**IT8701 Introduction to Programming for Data Science**

**Lab 05 – Creating and manipulating data in relational databases**

**What you will learn / do in this lab**

1. How to store and read data from a relational database such as mySQL
2. How to integrate what you have previously learnt to create a complete Python program that reads data from a relational databases, manipulate the data using Numpy / Pandas package and visualize the data using Matplotlib
3. https://www.udemy.com/introduction-to-mysql-database-training-course

**Table of Contents**

[1. Overview 3](#_Toc500252935)

[A. What you will do for this lab 3](#_Toc500252936)

[B. Intro to mySQL 3](#_Toc500252937)

[2. Install mySQL 3](#_Toc500252938)

[A. Windows Installation 3](#_Toc500252939)

[B. MacOS Installation 4](#_Toc500252940)

[3. Start / Stop mySQL 5](#_Toc500252941)

[A. On Windows (command-line) 5](#_Toc500252942)

[Start mySQL 5](#_Toc500252943)

[Stop mySQL 6](#_Toc500252944)

[Troubleshooting 6](#_Toc500252945)

[B. On Windows (mySQL Notifier) 7](#_Toc500252946)

[What is MySQL Notifier? 7](#_Toc500252947)

[C. On MacOS 7](#_Toc500252948)

[4. Use MySQL Workbench 9](#_Toc500252949)

[A. Connect to your database server 9](#_Toc500252950)

[Start mySQL Workbench 9](#_Toc500252951)

[Create a new MySQL Connection 9](#_Toc500252952)

[Test connection to localhost 10](#_Toc500252953)

[B. Create a new database 11](#_Toc500252954)

[Connect to localhost 11](#_Toc500252955)

[Use the CREATE DATABASE SQL statement 13](#_Toc500252956)

[C. Create a new table 13](#_Toc500252957)

[Use the CREATE TABLE SQL statement 13](#_Toc500252958)

[D. Insert new records 14](#_Toc500252959)

[Insert one record using INSERT INTO 14](#_Toc500252960)

[Insert multiple records using INSERT INTO 16](#_Toc500252961)

[E. View records 17](#_Toc500252962)

[Task 1 - View all records using SELECT 17](#_Toc500252963)

[Task 2 - View records by criteria using WHERE= 18](#_Toc500252964)

[Task 3 - View records by criteria using LIKE ‘%keyword%’ 19](#_Toc500252965)

[F. Update records 20](#_Toc500252966)

[Task 1 – Update one column in a selected record 20](#_Toc500252967)

[Use the UPDATE SQL statement to update a record 20](#_Toc500252968)

[Task 2– Update multiple columns in a selected record 20](#_Toc500252969)

[Use the UPDATE SQL statement to update a record 20](#_Toc500252970)

[G. Delete records 21](#_Toc500252971)

[Task 1 – Delete a specific record using DELETE WHERE 21](#_Toc500252972)

[5. Write Python code to connect to MySQL 22](#_Toc500252973)

[A. Install mysql-connector-python package 22](#_Toc500252974)

[Use conda to install the mysql-connector-python library 22](#_Toc500252975)

[B. Display records from MySQL 23](#_Toc500252976)

[C. Create new table 23](#_Toc500252977)

[D. Insert records 24](#_Toc500252978)

[E. Update records 25](#_Toc500252979)

[F. Delete records 26](#_Toc500252980)

# Overview

## What you will do for this lab

In this lab, you will learn how to set up your own relational database server on your laptop and manipulate the data stored in the database using SQL statements as well as Python code.

## Intro to mySQL

MySQL is a leading open source database management system. It is a multi-user, multithreaded database management system. MySQL is especially popular on the web. It is part of the very popular LAMP stack: **L**inux, **A**pache, **My**SQL and **P**HP. MySQL is available on Linux, Windows or Mac.

# Install mySQL

## Windows Installation

| No | Task | |
| --- | --- | --- |
|  | If you are on the Microsoft Windows operating system, click the link below to download the mySQL installer.  It is almost 400mb, so please be patient for it to be downloaded completely.  <https://dev.mysql.com/downloads/file/?id=470091> | |
|  | Once the file has been downloaded to your laptop, double-click on the file to start the installation process. | |
|  | You can use most of the defaults when installing.  However, do note that you should ensure that the following four options of products are included |  |
|  | Also, if you are prompted for the MySQL root password, don’t forget to note it down in case you forget it later. |  |

## MacOS Installation

| No | Task |
| --- | --- |
|  | Visit the mySQL download website  <https://dev.mysql.com/downloads/file/?id=486026>  Click on the appropriate version for your operating system. |

