Anthony Meunier

DeVry University

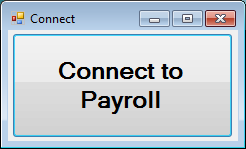
DBM 405A

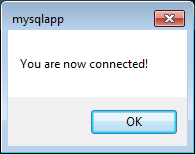
Week 7 Course Project

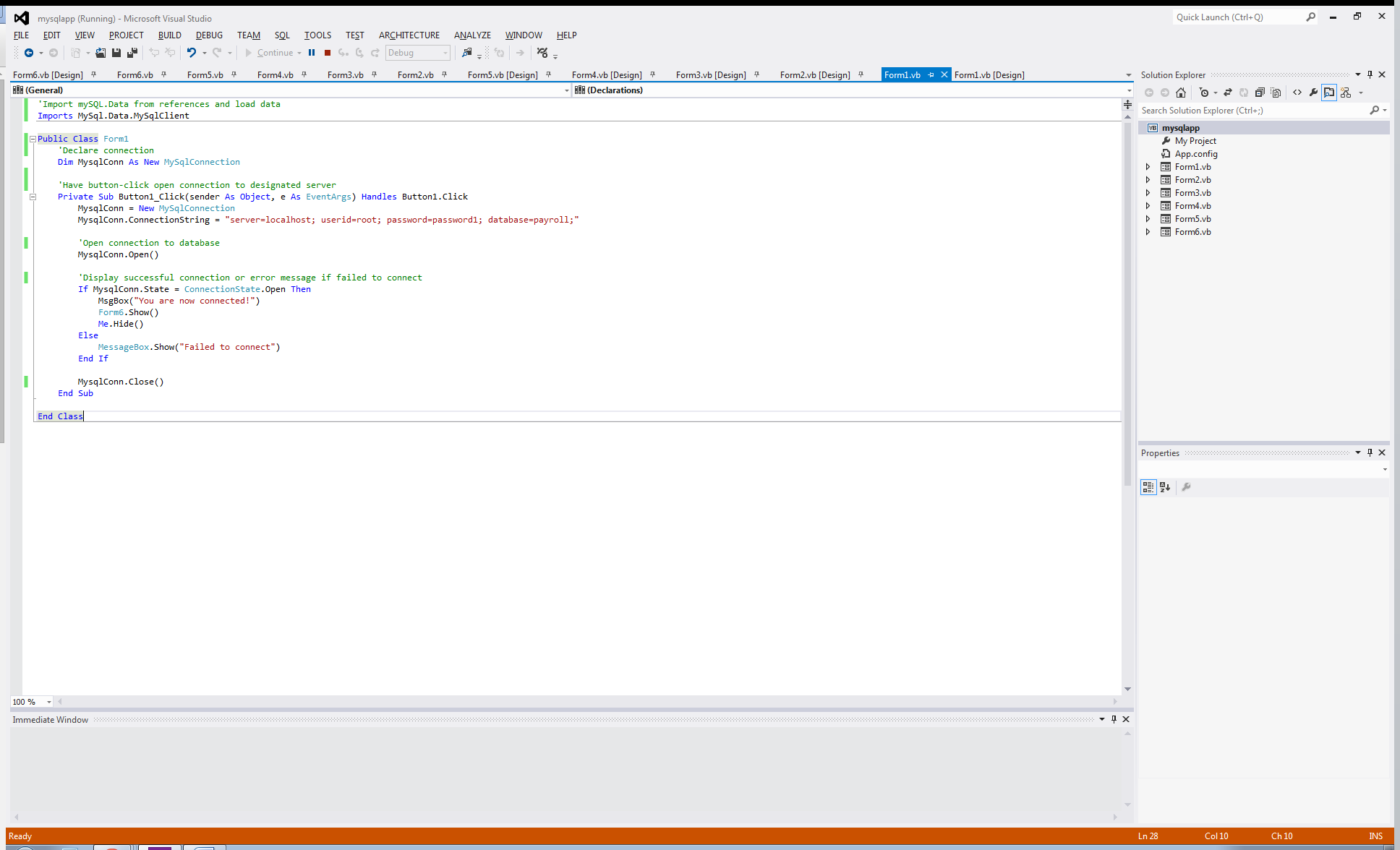
Write the Report

**Interface**

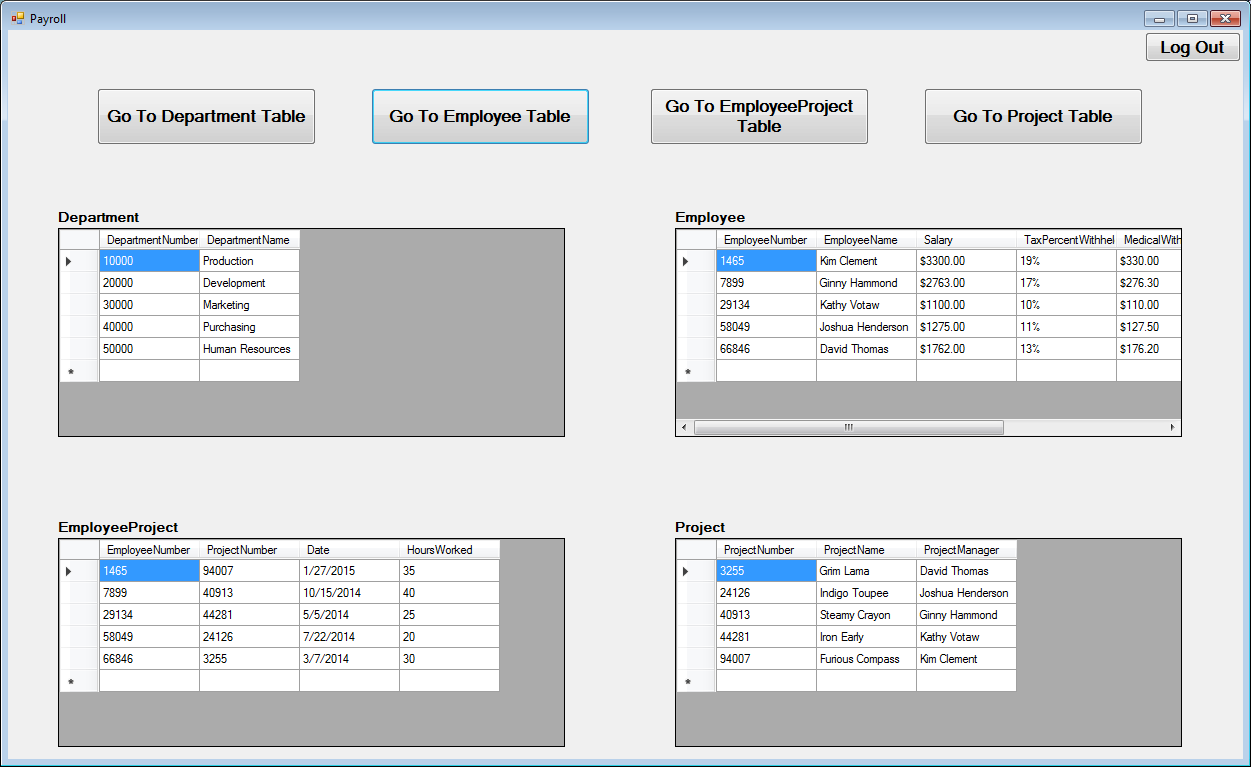
For the interface of the application, I created a GUI via Visual Basic with Windows Forms. Each button functions to take the user to the entitled screen when the button is clicked. In addition, each individual form pertaining to its corresponding table loads the data on its own when the form is loaded. There is also obviously an established connection from Visual Studio to the MySQL server which functions to pull of the data in (ability to add, modify, and delete will be in future project). ‘Log Out’ button takes the user back to the ‘Connect’ screen, and ‘View All’ returns user to the main menu where they can view all data as well as navigate to other table pages.

