

삼성 청년 SW 아카데미

Database - MySQL

<알림>

본 강의는 삼성 청년 SW아카데미의 컨텐츠로
보안서약서에 의거하여
강의 내용을 어떠한 사유로도 임의로 복사,
촬영, 녹음, 복제, 보관, 전송하거나
허가 받지 않은 저장매체를
이용한 보관, 제3자에게 누설, 공개,
또는 사용하는 등의 행위를 금합니다.

1장. DB 트랜드와 MySQL 개요

챕터의 포인트

- Database의 필요성
- Database 트랜드
- MySQL DBMS 설치
- DB 계정 생성 및 접속

Database의 필요성

DB

- Database
- 데이터의 기지 (Base), Data들의 묶음
- 데이터의 집합을 뜻하는 개념적 단어

DBMS

- Database (Data들) 을 관리하기 도구들을 모아 둔 시스템
- MySQL / Oracle / Mongo DB / ProtgreSQL 등 존재
- 프로그래밍에서 DB라고 부르는 것은 DBMS를 포함

DBMS을 통상 DB라고 부른다.

- 대표적인 DB 종류

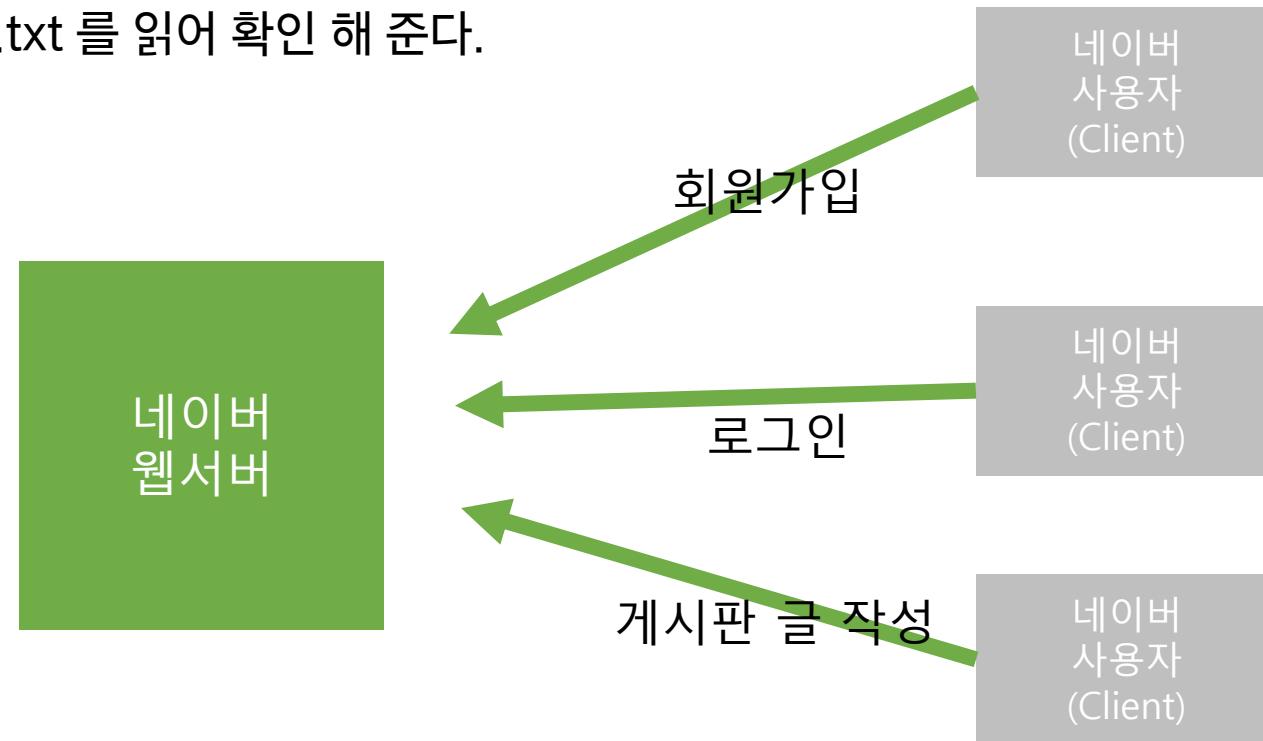
- Oracle사가 소유한 MySQL - 오픈소스 업계 1위
- Oracle사가 소유한 Oracle - 유료(기업대상) 업계 1위



사용자 데이터를 DB가 아닌 Text file에 저장하면 된다.

- **DB 없이 웹 서비스 개발 방법**

- 웹서버에 id.txt / pass.txt 파일을 만들어서 저장을 해 둔다.
- 게시판 글은 content.txt 파일을 만들어 저장을 해 둔다.
- 로그인 요청시 id.txt / pass.txt 를 읽어 확인 해 준다.



버그 없이 고성능으로 구현해야 한다.

- 개발자가 구현해야 할 것
 - 해커의 모든 행동을 고려하여 구현 필요
 - 로그인시 빠르게 text file을 찾는 알고리즘 구현
 - 고객 데이터가 버그로 인해 망가지는 일이 없어야 함
- 개발자 실수로 인해 해킹을 당하거나,
고객 데이터가 망가질 확률이 매우 크다.



고객 데이터를 다루는 일은
위험 물질을 다루는 것과 같다.

이 위험한 데이터들을 안전하게 관리해준다.

- 데이터들을 안전하게 다루도록 해준다.
- 고속으로 데이터들을 Read / Write 할 수 있음
- 세계 고수들의 알고리즘들이 모두 구현되어 있음

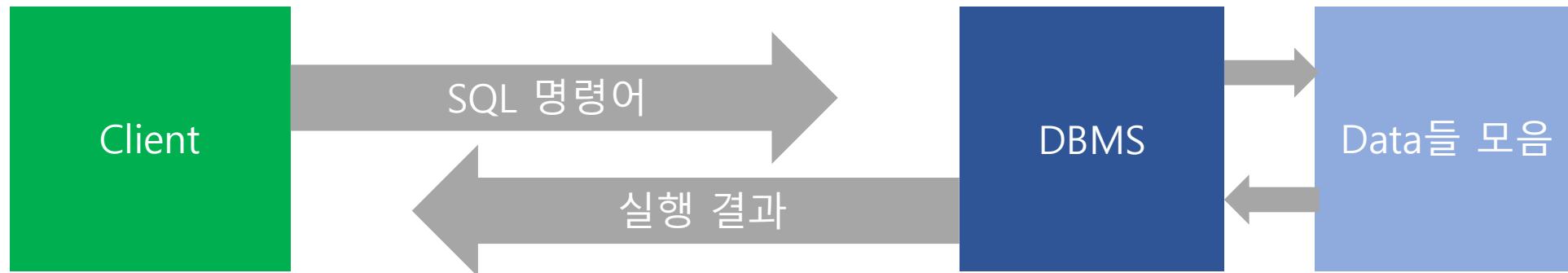
데이터들을 편리하고, 빠르고, 안전하고, 무료로 사용할 수 있다.

- 따라서, 데이터 관리는 DB가 필수



데이터 검색 알고리즘 / 암호화 등을, 직접 구현을 안해도 됨

1. DBMS에서 데이터 관리 방법을 알고,
2. DBMS를 제어하는 “**SQL 명령어**”만 학습하면 데이터 제어를 손쉽게 가능



Database 트랜드

DB는 RDBMS (관계형 DBMS) / 비관계형 DBMS으로 나뉜다.

- **관계형 DBMS**

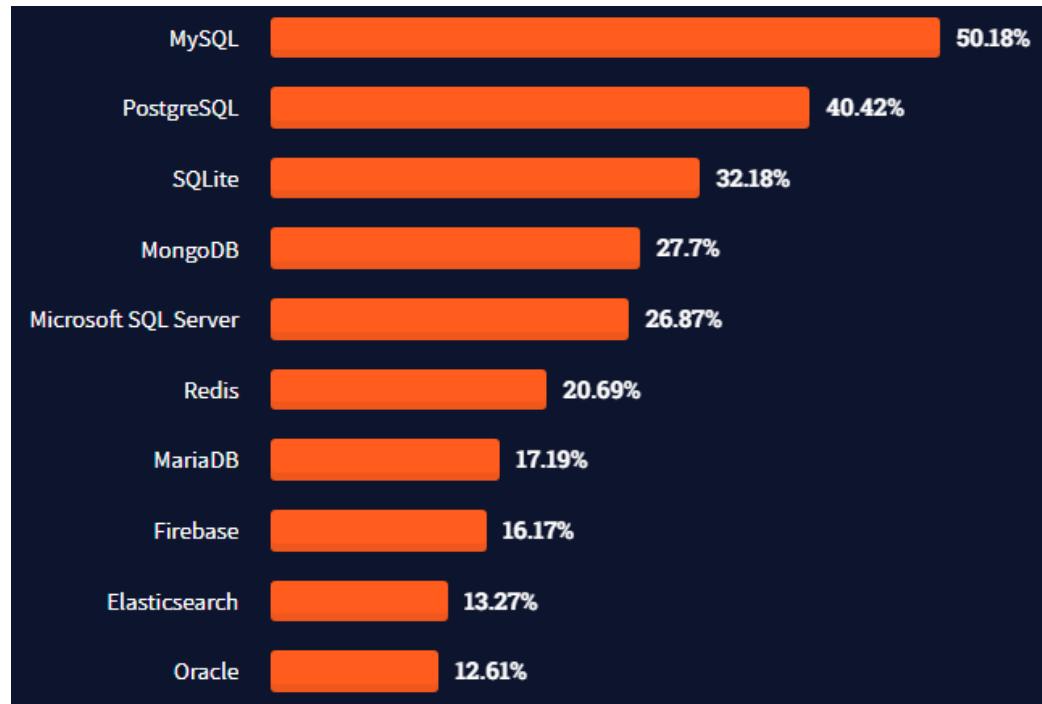
- 구조 및 제약조건 (스키마)를 만들고 값을 채워야 함
- 대표적 RDBMS : MySQL / Oracle

- **비관계형 DBMS**

- RDBMS의 구조로 저장하지 않는 DBMS를 뜻한다.
- 비관계형 DBMS 마다 구조가 다양하다.
- 대표적인 비관계형 DBMS
 - Mongo DB
 - Redis

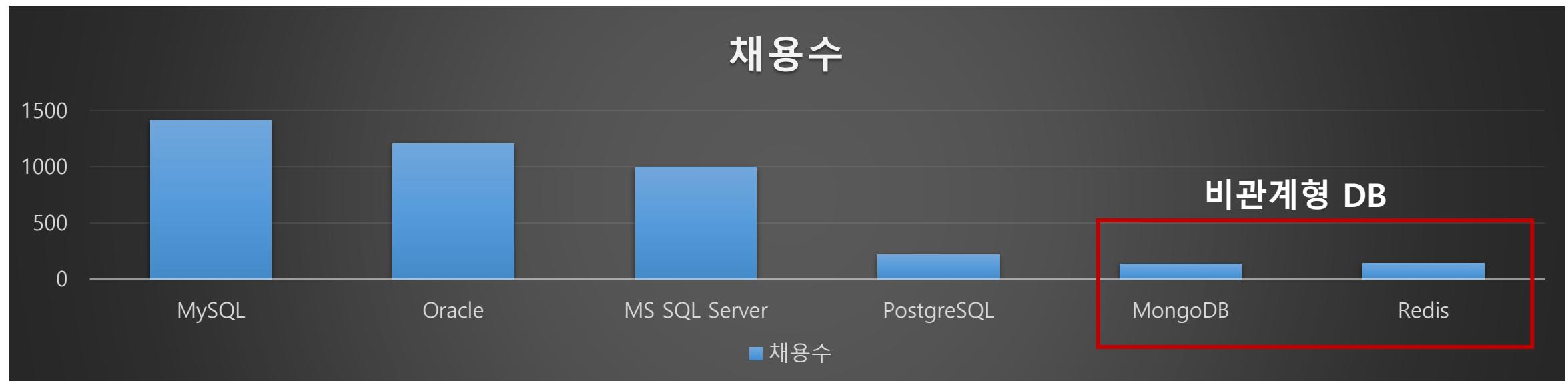
Database (DBMS) 순위

- stackoverflow 2021 Survey
- 전 세계 트랜드 확인 가능 (<https://insights.stackoverflow.com/survey/2021>)
 - MySQL은 부동의 1위이며, 국외에서는 PostgreSQL (포스트 그레스 큐엘)이 인기가 좋은 편



국내 채용사이트 기준

- 사람인 2020년, Backend / Server 개발자 기준
 - 1위 : MySQL
 - 2위 : Oracle (기업에서 유료)
 - 3위 : Microsoft SQL Server



MySQL

- 중요정보

- Open Source / 기업도 무료
- DBMS에 포함된 도구
 - MySQL Server 를 기본적으로 포함
 - MySQL Client 프로그램
 - GUI 용 : MySQL Workbench
 - CLI 용 : mysql

- 덜 중요 정보

- C / C++ 로 제작 됨
- 제작자 딸 이름 “My” 에서 지어진 이름
- Oracle 이 인수 함

DB 명령어 (SQL문)을 작성하는 방식의 코딩테스트

- Web 개발 Startup 회사들

- 당근마켓
- 왓챠
- 등등

- SK 주식회사 (C&C)

— 코드를 입력하세요
1. SELECT

SQL 문 작성

mysql / oracle 선택 가능

MySQL 기본기 확립

1. 임베디드 / IoT 개발시 Database 활용을 위함
2. 웹 개발시 활용을 위함
3. Database 코테 합격을 위함

MySQL DBMS 설치

MySQL 웹사이트 접속하여 MySQL Community 다운로드

The screenshot shows the MySQL Database Service website. At the top, there's a navigation bar with the MySQL logo, a search icon, and links for 'MySQL.COM', 'DOWNLOADS' (which is underlined), 'DOCUMENTATION', and 'DEVELOPER ZONE'. To the right of the navigation are 'Contact MySQL', 'Login', and 'Register' links, along with social media icons for Facebook, Twitter, LinkedIn, and YouTube. The main content area features a large blue background with a white cloud graphic. On the left, the text 'MySQL Database Service' and 'with HeatWave' is displayed. On the right, there are two sections: 'Faster Performance' with a list of competitive advantages, and 'Lower Total Cost of Ownership' with a list of cost savings. At the bottom left is a 'Try Now' button.

The world's most popular open source database

MySQL™

MySQL.COM DOWNLOADS DOCUMENTATION DEVELOPER ZONE

Contact MySQL | Login | Register

f t in Y

MySQL Database Service

with HeatWave

Faster Performance

- 5400x Faster than Amazon RDS
- 1400x Faster than Amazon Aurora
- 6.5x Faster than Amazon Redshift AQUA
- 7x Faster than Snowflake

Lower Total Cost of Ownership

- 2/3 the cost of Amazon RDS
- 1/2 the cost of Amazon Aurora
- 1/2 the cost of Amazon Redshift AQUA
- 1/5 the cost of Snowflake

Try Now >

[MySQL Community \(GPL\) Downloads »](#)

MySQL Installer for Windows 클릭

④ MySQL Community Downloads

- [MySQL Yum Repository](#)
- [MySQL APT Repository](#)
- [MySQL SUSE Repository](#)
- [MySQL Community Server](#)
- [MySQL Cluster](#)
- [MySQL Router](#)
- [MySQL Shell](#)
- [MySQL Workbench](#)
- [MySQL Installer for Windows](#)
- [MySQL for Visual Studio](#)
- [C API \(libmysqlclient\)](#)
- [Connector/C++](#)
- [Connector/J](#)
- [Connector/.NET](#)
- [Connector/Node.js](#)
- [Connector/ODBC](#)
- [Connector/Python](#)
- [MySQL Native Driver for PHP](#)
- [MySQL Benchmark Tool](#)
- [Time zone description tables](#)
- [Download Archives](#)

- 항상 버전체크

8.0.20

- 이슈가 있으므로 다운그레이드

- ④ MySQL Community Downloads

◀ MySQL Installer

The screenshot shows the MySQL Community Downloads page. At the top, there are two tabs: "General Availability (GA) Releases" (highlighted with a red box) and "Archives". Below the tabs, the title "MySQL Installer 8.0.28" is displayed. A dropdown menu "Select Operating System:" is set to "Microsoft Windows". To the right, a button says "Looking for previous GA versions?". The main content area lists two installer packages:

Installer Type	Version	File Size	Action
Windows (x86, 32-bit), MSI Installer (mysql-installer-web-community-8.0.28.0.msi)	8.0.28	2.3M	Download
Windows (x86, 32-bit), MSI Installer (mysql-installer-community-8.0.28.0.msi)	8.0.28	435.7M	Download

At the bottom, a note says: "We suggest that you use the MD5 checksums and GnuPG signatures to verify the integrity of the packages you download."

- 항상 버전체크

8.0.20

- 이슈가 있으므로 다운그레이드

- ① MySQL Product Archives

- ◀ MySQL Installer (Archived Versions)

⚠ Please note that these are old versions. New releases will have recent bug fixes and features!
To download the latest release of MySQL Installer, please visit [MySQL Downloads](#).

Product Version: **8.0.20**

Operating System: Microsoft Windows

Windows (x86, 32-bit), MSI Installer	Apr 14, 2020	24.4M	Download
(mysql-installer-web-community-8.0.20.0.msi)			MD5: 26ae47807122bf0052b99ebf893b0dac Signature

Windows (x86, 32-bit), MSI Installer	Apr 14, 2020	420.6M	Download
(mysql-installer-community-8.0.20.0.msi)			MD5: a69c77fe737654d8931079b4623b9e1a Signature

! We suggest that you use the [MD5 checksums](#) and [GnuPG signatures](#) to verify the integrity of the packages you download.

