



이수안 컴퓨터 연구소

suan computer laboratory



MySQL 한번에 끝내기

The screenshot displays the MySQL Workbench interface with three main panes:

- EER Diagram:** Shows a database schema with tables like 'customer', 'address', 'film', and 'film_actor' connected by relationships.
- Query Editor:** Contains a SQL query:

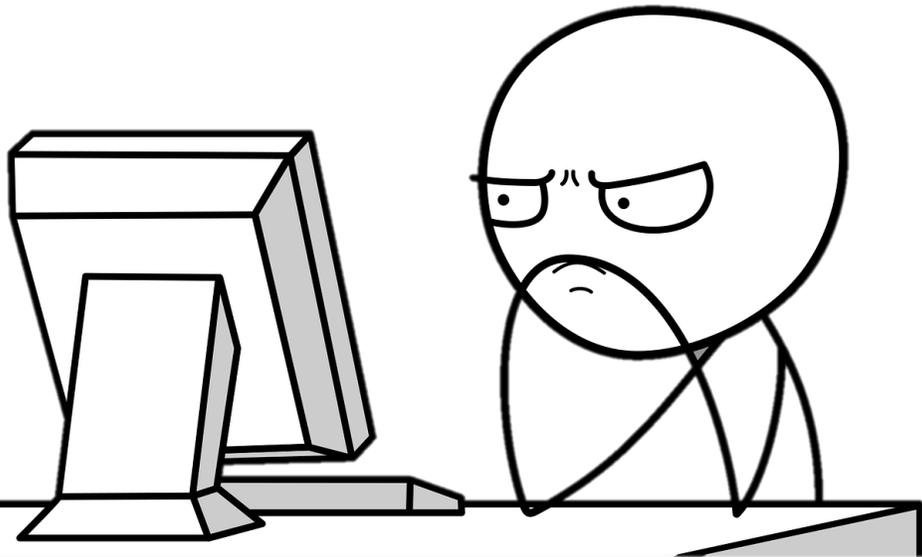
```
SELECT actor_id, actor_name, last_name, last_update FROM sakila.actor;
```
- Query Results:** Shows a table with columns: film_id, title, description, release_year, language_id, original_language, rental_duration, rental_rate, length. The results list various films such as 'ACADEMY DIN...', 'ACI COLDFIN...', 'ADAPTATION...', etc.



목차

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

1. MySQL 소개

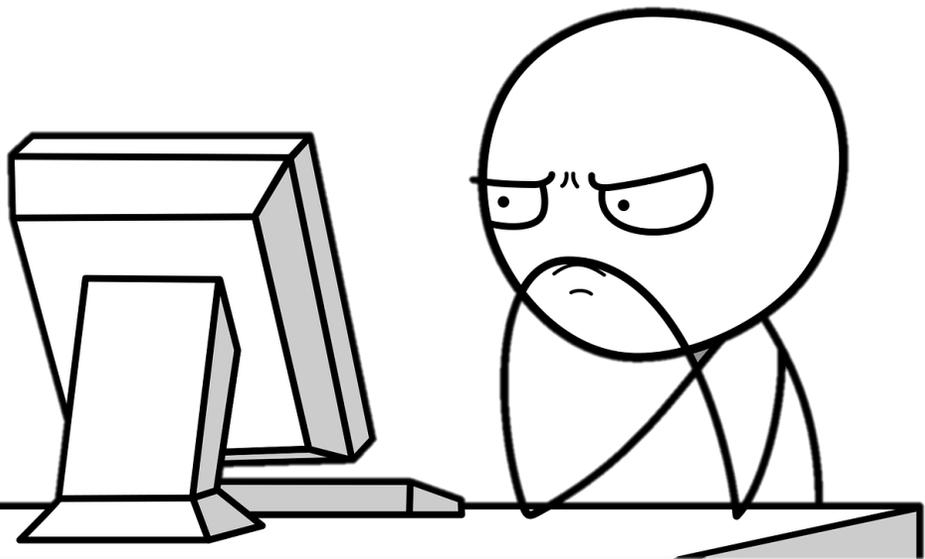


MySQL 소개

- 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 PHP와 함께 웹 개발에 자주 사용
- MySQL은 오픈 소스 라이선스를 따르기는 하지만, 상업적으로 사용할 때는 상업용 라이선스 구입 필요



2. MySQL 설치



MySQL 다운로드



The world's most popular open source database



[Contact MySQL](#) | [Login](#) | [Register](#)

[MYSQL.COM](#)

[DOWNLOADS](#)

[DOCUMENTATION](#)

[DEVELOPER ZONE](#)



[Products](#) [Cloud](#) [Services](#) [Partners](#) [Customers](#) [Why MySQL?](#) [News & Events](#) [How to Buy](#)



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 다운로드



The world's most popular open source database



[Contact MySQL](#) | [Login](#) | [Register](#)

[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 on Windows

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](#)

MySQL open source software is provided under the GPL License.

OEMs, ISVs and VARs can purchase commercial licenses.

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

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

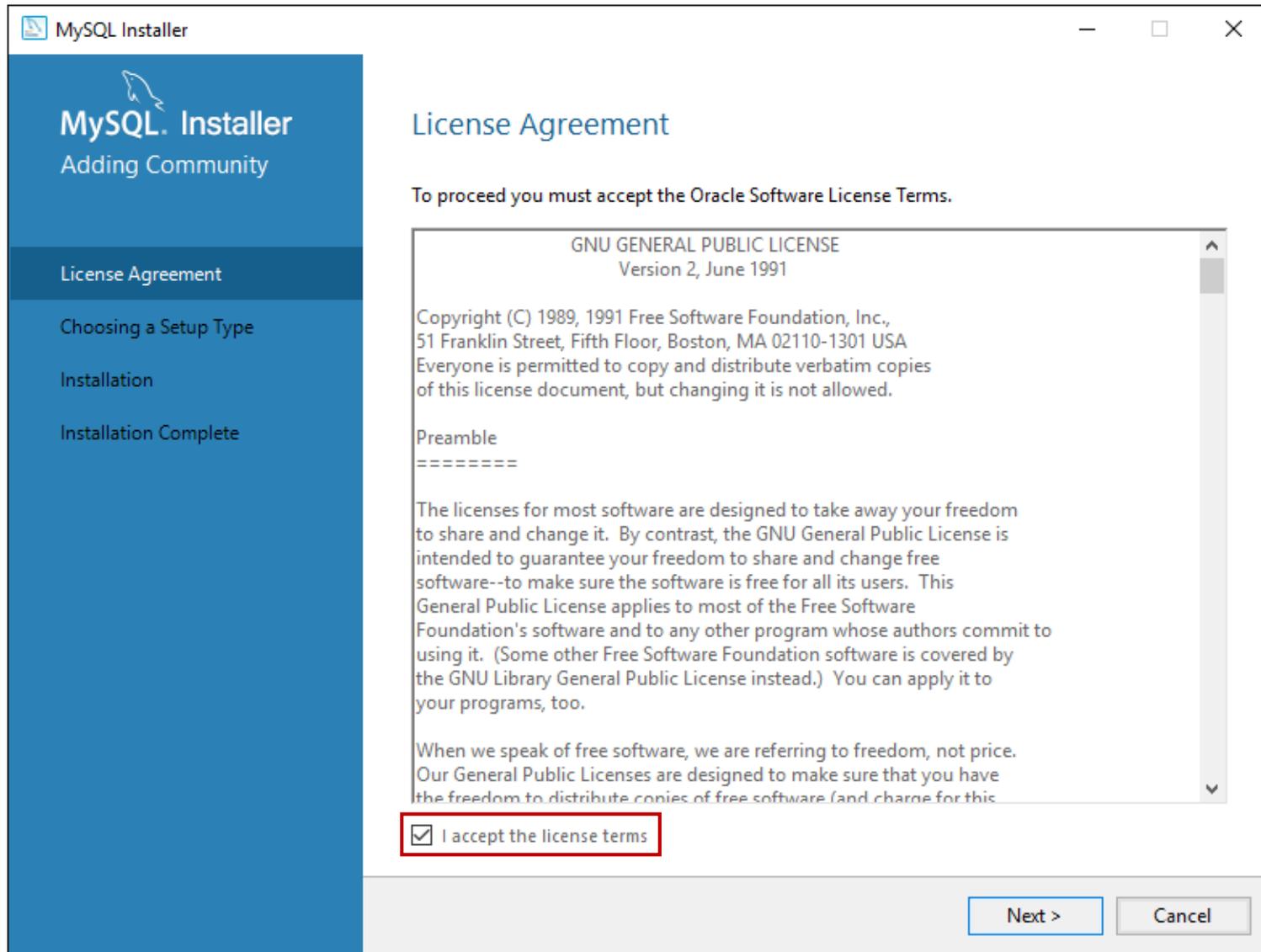
[Looking for previous GA versions?](#)

Windows (x86, 32-bit), MSI Installer <small>(mysql-installer-web-community-8.0.13.0.msi)</small>	8.0.13	16.3M	Download
Windows (x86, 32-bit), MSI Installer <small>(mysql-installer-community-8.0.13.0.msi)</small>	8.0.13	313.8M	Download

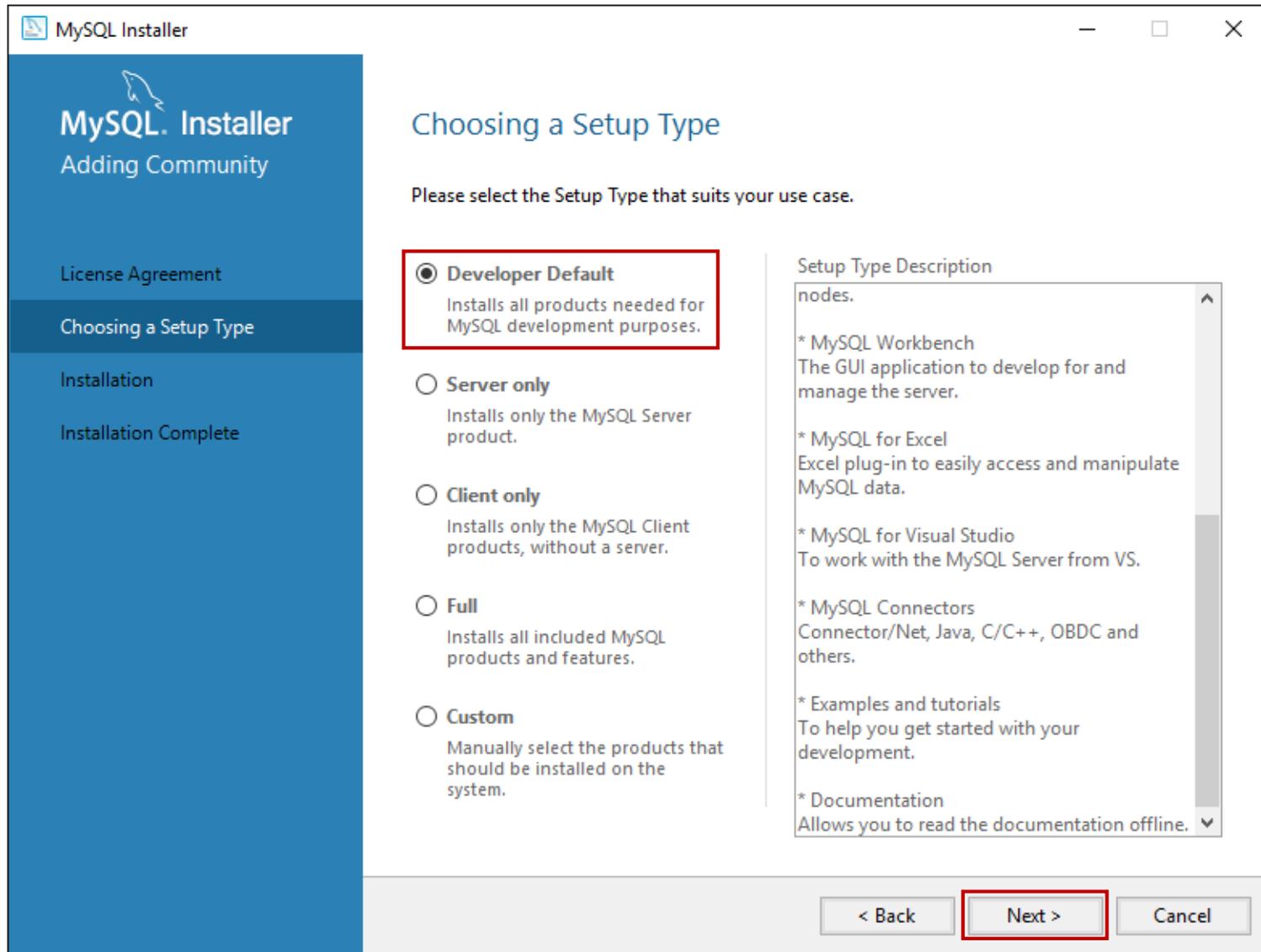


We suggest that you use the [MD5 checksums](#) and [GnuPG signatures](#) to verify the integrity of the packages you download.

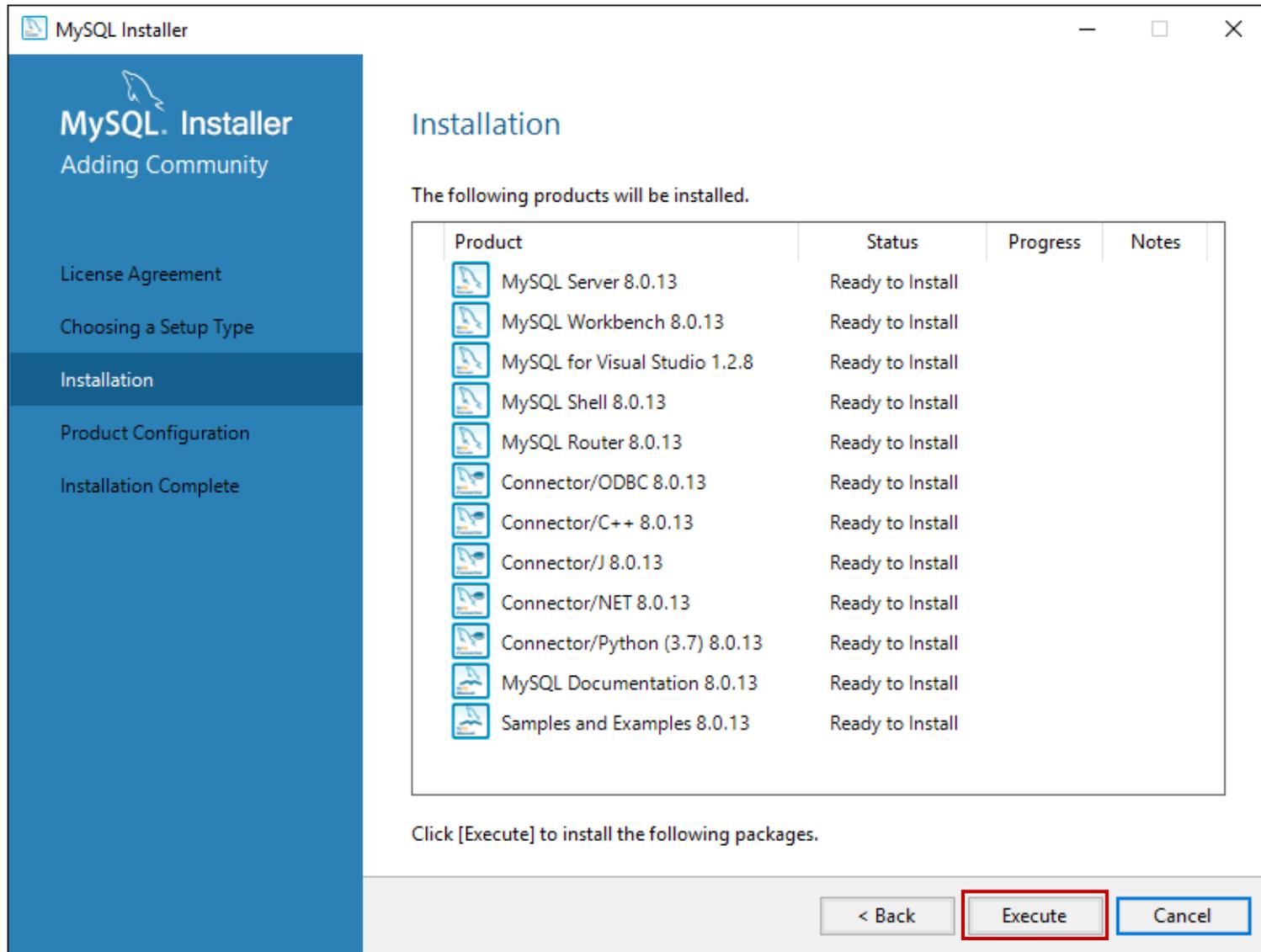
MySQL 설치



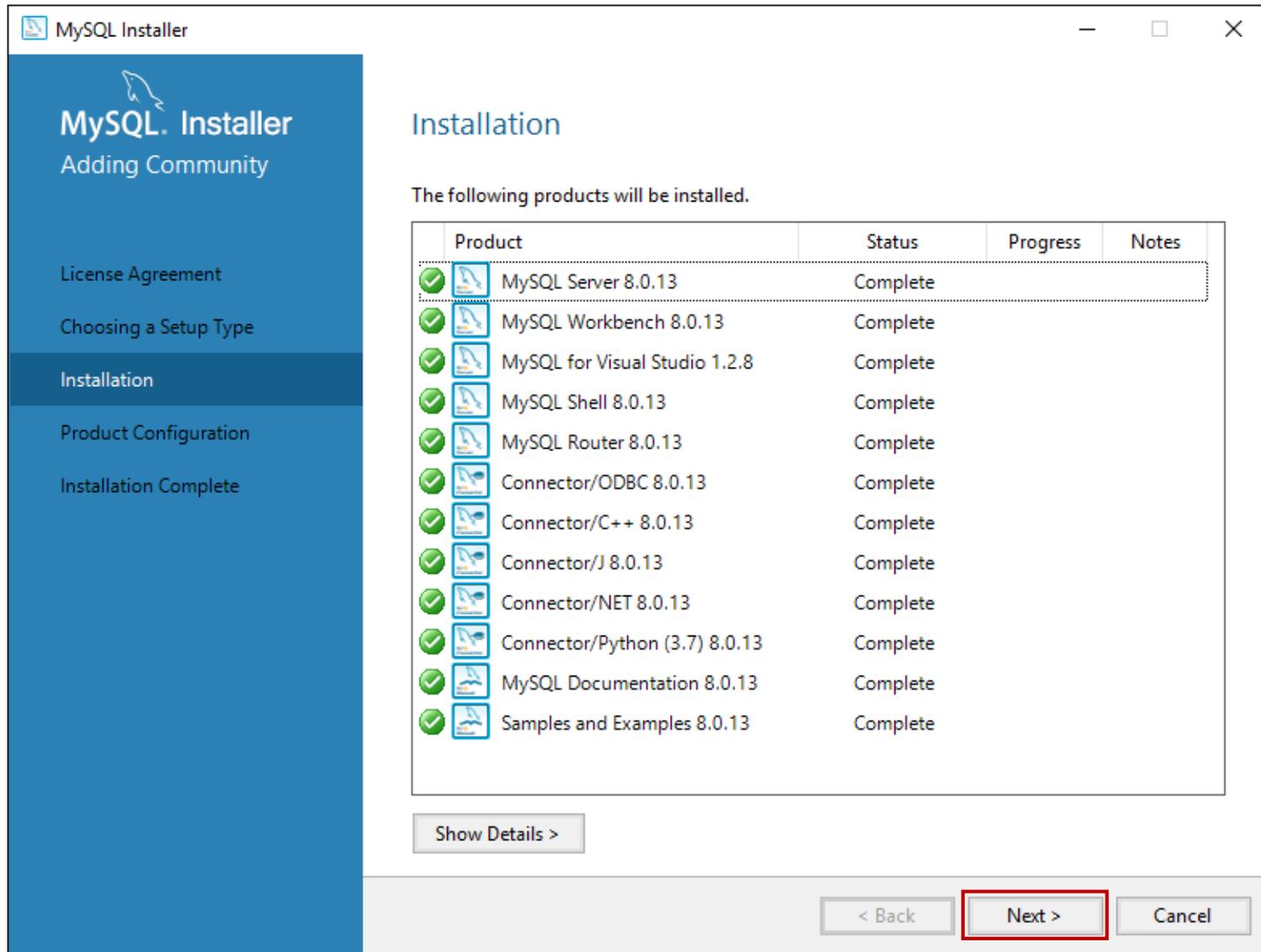
MySQL 설치



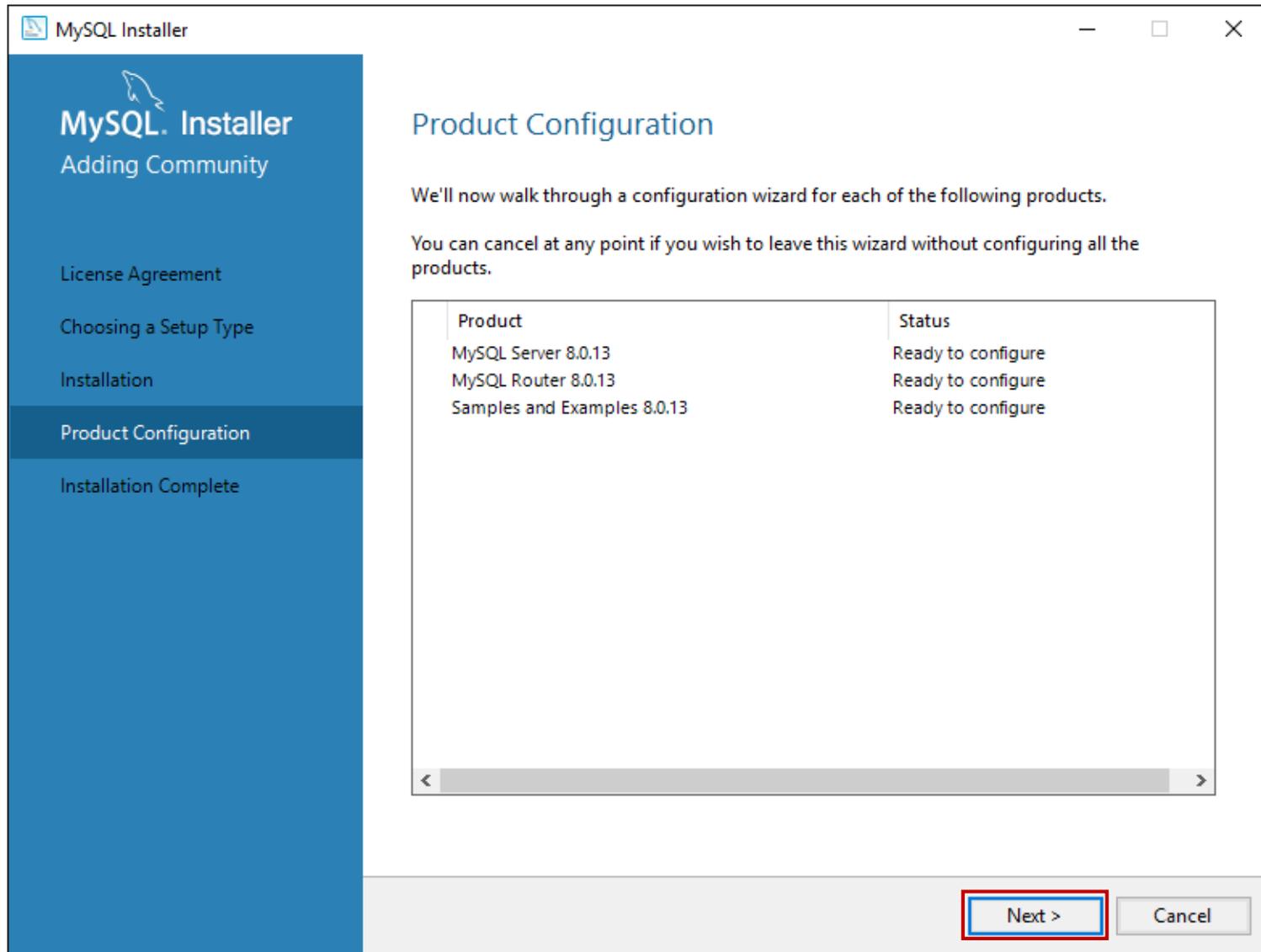
MySQL 설치



MySQL 설치



MySQL 설치



MySQL 설치

MySQL Installer

MySQL Server 8.0.13

Group Replication

Type and Networking

Authentication Method

Accounts and Roles

Windows Service

Apply Configuration

Group Replication

Standalone MySQL Server / Classic MySQL Replication
Choose this option if you want to run the MySQL Server either standalone with the opportunity to later configure classic MySQL Replication.

Using this option you can manually configure your replication setup and provide your own high availability solution if required.

Sandbox InnoDB Cluster Setup (for testing only)
The [InnoDB cluster](#) technology provides an out-of-the-box HA (high availability) solution for MySQL using Group Replication technology.

This option allows you to test an InnoDB cluster setup on your local computer using several MySQL Server sandbox instances. Read more about this [here](#).

To setup a real-world production InnoDB cluster please choose the standard MySQL Server configuration instead on all desired hosts and use the MySQL Shell afterwards to create or expand the InnoDB cluster setup.

Client App ↔ MySQL Router ↔ MySQL Shell ↔ InnoDB Cluster

Next > Cancel

MySQL 설치

MySQL Installer

MySQL Server 8.0.13

Group Replication

Type and Networking

Authentication Method

Accounts and Roles

Windows Service

Apply Configuration

Type and Networking

Server Configuration Type

Choose the correct server configuration type for this MySQL Server installation. This setting will define how much system resources are assigned to the MySQL Server instance.

Config Type: **Development Computer**

Connectivity

Use the following controls to select how you would like to connect to this server.

TCP/IP Port: X Protocol Port:

Open Windows Firewall ports for network access

Named Pipe Pipe Name:

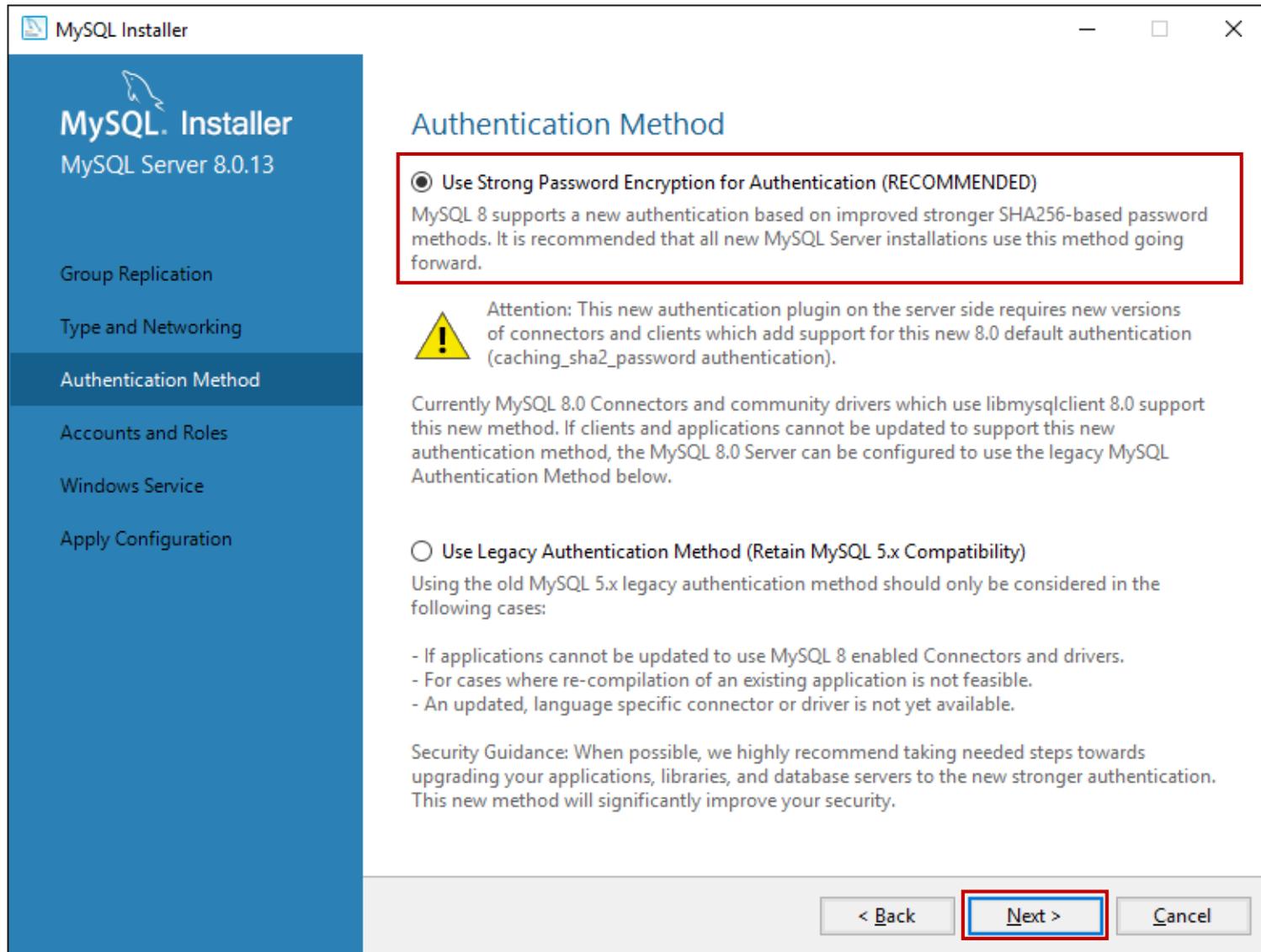
Shared Memory Memory Name:

Advanced Configuration

Select the check box below to get additional configuration pages where you can set advanced and logging options for this server instance.

