



이수안 컴퓨터 연구소

suan computer laboratory

MySQL 한번에 끝내기

The screenshot shows the MySQL Workbench interface. On the left, the Navigator pane displays the database schema with tables like actor, address, category, city, country, customer, film, film_actor, film_category, film_text, inventory, language, and rental. The central area shows an EER Diagram with entities like Customer, Address, City, Country, Actor, Film, Inventory, Language, and Rental, connected by various relationships. Below the diagram, the Query Editor pane contains two queries:

```
Query 1:
1 # *SELECT `actor`.`actor_id` *
2   `actor`.`first_name` ,
3   `actor`.`last_name` ,
4   `actor`.`gender` ,
5   FROM `sakila`.`actor`;
6
7 *SELECT `film`.`film_id` *
8   `film`.`title` ,
9   `film`.`description` ,
10  `film`.`release_year` ,
11  `film`.`language_id` ,
12  `film`.`original_language_id` ,
13  `film`.`rental_duration` ,
14  `film`.`rental_rate` ,
15  `film`.`length` ,
16  `film`.`replacement_cost` ,
17  `film`.`special_features` ,
18  `film`.`full_text` ,
```

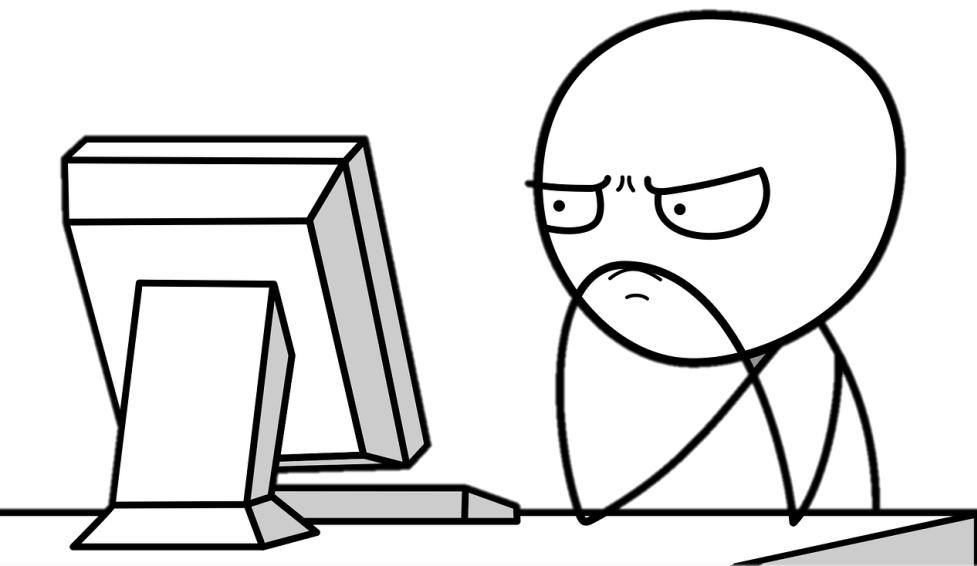
```
Query 2:
1 *SELECT DISTINCT `language` *
2   `language` ,
3   `language`.`name` ,
4   `language`.`iso_639_1` ,
5   `language`.`iso_639_2` ,
6   `language`.`english_name` ,
7   `language`.`description` ,
8   `language`.`re_release_year` ,
9   `language`.`language_id` ,
10  `language`.`original_language_id` ,
11  `language`.`rental_rate` ,
12  `language`.`length` ,
13  `language`.`replacement_cost` ,
14  `language`.`special_features` ,
```



목차

1. MySQL 소개
2. MySQL 설치
3. SQL 기본
4. SQL 고급

1. MySQL 소개

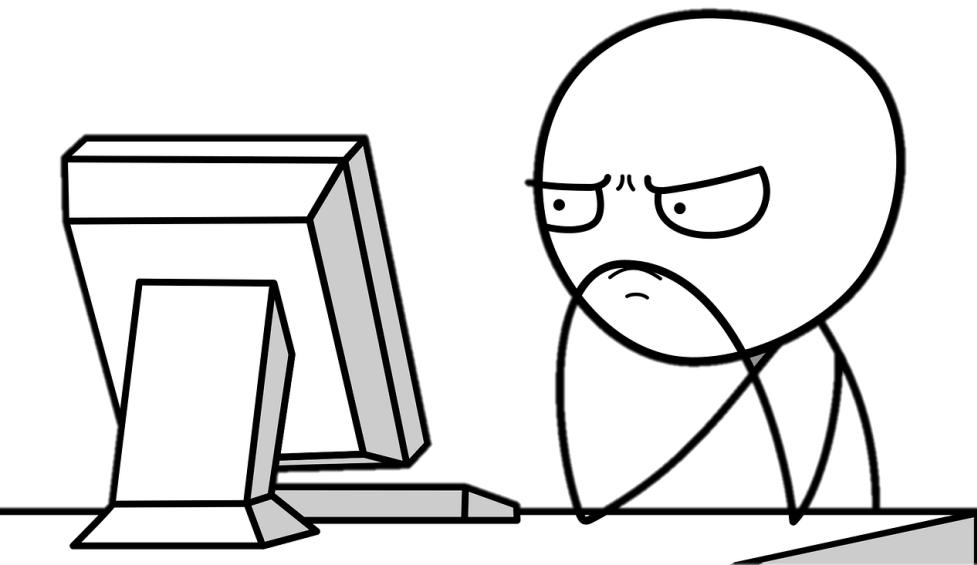


MySQL 소개

- MySQL은 가장 널리 사용되고 있는 관계형 데이터베이스 관리 시스템(RDBMS: Relational DBMS)
- MySQL은 오픈 소스이며, 다중 사용자와 다중 스레드를 지원
- C언어, C++, JAVA, PHP 등 여러 프로그래밍 언어를 위한 다양한 API를 제공
- MySQL은 유닉스, 리눅스, 윈도우 등 다양한 운영체제에서 사용할 수 있으며, 특히 PHP 와 함께 웹 개발에 자주 사용
- MySQL은 오픈 소스 라이센스를 따르기는 하지만, 상업적으로 사용할 때는 상업용 라이센스 구입 필요



2. MySQL 설치



MySQL 다운로드

The world's most popular open source database 

Contact MySQL | Login | Register

MySQL™ 

MYSQL.COM DOWNLOADS DOCUMENTATION DEVELOPER ZONE 

Products Cloud Services Partners Customers Why MySQL? News & Events How to Buy



MySQL™
ORACLE® CLOUD

New! Oracle MySQL Cloud Service

MySQL Enterprise Edition powered by Oracle Cloud

LEARN MORE



MySQL Enterprise Edition

The most comprehensive set of advanced features, management tools and technical support to achieve the highest levels of MySQL scalability, security, reliability, and uptime.



Oracle MySQL Cloud Service

Built on MySQL Enterprise Edition and powered by the Oracle Cloud, Oracle MySQL Cloud Service provides a simple, automated, integrated and enterprise ready MySQL cloud service, enabling organizations to

- MySQL 한번에 끝내기

MySQL 다운로드

The world's most popular open source database 

Contact MySQL | Login | Register

MySQL™  MYSQL.COM DOWNLOADS DOCUMENTATION DEVELOPER ZONE 

Enterprise Community Yum Repository APT Repository SUSE Repository Windows Archives

MySQL on Windows 

- MySQL Installer
- MySQL Connectors
- MySQL Workbench
- MySQL for Excel
- MySQL Notifier
- MySQL for Visual Studio
- MySQL Yum Repository
- MySQL APT Repository
- MySQL SUSE Repository
- MySQL Community Server
- MySQL Cluster

[MySQL Installer](https://dev.mysql.com/downloads/installer/) 

MySQL provides you with a suite of tools for developing and managing business critical applications on Windows.

[MySQL Installer](#)

MySQL Installer provides an easy to use, wizard-based installation experience for all MySQL software on Windows.

[MySQL Connectors](#)

MySQL offers industry standard database driver connectivity for using MySQL with applications and tools.

[MySQL Workbench](#)

MySQL Workbench provides DBAs and developers an integrated tools environment for database design, administration, SQL development and database migration.

[MySQL for Excel](#)

Enables users to import, export and edit MySQL data using Microsoft Excel. Available with MySQL Installer.

<https://dev.mysql.com/downloads/installer/>

MySQL 다운로드

- MySQL Shell
- MySQL Workbench
- MySQL Connectors
- Other Downloads

- If you have an online connection while running the MySQL Installer, choose the [mysql-installer-web-community](#) file.
- If you do NOT have an online connection while running the MySQL Installer, choose the [mysql-installer-community](#) file.

Note: MySQL Installer is 32 bit, but will install both 32 bit and 64 bit binaries.

Online Documentation

- [MySQL Installer Documentation](#) and [Change History](#)

Please report any bugs or inconsistencies you observe to our [Bugs Database](#).

Thank you for your support!

Generally Available (GA) Releases

MySQL Installer 8.0.13

Select Operating System:

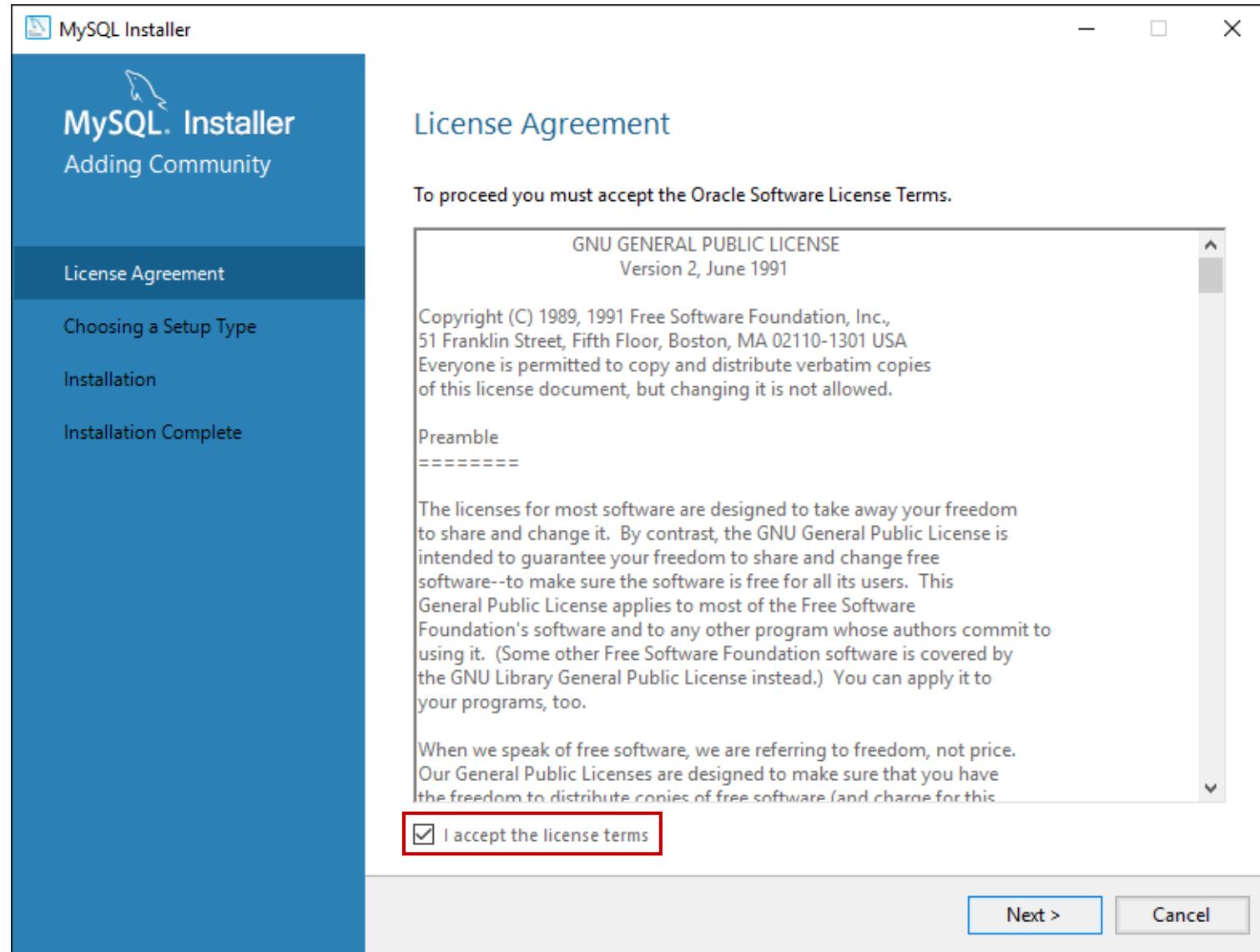
Microsoft Windows

Looking for previous GA versions?

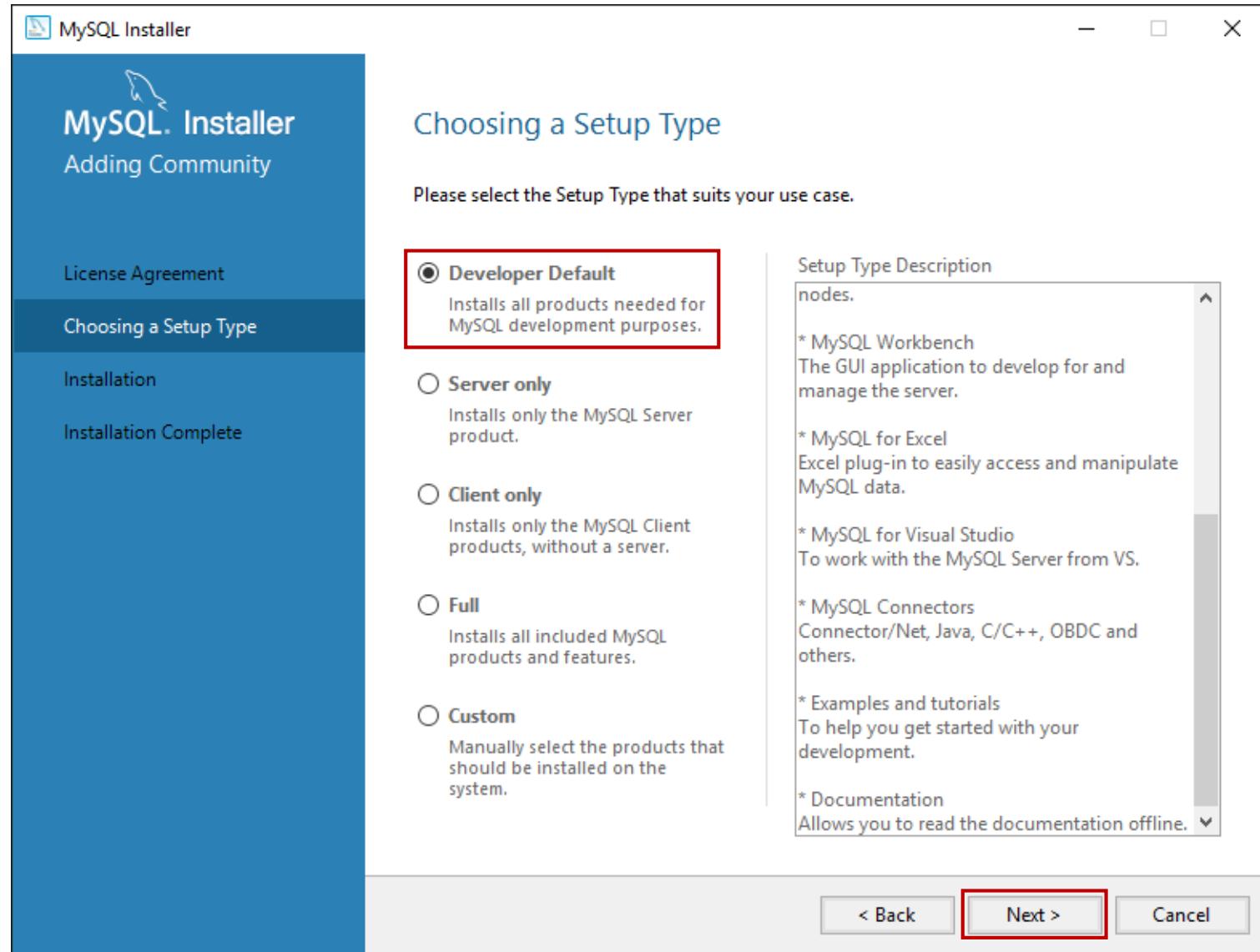
Version	File Size	Download
Windows (x86, 32-bit), MSI Installer (mysql-installer-web-community-8.0.13.0.msi)	8.0.13 16.3M	Download
Windows (x86, 32-bit), MSI Installer (mysql-installer-community-8.0.13.0.msi)	8.0.13 313.8M	Download