- 로그인은 무시

⑤ 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 »](#)

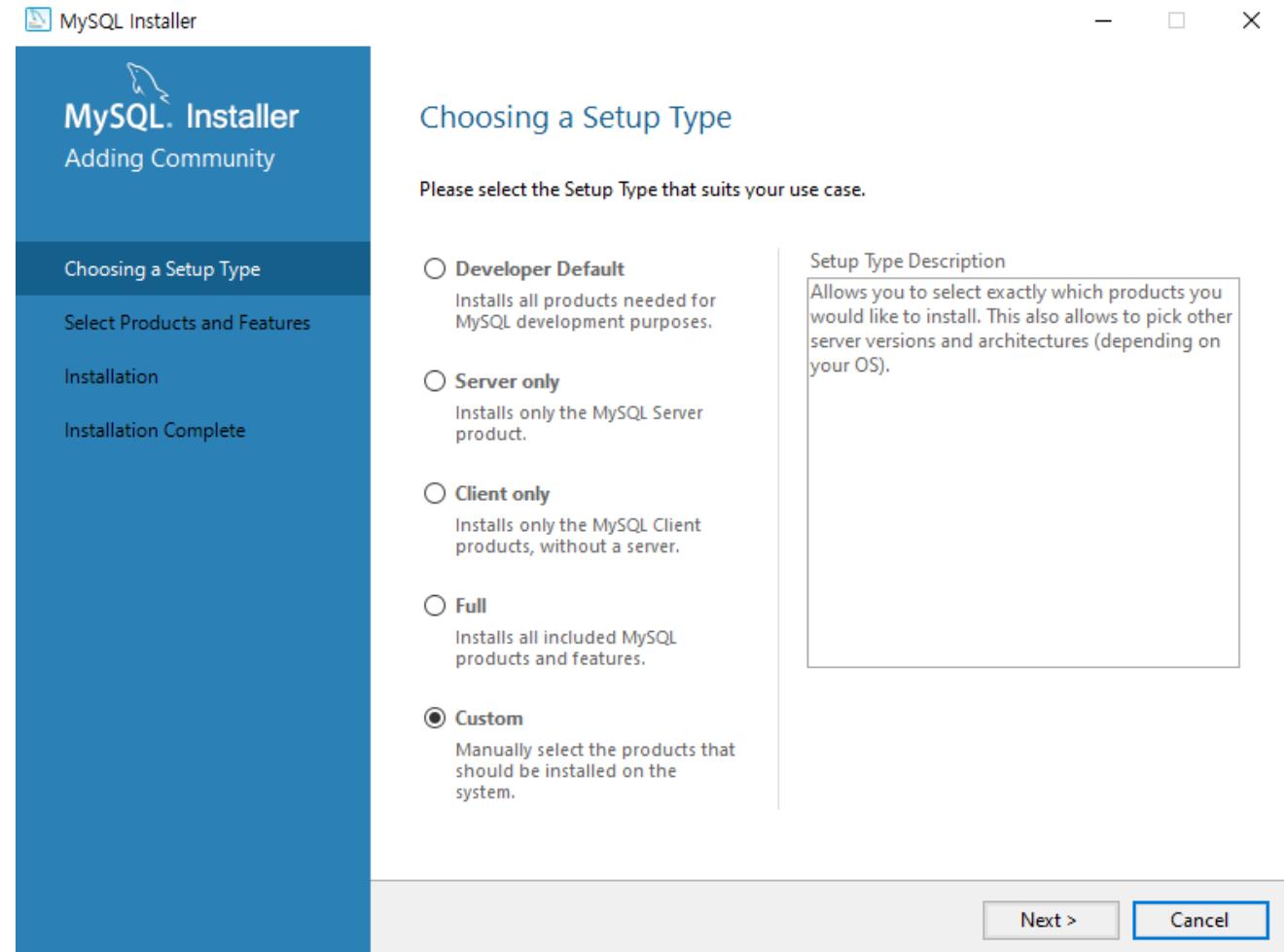
for an Oracle Web account

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

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

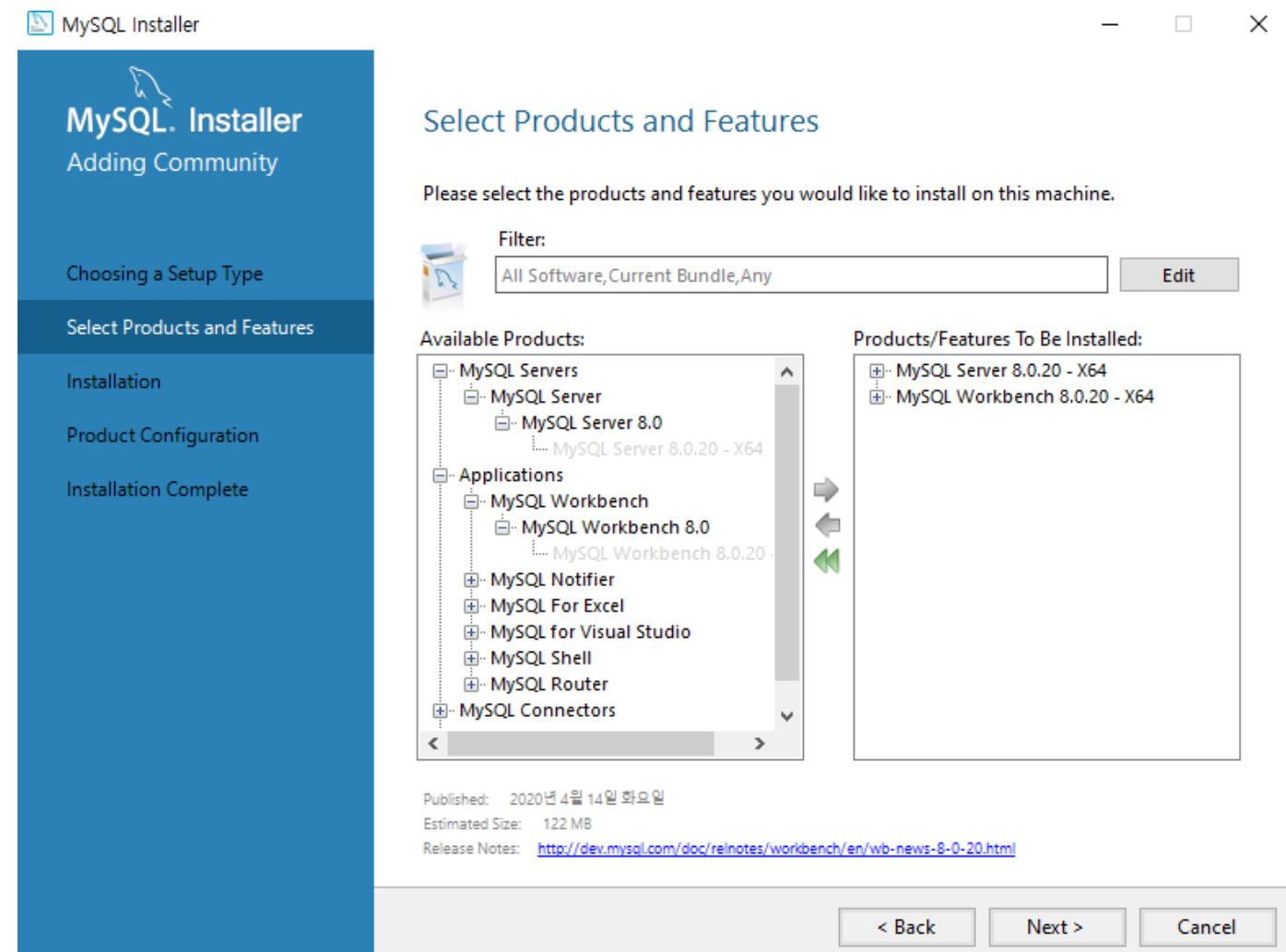
Custom 설정으로 설치를 한다.

- 실제 강의에 필요한 도구들
 - MySQL Server
 - MySQL Workbench

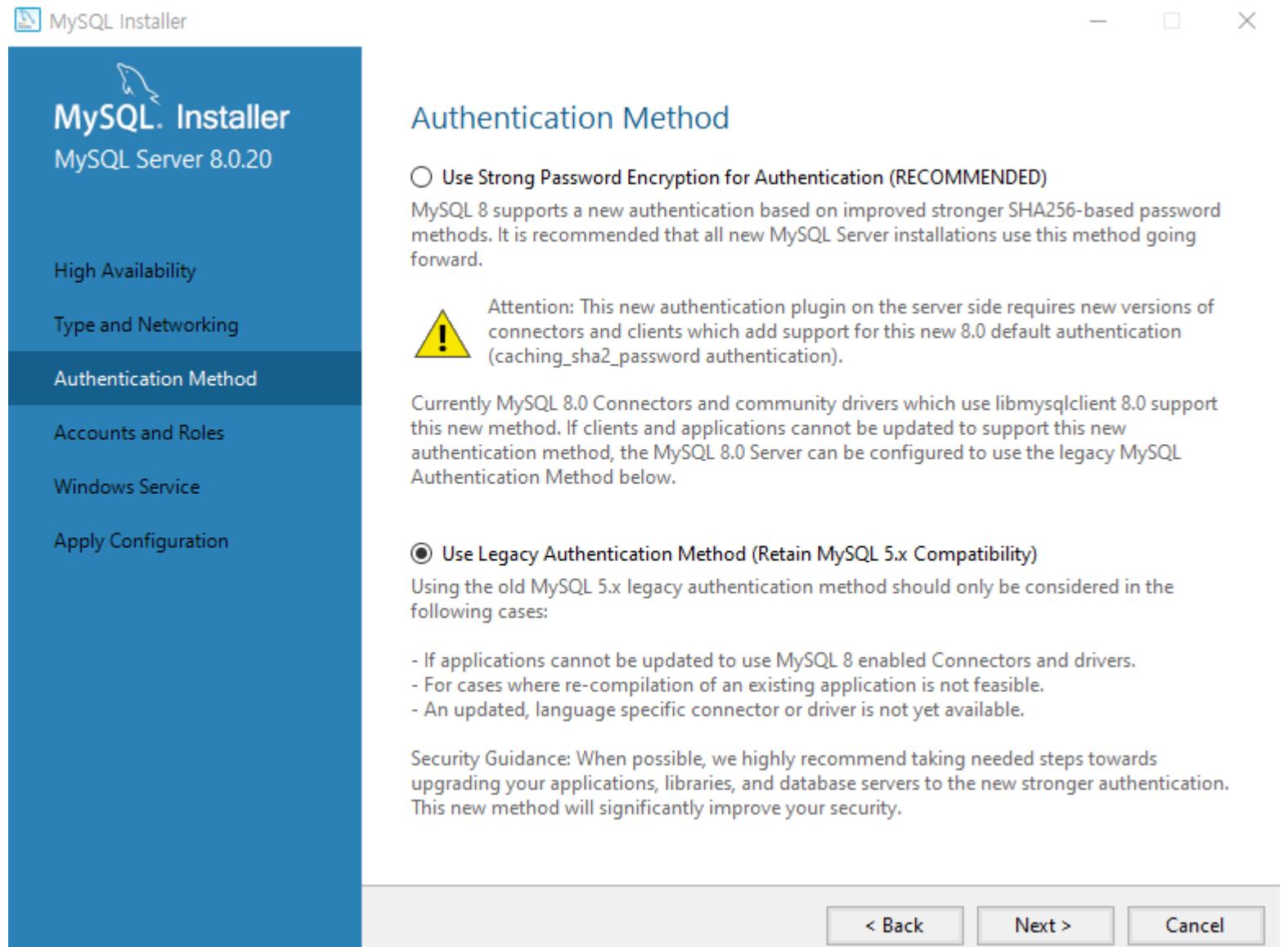


Custom 설정으로 설치를 한다.

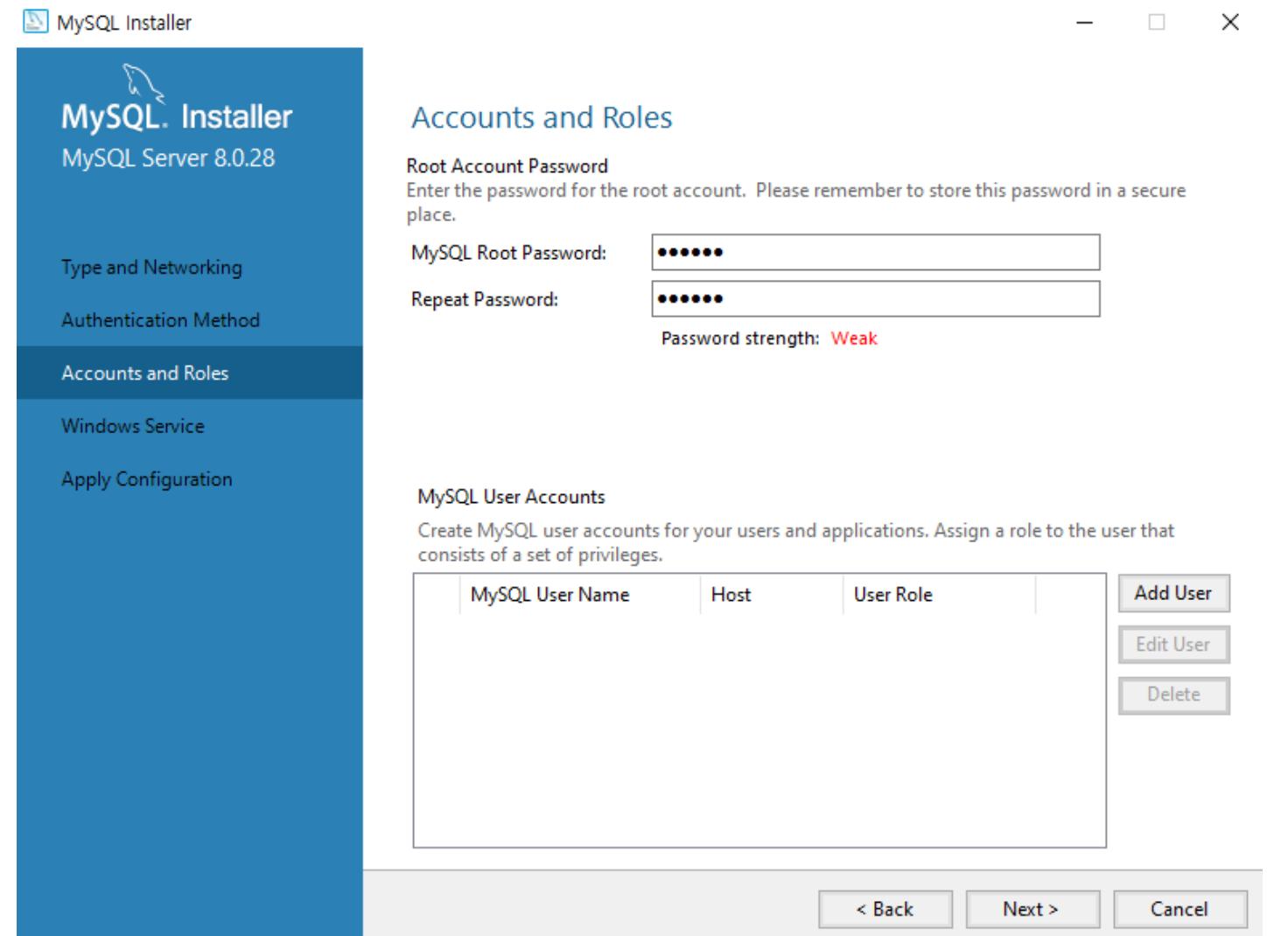
- 실제 강의에 필요한 도구들
 - MySQL Server
 - MySQL Workbench



• 레거시로 선택



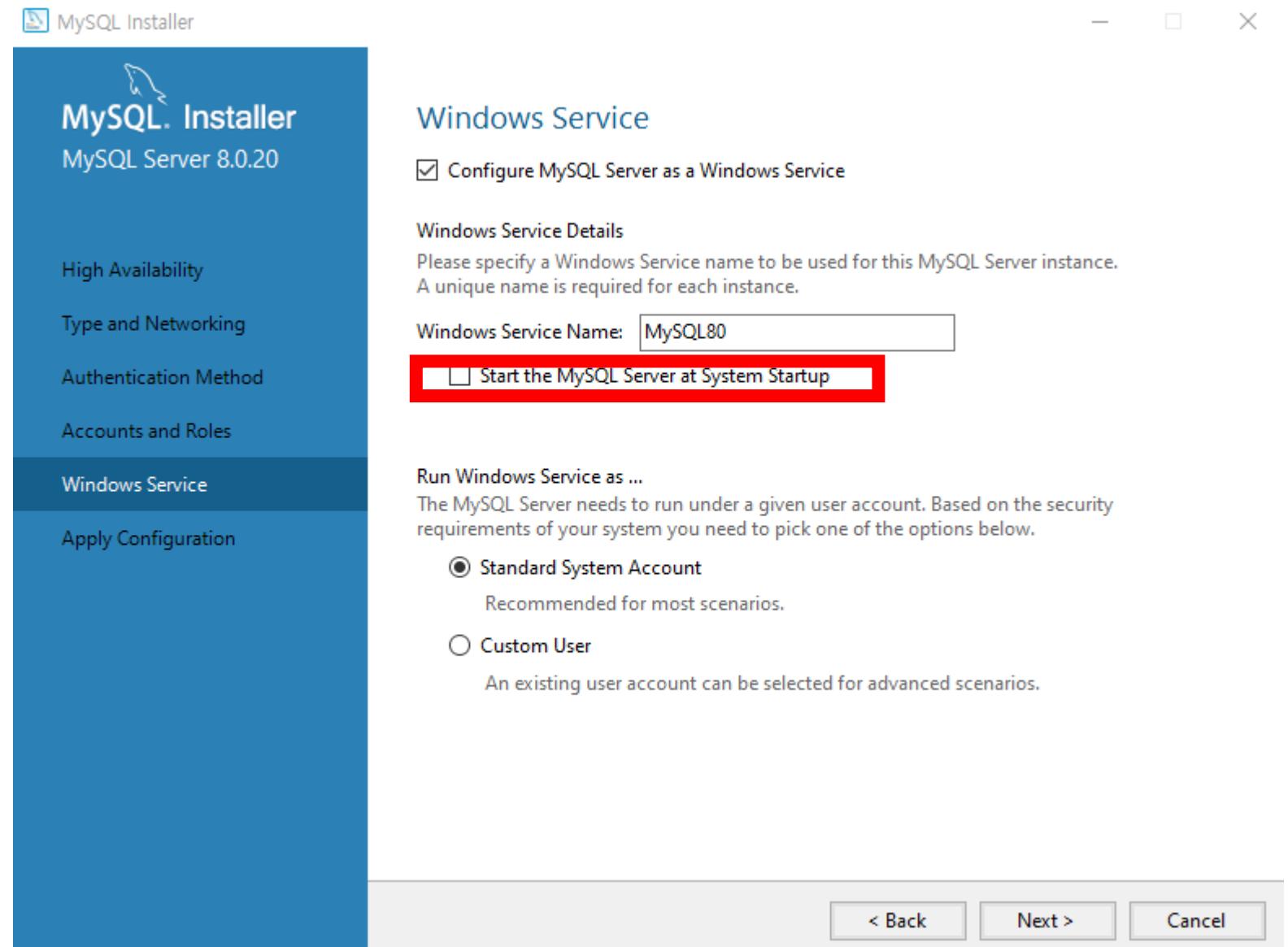
- 비밀번호는 1 여섯개
- 실전에선 말도 안되지만,
실습목적.



시작 시 자동실행 안되도록 조치

Confidential

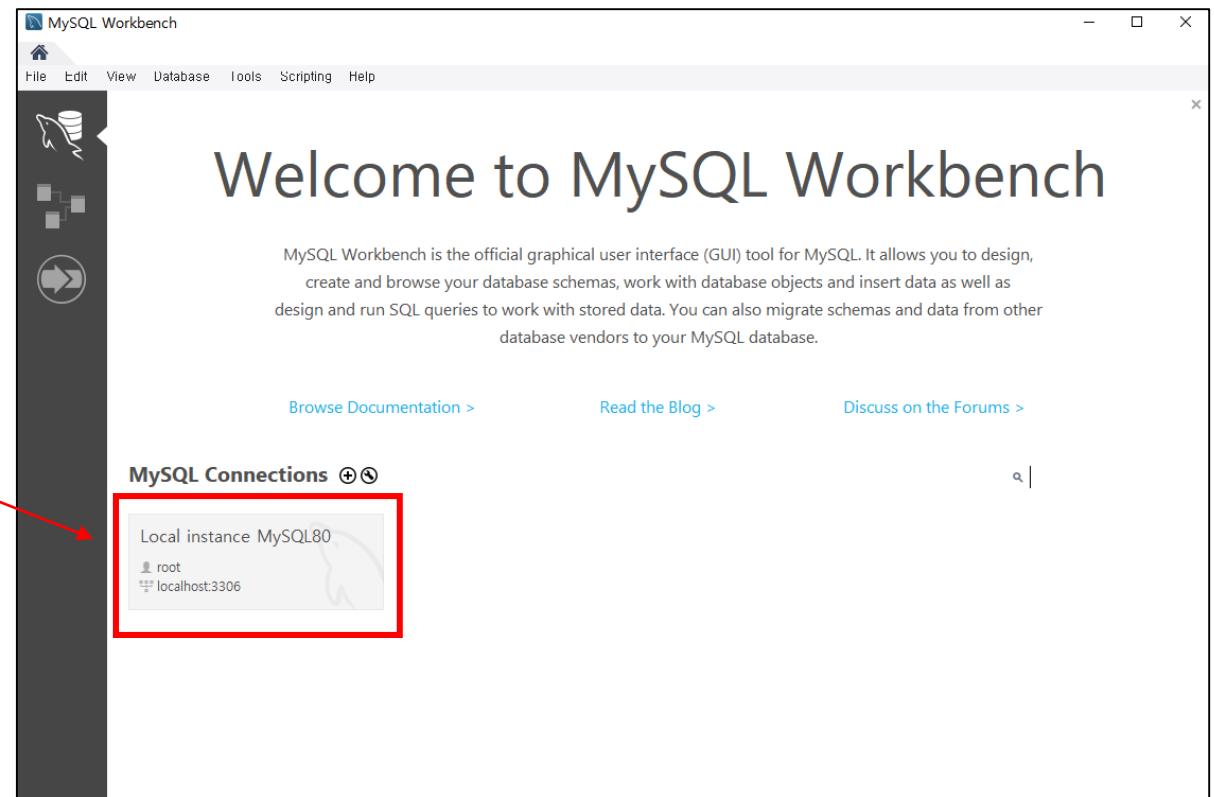
- 체크박스를 해제해야
- 시스템 부팅 시 자동시작 안됨
- 서비스로 가서
- 사용할때만 수동으로 시작



MySQL Workbench 동작 Test

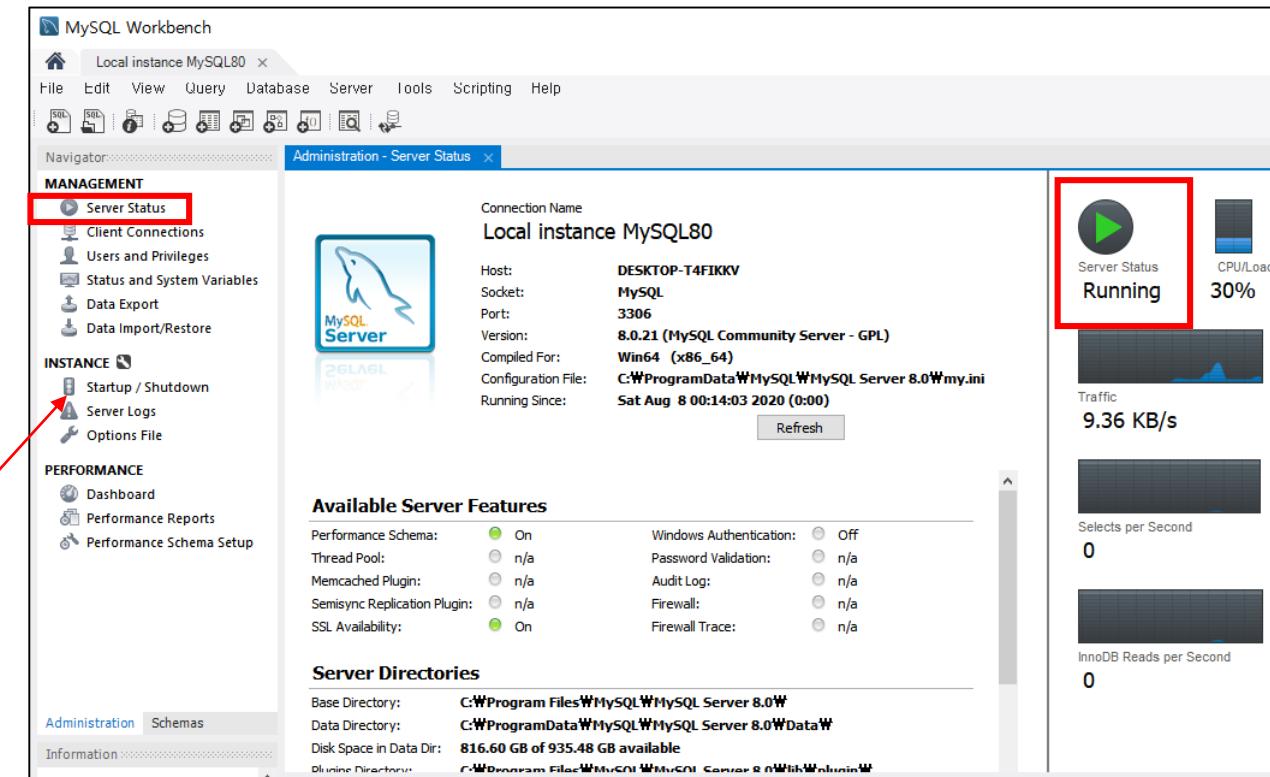
- 생성된 DB Instance에 접속할 수 있도록, 접속 정보가 등록 되어있음

설치 과정에서 추가된
DB Server (DB Instance)에
접속할 수 있다.



