// Step 1:

using System.Data.SqlClient;

// Step 2:

SqlConnection con = new SqlConnection();

Step 3:

Enter Details of ConnectionString

connectionString : contains information about the database

data source -> Server name

initial catalog > database name

credentials > If I connect by windows authentication , I will use integrated security = true

If I connect by Sql Server authentication , I will use user id=userid;password=password

string connectionString = @"data source=LAPTOP-53S2KQS8\SQLEXPRESS; initial catalog=PracticeDB;integrated security=true"

con.ConnectionString = connectionString;

Step 4:

SqlCommand com= new SqlCommand(“Select \* from Employee”, con);

create database PracticeDB

use PracticeDB

create table Employee (id int primary key, name varchar(20),

department varchar(20), salary int)

select \* from Employee

using System;

// Step 1

using System.Data.SqlClient;

namespace ADO.NetDemo

{

class Program

{

static void Main(string[] args)

{

// Step 2:

SqlConnection con = new SqlConnection();

// Step 3:

string connectionString = @"data source=LAPTOP-53S2KQS8\SQLEXPRESS; initial catalog=PracticeDB;integrated security=true";

con.ConnectionString = connectionString;

//Step 4:

// Create object of SqlCommand

// SqlCommand com = new SqlCommand("Select \* from Employee", con);

SqlCommand com = new SqlCommand();

com.CommandText = "Select \* from Employee";

com.Connection = con;

// Step 5: Open Connection

con.Open();

//Step 6: To Read Records / Fetch Records , we use ExecuteReader() Method

SqlDataReader reader = com.ExecuteReader();

// Step 7: Iterate through the Records

while (reader.Read())

{

Console.WriteLine("{0} \t {1} \t {2} \t {3}", reader[0], reader[1], reader[2], reader[3]);

}

con.Close();

}

}

}

using System;

// Step 1

using System.Data.SqlClient;

namespace ADO.NetDemo

{

class Program

{

static void Main(string[] args)

{

InsertEmployee();

GetEmployees();

UpdateEmployee();

GetEmployees();

DeleteEmployee();

GetEmployees();

}

static void InsertEmployee()

{

SqlConnection con = new SqlConnection();

string connectionString = @"data source=LAPTOP-53S2KQS8\SQLEXPRESS; initial catalog=PracticeDB;integrated security=true";

con.ConnectionString = connectionString;

SqlCommand com = new SqlCommand();

com.CommandText = "Insert into Employee(id,name,department,salary) values(5,'Ajay','Delhi',90000)";

com.Connection = con;

con.Open();

com.ExecuteNonQuery();

con.Close();

}

static void UpdateEmployee()

{

SqlConnection con = new SqlConnection();

string connectionString = @"data source=LAPTOP-53S2KQS8\SQLEXPRESS; initial catalog=PracticeDB;integrated security=true";

con.ConnectionString = connectionString;

SqlCommand com = new SqlCommand();

com.CommandText = "Update Employee set salary= 12000 ,department='HR' where id = 2";

com.Connection = con;

con.Open();

com.ExecuteNonQuery();

con.Close();

}

static void DeleteEmployee()

{

SqlConnection con = new SqlConnection();

string connectionString = @"data source=LAPTOP-53S2KQS8\SQLEXPRESS; initial catalog=PracticeDB;integrated security=true";

con.ConnectionString = connectionString;

SqlCommand com = new SqlCommand();

com.CommandText = "Delete Employee where id= 2";

com.Connection = con;

con.Open();

com.ExecuteNonQuery();

con.Close();

}

static void GetEmployees()

{

// Step 2:

SqlConnection con = new SqlConnection();

// Step 3:

string connectionString = @"data source=LAPTOP-53S2KQS8\SQLEXPRESS; initial catalog=PracticeDB;integrated security=true";

con.ConnectionString = connectionString;

//Step 4:

// Create object of SqlCommand

// SqlCommand com = new SqlCommand("Select \* from Employee", con);

SqlCommand com = new SqlCommand();

com.CommandText = "Select \* from Employee";

com.Connection = con;

// Step 5: Open Connection

con.Open();

//Step 6: To Read Records / Fetch Records , we use ExecuteReader() Method

SqlDataReader reader = com.ExecuteReader();

// Step 7: Iterate through the Records

while (reader.Read())

{

Console.WriteLine("{0} \t {1} \t {2} \t {3}", reader[0], reader[1], reader[2], reader[3]);

}

con.Close();

}

}

}

using System;

// Step 1

using System.Data.SqlClient;

namespace ADO.NetDemo

{

class Program

{

static SqlConnection con = new SqlConnection();

static SqlCommand com = new SqlCommand();

static void Main(string[] args)

{

InsertEmployee();

GetEmployees();

UpdateEmployee();

GetEmployees();

DeleteEmployee();

GetEmployees();

}

static SqlConnection GetConnection()

{

string connectionString = @"data source=LAPTOP-53S2KQS8\SQLEXPRESS; initial catalog=PracticeDB;integrated security=true";

con.ConnectionString = connectionString;

return con;

}

static void InsertEmployee()

{

con = GetConnection();

com.CommandText = "Insert into Employee(id,name,department,salary) values(5,'Ajay','Delhi',90000)";

com.Connection = con;

con.Open();

com.ExecuteNonQuery();

con.Close();

}

static void UpdateEmployee()

{

con = GetConnection();

com.CommandText = "Update Employee set salary= 12000 ,department='HR' where id = 2";

com.Connection = con;

con.Open();

com.ExecuteNonQuery();

con.Close();

}

static void DeleteEmployee()

{

con = GetConnection();

com.CommandText = "Delete Employee where id= 2";

com.Connection = con;

con.Open();

com.ExecuteNonQuery();

con.Close();

}

static void GetEmployees()

{

// Step 2:

SqlConnection con = new SqlConnection();

// Step 3:

con = GetConnection();

com.CommandText = "Select \* from Employee";

com.Connection = con;

// Step 5: Open Connection

con.Open();

//Step 6: To Read Records / Fetch Records , we use ExecuteReader() Method

SqlDataReader reader = com.ExecuteReader();

// Step 7: Iterate through the Records

while (reader.Read())

{

Console.WriteLine("{0} \t {1} \t {2} \t {3}", reader[0], reader[1], reader[2], reader[3]);

}

con.Close();

}

}

}

static void DeleteEmployee()

{

con = GetConnection();

com.CommandText = "Delete Employee";

com.Connection = con;

con.Open();

int x= com.ExecuteNonQuery(); // Return type ExecuteNonQuery is int , which gives you no. of Records affcted

Console.WriteLine("Records Deleted are " + x.ToString());

con.Close();

}

ExecuteScalar() Returns a single value

using System;

// Step 1

using System.Data.SqlClient;

namespace ADO.NetDemo

{

class Program

{

static SqlConnection con = new SqlConnection();

static SqlCommand com = new SqlCommand();

static void Main(string[] args)

{

//InsertEmployee();

//GetEmployees();

GetEmployeeNameById();

// UpdateEmployee();

// GetEmployees();

// DeleteEmployee();

// GetEmployees();

}

private static void GetEmployeeNameById()

{

con = GetConnection();

com.CommandText = "Select name from Employee where id=2";

com.Connection = con;

con.Open();

string name = com.ExecuteScalar().ToString();

Console.WriteLine("Name is " + name);

con.Close();

}

static SqlConnection GetConnection()

{

string connectionString = @"data source=LAPTOP-53S2KQS8\SQLEXPRESS; initial catalog=PracticeDB;integrated security=true";

con.ConnectionString = connectionString;

return con;

}

static void InsertEmployee()

{

con = GetConnection();

com.CommandText = "Insert into Employee(id,name,department,salary) values(1,'Ajay','Delhi',90000),(2,'Vijay','Delhi',90000)," +

"(3,'Preet','Delhi',90000),(4,'Sagar','Delhi',90000)";

com.Connection = con;

con.Open();

com.ExecuteNonQuery();

con.Close();

}

static void UpdateEmployee()

{

con = GetConnection();

com.CommandText = "Update Employee set salary= 12000 ,department='HR' where id = 2";

com.Connection = con;

con.Open();

com.ExecuteNonQuery();

con.Close();

}

static void DeleteEmployee()

{

con = GetConnection();

com.CommandText = "Delete Employee";

com.Connection = con;

con.Open();

int x= com.ExecuteNonQuery(); // Return type ExecuteNonQuery is int , which gives you no. of Records affcted

Console.WriteLine("Records Deleted are " + x.ToString());

con.Close();

}

}

When we inherit a class

When we create object of child class , it always instantiates object of base class

Inheritance : Constructors are not inherited

class A

{

public A()

{

}

}

class B

{

public B() : base (){}

}

When you create object of child class, it will invoke of child class , and it invokes def cons of base class