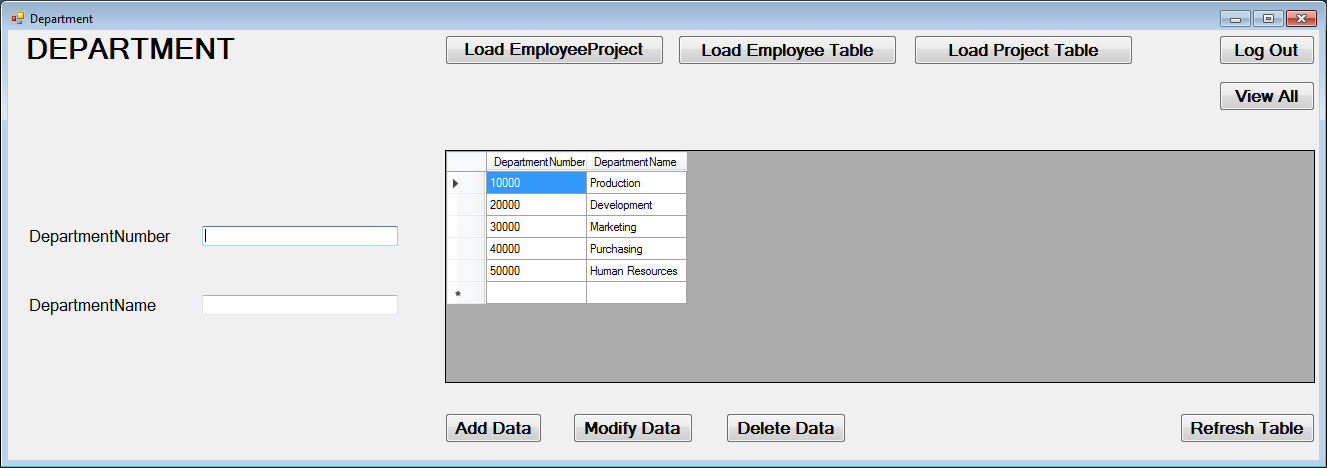


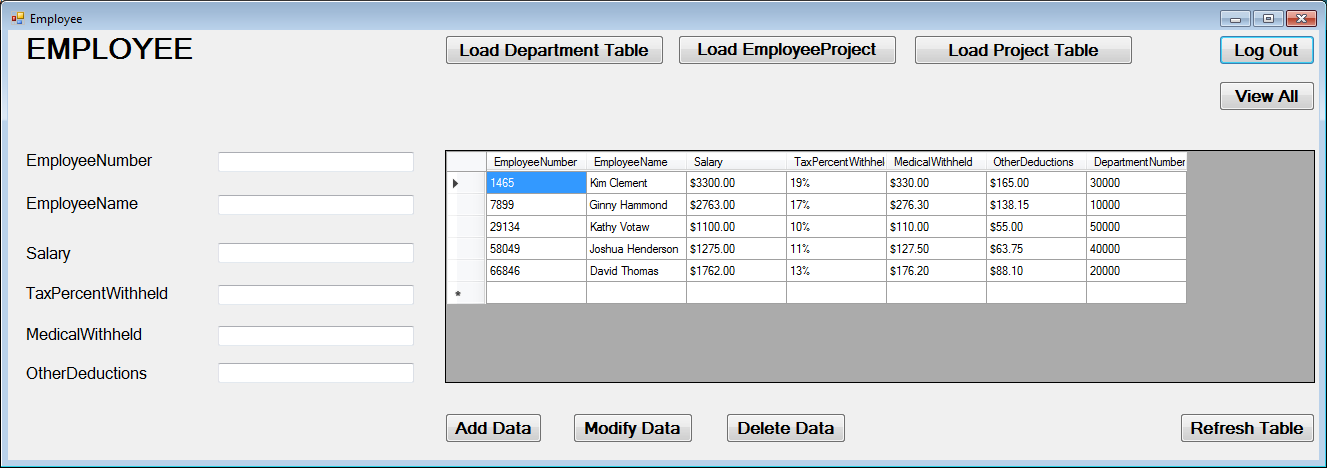


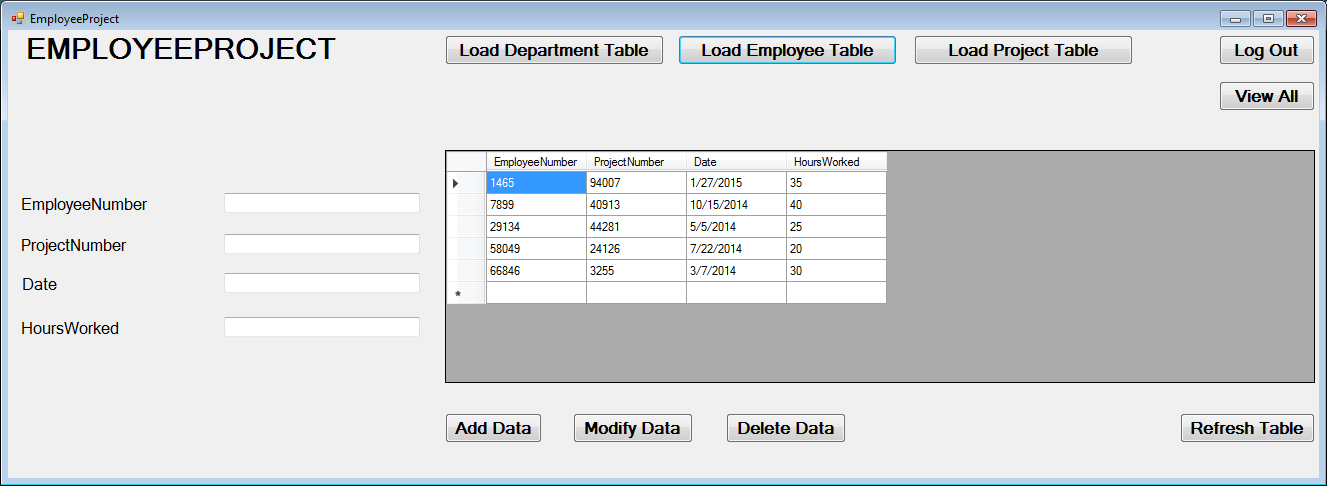


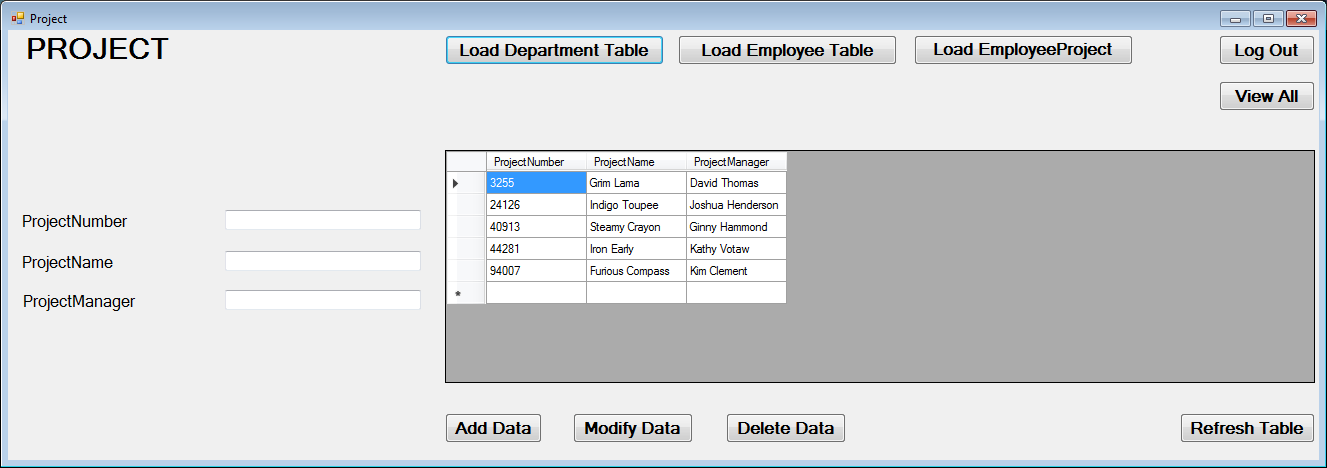
Please note the Imports MySql.Data.MySqlClient statement at the top of the code, as well as the connection string MysqlConn.ConnectionString = "server=localhost; userid=root; password=password1; database=payroll;" that references my specific MySQL server and database. Upon successful execution of this statement we have the ensuing MysqlConn.Open() that connects out application to the database for use/reference in Visual Studio.











**Add/Delete/Update**

Writing the add/change/delete routines was fairly straightforward. Because I am using Windows Forms pages to serve as the interface for each respective table in the database, I simply had to write the routines one time (add, update/modify, delete) and then transpose them into the other Forms pages and then change the appropriate text to have them match the appropriate table and columns within the database. I have included in these routines a try/catch block that will either show an error if the data cannot be manipulated in the desired way or a window pop-up that displays a message to notify the user that changes have been made. In the example screenshots I will attach, you can see this process for each of the three routines. First I show the table as-is, type in the new information, press the appropriate button, display the successful message, show the table after the refresh button has been pressed, and also note the changes in the database application (Workbench) itself after we refresh in there. For the sake of space, I showed each of the routines functioning in one of the project tables, but since the code is appropriately updated in all Forms pages we can know that the routines are effective there as well.

Here I will paste each of the routines from Visual Basic for my “Project” table of the database. The attached project files will show that the code is exactly the same for all other tables, just updated accordingly to match those table’s names so the correct information is updated:

'Add data on button click

