Step 1: Add reference to namespace / dll

Using System.Data.SqlClient

Step 2: Create object of SqlConnection Class, in this pass connectionString

Step 3: Create object of SqlCommand - 2 parameters > Command , Connection object

Step 4: Execute your Query , ExecuteReader() , ExecuteNonQuery(), ExecuteScalar()

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data.SqlClient;

namespace AdoDemo13thNov

{

class Program

{

static void Main(string[] args)

{

//SqlConnection con = new SqlConnection();

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

//con.ConnectionString = connectionString;

SqlConnection con = new SqlConnection(@"data source = LAPTOP-53S2KQS8\SQLEXPRESS; initial catalog = practiceDB; integrated security = true");

//SqlCommand com = new SqlCommand();

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

//com.Connection = con;

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

con.Open();

SqlDataReader reader = com.ExecuteReader();

while(reader.Read())

{

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

}

con.Close();

}

}

}

Garbage Collection runs to delete which objects : Managed one > Managed Code is the code Which is understood by CLR

Database Connection it is not managed one

con.Dispose();

com.Dispose();

con.Dispose();

Call Dispose explicitly to dispose of object after its being used

OR Use using block

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data.SqlClient;

namespace AdoDemo13thNov

{

class Program

{

static void Main(string[] args)

{

//SqlConnection con = new SqlConnection();

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

//con.ConnectionString = connectionString;

using (SqlConnection con = new SqlConnection(@"data source = LAPTOP-53S2KQS8\SQLEXPRESS; initial catalog = practiceDB; integrated security = true"))

{

//SqlCommand com = new SqlCommand();

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

//com.Connection = con;

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

{

con.Open();

SqlDataReader reader = com.ExecuteReader();

while (reader.Read())

{

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

}

con.Close();

}

}

}

}

}

Till now, our queries were hard coded

But now , we want to parameters , parameterized Queries

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data.SqlClient;

namespace AdoDemo13thNov

{

class Program

{

static void Insert()

{

int id;

string name;

string dept;

int salary;

Console.WriteLine("ENter ID");

id = Convert.ToByte(Console.ReadLine());

Console.WriteLine("ENter Name");

name = Console.ReadLine();

Console.WriteLine("ENter Dept");

dept = Console.ReadLine();

Console.WriteLine("Enter Salary");

salary = int.Parse(Console.ReadLine());

SqlConnection con = new SqlConnection(@"data source = LAPTOP-53S2KQS8\SQLEXPRESS; initial catalog = practiceDB; integrated security = true");

SqlCommand com = new SqlCommand("Insert into employee(id, name, department, salary) values (@id,@name,@department,@salary)", con);

SqlParameter p1 = new SqlParameter("@id", id);

SqlParameter p2 = new SqlParameter("@name", name);

SqlParameter p3 = new SqlParameter("@department", dept);

SqlParameter p4 = new SqlParameter("@salary", salary);

com.Parameters.Add(p1);

com.Parameters.Add(p2);

com.Parameters.Add(p3);

com.Parameters.Add(p4);

con.Open();

com.ExecuteNonQuery();

con.Close();

}

static void Main(string[] args)

{

//SqlConnection con = new SqlConnection();

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

//con.ConnectionString = connectionString;

Insert();

// using (SqlConnection con = new SqlConnection(@"data source = LAPTOP-53S2KQS8\SQLEXPRESS; initial catalog = practiceDB; integrated security = true"), con)

//{

// //SqlCommand com = new SqlCommand();

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

// //com.Connection = con;

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

// {

// con.Open();

// SqlDataReader reader = com.ExecuteReader();

// while (reader.Read())

// {

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

// }

// con.Close();

// }

//}

}

}

}

static void Insert()

{

int id;

string name;

string dept;

int salary;

Console.WriteLine("ENter ID");

id = Convert.ToByte(Console.ReadLine());

Console.WriteLine("ENter Name");

name = Console.ReadLine();

Console.WriteLine("ENter Dept");

dept = Console.ReadLine();

Console.WriteLine("Enter Salary");

salary = int.Parse(Console.ReadLine());

using (SqlConnection con = new SqlConnection(@"data source = LAPTOP-53S2KQS8\SQLEXPRESS; initial catalog = practiceDB; integrated security = true"))

{

using (SqlCommand com = new SqlCommand("Insert into employee(id, name, department, salary) values (@id,@name,@department,@salary)", con))

{

SqlParameter p1 = new SqlParameter("@id", id);

SqlParameter p2 = new SqlParameter("@name", name);

SqlParameter p3 = new SqlParameter("@department", dept);

SqlParameter p4 = new SqlParameter("@salary", salary);

com.Parameters.Add(p1);

com.Parameters.Add(p2);

com.Parameters.Add(p3);

com.Parameters.Add(p4);

con.Open();

com.ExecuteNonQuery();

con.Close();

