



# **ALAGAPPA UNIVERSITY**

[Accredited with 'A+' Grade by NAAC (CGPA:3.64) in the Third Cycle  
and Graded as Category-I University by MHRD-UGC]

(A State University Established by the Government of Tamil Nadu)

**KARAIKUDI – 630 003**



## **Directorate of Distance Education**

### **M.Sc. [Information Technology]**

**II - Semester**

**313 24**

## **VB.NET AND RDBMS LAB**

<b>Reviewer</b>	
Dr. S. Santhosh Kumar	Assistant Professor Department of Computer Science, Science Campus, Alagappa University, Karaikudi

#### **Author**

**Dr. Kavita Saini**, Associate Professor, School of Computer Science & Engineering, Galgotias University, Greater Noida.

"The copyright shall be vested with Alagappa University"

All rights reserved. No part of this publication which is material protected by this copyright notice may be reproduced or transmitted or utilized or stored in any form or by any means now known or hereinafter invented, electronic, digital or mechanical, including photocopying, scanning, recording or by any information storage or retrieval system, without prior written permission from the Alagappa University, Karaikudi, Tamil Nadu.

Information contained in this book has been published by VIKAS® Publishing House Pvt. Ltd. and has been obtained by its Authors from sources believed to be reliable and are correct to the best of their knowledge. However, the Alagappa University, Publisher and its Authors shall in no event be liable for any errors, omissions or damages arising out of use of this information and specifically disclaim any implied warranties or merchantability or fitness for any particular use.



VIKAS® is the registered trademark of Vikas® Publishing House Pvt. Ltd.

VIKAS® PUBLISHING HOUSE PVT. LTD.

E-28, Sector-8, Noida - 201301 (UP)

Phone: 0120-4078900 • Fax: 0120-4078999

Regd. Office: 7361, Ravindra Mansion, Ram Nagar, New Delhi 110 055

• Website: [www.vikaspublishing.com](http://www.vikaspublishing.com) • Email: [helpline@vikaspublishing.com](mailto:helpline@vikaspublishing.com)

**Work Order No. AU/DDE/DE1-291/Preparation and Printing of Course Materials/2018 Dated 19.11.2018 Copies - 500**

---

# VB.NET AND RDBMS LAB

---

## Syllabi

---

### BLOCK 1: SIMPLE APPLICATIONS

1. **Simple Applications:** Developing simple applications using VB.NET
  - a. Finding factorial Value
  - b. Money Conversion
  - c. Quadratic Equatin
  - d. Temperature Conversion
  - e. Login control
2. **Login form:** Create and Validate Login Form, Program to design Class, Program to demonstrate Inheritance, Polymorphism and Interfaces.

### BLOCK 2: CONTROLS

3. **Controls:** 1. Advance Controls, Common Dialog Controls.
  2. Adrotator Control
  3. Calendar control
    - a. Display messages in a calendar control
    - b. Display vacation in a calendar control
    - c. Selected day in a calendar control using style
    - d. Difference between two calendar dates
  4. Treeview control
    - a. Treeview control and datalist
    - b. Treeview operations
  5. Validation controls
4. **Active X Controls:** Working with intrinsic controls and ActiveX controls

### BLOCK 3: MDI AND DATA CONTROLS

5. **MDI:** Application with multiple forms
6. **Data controls:** Application using data controls

### BLOCK 4: DIALOGS AND MENU

7. **Dialogs:** Application with dialogs
8. **Common Dialogs:** Application using Common Dialogs
9. **Menus:** Application with Menus

### BLOCK 5: EVENTS AND DATABASE

10. **Events and Database:** Drag and Drop Events Database Management Creating
11. **DataGridView:** ADO.NET Code to show records in DataGridView Control.
  1. Databinding using datalist control
  2. Datalist control templates
  3. Databinding using datagrid
  4. Datagrid control template
  5. Datagrid hyperlink
  6. Datagrid button column
  7. Datalist event
  8. Datagrid paging
12. **Database Operations:** ADO.NET Code to perform Insert, Delete, Update and Select operations. Crystal Reports Web Application using ASP.NET that uses validation controls. Table creation, Renaming a Table, Copying another table, Dropping a Table Table Description: Describing Table Definitions, Modifying Tables, Joining tables, Number and Date functions.

### BLOCK 6: SQL QUERIES AND SUB QUERIES

13. **SQL Queries:** Queries, Sub Queries, and aggregate functions DDL: Experiments using database DDL SQL statements DML: Experiment using database DML SQL statements DCL: Experiment using database DCL SQL statements
14. **Application Development:** Design and Develop Application: Library information system, Students mark sheet processing, Telephone directory maintenance, Gas booking and delivering, Electricity bill processing, Bank Transaction, Pay roll processing. Personal information system, Question database and conducting Quiz and Personal diary

---

## INTRODUCTION

---

### NOTES

Visual Basic .NET or VB.NET, the next generation of the Visual Basic language, is a fast and easy way to create .NET-based applications, including XML Web services and Web applications. Visual Basic .NET has many new and improved features that make it a powerful object oriented programming language, including inheritance, interfaces, and overloading. Other new language features include free threading and structured exception handling. Visual Basic .NET also fully integrates the .NET Framework and the Common Language Runtime, which provide language interoperability, garbage collection, enhanced security, and improved versioning support. Visual Basic .NET is one of the two flagship languages (with C#) for the .NET framework from Microsoft. VB.NET includes full-blown support for object-oriented concepts, including simple inheritance. Everything in VB.NET is an object, including all of the primitives (Short, Integer, Long, String, Boolean, etc.) as well as types, events, and even assemblies. Everything inherits from the Object base class.

The most widely used commercial and open source databases are based on the relational model. Characteristically, a RDBMS is a DBMS in which data is stored in tables and the relationships among the data are also stored in tables. This stored data can be accessed or reassembled in many different ways without having to change the table forms. RDBMS program lets you create, update and manage a relational database. In spite of repeated challenges by competing technologies, as well as the claim by some experts that no current RDBMS has fully implemented relational principles, the majority of new corporate databases are still being created and managed with an RDBMS. So, understanding RDBMS through lab manuals has become extremely important.

This Lab Manual is intended for the students of M.Sc.-IT in the subject of *VB.NET and RDBMS*. This manual typically contains practical/Lab Sessions, covering various aspects related to the subject to enhanced understanding. Students are advised to thoroughly go through this manual rather than only topics mentioned in the syllabus as practical aspects are the key to understanding and conceptual visualization of theoretical aspects covered in the textbooks.

## Software Requirements for Program Implementation

VB.NET & RDBMS Lab

Microsoft Visual Studio 2010/ 2015, Asp .net Framework 3.0/3.5/4.0/4.5  
Microsoft SQL Express 2005 database engine.

## Hardware Requirements

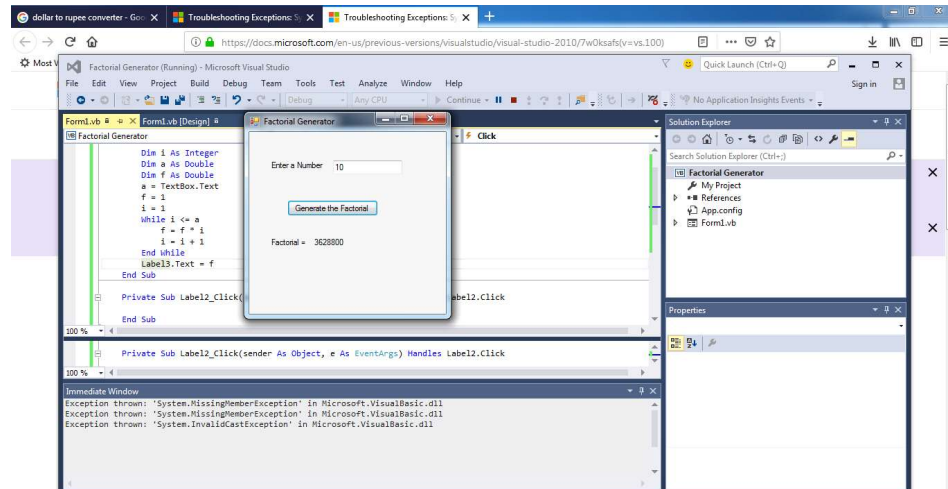
Any computer hardware capable of running DOS can be used.

## PROGRAMS:

### 1. Write a program to generate the factorial of a given number.

```
PublicClassForm1
PrivateSub Button1_Click(sender AsObject, e AsEventArgs)
Handles Button1.Click
Dim i AsInteger
Dim a AsDouble
Dim f AsDouble
    a = TextBox.Text
    f = 1
    i = 1
While i <= a
    f = f * i
    i = i + 1
EndWhile
    Label3.Text = f
EndSub
PrivateSub Label2_Click(sender AsObject, e AsEventArgs)
Handles Label2.Click
EndSub
PrivateSub Form1_Load(sender AsObject, e AsEventArgs)
Handles MyBase.Load
EndSub
EndClass
```

## NOTES

**Output:****NOTES****2. Write a program to perform money conversion.**

```

Public Class Form1
    Public Sub Form1_Load(sender As Object, e As EventArgs)
        Handles MyBase.Load
    EndSub

    Public Sub ComboBox1_SelectedIndexChanged(sender As Object,
        e As EventArgs) Handles ComboBox1.SelectedIndexChanged
    EndSub

    Private Sub Button1_Click(sender As Object, e As EventArgs)
        Handles Button1.Click
        Dim amount As Double
            amount = TextBox1.Text
        If ComboBox1.SelectedItem.Value = "IndianRupees" And
            ComboBox2.SelectedItem.Value = "Dollar" Then
            Label5.Text = amount / 72 & "Dollar"
        ElseIf ComboBox1.SelectedItem.Value = "IndianRupees" And
            ComboBox2.SelectedItem.Value = "Japanese
            Yen" Then
            Label5.Text = amount * 1.54 & "Japanese Yen"
        ElseIf ComboBox1.SelectedItem.Value = "IndianRupees" And
            ComboBox2.SelectedItem.Value = "Indian
            Rupees" Then
            Label5.Text = amount & "Indian Rupee"
        ElseIf ComboBox1.SelectedItem.Value = " Dollar" And
            ComboBox2.SelectedItem.Value = "IndianRupees" Then
            Label5.Text = amount * 72 & "Indian Rupee"
        ElseIf ComboBox1.SelectedItem.Value = " Dollar" And
            ComboBox2.SelectedItem.Value = "Japanese
            Yen" Then
            Label5.Text = amount * 109.78 & "Japanese Yen"
    EndSub
End Class

```

```

ElseIf ComboBox1.SelectedItem.Value = "Dollar"And
    ComboBox2.SelectedItem.Value = " Dollar"Then
    Label5.Text = amount &"Dollar"
ElseIf ComboBox1.SelectedItem.Value = "Japanese Yen"And
ComboBox2.SelectedItem.Value = "IndianRupees"Then
    Label5.Text = amount / 1.54 &"Indian Rupee"
ElseIf ComboBox1.SelectedItem.Value = "Japanese Yen"And
ComboBox2.SelectedItem.Value = " Dollar"Then
    Label5.Text = amount / 109.78 &"Dollar"
ElseIf ComboBox1.SelectedItem.Value = "Japanese Yen"And
ComboBox2.SelectedItem.Value = "Japanese Yen"Then
    Label5.Text = amount &"Japanese Yen"
Else : ComboBox1.SelectedItem.Value =
ComboBox2.SelectedItem.Value
    MsgBox("You Select Same Currency")

EndIf
EndSub

PrivateSub TextBox1_TextChanged(sender AsObject, e
AsEventArgs) Handles TextBox1.TextChanged
EndSub

PrivateSub Label4_Click(sender AsObject, e AsEventArgs)
Handles Label4.Click
EndSub

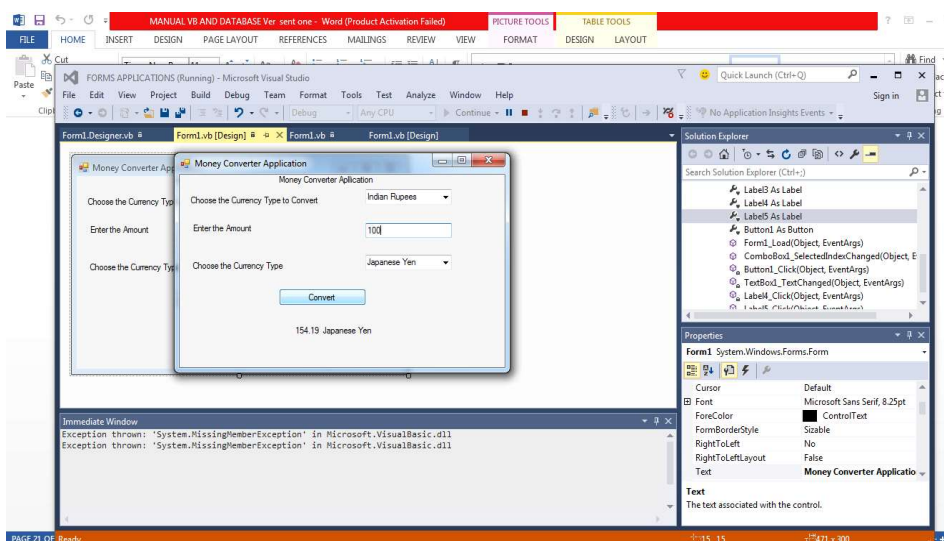
PrivateSub Label5_Click(sender AsObject, e AsEventArgs)
Handles Label5.Click
EndSub

EndClass

```

## NOTES

### Output:



**3. Write a program to find out roots of the quadratic equation.****NOTES**

```

PublicClassForm1
PrivateSub Button1_Click(sender AsObject, e AsEventArgs)
Handles Button1.Click
Dim a AsDouble
Dim b AsDouble
Dim c AsDouble
Dim s AsDouble
Dim x1 AsDouble
Dim x2 AsDouble
    a = TextBox1.Text
    b = TextBox2.Text
    c = TextBox3.Text
    s = Math.Sqrt(b * b - 4 * a * c)
If (b * b - 4 * a * c) = 0 Then
    Label4.Text = "Result Is:"
    Label5.Text = "Roots are Equal"
    x1 = (-b + s) / 2 * a
    x2 = (-b - s) / 2 * a
    Label4.Text = "Result Is:"
    Label5.Text = "The Roots : "& x1
ElseIf (b * b - 4 * a * c) < 0 Then
    Label4.Text = "Result Is:"
    Label5.Text = "The Roots are Imaginary"
Else Label4.Text = "Result Is:"
    Label5.Text = "The Roots are Not Equal"
    x1 = (-b + s) / 2 * a
    x2 = (-b - s) / 2 * a
    Label4.Text = "Result Is:"
    Label5.Text = "The Roots "& x1 &"or"& x2
EndIf
EndSub

PrivateSub Form1_Load(sender AsObject, e AsEventArgs)
Handles MyBase.Load

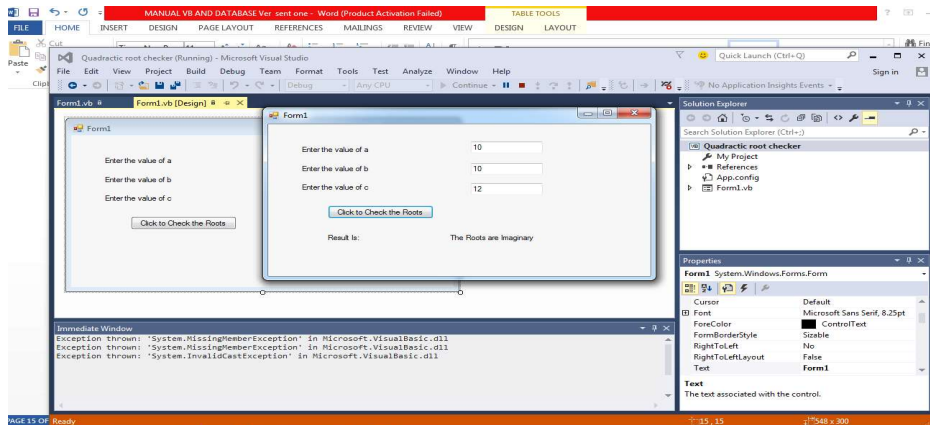
EndSub

EndClass

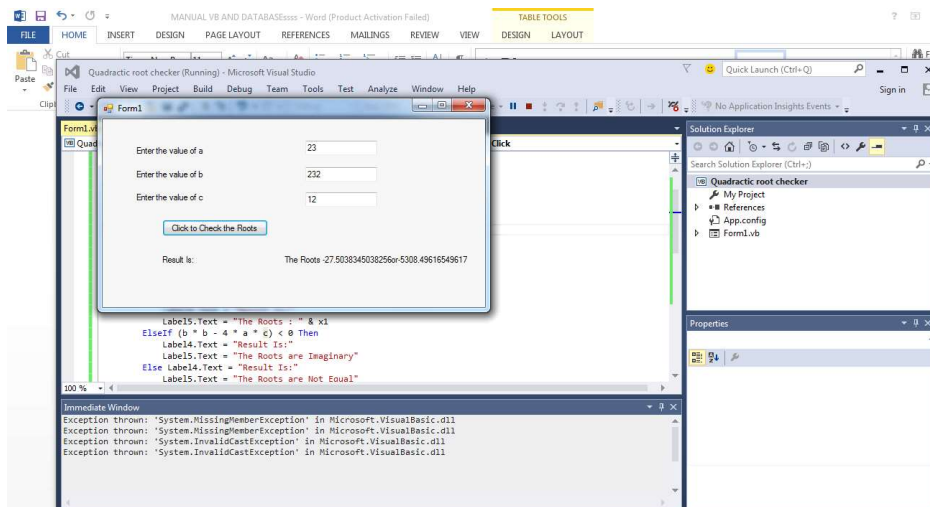
```



## Output:



## NOTES



### 4. Write a program to convert temperature from Fahrenheit to Celsius or vice versa.

```
Public Class Form1
    Private Sub RadioButton1_CheckedChanged(sender As Object, e As EventArgs) Handles RadioButton1.CheckedChanged
    EndSub

    Private Sub RadioButton2_CheckedChanged(sender As Object, e As EventArgs) Handles RadioButton2.CheckedChanged
    EndSub

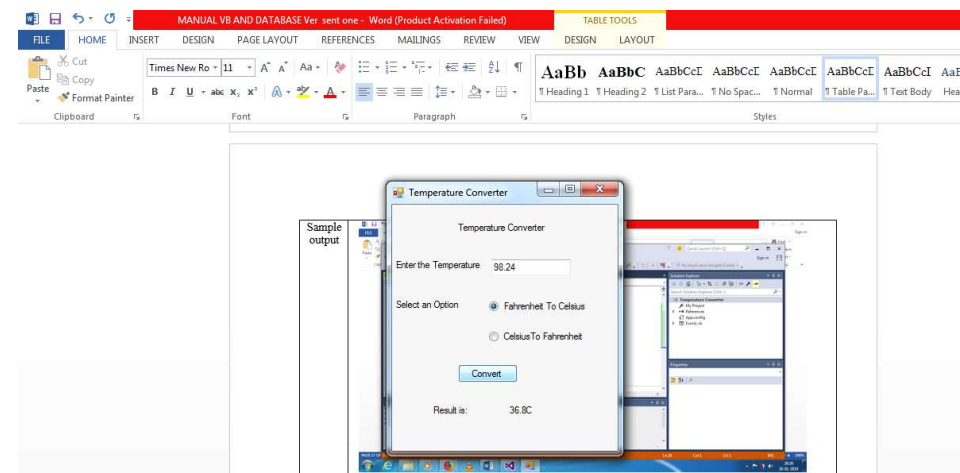
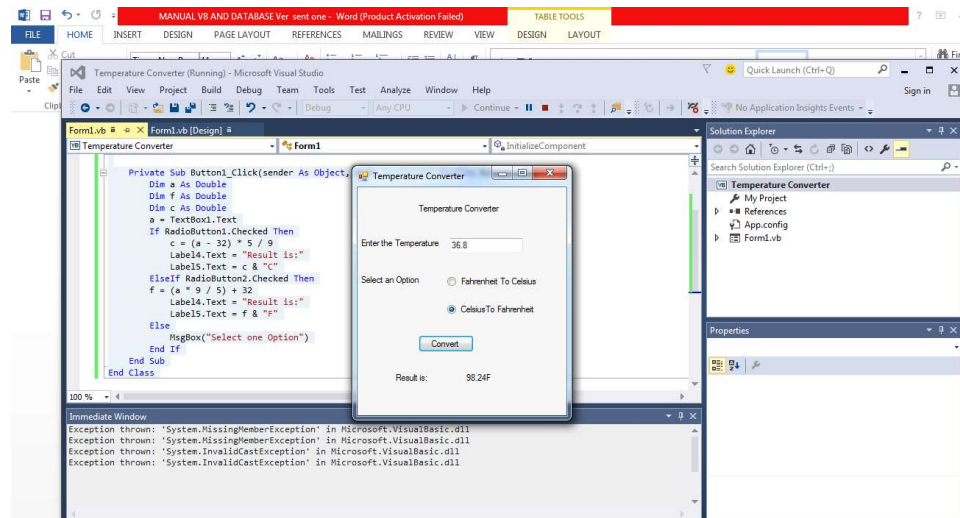
    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        Dim a As Double
        Dim f As Double
        Dim c As Double
        a = TextBox1.Text
        If RadioButton1.Checked Then
            c = (a - 32) * 5 / 9
```