정상적으로 Local Server가 운영되고 있음을 확인할 수 있다.

- DB로 접속하여 값을 R/W 할 수 있는 상태

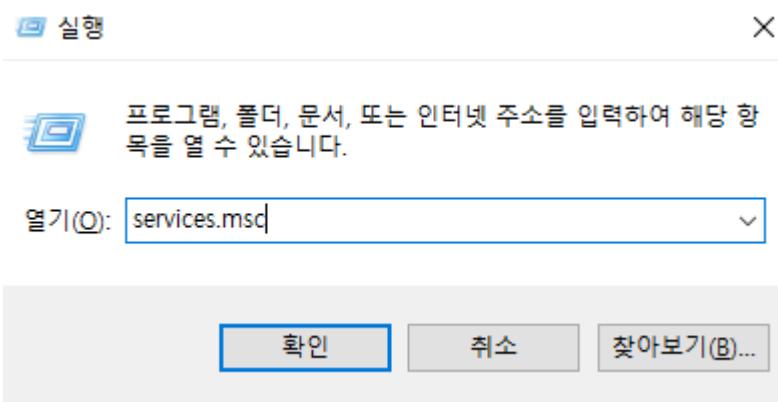


만약 서버가 중지되어있다면
Startup 해주면 된다.

컴퓨터 재시작 후 수동으로 시작

Confidential

- 윈도우 + r services.msc

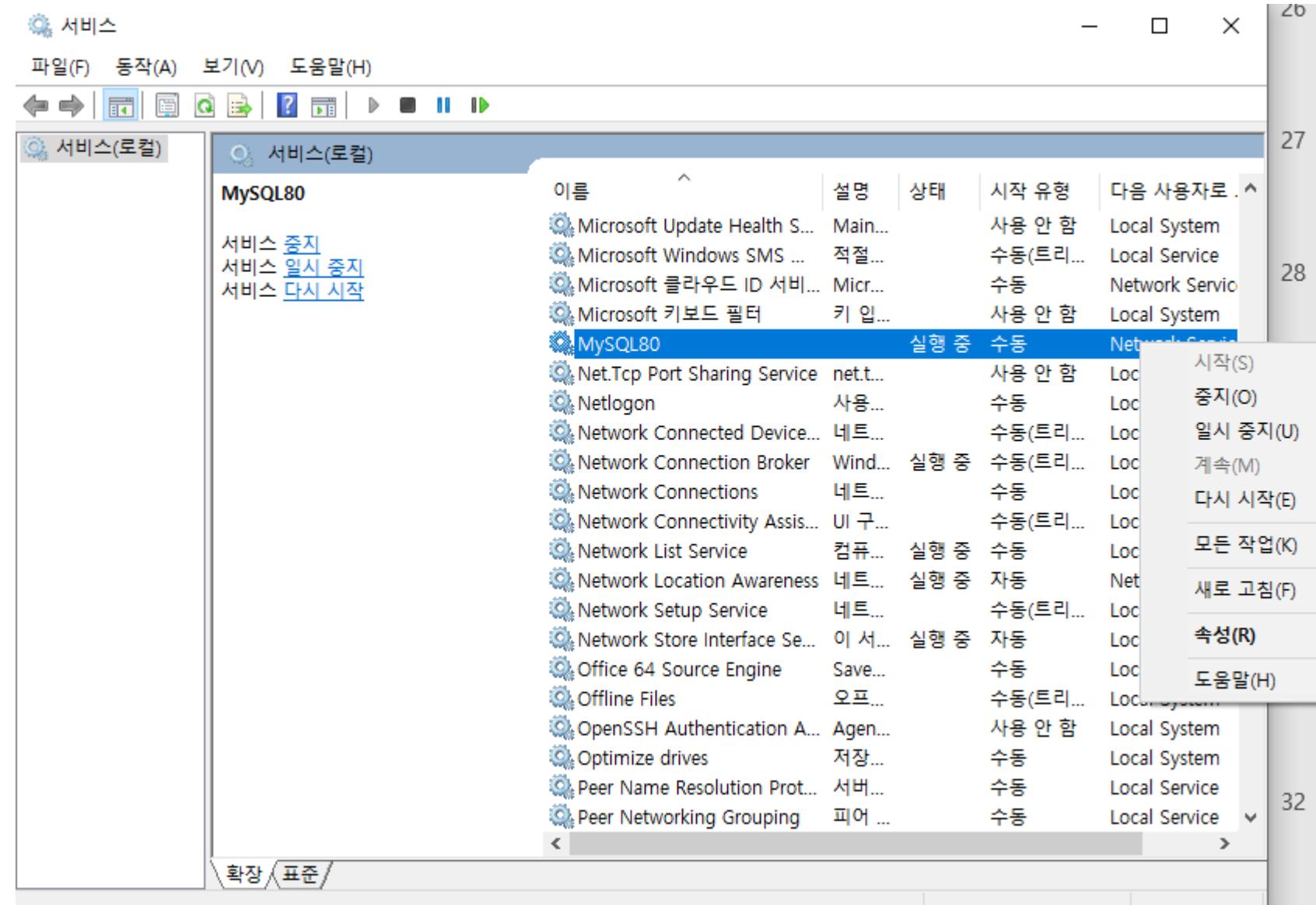


- 또는, 윈도우 검색창에 ‘서비스’

컴퓨터 재시작 후 수동으로 시작

Confidential

- MySQL80 찾은 후,
- 오른쪽버튼 눌러 수동으로 시작
- 이렇게 수동으로 조작해야
- 매번 컴퓨터 메모리를 소모 안함



2장. DB 세팅하기

챕터의 포인트

- MySQL 구조
- DB 계정 생성 및 접속
- Sample Table 생성하기

MySQL 구조

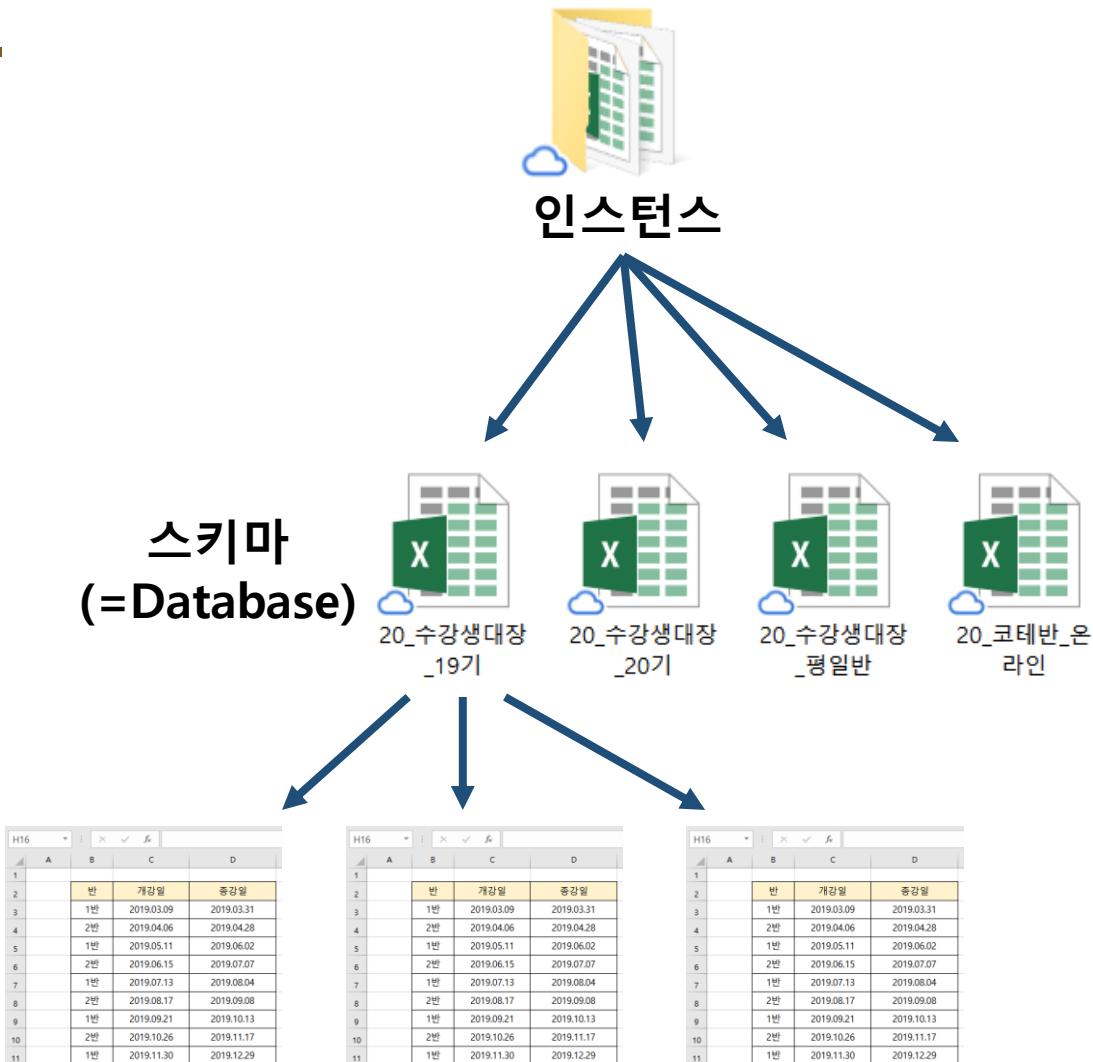
DB는 3 계층 구조로 이루어진다.

- MySQL 8 기준
 - 인스턴스 = DB 서버
 - 스키마 = Database
 - 테이블

- 유의사항

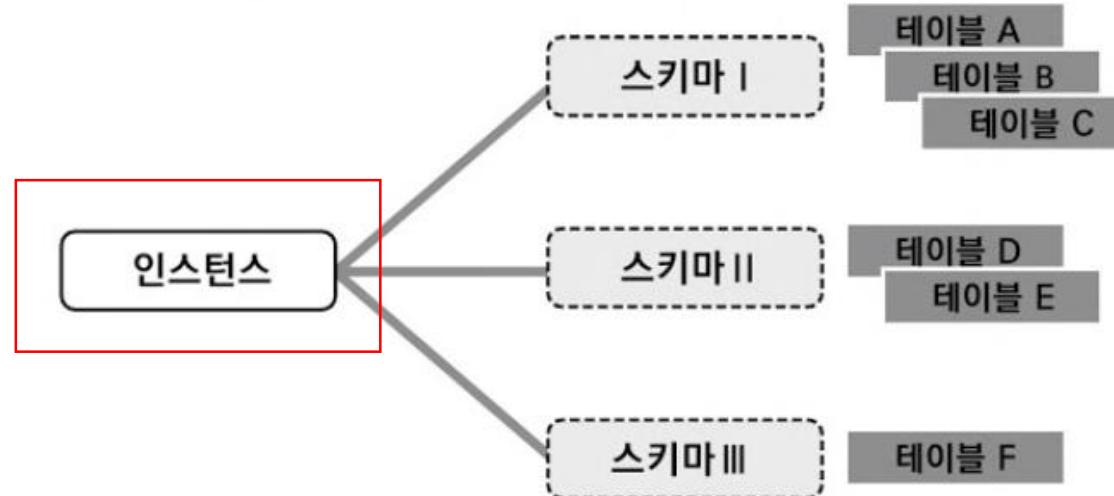
Oracle에서는 Database와 Schema 독립된 4 계층 구조로 되어있어 MySQL 구조와 혼동되기 쉽다.

Table



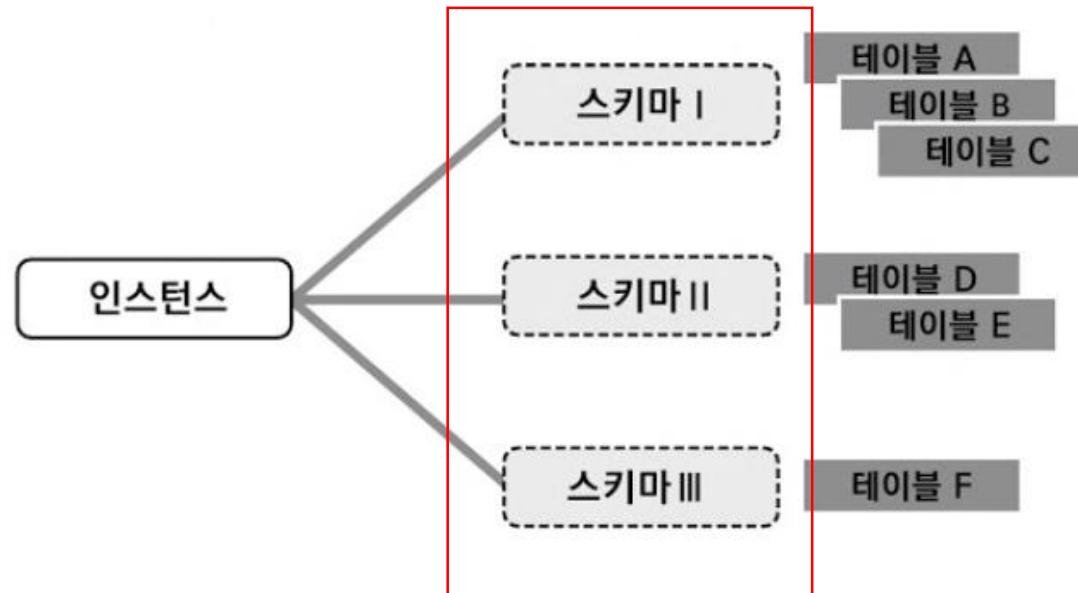
서버 인스턴스는 하나의 DB Server 를 나타낸다.

- DBMS를 설치되면 인스턴스가 자동 생성된다.
- 하나의 DB를 운영하기 위해 내부 Buffer / 내부 저장공간 / 관리 도구들이 동작되어야 한다. 운영이 필요한 모든 도구들을 모아 “서버 인스턴스” 라고 부른다.



スキ마는 Database와 동일한 뜻이다.

- 스키마 생성하는 SQL 명령어 1 : CREATE DATABASE minco;
- 스키마 생성하는 SQL 명령어 2 : CREATE SCHEMA minco;

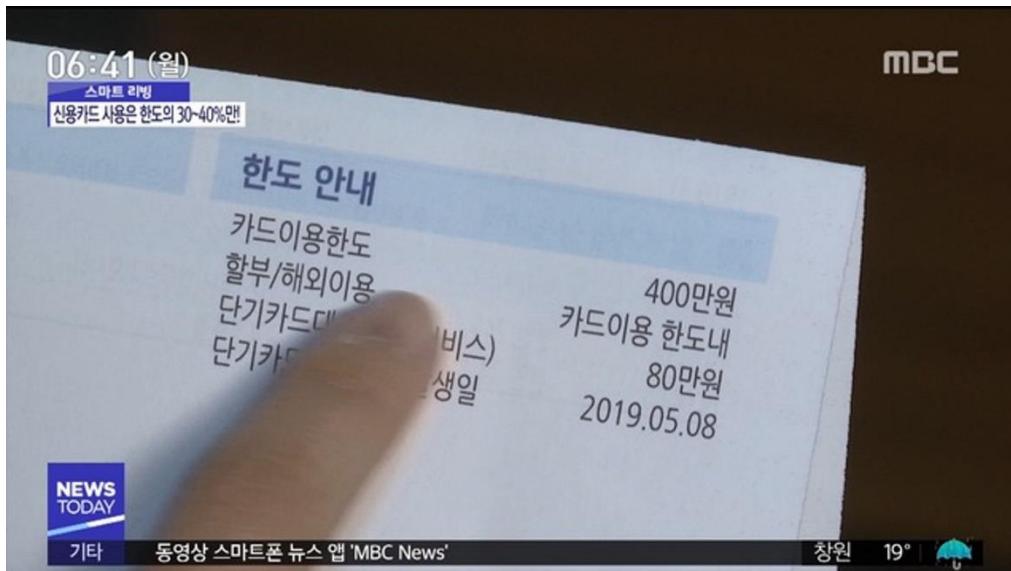


DB 계정 생성 및 접속

보안을 위해서, 모든 변경 권한이 있는 root 아이디를 쓰지 말자.

- 해킹 당했을 때, 피해를 최소화 하기 위함

- 사용자 계정 해킹 당하는 경우 : 권한 허용한도 내 해킹 피해를 입는다.
- root 계정 해킹 당하는 경우 : 해킹 피해 MAX



무제한 한도 카드를 도난당했을 때 피해
VS
카드이용한도 카드를 도난당했을 때 피해

root 계정 사용 하는 경우

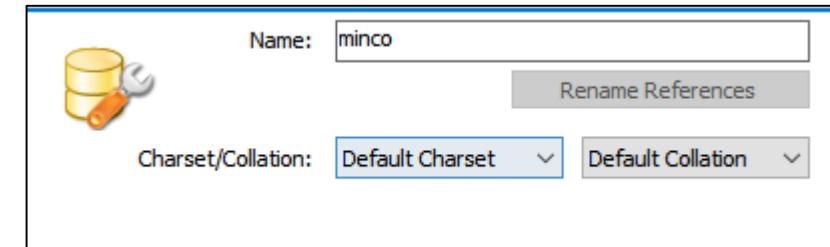
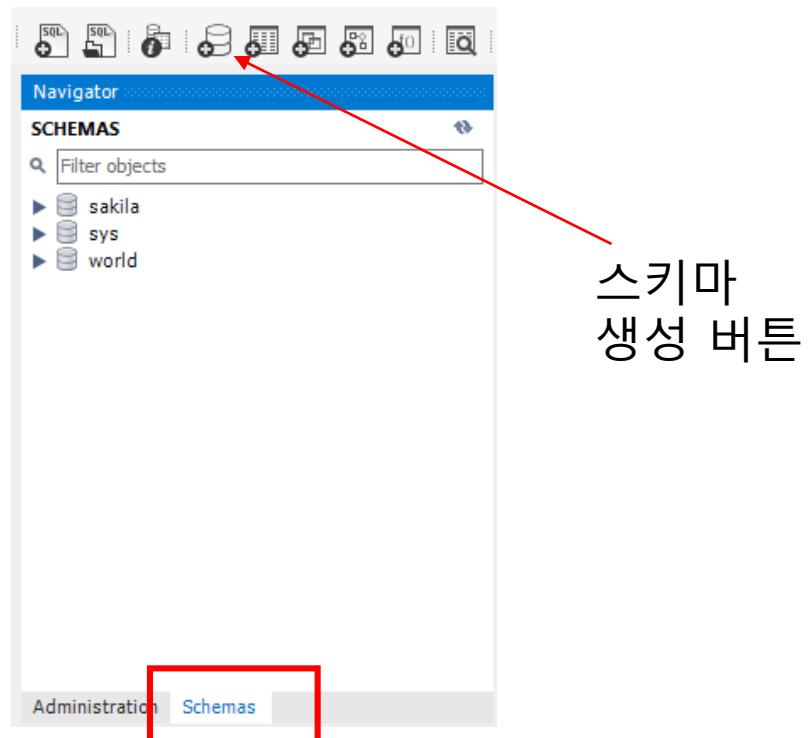
- 평소에 사용하지 않는다. 사용자 계정 권한이 없는 일을 할때만 접속하여 사용한다.
- 사용자 계정 추가 / 스키마 (Database) 생성 등

사용자 계정 사용 하는 경우

- 허용된 스키마 (Database)만 사용 할 권한이 부여된다.
- 허용된 스키마에 Table 추가 후 SQL 명령어 사용 가능
- 스키마 편집 / 제거 불가

사용자 계정이 사용할 스키마를 생성해두자.

