

# Maria DB

# Maria DB

## ❖ Maria DB

- ✓ SQL에 기반을 둔 관계형 DBMS로 Open Source 로 제공
- ✓ 리눅스, 유닉스, 윈도우 등 거의 모든 운영체제에서 사용 가능
- ✓ 처리 속도가 빠르고 대용량 데이터 처리 용이
- ✓ 설치 방법이 쉽고 초보자도 익히기 쉬움
- ✓ 작업 단위
  - 데이터베이스 > 테이블
  - 데이터베이스는 사용자 와 상관없이 생성되며 사용자에게 권한을 부여해서 사용
  - 데이터베이스를 생성하면 사용자 별로 만들어지지 않음
  - 하나의 데이터베이스를 여러 사용자가 공유 가능



# Server 설치

## ❖ Maria DB 서버 설치

### ✓ Windows 설치

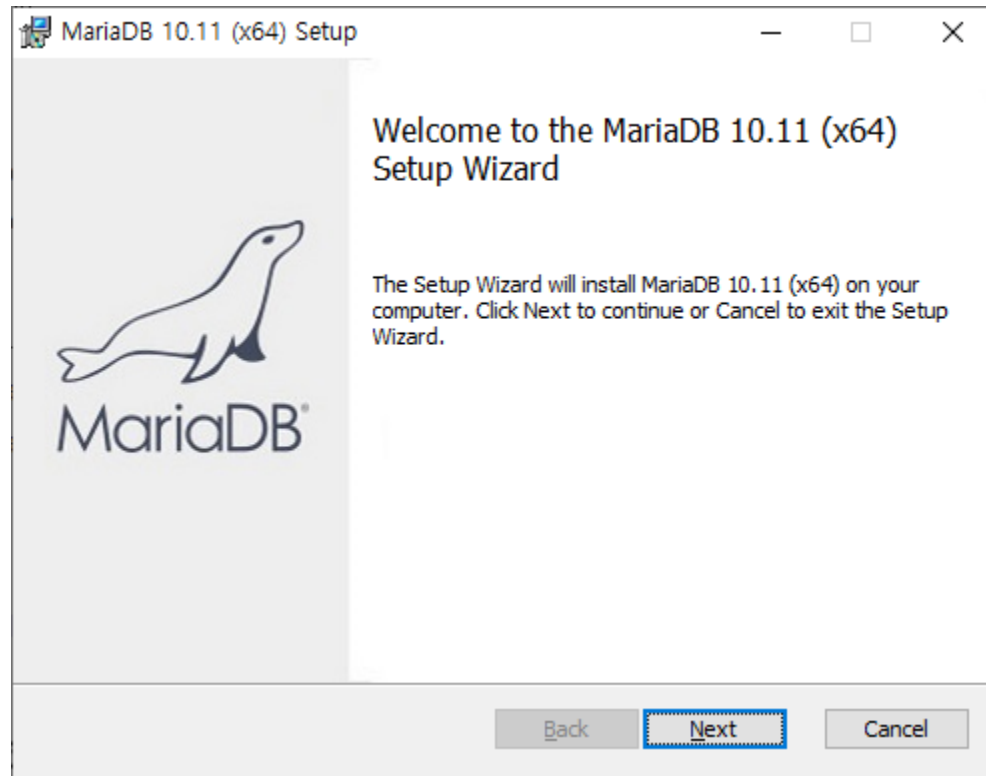
- 다운로드: <https://mariadb.org/download/>



The screenshot shows the MariaDB Foundation website's download page. The browser's address bar displays <https://mariadb.org/download/>. The website header includes the MariaDB Foundation logo, navigation links (Download, Documentation, Contribute, Server Fest, Events, Sponsor, Blog, Planet MariaDB, About), and a search bar. A banner below the header lists the latest releases: 10.11.0 (Alpha), 10.10.1 (RC), 10.9.3, 10.8.5, 10.7.6, 10.6.10, 10.5.17, 10.4.26, and 10.3.36, with a link to vote in a poll. The main content area features a 'WATCH MORE!' YouTube subscription button on the left. The central heading is 'Download MariaDB Server', followed by a paragraph explaining that MariaDB Server is a popular open-source relational database available in Linux repositories or via package managers. Below this, there are tabs for 'MariaDB Server', 'MariaDB Repositories', and 'Connectors'. The 'MariaDB Server' tab is active, showing a 'Download MariaDB Server' link, a 'REST API' link, and a 'Release Schedule' link. A purple banner at the bottom of the page states: 'This is a preview release with all potential upcoming features bundled into one package. Not all these features may make it into the final stable release of 10.11.'