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

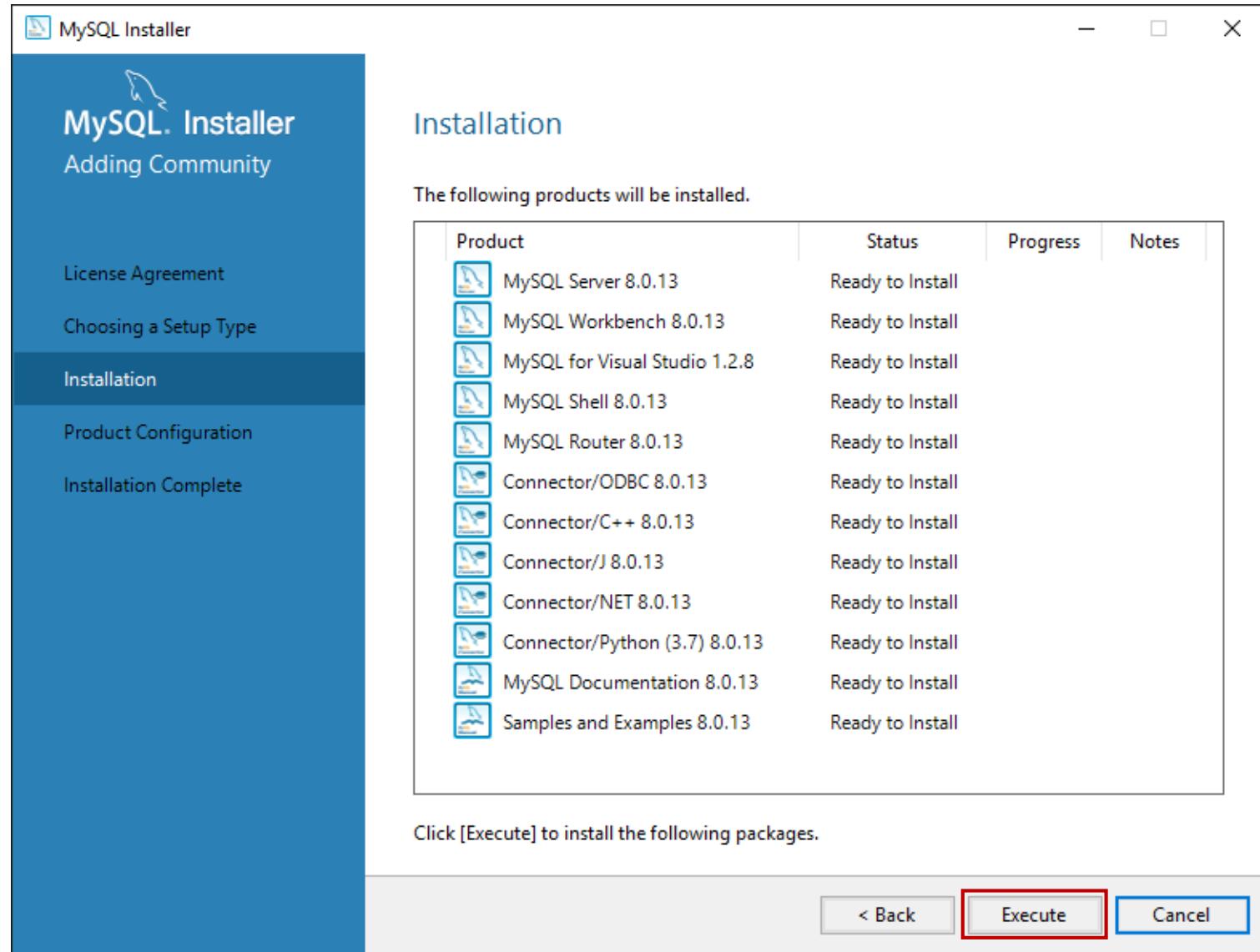
MySQL 설치



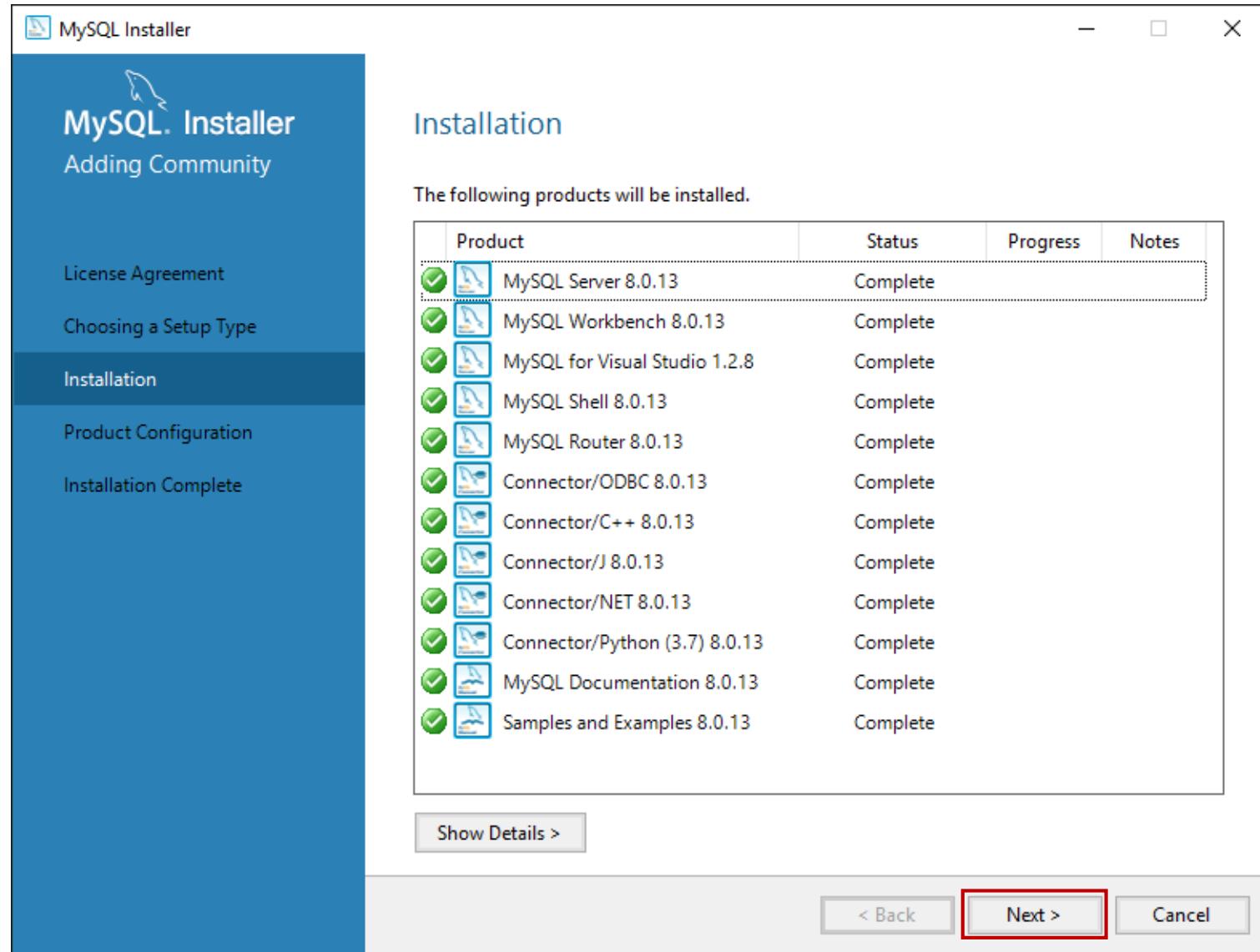
MySQL 설치



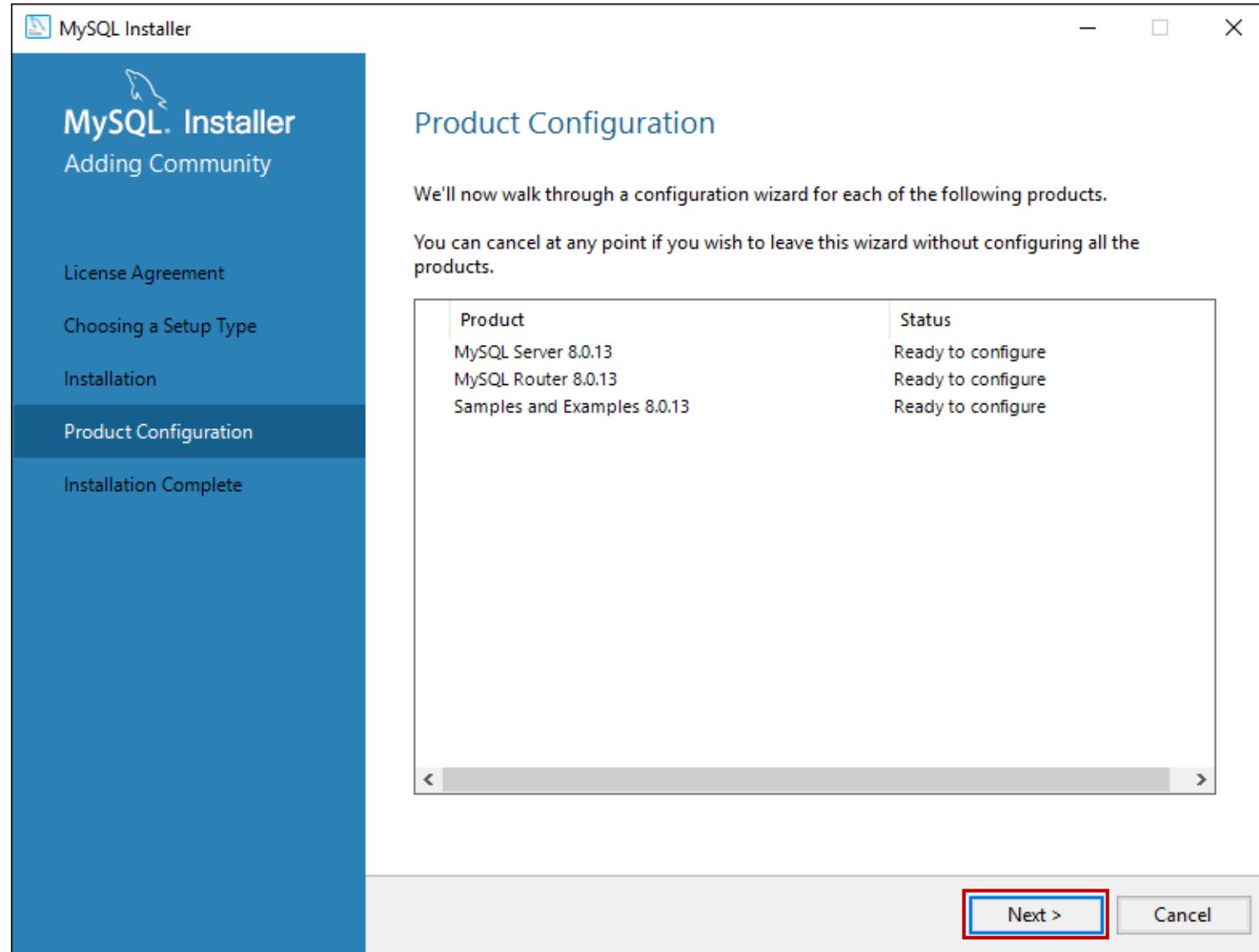
MySQL 설치



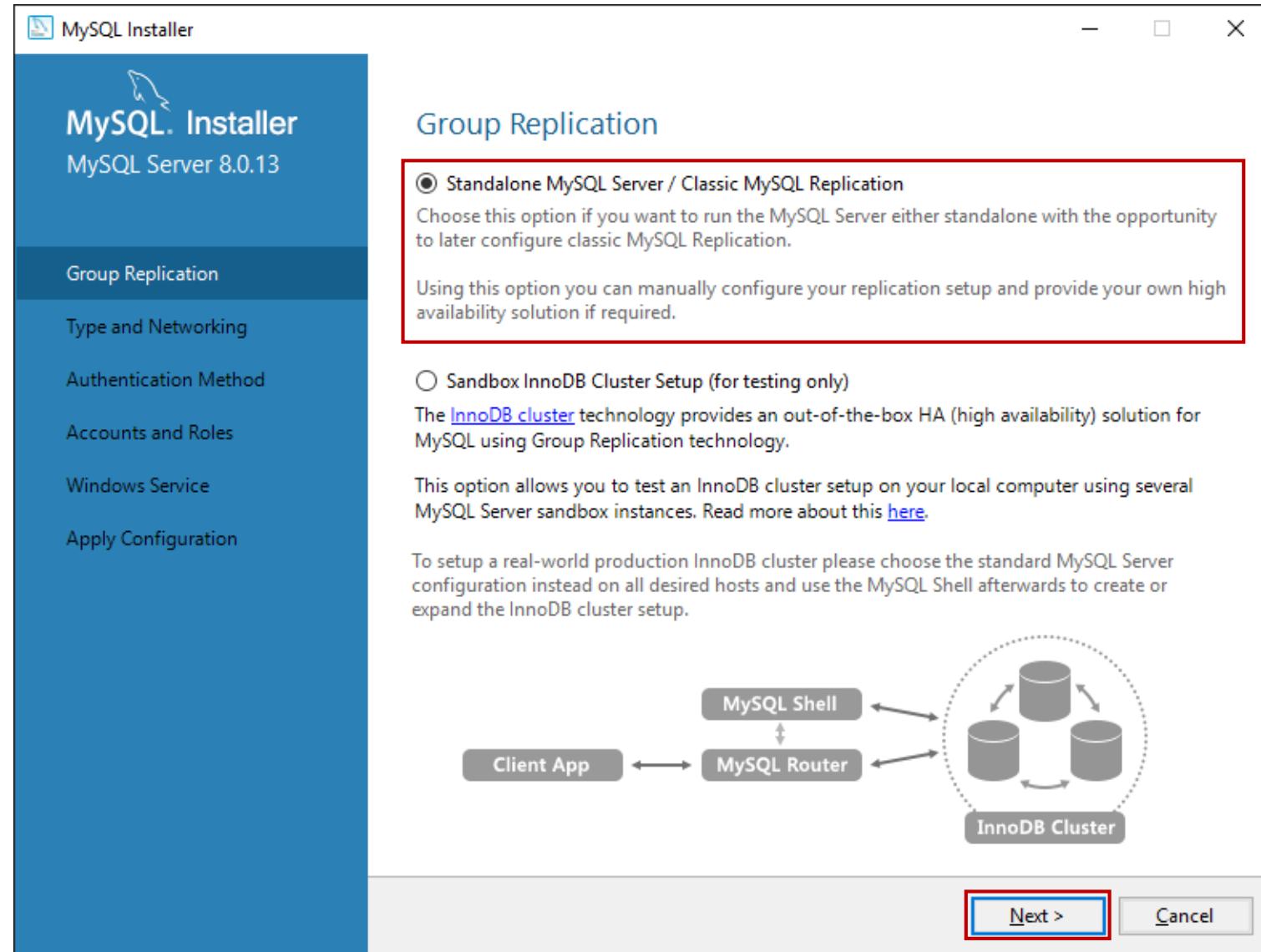
MySQL 설치



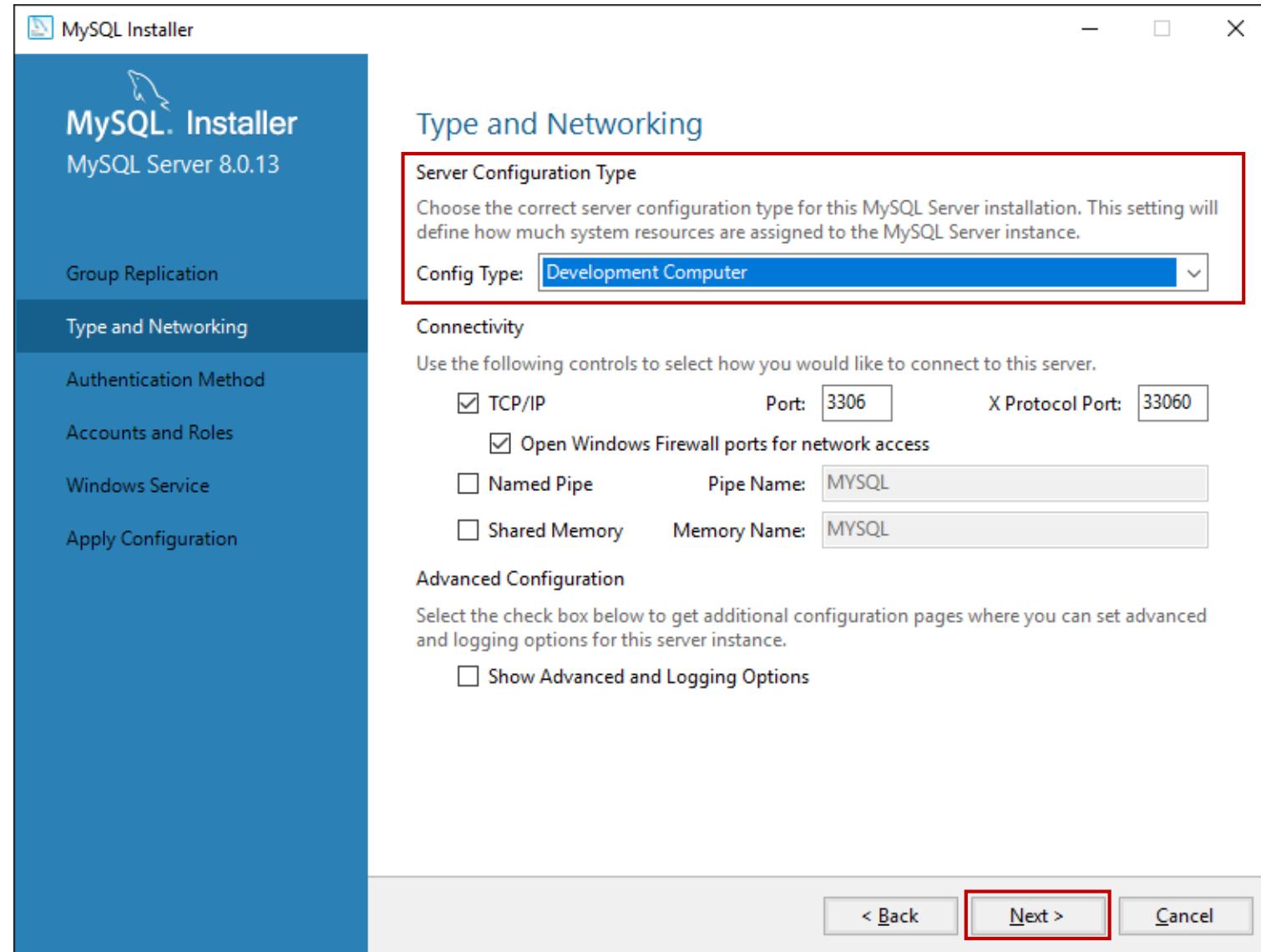
MySQL 설치



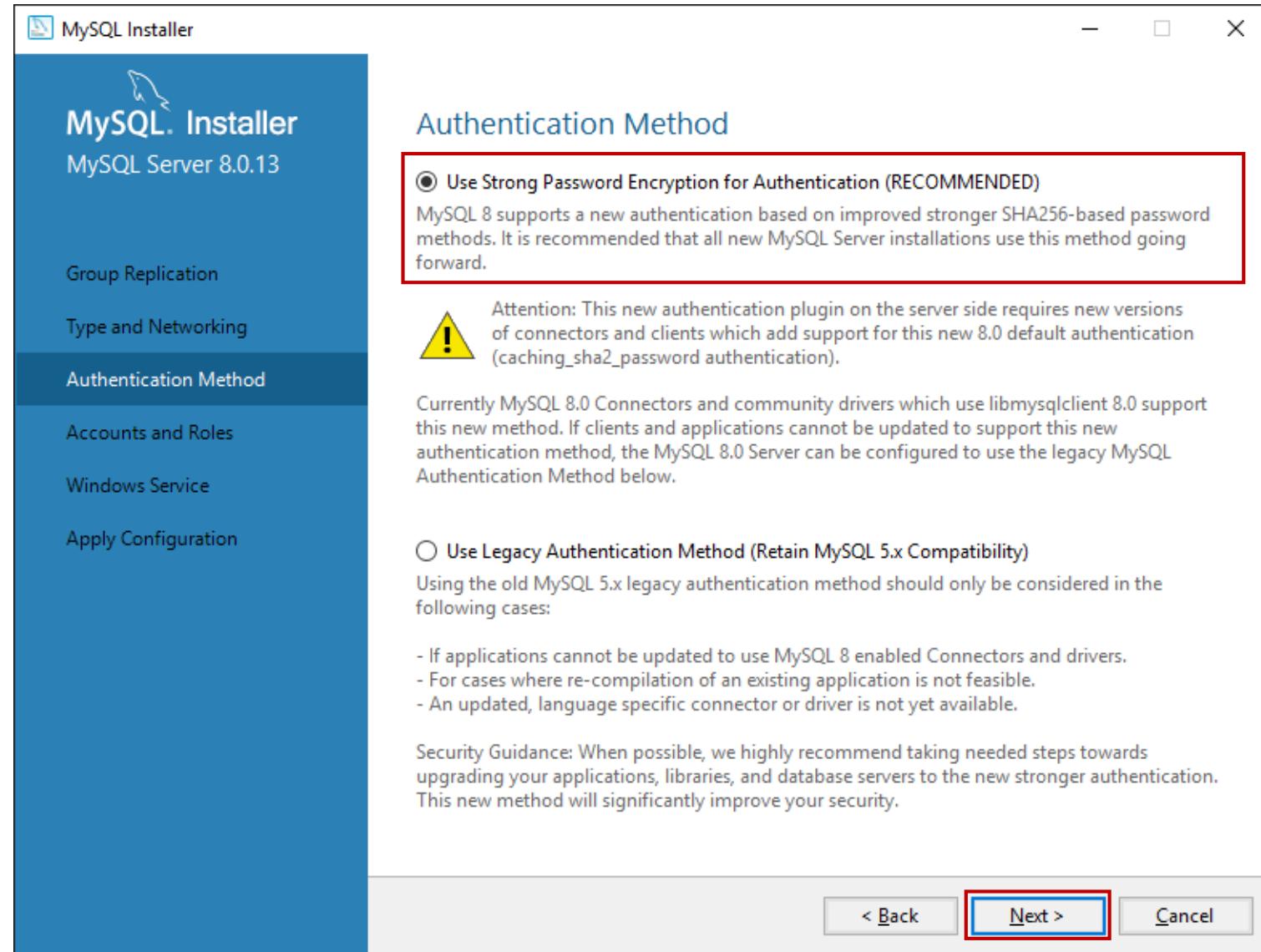
MySQL 설치



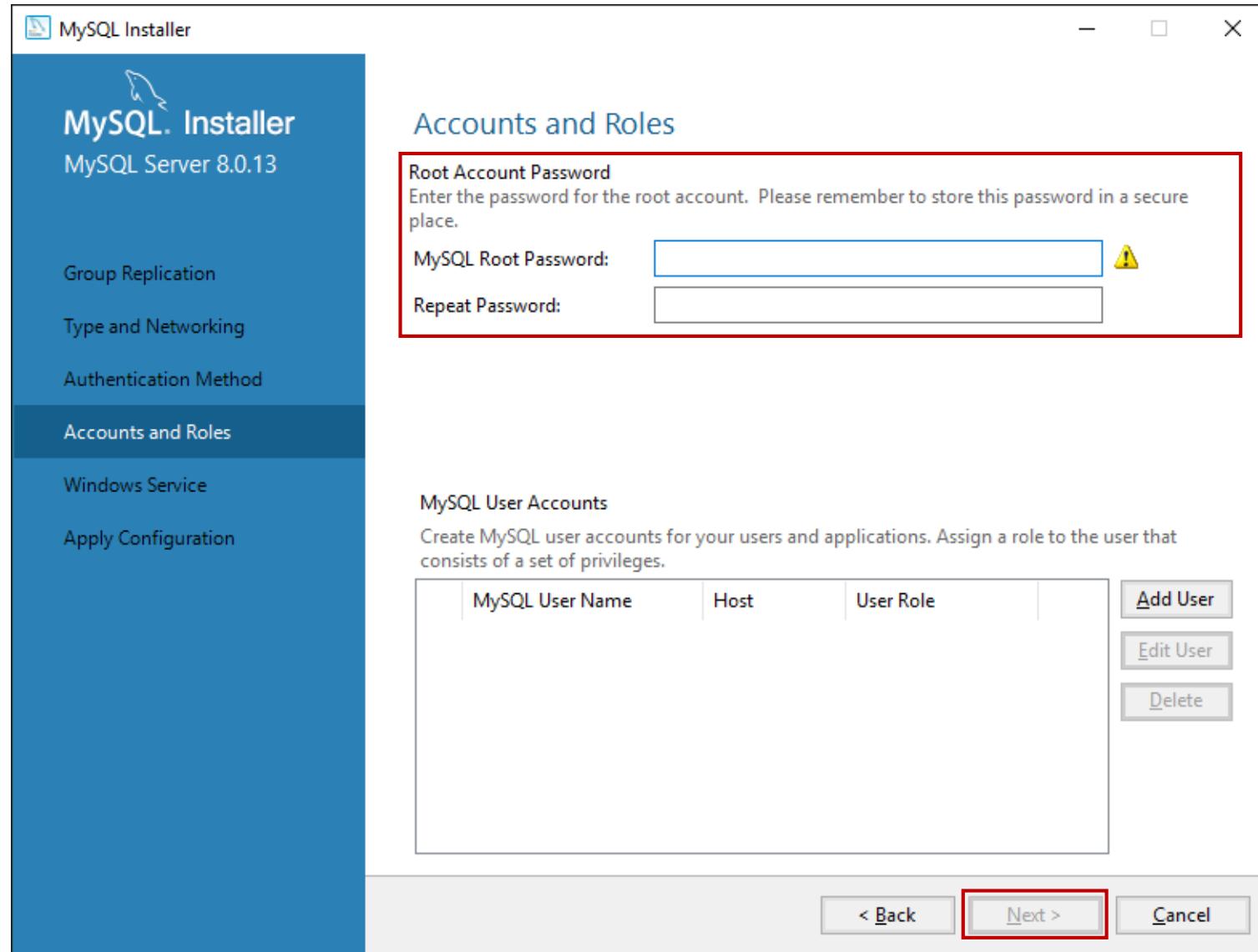
MySQL 설치



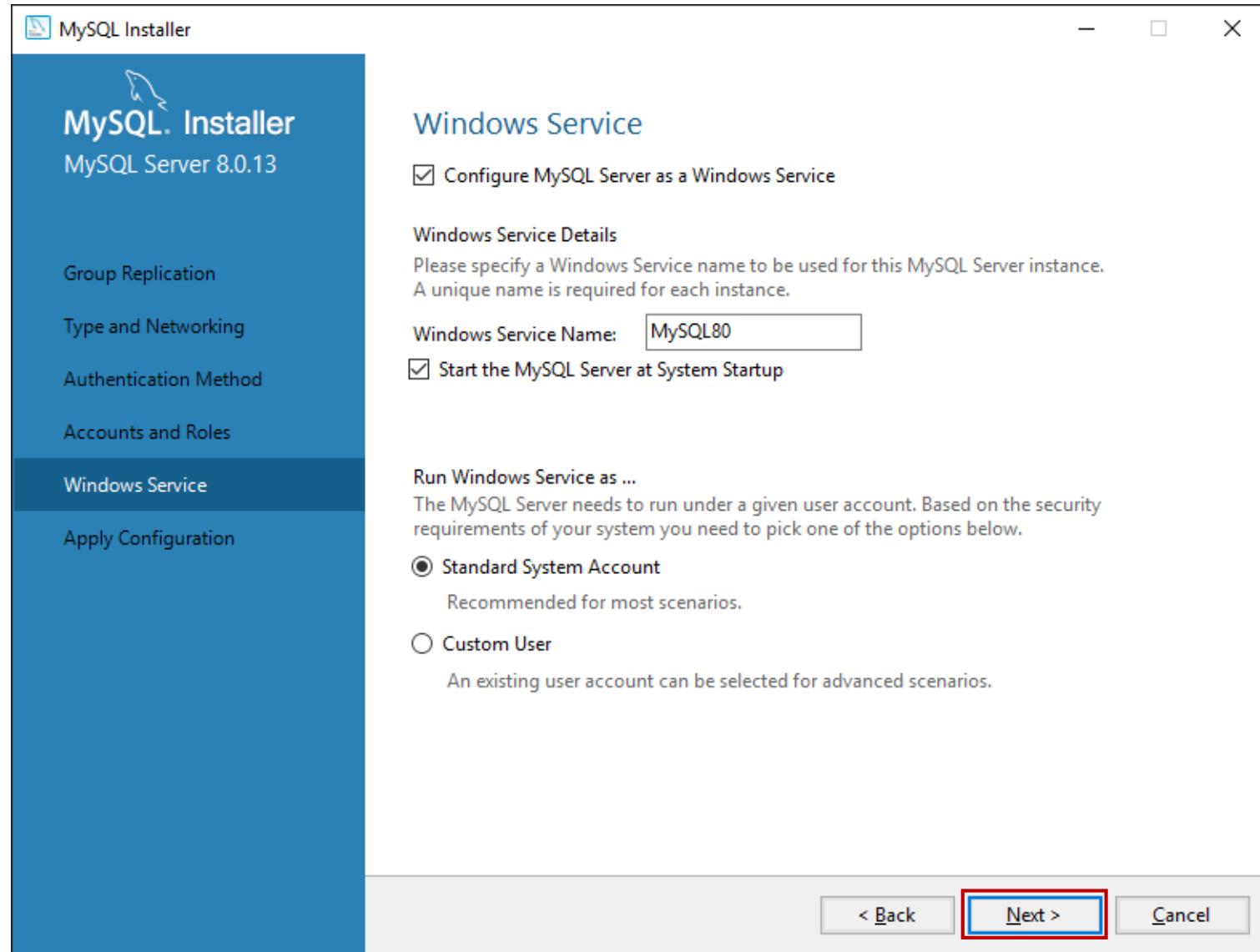
MySQL 설치



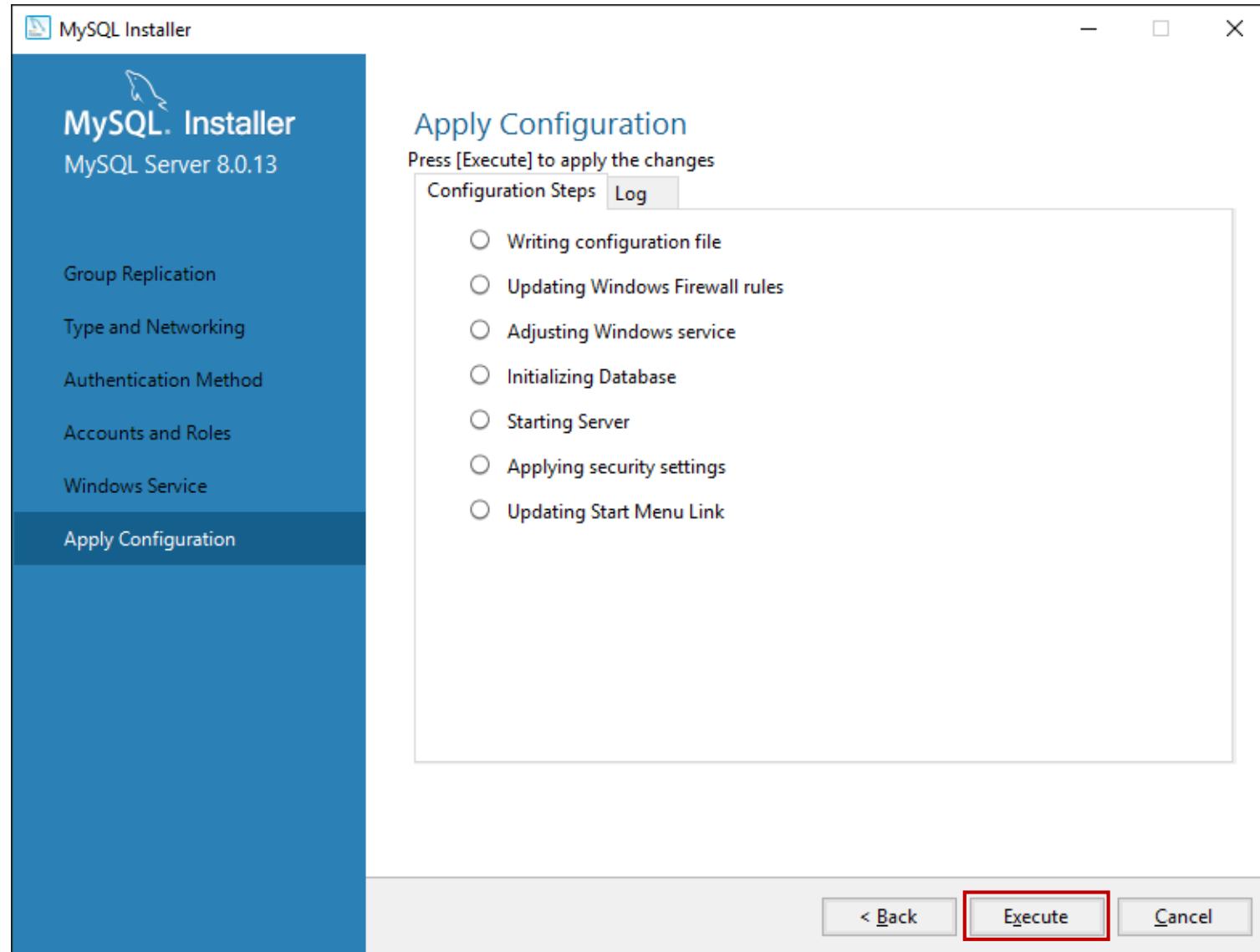
MySQL 설치



MySQL 설치

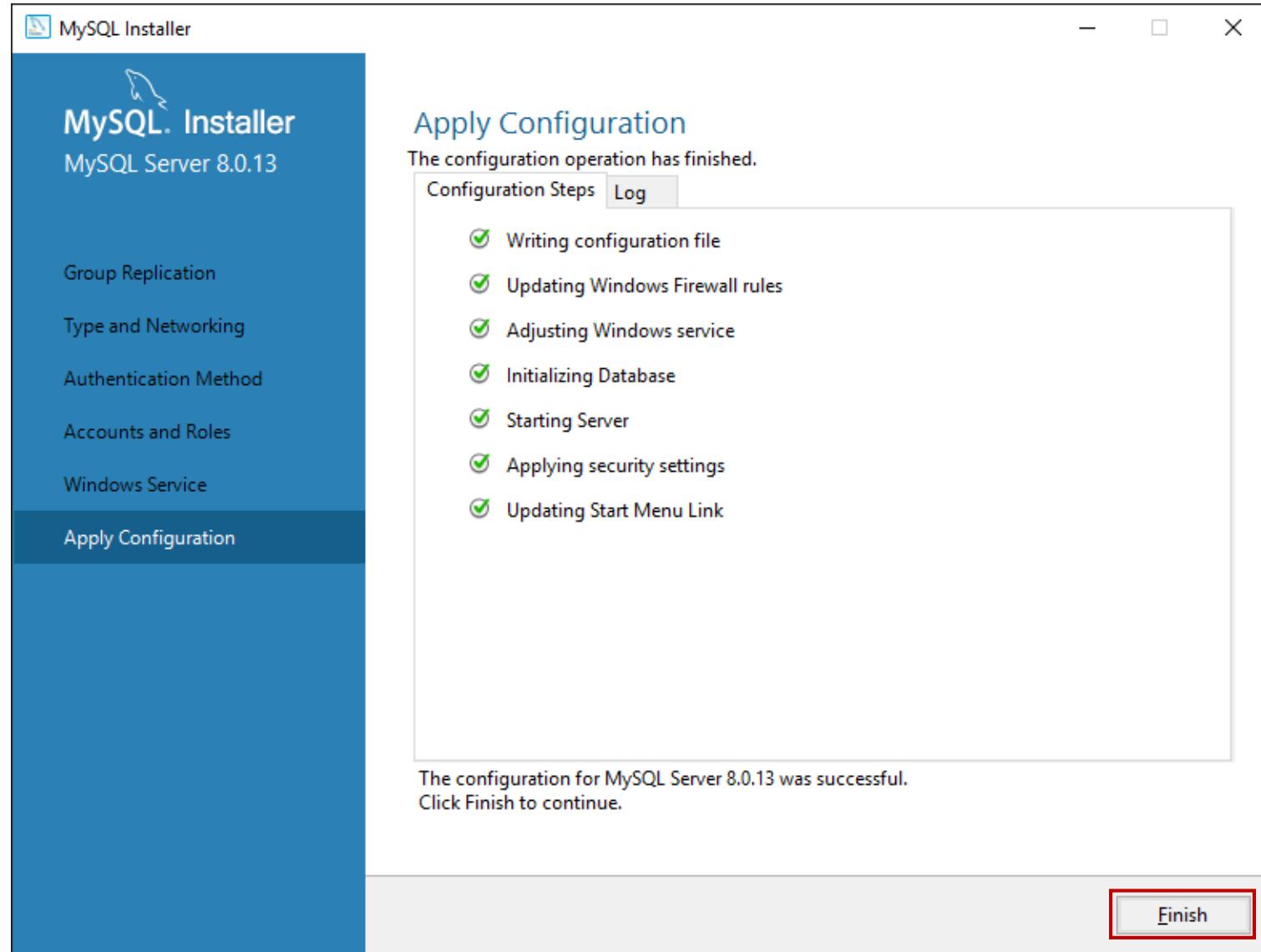


MySQL 설치

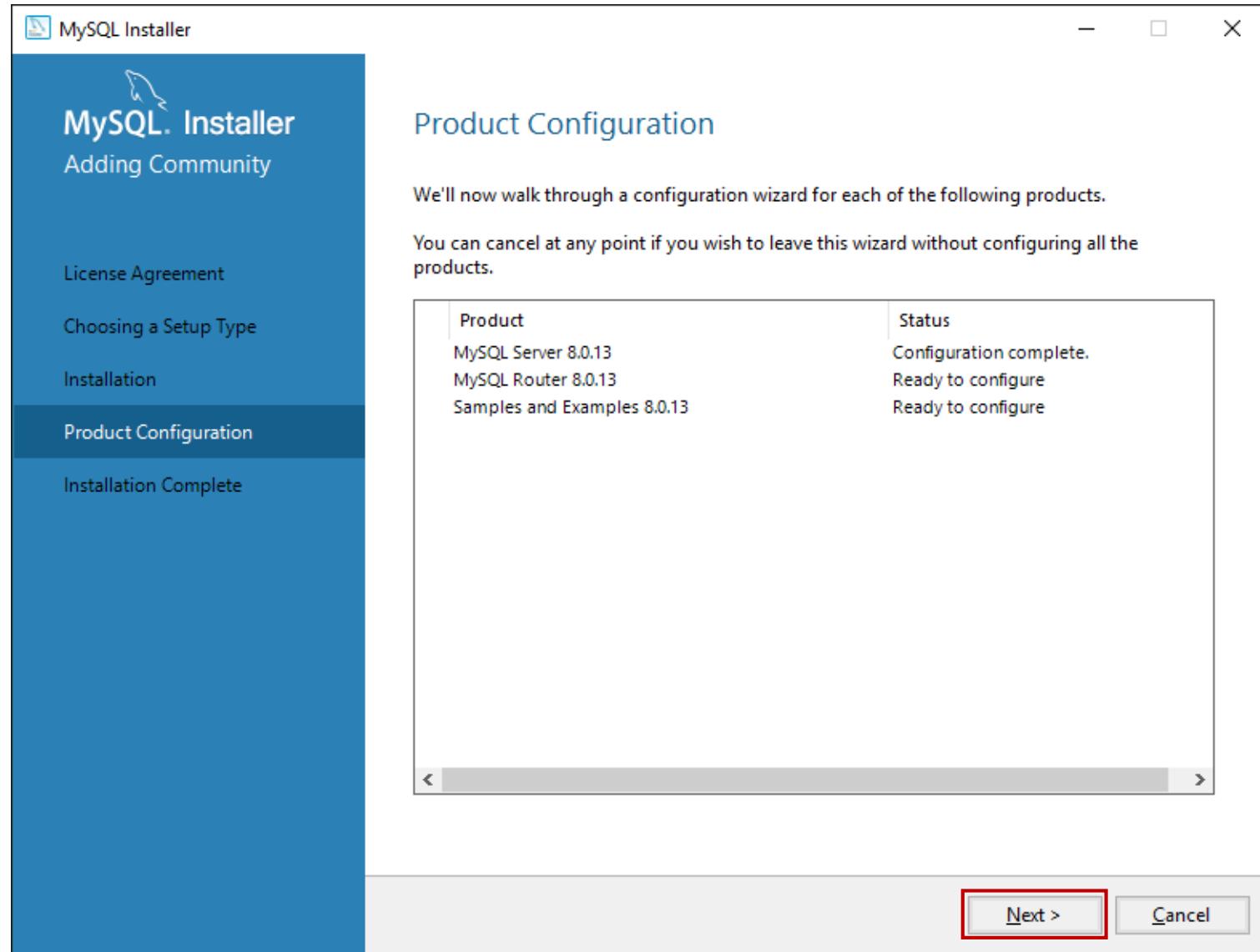


- MySQL 한번에 끝내기

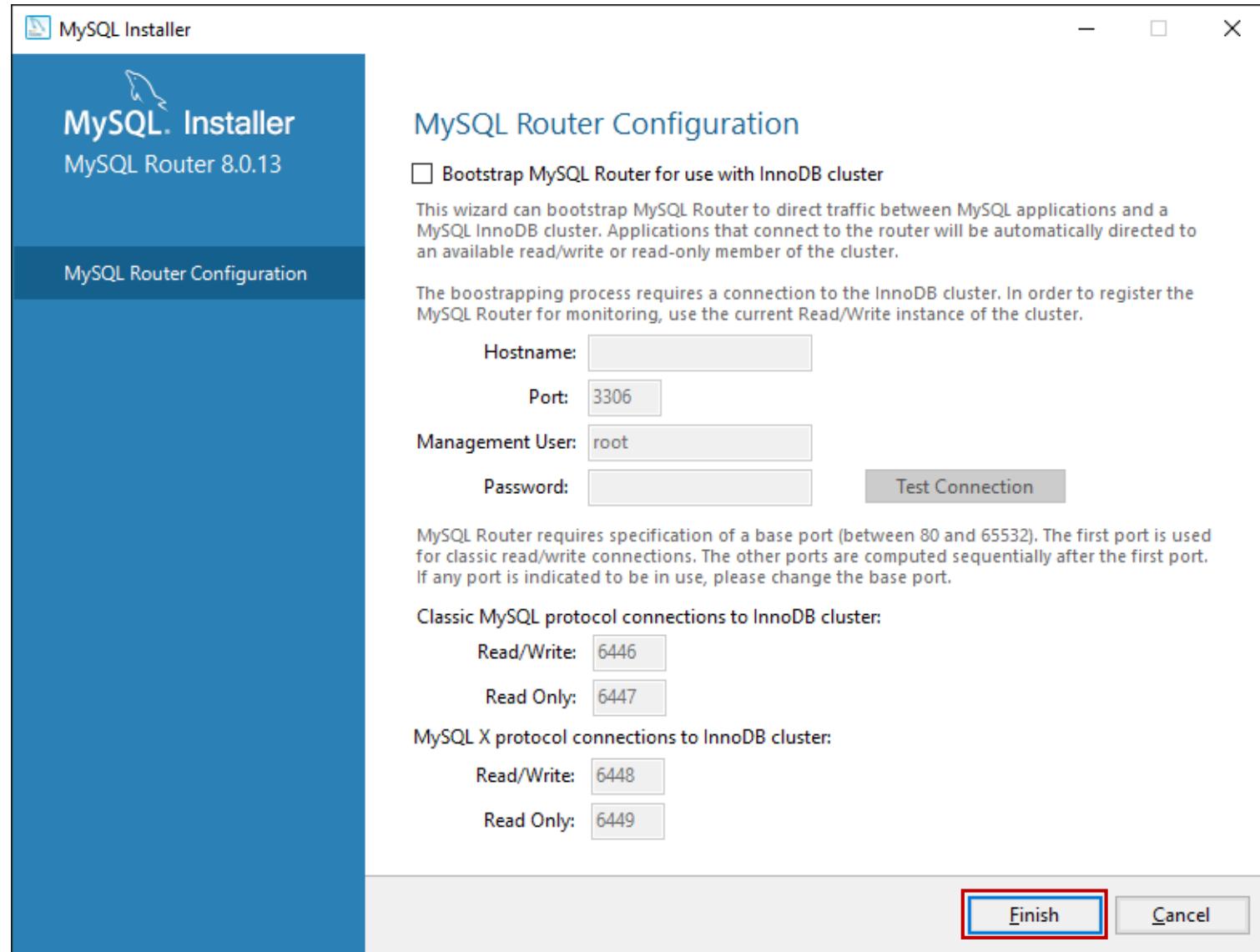
MySQL 설치



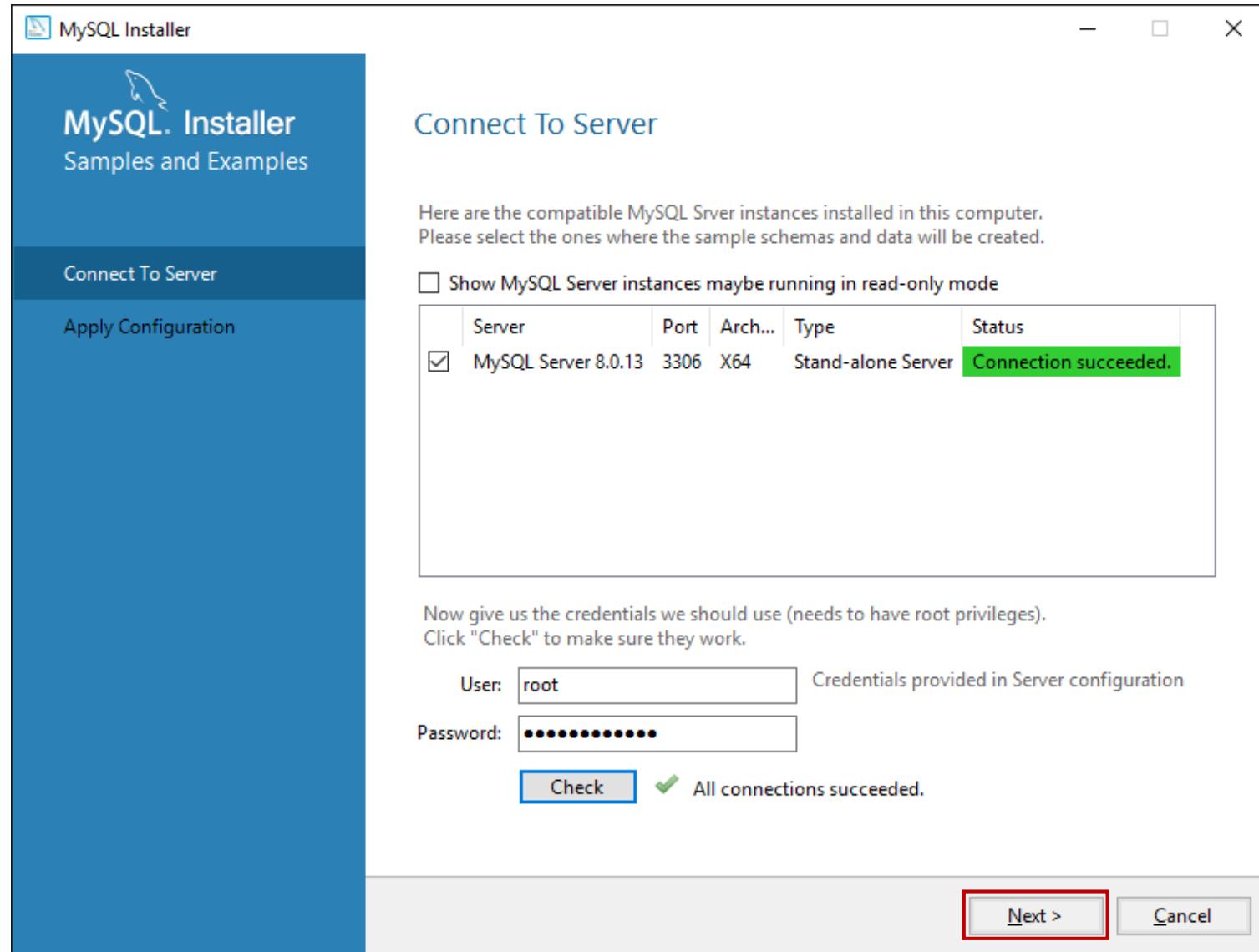
MySQL 설치



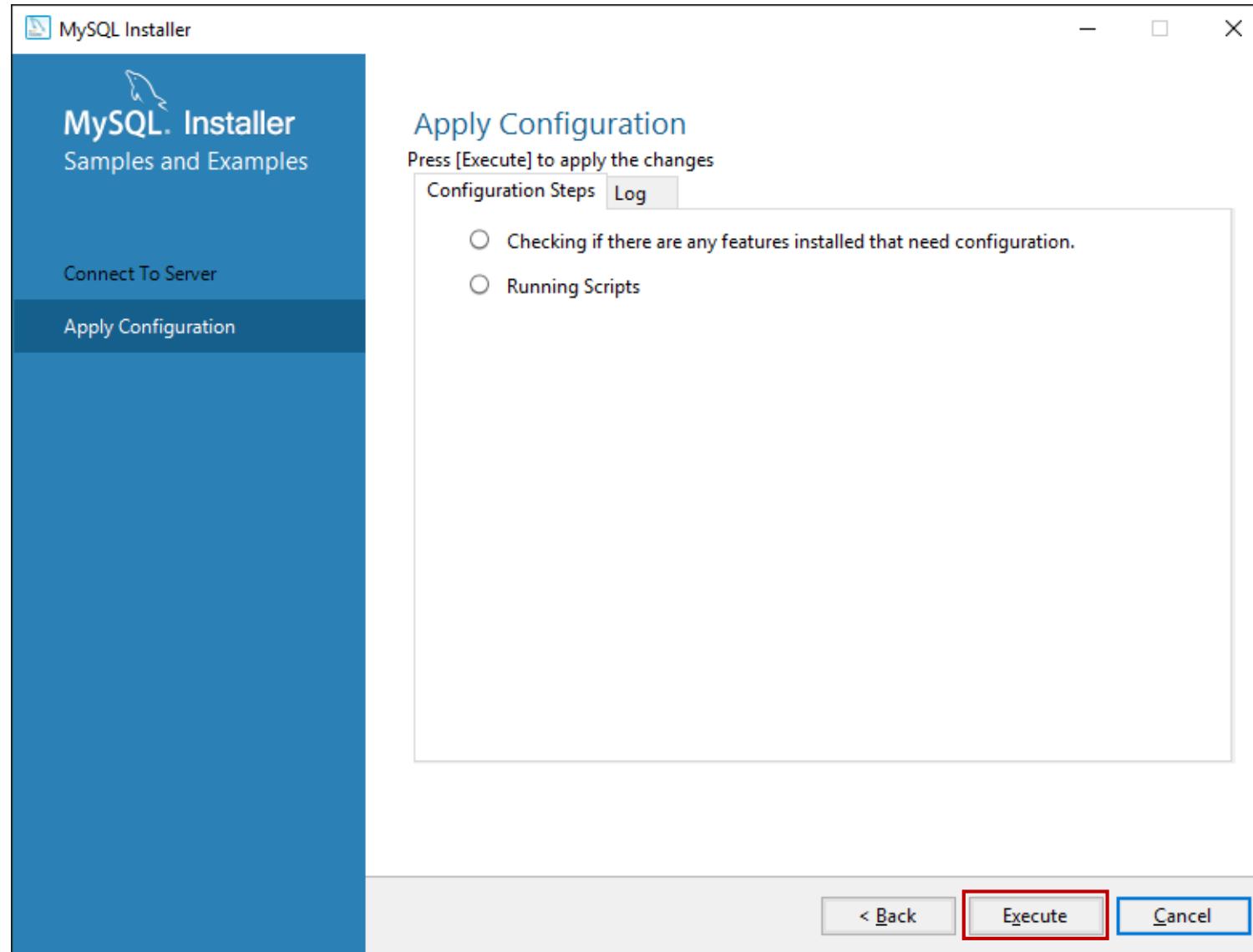
MySQL 설치



MySQL 설치

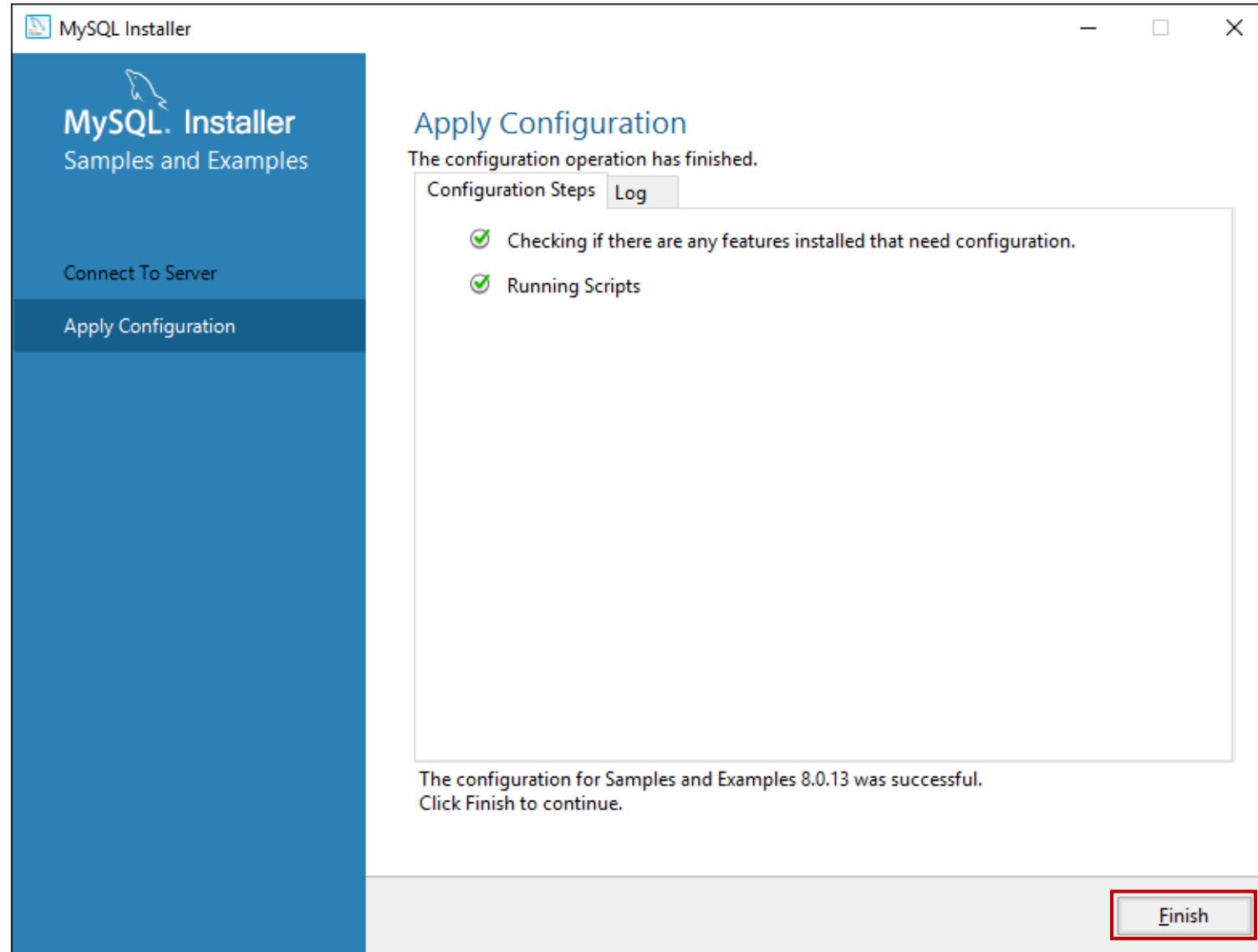


MySQL 설치

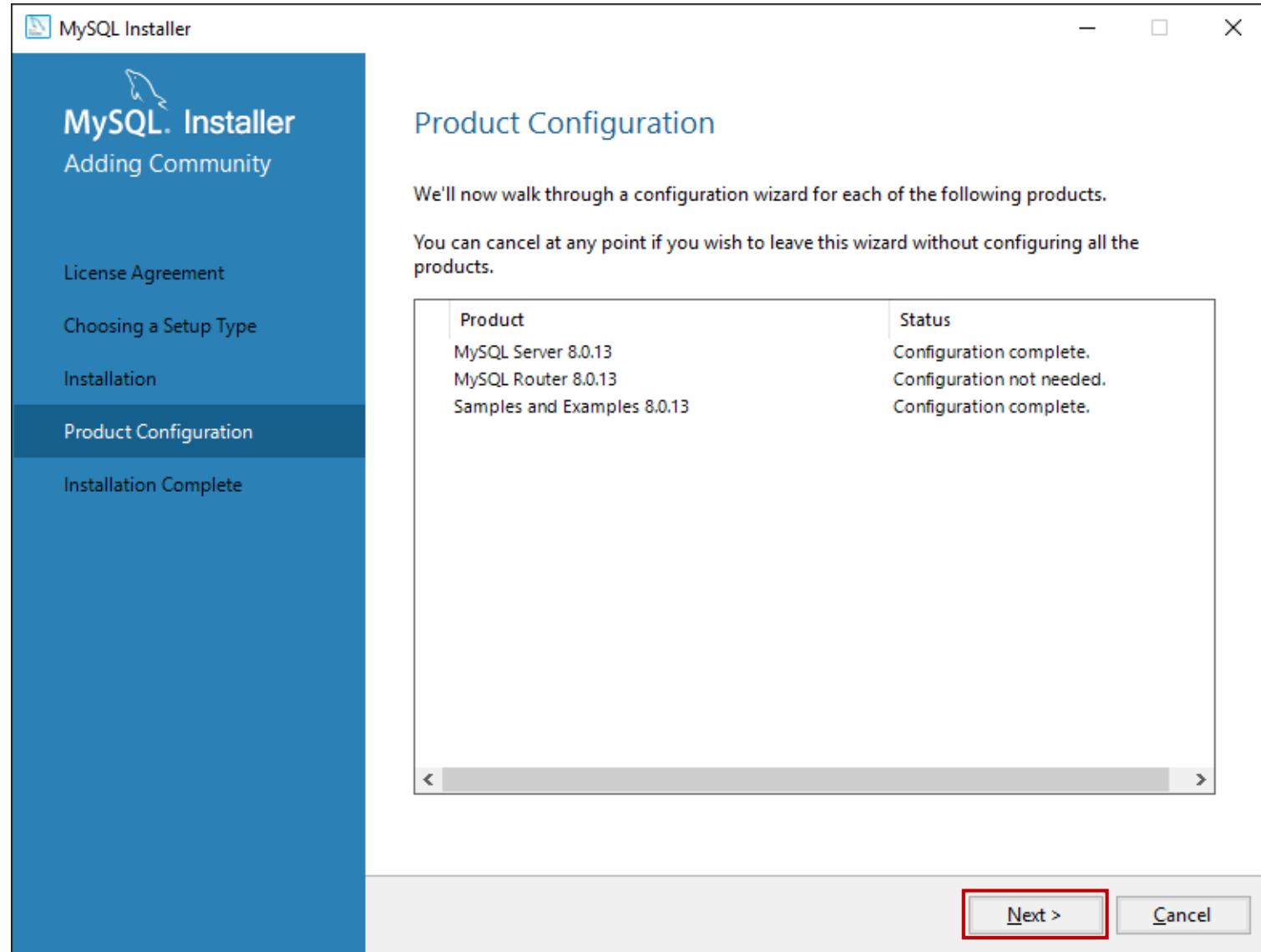


- MySQL 한번에 끝내기

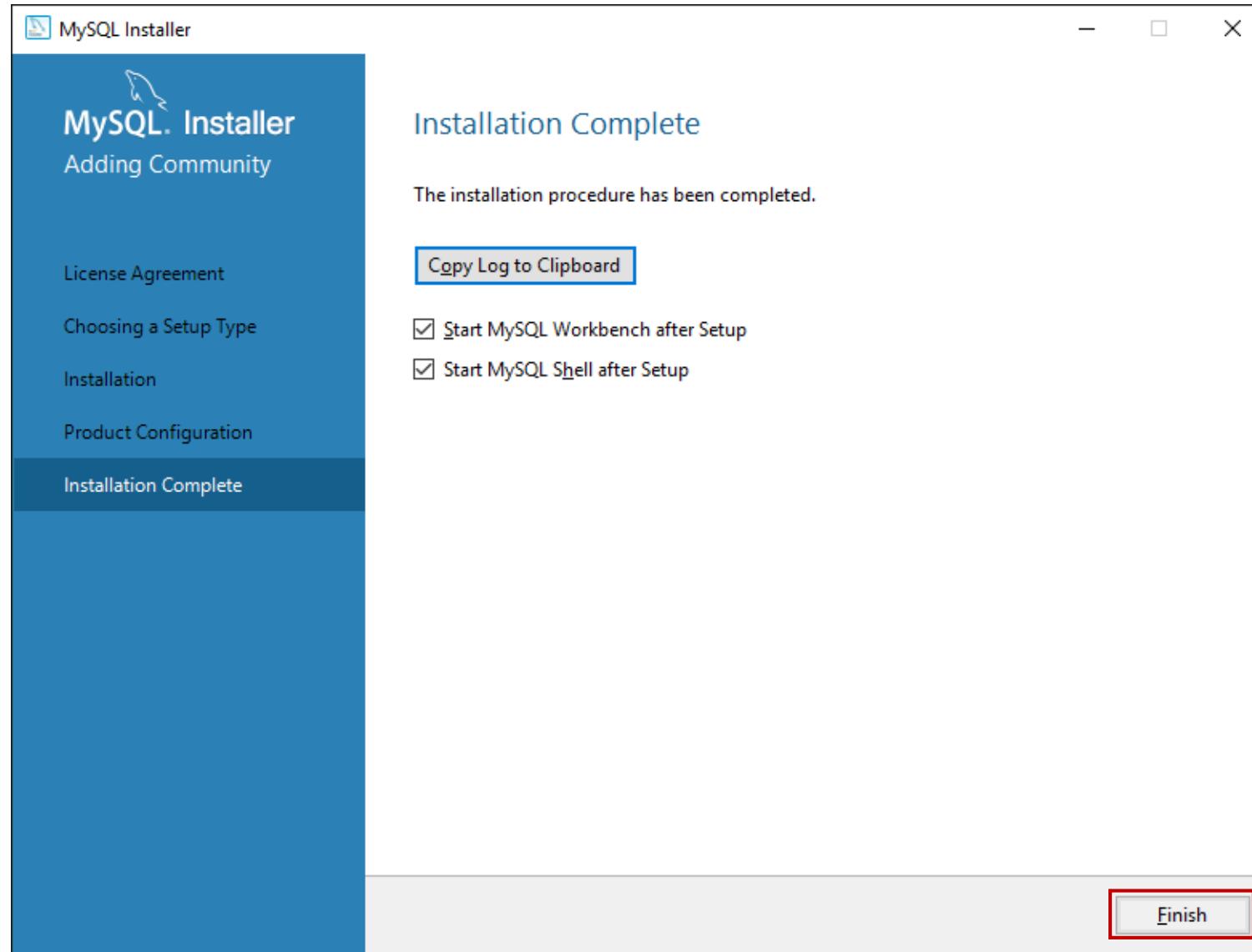
MySQL 설치



MySQL 설치

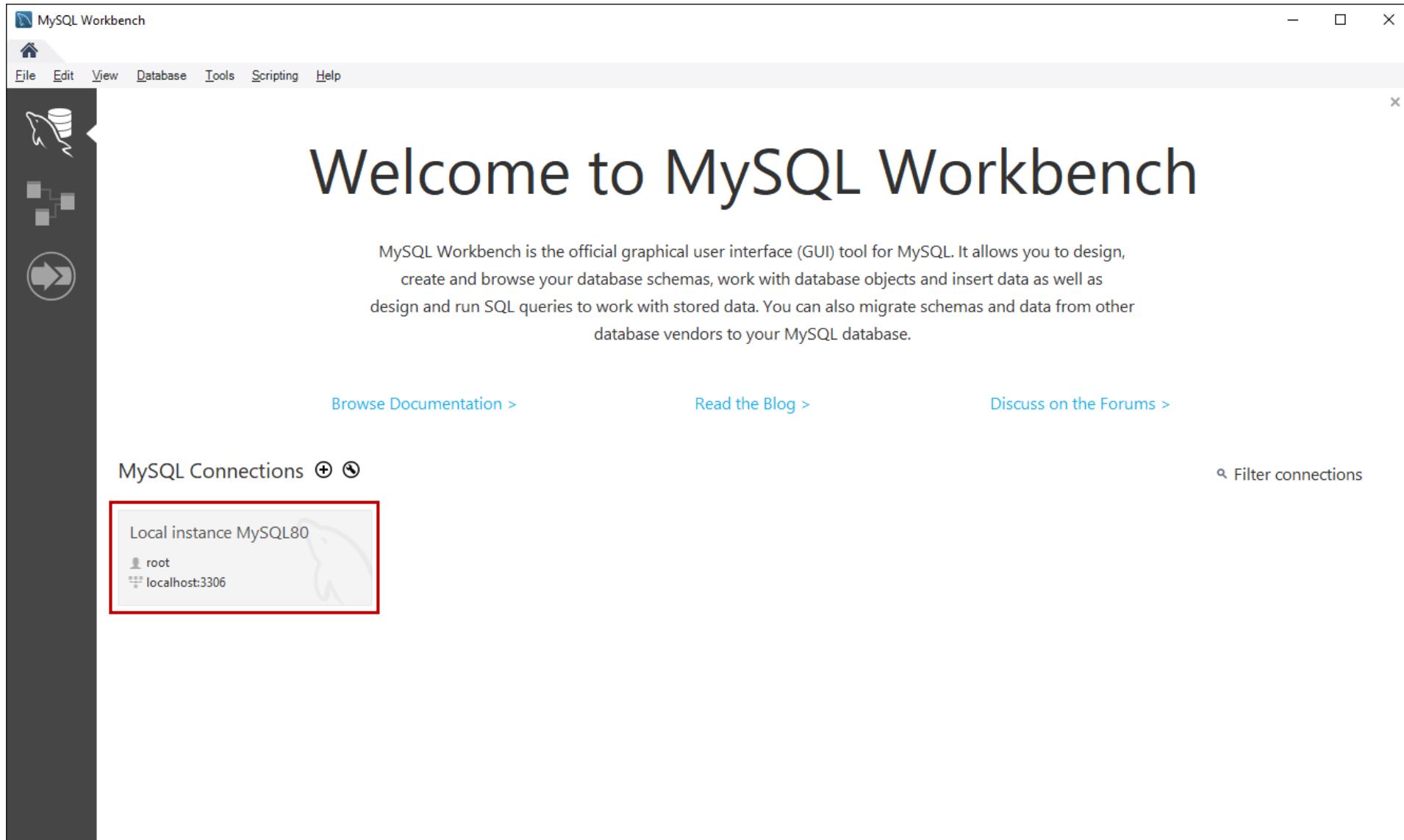


MySQL 설치

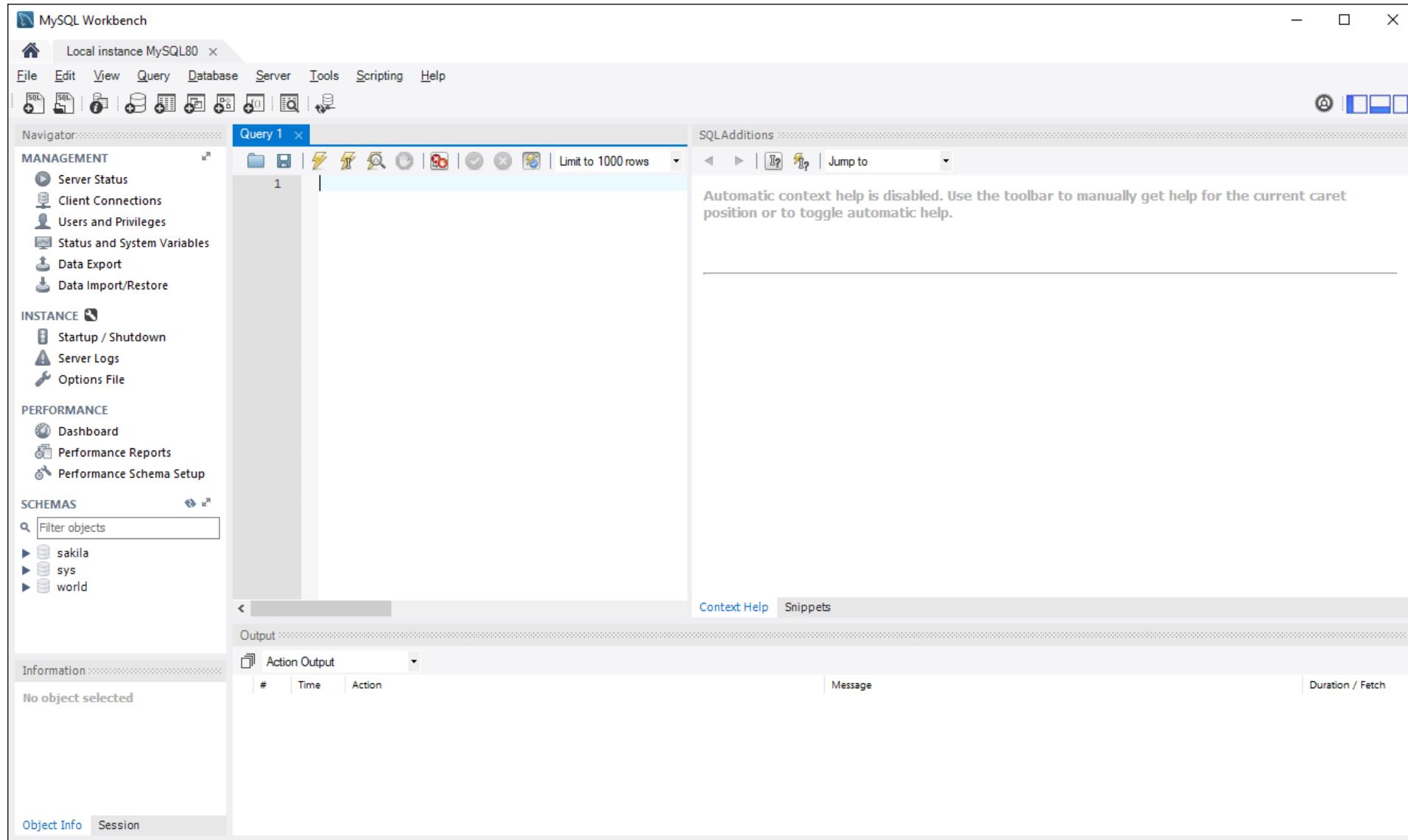


- MySQL 한번에 끝내기

MySQL Workbench

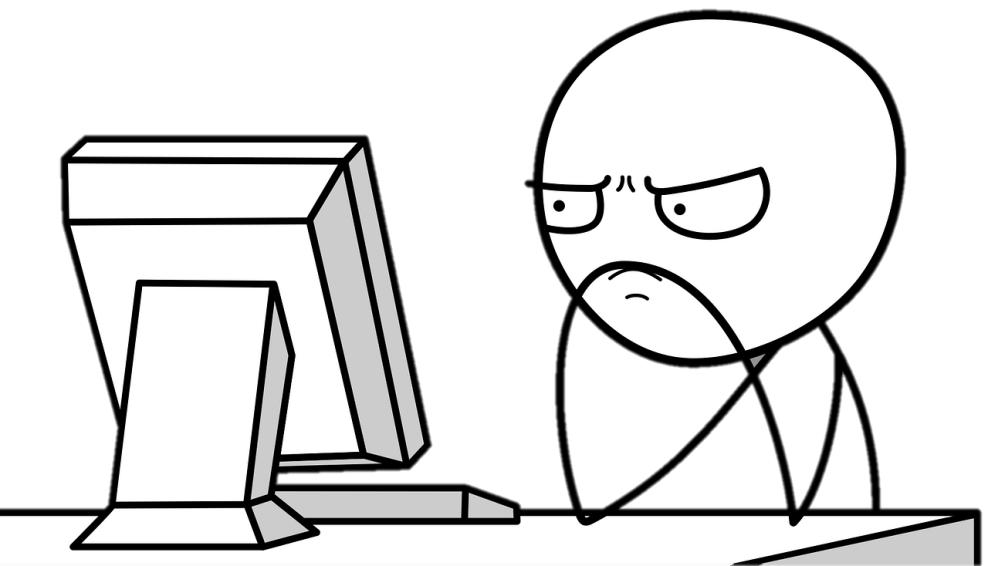


MySQL Workbench



- MySQL 한번에 끝내기

3. SQL 기본



SQL의 분류

■ DML Data Manipulation Language

- 데이터 조작 언어
- 데이터를 조작(선택, 삽입, 수정, 삭제)하는데 사용되는 언어
- DML 구문이 사용되는 대상은 테이블의 행
- DML 사용하기 위해서는 꼭 그 이전에 테이블이 정의되어 있어야 함
- SQL문 중 SELECT, INSERT, UPDATE, DELETE가 이 구문에 해당
- 트랜잭션 Transaction이 발생하는 SQL도 이 DML에 속함
 - 테이블의 데이터를 변경(입력/수정/삭제)할 때 실제 테이블에 완전히 적용하지 않고, 임시로 적용시키는 것
 - 취소 가능

■ DDL Data Definition Language

- 데이터 정의 언어
- 데이터베이스, 테이블, 뷰, 인덱스 등의 데이터베이스 객체를 생성/삭제/변경하는 역할
- CREATE, DROP, ALTER 구문
- DDL은 트랜잭션 발생시키지 않음
- ROLLBACK이나 COMMIT 사용 불가
- DDL문은 실행 즉시 MySQL에 적용

■ DCL Data Control Language

- 데이터 제어 언어
- 사용자에게 어떤 권한을 부여하거나 빼앗을 때 주로 사용하는 구문
- GRANT/REVOKE/DENY 구문

SHOW DATABASES

- 현재 서버에 어떤 DB가 있는지 보기

The screenshot shows the MySQL Workbench interface. At the top, there is a toolbar with various icons. Below the toolbar, a SQL editor window titled "SQL File 1" contains the command "SHOW DATABASES;". In the main area, a "Result Grid" displays the output of the query:

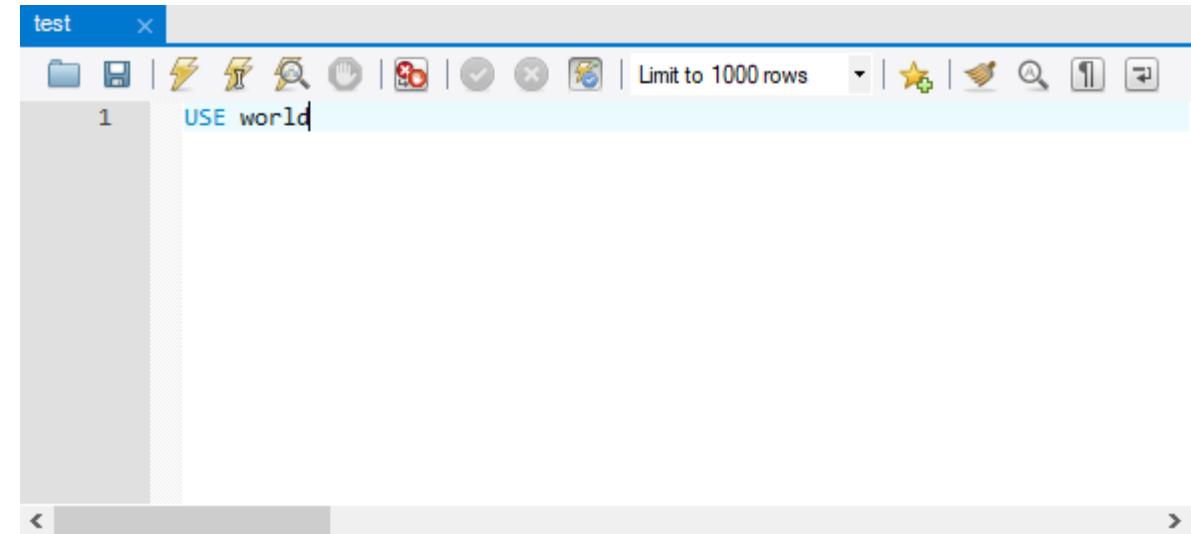
Database
information_schema
mysql
performance_schema
sakila
sys
world

The "Result Grid" tab is selected at the bottom of the results pane. To the right of the results pane, there is a vertical toolbar with four options: "Result Grid" (selected), "Form Editor", "Field Types", and "Query Stats". At the bottom right of the results pane, there is a "Read Only" status indicator.

USE

- 사용할 데이터베이스 지정
- 지정해 놓은 후 특별히 다시 USE문 사용하거나 다른 DB를 사용하겠다고 명시하지 않는 이상 모든 SQL문은 지정 DB에서 수행

```
USE database_name
```



- Workbench에서 직접 선택해 사용 가능
 - [Navigator] → [SCHEMAS] → 데이터베이스 선택

SHOW TABLE

- 데이터베이스 world의 테이블 이름 보기

The screenshot shows the MySQL Workbench interface. In the top SQL editor tab, the command `SHOW TABLES;` is entered. The results are displayed in a grid titled "Tables_in_world" under the "Result Grid" tab. The grid contains three rows: "city", "country", and "countrylanguage". The "country" row is currently selected. On the right side of the interface, there is a vertical toolbar with four icons: "Result Grid" (selected), "Form Editor", "Field Types", and "Query Stats". At the bottom right of the interface, there is a "Read Only" status indicator.