Private Sub Button3\_Click(sender As Object, e As EventArgs) Handles Button3.Click

MysqlConn = New MySqlConnection

MysqlConn.ConnectionString = "server=localhost; userid=root; password=password1; database=payroll;"

Dim READER As MySqlDataReader

Try

MysqlConn.Open()

Dim Query As String

Query = "insert into payroll.project (ProjectNumber, ProjectName, ProjectManager) values ('" & TextBox\_ProjectNumber.Text & "', '" & TextBox\_ProjectName.Text & "', '" & TextBox\_ProjectManager.Text & "')"

COMMAND = New MySqlCommand(Query, MysqlConn)

READER = COMMAND.ExecuteReader

MessageBox.Show("Data Added")

MysqlConn.Close()

Catch ex As Exception

MessageBox.Show(ex.Message)

Finally

MysqlConn.Dispose()

End Try

End Sub

'Update data on button click

Private Sub Button4\_Click(sender As Object, e As EventArgs) Handles Button4.Click

MysqlConn = New MySqlConnection

MysqlConn.ConnectionString = "server=localhost; userid=root; password=password1; database=payroll;"

Dim READER As MySqlDataReader

Try

MysqlConn.Open()

Dim Query As String

Query = "update payroll.project set ProjectNumber='" & TextBox\_ProjectNumber.Text & "', ProjectName='" & TextBox\_ProjectName.Text & "', ProjectManager='" & TextBox\_ProjectManager.Text & "' where ProjectNumber='" & TextBox\_ProjectNumber.Text & "'"

COMMAND = New MySqlCommand(Query, MysqlConn)

READER = COMMAND.ExecuteReader

MessageBox.Show("Data Modified")

MysqlConn.Close()

Catch ex As Exception

MessageBox.Show(ex.Message)

Finally

MysqlConn.Dispose()

End Try

End Sub

'Delete data on button click

Private Sub Button5\_Click(sender As Object, e As EventArgs) Handles Button5.Click

MysqlConn = New MySqlConnection

MsqlConn.ConnectionString = "server=localhost; userid=root; password=password1; database=payroll;"

Dim READER As MySqlDataReader

Try

MysqlConn.Open()

Dim Query As String

Query = "Delete from payroll.project where ProjectNumber='" & TextBox\_ProjectNumber.Text & "'"

COMMAND = New MySqlCommand(Query, MysqlConn)

READER = COMMAND.ExecuteReader

MessageBox.Show("Data Deleted")

MysqlConn.Close()

Catch ex As Exception

MessageBox.Show(ex.Message)

Finally

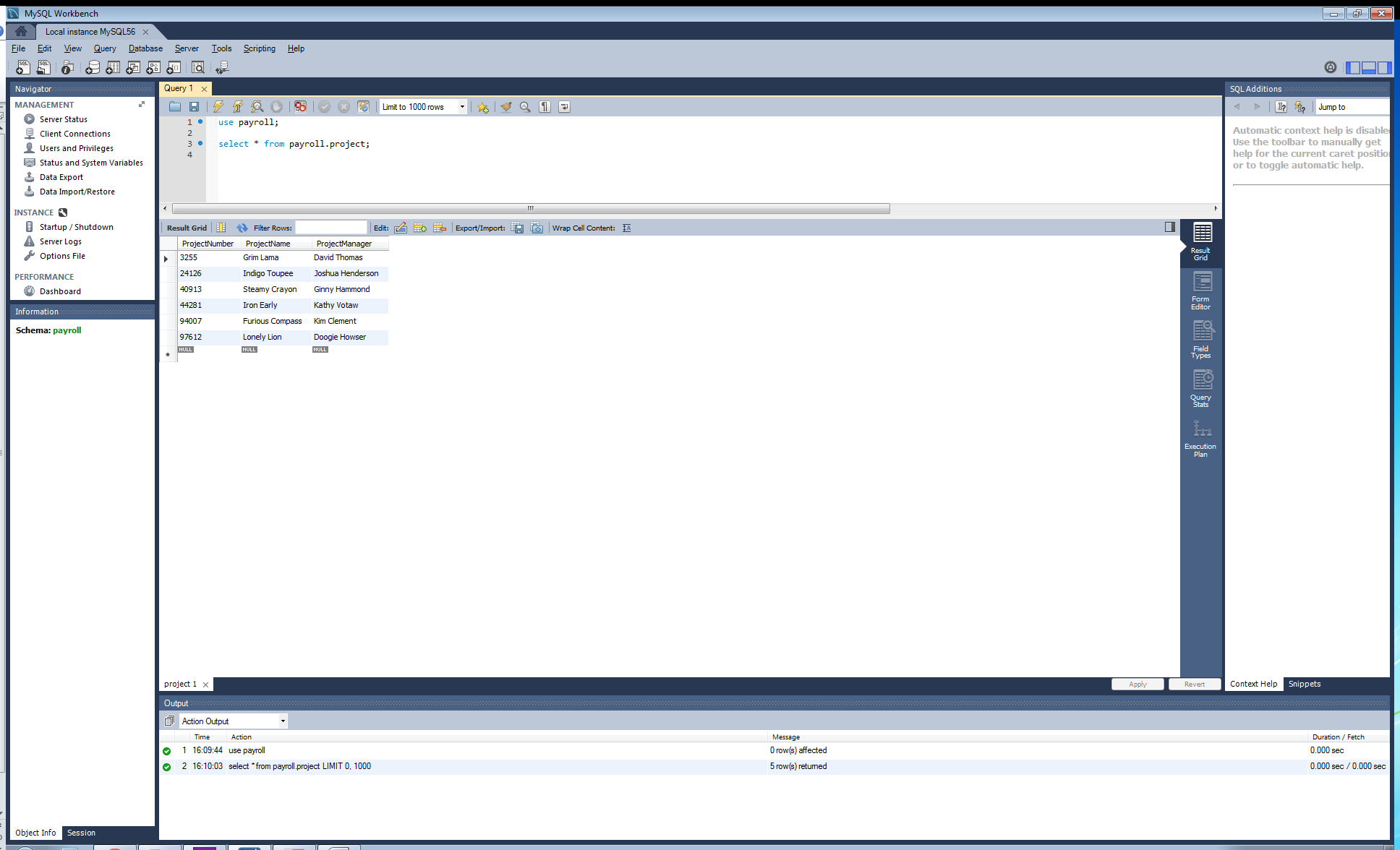
MysqlConn.Dispose()

