

# 7장

---

7.1 데이터베이스

7.2 MySQL, 워크벤치 설치하기

7.3 데이터베이스, 테이블 생성하기

7.4 CRUD 작업하기

## 7.1 데이터베이스

---



# 1. 데이터베이스란

» 지금까지는 데이터를 서버 메모리에 저장했음

- 서버를 재시작하면 데이터도 사라져버림 -> 영구적으로 저장할 공간 필요

» MySQL 관계형 데이터베이스 사용

- 데이터베이스: 관련성을 가지며 중복이 없는 데이터들의 집합
- DBMS: 데이터베이스를 관리하는 시스템
- RDBMS: 관계형 데이터베이스를 관리하는 시스템
- 서버의 하드 디스크나 SSD 등의 저장 매체에 데이터를 저장
- 서버 종료 여부와 상관 없이 데이터를 계속 사용할 수 있음
- 여러 사람이 동시에 접근할 수 있고, 권한을 따로 줄 수 있음

▼ 그림 7-2 데이터베이스는 흔히 원기둥 세 개를 겹친 모양으로 표현합니다.



## 7.2 MySQL, 워크벤치 설치하기

---

# 1. 원도

» 공식 사이트(<https://dev.mysql.com/downloads/installer>)

▼ 그림 7-3 MySQL Installer 다운로드 화면

MySQL Community Downloads

MySQL Installer

General Availability (GA) Releases Archives

### MySQL Installer 8.0.19

Select Operating system: Microsoft Windows Looking for previous GA version?

Windows (x86, 32-bit), MSI Installer	8.0.19	18.6M	<a href="#">Download</a>
<small>(mysql-installer-web-community-8.0.19.0.msi) MD5: 2540776a27584d1c1a68a2c071 Signature</small>			
Windows (x86, 32-bit), MSI Installer	8.0.19	296.9M	<a href="#">Download</a>
<small>(mysql-installer-community-8.0.19.0.msi) MD5: 1d00305a71d002e477642b017c Signature</small>			

We suggest that you use the MD5 checksums and GnuPG signatures to verify the integrity of the packages you download.

**ORACLE** © 2019, Oracle Corporation and/or its affiliates.  
Legal Notices | Your Privacy Rights | Terms of Use | Trademark Policy | Contributor Agreement | Cookie Preferences

▼ 그림 7-4 로그인하지 않아도 됩니다.

MySQL Community Downloads

Login Now or Sign Up for a free account.

An Oracle Web Account provides you with the following advantages:

- Fast access to MySQL software downloads
- Download technical White Papers and Presentations
- Post messages in the MySQL Discussion Forums
- Report and track bugs in the MySQL bug system

[Login »](#)  
Using my Oracle Web account

[Sign Up »](#)  
For an Oracle Web account

MySQL.com is using Oracle SSO for authentication. If you already have an Oracle Web account, click the Login link. Otherwise, you can sign up for a free account by clicking the Sign Up link and following the instructions.

[No thanks, just start my download.](#)

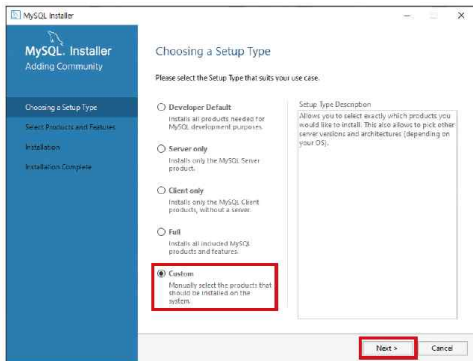
**ORACLE** © 2019, Oracle Corporation and/or its affiliates.  
Legal Notices | Your Privacy Rights | Terms of Use | Trademark Policy | Contributor Agreement | Cookie Preferences

