

당뇨 환자를 위한 식단 테이블 생성

유재명

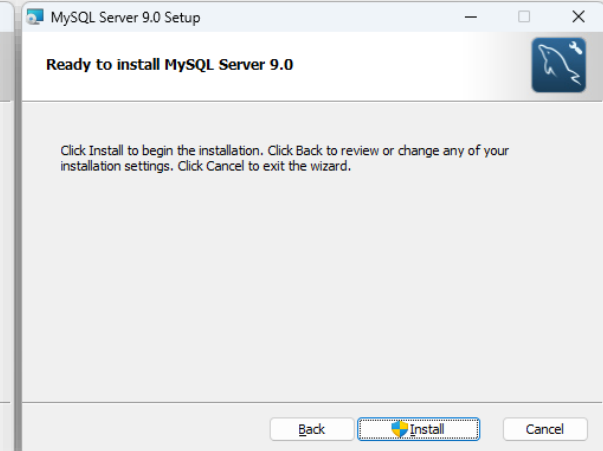
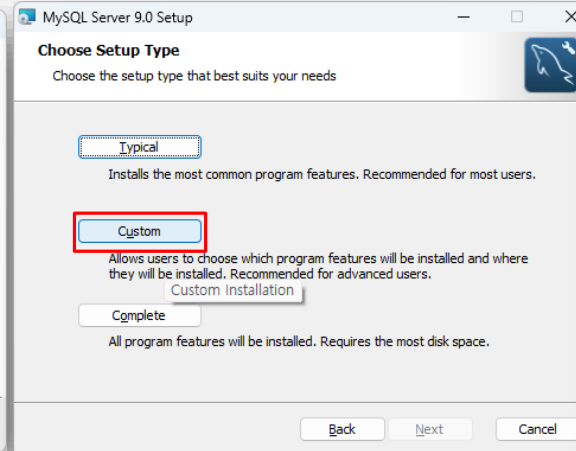
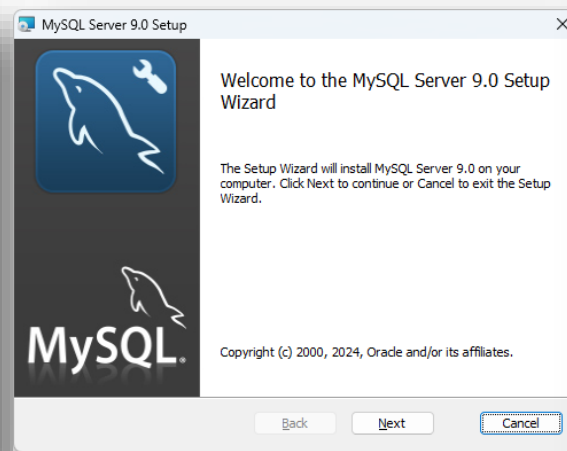
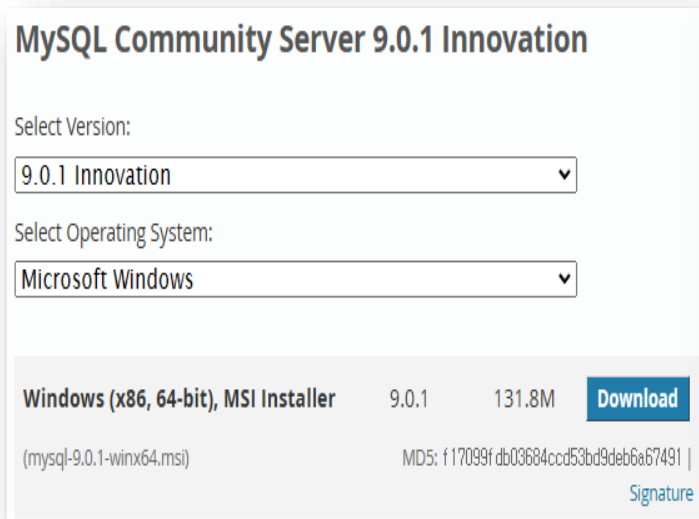
목차

1. MySQL 설치
2. 데이터베이스 생성
3. diabetic_meals 테이블 생성
4. 파이썬에서 csv 파일 읽고 MySQL diabetic DB / diabetic_meals 테이블에 입력
5. 파이썬을 이용해서 MySQL 테이블 조회

