# Introduction to Databases with VB.net and MySQL

### Aim

* Experience how data can be stored in a database.
* Be able to Create, and alter the structure of the database using DDL
* Be able to Add, Edit, and retrieve data using SQL
* Perform the above operations with VB.Net (or the language of your choice)
* Be able to visualise the structure of the data

### Tasks

You are going to create a database using VB.net and DDL (Data Definition Language)… We are going to use a MySQL database server. XAMPP provides a fairly lightweight set of servers Apache Webserver, MySQL database server & PHP interpreter. Eventually we will create a Database driven web project.. But to get us started we will create another vb.net forms project that uses a database.

### Before we begin

I have provided a zip with all the components sorted and the username and passwords set. The connector is downloaded and placed in the MYSQL folder. This should work for any windows machine.

* Download Xampp from GoL and unzip into the root of your C:/drive
* Unzip the compressed file to a folder in the root of you C:/drive
* Open the folder and run Xampp-Control: the following should appear
* Click start on Apache and MysqL
you may need to allow some firewall changes.
You may need to run “as administrator”



Start Apache and MySql: Your computer is now a web server and a database server.

To test this further Click on the Admin button for MySqL. It should open PHPMyAdmin. A web based Database management system. We are going to use this a lot! It should look something like this:



There is very little security on this system. Using it behind your home router is fine, but be careful not to expose the ports to the internet without careful consideration.

We are now going to create a MySql Database using Date Definition Language in VB.net

### Task1 – Create VB Form Project

The Database we are creating will store student exam results. There will be several forms but the first will simple be there to create the database.

1. Create a new Forms Project. For ease you can create a new button for each of the following tasks. So by the end you will have a VBform with lots of buttons on it…
2. Name the buttons something sensible (e.g. btnCreateDataBase).. You should also alter the display text so each button can be identified when the program runs.
3. You need to add the following reference to the project
	1. MySql.Data.Client (Project🡪 Add reference 🡪Browse🡪 find the file in Xampp/MySql)
4. Now type the following to make use of this library

Imports MySql.Data.MySqlClient

We’re going to be a smidge lazy and have a few globals (or are they properties?) that we will use throughout. Put these at the top of the form class.

 Dim MyCommand As MySqlCommand

 Dim MyConnection As MySqlConnection

 Dim MyReader As MySqlDataReader

 Dim DDLstr As String

 Dim SQLstr As String

 Dim ConnStr As String = "server=localhost;user=CPBuser;port=3306;password=VBmysql123;"

Google: “localhost”, The User and Password I have set-up for you already. If you want to change it use phpmyadmin. Don’t worry about these at the moment.. all will become clear.

### Task 2 - Create a Database

Go back to the form and double click on the button to create the button.cliock event.

Use the following code to create a new database…

 Private Sub BtnCreateDataBase\_Click(sender As Object, e As EventArgs) Handles btnCreateDataBase.Click

 Try

 MyConnection = New MySqlConnection(ConnStr) ' creates a new connection using the string

 MyConnection.Open() ' opens the connection

 MsgBox("open :)") ' just a msg so we know it's worked

 DDLstr = "CREATE DATABASE IF NOT EXISTS `StudentDB`;" ' the DDL command that needs to be sent to the server

 MyCommand = New MySqlCommand(DDLstr, MyConnection) 'sends the command through the connection

 MyCommand.ExecuteNonQuery() ' tells the sever to execute the command

 MsgBox("created :)") ' 'msg so we know it has worked

 Catch ex As Exception

 MsgBox(ex.Message) ' msg giving us the error that came back from the server

 End Try

 MyConnection.Close() ' don't forget to close the connection!

 End Sub

Execute the code and click on the button… Hopefully you get smilie faces and it works. Open phpMyAdmin you should now see a database called “StudentDB”. (NB. Database engine are VERY fussy. Wrong speech marks or any syntax slightly wrong will throw an error!)

### Task 3 Opening the connection to the Database

From now on before you execute the code for a following tasks the database connection needs to be opened… and after needs to be closed. The connection string needs slightly changing to point to the database. Having a separate method with its own exception handling will save debugging time in the future (especially if you forget to turn the database server on)
Add a database parameter to the ConnStr string. (how are you going to test this new sub?)

Dim ConnStr As String = "server=localhost;Database=StudentDB;user=CPBuser;port=3306;password=VBmysql123;"

 Sub OpenDB()

 Try

 MyConnection = New MySqlConnection(ConnStr)

 MyConnection.Open()

 Catch ex As Exception

 MsgBox(ex.Message)

 End Try

 End Sub

### Task 4 – Creating tables

Now we are going to use more DDL commands We will be creating 3 tables: **Students, Exams, Results**

I’ll give you the code for **Students** and specify the other 2 tables for you to code

 Sub CreateStudentTable()

 'this is the actual DDL command string we are sending to the database engine. The exact syntax (brackets, quaotes etc..) varies slightly for each type of database.. (this is all one line)

 DDLstr = "CREATE TABLE Students

(Username Varchar(20) NOT NULL PRIMARY KEY,

FirstName Varchar(25),

Surname Varchar(25))"

 Try

 MyCommand = New MySqlCommand(DDLstr, MyConnection)

 MyCommand.ExecuteNonQuery()