## 2. 윈도우 MySQL Installer

### » 다운로드 받은 MySQL Installer 실행

- 설치 지혜 주 Choosing a Setup Type 브부에서 Custom 선택

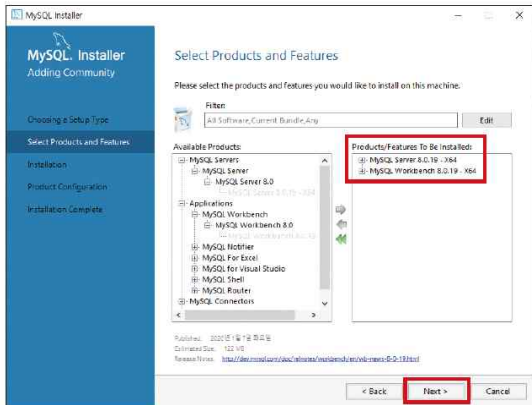
♥ 그림 7-5 Custom을 선택합니다.



# 3. Server와 Workbench 설치

» 자שי이 으여 체제 (x86 과 x64) 에서 Server와 Workbench만 선택하여 설치

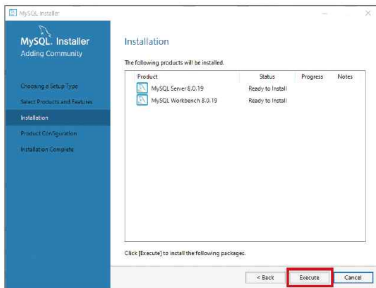
♥ 그림 7-6 MySQL Server와 MySQL Workbench 선택



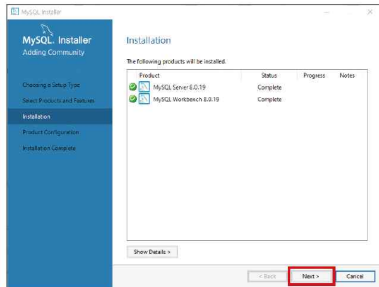
## 4. Execute

» Execute로 설치한 후 Next로 마무리

▼ 그림 7-7 설치 화면



▼ 그림 7-8 설치 완료 화면

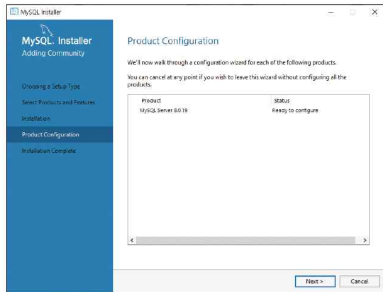




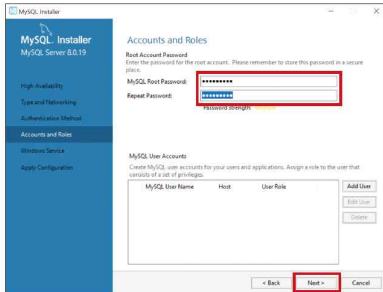
# 5. MySQL 기본 설정하기

» 계속 Next를 누르다가 비밀번호 설정하고 넘어가기

▼ 그림 7-9 MySQL 설정 화면



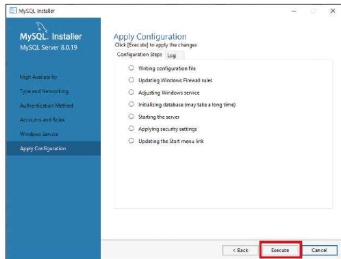
▼ 그림 7-10 root 비밀번호 설정 화면



# 6. 설정 적용하기

## »Execute를 눌러 적용

▼그림 7-11 설정 반영 전 화면



### Note 三 Starting Server에서 막한다면

Apply Configuration을 진행하는 도중 Starting Server에서 넘어가지 못하고 에러가 발생한다면 MySQL 설치 경로나 사용자 계정 이름, 컴퓨터 이름 등에 한글이 포함되어 있는 것은 아닌지 살펴봐야 합니다. 한글이 있다면 영어로 바꾸고 다시 시도해보세요. 프로그래밍 관련 프로그램을 설치할 때는 한글로 된 경로, 계정, 컴퓨터명 등을 사용하지 않는 것이 좋습니다.