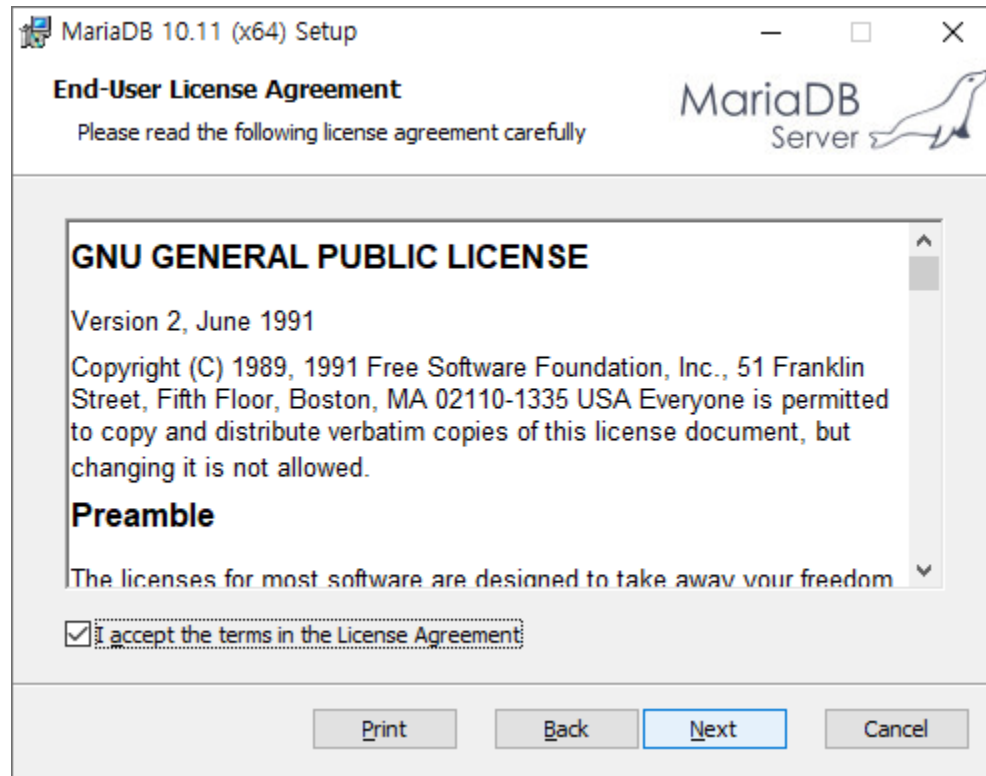
# Server 설치

- ❖ Maria DB 서버 설치
  - ✓ Windows 설치



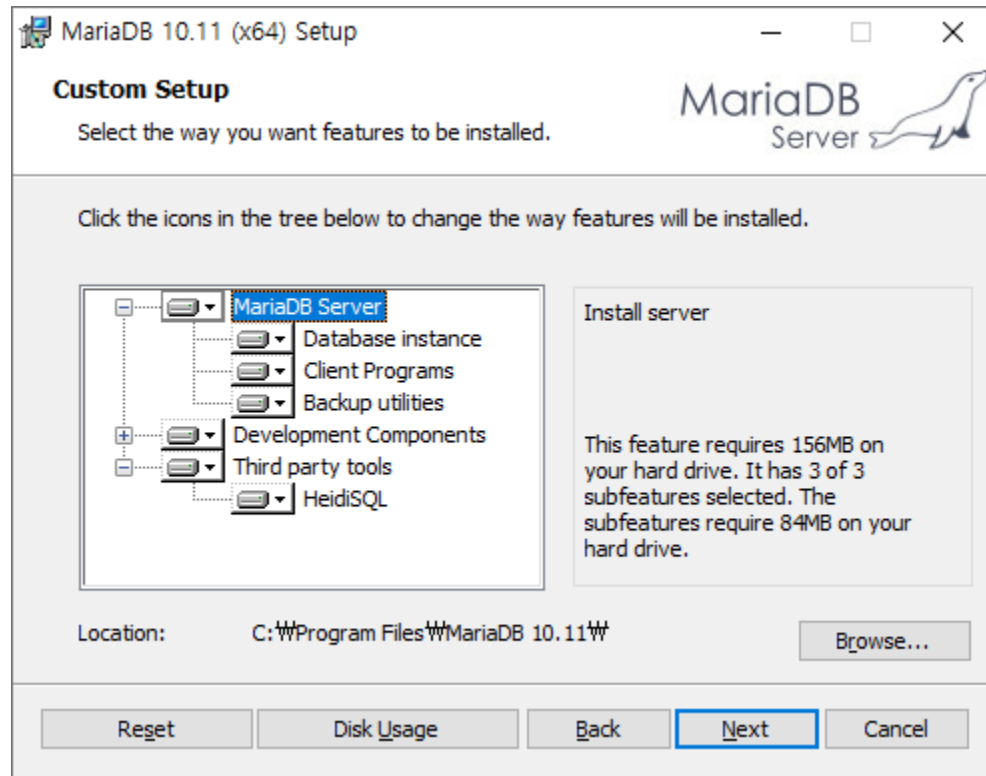
# Server 설치

- ❖ Maria DB 서버 설치
  - ✓ Windows 설치



# Server 설치

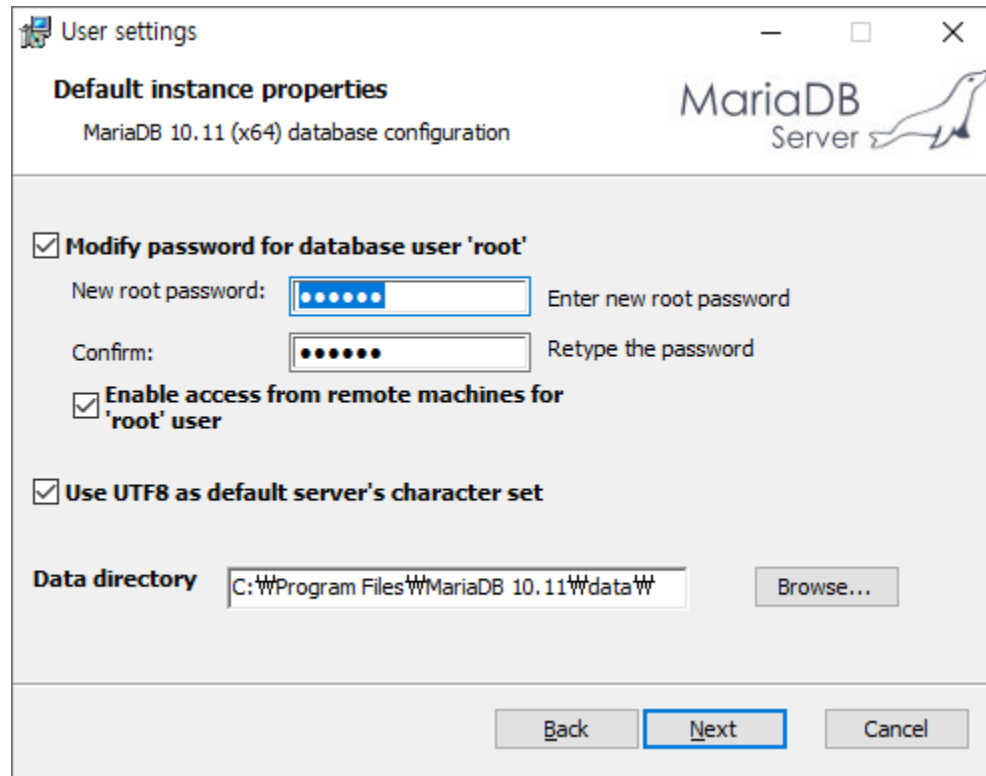
- ❖ Maria DB 서버 설치
  - ✓ Windows 설치



# Server 설치

## ❖ Maria DB 서버 설치

- ✓ Windows 설치: root 비밀번호 설정 및 원격 접속 허용 여부 및 인코딩 설정



The screenshot shows the 'User settings' window for MariaDB 10.11 (x64) database configuration. The window has a title bar with standard Windows window controls and the MariaDB Server logo. The main content area is titled 'Default instance properties' and contains several configuration options:

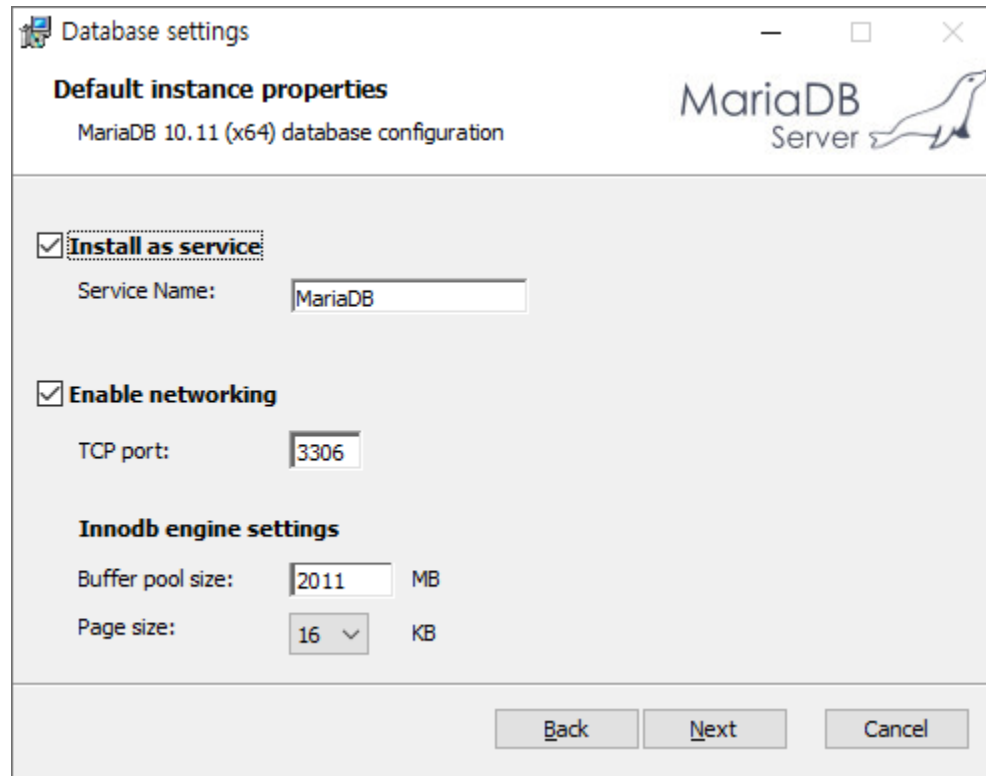
- ☒ **Modify password for database user 'root'**
  - New root password: [password field] Enter new root password
  - Confirm: [password field] Retype the password
- ☒ **Enable access from remote machines for 'root' user**
- ☒ **Use UTF8 as default server's character set**
- Data directory**: C:\Program Files\MariaDB 10.11\data [Browse...]

At the bottom of the window, there are three buttons: 'Back', 'Next' (which is highlighted with a blue border), and 'Cancel'.

# Server 설치

## ❖ Maria DB 서버 설치

- ✓ Windows 설치: 서비스 이름 과 포트번호 및 버퍼 사이즈 설정



The screenshot shows the 'Database settings' window for MariaDB 10.11 (x64) database configuration. The window has a title bar with standard Windows window controls. The main content area is divided into sections. The first section, 'Default instance properties', includes a checkbox for 'Install as service' which is checked, a text field for 'Service Name' containing 'MariaDB', and a checkbox for 'Enable networking' which is also checked. Below this, the 'TCP port' is set to '3306'. The second section, 'InnoDB engine settings', shows the 'Buffer pool size' set to '2011 MB' and the 'Page size' set to '16 KB'. At the bottom of the window are three buttons: 'Back', 'Next', and 'Cancel'.