Show Advanced and Logging Options

< Back **Next >** Cancel



MySQL 설치

MySQL Installer

MySQL Installer
MySQL Server 8.0.13

Group Replication

Type and Networking

Authentication Method

Accounts and Roles

Windows Service

Apply Configuration

Accounts and Roles

Root Account Password
Enter the password for the root account. Please remember to store this password in a secure place.

MySQL Root Password: 

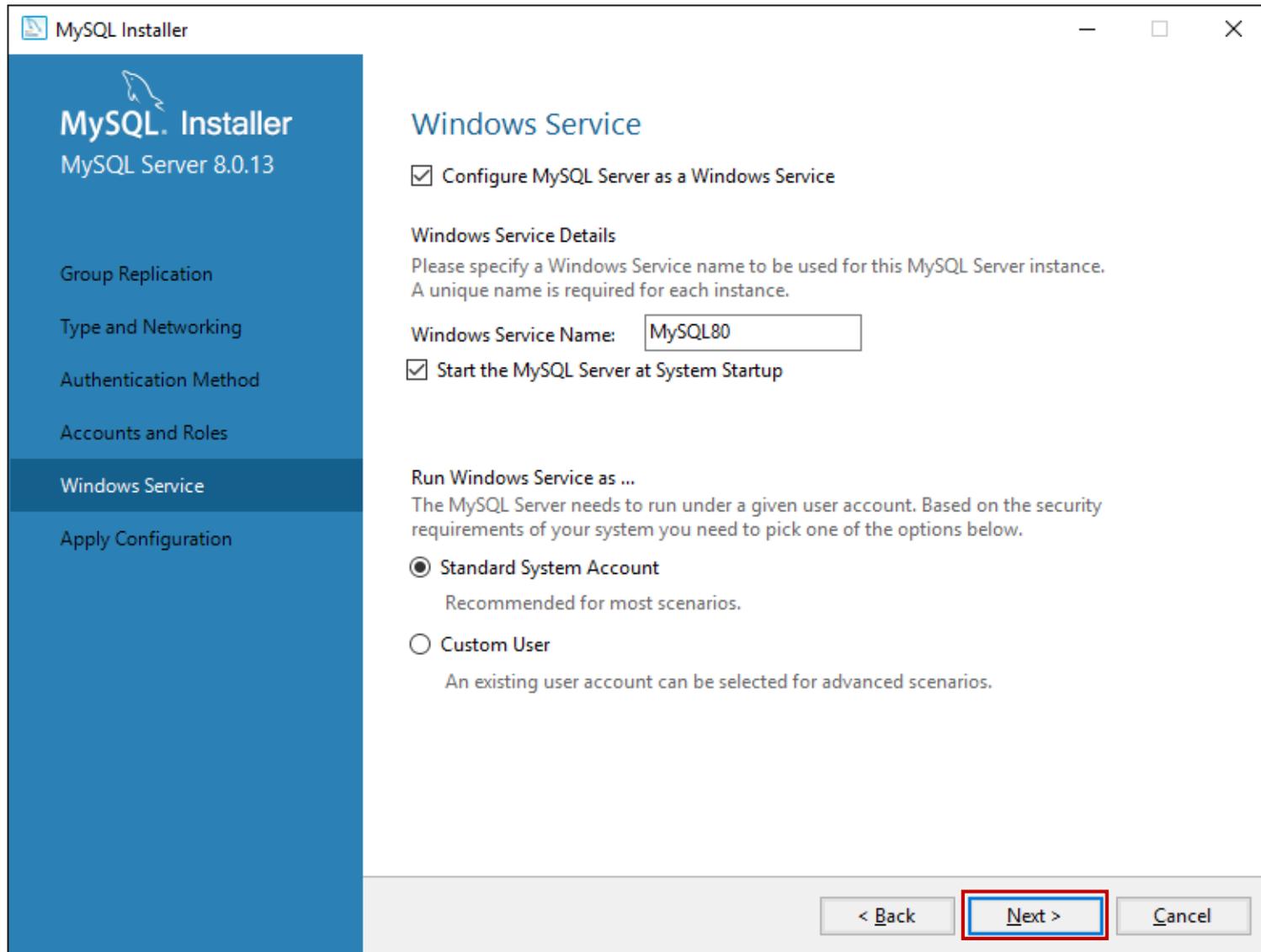
Repeat Password:

MySQL User Accounts
Create MySQL user accounts for your users and applications. Assign a role to the user that consists of a set of privileges.

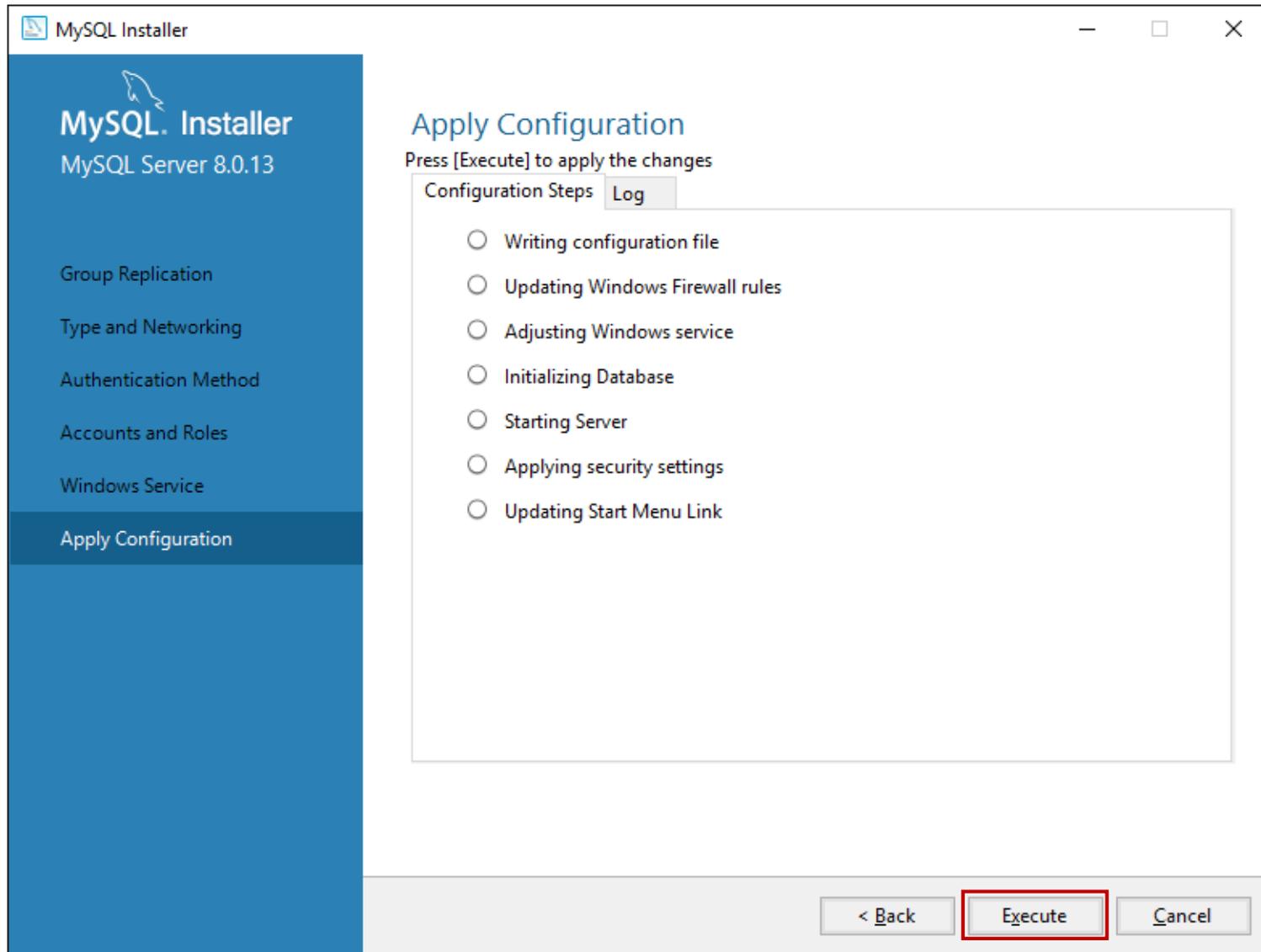
MySQL User Name	Host	User Role
-----------------	------	-----------

< Back **Next >** Cancel

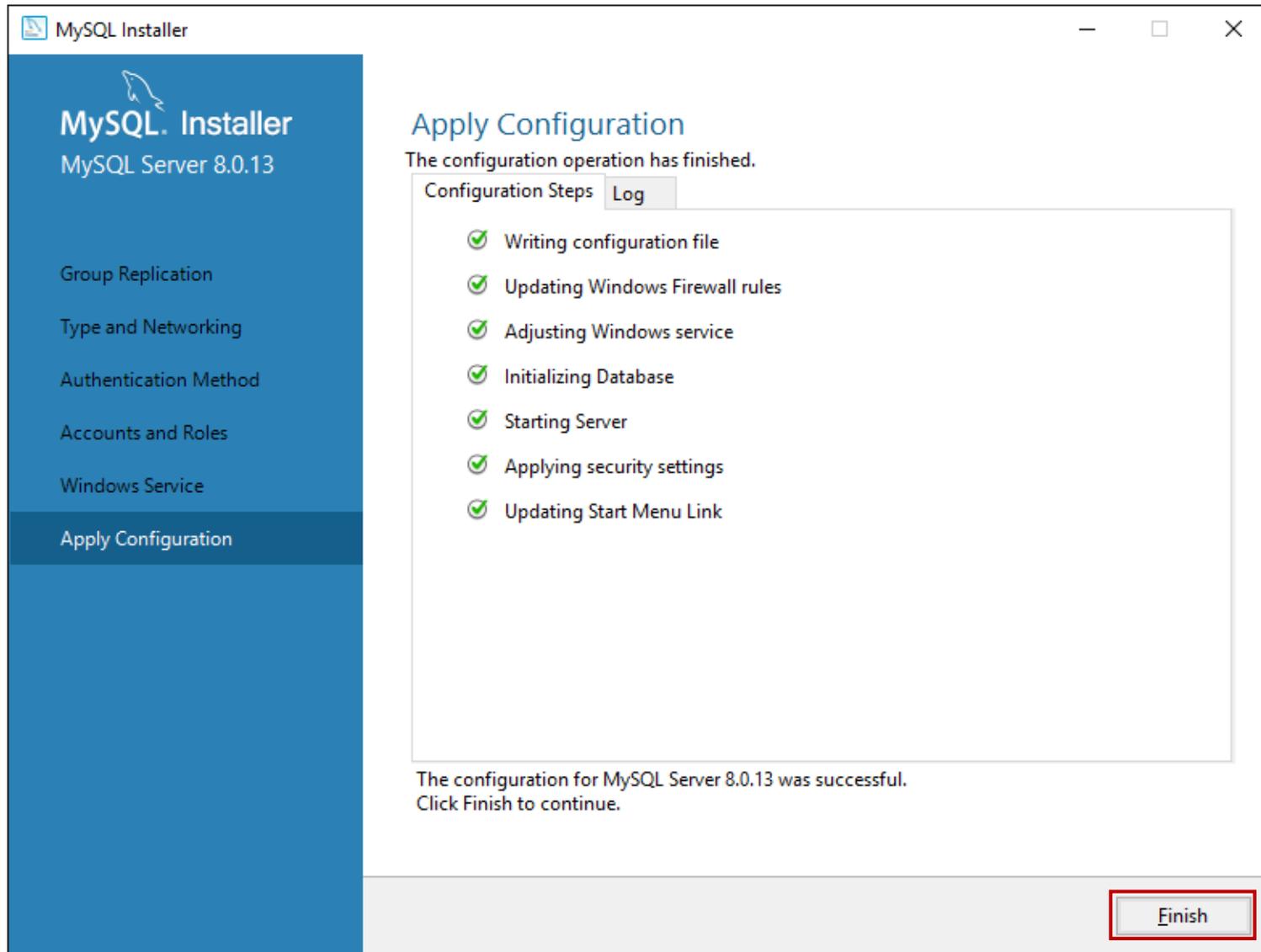
MySQL 설치



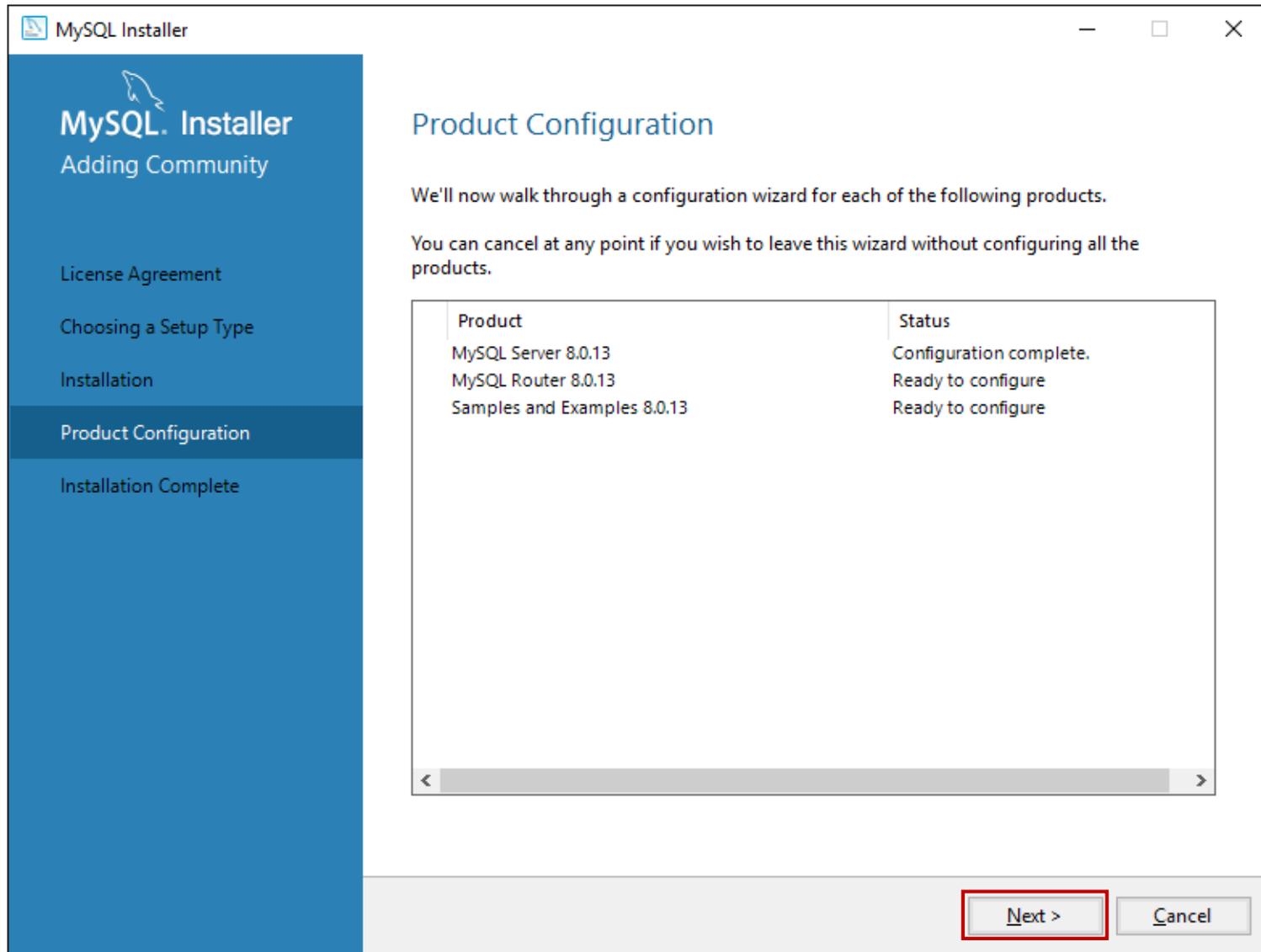
MySQL 설치



MySQL 설치



MySQL 설치



MySQL 설치

MySQL Installer

MySQL Router 8.0.13

MySQL Router Configuration

Bootstrap MySQL Router for use with InnoDB cluster

This wizard can bootstrap MySQL Router to direct traffic between MySQL applications and a MySQL InnoDB cluster. Applications that connect to the router will be automatically directed to an available read/write or read-only member of the cluster.

The bootstrapping process requires a connection to the InnoDB cluster. In order to register the MySQL Router for monitoring, use the current Read/Write instance of the cluster.

Hostname:

Port:

Management User:

Password:

Test Connection

MySQL Router requires specification of a base port (between 80 and 65532). The first port is used for classic read/write connections. The other ports are computed sequentially after the first port. If any port is indicated to be in use, please change the base port.

Classic MySQL protocol connections to InnoDB cluster:

Read/Write:

Read Only:

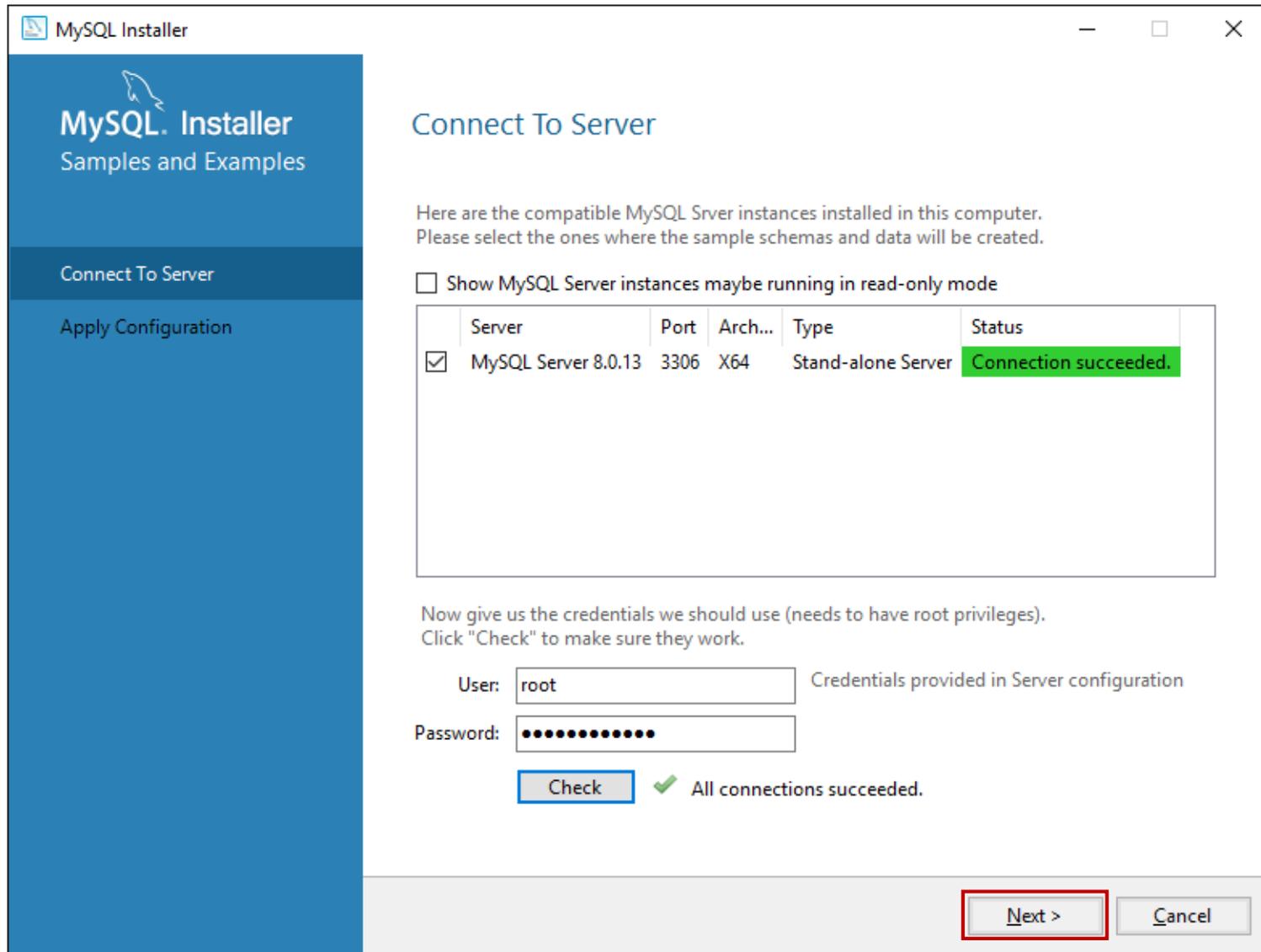
MySQL X protocol connections to InnoDB cluster:

Read/Write:

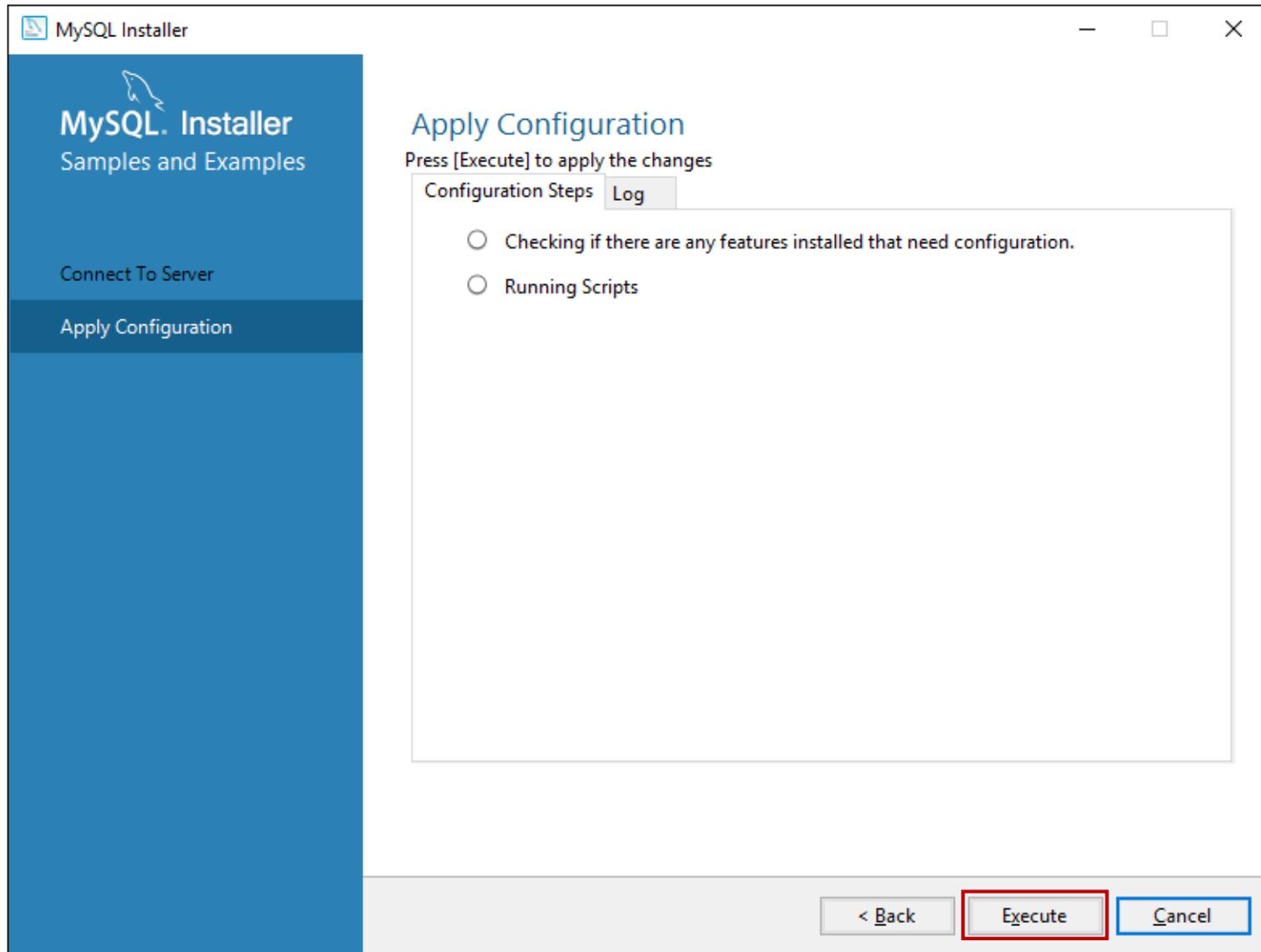
Read Only:

Finish Cancel

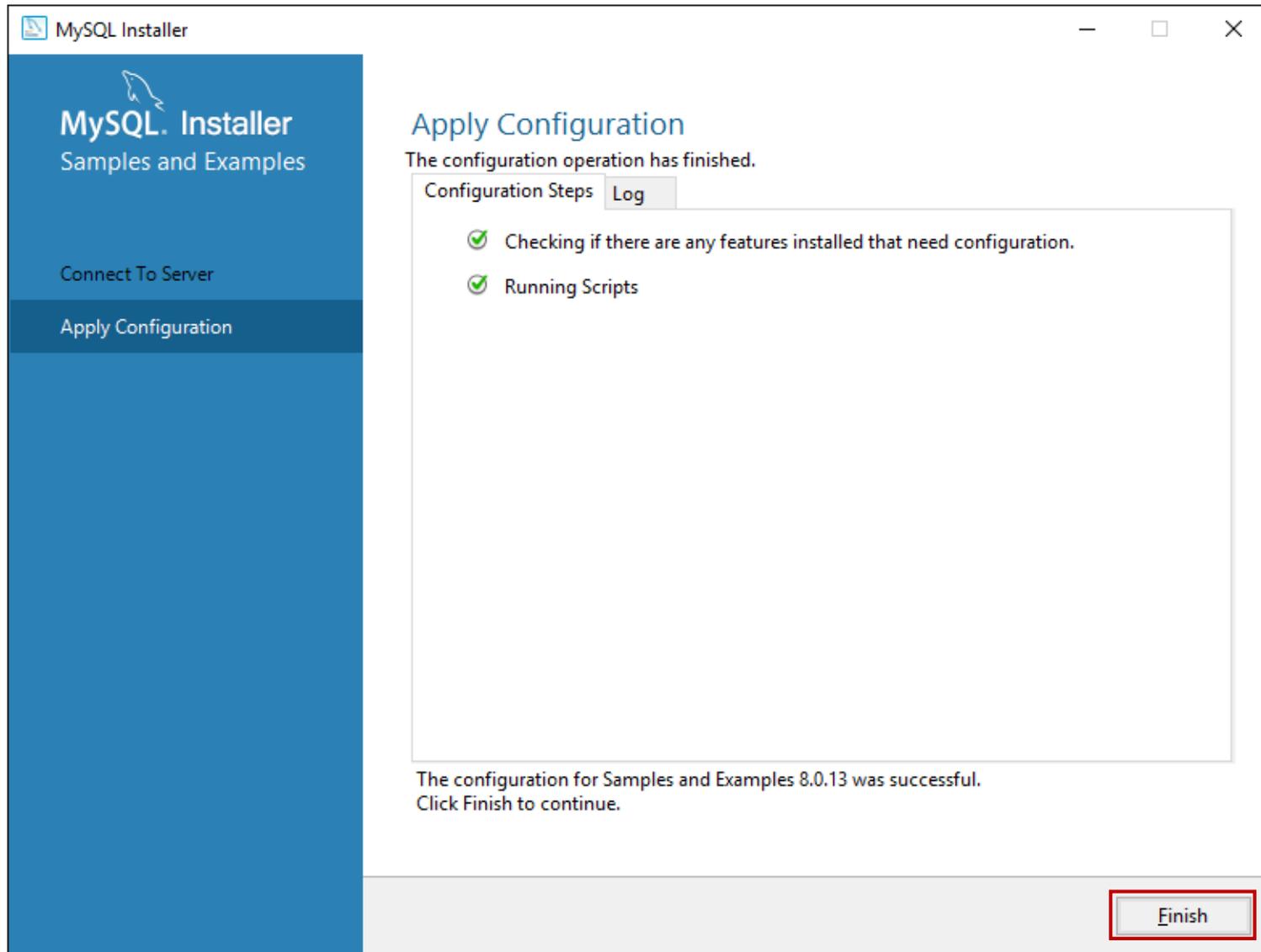
MySQL 설치



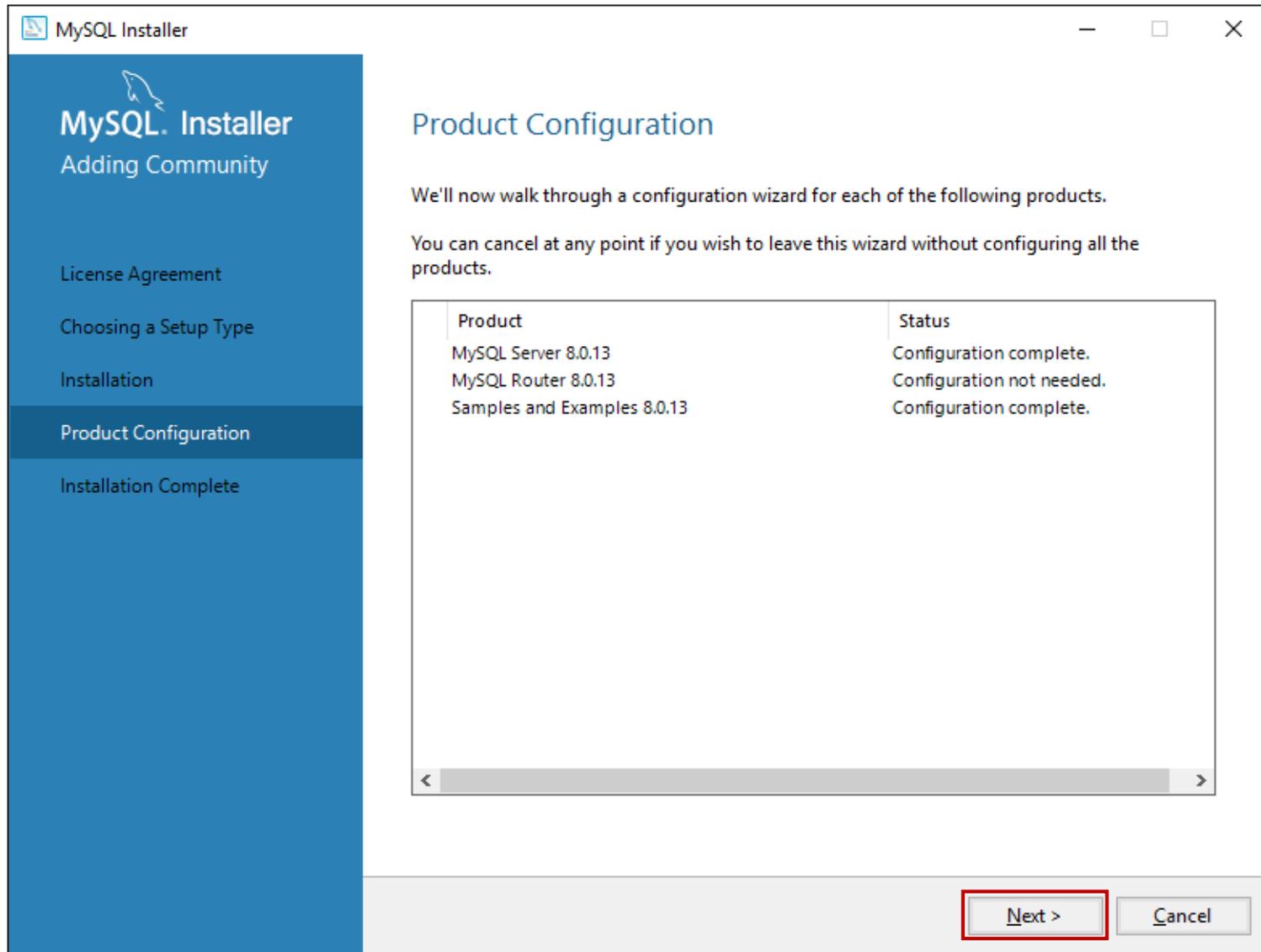
MySQL 설치



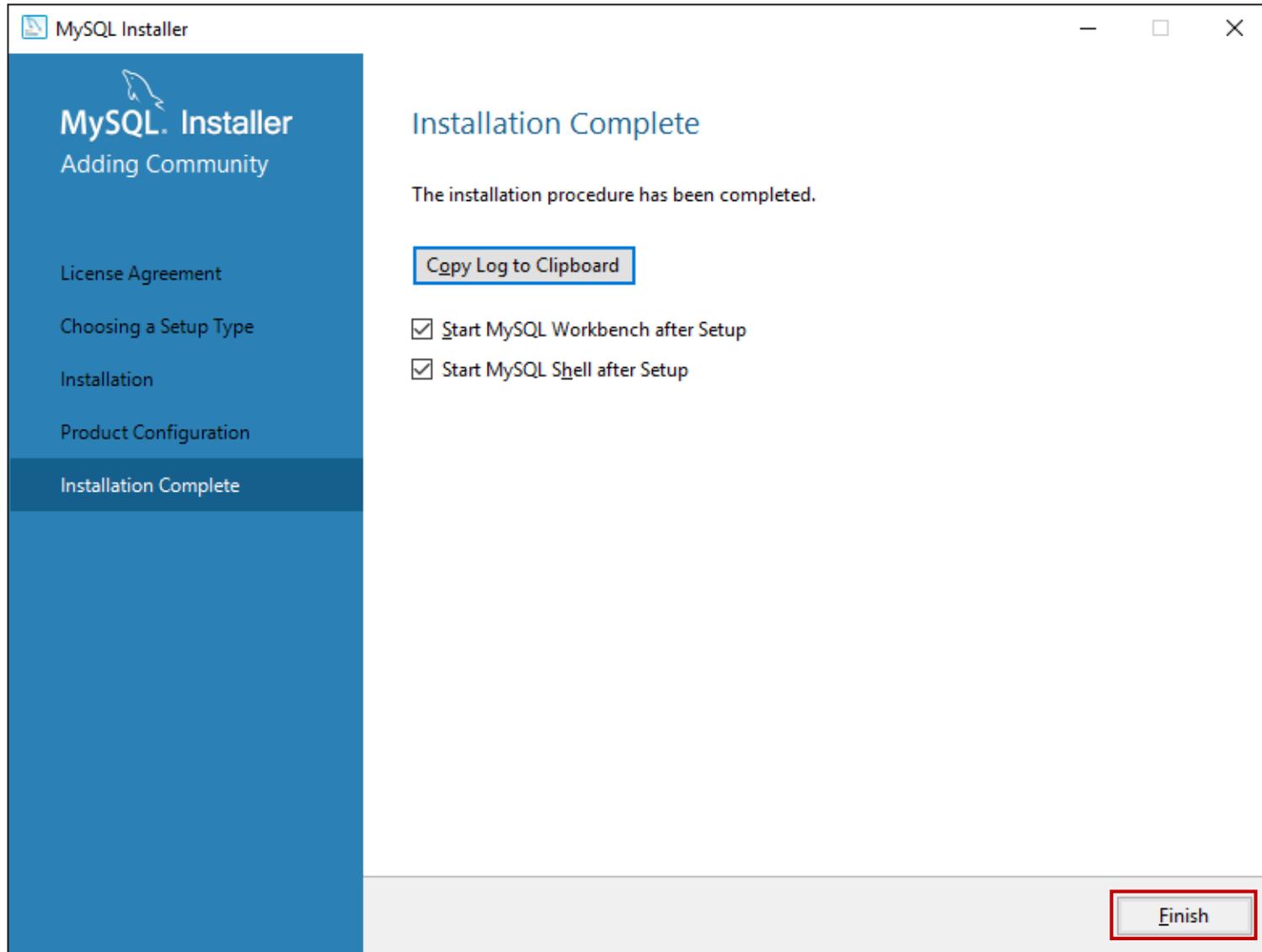
MySQL 설치



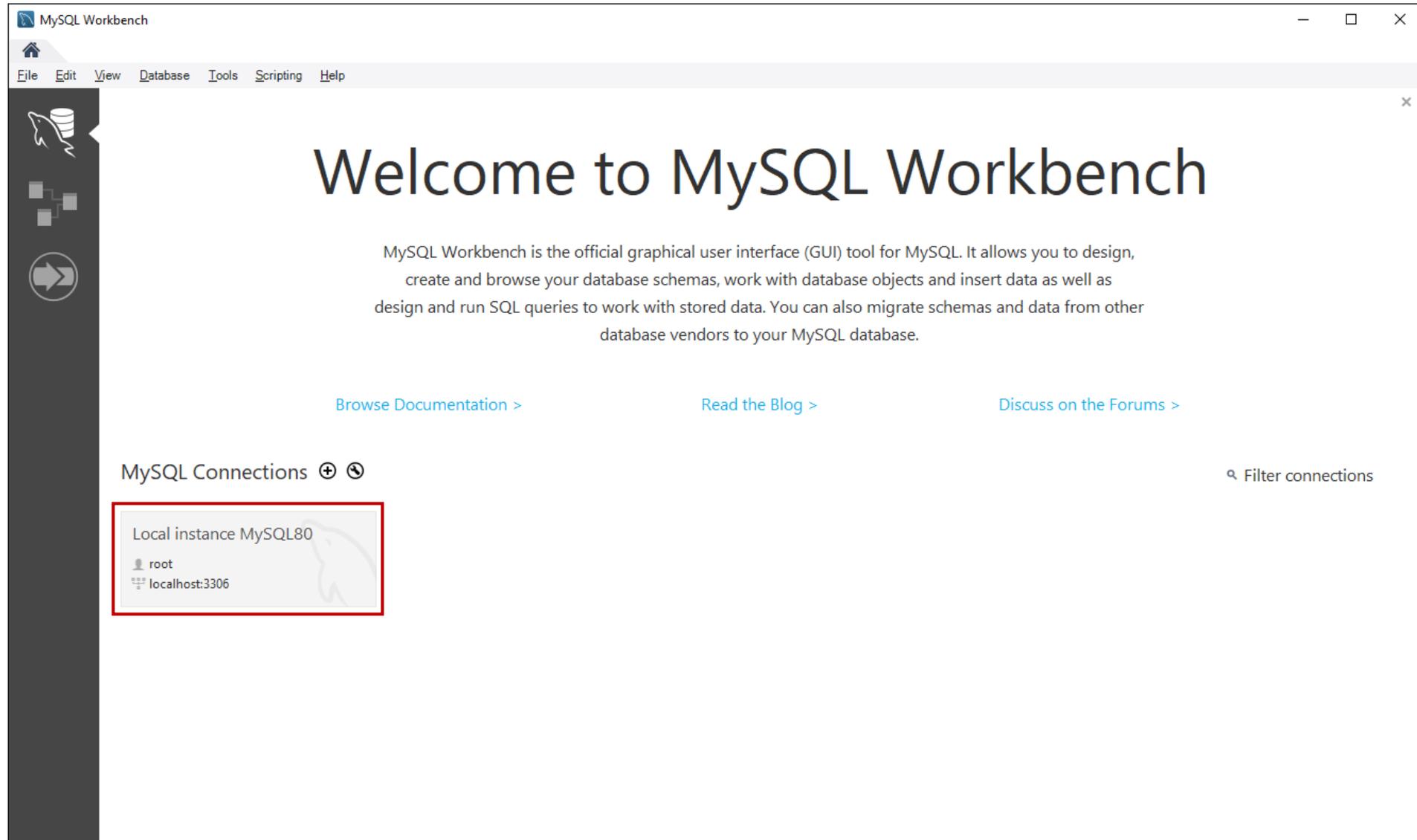
MySQL 설치



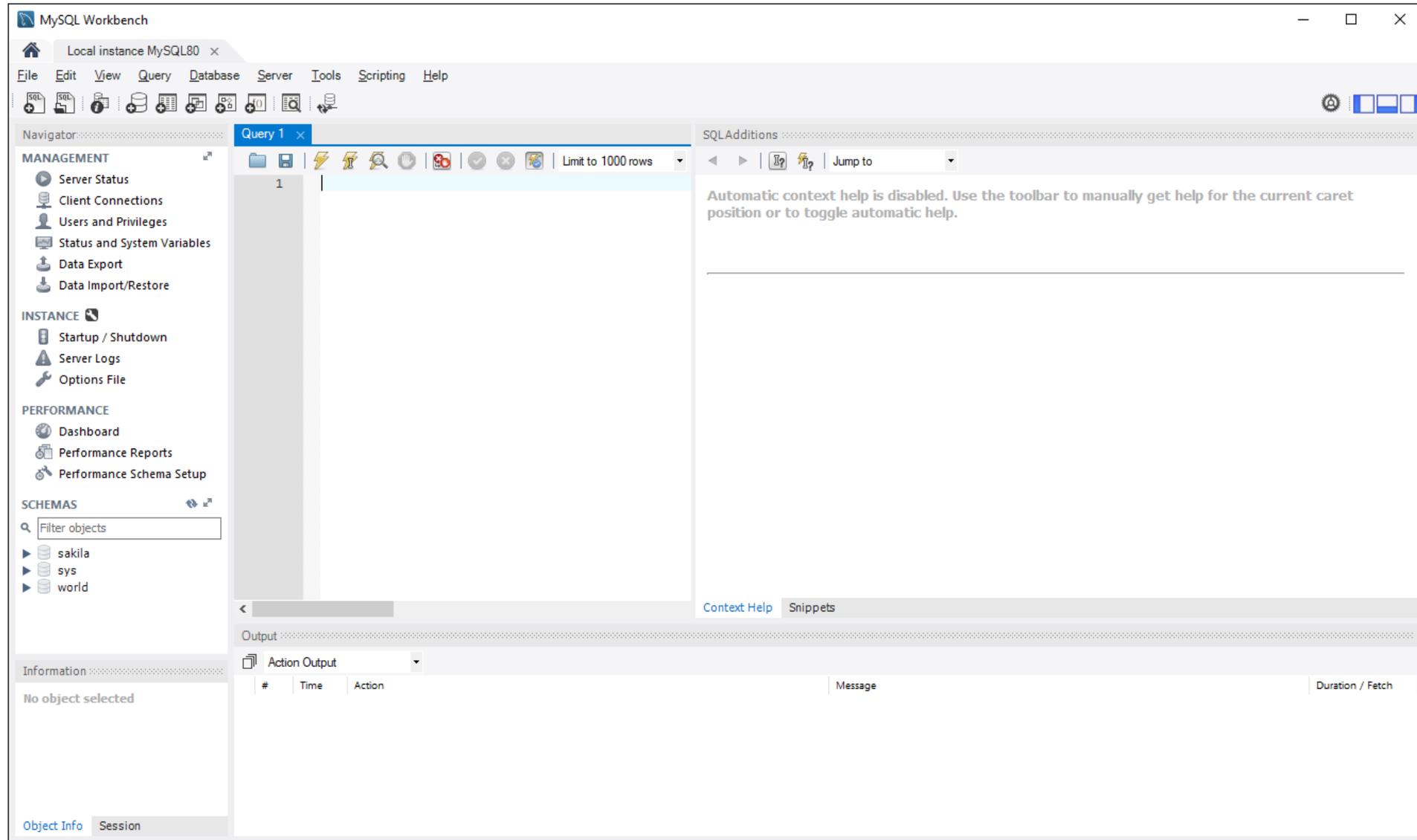
MySQL 설치



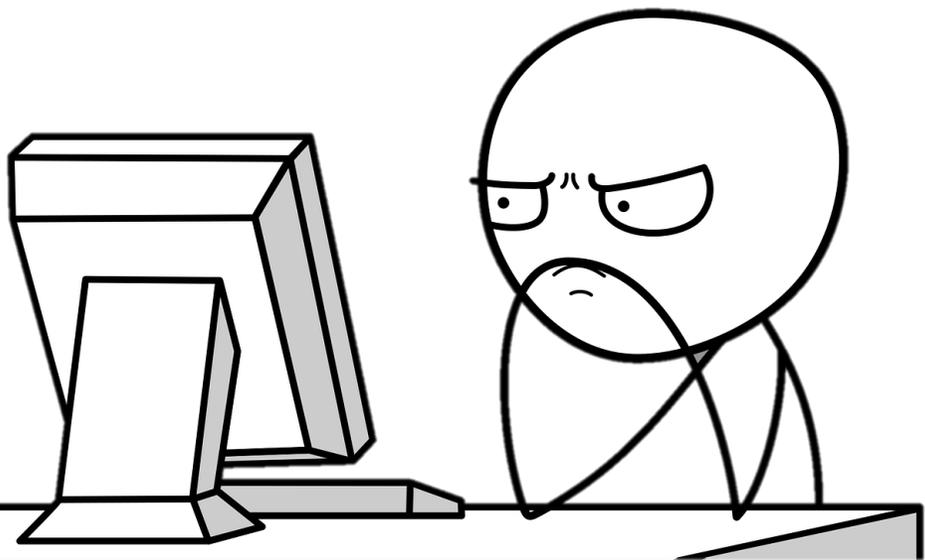
MySQL Workbench



MySQL Workbench



3. SQL 기본



SQL의 분류

■ DML Data Manipulation Language

- 데이터 조작 언어
- 데이터를 조작(선택, 삽입, 수정, 삭제)하는 데 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

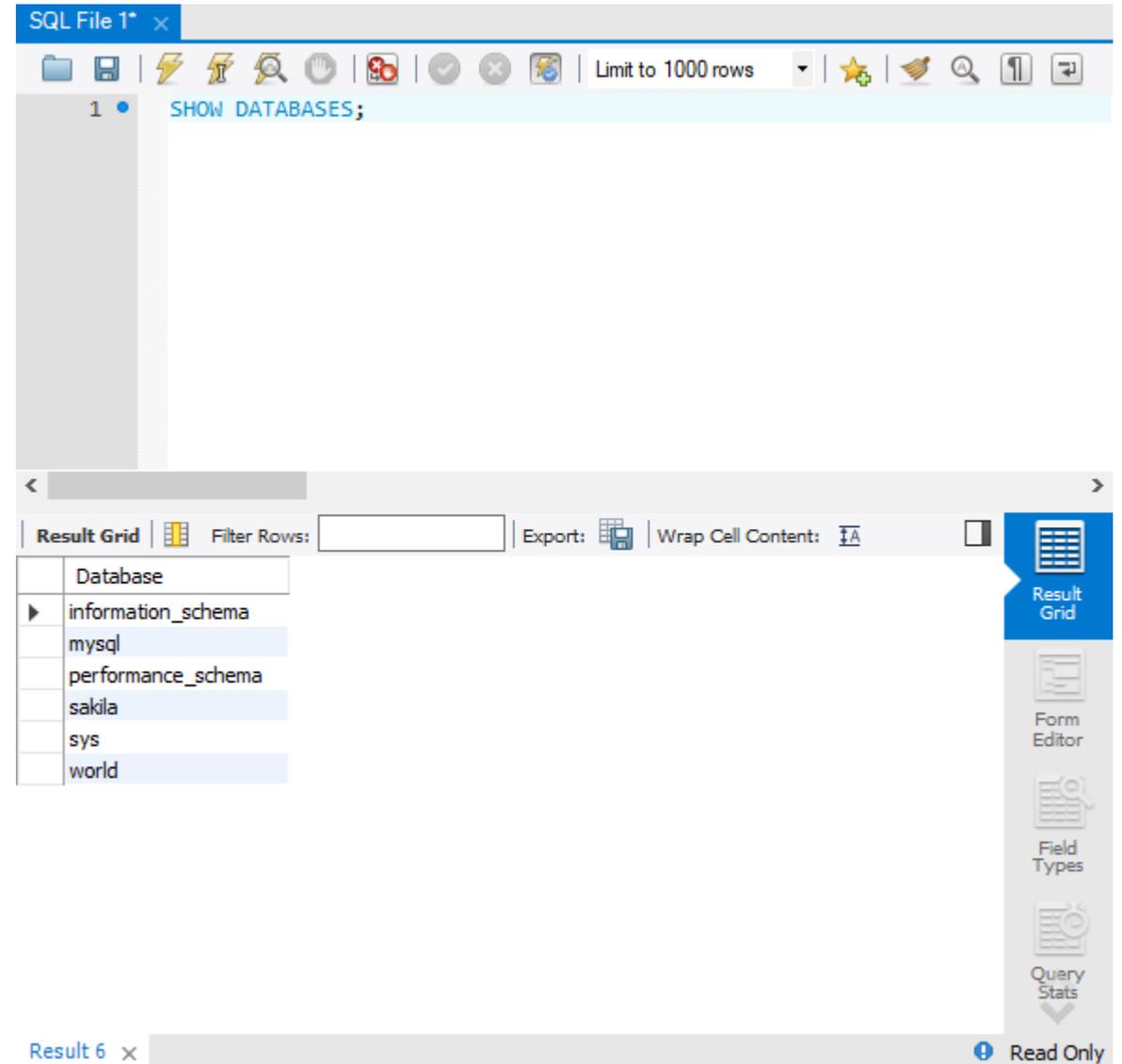
- 데이터 정의 언어
- 데이터베이스, 테이블, 뷰, 인덱스 등의 데이터베이스 개체를 생성/삭제/변경하는 역할
- CREATE, DROP, ALTER 구문
- DDL은 트랜잭션 발생시키지 않음
- ROLLBACK이나 COMMIT 사용 불가
- DDL문은 실행 즉시 MySQL에 적용

■ 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 a MySQL client window with a single SQL file open. The command `SHOW DATABASES;` is entered in the query editor. The results are displayed in a table with the following data:

Database
information_schema
mysql
performance_schema
sakila
sys
world

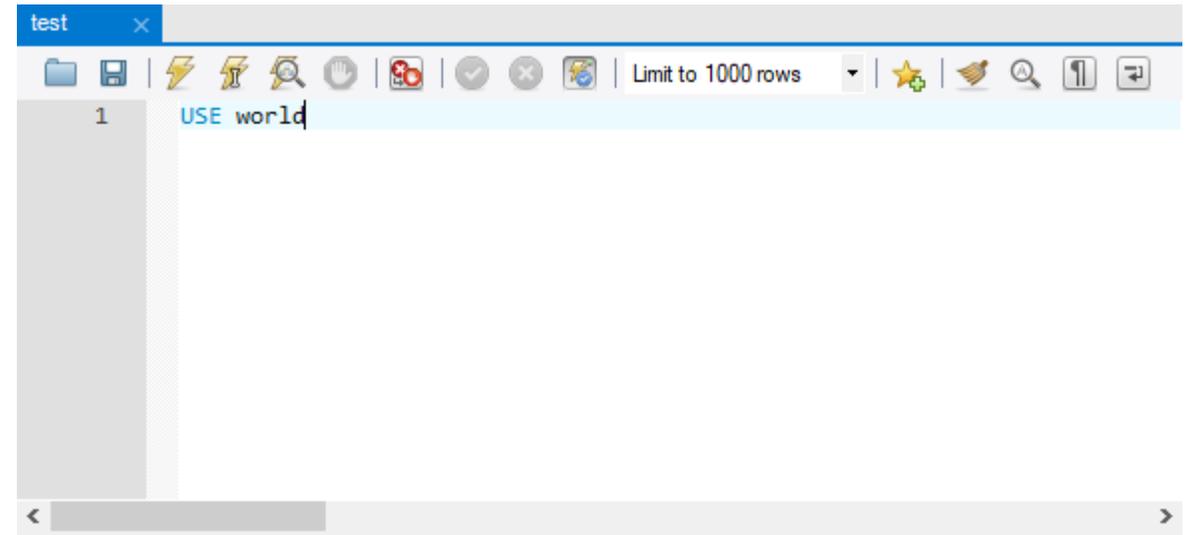
The interface includes a toolbar with various icons, a 'Limit to 1000 rows' dropdown, and a 'Result Grid' button. The bottom status bar shows 'Result 6 x' and 'Read Only'.

USE

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

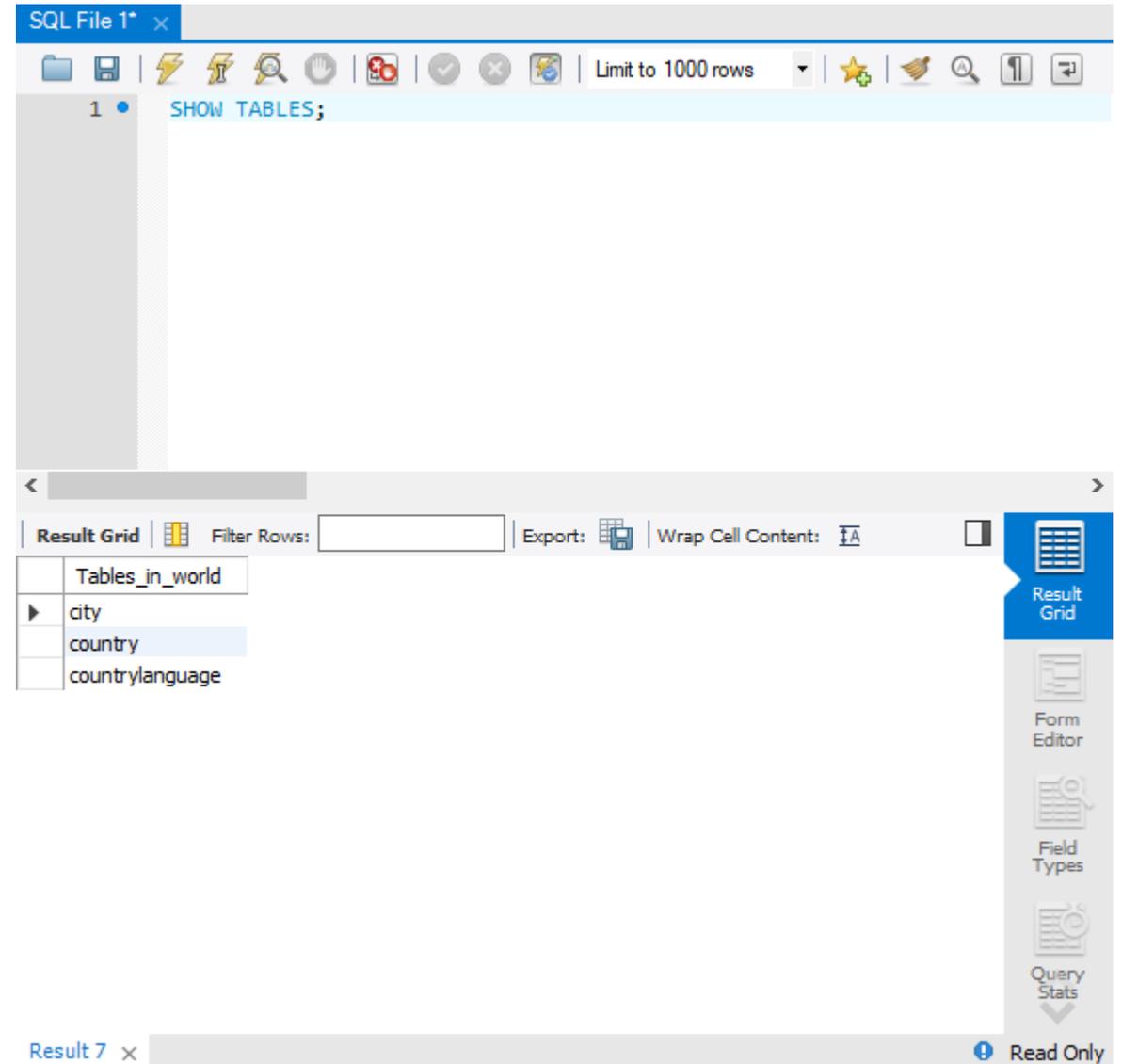
```
USE database_name
```

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



SHOW TABLE

- 데이터베이스 world의 테이블 이름 보기



The screenshot shows a SQL client window titled "SQL File 1*" with a toolbar and a text area containing the command "SHOW TABLES;". Below the text area is a "Result Grid" section with a "Filter Rows:" input field, an "Export:" button, and a "Wrap Cell Content:" checkbox. The result grid displays a list of tables:

Tables_in_world
▶ city
country
countrylanguage

At the bottom of the window, there is a "Result 7" tab and a "Read Only" indicator.

SHOW TABLE STATUS

- 데이터베이스 world의 테이블 정보 조회

The screenshot shows a SQL client window with the following details:

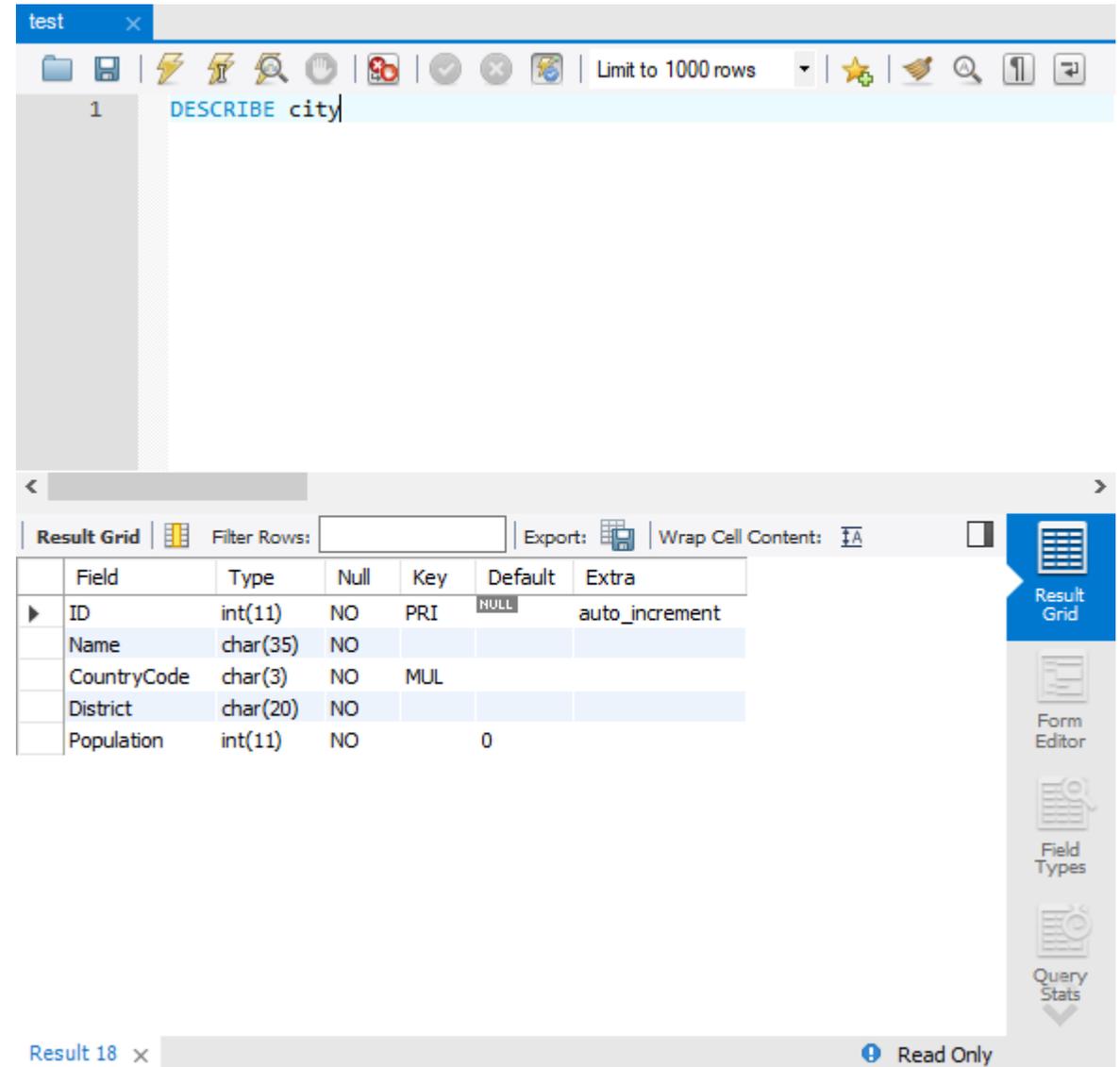
- SQL File 1* x
- Limit to 1000 rows
- Query: SHOW TABLE STATUS;
- Result Grid: Filter Rows: [] | Export: [] | Wrap Cell Content: []
- Table Data:

Name	Engine	Version	Row_format	Rows	Avg_row_length	Data_length	...
city	InnoDB	10	Dynamic	4188	97	409600	0
country	InnoDB	10	Dynamic	239	411	98304	0
countrylanguage	InnoDB	10	Dynamic	984	99	98304	0

Result 8 x Read Only

DESCRIBE (DESC)

- city 테이블에 무슨 열이 있는지 확인
 - DESCRIBE city;
 - DESC city;



The screenshot shows a MySQL IDE window titled 'test'. The command editor contains the text 'DESCRIBE city|'. Below the editor, the 'Result Grid' is displayed, showing the structure of the 'city' table. The grid has columns for Field, Type, Null, Key, Default, and Extra. The rows are: ID (int(11), NO, PRI, NULL, auto_increment), Name (char(35), NO), CountryCode (char(3), NO, MUL), District (char(20), NO), and Population (int(11), NO, 0). The interface includes a toolbar with icons for file operations, a 'Limit to 1000 rows' dropdown, and a sidebar with buttons for 'Result Grid', 'Form Editor', 'Field Types', and 'Query Stats'. The status bar at the bottom indicates 'Result 18' and 'Read Only'.