# 7. MySQL 접속해보기

» 콘솔(CMD)에서 MySQL이 설치된 경로로 이동

- 기본 경로는 C:\Program Files\MySQL\MySQL Server 8.0\bin
- -h는 호스트, -u는 사용자 -p는 비밀번호 의미

콘솔

```
$ mysql -h localhost -u root -p
Enter password: [비밀번호 입력]
mysql>
```

- 프롬프트가 mysql>로 바뀐다면 성공
- 프롬프트를 종료하려면 exit 입력

콘솔

```
mysql> exit
Bye
```



# 8. macOS

## » 홈브루(homebrew)를 통해 설치하는 게 편함

콘솔

```
$ /usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
```

## » brew 설치 후 MySQL 설치

콘솔

```
$ brew install mysql  
$ brew services start mysql  
$ mysql_secure_installation
```

- » root 비밀번호 설정 후, validate\_password 플러그인을 설치하겠냐고 물으면 모두 n을 입력하고 엔터를 눌러 건너 뛴
- 실서버 운영 시에는 설정해주는 것이 좋음
  - 윈도우와 같은 방법으로 MySQL에 접속



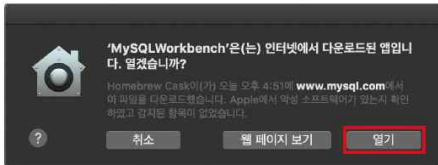
# 9. macOS 워크벤치 설치

## » Homebrew로 다운로드

콘솔

```
$ brew cask install mysqlworkbench
```

▼ 그림 7-17 워크벤치 경고창





# 10. 리눅스(우분투)에 MySQL 설치

» GUI를 사용하지 않으므로 콘솔에 다음 명령어를 순서대로 입력

콘솔

```
$ sudo apt-get update  
$ sudo apt-get install -y mysql-server  
$ mysql_secure_installation
```

» root 비밀번호 설정



» 원도와 같은 방법으로 mysql에 접속

» 우분투의 경우는 워크벤치 대신 콘솔에서 작업

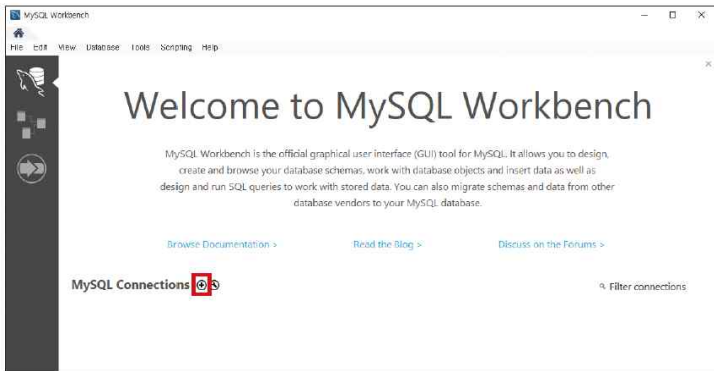


# 11. 윈도우, 맥 워크벤치에서 커넥션 생성

» 워크벤치 프로그램 실행

» MySQL Connections 옆에 + 모양 클릭

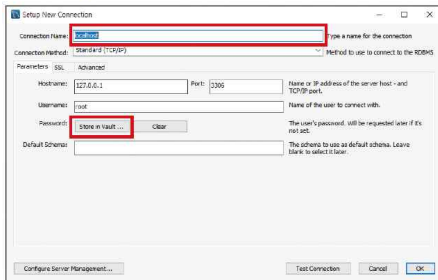
▼ 그림 7-18 워크벤치 실행 화면



## 12. MySQL 서버 주소 입력

» Connection Name에 localhost 적고 비밀번호 입력

♥ 그림 7-19 커넥션 생성 화면



The 'Setup New Connection' dialog box is shown. The 'Connection Name' field is highlighted with a red rectangle and contains the text 'localhost'. The 'Connection Method' is set to 'Standard (TCP/IP)'. The 'Parameters' tab is selected, showing 'Hostnames' as '127.0.0.1', 'Port' as '3306', 'Username' as 'root', and 'Password' as 'Store in Vault ...'. The 'Default Schema' field is empty. At the bottom, there are buttons for 'Configure Server Management...', 'Test Connection', 'Cancel', and 'OK'.