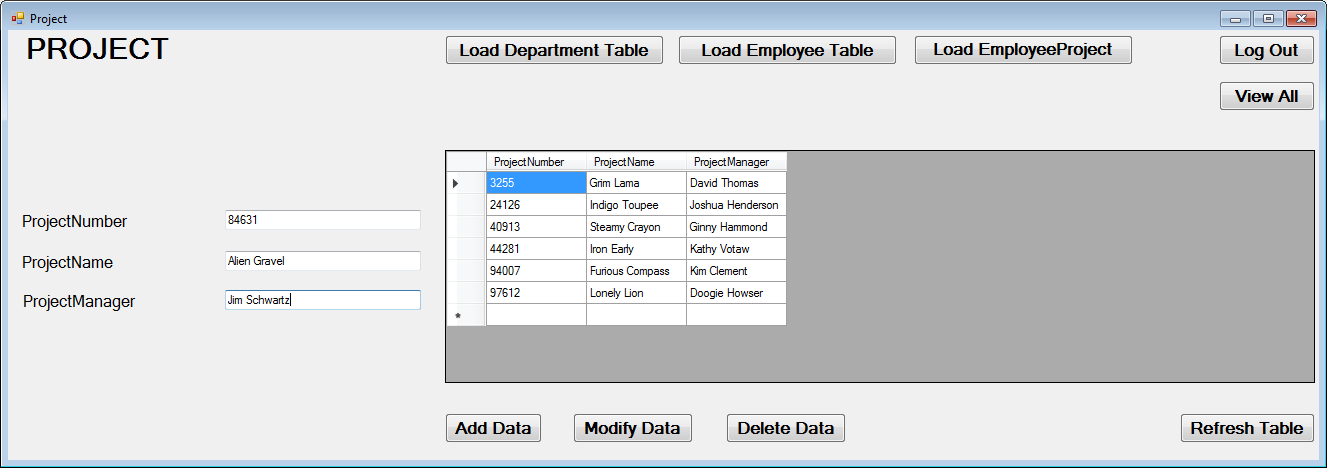
End Try

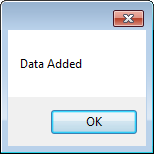
End Sub

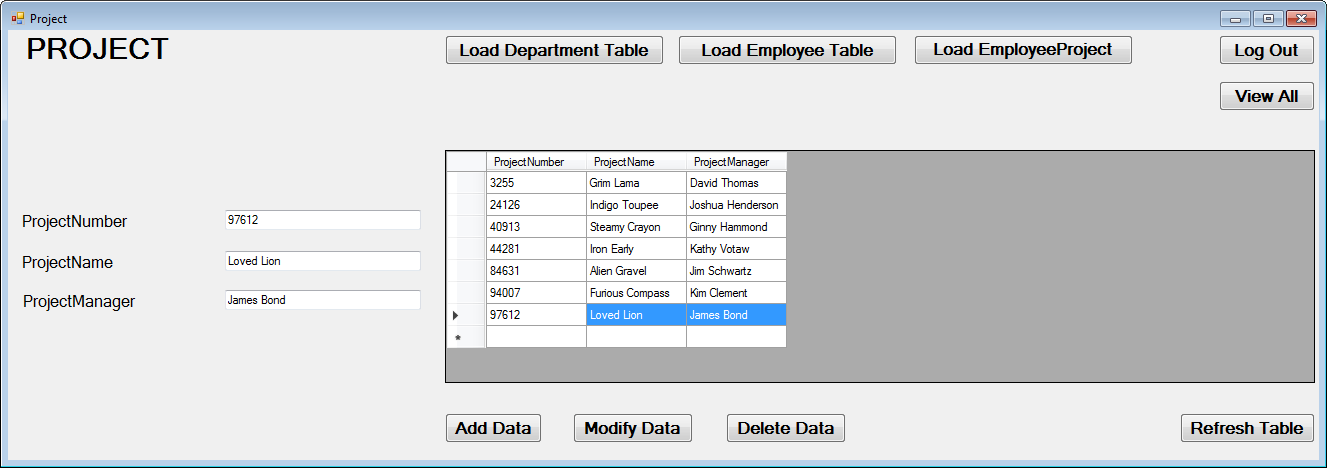
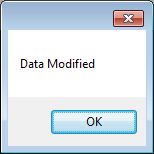