	Tables_in_world
▶	city
▶	country
▶	countrylanguage

SHOW TABLE STATUS

- 데이터베이스 world의 테이블 정보 조회

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

- SQL Editor:** Shows the query `SHOW TABLE STATUS;` entered.
- Result Grid:** Displays the results of the query in a tabular format. The columns are: Name, Engine, Version, Row_format, Rows, Avg_row_length, Data_length, and Extra.
- Results:**

Name	Engine	Version	Row_format	Rows	Avg_row_length	Data_length	Extra
city	InnoDB	10	Dynamic	4188	97	409600	0
country	InnoDB	10	Dynamic	239	411	98304	0
countrylanguage	InnoDB	10	Dynamic	984	99	98304	0
- Right Panel:** Shows icons for Result Grid, Form Editor, Field Types, and Query Stats.
- Status Bar:** Shows "Result 8" and "Read Only".

DESCRIBE (DESC)

■ city 테이블에 무슨 열이 있는지 확인

- DESCRIBE city;
- DESC city;

The screenshot shows the MySQL Workbench interface with a query editor window titled 'test'. The query 'DESCRIBE city' is entered in the text area. Below the query, the results are displayed in a 'Result Grid' table:

	Field	Type	Null	Key	Default	Extra
▶	ID	int(11)	NO	PRI	NULL	auto_increment
	Name	char(35)	NO			
	CountryCode	char(3)	NO	MUL		
	District	char(20)	NO			
	Population	int(11)	NO		0	

The right side of the interface features a vertical toolbar with icons for Result Grid, Form Editor, Field Types, and Query Stats.

country 테이블과 countrylanguage 테이블 정보 보기

SELECT

- <SELECT... FROM>
- 요구하는 데이터를 가져오는 구문
- 일반적으로 가장 많이 사용되는 구문
- 데이터베이스 내 테이블에서 원하는 정보를 추출
- SELECT의 구문 형식

```
SELECT select_expr  
      [FROM table_references]  
      [WHERE where_condition]  
      [GROUP BY {col_name | expr | position}]  
      [HAVING where_condition]  
      [ORDER BY {col_name | expr | position}]
```

SELECT

■ SELECT *

The screenshot shows the MySQL Workbench interface with a query editor and a result grid.

Query Editor:

```
test
1
SELECT * FROM city
```

Result Grid:

	ID	Name	CountryCode	District	Population
▶	1	Kabul	AFG	Kabul	1780000
	2	Qandahar	AFG	Qandahar	237500
	3	Herat	AFG	Herat	186800
	4	Mazar-e-Sharif	AFG	Balkh	127800
	5	Amsterdam	NLD	Noord-Holland	731200
	6	Rotterdam	NLD	Zuid-Holland	593321
	7	Haag	NLD	Zuid-Holland	440900
	8	Utrecht	NLD	Utrecht	234323
	9	Eindhoven	NLD	Noord-Brabant	201843
	10	Tilburg	NLD	Noord-Brabant	193238
	11	Groningen	NLD	Groningen	172701
	12	Breda	NLD	Noord-Brabant	160398
	13	Apeldoorn	NLD	Gelderland	153491
	14	Nijmeegen	NLD	Gelderland	152463

Toolbar: The toolbar includes icons for file operations, search, and various database management functions. A "Limit to 1000 rows" dropdown is also present.

Right Panel: A sidebar on the right provides quick access to "Result Grid", "Form Editor", "Field Types", and "Query Stats".

SELECT

■ SELECT 열 이름

- 테이블에서 필요로 하는 열만 가져오기 가능
- 여러 개의 열을 가져오고 싶을 때는 콤마로 구분
- 열 이름의 순서는 출력하고 싶은 순서대로 배열 가능

The screenshot shows the MySQL Workbench interface with a query editor window titled 'city'. The query entered is:

```
SELECT Name, District FROM city;
```

The results are displayed in a 'Result Grid' table:

	Name	District
▶	Kabul	Kabul
	Qandahar	Qandahar
	Herat	Herat
	Mazar-e-Sharif	Balkh
	Amsterdam	Noord-Holland
	Rotterdam	Zuid-Holland
	Haag	Zuid-Holland
	Utrecht	Utrecht
	Eindhoven	Noord-Brabant
	Tilburg	Noord-Brabant
	Groningen	Groningen
