

[TA 3-3]

Database and SQL

- Basics

빅데이터 프로그래밍 기초

박진수 교수

Big Data Institute,
Seoul National University

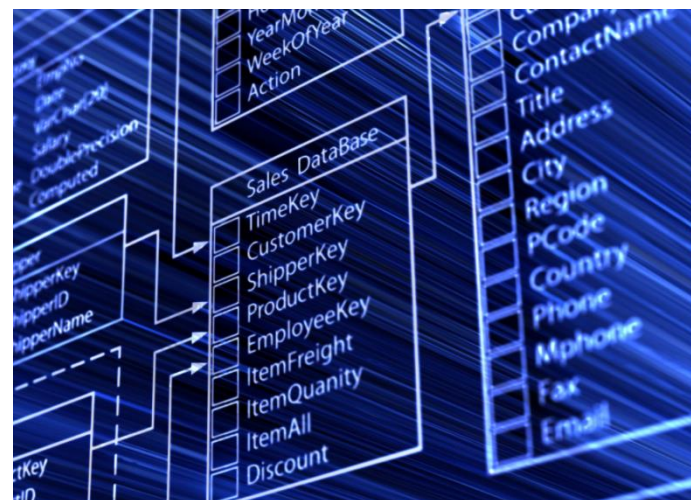
데이터베이스 시스템(Database Systems)



데이터베이스(Database)

· 데이터베이스: 체계화된 데이터의 모임("the collection of data"). 작성된 목록으로써 여러 응용 시스템의 통합된 정보를 저장하여 운영할 수 있는 공용 데이터들의 묶음

- 여러 사람이 공유하고 사용할 목적으로 통합 관리되는 정보의 집합
- 논리적으로 연관된 하나 이상의 자료의 모음으로, 그 내용을 고도로 구조화함으로써 검색과 갱신의 효율화를 꾀한 것
- “몇 개의 자료 파일을 조직적으로 통합하여 자료 항목의 중복을 없애고 자료를 구조화하여 기억시켜 놓은 자료의 집합체”



데이터베이스(Database)

- 데이터베이스의 장점

- 데이터 중복(Data redundancy) 최소화
- 데이터 공유
- 일관성(consistency), 무결성(integrity), 보안성(security) 유지
- 최신 데이터 유지
- 데이터의 표준화(standardization)
- 용이한 데이터 접근
- 데이터 저장공간 절약

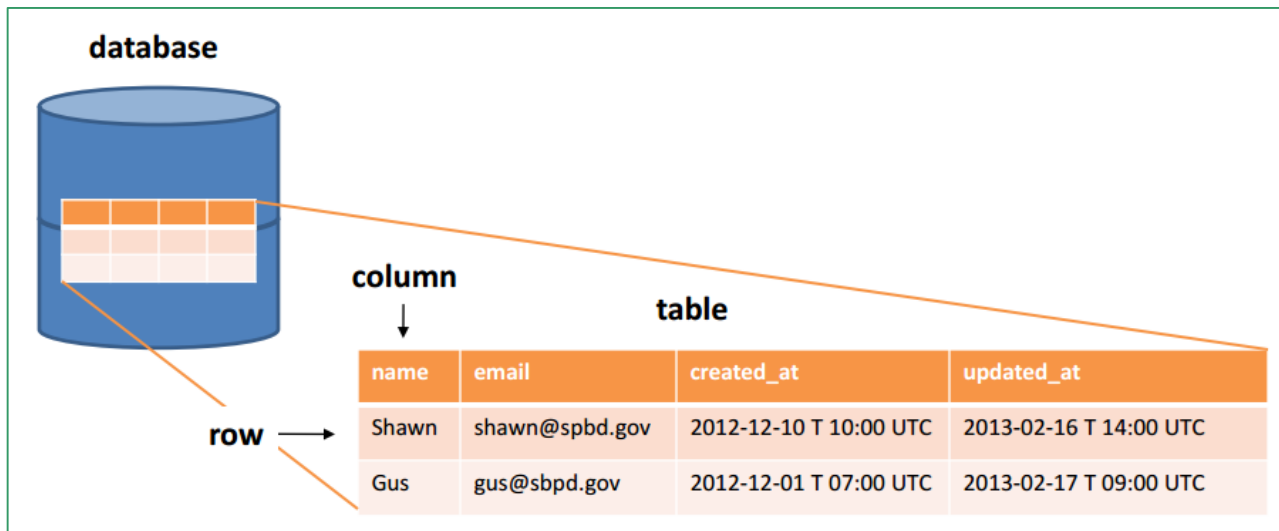
관계형 데이터베이스(Relational Database)

· 관계형 데이터베이스(RDB): 데이터를 표(table)로 표현하는 형식의 데이터베이스

- 1970년 커드(codd)가 제시한 데이터 관계형 모델에 기반한 키(key)와 값(value)들의 간단한 관계를 테이블화 시킨 매우 간단한 데이터베이스

- 데이터를 표를 이용해 구조화하고, 그들 간의 관계(relationship)를 바탕으로 서로를 연결(link)한다

- 그러한 “관계”들은 하나의 쿼리(query)로 여러 표에 산재한 정보를 불러오는 것을 가능케 한다



관계형 데이터베이스(Relational Database)

· 관계형 데이터베이스 예시 – 대학교 데이터베이스(University database)

- 교수자(instructor) 테이블

- 부서(department) 테이블

ID	name	dept_name	salary
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

(a) The *instructor* table

dept_name	building	budget
Comp. Sci.	Taylor	100000
Biology	Watson	90000
Elec. Eng.	Taylor	85000
Music	Packard	80000
Finance	Painter	120000
History	Painter	50000
Physics	Watson	70000

(b) The *department* table

관계형 데이터베이스(Relational Database)

· 관계형 데이터베이스 예시 – 대학교 데이터베이스(University database)

- 관계를 나타내기 위한 E-R 다이어그램(Entity-Relationship Diagram)

- E-R 다이어그램: 개체(entities)와 개체들 간의 관계(relationships)를 바탕으로 데이터를 표현하는 방식

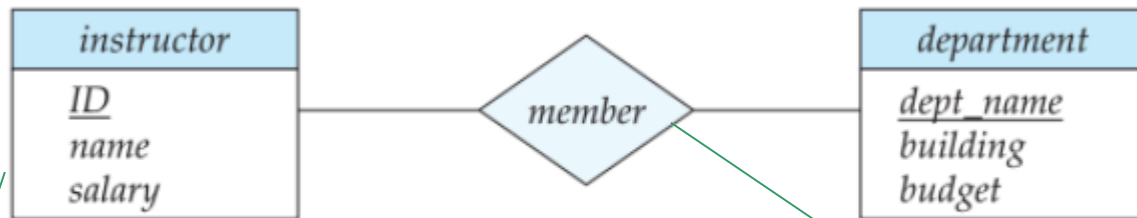


Figure 1.3 A sample E-R diagram.

Entity

Relationship

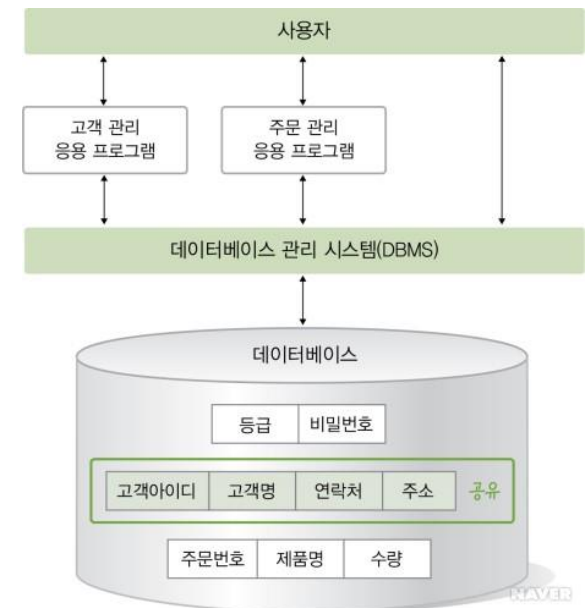
데이터베이스 관리 시스템(DBMS)

· 데이터베이스 관리 시스템: 데이터베이스를 관리하는데 필요한 데이터의 추가, 변경, 삭제, 검색 등의 기능을 집대성한 소프트웨어 패키지

- 데이터베이스에 존재하는 데이터의 검색/삽입/삭제/수정을 가능하게 하고, 모든 응용프로그램이 데이터베이스를 공유할 수 있게 한다

- DBMS의 주요 기능

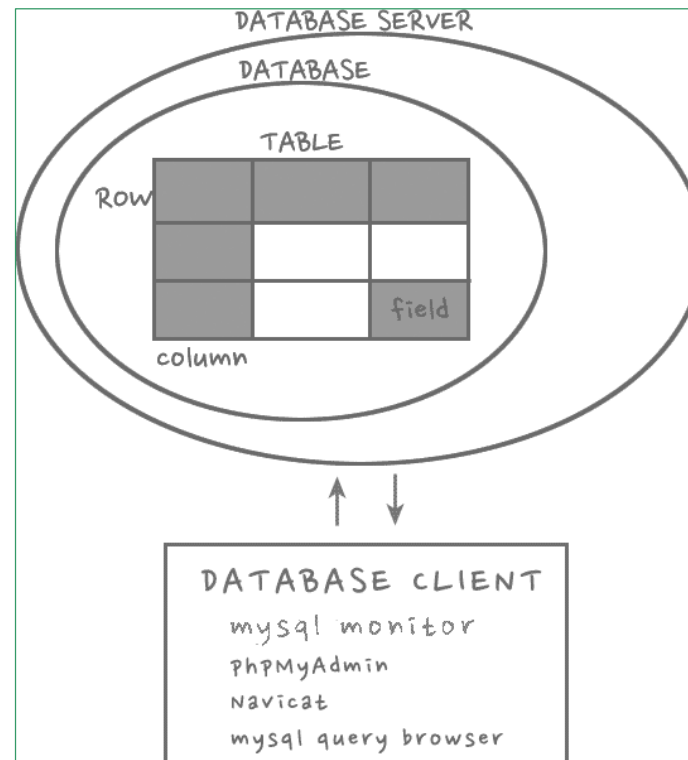
- 정의: 데이터베이스 구조를 정의하거나 수정할 수 있다
- 조작: 데이터를 삽입/삭제/수정/검색하는 연산을 할 수 있다
- 제어: 데이터를 항상 정확하고 안전하게 유지할 수 있다



데이터베이스 관리 시스템(DBMS)

· 관계형 데이터베이스 관리시스템: 관계형 모델을 기반으로 하는 DBMS. 현재 상업적으로 사용되는 대부분의 DB는 RDB 모델을 기반으로 한다

- 마이크로소프트의 액세스(Access)와 MSSQL, 오라클의 오라클(Oracle)과 MySQL 등
- DB 내에 테이블들이 존재하고 테이블 내에 데이터가 존재



구조화 질의 언어(SQL)

· 구조화 질의 언어(Structured Query Language): 데이터베이스에서 데이터를 저장하거나 얻기 위해서 사용하는 표준 언어

- 데이터 정의 언어: 테이블과 인덱스 구조 등을 관리

■ CREATE: 테이블 등을 생성

■ DROP: 테이블 등을 삭제

- 데이터 조작 언어: 데이터 변경

■ INSERT: 데이터 삽입

■ DELETE: 행 삭제

■ SELECT: 검색 결과 집합의 취득

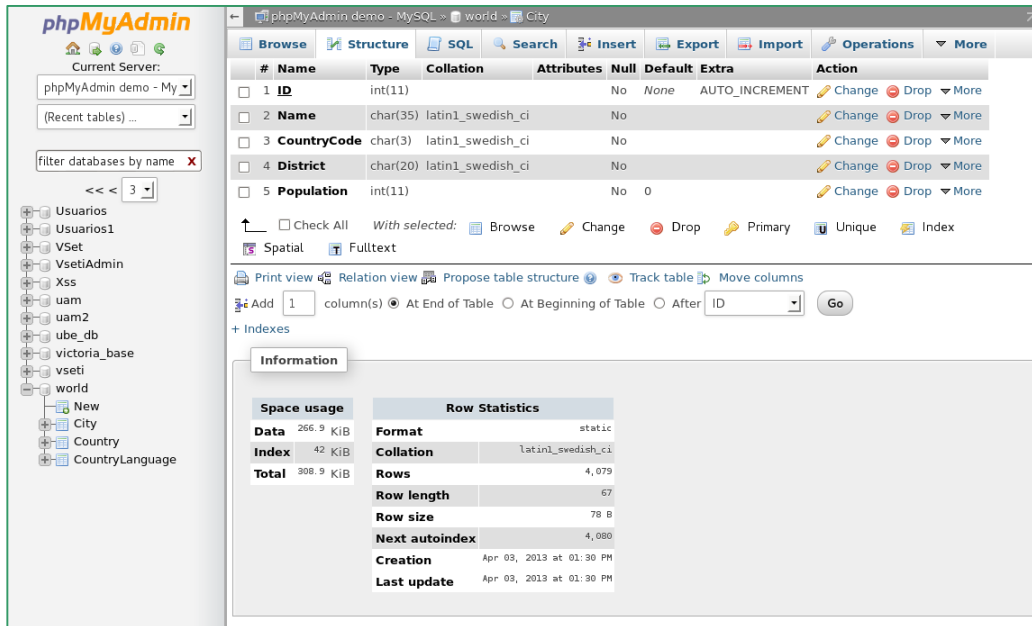
- 데이터 제어 언어: 데이터 작업 권한 등을 관리

MySQL 설치



MySQL이란?

- MySQL: 세계에서 가장 많이 쓰이는 오픈 소스 관계형 데이터베이스 관리 시스템(RDBMS)
- 파이썬의 MySQL API를 활용하여 MySQL 데이터베이스에 접근할 수 있다(pymysql 등)



The screenshot displays the phpMyAdmin interface for a MySQL database. The left sidebar shows a list of databases, including 'Usuarios', 'VSet', 'uam', 'uam2', 'ube_db', 'victoria_base', 'vseti', 'world', and 'New'. The main panel shows the structure of the 'City' table, which has five columns: ID, Name, CountryCode, District, and Population. The table is using the latin1_swedish_ci collation. Below the table structure, there is an 'Information' tab showing space usage and row statistics.