♥ 그림 7-20 비밀번호 저장하기



The 'Store Password For Connection' dialog box is shown. It prompts the user to enter a password for the service. The 'Service' is 'Mysqld@127.0.0.1:3306' and the 'User' is 'root'. The 'Password' field is empty. The 'OK' button is highlighted with a red rectangle.

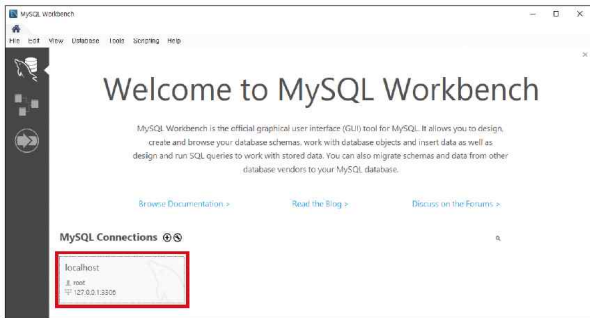




# 13. localhost 서버 접속하기

» 아까 설정했던 비밀번호를 입력하여 접속

♥ 그림 7-21 localhost 접속하기



» 접속이 되었다면 성공

## 7.3 데이터베이스, 테이블 생성하기

---



# 1. 데이터베이스 생성하기

» 콘솔에서 MySQL 프롬프트에 접속

- **CREATE SCHEMA nodejs;**로 nodejs 데이터베이스 생성
- **use nodejs;**로 생성한 데이터베이스 선택

### 콘솔

```
mysql> CREATE SCHEMA `nodejs` DEFAULT CHARACTER SET utf8;  
Query OK, 1 row affected (0.01sec)  
mysql> use nodejs;  
Database changed
```



# 2. 테이블 생성하기

## » MySQL 프롬프트에서 테이블 생성

- CREATE TABLE [데이터베이스명.테이블명]으로 테이블 생성
- 사용자 정보를 저장하는 테이블

### 콘솔

```
mysql> CREATE TABLE nodejs.comments (  
  -> id INT NOT NULL AUTO_INCREMENT,  
  -> commenter INT NOT NULL,  
  -> comment VARCHAR(100) NOT NULL,  
  -> created_at DATETIME NOT NULL DEFAULT now(),  
  -> PRIMARY KEY(id),  
  -> INDEX commenter_idx (commenter ASC),  
  -> CONSTRAINT commenter  
  -> FOREIGN KEY (commenter)  
  -> REFERENCES nodejs.users (id)  
  -> ON DELETE CASCADE  
  -> ON UPDATE CASCADE)  
  -> COMMENT = '댓글'  
  -> DEFAULT CHARSET=utf8mb4  
  -> ENGINE=InnoDB;
```

Query OK, 0 row affected (0.09 sec)



### 3. 컬럼과 로우

» 나이, 결혼 여부, 성별같은 정보가 컬럼

» 실제로 들어가는 데이터는 로우

Note ≡ 컬럼과 로우

♥ 그림 7-27 컬럼과 로우



id	name	age	married
1	zero	24	false
2	nero	32	true
3	hero	28	false



# 4. 컬럼 옵션들

## » id INT NOT NULL AUTO\_INCREMENT

- 컬럼명 옆의 것들은 컬럼에 대한 옵션들
- INT: 정수 자료형(FLOAT, DOUBLE은 실수)
- VARCHAR: 문자열 자료형, 가변 길이(CHAR은 고정 길이)
- TEXT: 긴 문자열은 TEXT로 별도 저장
- DATETIME: 날짜 자료형 저장
- TINYINT: -128에서 127까지 저장하지만 여기서는 1 또는 0만 저장해 볼 값 표현
- NOT NULL: 빈 값은 받지 않는다는 뜻(NULL은 빈 값 허용)
- AUTO\_INCREMENT: 숫자 자료형인 경우 다음 로우가 저장될 때 자동으로 1 증가
- UNSIGNED: 0과 양수만 허용
- ZEROFILL: 숫자의 자리 수가 고정된 경우 빈 자리에 0을 넣음
- DEFAULT now(): 날짜 컬럼의 기본값을 현재 시간으로