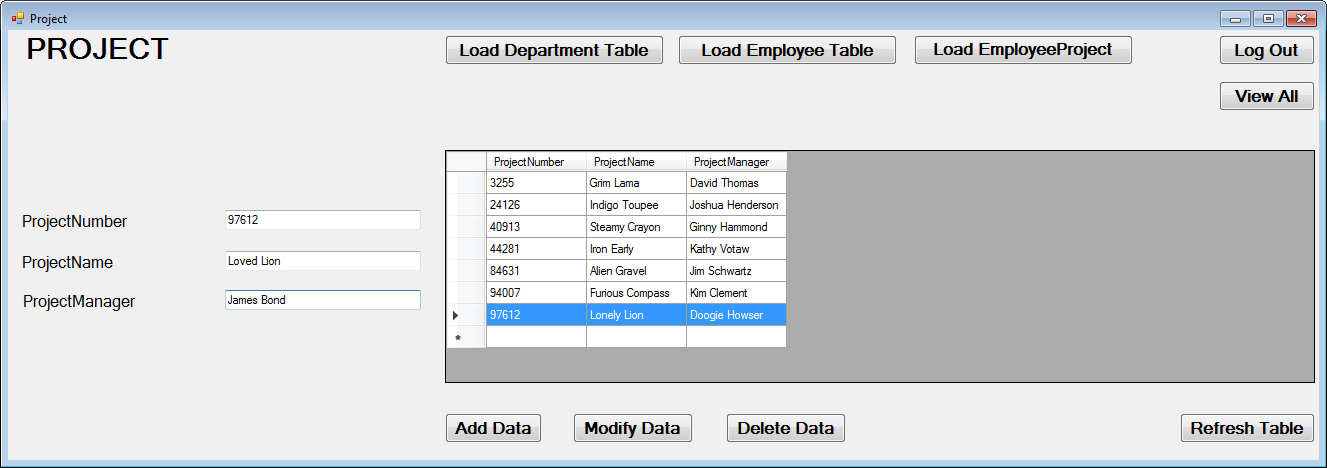
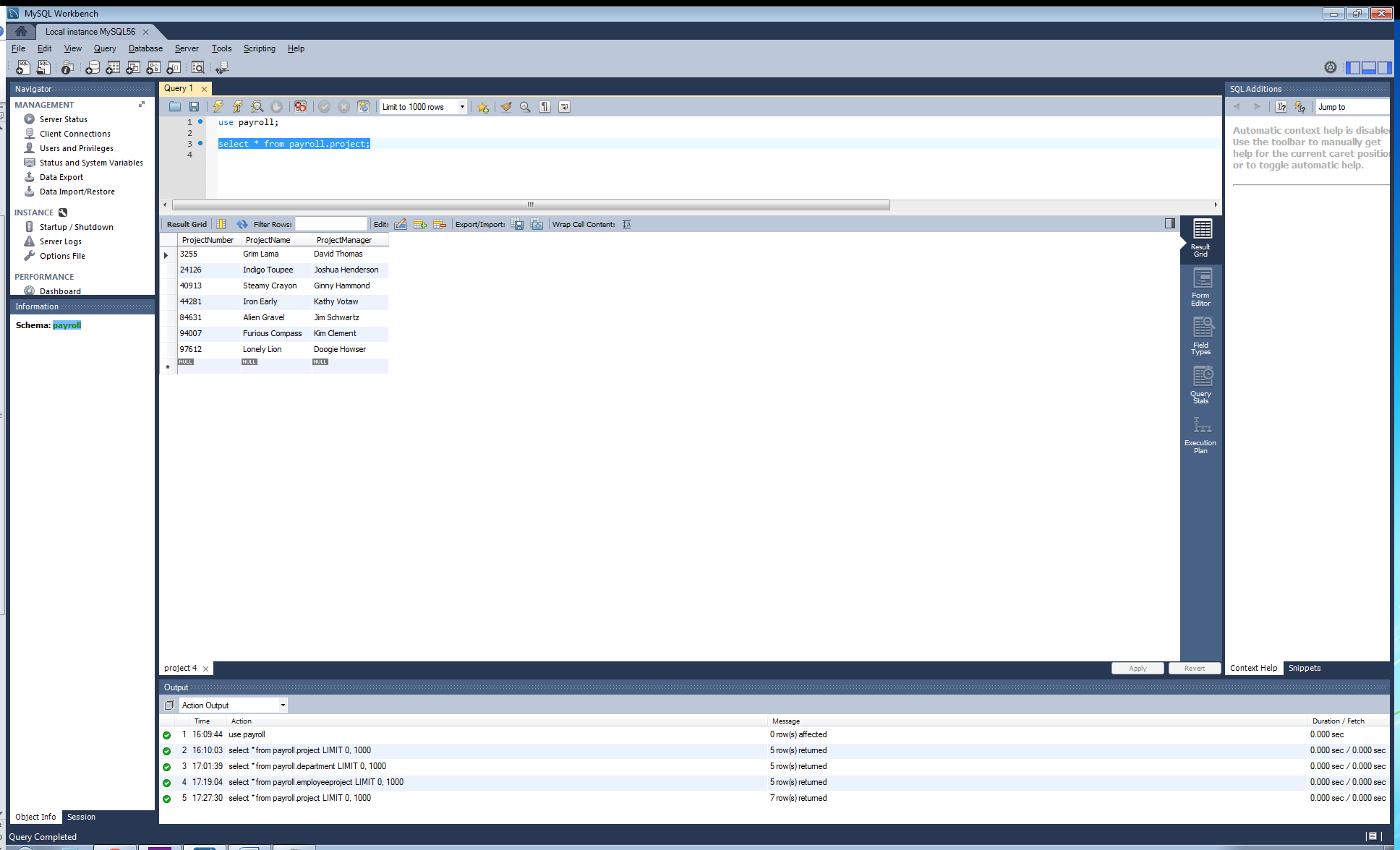
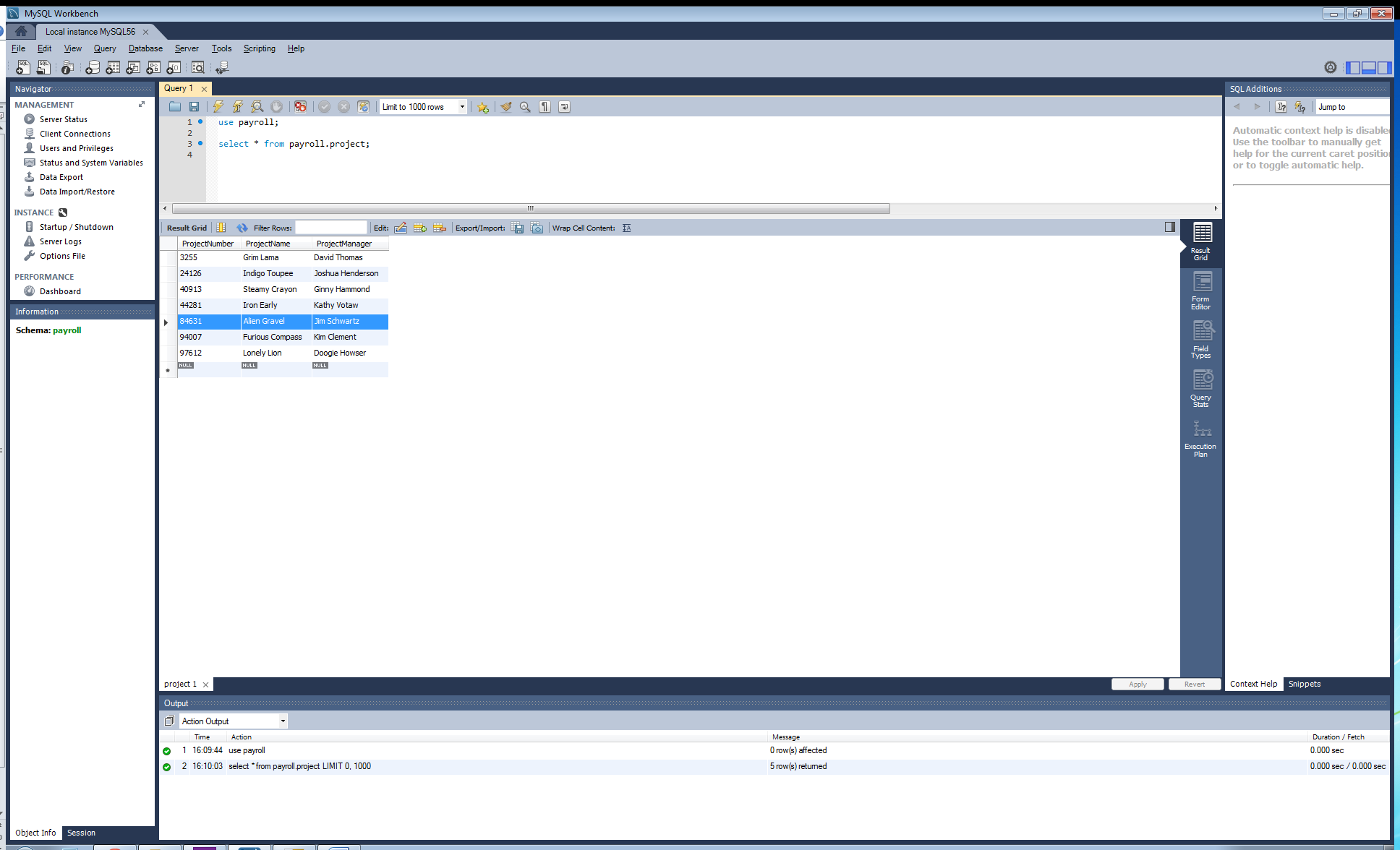
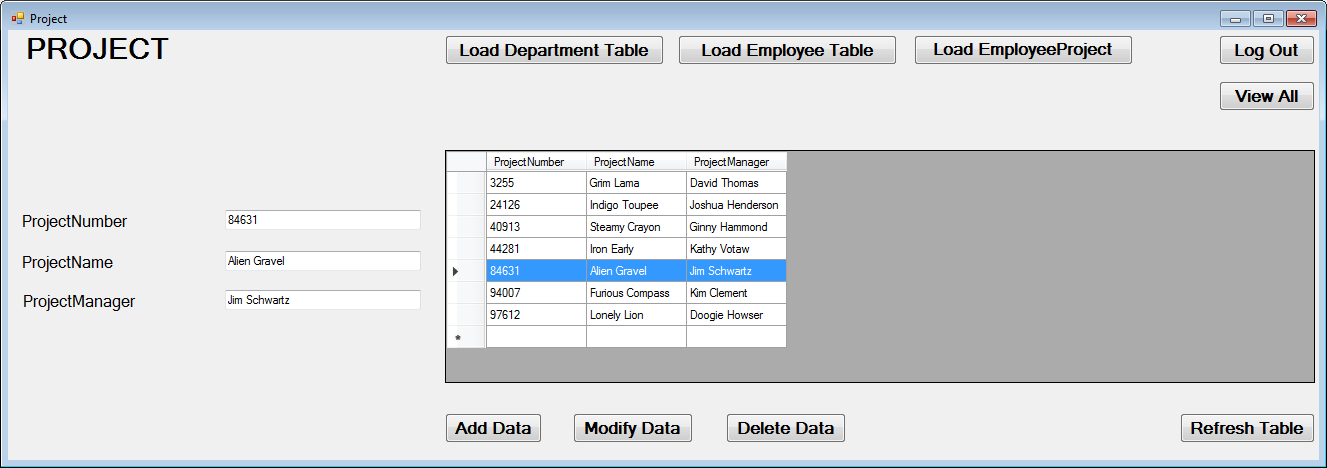
**Screenshots of the Results**