#	Name	Type	Collation	Attributes	Null	Default	Extra	Action
1	ID	int(11)			No	None	AUTO_INCREMENT	Change Drop More
2	Name	char(35)	latin1_swedish_ci		No			Change Drop More
3	CountryCode	char(3)	latin1_swedish_ci		No			Change Drop More
4	District	char(20)	latin1_swedish_ci		No			Change Drop More
5	Population	int(11)			No	0		Change Drop More

Space usage

Space usage	Value
Data	266.9 KiB
Index	42 KiB
Total	308.9 KiB

Row Statistics

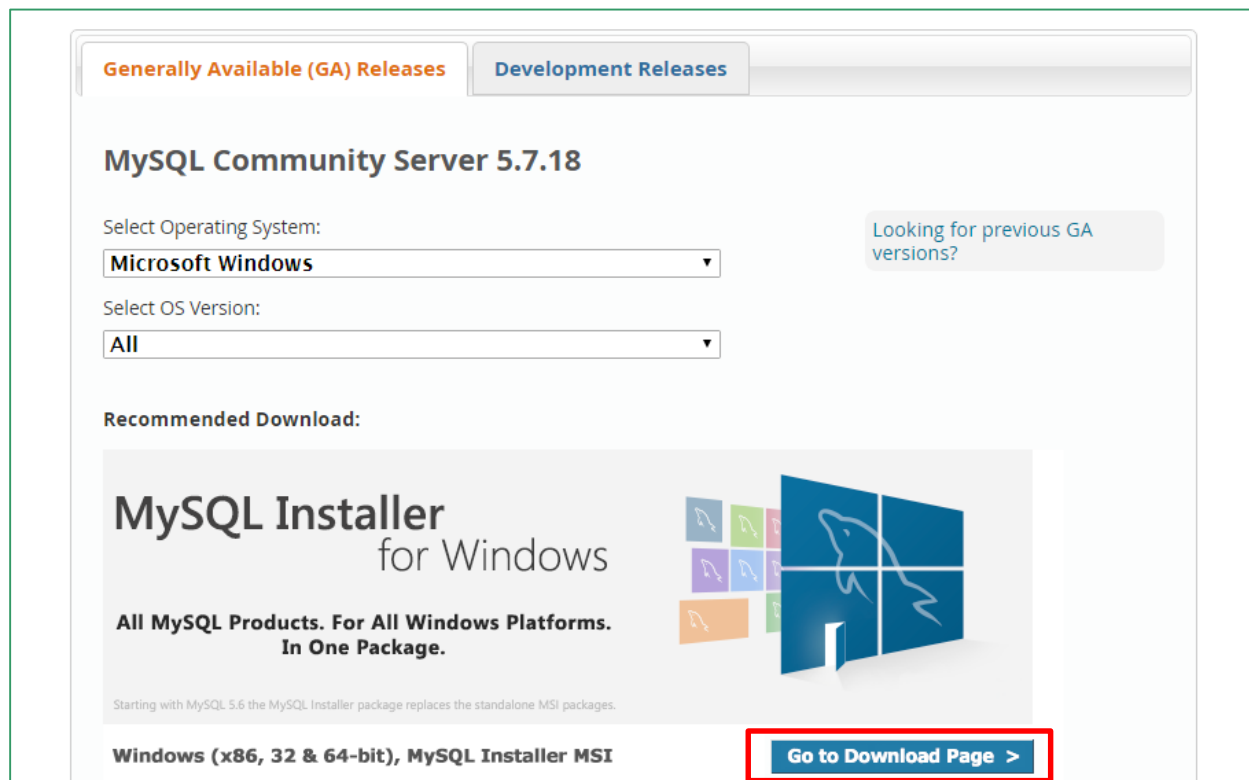
Row Statistics	Value
Format	static
Collation	latin1_swedish_ci
Rows	4,979
Row length	67
Row size	78 B
Next autoindex	4,080
Creation	Apr 03, 2013 at 01:30 PM
Last update	Apr 03, 2013 at 01:30 PM



MySQL 설치(for Windows)

· MySQL 홈페이지에서 community version 다운로드 가능(무료)

- URL: <https://dev.mysql.com/downloads/mysql/>



MySQL 설치(for Windows)

· MySQL 홈페이지에서 community version 다운로드 가능(무료)

- 운영체제와 비트수에 맞는 installer 다운로드

Generally Available (GA) Releases **Development Releases**

MySQL Community Server 5.7.18

Select Operating System:
Mac OS X

[Looking for previous GA versions?](#)

Packages for Sierra (10.12) are compatible with El Capitan (10.11)

Mac OS X 10.12 (x86, 64-bit), DMG Archive (mysql-5.7.18-macos10.12-x86_64.dmg)	5.7.18	331.7M	Download
MD5: 8e8e3dea84afdaa7995f39fb025c418e Signature			
Mac OS X 10.12 (x86, 64-bit), Compressed TAR Archive (mysql-5.7.18-macos10.12-x86_64.tar.gz)	5.7.18	307.3M	Download
MD5: 0fcd44dde766c20b91b79982e7a5252d Signature			
Mac OS X 10.12 (x86, 64-bit), Compressed TAR Archive Test Suite (mysql-test-5.7.18-macos10.12-x86_64.tar.gz)	5.7.18	24.3M	Download
MD5: 7df5f67383a5dcec61902ab90cf5a88f Signature			

report any bugs or inconsistencies you observe to our [Bugs Database](#).
you for your support!

Generally Available (GA) Releases **Development Releases**

MySQL Installer 5.7.18

Select Operating System:
Microsoft Windows

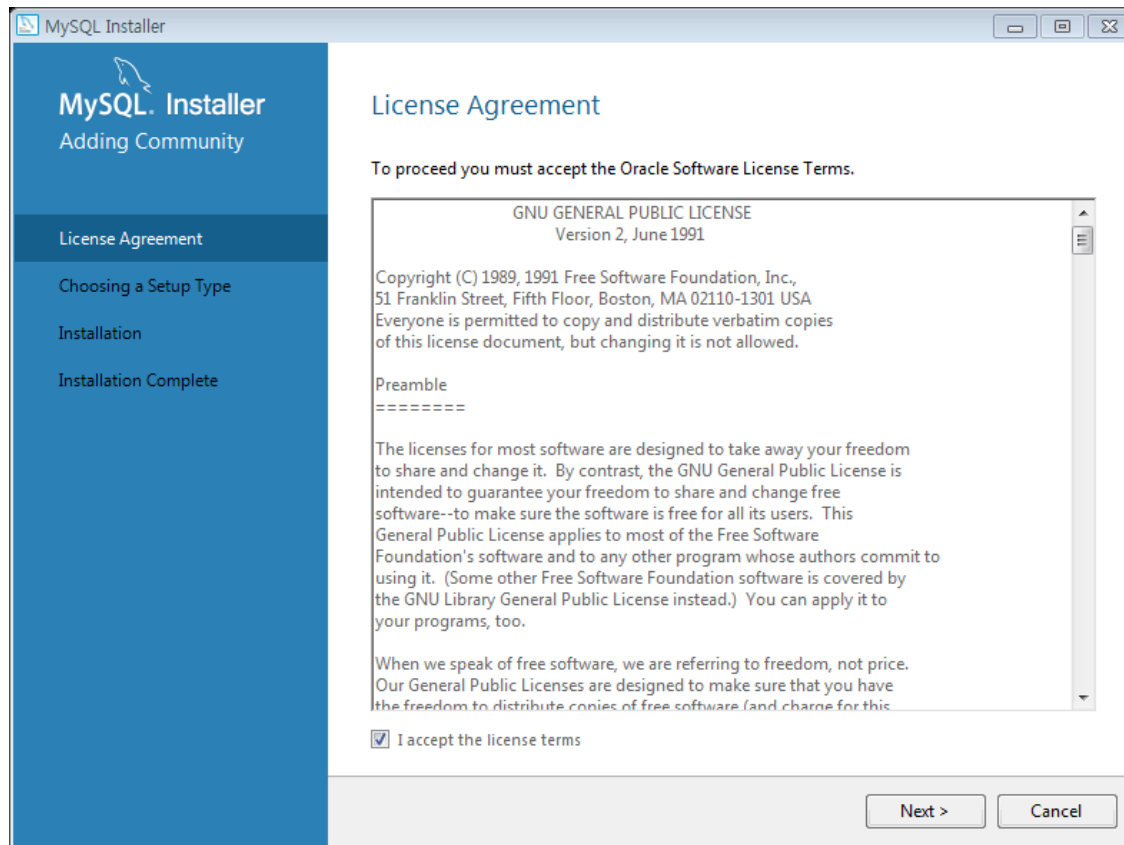
[Looking for previous GA versions?](#)

Windows (x86, 32-bit), MSI Installer (mysql-installer-web-community-5.7.18.1.msi)	5.7.18	18.5M	Download
MD5: acc7a5ec61c2dafbe975011fd8c8ed90 Signature			
Windows (x86, 32-bit), MSI Installer (mysql-installer-community-5.7.18.1.msi)	5.7.18	405.8M	Download
MD5: 3c4bfbc433c78fc082093d29c73bee66 Signature			

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

MySQL 설치(for Windows)

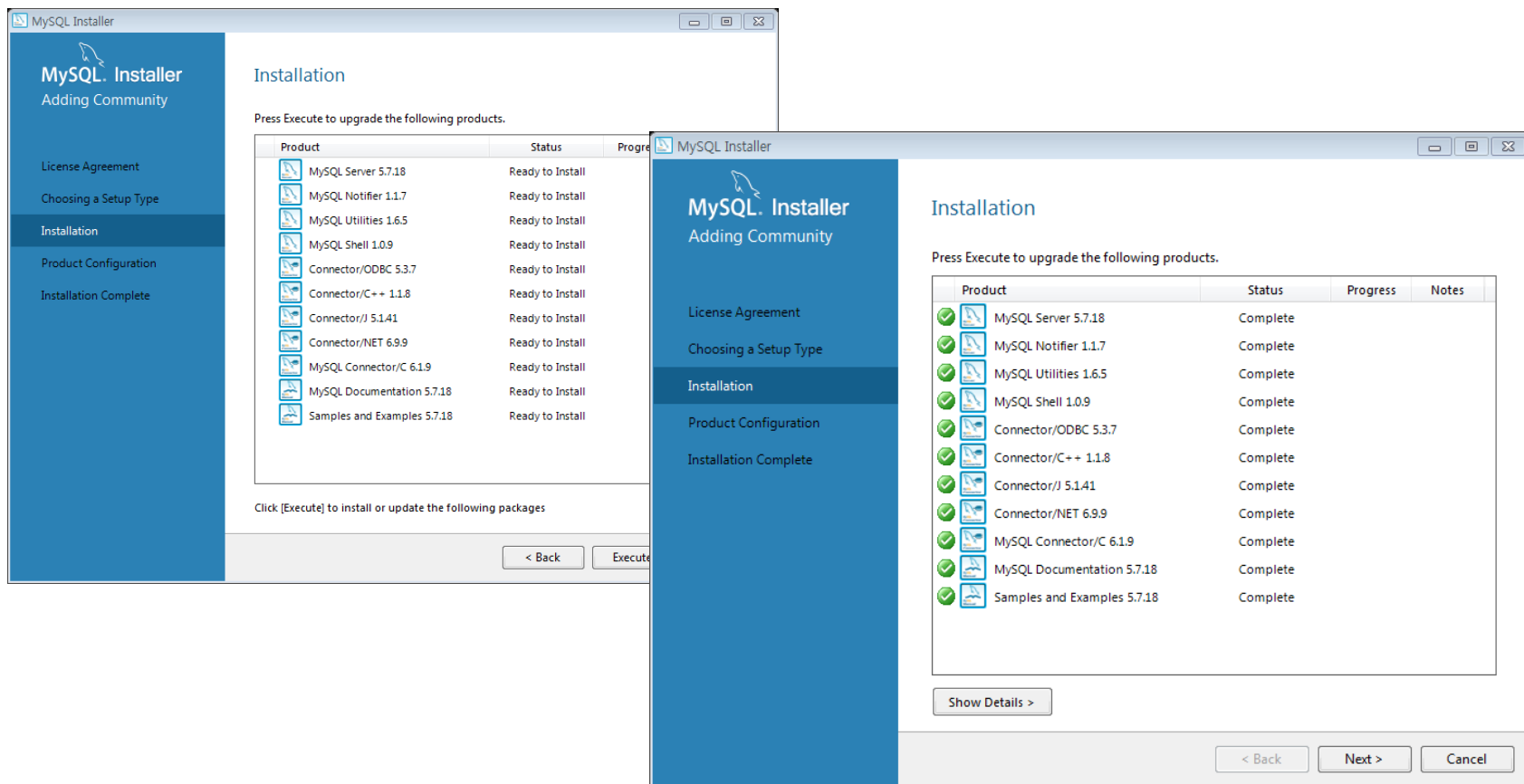
- 불린(Boolean) 타입



MySQL 설치(for Windows)

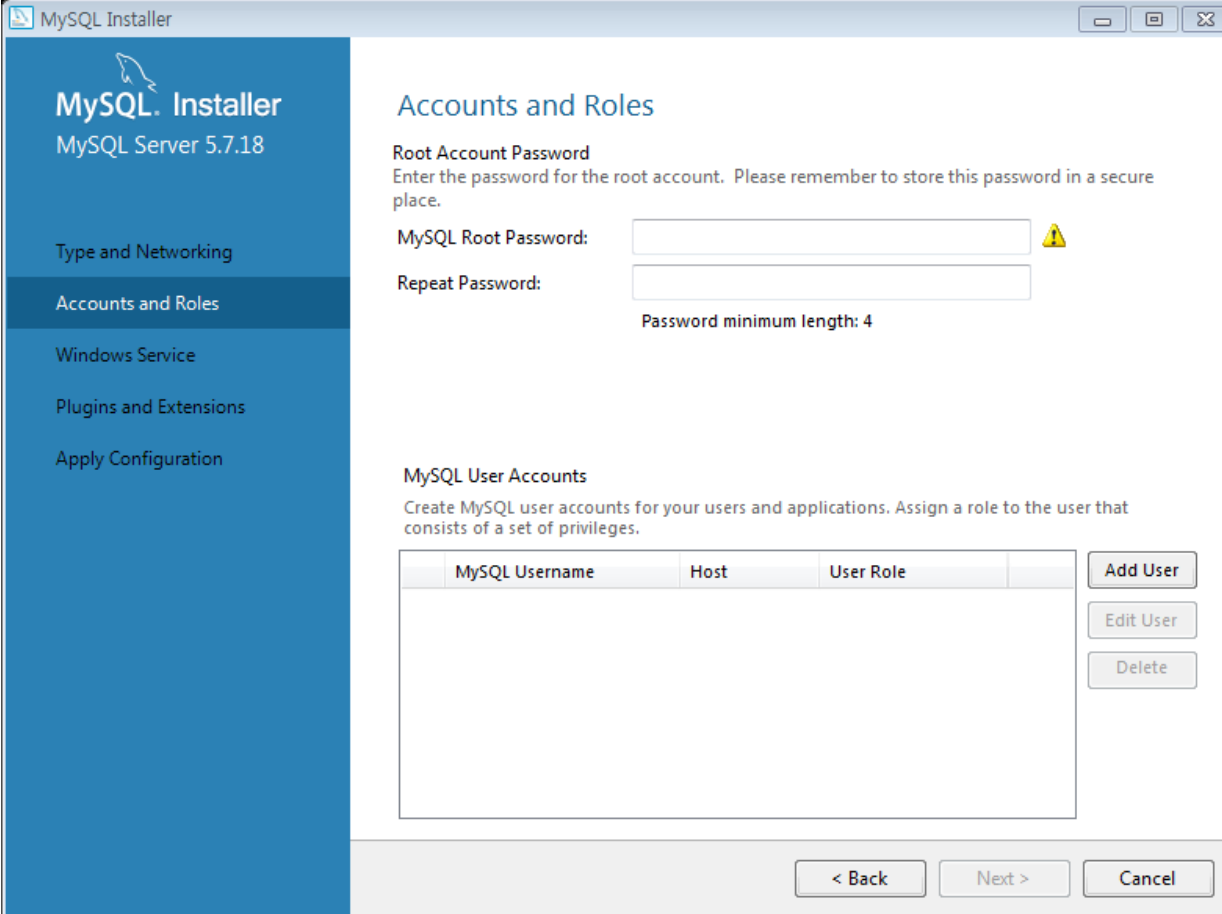
· 필요한 프로그램 설치

- 필수 프로그램: MySQL Server



MySQL 설치(for Windows)

- Root 비밀번호 입력



The screenshot shows the 'Accounts and Roles' step of the MySQL Installer for Windows 5.7.18. The left sidebar contains navigation links: 'Type and Networking', 'Accounts and Roles' (selected), 'Windows Service', 'Plugins and Extensions', and 'Apply Configuration'. The main area is titled 'Accounts and Roles' and contains two sections. The first section, 'Root Account Password', instructs the user to enter a password for the root account, with a warning icon indicating a minimum length of 4 characters. It includes input fields for 'MySQL Root Password' and 'Repeat Password'. The second section, 'MySQL User Accounts', instructs the user to create MySQL user accounts and assign roles. It features a table with columns for 'MySQL Username', 'Host', and 'User Role', and buttons for 'Add User', 'Edit User', and 'Delete'. At the bottom, there are navigation buttons: '< Back', 'Next >', and 'Cancel'.