MySQL 설치

MySQL 다운로드

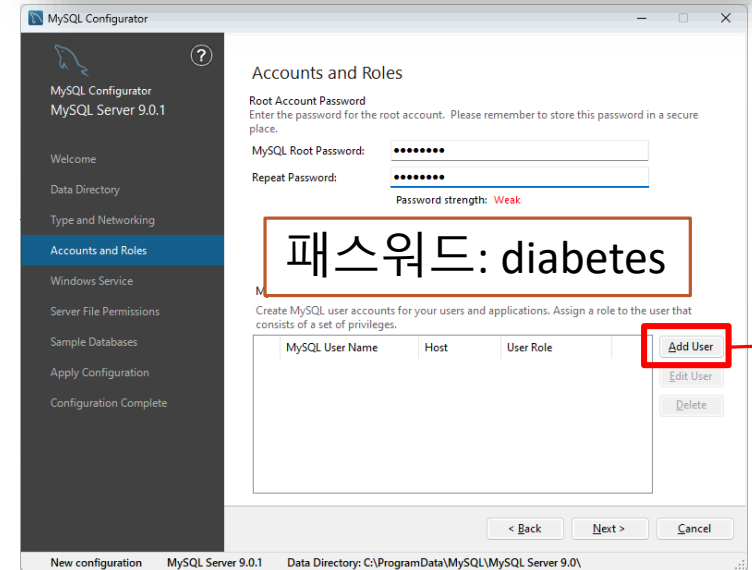
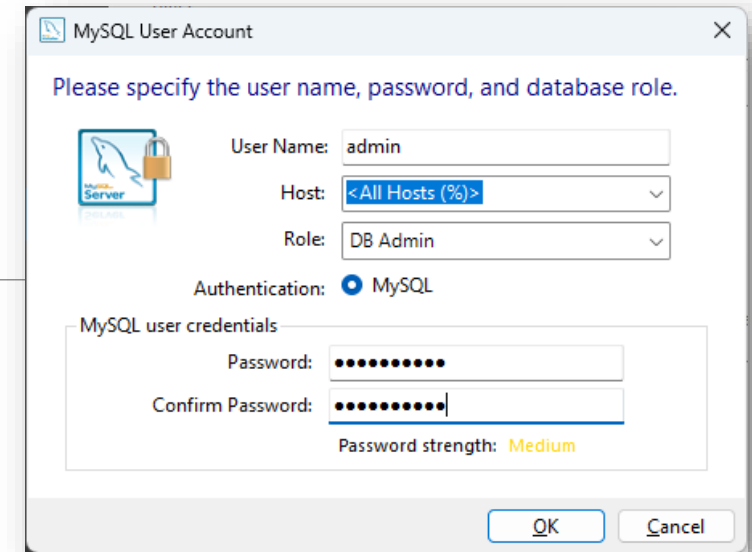
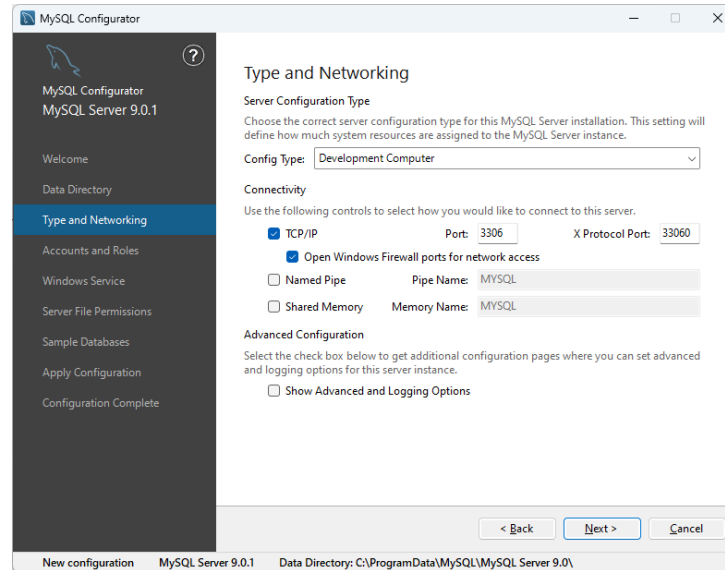
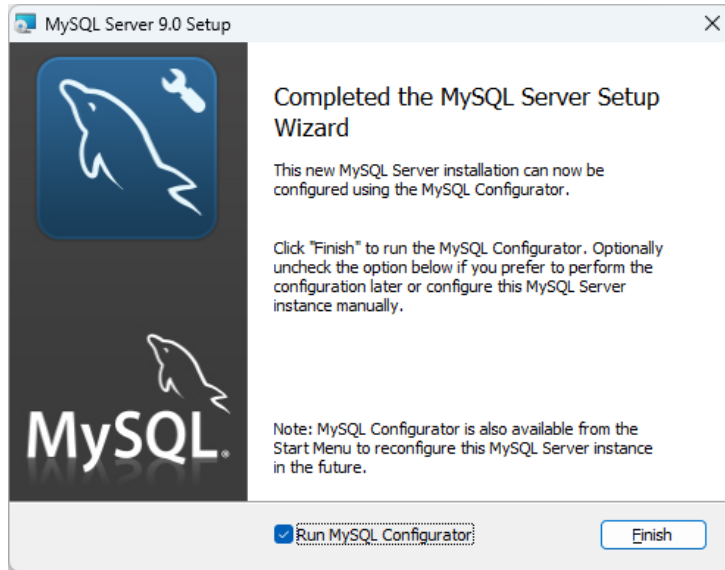
- <https://dev.mysql.com/downloads/> 에 접속
- "MySQL Community Server" 선택하여 다운로드 및 설치