- Navigator에서 Schemas 선택 후, Schema 추가
- Default Charset을 해도 utf-8로 자동 선택 된다.



새로운 계정을 만든다.

The screenshot shows the MySQL Workbench interface. On the left, the Navigator pane is visible with various management options like Server Status, Client Connections, and Users and Privileges. The 'Users and Privileges' option is highlighted with a red box. The main window title is 'Administration - Users and Privileges'. It displays a list of existing users: mysql.infoschema, mysql.session, mysql.sys, newuser, and root. A detailed view of the 'newuser' account is shown on the right, with tabs for Login, Account Limits, Administrative Roles, and Schema Privileges. The 'Login' tab is selected. The 'Login Name' field contains 'minco'. The 'Authentication Type' is set to 'Standard'. The 'Limit to Hosts Matching' field contains '%'. The 'Password' field is filled with '*****' and has a red box around it, with a note below saying 'Weak password.'. The 'Confirm Password' field also contains '*****'. A button for 'Expire Password' is at the bottom. The entire 'Password' and 'Confirm Password' input area is also enclosed in a red box.

Navigator: :::::::::::::::::::::

Query 1 Administration - Users and Privileges

MANAGEMENT

- Server Status
- Client Connections
- Users and Privileges**
- Status and System Variables
- Data Export
- Data Import/Restore

INSTANCE

- Startup / Shutdown
- Server Logs
- Options File

Local instance MySQL80

Users and Privileges

User Accounts

User	From Host
mysql.infoschema	localhost
mysql.session	localhost
mysql.sys	localhost
newuser	%
root	localhost

Details for account newuser@%

Tab	Description
Login	Login Name: minco You may create multiple accounts with the same login name to connect from different hosts.
Account Limits	Authentication Type: Standard For the standard password and/or host based authentication select 'Standard'.
Administrative Roles	Limit to Hosts Matching: % % and _ wildcards may be used
Schema Privileges	Password: ***** Weak password. Confirm Password: ***** Enter password again to confirm. Expire Password

생성된 계정이 사용할 스키마(DB)를 지정해준다.

- 기존에 있는 권한은 지운다.

The screenshot shows the MySQL Workbench 'Users and Privileges' interface. On the left, a sidebar lists user accounts: minco, mysql.infoschema, mysql.session, mysql.sys, and root. The main panel displays the 'Details for account minco@%' for the 'Schema Privileges' tab. A red box highlights the 'Schema Privileges' tab. Below it, a table shows the current schema privileges: '%' has 'none'. At the bottom right of the main panel, another red box highlights the 'Add Entry...' button. In the foreground, a modal dialog titled 'New Schema Privilege Definition' is open. It asks to select a schema for the user 'minco'. Three options are available: 'All Schema (%)', 'Schemas matching pattern:' (with a text input field), and 'Selected schema:' (with a dropdown menu). The 'Selected schema:' option is selected, and 'minco' is chosen from the dropdown. A red box highlights the 'Selected schema:' dropdown.

해당 스키마에 대해서, 모든 SQL 명령어 권한 주기

Details for account minco@%

Login Account Limits Administrative Roles Schema Privileges

Schema	Privileges
minco	ALTER, ALTER ROUTINE, CREATE, CREATE ROUTINE, CREATE TEMPORARY TABLES, CREATE VIEW, DELETE, DROP, EVENT, EXECUTE, INDEX, INSERT, LOAD, SELECT, TRIGGER, UPDATE

Schema and Host fields may use % and _ wildcards.
The server will match specific entries before wildcarded ones.

The user 'minco'@'%' will have the following access rights to the schema 'minco':

Object Rights

- SELECT
- INSERT
- UPDATE
- DELETE
- EXECUTE
- SHOW VIEW

DDL Rights

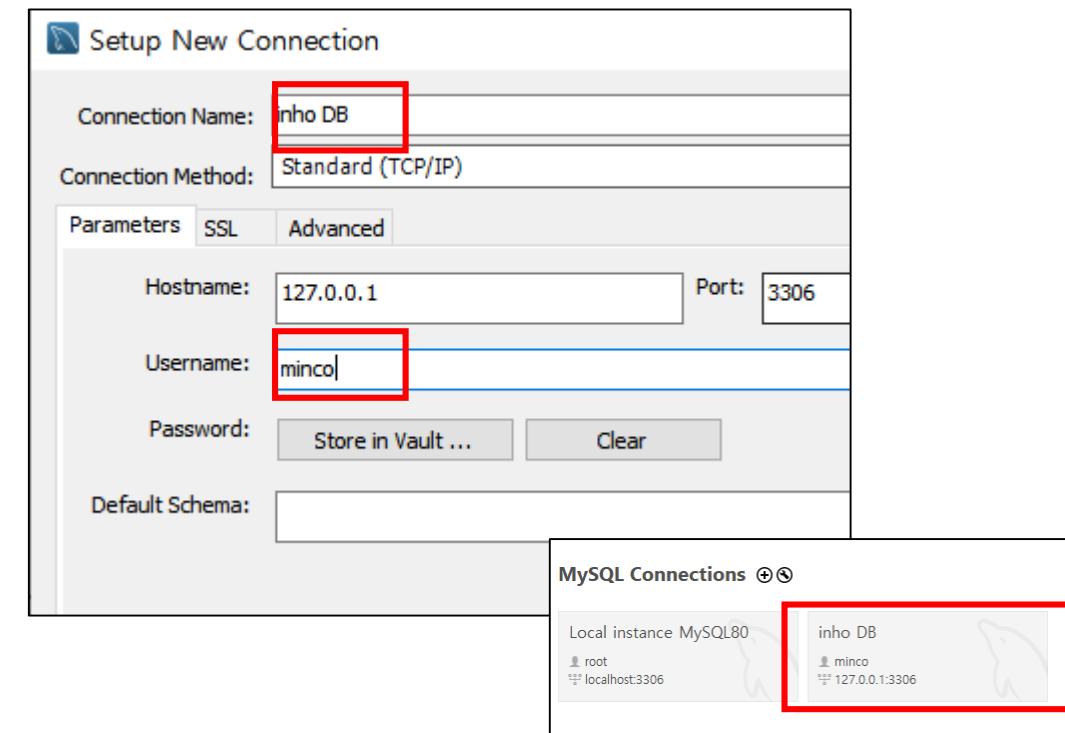
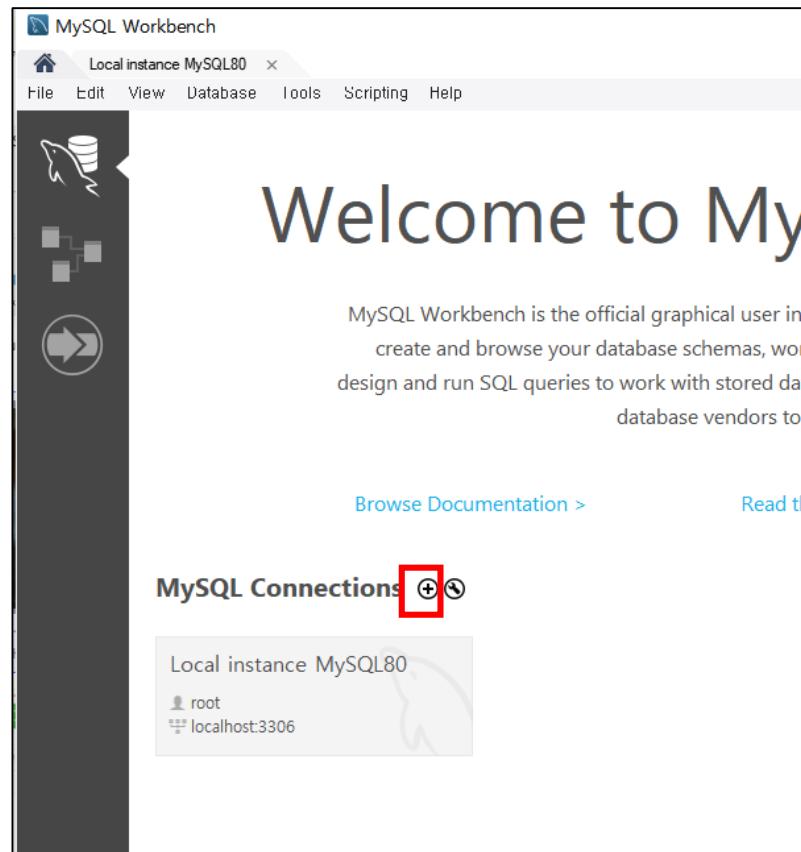
- CREATE
- ALTER
- REFERENCES
- INDEX
- CREATE VIEW
- CREATE ROUTINE
- ALTER ROUTINE
- EVENT
- DROP
- TRIGGER

Other Rights

- GRANT OPTION
- CREATE TEMPORARY TABLES
- LOCK TABLES

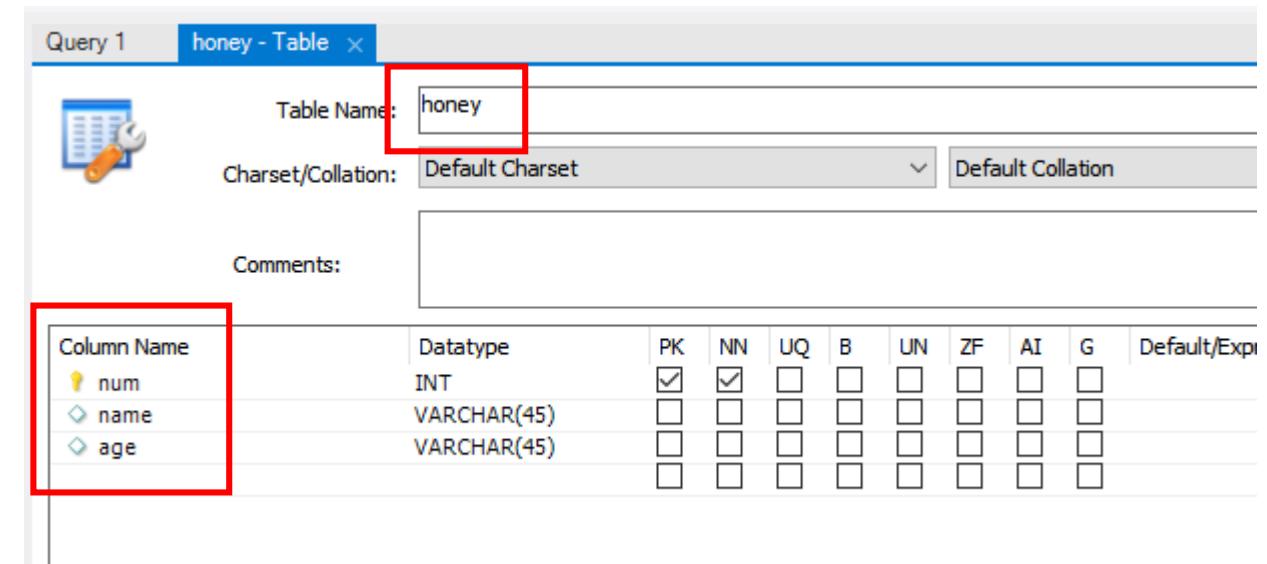
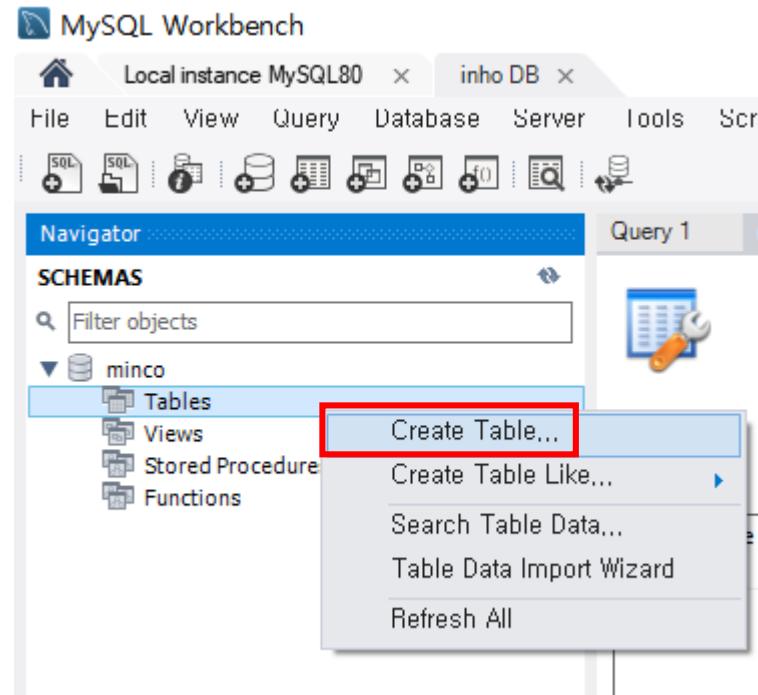
root를 종료하고, HOME으로 와서 Connection을 추가한다.

- + 버튼을 눌러, 사용자 계정으로 DB를 새롭게 접속한다.



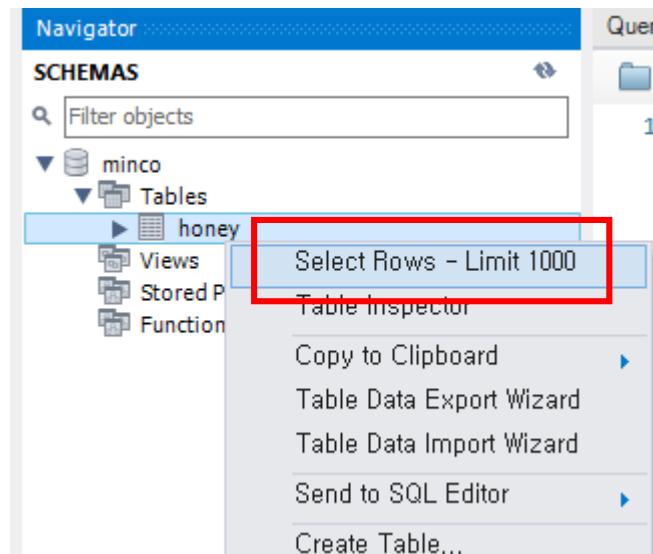
Sample Table 생성하기

Test 할 honey Table을 생성하자.



Navigator에서 생성된 honey를 확인 해보자

- Select Rows 클릭
- 생성된 Table 확인 가능

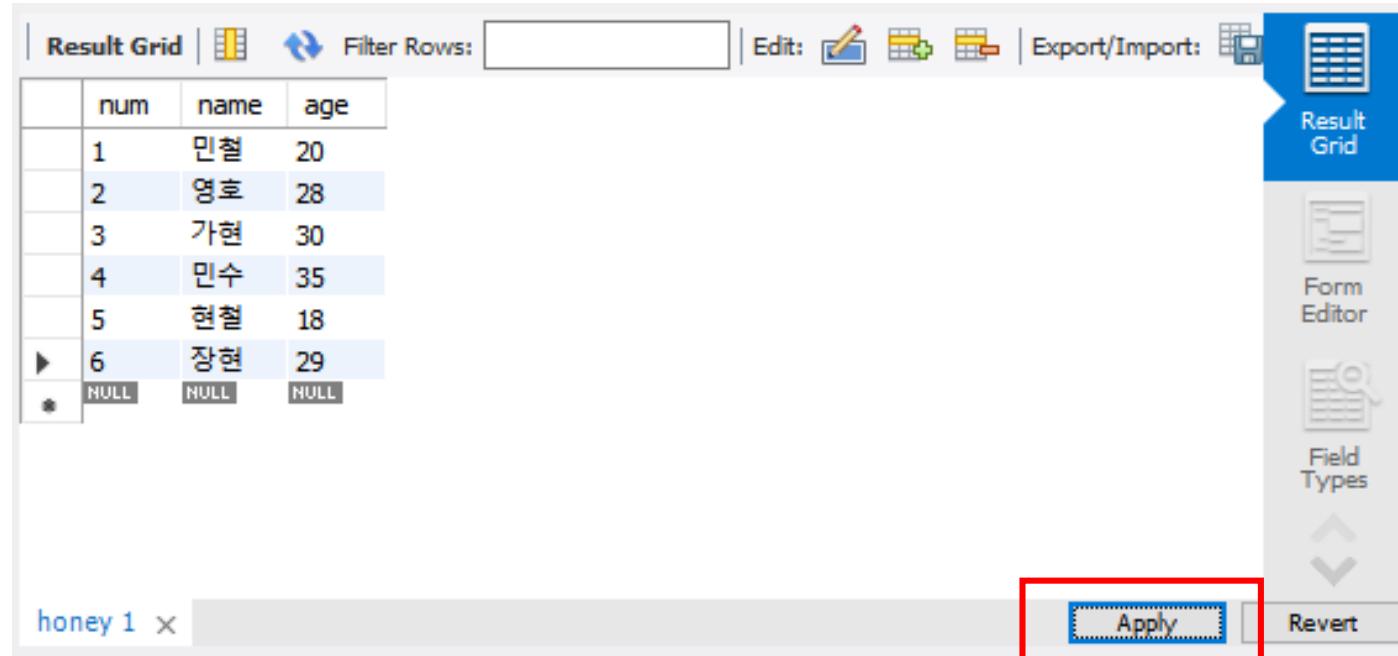


*	num	name	age
*	NULL	NULL	NULL
*	NULL	NULL	NULL
*	NULL	NULL	NULL

아직 값을 넣지 않은 상태

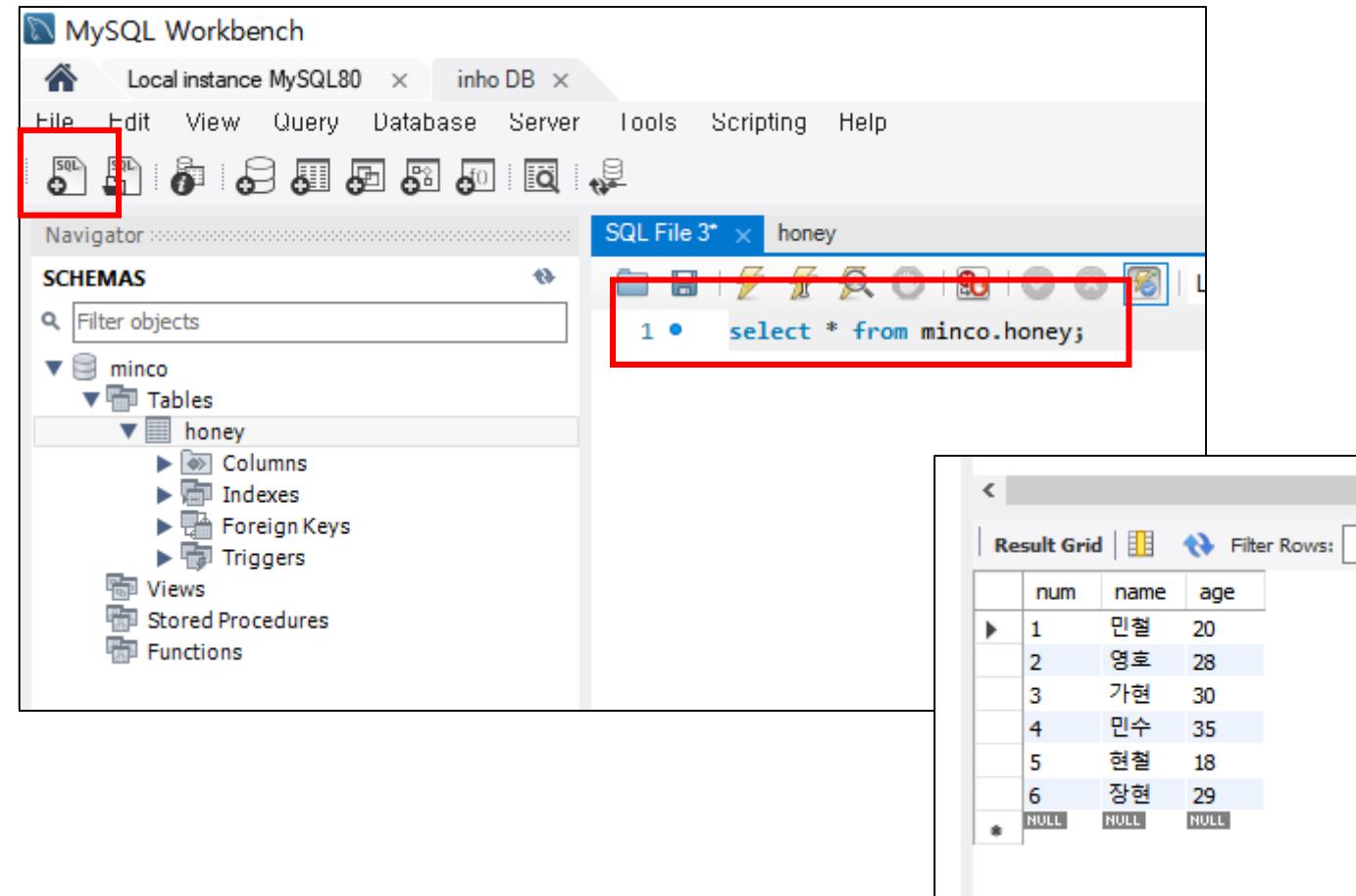
샘플 데이터를 입력 후 Apply 클릭

- TAB 키를 누르면 빠르게 입력이 가능



SQL File을 하나 생성하기

- select * from minco.honey 입력
- 번개 아이콘을 눌러 결과 확인하기
- execute 단축키 : **Ctrl + Shift + Enter**



