# DB MySQL:

- MySQL is an open-source relational database management system (RDBMS).
- Its name is a combination of "My", the name of co-founder Michael Widenius's daughter, and "SQL".
- The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements.
- MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation.
- For proprietary use, several paid editions are available, and offer additional functionality.

## Terminology:

- **Database** − A database is a collection of tables, with related data.
- **Table** − A table is a matrix with data. A table in a database looks like a simple spreadsheet.
- **Column** − One column (data element) contains data of one and the same kind, for example the column id.
- **Row** − A row (entry or record) is a group of related data, for example the data of one user.

## PyMySQL:

- PyMySQL is an interface for connecting to a MySQL database server from Python.
- It implements the Python Database API v2.0 and contains a pure-Python MySQL client library.
- PyMySql can be install through PIP, and can be used from any Python project.
- Open CMD/Terminal and type **pip pymysql**

## Implantation

- Importing PyMySql module:

```
import pymysql
```

- Open a DB connection:

```
conn = pymysql.connect(host='localhost', port=3306, user='root', passwd='Aa123456', db='table')
```

- Using a cursor to run DB queries

```
cursor.execute("SELECT * FROM table.users;")
```

- Closing cursor connection

```
cursor.close()
conn.close()
```

## Cursor

- A database **cursor** is a control structure that enables iteration over the records in a database.
- Cursors are used for actions such as retrieval, addition and removal of database records.
- A cursor can be viewed as a pointer to one row in a set of rows.
- The cursor can only reference one row at a time, but can move to other rows of the result set as needed.
- One of the main advantages is the ability to make a single call to the database and work on the returned data within our code.

## MySQL statements:

- The **SELECT** statement is used to select data from one or more tables.

```
# Getting all data from table "users"
cursor.execute("SELECT * FROM table.users;")

# Iterating table and printing all users
for row in cursor:
    print(row)
```

- The **INSERT** INTO statement is used to insert new records in a table.

```
# Inserting data into table
cursor.execute("INSERT into `table`.`users` (name, id) VALUES ('john', 5)")
```

- The **DELETE** statement is used to delete existing records in a table.

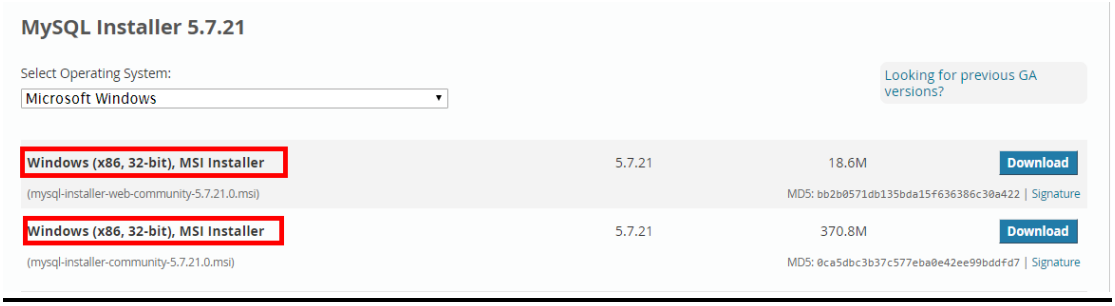
```
# Deleting data into table
cursor.execute("DELETE FROM `table`.`users` WHERE name = 'john'")
```

- The UPDATE statement is used to modify the existing records in a table.

```
# Deleting data into table
cursor.execute("UPDATE `table`.`users` SET id = '10' WHERE name = 'john'")
```

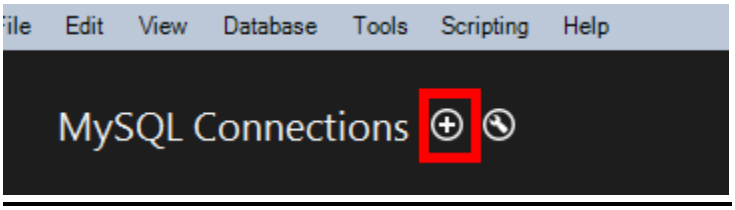
# Setup:

- MySQL workbench is software including all we need to work with MySQL including MySQL server, connectors documentation and more.
- Download MySQL Workbench software from https://dev.mysql.com/downloads/windows/installer/5.7.html (or search MySQL workbench download).



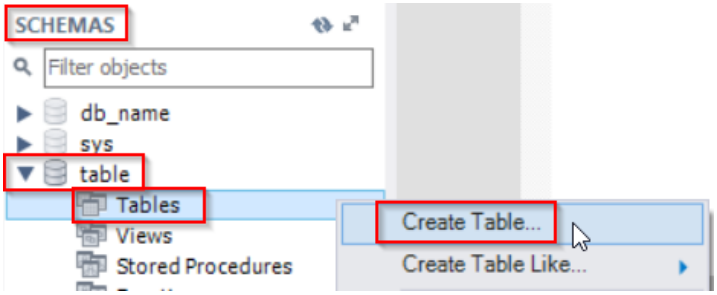

# Creating a connection

- Enter software once it's installed and press the plus button

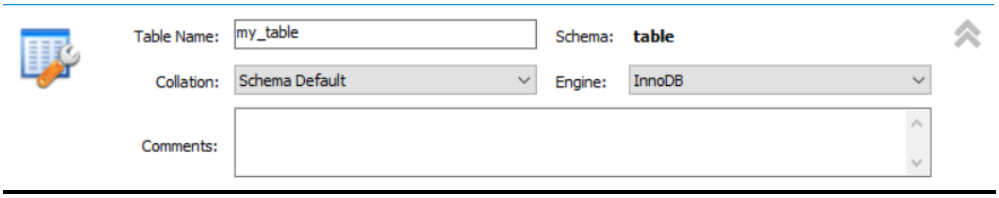



- You will need to enter the following information:
- **Name** – whatever you want
- **Method** – Standard (TCP/IP)
- **Hostname** – 127.0.0.1(local host)
- **Port** – 3306 (mysql default)
- **Username** – root (mysql default)
- **Password** – no password by default

- Once you are connected, to create a new table, simply go under **schemas** → expand **table** → right click on **Tables** → **Create Table..**




- We will now see the following screen, which is where we are going to define our table structure:




- Here is the explanation of each of the flags available for each of your table column.
  - **PK** - Primary Key
  - **NN** - Not Null
  - **BIN** - Binary (stores data as binary strings)
  - **UN** - Unsigned (non-negative numbers only. so if the range is -500 to 500, instead its 0 - 1000, the range is the same but it starts at 0)
  - **UQ** - Unique Key
  - **ZF** - Zero-Filled (if the length is 5 like INT(5) then every field is filled with 0's to the 5th value. 12 = 00012, 400 = 00400, etc. )
  - **AI** - Auto Increment
  - **G** - Generated column. i.e. value generated by a formula based on the other columns

- There are many data types your column can contain, to simplify I’m going to show you only two popular types:
- **INT** – is a 4 bytes type and supports numbers between -2147483648 to 2147483647.
- **VARCHAR -** Contains strings. Columns are variable-length strings which can contain a value between 0 to 255.