	Breda	Noord-Brabant
	Apeldoorn	Gelderland
	Nijmegen	Gelderland

The interface includes various toolbars and panels on the right side, such as 'Result Grid', 'Form Editor', 'Field Types', and 'Query Stats'.

SELECT FROM WHERE

■ 기본적인 WHERE절

- 조회하는 결과에 특정한 조건 줘서 원하는데 이터만 보고 싶을 때 사용
- SELECT 필드이름 FROM 테이블이름 WHERE 조건식;
- 조건이 없을 경우 테이블의 크기가 클수록 찾는 시간과 노력이 증가

The screenshot shows the MySQL Workbench interface. At the top, there is a toolbar with various icons. Below it is a query editor window titled "SQL File 1" containing the following SQL code:

```
1 • SELECT *
 2   FROM city
 3 WHERE Population > 8000000
```

Below the query editor is a "Result Grid" window displaying the results of the query. The grid has columns: ID, Name, CountryCode, District, and Population. The data includes rows for São Paulo, Jakarta, Mumbai, Shanghai, Seoul, Ciudad de México, Karachi, Istanbul, Moscow, and New York. The "Population" column for São Paulo is listed as 9968485.

ID	Name	CountryCode	District	Population
206	São Paulo	BRA	São Paulo	9968485
939	Jakarta	IDN	Jakarta Raya	9604900
1024	Mumbai (Bombay)	IND	Maharashtra	10500000
1890	Shanghai	CHN	Shanghai	9696300
2331	Seoul	KOR	Seoul	9981619
2515	Ciudad de México	MEX	Distrito Federal	8591309
2822	Karachi	PAK	Sindh	9269265
3357	Istanbul	TUR	Istanbul	8787958
3580	Moscow	RUS	Moscow (City)	8389200
3793	New York	USA	New York	8008278
*	NULL	NULL	NULL	NULL

SELECT FROM WHERE

■ 관계 연산자의 사용

- '...했거나', '... 또는' -OR 연산자
- '...하고', '...면서', '... 그리고' -AND 연산자
- 조건 연산자(=, <, >, <=, >=, <>, != 등)와 관계 연산자(NOT, AND, OR 등)의 조합으로 알맞은 데이터를 효율적으로 추출

■ MySQL 함수 및 연산자:

<https://dev.mysql.com/doc/refman/8.0/en/functions.html>

The screenshot shows the MySQL Workbench interface. In the top-left window, titled 'SQL File 1*', a SQL query is written:

```
1 • SELECT *
 2   FROM city
 3 WHERE Population > 7000000 AND Population < 8000000
```

The bottom-right window, titled 'city 25', displays the results of the query in a 'Result Grid'. The grid has columns: ID, Name, CountryCode, District, and Population. The data is as follows:

	ID	Name	CountryCode	District	Population
▶	456	London	GBR	England	7285000
	1025	Delhi	IND	Delhi	7206704
	1532	Tokyo	JPN	Tokyo-to	7980230
	1891	Peking	CHN	Peking	7472000
*	NULL	NULL	NULL	NULL	NULL

한국에 있는 도시들만 보기

미국에 있는 도시들만 보기

한국에 있는 도시들 중에 인구 수가 1,000,000 이상인 도시
만 SELECT 하기

BETWEEN

- 데이터가 숫자로 구성되어 있어 연속적인 값은 BETWEEN... AND 사용 가능

The screenshot shows the MySQL Workbench interface. At the top, there is a toolbar with various icons. Below the toolbar, a SQL editor window titled "SQL File 1" contains the following query:

```
1 • SELECT *
2   FROM city
3 WHERE Population BETWEEN 7000000 AND 8000000
```

Below the SQL editor is a "Result Grid" window displaying the results of the query. The grid has columns labeled ID, Name, CountryCode, District, and Population. The data is as follows:

	ID	Name	CountryCode	District	Population
▶	456	London	GBR	England	7285000
	1025	Delhi	IND	Delhi	7206704
	1532	Tokyo	JPN	Tokyo-to	7980230
	1891	Peking	CHN	Peking	7472000
*	NULL	NULL	NULL	NULL	NULL

On the right side of the interface, there is a vertical sidebar with four tabs: "Result Grid" (which is selected), "Form Editor", "Field Types", and "Query Stats". At the bottom right of the main area, there is a "Apply" button.