}

}

Complete Program

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Data.SqlClient;

namespace AdoDemo13thNov

{

class Program

{

static SqlConnection con;

static SqlCommand com;

static string GetConnectionString()

{

return @"data source = LAPTOP-53S2KQS8\SQLEXPRESS; initial catalog = practiceDB; integrated security = true";

}

static void Insert()

{

int id;

string name;

string dept;

int salary;

Console.WriteLine("ENter ID");

id = Convert.ToByte(Console.ReadLine());

Console.WriteLine("ENter Name");

name = Console.ReadLine();

Console.WriteLine("ENter Dept");

dept = Console.ReadLine();

Console.WriteLine("Enter Salary");

salary = int.Parse(Console.ReadLine());

using (con = new SqlConnection(GetConnectionString()))

{

using (com = new SqlCommand("Insert into employee(id, name, department, salary) values (@id,@name,@department,@salary)", con))

{

SqlParameter p1 = new SqlParameter("@id", id);

SqlParameter p2 = new SqlParameter("@name", name);

SqlParameter p3 = new SqlParameter("@department", dept);

SqlParameter p4 = new SqlParameter("@salary", salary);

com.Parameters.Add(p1);

com.Parameters.Add(p2);

com.Parameters.Add(p3);

com.Parameters.Add(p4);

con.Open();

com.ExecuteNonQuery();

con.Close();

}

}

}

static void UpdateEmployee()

{

int id;

string dept;

int salary;

Console.WriteLine("ENter ID whose Record to be updated");

id = Convert.ToByte(Console.ReadLine());

Console.WriteLine("ENter Dept");

dept = Console.ReadLine();

Console.WriteLine("Enter Salary");

salary = int.Parse(Console.ReadLine());

using (con = new SqlConnection(GetConnectionString()))

{

using (com = new SqlCommand("update employee set department= @department , salary = @salary where id = @id", con))

{

SqlParameter p1 = new SqlParameter("@id", id);

SqlParameter p3 = new SqlParameter("@department", dept);

SqlParameter p4 = new SqlParameter("@salary", salary);

com.Parameters.Add(p1);

com.Parameters.Add(p3);

com.Parameters.Add(p4);

con.Open();

com.ExecuteNonQuery();

con.Close();

}

}

}

static void DeleteEmployee()

{

int id;

Console.WriteLine("ENter ID whose Record to be Deleted");

id = Convert.ToByte(Console.ReadLine());

using (con = new SqlConnection(GetConnectionString()))

{

using (com = new SqlCommand("delete employee where id = @id", con))

{

SqlParameter p1 = new SqlParameter("@id", id);

com.Parameters.Add(p1);

con.Open();

com.ExecuteNonQuery();

con.Close();

}

}

}

static void GetEmployee()

{

using (con = new SqlConnection(GetConnectionString()))

{

//SqlCommand com = new SqlCommand();

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

//com.Connection = con;

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

{

con.Open();

SqlDataReader reader = com.ExecuteReader();

while (reader.Read())

{

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

}

con.Close();

}

}

}

static void GetEmployeesCount()

{

using (con = new SqlConnection(GetConnectionString()))

{

//SqlCommand com = new SqlCommand();

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

//com.Connection = con;

using (SqlCommand com = new SqlCommand("Select count(\*) from Employee", con))

{

con.Open();

int count = (int)com.ExecuteScalar();

con.Close();

Console.WriteLine("No of employees are " + count);

}

}

}

static void Main(string[] args)

{

//SqlConnection con = new SqlConnection();

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

//con.ConnectionString = connectionString;

Insert();

UpdateEmployee();

DeleteEmployee();

GetEmployee();

GetEmployeesCount();

}

}

}

Stored Procedures

use PracticeDB

select \* from employee

create proc GetEmployees

As

Begin

Select \* from Employee

End

exec GetEmployees

create proc InsertEmployee(@id int , @name varchar(20), @department varchar(20),

@salary int)

As

Begin

Insert into EMployee(id, name, department, salary ) values(@id, @name, @department,@salary)

ENd

create proc UpdateEmployee(@id int , @department varchar(20),

@salary int)

As

Begin

update EMployee set department = @department , salary=@salary where id = @id

ENd

create proc DeleteEmployee(@id int)

As

Begin

Delete EMployee where id = @id

ENd

using System;

using System.Collections.Generic;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace AdoDemo13thNov

{

class ADONetProcedureDemo

{

static SqlConnection con;

static SqlCommand com;

static string GetConnectionString()

{

return @"data source = LAPTOP-53S2KQS8\SQLEXPRESS; initial catalog = practiceDB; integrated security = true";

}

static void GetEmployee()

{

using (con = new SqlConnection(GetConnectionString()))

{

//SqlCommand com = new SqlCommand();

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

//com.Connection = con;

using (SqlCommand com = new SqlCommand("GetEmployees", con))