MySQL 설치

MySQL 다운로드

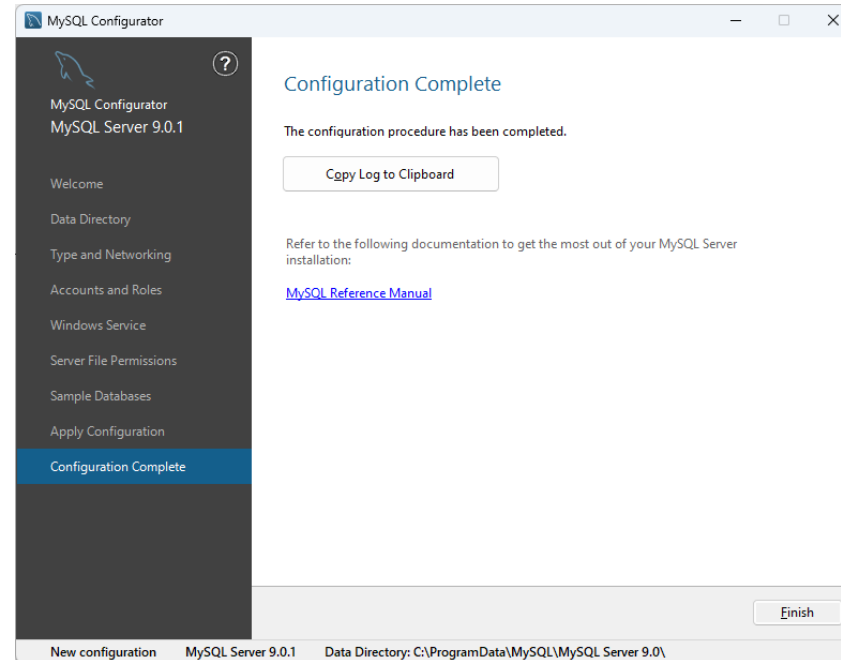
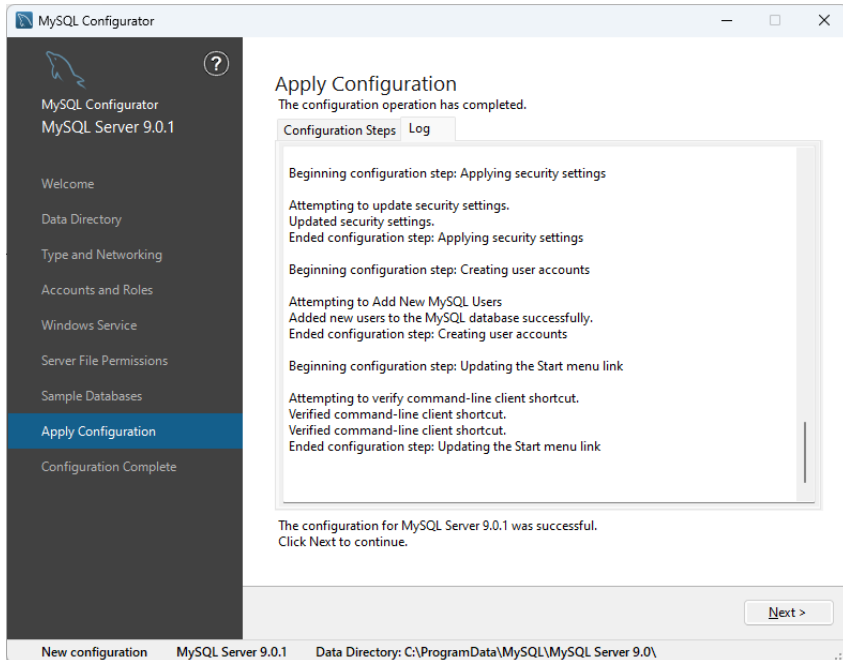
- <https://dev.mysql.com/downloads/> 에 접속
- "MySQL Community Server" 선택하여 다운로드 및 설치