**NOTES**

```

Label4.Text = "Result is:"
Label5.Text = c & "C"
ElseIf RadioButton2.Checked Then
    f = (a * 9 / 5) + 32
    Label4.Text = "Result is:"
    Label5.Text = f & "F"
Else
    MsgBox("Select one Option")
EndIf
EndSub
EndClass
    
```

**Output:**



## 5. Write a program to generate/develop the login control.

VB.NET & RDBMS Lab

```
Partial Class _Default
```

```
Inherits System.Web.UI.Page
```

```
Protected Sub Login1_Authenticate(ByVal sender As Object,  
ByVal e As System.Web.UI.WebControls.AuthenticateEventArgs)  
Handles Login1.Authenticate
```

```
If Login1.UserName = "Database" And Login1.Password =  
"Jaiswal" Then MsgBox("You are successfully Logged in")
```

```
Else MsgBox("Error:Loggedin") End If
```

```
If Application("i") = 3 Then MsgBox("You are Blocked")  
Login1.Enabled = False
```

```
End If EndSub
```

```
Protected Sub Page_Load(ByVal sender As Object, ByVal e As  
System.EventArgs) Handles Me.Load
```

```
Application("i") = Int(Application("i") + 1) If  
Application("i") > 3 Then Application("i") = 0
```

```
End If End Sub End Class
```

```
Global Application:
```

```
<%@ Application Language="VB" %><script runat="server">
```

```
Sub Application_Start(ByVal sender As Object, ByVal e As  
EventArgs) Application("i") = 0End Sub
```

```
Sub Application_End(ByVal sender As Object, ByVal e As  
EventArgs) EndSub
```

```
Sub Application_Error(ByVal sender As Object, ByVal e As  
EventArgs) EndSub
```

```
Sub Session_Start(ByVal sender As Object, ByVal e As  
EventArgs) EndSub
```

```
Sub Session_End(ByVal sender As Object, ByVal e As  
EventArgs) EndSub
```

```
</script>
```

## NOTES

**Output:****After enter the wrong password****NOTES**
**After log in three times the login will be blocked**
**6. Write a program to create an Advertisement using Ad rotator.**

```
<?xml version="1.0" encoding="utf-8" ?>
```

```
<Advertisements>
```

```
<Ad>
```

```
<ImageUrl>.\image\p1.jpg</ImageUrl>
```

```
<NavigateUrl>www.pepsi.com</NavigateUrl>
```

```
<AlternativeText>Pepsi</AlternativeText>
```

```
<Keyword>softdrink</Keyword><Impression>2</Impression>
```

```
</Ad><Ad>
```

```
<ImageUrl>.\image\p2.jpg</
```

```
ImageUrl><NavigateUrl>www.Excel.com</NavigateUrl>
```

```
<AlternativeText>Excel</
```

```
AlternativeText><Keyword>softdrink</Keyword>
```

```
<Impression>2</Impression>
```

```
</Ad><Ad>
```

```
<ImageUrl>.\image\p3.jpg</
```

```
ImageUrl><NavigateUrl>www.7up.com</NavigateUrl>
```

```
<AlternativeText>7up</
```

```
AlternativeText><Keyword>softdrink</Keyword>
```

```
<Impression>2</Impression>
```

```
</Ad><Ad>
```

VB.NET & RDBMS Lab

```
<ImageUrl>.\image\p4.jpg</  
ImageUrl><NavigateUrl>www.Mirinda.com</NavigateUrl>  
<AlternativeText>Mirinda</  
AlternativeText><Keyword>softdrink</Keyword>  
<Impression>2</Impression>
```

```
</Ad>
```

```
</Advertisements>
```

## NOTES

### Output:



VIEW1



VIEW2

### 7. Write a program to display the holiday in calendar.

```
Partial Class _Default Inherits System.Web.UI.Page Dim  
Holidays(13,32)  
Protected Sub Page_Load(ByVal sender As Object, ByVal e  
As System.EventArgs) Handles Me.Load  
Holidays(10, 12) = "Birthdays" Holidays(10, 15) =  
"Aniversary"  
  
End Sub
```

```
Protected Sub Calendar1_DayRender(ByVal sender As  
Object, ByVal e As  
System.Web.UI.WebControls.DayRenderEventArgs) Handles  
Calendar1.DayRender If e.Day.IsOtherMonth Then  
e.Cell.Controls.Clear()  
Else  
Dim adate As Date = e.Day.Date  
Dim aHolidays As String = Holidays(adate.Month,  
adate.Day) If (Not aHolidays Is Nothing) Then
```

```
Dim aLB As New LB() aLB.Text = "<br>"
&aHolidayse.Cell.Controls.Add(aLB)
```

**NOTES**

```
End If
End If
End Sub
End Class
```

**Output:**

November 2009						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
25	26	27	28	29	30	31
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	1	2	3	4	5

**8. Write a program to display the vacation in calendar.**

```
Partial Class _Default Inherits System.Web.UI.Page
```

```
Protected Sub Calendar1_DayRender(ByVal sender As
Object, ByVal e As
System.Web.UI.WebControls.DayRenderEventArgs) Handles
Calendar1.DayRender
```

```
Dim vocationstyle As New Style() With vocationstyle
.BackgroundColor =System.Drawing.Color.Yellow
.BorderColor =System.Drawing.Color.Black
.BorderWidth = New Unit(3) End With
Dim weekendstyle As New Style()
weekendstyle.BackgroundColor =
System.Drawing.Color.SpringGreen
If ((e.Day.Date>= New Date(2009, 11, 23)) And
(e.Day.Date<= New Date(2009, 11, 30)))
Then
```

```
e.Cell.ApplyStyle(vacationstyle) ElseIf
(e.Day.IsWeekend) Then e.Cell.ApplyStyle(weekendstyle)
End If
End Sub
```

**Output:**

November 2009						
Oct						Dec
Go to the previous month						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
25	26	27	28	29	30	31
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30	1	2	3	4	5

**9. Write a program to display the selected date in the calendar.**

```
Partial Class _Default Inherits System.Web.UI.Page Dim
Holidays(13,32)
```

```
Protected Sub Calendar3_DayRender(ByVal sender As
Object, ByVal e As
System.Web.UI.WebControls.DayRenderEventArgs) Handles
Calendar3.DayRender
```

```
Dim vacationstyle As New Style() With vacationstyle
.BackColor = System.Drawing.Color.Yellow
.BorderColor = System.Drawing.Color.Purple
.BorderWidth = New Unit(3) End With
Dim weekendstyle As New Style()
```

```
weekendstyle.BackColor = System.Drawing.Color.Green
If ((e.Day.Date >= Calendar1.SelectedDate) And
(e.Day.Date <= Calendar2.SelectedDate))
Then
e.Cell.ApplyStyle(vacationstyle) ElseIf
(e.Day.IsWeekend) Then e.Cell.ApplyStyle(weekendstyle)
End If
If e.Day.IsOtherMonth Then e.Cell.Controls.Clear() Else
Dim aDate As Date = e.Day.Date
```

**NOTES**

**NOTES**

```
Dim aHolidays As String = Holidays(aDate.Month,
aDate.Day) If (Not aHolidays Is Nothing) Then
Dim aLB As LB = New LB() aLB.Text = "<br>"
&aHolidayse.Cell.Controls.Add(aLB)
End If End If EndSub
```

```
Protected Sub Page_Load(ByVal sender As Object, ByVal e
As System.EventArgs) Handles Me.Load
Holidays(8, 15) = "IndependenceDay" Holidays(1, 26) =
"RepublicDay" Holidays(10, 12) = "Birthday"
End Sub
```

```
Protected Sub Calendar1_SelectionChanged(ByVal sender
As Object, ByVal e As System.EventArgs) Handles
Calendar1.SelectionChanged
LB1.Text = Calendar1.SelectedDate End Sub
```

```
Protected Sub Calendar2_SelectionChanged(ByVal sender
As Object, ByVal e As System.EventArgs) Handles
Calendar2.SelectionChanged
LB2.Text = Calendar2.SelectedDate End Sub
End Class
```

**Output:**

< April 2010 >							< April 2010 >						
Sun	Mon	Tue	Wed	Thu	Fri	Sat	Sun	Mon	Tue	Wed	Thu	Fri	Sat
28	29	30	31	1	2	3	28	29	30	31	1	2	3
4	5	6	7	8	9	10	4	5	6	7	8	9	10
11	12	13	14	15	16	17	11	12	13	14	15	16	17
18	19	20	21	22	23	24	18	19	20	21	22	23	24
25	26	27	28	29	30	1	25	26	27	28	29	30	1
2	3	4	5	6	7	8	2	3	4	5	6	7	8

FROM TO 4/6/2010 FROM TO 4/12/2010

< April 2010 >						
Sun	Mon	Tue	Wed	Thu	Fri	Sat
				1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	



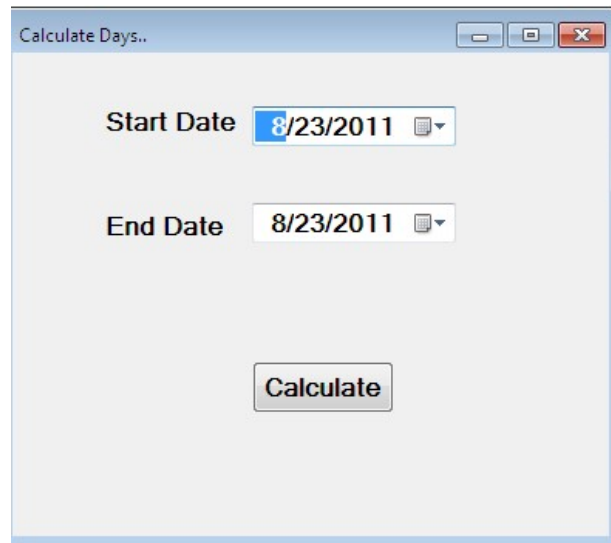
**10. Implement a program that display the difference between two dates.**

User can select two dates by DateTimePicker and also calculate number of days. For this, take a window form, two labels, two DateTimePicker and a Button. Then, set the properties of these controls as:

**Label:** Set its name as Start Date and End Date by its Text Property.

**DateTimePicker:** Set its Format Property as Short.

**Button:** Set its name as Calculate by its Text Property. Then, Form will look as shown below:



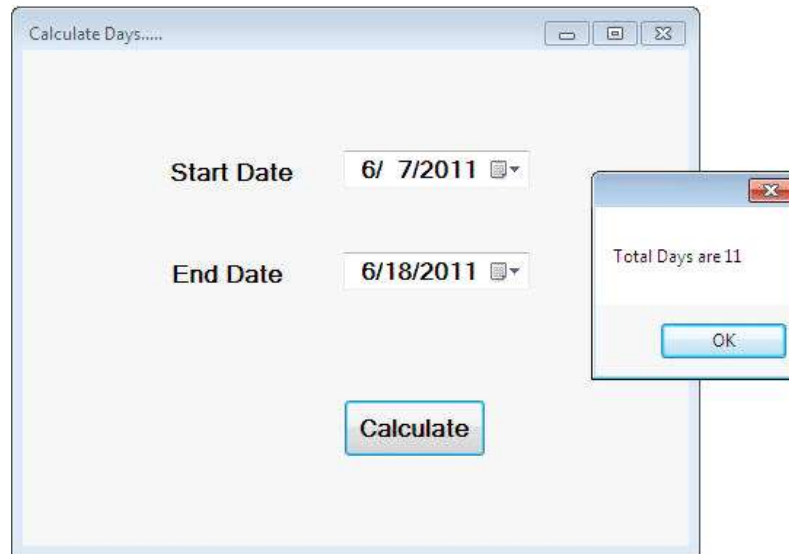
After that, write the following code on double click of Calculate button.

```
Private Sub btnCalculate_Click(sender As System.Object,
    e As System.EventArgs) Handles btnCalculate.Click
    If True Then
        Dim dt1 As DateTime = Convert.ToDateTime(DateTimePicker1.Text)
        Dim dt2 As DateTime = Convert.ToDateTime(DateTimePicker2.Text)
        Dim ts As TimeSpan = dt2.Subtract(dt1)
        If Convert.ToInt32(ts.Days) >= 0 Then
            MessageBox.Show("Total Days are
" & Convert.ToInt32(ts.Days))
        Else
            MessageBox.Show("Invalid Input")
        End If
    End If
End Sub
```

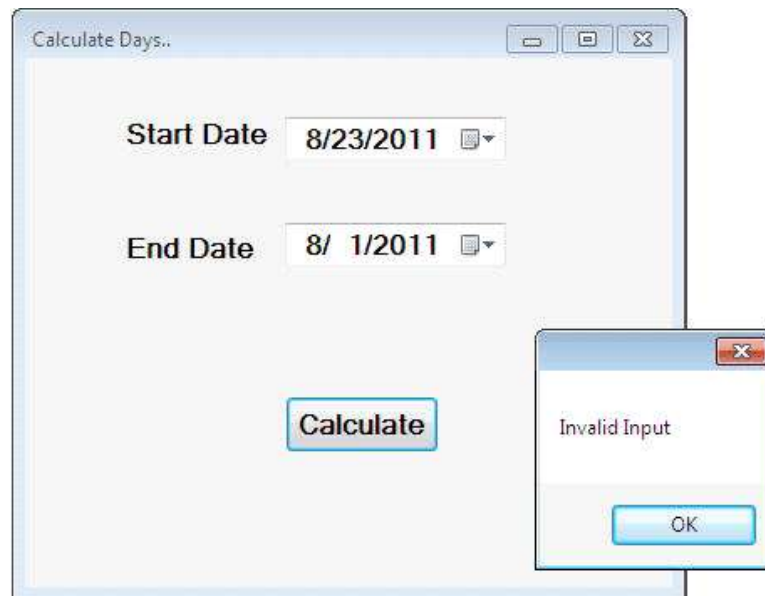
**NOTES**

Then, Select two dates and click Calculate button, The output will look like as shown below:

**NOTES**



If user enter wrong data, means if End Date comes before Start Date, then error message will show as “Invalid Input”.



**11. Write a program to perform Treeview operation using data list.**

```
Partial Class _Default Inherits System.Web.UI.Page
Protected Sub Button1_Click(ByVal sender As Object,
ByVal e As System.EventArgs) Handles Button1.Click
```

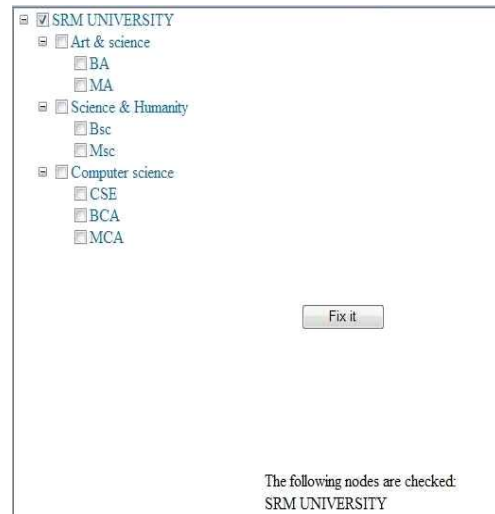
```

Dim checkednodes As TreeNodeCollection =
TreeView1.CheckedNodes
DataList1.DataSource =
checkednodes
DataList1.DataBind()

```

```
End Sub
```

### Output:



### 12. Write a program to perform Treeview operation.

```

Partial Class _Default Inherits System.Web.UI.Page
Protected Sub Page_Load(ByVal sender As Object, ByVal e
As System.EventArgs) Handles Me.Load
EndSub

Protected Sub TreeView1_SelectedNodeChanged(ByVal
sender As Object, ByVal e As System.EventArgs) Handles
TreeView1.SelectedNodeChanged
Dim value As String = TreeView1.SelectedNode.Value
Dim path As String = TreeView1.SelectedNode.ValuePath
Response.Write("The value select was" & value & "<br>")
Response.Write("The value path is " & value & "<br>")
End Sub

Protected Sub TreeView1_TreeNodeCheckChanged(ByVal
sender As Object, ByVal e As
System.Web.UI.WebControls.TreeNodeEventArgs) Handles
TreeView1.TreeNodeCheckChanged
Dim value As String = e.Node.Value
Response.Write("The value collapsed was" & value &
"<br>") End Sub

```

### NOTES

**NOTES**

```
Protected Sub TreeView1_TreeNodeExpanded(ByVal sender
As Object, ByVal e As
System.Web.UI.WebControls.TreeNodeEventArgs) Handles
TreeView1.TreeNodeExpanded Dim value As String =
e.Node.Value
Response.Write("The value collapsed was" & value &
"<br>") End Sub
End Class
```

**Output:**



**13. Write a program to perform validation operation.**

```
Protected Sub Button1_Click(ByVal sender As Object, ByVal
e As System.EventArgs)
Handles Button1.Click
If IsValid Then LB9.Enabled = False
TBox7.Enabled = False
End If
```

**Output:**

Registration form

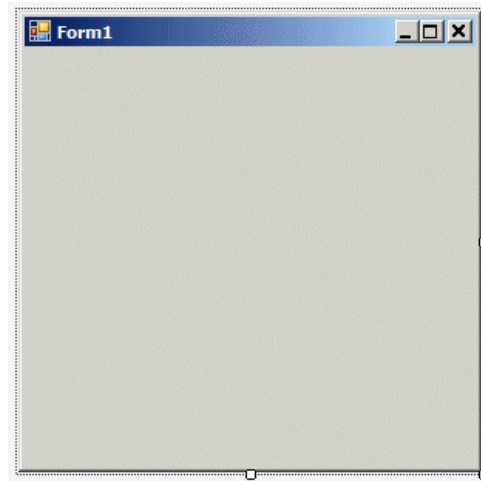
Name	<input type="text" value="Rajeev Ranjan"/>
Reg_no	<input type="text" value="35208095"/>
Date_Of_Birth	<input type="text" value="05-08-1986"/>
Department	<input type="text" value="MCA"/>
Address	<input type="text" value="CHENNAI"/>
Phone number	
personal phone no	<input type="text" value="9962941836"/>
Home phone no	<input type="text" value="9962941836"/>
Email_id	<input type="text" value="Rajeevj38@gmail.com"/>

**Note:** ActiveX Controls exist as separate files. Earlier versions of Visual Basic called VBX's (Visual Basic Extensions) and later OCX's .ocx extensions. One may include these controls by adding them to Toolbox. Microsoft includes many ActiveX controls that may enhance the performance of application.

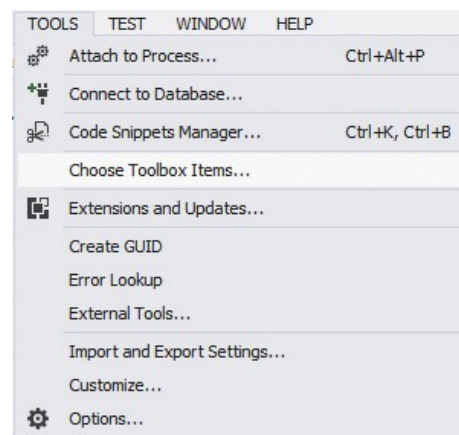
#### 14. Write a program to demonstrate intrinsic controls and ActiveX controls.

To use the control in a Visual Studio project, it must first be added to the Tool Box. Select "Choose Toolbox Items..." from the Tools drop down menu or by right clicking on the Tool Box itself.

##### Create Form in design view:



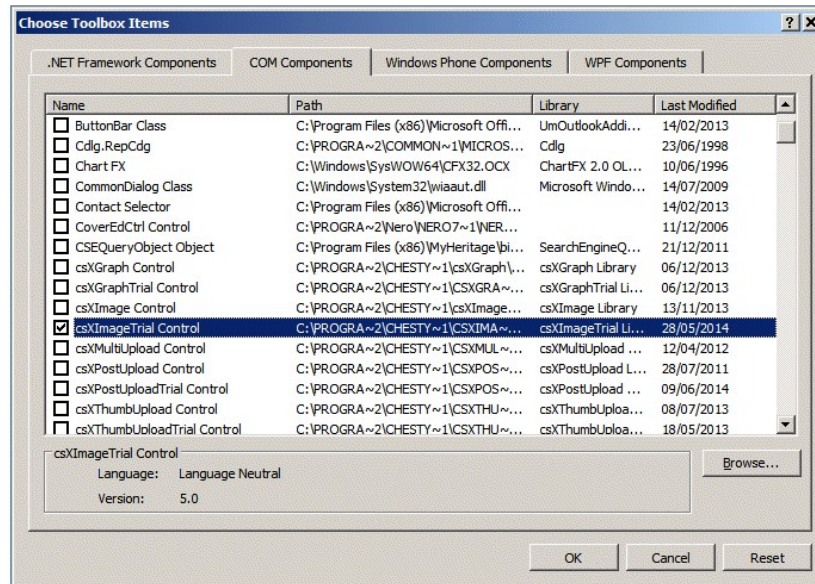
Choose the "Choose Toolbox Items" as shown below:



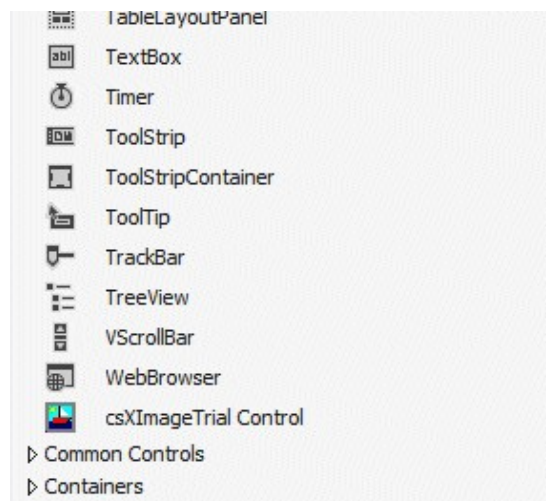
## NOTES

The Choose Toolbox Items dialogue box has four tabs. Select COM Components tab as shown in screenshot given below. After that click on box for csXImage. The trial version is shown below:

## NOTES



After clicking on OK button, the control will appear in the Tool Box and it can be added to a form like any other control.