- 이산적인 Discrete 값의 조건에서는 IN() 사용 가능

The screenshot shows the MySQL Workbench interface. In the top window, titled "SQL File 1*", a query is written:

```
1 • SELECT *
 2   FROM city
 3 WHERE Name IN('Seoul', 'New York', 'Tokyo')
```

The bottom window, titled "city 33", displays the results in a grid:

	ID	Name	CountryCode	District	Population
▶	1532	Tokyo	JPN	Tokyo-to	7980230
	2331	Seoul	KOR	Seoul	9981619
	3793	New York	USA	New York	8008278
*	NULL	NULL	NULL	NULL	NULL

한국, 미국, 일본의 도시들에 대해서 보기

LIKE

- 문자열의 내용 검색하기 위해 LIKE 연산자 사용
- 문자 뒤에 % - 무엇이든(%) 허용
- 한 글자와 매치하기 위해서는 '_' 사용

SQL File 1* x

```
1 • SELECT *
 2 FROM city
 3 WHERE CountryCode LIKE 'KO_'
```

Result Grid | Filter Rows: [] | Edit: [] | Export/Import: [] | Result Grid | Form Editor | Field Types | Query Stats | Apply

ID	Name	CountryCode	District	Population
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
2337	Ulsan	KOR	Kyongsangnam	1084891
2338	Songnam	KOR	Kyonggi	869094
2339	Puchon	KOR	Kyonggi	779412
2340	Suwon	KOR	Kyonggi	755550
2341	Anyang	KOR	Kyonggi	591106
2342	Chonju	KOR	Chollabuk	563153
2343	Chongju	KOR	Chungchongbuk	531376
2344	Koyang	KOR	Kyonqgi	518282

Sub Query

- 서브 쿼리 SubQuery
- 쿼리문 안에 또 쿼리문이 들어 있는 것
- 서브 쿼리의 결과가 둘 이상이 되면 에러 발생

SQL File 1* x

```
1 •   SELECT *
2     FROM city
3 WHERE CountryCode = (
4             SELECT CountryCode
5               FROM city
6             WHERE Name = 'Seoul' );
```

Result Grid | Filter Rows: [] | Edit: [] | Export/Import: [] | Result Grid | Form Editor | Field Types | Query Stats | Apply

ID	Name	CountryCode	District	Population
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
2337	Ulsan	KOR	Kyongsangnam	1084891
2338	Songnam	KOR	Kyonggi	869094
2339	Puchon	KOR	Kyonggi	779412
2340	Suwon	KOR	Kyonggi	755550
2341	Anyang	KOR	Kyonggi	591106
2342	Chonju	KOR	Chollabuk	563153
2343	Chongju	KOR	Chungchongbuk	531376
2344	Koyang	KOR	Kyonqgi	518282

ANY

- 서브쿼리의 여러 개의 결과 중 한 가지만 만족해도 가능
- SOME은 ANY와 동일한 의미로 사용
- = ANY 구문은 IN과 동일한 의미

SQL File 1* x

```
1  SELECT *
2  FROM city
3  WHERE Population > ANY (
4      SELECT Population
5      FROM city
6      WHERE District = 'New York' );
```

Result Grid | Filter Rows: [] | Edit: [] | Export/Import: [] | Result Grid | Form Editor | Field Types | Query Stats

ID	Name	CountryCode	District	Population
1	Kabul	AFG	Kabul	1780000
2	Qandahar	AFG	Qandahar	237500
3	Herat	AFG	Herat	186800
4	Mazar-e-Sharif	AFG	Balkh	127800
5	Amsterdam	NLD	Noord-Holland	731200
6	Rotterdam	NLD	Zuid-Holland	593321
7	Haag	NLD	Zuid-Holland	440900
8	Utrecht	NLD	Utrecht	234323
9	Eindhoven	NLD	Noord-Brabant	201843
10	Tilburg	NLD	Noord-Brabant	193238
11	Groningen	NLD	Groningen	172701
12	Breda	NLD	Noord-Brabant	160398
13	Apeldoorn	NLD	Gelderland	153491
14	Nijmeegen	NLD	Gelderland	152463

city 96 x Apply Revert

ALL

- 서브쿼리의 여러 개의 결과를 모두 만족 시켜야 함

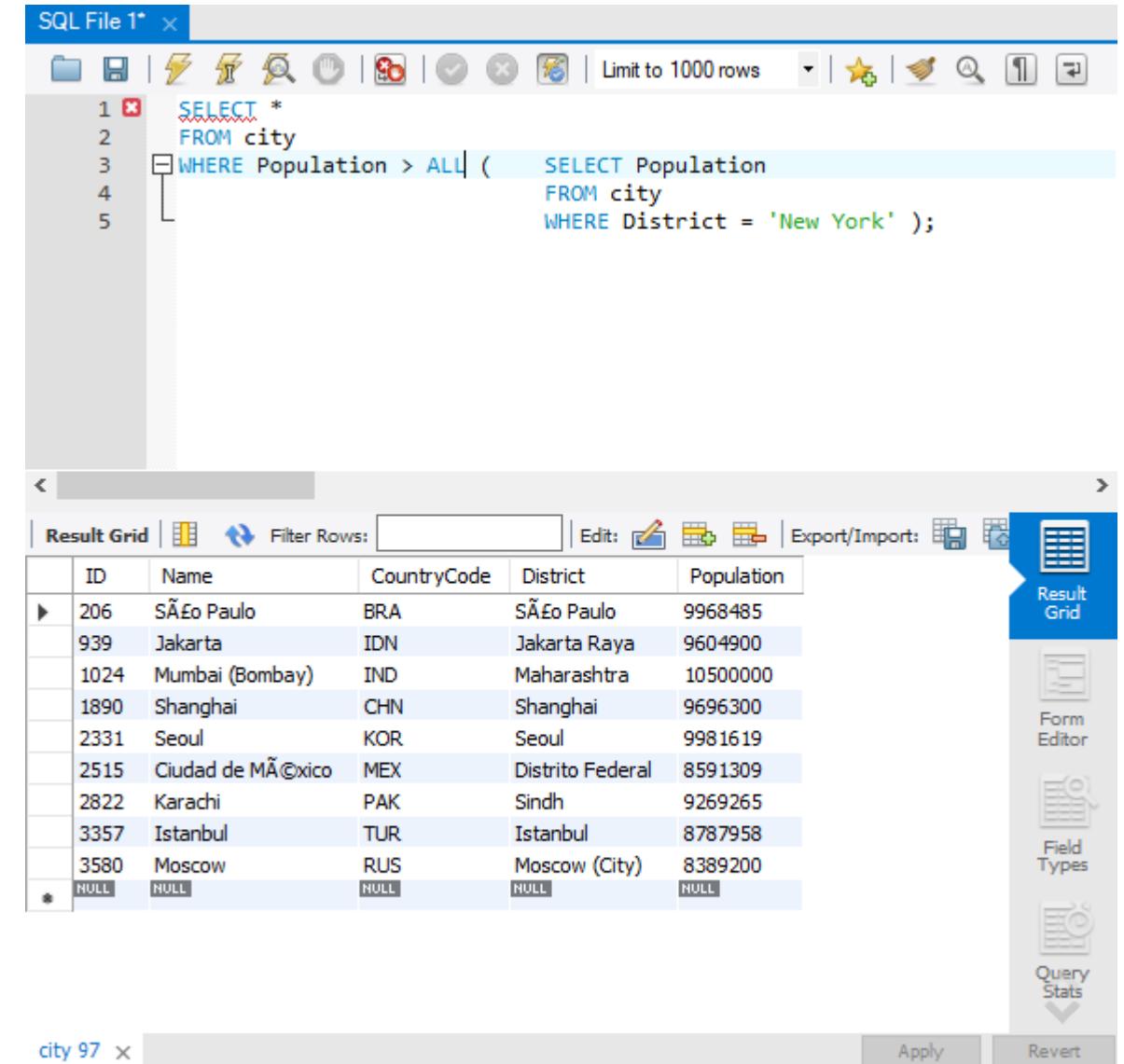
SQL File 1* x

```
1  SELECT *
2  FROM city
3  WHERE Population > ALL (   SELECT Population
4                                FROM city
5                                WHERE District = 'New York' );
```

Result Grid | Filter Rows: [] | Edit: [] | Export/Import: [] | Result Grid | Form Editor | Field Types | Query Stats

ID	Name	CountryCode	District	Population
206	São Paulo	BRA	São Paulo	9968485
939	Jakarta	IDN	Jakarta Raya	9604900
1024	Mumbai (Bombay)	IND	Maharashtra	10500000
1890	Shanghai	CHN	Shanghai	9696300
2331	Seoul	KOR	Seoul	9981619
2515	Ciudad de México	MEX	Distrito Federal	8591309
2822	Karachi	PAK	Sindh	9269265
3357	Istanbul	TUR	Istanbul	8787958
3580	Moscow	RUS	Moscow (City)	8389200
*	NULL	NULL	NULL	NULL

city 97 x Apply Revert



ORDER BY

- 결과물에 대해 영향을 미치지는 않음
- 결과가 출력되는 순서를 조절하는 구문
- 기본적으로 오름차순^{ASCENDING} 정렬
- 내림차순^{DESCENDING} 으로 정렬
 - 열 이름 뒤에 DESC 적어줄 것
- ASC(오름차순)는 디폴트 값이므로 생략 가능

SQL File 1* x

```
1 • SELECT *
  FROM city
  ORDER BY Population DESC
```

Result Grid | Filter Rows: [] Edit: [] Export/Import: []

ID	Name	CountryCode	District	Population
1024	Mumbai (Bombay)	IND	Maharashtra	10500000
2331	Seoul	KOR	Seoul	9981619
206	São Paulo	BRA	São Paulo	9968485
1890	Shanghai	CHN	Shanghai	9696300
939	Jakarta	IDN	Jakarta Raya	9604900
2822	Karachi	PAK	Sindh	9269265
3357	Istanbul	TUR	Istanbul	8787958
2515	Ciudad de MÃ©xico	MEX	Distrito Federal	8591309
3580	Moscow	RUS	Moscow (City)	8389200
3793	New York	USA	New York	8008278
1532	Tokyo	JPN	Tokyo-to	7980230
1891	Peking	CHN	Peking	7472000
456	London	GBR	England	7285000
1025	Delhi	IND	Delhi	7206704

city 101 x Apply Revert

The screenshot shows the MySQL Workbench interface. In the top window, there is a SQL editor tab titled "SQL File 1* x" containing the following code:

```
1 • SELECT *
  FROM city
  ORDER BY Population DESC
```

Below the editor is a "Result Grid" tab showing the results of the query. The grid has columns: ID, Name, CountryCode, District, and Population. The data lists major cities from around the world, ordered by population in descending order. The first few rows are Mumbai (Bombay), Seoul, São Paulo, and Shanghai.

ORDER BY

- ORDER BY 구문을 혼합해 사용하는 구문도 가능

The screenshot shows the MySQL Workbench interface. At the top, there's a toolbar with various icons. Below it is a SQL editor window titled "SQL File 1" containing the following code:

```
1 • SELECT *
 2 FROM city
 3 ORDER BY CountryCode ASC, Population DESC
```

Below the SQL editor is a "Result Grid" window displaying the query results. The results are a table with columns: ID, Name, CountryCode, District, and Population. The data is ordered by CountryCode (ASC) and Population (DESC). The results show the top cities by population within each country.

ID	Name	CountryCode	District	Population
129	Oranjestad	ABW	Â-	29034
1	Kabul	AFG	Kabol	1780000
2	Qandahar	AFG	Qandahar	237500
3	Herat	AFG	Herat	186800
4	Mazar-e-Sharif	AFG	Balkh	127800
56	Luanda	AGO	Luanda	2022000
57	Huambo	AGO	Huambo	163100
58	Lobito	AGO	Benguela	130000
59	Benguela	AGO	Benguela	128300
60	Namibe	AGO	Namibe	118200
61	South Hill	AIA	Â-	961
62	The Valley	AIA	Â-	595
34	Tirana	ALB	Tirana	270000
55	Andorra la Vella	AND	Andorra l...	21189

한국에 있는 도시들에 대해 인구수로 내림차순하여 보기

country 테이블을 이용하여 국가 면적 크기로 내림차순하여 보기

DISTINCT

- 중복된 것은 1개씩만 보여주면서 출력
- 테이블의 크기가 클수록 효율적

SQL File 1* x

```
1 • 2
SELECT DISTINCT CountryCode
FROM city
```

Result Grid | Filter Rows: [] Export: [] Wrap Cell Content: []

CountryCode
ABW
AFG
AGO
AIA
ALB
AND
ANT
ARE
ARG
ARM
ASM
ATG
AUS
AUT

city 105 x Read Only

Result Grid
Form Editor
Field Types
Query Stats

LIMIT

- 출력 개수를 제한
- 일부를 보기 위해 여러 건의 데이터를 출력하는 부담 줄임
- 상위의 N개만 출력하는 'LIMIT N' 구문 사용
- 서버의 처리량을 많이 사용해 서버의 전반적인 성능을 나쁘게 하는 악성 쿼리문 개선할 때 사용

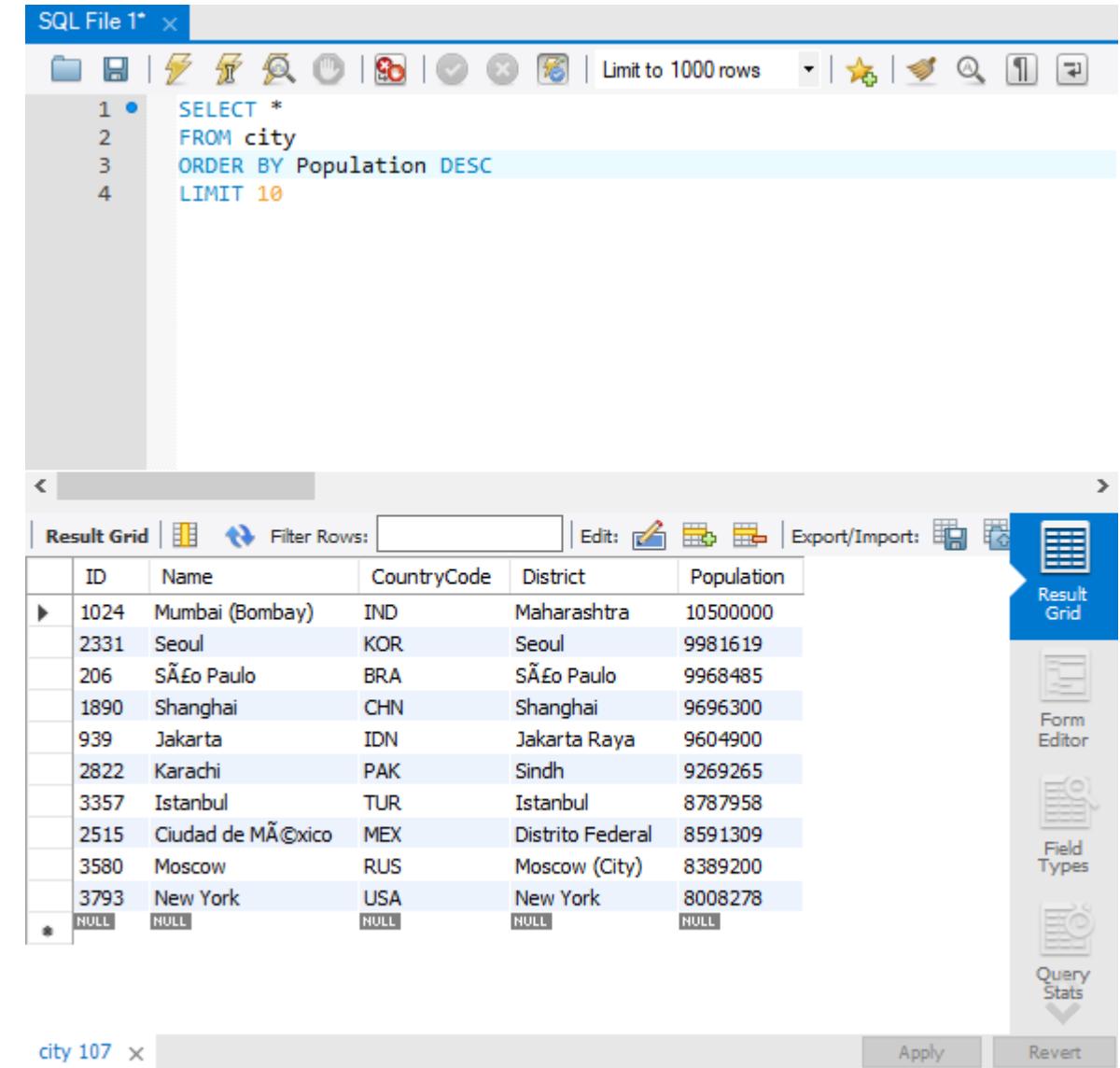
SQL File 1* x

```
1 • SELECT *
  FROM city
  ORDER BY Population DESC
  LIMIT 10
```

Result Grid | Filter Rows: Edit: Export/Import: Result Grid | Form Editor | Field Types | Query Stats

ID	Name	CountryCode	District	Population
1024	Mumbai (Bombay)	IND	Maharashtra	10500000
2331	Seoul	KOR	Seoul	9981619
206	São Paulo	BRA	São Paulo	9968485
1890	Shanghai	CHN	Shanghai	9696300
939	Jakarta	IDN	Jakarta Raya	9604900
2822	Karachi	PAK	Sindh	9269265
3357	Istanbul	TUR	Istanbul	8787958
2515	Ciudad de México	MEX	Distrito Federal	8591309
3580	Moscow	RUS	Moscow (City)	8389200
3793	New York	USA	New York	8008278
*	NULL	NULL	NULL	NULL

city 107 x Apply Revert



GROUP BY

- 그룹으로 묶어주는 역할
- 집계 함수 Aggregate Function 를 함께 사용
 - AVG(): 평균
 - MIN(): 최소값
 - MAX(): 최대값
 - COUNT(): 행의 개수
 - COUNT(DISTINCT): 중복 제외된 행의 개수
 - STDEV(): 표준 편차
 - VARIANCE(): 분산
- 효율적인 데이터 그룹화 Grouping
- 읽기 좋게 하기 위해 별칭 Alias 사용

```
1  SELECT CountryCode, MAX(Population) AS 'Population'  
2  FROM city  
3  GROUP BY CountryCode
```

The screenshot shows the MySQL Workbench interface. At the top, there's a toolbar with various icons. Below it is a SQL editor window titled "SQL File 1" containing the following query:

```
1  SELECT CountryCode, MAX(Population)  
2  FROM city  
3  GROUP BY CountryCode
```

Below the SQL editor is a "Result Grid" window showing the results of the query. The results are as follows:

	CountryCode	MAX(Population)
▶	ABW	29034
	AFG	1780000
	AGO	2022000
	AIA	961
	ALB	270000
	AND	21189
	ANT	2345
	ARE	669181
	ARG	2982146
	ARM	1248700
	ASM	5200
	ATG	24000
	AUS	3276207
	AUT	1608144

The bottom right corner of the result grid has a "Read Only" button.

전체 도시는 몇개인가?

전체 도시들의 평균 인구수는?

HAVING

- WHERE과 비슷한 개념으로 조건 제한
- 집계 함수에 대해서 조건 제한하는 편리한 개념
- HAVING절은 반드시 GROUP BY절 다음에 나와야 함

The screenshot shows the MySQL Workbench interface. In the top SQL Editor window, the following query is displayed:

```
SQL File 1* x
SELECT CountryCode, MAX(Population) 'MAX Population'
FROM city
GROUP BY CountryCode
HAVING MAX(Population) > 8000000
```

The Result Grid below shows the output of the query:

	CountryCode	MAX Population
▶	BRA	9968485
	CHN	9696300
	IDN	9604900
	IND	10500000
	KOR	9981619
	MEX	8591309
	PAK	9269265
	RUS	8389200
	TUR	8787958
	USA	8008278

The bottom Result 117 window shows the count of rows returned by the query.

ROLLUP

- 총합 또는 중간합계가 필요할 경우 사용'
- GROUP BY절과 함께 WITH ROLLUP문 사용

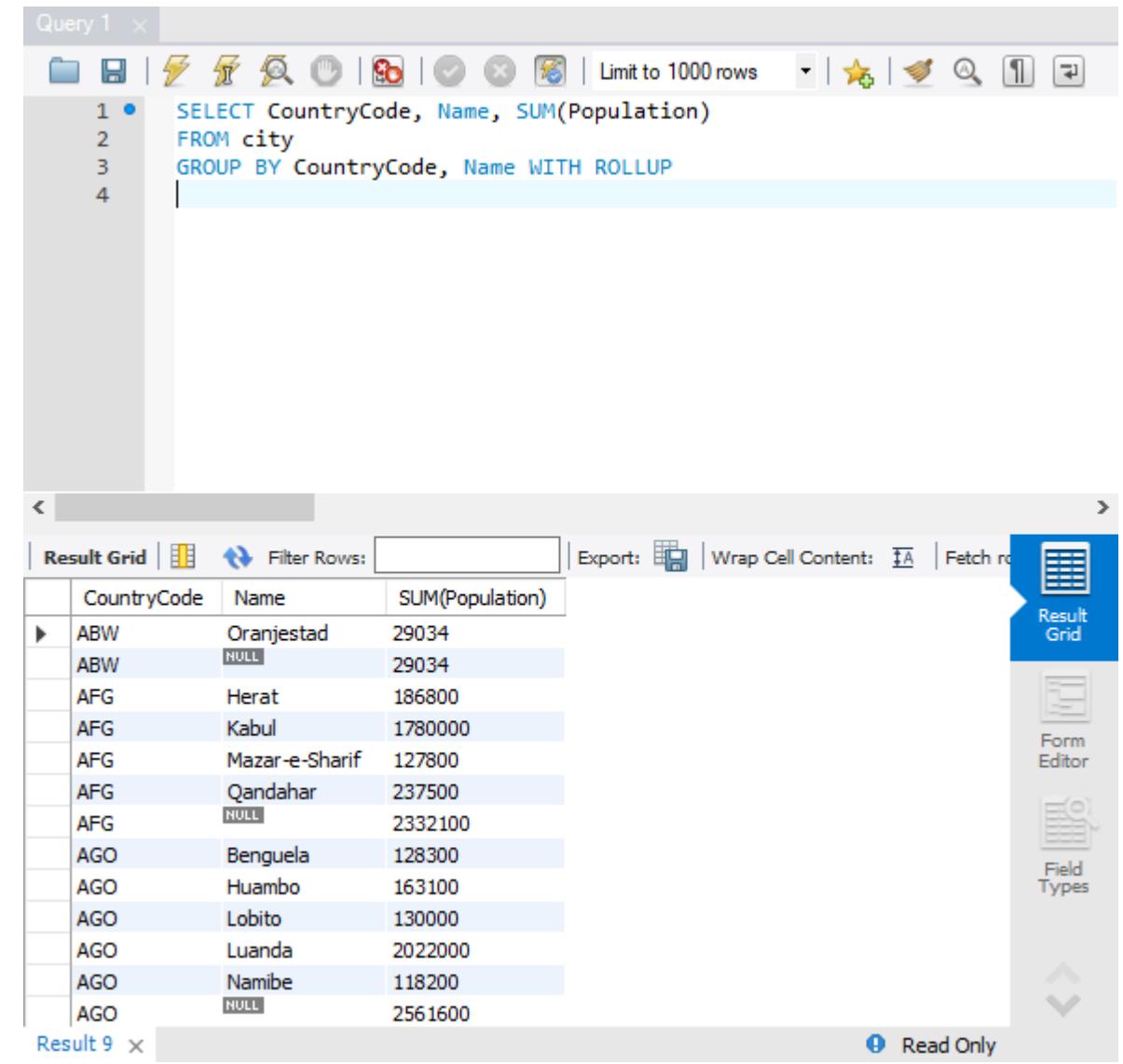
Query 1

```
1 • SELECT CountryCode, Name, SUM(Population)
  2   FROM city
  3   GROUP BY CountryCode, Name WITH ROLLUP
```

Result Grid | Filter Rows: [] | Export: [] | Wrap Cell Content: [] | Fetch rows: []

	CountryCode	Name	SUM(Population)
▶	ABW	Oranjestad	29034
	ABW	NULL	29034
▶	AFG	Herat	186800
	AFG	Kabul	1780000
▶	AFG	Mazar-e-Sharif	127800
	AFG	Qandahar	237500
▶	AFG	NULL	2332100
	AGO	Benguela	128300
▶	AGO	Huambo	163100
	AGO	Lobito	130000
▶	AGO	Luanda	2022000
	AGO	Namibe	118200
▶	AGO	NULL	2561600

Result 9 | Read Only



JOIN

- JOIN은 데이터베이스 내의 여러 테이블에서 가져온 레코드를 조합하여 하나의 테이블이나 결과 집합으로 표현

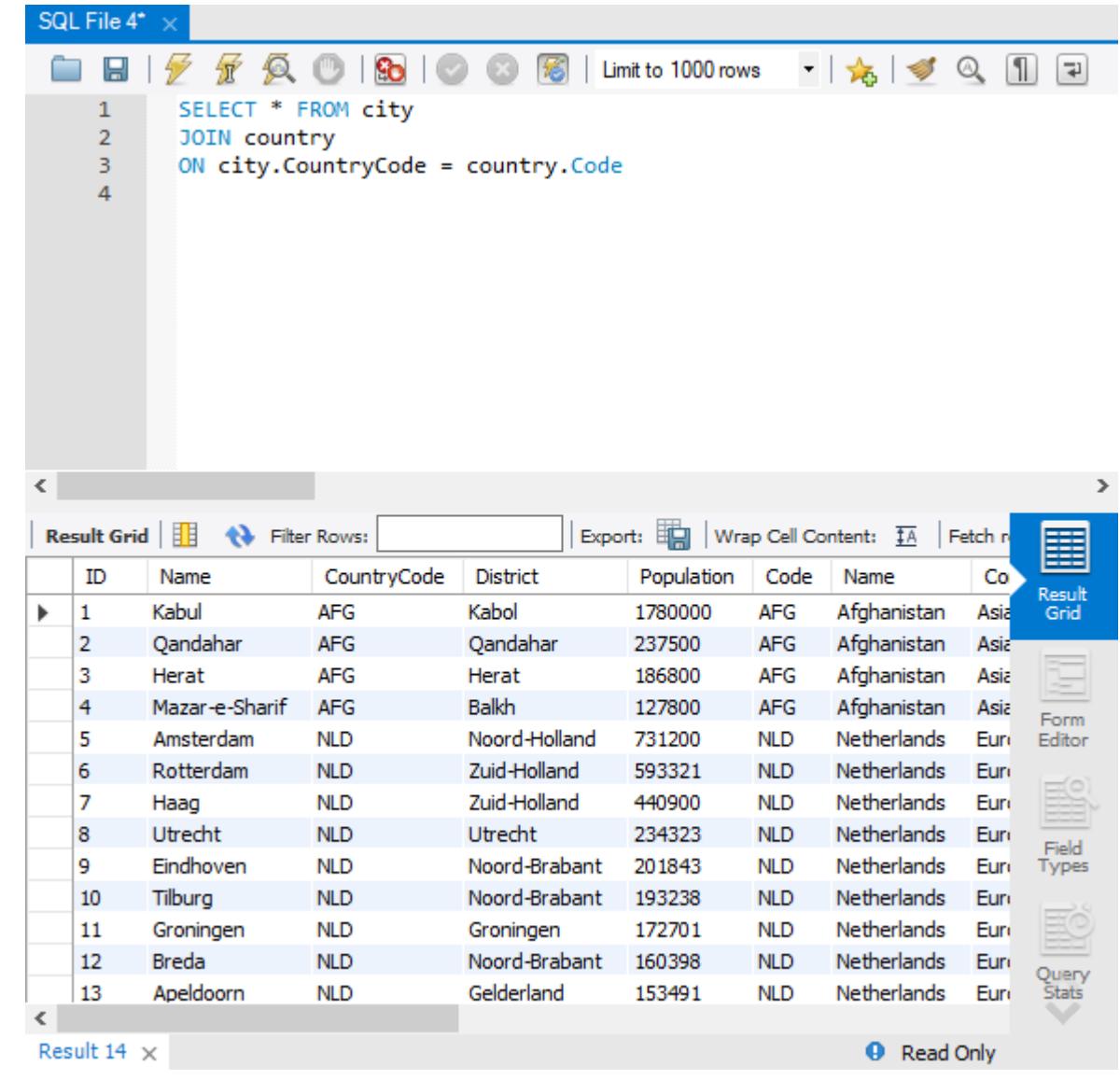
SQL File 4* x

```
1 SELECT * FROM city
2 JOIN country
3 ON city.CountryCode = country.Code
4
```

Result Grid | Filter Rows: Export: Wrap Cell Content: Fetch n Result Grid

ID	Name	CountryCode	District	Population	Code	Name	Co
1	Kabul	AFG	Kabul	1780000	AFG	Afghanistan	Asia
2	Qandahar	AFG	Qandahar	237500	AFG	Afghanistan	Asia
3	Herat	AFG	Herat	186800	AFG	Afghanistan	Asia
4	Mazar-e-Sharif	AFG	Balkh	127800	AFG	Afghanistan	Asia
5	Amsterdam	NLD	Noord-Holland	731200	NLD	Netherlands	Europe
6	Rotterdam	NLD	Zuid-Holland	593321	NLD	Netherlands	Europe
7	Haag	NLD	Zuid-Holland	440900	NLD	Netherlands	Europe
8	Utrecht	NLD	Utrecht	234323	NLD	Netherlands	Europe
9	Eindhoven	NLD	Noord-Brabant	201843	NLD	Netherlands	Europe
10	Tilburg	NLD	Noord-Brabant	193238	NLD	Netherlands	Europe
11	Groningen	NLD	Groningen	172701	NLD	Netherlands	Europe
12	Breda	NLD	Noord-Brabant	160398	NLD	Netherlands	Europe
13	Apeldoorn	NLD	Gelderland	153491	NLD	Netherlands	Europe

Result 14 x Read Only



city, country, countrylanguage 테이블 3개를 JOIN 하기

MySQL 내장함수

- 사용자의 편의를 위해 다양한 기능의 내장 함수를 미리 정의하여 제공
- 대표적인 내장 함수의 종류
 - 문자열 함수
 - 수학 함수
 - 날짜와 시간 함수

LENGTH()

- 전달받은 문자열의 길이를 반환

The screenshot shows the MySQL Workbench interface. In the top query editor window titled "Query 1", the SQL command `SELECT LENGTH('123456789');` is entered. The result grid below displays a single row with two columns: the first column contains the function call `LENGTH('123456789')`, and the second column contains the result value 9. The interface includes various toolbars and a sidebar with tabs for "Result Grid", "Form Editor", and "Field Types".