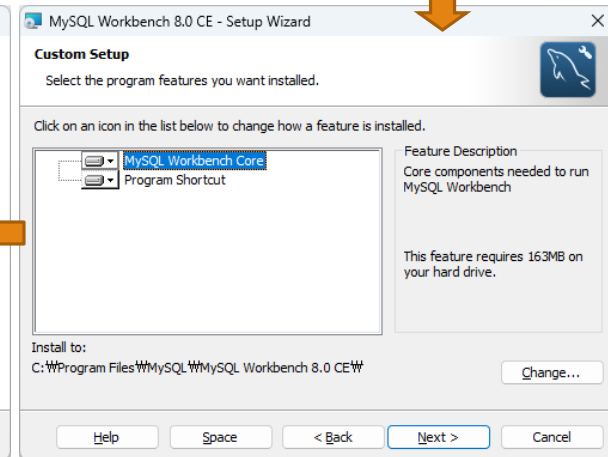
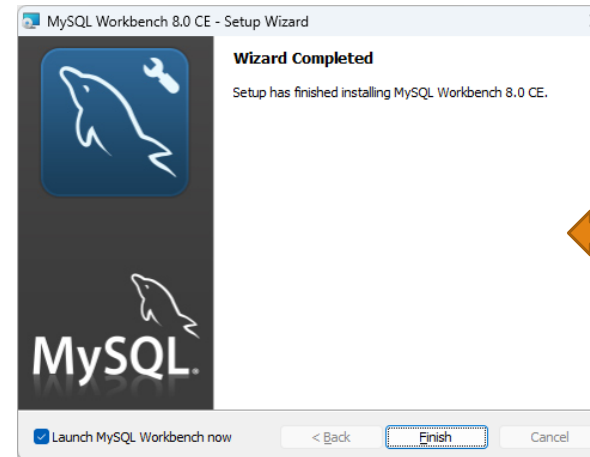
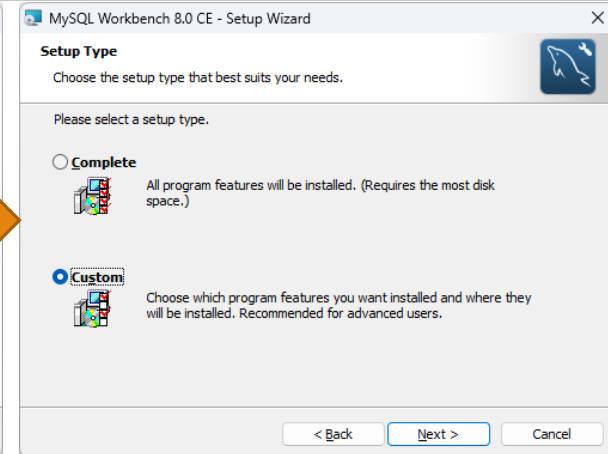
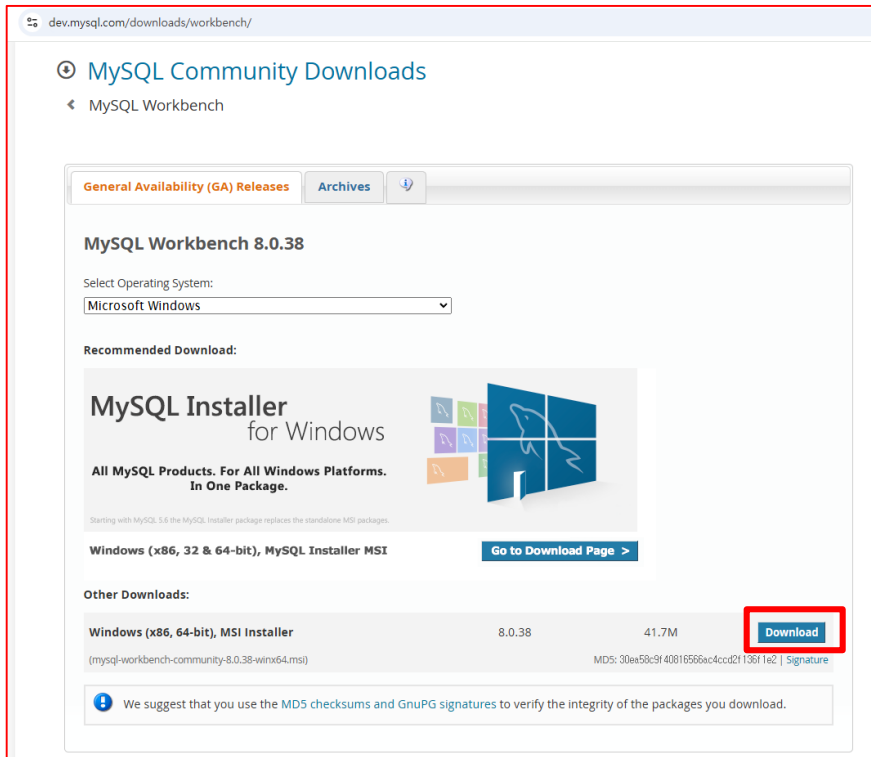
MySQL 설치