Database settings

Default instance properties

MariaDB 10.11 (x64) database configuration

☒ Install as service

Service Name:

☒ Enable networking

TCP port:

InnoDB engine settings

Buffer pool size:  MB

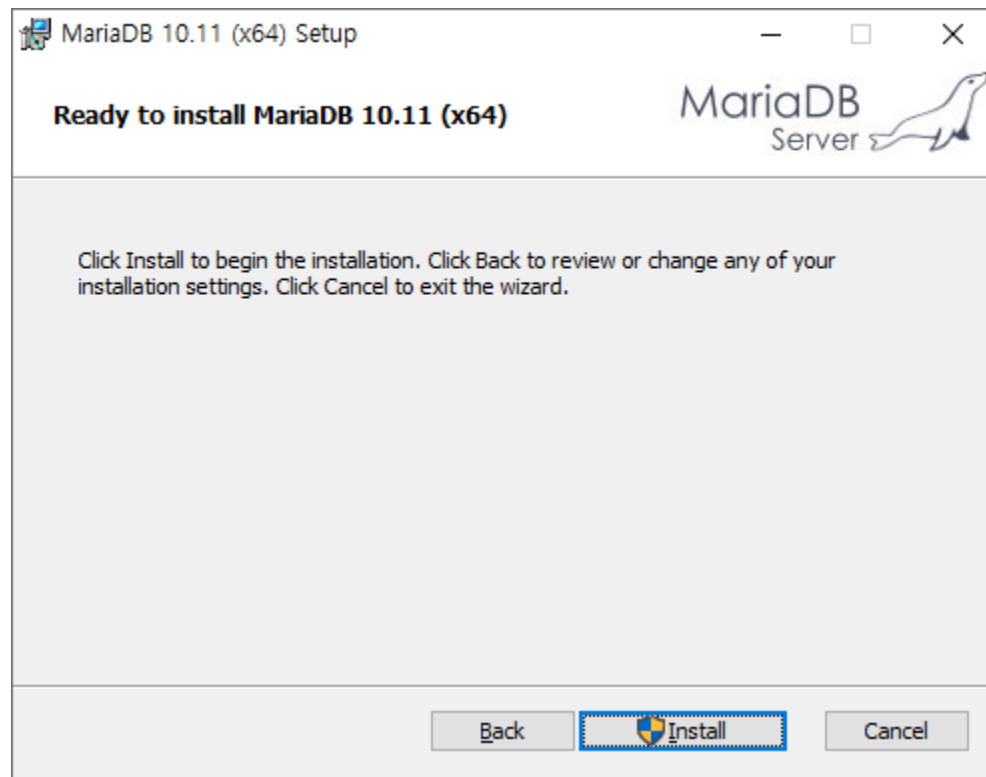
Page size:  KB

Back Next Cancel



# Server 설치

- ❖ Maria DB 서버 설치
  - ✓ Windows 설치: 설치 완료



# Server 설치

## ❖ Maria DB 서버 설치

### ✓ Docker를 이용한 설치

- `docker pull mariadb` 버전(생략하면 최신 버전)
- 컨테이너 생성: `docker run --name mariadb -d -p 외부에서접속할포트번호:MariaDB포트번호 -e MYSQL_ROOT_PASSWORD=비밀번호 컨테이너이름`




# 실습

## ❖ Maria DB 접속

Connection "mariadb" configuration

Connection settings  
MariaDB connection settings



✓ Connection settings  
Initialization  
Shell Commands  
Client identification  
Transactions  
General  
Metadata  
Errors and timeouts  
> Data editor  
> SQL Editor

Main Driver properties SSH Proxy SSL

Server

Server Host: localhost Port: 3306

Database: mysql

Authentication (Database Native)

Username: root

Password: ..... ☒ Save password loc:

Advanced

Server Time Zone: Auto-detect

Local Client:

*i* You can use variables in connection parameters.

Driver name: MariaDB Edit Driver Settings

Test Connection ... 취소 확인

# 실습

## ❖ 데이터베이스 관련 명령

- ✓ 존재하는 데이터베이스 목록보기

show databases;

- ✓ 데이터베이스 생성

create database 데이터베이스이름;

- ✓ 데이터베이스 삭제

drop database 데이터베이스이름;

- ✓ 데이터베이스 사용

use 데이터베이스이름;

- ✓ 테이블 목록 보기

show tables;



# 실습

## ❖ 유저 생성

### ✓ 외부 접속 허용

#### ● Windows 는 아래 작업만 수행

##### ◆ 유저 생성

`create user '사용자계정'@'%' identified by '비밀번호';`

##### ◆ 유저에 권한 부여

`GRANT all privileges on *.* TO '계정'@'%';`

- \*.\* 대신에 데이터베이스 이름을 대입하면 모든 데이터베이스 사용 가능
- % 대신에 ip를 기재하면 특정 ip에서만 접속이 허용되며 localhost로 지정하면 현재 컴퓨터에서만 접속이 가능