As a quick test, you can import a csXImage control and a button and run the following code in the button click event handler:

```
Private Sub Button1_Click(sender As Object, e As
EventArgs) Handles Button1.Click
    AxImageBox1.LoadDialog()
End Sub
```

The default name for a `csXImage` object when it is placed on a form is `AxImageBox1`. This can be changed in the properties box at the right hand side of the IDE. This code calls the `LoadDialog` command which allows the user to select and load an image from disk.

## MDI CONTROLS

In Multiple Document Interface (MDI) application, we can view and work with several documents at the same time, similar to Microsoft Excel or word. MDI applications have a feature of MDI child forms and its very essential element of it.

### 15. Write a program to demonstrate an application with multiple forms.

Parent and Child Forms in MDI

MDI applications consist of a main form, which does not display any data itself, and one or more child forms which appear only within the main form are used for displaying documents. The main form is called the *MDI parent*, and the child forms are called the *MDI children*. The Form class has two properties that control whether a given form is an MDI parent, MDI child, or neither.

Note: The `MdiParent` property (which is of type Form) controls whether a form behaves as an MDI child.

*MDI application an example program:*

```
Imports System
Imports System.Windows.Forms

Public Module AppModule
    Public Sub Main( )
        Application.Run(New MainForm( ))
    End Sub
End Module

Public Class MainForm
    Inherits Form

    Public Sub New( )
        Text = "My MDI Application"
        'MDI parent form
        IsMdiContainer = True
        'Child form
        Dim myChild As New DocumentForm("My Document", Me)
        myChild.Show
    End Sub
```

## NOTES

## NOTES

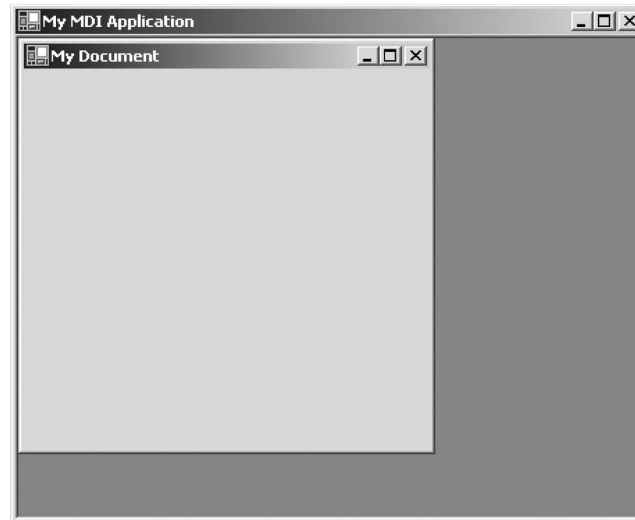
```
End Class
Public Class DocumentForm
    Inherits Form
    Public Sub New(ByVal name As String, ByVal parent As
Form)
        Text = name
        MdiParent = parent
    End Sub
End Class
```

End Class

Save the code in in a file named `MyApp.vb`, it can be compiled from the command line with this command:

```
vbc MyApp.vb /r:System.dll,System.Windows.Forms.dll
```

### Output:

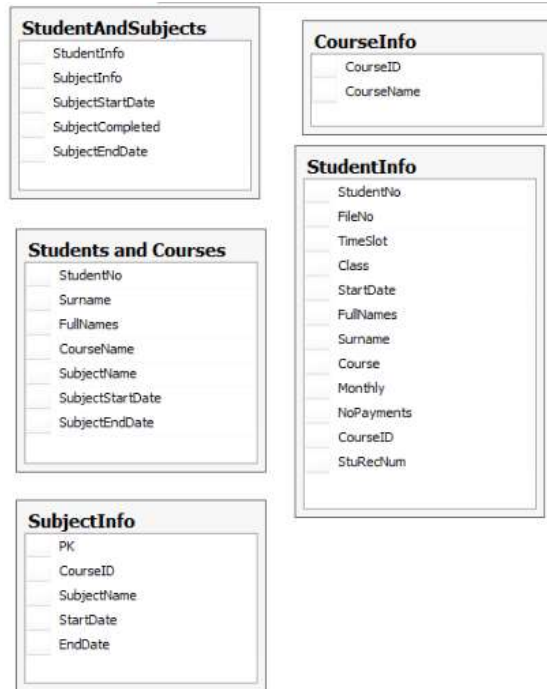


### Data Controls

The controls on the form can't see the database directly. They see the database through the Data control. The Data control in turn sees a RecordSet which happens to be a table of the database. You'll learn later how to specify other types of RecordSets for the Data control. As a relationship among the TextBox and other controls on the Form, the Data control, and the database is used.



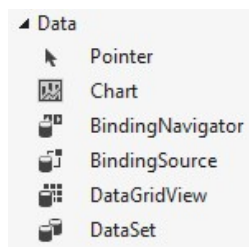
A database consists of various tables, consider the following tables:



## NOTES

### 16. Write a program to demonstrate data controls using multiple inter-connected tables.

Visual Basic 2012 Toolbox provide the following data controls as shown below:



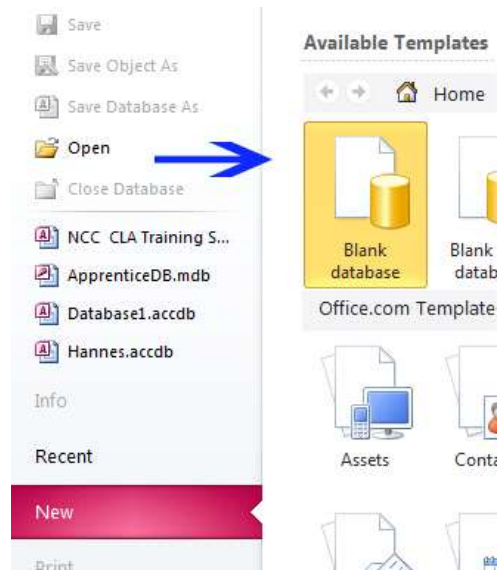
#### Step 1: Create the database.

Microsoft Access or SQL Server or other database can be used to create database. Following are the steps to create database using MS Access.

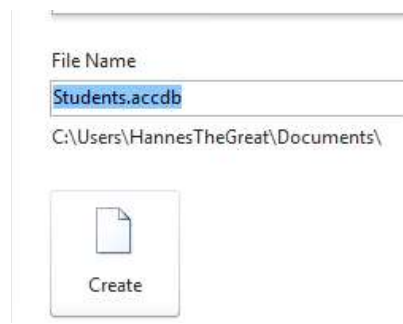
1. Open Microsoft Access.

**NOTES**

2. In the Available Templates, select Blank database, as shown below:



3. On the right side of the screen enter the File name, in this case: Students.accdb, then click Create, as shown below:



4. Inside the new Screen, edit the columns and data as shown below:

StudentNumber	StuydentName	StudentSurname
1	Hannes	du Preez
2	YourName	YourSurname
*	(New)	

5. Save the table as StudentInfo.

**VB.NET Project**

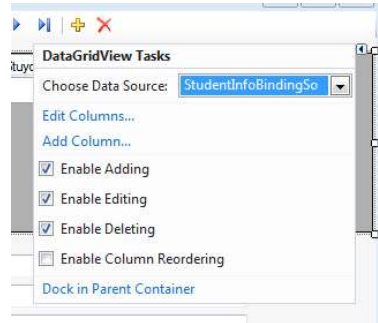
After creating database, next phase is to create a new VB.NET Windows Forms Project in VB.NET.

**Connecting to an MS Access 2010 Database using the Data Controls**

To connect to your Database, follow the steps given :

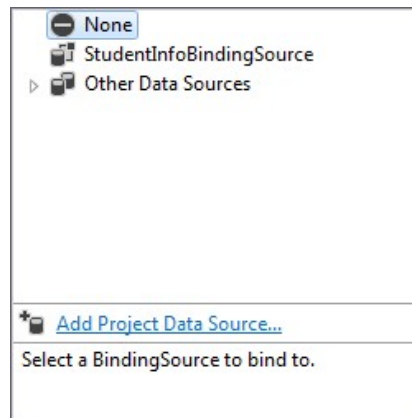
1. Click on the DataGridView on your Form.
2. A small right pointing triangle will appear.

3. Click on it. This will give you a screen similar as given below:
4. Click on the drop down arrow next to Choose Data Source. This will produce a screen that resembles as given below:

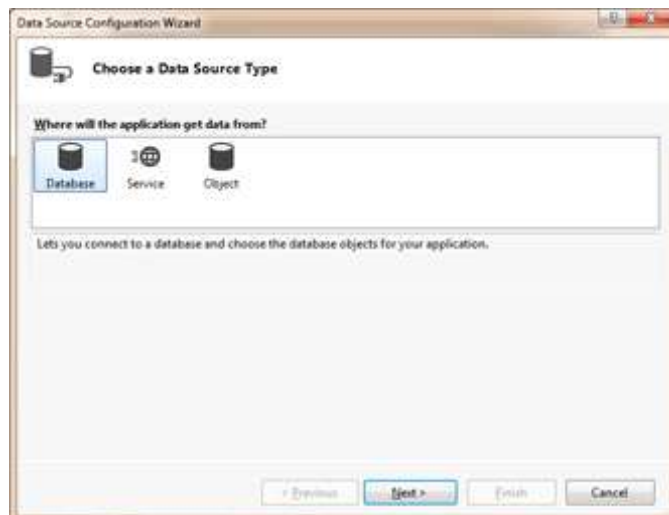


## NOTES

5. Click on Add Project Data Source. The Wizard will open as shown below:

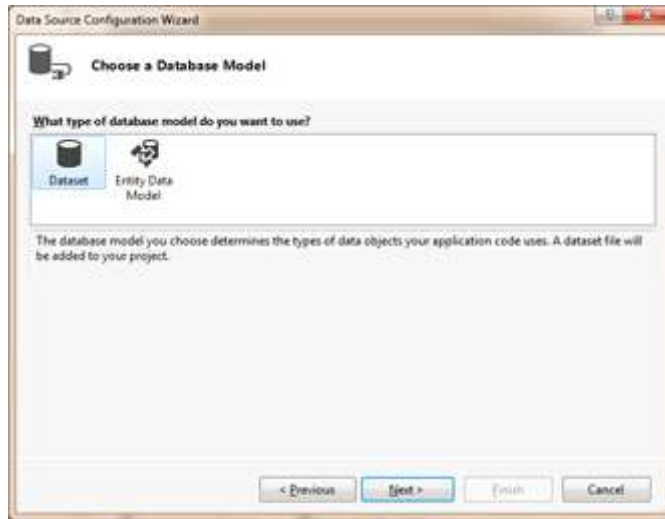


6. Make sure Database is selected, then click Next.

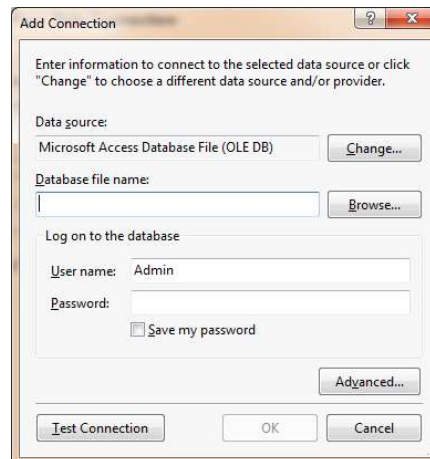


7. Make sure Dataset is selected, then click on Next.

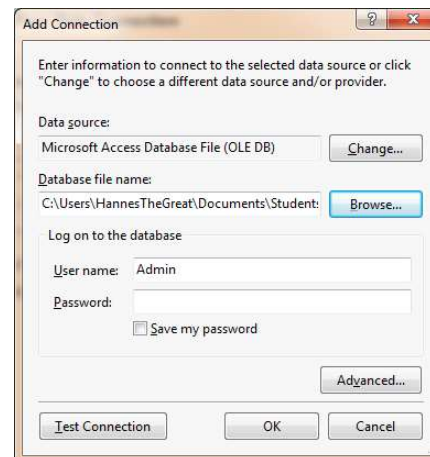
**NOTES**



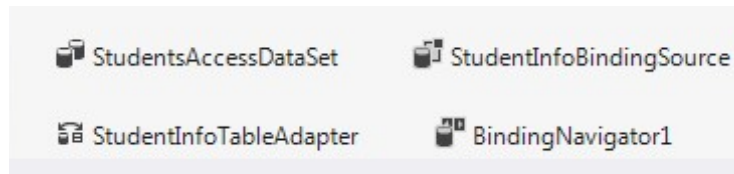
8. Click on New Connection.



9. Click on Browse next to Database name, and select your Students.accdb database, as shown below:



10. Click OK
11. Select Tables. Give the ConnectionString a name such as StudentAccessDataSet.
12. Click Finish.
13. The following controls will appear inside your design window.

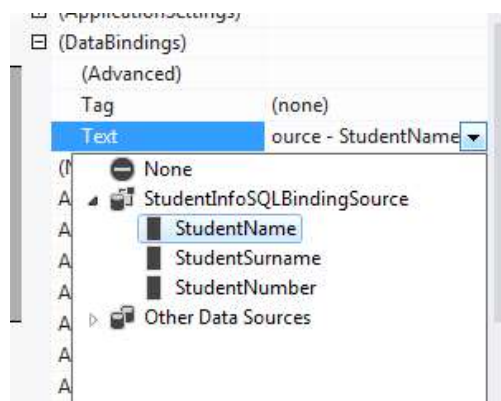


You will now see that your **DataGridView** shows the three columns of table. The DataGridView is now connected to your database. We still need to connect our Textboxes as well as the BindingNavigator to the database.

### Connecting the TextBoxes to Access Database

Follow the steps given below:

1. Select a TextBox and open the Properties Window.
2. Expand the DataBindings property (at the top of the list).
3. Select Text.
4. Select the appropriate Field you want to connect to from the displayed combobox, as shown below:



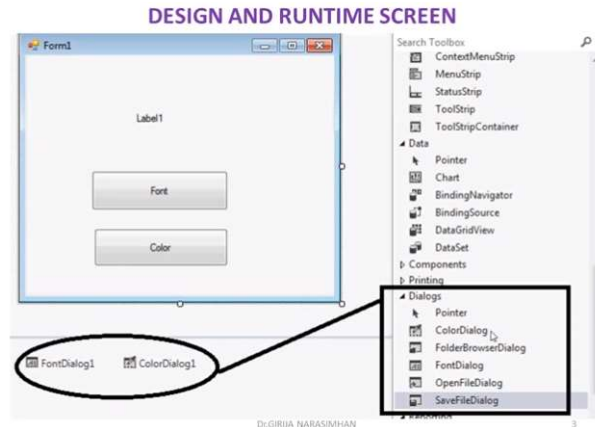
### Try Yourself:

- (i) Develop an application to demonstrate MDI form for student registration.
- (ii) Develop an application to demonstrate use of data control.

## NOTES

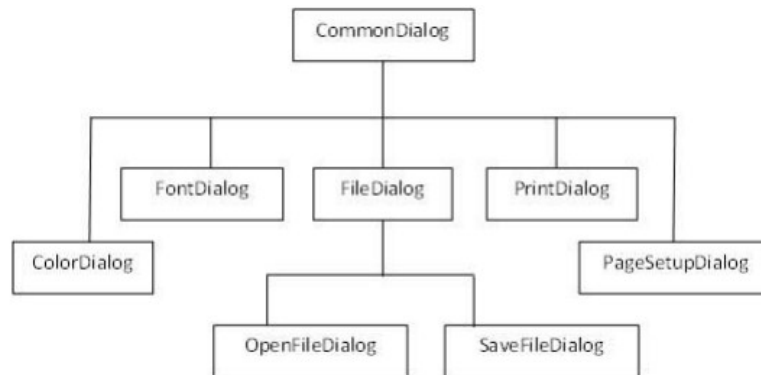
**17. Write a program to demonstrate application with common dialogs.**

**NOTES**

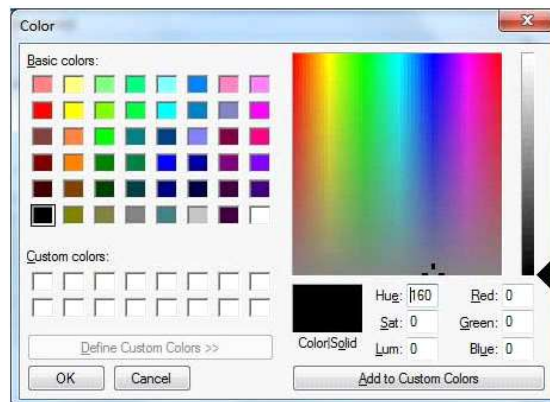


**18. Write a program to demonstrate application with common dialogs.**

There are many built-in dialog boxes to be used in Windows forms for various tasks like opening and saving files, printing a page, providing choices for colors, fonts, page setup, etc., to the user of an application. These built-in dialog boxes reduce the developer's time and workload.



The ColorDialog control class represents a common dialog box that displays available colors along with controls that enable the user to define custom colors. It lets the user select a color.



## Develop an application to change the forecolor of a label control using the color dialog box.

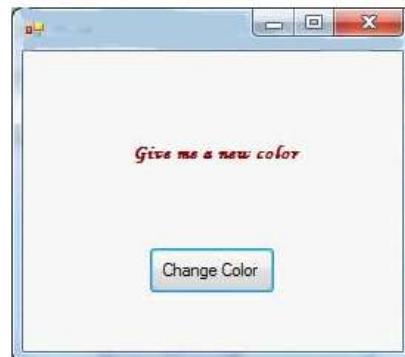
### Solution:

- Create a form and Drag and drop a label control, a button control and a ColorDialog control on the form.
- Set the Text property of the label and the button control to 'Give me a new Color' and 'Change Color', respectively.
- Change the font of the label as per your likings.
- Double-click the Change Color button and modify the code of the Click event.