 MsgBox("Students Table created")

 Catch ex As Exception

 MsgBox("Students Table not Created " & ex.Message)

 End Try

 End Sub



Now create the Exams Table, which will hold information about the different exam each subject has

|  |  |
| --- | --- |
| **Field** | **Properties** |
| ExamCode | Varchar(6) , Primary Key |
| ExamTitle | Varchar(25) |
| Subject | Varchar(25) |

Now create the Results Table

|  |  |
| --- | --- |
| **Field** | **Properties** |
| ResultID | Varchar(6) , Primary Key |
| UserName | Varchar(20) |
| ExamCode | Varchar(6) |
| Score | Int |

We will be adding extra fields later to the 3 table

View the database through phpmyadmin it should look like this

### Task 5 Adding Fields to a table

Here is an example of some code that adds a field to an existing table:

 Sub CreateField()

 'Purpose: Illustrates how to add a field to a table using DDL.

 Try

 DDLstr = "ALTER TABLE Exams ADD COLUMN Weight REAL"

 MyCommand = New MySqlCommand(DDLstr, MyConnection)

 MyCommand.ExecuteNonQuery()

 MsgBox("Field Added Sucessfully")

 Catch ex As Exception

 MsgBox("Field NOT Added Sucessfully " & ex.Message)

 End Try

 End Sub

Now add the following fields

* RawScore to Results as a Real number
* MaxScore to Exam with type integer
* DateofBirth to Students with type date
* Gender to Students Table. (you choose how to encode this)
* ExamYear to Results (storing the year as an integer)

### Task 6 Adding data to a table

Let’s start with adding some hard coded data (you will never do this in practice but that way you can see the syntax… (the strings are made smaller to fit on one line… We will look at ways to make you code more readable later)

Sub InsertOneRecord() 'A literal SQL string

 ' we are now using SQL rather than DDL

 ' here is an insertion of one record using some hard coded values

 Dim SQLstr As String

 Try

 SQLstr = " INSERT INTO Students(Username, FirstName, Surname) VALUES ('BB','Robert','theBuilder')"

 MyCommand = New MySqlCommand(SQLstr, MyConnection)

 MyCommand.ExecuteNonQuery()

 MsgBox("inserted")

 Catch ex As Exception

 MsgBox(ex.Message)

 End Try

End Sub

Make sure you understand the syntax…

* why are bits of the string in square brackets?
* Why are sections in single quotes?

Now we will do the same thing again BUT with data stored in variables

 Sub InsertAnotherRecord()

 ' Building SQL strings with variables takes care..... you need to be alert... the world needs more lerts

 Dim SQLstr As String

 Dim UName As String = "WB"

 Dim fname As String = "Wendy"

 Dim sName As String = "theBuilder" ' are bob and wendy married?

 Try

 'Make sure you REALLY understand how this concatentaion is working!

 SQLstr = "INSERT INTO Students(Username, FirstName, Surname) VALUES ('" & UName & "','" & fname & "','" & sName & "')"

 MyCommand = New MySqlCommand(SQLstr, MyConnection)

 MyCommand.ExecuteNonQuery()

 Console.WriteLine("inserted")

 Catch ex As Exception

 Console.WriteLine(ex.Message)

 End Try

 End Sub

**NB. This method is vulnerable to SQL attacks… we will learn how to protect against this later.**

Alter the Procedure to add the following data to the EXAMS Table. The column headings are the exact field names..

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ExamCode** | **ExamTitle** | **Subject** | **Weight** | **MaxScore** |
| COMP999 | Geekdom | Computing | 0.6 | 100 |

Add Textboxes to your MainForm to allow you to enter data into the Students and Exams Tables.

(probably best to handle as separate events )

### Task 7 Adding data to the database from CSV files.

It is often useful to have sets of test data stored as text files. As it happens most DBMS also have CSV import… but that would be no fun! It gives us a chance to see a sound way to manipulate files in .net.

Here is an example for the Student database… Make sure this inserts correctly

  Sub InserttoStudentFromCsv()

 ' here is an insertion of a set of records taken from a correctly formatted CSV file.

 'You need to have put the CSV files into the Debug bin folder

 'get them here <https://drive.google.com/file/d/0B1nECgSBg0g7S0NDamxOTDFWM0E/view?usp=sharing>

 Dim SQLstr As String

 Dim SR As New StreamReader("students.csv") 'hope you have this file in the debug bin folder!

 Dim txtLine As String

 Dim FieldStr As String()

 Do Until SR.EndOfStream ' keeps going until end of file

 Try

 txtLine = SR.ReadLine()

 FieldStr = txtLine.Split(",")

 SQLstr = "INSERT INTO Students(Username,FirstName, Surname) VALUES ('" & FieldStr(0) & "','" & FieldStr(1) & "','" & FieldStr(2) & "')"

 MyCommand = New MySqlCommand(SQLstr, MyConnection)

 MyCommand.ExecuteNonQuery()

 Console.WriteLine("inserted")

 Catch ex As Exception

 Console.WriteLine(ex.Message)

 End Try

 Loop

 End Sub

Alter the above code to insert the csv data for the Exams and Results Tables…

**Congratulations! You should have a database with 3 tables and enough data for us to play with.**