MySQL 다운로드

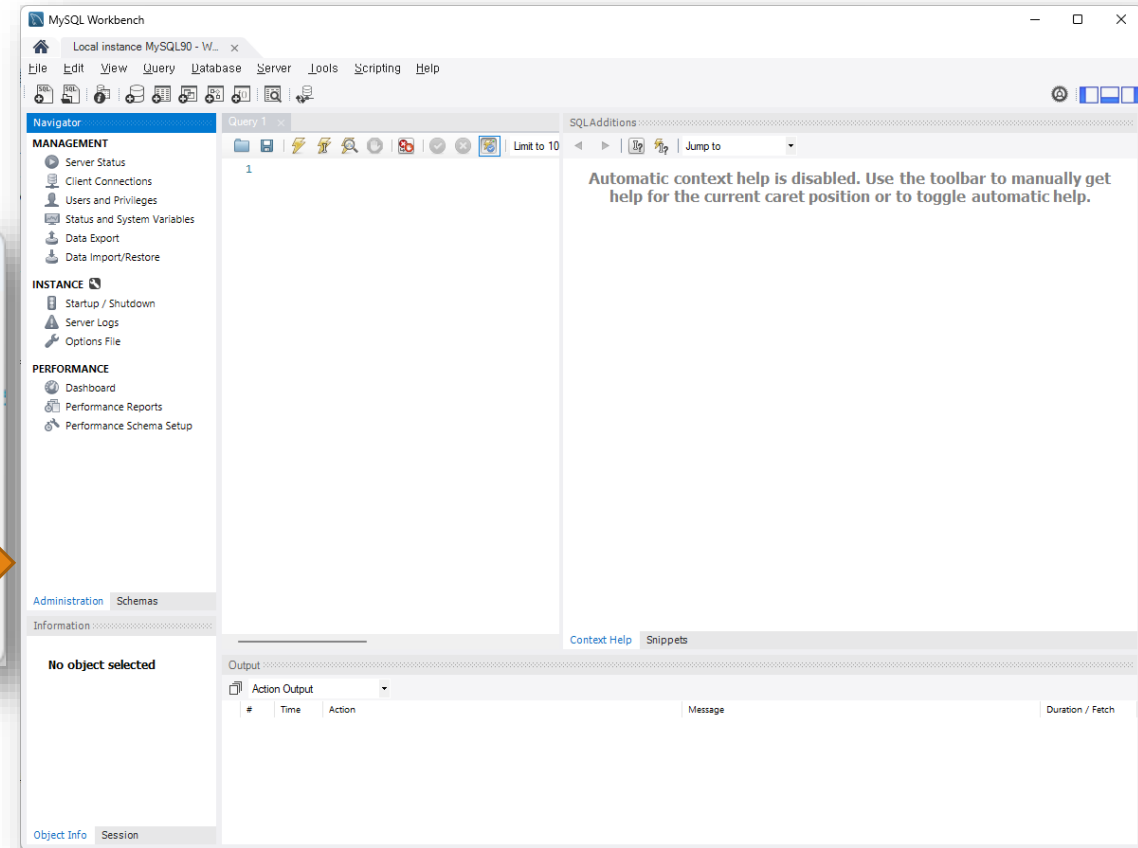
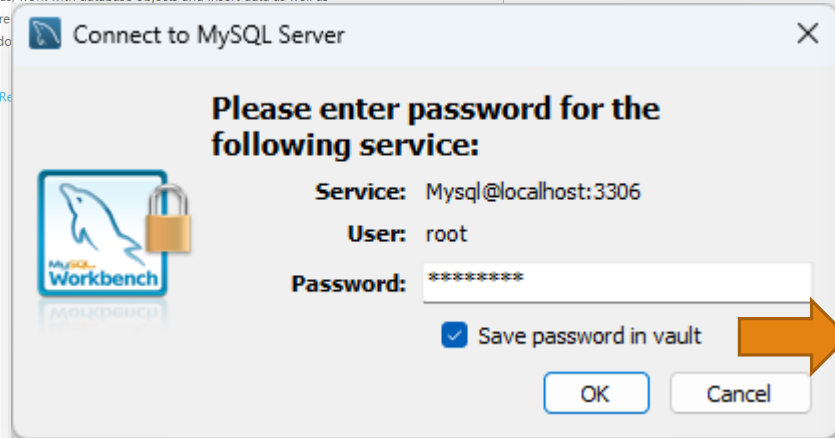
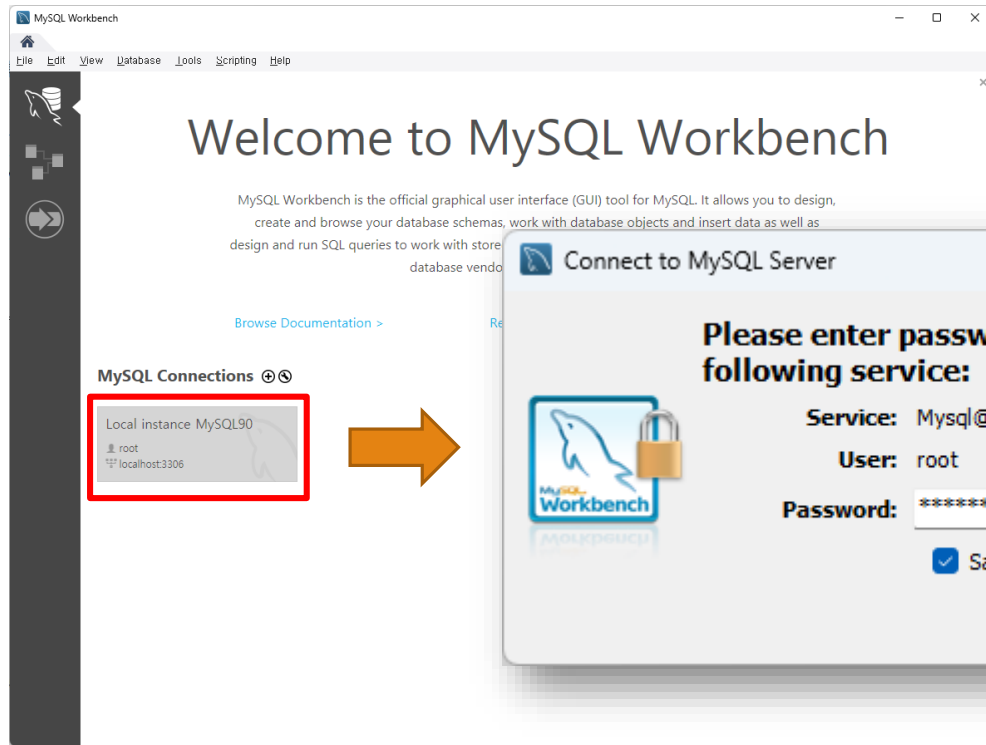
- <https://dev.mysql.com/downloads/> 에 접속
- "MySQL Community Server" 선택하여 다운로드 및 설치