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    If ColorDialog1.ShowDialog() <> Windows.Forms.DialogResult.Cancel Then
        Label1.ForeColor = ColorDialog1.Color
    End If
End Sub
```

When the application is compiled and run using Start button available at the Microsoft Visual Studio tool bar, it will show the following window:

### Output:



## 19. Write the steps to demonstrate Application with Menus.

In VB .NET MainMenu is the container for the Menu structure of the form. Menus are made of MenuItem objects that represent individual parts of a menu (like File->New, Open, Save, Save As etc). It is MenuItem's click event that makes Menu Event of the MenuItem.

### Creating Menus

Drag a MainMenu component from the toolbar onto the form. When you add a MainMenu component to the form, it appears in the component tray below the form.

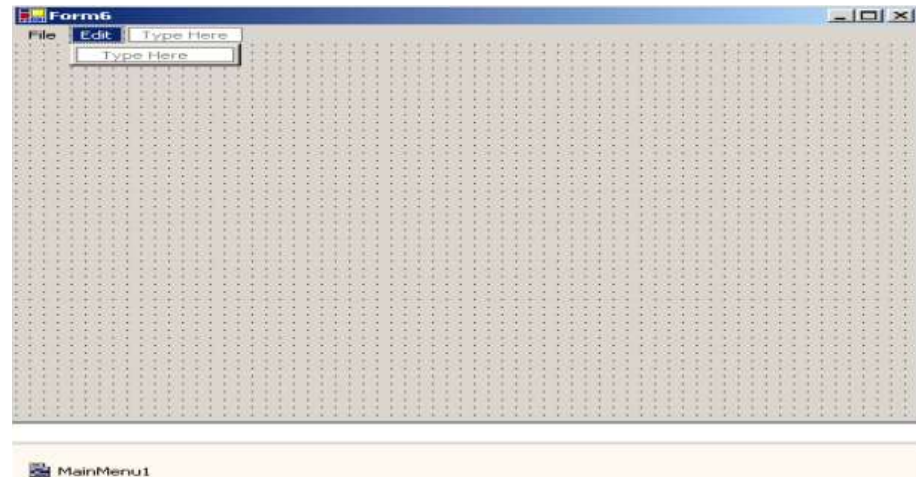
## NOTES

**NOTES**

Windows form designer will add the MenuItem's for this by default, you need not add this. Once when you finish adding a MainMenu component to the form you will notice a "TypeHere" box towards the top-left corner of the form.

To create a menu all you have to do is click on the "TypeHere" text which opens up a small textbox allowing you to enter text for the menu.

You can use the arrow keys on the keyboard to create a submenu or add other items to that menu or click on the first menu item and use the left/right arrow keys on the keyboard to create a new menu item as shown below:



**20. Write the steps to demonstrate how to open and save files using menus. Design a form using menus - Create submenus to open and save files using open and save dialog boxes.**

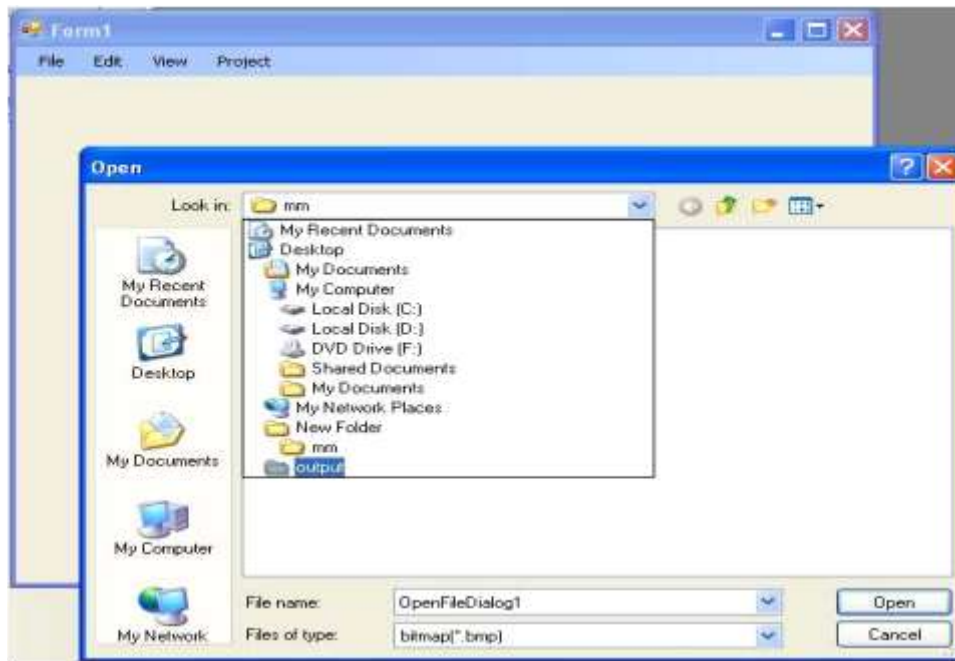
Steps to create windows application:

- Select File->New-> Project. Visual studio will display the new project dialog box.
- In the new project dialog box, click the windows Application icon. In the name field, type a project name that describes the program you are binding, such as Demo program.
- Click ok button, you will get a design window where you can drag and drop controls onto form.
- To display the toolbox that contains the control you can drag and drop on to your form, select View->Toolbox which will open Toolbox window.
- In the toolbox window, drag and drop menu control - Create submenus like copy, paste, undo - Create submenus like new, open, save, save as etc.
- Drag and drop opendirialogbox and savedialogbox on to the form.
- Write the code to open the dialogbox and save the respected form using various methods.
- Select Debug menu and then start to run the program.

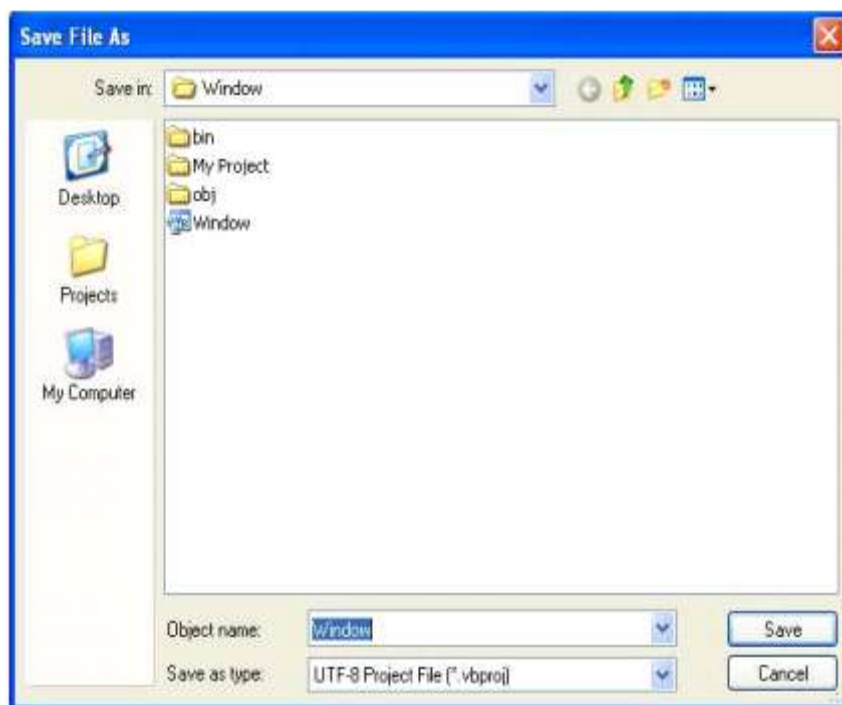


**Output:**

*VB.NET & RDBMS Lab*



**NOTES**



**Try Yourself:**

- (i) Develop an application to design menus including File, Edit, and Help.
- (ii) Develop an application to demonstrate sub-menu.

**21. Write a program to demonstrate Drag and Drop operation.**

Let us look at some examples, starting with simple drag and drop operation. Create a Visual Basic.net windows application and design a form with control & Drag Drop event procedure as follows:

**NOTES**

To enable drag & drop for text, follow the steps.

1. Place two **textboxes** and set **Allowdrop** property of a second **textbox** to **true**.

2. Add the following code

```
Private MouseIsDown As Boolean = False 'variable
declaration

Private Sub TextBox1_MouseDown(ByVal sender As Object,
ByVal e As _
System.Windows.Forms.MouseEventArgs) Handles
TextBox1.MouseDown
    'Set a flag to show that the mouse is down.
    MouseIsDown = True
End Sub

Private Sub TextBox1_MouseMove(ByVal sender As Object,
ByVal e As _
System.Windows.Forms.MouseEventArgs) Handles
TextBox1.MouseMove
    If MouseIsDown Then
        'Initiate dragging.
        TextBox1.DoDragDrop(TextBox1.Text, DragDropEffects.Copy)
    End If
    MouseIsDown = False
End Sub

Private Sub TextBox2_DragEnter(ByVal sender As Object,
ByVal e As _
System.Windows.Forms.DragEventArgs) Handles
TextBox2.DragEnter
    'Check the format of the data being dropped.
    If (e.Data.GetDataPresent(DataFormats.Text)) Then
        'Display the copy cursor.
        e.Effect = DragDropEffects.Copy
    Else
        'Display the no-drop cursor.
        e.Effect = DragDropEffects.None
    End If
End Sub
```

```

Private Sub TextBox2_DragDrop(ByVal sender As Object,
ByVal e As _
System.Windows.Forms.DragEventArgs) Handles
TextBox2.DragDrop
    'Paste the text.
    TextBox2.Text = e.Data.GetData(DataFormats.Text)
End Sub

```

In the above code, the **MouseDown** event is used to set a flag showing that the mouse is down, and then the **DoDragDrop** method is called in the **MouseMove** event. Although you could initiate the drag in the **MouseDown** event, doing so would create undesirable behavior: Every time a user clicks the control, the no-drag cursor would be displayed.

The **DoDragDrop** method takes two parameters:

- **Data** parameter, which in this case takes the **Text** property of the **TextBox**
- **allowedEffects** parameter, which in this case only allows copying

Also in the **MouseMove** event the **MouseIsDown** flag is set to **False**. Although unnecessary in this example, if you had multiple controls that support dragging, you could get a run-time exception.

In the **DragEnter** event, the **GetDataPresent** method checks the format of the data being dragged. In this case it is text, so the **Effect** property is set to **Copy**, which in turn displays the copy cursor.

In the **DragDrop** event, the **GetData** method is used to retrieve the text from the **DataObject** and assign it to the target **TextBox**.

The next section provides an example of dragging a different type of data and providing support for both cutting and copying.

To enable drag and drop for a picture:

1. Add two **picturebox** control to a form
2. Add the following code.

```

Private Sub Form1_Load(ByVal sender As System.Object,
ByVal e As _
System.EventArgs) Handles MyBase.Load
    'Enable dropping.
    PictureBox2.AllowDrop = True
End Sub

Private Sub PictureBox1_MouseDown(ByVal sender As Object,
ByVal e As _
System.Windows.Forms.MouseEventArgs) Handles
PictureBox1.MouseDown
    If Not PictureBox1.Image Is Nothing Then
        'Set a flag to show that the mouse is down.

```

## NOTES

## NOTES

```
m_MouseIsDown = True
End If
End Sub

Private Sub PictureBox1_MouseMove(ByVal sender As Object,
ByVal e As _
System.Windows.Forms.MouseEventArgs) Handles
PictureBox1.MouseMove
If m_MouseIsDown Then
`Initiate dragging and allow either copy or move.
PictureBox1.DoDragDrop(PictureBox1.Image,
DragDropEffects.Copy Or _
DragDropEffects.Move)
End If
m_MouseIsDown = False
End Sub

Private Sub PictureBox2_DragEnter(ByVal sender As Object,
ByVal e As _
System.Windows.Forms.DragEventArgs) Handles
PictureBox2.DragEnter
If e.Data.GetDataPresent(DataFormats.Bitmap) Then
`Check for the CTRL key.
If e.KeyState = 9 Then
e.Effect = DragDropEffects.Copy
Else
e.Effect = DragDropEffects.Move
End If
Else
e.Effect = DragDropEffects.None
End if
End sub

Private Sub PictureBox2_DragDrop(ByVal sender As Object,
ByVal e As _
System.Windows.Forms.DragEventArgs) Handles
PictureBox2.DragDrop
`Assign the image to the PictureBox.
PictureBox2.Image = e.Data.GetData(DataFormats.Bitmap)
`If the CTRL key is not pressed, delete the source picture.
If Not e.KeyState = 8 Then
PictureBox1.Image = Nothing
End If
End Sub
```

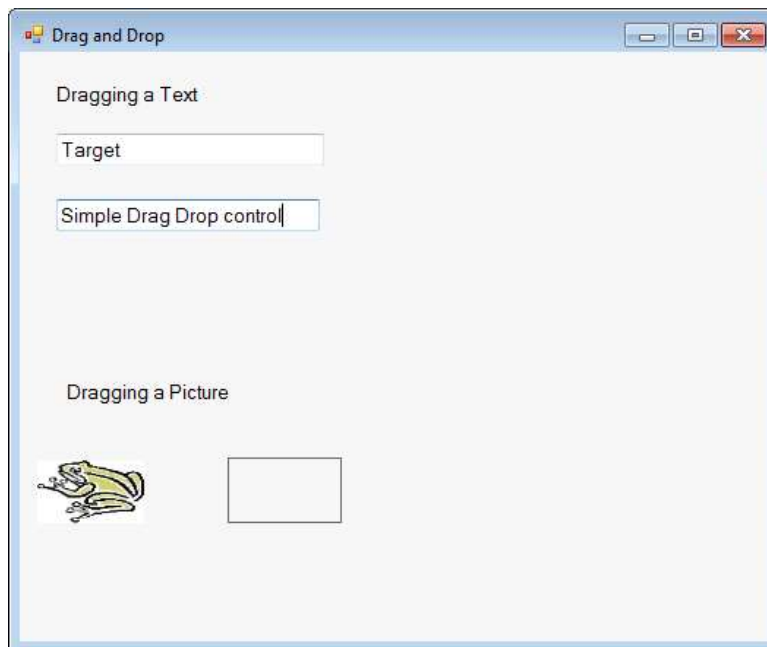
In the above code, note that the **AllowDrop** property for the second **PictureBox** control is set in the **Form1\_Load** event. This is necessary because the **AllowDrop** property is not available at design time.

In the **MouseDown** event, the code first checks to make sure that there is an image assigned to the **PictureBox**; otherwise, after you moved the picture, subsequent clicks would raise an exception.

Also note that in both the **DragEnter** and **DragDrop** events the code checks to see if the CTRL key is pressed to determine whether to copy or move the picture. Why are the values different? In the **DragEnter** event, the left mouse button is down, resulting in a value of 8 for the CTRL key plus 1 for the left mouse button.

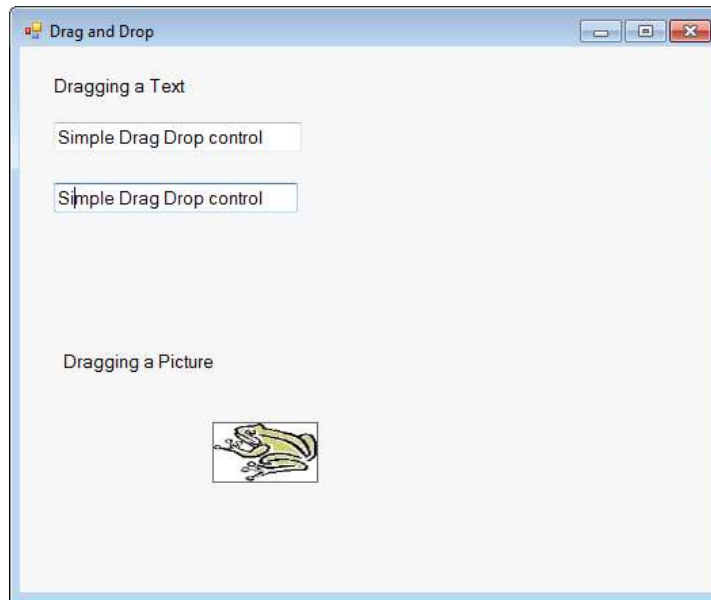
Both examples, so far have dealt with dragging between two controls on the same form; they would also work for dragging items between controls on different forms within an application. The next example demonstrates accepting items dropped from another application — in this case, files that are dragged from Windows Explorer.

### Output:



### NOTES

## NOTES



Applications need to respond to events when they occur. There are mainly two types of events **Mouse events and Keyboard events**.

Mouse Events:

- **MouseDown** ” it occurs when a mouse button is pressed
- **MouseEnter** ” it occurs when the mouse pointer enters the control
- **MouseHover** ” it occurs when the mouse pointer hovers over the control
- **MouseLeave** ” it occurs when the mouse pointer leaves the control
- **MouseMove** ” it occurs when the mouse pointer moves over the control
- **MouseUp** ” it occurs when the mouse pointer is over the control and the mouse button is released
- **MouseWheel** ” it occurs when the mouse wheel moves and the control has focus

**Try Yourself:**

- (i) Develop an application to demonstrate MouseMove and MouseUp events.
- (ii) Develop an application to demonstrate Drag and Drop Events.

**22. Write a program to demonstrate the use of mouse events.**

Apply the following steps:

- (i) Create a new form.
- (ii) Add three labels, three text boxes and a button control in the form.
- (iii) Change the text properties of the labels to - Customer ID, Name and Address, respectively.
- (iv) Change the name properties of the text boxes to txtID, txtName and txtAddress, respectively.

(v) Change the text property of the button to 'Submit'.

(vi) Add the following code in the code editor window:

```
Public Class Form1
    Private Sub Form1_Load(sender As Object, e
    As EventArgs) Handles MyBase.Load
        ` Set the caption bar text of the form.
            Me.Text = "tutorialspont.com"
        End Sub

        Private Sub txtID_MouseEnter(sender As Object, e As
    EventArgs) _
            Handles txtID.MouseEnter
            `code for handling mouse enter on ID textbox
            txtID.BackColor=Color.CornflowerBlue
            txtID.ForeColor=Color.White
        EndSub

        Private Sub txtID_MouseLeave(sender As Object, e
    As EventArgs) _
            Handles txtID.MouseLeave
            `code for handling mouse leave on ID textbox
            txtID.BackColor = Color.White
            txtID.ForeColor = Color.Blue
        End Sub

        Private Sub txtName_MouseEnter(sender As Object, e
    As EventArgs) _
            Handles txtName.MouseEnter
            `code for handling mouse enter on Name textbox
            txtName.BackColor=Color.CornflowerBlue
            txtName.ForeColor=Color.White
        EndSub

        Private Sub txtName_MouseLeave(sender As Object, e
    As EventArgs) _
            Handles txtName.MouseLeave
            `code for handling mouse leave on Name textbox
            txtName.BackColor = Color.White
            txtName.ForeColor = Color.Blue
        End Sub
    End Class
```

## NOTES

## NOTES

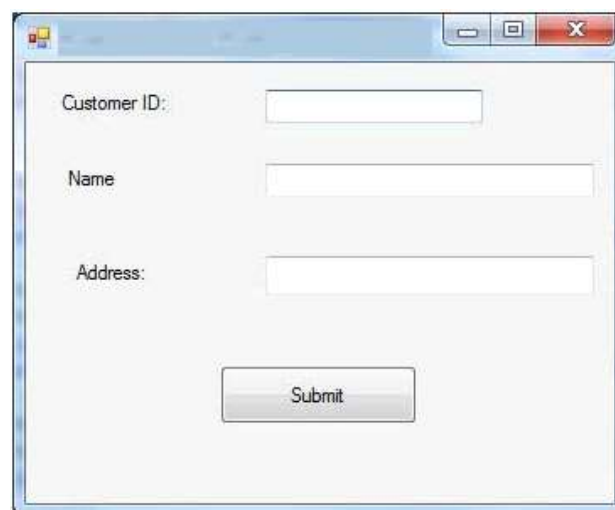
```
Private Sub txtAddress_MouseEnter(sender As Object,
e As EventArgs) _
    Handles txtAddress.MouseEnter
    `codefor handling mouse enter on Address textbox
    txtAddress.BackColor=Color.CornflowerBlue
    txtAddress.ForeColor=Color.White
EndSub

PrivateSub txtAddress_MouseLeave(sender AsObject, e
AsEventArgs) _
Handles txtAddress.MouseLeave
`code for handling mouse leave on Address textbox
    txtAddress.BackColor = Color.White
    txtAddress.ForeColor = Color.Blue
End Sub

Private Sub Button1_Click(sender As Object, e As
EventArgs) _
    Handles Button1.Click
    MsgBox("Thank you " & txtName.Text & ", for your kind
cooperation")
End Sub
End Class
```

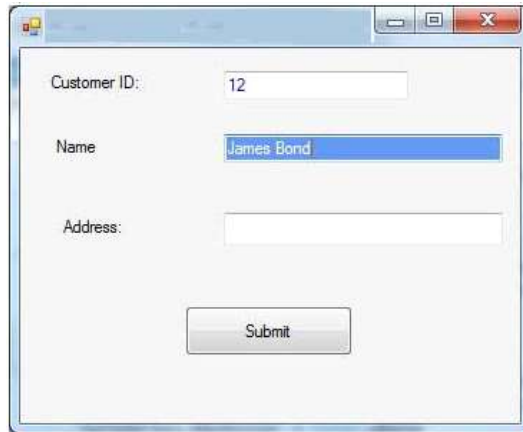
### Output:

When the above code is executed and run using **Start** button available at the Microsoft Visual Studio tool bar, it will show the following window:





Try to enter text in the text boxes and check the mouse events:



## NOTES

### Keyboard Events

Following are the various keyboard events related with a Control class –

- **KeyDown** – occurs when a key is pressed down and the control has focus
- **KeyPress** – occurs when a key is pressed and the control has focus
- **KeyUp** – occurs when a key is released while the control has focus

The event handlers of the KeyDown and KeyUp events get an argument of type **KeyEventArgs**. This object has the following properties –

- **Handled** – indicates if the KeyPress event is handled
- **KeyChar** – stores the character corresponding to the key pressed

### 23. Write a program to demonstrate the use of keyboard events.

#### Solution:

Let us continue with the previous example to show how to handle keyboard events. The code will verify that the user enters some numbers for his customer ID and age.

- Create a new form
- Add a label with text Property as 'Age' and add a corresponding text box named txtAge.
- Add the following codes for handling the KeyUP events of the text box txtID.

```
PrivateSub txtID_KeyUP(sender AsObject, e AsKeyEventArgs)
    -
    Handles txtID.KeyUp
    If (NotChar.IsNumber (ChrW (e.KeyCode))) Then
    MessageBox.Show("Enter numbers for your Customer ID")
        txtID.Text=""
    EndIf
EndSub
```

(iv) Add the following codes for handling the KeyUP events of the text box txtID.

```
PrivateSub txtAge_KeyUP(sender AsObject, e AsKeyEventArgs)
```

```
Handles txtAge.KeyUp
```

```
If (NotChar.IsNumber(ChrW(e.keyCode))) Then
```

```
MessageBox.Show("Enter numbers for age")
```

```
txtAge.Text=""
```

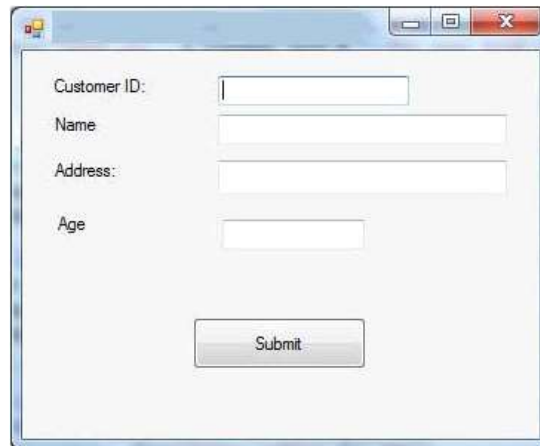
```
EndIf
```

```
EndSub
```

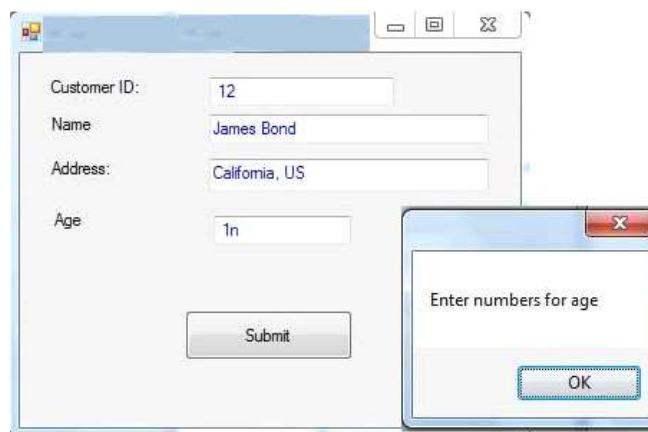
## NOTES

### Output:

When the above code is executed and run using **Start** button available at the Microsoft Visual Studio tool bar, it will show the following window:

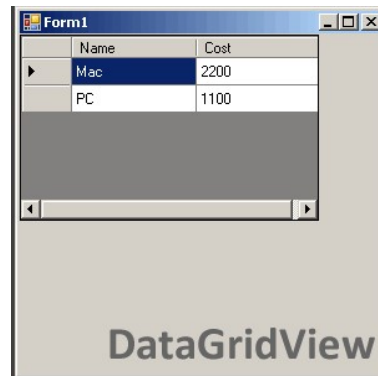


If you leave the text for age or ID as blank or enter some non-numeric data, it gives a warning message box and clears the respective text:



**24. Write an ADO.NET code to show records in DataGridView Control.**

**DataGridView** provides a visual interface to data. It is an excellent way to display and allow editing for your data. It is accessed with VB.NET code. Data edited in the DataGridView can then be persisted in the database.

**25. Write a program to demonstrate the concept DataGridView control.****Solution:**

First, you should add a DataGridView control to your Windows Forms application by double-clicking on the control name in the Visual Studio designer panel. After you add the control, you can add the Load event on the form.

**Load:** You can create the Load event on the Form's event pane in Visual Studio. We use Load in this example. Here, We use an empty DataTable on the DataGridView control. We assign the DataSource property.