# Start / Stop mySQL

## On Windows (command-line - OPTIONAL)

### Start mySQL

| No | Task | | |
| --- | --- | --- | --- |
|  | Ensure that you have a sub-folder named data under your install location of MySQL as shown on the right | |  |
|  | The MySQL server (mysqld.exe) can be started manually from the command line. This can be done on any version of Windows.  To start the mysqld server from the command line, you should start a console window (or “DOS window”) with **ADMINISTRATIVE rights** and enter the mysqld command | The path to mysqld may vary depending on the install location of MySQL on your system.  "C:\Program Files\MySQL\MySQL Server 5.7\bin\mysqld.exe" | |

### Stop mySQL

| No | Task |
| --- | --- |
|  | You can stop the MySQL server by executing the command as shown below.  If the MySQL root user account has a password, you need to invoke mysqladmin with the -p option and supply the password when prompted.  "C:\Program Files\MySQL\MySQL Server 5.7\bin\mysqladmin" -u root shutdown” |

### Troubleshooting

| No | Task |
| --- | --- |
|  | * If mysqld doesn't start successfully, check the error log to see whether the server wrote any messages there to indicate the cause of the problem * By default, the error log is located in the C:\Program Files\MySQL\MySQL Server 5.7\data directory with a suffix of .err * Alternatively, you can try to start the server with the --console option; in this case, the server may display some useful information on the screen that will help solve the problem * The last option is to start mysqld with the --standalone and --debug options. In this case, mysqld writes a log file C:\mysqld.trace that should contain the reason why mysqld doesn't start * Use mysqld --verbose --help to display all the options that mysqld supports. |

## On Windows (mySQL Notifier)

### What is MySQL Notifier?

| No | Task | |
| --- | --- | --- |
|  | MySQL Notifier is a tool that enables you to monitor and adjust the status of your MySQL server instances through an indicator that resides in the system tray. MySQL Notifier also gives quick access to MySQL Workbench through its context menu. | The MySQL Notifier is installed by MySQL Installer, and (by default) will start-up when Microsoft Windows is started. |

## On MacOS

| No | Task | |
| --- | --- | --- |
|  | OS X uses launch daemons to automatically start, stop, and manage processes and applications such as MySQL.  To enable the launchd service, click Start MySQL Server from the MySQL preference pane. | MySQL Preference Pane: Location |
|  | You will be presented with the opening installer dialog. Click Continue to begin installation. | |
|  | MySQL Preference Pane: Usage | |

# Use MySQL Workbench

## Connect to your database server

### Start mySQL Workbench

|  |  | |
| --- | --- | --- |
|  | Using the search function on your laptop, start up the MySQL Workbench software |  |

### Create a new MySQL Connection

|  |  |
| --- | --- |
|  | Click on the MySQL connections (+) button to create a connection to your server. |
|  | |
|  | |

### Test connection to localhost

|  |  |
| --- | --- |
|  | Key in the parameters as shown below, including the password you have used when installing mySQL, then click the “Test Connection” button.  If all goes well, then proceed to the next section.   |  |  | | --- | --- | | **Connection Name** | Localhost | | **Connection Method** | Standard (TCP/IP) | | **Hostname** | 127.0.0.1 | | **Port** | 3306 | | **Username** | root | |
|  | |

## Create a new database

### Connect to localhost

|  | Task |
| --- | --- |
|  | Double-click on the localhost connection you have just created in the previous section  You should be brought to a screen that looks similar to this |
|  | |

### Use the CREATE DATABASE SQL statement

|  | Task | | | |
| --- | --- | --- | --- | --- |
|  | Type the following SQL query statement in the Query 1 window   |  | | --- | | create database mydatabase | | | |  |
|  | Click the “lightning” icon beside the save button to execute the statement you typed above. | | | |
|  | If all went well, you should see the following message in the Output window. |  | | |
|  | Click the refresh button near the top right hand corner of the SCHEMAS pane  You should see your newly created database in the list. | |  | |

## Create a new table

### Use the CREATE TABLE SQL statement

|  | Task | | |
| --- | --- | --- | --- |
|  | Press Ctrl-T to start a new Query tab.  Type the following SQL query statement in the Query tab window. | |  | | --- | | use mydatabase;  CREATE TABLE Customer (  CustomerID int NOT NULL,  CustomerName varchar(50) NOT NULL,  CustomerAddress varchar(99) NOT NULL,  CustomerPhoneNumber varchar(25) NOT NULL,  Country varchar(25) NULL,  PRIMARY KEY (CustomerID)  ); | | |
|  | Click the “lightning” icon beside the save button to execute the statement you typed above. | | |
|  | If all went well, you should be able to see the newly created table in the Schemas pane when you do a refresh | |  |