MySQL Installer

MySQL Server 5.7.18

Type and Networking

Accounts and Roles

Windows Service

Plugins and Extensions

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:

Password minimum length: 4

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 Username	Host	User Role
----------------	------	-----------

Add User

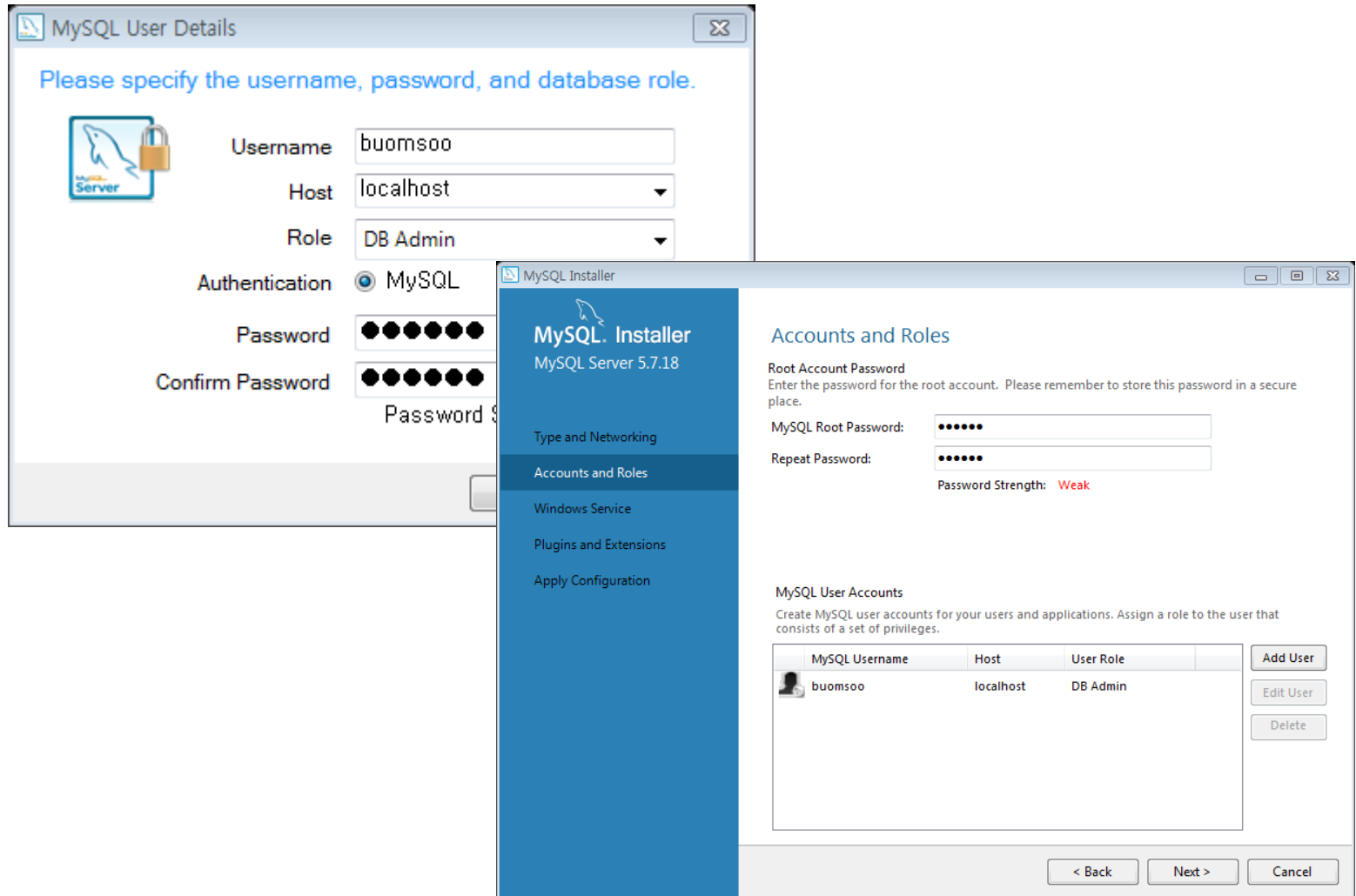
Edit User

Delete

< Back Next > Cancel

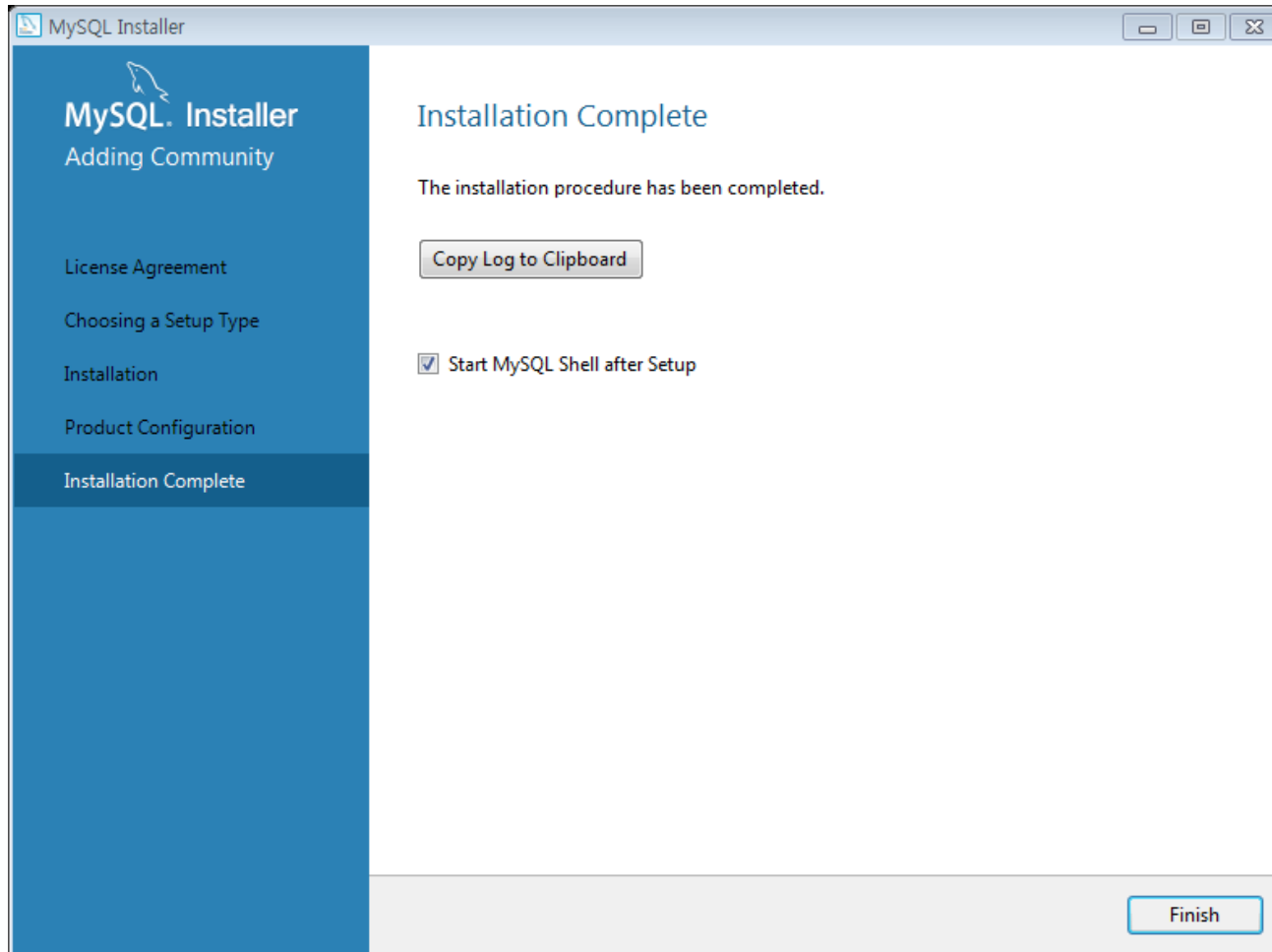
MySQL 설치(for Windows)

- 기타 사용자 등록(설치 후에도 수정 가능)



MySQL 설치(for Windows)

· 설치 완료



MySQL 설치하기(for MAC OSX)

- MySQL을 맥에 설치하기 위해서 우선 HomeBrew를 먼저 설치한다

- 터미널에 아래 명령어를 입력한다

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

- 참고: https://brew.sh/index_ko.html

- HomeBrew는 Apple에서 번들로 제공되지 않는 다양한 패키지 관리를 지원한다

- 참고: <https://github.com/helloheesu/SecretlyGreatly/wiki/%EB%A7%A5%EC%97%90%EC%84%9C-mysql-%EC%84%A4%EC%B9%98-%ED%9B%84-%ED%99%98%EA%B2%BD%EC%84%A4%EC%A0%95%ED%95%98%EA%B8%B0>
- 영상: <https://www.youtube.com/watch?v=xX9W5dmEpO0>

MySQL 설치하기(for MAC OSX)

- brew 명령어를 이용해 MySQL을 설치한다

- 터미널에 아래 명령어를 입력한다

```
> brew install mysql
```

- MySQL 구동하기

- MySQL 시작

```
> mysql.server start
```

- Root 비밀번호 설정하기

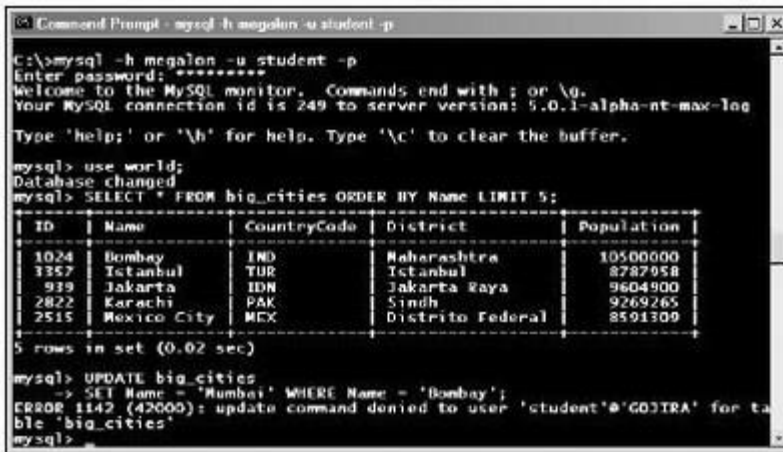
```
> mysql_secure_installation
```


SQL & MySQL basics



MySQL Monitor

- MySQL 서버의 번들로 제공되는 기본 프로그램
- 명령어 기반으로, 커맨드 라인 인터페이스를 활용하여 MySQL 데이터베이스를 이용할 수 있다
 - GUI(Graphic User Interface)가 제공되지 않는다는 단점이 있다

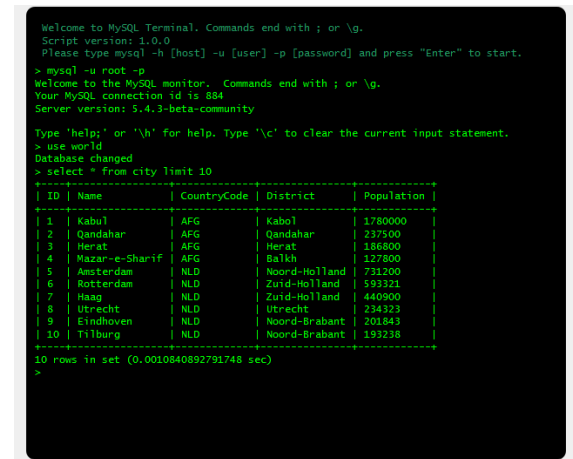


```
C:\>mysql -h megalon -u student -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 249 to server version: 5.0.1-alpha-nt-max-log

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> use world;
Database changed
mysql> SELECT * FROM big_cities ORDER BY Name LIMIT 5;
+----+-----+-----+-----+-----+
| ID | Name | CountryCode | District | Population |
+----+-----+-----+-----+-----+
| 1024 | Bombay | IND | Maharashtra | 10500000 |
| 1357 | Istanbul | TUR | Istanbul | 8787958 |
| 939 | Jakarta | IDN | Jakarta Raya | 9604900 |
| 2822 | Karachi | PAK | Sindh | 9269265 |
| 2515 | Mexico City | MEX | Distrito Federal | 8591309 |
+----+-----+-----+-----+-----+
5 rows in set (0.02 sec)

mysql> UPDATE big_cities
-> SET Name = 'Mumbai' WHERE Name = 'Bombay';
ERROR 1142 (42000): update command denied to user 'student'@'GO3TRA' for table 'big_cities'
mysql>
```



```
Welcome to MySQL Terminal. Commands end with ; or \g.
Script version: 1.0.0
Please type mysql -h [host] -u [user] -p [password] and press "Enter" to start.

> mysql -u root -p
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 884
Server version: 5.4.3-beta-community

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.
> use world
Database changed
> select * from city limit 10
+----+-----+-----+-----+-----+
| 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 |
+----+-----+-----+-----+-----+
10 rows in set (0.0010840892791748 sec)
>
```