	LENGTH('123456789')
▶	9

CONCAT()

- 전달받은 문자열을 모두 결합하여 하나의 문자열로 반환
- 전달받은 문자열 중 하나라도 NULL이 존재하면 NULL을 반환

The screenshot shows the MySQL Workbench interface. In the Query Editor (Query 1), two SQL statements are run:

```
SELECT CONCAT('My', 'sql Op', 'en Source'),  
CONCAT('MySQL', NULL, 'OpenSource');
```

The Result Grid displays the output:

	CONCAT('My', 'sql Op', 'en Source')	CONCAT('MySQL', NULL, 'OpenSource')
▶	Mysql Open Source	NULL

The 'Result Grid' tab is selected at the bottom of the results pane.

LOCATE()

- 문자열 내에서 찾는 문자열이 처음으로 나타나는 위치를 찾아서 해당 위치를 반환
- 찾는 문자열이 문자열 내에 존재하지 않으면 0을 반환
- MySQL에서는 문자열의 시작 인덱스를 1부터 계산

Query 1

```
SELECT LOCATE('abc', 'abababcbacdABCDabcAB'),  
LOCATE('abc', 'abababcbacdABCDabcAB', 7);
```

Result Grid

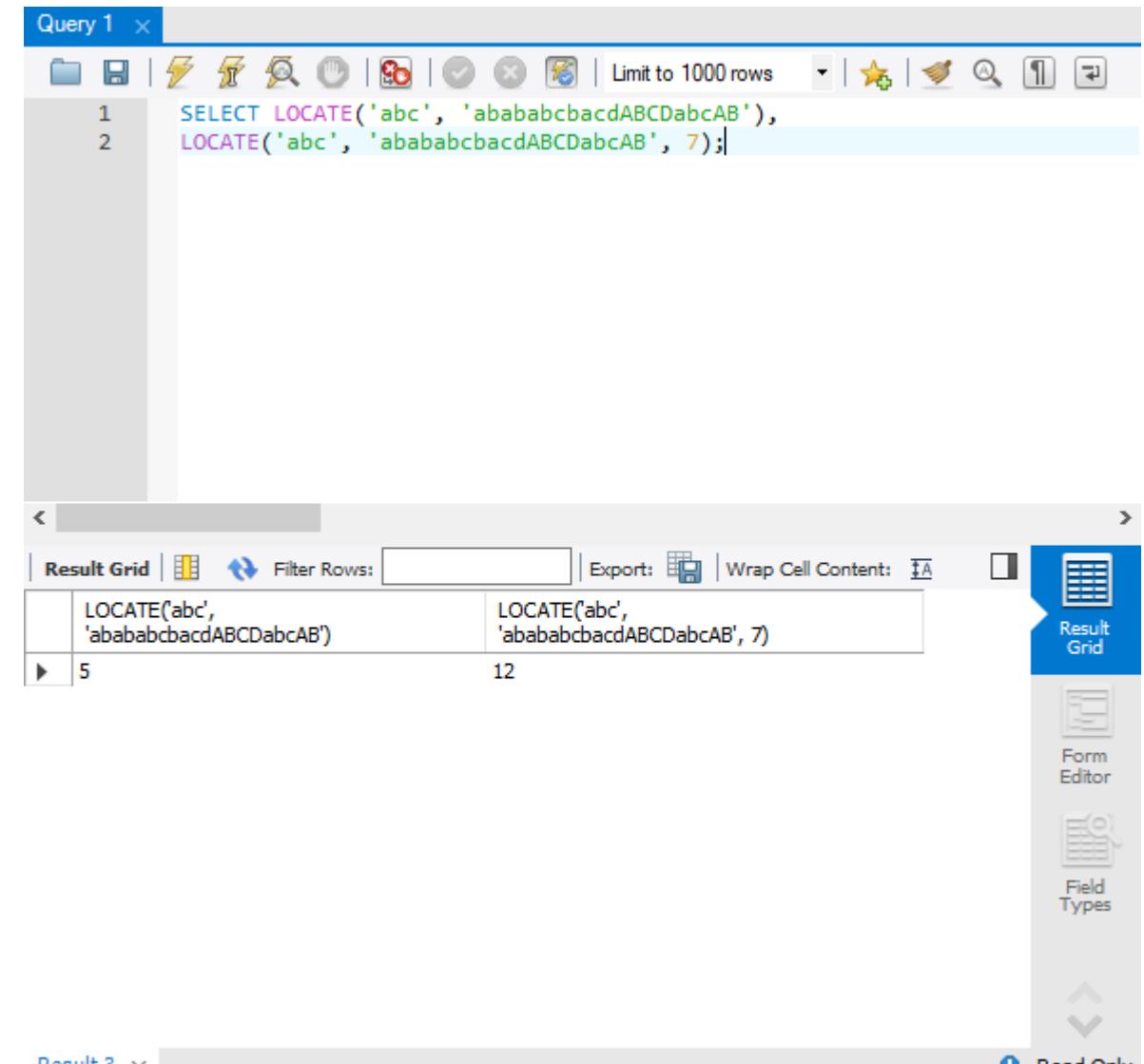
	LOCATE('abc', 'abababcbacdABCDabcAB')	LOCATE('abc', 'abababcbacdABCDabcAB', 7)
▶	5	12

Form Editor

Field Types

Result 3

Read Only



LEFT(), RIGHT()

- LEFT(): 문자열의 왼쪽부터 지정한 개수만큼의 문자를 반환
- RIGHT(): 문자열의 오른쪽부터 지정한 개수만큼의 문자를 반환

The screenshot shows the MySQL Workbench interface with a query editor and a results grid.

Query Editor:

```
Query 1 ×
SELECT
    LEFT('MySQL is an open source relational database management system', 5),
    RIGHT('MySQL is an open source relational database management system', 6)
```

Results Grid:

	LEFT('MySQL is an open source relational database management system', 5)	RIGHT('MySQL is an open source relational database management system', 6)
▶	MySQL	system

The results grid displays the output of the SQL query, showing 'MySQL' as the result of the LEFT function and 'system' as the result of the RIGHT function.

LOWER(), UPPER()

- LOWER(): 문자열의 문자를 모두 소문자로 변경
- UPPER(): 문자열의 문자를 모두 대문자로 변경

The screenshot shows the MySQL Workbench interface with a query editor and a results grid.

Query Editor:

```
Query 1 ×
SELECT
    LOWER('MySQL is an open source relational database management system'),
    UPPER('MySQL is an open source relational database management system');
```

Result Grid:

	LOWER('MySQL is an open source relational database management system')	UPPER('MySQL is an open source relational database management system')
▶	mysql is an open source relational database ma... MySQL IS AN OPEN SOURCE RELATIONAL DAT...	

The results show that the first row contains the original string and its lowercase and uppercase equivalents respectively. The second row shows the truncated versions of the strings due to the limit of 1000 rows.

REPLACE()

- 문자열에서 특정 문자열을 대체 문자열로 교체

The screenshot shows the MySQL Workbench interface. In the top query editor window, titled 'Query 1', the following SQL code is written:

```
SELECT REPLACE('MSSQL', 'MS', 'My');
```

The result of this query is displayed in the 'Result Grid' tab below, showing a single row:

REPLACE('MSSQL', 'MS', 'My')
MySQL

The right sidebar of the interface includes tabs for 'Result Grid', 'Form Editor', and 'Field Types'. At the bottom of the interface, there is a status bar with the text 'Result 21' and 'Read Only'.

TRIM()

- 문자열의 앞이나 뒤, 또는 양쪽 모두에 있는 특정 문자를 제거
- TRIM() 함수에서 사용할 수 있는 지정자
 - BOTH: 전달받은 문자열의 양 끝에 존재하는 특정 문자를 제거 (기본 설정)
 - LEADING: 전달받은 문자열 앞에 존재하는 특정 문자를 제거
 - TRAILING: 전달받은 문자열 뒤에 존재하는 특정 문자를 제거
- 만약 지정자를 명시하지 않으면, 자동으로 BOTH로 설정
- 또한, 제거할 문자를 명시하지 않으면, 자동으로 공백을 제거

The screenshot shows the MySQL Workbench interface. In the top-left pane, titled 'Query 1', there is a code editor containing the following SQL query:

```
SELECT TRIM('##MySQL##'),  
       TRIM(LEADING '#' FROM '##MySQL##'),  
       TRIM(TRAILING '#' FROM '##MySQL##')
```

The results are displayed in a 'Result Grid' below:

	TRIM('##MySQL##')	TRIM(LEADING '#' FROM '##MySQL##')	TRIM(TRAILING '#' FROM '##MySQL##')
▶	##MySQL##	MySQL##	##MySQL

The bottom status bar indicates 'Result 8'.

FORMAT()

- 숫자 타입의 데이터를 세 자리마다 쉼표(,)를 사용하는 '#,###,###.##' 형식으로 변환
- 반환되는 데이터의 형식은 문자열 타입
- 두 번째 인수는 반올림할 소수 부분의 자릿수

The screenshot shows the MySQL Workbench interface. In the top query editor window (Query 1), the following SQL code is written:

```
SELECT FORMAT(123456789.123456, 3);
```

The result grid below displays the output of the query:

FORMAT(123456789.123456, 3)
123,456,789.123

The right sidebar of the interface includes tabs for "Result Grid", "Form Editor", and "Field Types".

FLOOR(), CEIL(), ROUND()

- FLOOR(): 내림
- CEIL(): 올림
- ROUND(): 반올림

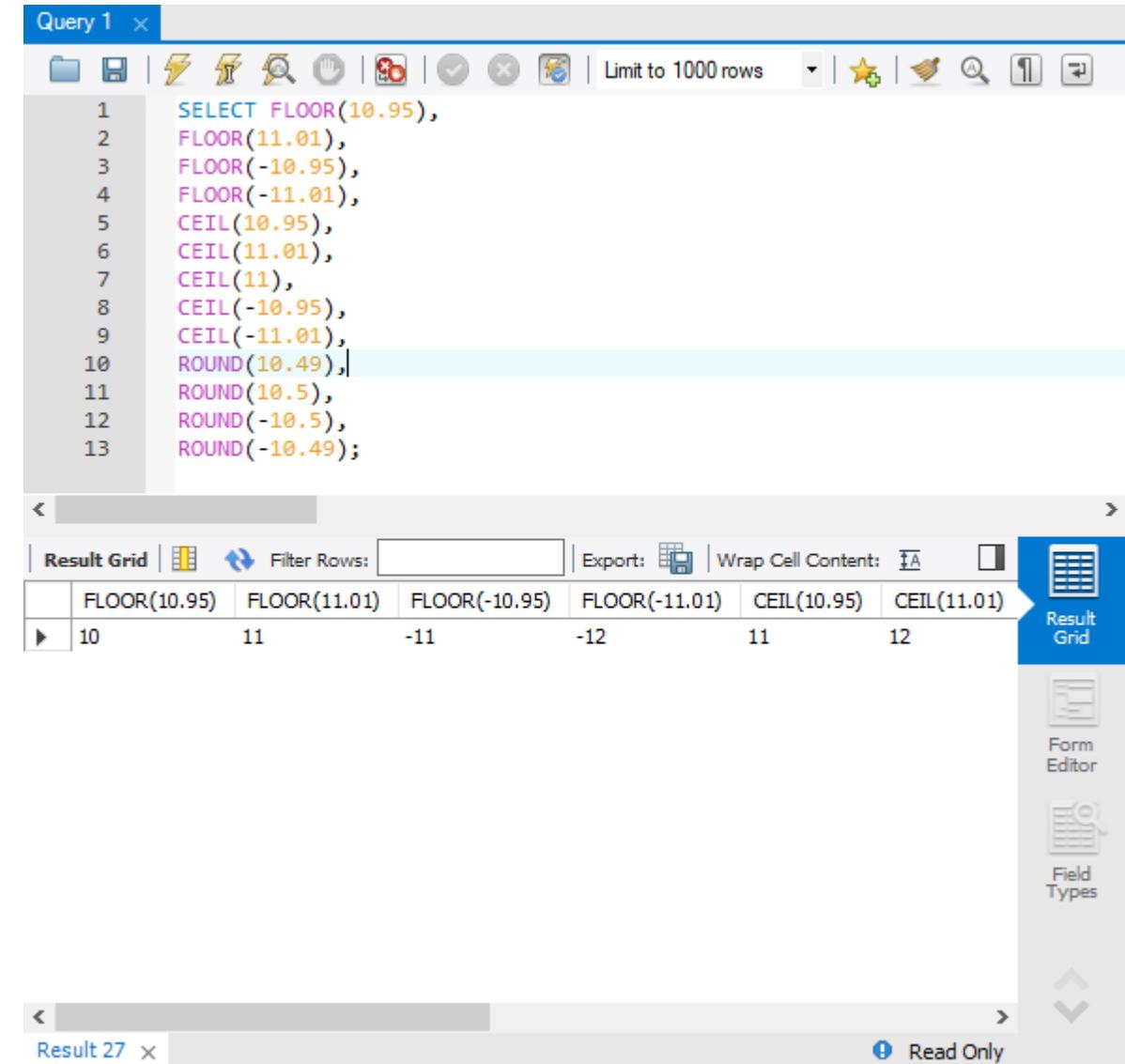
Query 1

```
1 SELECT FLOOR(10.95),
2 FLOOR(11.01),
3 FLOOR(-10.95),
4 FLOOR(-11.01),
5 CEIL(10.95),
6 CEIL(11.01),
7 CEIL(11),
8 CEIL(-10.95),
9 CEIL(-11.01),
10 ROUND(10.49),
11 ROUND(10.5),
12 ROUND(-10.5),
13 ROUND(-10.49);
```

Result Grid | Filter Rows: Export: Wrap Cell Content: Result Grid

	FLOOR(10.95)	FLOOR(11.01)	FLOOR(-10.95)	FLOOR(-11.01)	CEIL(10.95)	CEIL(11.01)
▶	10	11	-11	-12	11	12

Result 27 x Read Only



SQRT(), POW(), EXP(), LOG()

- SQRT(): 양의 제곱근
- POW(): 첫 번째 인수로는 밑수를 전달하고, 두 번째 인수로는 지수를 전달하여 거듭제곱 계산
- EXP(): 인수로 지수를 전달받아, e의 거듭제곱을 계산
- LOG(): 자연로그 값을 계산

The screenshot shows the MySQL Workbench interface with a query editor and a results grid.

Query Editor:

```
Query 1
SELECT SQRT(4),
       POW(2, 3),
       EXP(3),
       LOG(3);
```

Results Grid:

	SQRT(4)	POW(2, 3)	EXP(3)	LOG(3)
▶	2	8	20.085536923187668	1.0986122886681098

The results grid displays the output of the query. The first row contains the column headers: SQRT(4), POW(2, 3), EXP(3), and LOG(3). The second row contains the corresponding values: 2, 8, 20.085536923187668, and 1.0986122886681098.

SIN(), COS(), TAN()

- SIN(): 사인값 반환
- COS(): 코사인값 반환
- TAN(): 탄젠트값 반환

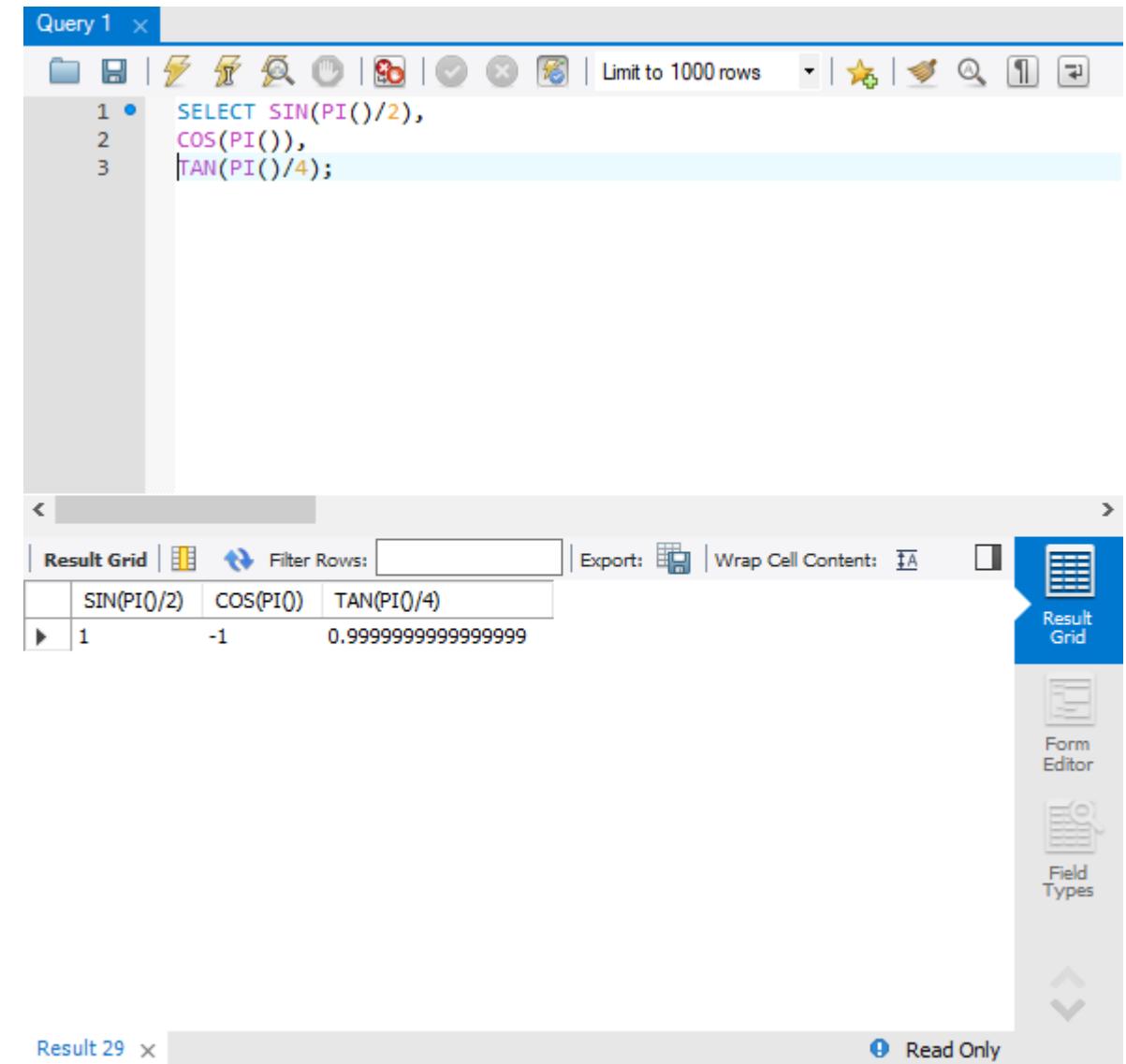
Query 1

```
1 •   SELECT SIN(PI()/2),
2      COS(PI()),
3      TAN(PI()/4);
```

Result Grid | Filter Rows: Export: Wrap Cell Content: Result Grid

	SIN(PI/2)	COS(PI)	TAN(PI/4)
▶	1	-1	0.999999999999999

Result 29 | Read Only



ABS(), RAND()

- ABS(X): 절대값을 반환
- RAND(): 0.0보다 크거나 같고 1.0보다 작은 하나의 실수를 무작위로 생성

Query 1

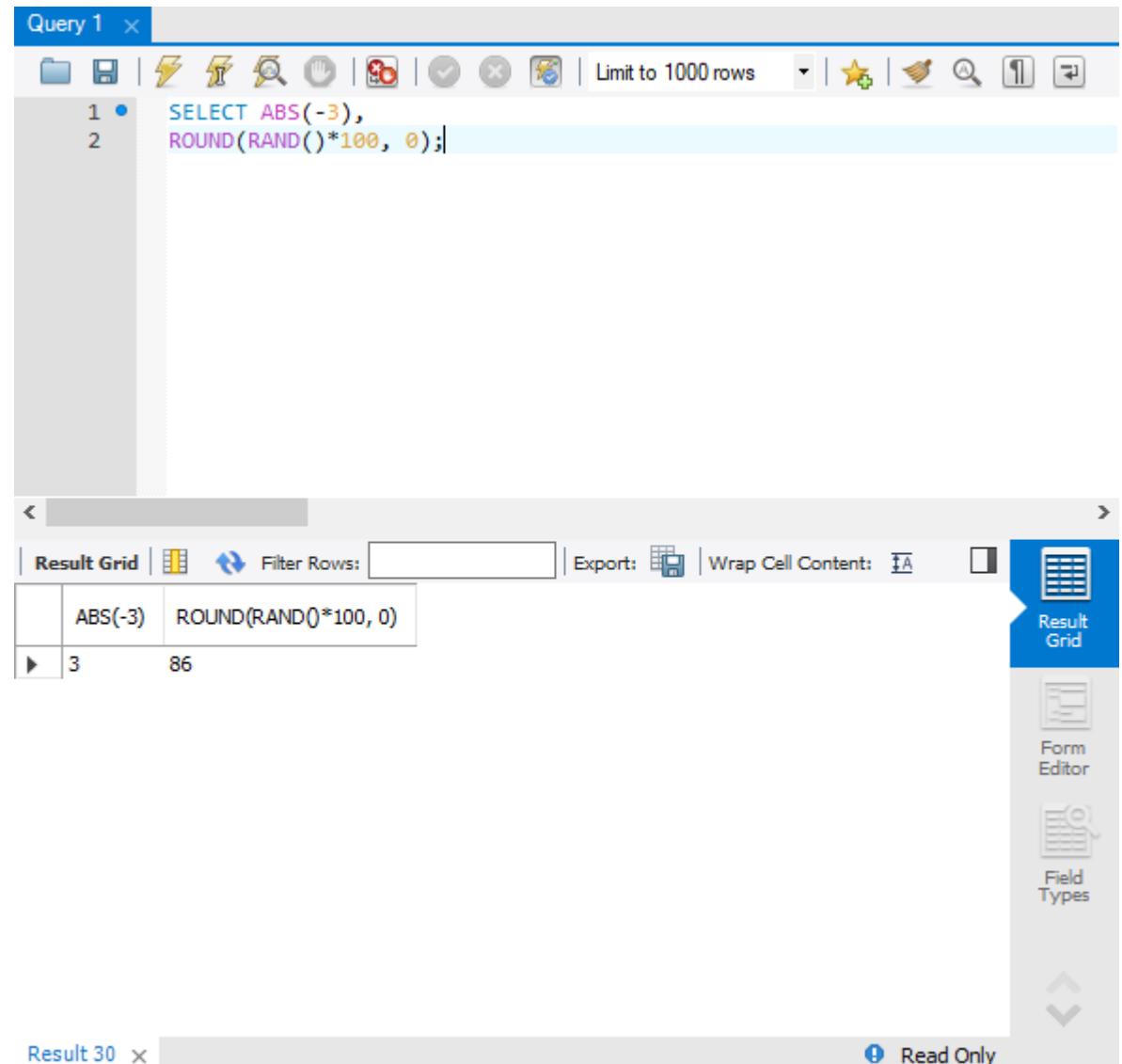
```
1 • SELECT ABS(-3),  
2     ROUND(RAND()*100, 0);
```

Result Grid | Filter Rows: [] | Export: [] | Wrap Cell Content: []

	ABS(-3)	ROUND(RAND()*100, 0)
▶	3	86

Result 30

Read Only

A screenshot of the MySQL Workbench interface. The top bar shows 'Query 1' and various toolbar icons. The main area displays a query window with two rows of code. Row 1 starts with '1 •' followed by 'SELECT ABS(-3),'. Row 2 starts with '2' followed by 'ROUND(RAND()*100, 0);'. Below the query window is a 'Result Grid' section with a table containing one row of data: 'ABS(-3)' is 3 and 'ROUND(RAND()*100, 0)' is 86. The right side of the interface features a vertical sidebar with tabs for 'Result Grid' (which is selected), 'Form Editor', and 'Field Types'. At the bottom, it says 'Result 30' and 'Read Only'.

NOW(), CURDATE(), CURTIME()

- NOW(): 현재 날짜와 시간을 반환, 반환되는 값은 'YYYY-MM-DD HH:MM:SS' 또는 YYYYMMDDHHMMSS 형태로 반환
- CURDATE(): 현재 날짜를 반환, 이때 반환되는 값은 'YYYY-MM-DD' 또는 YYYYMMDD 형태로 반환
- CURTIME(): 현재 시각을 반환, 이때 반환되는 값은 'HH:MM:SS' 또는 HHMMSS 형태로 반환

The screenshot shows the MySQL Workbench interface with a query editor titled 'Query 1'. The query is:

```
SELECT NOW(),
       CURDATE(),
       CURTIME();
```

The results are displayed in a grid:

	NOW()	CURDATE()	CURTIME()
▶	2018-11-21 16:23:08	2018-11-21	16:23:08

The interface includes various toolbars and panels on the right side.