## Insert new records

### Insert one record using INSERT INTO

|  | Task |
| --- | --- |
|  | Press Ctrl-T to start a new Query tab.  Type the following SQL query statement in the Query tab window. |
| |  | | --- | | use mydatabase;  INSERT INTO Customer (CustomerID, CustomerName, CustomerAddress, CustomerPhoneNumber)  VALUES (1, 'Adampak Limited', '6 Loyang Way 4 Singapore 507605', '+6565539767'); | | |
|  | Click the “lightning” icon beside the save button to execute the statement you typed above. |
|  | If all went well, you should see the following message in the Output window. |
|  | |

### Insert multiple records using INSERT INTO

|  | Task |
| --- | --- |
|  | Press Ctrl-T to start a new Query tab.  Type the following SQL query statements in the Query tab window. |
| |  | | --- | | use mydatabase;  INSERT INTO Customer (CustomerID, CustomerName, CustomerAddress, CustomerPhoneNumber)  VALUES (2, 'Ban Leong Technologies Limited', '21 Kaki Bukit Cres Singapore 416252', '+6565482343');  INSERT INTO Customer (CustomerID, CustomerName, CustomerAddress, CustomerPhoneNumber)  VALUES (3, 'CH Offshore Ltd', '388 Jln Ahmad Ibrahim Singapore 629157', '+6565703336');  INSERT INTO Customer (CustomerID, CustomerName, CustomerAddress, CustomerPhoneNumber)  VALUES (4, 'Dragon Group International Limited', '25 Kallang Ave Singapore 339416', '+6565379406');  INSERT INTO Customer (CustomerID, CustomerName, CustomerAddress, CustomerPhoneNumber)  VALUES (5, 'Eu Yan Sang International Ltd', 'Eu Yan Sang Centre 21 Tai Seng Drive Singapore 535223', '+6565474002');  INSERT INTO Customer (CustomerID, CustomerName, CustomerAddress, CustomerPhoneNumber)  VALUES (6, 'Fischer Tech Ltd', '12 Loyang Way 4 Singapore 507602', '+6565829983');  INSERT INTO Customer (CustomerID, CustomerName, CustomerAddress, CustomerPhoneNumber)  VALUES (7, 'GMG Global Ltd', '8 Marina View Asia Square Tower 1 Singapore 018960', '+6565642539');  INSERT INTO Customer (CustomerID, CustomerName, CustomerAddress, CustomerPhoneNumber)  VALUES (8, 'Hyflux Limited', '202 Kallang Bahru Singapore 339339', '+6565776668');  INSERT INTO Customer (CustomerID, CustomerName, CustomerAddress, CustomerPhoneNumber)  VALUES (9, 'Innotek Limited', '1 Finlayson Green Singapore 049246', '+6565549383');  INSERT INTO Customer (CustomerID, CustomerName, CustomerAddress, CustomerPhoneNumber)  VALUES (10, 'Jadason Enterprises Ltd', '3 Kaki Bukit Cres Singapore 416237', '+6565719466'); | | |
|  | Click the “lightning” icon beside the save button to execute the statement you typed above. |
|  | If all went well, you should see the following message in the Output window. |
|  | |

## View records

### Task 1 - View all records using SELECT

|  | Task |
| --- | --- |
|  | Press Ctrl-T to start a new Query tab.  Type the following SQL query statements in the Query tab window. |
| |  | | --- | | use mydatabase;  SELECT \* from Customer; | | |
|  | Click the “lightning” icon beside the save button to execute the statement you typed above. |
|  | If all went well, you should see the following message in the Result Grid window. |
|  | |

### Task 2 - View records by criteria using WHERE=

|  | Task |
| --- | --- |
|  | Press Ctrl-T to start a new Query tab.  Type the following SQL query statements in the Query tab window. |
| |  | | --- | | use mydatabase;  SELECT \* from Customer WHERE CustomerID=5 | | |
|  | Click the “lightning” icon beside the save button to execute the statement you typed above. |
|  | If all went well, you should see the following message in the Result Grid window. |
|  | |

### Task 3 - View records by criteria using LIKE ‘%keyword%’

|  | Task |
| --- | --- |
|  | Press Ctrl-T to start a new Query tab.  Type the following SQL query statements in the Query tab window. |
| |  | | --- | | use mydatabase;  SELECT \* from Customer WHERE CustomerAddress LIKE '%Loyang%' | | |
|  | Click the “lightning” icon beside the save button to execute the statement you typed above. |
|  | If all went well, you should see the following message in the Result Grid window. |
|  | |