MySQL Monitor

- MySQL Monitor에 접속하기

- 커맨드 라인에 아래와 같이 입력한다

```
> mysql -u[아이디] -p  
Enter password: [비밀번호 입력]
```

- 사용자를 따로 생성하지 않았으면 아이디는 '**root**'(superuser)를 사용한다

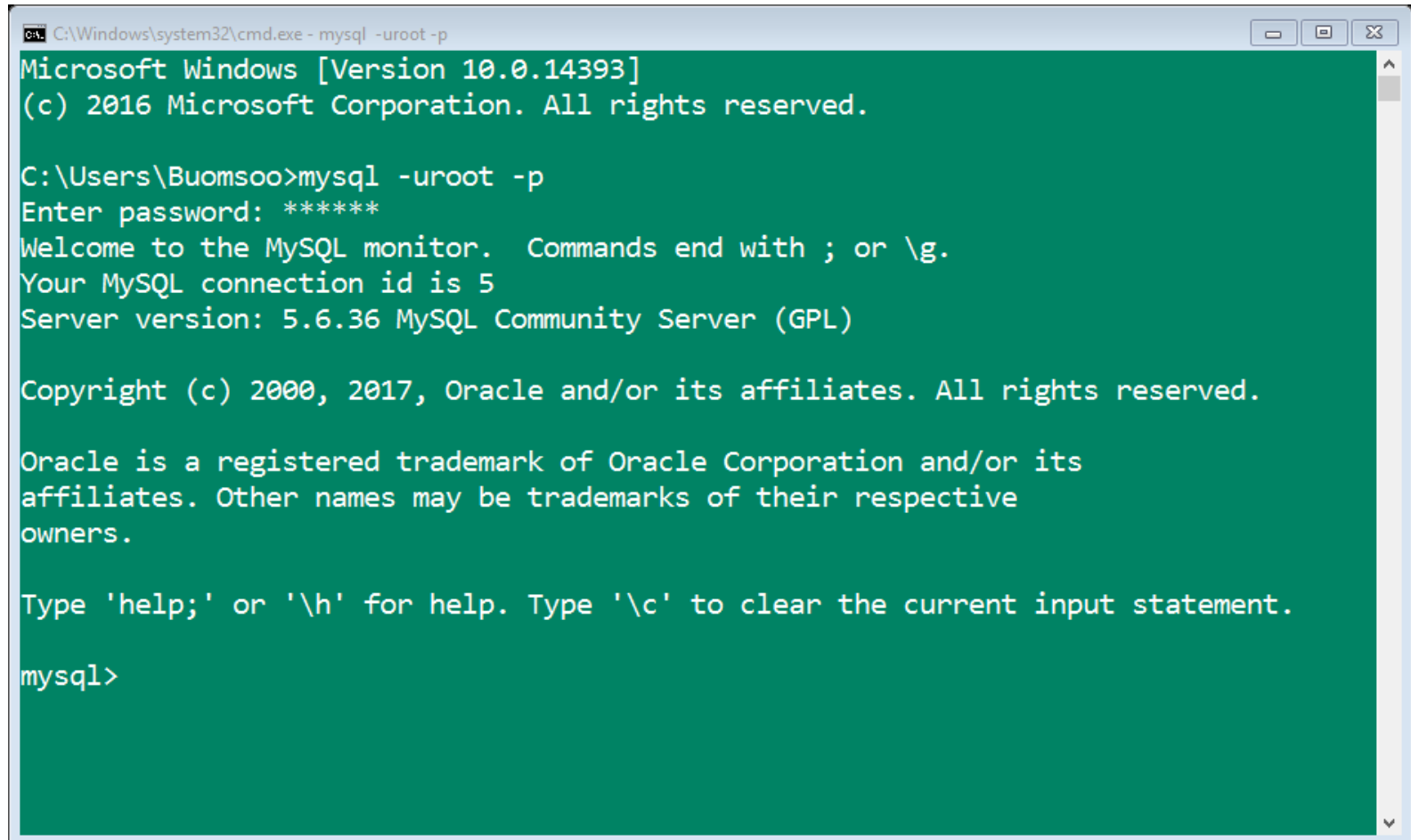
- 비밀번호는 설치하면서 설정 → 꼭 기억해 둔다!

- 커맨드 라인에서 MySQL이 실행이 안될 경우:

- 환경변수 설정을 확인해 본다(MySQL이 설치된 디렉토리)

MySQL Monitor

- MySQL Monitor에 접속하기

A screenshot of a Windows command prompt window. The title bar shows the path 'C:\Windows\system32\cmd.exe - mysql -uroot -p'. The window has a green background. The text inside shows the user logging in as 'root' with a password. The MySQL Monitor interface displays the connection ID (5) and the server version (5.6.36). It also includes copyright information for Oracle and instructions on how to use help or clear the input.

```
C:\Windows\system32\cmd.exe - mysql -uroot -p
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\Buomsoo>mysql -uroot -p
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 5
Server version: 5.6.36 MySQL Community Server (GPL)

Copyright (c) 2000, 2017, Oracle and/or its affiliates. All rights reserved.

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

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

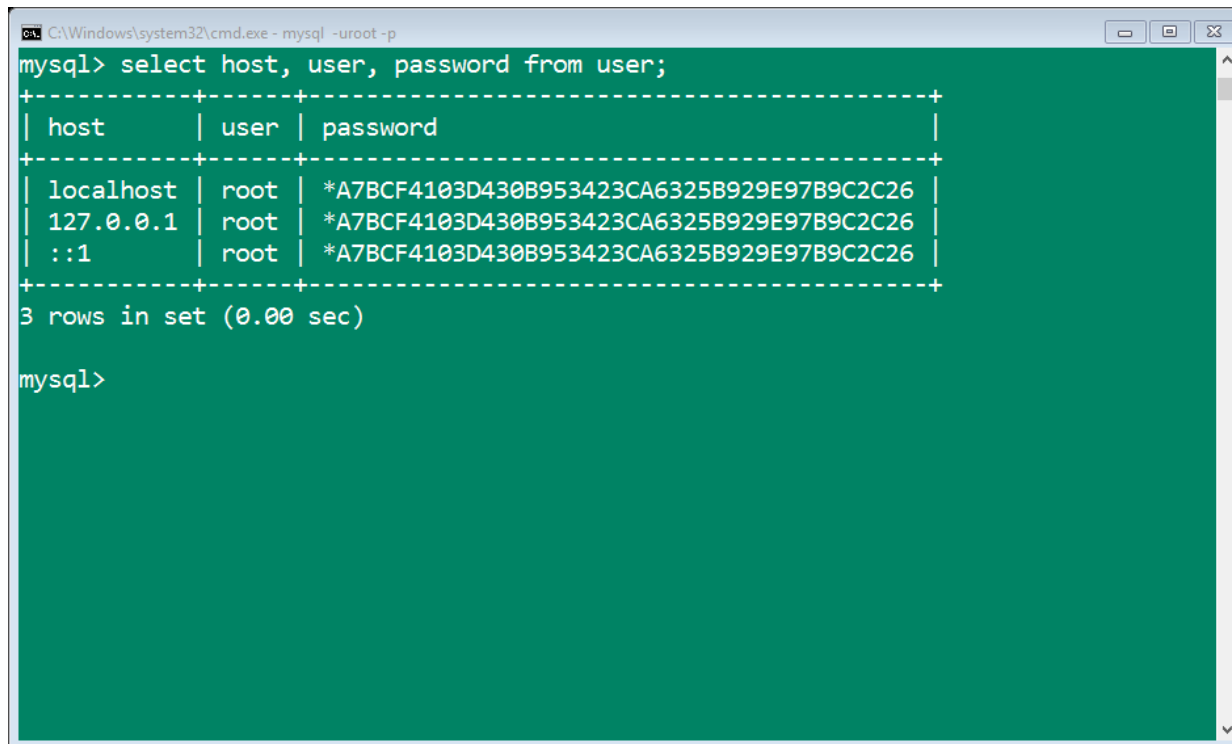
mysql>
```

사용자 관리하기

- 현재 사용자 목록 조회하기

- root 사용자로 접속 후 아래 명령어 입력

```
> use mysql;  
> select host, user, password from user;
```



The screenshot shows a Windows command prompt window titled "C:\Windows\system32\cmd.exe - mysql -uroot -p". The prompt is "mysql>". The user has entered the command "select host, user, password from user;". The output is a table with three columns: host, user, and password. The table contains three rows of data, all with the same password hash. Below the table, it says "3 rows in set (0.00 sec)". The prompt is now "mysql>".

```
mysql> select host, user, password from user;  
+-----+-----+-----+  
| host      | user  | password |  
+-----+-----+-----+  
| localhost | root  | *A7BCF4103D430B953423CA6325B929E97B9C2C26 |  
| 127.0.0.1 | root  | *A7BCF4103D430B953423CA6325B929E97B9C2C26 |  
| ::1       | root  | *A7BCF4103D430B953423CA6325B929E97B9C2C26 |  
+-----+-----+-----+  
3 rows in set (0.00 sec)  
  
mysql>
```

사용자 관리하기

- 새로운 사용자 생성하기

- ID 와 PASSWORD로 식별되는 사용자 생성

```
> CREATE USER '[사용자 ID]'@LOCALHOST IDENTIFIED BY '[비밀번호]';
```

- 권한 부여하기

```
> GRANT ALL PRIVILEGES ON *.* TO '[사용자 이름]'@'localhost'  
WITH GRANT OPTION;
```

- **TODO:** 사용자를 자유롭게 새로이 생성해 본다

- 예시: ID는 'buomsoo', 비밀번호는 '12345'인 사용자 생성

```
1 CREATE USER 'buomsoo'@localhost IDENTIFIED BY '12345';
```

사용자 관리하기

- 새로운 사용자 생성하기

- ID 와 PASSWORD로 식별되는 사용자 생성

```
C:\Windows\system32\cmd.exe - mysql -uroot -p
mysql> select host, user, password from user;
+-----+-----+-----+
| host      | user  | password                                     |
+-----+-----+-----+
| localhost | root  | *A7BCF4103D430B953423CA6325B929E97B9C2C26 |
| 127.0.0.1 | root  | *A7BCF4103D430B953423CA6325B929E97B9C2C26 |
| ::1       | root  | *A7BCF4103D430B953423CA6325B929E97B9C2C26 |
| localhost | buomsoo | *00A51F3F48415C7D4E8908980D443C29C69B60C9 |
+-----+-----+-----+
4 rows in set (0.00 sec)

mysql>
```

데이터베이스 관리하기

- 현재 DBMS에 존재하는 데이터베이스 조회하기

> **SHOW DATABASES;**

```
C:\Windows\system32\cmd.exe - mysql -uroot -p

mysql> show databases;
+-----+
| Database                |
+-----+
| information_schema       |
| mysql                    |
| performance_schema       |
| test                     |
+-----+
4 rows in set (0.04 sec)

mysql>
```

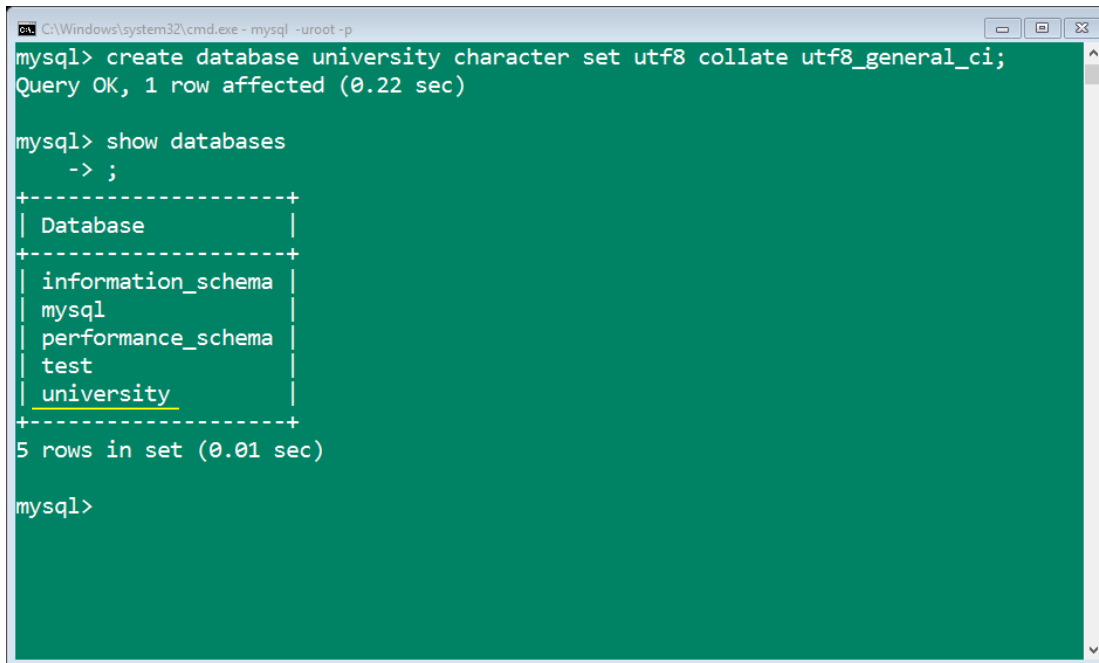
현재 시스템 내의
데이터베이스 목록

데이터베이스 관리하기

- 새로운 데이터베이스 생성하기

```
> CREATE DATABASE [데이터베이스 이름] CHARACTER SET utf8  
COLLATE utf8_general_ci;
```

- **TODO:** 'university' 데이터베이스를 생성해 본다



```
C:\Windows\system32\cmd.exe - mysql -uroot -p  
mysql> create database university character set utf8 collate utf8_general_ci;  
Query OK, 1 row affected (0.22 sec)  
  
mysql> show databases  
-> ;  


| Database           |
|--------------------|
| information_schema |
| mysql              |
| performance_schema |
| test               |
| <u>university</u>  |

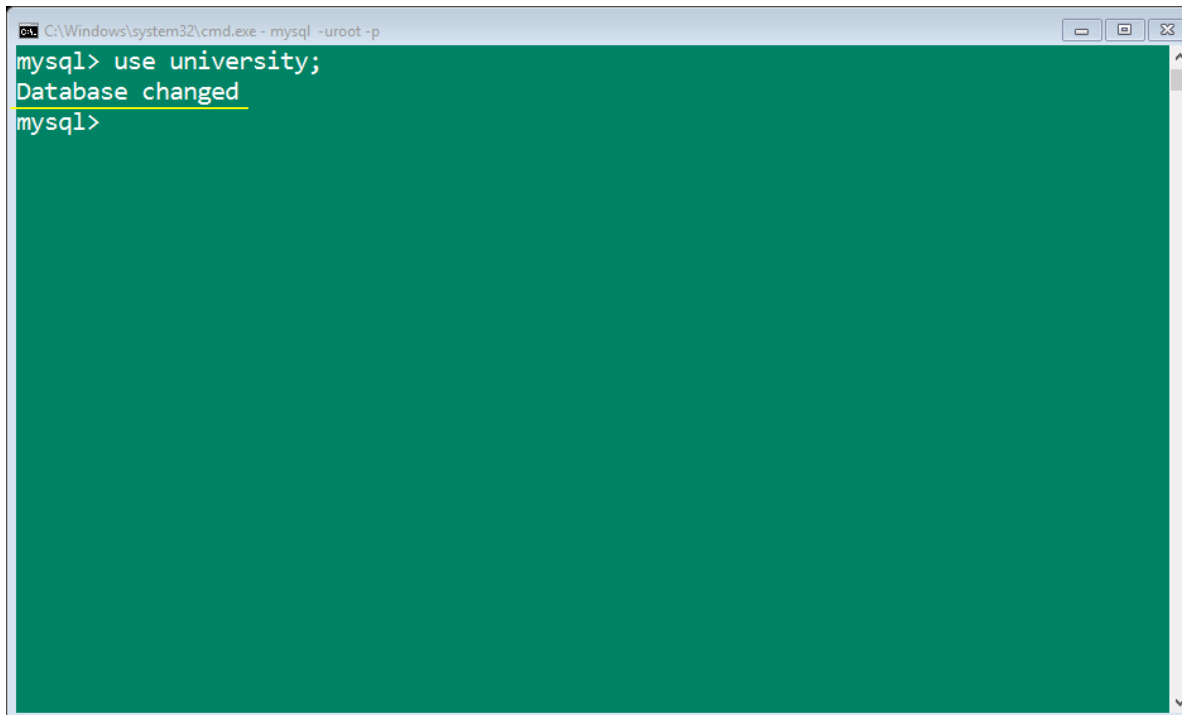
  
5 rows in set (0.01 sec)  
  
mysql>
```