DATE(), MONTH(), DAY(), HOUR(), MINUTE(), SECOND()

- DATE(): 전달받은 값에 해당하는 날짜 정보를 반환
- MONTH(): 월에 해당하는 값을 반환하며, 0부터 12 사이의 값을 가짐
- DAY(): 일에 해당하는 값을 반환하며, 0부터 31 사이의 값을 가짐
- HOUR(): 시간에 해당하는 값을 반환하며, 0부터 23 사이의 값을 가짐
- MINUTE(): 분에 해당하는 값을 반환하며, 0부터 59 사이의 값을 가짐
- SECOND(): 초에 해당하는 값을 반환하며, 0부터 59 사이의 값을 가짐

The screenshot shows the MySQL Workbench interface. In the top window, titled 'Query 1', there is a code editor containing the following SQL query:

```
1 • SELECT DATE('2018-02-19 12:34:56'),
2      MONTH('2018-01-02 12:34:56'),
3      DAY('2018-01-02 12:34:56'),
4      HOUR('12:34:56'),
5      MINUTE('12:34:56'),
6      SECOND('12:34:56');
```

The results are displayed in a grid below:

	DATE('2018-02-19 12:34:56')	MONTH('2018-01-02 12:34:56')	DAY('2018-01-02 12:34:56')	HOUR('12:34:56')	MINUTE('12:34:56')	SECOND('12:34:56')
▶	2018-02-19	1	2	12	34	56

The results show the breakdown of the timestamp '2018-02-19 12:34:56' into its components: Date (2018-02-19), Month (1), Day (2), Hour (12), Minute (34), and Second (56).

MONTHNAME(), DAYNAME()

- MONTHNAME(): 월에 해당하는 이름을 반환
- DAYNAME(): 요일에 해당하는 이름을 반환

The screenshot shows a MySQL Workbench interface with a query editor and a results grid.

Query Editor:

```
Query 1
SELECT MONTHNAME('2018-02-19'),
       DAYNAME('2018-02-19');
```

Result Grid:

	MONTHNAME('2018-02-19')	DAYNAME('2018-02-19')
▶	February	Monday

The results show that for the date '2018-02-19', the month is February and the day is Monday.

DAYOFWEEK(), DAYOFMONTH(), DAYOFYEAR()

- DAYOFWEEK(): 일자가 해당 주에서 몇 번째 날인지를 반환, 1부터 7 사이의 값을 반환 (일요일 = 1, 토요일 = 7)
- DAYOFMONTH(): 일자가 해당 월에서 몇 번째 날인지를 반환, 0부터 31 사이의 값을 반환
- DAYOFYEAR(): 일자가 해당 연도에서 몇 번째 날인지를 반환, 1부터 366 사이의 값을 반환

The screenshot shows the MySQL Workbench interface with a query editor and a results grid.

Query Editor:

```
Query 1
SELECT DAYOFMONTH('2018-02-19'),
       DAYOFWEEK('2018-02-19'),
       DAYOFYEAR('2018-02-19');
```

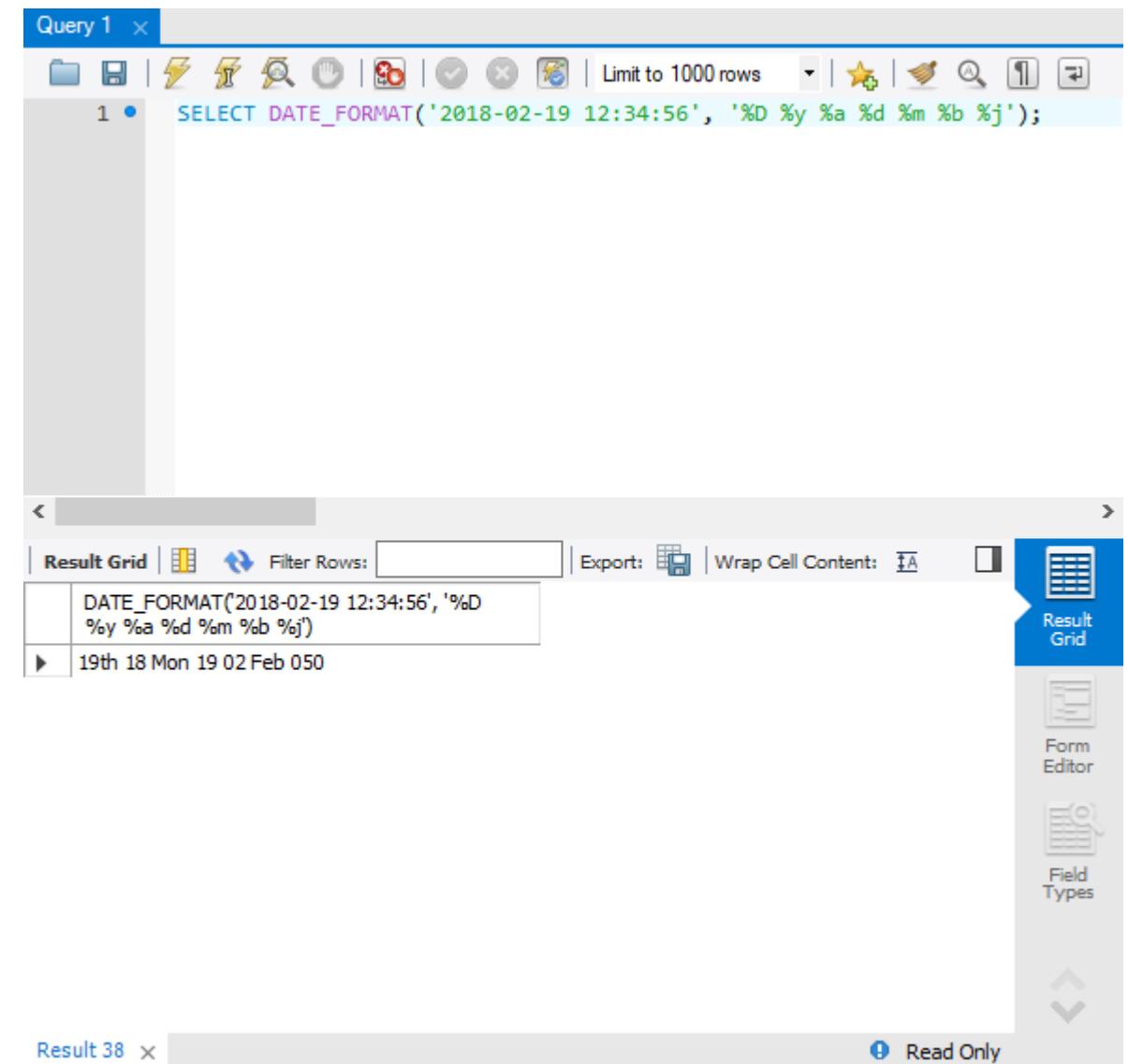
Result Grid:

	DAYOFMONTH('2018-02-19')	DAYOFWEEK('2018-02-19')	DAYOFYEAR('2018-02-19')
▶	19	2	50

The results show that for the date '2018-02-19', the day is the 19th of the month, it is the 2nd day of the week (Monday), and it is the 50th day of the year.

DATE_FORMAT()

- 전달받은 형식에 맞춰 날짜와 시간 정보를 문자열로 반환
- MySQL Date and Time Function:
<https://dev.mysql.com/doc/refman/8.0/en/date-and-time-functions.html>



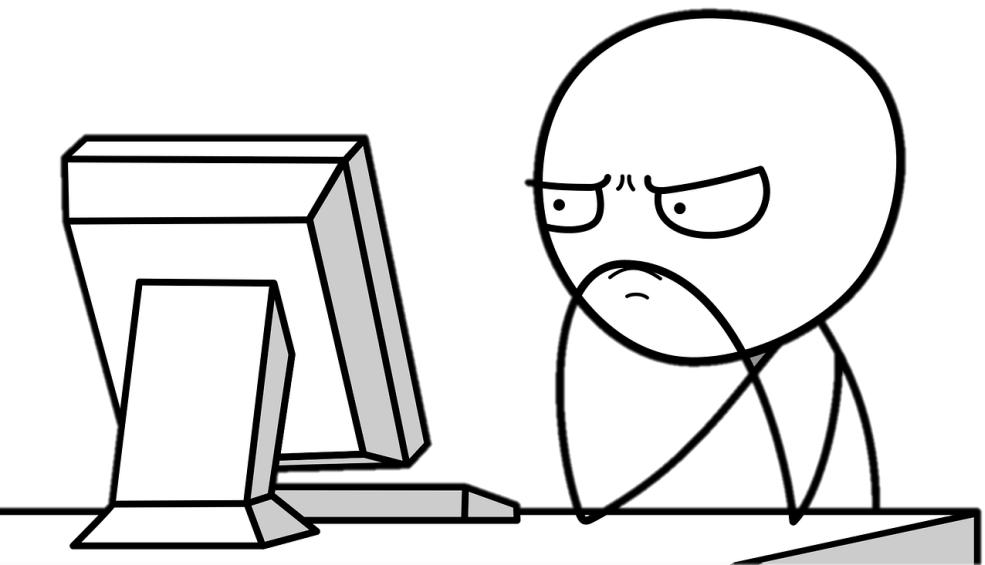
The screenshot shows the MySQL Workbench interface. In the Query Editor (Query 1), the following SQL query is displayed:

```
SELECT DATE_FORMAT('2018-02-19 12:34:56', '%D %y %a %d %m %b %j');
```

The Result Grid displays the output of the query:

	DATE_FORMAT('2018-02-19 12:34:56', '%D %y %a %d %m %b %j')
▶	19th 18 Mon 19 02 Feb 050

4. SQL 고급



CREATE TABLE AS SELECT

- city 테이블과 똑같은 city2 테이블 생성

The screenshot shows the MySQL Workbench interface. In the top panel, there is a SQL editor window titled "SQL File 1" containing two statements:

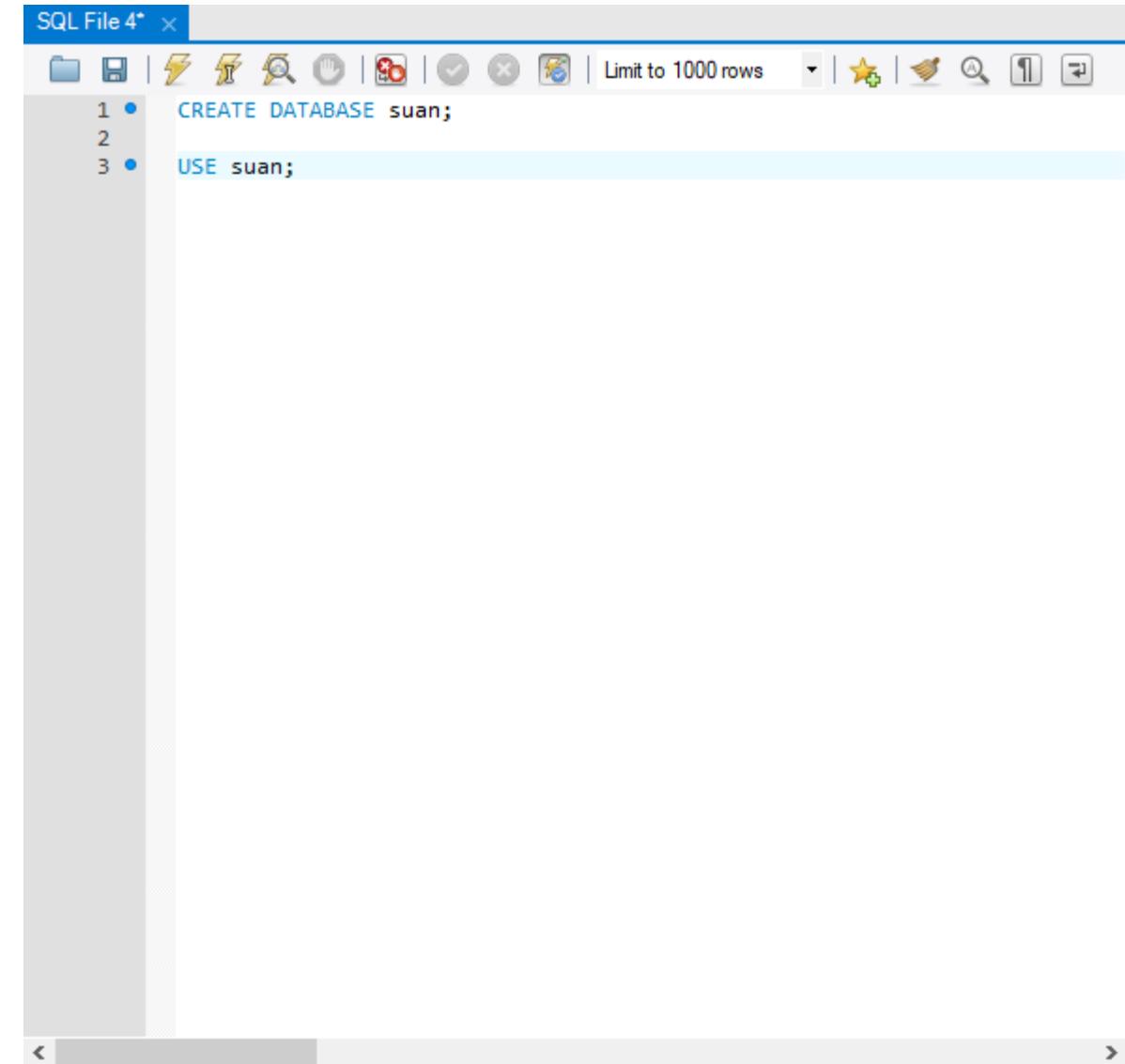
```
CREATE TABLE city2 AS SELECT * FROM city;
SELECT * FROM city2;
```

Below the editor is a "Result Grid" window displaying the data from the "city" table. The grid has columns: ID, Name, CountryCode, District, and Population. The data consists of 14 rows, starting with Kabul (ID 1) and ending with Nijmeagen (ID 14). The "Result Grid" tab is selected at the bottom of the window.

ID	Name	CountryCode	District	Population
1	Kabul	AFG	Kabul	1780000
2	Qandahar	AFG	Qandahar	237500
3	Herat	AFG	Herat	186800
4	Mazar-e-Sharif	AFG	Balkh	127800
5	Amsterdam	NLD	Noord-Holland	731200
6	Rotterdam	NLD	Zuid-Holland	593321
7	Haag	NLD	Zuid-Holland	440900
8	Utrecht	NLD	Utrecht	234323
9	Eindhoven	NLD	Noord-Brabant	201843
10	Tilburg	NLD	Noord-Brabant	193238
11	Groningen	NLD	Groningen	172701
12	Breda	NLD	Noord-Brabant	160398
13	Apeldoorn	NLD	Gelderland	153491
14	Nijmeagen	NLD	Gelderland	152463

CREATE DATABASE

- CREATE DATABASE 문은 새로운 데이터베이스를 생성
- USE문으로 새 데이터베이스를 사용



The screenshot shows a MySQL Workbench interface with a SQL editor titled "SQL File 4". The editor contains two lines of SQL code:

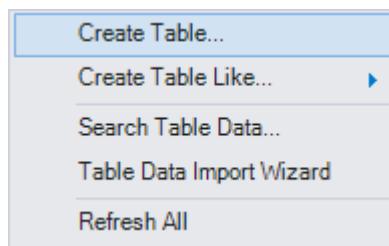
```
CREATE DATABASE suan;  
USE suan;
```

The "suan" database is listed in the left sidebar under the "Databases" section.

CREATE TABLE (MySQL Workbench)

■ 데이터 타입:

<https://dev.mysql.com/doc/refman/8.0/en/data-types.html>



test - Table X

Table Name: test Schema: suan

Charset/Collation: Default C Engine: InnoDB

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Defa
id	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>						
col1	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
col2	FLOAT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
col3	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Column Name: Data Type:

Charset/Collation: Expression:

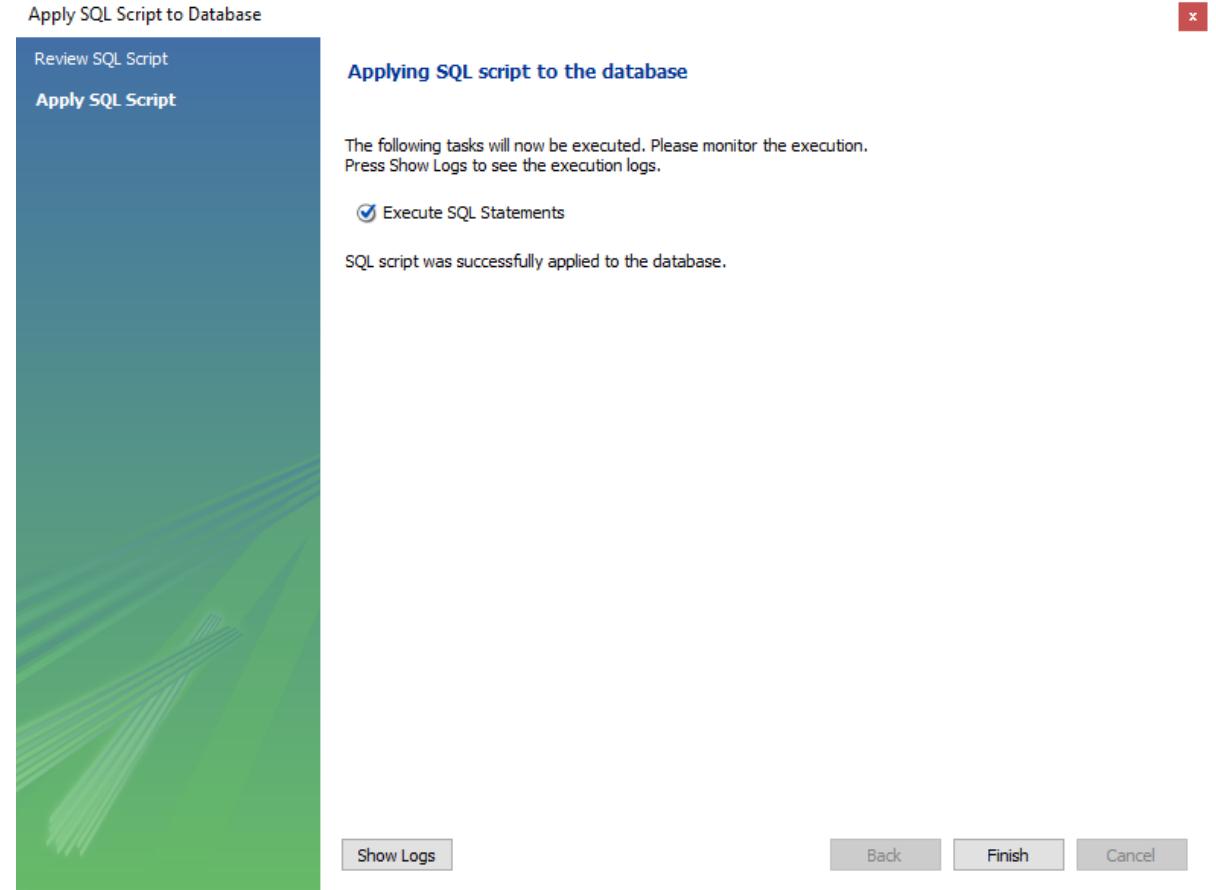
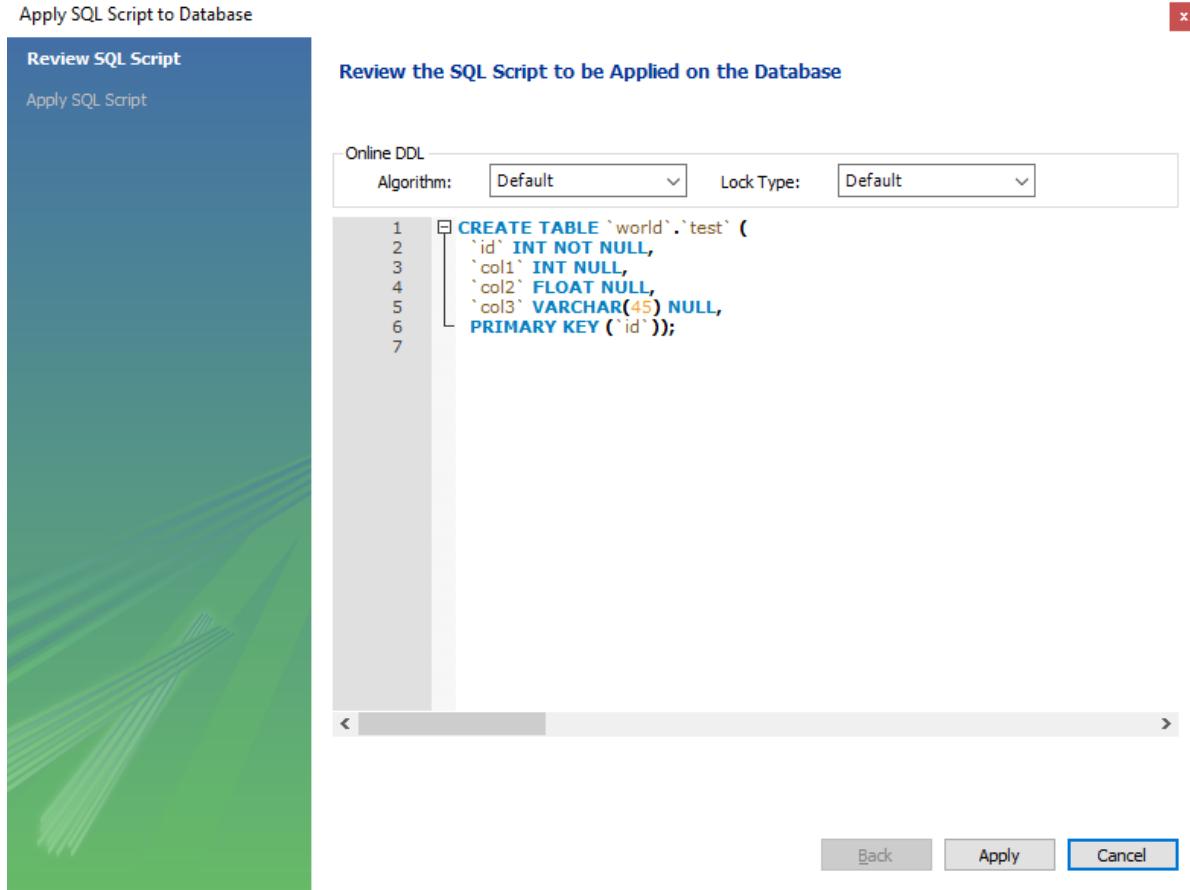
Comments: Storage:

Virtual Stored
 Primary Key Not Null Unique
 Binary Unsigned Zero Fill
 Auto Increment Generated

Columns Indexes Foreign Keys Triggers Partitioning Options

Apply Revert

CREATE TABLE (MySQL Workbench)



CREATE TABLE (MySQL Workbench)

- test 테이블 생성 완료

The screenshot shows the MySQL Workbench interface with a query editor window titled 'test'. The SQL command entered is 'SELECT * FROM test;'. Below the query results, a 'Result Grid' displays the data from the 'test' table, which has four columns: id, col1, col2, and col3. All rows show 'NULL' values. The right side of the interface features a vertical toolbar with icons for different editors: Result Grid (selected), Form Editor, Field Types, and Query Stats.

	id	col1	col2	col3
*	NULL	NULL	NULL	NULL

CREATE TABLE

- test2 테이블 생성 완료

The screenshot shows the MySQL Workbench interface. In the top query editor window titled 'test', there are two statements:

```
1 CREATE TABLE test2 (
2     id      INT NOT NULL PRIMARY KEY,
3     col1    INT NULL,
4     col2    FLOAT NULL,
5     col3    VARCHAR(45) NULL
6 );
7
8 • SELECT * FROM test2;
```

The second statement is highlighted with a blue dot. Below the editor is a 'Result Grid' pane displaying the results of the SELECT query:

	id	col1	col2	col3
*	NULL	NULL	NULL	NULL

On the right side of the interface, there is a vertical toolbar with several icons:

- Result Grid (selected)
- Form Editor
- Field Types
- Query Stats

At the bottom of the interface, there are buttons for 'Apply' and 'Revert'.

ALTER TABLE

- ALTER TABLE 문과 함께 ADD 문을 사용하면, 테이블에 컬럼을 추가할 수 있음

The screenshot shows the MySQL Workbench interface. In the top SQL editor window (SQL File 4), the following code is displayed:

```
1 • ALTER TABLE test2
2   ADD col4 INT NULL;
3
4 • DESC test2;
```

The bottom Result Grid shows the table structure after the modification:

	Field	Type	Null	Key	Default	Extra
▶	id	int(11)	NO	PRI	NULL	
	col1	int(11)	YES		NULL	
	col2	float	YES		NULL	
	col3	varchar(45)	YES		NULL	
	col4	int(11)	YES		NULL	

The sidebar on the right lists other tabs: Result Grid (selected), Form Editor, Field Types, and Query Stats.

ALTER TABLE

- ALTER TABLE 문과 함께 MODIFY 문을 사용하면, 테이블의 컬럼 타입을 변경할 수 있음

The screenshot shows the MySQL Workbench interface. In the top SQL editor window (SQL File 4), the following code is displayed:

```
1 • ALTER TABLE test2
2     MODIFY col4 VARCHAR(20) NULL;
3
4 • DESC test2;
```

The Result Grid below shows the current structure of the test2 table:

	Field	Type	Null	Key	Default	Extra
▶	id	int(11)	NO	PRI	NULL	
	col1	int(11)	YES		NULL	
	col2	float	YES		NULL	
	col3	varchar(45)	YES		NULL	
	col4	varchar(20)	YES		NULL	

The sidebar on the right contains icons for Result Grid, Form Editor, Field Types, and Query Stats.

ALTER TABLE

- ALTER TABLE 문과 함께 DROP 문을 사용하면, 테이블에 컬럼을 제거할 수 있음

The screenshot shows the MySQL Workbench interface. In the top-left pane, titled "SQL File 4*", there are two lines of code:

```
1 • ALTER TABLE test2  
2      DROP col4;  
3  
4 • DESC test2;  
5
```

In the bottom-right pane, titled "Result Grid", there is a table showing the structure of the "test2" table:

	Field	Type	Null	Key	Default	Extra
▶	id	int(11)	NO	PRI	NULL	
	col1	int(11)	YES		NULL	
	col2	float	YES		NULL	
	col3	varchar(45)	YES		NULL	

The sidebar on the right contains icons for "Result Grid", "Form Editor", "Field Types", and "Query Stats".