## Update records

### Task 1 – Update one column in a selected record

|  | Task |
| --- | --- |
|  | Use the UPDATE SQL statement to update a record Press Ctrl-T to start a new Query tab.  Type the following SQL query statements in the Query tab window |
| |  | | --- | | use mydatabase;  ## Update telephone number of Customer ID 1 (Adampak)  UPDATE Customer  SET CustomerPhoneNumber='+6565539766'  WHERE CustomerID=1 | | |
|  | Click the “lightning” icon beside the save button to execute the statement you typed above. |
|  | If all went well, you should see the following message in the **Output** window. |
|  | |

### Task 2– Update multiple columns in a selected record

|  | Task |
| --- | --- |
|  | Use the UPDATE SQL statement to update a record Press Ctrl-T to start a new Query tab.  Type the following SQL query statements in the Query tab window |
| |  | | --- | | use mydatabase;  ## Update country and name of Customer ID 1 (Adampak)  UPDATE Customer  SET Country='Singapore', CustomerName='Adampak Pte Ltd'  WHERE CustomerID=1 | | |
|  | Click the “lightning” icon beside the save button to execute the statement you typed above. |
|  | If all went well, you should see the following message in the **Output** window. |
|  | |

## Delete records

### Task 1 – Delete a specific record using DELETE WHERE

|  | Task |
| --- | --- |
|  | Press Ctrl-T to start a new Query tab.  Type the following SQL query statements in the Query tab window |
| |  | | --- | | use mydatabase;  DELETE FROM Customer  WHERE CustomerID=10 | | |
|  | Click the “lightning” icon beside the save button to execute the statement you typed above. |
|  | If all went well, you should see the following message in the **Output** window. |
|  | |

# Write Python code to connect to MySQL

Now that you have mastered the basics of using SQL statements to view and manipulate the data in a relational database such as MySQL, let’s hone your skills further by writing Python code that reads data from various CSV file, stores the data into your MySQL database and then retrieves it and shows it as a visualization based on the user’s inputs.

## Install mysql-connector-python package

In order to write Python code that can interface with the mySQL database, you need to install the mysql-connector-python library first. Follow the instructions below to install it on your laptop.

### Use conda to install the mysql-connector-python library

|  |  |
| --- | --- |
|  | Run the following command in a terminal window. If the command runs successfully, Anaconda will start to install the library and your package will be ready for use in less than 60 seconds.  conda install mysql-connector-python |

## Display records from MySQL

Once you have installed the mysql-connector-python package successfully, you may start to connect to your MySQL database using Python code.

Execute the example below to connect to the Customers table in mydatabase and display the records inside the table.

|  |  |
| --- | --- |
|  | Type the following code in a new cell in Jupyter |

|  |
| --- |
| import mysql.connector  import pandas as pd  user,pw, host,db = 'root','<type yr password here>','127.0.0.1','mydatabase'  cnx = mysql.connector.connect(user=user, password=pw, host=host, database=db)  sql ='SELECT \* FROM Customer'  df = pd.read\_sql(sql, con=cnx)  print(df) |

## Create new table

Execute the example below to learn how to create a new table inside the database

|  |  |
| --- | --- |
|  | Type the following code in a new cell in Jupyter |

|  |
| --- |
| import mysql.connector,sys  user,pw, host,db = 'root', 'yourpasswordhere','127.0.0.1','mydatabase'  cnx = mysql.connector.connect(user=user, password=pw, host=host, database=db)  cursor = cnx.cursor()  query\_for\_creating\_table = ("CREATE TABLE `cea\_salespersons` ("  "`cea\_salesperson\_id` int(11) NOT NULL AUTO\_INCREMENT,"  "`salesperson\_name` varchar(100) NOT NULL,"  "`registration\_no` varchar(50) NOT NULL,"  "`registration\_start\_date` date NOT NULL,"  "`registration\_end\_date` date NOT NULL,"  "`estate\_agent\_name` varchar(100) NOT NULL,"  "`estate\_agent\_license\_no` varchar(50) NOT NULL,"  "PRIMARY KEY (`cea\_salesperson\_id`)"  ") ENGINE=InnoDB")  try:  print("Executing CREATE TABLE command")  cursor.execute(query\_for\_creating\_table)  cnx.commit()  print("Table created!")  except:  print("Unexpected error:", sys.exc\_info()[0])  exit()  finally:  cursor.close()  cnx.close() |

## Insert records

Execute the example below to learn how you can read a CSV file that contains several records, and store these records in MySQL in the table you have created in Section C.