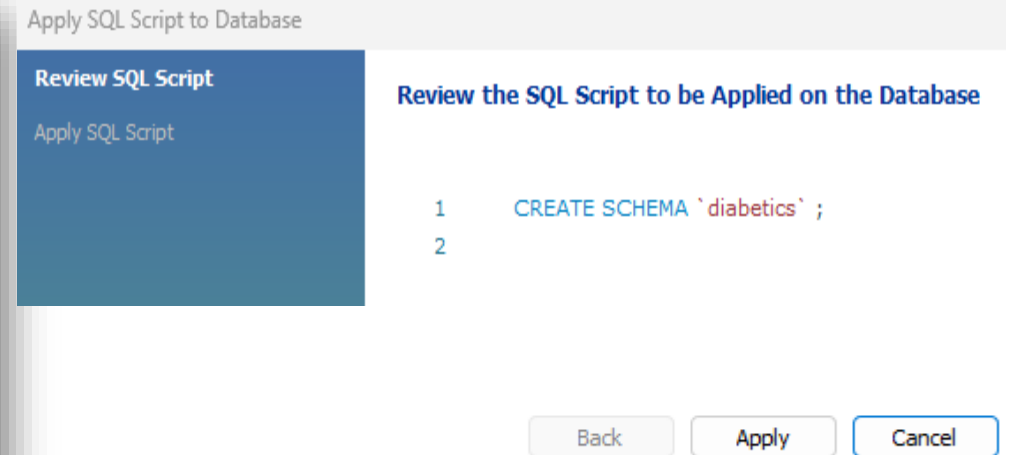
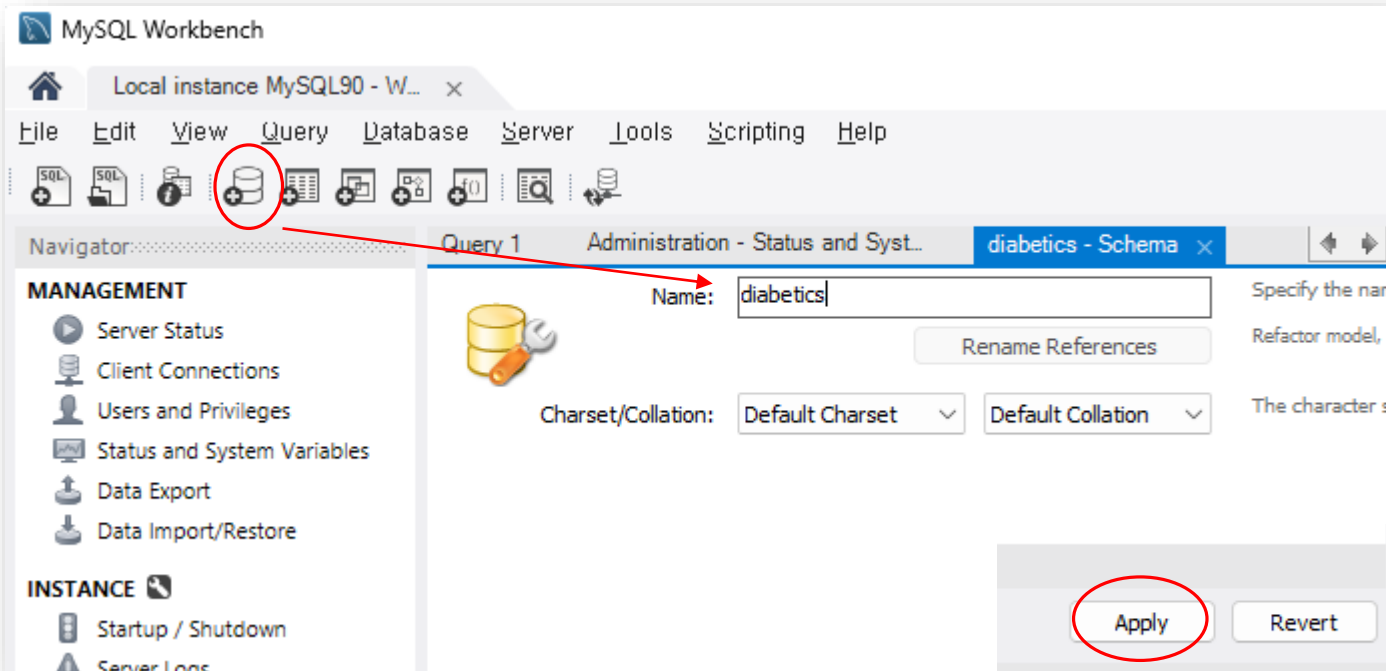
MySQL Workbench(GUI도구) 설치