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	

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 a database client window titled 'test'. The SQL editor contains the query: `SELECT * FROM city`. Below the editor, the 'Result Grid' displays the following data:

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	Nijmegen	NLD	Gelderland	152463

The interface includes a toolbar with icons for file operations, a 'Limit to 1000 rows' dropdown, and a 'Result Grid' toolbar with options for 'Filter Rows', 'Edit', and 'Export/Import'. A vertical sidebar on the right contains icons for 'Result Grid', 'Form Editor', 'Field Types', and 'Query Stats'. At the bottom, there are 'Apply' and 'Revert' buttons.

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 a database client interface with a query editor and a result grid. The query editor contains the following SQL statement:

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

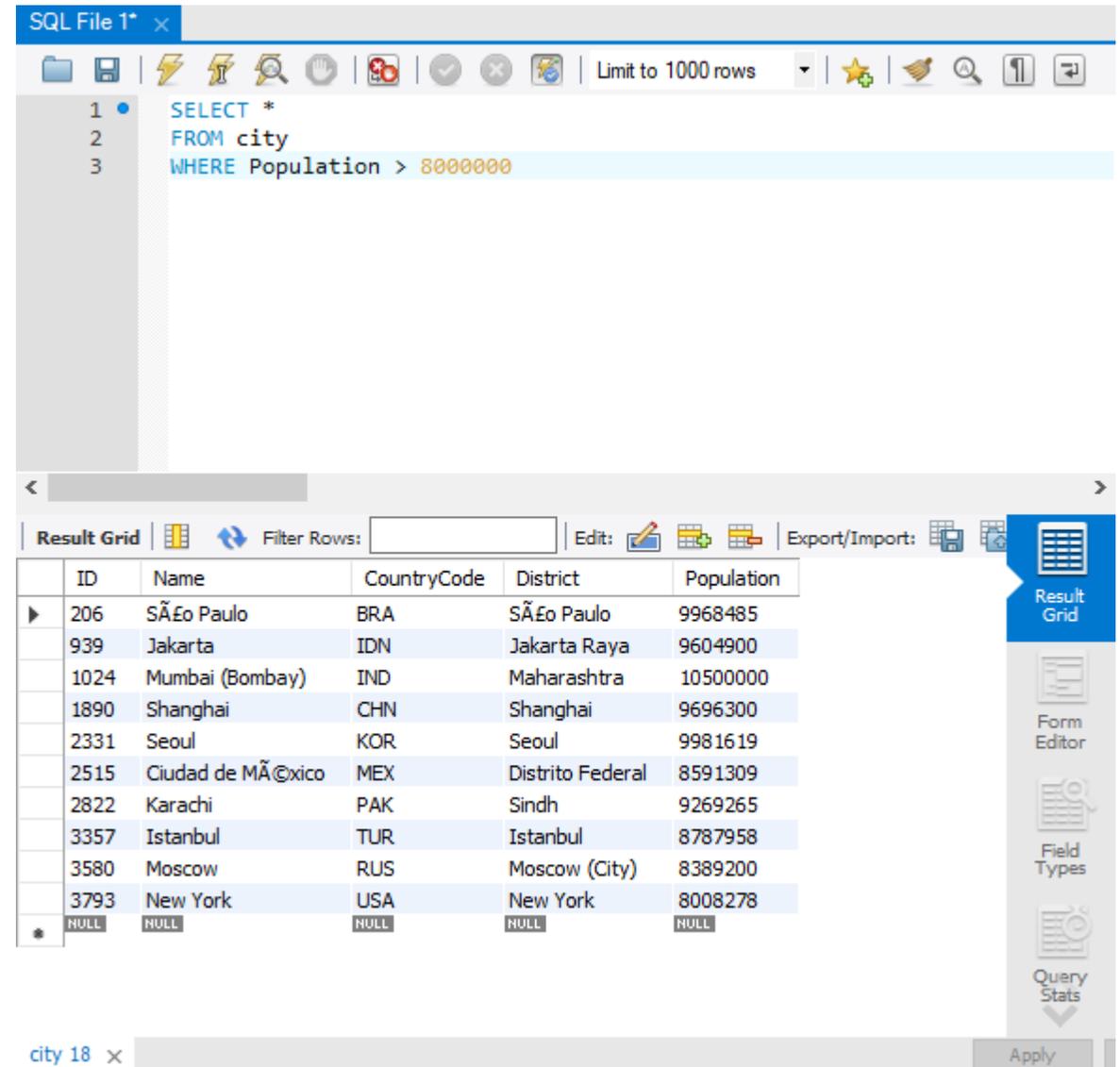
The result grid displays the following data:

Name	District
Kabul	Kabol
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
Eindhoven	Overijssel

SELECT FROM WHERE

■ 기본적인 WHERE절

- 조회하는 결과에 특정한 조건으로 원하는 데이터만 보고 싶을 때 사용
- SELECT 필드이름 FROM 테이블이름 WHERE 조건식;
- 조건이 없을 경우 테이블의 크기가 클수록 찾는 시간과 노력이 증가



The screenshot shows a SQL IDE window titled "SQL File 1*" with a query editor and a result grid. The query in the editor is:

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

The result grid displays the following data:

ID	Name	CountryCode	District	Population
206	SÃ£o Paulo	BRA	SÃ£o 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 MÃ©xico	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

The interface includes a toolbar with icons for file operations, a "Limit to 1000 rows" dropdown, and a "Result Grid" tab. The bottom status bar shows "city 18 x" and an "Apply" button.

SELECT FROM WHERE

- 관계 연산자의 사용
 - OR 연산자
 - AND 연산자
 - 조건 연산자(=, <, >, <=, >=, <>, != 등)
 - 관계 연산자(NOT, AND, OR 등)
 - 연산자의 조합으로 데이터를 효율적으로 추출
- MySQL 함수 및 연산자:
<https://dev.mysql.com/doc/refman/8.0/en/functions.html>

The screenshot shows a SQL IDE window titled "SQL File 1*" with a toolbar and a "Limit to 1000 rows" dropdown. The query editor contains the following SQL code:

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

Below the query editor is a "Result Grid" showing the results of the query. The grid has columns for ID, Name, CountryCode, District, and Population. The results are 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

The IDE also features a sidebar with icons for "Result Grid", "Form Editor", "Field Types", and "Query Stats". At the bottom, there is a "city 25 x" tab and an "Apply" button.

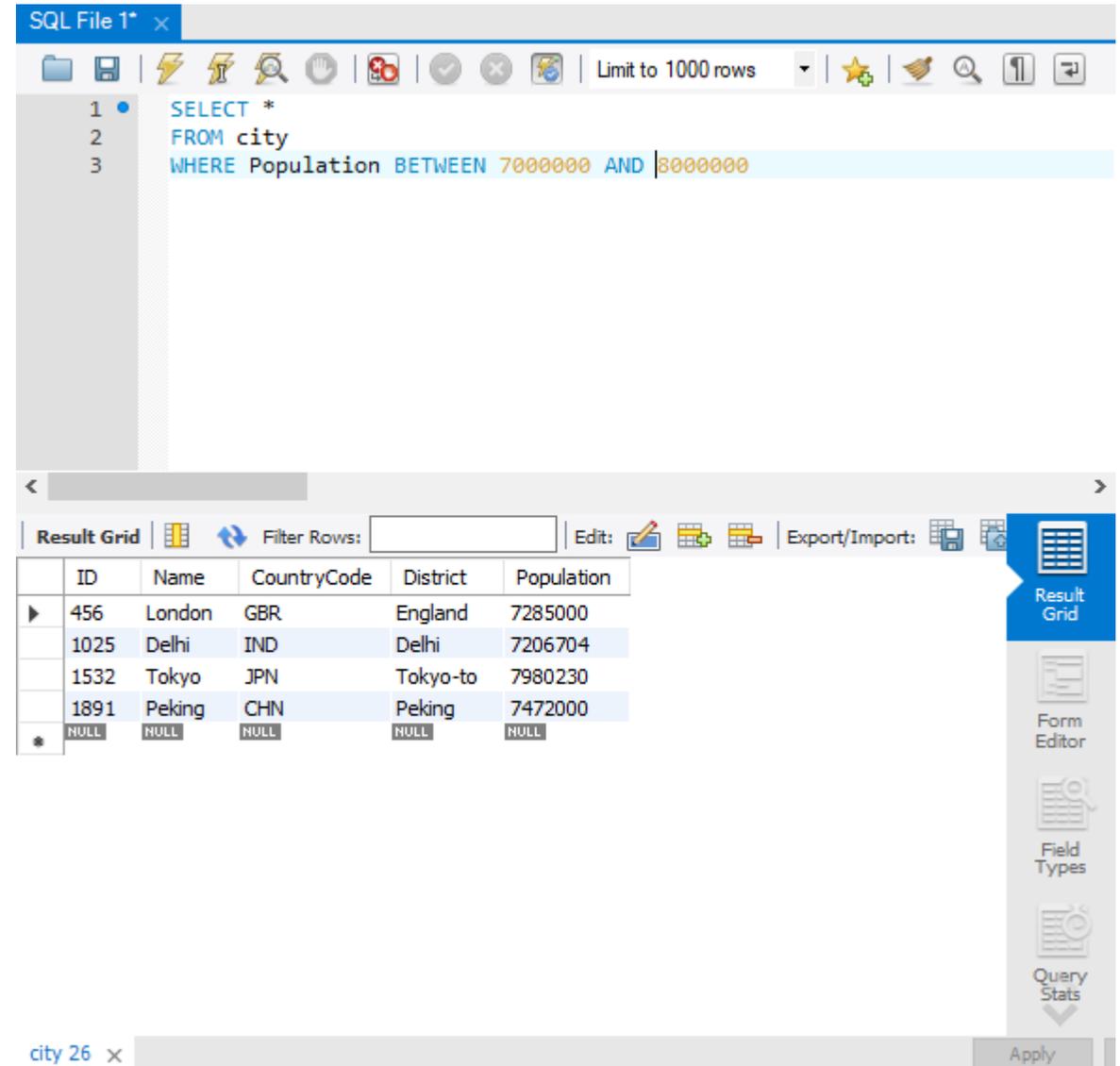
한국에 있는 도시들 보기

미국에 있는 도시들 보기

한국에 있는 도시들 중에
인구 수가 1,000,000 이상인 도시 보기

BETWEEN

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



The screenshot shows a SQL IDE window titled "SQL File 1* x". The query editor contains the following SQL code:

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

The results are displayed in a "Result Grid" below the query editor. The grid has the following columns: ID, Name, CountryCode, District, and Population. The data rows are:

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

The IDE interface includes a toolbar with various icons, a "Limit to 1000 rows" dropdown, and a "Filter Rows:" input field. The bottom status bar shows "city 26 x" and an "Apply" button.

IN

- 이산적인 Discrete 값의 조건에서는 IN() 사용 가능

The screenshot shows a SQL IDE interface. The top pane displays the following SQL query:

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

The bottom pane shows the results in a table format:

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

The interface includes various toolbars and a sidebar with options like 'Result Grid', 'Form Editor', 'Field Types', and 'Query Stats'. The status bar at the bottom indicates 'city 33 x' and 'Apply'.

한국, 미국, 일본의 도시들 보기

LIKE

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

The screenshot shows a SQL IDE window titled "SQL File 1*" with a toolbar and a query editor. The query is:

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

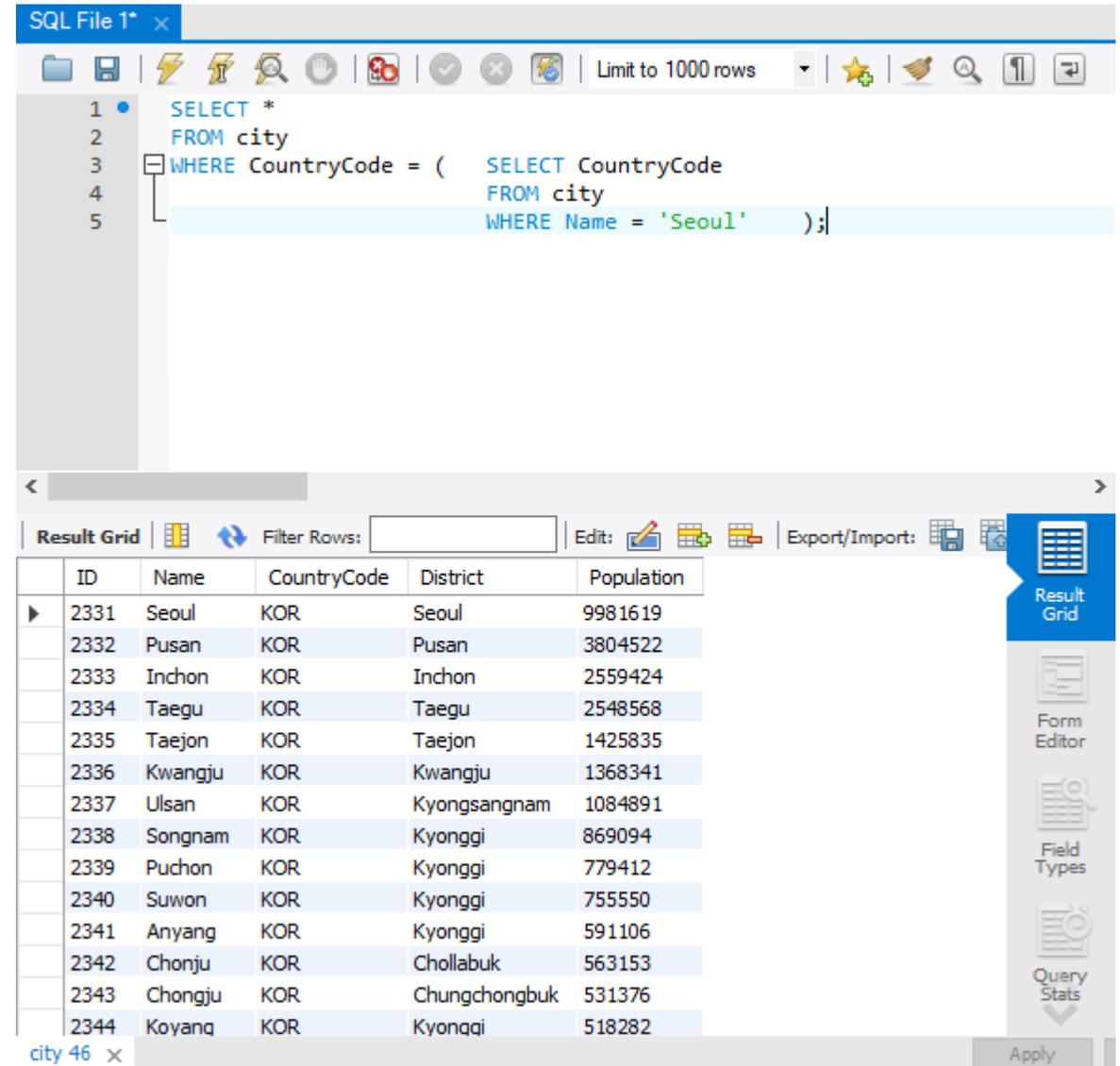
Below the query editor is a "Result Grid" showing the results of the query. The grid has columns for ID, Name, CountryCode, District, and Population. The results are as follows:

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	Kyonggi	518282

The interface also includes a "Filter Rows" field, "Edit", "Export/Import", and "Apply" buttons. The status bar at the bottom shows "city 44 x" and "Apply".

Sub Query

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



The screenshot shows a SQL IDE window titled "SQL File 1* x". The SQL editor contains the following query:

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

Below the editor is the "Result Grid" showing the output of the query. The grid has columns for ID, Name, CountryCode, District, and Population. The first row is highlighted, showing the result for Seoul.

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	Kyonggi	518282

The status bar at the bottom indicates "city 46 x" and "Apply".

ANY

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

The screenshot shows a SQL IDE window titled "SQL File 1*" with a query editor and a result grid. The query is:

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

The result grid displays 14 rows of data:

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	Nijmegen	NLD	Gelderland	152463

ALL

- 서브쿼리의 여러 개의 결과를 모두 만족시켜야 함

The screenshot shows a SQL IDE window titled "SQL File 1* x". The query editor contains the following SQL code:

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

The result grid below the query shows the following data:

ID	Name	CountryCode	District	Population
206	SÃ£o Paulo	BRA	SÃ£o 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 MÃ©xico	MEX	Distrito Federal	8591309
2822	Karachi	PAK	Sindh	9269265
3357	Istanbul	TUR	Istanbul	8787958
3580	Moscow	RUS	Moscow (City)	8389200
NULL	NULL	NULL	NULL	NULL

The interface includes a toolbar with icons for file operations, a "Limit to 1000 rows" dropdown, and a "Result Grid" section with a "Filter Rows" input field. The bottom status bar shows "city 97 x" and "Apply Revert" buttons.

ORDER BY

- 결과가 출력되는 순서를 조절하는 구문
- 기본적으로 오름차순 ASCENDING 정렬
- 내림차순 DESCENDING 으로 정렬
 - 열 이름 뒤에 DESC 적어줄 것
- ASC(오름차순)는 default이므로 생략 가능

The screenshot shows a SQL IDE window titled "SQL File 1* x". The query editor contains the following SQL code:

```
1 SELECT *
2 FROM city
3 ORDER BY Population DESC
```

The results are displayed in a "Result Grid" table with the following columns: ID, Name, CountryCode, District, and Population. The data is sorted in descending order of population.

ID	Name	CountryCode	District	Population
1024	Mumbai (Bombay)	IND	Maharashtra	10500000
2331	Seoul	KOR	Seoul	9981619
206	SÃ£o Paulo	BRA	SÃ£o 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

The interface includes a toolbar with icons for file operations, a "Limit to 1000 rows" dropdown, and a "Result Grid" section with a "Filter Rows:" input field. The bottom of the window shows "city 101 x" and "Apply" and "Revert" buttons.

ORDER BY

- ORDER BY 구문을 혼합해 사용하는 구문도 가능

The screenshot shows a SQL IDE window titled "SQL File 1* x". The query editor contains the following SQL code:

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

The results are displayed in a "Result Grid" table with the following data:

ID	Name	CountryCode	District	Population
▶ 129	Oranjestad	ABW	Å-	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	Å-	961
62	The Valley	AIA	Å-	595
34	Tirana	ALB	Tirana	270000
55	Andorra la Vella	AND	Andorra l...	21189

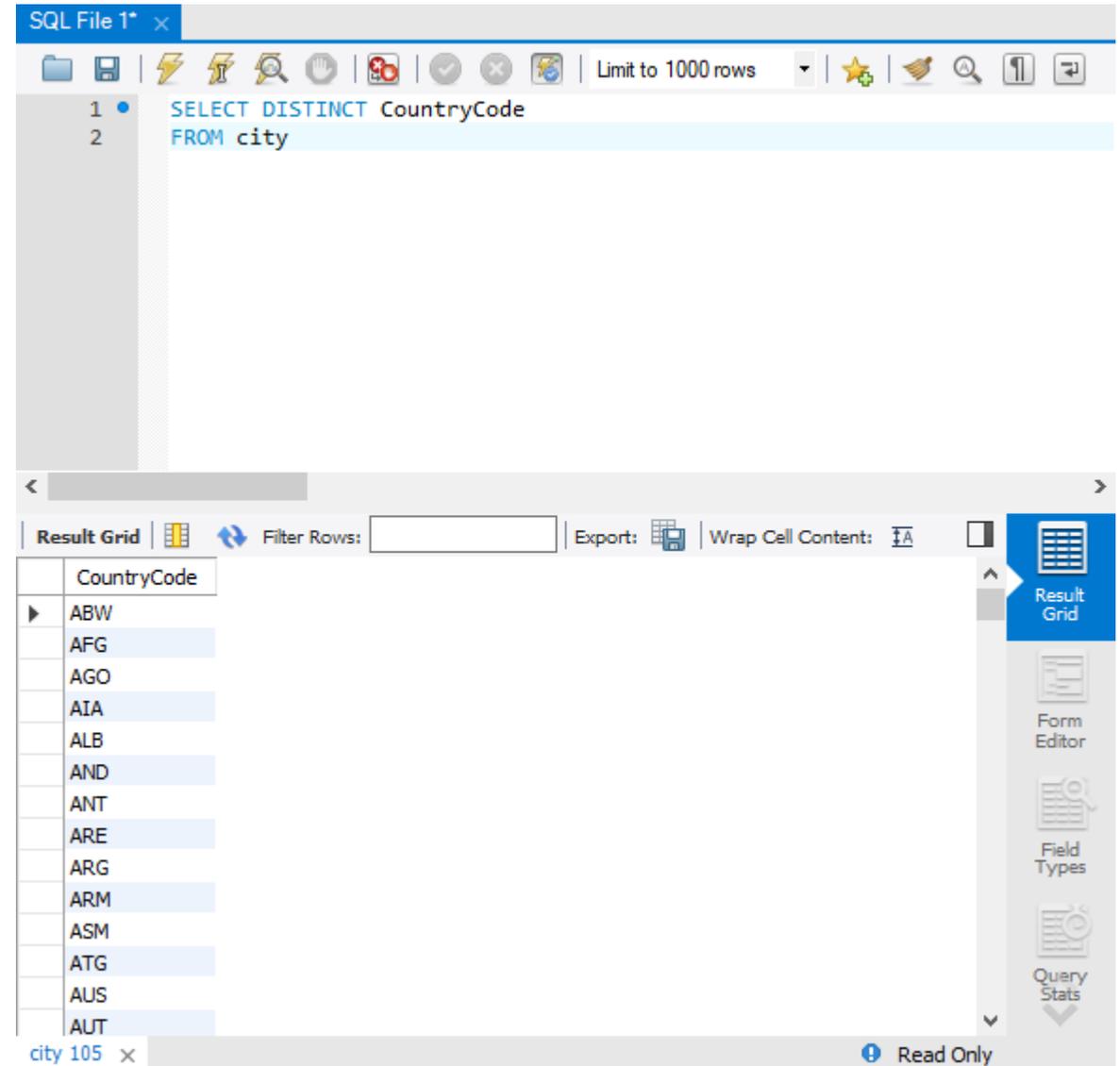
The interface includes a toolbar with icons for file operations, a "Limit to 1000 rows" dropdown, and a "Result Grid" section with a "Filter Rows:" input field. On the right side, there are buttons for "Result Grid", "Form Editor", "Field Types", and "Query Stats". At the bottom, there are "Apply" and "Revert" buttons.

인구수로 내림차순하여 한국에 있는 도시 보기

**국가 면적 크기로 내림차순하여 나라 보기
(country table)**

DISTINCT

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



The screenshot shows a SQL IDE interface. The top pane displays the following SQL query:

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

The bottom pane shows the 'Result Grid' with the following data:

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

The interface also includes a toolbar with various icons, a 'Limit to 1000 rows' dropdown, and a 'Read Only' indicator at the bottom right.

LIMIT

- 출력 개수를 제한
- 상위의 N개만 출력하는 'LIMIT N' 구문
- 서버의 처리량을 많이 사용해 서버의 전반적인 성능을 나쁘게 하는 악성 쿼리문 개선할 때 사용

The screenshot shows a SQL IDE window titled 'SQL File 1*'. The query editor contains the following SQL code:

```
1 SELECT *
2 FROM city
3 ORDER BY Population DESC
4 LIMIT 10
```

The 'Result Grid' below the query shows the top 10 cities by population, ordered in descending order. The columns are ID, Name, CountryCode, District, and Population.

ID	Name	CountryCode	District	Population
1024	Mumbai (Bombay)	IND	Maharashtra	10500000
2331	Seoul	KOR	Seoul	9981619
206	SÃ£o Paulo	BRA	SÃ£o 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
NULL	NULL	NULL	NULL	NULL

The IDE interface includes a toolbar with icons for file operations, a 'Limit to 1000 rows' dropdown, and a 'Result Grid' toolbar with options for 'Filter Rows', 'Edit', 'Export/Import', 'Form Editor', 'Field Types', and 'Query Stats'. The status bar at the bottom shows 'city 107 x', 'Apply', and 'Revert' buttons.

GROUP BY

- 그룹으로 묶어주는 역할
- 집계 함수 Aggregate Function를 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 a SQL IDE window titled 'SQL File 1*'. The query editor contains the following SQL code:

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

Below the query editor, the 'Result Grid' is displayed, showing the results of the query. The grid has two columns: 'CountryCode' and 'MAX(Population)'. 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 IDE interface includes various toolbars and a sidebar on the right with options like 'Result Grid', 'Form Editor', 'Field Types', and 'Query Stats'. The status bar at the bottom indicates 'Result 110 x' and 'Read Only'.

도시는 몇개인가?

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 a SQL IDE window titled "SQL File 1* x". The query editor contains the following SQL code:

```
1 SELECT CountryCode, MAX(Population) 'MAX Population'  
2 FROM city  
3 GROUP BY CountryCode  
4 HAVING MAX(Population) > 8000000
```

The results pane below shows a table with the following data:

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

The IDE interface includes a toolbar with icons for file operations, a "Limit to 1000 rows" dropdown, and a "Result Grid" section with options for "Filter Rows", "Export", and "Wrap Cell Content". The status bar at the bottom indicates "Result 117 x" and "Read Only".

ROLLUP

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

The screenshot shows a database query editor with the following SQL query:

```
1 SELECT CountryCode, Name, SUM(Population)
2 FROM city
3 GROUP BY CountryCode, Name WITH ROLLUP
4
```

The result grid displays the following data:

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

JOIN

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

The screenshot shows a SQL IDE window titled "SQL File 4*" with a query editor and a result grid. The query is:

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

The result grid displays 13 rows of data. The first four rows represent cities in Afghanistan, and the remaining nine rows represent cities in the Netherlands. The columns are: ID, Name, CountryCode, District, Population, Code, Name, and Co.