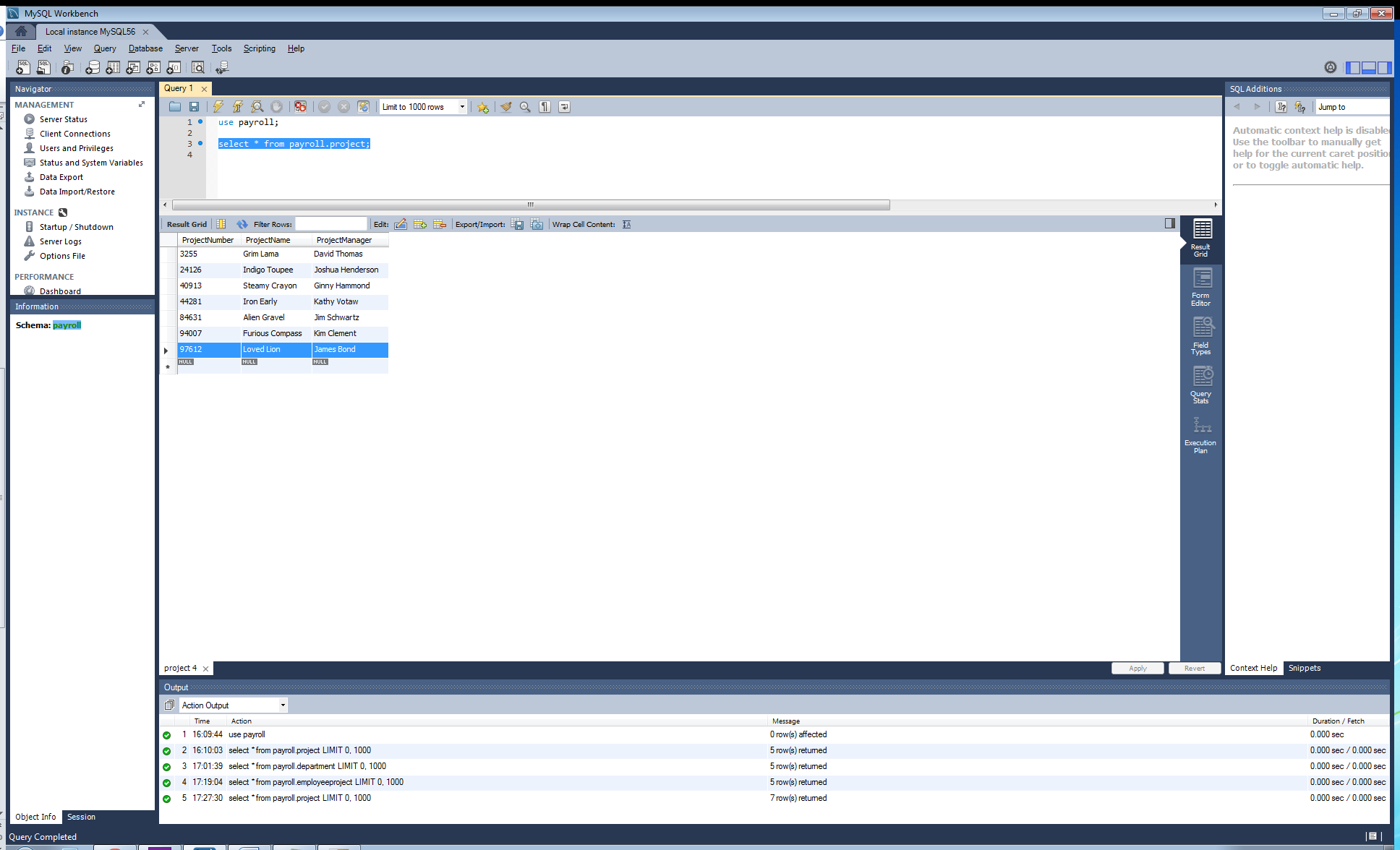
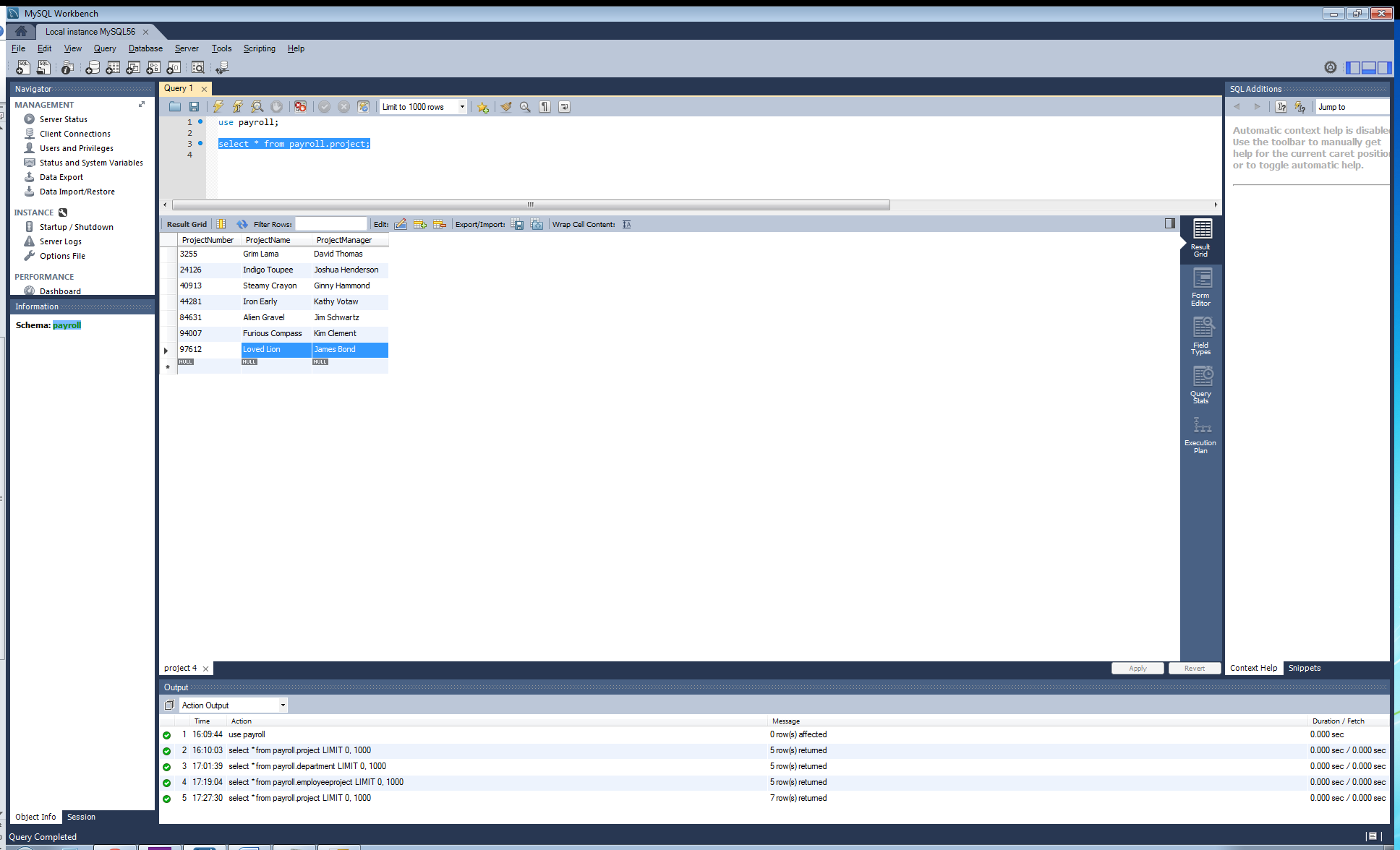
Please see the below screenshots to see the effects of the add/delete/update routines on the “Project” table, both within the application and also the ensuing changes they have in the database environment as well. Again, code is the same with replaced table names for each table within the application; for the sake of space I am illustrating one example.