MySQL Workbench(GUI도구) 설치



데이터베이스 생성



The screenshot shows the 'Schemas' view in SQL Server Enterprise Manager. The 'diabetics' schema is expanded, revealing its contents: Tables, Views, Stored Procedures, and Functions. Below it, the 'sys' schema is partially visible. The interface includes a 'Navigator' pane on the left and a 'Filter objects' search bar at the top.



Table Name: Schema:

Charset/Collation: Engine:

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF
<input type="text" value="id"/>	<input type="text" value="INT"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="text" value="d_meal1"/>	<input type="text" value="VARCHAR(50)"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="text" value="d_meal2"/>	<input type="text" value="VARCHAR(100)"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Column Name: Type:

Charset/Collation:

Comments:

Storage: ☐ Virtual ☐ Stored

☐ Primary Key ☐ Not Null ☐ Unique

☐ Binary ☐ Unsigned ☐ Zero Fill

☐ Auto Increment ☐ Generated

Columns Indexes Foreign Keys Triggers Partitioning Options

데이터베이스 – 테이블

The screenshot displays a database management interface with three main panels:

- Left Panel: Review SQL Script to be Applied on the Database**
This panel shows a SQL script for creating a table named `diabetics.diabetic_meal`. The script is as follows:

```
1 CREATE TABLE `diabetics`.`diabetic_meal` (  
2   `id` INT NOT NULL AUTO_INCREMENT,  
3   `d_meal1` VARCHAR(50) NULL,  
4   `d_meal2` VARCHAR(100) NULL,  
5   PRIMARY KEY (`id`))  
6 ENGINE = InnoDB  
7 DEFAULT CHARACTER SET = utf8  
8 COLLATE = utf8_unicode_ci  
9 COMMENT = '당뇨환자식단테이블';  
10
```
- Middle Panel: Navigator**
This panel shows the database schema structure. A context menu is open over the `diabetic_meal` table, listing various actions:
 - Select Rows - Limit 1000
 - Table Inspector
 - Copy to Clipboard
 - Table Data Export Wizard
 - Table Data Import Wizard
 - Send to SQL Editor
 - Create Table...
 - Create Table Like...
 - Alter Table...
 - Table Maintenance...
 - Drop Table...
 - Truncate Table...
 - Search Table Data...
 - Refresh All
- Right Panel: diabetics - Schema diabetics_meal - Table**
This panel shows the SQL query `SELECT * FROM diabetics.diabetic_meal;` and its results in a table grid. The table has three columns: `id`, `d_meal1`, and `d_meal2`. The first row shows all three columns as `NULL`.

테이블 생성

```
CREATE TABLE students (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    name VARCHAR(50),  
    age INT,  
    grade VARCHAR(10)  
);
```

데이터 질의(SQL, Structure Query Language)

* 데이터 삽입

```
INSERT INTO students (name, age, grade) VALUES ('John Doe', 14, '8th Grade');
```

* 데이터 조회

```
SELECT * FROM students;
```

* 데이터 갱신

```
UPDATE students SET name='Hong', age=20 WHERE name='John Doe';
```

* 데이터 삭제

```
DELETE FROM students WHERE name='John Doe';
```

SQL문법 연습 : <https://www.w3schools.com/sql/>

관계형 데이터베이스(예시)