##### ◆ 변경 내용 적용

`FLUSH privileges;`



# 실습

## ❖ 유저 생성

### ✓ 외부 접속 허용

- Mac 이나 Linux 에서는 아래 위치의 설정 파일을 수정

#### ◆ 터미널에서 아래 명령 수행

~# `vi /etc/mysql/mariadb.conf.d/50-server.cnf`

- ◆ `bind-address = 127.0.0.1` 부분을 `0.0.0.0` 이나 허용할 IP를 나열



# 실습

## ❖ 유저 생성

### ✓ 외부 접속 허용

#### ● Docker에 설치한 경우

##### ◆ MariaDB가 설치된 도커 컨테이너로 접속

```
$ docker exec -it [mariadb컨테이너이름] bash
```

##### ◆ MariaDB 외부 접근 허용을 위한 설정 파일을 수정

```
# vim /etc/mysql/mariadb.conf.d/50-server.cnf
```

- bind-address 의 값을 0.0.0.0 으로 수정

##### ◆ vi, vim 설치되어 있지 않으면 설치

- # apt update
- # apt upgrade
- # apt install vim



# 실습

## ❖ 유저 생성

### ✓ 유저 생성

create user '사용자계정'@'%' identified by '비밀번호';

### ✓ 유저에 권한 부여

GRANT all privileges on \*.\* TO '계정'@'%';

- \*.\* 대신에 데이터베이스 이름을 대입하면 모든 데이터베이스 사용 가능
- % 대신에 ip를 기재하면 특정 ip에서만 접속이 허용되며 localhost로 지정하면 현재 컴퓨터에서만 접속이 가능

### ✓ 변경 내용 적용

FLUSH privileges;





# 샘플 데이터

## ❖ 샘플 데이터 생성

```
CREATE TABLE tCity
(
    name CHAR(10) PRIMARY KEY,
    area INT NULL ,
    popu INT NULL ,
    metro CHAR(1) NOT NULL,
    region CHAR(6) NOT NULL
);

INSERT INTO tCity VALUES ('서울',605,974,'y','경기');
INSERT INTO tCity VALUES ('부산',765,342,'y','경상');
INSERT INTO tCity VALUES ('오산',42,21,'n','경기');
INSERT INTO tCity VALUES ('청주',940,83,'n','충청');
INSERT INTO tCity VALUES ('전주',205,65,'n','전라');
INSERT INTO tCity VALUES ('순천',910,27,'n','전라');
INSERT INTO tCity VALUES ('춘천',1116,27,'n','강원');
INSERT INTO tCity VALUES ('홍천',1819,7,'n','강원');

SELECT * FROM tCity;
```

# 샘플 데이터

## ❖ 샘플 데이터 생성

```
CREATE TABLE tStaff  
(  
    name CHAR (15) PRIMARY KEY,  
    depart CHAR (10) NOT NULL,  
    gender CHAR(3) NOT NULL,  
    joindate DATE NOT NULL,  
    grade CHAR(10) NOT NULL,  
    salary INT NOT NULL,  
    score DECIMAL(5,2) NULL  
);
```