- 테이블에서 원하는 데이터를 빠르게 찾기 위해 사용
- 일반적으로 데이터를 검색할 때 순서대로 테이블 전체를 검색하므로 데이터가 많으면 많을수록 탐색하는 시간이 늘어남
- 검색과 질의를 할 때 테이블 전체를 읽지 않기 때문에 빠름
- 설정된 컬럼 값을 포함한 데이터의 삽입, 삭제, 수정 작업이 원본 테이블에서 이루어질 경우, 인덱스도 함께 수정되어야 함
- 인덱스가 있는 테이블은 처리 속도가 느려질 수 있으므로 수정보다는 검색이 자주 사용되는 테이블에서 사용하는 것이 좋음

CREATE INDEX

- CREATE INDEX 문을 사용하여 인덱스를 생성

The screenshot shows the MySQL Workbench interface with a SQL editor window titled "SQL File 4". The editor contains the following SQL code:

```
CREATE INDEX Col1Idx  
ON test (col1);
```

The code is highlighted in blue, indicating it is syntax-highlighted SQL. The interface includes standard database management tools like file operations, search, and refresh buttons at the top, and a toolbar below.

SHOW INDEX

■ 인덱스 정보 보기

The screenshot shows the MySQL Workbench interface with a SQL editor window titled "SQL File 4". The query "SHOW INDEX FROM test;" is entered. Below the editor is a "Result Grid" pane displaying the results of the query. The results are as follows:

	Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part
▶	test	0	PRIMARY	1	id	A	0	NULL
	test	1	Col1Idx	1	col1	A	0	NULL

The "Result Grid" tab is selected in the grid toolbar. To the right of the result grid are icons for "Form Editor" and "Field Types". At the bottom right of the result grid is a "Read Only" button.

CREATE UNIQUE INDEX

- 중복 값을 허용하지 않는 인덱스

The screenshot shows the MySQL Workbench interface with a SQL editor and a results grid.

In the SQL editor (top window), the following commands are run:

```
CREATE UNIQUE INDEX Col2Idx  
ON test (col2);  
SHOW INDEX FROM test;
```

The results grid (bottom window) displays the index structure for the 'test' table:

	Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part
▶	test	0	PRIMARY	1	id	A	0	NULL
	test	0	Col2Idx	1	col2	A	0	NULL
	test	1	Col1Idx	1	col1	A	0	NULL

The results grid has tabs for 'Result Grid' (selected), 'Form Editor', and 'Field Types'. A 'Read Only' status is indicated at the bottom right.

FULLTEXT INDEX

- FULLTEXT INDEX는 일반적인 인덱스와는 달리 매우 빠르게 테이블의 모든 텍스트 컬럼을 검색

The screenshot shows the MySQL Workbench interface with two tabs open: 'SQL File 4*' and 'Result Grid'.

In the SQL tab, the following queries are run:

```
ALTER TABLE test
ADD FULLTEXT Col3Idx (col3);

SHOW INDEX FROM test;
```

The Result Grid tab displays the index information for the 'test' table:

	Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part
▶	test	0	PRIMARY	1	id	A	0	NULL
	test	0	Col2Idx	1	col2	A	0	NULL
	test	1	Col1Idx	1	col1	A	0	NULL
	test	1	Col3Idx	1	col3	NULL	0	NULL

At the bottom of the interface, there is a note: 'Read Only'.

INDEX 삭제 (ALTER)

- ALTER 문을 사용하여 테이블에 추가된 인덱스 삭제

The screenshot shows the MySQL Workbench interface. In the top SQL editor window, two statements are run:

```
SQL File 4* x
1 • ALTER TABLE test
2   DROP INDEX Col3Idx;
3
4 • SHOW INDEX FROM test;
```

The bottom Result Grid displays the current state of the test table's indexes:

	Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part
▶	test	0	PRIMARY	1	id	A	0	NULL
	test	0	Col2Idx	1	col2	A	0	NULL
	test	1	Col1Idx	1	col1	A	0	NULL

The Result Grid tab is selected at the bottom.

INDEX 삭제 (DROP INDEX)

- DROP 문을 사용하여 해당 테이블에서 명시된 인덱스를 삭제
- DROP 문은 내부적으로 ALTER 문으로 자동 변환되어 명시된 이름의 인덱스를 삭제

The screenshot shows the MySQL Workbench interface. At the top, there is a toolbar with various icons. Below the toolbar, the SQL editor window contains two statements:

```
SQL File 4* x
1 • DROP INDEX Col2Idx ON test;
2 • SHOW INDEX FROM test;|
```

Below the SQL editor is the Result Grid pane, which displays the results of the SHOW INDEX query. The results are as follows:

	Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part
▶	test	0	PRIMARY	1	id	A	0	NULL
	test	1	Col2Idx	1	col1	A	0	NULL

The Result Grid pane has a "Result 13" tab at the bottom. On the right side of the interface, there is a vertical toolbar with icons for Result Grid, Form Editor, and Field Types, and a "Read Only" status indicator.

- 뷰^{view}는 데이터베이스에 존재하는 일종의 가상 테이블
- 실제 테이블처럼 행과 열을 가지고 있지만, 실제로 데이터를 저장하진 않음
- MySQL에서 뷰는 다른 테이블이나 다른 뷰에 저장되어 있는 데이터를 보여주는 역할만 수행
- 뷰를 사용하면 여러 테이블이나 뷰를 하나의 테이블처럼 볼 수 있음

■ 뷰의 장점

- 특정 사용자에게 테이블 전체가 아닌 필요한 컬럼만 보여줄 수 있음
- 복잡한 쿼리를 단순화해서 사용
- 쿼리 재사용 가능

■ 뷰의 단점

- 한 번 정의된 뷰는 변경할 수 없음
- 삽입, 삭제, 갱신 작업에 많은 제한 사항을 가짐
- 자신만의 인덱스를 가질 수 없음

CREATE VIEW

- CREATE VIEW 문을 사용하여 뷰 생성

The screenshot shows the MySQL Workbench interface. In the top tab, it says "SQL File 4*". Below the tabs are several icons: folder, file, lightning bolt, wrench, magnifying glass, hand, red circle, checkmark, close, and a gear. To the right of these are buttons for "Limit to 1000 rows", a star, a pencil, a magnifying glass, and a refresh symbol. The main SQL editor area contains the following code:

```
CREATE VIEW testView AS
SELECT Col1, Col2
FROM test;
SELECT * FROM testView;
```

Below the editor is a toolbar with "Result Grid", "Filter Rows", "Export", and "Wrap Cell Content" buttons. A small preview grid shows two columns labeled "Col1" and "Col2". On the right side of the interface, there is a vertical panel with three tabs: "Result Grid" (which is selected), "Form Editor", and "Field Types". At the bottom of the screen, there is a status bar with the text "testView 14" and a "Read Only" indicator.

ALTER VIEW

■ ALTER 문을 사용하여 뷰를 수정

The screenshot shows the MySQL Workbench interface. In the top-left pane, titled "SQL File 4*", there are two SQL statements:

```
1 • ALTER VIEW testView AS  
2     SELECT Col1, Col2, Col3  
3     FROM test;  
4  
5 • SELECT * FROM testView;
```

The second statement is highlighted with a light blue background. Below the SQL editor is the "Result Grid" tab, which displays a single row of results:

	Col1	Col2	Col3

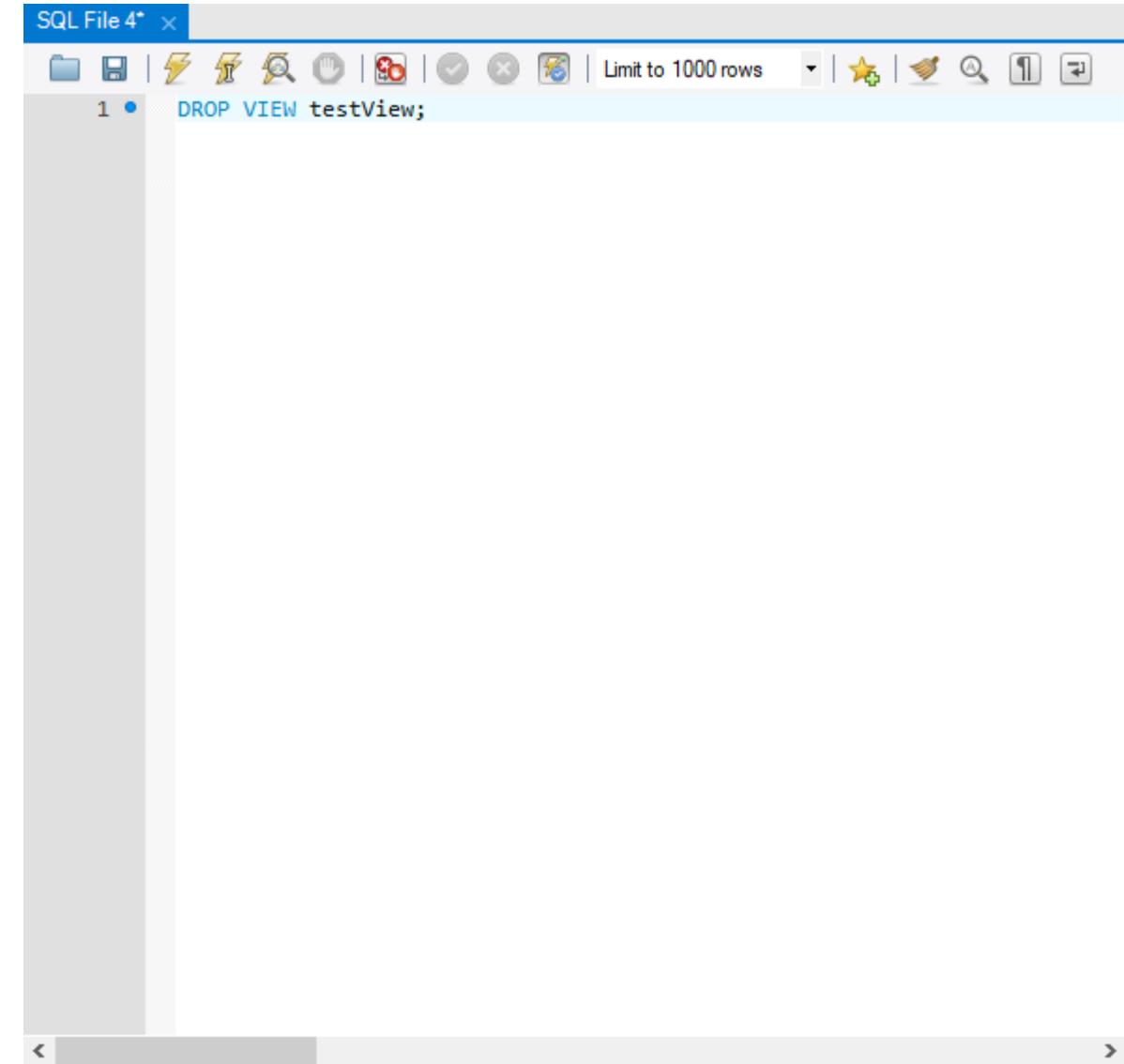
On the right side of the interface, there is a vertical toolbar with three items:

- Result Grid (selected)
- Form Editor
- Field Types

At the bottom of the window, there is a status bar with the text "testView 15" and a "Read Only" indicator.

DROP VIEW

- DROP 문을 사용하여 생성된 뷰를 삭제



The screenshot shows the MySQL Workbench interface with the SQL editor tab selected. The title bar says "SQL File 4". The editor window contains the SQL command:

```
1 ● DROP VIEW testView;
```

The interface includes standard database management tools like file operations, search, and refresh buttons at the top, and a toolbar with various icons below the title bar.

**city, country, countrylanguage 테이블을 JOIN하고,
한국에 대한 정보만 뷰 생성하기**

INSERT

- 테이블 이름 다음에 나오는 열 생략 가능
- 생략할 경우에 VALUE 다음에 나오는 값들의 순서 및 개수가 테이블이 정의된 열 순서 및 개수와 동일해야 함

The screenshot shows the MySQL Workbench interface with a query editor window titled 'test'. The code entered is:

```
1 • INSERT INTO test  
2     VALUE(1, 123, 1.1, "Test");  
3  
4 • SELECT *  
5     FROM test;  
6
```

The 'Result Grid' tab is selected, displaying the following data:

	id	col1	col2	col3
▶	1	123	1.1	Test
*	NULL	NULL	NULL	NULL

The right sidebar contains icons for Result Grid, Form Editor, Field Types, and Query Stats.

INSERT (MySQL Workbench)

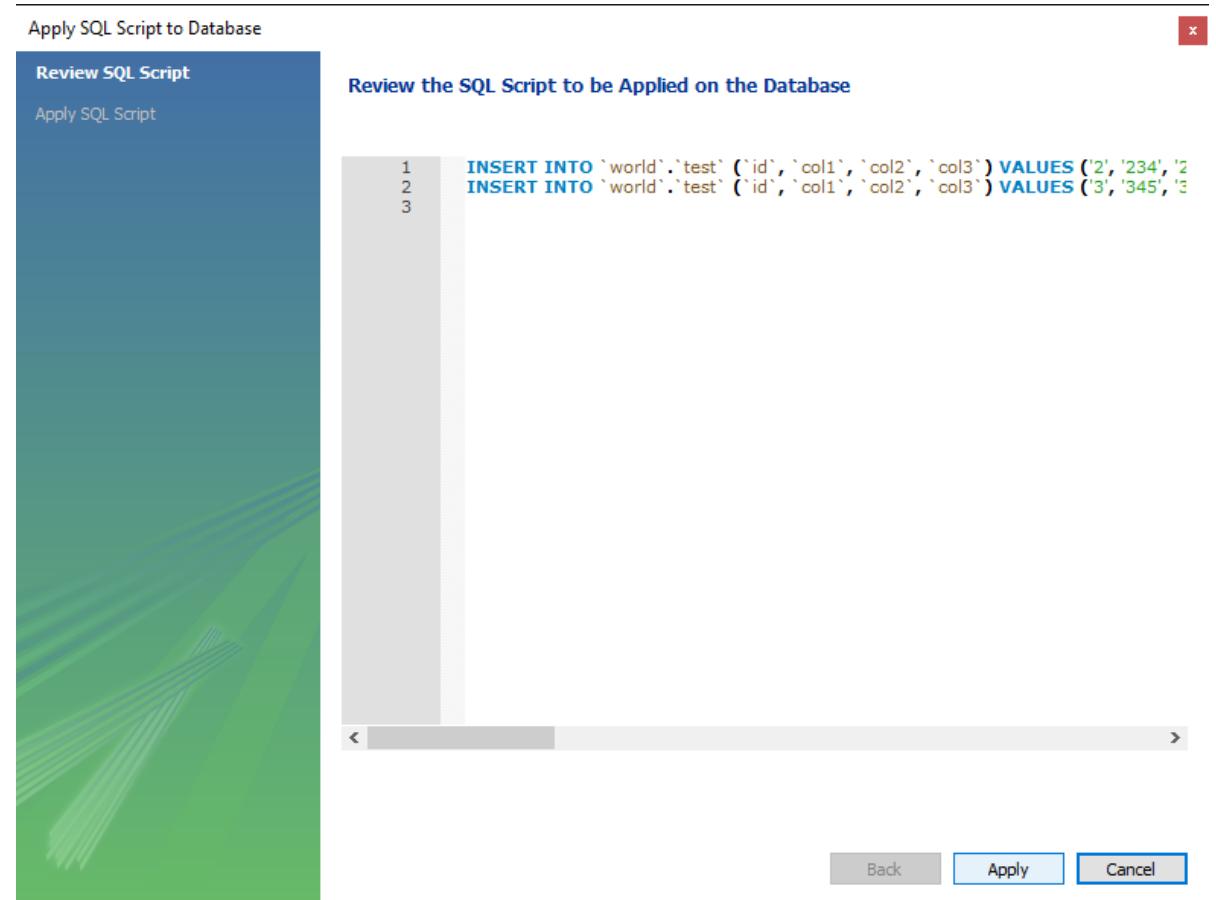
The screenshot shows the MySQL Workbench interface. In the top-left window, titled 'test', there is a query editor with the following SQL script:

```
1 • INSERT INTO test
2   VALUE(1, 123, 1.1, "Test");
3
4 • SELECT *
5   FROM test;
6
```

Below the editor is a results grid displaying the data from the 'test' table:

	id	col1	col2	col3
1	123	1.1	Test	
NULL	234	2.2	Test	
NULL	345	3.3	Test	
NULL	NULL	NULL		

The bottom part of the interface shows tabs for 'test 4' and 'test 5', along with 'Apply' and 'Revert' buttons.



INSERT INTO SELECT

- test 테이블에 있는 내용을 test2 테이블에 삽입

The screenshot shows the MySQL Workbench interface with a query editor window titled 'test'. The editor contains two queries:

```
1 • INSERT INTO test2 SELECT * FROM test;
2 • SELECT * FROM test2;
```

The result grid displays the data from the 'test2' table, which has four columns: id, col1, col2, and col3. The data is as follows:

	id	col1	col2	col3
▶	1	123	1.1	Test
	2	234	2.2	Test
	3	345	3.3	Test
*	NULL	NULL	NULL	NULL

The right side of the interface features a vertical toolbar with icons for different editors: Result Grid, Form Editor, Field Types, and Query Stats.

UPDATE

- 기존에 입력되어 있는 값 변경하는 구문
- WHERE절 생략 가능하나 테이블의 전체 행의 내용 변경

The screenshot shows the MySQL Workbench interface with a query editor window titled 'test'. The editor contains the following SQL code:

```
1 • UPDATE test
2 SET col1=1, col2=1.0, col3='test'
3 WHERE id = 1;
4
5 • SELECT * FROM test;
```

Below the editor is a 'Result Grid' showing the current state of the 'test' table:

	id	col1	col2	col3
▶	1	1	1	test
	2	234	2.2	Test
	3	345	3.3	Test
*	NULL	NULL	NULL	NULL

The right side of the interface features a vertical toolbar with icons for different editors: Result Grid, Form Editor, Field Types, and Query Stats.

DELETE

- 행 단위로 데이터 삭제하는 구문
- DELETE FROM 테이블이름 WHERE 조건;
- 데이터는 지워지지만 테이블 용량은 줄어들지 않음
- 원하는 데이터만 지울 수 있음
- 삭제 후 잘못 삭제한 것을 되돌릴 수 있음

The screenshot shows the MySQL Workbench interface. In the top-left pane, there is a query editor window titled 'test'. It contains two queries:

```
1 • DELETE FROM test  
WHERE id = 1;  
2  
3  
4 • SELECT * FROM test;
```

The result grid below shows the state of the 'test' table:

	id	col1	col2	col3
▶	2	234	2.2	Test
	3	345	3.3	Test
*	NULL	NULL	NULL	NULL

The bottom right corner of the interface has a vertical toolbar with icons for 'Result Grid', 'Form Editor', 'Field Types', and 'Query Stats'.

TRUNCATE

- 용량이 줄어 들고, 인덱스 등도 모두 삭제
- 테이블은 삭제하지는 않고, 데이터만 삭제
- 한꺼번에 다 지워야 함
- 삭제 후 절대 되돌릴 수 없음

The screenshot shows the MySQL Workbench interface with a query editor window titled 'test'. The editor contains the following SQL code:

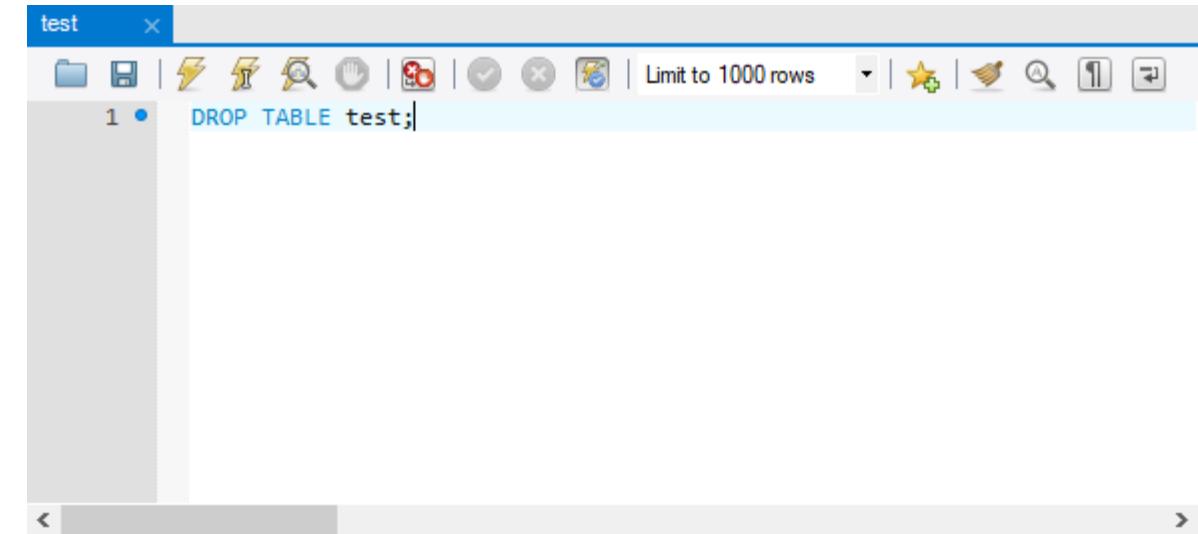
```
TRUNCATE TABLE test;
SELECT * FROM test;
```

The results pane below shows a table structure with columns: id, col1, col2, col3. There are no rows present in the result grid.

On the right side of the interface, there is a vertical toolbar with icons for different editors: Result Grid (selected), Form Editor, Field Types, and Query Stats.

DROP TABLE

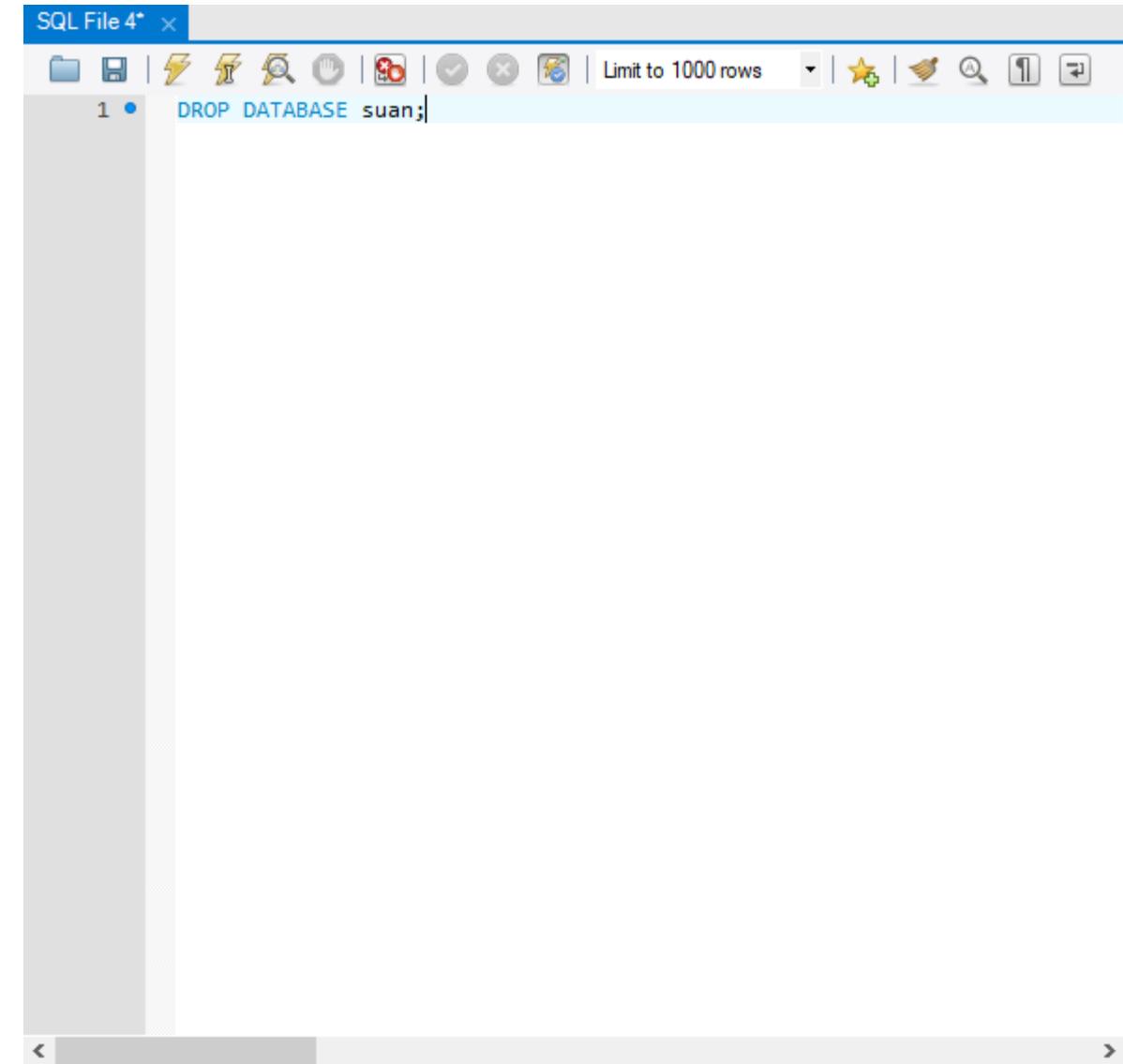
- 데이터 전체를 삭제, 공간, 객체를 삭제
- 삭제 후 절대 되돌릴 수 없음



The screenshot shows the MySQL Workbench interface. The title bar says 'test'. The main area contains the SQL command 'DROP TABLE test;'. The status bar at the bottom right indicates 'Limit to 1000 rows'.

DROP DATABASE

- DROP DATABASE 문은 해당 데이터베이스를 삭제



The screenshot shows the MySQL Workbench interface with the title bar "SQL File 4". The main area contains the SQL command:

```
1 • DROP DATABASE suan;
```

The interface includes various toolbars and buttons typical of a database management tool.

자신만의 연락처 테이블 만들기
이름, 전화번호, 주소, 이메일, ...

자유 주제로 데이터들을 저장할 테이블 만들기

(참고) 데이터 타입:

<https://dev.mysql.com/doc/refman/8.0/en/data-types.html>