```
Public Class Form1
    Private Sub Form1_Load(ByVal sender As
System.Object, _
                                ByVal e As System.EventArgs)
Handles MyBase.Load
    \
    \ Fill in the data grid on form load.
    \
    DataGridView1.DataSource = GetDataTable()
    End Sub

    Private Function GetDataTable() As DataTable
    \
    \This Function needs to build the data table.
    \
    Return New DataTable()
    End Function
End Class
```

**NOTES**

**NOTES**

This event handler is executed when the program starts up and when the DataGridView control is displayed. The Form1\_Load sub calls into the GetDataTable function, which would return a DataTable from your database in SQL Server.

Assigning the DataSource property on DataGridView copies no data. It allows the DataGridView to read in the DataTable and display all its contents on the screen in grid form. This is an efficient way to populate DataGridView.

**DataTable**

When using the DataGridView control in Windows Forms, you should use the lightning bolt panel. This allows you to manipulate the events on the control. DataGridView has many events, and this article doesn't describe them all. However, you will often want the CellClick, SelectionChanged, CellDoubleClick, and KeyPress events, depending on your requirements.

**Objects**

You can use an object collection, such as a List(Of String), in your DataGridView using the Visual Basic language. The object collection will be read. Its properties (get accessors) will be used to display the values on the screen.

**List**

This is the easiest way to get started with DataGridView. But it may be less effective than more complex approaches.

```
Public Class Test
    Public Sub New(ByVal name As String, ByVal cost As
String)
        _name = name
        _cost = cost
    End Sub

    Private _name As String
    Public Property Name() As String
        Get
            Return _name
        End Get
    Set(ByVal value As String)
        _name = value
    End Set
End Property

    Private _cost As String
    Public Property Cost() As String
        Get
```

```

        Return _cost
    End Get
Set (ByVal value As String)
    _cost = value
End Set
End Property
End Class

```

## 26. Write a program that uses DataGridView with class.

```

Public Class Form1

    Private Sub Form1_Load(ByVal sender As System.Object,
        -
        ByVal e As System.EventArgs)
        Handles MyBase.Load
        Dim list = New List(Of Test)
        list.Add(New Test("Mac", 2200))
        list.Add(New Test("PC", 1100))
        DataGridView1.DataSource = list
    End Sub
End Class

```

This program includes the Public Class Test definition, which encapsulates two properties with backing stores. The names of these properties are Name and Cost. These could be used for an inventory of merchandise.

After the Dim List is allocated, two new Test objects are added to its contents. These two objects are reflected in the DataGridView output to the screen. You can see the four cells from the four values in the example in the screenshot.

You can hide the row headers, which are the boxes on the left of the DataGridView control, from appearing on the screen. The screenshot shows what the row headers look like on DataGridView controls.

## 27. Write a program that adds rows in database.

```

Public Class Form1

    Private Sub Form1_Load(ByVal sender As System.Object,
        -
        ByVal e As System.EventArgs)
        Handles MyBase.Load
        ` Add row using the Add subroutine.
        Dim n As Integer = DataGridView1.Rows.Add()
        DataGridView1.Rows.Item(n).Cells(0).Value = "First"
        DataGridView1.Rows.Item(n).Cells(1).Value = "Second"
    End Sub
End Class

```

## NOTES

We can obtain the location of the current cell in VB.NET code. One way you can do this is add the SelectionChanged event. As a reminder, you can add events easily in Visual Studio by using the lightning bolt panel.

**NOTES**

**Next**, in the DataGridView1\_SelectionChanged event, you can access the DataGridView1.CurrentCellAddress property. This is a System.Drawing.Point type that has two instance properties, X and Y.

**28. Write a program that gets current record.**

```
Public Class Form1

    Private Sub Form1_Load(ByVal sender As
System.Object, _
                                ByVal e As System.EventArgs)
Handles MyBase.Load
Dim list = New List(Of Test)
list.Add(New Test("Mac", 2200))
list.Add(New Test("PC", 1100))
        DataGridView1.DataSource = list
    End Sub

    Private Sub DataGridView1_SelectionChanged(ByVal
sender As System.Object, _
                                ByVal e
As System.EventArgs) _
                                Handles
DataGridView1.SelectionChanged
Dim y As Integer = DataGridView1.CurrentCellAddress.Y
        Dim x As Integer =
DataGridView1.CurrentCellAddress.X

    ` Write coordinates to console.
Console.WriteLine(y.ToString + " " + x.ToString)
    End Sub
End Class
```

**29. Write a program to bind data in a multiline TBox by querying in another TBox.**

```
Imports System.Data
Imports System.Data.SqlClient

Partial Class _Default Inherits System.Web.UI.Page

Dim constr As String =
```

```

ConfigurationManager.ConnectionStrings("DatabaseConnectionString1").Connection
String

```

```

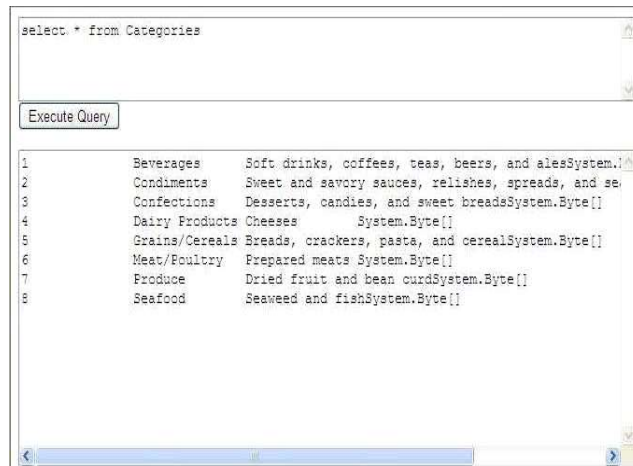
Protected Sub Button_Click(ByVal sender As Object,
ByVal e As System.EventArgs) Handles Button1.Click
Dim conPubs As SqlConnection Dim cmdSelect As
SqlCommand Dim dtrResults As SqlDataReader Dim intField
As Integer conpubs = New SqlConnection(constr)
conPubs.Open()
cmdSelect = New SqlCommand(txtQuery.Text, conPubs)
dtrResults = cmdSelect.ExecuteReader()
txtResults.Text=""
While dtrResults.Read() txtResults.Text&=vbNewLine For
intField = 0 To dtrResults.FieldCount - 1
txtResults.Text&=
dtrResults(intField).ToString().PadRight(15) Next
End While dtrResults.Close() conPubs.Close()

End Sub
End Class

```

## NOTES

### Output:



### 30. Write a program to display the phone number of an author from database.

```

Imports System.Data.SqlClient Partial Class _Default
Inherits System.Web.UI.Page Dim constr As String =
ConfigurationManager.ConnectionStrings
("DatabaseConnectionString1").Connection String

Protected Sub Button1_Click(ByVal sender As Object,
ByVal e As System.EventArgs) Handles Button1.Click

```

**NOTES**

```

Dim conpubs As SqlConnection Dim strselect AsString
Dim cmdselect As SqlCommand
conpubs = NewSqlConnection(constr)
strselect = "select phone from Author

cmdselect = New SqlCommand(strselect,
cmdselect.Parameters.Add("@Au_fname",
cmdselect.Parameters.Add("@Au_lname",
conpubs.Open() label5.Text = cmdselect.ExecuteScalar()
conpubs.Close() End SubEnd Class

```

**Output:**

Another PhoneLookup

Au\_fname  
Rajeev

Au\_lname  
Ranjan

lookup

phone 456987

**31. Write a program showing data bind using dropdown list.**

```

Imports System.Data.SqlClient Partial Class _Default
Inherits System.Web.UI.Page Dim constr As String=
ConfigurationManager.ConnectionStrings
("DatabaseConnectionString1").Connection String
Protected Sub Button1_Click(ByVal sender As Object,
ByVal e As System.EventArgs) Handles Button1.Click
LB1.Text = "you pick" & ComboBox1.SelectedValue EndSub
Protected Sub Page_Load(ByVal sender As Object, ByVale
As System.EventArgs) Handles Me.Load
If Not IsPostBack Then
Dim conpubs As SqlConnection Dim cmdselect As
SqlCommand Dim dtrAuthor AsSqlDataReader
conpubs = New SqlConnection(constr)
cmdselect = New SqlCommand("select Au_fname from
Author", conpubs) conpubs.Open()
dtrAuthor = cmdselect.ExecuteReader()
ComboBox1.DataSource = dtrAuthor

```



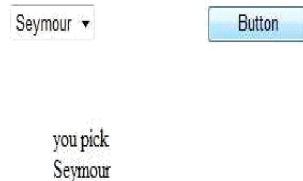
```

ComboBox1.DataTextField = "Au_fname"
ComboBox1.DataBind() dtrAuthor.Close() conpubs.Close()
End If
End Sub
EndClass

```

VB.NET & RDBMS Lab

### Output:



### NOTES

### 32. Write a program to insert the data into database using Execute-Non Query.

```

Imports System.Data.SqlClient Partial Class _Default
Inherits System.Web.UI.Page Dim constr As String=
ConfigurationManager.ConnectionStrings
("DatabaseConnectionString1").Connection String
Protected Sub Button1_Click(ByVal sender As Object,
ByVal e As System.EventArgs) Handles Button1.Click
Dim conpubs As SqlConnection Dim strInsert As String
Dim cmdInsert As SqlCommand
conpubs = New SqlConnection(constr)
strInsert = "Insert
Author(Au_fname,Au_lname) values (@Au_fname,@Au_lname)"
cmdInsert = New SqlCommand(strInsert, conpubs)
cmdInsert.Parameters.Add("@Au_fname", txtfname.Text)
cmdInsert.Parameters.Add("@Au_lname", txtlname.Text)
conpubs.Open() cmdInsert.ExecuteNonQuery()
conpubs.Close()
Response.Write("New Rowinserted")
End Sub
End Class

```

### Output:



## NOTES

**33. Write a program to delete the data in database using Execute non-query.**

```
Imports System.Data.SqlClient
Partial Class _Default
Inherits System.Web.UI.Page
Dim constr As String=
ConfigurationManager.ConnectionStrings
("RajeevConnectionString1").ConnectionString
Protected Sub Button1_Click(ByVal sender As Object, ByVal e As
System.EventArgs) Handles Button1.Click
Dim conpubs As SqlConnection
Dim strdelete As String
Dim cmddelete As SqlCommand
conpubs = New SqlConnection(constr)
strdelete = "delete from Author where
Au_fname=@Au_fname"
cmddelete = New SqlCommand(strdelete, conpubs)
cmddelete.Parameters.Add("@Au_fname", TextBox1.Text)
conpubs.Open()
cmddelete.ExecuteNonQuery()
Response.Write("Row delete")
conpubs.Close()
End Sub
End Class
```

**Output:**

Row delete

Au\_fname

Rajeev

Au\_lname

Remove

**34. Write a program for Databinding using datalist control.**

```
Imports System.Data.SqlClient
Partial Class _Default
Inherits System.Web.UI.Page
Dim constr As String=
ConfigurationManager.ConnectionStrings("DatabaseConnectionString1").ConnectionString
Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load
Dim conpubs As SqlConnection
Dim cmdselect As SqlCommand
Dim dtrAuthor As SqlDataReader
conpubs = New SqlConnection(constr)
```

```
cmdselect = New SqlCommand("Select Au_fname from
Author", conpubs) conpubs.Open() dtrAuthor =
cmdselect.ExecuteReader() dlstAuthor.DataSource =
dtrAuthordlstAuthor.DataBind() dtrAuthor.Close()
conpubs.Close() End Sub
```

End Class

### Output:

```
Rajeev
Rajeev
Sonu
sanjay
```

### 35. Write a program to bind data using template in datalist.

```
Imports System.Data.SqlClient Partial Class _Default
Inherits System.Web.UI.Page Dim constr As String =
ConfigurationManager.ConnectionStrings
("DatabaseConnectionString1").Connection String
Protected Sub Page_Load(ByVal sender As Object, ByVale
As System.EventArgs) Handles Me.Load
Dim conpubs As SqlConnection Dim cmdselect As
SqlCommand Dim dtrAuthor As SqlDataReader
conpubs = New SqlConnection(constr)
cmdselect = New SqlCommand("Select * from Author",
conpubs) conpubs.Open() dtrAuthor =
cmdselect.ExecuteReader() dtrlstAuthor.DataSource =
dtrAuthordtrlstAuthor.DataBind() dtrAuthor.Close()
conpubs.Close()
End Sub
End Class
```

### Output:

#### Author Phone Number

```
Au_name:Rajeev
Au_lname:Jaiswal
Phone:234567
Au_name:Soni
Au_lname:kumar
Phone:345234
Au_name:trideep
Au_lname:sharma
Phone:345675
Au_name:Deepak
Au_lname:shariya
Phone:565432
Au_name:Ranjan
Au_lname:jaiswal
Phone:876875
```

## NOTES

**36. Write a program to bind data using data grid.**

```
Imports System.Data.SqlClient
```

**NOTES**

```
Partial Class _Default Inherits System.Web.UI.Page Dim
constr As String=
ConfigurationManager.ConnectionStrings("RajeevConnectionString1").ConnectionSt
ring Protected Sub Page_Load(ByVal sender As Object,
ByVal e As System.EventArgs) Handles Me.Load
Dim conpubs As SqlConnection Dim cmdselect As
SqlCommand
conpubs = New SqlConnection(constr)
cmdselect = New SqlCommand("select * from Author",
conpubs) conpubs.Open()
datagrid.DataSource = cmdselect.ExecuteReader()
datagrid.DataBind()
conpubs.Close() End Sub
Protected Sub datagrid_SelectedIndexChanged(ByVal sender
As Object, ByVal e As System.EventArgs) Handles
datagrid.SelectedIndexChanged
End Sub
End Class
```

**Output:**


Rajeev Ranjan
Ranjan Jaiswal
Somu jaiswal
Trideep Sharma
Deepak Shariya

**37. Write a program to bind data using template column in data grid.**

```
Imports System.Data.SqlClient Partial Class _Default
Inherits System.Web.UI.Page Dim constr As String =
ConfigurationManager.ConnectionStrings
("DatabaseConnectionString1").Connection String
Protected Sub Page_Load(ByVal sender As Object, ByVal e
As System.EventArgs) Handles Me.Load
Dim conpubs As SqlConnection Dim cmdselect As
SqlCommand conpubs = New SqlConnection(constr)
cmdselect = New SqlCommand("select * from Author",
conpubs) conpubs.Open() datagrid.DataSource =
cmdselect.ExecuteReader() datagrid.DataBind()
conpubs.Close()
End Sub
End Class
```

**Output:**

Rajeev	phone: 565434 city: chennai
Ranjan	phone: 676545 city: chapra
Trideep	phone: 676546 city: Guwahati
sonu	phone: 565678 city: chapra

**NOTES****38. Write a program to bind data using hyperlink column in data grid.**

```
Imports System.Data.SqlClient Partial Class _Default
Inherits System.Web.UI.Page

Dim constr As String =
ConfigurationManager.ConnectionStrings
("DatabaseConnectionString1").Connection String
Protected Sub Page_Load(ByVal sender As Object, ByVal e
As System.EventArgs) Handles Me.Load

Dim conpubs As SqlConnection Dim cmdselect As
SqlCommand
conpubs = New SqlConnection(constr)
cmdselect = New SqlCommand("select * from Author",
conpubs)

conpubs.Open()
datagrlnk.DataSource = cmdselect.ExecuteReader()
datagrlnk.DataBind()
conpubs.Close()

End Sub
End Class
```

**Output:****39. Write a program to bind data using button column in data grid.**

```
Imports System.Data.SqlClient Partial Class _Default
Inherits System.Web.UI.Page Dim constr As String =
ConfigurationManager.ConnectionStrings
("DatabaseConnectionString1").Connection String
```

**NOTES**

```

Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load
    If Not IsPostBack Then
        Dim conpubs As SqlConnection Dim cmdselect As SqlCommand
        conpubs = New SqlConnection(constr)
        cmdselect = New SqlCommand("select * from Author", conpubs)
        conpubs.Open() GridView1.DataSource = cmdselect.ExecuteReader()
        GridView1.DataBind() conpubs.Close() End If
    End Sub

Protected Sub GridView1_RowCommand(ByVal sender As Object, ByVal e As System.Web.UI.WebControls.GridViewCommandEventArgs) Handles GridView1.RowCommand
    If e.CommandName = "select" Then
        GridView1.SelectedRowStyle.BackColor = Drawing.Color.Green
    Else
        GridView1.SelectedRowStyle.BackColor = Drawing.Color.White End If
    End Sub
End Class

```

**40. Write a program to create link using event in data list.**

```

Imports System.Data.SqlClient Partial Class _Default
    Inherits System.Web.UI.Page Dim constr As String = ConfigurationManager.ConnectionStrings("RajeevConnectionString1").ConnectionString
    Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load
        If Not IsPostBack Then
            Dim conpubs As SqlConnection
            Dim cmdselect As SqlCommand
            Dim dtrAuthor As SqlDataReader
            conpubs = New SqlConnection(constr)
            cmdselect = New SqlCommand("Select * from Author", conpubs)
            conpubs.Open()
            dtrAuthor = cmdselect.ExecuteReader()
            dtrlstAuthor.DataSource = dtrAuthor
            dtrlstAuthor.DataBind() dtrAuthor.Close()
            conpubs.Close()
        End If
    End Sub

Protected Sub dtrlstAuthor_CancelCommand(ByVal source As Object, ByVal e As System.Web.UI.WebControls.DataListCommandEventArgs)

```

```

Handles dtrlstAuthor.CancelCommand
lblMessage.Text = "<li> Cancel Item!" End Sub
Protected Sub dtrlstAuthor_DeleteCommand(ByVal source
As Object, ByVal e As
System.Web.UI.WebControls.DataListCommandEventArgs)
Handles dtrlstAuthor.DeleteCommand
lblMessage.Text = "<li> Delete Item!" End Sub
Protected Sub dtrlstAuthor_EditCommand(ByVal source As
Object, ByVal e As
System.Web.UI.WebControls.DataListCommandEventArgs)
Handles dtrlstAuthor.EditCommand
lblMessage.Text &= "<li> Edit item!" End Sub
Protected Sub dtrlstAuthor_ItemCommand(ByVal source As
Object, ByVal e As
System.Web.UI.WebControls.DataListCommandEventArgs)
Handles dtrlstAuthor.ItemCommand
lblMessage.Text = "<li> Item command!" End Sub
Protected Sub dtrlstAuthor_UpdateCommand(ByVal source
As Object, ByVal e As
System.Web.UI.WebControls.DataListCommandEventArgs)
Handles dtrlstAuthor.UpdateCommand
lblMessage.Text = "<li> Update Item!" End Sub
End Class

```

## NOTES

### Output:

- Item command!
- Edit item!

Rajeev

[Edit](#) [Delete](#) [Update](#) [Cancel](#) [Item](#)

Ranjan

[Edit](#) [Delete](#) [Update](#) [Cancel](#) [Item](#)

Somu

[Edit](#) [Delete](#) [Update](#) [Cancel](#) [Item](#)

Trideep

[Edit](#) [Delete](#) [Update](#) [Cancel](#) [Item](#)

### 41. Write a program to implement paging concept data grid and dataset.

