL	Sybase
IBM DB2	MySQL	Velocis
Informaix	ODBC	Unix dbm

예는 조금 다른데  
DB간의 호환성을 위해 사용하는 것?

## 8. 보고서

다음 용어의 원어와 개념을 조사하세요

1. DBMS란?
2. Data, DataBase, Record, Tuple, Attribute, Domain의 개념을 설명하세요
3. MySQL 서버의 활용 방법과 실제 예를 들어 설명하세요.
4. HTML Tag의 종류와 기능을 설명하세요.
5. 실습 내용을 실행한 후 캡처 하세요
6. MySQL Server의 주요기능과 설치방법을 설명하세요
7. PHP 게시판인 GnuBoard를 설치하고 Web Server(httpd)와 Database server(MariaDB) , 그리고 PHP Programming(웹 프로그램)을 연동하여 자신을 소개하는 홈페이지를 만들고 화면을 캡처하여 첨부 하세요. (화면에 자신의 학번과 이름이 표시 될 것)