Practical -6
Database connectivity with SQL server

1. Create a Web application to display all the Empname and Deptid of the employee from the database using data reader. Database fields are (DeptId, DeptName, EmpName, Salary).

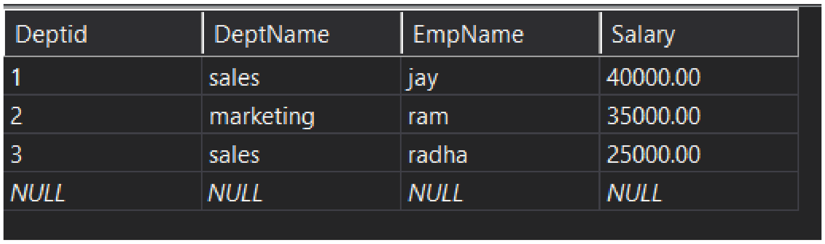
Source code:

```
<%@ Page Language="C#" AutoEventWireup="true" CodeBehind="Employee.aspx.cs"
Inherits="lab6.Employee" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            <h1>Employee Information</h1>

            <asp:GridView ID="GridView1" runat="server" AutoGenerateColumns="False"
DataSourceID="empdataSource">
            </asp:GridView>
            <asp:SqlDataSource ID="empdataSource" runat="server" ConnectionString="<%$
ConnectionStrings:EmployeeConnectionString %>" SelectCommand="SELECT [DeptId], [EmpName]
FROM [Employee]"></asp:SqlDataSource>
        </div>
    </form>
</body>
 </html>
```

employee.cs

```
using System;
using System.Collections.Generic;
using System.Data.SqlClient;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;

namespace lab6
{
    public partial class Employee : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
            {
                BindData();
            }
        }
        private void BindData()
        {
            string connectionString = "Data Source=(localdb)\\ProjectsV13;Initial
Catalog=master;Integrated Security=True;Connect
Timeout=30;Encrypt=False;TrustServerCertificate=False;ApplicationIntent=ReadWrite;MultiSubn
etFailover=False";
            using (SqlConnection connection = new SqlConnection(connectionString))
            {
                connection.Open();
                using (SqlCommand command = new SqlCommand("SELECT DeptId, EmpName
FROM Employee", connection))
                {
                    SqlDataReader reader = command.ExecuteReader();
                    GridView1.DataSource = reader;
                    GridView1.DataBind();
                }
            }
        }
    }
}
```

OUTPUT:



Deptid	DeptName	EmpName	Salary
1	sales	jay	40000.00
2	marketing	ram	35000.00
3	sales	radha	25000.00
NULL	*NULL*	*NULL*	*NULL*

2. Create an application which display Student table (StudentId, Fname, Mname, SName, Sem, Branch, Address, City, ContactNo, EmailId) data in a table format.

Source code:

```
using System;
using System.Data;
using System.Data.SqlClient;

public partial class Default : System.Web.UI.Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            BindStudentData();
        }
    }

    private void BindStudentData()
    {
        string connectionString =
System.Configuration.ConfigurationManager.ConnectionStrings["MyConnectionString"].ConnectionString;
        using (SqlConnection connection = new SqlConnection(connectionString))
        {
            string query = "SELECT * FROM Student";
            using (SqlDataAdapter adapter = new SqlDataAdapter(query, connection))
            {
                DataTable dt = new DataTable();
                adapter.Fill(dt);
                GridView1.DataSource = dt;
                GridView1.DataBind();
            }
        }
    }
}
```

Output:

```
StudentId | Fname | Mname | Sname | Sem | Branch | Address | City  | ContactNo | EmailId
------------------------------------------------------------------------------------------
1         | John  | M     | Doe   | 1   | IT     | Address1 | City1 | 123456789 | john@example.com
2         | Jane  | M     | Smith | 2   | CS     | Address2 | City2 | 987654321 | jane@example.com
3         | Bob  | A      | Johnson | 3 | EE     | Address3 | City3 | 456789123 | bob@example.com
```

3. **Implement CRUD operations for both the applications.**

Source code:

1st Application:

```
using System;
using System.Collections.Generic;
using System.Configuration;
using System.Data;
using System.Data.SqlClient;

public class EmployeeDataAccess
{
    private readonly string connectionString =
ConfigurationManager.ConnectionStrings["MyConnectionString"].ConnectionString;

    public List<Employee> GetAllEmployees()
```

```
{
    List<Employee> employees = new List<Employee>();

    using (SqlConnection connection = new SqlConnection(connectionString))
    {
        string query = "SELECT EmpName, DeptId FROM Employee";
        using (SqlCommand command = new SqlCommand(query, connection))
        {
            connection.Open();
            SqlDataReader reader = command.ExecuteReader();

            while (reader.Read())
            {
                Employee employee = new Employee
                {
                    EmpName = reader["EmpName"].ToString(),
                    DeptId = Convert.ToInt32(reader["DeptId"])
                };
                employees.Add(employee);
            }
            reader.Close();
        }
    }

    return employees;
}
        public void CreateEmployee(Employee newEmployee)
        {
            using (SqlConnection connection = new SqlConnection(connectionString))
            {
                string query = "INSERT INTO Employee (EmpName, DeptId) VALUES
        (@EmpName, @DeptId)";
                using (SqlCommand command = new SqlCommand(query, connection))
                {
                    command.Parameters.AddWithValue("@EmpName", newEmployee.EmpName);
                    command.Parameters.AddWithValue("@DeptId", newEmployee.DeptId);

                    connection.Open();
                    command.ExecuteNonQuery();
                }
            }

        public void UpdateEmployee(Employee updatedEmployee)
        {
            using (SqlConnection connection = new SqlConnection(connectionString))
            {
                string query = "UPDATE Employee SET EmpName = @EmpName, DeptId =
        @DeptId WHERE EmployeeId = @EmployeeId";
                using (SqlCommand command = new SqlCommand(query, connection))
                {
                    command.Parameters.AddWithValue("@EmpName",
        updatedEmployee.EmpName);
                    command.Parameters.AddWithValue("@DeptId", updatedEmployee.DeptId);
                    command.Parameters.AddWithValue("@EmployeeId",
        updatedEmployee.EmployeeId); // Assuming you have an EmployeeId property

                    connection.Open();
                    command.ExecuteNonQuery();
                }
            }
        }

        public void DeleteEmployee(int employeeId)
        {
            using (SqlConnection connection = new SqlConnection(connectionString))
            {
                string query = "DELETE FROM Employee WHERE EmployeeId = @EmployeeId";
                using (SqlCommand command = new SqlCommand(query, connection))
                {
                    command.Parameters.AddWithValue("@EmployeeId", employeeId);

                    connection.Open();
                    command.ExecuteNonQuery();
                }
            }
        }

}

}
```

Bind the Data in to GridView:

```
using System;
using System.Web.UI;
using System.Data;
using System.Data.SqlClient;
```

```
public partial class Default : Page
{
    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
            BindEmployeeData();
        }
    }

    private void BindEmployeeData()
    {
        string connectionString =
ConfigurationManager.ConnectionStrings["MyConnectionString"].ConnectionString;
        using (SqlConnection connection = new SqlConnection(connectionString))
        {
            string query = "SELECT EmpName, DeptId FROM Employee";
            using (SqlCommand command = new SqlCommand(query, connection))
            {
                connection.Open();
                SqlDataAdapter adapter = new SqlDataAdapter(command);
                DataTable dt = new DataTable();
                adapter.Fill(dt);
                GridView1.DataSource = dt;
                GridView1.DataBind();
            }
        }
    }
}
```

Output:

```
EmpName      | DeptId
----------------------
John Doe     | 101
Jane Smith | 102
Bob Johnson  | 101
Alice Brown  | 103
```

2nd Application:

```
using System;
using System.Collections.Generic;
using System.Configuration;
using System.Data;
using System.Data.SqlClient;

public class StudentDataAccess
{
    private readonly string connectionString =
ConfigurationManager.ConnectionStrings["MyConnectionString"].ConnectionString;

    public List<Student> GetAllStudents()
    {
        List<Student> students = new List<Student>();

        using (SqlConnection connection = new SqlConnection(connectionString))
        {
            string query = "SELECT * FROM Student";
            using (SqlCommand command = new SqlCommand(query, connection))
            {
                connection.Open();
                SqlDataReader reader = command.ExecuteReader();

                while (reader.Read())
                {
                    Student student = new Student
                    {
                        StudentId = Convert.ToInt32(reader["StudentId"]),
                        Fname = reader["Fname"].ToString(),
                        Mname = reader["Mname"].ToString(),
                        SName = reader["SName"].ToString(),
                        Sem = Convert.ToInt32(reader["Sem"]),
                        Branch = reader["Branch"].ToString(),
                        Address = reader["Address"].ToString(),
                        City = reader["City"].ToString(),
                        ContactNo = reader["ContactNo"].ToString(),
                        EmailId = reader["EmailId"].ToString()
                    };
                    students.Add(student);
                }
                reader.Close();
            }
        }

        return students;
    }
    public void InsertStudent(Student student)
    {
        using (SqlConnection connection = new SqlConnection(connectionString))
        {
            string query = "INSERT INTO Student (Fname, Mname, SName, Sem, Branch, Address,
City, ContactNo, EmailId) " +
                           "VALUES (@Fname, @Mname, @SName, @Sem, @Branch, @Address,
@City, @ContactNo, @EmailId)";

            using (SqlCommand command = new SqlCommand(query, connection))
            {
                command.Parameters.AddWithValue("@Fname", student.Fname);
                command.Parameters.AddWithValue("@Mname", student.Mname);
                command.Parameters.AddWithValue("@SName", student.SName);
                command.Parameters.AddWithValue("@Sem", student.Sem);
                command.Parameters.AddWithValue("@Branch", student.Branch);
                command.Parameters.AddWithValue("@Address", student.Address);
                command.Parameters.AddWithValue("@City", student.City);
                command.Parameters.AddWithValue("@ContactNo", student.ContactNo);
                command.Parameters.AddWithValue("@EmailId", student.EmailId);

                connection.Open();
                command.ExecuteNonQuery();
            }
        }
    }

    public void UpdateStudent(Student student)
    {
        using (SqlConnection connection = new SqlConnection(connectionString))
        {
            string query = "UPDATE Student SET Fname = @Fname, Mname = @Mname, SName
= @SName, " +
                           "Sem = @Sem, Branch = @Branch, Address = @Address, City = @City, " +
                           "ContactNo = @ContactNo, EmailId = @EmailId " +
                           "WHERE StudentId = @StudentId";

            using (SqlCommand command = new SqlCommand(query, connection))
            {
                command.Parameters.AddWithValue("@Fname", student.Fname);
                command.Parameters.AddWithValue("@Mname", student.Mname);
                command.Parameters.AddWithValue("@SName", student.SName);
                command.Parameters.AddWithValue("@Sem", student.Sem);
                command.Parameters.AddWithValue("@Branch", student.Branch);
                command.Parameters.AddWithValue("@Address", student.Address);
                command.Parameters.AddWithValue("@City", student.City);
                command.Parameters.AddWithValue("@ContactNo", student.ContactNo);
                command.Parameters.AddWithValue("@EmailId", student.EmailId);
                command.Parameters.AddWithValue("@StudentId", student.StudentId);

                connection.Open();
                command.ExecuteNonQuery();
            }
        }
    }

    public void DeleteStudent(int studentId)
    {
        using (SqlConnection connection = new SqlConnection(connectionString))
        {
            string query = "DELETE FROM Student WHERE StudentId = @StudentId";

            using (SqlCommand command = new SqlCommand(query, connection))
            {
                command.Parameters.AddWithValue("@StudentId", studentId);

                connection.Open();
                command.ExecuteNonQuery();
            }
        }
    }
}
```

Bind Data to GridView:

```
using System;
using System.Web.UI;
using System.Collections.Generic;

public partial class Default : Page
{
    private readonly StudentDataAccess dataAccess = new StudentDataAccess();

    protected void Page_Load(object sender, EventArgs e)
    {
        if (!IsPostBack)
        {
```

```
        BindStudentData();
    }
}

private void BindStudentData()
{
    List<Student> students = dataAccess.GetAllStudents();
    GridView1.DataSource = students;
    GridView1.DataBind();
}

protected void InsertButton_Click(object sender, EventArgs e)
{
    Student newStudent = new Student
    {
        Fname = FirstNameTextBox.Text,
        Mname = MiddleNameTextBox.Text,
        SName = LastNameTextBox.Text,
        Sem = Convert.ToInt32(SemesterTextBox.Text),
        Branch = BranchTextBox.Text,
        Address = AddressTextBox.Text,
        City = CityTextBox.Text,
        ContactNo = ContactNoTextBox.Text,
        EmailId = EmailIdTextBox.Text
    };

    dataAccess.InsertStudent(newStudent);

    BindStudentData();

    ClearForm();
}

protected void UpdateButton_Click(object sender, EventArgs e)
{
    Student updatedStudent = new Student
    {
        StudentId = Convert.ToInt32(StudentIdTextBox.Text),
        Fname = FirstNameTextBox.Text,
        Mname = MiddleNameTextBox.Text,
        SName = LastNameTextBox.Text,
        Sem = Convert.ToInt32(SemesterTextBox.Text),
        Branch = BranchTextBox.Text,
        Address = AddressTextBox.Text,
            City = CityTextBox.Text,
            ContactNo = ContactNoTextBox.Text,
            EmailId = EmailIdTextBox.Text
        };

        dataAccess.UpdateStudent(updatedStudent);

        BindStudentData();

        ClearForm();
    }

    protected void DeleteButton_Click(object sender, EventArgs e)
    {
        int studentId = Convert.ToInt32(StudentIdToDelete.Text);

        dataAccess.DeleteStudent(studentId);

        BindStudentData();

        ClearForm();
    }

    private void ClearForm()
    {
        StudentIdTextBox.Text = string.Empty;
        FirstNameTextBox.Text = string.Empty;
        MiddleNameTextBox.Text = string.Empty;
        LastNameTextBox.Text = string.Empty;
        SemesterTextBox.Text = string.Empty;
        BranchTextBox.Text = string.Empty;
        AddressTextBox.Text = string.Empty;
        CityTextBox.Text = string.Empty;
        ContactNoTextBox.Text = string.Empty;
        EmailIdTextBox.Text = string.Empty;
    }
}
```

Output:

```
StudentId | Fname | Mname | SName | Sem | Branch | Address | City  | ContactNo     | EmailId
------------------------------------------------------------------------------------------------
1         | John  | M     | Doe   | 1   | IT     | Addr1  | City1 | 123-456-7890 | john@example.com
2         | Jane  | M     | Smith | 2   | CS     | Addr2  | City2 | 987-654-3210 | jane@example.com
3         | Bob   | A     | Johnson | 3 | EE     | Addr3  | City3 | 456-789-1230 | bob@example.com
```