```

Imports System.Data
Imports System.Data.SqlClient Partial Class _Default
Inherits System.Web.UI.Page Dim constr As String =
ConfigurationManager.ConnectionStrings
("DatabaseConnectionString1").Connection String
Protected Sub Page_Load(ByVal sender As Object, ByVale
As System.EventArgs) Handles Me.Load
If Not IsPostBack Then Binddatagrid()
End If End Sub
Sub Binddatagrid()

```

**NOTES**

```

Dim conpubs As SqlConnection Dim dtrProgram Title As
SqlDataAdapter Dim datset As DataSetconpubs = New
SqlConnection(constr)
dtrProgram Title = New SqlDataAdapter("select * from
Author", conpubs) datset = New
DataSetdtrProgramTitle.Fill(datset)
GridView1.DataSource = datset GridView1.DataBind()
End Sub

Protected Sub GridView1_PageIndexChanged(ByVal sender
As Object, ByVal e As System.EventArgs) Handles
GridView1.PageIndexChanged

End Sub

Protected Sub GridView1_PageIndexChanging(ByVal sender
As Object, ByVal e As
System.Web.UI.WebControls.GridViewPageEventArgs)
Handles GridView1.PageIndexChanging
GridView1.PageIndex = e.NewPageIndexBinddatagrid()
End Sub

Protected Sub GridView1_SelectedIndexChanged(ByVal
sender As Object, ByVal e As System.EventArgs) Handles
GridView1.SelectedIndexChanged

End Sub

End Sub
End Class

```

**Output:**

Au_id	Au_fname	Au_lname	phone
01	Rajeev	Ranjan	453234
02	Ranjan	Jaiswal	565456

1 2 3

**After click on second page**

Au_id	Au_fname	Au_lname	phone
03	Somu	Kumar	565467
04	Trideep	sarna	565456

1 2 3

**42. Write a program to create an own table and bind data using data grid.**

```

Imports System.Data Partial Class _Default
Inherits System.Web.UI.Page
Function CreateDatasource() As ICollection Dim dt As
New DataTable()
Dim dr As DataRow

```



```

dt.Columns.Add(New DataColumn("integervalue",
GetType(Int32))) dt.Columns.Add(New
DataColumn("stringvalue", GetType(String)))
dt.Columns.Add(New DataColumn("currencyvalue",
GetType(Double))) Dim i As Integer
For i = 0 To 8
dr = dt.NewRow() dr(0) = i
dr(1) = "item" + i.ToStringdr(2) = 1.23 * (i + 1)
dt.Rows.Add(dr) Next i
Dim dv As New DataView(dt) Return dv End Function
Protected Sub Page_Load(ByVal sender As Object, ByVal e
As System.EventArgs) Handles Me.Load
If Not IsPostBack Then GridView1.DataSource =
CreateDatasource() GridView1.DataBind()
End If
End Sub
End Class

```

**NOTES****Output:**

integervalue	stringvalue	currencyvalue
0	item0	1.23
1	item1	2.46
2	item2	3.69
3	item3	4.92
4	item4	6.15
5	item5	7.38
6	item6	8.61
7	item7	9.84
8	item8	11.07

**Try Yourself:**

- (i) Develop an application to display student details in form.
- (ii) Develop an application to demonstrate bind data using data grid control to display student details in form.

**43. Write a program to demonstrate select operation using ADO.Net Code.**

We will work with the sample Emp table in Oracle.

**Retrieving Records:**

```

Imports System.Data.OleDb
Public Class Form1 Inherits System.Windows.Forms.Form
Dim myConnection As OleDbConnection
Dim myCommand As OleDbCommand
Dim dr As New OleDbDataReader()

```

**NOTES**

```

Private Sub Form1_Load(ByVal sender As System.Object,
ByVal e As
    System.EventArgs) _
Handles MyBase.Load
myConnection = New OleDbConnection_
("Provider=MSDAORA.1;UserID=scott;password=tiger;
database=ora")
'MSDORA is the provider when working with Oracle
Try
myConnection.Open()
'opening the connection
myCommand = New OleDbCommand("Select * from emp",
myConnection)
'executing the command and assigning it to connection
dr = myCommand.ExecuteReader()
While dr.Read()
'reading from the datareader

MessageBox.Show("EmpNo" & dr(0))
MessageBox.Show("ENAME" & dr(1))
MessageBox.Show("Job" & dr(2))
MessageBox.Show("Mgr" & dr(3))
MessageBox.Show("HireDate" & dr(4))
'displaying data from the table
End While

dr.Close()
myConnection.Close()
Catch e As Exception
End Try
End Sub
End Class

```

The above code displays first 5 columns from the Emp table in Oracle.

**44. Write a program to demonstrate insert operation using ADO.Net Code.**

Drag a Button from the toolbox onto the Form. When this Button is clicked the values specified in code will be inserted into the Emp table.

```

Imports System.Data.OleDb
Public Class Form2 Inherits System.Windows.Forms.Form
Dim myConnection As OleDbConnection
Dim myCommand As OleDbCommand

```

```

Dim ra as Integer
    `integer holds the number of records inserted
Private Sub Form2_Load(ByVal sender As System.Object,
ByVal e As_
System.EventArgs) Handles MyBase.Load
End Sub

Private Sub Button1_Click(ByVal sender As
System.Object, ByVal e As _
System.EventArgs) Handles Button1.Click
myConnection = New
OleDbConnection(Provider=MSDAORA.1;User_
ID=scott;password=tiger;database=ora")
Try

myConnection.Open() myCommand = New
OleDbCommand("Insert into emp values
12,'Ben','Salesman',300,_ 12-10-2001,3000,500,10 ",
myConnection)

`emp table has 8 columns. You can work only with the
columns you want ra=myCommand.ExecuteNonQuery()
MessageBox.Show("Records Inserted" & ra)
myConnection.Close()

Catch
End Try
End Sub
End Class

```

**45. Write a program to demonstrate delete operation using ADO.Net Code.**

Drag a Button on a new form and paste the following code.

```

Imports System.Data.OleDb
Public Class Form3 Inherits System.Windows.Forms.Form
Dim myConnection As OleDbConnection
Dim myCommand As OleDbCommand
Dim ra as Integer

Private Sub Form3_Load(ByVal sender As System.Object,
ByVal e As_
System.EventArgs) Handles MyBase.Load
End Sub

```

## NOTES

**NOTES**

```
Private Sub Button1_Click(ByVal sender As
System.Object, ByVal e_
As System.EventArgs) Handles Button1.Click
Try
```

**Try yourself:**

(i) Develop an application to display student details in for using ADO control.

**46. Write a program to demonstrate update operation using ADO.Net Code.**

Drag a Button on a new form and paste the following code.

```
Imports System.Data.OleDb
Public Class Form4 Inherits System.Windows.Forms.Form
Dim myConnection As OleDbConnection
Dim myCommand As OleDbCommand
Dim ra as Integer

Private Sub Form4_Load(ByVal sender As System.Object,
ByVal e As_ System.EventArgs) Handles MyBase.Load
End Sub

Private Sub Button1_Click(ByVal sender As
System.Object, ByVal e_ As System.EventArgs) Handles
Button1.Click
Try
myConnection = New
OleDbConnection(Provider=MSDAORA.1;User_
ID=scott;password=tiger;database=ora")
myConnection.Open()
myCommand = New OleDbCommand("Update emp Set DeptNo=65
where DeptNo=793410",_ myConnection)
ra=myCommand.ExecuteNonQuery()
MessageBox.Show("Records Updated" & ra)
myConnection.Close()
Catch
End Try
End Sub
End Class
```

## Crystal Reports

You can create a Crystal Report using three methods:

1. Manually i.e. from a blank document
2. Using Standard Report Expert
3. From an existing report Using Pull Method Creating Crystal Reports Manually.

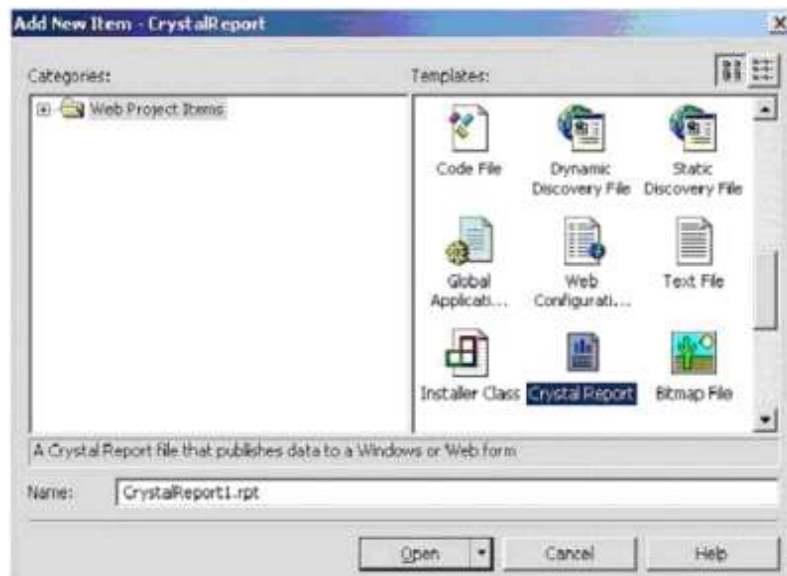
We would use the following steps to implement Crystal Reports using the Pull Model:

1. Create the .rpt file (from scratch) and set the necessary database connections using the Crystal Report Designer interface.
2. Place a CrystalReportViewer control from the toolbox on the .aspx page and set its properties to point to the .rpt file that we created in the previous step.

### 47. Write the steps for creating a Crystal Reports.

Following are the steps to create the report i.e. the .rpt file.

1. Add a new Crystal Report to the web form by right clicking on the “Solution Explorer”, selecting “Add” → “Add New Item” → “CrystalReport”.



## NOTES

2. On the “Crystal Report Gallery” pop up, select the “As a Blank Report” radio button and click “ok”.

## NOTES



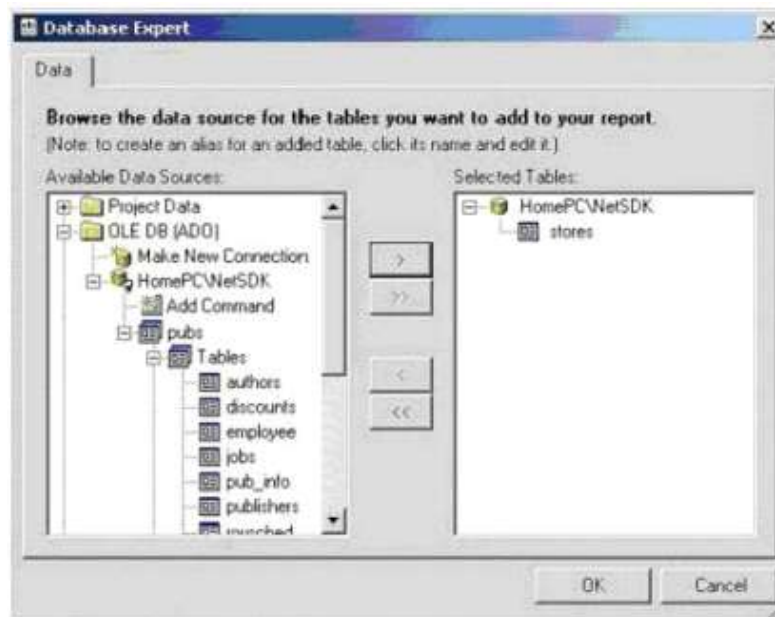
3. This should open up the Report File in the Crystal Report Designer.



4. Right click on the “Details Section” of the report, and select “Database” →”Add/Remove Database”.

5. In the “Database Expert” pop up window, expand the “OLE DB (ADO)” option by clicking the “+” sign, which should bring up another “OLE DB (ADO)” pop up.
6. In the “OLE DB (ADO)” pop up, Select “Microsoft OLE DB Provider for SQL Server” and click Next.
7. Specify the connection information.
8. Click “Next” and then click “Finish”.
9. Now you should be able to see the Database Expert showing the table that have been selected.
10. Expand the “Pubs” database, expand the “Tables”, select the “Stores” table and click on “>” to include it into the “Selected Tables” section.

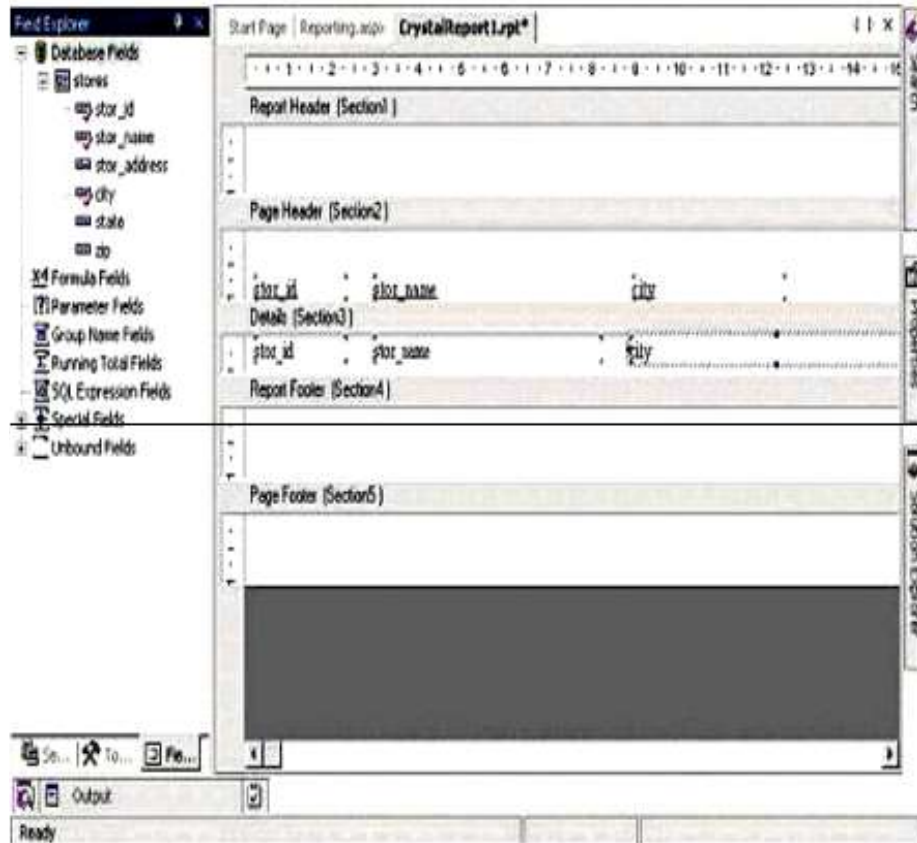
## NOTES



11. Now the Field Explorer should show you the selected table and its fields under the “Database Fields” section, in the left window.
12. Drag and drop the required fields into the “Details” section of the report. The field names would automatically appear in the “Page Header” section of the report. If you want to modify the header text then right click on the text of the “Page Header” section, select “Edit Text Object” option and edit it.

13. Save it.

**NOTES**



**Try Yourself:**

- (i) Generate a crystal report to display students data.

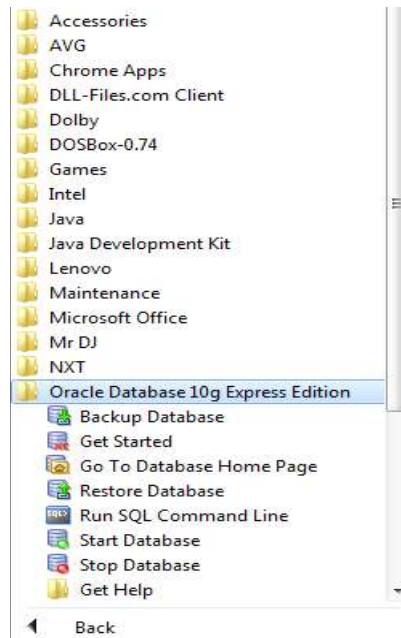
**Getting Started with SQL**

To work with SQL \*Plus Oracle (in our case) should to be installed on computer system. The following steps are required to follow to invoke SQL plus:

1. Click on Start button
2. Point on All Programs

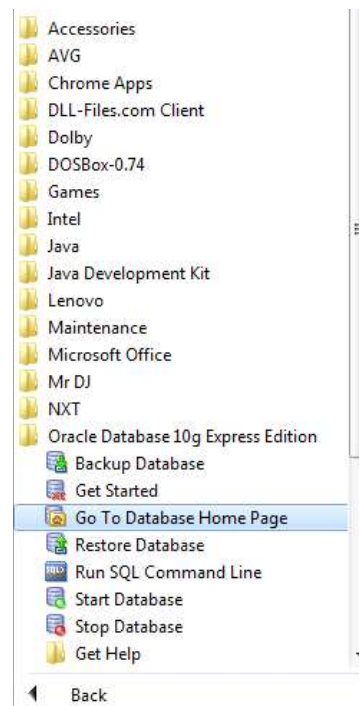


### 3. Point on Oracle Database 10g Express Edition



### NOTES

### 4. Click on Go to Database Home Page

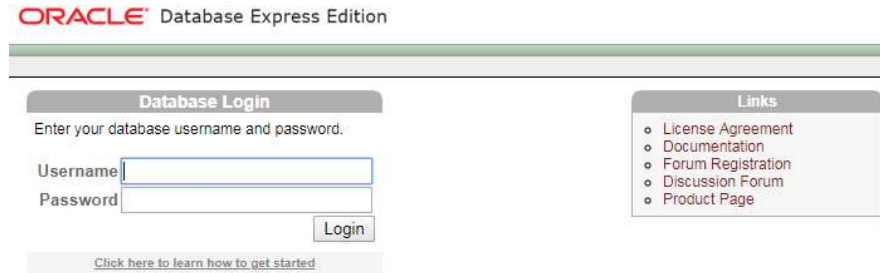


The following Screen given below will appear:

**Note:** Oracle user name and password may be different and need to be verified in lab. In this manual User name is Demo and Password in Demo.

1. Enter the User Name **Demo**, password **Demo** (Consult to your Lab Instructor for user name and password)
2. Click on “Login” button.

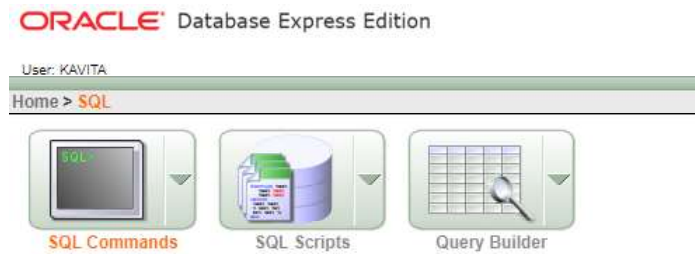
**NOTES**



Enter the User name and Password as created during installation. The following screen will appear. In this screen, click on **SQL**.



After clicking on SQL following screen will appear. Click on **SQL Command** to go to SQL command window.



After clicking on **SQL Command** following command screen will appear, where we can type and run all SQL commands.

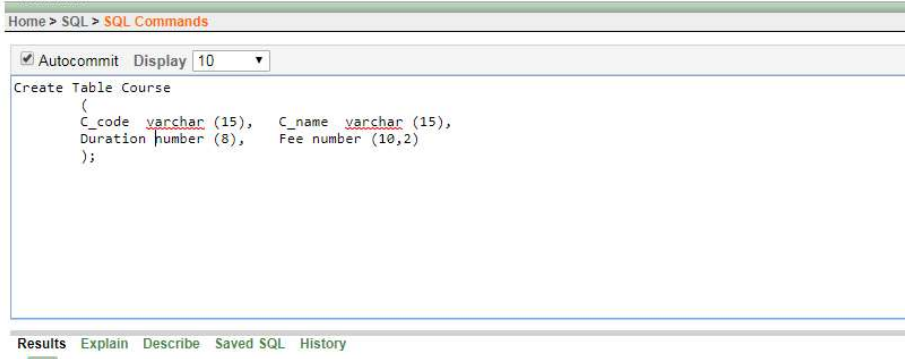


Enter SQL statement or PL/SQL command and click Run to see the results.

## SQL Queries:

### Data Definition Language (DDL)

#### Creating a Table

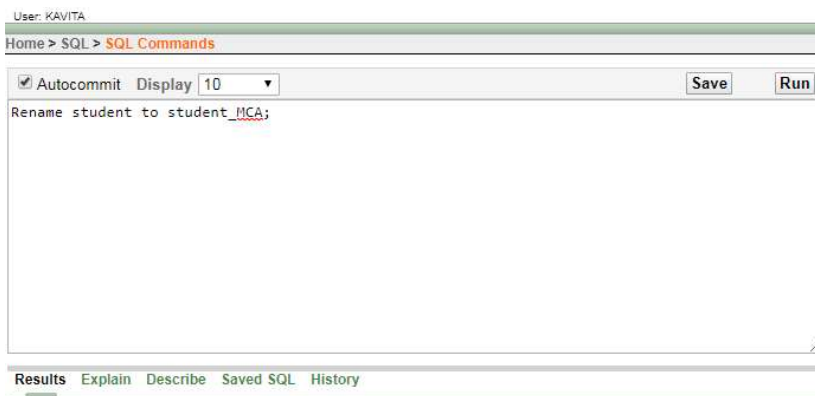


```

Home > SQL > SQL Commands
Autocommit Display 10
Create Table Course
(
  C_code varchar (15),  C_name varchar (15),
  Duration number (8),  Fee number (10,2)
);
Results Explain Describe Saved SQL History
    
```

Table created.

#### Renaming Tables

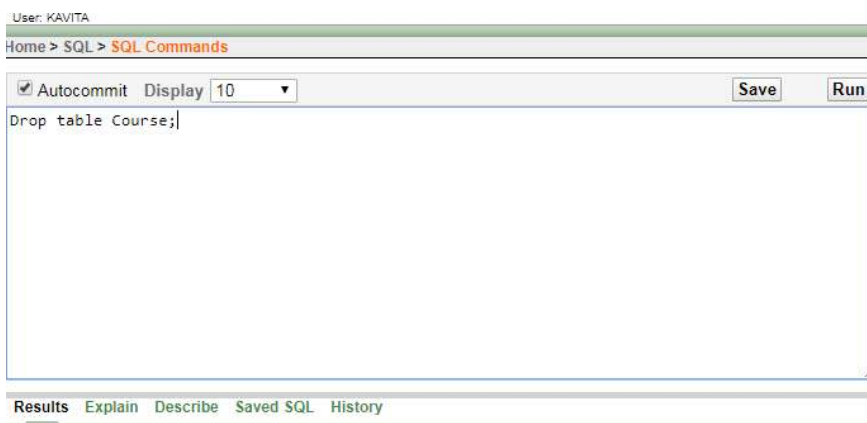


```

User: KAVITA
Home > SQL > SQL Commands
Autocommit Display 10 Save Run
Rename student to student_MCA;
Results Explain Describe Saved SQL History
    