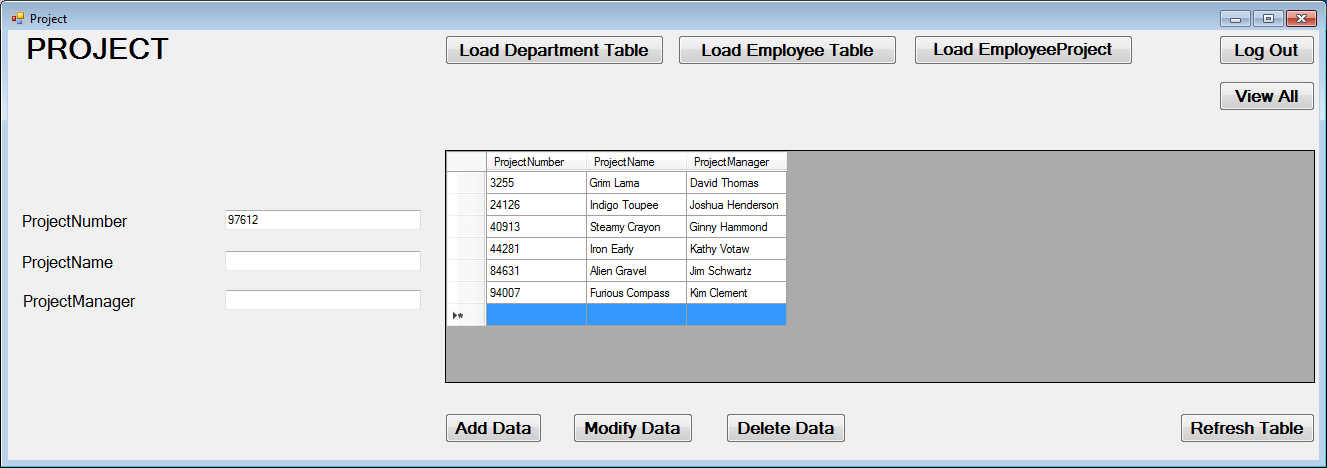
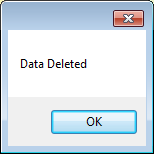
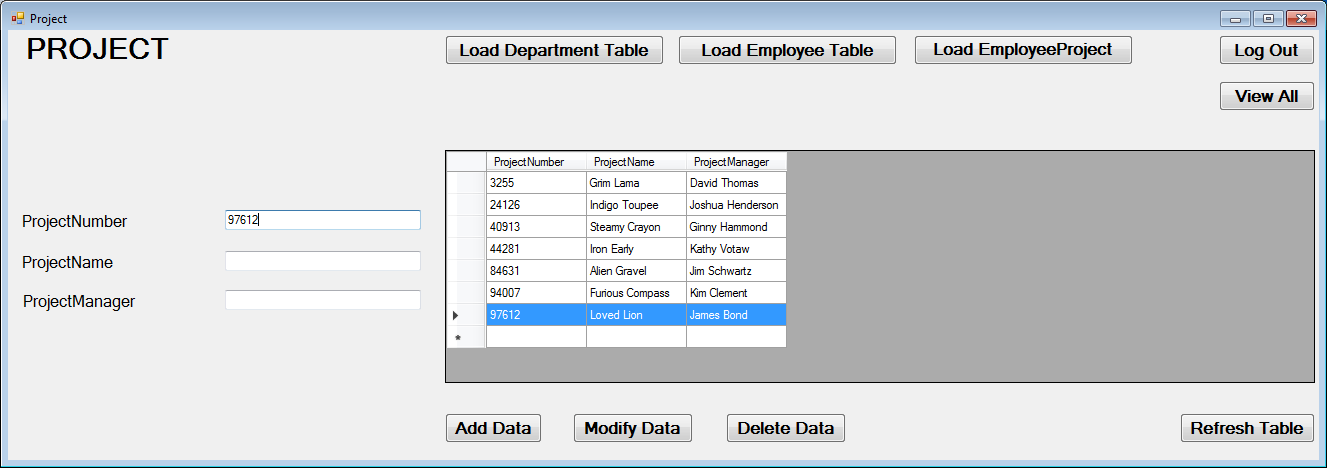