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

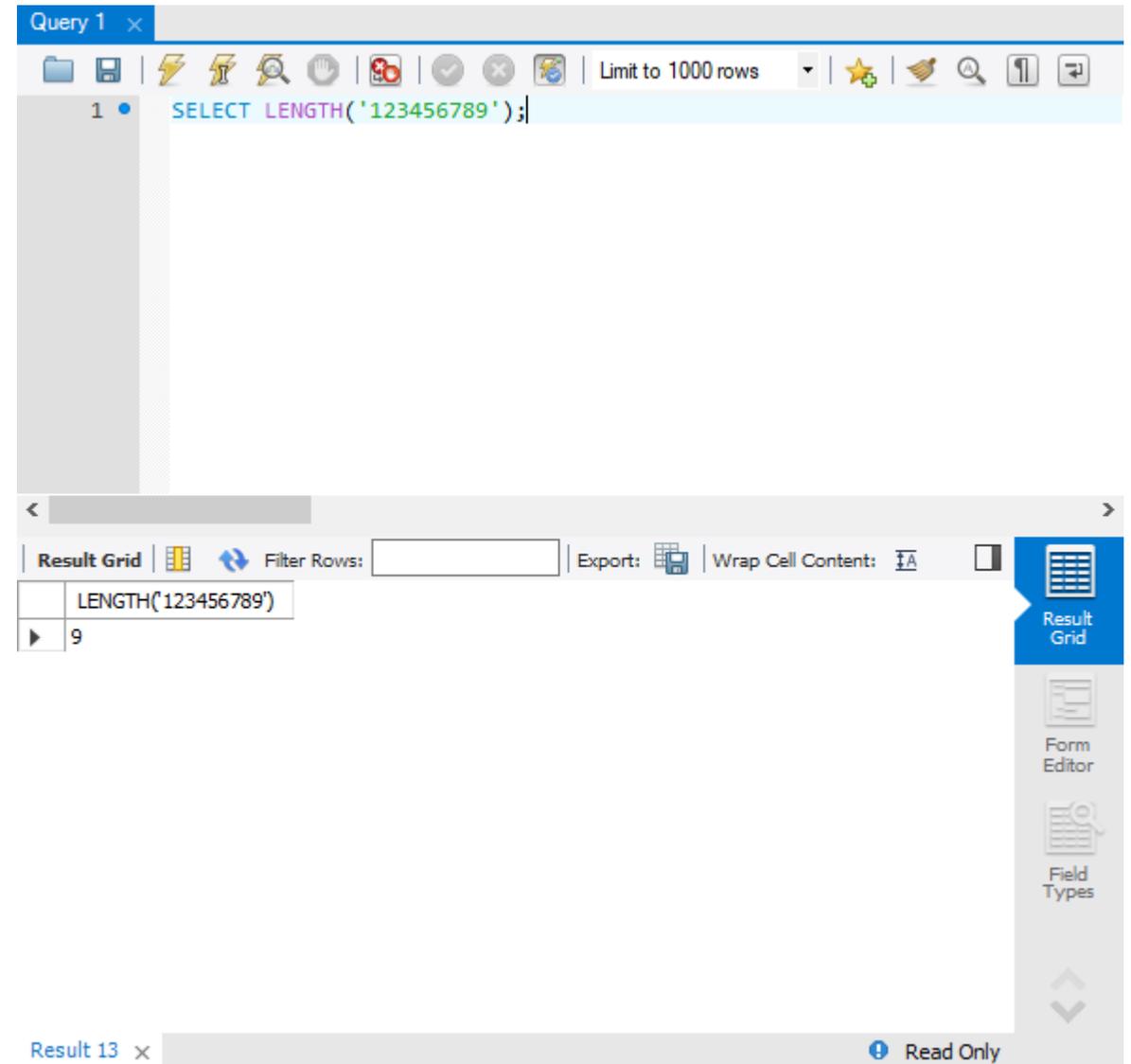
city, country, countrylanguage 테이블 3개를 JOIN 하기

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 a SQL query editor interface. The query editor contains the following SQL statement:

```
1 • SELECT LENGTH('123456789');
```

The result grid below the query editor displays the output of the query:

LENGTH('123456789')
9

The interface includes a toolbar with various icons for file operations, execution, and search. The result grid has a 'Filter Rows' input field and an 'Export' button. The right sidebar contains buttons for 'Result Grid', 'Form Editor', and 'Field Types'. The bottom status bar shows 'Result 13 x' and 'Read Only'.

CONCAT()

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

The screenshot shows a SQL query editor with the following query:

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

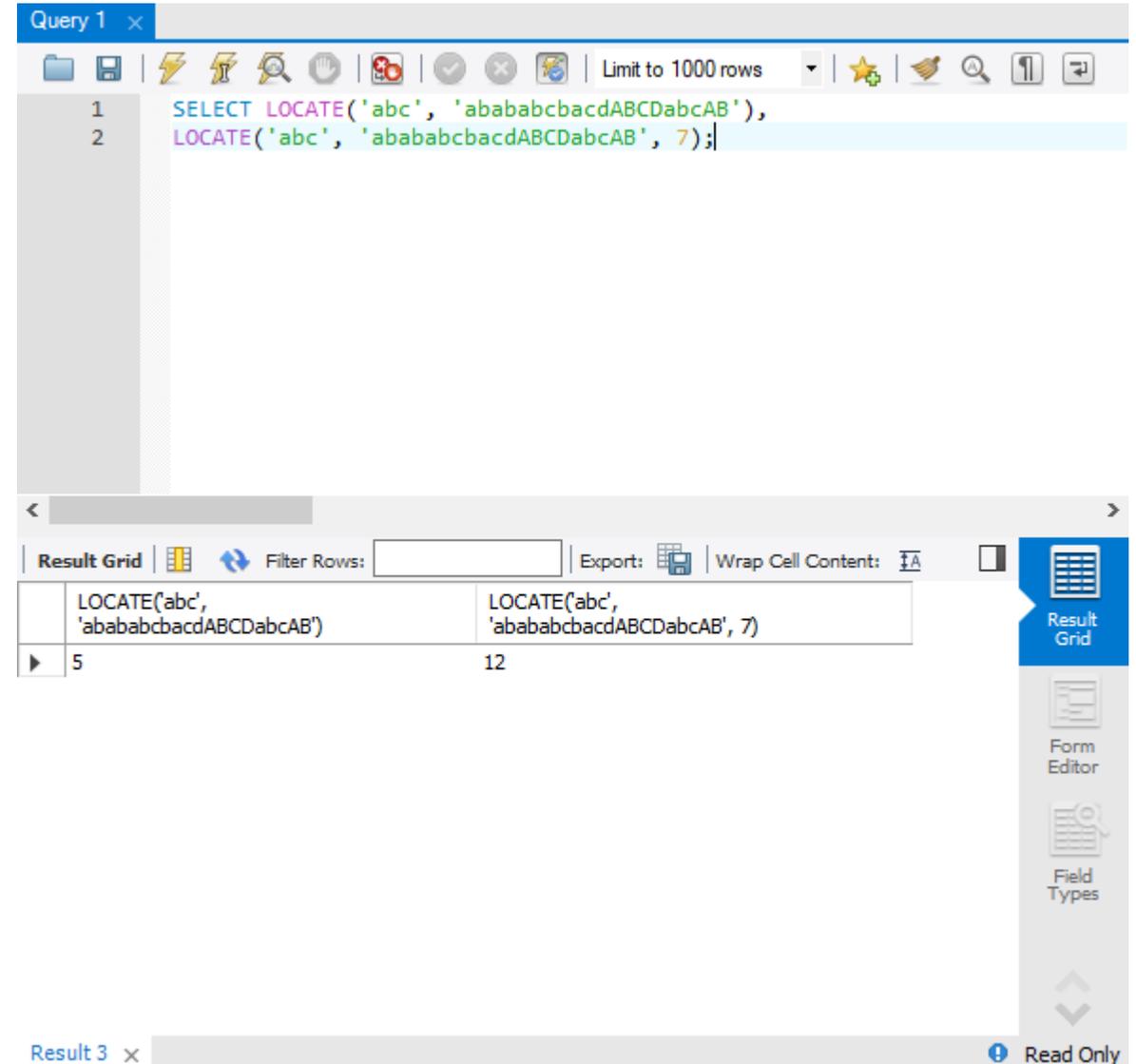
The result grid below the query shows the output of the query:

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

The interface includes a toolbar with icons for file operations, execution, and search. The result grid has a 'Filter Rows' field and an 'Export' button. The bottom right corner shows 'Result 15 x' and 'Read Only'.

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



The screenshot shows a MySQL query editor window titled "Query 1". The query text is:

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

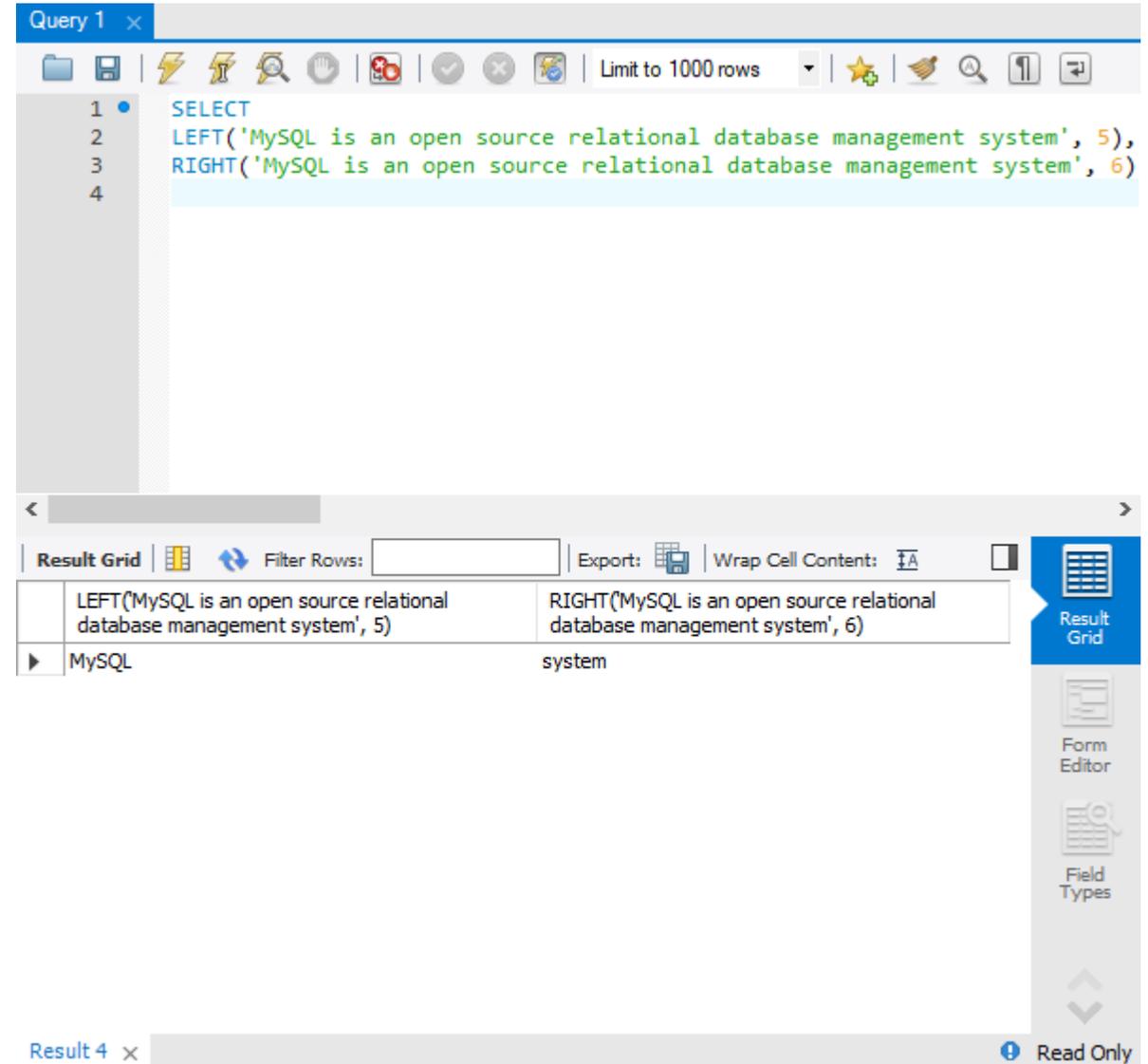
The results are displayed in a table with two columns:

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

The interface includes a toolbar with various icons, a "Limit to 1000 rows" dropdown, and a "Result Grid" button on the right side. The bottom status bar shows "Result 3 x" and "Read Only".

LEFT(), RIGHT()

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



The screenshot shows a MySQL query editor window titled "Query 1". The query text is as follows:

```
1 SELECT
2 LEFT('MySQL is an open source relational database management system', 5),
3 RIGHT('MySQL is an open source relational database management system', 6)
4
```

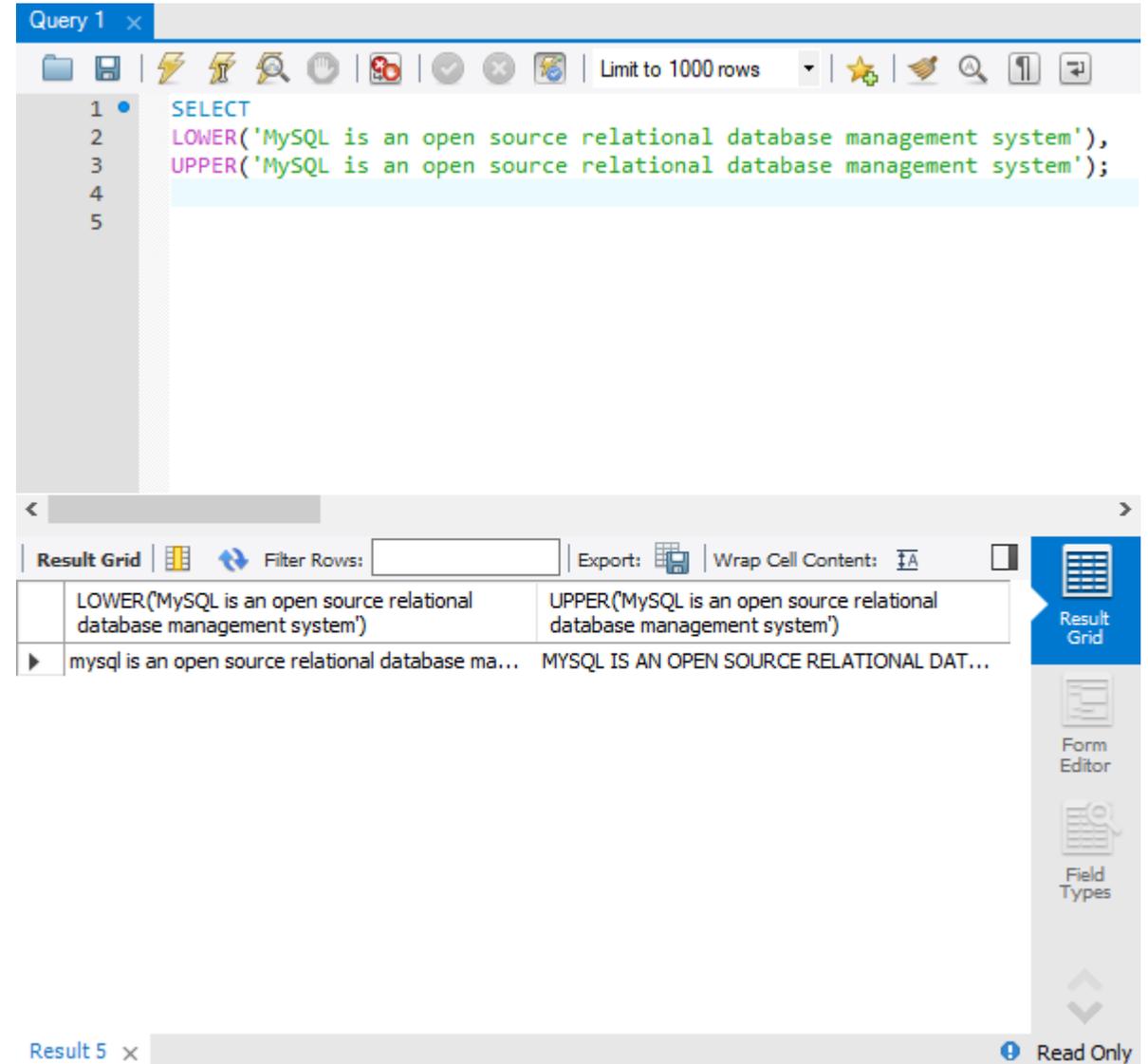
Below the query editor, the "Result Grid" is displayed, showing the output of the query:

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 interface includes various toolbars and a sidebar with options like "Result Grid", "Form Editor", and "Field Types". The status bar at the bottom indicates "Result 4 x" and "Read Only".

LOWER(), UPPER()

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



The screenshot shows a MySQL query editor window titled "Query 1". The query text is as follows:

```
1 SELECT
2 LOWER('MySQL is an open source relational database management system'),
3 UPPER('MySQL is an open source relational database management system');
4
5
```

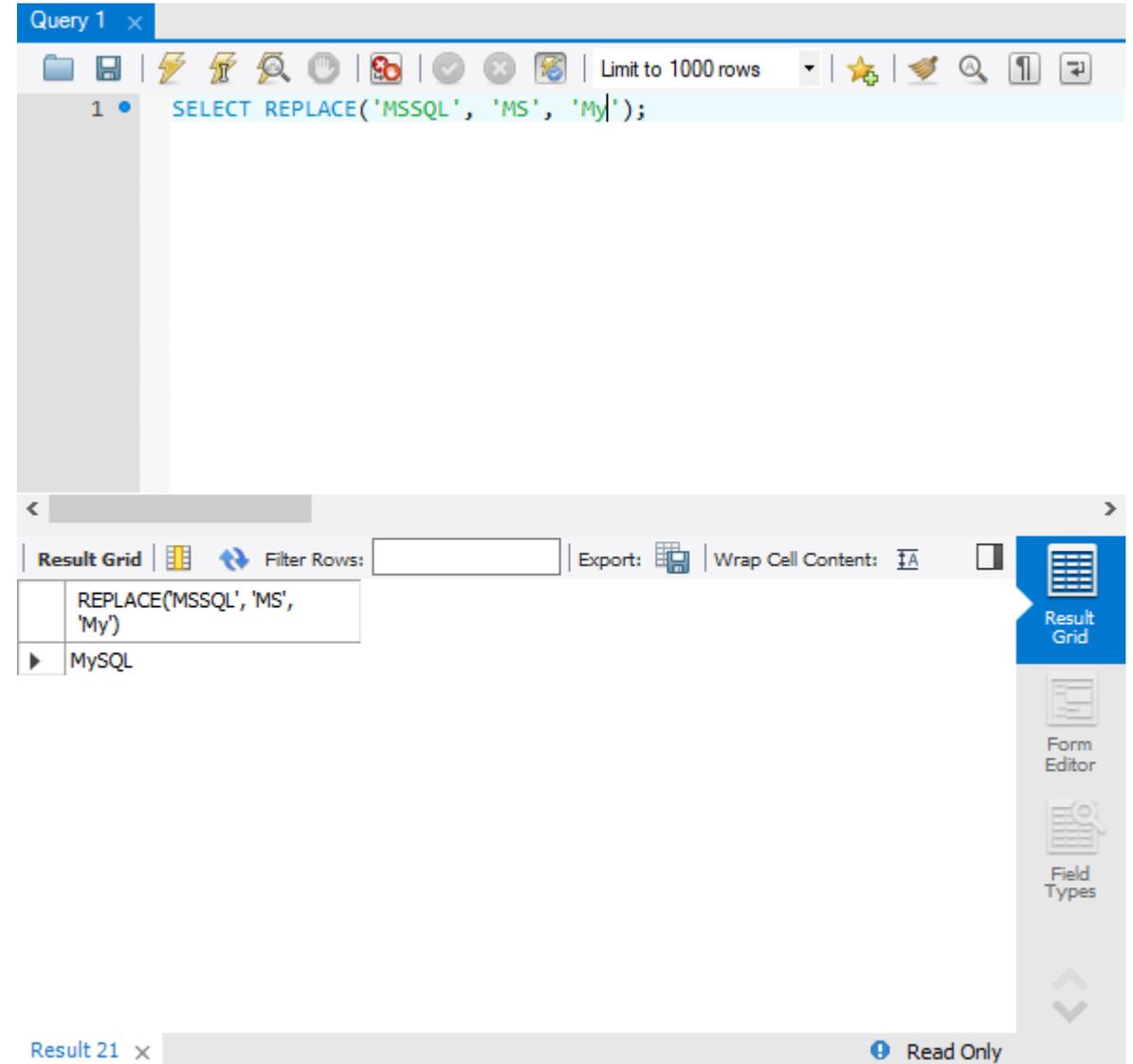
Below the query editor, the "Result Grid" is displayed. It shows the output of the query:

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 interface includes various toolbars and a sidebar on the right with options like "Result Grid", "Form Editor", and "Field Types". The status bar at the bottom indicates "Result 5 x" and "Read Only".

REPLACE()

- 문자열에서 특정 문자열을 대체 문자열로 교체



The screenshot shows a MySQL query editor window titled "Query 1". The query text is `SELECT REPLACE('MSSQL', 'MS', 'My');`. Below the query editor, the "Result Grid" is visible, showing the output of the query. The result is a single row with the value `REPLACE('MSSQL', 'MS', 'My')` in the first column and `MySQL` in the second column. The interface includes various toolbars and a sidebar with options like "Result Grid", "Form Editor", and "Field Types".

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

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
- 제거할 문자를 명시하지 않으면, 자동으로 공백을 제거

The screenshot shows a MySQL query editor window titled "Query 1". The query text is as follows:

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

Below the query editor, the "Result Grid" is displayed, showing the results of the three queries. The grid has three columns corresponding to the queries and one row of results.

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

The interface includes various toolbars and a sidebar with options like "Result Grid", "Form Editor", and "Field Types". The status bar at the bottom indicates "Result 8 x" and "Read Only".

FORMAT()

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

The screenshot shows a MySQL query editor window titled "Query 1". The query entered is `SELECT FORMAT(123456789.123456, 3);`. The result grid below shows the output of the query:

FORMAT(123456789.123456, 3)
123,456,789.123

The interface includes a toolbar with various icons, a "Limit to 1000 rows" dropdown, and a "Result Grid" button on the right side. The status bar at the bottom indicates "Result 23" and "Read Only".

FLOOR(), CEIL(), ROUND()

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

The screenshot shows a SQL query editor window titled "Query 1". The query text is as follows:

```
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);
```

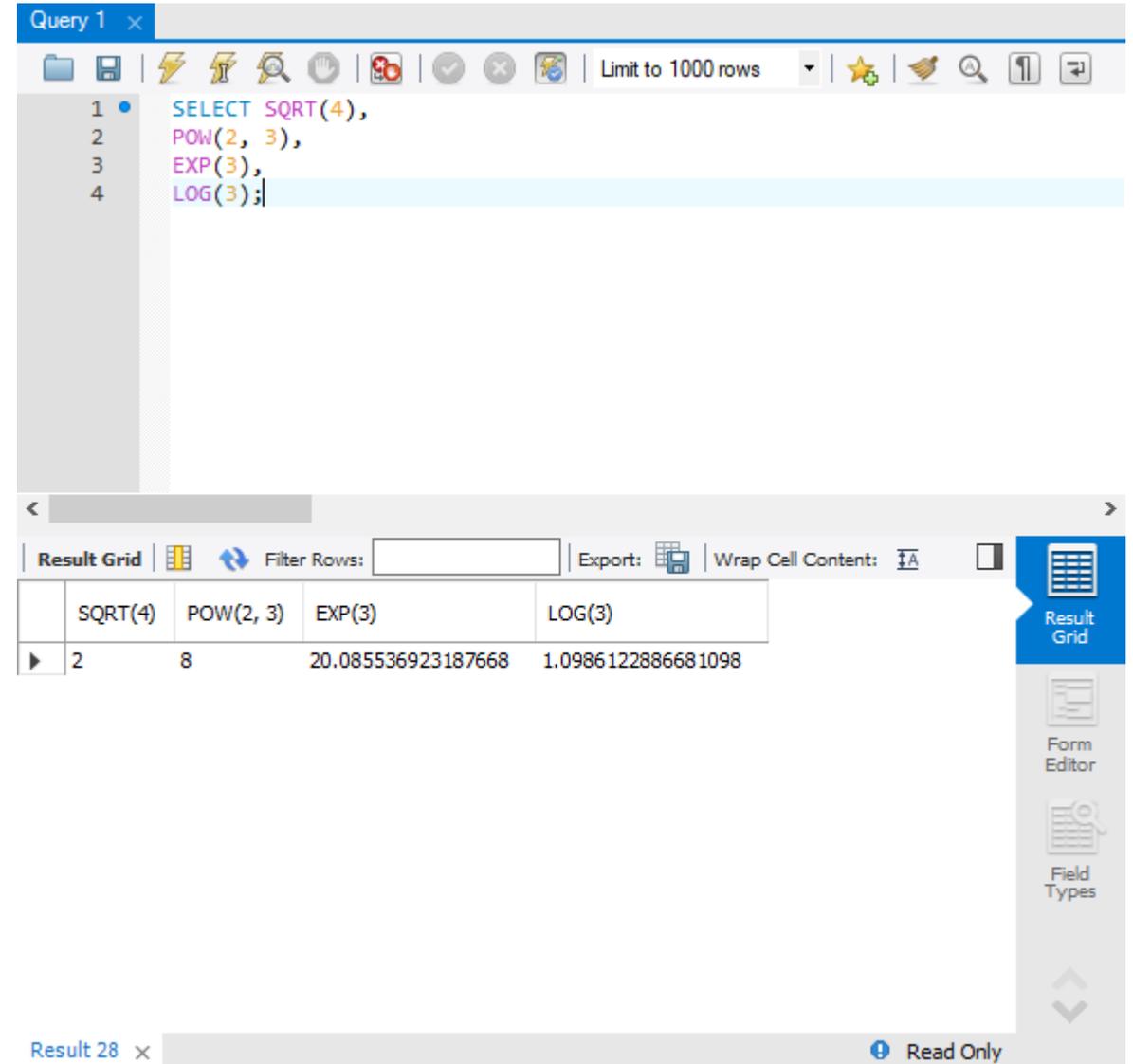
Below the query editor is a "Result Grid" showing the output of the query. The grid has 7 columns corresponding to the functions in the query. The first row of data shows the results for the first six functions.

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

The interface also includes a "Filter Rows" field, an "Export" button, and a "Wrap Cell Content" checkbox. On the right side, there are buttons for "Result Grid", "Form Editor", and "Field Types". At the bottom, there is a "Read Only" indicator.

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 a MySQL query editor window titled "Query 1". The query text is:

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

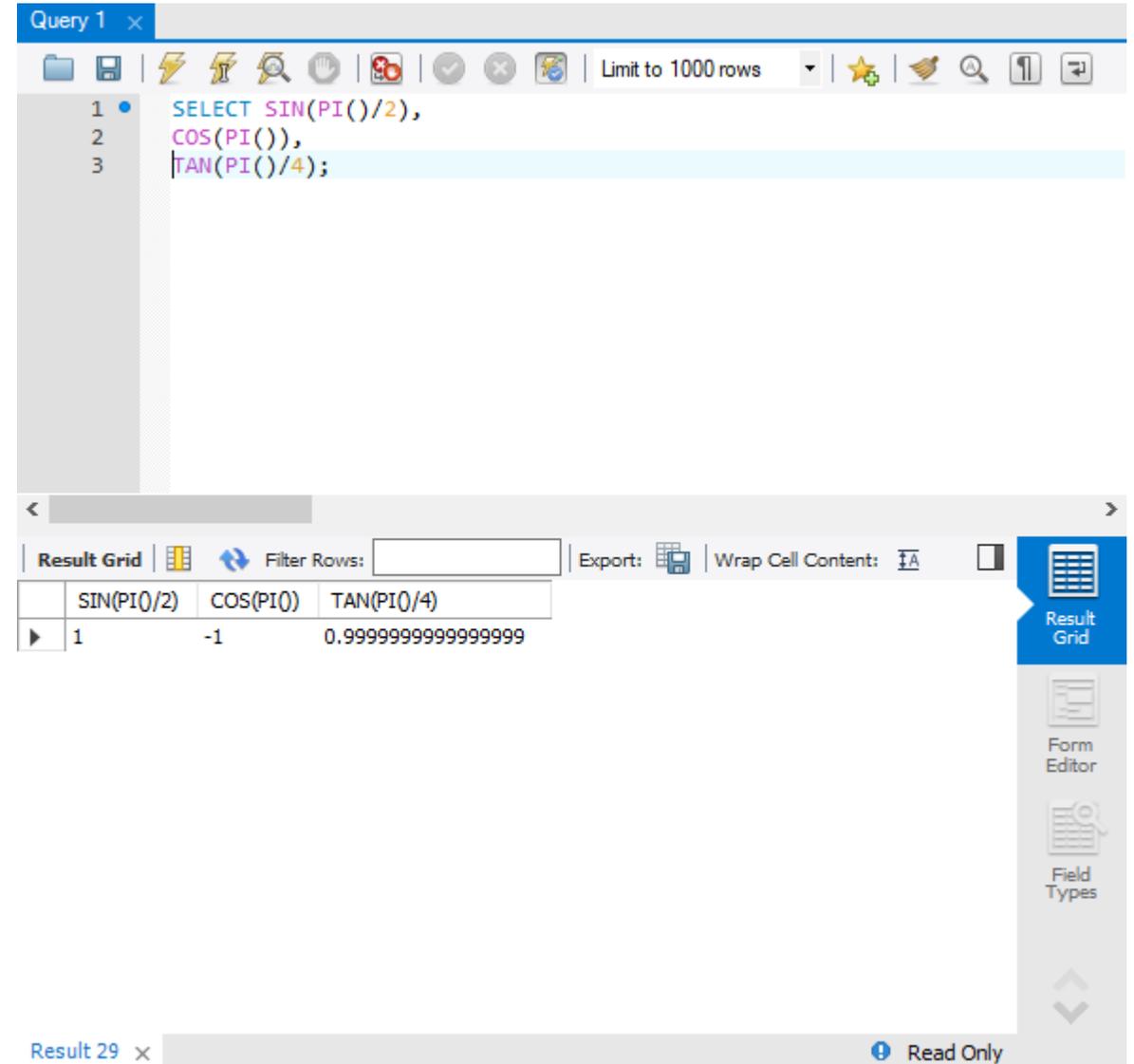
Below the query editor, the "Result Grid" is displayed with the following data:

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

The interface includes various toolbars and a sidebar on the right with options like "Result Grid", "Form Editor", and "Field Types". The status bar at the bottom indicates "Result 28" and "Read Only".