|  |  |
| --- | --- |
|  | Ensure the “cea-salesperson-information-utf8.csv” is inside your data folder.  Type the following code in a new cell in Jupyter |

|  |
| --- |
| import mysql.connector,sys, pandas as pd  from datetime import date, datetime, timedelta  from dateutil.parser import parse  df = pd.read\_csv("data/cea-salesperson-information-utf8.csv")  user,pw, host,db = 'root','yourpasswordhere','127.0.0.1','mydatabase'  cnx = mysql.connector.connect(user=user, password=pw, host=host, database=db)  cursor = cnx.cursor()  for index, col in df.iterrows():  startdate = parse(col[2]).strftime("%d/%m/%Y")  enddate = parse(col[3]).strftime("%d/%m/%Y")  dt\_obj\_startdate = datetime.strptime(startdate, "%d/%m/%Y")  dt\_obj\_enddate = datetime.strptime(enddate, "%d/%m/%Y")    data = {  'salesperson\_name': col[0],  'registration\_no': col[1],  'registration\_start\_date': dt\_obj\_startdate,  'registration\_end\_date': dt\_obj\_enddate,  'estate\_agent\_name' : col[4],  'estate\_agent\_license\_no': col[5]  }      query = insert\_stmt = ("INSERT INTO cea\_salespersons (salesperson\_name, registration\_no, registration\_start\_date, registration\_end\_date,estate\_agent\_name, estate\_agent\_license\_no)"  "VALUES (%(salesperson\_name)s, %(registration\_no)s, %(registration\_start\_date)s, %(registration\_end\_date)s,%(estate\_agent\_name)s, %(estate\_agent\_license\_no)s)")  # Insert new cea\_salesperson  cursor.execute(query, data)  print("Adding row " + str(index))  # Make sure data is committed to the database  cnx.commit()    print("All data inserted!")  # We are done, let's close the connection  cursor.close()  cnx.close() |

## Display records from MySQL

Execute the example below to connect to the cea\_salespersons table in mydatabase and display the records inside the table.

|  |  |
| --- | --- |
|  | Type the following code in a new cell in Jupyter |

|  |
| --- |
| import mysql.connector  import pandas as pd  user,pw, host,db = 'root','<type yr password here>','127.0.0.1','mydatabase'  cnx = mysql.connector.connect(user=user, password=pw, host=host, database=db)  sql ='SELECT \* FROM cea\_salespersons '  df = pd.read\_sql(sql, con=cnx)  print(df) |

## Update records

Execute the example below to learn how you can update existing records in the table you created and populated with records in the previous sections.

|  |  |
| --- | --- |
|  | Ensure the “cea-salesperson-information-utf8.csv” is inside your data folder.  Type the following code in a new cell in Jupyter |

|  |
| --- |
| import mysql.connector  from datetime import date, datetime, timedelta  user,pw, host,db = 'root', 'yourpasswordhere','127.0.0.1','mydatabase'  cnx = mysql.connector.connect(user=user, password=pw, host=host, database=db)  cursor = cnx.cursor()  today = datetime.now().date()  tomorrow = datetime.now().date() + timedelta(days=1)  data\_cea\_salesperson = {  'registration\_end\_date': datetime.strptime('31/12/2017', "%d/%m/%Y")  }  update\_cea\_salesperson = ("UPDATE cea\_salespersons SET registration\_end\_date = %(registration\_end\_date)s")  cursor.execute(update\_cea\_salesperson, data\_cea\_salesperson)  # Make sure data is committed to the database  cnx.commit()  # We are done, let's close the connection  cursor.close()  cnx.close() |

## Delete records

Execute the example below to learn how you can delete existing records in the table you created and populated with records in the previous sections.

|  |  |
| --- | --- |
|  | Ensure the “cea-salesperson-information-utf8.csv” is inside your data folder.  Type the following code in a new cell in Jupyter |

|  |
| --- |
| import mysql.connector, sys  user,pw, host,db = 'root','yourpasswordhere','127.0.0.1','mydatabase'  cnx = mysql.connector.connect(user=user, password=pw, host=host, database=db)  cursor = cnx.cursor()  delete\_stmt = ("DELETE FROM cea\_salespersons WHERE salesperson\_name = %(salesperson\_name)s")  data = { 'salesperson\_name': 'AARON YEO YI'}  try:  print("Going to run the DELETE script")  cursor.execute(delete\_stmt, data)  cnx.commit()  print("Delete done!")  except:  print("Unexpected error:", sys.exc\_info()[0])  exit()  finally:  cursor.close()  cnx.close() |

**-- End of Lab --**