접속하기

- https://www.w3schools.com/mysql/trysql.asp?filename=trysql_select_all

3장. Select (훈련 Level 1)

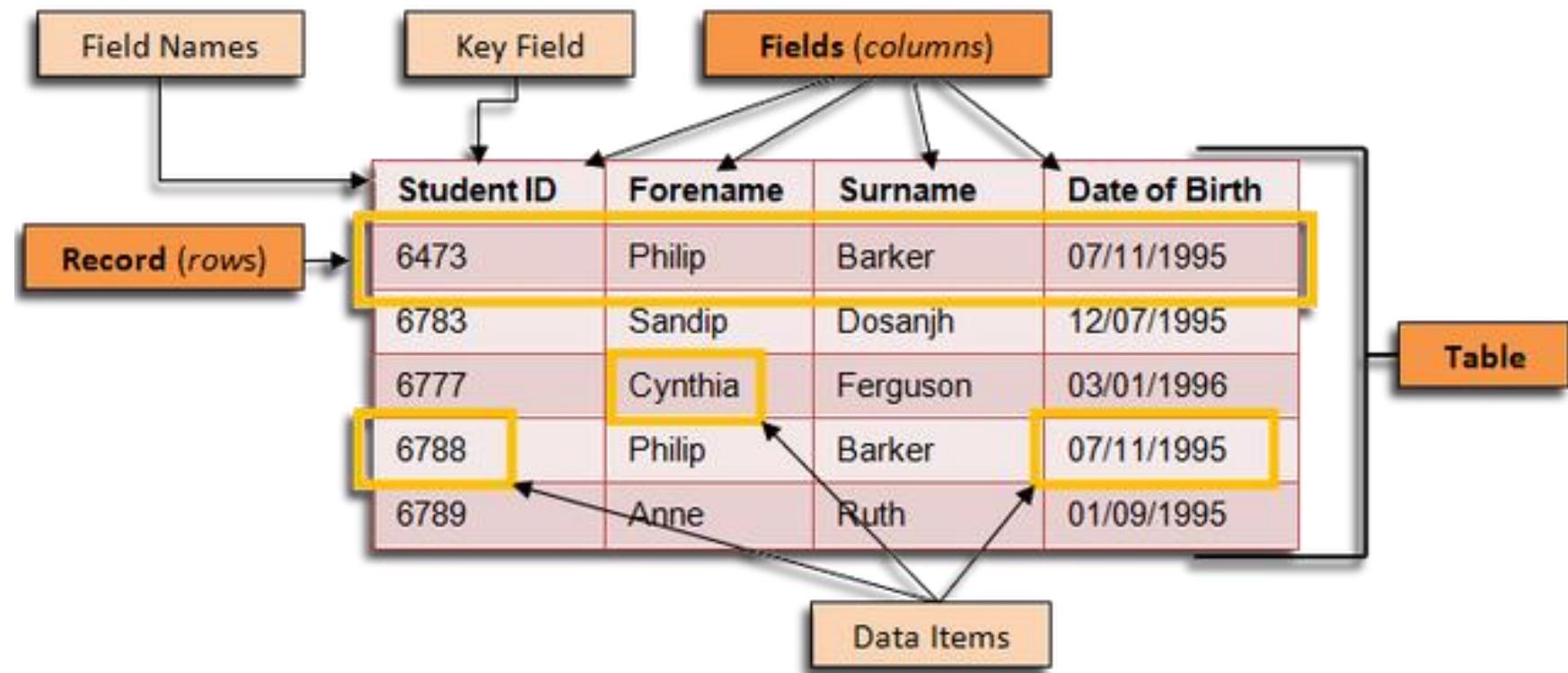
챕터의 포인트

- Select 연습 - Level 1
- DB Backup

Select 연습 - Level 1

세 가지 용어들을 암기하자

- Field (columns)
- Record
- Items



SELECT 필드명 FROM 테이블

- 모든 필드를 출력하고 싶을 때는 * 을 쓰면 된다.

SELECT * FROM TABLE

	num	name	age
▶	1	민철	20
	2	영호	28
	3	가현	30
	4	민수	35
	5	현철	18
	6	장현	29
*	NULL	NULL	NULL

```
select * from minco.honey;
```

	name	age
▶	민철	20
	영호	28
	가현	30
	민수	35
	현철	18
	장현	29

```
select name, age from minco.honey;
```

WHERE 절 사용하기

- SELECT * FROM TABLE WHERE 조건1 AND 조건2 AND 조건3 ...
- 특정 조건에 해당하는 레코드만 출력할 때 사용
 - 특정 필드만 출력 : SELECT [어떤 필드?] FROM TABLE
 - 특정 레코드만 출력 : SELECT * FROM TABLE WHERE [어떤 레코드?]

	num	name	age
▶	2	영호	28
	3	가현	30
	4	민수	35
*	NULL	NULL	NULL

```
select * from minco.honey where num >= 2 and num <= 4
```

	num	name	age
▶	2	영호	28
*	NULL	NULL	NULL

```
select * from minco.honey where name='영호'
```

SELECT 123 + 45

123 + 45

168

SELECT 'SSAFY' + 123

'SSAFY' + 123

123

```
SELECT OrderDetailID, ProductID, OrderDetailID + ProductID FROM OrderDetails;
```

Number of Records: 2155

OrderDetailID	ProductID	OrderDetailID + ProductID
1	11	12
2	42	44
3	72	75
4	14	18
5	51	56
6	41	47
7	51	58

조건문에서 AND

```
SELECT * FROM Orders WHERE CustomerID > 33 and EmployeeID < 5
```

Number of Records: 331

OrderID	CustomerID	EmployeeID	OrderDate	ShipperID
10250	34	4	1996-07-08	2
10251	84	3	1996-07-08	1
10252	76	4	1996-07-09	2
10253	34	3	1996-07-10	2
10256	88	3	1996-07-15	2
10257	35	4	1996-07-16	3
10260	55	4	1996-07-19	1

- 조건문에서 OR

```
SELECT * FROM Orders WHERE OrderID < 10000 or ShipperID < 5
```

Number of Records: 830

OrderID	CustomerID	EmployeeID	OrderDate	ShipperID
10248	90	5	1996-07-04	3
10249	81	6	1996-07-05	1
10250	34	4	1996-07-08	2
10251	84	3	1996-07-08	1
10252	76	4	1996-07-09	2
10253	34	3	1996-07-10	2
10254	14	5	1996-07-11	2

다음 미션을 순서대로 수행하자

1. 모든 필드 출력 + num이 3, 5번 레코드만 출력하기
2. name, age 필드 출력하고, $2 \leq \text{num} \leq 5$ 번 레코드만 출력하기 (조건에 AND 사용)
3. 나이가 20대인 친구들만 출력 (모든 필드 출력)
4. num이 짝수이면서, age는 홀수인 친구들만 출력 (사람 이름만 출력)

- 다음 미션을 순서대로 수행하자

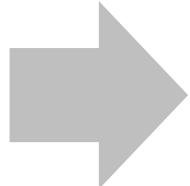
1. OrderDetails의 ProductID가 14보다 큰 모든 필드 출력
2. CustomerName, city 필드 출력 $10 < \text{CustomerID} \leq 50$ 레코드 출력 Customers에 데이터 존재
3. Orders에 CustomerId 가 30번대만 출력 (모든 필드 출력, and 사용)
4. ProductID가 짹수이면서, SupplierId 홀수인 ProductName과 ProductID 출력 (Products)

alias 지정하여 필드 이름 변경

- SELECT 필드명 AS [별칭] FROM TABLE

name	
민철	
영호	
가현	
민수	
현철	
장현	

```
select name from minco.honey
```



이름	
민철	
영호	
가현	
민수	
현철	
장현	

```
select name as '이름' from minco.honey
```

특정 필드를 정렬한다

- 오름차순 : SELECT * FROM TABLE ORDER BY [필드명]
- 내림차순 : SELECT * FROM TABLE ORDER BY [필드명] DESC;

	num	name	age
▶	5	현철	18
	1	민철	20
	2	영호	28
	6	장현	29
	3	가현	30
	4	민수	35
*	NULL	NULL	NULL

```
select * from minco.honey order by age;
```

	num	name	age
▶	4	민수	35
	3	가현	30
	6	장현	29
	2	영호	28
	1	민철	20
	5	현철	18
*	NULL	NULL	NULL

```
select * from minco.honey order by age desc;
```

원하는 데이터 개수만큼만 가져온다.

- LIMIT 가져올 개수 또는 LIMIT 건너뛸 개수, 가져올 개수

```
SELECT * FROM minco.honey LIMIT 3;
```

```
SELECT * FROM minco.honey LIMIT 3, 3;
```

다음 설명에 맞는 Query 작성하기

1. num을 “번호”, name을 “성함”, age를 “나이”로 출력 + 나이 오름차순으로 정렬
2. 이름 사전 순서대로(오름차순) 나이, 이름만 출력하기

	번호	성함	나이
▶	5	현철	18
	1	민철	20
	2	영호	28
	6	장현	29
	3	가현	30
	4	민수	35

age	name
30	가현
35	민수
20	민철
28	영호
29	장현
18	현철

다음 설명에 맞는 Query 작성하기

1. Customers에 Address를 “주소”, city를 “도시”, PostalCode를 “우편번호”로 출력, 우편번호 내림차순으로 정렬
2. OrderDetails에 OrderDetailID를 기준으로 내림차순으로 정렬하고, 5개만 출력하기 (모든 컬럼)
3. Orders에 모든 컬럼을 OrderId를 기준으로 내림차순 정렬후, 3개를 건너뛰고 7개만큼 출력하기

SELECT

- OrderDetailID AS '상세주문번호',
- OrderID AS '주문번호',
- ProductID AS '상품번호',
- Quantity AS '수량'
- FROM OrderDetails
- WHERE OrderID <= 10000 OR ProductID >
- ORDER BY OrderDetailID DESC
- LIMIT 1, 8;

Result:

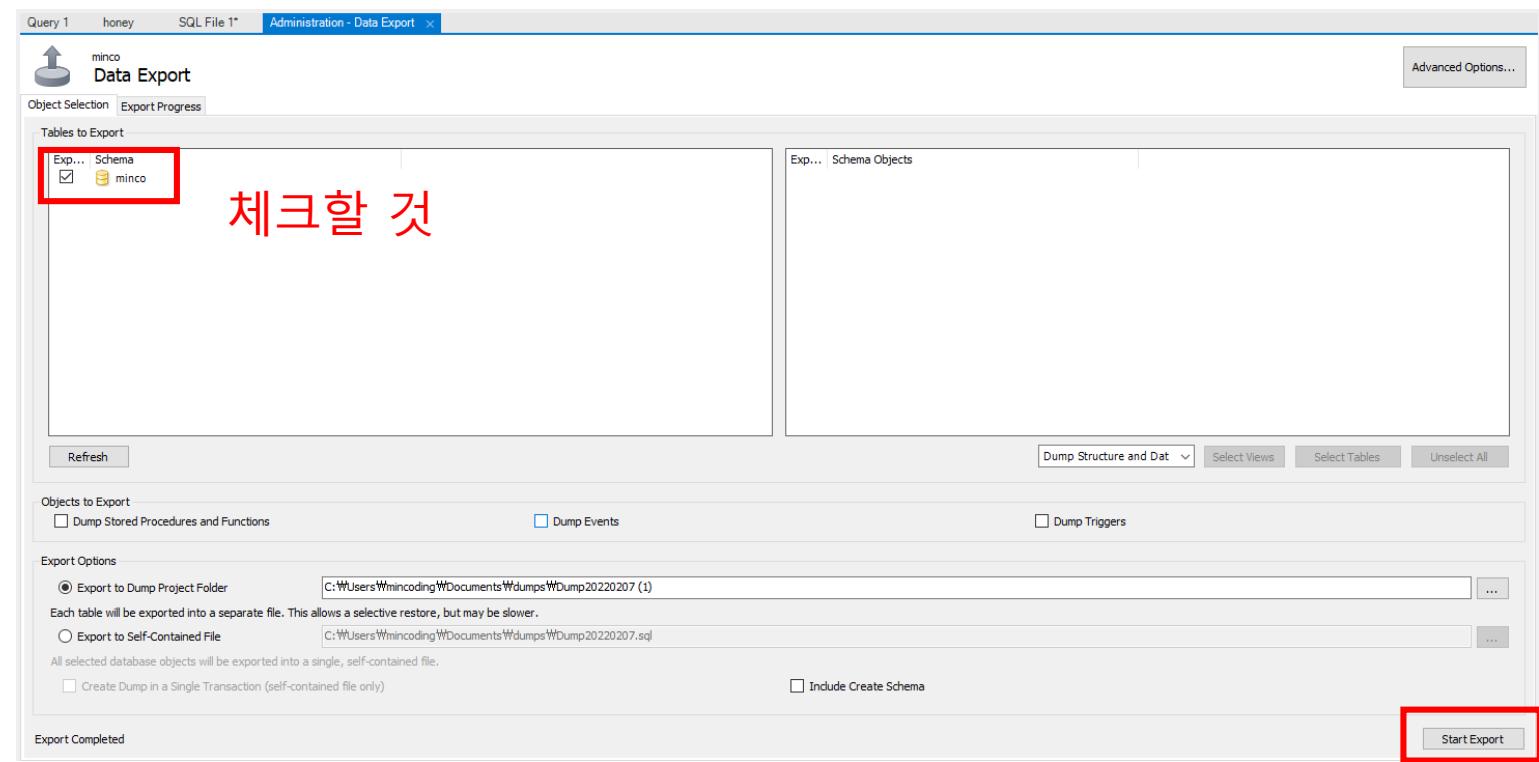
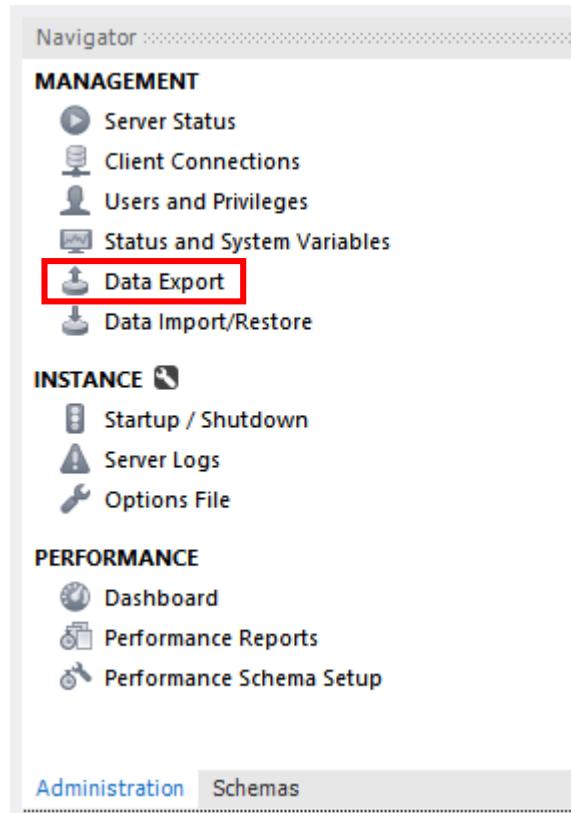
Number of Records: 8

상세주문번호	주문번호	상품번호	수량
2154	11077	75	4
2153	11077	73	2
2152	11077	66	1
2151	11077	64	2
2150	11077	60	2
2149	11077	55	2
2148	11077	52	2
2127	11075	76	2

DB Backup

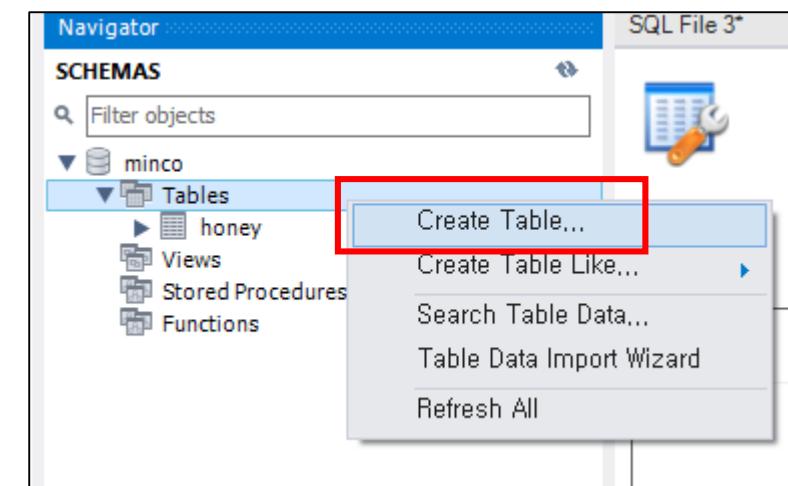
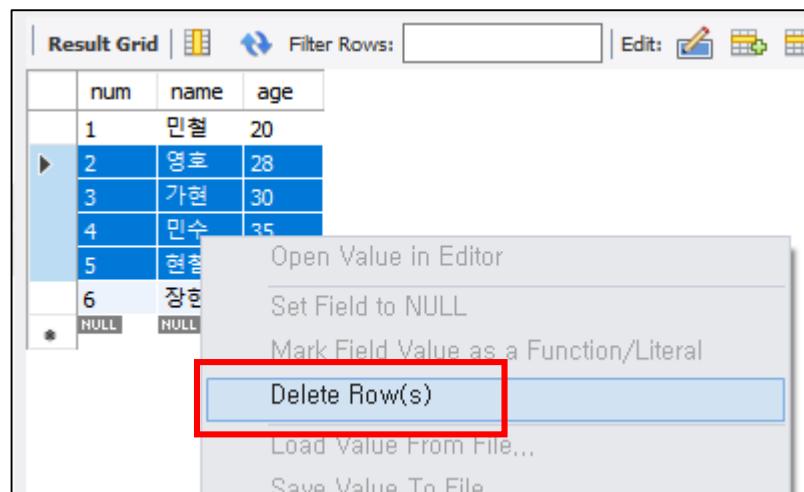
Data Export

- 현재 Data를 외부 파일로 추출한다.