데이터베이스 관리하기

- 데이터베이스 선택(접근)하기

```
> USE [데이터베이스 이름];
```

- **TODO:** 'university' 데이터베이스를 선택한다



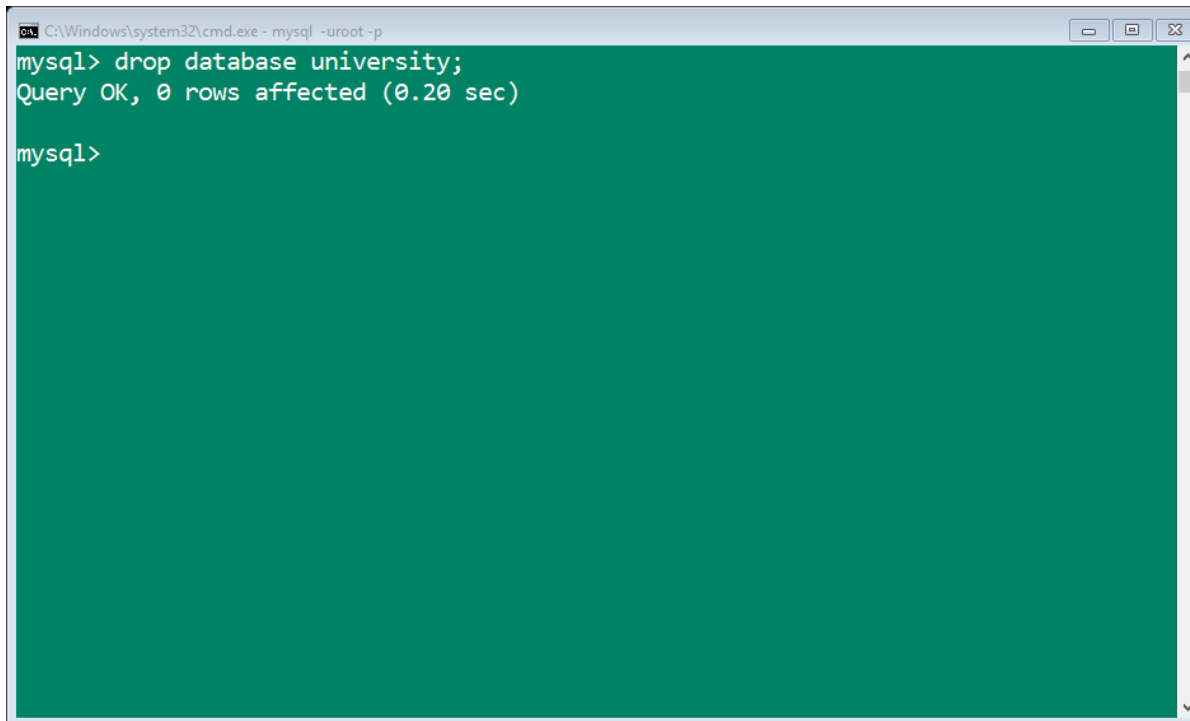
```
C:\Windows\system32\cmd.exe - mysql -uroot -p
mysql> use university;
Database changed
mysql>
```


데이터베이스 관리하기

- 데이터베이스 삭제하기

```
> DROP DATABASE [데이터베이스 이름];
```

- **TODO:** 'university' 데이터베이스를 삭제한다



```
C:\Windows\system32\cmd.exe - mysql -uroot -p
mysql> drop database university;
Query OK, 0 rows affected (0.20 sec)

mysql>
```

테이블 관리하기

- 새로운 테이블 생성하기

```
> CREATE TABLE `[테이블 이름]` (  
    `[컬럼1 이름]` `[컬럼1 데이터 타입]`,  
    ...  
    `[컬럼n 이름]` `[컬럼n 데이터 타입]`,  
) ENGINE = innodb;
```

- **TODO:** 'university' 데이터베이스를 다시 생성하고 'student' 테이블을 생성해 본다

- 컬럼1: 이름은 'name', 데이터 타입은 'varchar(20)'
- 컬럼2: 이름은 'id', 데이터 타입은 'int(5)'
- 컬럼3: 이름은 'department', 데이터 타입은 'varchar(30)'
- 컬럼4: 이름은 'birthday', 데이터 타입은 'date'

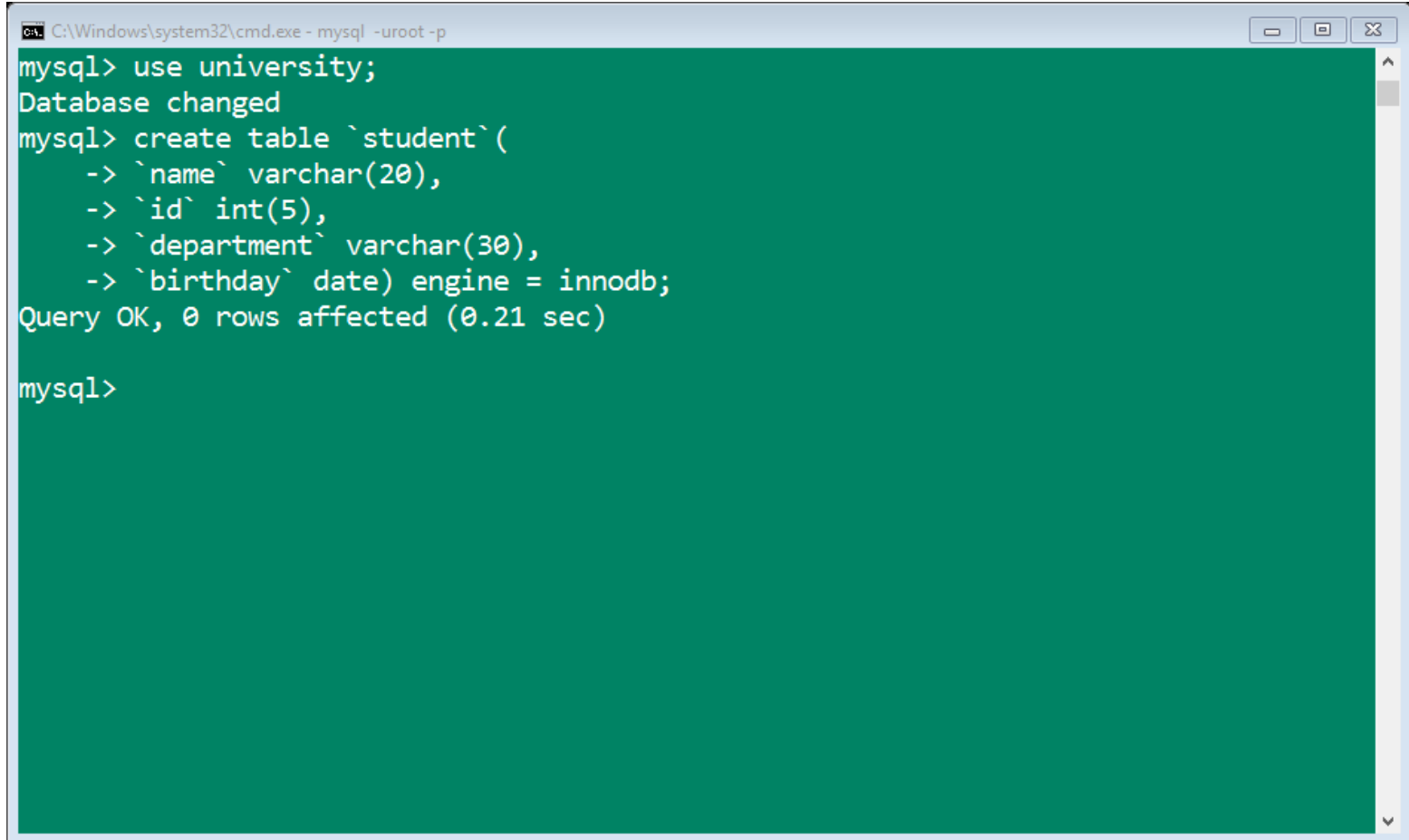
테이블 관리하기

· 참고: 데이터 타입

타입	설명	예시
CHAR()	고정 문자(0-255)	성명
VARCHAR()	가변 문자(0-65535)	상품 리뷰
INT()	정수형	나이
DATE	날짜(YYYY-MM-DD)	생년월일
ENUM(V1, V2, ... VN)	V1,...,VN 중 하나의 값	상품 타입(사이즈 등)

테이블 관리하기

- 새로운 테이블 생성하기

A screenshot of a Windows command prompt window with a green background. The title bar at the top reads "C:\Windows\system32\cmd.exe - mysql -uroot -p". The command history shows the user switching to the 'university' database and then creating a new table named 'student'. The table has four columns: 'name' (varchar(20)), 'id' (int(5)), 'department' (varchar(30)), and 'birthday' (date). The table is created using the InnoDB engine. The output shows "Query OK, 0 rows affected (0.21 sec)". The prompt ends with "mysql>".

```
C:\Windows\system32\cmd.exe - mysql -uroot -p
mysql> use university;
Database changed
mysql> create table `student` (
  -> `name` varchar(20),
  -> `id` int(5),
  -> `department` varchar(30),
  -> `birthday` date) engine = innodb;
Query OK, 0 rows affected (0.21 sec)

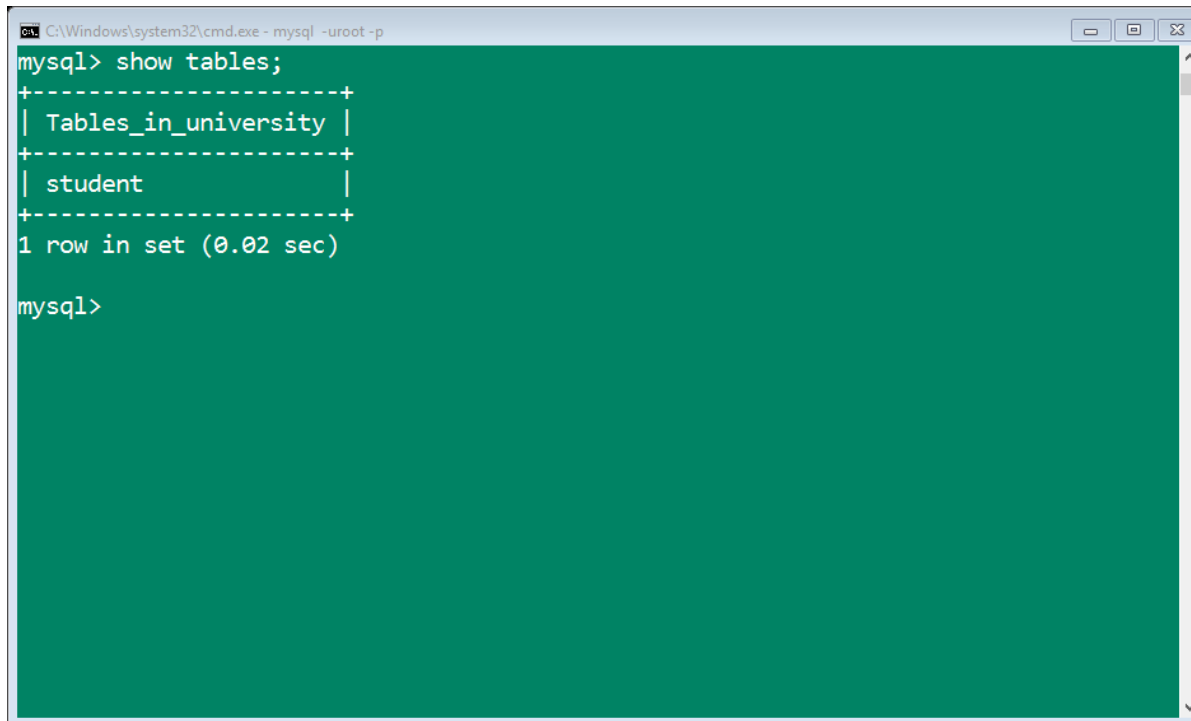
mysql>
```

테이블 관리하기

- 테이블 목록 보기

```
> SHOW TABLES;
```

- **TODO:** 현재 생성된 테이블 목록을 확인한다



```
C:\Windows\system32\cmd.exe - mysql -uroot -p
mysql> show tables;
+-----+
| Tables_in_university |
+-----+
| student               |
+-----+
1 row in set (0.02 sec)

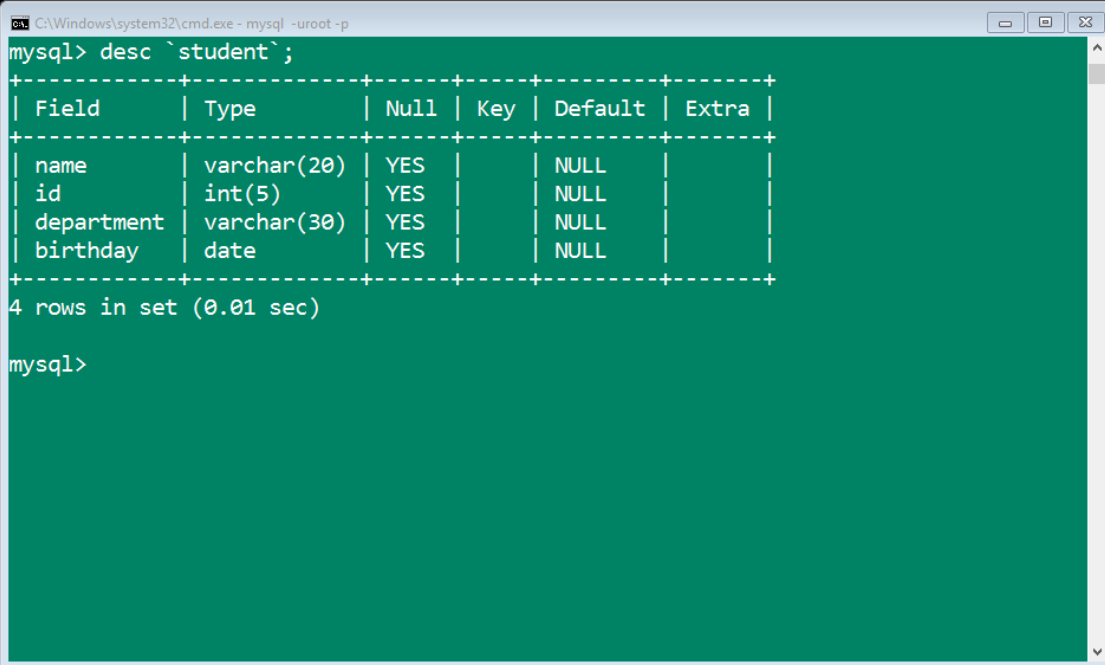
mysql>
```