기본키와 외래키의 관계

- 학생, 수업, 등록 테이블간 연관성 유지 및 조회 용이
- 데이터의 효율적 관리 및 데이터 무결성 보장

```
CREATE TABLE Students (  
    student_id INT AUTO_INCREMENT PRIMARY KEY,  
    name VARCHAR(100),  
    age INT,  
    gender VARCHAR(10)  
);
```

```
INSERT INTO Students (name, age, gender) VALUES ('John Doe', 20, 'Male');  
INSERT INTO Students (name, age, gender) VALUES ('Jane Smith', 22, 'Female');
```

```
INSERT INTO Courses (course_name, teacher) VALUES ('Math 101', 'Mr. Brown');  
INSERT INTO Courses (course_name, teacher) VALUES ('History 201', 'Ms. Green');
```

```
INSERT INTO Enrollments (student_id, course_id, enrollment_date) VALUES (1, 1, '2023-09-01');  
INSERT INTO Enrollments (student_id, course_id, enrollment_date) VALUES (1, 2, '2023-09-02');  
INSERT INTO Enrollments (student_id, course_id, enrollment_date) VALUES (2, 1, '2023-09-03');
```

```
SELECT Students.name, Courses.course_name, Enrollments.enrollment_date  
FROM Enrollments  
JOIN Students ON Enrollments.student_id = Students.student_id  
JOIN Courses ON Enrollments.course_id = Courses.course_id;
```

```
CREATE TABLE Courses(  
    course_id INT AUTO_INCREMENT PRIMARY KEY,  
    course_name VARCHAR(100),  
    teacher VARCHAR(100)  
);
```

```
CREATE TABLE Enrollments (  
    enrollment_id INT AUTO_INCREMENT PRIMARY KEY,  
    student_id INT,  
    course_id INT,  
    enrollment_date DATE,  
    FOREIGN KEY (student_id) REFERENCES Students(student_id),  
    FOREIGN KEY (course_id) REFERENCES Courses(course_id)  
);
```

파이썬에서 활용

선수조건) 관련 라이브러리 설치 : pip install pymysql pandas

MySQL 서버 연결

```
import pymysql
```

```
import pandas as pd
```

```
connection = pymysql.connect(
```

```
    host='localhost',    # MySQL 서버 호스트명 (로컬호스트의 경우 'localhost')
```

```
    user='root',        # MySQL 사용자 이름
```

```
    password='diabetes', # MySQL 루트 비밀번호
```

```
    database='diabetics', # 사용할 데이터베이스 이름
```

```
    charset='utf8',
```

```
    cursorclass=pymysql.cursors.DictCursor
```

```
)
```

파이썬에서 CSV 읽기

CSV 파일을 pandas로 읽기

diafile = 'diabetic_meals.csv' # CSV 파일 경로

df = pd.read_csv(diafile)

파이썬에서 CSV -> MySQL로 입력

```
# 데이터베이스에 데이터를 삽입하는 함수
def insert_data_to_mysql(dataframe, connection):
    with connection.cursor() as cursor:
        # 데이터프레임의 각 행(row)을 MySQL에 삽입
        for _, row in dataframe.iterrows():
            sql = """
            INSERT INTO diabetic_meal (d_meal1, d_meal2, calories, carbohydrates, protein, fat)
            VALUES (%s, %s, %d, %f, %f,%f)
            """
            cursor.execute(sql, (
                row['d_meal1'],
                row['d_meal2'],
                row['calories'],
                row['carbohydrates'],
                row['protein'],
                row['fat']
            ))
        # 변경사항을 커밋
        connection.commit()
```


파이썬에서 MySQL 모든 내용 조회

```
# 데이터베이스에 데이터를 조회하는 함수
def query_data_from_mysql(connection):
    with connection.cursor() as cursor:
        sql = "SELECT * FROM diabetic_meals"
        cursor.execute(sql)
        result = cursor.fetchall() # 조회된 모든 데이터를 가져옴
    return result
```

파이썬에서 MySQL 특정값 조회

```
# 특정 d_meal1 에 대한 데이터를 조회하는 함수
def query_by_meal_name(connection, meal_name):
    with connection.cursor() as cursor:
        sql = "SELECT * FROM diabetic_meals WHERE d_meal1 = %s"
        cursor.execute(sql, (meal_name,))
        result = cursor.fetchall() # 조회된 모든 데이터를 가져옴
    return result
```

파이썬 실행

```
try:
    insert_data_to_mysql(df, connection)
    print("데이터 삽입 완료!")

    # 데이터베이스에서 전체 데이터 조회
    query_result = query_data_from_mysql(connection)
    for row in query_result:
        print(row)

    # 특정값 조회
    meal_name = input('조회할 meal 입력: ')
    query_result = query_by_meal_name(connection, meal_name_input)
    if query_result:
        print(f'{meal_name} 조회')
        for row in query_result:
            print(row)
    else:
        print(f'{meal_name} 자료 없음')
except Exception as e:
    print(f"오류 발생: {e}")
finally:
    connection.close()
```