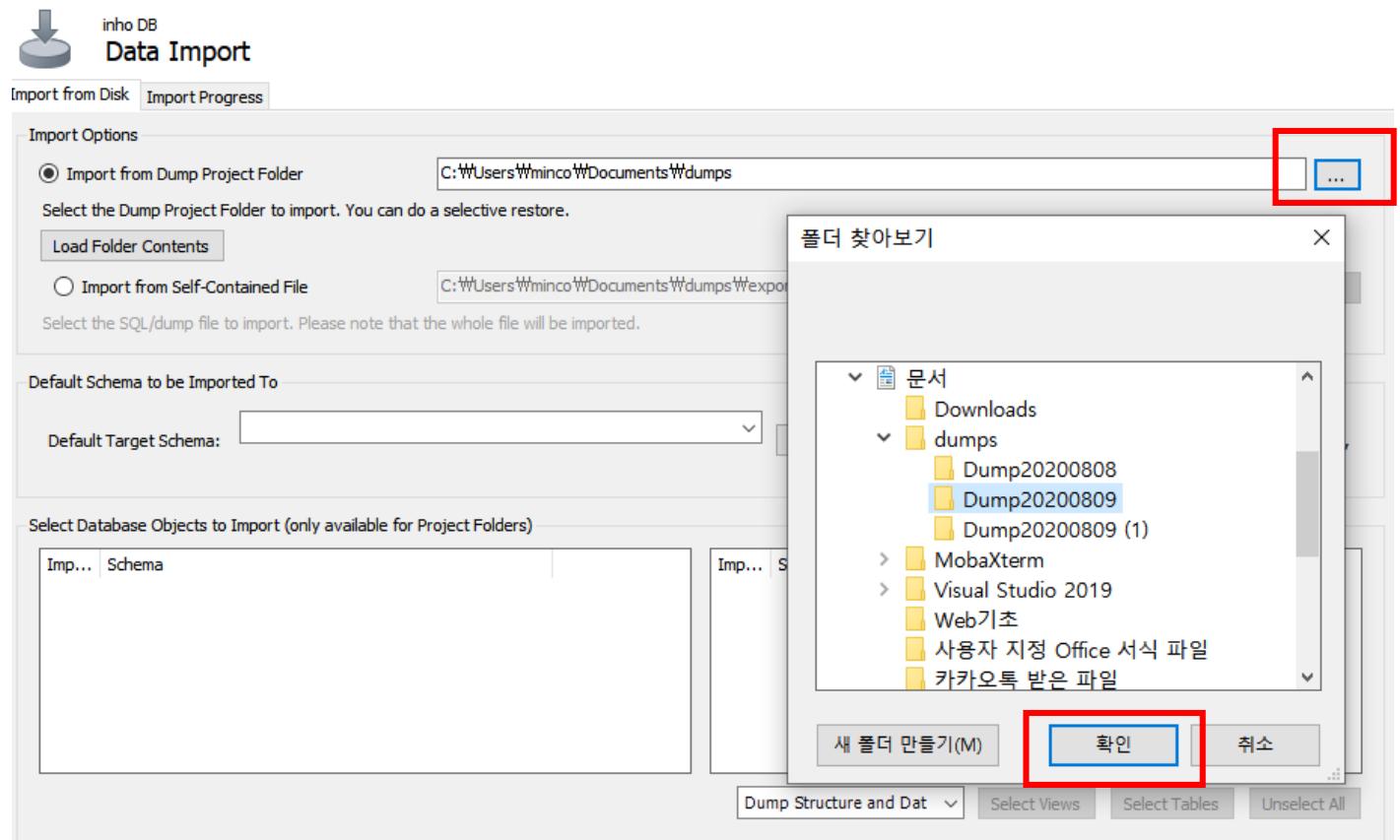
데이터 변경

- 데이터를 변경한 후, Restore 해보자
 - Honey Table에서 레코드 추가 및 제거
 - 새로운 Table 추가하기



데이터 복구 하기

- 제거된 데이터는 복구가 된다.
- 복구 전 추가된 Table은 삭제하지 않는다.



4장. CRUD와 Console

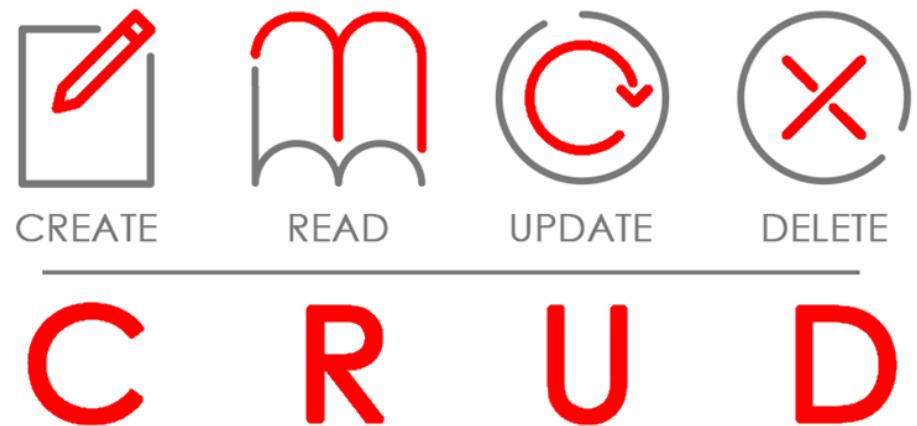
챕터의 포인트

- CRUD
- MySQL Console

CRUD

CRUD : Create + Read + Update + Delete

- 데이터를 다루는 Software의 기본적인 인터페이스를 뜻함
- Database에서 CRUD (암기 추천)
 - C : INSERT INTO ~ VALUES
 - R : SELECT
 - U : UPDATE ~ SET WHERE
 - D : DELETE FROM ~ WHERE



웹서비스에서 해당 Query를 쓰는 이유에 대해 생각해보자.

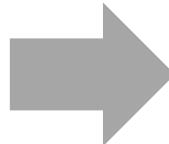
- 로그인 : SELECT
- 회원가입 : SELECT, INSERT, UPDATE
- 회원탈퇴 : DELETE
- 게시판은 CRUD 전체 사용

INSERT INTO 테이블 VALUES (값1, 값2, 값3 ...)

- 값 (7번, 이순신, 20세)을 추가하자.
- 값 (8번, 박중박, 50세)를 추가하자.

```
insert into minco.honey value(8, '박중박', 50);
```

	num	name	age
▶	1	민철	20
	2	영호	28
	3	가현	30
	4	민수	35
	5	현철	18
	6	장현	29
*	NULL	NULL	NULL

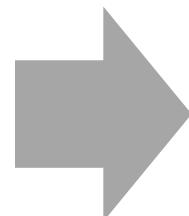


	num	name	age
▶	1	민철	20
	2	영호	28
	3	가현	30
	4	민수	35
	5	현철	18
	6	장현	29
	7	이순신	20
	8	박중박	50
*	NULL	NULL	NULL

UPDATE 테이블 SET 필드=값 WHERE 레코드 조건

- 값을 수정할 때는 통상적으로 레이블 + 필드를 모두 지정해준다.
- '이순신' 나이를 320세로 수정하자.
- '박중박'의 이름을 '박재훈'으로 수정하자.

	num	name	age
▶	1	민철	20
	2	영호	28
	3	가현	30
	4	민수	35
	5	현철	18
	6	장현	29
	7	이순신	20
*	8	박중박	50
	NULL	NULL	NULL



	num	name	age
▶	1	민철	20
	2	영호	28
	3	가현	30
	4	민수	35
	5	현철	18
	6	장현	29
	7	이순신	320
*	8	박재훈	50
	NULL	NULL	NULL

```
UPDATE minco.honey SET age=320 WHERE num = 7;  
UPDATE minco.honey SET name='박재훈' WHERE num = 8;  
SELECT * FROM minco.honey;
```

Safe update mode

- 다수의 레코드들이 한꺼번에 UPDATE 되는 사고를 막기 위함
- 설정에서 해당 Mode를 OFF 할 수 있음
- Safe mode Error가 발생되는 케이스
 - 조건문을 붙이지 않는 경우
 - WHERE문에 Key column 조건이 없는 경우

```
UPDATE minco.honey SET age=320;
```

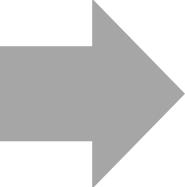
```
UPDATE minco.honey SET age=320 where name='이순신';
```

DELETE FROM 테이블 WHERE 레코드 조건

- 7번 레코드 삭제
- 8번 레코드 삭제

```
delete from minco.honey where id=7;  
delete from minco.honey where id=8;
```

	num	name	age
▶	1	민철	20
	2	영호	28
	3	가현	30
	4	민수	35
	5	현철	18
	6	장현	29
	7	이순신	320
	8	박재훈	50
*	NULL	NULL	NULL



	num	name	age
▶	1	민철	20
	2	영호	28
	3	가현	30
	4	민수	35
	5	현철	18
	6	장현	29
*	NULL	NULL	NULL

CRUD 명령어 Format을 암기 해두자

- INSERT INTO 테이블 VALUES()
- UPDATE 테이블 SET 필드 WHERE
- DELETE FROM 테이블 WHERE

제한시간 : 10 분

1. 모든 레코드 출력
2. NUM 2 번과 4 번 레코드의 나이를 모두 15로 수정
모든 레코드 출력
3. 1번 레코드 삭제 후
모든 레코드 출력
4. 레코드 추가 (50, 촉촉, 100)
모든 레코드 출력

MySQL Console

GUI 환경 : MySQL Workbench

- 대량 데이터에도, 구조를 쉽게 파악할 수 있다.
- 사용하기 편리한 장점

CLI 환경 : MySQL Console

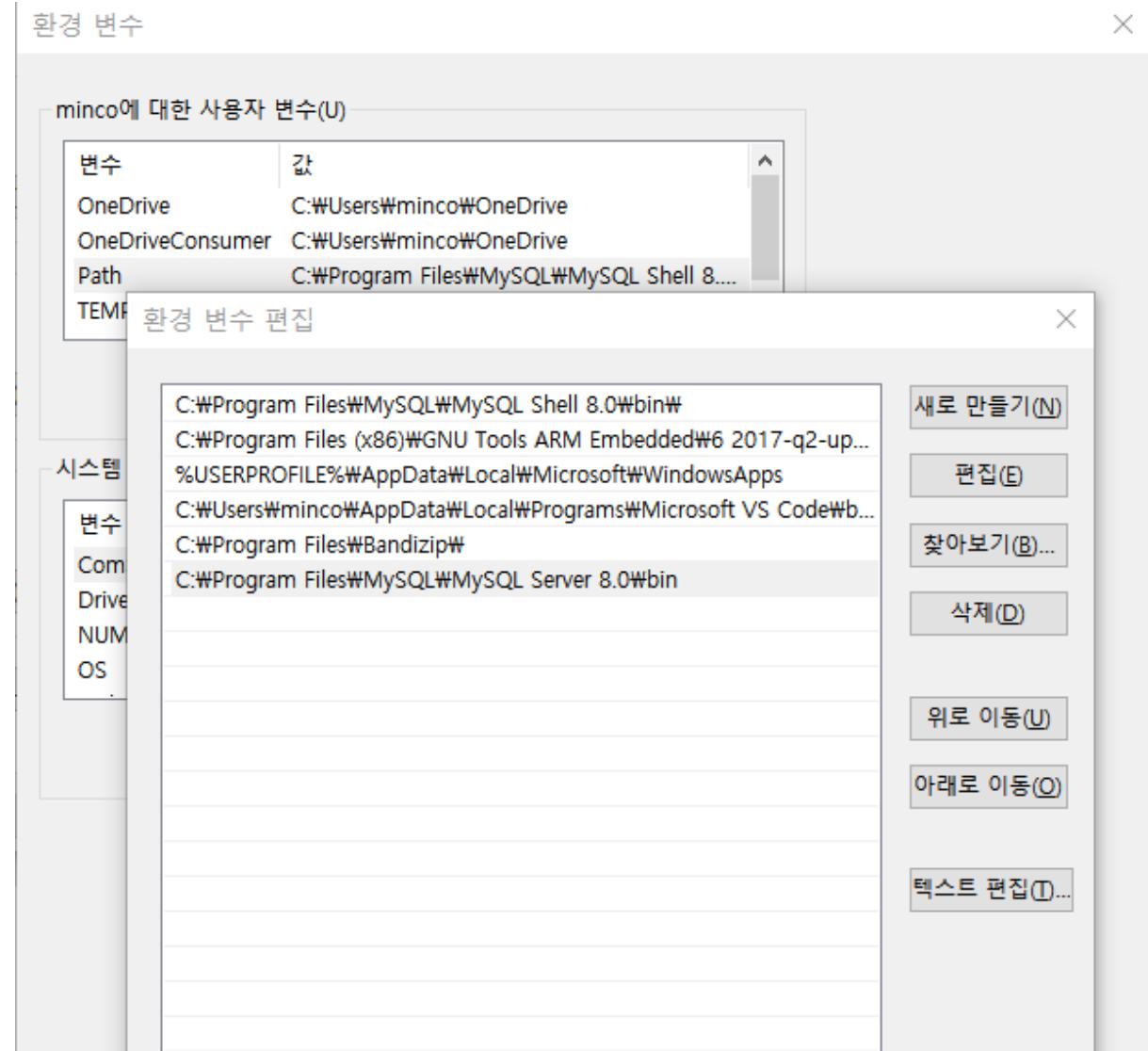
- Linux + DB Server + SSH에서 DB 초기설정에 사용된다.

MySQL을 Console에서 다룬다.

- Path 등록을 수동으로 해 주어야 한다.

- Windows 경로
C:\Program Files\MySQL\MySQL Server 8.0\bin

- Mac에서 mysql 위치
cd /usr/local/mysql/bin



명령 프롬프트 창을 띄운 후 mysql 입력

- 해당 에러메세지가 뜨면 path 등록이 잘 된 것이다.

```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.18363.959]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\minco>mysql
ERROR 1045 (28000): Access denied for user 'ODBC'@'localhost' (using password: NO)

C:\Users\minco>
```

mysql -u [계정아이디] -p

```
C:\Users\minco>
C:\Users\minco>mysql -u minco -p
Enter password: ****■
```

```
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 76
Server version: 8.0.21 MySQL Community Server - GPL

Copyright (c) 2000, 2020, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

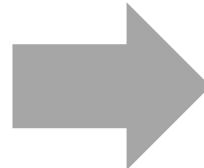
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

show databases / use [DB명]

- 현 계정이 접근 권한이 있는 Database 목록을 확인한다.
- 사용할 Databases 선택 명령어 : use [DB이름]

```
mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| minco |
+-----+
2 rows in set (0.01 sec)
```



```
mysql> use minco
Database changed
```

show tables;

- 테이블 목록 확인

desc [테이블명]

- 테이블 구조 확인

```
mysql> show tables;
+-----+
| Tables_in_minco |
+-----+
| honey           |
+-----+
1 row in set (0.01 sec)
```

```
mysql> desc honey
-> ;
+-----+-----+-----+-----+-----+-----+
| Field | Type   | Null | Key  | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| num   | int    | NO   | PRI  | NULL    |       |
| name  | varchar(45) | YES  |       | NULL    |       |
| age   | varchar(45) | YES  |       | NULL    |       |
+-----+-----+-----+-----+-----+-----+
3 rows in set (0.00 sec)
```

UPDATE시 실수로 Where를 쓰지 않으면 데이터가 한꺼번에 변경 됨

- Workbench와 달리, Safe Update Mode가 없다.
- update honey set age = 77 수행

```
mysql> update honey set age = 77;
Query OK, 6 rows affected (0.01 sec)
Rows matched: 6  Changed: 6  Warnings: 0

mysql>
mysql> select * from honey;
+---+---+---+
| num | name | age |
+---+---+---+
| 1 | 민철 | 77 |
| 2 | 영호 | 77 |
| 3 | 가현 | 77 |
| 4 | 민수 | 77 |
| 5 | 현철 | 77 |
| 6 | 장현 | 77 |
+---+---+---+
6 rows in set (0.01 sec)