```

Statement processed.

#### Dropping a table



```

User: KAVITA
Home > SQL > SQL Commands
Autocommit Display 10 Save Run
Drop table Course;
Results Explain Describe Saved SQL History
    
```

Table dropped.

## NOTES

## Table description

### NOTES

Home > SQL > **SQL Commands**

Autocommit Display 10 Save Run

Describe Course

---

Results Explain **Describe** Saved SQL History

Object Type **TABLE** Object **COURSE**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
COURSE	C_CODE	Varchar2	15	-	-	-	✓	-	-
	C_NAME	Varchar2	15	-	-	-	✓	-	-
	DURATION	Number	-	8	0	-	✓	-	-
	FEE	Number	-	10	2	-	✓	-	-
									1-4

## Modifying table

### Add a New Column:

ORACLE Database Express Edition

User: KAVITA Home Logout Help

Home > SQL > **SQL Commands**

Autocommit Display 10 Save Run

Alter table student add (mobile Number (10)) ;

---

Results Explain Describe Saved SQL History

Table altered.

0.03 seconds

The above command will add a new column mobile in student table. You could see the new structure of student table as shown below:

The screenshot shows the Oracle SQL Developer interface. The SQL Command window contains the command: `desc student;`. Below the command window, the 'Describe' tab is active, displaying the structure of the **STUDENT** table. The table has five columns: ROLL\_NO, NAME, ADDRESS, C\_CODE, and MOBILE. The MOBILE column is a Number type with a length of 10 and a scale of 0.

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
STUDENT	ROLL_NO	Varchar2	10	-	-	-	✓	-	-
	NAME	Varchar2	20	-	-	-	✓	-	-
	ADDRESS	Varchar2	30	-	-	-	✓	-	-
	C_CODE	Varchar2	8	-	-	-	✓	-	-
	MOBILE	Number	-	10	0	-	✓	-	-

## NOTES

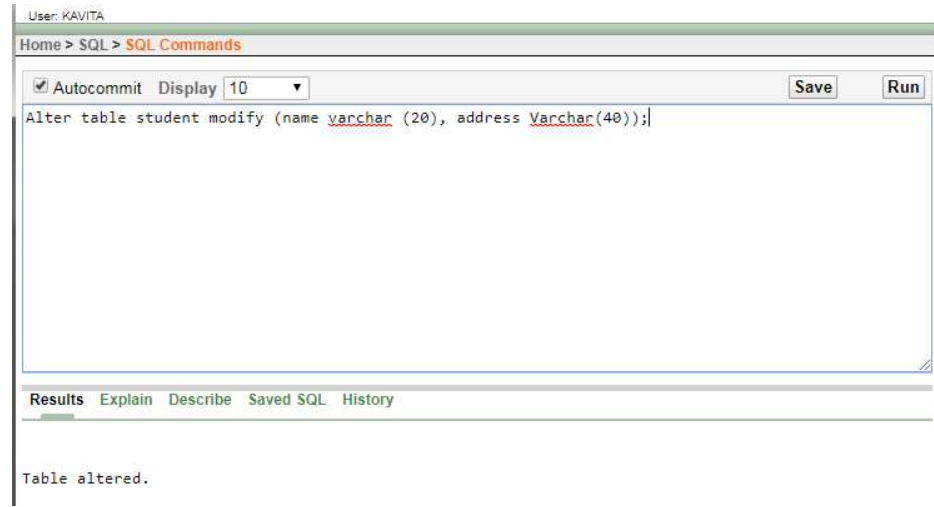
### Changing data type of an existing column

The screenshot shows the Oracle Database Express Edition interface. The SQL Command window contains the command: `Alter table course modify c_code char (15);`. Below the command window, the 'Results' tab is active, displaying the message: `Table altered.`

The above command will change the data type of c\_code field from varchar to char.

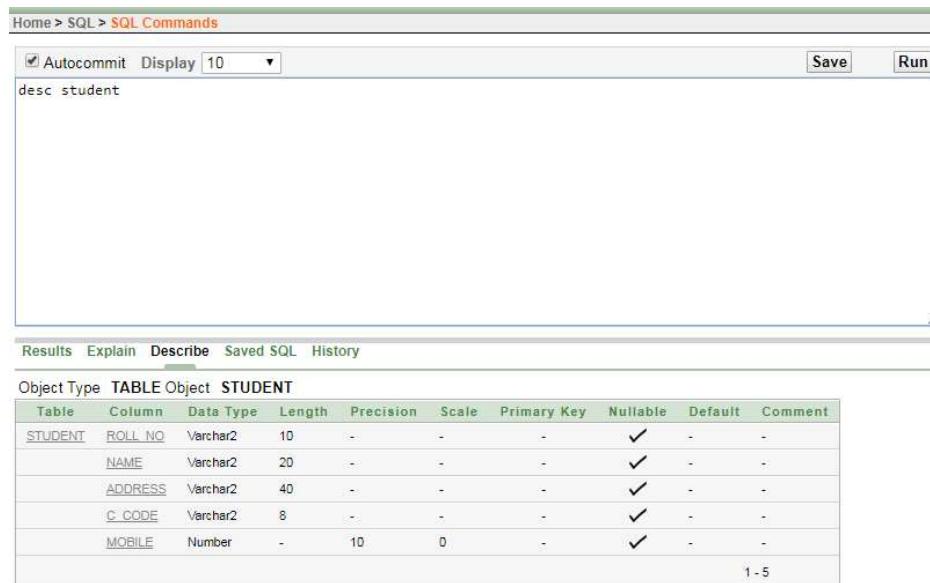
**Modifying the length of on existing column**

**NOTES**



The above command will change the length of name column from 15 to 20 and address from 35 to 40.

After altering student table structure will look like:



## Deleting any Column

The screenshot shows the Oracle Database Express Edition interface. At the top, it says "ORACLE Database Express Edition" and "User: KAVITA". There are navigation buttons for Home, Logout, and Help. Below the header, the breadcrumb "Home > SQL > SQL Commands" is visible. The main area contains a text input field with the SQL command "Alter table student drop column mobile;". Above the input field are checkboxes for "Autocommit" (checked) and a "Display" dropdown set to "10". To the right of the input field are "Save" and "Run" buttons. Below the input field, there are tabs for "Results", "Explain", "Describe", "Saved SQL", and "History".

Table dropped.

## Data Manipulation Language (DML)

The data manipulation language statements are INSERT, DELETE, and UPDATE

### Insert Records in Table

The screenshot shows the Oracle Database Express Edition interface. At the top, it says "User: KAVITA". Below the header, the breadcrumb "Home > SQL > SQL Commands" is visible. The main area contains a text input field with the SQL command "Insert into course values ('PG001', 'MCA', 3,32000.00)". Above the input field are checkboxes for "Autocommit" (checked) and a "Display" dropdown set to "10". To the right of the input field are "Save" and "Run" buttons. Below the input field, there are tabs for "Results", "Explain", "Describe", "Saved SQL", and "History".

### Output:

The screenshot shows the Oracle Database Express Edition Results window. At the top, there are tabs for "Results", "Explain", "Describe", "Saved SQL", and "History". Below the tabs, the text "1 row(s) inserted." is displayed. Below that, the text "0.08 seconds" is displayed. At the bottom of the window, there is a footer with "Application Express 2.1.0.00.39" and "Copyright © 1999, 2006, Oracle. All rights reserved." and "Language: en-us".

## NOTES

**Try Yourself:**

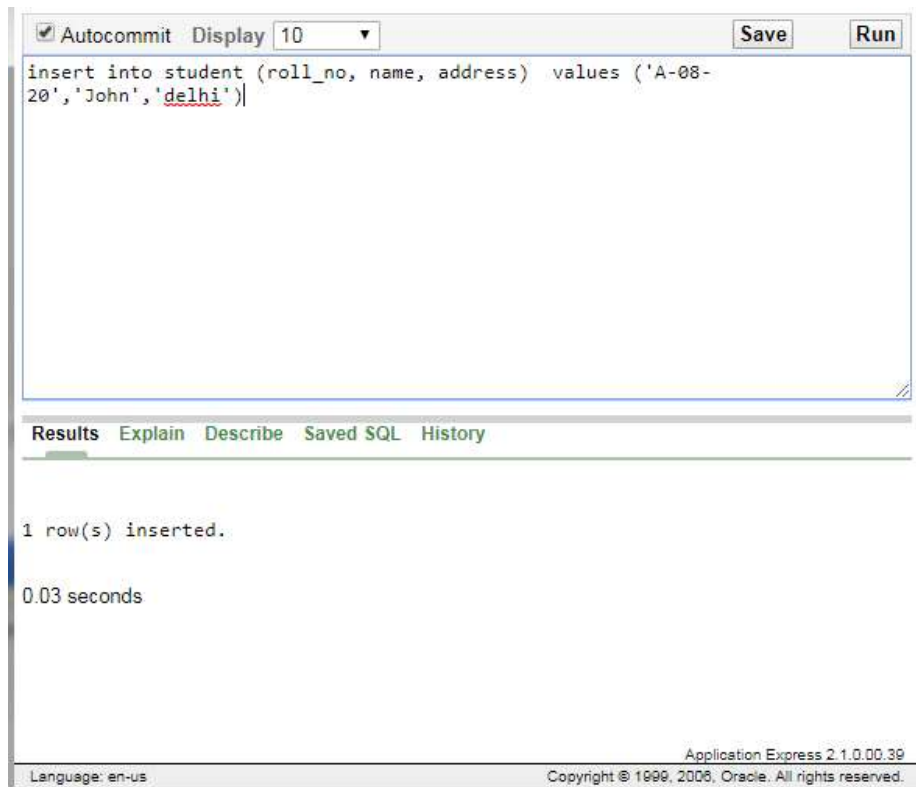
1. Add five records in course table
2. Create a new table **Book** with the following fields and data types.

**NOTES**

Field Name	Data Type	Size
B_Code	varchar	15
Title	varchar	30
Author	varchar	15
Price	Number	6,2

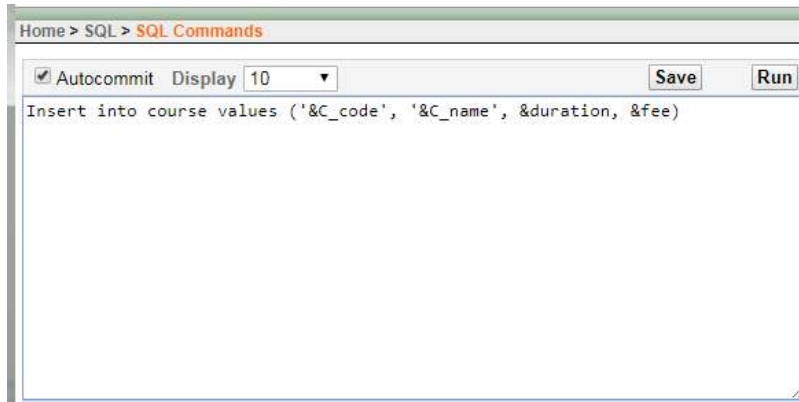
3. View the structure of Book table.
4. Add five records in Book table.

**Insert Data into Specific Fields**



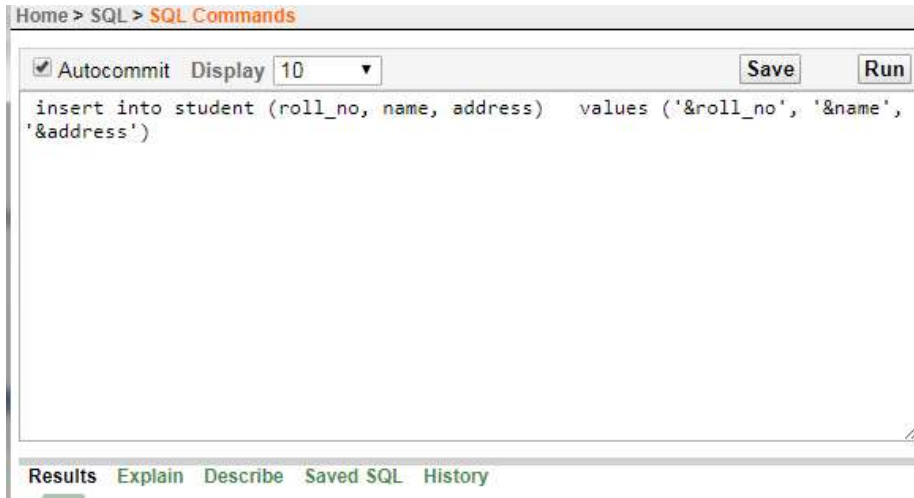


## Insert Data with User Interaction



## NOTES

To insert more record the same command could be repeated by putting / and pressing enter key at SQL prompt.



### Try Yourself:

1. Add the following data into C\_code, C\_name and duration fields of Course table.

C_code	C_name	Duration
UG001	BCA	3
UG002	B Sc-IT	3
PG003	M Sc-IT	2
2. Add three 10 records into student table with the user interaction.
3. Add data into b\_code, title, and author fields of book table with the user interaction.

## Display Table Records

*To view all the columns*

### NOTES

Home > SQL > SQL Commands

Autocommit Display 10 Save Run

```
Select * from course
```

Results Explain Describe Saved SQL History

C_CODE	C_NAME	DURATION	FEE
PG007	M Sc-CS	3	32000
UG001	BCA	3	29000
UG002	B SC-IT	3	25000
PG002	MBA	2	40000
PG003	M Sc-IT	3	32000
PG001	MCA	3	32000

6 rows returned in 0.07 seconds [CSV Export](#)

Application Express 2.1.0.00.39

*To view selective columns*

User: KAVITA

Home > SQL > SQL Commands

Autocommit Display 10 Save Run