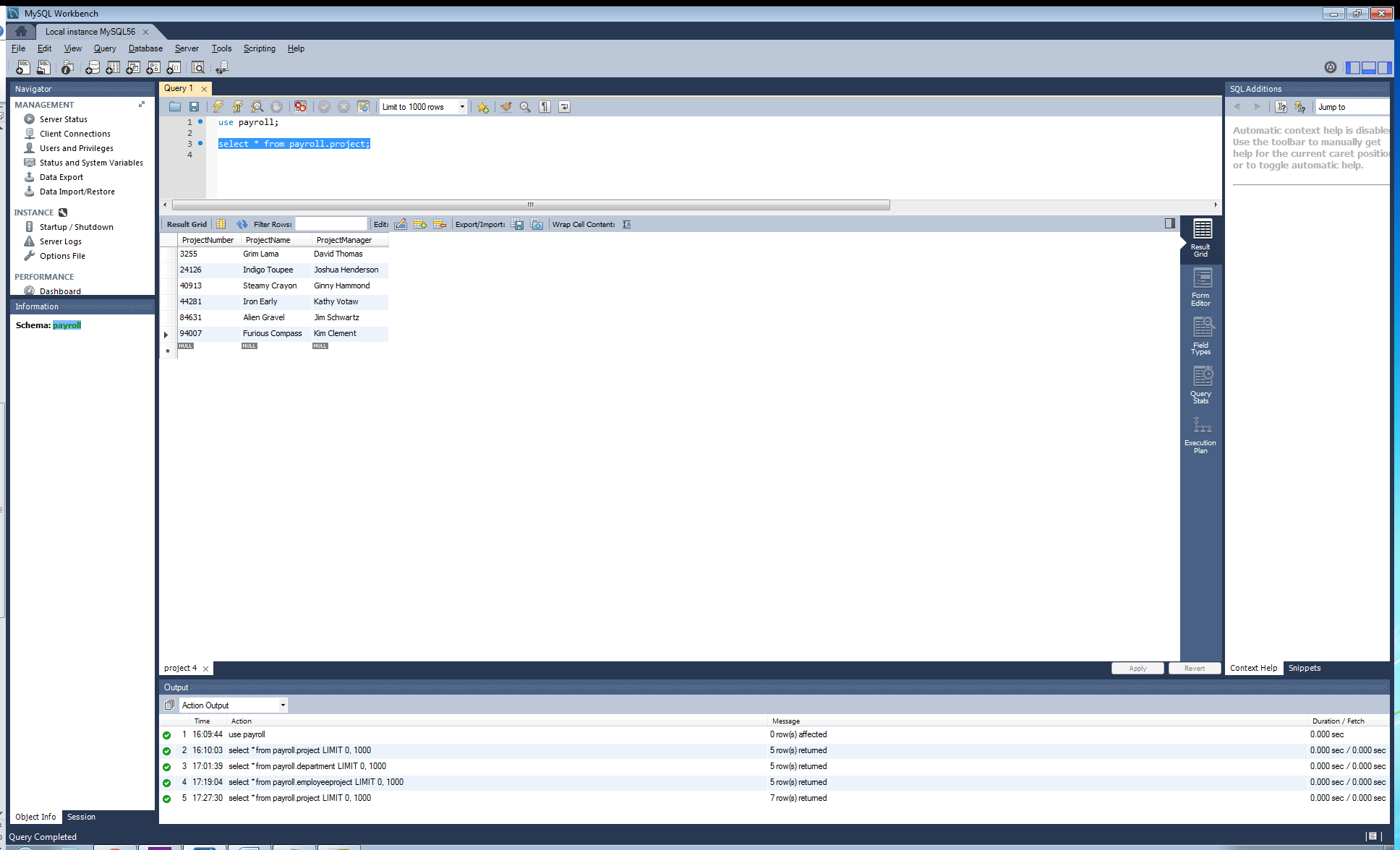












**Display/Print the Results**

I chose to use DataGridViews to serve as the container for my data to fill into to create a report to print/display the data from each table in the database. For these DataGridViews, I had one to coincide with each of the tables in my database, therefore you will find the DataGridView and the correct data for each table on each respective Forms page (i.e., “Department” Forms page loads department table data to DataGridView). In addition, I added two separate code blocks that load/display this data. First we have a structure that responds to a button press to load/refresh the data in the table/gridview. Second is a structure that automatically populates the DataGridViews upon loading each Forms page. Therefore we always have the data displayed to us on each respective Forms page, and then also have a button to refresh the data display if we have made changes using the add/change/delete routines. In addition, the “main menu” Forms page displays all data for all tables, all on the same page upon loading said page. Please see the code segment below for an example of one of my page’s DataGridView; first displaying the data on button press, and then just automatically loading/populating upon page load as well.

Code for one page is provided as the routines are the same for every other page except for switching the query to name the appropriate table.

'Fill DataGridView with MySQL table values when using "Refresh" button-click

Private Sub Load\_Table\_Btn\_Click(sender As Object, e As EventArgs) Handles Load\_Table\_Btn.Click

MysqlConn = New MySqlConnection

MysqlConn.ConnectionString = "server=localhost; userid=root; password=password1; database=payroll;"

'Connect DataGridView with database table

'Get data with adapter

Dim SDA As New MySqlDataAdapter

'Bind data from adapter to DataGridView

Dim dbDataSet As New DataTable

Dim bSource As New BindingSource

Try

'Open connection to database

MysqlConn.Open()

'Declare query

Dim Query As String

'Query that will pull from database and return to DataGridView

Query = "select \* from payroll.department"

COMMAND = New MySqlCommand(Query, MysqlConn)

SDA.SelectCommand = COMMAND

'Take values from query and fill to DataSet

SDA.Fill(dbDataSet)

'Bind DataSet to DataGridView

bSource.DataSource = dbDataSet

DataGridView1.DataSource = bSource

SDA.Update(dbDataSet)

MysqlConn.Close()

Catch ex As Exception

MessageBox.Show(ex.Message)

Finally

MysqlConn.Dispose()

End Try

End Sub

'Automatically load DataGridView upon page creation/load (same as above, just automatic and does not require refresh button press)

Private Sub load\_table()

MysqlConn = New MySqlConnection

MysqlConn.ConnectionString = "server=localhost; userid=root; password=password1; database=payroll;"

Dim SDA As New MySqlDataAdapter

Dim dbDataSet As New DataTable

Dim bSource As New BindingSource

Try

MysqlConn.Open()

Dim Query As String

Query = "select \* from payroll.department"

COMMAND = New MySqlCommand(Query, MysqlConn)

SDA.SelectCommand = COMMAND

SDA.Fill(dbDataSet)

bSource.DataSource = dbDataSet

DataGridView1.DataSource = bSource

SDA.Update(dbDataSet)

MysqlConn.Close()

Catch ex As Exception

MessageBox.Show(ex.Message)

Finally

MysqlConn.Dispose()

End Try

End Sub

Screenshots for each page successfully displaying the data will be provided below:

