

**Name of Experiment:** Student Database

**Exp No:** DB1

**Background:** Student should have basic knowledge of C#.

**Summary:** DBMS is a necessary requirement for any Mobile Application. We need to store and retrieve certain information at any point of time. Though the cloud services are available in the eco-system, we even require to store data locally.  
SQL Server CE (for Mobile) is embedded with the mango version Windows Phone. Hence we don't need to look for explicit database server to manage the user data.

**Learning Objective:** To learn the steps for database handling in windows phone 7 (mango version) locally through this experiment. Each steps are clearly mentioned (i.e. CRUD) and explained properly in order to proper understanding of the scenario.

**Target Platforms:** This experiment is tested on Windows Phone Emulator and Nokia Lumia 800.

**Procedure:**

Step 1. Repeat steps [1-4] as in Experiment no MC1.[Refer Hello World]

Step 2. Add a reference for System.Data.Linq by making right click on the project and add this library to the project.

Step 3. Now, add a new class and named it Student.cs by making right click on the project.

Step 4. Open Student.cs class and add using System.Data.Linq.Mapping.

Step 5. Define the Table and its schema. [Refer Student.cs from Source Code Section]

Step 6. Now again add one more class and named it as StudentDataContext.cs

Step 7. Open StudentDataContext.cs and add using System.Data.Linq.

Step 8. Here just define the ModelView of the MVVM architecture.[Refer StudentDataContext.cs from Source Code Section]

Step 9. Now go back to the MainPage.xaml.cs and add using System.Data.Linq, using System, using System.Collections.Generic to the class.

Step 10. Define the connection string that is called at each time in order to communicate with the database as @"isostore:/StudentDB.sdf". [It is a standard way to define the connection string, you can change the sdf file name as your convenience (all remains the same)]

Step 11. Now for each button handler define the action performed on OnClick event.[Refer MainPage.xaml.cs from Source code]

Step 12. Save all changes made to the project, by clicking on Ctrl + S.

Step 13. Press F5, in order to run the experiment on the Windows Phone Emulator.

\*Note- All SQL queries are made through Linq to SQL queries.

Source Code	Comments
<pre> MainPage.xaml  &lt;phone:PhoneApplicationPage   x:Class="StudentDB.MainPage"  xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation" xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml" xmlns:phone="clr- namespace:Microsoft.Phone.Controls;assembly=Microsoft.Phone" xmlns:shell="clr- namespace:Microsoft.Phone.Shell;assembly=Microsoft.Phone" xmlns:d="http://schemas.microsoft.com/expression/blend/2008" xmlns:mc="http://schemas.openxmlformats.org/markup- compatibility/2006" mc:Ignorable="d" d:DesignWidth="480" d:DesignHeight="768" FontFamily="{StaticResource PhoneFontFamilyNormal}" FontSize="{StaticResource PhoneFontSizeNormal}" Foreground="{StaticResource PhoneForegroundBrush}" SupportedOrientations="Portrait" Orientation="Portrait" shell:SystemTray.IsVisible="True"&gt;    &lt;!--LayoutRoot is the root grid where all page content is placed--&gt;   &lt;Grid x:Name="LayoutRoot" Background="Transparent"&gt;     &lt;Grid.RowDefinitions&gt;       &lt;RowDefinition Height="173"/&gt;       &lt;RowDefinition Height="595*"/&gt;     &lt;/Grid.RowDefinitions&gt;     &lt;!--TitlePanel contains the name of the application and page title--&gt;     &lt;StackPanel x:Name="TitlePanel" Grid.Row="0" Margin="12,17,0,28"&gt;       &lt;TextBlock x:Name="ExperieimentTitle" Text="Exp No:DB1" TextAlignment="Right" Style="{StaticResource PhoneTextNormalStyle}"/&gt;       &lt;TextBlock x:Name="ApplicationTitle" Text="Local DBMS" Style="{StaticResource PhoneTextNormalStyle}"/&gt;       &lt;TextBlock x:Name="PageTitle" Text="Student DB" Margin="9,-7,0,0" Style="{StaticResource PhoneTextTitle1Style}"/&gt;     &lt;/StackPanel&gt;      &lt;!--ContentPanel - place additional content here--&gt;     &lt;Grid x:Name="ContentPanel" Grid.Row="1" Margin="12,0,12,0"&gt;       &lt;StackPanel Orientation="Vertical"&gt;         &lt;StackPanel Orientation="Horizontal"&gt;           &lt;TextBlock Text="Student ID" Width="100" Margin="12" Height="75"/&gt;           &lt;TextBox x:Name="txtID" Width="335" Height="75" /&gt;         &lt;/StackPanel&gt;         &lt;StackPanel Orientation="Horizontal"&gt;           &lt;TextBlock Text="Student Name" Width="130" Margin="0" Height="75"/&gt;           &lt;TextBox x:Name="txtName" Width="330" Height="75"/&gt;         &lt;/StackPanel&gt;         &lt;StackPanel Orientation="Horizontal"&gt;           &lt;TextBlock Text="Student Age" Width="130" Height="75"/&gt;           &lt;TextBox x:Name="txtAge" Width="330" Height="75"/&gt;         &lt;/StackPanel&gt; </pre>	<p>← Experiment Title(Exp No:DB1)</p> <p>← Application Title(Local DBMS)</p> <p>← Page Title(Student DB)</p> <p>← Controls Elements</p>

```

<Button x:Name="bCreateDB" Content="Create Database"
Width="400" Height="70" Click="bCreateDB_Click"/>
<Button x:Name="bAdd" Content="Add Student" Width="400"
Height="70" Click="bAdd_Click"/>
<Button x:Name="bDelete" Content="Delete Student" Width="400"
Height="70" Click="bDelete_Click"/>
<Button x:Name="bRetrieve" Content="Fetech all Students"
Width="400" Height="70" Click="bRetrieve_Click"/>
<Button x:Name="bDeleteDB" Content="Delete Database"
Width="400" Height="70" Click="bDeleteDB_Click"/>
</StackPanel>
</Grid>
</Grid>
</phone:PhoneApplicationPage>

```

### MainPage.xaml.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Windows;
using Microsoft.Phone.Controls;
using System.Text;

namespace StudentDB
{
    public partial class MainPage : PhoneApplicationPage
    {
        private const string connectionString =
@"isostore:/StudentDB.sdf";
        // Constructor
        public MainPage()
        {
            InitializeComponent();
        }

        private void bCreateDB_Click(object sender,
RoutedEventArgs e)
        {
            // for creation of database
            using (StudentDataContext Studb = new
StudentDataContext(connectionString)) {
                if (Studb.DatabaseExists() == false)
                {
                    Studb.CreateDatabase();
                    MessageBox.Show("Student Database Created
Successfully!!");
                }
                else {
                    MessageBox.Show("Student Database already
exists!!");
                }
            }
        }

        private void bAdd_Click(object sender, RoutedEventArgs e)
        {
            // for inserting the values in the database
            using (StudentDataContext Studb = new

```

← Add namespaces

← Define Connection string  
(for Database)

← Create Database if it is  
not existing.

```

StudentDataContext(connectionString) {
    Student newStudent = new Student
    {
        studentID = Convert.ToInt32(txtID.Text),
        studentAge = Convert.ToInt16(txtAge.Text),
        studentName = txtName.Text.ToString()
    };
    Studb.Students.InsertOnSubmit(newStudent);
    Studb.SubmitChanges();
    MessageBox.Show("Student Added Successfully!!");
}
}

private void bDelete_Click(object sender, RoutedEventArgs
e)
{
    // for deleting the values in the database
    using (StudentDataContext Studb = new
StudentDataContext(connectionString) {
        IQueryable<Student> StuQuery = from Stu in
Studb.Students where Stu.studentName == txtName.Text select Stu;
        Student StuRemove = StuQuery.FirstOrDefault();
        Studb.Students.DeleteOnSubmit(StuRemove);
        Studb.SubmitChanges();
        MessageBox.Show("Student removed
successfully!!");
    }
}

private void bRetrieve_Click(object sender,
RoutedEventArgs e)
{
    // for feteching all values from the database
    IList<Student> StudentList = this.GetStudentList();
    StringBuilder str = new StringBuilder();
    str.AppendLine("Student Details");
    foreach (Student stu in StudentList) {
        str.AppendLine("Name-" + stu.studentName + " " +
"Age-" + stu.studentAge);
    }
    MessageBox.Show(str.ToString());
}

private void bDeleteDB_Click(object sender,
RoutedEventArgs e)
{
    //delete database
    using (StudentDataContext Studb = new
StudentDataContext(connectionString) {
        if (Studb.DatabaseExists()) {
            Studb.DeleteDatabase();
            MessageBox.Show("Student Database Deleted
Successfully!! ");
        }
    }
}

public IList<Student> GetStudentList() {
    //Feteching data from local database
    IList<Student> StudentList = null;
    using (StudentDataContext Studb = new
StudentDataContext(connectionString) {

```

← Get the values from UI controls

← Insert values into database

← Query to delete the record from database

← Delete the complete database.

```

        IQueryable<Student> StuQuery = from Stu in
Studb.Students select Stu;
        StudentList = StuQuery.ToList();
    }
    return StudentList;
}
}
}

```

← Query to get the list of records

### Student.cs

```

using System.Data.Linq.Mapping;

namespace StudentDB
{
    //Schema for Database
    [Table]
    public class Student
    {
        [Column(IsPrimaryKey=true,IsDbGenerated=true)]
        public int studentID
        {get;
        set;}
        [Column(CanBeNull = false)]
        public int studentAge { get; set; }
        [Column(CanBeNull = false)]
        public string studentName
        {
            get;
            set;
        }
    }
}

```

← Define the Schema for the Database.

### StudentDataContext.cs

```

using System.Data.Linq;

namespace StudentDB
{
    public class StudentDataContext: DataContext
    { //View Model for MVVM
        public StudentDataContext(string connectionString) :
base(connectionString) {
        }
        public Table<Student> Students {
            get { return this.GetTable<Student>(); }
        }
    }
}

```

← Define the Model View for the MVVM architecture

## Screenshots

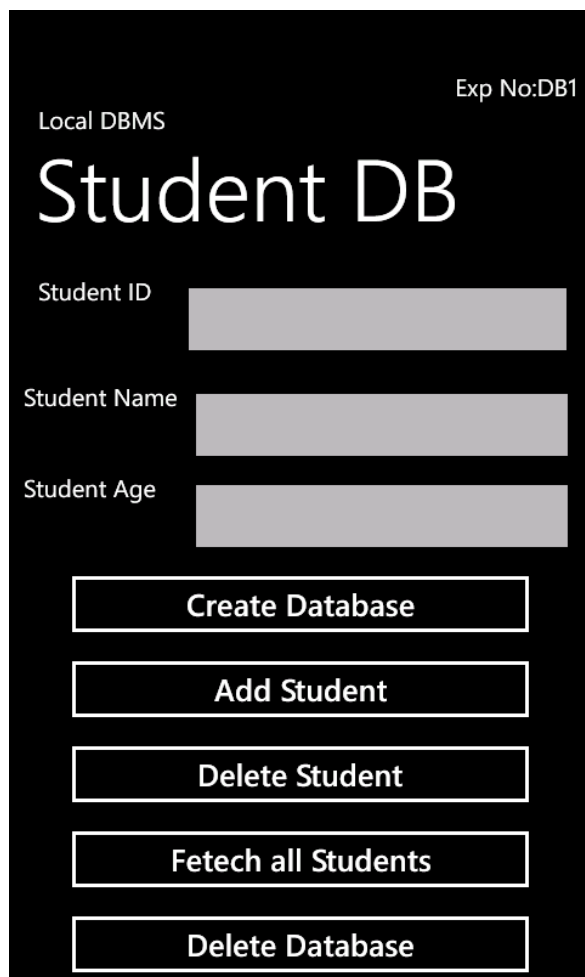


Fig. No. 1 Home Page

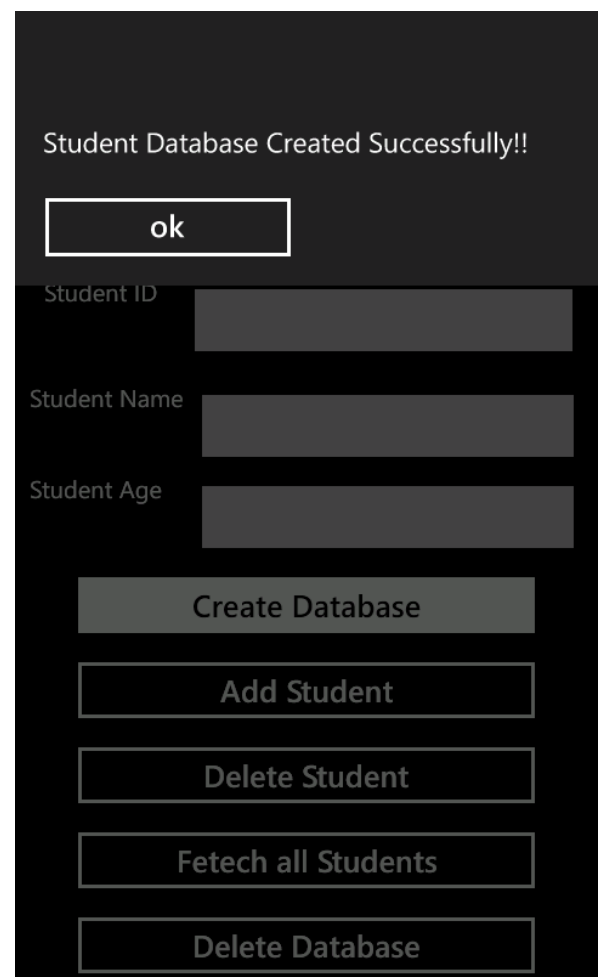


Fig. No. 2 Creating Schema for Database

Student Added Successfully!!

Student ID

Student Name

Student Age

Fig. No. 3 Inserting values in the table

Student Details  
Name-VINEET Age-25  
Name-NEHA Age-26

Student Name

Student Age

Fig. No. 4 Fetching the records

Student removed successfully!!

Student ID

Student Name

Student Age

Fig. No. 5 Deleting the records

Student Details  
Name-VINEET Age-25

Student Name

Student Age

Fig. No. 6 Displaying the records after deletion

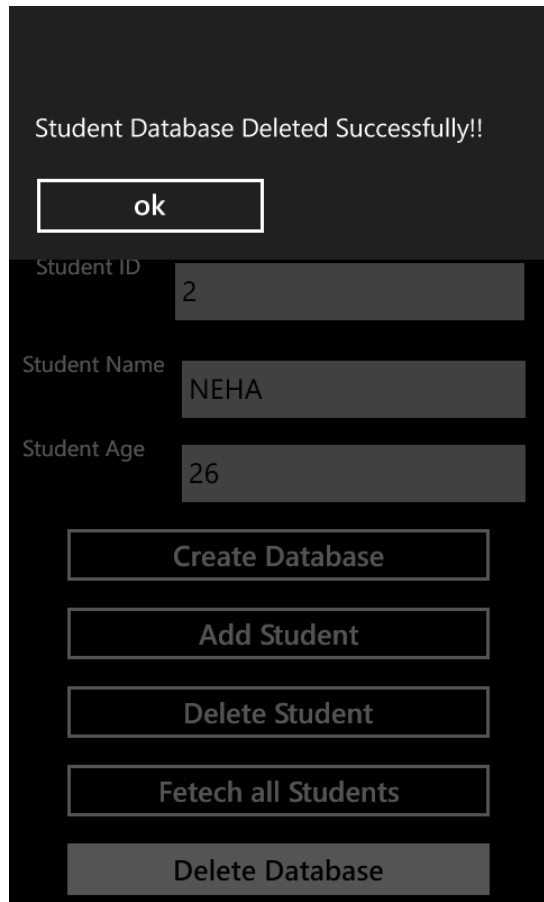


Fig. No. 7 Deletion of Database

**Observations:** It is observed by the programmer that we can easily handle the databases using inbuilt SQL server compact edition for Mobiles. The local database is stored at Isolated Storage of the phone. User cannot see the database file by normal procedure.

\*Note- Use ISETool.exe tool to copy the database file at your local machine through certain procedure explained at [http://msdn.microsoft.com/en-us/library/windowsphone/develop/hh286408\(v=vs.105\).aspx](http://msdn.microsoft.com/en-us/library/windowsphone/develop/hh286408(v=vs.105).aspx).