SIN(), COS(), TAN()

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



The screenshot shows a SQL query editor window titled "Query 1". The query text is:

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

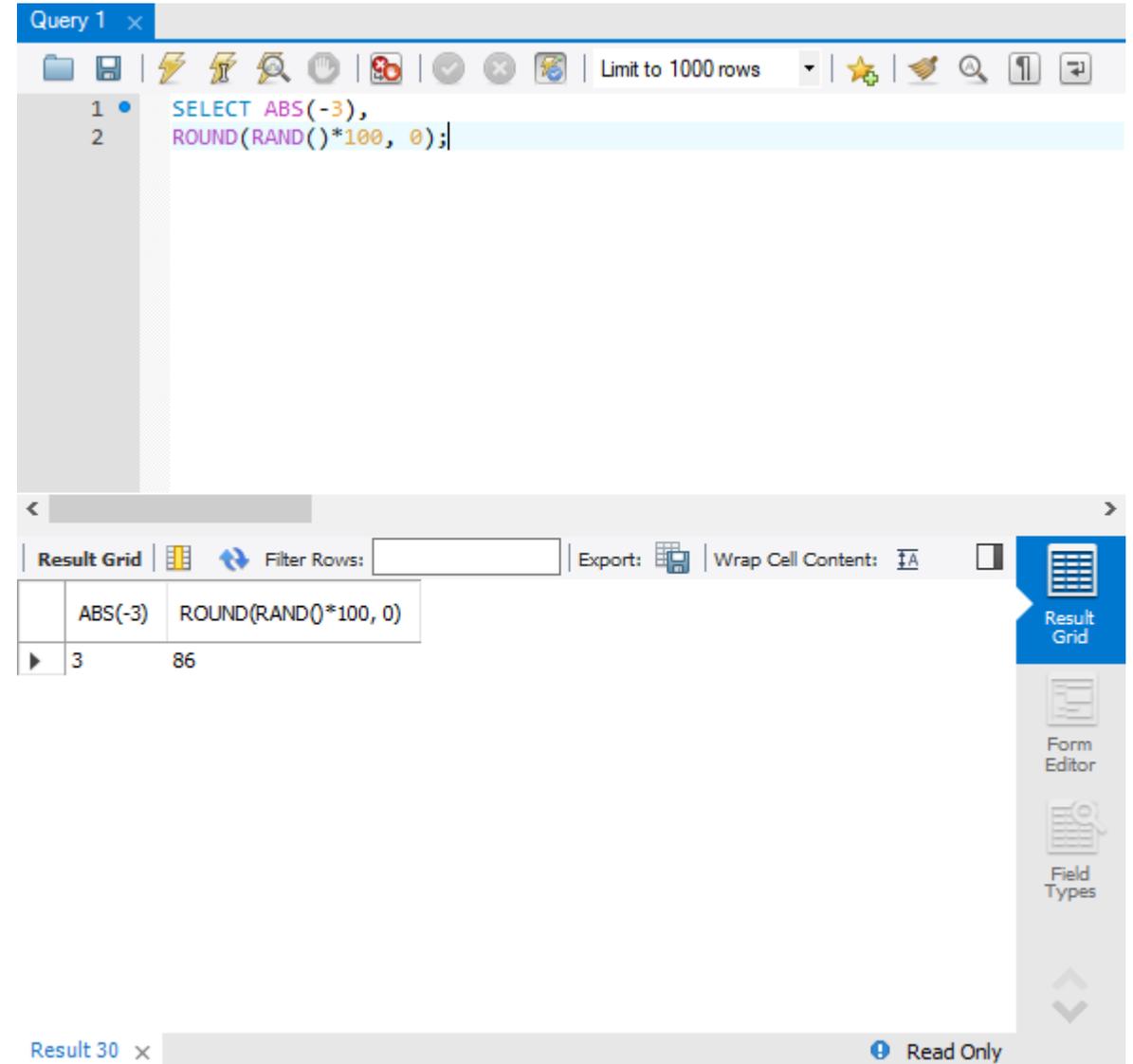
The results are displayed in a table with the following columns: SIN(PI()/2), COS(PI()), and TAN(PI()/4). The first row shows the values 1, -1, and 0.9999999999999999.

	SIN(PI()/2)	COS(PI())	TAN(PI()/4)
1	1	-1	0.9999999999999999

The interface includes a toolbar with various icons, a "Limit to 1000 rows" dropdown, and a "Result Grid" button. The bottom right corner shows "Result 29" and "Read Only" status.

ABS(), RAND()

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



The screenshot shows a MySQL query editor window titled "Query 1". The query text is:

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

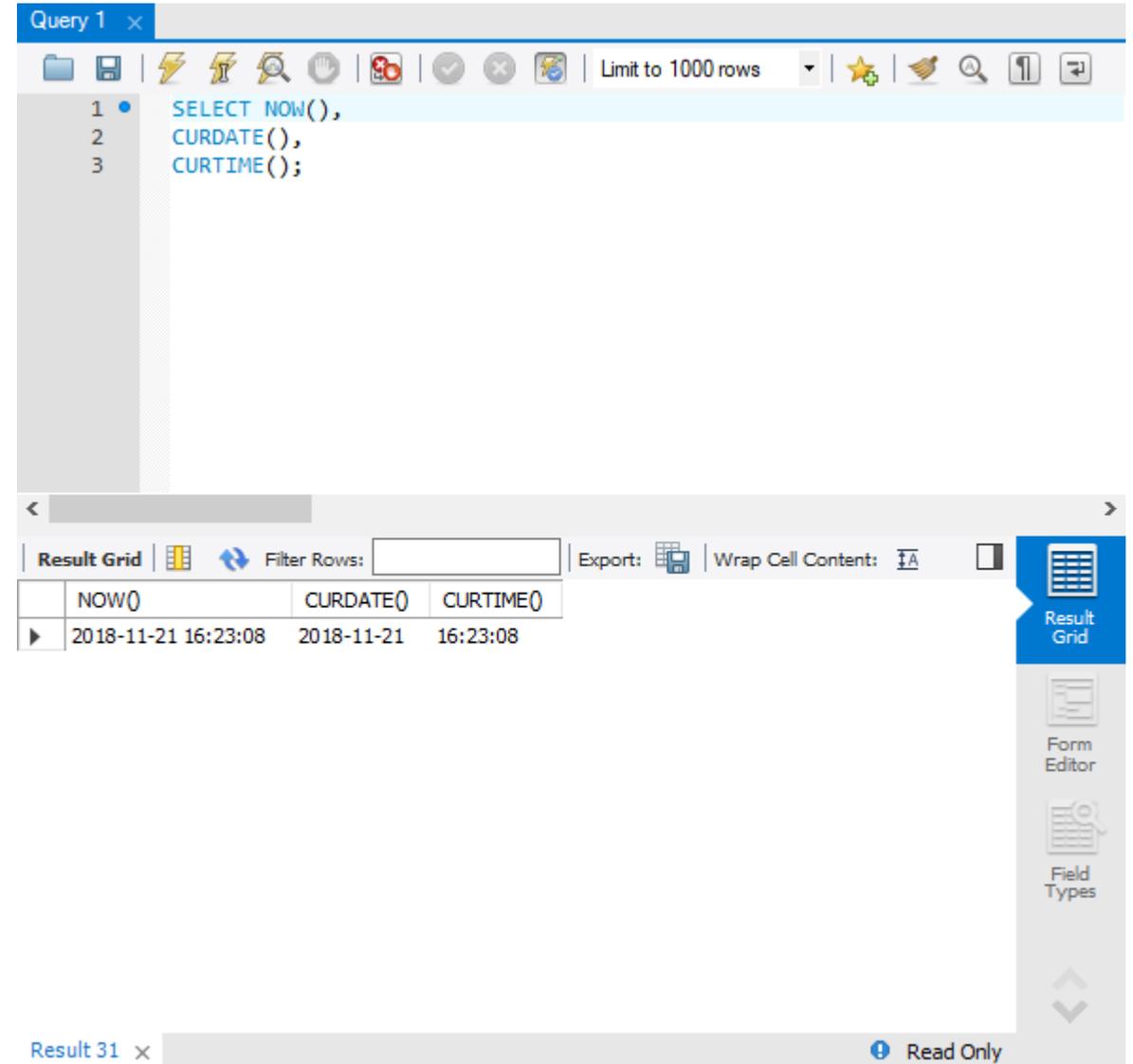
The result grid below the query shows the following output:

	ABS(-3)	ROUND(RAND()*100, 0)
▶ 3		86

The interface includes a toolbar with various icons, a "Limit to 1000 rows" dropdown, and a "Result Grid" button. The bottom right corner shows "Result 30 x" 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



```
Query 1 x
SELECT NOW(),
CURDATE(),
CURTIME();
```

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

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 a MySQL query editor window titled 'Query 1'. The query text is as follows:

```
1 SELECT
2 NOW(),
3 DATE(NOW()),
4 MONTH(NOW()),
5 DAY(NOW()),
6 HOUR(NOW()),
7 MINUTE(NOW()),
8 SECOND(NOW())
9
```

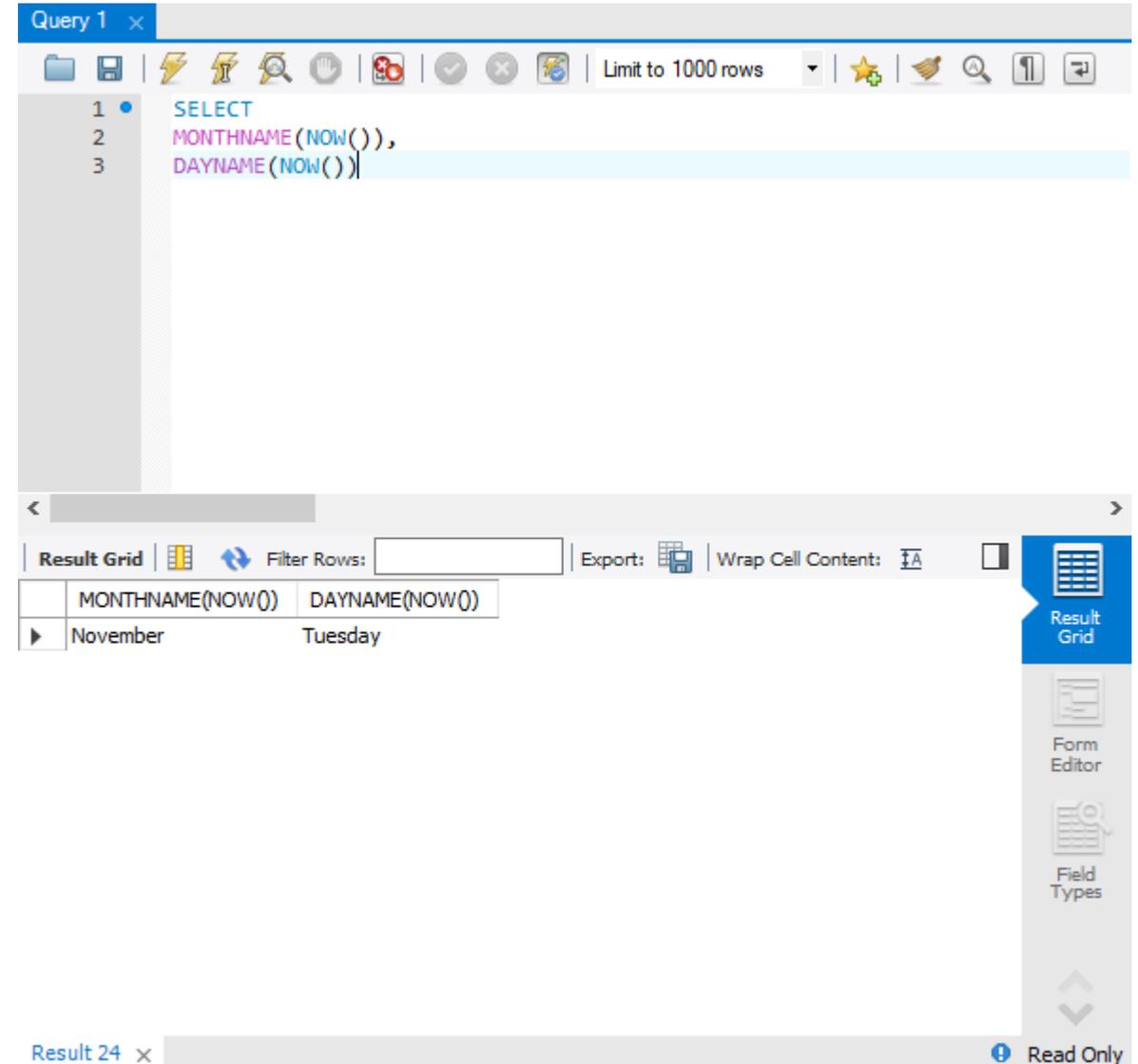
Below the query editor, the 'Result Grid' is displayed with the following data:

NOW()	DATE(NOW())	MONTH(NOW())	DAY(NOW())	HOUR(NOW())	MINUTE
2018-11-27 01:18:06	2018-11-27	11	27	1	18

The interface includes a toolbar with various icons, a 'Limit to 1000 rows' dropdown, and a 'Read Only' indicator at the bottom right.

MONTHNAME(), DAYNAME()

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



The screenshot shows a MySQL query editor window titled "Query 1". The query text is:

```
1 SELECT
2 MONTHNAME(NOW()),
3 DAYNAME(NOW())
```

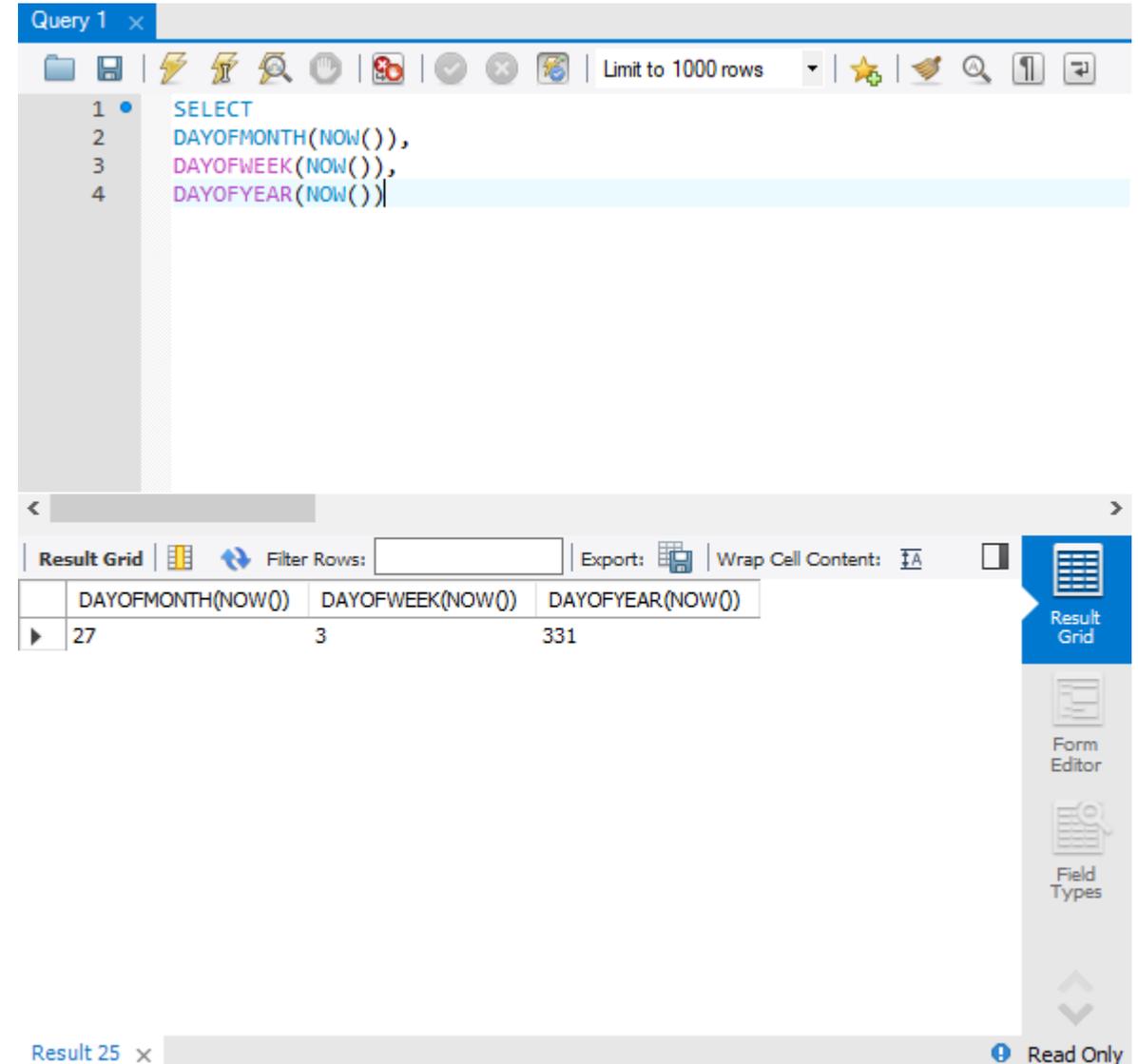
The result is displayed in a table with two columns: MONTHNAME(NOW()) and DAYNAME(NOW()). The result row shows "November" and "Tuesday".

MONTHNAME(NOW())	DAYNAME(NOW())
November	Tuesday

The interface includes a toolbar with various icons, a "Limit to 1000 rows" dropdown, and a "Result Grid" button. The bottom status bar shows "Result 24 x" and "Read Only".

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 a MySQL query editor window titled "Query 1". The query text is:

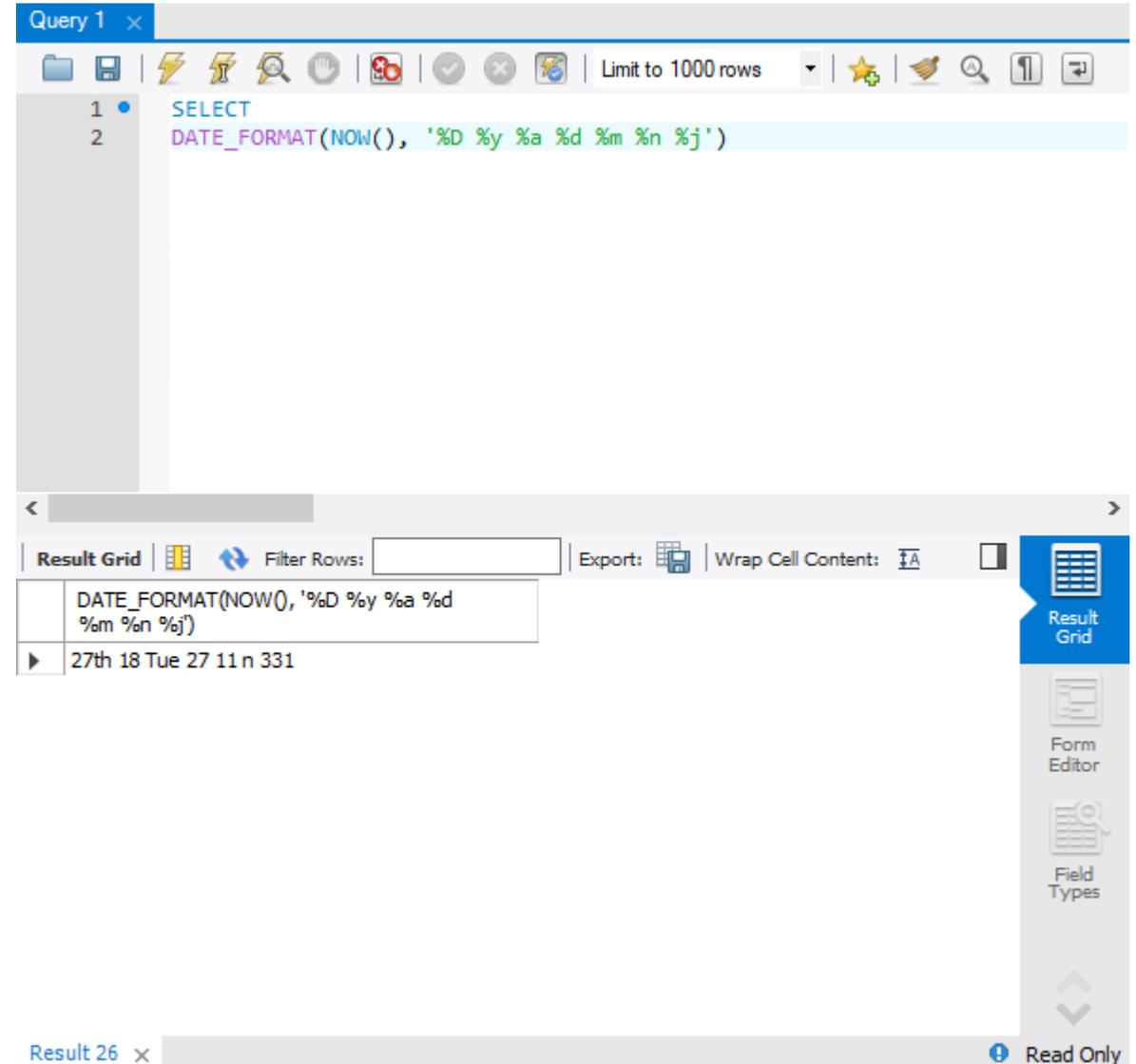
```
SELECT  
DAYOFMONTH(NOW()),  
DAYOFWEEK(NOW()),  
DAYOFYEAR(NOW())
```

The results are displayed in a table with the following columns: DAYOFMONTH(NOW()), DAYOFWEEK(NOW()), and DAYOFYEAR(NOW()). The first row shows the values 27, 3, and 331 respectively.

DAYOFMONTH(NOW())	DAYOFWEEK(NOW())	DAYOFYEAR(NOW())
27	3	331

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 a MySQL query editor window titled "Query 1". The query text is:

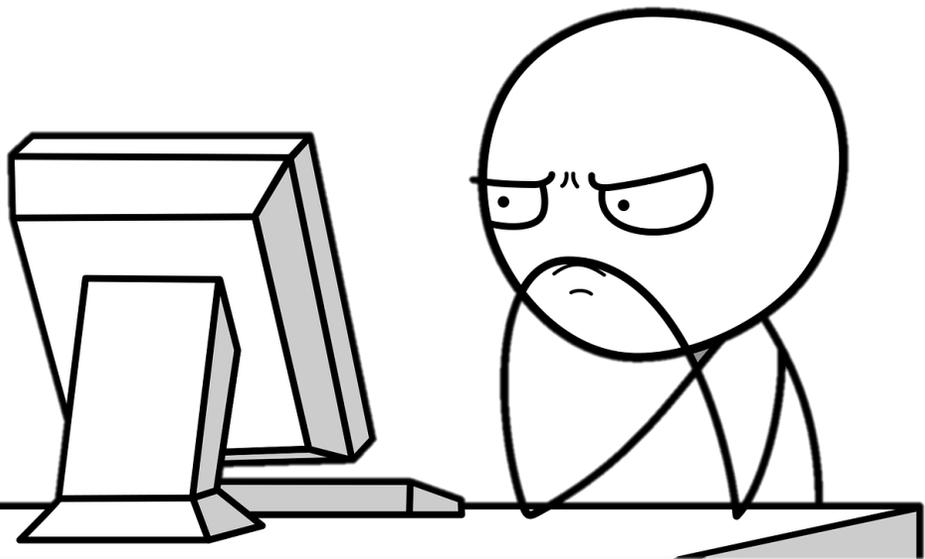
```
1 SELECT
2 DATE_FORMAT(NOW(), '%D %y %a %d %m %n %j')
```

The result grid below the query shows the output of the function:

DATE_FORMAT(NOW(), '%D %y %a %d %m %n %j')
27th 18 Tue 27 11 n 331

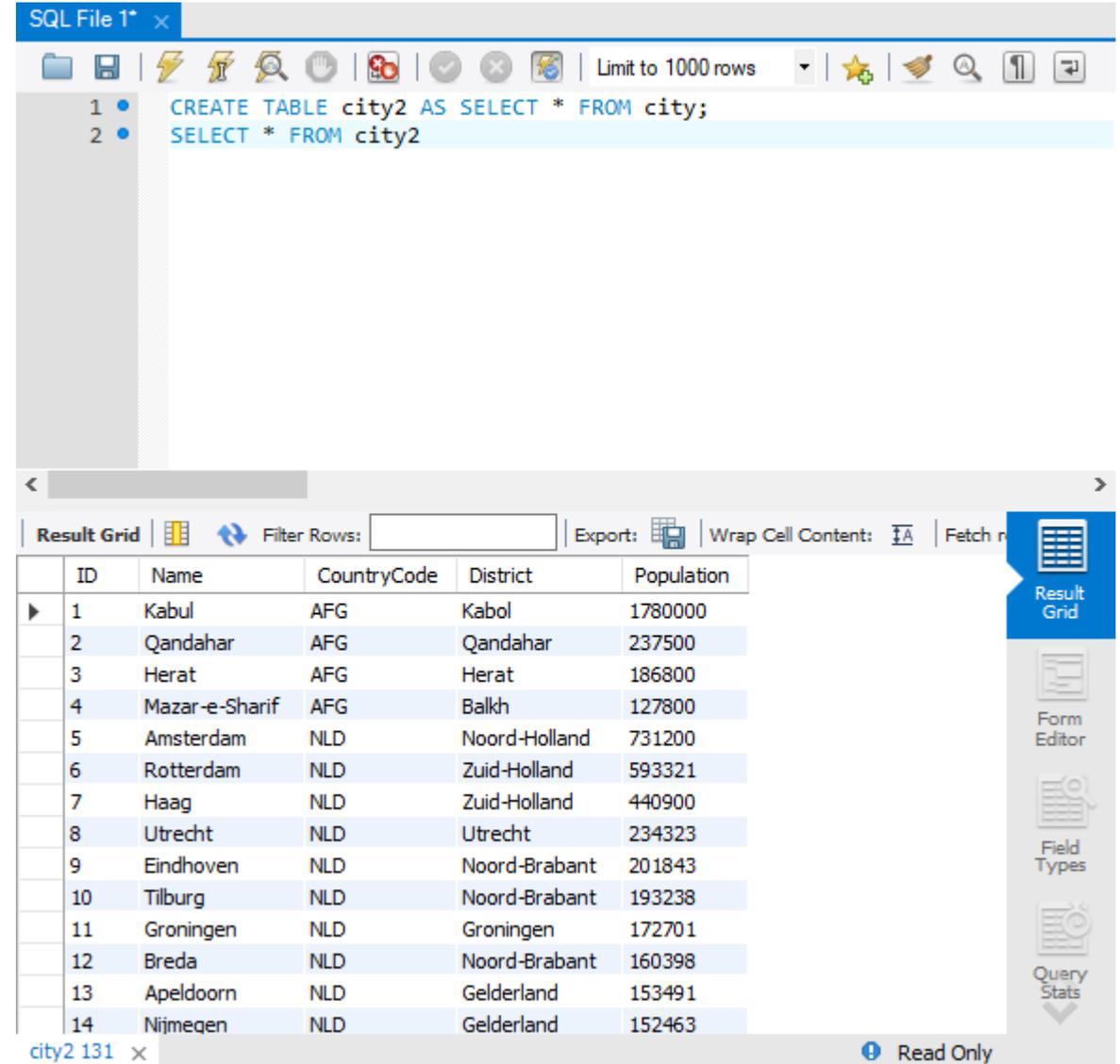
The interface includes a toolbar with various icons, a "Limit to 1000 rows" dropdown, and a "Result Grid" button. The bottom status bar shows "Result 26" and "Read Only".

