# **Labs for Automating Database Tasks Using Python Programming Language**

**Duration:** 90 minutes. Please note there are three parts in this lab.

## Objective

This is a short exercise to practice executing SQL command using Python.

It is not meant to be challenging but just to get you to use some of the techniques mentioned in this chapter.

## Part 1 – Create Table

Use the slide code to create a table called Student for storing student data.

Check that the table is created using the SQL Management Studio.

## Part 2 – Practice Executing Insert Commands

Insert a few students' records into the Student table.

Be inventive! You can store the records in a CSV file. Read the data and then insert the data into the database.

You can use the names of the students in the class and details of their course and the city they live in.

Check the records are inserted using the Database Management Studio.

## Part 3 – Practice Executing an Update Command

Update the record of one of the students.

Check that it has worked by reading the data in the Management Studio.

**\*\* End \*\***

**import pyodbc**

**connectionString = r'DRIVER={ODBC Driver 13 for SQL Server};SERVER=.\SQLExpress;DATABASE=northwind;Trusted\_Connection=yes'**

**sqlStr = "SELECT \* FROM customers ORDER BY contactName"**

**conn = pyodbc.connect(connectionString)**

**cur = conn.cursor()**

**result = cur.execute(sqlStr).fetchall()**

**conn.close()**

**for row in result:**

**print(row)**

**import pyodbc**

**connectionString = r'DRIVER={ODBC Driver 13 for SQL Server};SERVER=.\SQLExpress;DATABASE=northwind;Trusted\_Connection=yes'**

**sqlStr="""CREATE TABLE Student (**

**StudentID int NOT NULL,**

**FirstName nvarchar(40) NOT NULL,**

**Surname nvarchar(30) NULL,**

**Course nvarchar(30) NULL,**

**City nvarchar(15) NULL)"""**

**conn = pyodbc.connect(connectionString)**

**cur = conn.cursor()**

**cur.execute(sqlStr)**

**conn.commit()**

**conn.close()**