mysql>
```

제한시간 : 10 분

1. 모든 레코드 출력
2. 6 번 레코드 삭제 후 모든 레코드 출력
3. 레코드 추가 (99, 동동주, 50)
모든 레코드 출력
4. Workbench 에도 적용이 되어있는지 확인 해 본다.

5장. Select (훈련 Level 2)

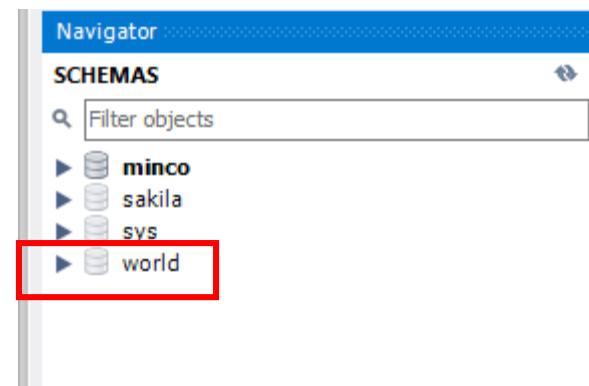
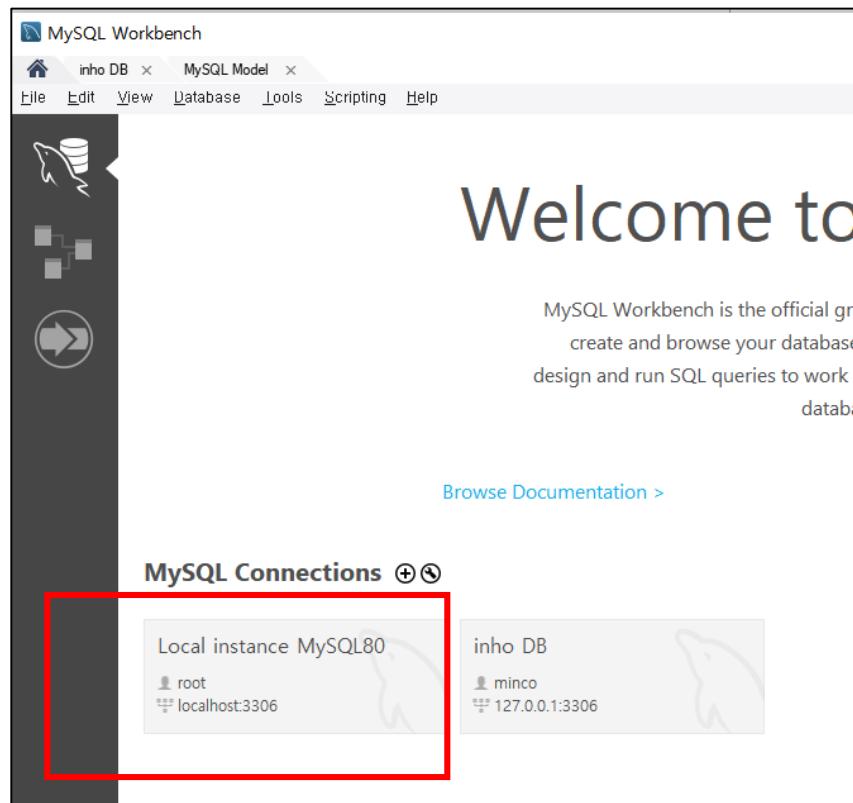
챕터의 포인트

- 샘플 스키마 준비
- BETWEEN / IN / LIKE
- GROUP BY / HAVING

샘플 스키마 준비

root 권한으로 접속하기

- world 스키마를 사용자 계정에서 사용할 수 있게 권한 설정



Users > 사용자계정명 > Schema 권한 탭 선택

The screenshot shows the MySQL Workbench interface with the following details:

- MySQL Workbench Title Bar:** inho DB, MySQL Model, Local instance MySQL80.
- File Menu:** File, Edit, View, Query, Database, Server, Tools, Scripting, Help.
- Toolbar:** SQL, SQL*, Text, Data, Structure, DDL, DML, Find, Refresh.
- Navigator:** Local instance MySQL80.
- Management Panel:**
 - MANAGEMENT:** Server Status, Client Connections, **Users and Privileges** (highlighted with a red box), Status and System Variables, Data Export, Data Import/Restore.
 - INSTANCE:** Startup / Shutdown, Server Logs, Options File.
 - PERFORMANCE:** Dashboard, Performance Reports, Performance Schema Setup.
- Central Area:** Local instance MySQL80, Users and Privileges.
- User Accounts Table:**

User	From Host
minco	%
mysql.infoschema	localhost
mysql.session	localhost
mysql.sys	localhost
root	localhost
- Details for account minco@%** panel:
 - Login:** Account Limits, Administrative Roles, **Schema Privileges** (highlighted with a red box).
 - Schema:** minco
 - Privileges:** ALTER, ALTER ROUTINE, CREATE, CREATE ROUTINE, CRE...
- Text Labels:** Schema and Host fields may use % and _ wildcards. The server will match specific entries before wildcarded ones.
- Buttons:** Revoke All Privileges.
- Checkboxes:** Object Rights: SELECT, INSERT, UPDATE. DDL Rights: CREATE, ALTER, REFERENCES.

world 스키마의 모든 query 문을 사용할 수 있도록 설정

Local instance MySQL80

Users and Privileges

User Accounts

User	From Host
minco	%
mysql.infoschema	localhost
mysql.session	localhost
mysql.sys	localhost
root	localhost

Details for account minco@%

Schema Privileges

Schema	Privileges
minco	ALTER, ALTER ROUTINE, CREATE, CREATE ROUTINE, CREATE TEMPORARY TABLES, CREATE VIEW, DELETE, DROP, EVENT, EXECUTE, INDEX, INSERT, LO...
world	ALTER, ALTER ROUTINE, CREATE, CREATE ROUTINE, CREATE TEMPORARY TABLES, CREATE VIEW, DELETE, DROP, EVENT, EXECUTE, INDEX, INSERT, LO...

Schema and Host fields may use % and _ wildcards.
The server will match specific entries before wildcarded ones.

The user 'minco'@'%' will have the following access rights to the schema 'world':

Object Rights

- SELECT
- INSERT
- UPDATE
- DELETE
- EXECUTE
- SHOW VIEW

DDL Rights

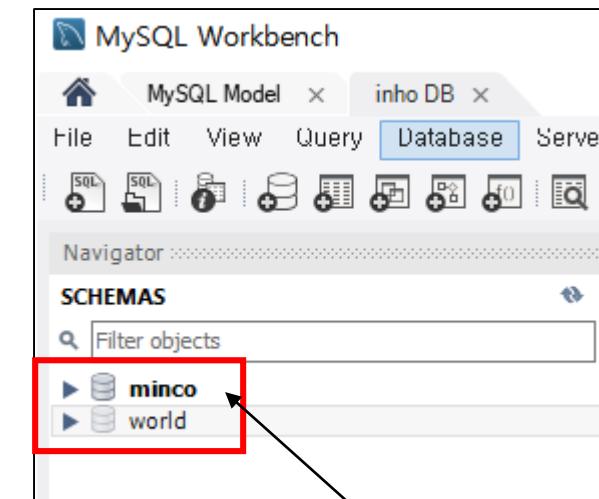
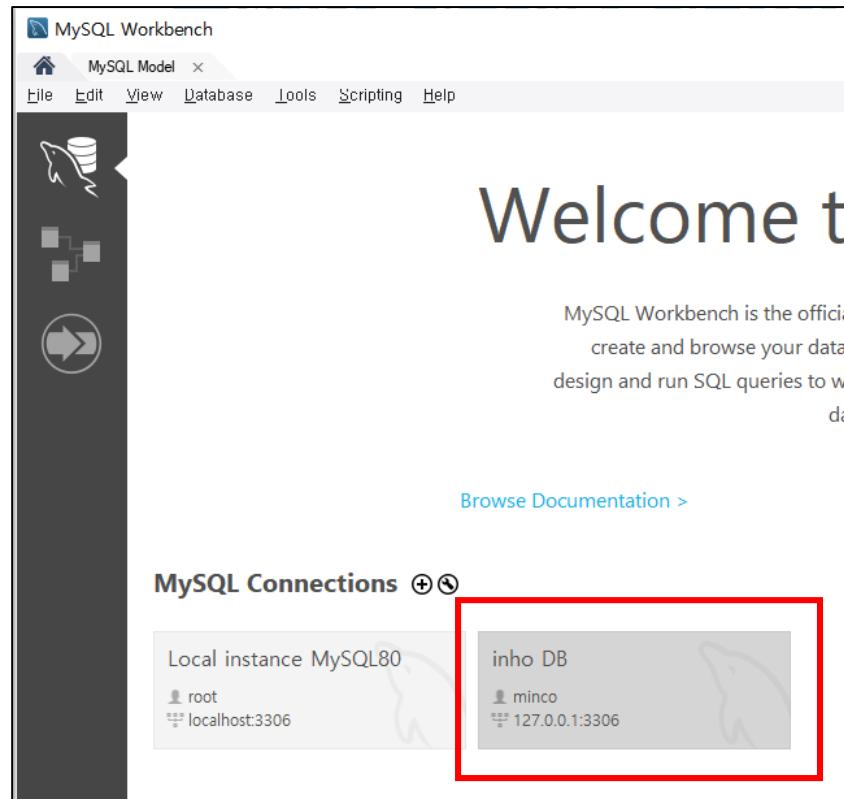
- CREATE
- ALTER
- REFERENCES
- INDEX
- CREATE VIEW
- CREATE ROUTINE
- ALTER ROUTINE
- EVENT
- DROP
- TRIGGER

Other Rights

- GRANT OPTION
- CREATE TEMPORARY TABLES
- LOCK TABLES

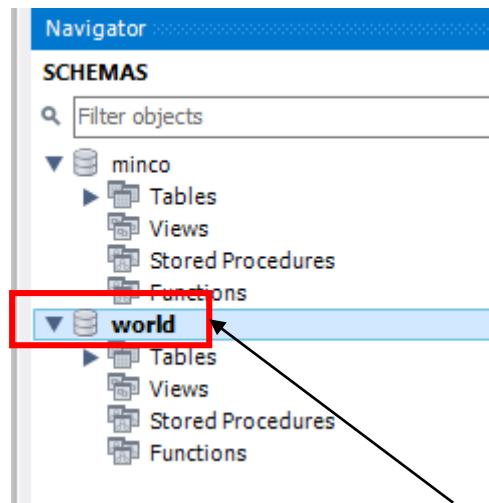
사용자 계정에 접속하여 권한이 잘 부여되었는지 확인한다.

- 스키마 창에 [사용자계정명] / [world] 스키마 2개가 있어야 한다.

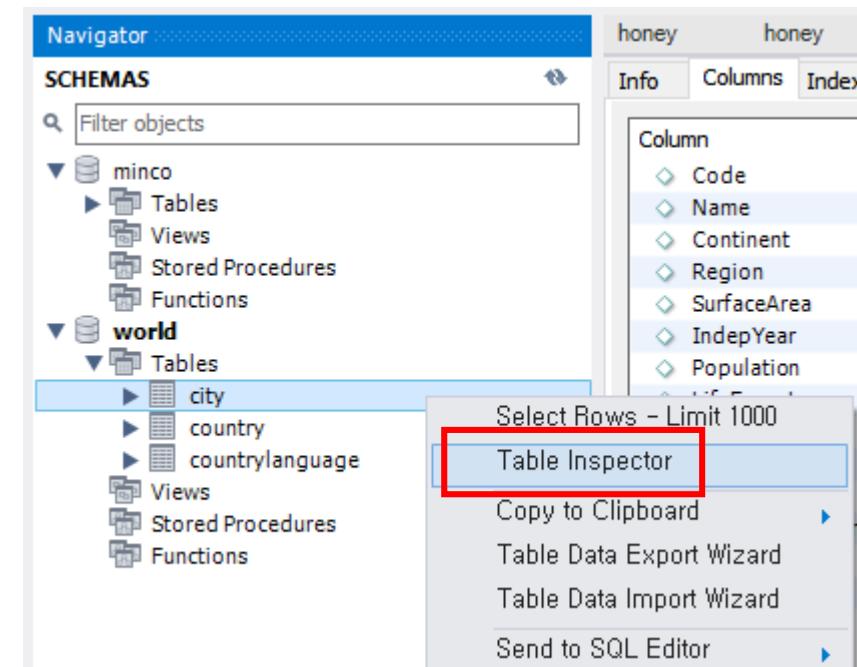


굵은 글씨 : 현재 use 상태

연습용 스키마 world 구조 확인하기



더블 클릭시
자동으로 use world;
명령어가 수행된다.



Columns 구조를 확인할 수 있다.

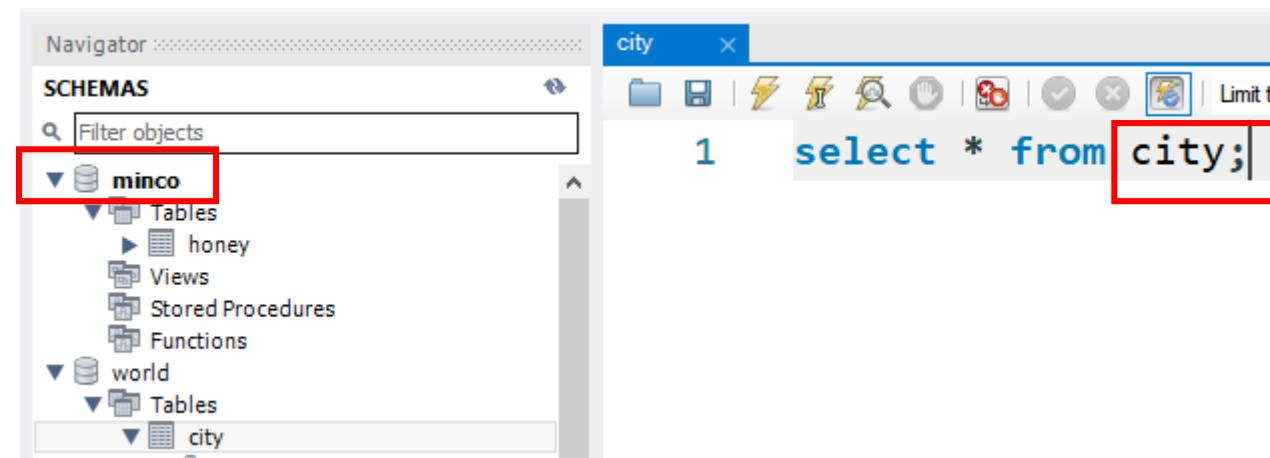
The screenshot shows the MySQL Workbench interface for the 'world.city' table. The 'Columns' tab is selected and highlighted with a red box. The table structure is displayed with columns for Column, Type, Default Value, Nullable, Character Set, Collation, and Privileges. The MySQL command 'desc city;' is run in the terminal window below, and its output is shown, matching the structure in Workbench.

Column	Type	Default Value	Nullable	Character Set	Collation	Privileges
ID	int		NO			select,insert,update,references
Name	char(35)		NO	latin1	latin1_swedish_ci	select,insert,update,references
CountryCode	char(3)		NO	latin1	latin1_swedish_ci	select,insert,update,references
District	char(20)		NO	latin1	latin1_swedish_ci	select,insert,update,references
Population	int	0	NO			select,insert,update,references


```
mysql> desc city;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| ID   | int  | NO   | PRI | NULL    | auto_increment |
| Name | char(35) | NO  | MUL |          |                |
| CountryCode | char(3) | NO  |      |          |                |
| District | char(20) | NO  |      |          |                |
| Population | int  | NO  |      | 0        |                |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

world 스키마의 city table을 query 하기 위해서 use 필수

- 아래 그림과 같이 world가 굵은 글씨가 아닌 상태에서, 에러가 발생한다.
- “world”에 더블클릭 해주면 된다.



BETWEEN / IN / LIKE

구간 사이 데이터 추출

- WHERE 절 다음 입력한다.
- Population ≥ 500 AND Population ≤ 900 과 동일하다.

```
select * from city
```

```
where Population BETWEEN 500 AND 900;
```

	id	name	country_code	district	Population
▶	62	The Valley	AIA	Home Island	595
	1791	Flying Fish Cove	CXR	Home Island	700
	2316	Bantam	CCK	Home Island	503
	2728	Yaren	NRU	Home Island	559
	2805	Alofi	NIU	Home Island	682
*	2806	Kingston	NFK	Home Island	800
		NULL	NULL	NULL	NULL

후보들 중 나오는 값을 의미

- WHERE 절 다음 입력한다.
- IN (후보 Item1, 후보 Item2, 후보 Item3 ..)

```
select * from city
where Name IN('Seoul', 'Sydney', 'Oxford');
```

ID	Name	CountryCode	District	Population
130	Sydney	AUS	New South Wales	3276207
498	Oxford	GBR	England	144000
2331	Seoul	KOR	Seoul	9981619
*	NULL	NULL	NULL	NULL

문자열 검색

- WHERE 절 다음에 사용
- % : 다중 문자를 의미한다. (wildcard 문자)
- _ : 한 글자를 의미한다.

```
select * from city  
where Name LIKE 'New%';
```

ID	Name	CountryCode	District	Population
137	Newcastle	AUS	New South Wales	270324
482	Newcastle upon Tyne	GBR	England	189150
502	Newport	GBR	Wales	139000
734	Newcastle	ZAF	KwaZulu-Natal	222993
1106	New Bombay	IND	Maharashtra	307297
1109	New Delhi	IND	Delhi	301297
3793	New York	USA	New York	8008278
3823	New Orleans	USA	Louisiana	484674

```
select * from city  
where CountryCode LIKE 'K_R';
```

ID	Name	CountryCode	District	Population
2255	Bikenibeu	KIR	South Tarawa	5055
2256	Bairiki	KIR	South Tarawa	2226
2331	Seoul	KOR	Seoul	9981619
2332	Pusan	KOR	Pusan	3804522
2333	Inchon	KOR	Inchon	2559424
2334	Taegu	KOR	Taegu	2548568
2335	Taejon	KOR	Taejon	1425835
2336	Kwangju	KOR	Kwangju	1368341

- LIKE % -> 0~N개
- LIKE _ -> 1개

```
SELECT * FROM OrderDetails WHERE ProductID LIKE '1%';
```

Result:

	OrderDetailID	OrderID	ProductID	Quantity
1	1	10248	11	12
2	4	10249	14	9
3	22	10255	16	35
4	44	10263	16	60
5	50	10265	17	30
6	52	10266	12	12
7	66	10273	10	24
8	75	10276	10	15
9	76	10276	13	10
10	83	10279	17	15
11	87	10281	19	1
12	92	10283	15	20
13	93	10283	19	18
14	100	10285	1	45

Number of Records: 297

```
SELECT * FROM OrderDetails WHERE ProductID LIKE '1_';
```

OrderDetailID	OrderID	ProductID	Quantity
1	10248	11	12
4	10249	14	9
22	10255	16	35
44	10263	16	60
50	10265	17	30
52	10266	12	12
66	10273	10	24
75	10276	10	15

1. 인구수가 1000명 미만이면서, A로 시작하는 도시 찾기
2. 일본 (JPN)에서, 인구수가 100만명 ~ 200만명인 도시를 찾아
도시 이름과, 국가 코드 (JPN) 필드만 출력하기
3. kim이라는 글자가 포함된 도시명 찾기

GROUP BY / HAVING

그룹을 지어 데이터를 묶는다.

- 집계 함수를 사용하기 위해 묶는다.
 - SUM() / AVG()
 - MIN() / MAX()
 - COUNT()

ID	합계
129	29034
1	2332100
56	2561600
61	1556
34	270000
55	21189
33	2345

```
select ID, sum(Population) as 합계 from city  
group by CountryCode;
```

```
select CountryCode, count(Population) as '도시수' from city  
group by CountryCode;
```

CountryCode	도시수
ABW	1
AFG	4
AGO	5
AIA	2
ALB	1
AND	1
ANT	1

GROUP BY을 썼을 때 쓸 수 있는 조건 절

- 집계함수에 대해 조건을 걸 수 있음

```
select CountryCode, count(Population) as '도시수' from city  
group by CountryCode  
having count(Population) > 200;
```

	CountryCode	도시수
▶	BRA	250
	CHN	363
	IND	341
	JPN	248
	USA	274

특정 개수만 보여줌

- 빠른 Query 성능을 위해 사용 함
- 웹 개발에서, 최근 게시물에 자주 사용함

select * from city limit 3;					
	ID	Name	CountryCode	District	Population
▶	1	Kabul	AFG	Kabul	1780000
	2	Qandahar	AFG	Qandahar	237500
	3	Herat	AFG	Herat	186800
	NULL	NULL	NULL	NULL	NULL

1. world 스키마의 레코드 총 개수 출력

▶	레코 드수
▶	4079

2. 하위 5개의 도시 코드, 인구수 출력

도시명	인구수
Adamstown	42
West Island	167
Fakaofo	300
CittÃ del Vaticano	455
Bantam	503

3. TOP 5개의 국가 코드, 인구수 출력

	CountryCode	인구수
▶	IND	10500000
	KOR	9981619
	BRA	9968485
	CHN	9696300
	IDN	9604900

6장. 데이터 탑입

챕터의 포인트

- 데이터 타입 종류
- 샘플 데이터 다운로드
- DATE 타입 다루기
- NULL 값 처리

데이터 타입 종류

숫자형

- int / float

문자형

- char / varchar / text

날짜형

- date / datetime

world 스키마의 city table 확인

- char(35) : 35 Byte 만큼 char 사용
- int : 약 -21 억 부터 약 21억 사이 정수

The screenshot shows a database interface with the following details:

- Navigator:** Shows SCHEMAS: minco (Tables: honey, Views, StoredProcedure, Functions), world (Tables: city, country, countrylang, Views, StoredProcedure, Functions).
- Table View:** city table from the world schema.
- Columns Tab:** Selected tab. Data shown:

Column	Type
ID	int
Name	char(3)
CountryCode	char(3)
District	char(2)
Population	int
- Context Menu:** Opened over the table, with items: Select Rows - Limit 1000, Table Inspector (highlighted with a red box), Copy to Clipboard, Table Data Export Wizard.
- Callout Box:** Details the column types:

Column	Type
ID	int
Name	char(35)
CountryCode	char(3)
District	char(20)
Population	int

char

- 고정길이 데이터 타입, 빈 공간은 띄어쓰기로 채운다.
- 예를들어 char(5)라면, 5 Byte 를 사용하는 것이고 만약 해당 문자열에 'BTS' 를 넣으면 공백을 포함한 '**BTS**' 으로 저장된다.

varchar

- 가변길이 데이터 타입, 빈 공간을 채우지 않는다.
- varchar(20) 이지만, 'BTS'를 넣는다면 3 Byte만 사용한다.
- 공간을 아낄 수 있지만, 데이터 파편화로 성능 저하

문자열의 특징에 따라 쓰는 타입을 결정하자

- **char** : 길이가 일정한 문자 / 빠른 성능이 필요한, 자주 읽는 필드
- **varchar** : 주소 와 같이 길이 변화가 큰 문자
- **text** : 64KB 미만의 긴 문자열

대표적인 날짜형 데이터 타입

- **DATE**

- 날짜 타입 (year, month, day)
- 기본 Format은 YYYY-MM-DD

- **DATETIME**

- 날짜 + 시간을 나타냄
- YYYY-MM-DD HH:MM:SS

JSON Data를 저장할 수 있는 Type

- **INSERT 예시**

```
mysql> INSERT INTO t1 VALUES('{"key1": "value1", "key2": "value2"}');
Query OK, 1 row affected (0.01 sec)
```

샘플 데이터 다운로드

employees 데이터베이스 다운로드

- 쿼리문을 연습하기 좋은 데이터베이스
- <https://www.mysql.com/>
공식사이트 > document > More 클릭

The screenshot shows the MySQL Documentation page. At the top, there are navigation links: MySQL.COM, DOWNLOADS, DOCUMENTATION (which is highlighted with a red box), and DEVELOPER ZONE. Below this, there's a search bar and a sidebar with links to Archives and About. The main content area is titled "Other MySQL Documentation" and includes sections for MySQL Server Doxygen Documentation and Expert Guides. A table lists various documentation versions with their titles, versions, and download links.

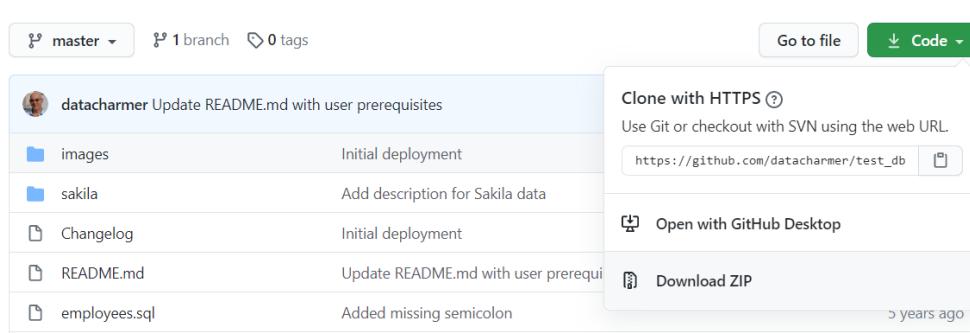
Language	Title	Version	HTML Online	PDF	HTML Download
English	Extending MySQL	8.0	View	US Ltr A4	TGZ Zip
English	Extending MySQL	5.7	View	US Ltr A4	TGZ Zip
English	Extending MySQL	5.6	View	US Ltr A4	TGZ Zip

The screenshot shows the "Example Databases" section. It lists several databases with their titles, download links, and setup guides. The first database, "employee data (large dataset, includes data and test/verification suite)", has its "Download DB" link highlighted with a red box.