# 샘플 데이터

## ❖ 샘플 데이터 생성

```
INSERT INTO tStaff VALUES ('김유신','총무부','남','2000-2-3','이사',420,88.8);
INSERT INTO tStaff VALUES ('유관순','영업부','여','2009-3-1','과장',380,NULL);
INSERT INTO tStaff VALUES ('안중근','인사과','남','2012-5-5','대리',256,76.5);
INSERT INTO tStaff VALUES ('윤봉길','영업부','남','2015-8-15','과장',350,71.25);
INSERT INTO tStaff VALUES ('강감찬','영업부','남','2018-10-9','사원',320,56.0);
INSERT INTO tStaff VALUES ('정몽주','총무부','남','2010-9-16','대리',370,89.5);
INSERT INTO tStaff VALUES ('허난설헌','인사과','여','2020-1-5','사원',285,44.5);
INSERT INTO tStaff VALUES ('신사임당','영업부','여','2013-6-19','부장',400,92.0);
INSERT INTO tStaff VALUES ('성삼문','영업부','남','2014-6-8','대리',285,87.75);
INSERT INTO tStaff VALUES ('논개','인사과','여','2010-9-16','대리',340,46.2);
INSERT INTO tStaff VALUES ('황진이','인사과','여','2012-5-5','사원',275,52.5);
INSERT INTO tStaff VALUES ('이율곡','총무부','남','2016-3-8','과장',385,65.4);
INSERT INTO tStaff VALUES ('이사부','총무부','남','2000-2-3','대리',375,50);
INSERT INTO tStaff VALUES ('안창호','영업부','남','2015-8-15','사원',370,74.2);
INSERT INTO tStaff VALUES ('을지문덕','영업부','남','2019-6-29','사원',330,NULL);
INSERT INTO tStaff VALUES ('정약용','총무부','남','2020-3-14','과장',380,69.8);
INSERT INTO tStaff VALUES ('홍길동','인사과','남','2019-8-8','차장',380,77.7);
INSERT INTO tStaff VALUES ('대조영','총무부','남','2020-7-7','차장',290,49.9);
INSERT INTO tStaff VALUES ('장보고','인사과','남','2005-4-1','부장',440,58.3);
INSERT INTO tStaff VALUES ('선덕여왕','인사과','여','2017-8-3','사원',315,45.1);
```

```
SELECT * FROM tStaff;
```

# 샘플 데이터

## ❖ 테이블 구조 확인

DESC tStaff;

DESC tCity;



# 샘플 데이터

## ❖ 샘플 데이터 생성

```
CREATE TABLE DEPT(  
    DEPTNO INT(2),  
    DNAME VARCHAR(14) ,  
    LOC VARCHAR(13),  
    CONSTRAINT PK_DEPT PRIMARY KEY(DEPTNO)  
);
```

```
CREATE TABLE EMP(  
    EMPNO INT(4),  
    ENAME VARCHAR(10),  
    JOB VARCHAR(9),  
    MGR INT(4),  
    HIREDATE DATE,  
    SAL FLOAT(7,2),  
    COMM FLOAT(7,2),  
    DEPTNO INT(2),  
    CONSTRAINT PK_EMP PRIMARY KEY(EMPNO),  
    CONSTRAINT FK_DEPTNO FOREIGN KEY(DEPTNO) REFERENCES DEPT(DEPTNO)  
);
```



# 샘플 데이터

## ❖ 샘플 데이터 생성

```
INSERT INTO DEPT VALUES(10,'ACCOUNTING','NEW YORK');  
INSERT INTO DEPT VALUES (20,'RESEARCH','DALLAS');  
INSERT INTO DEPT VALUES(30,'SALES','CHICAGO');  
INSERT INTO DEPT VALUES(40,'OPERATIONS','BOSTON');
```



# 샘플 데이터

## ❖ 샘플 데이터 생성

```
INSERT INTO EMP VALUES
(7369,'SMITH','CLERK',7902,'1980-12-17',800,NULL,20);
INSERT INTO EMP VALUES
(7499,'ALLEN','SALESMAN',7698,'1981-2-20',1600,300,30);
INSERT INTO EMP VALUES
(7521,'WARD','SALESMAN',7698,'1981-2-22',1250,500,30);
INSERT INTO EMP VALUES
(7566,'JONES','MANAGER',7839,'1981-4-2',2975,NULL,20);
INSERT INTO EMP VALUES
(7654,'MARTIN','SALESMAN',7698,'1981-9-28',1250,1400,30);
INSERT INTO EMP VALUES
(7698,'BLAKE','MANAGER',7839,'1981-5-1',2850,NULL,30);
INSERT INTO EMP VALUES
(7782,'CLARK','MANAGER',7839,'1981-6-9',2450,NULL,10);
INSERT INTO EMP VALUES
(7788,'SCOTT','ANALYST',7566,'1987-7-13',3000,NULL,20);
INSERT INTO EMP VALUES
(7839,'KING','PRESIDENT',NULL,'1981-11-17',5000,NULL,10);
INSERT INTO EMP VALUES
(7844,'TURNER','SALESMAN',7698,'1981-9-8',1500,0,30);
INSERT INTO EMP VALUES
(7876,'ADAMS','CLERK',7788,'1987-7-13',1100,NULL,20);
```