4. SQL 고급



CREATE TABLE AS SELECT

- city 테이블과 똑같은 city2 테이블 생성



The screenshot shows a SQL IDE interface. The top pane displays the following SQL query:

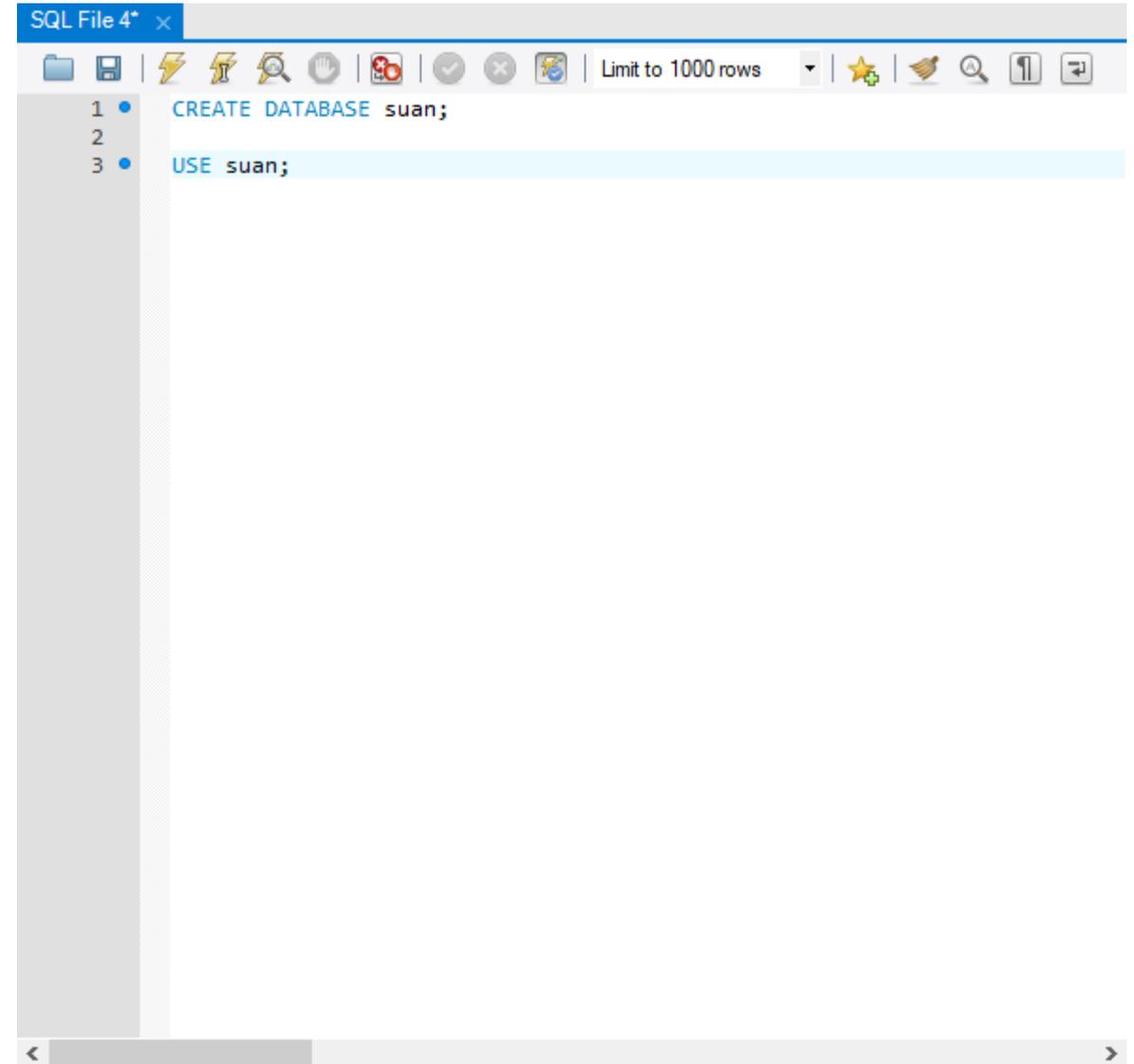
```
1 CREATE TABLE city2 AS SELECT * FROM city;  
2 SELECT * FROM city2
```

The bottom pane shows the result grid with the following data:

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	Nijmegen	NLD	Gelderland	152463

CREATE DATABASE

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



The screenshot shows a SQL editor window titled "SQL File 4* x". The editor contains two lines of SQL code:

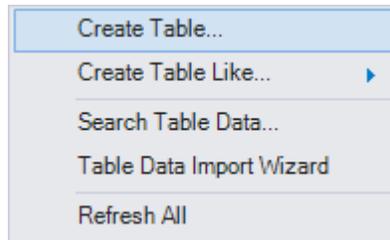
```
1 CREATE DATABASE suan;  
2  
3 USE suan;
```

The second line, "USE suan;", is highlighted in light blue. The editor interface includes a toolbar with various icons and a "Limit to 1000 rows" dropdown menu.

CREATE TABLE (MySQL Workbench)

■ 데이터 타입:

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



The screenshot shows the 'test - Table' dialog box in MySQL Workbench. The 'Table Name' is 'test' and the 'Schema' is 'suan'. The 'Engine' is set to 'InnoDB'. The 'Charset/Collation' is set to 'Default C'. The 'Comments' field is empty. Below the dialog box, there is a table with columns: Column Name, Datatype, PK, NN, UQ, B, UN, ZF, AI, G, and Defa. The table contains four rows: 'id' (INT, PK, NN), 'col1' (INT), 'col2' (FLOAT), and 'col3' (VARCHAR(45)). Below the table, there are fields for 'Column Name', 'Data Type', 'Charset/Collation', and 'Expression'. There are also radio buttons for 'Storage' (Virtual, Stored) and checkboxes for 'Primary Key', 'Not Null', 'Unique', 'Binary', 'Unsigned', 'Zero Fill', 'Auto Increment', and 'Generated'. At the bottom, there are tabs for 'Columns', 'Indexes', 'Foreign Keys', 'Triggers', 'Partitioning', and 'Options', and buttons for 'Apply' and 'Revert'.

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"/>	
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"/>	
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"/>	

CREATE TABLE (MySQL Workbench)

Apply SQL Script to Database

Review SQL Script

Apply SQL Script

Review the SQL Script to be Applied on the Database

Online DDL

Algorithm: Default

Lock Type: Default

```
1 CREATE TABLE `world`.`test` (  
2   `id` INT NOT NULL,  
3   `col1` INT NULL,  
4   `col2` FLOAT NULL,  
5   `col3` VARCHAR(45) NULL,  
6   PRIMARY KEY (`id`));  
7
```

Back

Apply

Cancel

Apply SQL Script to Database

Review SQL Script

Apply SQL Script

Applying SQL script to the database

The following tasks will now be executed. Please monitor the execution. Press Show Logs to see the execution logs.

Execute SQL Statements

SQL script was successfully applied to the database.

Show Logs

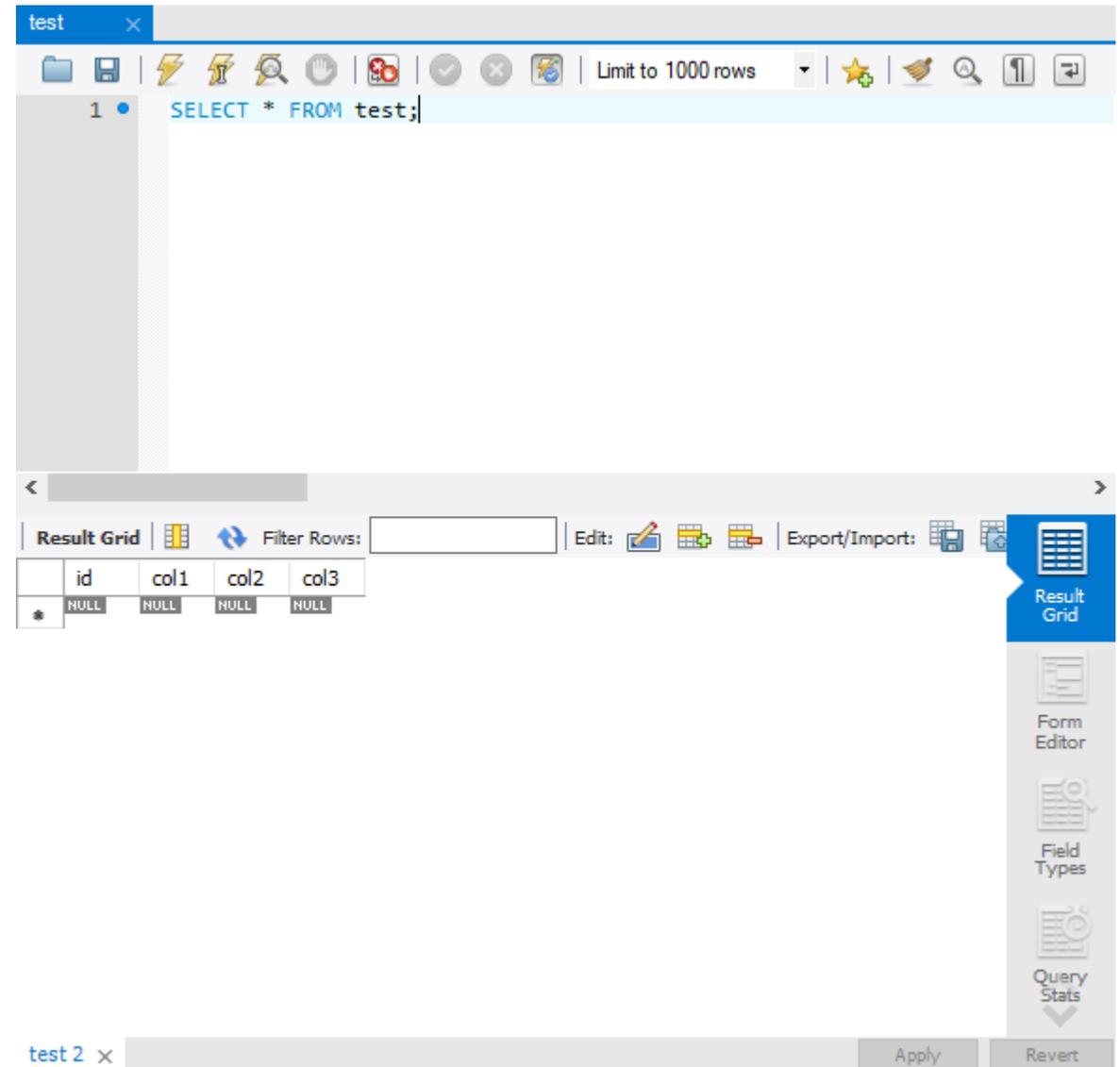
Back

Finish

Cancel

CREATE TABLE (MySQL Workbench)

- test 테이블 생성 완료



CREATE TABLE

- test2 테이블 생성 완료

The screenshot shows a database IDE window titled 'test'. The SQL editor contains the following code:

```
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;
```

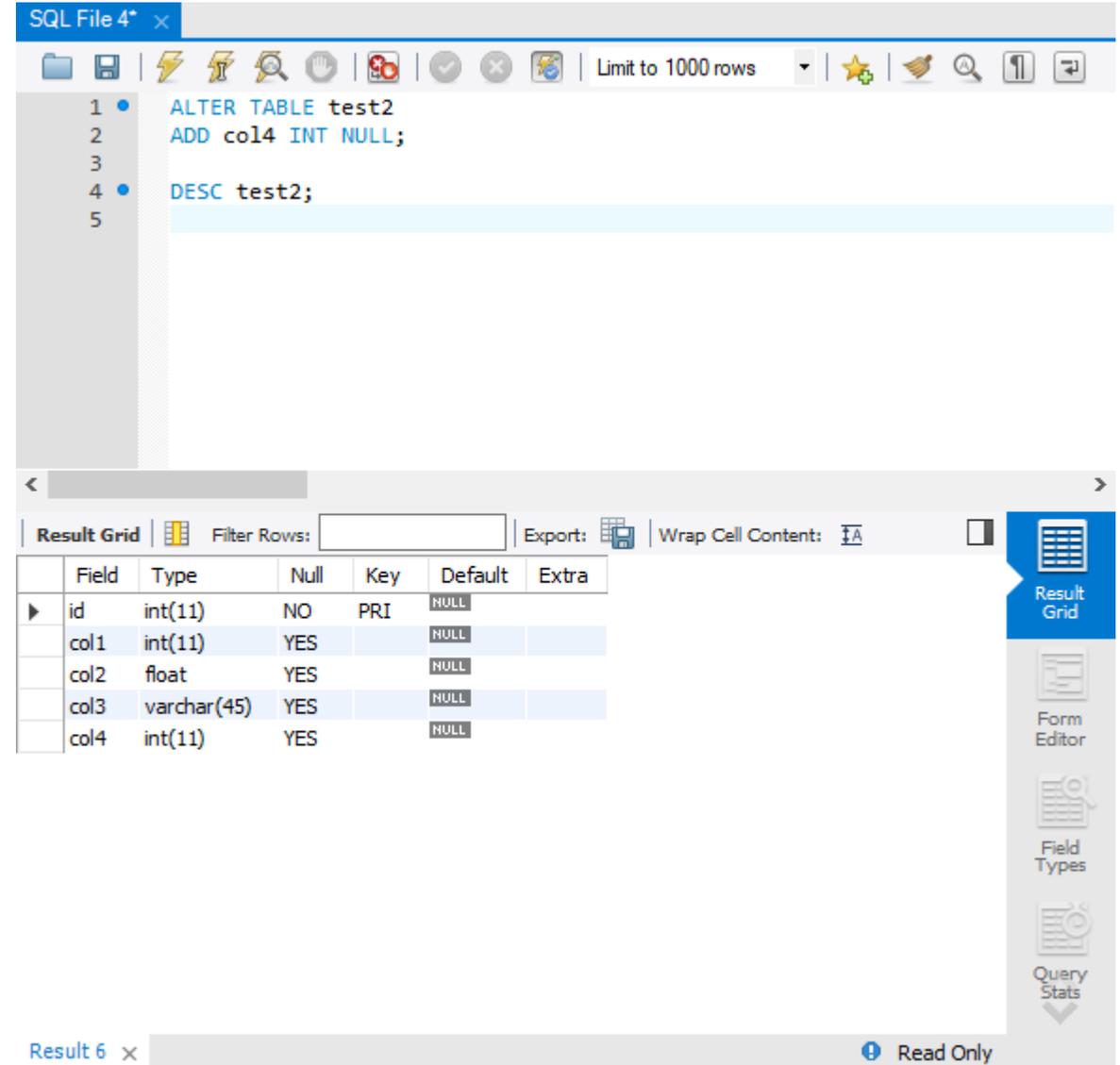
The 'Result Grid' below the editor shows the output of the query:

	id	col1	col2	col3
*	NULL	NULL	NULL	NULL

The IDE interface includes a toolbar at the top with icons for file operations, a 'Limit to 1000 rows' dropdown, and a right-hand sidebar with buttons for 'Result Grid', 'Form Editor', 'Field Types', and 'Query Stats'. At the bottom, there are 'Apply' and 'Revert' buttons.

ALTER TABLE

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



The screenshot shows a SQL IDE interface. The top window, titled "SQL File 4*", contains the following SQL code:

```
1 ALTER TABLE test2
2 ADD col4 INT NULL;
3
4 DESC test2;
5
```

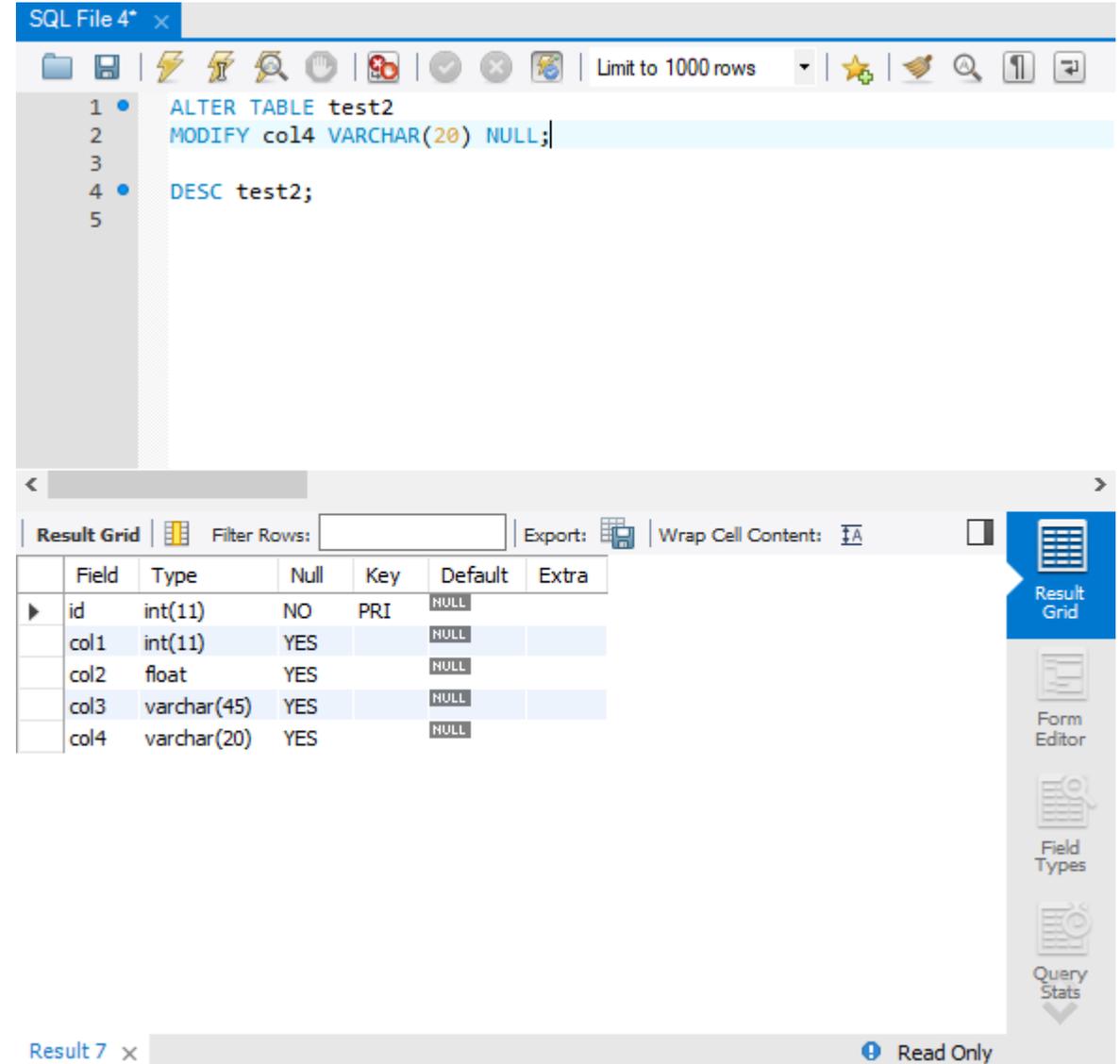
Below the code editor is a "Result Grid" showing the output of the query. The grid has columns for Field, Type, Null, Key, Default, and Extra. The data is as follows:

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 IDE also shows a "Result 6" window at the bottom, which is currently empty and marked as "Read Only".

ALTER TABLE

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



The screenshot shows a SQL IDE interface. The top window, titled "SQL File 4*", contains the following SQL code:

```
1 ALTER TABLE test2
2 MODIFY col4 VARCHAR(20) NULL;
3
4 DESC test2;
5
```

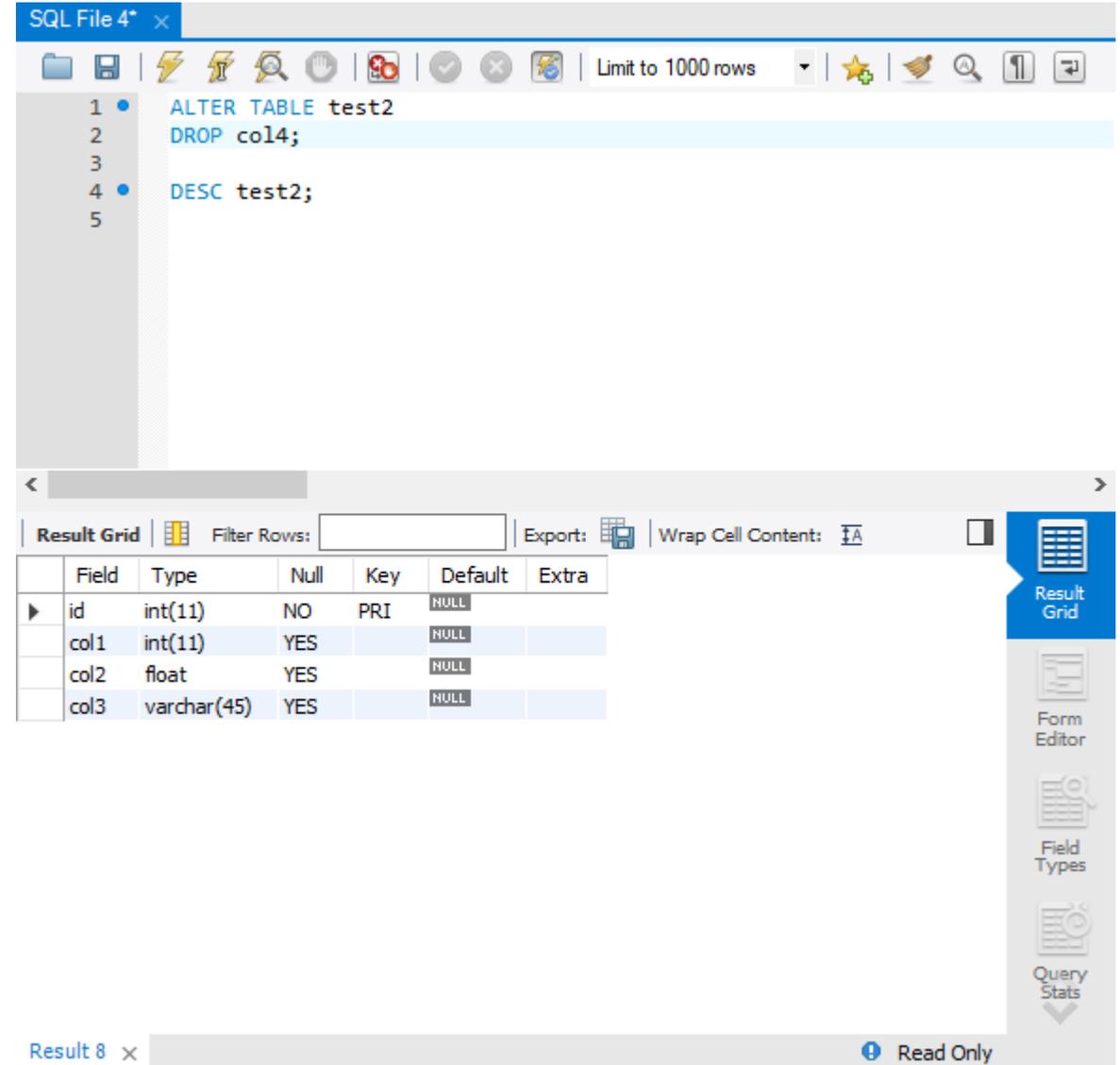
Below the code editor is a "Result Grid" showing the output of the query. The grid has columns for Field, Type, Null, Key, Default, and Extra. The data is as follows:

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 interface also includes a toolbar with various icons, a "Filter Rows" input field, and an "Export" button. On the right side, there is a vertical toolbar with icons for "Result Grid", "Form Editor", "Field Types", and "Query Stats". At the bottom, there is a "Result 7" tab and a "Read Only" indicator.

ALTER TABLE

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



The screenshot shows a SQL IDE interface. The top window, titled "SQL File 4*", contains the following SQL code:

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

Below the code editor is a "Result Grid" section. It includes a "Filter Rows:" input field, an "Export:" button, and a "Wrap Cell Content:" checkbox. The grid displays the following table structure:

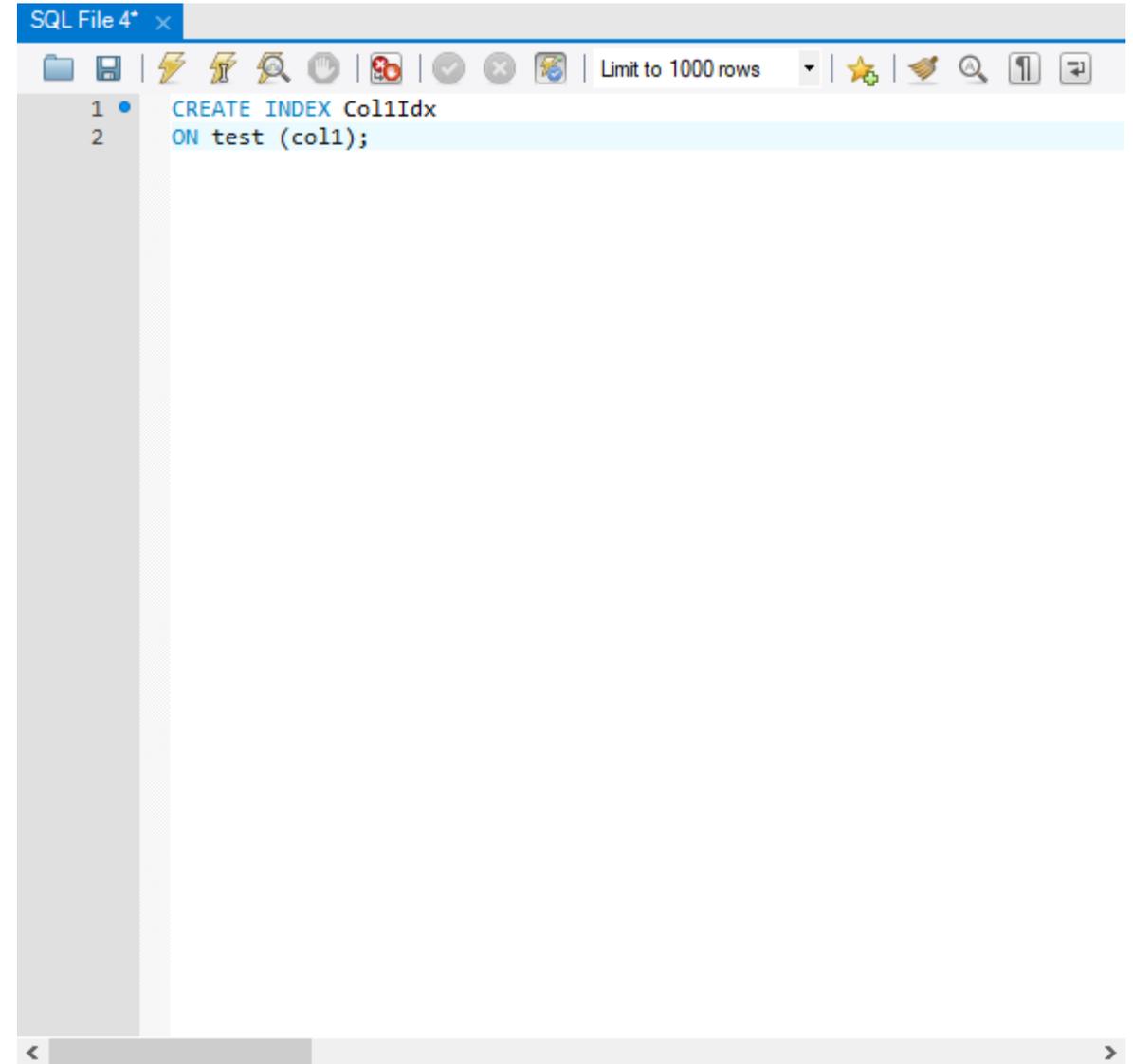
Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	
col1	int(11)	YES		NULL	
col2	float	YES		NULL	
col3	varchar(45)	YES		NULL	

At the bottom of the IDE, there is a "Result 8" tab and a "Read Only" indicator.

- 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 a SQL editor window titled "SQL File 4* x". The editor contains two lines of SQL code: "1 CREATE INDEX Col1Idx" and "2 ON test (col1);". The second line is highlighted in light blue. The editor has a toolbar with various icons and a "Limit to 1000 rows" dropdown menu. A scrollbar is visible at the bottom of the editor.

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

SHOW INDEX

- 인덱스 정보 보기

The screenshot shows a SQL IDE window titled "SQL File 4*" with the following SQL code:

```
1 SHOW INDEX
2 FROM test;
```

The result grid displays the following data:

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

The interface includes a toolbar with icons for file operations, a "Limit to 1000 rows" dropdown, and a "Result Grid" section with options for "Filter Rows:", "Export:", and "Wrap Cell Content:". A vertical sidebar on the right contains icons for "Result Grid", "Form Editor", and "Field Types". The bottom status bar shows "Result 6" and "Read Only".

CREATE UNIQUE INDEX

- 중복 값을 허용하지 않는 인덱스

The screenshot shows a SQL IDE window titled "SQL File 4*" with the following SQL code:

```
1 CREATE UNIQUE INDEX Col2Idx
2 ON test (col2);
3
4 SHOW INDEX FROM test;
```

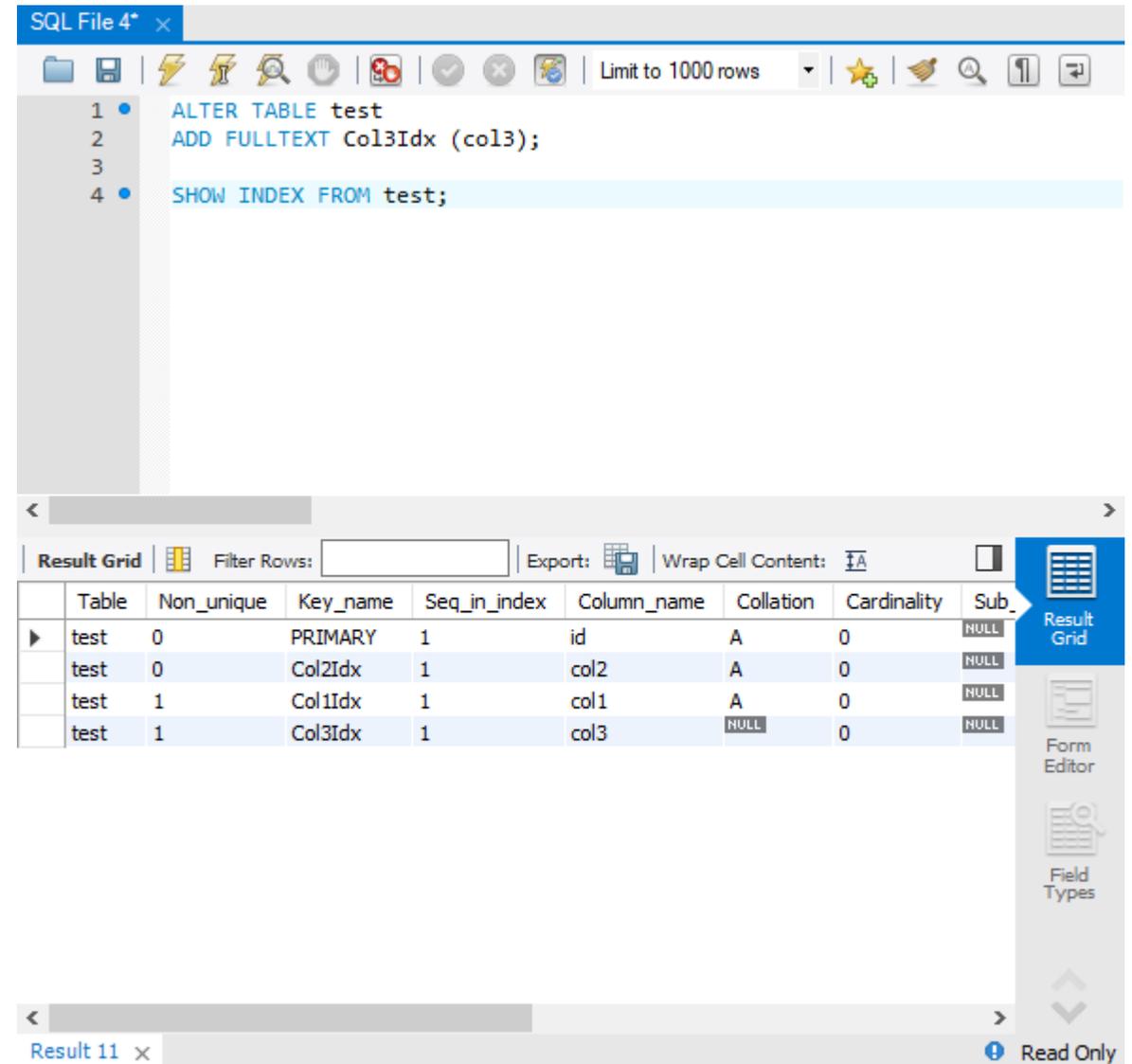
The "Result Grid" below the code displays the output of the SQL execution:

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_
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 interface also includes a toolbar with various icons, a "Limit to 1000 rows" dropdown, and a "Read Only" indicator at the bottom right.