# 5. Primary Key, Unique Index

## » PRIMARY KEY(id)

- id가 테이블에서 로우를 특정할 수 있게 해주는 고유한 값임을 의미
- 학번, 주민등록번호같은 개념

## » UNIQUE INDEX name\_UNIQUE (name ASC)

- 해당 컬럼(name)이 고유해야 함을 나타내는 옵션
- name\_UNIQUE는 이 옵션의 이름(아무거나 다른 걸로 지어도 됨)
- ASC는 인덱스를 오름차순으로 저장함의 의미(내림차순은 DESC)



# 6. 테이블 옵션

- » COMMENT: 테이블에 대한 보충 설명(필수 아님)
- » DEFAULT CHARSET: utf8로 설정해야 한글이 입력됨
- » ENGINE: InnoDB 사용(이외에 MyISAM이 있음, 엔진별로 기능 차이 존재)





# 7. 테이블 생성되었나 확인하기

## » DESC 테이블명

콘솔

```
mysql> DESC users;
```

♥ 그림 7-27 DESC 명령어 결과

```
mysql> DESC users;
```

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	auto_increment
name	varchar(20)	NO	UNI	NULL	
age	int unsigned	NO		NULL	
married	tinyint(1)	NO		NULL	
comment	text	YES		NULL	
created_at	datetime	NO		CURRENT_TIMESTAMP	DEFAULT_GENERATED

6 rows in set (0.00 sec)

## » 테이블 삭제하기: DROP TABLE 테이블명

콘솔

```
mysql> DROP TABLE users;
```



# 8. 댓글 테이블 저장하기

## » comments 테이블 생성

### 콘솔

```
mysql> CREATE TABLE nodejs.comments (  
  -> id INT NOT NULL AUTO_INCREMENT,  
  -> commenter INT NOT NULL,  
  -> comment VARCHAR(100) NOT NULL,  
  -> created_at DATETIME NOT NULL DEFAULT now(),  
  -> PRIMARY KEY(id),  
  -> INDEX commenter_idx (commenter ASC),  
  -> CONSTRAINT commenter  
  -> FOREIGN KEY (commenter)  
  -> REFERENCES nodejs.users (id)  
  -> ON DELETE CASCADE  
  -> ON UPDATE CASCADE)  
  -> COMMENT = '댓글'  
  -> DEFAULT CHARSET=utf8mb4  
  -> ENGINE=InnoDB;
```

Query OK, 0 row affected (0.09 sec)



# 9. 외래키(foreign key)

» 댓글 테이블은 사용자 테이블과 관계가 있음(사용자가 댓글을 달기 때문)

- 외래키를 두어 두 테이블이 관계가 있다는 것을 표시
- FOREIGN KEY (컬럼명) REFERENCES 데이터베이스.테이블명 (컬럼)
- FOREIGN KEY (commenter) REFERENCES nodejs.users (id)
- 댓글 테이블에는 commenter 컬럼이 생기고 사용자 테이블의 id값이 저장됨
- ON DELETE CASCADE, ON UPDATE CASCADE
- 사용자 테이블의 로우가 지워지고 수정될 때 댓글 테이블의 연관된 로우들도 같이 지워지고 수정됨
- 데이터를 일치시키기 위해 사용하는 옵션(CASCADE 대신 SET NULL과 NO ACTION도 있음)



# 10. 테이블 목록 보기

- SHOW TABLES;

콘솔

```
mysql> SHOW TABLES;
```

```
+-----+
```

```
| Tables_in_nodejs |
```

```
+-----+
```

```
| comments          |
```

```
| users              |
```

```
+-----+
```

```
2 rows in set (0.00 sec)
```