# 샘플 데이터

## ❖ 샘플 데이터 생성

```
INSERT INTO EMP VALUES  
(7900,'JAMES','CLERK',7698,'1981-12-3',950,NULL,30);  
INSERT INTO EMP VALUES  
(7902,'FORD','ANALYST',7566,'1981-12-3',3000,NULL,20);  
INSERT INTO EMP VALUES  
(7934,'MILLER','CLERK',7782,'1982-1-23',1300,NULL,10);
```

```
CREATE TABLE SALGRADE  
( GRADE INT,  
  LOSAL INT,  
  HISAL INT );
```

```
INSERT INTO SALGRADE VALUES (1,700,1200);  
INSERT INTO SALGRADE VALUES (2,1201,1400);  
INSERT INTO SALGRADE VALUES (3,1401,2000);  
INSERT INTO SALGRADE VALUES (4,2001,3000);  
INSERT INTO SALGRADE VALUES (5,3001,9999);
```

```
COMMIT;
```





# 샘플 데이터

## ❖ 샘플 데이터 생성

```
SELECT * FROM DEPT;  
SELECT * FROM EMP;  
SELECT * FROM SALGRADE;
```



# 샘플 데이터

## ❖ 샘플 데이터 생성

--회원테이블

```
create table usertbl(  
  userid char(15) not null primary key,  
  name varchar(20) not null,  
  birthyear int not null,  
  addr char(100),  
  mobile char(11),  
  mdate date)ENGINE=MyISAM AUTO_INCREMENT=1 DEFAULT CHARSET=utf8;
```

--구매테이블

```
create table buytbl(  
  num int auto_increment primary key,  
  userid char(8) not null,  
  productname char(10),  
  groupname char(10),  
  price int not null,  
  amount int not null,  
  foreign key (userid) references usertbl(userid) on delete cascade)ENGINE=MyISAM  
  AUTO_INCREMENT=1 DEFAULT CHARSET=utf8;
```

# 샘플 데이터

## ❖ 샘플 데이터 생성

--데이터 삽입

```
insert into usertbl values('kty', '김태연',1989,'전주','01011111111', '1989-3-9');
insert into usertbl values('bsj', '배수지',1994,'광주','01022222222', '1994-10-10');
insert into usertbl values('ksh', '김설현',1995,'부천','01033333333', '1995-1-3');
insert into usertbl values('bjh', '배주현',1991,'대구','01044444444', '1991-3-29');
insert into usertbl values('ghr', '구하라',1991,'광주','01055555555', '1991-1-13');
insert into usertbl values('san', '산다라박',1984,'부산','01066666666', '1984-11-12');
insert into usertbl values('jsm', '전소미',2001,'캐나다','01077777777', '2001-3-9');
insert into usertbl values('lhl', '이효리',1979,'서울','01088888888', '1979-5-10');
insert into usertbl values('iyou', '아이유',1993,'서울','01099999999', '1993-5-19');
insert into usertbl values('ailee', '에일리',1989,'미국','01000000000', '1989-5-30');
```

commit;



# 샘플 데이터

## ❖ 샘플 데이터 생성

```
insert into buytbl values(null, 'kty', '운동화', '잡화', 30, 2);
insert into buytbl values(null, 'kty', '노트북', '전자', 1000, 1);
insert into buytbl values(null, 'jsm', '운동화', '잡화', 30, 1);
insert into buytbl values(null, 'lhl', '모니터', '전자', 200, 1);
insert into buytbl values(null, 'bsj', '모니터', '전자', 200, 1);
insert into buytbl values(null, 'kty', '청바지', '잡화', 100, 1);
insert into buytbl values(null, 'lhl', '책', '서적', 15, 2);
insert into buytbl values(null, 'iyou', '책', '서적', 15, 7);
insert into buytbl values(null, 'iyou', '컴퓨터', '전자', 500, 1);
insert into buytbl values(null, 'bsj', '노트북', '전자', 1000, 1);
insert into buytbl values(null, 'bjh', '메모리', '전자', 50, 4);
insert into buytbl values(null, 'ailee', '운동화', '잡화', 30, 2);
insert into buytbl values(null, 'ghr', '운동화', '잡화', 30, 1);
```

```
commit;
```