FULLTEXT INDEX

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



The screenshot shows a SQL IDE window titled "SQL File 4*" with the following SQL commands:

```
1 ALTER TABLE test
2 ADD FULLTEXT Col3Idx (col3);
3
4 SHOW INDEX FROM test;
```

The result grid below shows the output of the "SHOW INDEX FROM test;" command:

	Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_
▶	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

The IDE interface includes a toolbar with icons for file operations, a "Limit to 1000 rows" dropdown, and a sidebar with buttons for "Result Grid", "Form Editor", and "Field Types". The status bar at the bottom indicates "Result 11 x" and "Read Only".

INDEX 삭제 (ALTER)

- ALTER 문을 사용하여 테이블에 추가된 인덱스 삭제

The screenshot shows a SQL IDE window titled "SQL File 4*" with the following SQL code:

```
1 ALTER TABLE test
2 DROP INDEX Col3Idx;
3
4 SHOW INDEX FROM test;
```

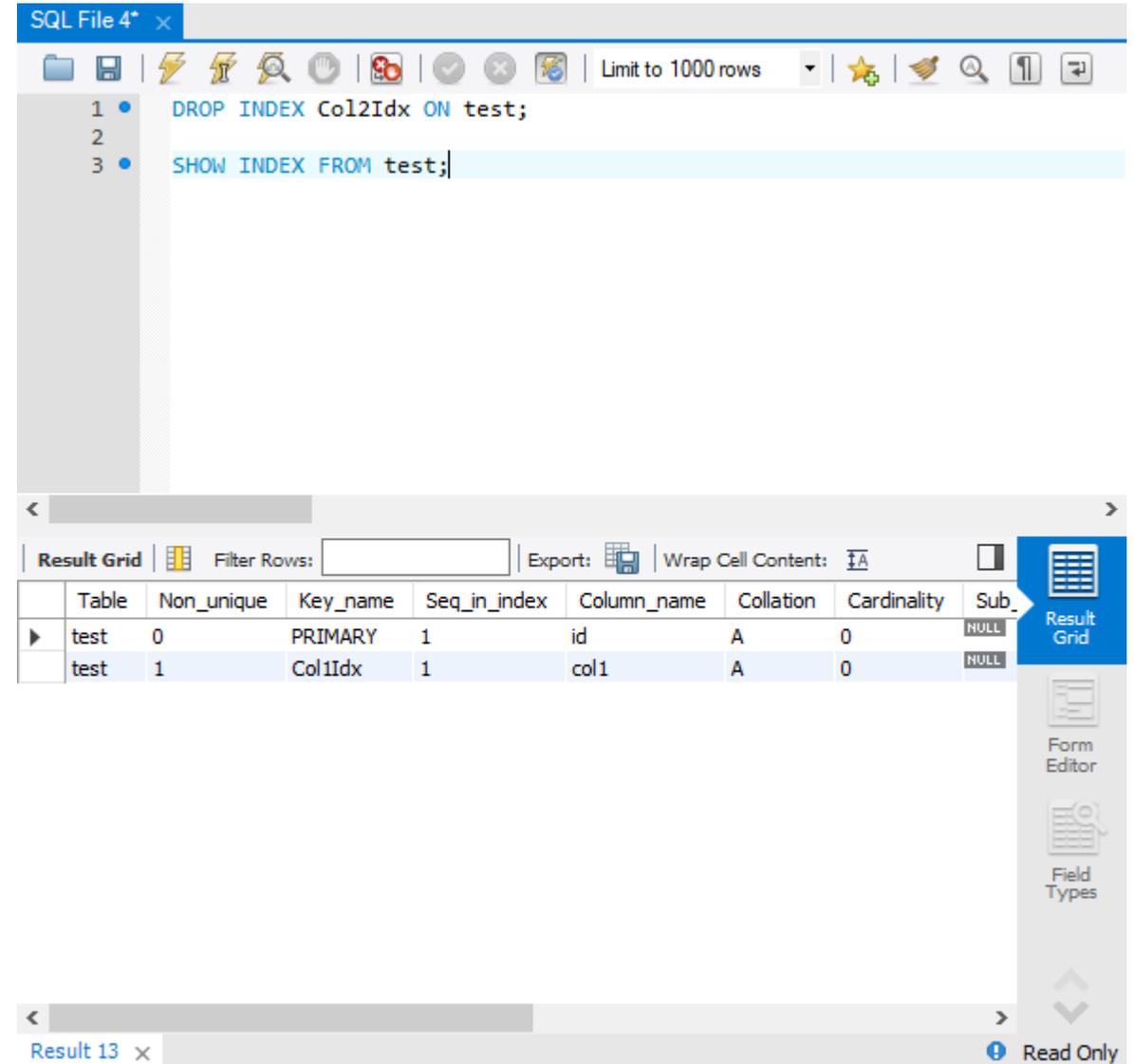
The result grid below shows the output of the SQL command:

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_
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 interface also includes a toolbar with various icons, a "Limit to 1000 rows" dropdown, and a "Result Grid" button. The status bar at the bottom indicates "Result 12 x" and "Read Only".

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 a SQL IDE window titled "SQL File 4*" with the following SQL code:

```
1 DROP INDEX Col1Idx ON test;
2
3 SHOW INDEX FROM test;
```

The result grid below the code shows the output of the `SHOW INDEX FROM test;` command:

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

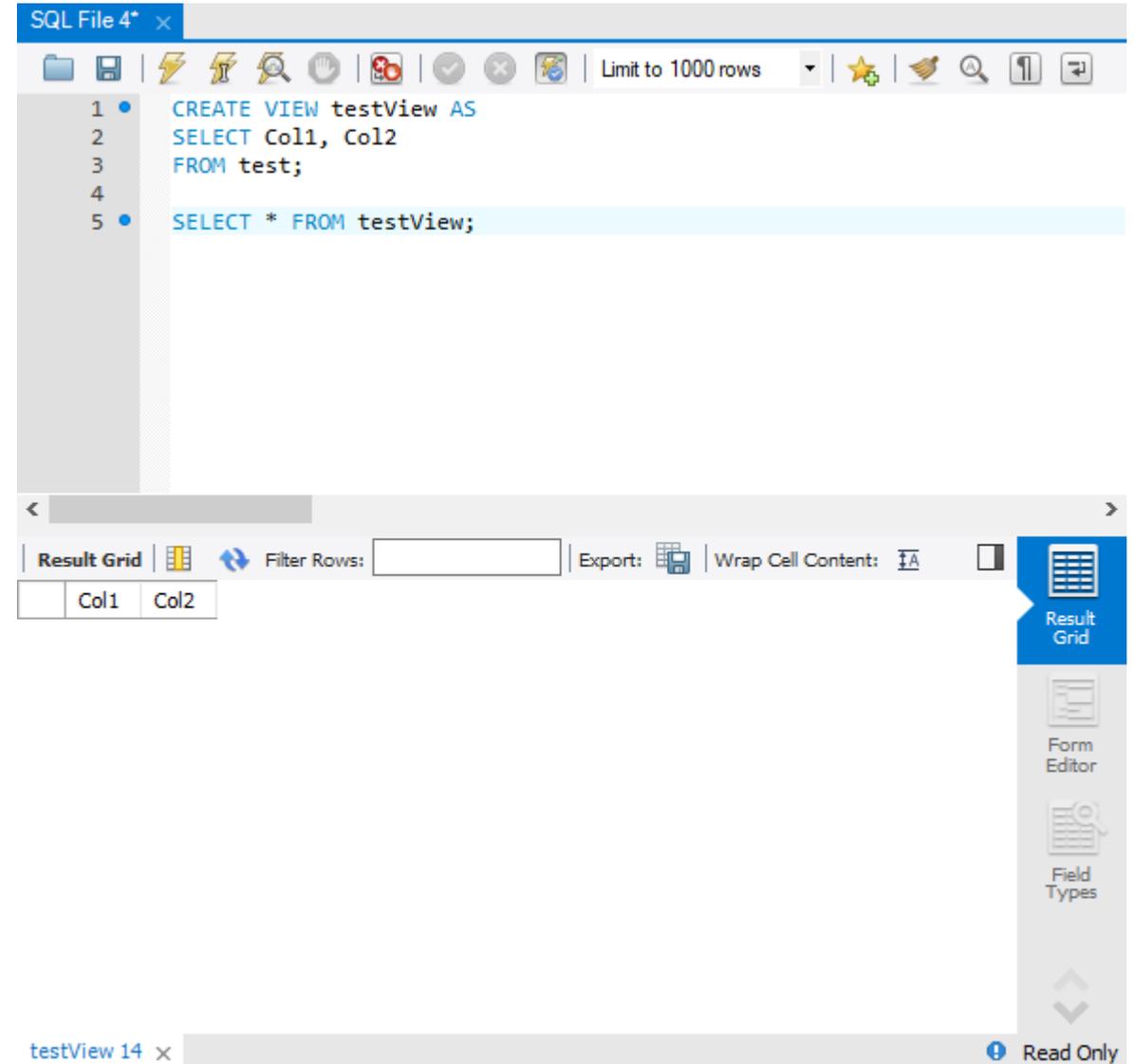
The interface also includes a toolbar with icons for file operations, a "Limit to 1000 rows" dropdown, and a "Result Grid" button. The status bar at the bottom indicates "Result 13 x" and "Read Only".

VIEW

- 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 a SQL IDE window titled "SQL File 4* x". The main editor area contains the following SQL code:

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

The second query, "SELECT * FROM testView;", is highlighted in light blue. Below the editor, there is a toolbar with options like "Result Grid", "Filter Rows", "Export", and "Wrap Cell Content". A small table is visible below the toolbar:

Col1	Col2
------	------

At the bottom of the IDE, a tab labeled "testView 14 x" is visible with a "Read Only" status.

ALTER VIEW

- ALTER 문을 사용하여 뷰를 수정

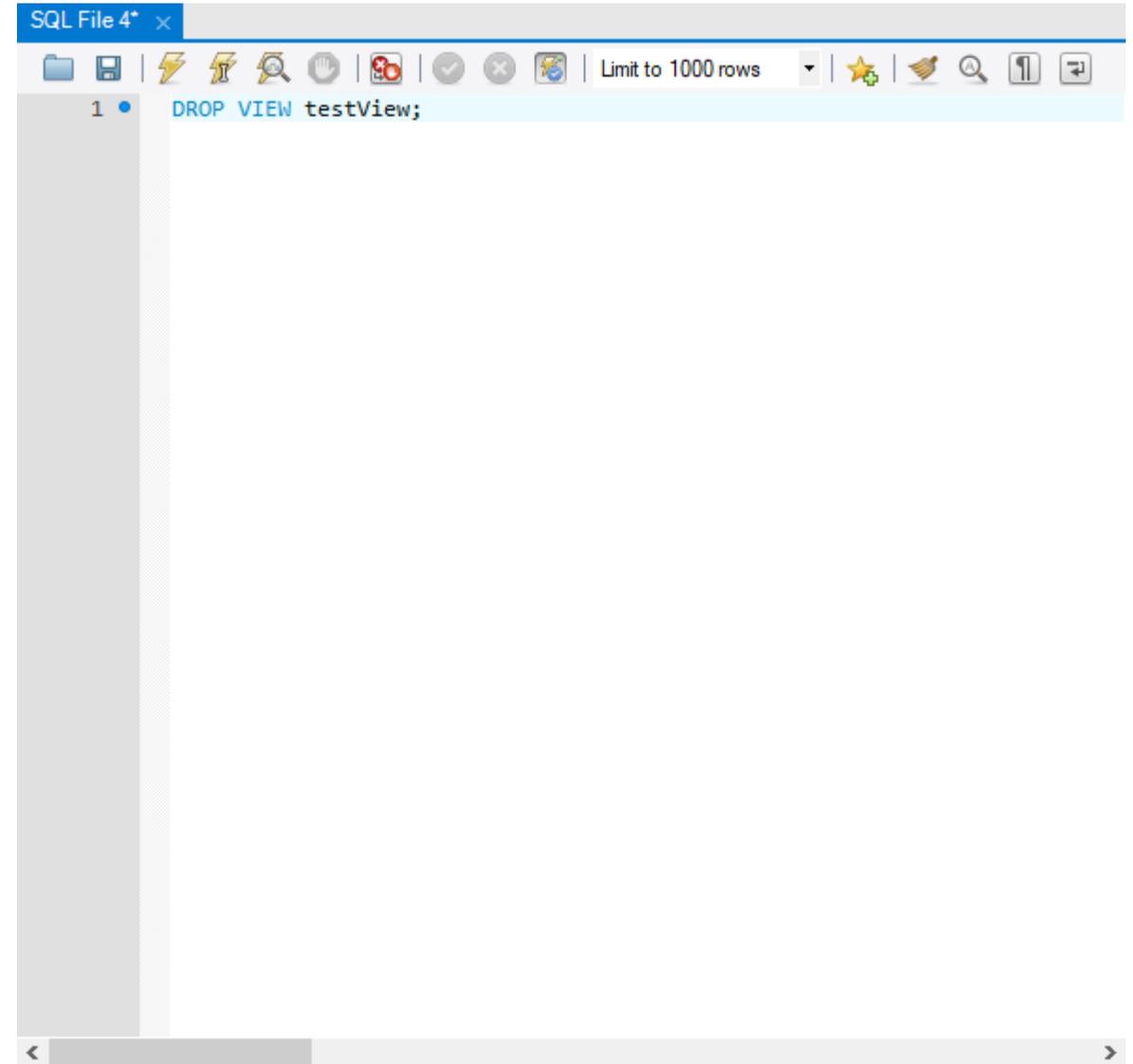
The screenshot shows a SQL IDE window titled "SQL File 4* x". The main editor contains the following SQL code:

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

The second query is highlighted in light blue. Below the editor, the "Result Grid" is visible, showing a table with three columns: Col1, Col2, and Col3. The table is currently empty. The interface includes a toolbar with various icons, a "Limit to 1000 rows" dropdown, and a "Filter Rows" input field. On the right side, there are buttons for "Result Grid", "Form Editor", and "Field Types". At the bottom, a tab labeled "testView 15 x" is shown with a "Read Only" indicator.

DROP VIEW

- DROP 문을 사용하여 생성된 뷰를 삭제



The image shows a screenshot of a SQL editor window titled "SQL File 4* x". The window has a toolbar with various icons for file operations, execution, and search. Below the toolbar, the text "1 • DROP VIEW testView;" is displayed in a light blue background, indicating the SQL command being entered or executed. The window also shows a "Limit to 1000 rows" dropdown menu and a search icon.

**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 a database client window titled 'test'. The SQL editor contains the following code:

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

Below the editor, the 'Result Grid' is displayed with the following data:

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

The interface also includes a toolbar with icons for file operations, a 'Limit to 1000 rows' dropdown, and a sidebar with options like 'Form Editor', 'Field Types', and 'Query Stats'. At the bottom, there are 'Apply' and 'Revert' buttons.

INSERT (MySQL Workbench)

The screenshot shows the MySQL Workbench interface. The SQL editor contains the following code:

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

The result grid below shows the output of the SQL script:

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

The interface also shows a toolbar with icons for file operations, a 'Limit to 1000 rows' dropdown, and a sidebar with options like 'Result Grid', 'Form Editor', 'Field Types', and 'Query Stats'. At the bottom, there are tabs for 'test 4' and 'test 5' and buttons for 'Apply' and 'Revert'.

The screenshot shows the 'Apply SQL Script to Database' dialog box. The 'Review SQL Script' section displays the following SQL code:

```
1 INSERT INTO `world`.`test` (`id`, `col1`, `col2`, `col3`) VALUES ('2', '234', '2
2 INSERT INTO `world`.`test` (`id`, `col1`, `col2`, `col3`) VALUES ('3', '345', '3
3
```

The dialog box has a 'Back' button, an 'Apply' button, and a 'Cancel' button at the bottom right.

INSERT INTO SELECT

- test 테이블에 있는 내용을 test2 테이블에 삽입

The screenshot shows a SQL IDE interface. The top window, titled 'test', contains the following SQL code:

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

Below the code editor is a 'Result Grid' showing the output of the query. The grid has four columns: 'id', 'col1', 'col2', and 'col3'. The data rows are:

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

The interface also includes a toolbar with various icons, a 'Filter Rows' input field, and a vertical sidebar on the right with buttons for 'Result Grid', 'Form Editor', 'Field Types', and 'Query Stats'. At the bottom, there are tabs for 'test 4', 'test 5', and 'test2 7', along with 'Apply' and 'Revert' buttons.

UPDATE

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

The screenshot shows a MySQL IDE window titled 'test'. The query 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 query editor, the 'Result Grid' is displayed, showing the results of the SELECT query. The grid has the following data:

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

The IDE interface includes various toolbars and a sidebar with options like 'Result Grid', 'Form Editor', 'Field Types', and 'Query Stats'. The status bar at the bottom shows 'test 4', 'test 5', 'test 8 x', 'Apply', and 'Revert'.

DELETE

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

The screenshot shows a database IDE window titled 'test'. The SQL editor contains the following code:

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

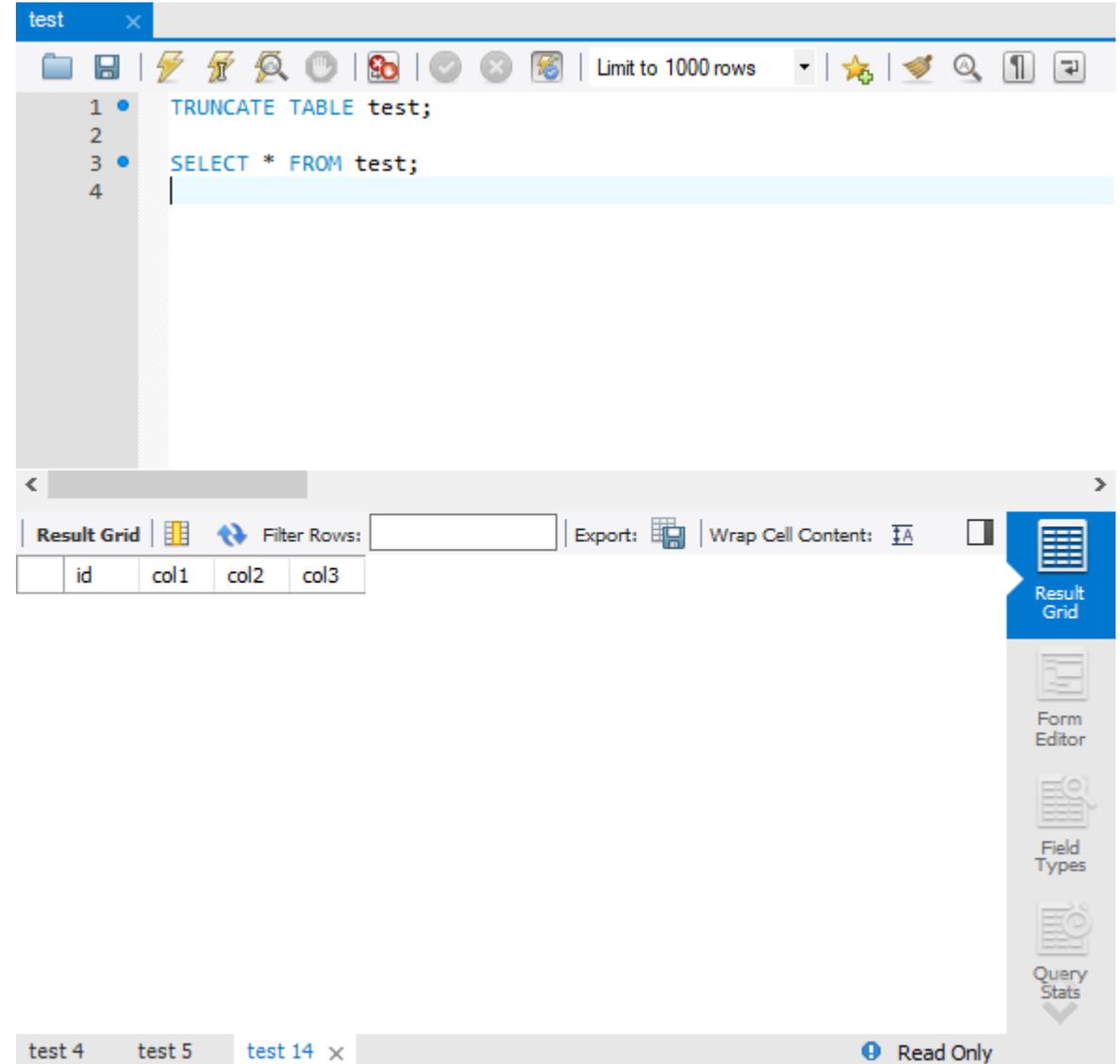
Below the editor is a 'Result Grid' showing the output of the query. The grid has columns 'id', 'col1', 'col2', and 'col3'. The first row is selected, showing '2', '234', '2.2', and 'Test'. The second row shows '3', '345', '3.3', and 'Test'. The third row is a summary row with 'NULL' values for all columns.

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

The IDE interface includes a toolbar with various icons, a 'Limit to 1000 rows' dropdown, and a sidebar with buttons for 'Result Grid', 'Form Editor', 'Field Types', and 'Query Stats'. The bottom status bar shows 'test 4', 'test 5', 'test 13 x', 'Apply', and 'Revert'.

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 a SQL IDE window titled 'test'. The query editor contains the following SQL code:

```
1 TRUNCATE TABLE test;  
2  
3 SELECT * FROM test;  
4
```

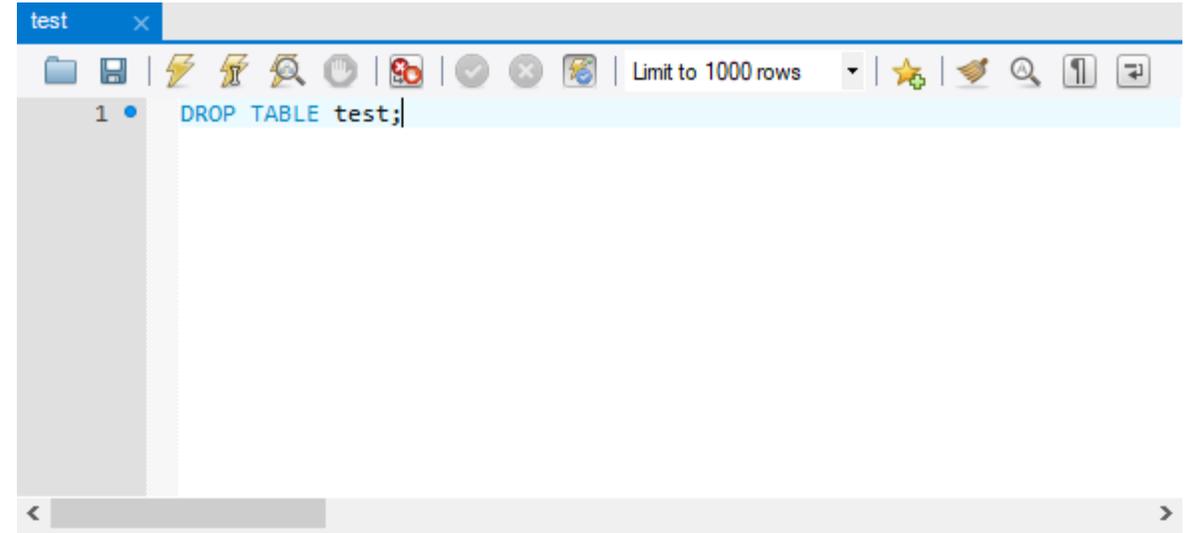
Below the query editor, the 'Result Grid' is visible, showing the following columns:

id	col1	col2	col3
----	------	------	------

The IDE interface includes various toolbars and a sidebar on the right with options like 'Result Grid', 'Form Editor', 'Field Types', and 'Query Stats'. The status bar at the bottom indicates 'test 4 test 5 test 14 x' and 'Read Only'.

DROP TABLE

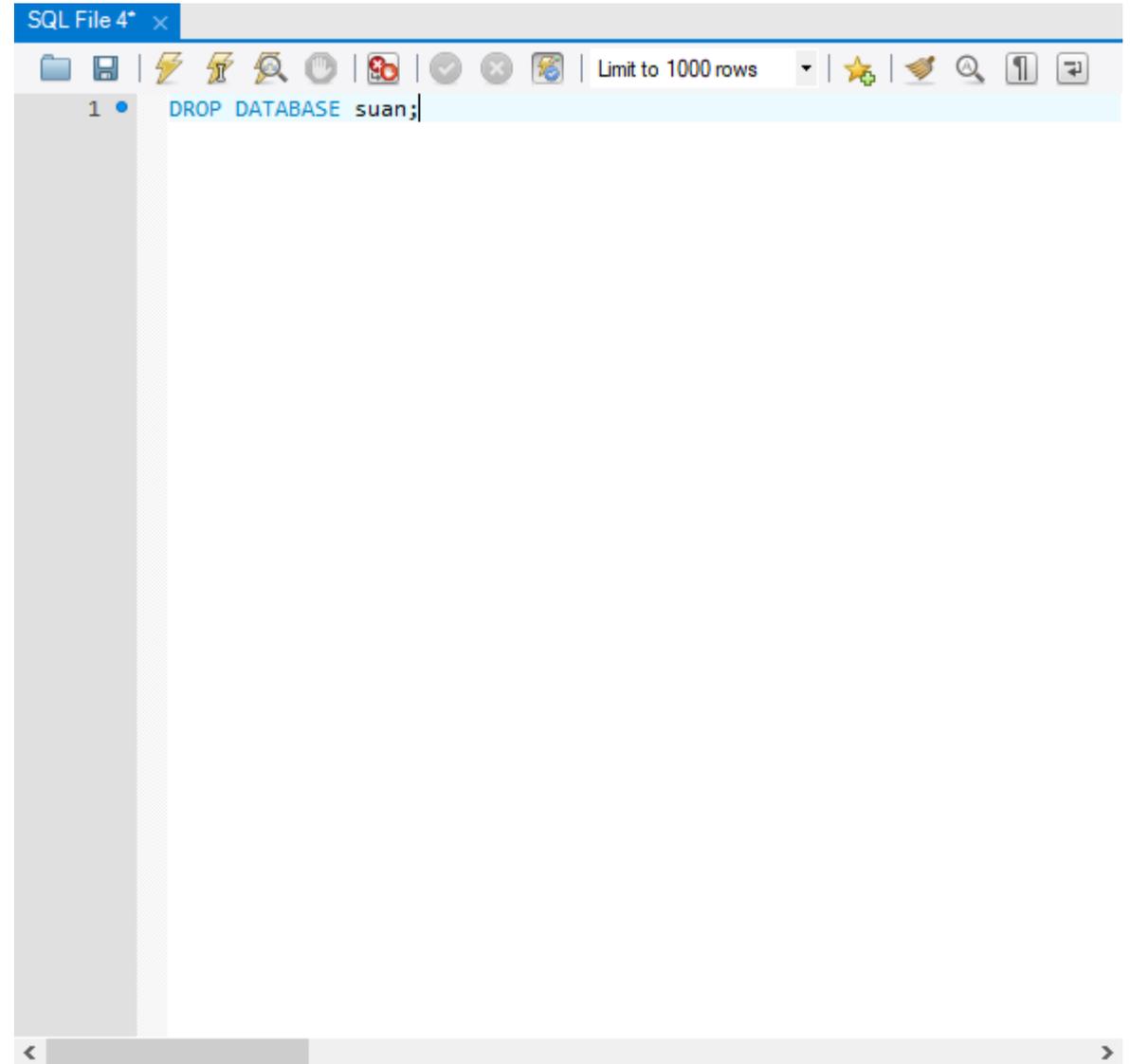
- 테이블 전체를 삭제, 공간, 객체를 삭제
- 삭제 후 절대 되돌릴 수 없음



The screenshot shows a MySQL command-line interface window titled 'test'. The window has a toolbar with various icons for file operations, execution, and search. Below the toolbar, a single line of SQL code is entered: 'DROP TABLE test;'. The text is highlighted in light blue. The window also shows a 'Limit to 1000 rows' dropdown menu and a scroll bar at the bottom.

DROP DATABASE

- DROP DATABASE 문은 해당 데이터베이스를 삭제



The screenshot shows a SQL editor window titled "SQL File 4* x". The toolbar includes icons for file operations, execution, search, and settings. A dropdown menu is set to "Limit to 1000 rows". The main text area contains the SQL command "1 DROP DATABASE suan;".

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

(참고) 데이터 타입:

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