## 7.4 CRUD 작업하기

---

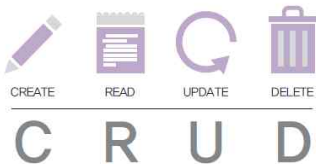


# 1. CRUD

» Create, Read, Update, Delete의 두문자어

- 데이터베이스에서 많이 하는 작업 4가지

♥ 그림 7-37 CRUD 작업





# 2. Create

## » INSERT INTO 테이블 (컬럼명들) VALUES (값들)

콘솔

```
mysql> INSERT INTO nodejs.users (name, age, married, comment) VALUES ('zero', 24, 0, '자기소개1');
```

```
Query OK, 1 row affected (0.01 sec)
```

```
mysql> INSERT INTO nodejs.users (name, age, married, comment) VALUES ('nero', 32, 1, '자기소개2');
```

```
Query OK, 1 row affected (0.02 sec)
```

콘솔

```
mysql> INSERT INTO nodejs.comments (commenter, comment) VALUES (1, '안녕하세요. zero의 댓글입니다');
```

```
Query OK, 1 row affected (0.02 sec)
```



## 3. Read

### » SELECT 컬럼 FROM 테이블명

- SELECT \* 은 모든 컬럼을 선택한다는 의미

콘솔

```
mysql> SELECT * FROM nodejs.users;
```

id	name	age	married	comment	created_at
1	zero	24	0	자기소개1	2017-10-25 07:06:33
2	nero	32	1	자기소개2	2017-10-25 09:25:40

2 rows in set (0.00 sec)

- 컬럼만 따로 추리는 것도 가능

콘솔

```
mysql> SELECT name, married FROM nodejs.users;
```

name	married
zero	0
nero	1

2 rows in set (0.00 sec)





## 4. Read 옵션들

### » WHERE로 조건을 주어 선택 가능

- AND로 여러가지 조건을 동시에 만족하는 것을 찾음

콘솔

```
mysql> SELECT name, age FROM nodejs.users WHERE married = 1 AND age > 30;
```

```
+-----+-----+  
| name | age |  
+-----+-----+  
| nero | 32 |  
+-----+-----+
```

1 row in set (0.00 sec)

- OR로 여러가지 조건 중 하나 이상을 만족하는 것을 찾음

콘솔

```
mysql> SELECT id, name FROM nodejs.users WHERE married = 0 OR age > 30;
```

```
+-----+-----+  
| id | name |  
+-----+-----+  
| 1 | zero |  
| 2 | nero |  
+-----+-----+
```

2 rows in set (0.01 sec)



# 5. 정렬해서 찾기

» ORDER BY로 특정 컬럼 값 순서대로 정렬 가능

- DESC는 내림차순, ASC 오름차순

콘솔

```
mysql> SELECT id, name FROM nodejs.users ORDER BY age DESC;
```

```
+-----+
```

```
| id | name |
```

```
+-----+
```

```
| 2 | nero |
```

```
| 1 | zero |
```

```
+-----+
```

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



## 6. LIMIT, OFFSET

### » LIMIT으로 조회할 개수 제한

콘솔

```
mysql> SELECT id, name FROM nodejs.users ORDER BY age DESC LIMIT 1;
```

```
+-----+
```

```
| id | name |
```

```
+-----+
```

```
| 2 | nero |
```

```
+-----+
```

```
1 row in set (0.00 sec)
```

### » OFFSET으로 앞의 로우들 스킵 가능(OFFSET 2면 세 번째 것부터 찾음)

콘솔

```
mysql> SELECT id, name FROM nodejs.users ORDER BY age DESC LIMIT 1 OFFSET 1;
```

```
+-----+
```

```
| id | name |
```

```
+-----+
```

```
| 1 | zero |
```

```
+-----+
```

```
1 row in set (0.00 sec)
```



# 7. Update

» 데이터베이스에 있는 데이터를 수정하는 작업

- UPDATE 테이블명 SET 컬럼=새값 WHERE 조건

콘솔

```
mysql> UPDATE nodejs.users SET comment = '바뀐 내용' WHERE id = 2;  
Query OK, 1 row affected (0.01 sec)  
Rows matched: 1  Changed: 1  Warnings: 0
```



# 8. Delete

» 데이터베이스에 있는 데이터를 삭제하는 작업

- DELETE FROM 테이블명 WHERE 조건

콘솔

```
mysql> DELETE FROM nodejs.users WHERE id = 2;  
Query OK, 1 row affected (0.00 sec)
```