Title	Download DB	HTML Setup Guide	PDF Setup Guide
employee data (large dataset, includes data and test/verification suite)	GitHub	View	US Ltr A4
world database	Gzip Zip	View	US Ltr A4
world_x database	TGZ Zip	View	US Ltr A4
sakila database	TGZ Zip	View	US Ltr A4
menagerie database	TGZ Zip		

mysql console에서 Script를 수행한다.

- git clone을 다운로드 > 압축해제
- CMD 창 띄우고, 해당 폴더로 이동하기

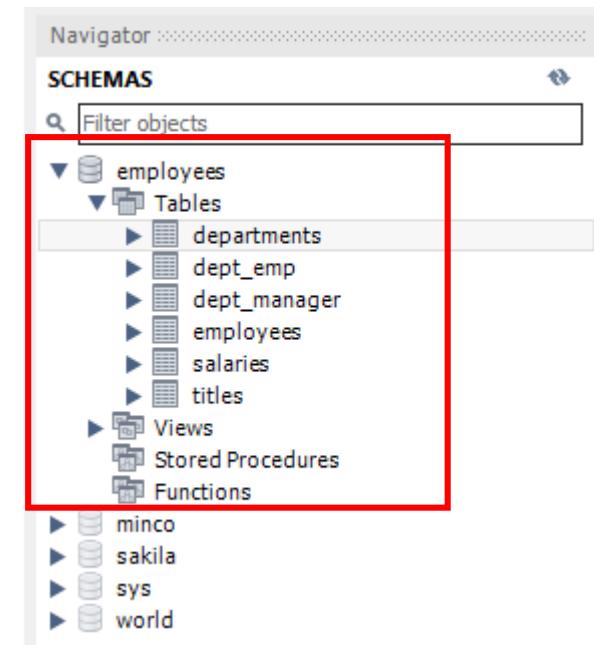
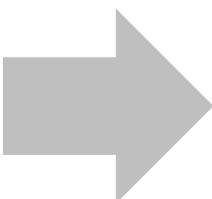


```
C:\Users\minco>cd C:\Users\minco\Downloads\test_db-master\test_db-master  
C:\Users\minco\Downloads\test_db-master\test_db-master>
```

root 계정으로 script를 수행한다.

- 약 1분 소요

```
mysql -u root -p < employees.sql
```



날짜 태입 다루기

날짜 타입 다루기

- 날짜 및 시간은 필수로 들어가는 데이터이다.
- Log / 고객 데이터가 언제 기록이 되었는지 필요

```
select DATE("2025-12-25 11:42:59");
```

	DATE "2025-12-25"
▶	2025-12-25

```
select TIME("2025-12-25 11:42:59");
```

	TIME("2025-12-25 11:42:59")
▶	11:42:59

날짜 / 시간 함수 다루기

다음 명령어는 각각 2025, 12, 25가 출력된다.

- select YEAR("2025-12-25 11:42:59");
- select MONTH("2025-12-25 11:42:59");
- select DAY("2025-12-25 11:42:59");

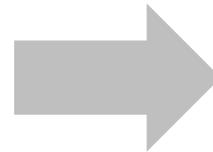
다음 명령어는 각각 11, 42, 59가 출력 된다.

- select hour("2025-12-25 11:42:59");
- select minute("2025-12-25 11:42:59");
- select second("2025-12-25 11:42:59");

해당 Table에서 from / to 필드의 Year 추출하기

- employees.dept_emp 테이블 사용
- emp_no 기준, 최상위 3개만 추출하기

	emp_no	dept_no	from_date	to_date
▶	10001	d005	1986-06-26	9999-01-01
	10002	d007	1996-08-03	9999-01-01
	10003	d004	1995-12-03	9999-01-01
	10004	d004	1986-12-01	9999-01-01
	10005	d003	1989-09-12	9999-01-01
	10006	d005	1990-08-05	9999-01-01
	10007	d008	1989-02-10	9999-01-01
	10008	d005	1998-03-11	2000-07-31



	시작 년도	종료 년도
▶	1986	9999
	1996	9999
	1995	9999

날짜 간격 구하기

- **timestampdiff**

- 예시 :`select timestampdiff(day, '2020-08-10', '2020-08-13');`
- day 대신 다음과 같은 단위 지정 가능

- second
- minute
- hour
- day
- week
- month
- year

	<code>timestampdiff(day, '2020-08-10', '2020-08-13')</code>
▶	3

날짜 더하기

- `date_add()` 함수 사용
- `select date_add(now(), interval 1 hour);`
- `select date_add(now(), interval 1 day);`
- `select date_add(now(), interval 1 month);`

날짜 빼기

- `date_sub()` 함수 사용
- `select date_sub(now(), interval 1 month);`

NULL 값과 문자열 함수

ALTER Table

- Table 구조를 수정할 때 사용
- dept_name에서 NOT NULL 체크 표시를 없애보자. (NULL을 허용하게끔 설정 변경)

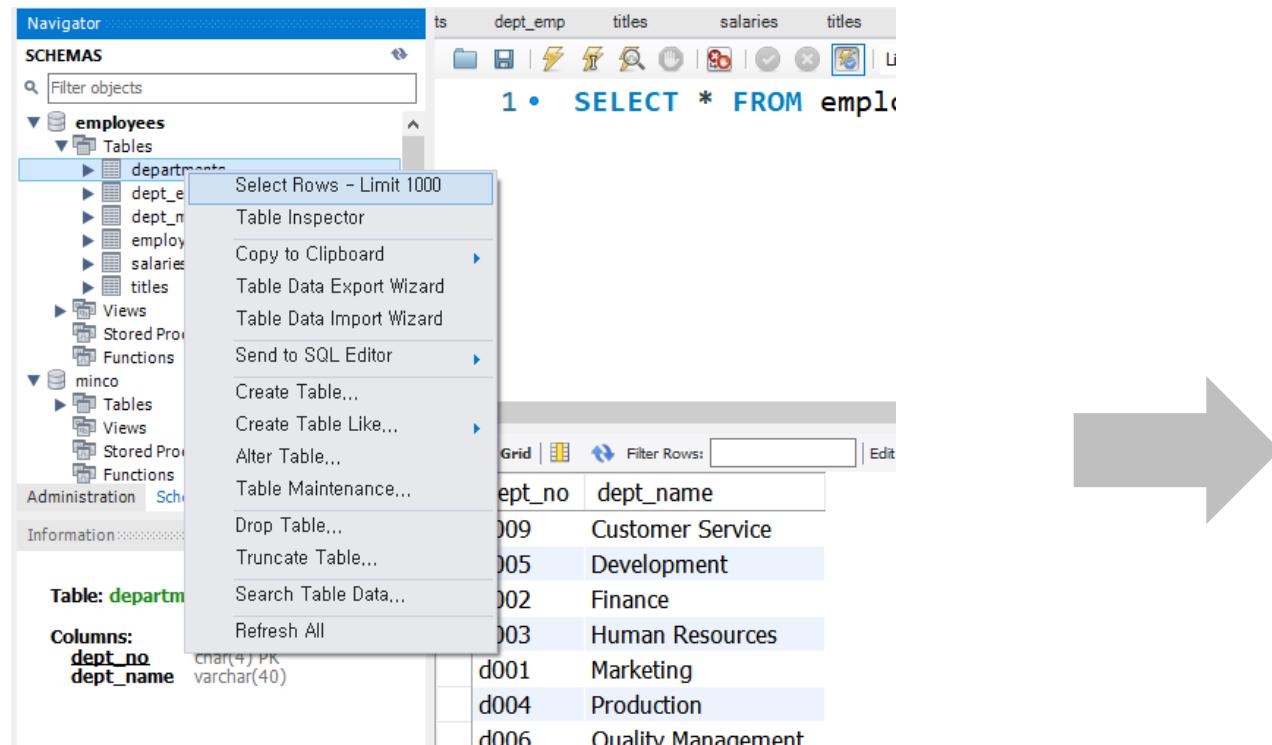
The image shows two screenshots of MySQL Workbench. On the left, the 'Navigator' pane shows the 'employees' schema with its tables: employees, dept_emp, dept_name, employee, salaries, and titles. A right-click context menu is open over the 'dept_name' table, with the 'Alter Table...' option highlighted. A large grey arrow points from this menu to the 'Alter Table' configuration window on the right. This window shows the 'departments' table with 'Charset/Collation' set to 'utf8mb4'. The 'dept_no' column is defined as CHAR(4) with PK and NN checked. The 'dept_name' column is defined as VARCHAR(40) with NN checked but UQ unchecked. A red box highlights the NN checkbox for the 'dept_name' column.

Column Name	Datatype	PK	NN	UQ
dept_no	CHAR(4)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
dept_name	VARCHAR(40)	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

실습 전, 백업을 수행하자.

일부 값을 지우고 적용해보자.

- Select Rows > 일부 값 선택 후 Delete 키 누르기



	dept_no	dept_name
	d009	Customer Service
	d005	Development
▶	d002	NULL
	d003	NULL
	d001	Marketing
	d004	Production
	d006	Quality Management
	d008	Research

IFNULL(필드, 대체 값)

- 만약 필드의 값이 NULL이라면, “대체 값”으로 출력하기
- SELECT dept_no, IFNULL(dept_name,'#') FROM employees.departments;

	dept_no	IFNULL(dept_name,'#')
▶	d002	#
	d003	#
	d009	Customer Service
	d005	Development
	d001	Marketing
	d004	Production
	d006	Quality Management

값을 합치는 함수

- `select concat('안녕', '하', '시오');`

concat('안녕', '하', '시 오')
▶ 안녕하시오

- `SELECT concat(emp_no, " - ", dept_no, " - ", from_date) FROM employees.dept_emp;`

	emp_no	dept_no	from_date	to_date
▶	10001	d005	1986-06-26	9999-01-01
	10002	d007	1996-08-03	9999-01-01
	10003	d004	1995-12-03	9999-01-01
	10004	d004	1986-12-01	9999-01-01
	10005	d003	1989-09-12	9999-01-01
	10006	d005	1990-08-05	9999-01-01
	10007	d008	1989-02-10	9999-01-01
	10008	d005	1998-03-11	2000-07-31
	10009	d006	1985-02-18	9999-01-01
	10010	d004	1996-11-24	2000-06-26
	10010	d006	2000-06-26	9999-01-01
	10011	d009	1990-01-22	1996-11-09
	10012	d005	1992-12-18	9999-01-01
	10013	d003	1985-10-20	9999-01-01
	10014	d005	1993-12-29	9999-01-01

concat(emp_no, " - ", dept_no, " - ", from_date)
▶ 10001 - d005 - 1986-06-26
10002 - d007 - 1996-08-03
10003 - d004 - 1995-12-03
10004 - d004 - 1986-12-01
10005 - d003 - 1989-09-12
10006 - d005 - 1990-08-05
10007 - d008 - 1989-02-10
10008 - d005 - 1998-03-11
10009 - d006 - 1985-02-18
10010 - d004 - 1996-11-24
10010 - d006 - 2000-06-26
10011 - d009 - 1990-01-22
10012 - d005 - 1992-12-18
10013 - d003 - 1985-10-20
10014 - d005 - 1993-12-29
10015 - d008 - 1992-10-10

타입을 변환하는 함수

- convert(필드명, 변환 Type)
- 변환 Type
 - char
 - int
 - datetime 등

DB KEY

PK(Primary Key)

- 한 테이블 내에서 중복되지 않는 값만 가질수 있는 키
- 유일한 값
- 중복 허용 X
- NULL 값 X

menus			
id	name	description	image_src
1	카페라떼	라떼는.. 말이야	public/image1.jpg
2	카푸치노	언빌리 "버블"	public/image2.jpg
3	아메리카노	아메아메 좋아	public/image3.jpg
4	복숭아 아이	상큼한 복숭아!	public/image4.jpg

FK (Foreign Key)

- 특정 테이블에 포함되어 있으면서 다른테이블의 기본키로 지정된 키
- 기본적으로는 다른 테이블의 기본 키를 참조한다.
- NULL, 중복 허용 OK

menus

id	name	description	image_src
1	카페라떼	라떼는.. 말이야	public/image1.jpg
2	카푸치노	언빌리 "버블"	public/image2.jpg
3	아메리카노	아메아메 좋아	public/image3.jpg
4	복숭아 아이	상큼한 복숭아!	public/image4.jpg

orders

id	quantity	request_detail	menus_id
1	1	얼음 많이많이 주세요.	1
2	3	시럽 넣지 말아주세요.	2
3	1	우유 많이 주세요.	2
4	2	복숭아 좋아요.	4

JOIN

관계형 데이터 베이스의 꽃

- 데이터베이스를 연결하여 데이터를 검색하는 방법
- PK, FK로 두 테이블을 연결한다.

왜 테이블을 둘로 나눴을까?

- 다음 상황을 가정해보자.
- 만약 100,000 개의 데이터를 저장한다면, 중복이 지나치게 많아 매우 비효율적
 - DB 용량 증가, 속도 저하
 - 비용 증가: Oracle 등의 유료 데이터베이스를 사용한다면, 훨씬 많은 비용 지불
 - 만약, menu_description 을 바꾸고 싶다면? 100,000 개의 데이터에서 찾아 전부 바꿔야한다

id	name	description	quantity	request_detail
1	카페라떼	라떼는.. 말이야	1	얼음 많이많이 주세요.
2	카페라떼	라떼는.. 말이야	3	시럽 넣지 말아주세요.
3	아메리카노	아메아메 좋아	1	
4	복숭아 아이스티	상큼한 복숭아!	2	

관계형 데이터베이스 (Relational Database)

- 여러 테이블의 "관계"로 이루어진다.
- 핵심은 FK (Foreign Key, 외래키) 와 JOIN
- 중복 제거
 - 만약 100,000 개의 데이터를 저장하더라도,
 - DB 용량을 훨씬 효율적으로 사용
 - 만약, menu_description 을 바꾸고 싶다면? menus 테이블에서 "단 하나만" 변경하면 끝

menus

id	name	description	image_src
1	카페라떼	라떼는.. 말이야	public/image1.jpg
2	카푸치노	언빌리 "버블"	public/image2.jpg
3	아메리카노	아메아메 좋아	public/image3.jpg
4	복숭아 아이	상큼한 복숭아!	public/image4.jpg

orders

id	quantity	request_detail	menus_id
1	1	얼음 많이많이 주세요.	1
2	3	시럽 넣지 말아주세요.	2
3	1	우유 많이 주세요.	2
4	2	복숭아 좋아요.	4

JOIN 문법을 사용하는 이유

- `orders`에서, 주문마다 `menus_id` 보유
- 즉, `id`만 알고 있다면 `menus`에 접근해서
 - 메뉴 이름, 메뉴 설명, 메뉴 이미지에 대한 정보를 가져올 수 있음
 - 이를 위해 **JOIN 문법** 사용

menus			
id	name	description	image_src
1	카페라떼	라떼는.. 말이야	public/image1.jpg
2	카푸치노	언빌리 "버블"	public/image2.jpg
3	아메리카노	아메아메 좋아	public/image3.jpg
4	복숭아 아이	상큼한 복숭아!	public/image4.jpg

orders			
id	quantity	request_detail	menus_id
1	1	얼음 많이많이 주세요.	1
2	3	시럽 넣지 말아주세요.	2
3	1	우유 많이 주세요.	2
4	2	복숭아 좋아요.	4



JOIN

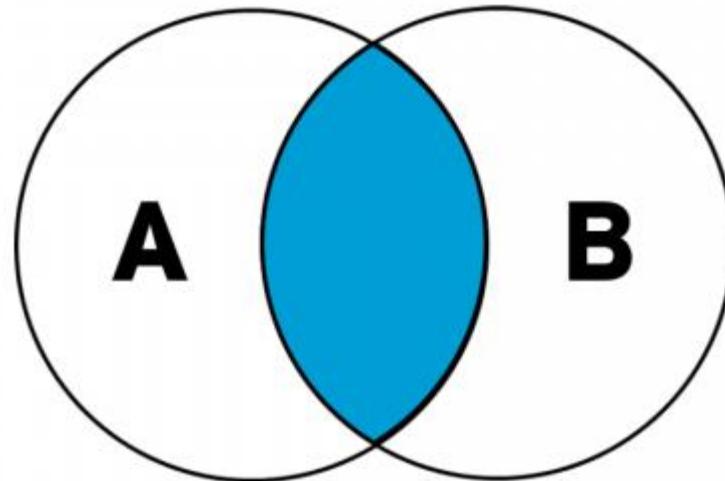
- SELECT <열 목록>
FROM <첫번째 테이블>
<조인의 종류> <두번째 테이블>
ON <조인될 조건>
[WHERE 검색조건]

INNER JOIN

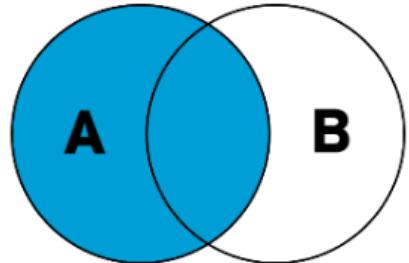
- ```
SELECT *
FROM A a
INNER JOIN B b
ON a.id = b.id
```

## **INNER JOIN**

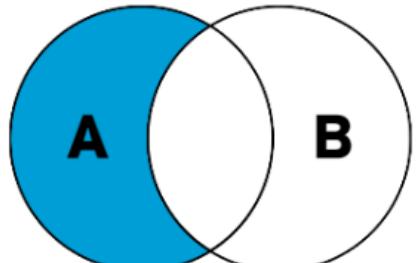
---



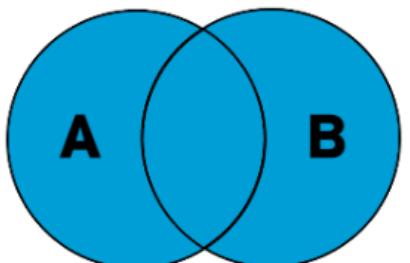
- 각 사원들의 사원번호, 부서번호를 가져오기
  - employees 테이블, departments 테이블 활용
- 각 사원들의 사원번호, first\_name, 현재 받고 있는 급여액(salary) 가져오기

**LEFT OUTER JOIN**

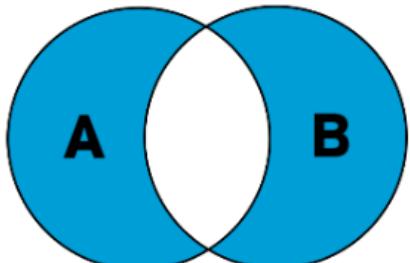
```
SELECT *
FROM A a
LEFT JOIN B b
ON a.KEY = b.KEY
```



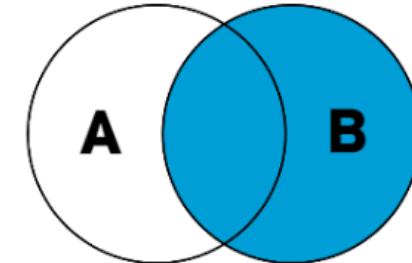
```
SELECT *
FROM A a
LEFT JOIN B b
ON a.KEY = b.KEY
WHERE b.KEY IS NULL
```

**FULL OUTER JOIN**

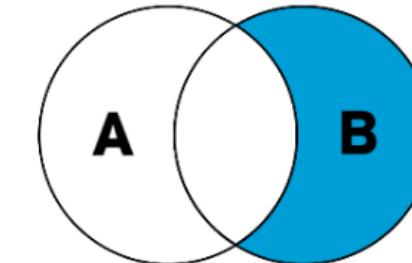
```
SELECT *
FROM A a
FULL OUTER JOIN B b
ON a.KEY = b.KEY
```



```
SELECT *
FROM A a
FULL OUTER JOIN B b
ON a.KEY = b.KEY
WHERE a.KEY IS NULL
OR b.KEY IS NULL
```

**RIGHT OUTER JOIN**

```
SELECT *
FROM A a
RIGHT OUTER JOIN B b
ON a.KEY = b.KEY
```



```
SELECT *
FROM A a
RIGHT OUTER JOIN B b
ON a.KEY = b.KEY
WHERE a.KEY IS NULL
```

내일  
방송에서  
만나요!

삼성 청년 SW 아카데미