```
Select c_name, fee from course
```

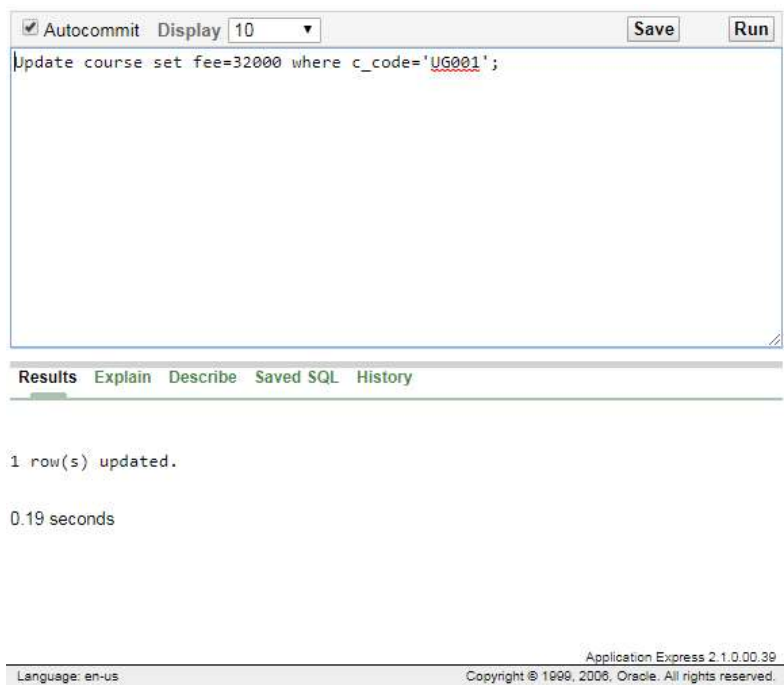
Results Explain Describe Saved SQL History

C_NAME	FEE
M Sc-CS	32000
BCA	29000
B SC-IT	25000
MBA	40000
M Sc-IT	32000
MCA	32000

6 rows returned in 0.02 seconds [CSV Export](#)

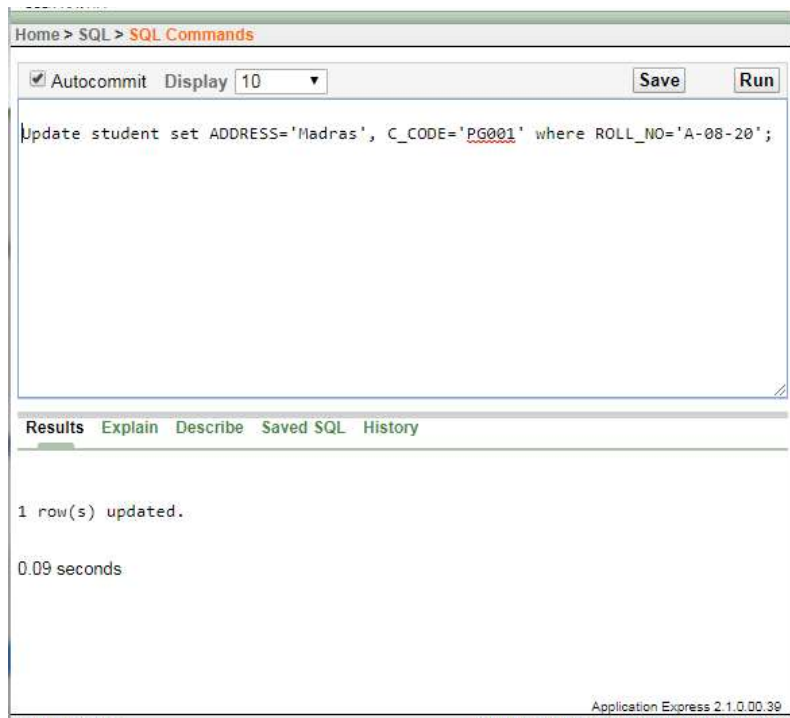
## Update Table Records

The *Example* for Update:



## NOTES

The *Example* to update multiple columns:



**NOTES****Try Yourself:**

1. Display name and c\_code of students.
2. Change the address from Madras to Delhi of student whose roll number is A-08-20.
3. Change the fee from Rs. 32000 to Rs. 38000 of course where c\_code is PG001.

**Data Control Language (DCL)**

Data Control Language are the commands that allow authorized database users to share the data with other users. The shared data could be accessed or manipulated by other users as per the permission granted to those users.

The data manipulation language statements are GRANT and REVOKE

- **GRANT**- It gives user's access privileges to database.
- **REVOKE**-withdraw user's access privileges given by using the GRANT command.

**AGGREGATE FUNCTIONS**

Aggregate functions work on a group of values (a column values) and returns a single value.

Few aggregate functions are listed below:

- SUM()
- MAX()
- MIN()
- AVG()
- COUNT()

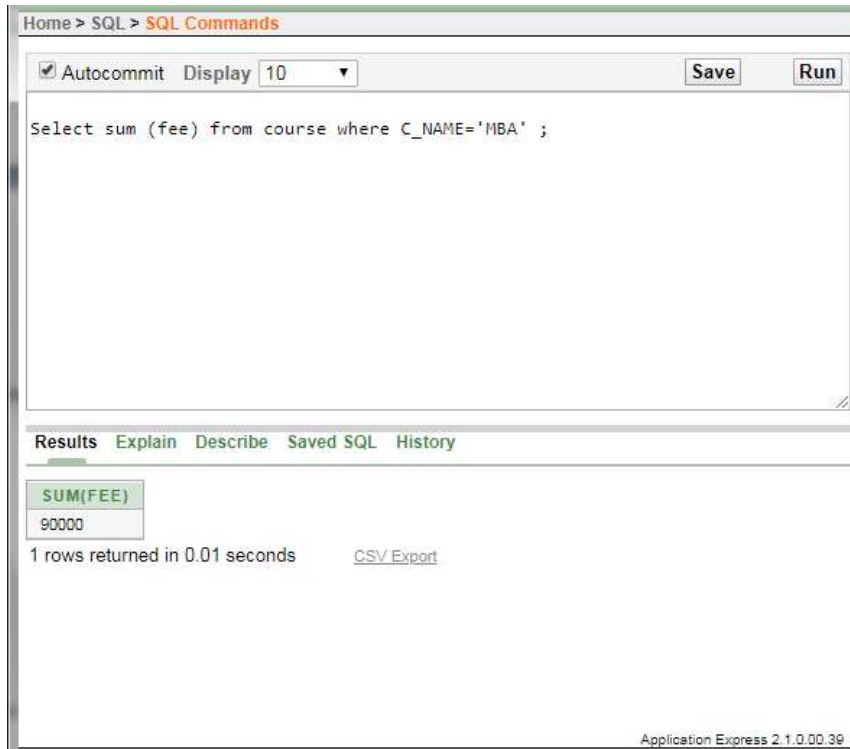
Let suppose we have a table Book with the following records.

*Table: Book*

C_CODE	C_NAME	DURATION	FEE
PG002	MBA	2	40000
PG006	MBA	2	50000
PG007	M Sc-CS	3	32000
UG001	BCA	3	32000
UG002	B SC-IT	3	25000
PG003	M Sc-IT	3	32000
PG001	MCA	3	32000

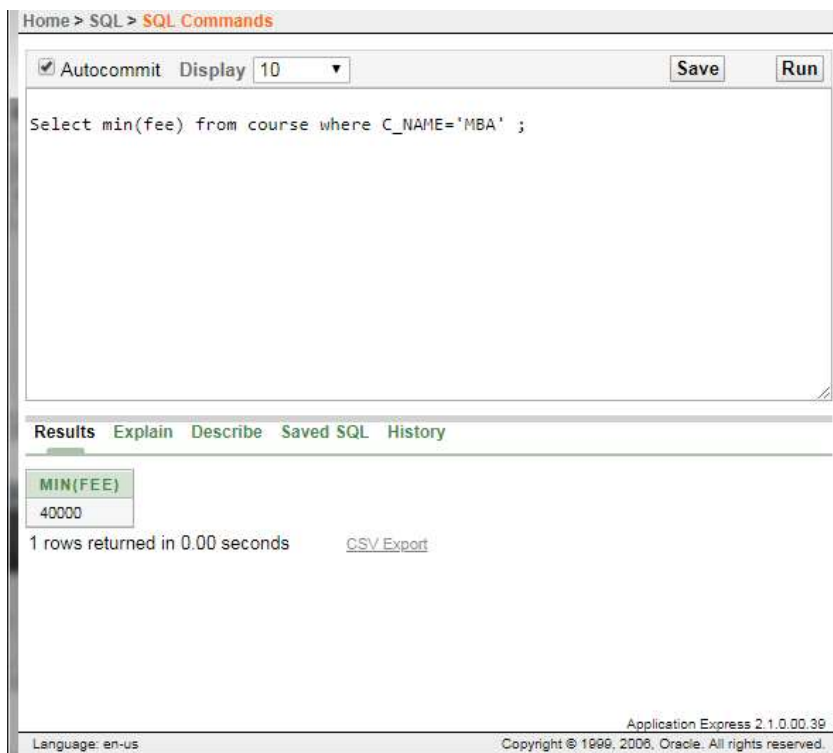
### Sum ()

To see the sum of fee where the course name is MBA, SQL query is as follows:



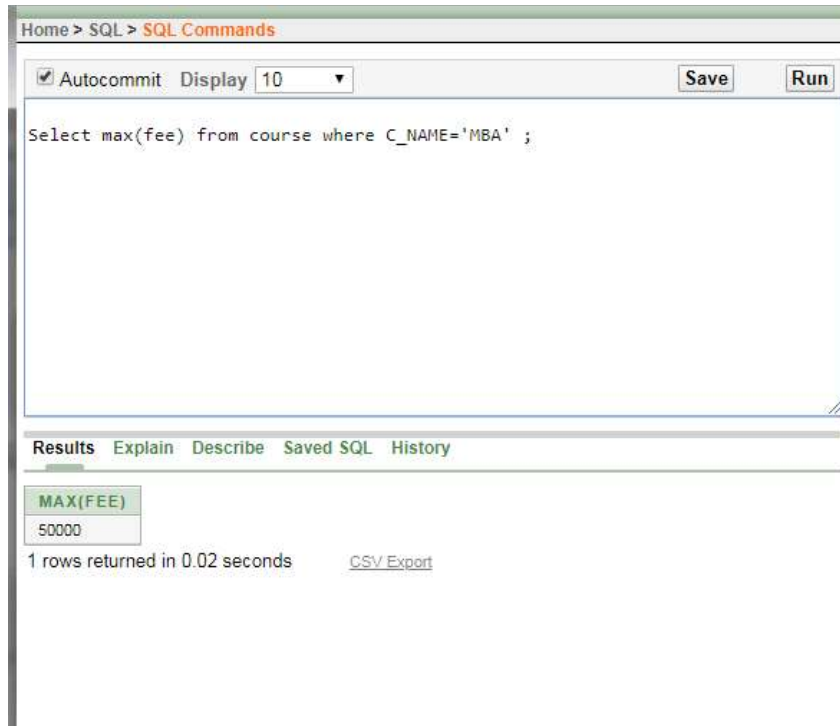
### NOTES

### Min ()



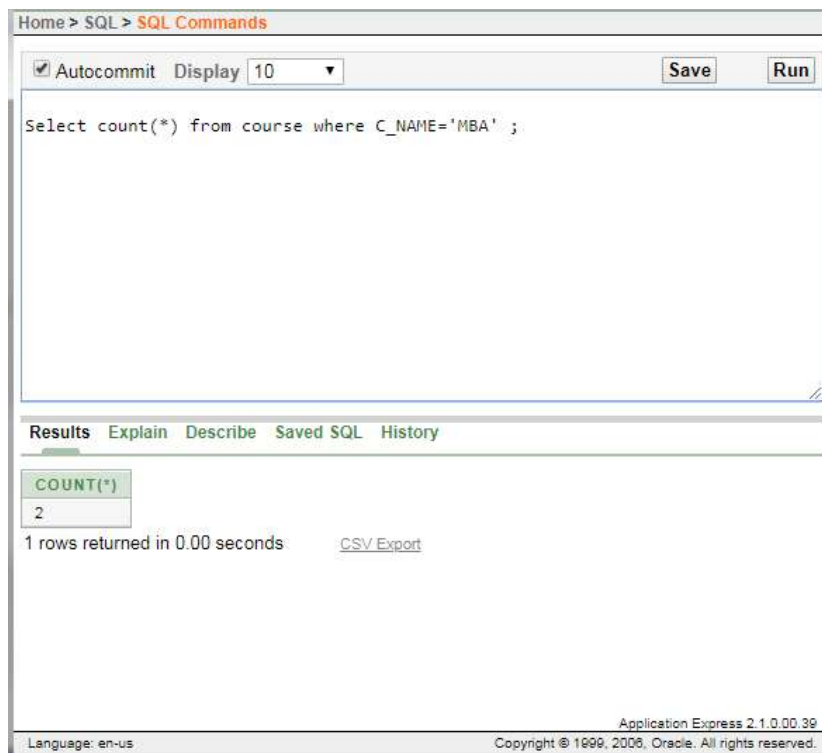
**Max ()**

**NOTES**



**Count (\*)**

To see the number of records in the table where course name is MBA, SQL query is as follows:



## LIBRARY INFORMATION SYSTEM

VB.NET & RDBMS Lab

Book Table for Keeping Track of Books

Field	Data type	Default	Key	Extra
Code	INT(11)	Not Null	Primary	Auto increment
Bookname	VARCHAR(255)	Null		
Author	VARCHAR(255)	Null		
Publication	VARCHAR(255)	Null		
Subject	VARCHAR(255)	Null		
No of copies	INT(10)	Null		

Student Table for Student Information

Field	Data type	Default	Key	Extra
libid	INT(11)	NOT NULL	Primary key	Auto increment
regno	INT(10)	NULL		
branch	VARCHAR(255)	NULL		
section	VARCHAR(255)	NULL		
semester	VARCHAR(255)	NULL		
section	VARCHAR(2)	NULL		
yearofadm	INT(5)	NULL		

Teacher Table to Keep Teacher Information

Field	Data Type	Default	Key	Extra
Tid	INT(11)	NOT NULL	Primary key	Auto increment
Name	VARCHAR(255)	NULL		
Designation	VARCHAR(255)	NULL		
Branch	VARCHAR(255)	NULL		
Contactno	INT(13)	NULL		
Lectures	LONG BLOB	NULL		

Issue Table to Keep Track of Books Issued

Field	Data Type	Default	Key	Extra
bookid	INT(11)	NOT NULL	Foreign key	References book
stuid	INT(11)	NOT NULL	Foreign key	References Student
issuedate	DATE	NULL		
returndate	DATE	NULL		

## NOTES

**NOTES**

Student Login Table

Field	Data type	Default	Key	Extra
logid	INT(11)	NOT NULL	Foreign key	References Student
Username	VARCHAR(255)	NULL		
Password	VARCHAR(255)	NULL		
numbooks	INT(1)	NULL		

Event Table for Event Information

Field	Data type	Default	Key	Extra
Name	Varchar(255)	NULL		
Date	Date(yyyy/mm/dd)	NULL		
Time	VARCHAR(255)	NULL		
Mname	VARCHAR(255)	NULL		
Contactno.	Int(30)	NULL		
Email	VARCHAR(255)	NULL		
Venue	varchar(255)	NULL		

Teacher Login Table

Field	Data Type	Default	Key	Extra
Loginid	INT(11)	NOT NULL	Foreign key	References teacher
Username	VARCHAR(255)	NULL		
Password	VARCHAR(255)	NULL		

**Student marks sheet processing**

Table Name: Login

Field Name	Data Type	Length	Constraints
Username	Varchar	13	Not null
Password	Varchar	13	Not null



Table Name: Student

Field Name	Data Type	Length	Constraints
Reg_no	Int	11	Primary key
Name	Varchar	20	Not null
Year_of_admission	Varchar	21	Not null
College	Varchar	40	Not null
Regular	Varchar	10	Not null
Course	Varchar	10	Not null
Branch	Varchar	20	Not null

**NOTES**

Table Name: Attendance

Field Name	Data Type	Length	Constraints
Staff_id	Int	11	Primary key, foreign key
Date	Date	10	Primary key, foreign key
attendance	int	10	Not null

Table Name: Syllabus

Field Name	Data Type	Length	Constraints
Course	Varchar	20	Not null
Branch	Varchar	41	Not null
Semester	Varchar	20	Not null
Year	Varchar	10	Primary key
Code_Subject	Varchar	40	Primary key
Credits	Varchar	40	Not null

Table Name: Marks

**NOTES**

Field Name	Data Type	Length	Constraints
Reg_no	int	11	Primary key
Code_Subject	varchar	40	Primary key
Attended_Exam	varchar	10	Not null
Internal_mark	double	10	Not null
First_valuation_mark	double	10	Not null
first_evaluators_id	int	10	Not null
Second_valuation_mark	double	20	Not null
Third_valuation_mark	double	10	Not null
Third_evaluators_id	int	20	Not null
External_mark	double	10	Not null
Final_mark	double	10	Not null
Semester	Varchar	20	Not null
moderation_given	char	3	Not null

Table Name: Camp\_Details

VB.NET & RDBMS Lab

For entering the details of the second valuation camp. Date\_of\_begin, Branch :  
Primary key

Field Name	Data Type	Length	Constraints
Staff_name	varchar	30	Not null
Staff_id	Int	11	Foreign key
Informed_not	varchar	10	Not null
Date_of_begin	date	10	Primary key
Subject_to_value	varchar	20	Not null
End_date	Date	10	Not null
Summary	varchar	200	Not null
Branch	varchar	10	Primary key
Semester	varchar	10	Not null
attendance	int	10	Not null

Table Name: Hall ticket

Field Name	Data Type	Length	Constraints
Reg_no	Int	11	Primary key
Attendance	Double	11	Not null
Fee_paid	Varchar	10	Not null
Condonation_paid	Varchar	10	Not null
Semester	Varchar	23	Not null
Year	Varchar	10	Not null

## NOTES

Table Name: Passing Board

**NOTES**

Field Name	Data Type	Length	Constraints
members_id	Int	10	Foreign key
Date	date	11	Primary key
Sem	Int	10	Not null
Branch	varchar	10	Primary key
Summary	mediumtext	200	Not null
max_marks_student	float	10	Not null
subject1	varchar	20	Not null
mark1	float	10	Not null
special_sub	varchar	20	Not null
special_marks	float	10	Not null

Table Name: Remuneration

Field Name	Data Type	Length	Constraints
Staff_id	Int	11	Primary key
Camp_attendence	float	11	Not null
no_of_papers	Int	11	Not null
payment_paid	varchar	10	Not null
Date	date	10	Not null

Table Name: Staff

Field Name	Data Type	Length	Constraints
Staff_id	Int	11	Primary key
Staff_name	Varchar	11	Not null
SCollege	Varchar	11	Not null
Designation	Varchar	10	Not null
Email_id	Varchar	10	Not null
Experience	Int	10	Not null

**NOTES**

**Telephone Directory**

Customer Table

S.No.	Field Name	Data Type
1	Id	Number
2	Name	Text
3	Address	Text
4	Provider	Text

Service Provider Table

S.No.	Field Name	Data Type
1	Id	Number
2	Plan	Number
3	Issue_date	Date/time
4	Due_date	Date/time
5	Providername	Text

Directory Table

S.No.	Field Name	Data Type
1	add new	Number
2	Group	Number
3	Move	Number
4	Copy	Number

**GAS BOOKING AND DELIVERING****Table Design Of Online Gas Booking Management System**

## Price Details

## NOTES

Column Name	Data Type	Remark
Cylntype	Varchar (10)	Cylinder Type
price	float (4,2)	Price

## Billing Details

Column Name	Data Type	Remark
Blid	Integer	Billing Id
Consno	Integer	Customer Number
Cylintype	Varchar (10)	Cylinder Type
bkdate	Datetime	Booked Date
deldate	Datetime	Delivery Date
amt	Float (4,2)	Amount

## Customer Details

Column Name	Data Type	Remark
Consno	Integer	Customer Number
Pass	Varchar (8)	Password
Cylintype	Varchar (10)	Cylinder Type
dtofconn	Datetime	Connection Date
Cname	Varchar (25)	Customer Name
Caddr	Varchar (50)	Customer address
City	Varchar(15)	Customer city
Phone	Number (10)	Phone number
Pincode	Number (6)	Pin code

## Stock Details

Column Name	Data Type	Remark
Sdt	Datetime	Date
Storeccom	Integer	Stock received Commercial
Storecdom	Integer	Stock received Domestic
Defcomm	Integer	Defective in Commercial
Defdom	Integer	Defective in Domestic
Cstocom	Integer	Current stock commercial
Cstodom	Integer	Current stock domestic

## Transaction Details

Column Name	Data Type	Remark
TId	Integer	Transaction Id
Consno	Integer	Customer Id
Cylntype	Varchar(10)	Cylinder Type
Bkdt	Datetime	Customer Name
Deldt	Datetime	Customer address
Status	Varchar(10)	Status of transaction

## ELECTRICITY BILL PROCESSING

Table Name: BILL

Field Name	Data Type	Size	Constraint
Bill_no	Number	Auto	Primary key
BillDate	Date		
Amount	Number	Auto	
City	Text	30	

Table Name: Customer

Field Name	Data Type	Size	Constraint
Customer First Name	Text	10	
Customer Middle Name	Text	10	
Customer Last Name	Text	10	
Cust_id	Number	Auto	Primary Key
Service_Centre	Text	20	
Street	Text	20	
City	Text	20	
City Code	Number	Auto	
State	Text	20	

Service Centre Table

Field Name	Data Type	Size	Constraint
Cust_id	Number	Auto	Foreign Key
Service_Centre	Text	20	
Street	Text	20	
City	Text	20	
City Code	Number	Auto	
State	Text	10	

Bill Charge Table

Field Name	Data Type	Size	Constraint
Bill_no	Number	Auto	Primary Key
Fix_charge	Number	Auto	
E_duty	Number	Auto	
Total Charge	Number	Auto	
Total Unit	Number	Auto	

Commercial Bill Table

Field Name	Data Type	Size	Constraint
Customer Name	Text	20	
Shop_no	Text	20	Primary Key
Street	Text	20	
Bill_no	Integer	Auto	Foreign Key
Bill Amount	Integer	Auto	
Bill Date	Date		

## NOTES

**BANK TRANSACTION**

Table Name: LOGIN

Description: This table is used to store Login details.

**NOTES**

Key	Field Name	Data Type	Length
PK	ACCOUNTNO	VARCHAR	12
	PASSWORD	VARCHAR	45

Table Name: CLIENTS

Description: This table is used to store customer details.

Key	Field Name	Data Type	Length
	NAME	VARCHAR	45
	SURNAME	VARCHAR	45
	INITIAL	VARCHAR	10
	ACCOUNTTYPE	VARCHAR	45
	SEX	VARCHAR	6
	D.O.B	DATE	
	ADDRESS	VARCHAR	200
	MOBILENO	VARCHAR	10
	TELEPHONENO	VARCHAR	10
	EMAIL	VARCHAR	45
PK	ID_PASSPORT	VARCHAR	45

Table Name: ACCOUNT

Description: This table is used to store account details.

Key	Field Name	Data Type	Length
FK	ACCOUNTNO	VARCHAR	12
	ACCOUNTTYPE	VARCHAR	45
	ACCOUNTHOLDER	VARCHAR	45
	DATEOPENED	DATE	
	BRANCHCODE	INT	5
	DATEAPPROVED	DATE	
	ACCOUNTBALANCE	DECIMAL	
	APPROVED	VARCHAR	6
	DISAPPROVED	VARCHAR	6



Table Name: TRANSACTION

Description: This table is used to store the transaction details

Key	Field Name	Data Type	Length
FK	ACCOUNTNO	VARCHAR	12
	TRANSACTIONID	INT	
	TYPEOFTRANSACTION	VARCHAR	45
	TRANSACTIONDATE	DATETIME	
	REFERENCE	VARCHAR	45

**NOTES**

**PAYROLL PROCESSING**

User Table:

user_id(int) – Primary Key	Id for the user.
user_name(varchar)	Enter the name of the user.
password(varchar)	Enter the password of the user.
email_id(varchar)	Enter the email_id of the user.
usertype(varchar)	Enter the type of user.

Department Table:

dept_id(int) – Primary key	Id of the Department.
dept_name(varchar)	Name of the Department.

Grade Table:

grade_id(int) – Primary Key	Id of the pay grade.
grade_name(varchar)	Name of the pay grade.
grade_short_name(varchar)	Short name of the pay grade.
grade_basic(int)	Enter the basic amount.
grade_ta(int)	The amount of the Travel Allowance.
grade_da(int)	The amount of the Dearness Allowance.
grade_hra(int)	The amount of the House Rent Allowance.
grade_ma(int)	The amount of Medical Allowance.
grade_bonus(int)	The amount of bonus received.
grade_pf(int)	Amount of Provident Fund to be deducted.
grade_pt(int)	Amount of Professional Tax to be deducted.

## Employee Table:

## NOTES

emp_id(int) – Primary Key	Id of the employee.
emp_title(varchar)	Enter the title of employee.
emp_name(varchar)	Enter the name of employee.
emp_dob(date)	Enter the date of birth of employee.
emp_doj(date)	Enter the date of join of employee.
emp_address(varchar)	Enter the address of the employee.
emp_city(varchar)	Enter the city of the employee.
emp_pincode(int)	Enter the pincode of the employee.
emp_mobile_no(int)	Enter the mobile number of the employee.
emp_state(varchar)	Enter the state of the employee.
emp_mail_id(varchar)	Enter the mail id of the employee.
emp_pan_no(varchar)	Enter the Pan number of the employee.
emp_upload_pan()	Enter the pan card image of the employee.

## Employee Grade Details:

transaction_id(int)- Primary Key	Unique transaction id.
emp_id(int)	Employee id of employee.
emp_dept_id(int)	Department Id of employee.
emp_grade_id(int)	Grade Id of employee.
emp_from_date(date)	Date of join of employee in a department.
emp_to_date(varchar)	Last date of an employee in a department.

## Employee Salary Details:

transaction_id(int) -Primary Key	Unique primary key.
emp_id(int)	Employee Id of employee.
emp_salary_month(varchar)	Employee Salary month.
emp_salary_year(varchar)	Employee salary year.
emp_salary_eimbursment_date(datetime)	The date and time when employee salary was generated.
emp_dept_id(int)	department Id of the employee.
emp_grade_id(int)	grade id of the employee

emp_basic(int)	Enter the amount of the basic.
emp_da(int)	The amount of dearness Allowance.
emp_ta(int)	The amount of travel allowance.
emp_hra(int)	The amount of House Rent Allowance.
emp_ma(int)	The amount of Medical Allowance.
emp_bonus(int)	The amount of Bonus.
emp_pf(int)	The amount of Provident Fund to be deducted.
emp_pt(int)	The amount of Professional Tax to be deducted.
emp_gross(int)	The gross total received by employee.
emp_total_salary(int)	The total salary received after deduction.

**NOTES****PERSONAL INFORMATION SYSTEM**

Question database and conducting quiz and personal diary

Student Table:

<u>s_id</u>	sname	dept	pass	e-mail
02201048	palash	cse	123	<a href="mailto:Msap71@yahoo.com">Msap71@yahoo.com</a>
02201018	saikot	cse	012	<a href="mailto:Saikot1@gmail.com">Saikot1@gmail.com</a>
02201070	rumi	cs	324	<a href="mailto:Rumi12@gmail.com">Rumi12@gmail.com</a>
02201154	nasim	cse	786	<a href="mailto:Nasim1@hotmail.com">Nasim1@hotmail.com</a>

Teacher Table

<u>T_id</u>	tname	dept	pass	e-mail
012	Dr. karim rahman	cse	0012	<a href="mailto:karim12@yahoo.com">karim12@yahoo.com</a>
009	Abdur rouf sarkar	cse	abc	<a href="mailto:sarkar@gmail.com">sarkar@gmail.com</a>
003	Dr.Rahim	ess	236	<a href="mailto:rahim@gmail.com">rahim@gmail.com</a>

Course Table

<u>c_id</u>	Cname	T_id
Cse420	compiler	012
Cse310	Java programming	009
Cse474	Simulation & modeling	003

Schedule Quiz Table

<u>g_num</u>	Tques	tmark	stime	etime	sdate	edate	c_id	T_id
1	10	100	12.00pm	12.30pm	3/6/07	5/6/07	Cse420	012
2	20	40	10.00am	10.20	10/5/07	15/5/07	Cse474	003
1	10	50	2.00pm	3.00pm	5/5/07	6/5/07	Cse310	009

Question Table:

qlevel	qnum	sub_id	question	Op1	Op2	Op3	Op4	ans	mark	q_num

**NOTES**

Admin Table:

a_id	aname	pass	e-mail
789	Mr.roton	mrt	<a href="mailto:Mrt12@gmail.com">Mrt12@gmail.com</a>
889	Mrs sharmin	mnb	<a href="mailto:Sha122@yahoo.com">Sha122@yahoo.com</a>
885	Mr. abul	ccz	Ant2000@yahoo.com

Registered Table:

s_id	c_id
Palash	Cse420
Soikot	Cse474
Nasim	Cse474

Quiz Info Table:

s_id	t_id	c_id	qnum	status
02201048	00010001	CSE391	1	yes
02201018	00010002	CSE421	2	Yes
02201154	00010001	CSE391	1	no