테이블 관리하기

- 테이블 스키마 보기

```
> DESC `[테이블 이름]`;
```

- **TODO:** 'student' 테이블의 스키마를 확인한다




```
C:\Windows\system32\cmd.exe - mysql -uroot -p
mysql> desc `student`;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| name       | varchar(20)   | YES  |     | NULL    |       |
| id         | int(5)        | YES  |     | NULL    |       |
| department | varchar(30)   | YES  |     | NULL    |       |
| birthday   | date          | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
4 rows in set (0.01 sec)

mysql>
```

테이블 관리하기



· 참고: 테이블 스키마[source: <https://stackoverflow.com/questions/298739/what-is-the-difference-between-a-schema-and-a-table-and-a-database>]

- 테이블 스키마(Relation schema): 테이블에 대한 논리적 정의("defines what the name of the table is, and what the name and type of each column is")
- 데이터베이스 스키마(Database schema): 전체 데이터베이스에 포함된 테이블 스키마의 총집합("the collection of relation schemas for a whole database")



schema : database : table :: floor plan : house : room


166



[share](#) [improve this answer](#)

edited Jun 1 at 13:36

answered Nov 18 '08 at 13:45

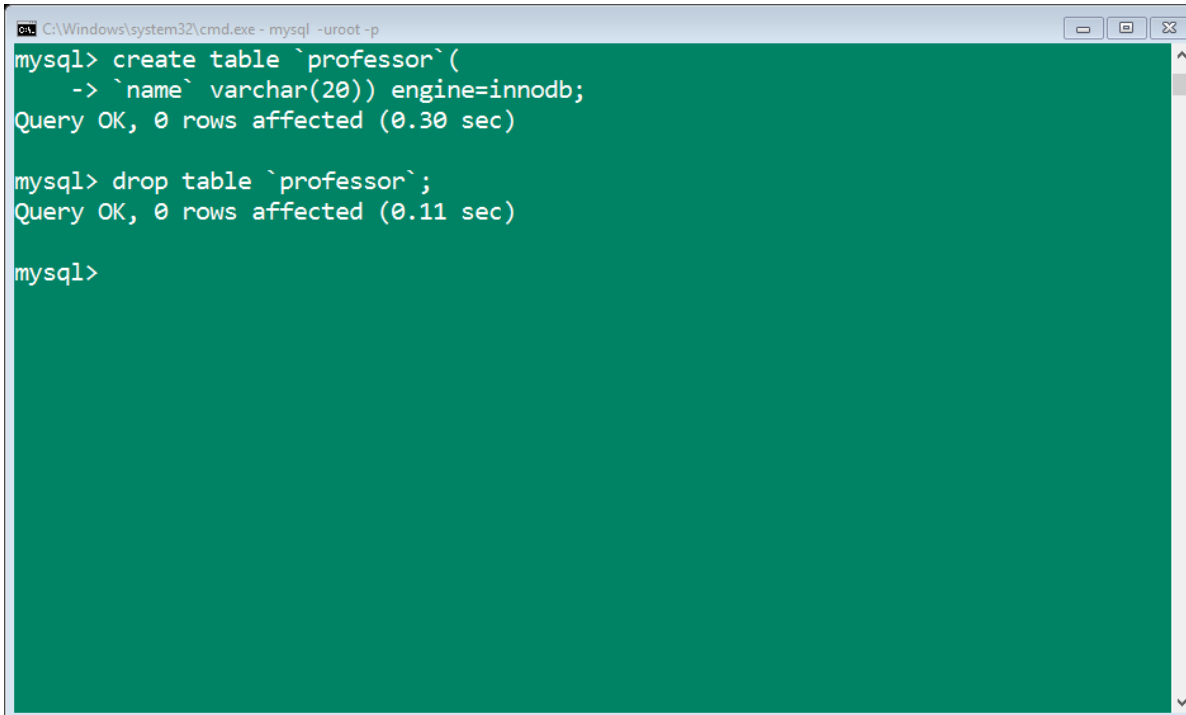
 **MusiGenesis**
58.2k ● 32 ● 149 ● 291

테이블 관리하기

- 테이블 제거하기

```
> DROP TABLE `[테이블 이름]`
```

- **TODO:** 임의의 테이블을 하나 만들고 삭제한다



```
C:\Windows\system32\cmd.exe - mysql -uroot -p
mysql> create table `professor` (
  -> `name` varchar(20)) engine=innodb;
Query OK, 0 rows affected (0.30 sec)

mysql> drop table `professor`;
Query OK, 0 rows affected (0.11 sec)

mysql>
```


데이터 관리하기

- 테이블에 로우(row) 삽입하기

> **INSERT INTO** `[테이블 이름]` **VALUES** (값1, 값2, 값3, ...)

> **INSERT INTO** `[테이블 이름]` (컬럼1, 컬럼2, ...) **VALUES** (값1, 값2, ...)

- 주의: 값을 입력할 때에는 `가 아닌 '를 사용한다



데이터 관리하기

· 테이블에 로우(row) 삽입하기

- **TODO:** 아래 데이터를 student 테이블에 삽입한다

name	id	department	birthday
Brad Pitt	13001	Humanities	1963-12-18
Leonardo Dicaprio	13002	Social sciences	1974-11-11
Tom Cruise	13003	Engineering	1962-07-03
John Travolta	13004	Engineering	1954-02-18
Kevin Costner	13005	Humanities	1955-01-18

데이터 관리하기

- 테이블에 로우(row) 삽입하기

- **TODO:** 아래 데이터를 student 테이블에 삽입한다

```
C:\Windows\system32\cmd.exe - mysql -uroot -p
mysql> insert into `student` values ('Brad Pitt', '13001', 'Humanities', '1963-12-18');
Query OK, 1 row affected (0.10 sec)

mysql> insert into `student` values ('Leonardo Dicaprio', '13002', 'Social sciences', '1974-11-11');
Query OK, 1 row affected (0.03 sec)

mysql> insert into `student` values ('Tom Cruise', '13003', 'Engineering', '1962-07-03');
Query OK, 1 row affected (0.03 sec)

mysql> insert into `student` values ('John Travolta', '13004', 'Engineering', '1954-02-18');
Query OK, 1 row affected (0.06 sec)

mysql> insert into `student` values ('Kevin Costner', '13005', 'Humanities', '1955-01-18');
Query OK, 1 row affected (0.04 sec)

mysql>
```

데이터 관리하기

- 테이블에 로우(row) 삽입하기

- **TODO:** 아래 데이터를 student 테이블에 삽입한다

```
C:\Windows\system32\cmd.exe - mysql -uroot -p
mysql> select * from student;
+-----+-----+-----+-----+
| name          | id    | department | birthday |
+-----+-----+-----+-----+
| Brad Pitt     | 13001 | Humanities | 1963-12-18 |
| Leonardo Dicaprio | 13002 | Social sciences | 1974-11-11 |
| Tom Cruise    | 13003 | Engineering | 1962-07-03 |
| John Travolta | 13004 | Engineering | 1954-02-18 |
| Kevin Costner  | 13005 | Humanities | 1955-01-18 |
+-----+-----+-----+-----+
5 rows in set (0.03 sec)

mysql>
```

데이터 관리하기

· 데이터 변경하기

```
> UPDATE `[테이블 이름]` SET 컬럼N = 컬럼N의 값
```

```
> UPDATE `[테이블 이름]` SET 컬럼N = 컬럼N의 값 WHERE 대상이 될 컬럼 이름  
= 컬럼의 값
```

- **TODO:** Brad Pitt의 생년월일을 '1963-12-24'로 변경한다

- **TODO:** Tom Cruise의 이름을 'Tom Hanks'로 변경한다

데이터 관리하기

- 데이터 변경하기

```
C:\Windows\system32\cmd.exe - mysql -uroot -p

mysql> update `student` set birthday = '1963-12-24' where id = '13001';
Query OK, 0 rows affected (0.03 sec)
Rows matched: 1  Changed: 0  Warnings: 0

mysql> update `student` set name = 'Tom Hanks' where id = '13003';
Query OK, 0 rows affected (0.03 sec)
Rows matched: 1  Changed: 0  Warnings: 0

mysql> select * from student;
+-----+-----+-----+-----+
| name          | id    | department | birthday |
+-----+-----+-----+-----+
| Brad Pitt     | 13001 | Humanities | 1963-12-24 |
| Leonardo Dicaprio | 13002 | Social sciences | 1974-11-11 |
| Tom Hanks     | 13003 | Engineering | 1962-07-03 |
| John Travolta | 13004 | Engineering | 1954-02-18 |
| Kevin Costner | 13005 | Humanities | 1955-01-18 |
+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql>
```

데이터 관리하기

· 참고: 일차 키(Primary key)

- 'student' 테이블에서 이름(name), 단과대학(department), 생년월일(birthday)이 같은 사람은 여럿 존재할 수 있지만, 일반적으로 대학교 데이터베이스에서 학번(id)은 각 학생이 고유한 값을 갖는다
- 즉, 학번을 이용해 각 학생을 고유하게 식별해낼 수가 있다. 이 경우, 이를 일차 키(primary key)로 지정할 수 있다 ("candidate key whose values are used to *identify* tuples in the relation").
- 일차 키 설정하기

```
1 create table `student`(  
2     `name` varchar(20),  
3     `id` int(5) PRIMARY KEY,  
4     `department` varchar(30),  
5     `birthday` date) engine = innodb;
```

데이터 관리하기

- 참고: 일차 키(Primary key)

```
C:\Windows\system32\cmd.exe - mysql -uroot -p
mysql> desc student;
```

Field	Type	Null	Key	Default	Extra
name	varchar(20)	YES		NULL	
id	int(5)	NO	PRI	NULL	
department	varchar(30)	YES		NULL	
birthday	date	YES		NULL	

```
4 rows in set (0.02 sec)

mysql>
```

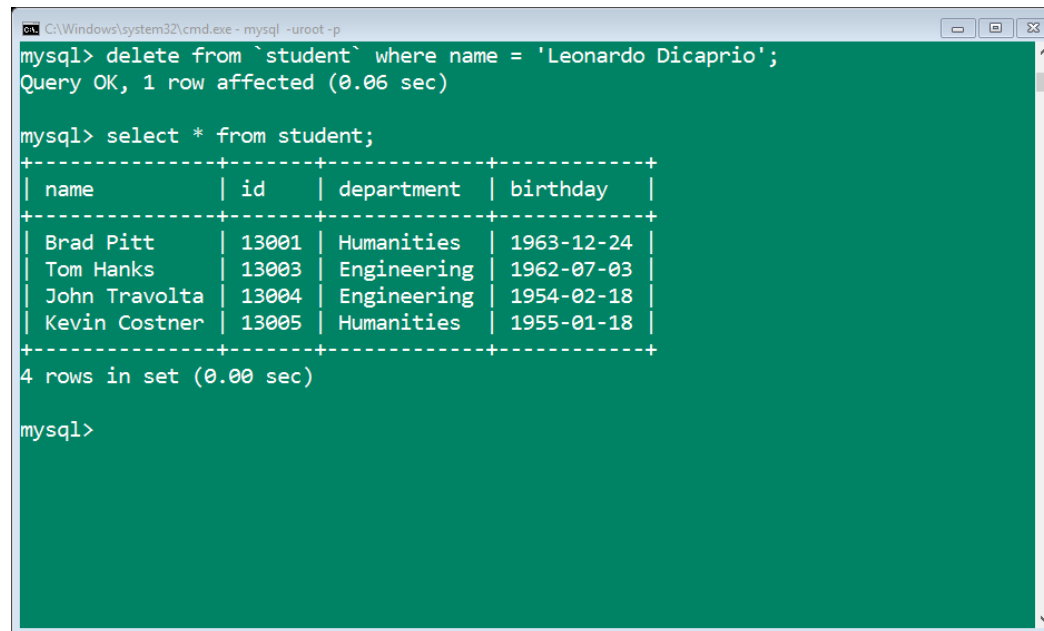

데이터 관리하기

· 데이터 삭제하기

> **DELETE FROM** `[테이블 이름]`;

> **DELETE FROM** `[테이블 이름]` **WHERE** 삭제하려는 컬럼 이름 = 값;

- **TODO:** 이름이 'Leonardo Dicaprio'인 행의 데이터를 삭제한다



```
C:\Windows\system32\cmd.exe - mysql -uroot -p
mysql> delete from `student` where name = 'Leonardo Dicaprio';
Query OK, 1 row affected (0.06 sec)

mysql> select * from student;
+-----+-----+-----+-----+
| name      | id    | department | birthday |
+-----+-----+-----+-----+
| Brad Pitt  | 13001 | Humanities  | 1963-12-24 |
| Tom Hanks  | 13003 | Engineering | 1962-07-03 |
| John Travolta | 13004 | Engineering | 1954-02-18 |
| Kevin Costner | 13005 | Humanities  | 1955-01-18 |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)

mysql>
```

데이터 관리하기

· 데이터 조회하기

```
> SELECT 컬럼 이름1, 컬럼 이름 2, ... 컬럼 이름 N  
      FROM `[테이블 이름]`  
      GROUP BY 컬럼 이름  
      ORDER BY 컬럼 이름  
      LIMIT 시작 행 번호, 조회할 행의 수
```

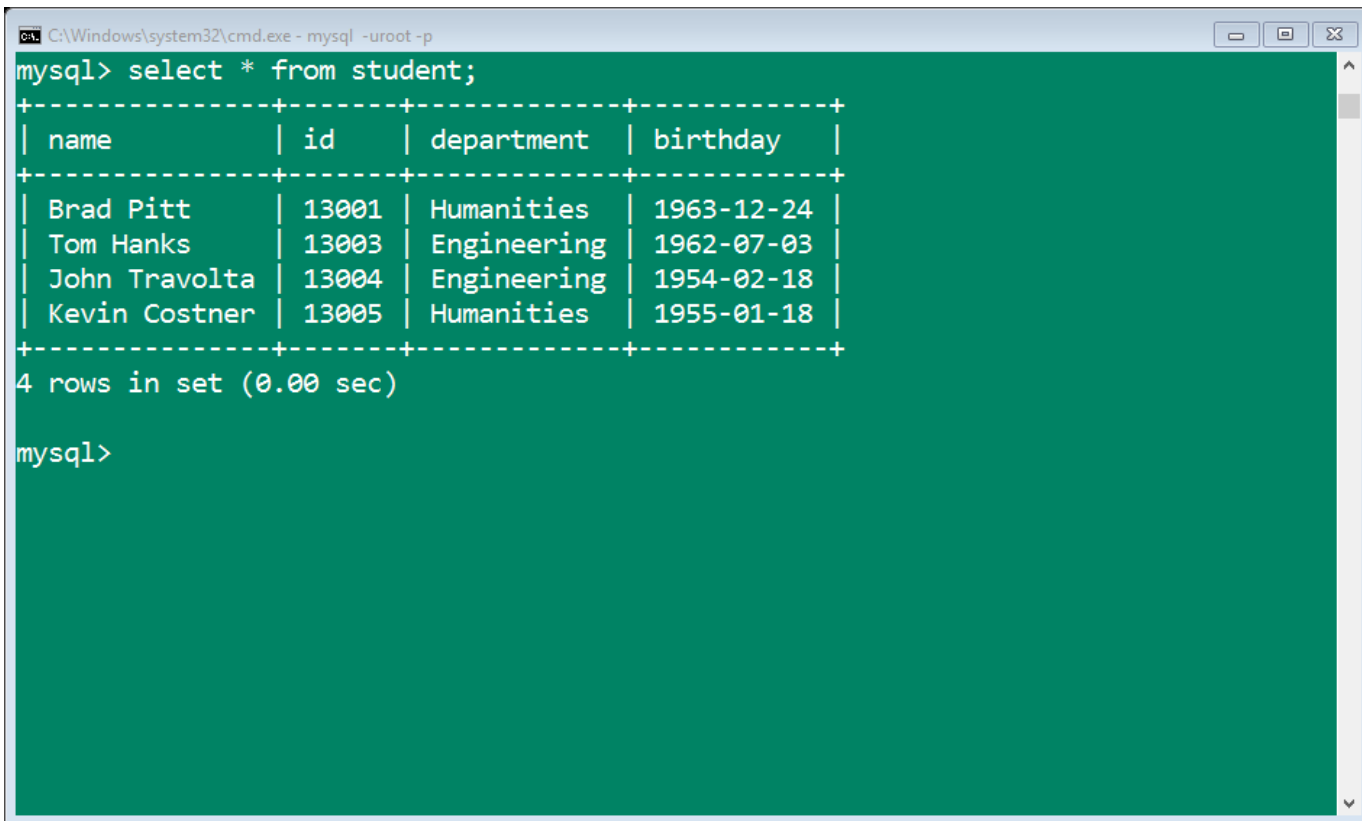
```
> SELECT *  
      FROM `[테이블 이름]`  
      GROUP BY 컬럼 이름  
      ORDER BY 컬럼 이름  
      LIMIT 시작 행 번호, 조회할 행의 수
```

- 전체 데이터를 조회하려면 와일드카드(*)를 활용한다

데이터 관리하기

- 데이터 조회하기

- **TODO:** student 테이블의 전체 데이터를 조회해 본다



```
C:\Windows\system32\cmd.exe - mysql -uroot -p
mysql> select * from student;
+-----+-----+-----+-----+
| name      | id    | department | birthday |
+-----+-----+-----+-----+
| Brad Pitt  | 13001 | Humanities  | 1963-12-24 |
| Tom Hanks  | 13003 | Engineering | 1962-07-03 |
| John Travolta | 13004 | Engineering | 1954-02-18 |
| Kevin Costner | 13005 | Humanities  | 1955-01-18 |
+-----+-----+-----+-----+
4 rows in set (0.00 sec)

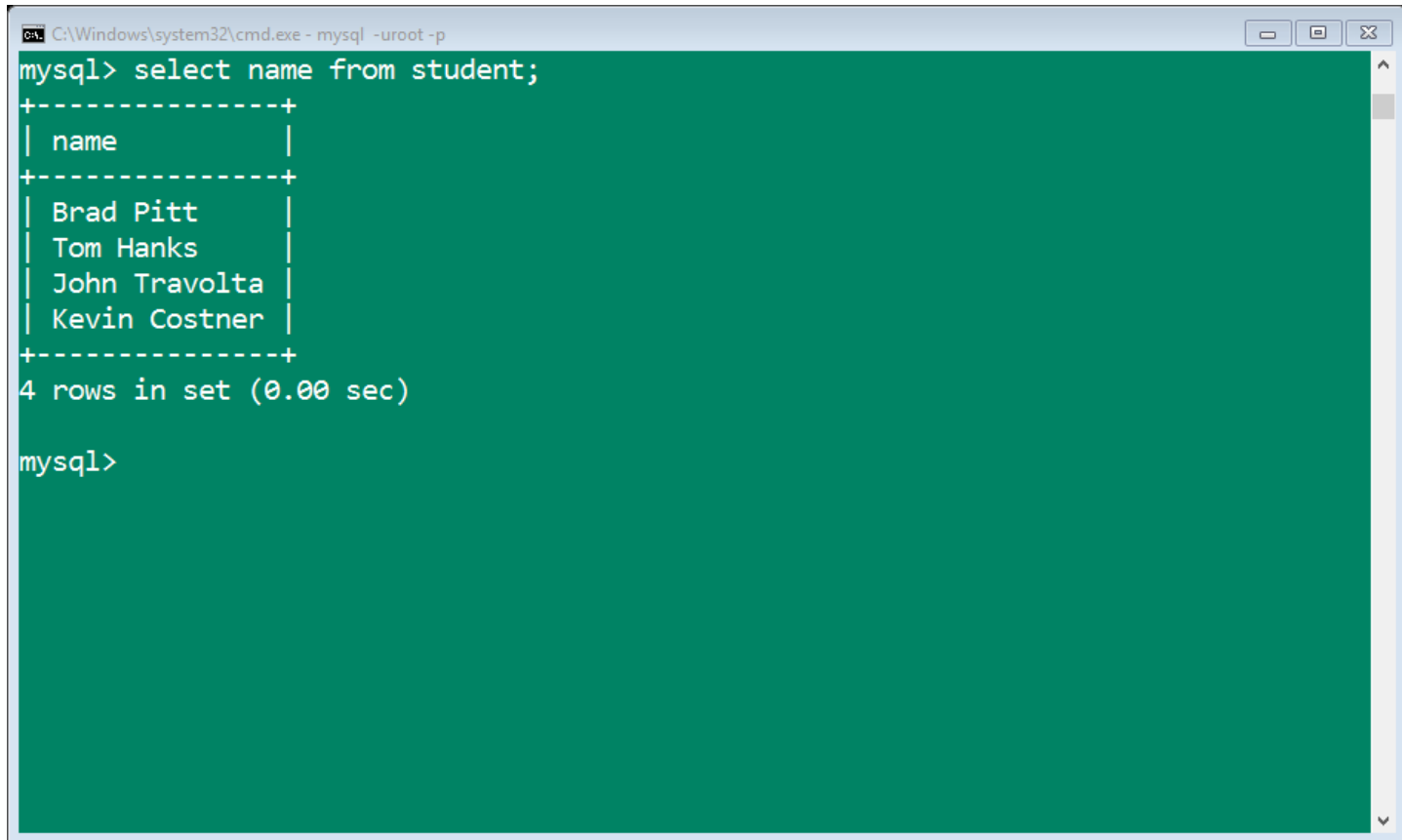
mysql>
```

name	id	department	birthday
Brad Pitt	13001	Humanities	1963-12-24
Tom Hanks	13003	Engineering	1962-07-03
John Travolta	13004	Engineering	1954-02-18
Kevin Costner	13005	Humanities	1955-01-18

데이터 관리하기

- 데이터 조회하기

- **TODO:** student 테이블의 이름(name)만 조회해 본다



A screenshot of a Windows command prompt window titled "C:\Windows\system32\cmd.exe - mysql -uroot -p". The prompt is "mysql>". The user has entered the command "select name from student;". The output is a table with one column named "name" and four rows of data: "Brad Pitt", "Tom Hanks", "John Travolta", and "Kevin Costner". The output is formatted with a header row and a footer row indicating "4 rows in set (0.00 sec)".

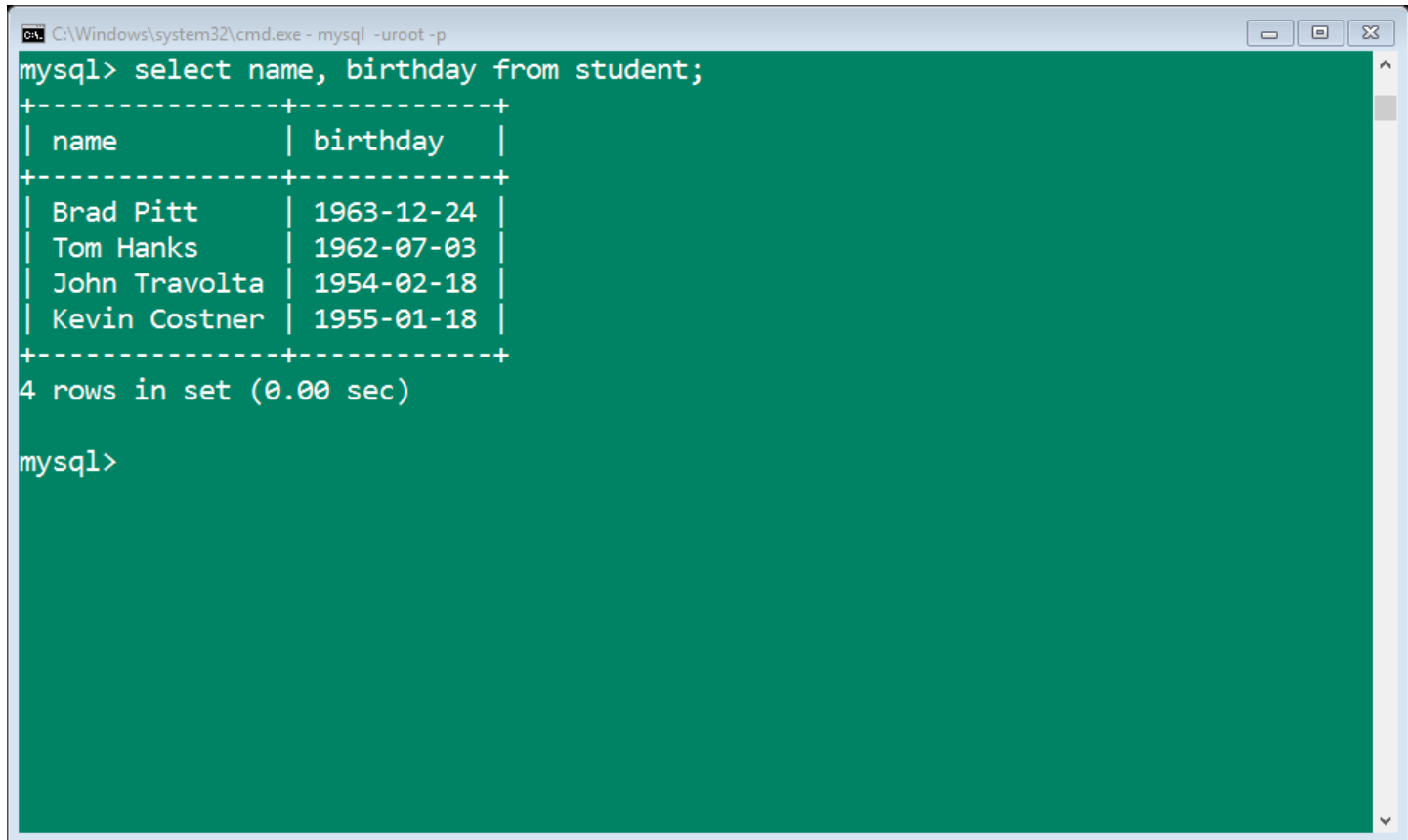
```
mysql> select name from student;
+-----+
| name          |
+-----+
| Brad Pitt     |
| Tom Hanks     |
| John Travolta |
| Kevin Costner |
+-----+
4 rows in set (0.00 sec)

mysql>
```

데이터 관리하기

- 데이터 조회하기

- **TODO:** student 테이블의 이름(name)과 생년월일(birthday)만 조회해 본다



```
C:\Windows\system32\cmd.exe - mysql -uroot -p
mysql> select name, birthday from student;
+-----+-----+
| name      | birthday |
+-----+-----+
| Brad Pitt  | 1963-12-24 |
| Tom Hanks  | 1962-07-03 |
| John Travolta | 1954-02-18 |
| Kevin Costner | 1955-01-18 |
+-----+-----+
4 rows in set (0.00 sec)

mysql>
```

데이터 관리하기

- 데이터 조회하기

- **TODO:** student 테이블에서 id가 13004인 사람의 데이터만 조회해 본다

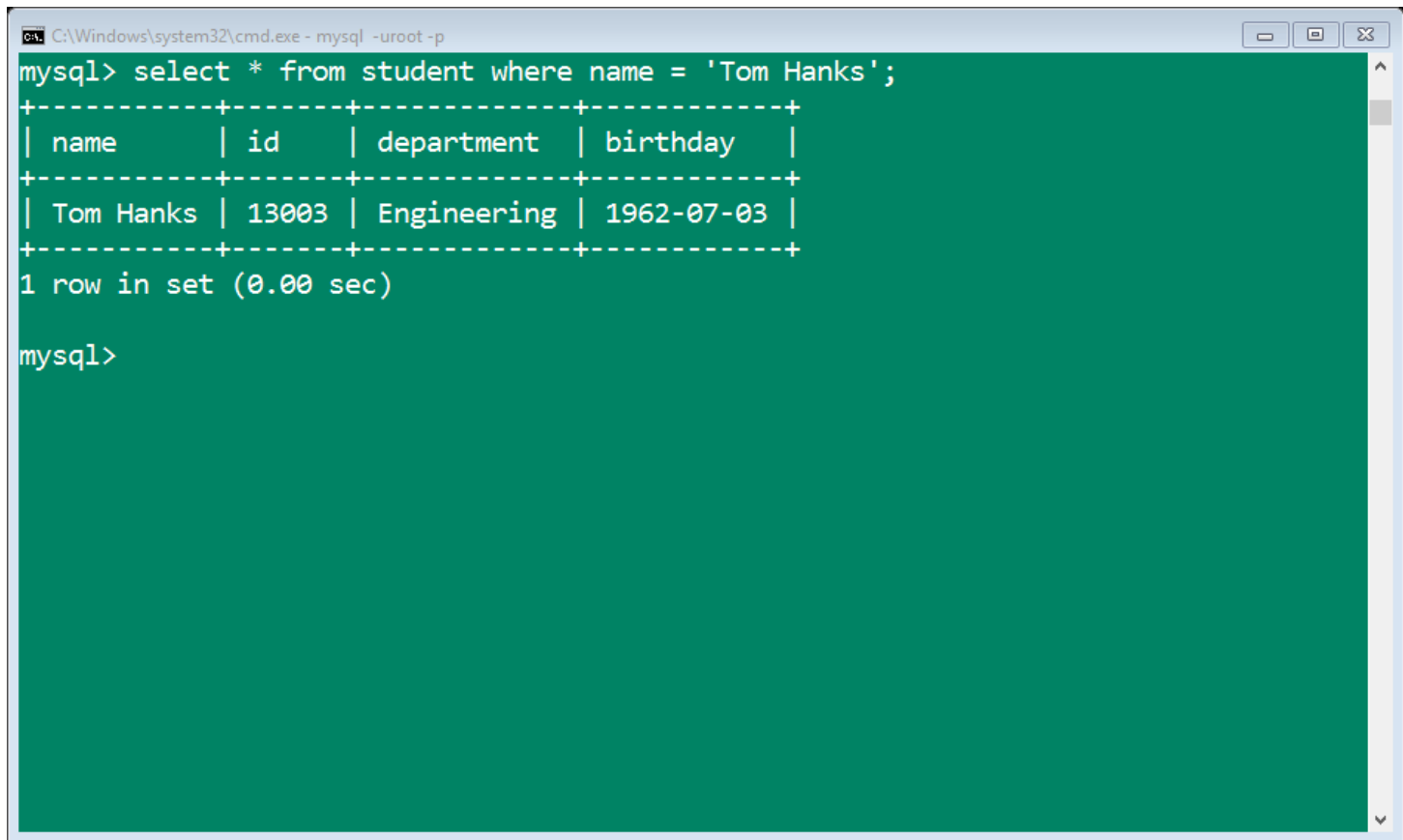
```
C:\Windows\system32\cmd.exe - mysql -uroot -p
mysql> select * from student where id = '13004';
+-----+-----+-----+-----+
| name      | id    | department | birthday |
+-----+-----+-----+-----+
| John Travolta | 13004 | Engineering | 1954-02-18 |
+-----+-----+-----+-----+
1 row in set (0.03 sec)

mysql>
```

데이터 관리하기

- 데이터 조회하기

- **TODO:** student 테이블에서 이름이 'Tom Hanks' 인 사람의 데이터만 조회해 본다



```
C:\Windows\system32\cmd.exe - mysql -uroot -p
mysql> select * from student where name = 'Tom Hanks';
+-----+-----+-----+-----+
| name      | id    | department | birthday |
+-----+-----+-----+-----+
| Tom Hanks | 13003 | Engineering | 1962-07-03 |
+-----+-----+-----+-----+
1 row in set (0.00 sec)

mysql>
```

실습 3-1-1. 데이터베이스 및 테이블 생성하기

· company 데이터베이스를 만들고, 데이터베이스에 아래의 데이터를 담은 employee 테이블을 생성해 본다

fname	lname	id	bdate	sex	salary
Scarlette	Johnasson	100001	1984-11-22	F	100000
Natalie	Portman	100002	1981-06-09	F	250000
Emma	Stone	100003	1988-11-06	F	350000
Ryan	Gosling	100004	1980-11-22	M	340000
Tom	Hardy	100005	1977-09-15	M	750000

실습 3-1-1. 데이터베이스 및 테이블 생성하기

· 수행 예시

```
C:\Windows\system32\cmd.exe - mysql -ubuomsoo -p
mysql> select * from employee;
+-----+-----+-----+-----+-----+-----+
| fname | lname | id    | bdate | sex | salary |
+-----+-----+-----+-----+-----+-----+
| Scarlett | Johansson | 100001 | 1984-11-22 | F | 100000 |
| Natalie | Portman | 100002 | 1981-06-09 | F | 250000 |
| Emma | Stone | 100003 | 1988-11-06 | F | 350000 |
| Ryan | Gosling | 100004 | 1980-11-12 | M | 340000 |
| Tom | Hardy | 100005 | 1977-09-15 | M | 750000 |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql>
```

PyMySQL



PyMySQL

- Python DB API 표준을 따르는 MySQL DB 모듈로, Python을 활용해 MySQL 데이터베이스에 접근을 가능하게 한다
- PyMySQL 설치(with conda)
 - 커맨드 라인에 아래 명령어를 입력한다

```
> conda install pymysql
```

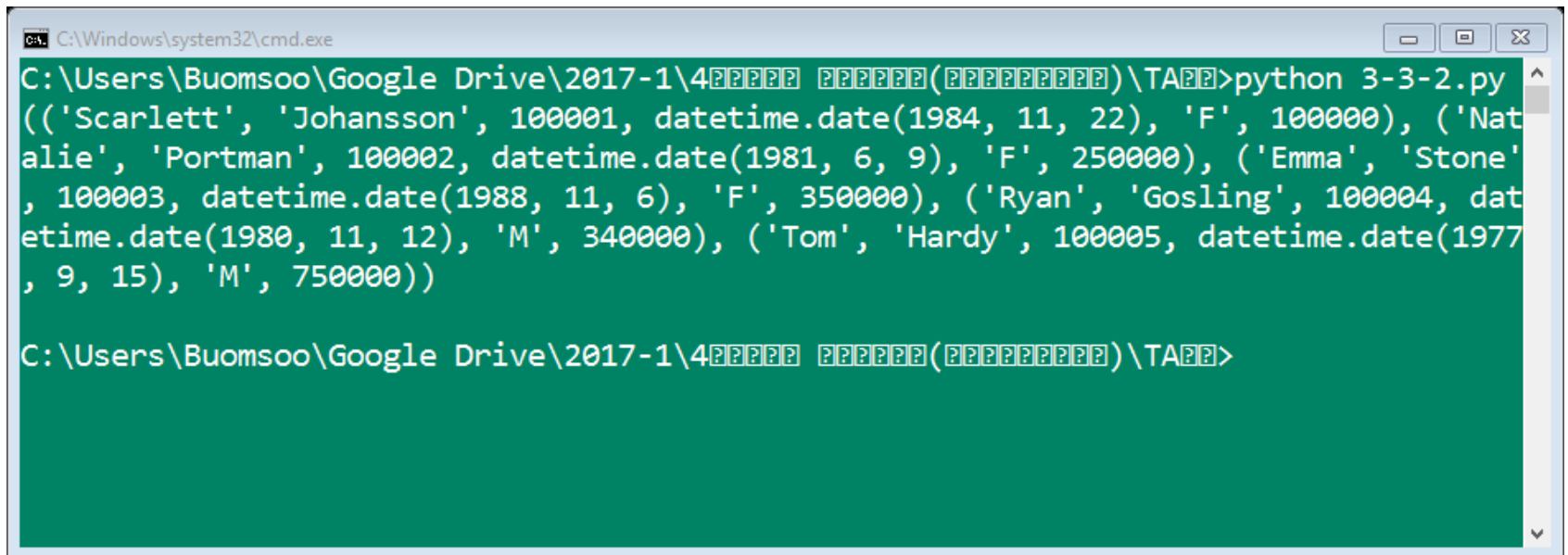
실습 3-3-2. DB에서 데이터 불러오기

· 실습 3-3-1에서 생성한 데이터베이스 테이블에서 데이터를 불러온다

- sql query '**select ~ from**'을 활용한다

- fetchall() 함수를 활용한다

- 수행 예시



```
C:\Windows\system32\cmd.exe
C:\Users\Buomsoo\Google Drive\2017-1\4????? ?????(????????)\TA??>python 3-3-2.py
(('Scarlett', 'Johansson', 100001, datetime.date(1984, 11, 22), 'F', 100000), ('Natalie', 'Portman', 100002, datetime.date(1981, 6, 9), 'F', 250000), ('Emma', 'Stone', 100003, datetime.date(1988, 11, 6), 'F', 350000), ('Ryan', 'Gosling', 100004, datetime.date(1980, 11, 12), 'M', 340000), ('Tom', 'Hardy', 100005, datetime.date(1977, 9, 15), 'M', 750000))

C:\Users\Buomsoo\Google Drive\2017-1\4????? ?????(????????)\TA??>
```

실습 3-3-3. DB에서 데이터 불러오기

- 테이블에서 일부 정보만 불러온다

- SQL query를 바꿀 수도 있고 불러온 이후에 데이터를 가공할 수도 있다

- 수행 예시

```
C:\Users\Buomsoo\Google Drive\2017-1\4????? ??????(?????????)\TA??>python 3-3-3.py
Name of the employee is: Scarlett Johansson
Name of the employee is: Natalie Portman
Name of the employee is: Emma Stone
Name of the employee is: Ryan Gosling
Name of the employee is: Tom Hardy

C:\Users\Buomsoo\Google Drive\2017-1\4????? ??????(?????????)\TA??>
```

```
C:\Users\Buomsoo\Google Drive\2017-1\4????? ??????(?????????)\TA??>python 3-3-3.py
('Scarlett', 'Johansson')
('Natalie', 'Portman')
('Emma', 'Stone')
('Ryan', 'Gosling')
('Tom', 'Hardy')

C:\Users\Buomsoo\Google Drive\2017-1\4????? ??????(?????????)\TA??>
```

실습 3-3-4. DB에 데이터 저장하기

· 테이블에 새로운 정보를 저장한다

- Insert into문을 활용해 아래 데이터를 담은 행을 하나 추가한다

fname	lname	id	bdate	sex	salary
Daniel	Craig	100006	1968-03-02	M	800000

- 수행 예시

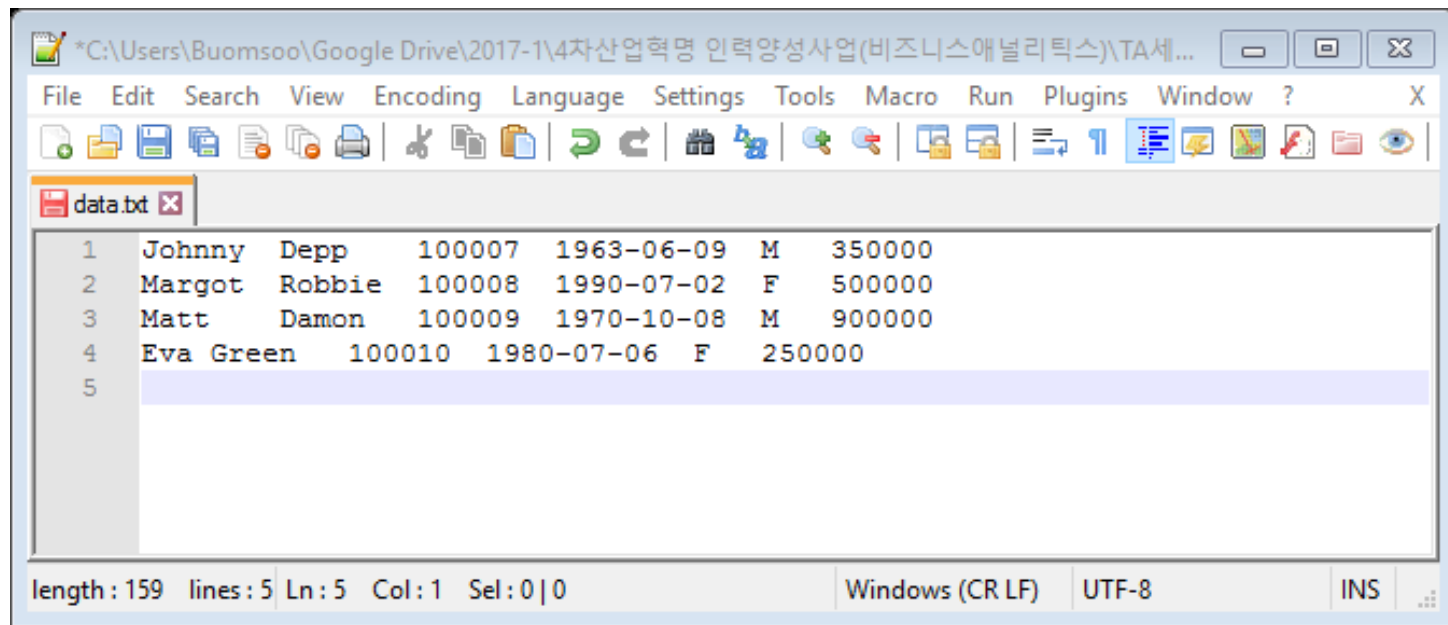
```
C:\Windows\system32\cmd.exe - mysql -ubuomsoo -p
mysql> select * from employee;
+-----+-----+-----+-----+-----+-----+
| fname | lname  | id    | bdate  | sex  | salary |
+-----+-----+-----+-----+-----+-----+
| Scarlett | Johansson | 100001 | 1984-11-22 | F    | 100000 |
| Natalie  | Portman  | 100002 | 1981-06-09 | F    | 250000 |
| Emma     | Stone    | 100003 | 1988-11-06 | F    | 350000 |
| Ryan     | Gosling  | 100004 | 1980-11-12 | M    | 340000 |
| Tom      | Hardy    | 100005 | 1977-09-15 | M    | 750000 |
| Daniel   | Craig    | 100006 | 1968-03-02 | M    | 800000 |
+-----+-----+-----+-----+-----+-----+
6 rows in set (0.00 sec)

mysql>
```

실습 3-3-5. DB에 데이터 저장하기 (2)

· 테이블에 새로운 정보를 저장한다

- Insert into문을 이용해 텍스트 파일(data.txt)에서 데이터를 불러와 이를 employee 테이블에 추가한다



A screenshot of a text editor window titled "data.txt". The window displays a table of data with 5 rows. The first four rows contain data for Johnny Depp, Margot Robbie, Matt Damon, and Eva Green. The fifth row is empty. The status bar at the bottom shows "length: 159 lines: 5 Ln: 5 Col: 1 Sel: 0 | 0", "Windows (CR LF)", "UTF-8", and "INS".

1	Johnny	Depp	100007	1963-06-09	M	350000
2	Margot	Robbie	100008	1990-07-02	F	500000
3	Matt	Damon	100009	1970-10-08	M	900000
4	Eva	Green	100010	1980-07-06	F	250000
5						

- 한 줄에 하나씩 추가한다

실습 3-3-5. DB에 데이터 저장하기 (2)

· 테이블에 새로운 정보를 저장한다

- 수행 예시

```
C:\Windows\system32\cmd.exe - mysql -ubuomsoo -p
mysql> select * from employee;
```

fname	lname	id	bdate	sex	salary
Scarlett	Johansson	100001	1984-11-22	F	100000
Natalie	Portman	100002	1981-06-09	F	250000
Emma	Stone	100003	1988-11-06	F	350000
Ryan	Gosling	100004	1980-11-12	M	340000
Tom	Hardy	100005	1977-09-15	M	750000
Daniel	Craig	100006	1968-03-02	M	800000
Johnny	Depp	100007	1963-06-09	M	350000
Margot	Robbie	100008	1990-07-02	F	500000
Matt	Damon	100009	1970-10-08	M	900000
Eva	Green	100010	1980-07-06	F	250000

```
10 rows in set (0.00 sec)

mysql>
```