{

com.CommandType = System.Data.CommandType.StoredProcedure;

con.Open();

SqlDataReader reader = com.ExecuteReader();

while (reader.Read())

{

for(int i =0;i<reader.FieldCount;i++)

Console.Write("{0}\t" ,reader[i]);

Console.WriteLine();

}

con.Close();

}

}

}

static void Insert()

{

int id;

string name;

string dept;

int salary;

Console.WriteLine("ENter ID");

id = Convert.ToByte(Console.ReadLine());

Console.WriteLine("ENter Name");

name = Console.ReadLine();

Console.WriteLine("ENter Dept");

dept = Console.ReadLine();

Console.WriteLine("Enter Salary");

salary = int.Parse(Console.ReadLine());

using (con = new SqlConnection(GetConnectionString()))

{

using (com = new SqlCommand("InsertEmployee", con))

{

com.CommandType = System.Data.CommandType.StoredProcedure;

SqlParameter p1 = new SqlParameter("@id", id);

SqlParameter p2 = new SqlParameter("@name", name);

SqlParameter p3 = new SqlParameter("@department", dept);

SqlParameter p4 = new SqlParameter("@salary", salary);

com.Parameters.Add(p1);

com.Parameters.Add(p2);

com.Parameters.Add(p3);

com.Parameters.Add(p4);

con.Open();

com.ExecuteNonQuery();

con.Close();

}

}

}

static void UpdateEmployee()

{

int id;

string dept;

int salary;

Console.WriteLine("ENter ID whose Record to be updated");

id = Convert.ToByte(Console.ReadLine());

Console.WriteLine("ENter Dept");

dept = Console.ReadLine();

Console.WriteLine("Enter Salary");

salary = int.Parse(Console.ReadLine());

using (con = new SqlConnection(GetConnectionString()))

{

using (com = new SqlCommand("UpdateEmployee", con))

{

com.CommandType = System.Data.CommandType.StoredProcedure;

SqlParameter p1 = new SqlParameter("@id", id);

SqlParameter p3 = new SqlParameter("@department", dept);

SqlParameter p4 = new SqlParameter("@salary", salary);

com.Parameters.Add(p1);

com.Parameters.Add(p3);

com.Parameters.Add(p4);

con.Open();

com.ExecuteNonQuery();

con.Close();

}

}

}

static void DeleteEmployee()

{

int id;

Console.WriteLine("ENter ID whose Record to be Deleted");

id = Convert.ToByte(Console.ReadLine());

using (con = new SqlConnection(GetConnectionString()))

{

using (com = new SqlCommand("DeleteEmployee", con))

{

com.CommandType = System.Data.CommandType.StoredProcedure;

SqlParameter p1 = new SqlParameter("@id", id);

com.Parameters.Add(p1);

con.Open();

com.ExecuteNonQuery();

con.Close();

}

}

}

static void Main()

{

GetEmployee();

UpdateEmployee();

DeleteEmployee();

Insert();

}

}

}

// Return Value & Output Parameters in Stored Procedures

using System;

using System.Collections.Generic;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace AdoDemo13thNov

{

class ADONetProcedureDemo

{

static SqlConnection con;

static SqlCommand com;

static string GetConnectionString()

{

return @"data source = LAPTOP-53S2KQS8\SQLEXPRESS; initial catalog = practiceDB; integrated security = true";

}

static void GetEmployee()

{

using (con = new SqlConnection(GetConnectionString()))

{

//SqlCommand com = new SqlCommand();

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

//com.Connection = con;

using (SqlCommand com = new SqlCommand("GetEmployees", con))

{

com.CommandType = System.Data.CommandType.StoredProcedure;

con.Open();

SqlDataReader reader = com.ExecuteReader();

while (reader.Read())

{

for(int i =0;i<reader.FieldCount;i++)

Console.Write("{0}\t" ,reader[i]);

Console.WriteLine();

}

con.Close();

}

}

}

//static void Insert()

//{

// int id;

// string name;

// string dept;

// int salary;

// Console.WriteLine("ENter ID");

// id = Convert.ToByte(Console.ReadLine());

// Console.WriteLine("ENter Name");

// name = Console.ReadLine();

// Console.WriteLine("ENter Dept");

// dept = Console.ReadLine();

// Console.WriteLine("Enter Salary");

// salary = int.Parse(Console.ReadLine());

// using (con = new SqlConnection(GetConnectionString()))

// {

// using (com = new SqlCommand("InsertEmployee", con))

// {

// com.CommandType = System.Data.CommandType.StoredProcedure;

// SqlParameter p1 = new SqlParameter("@id", id);

// SqlParameter p2 = new SqlParameter("@name", name);

// SqlParameter p3 = new SqlParameter("@department", dept);

// SqlParameter p4 = new SqlParameter("@salary", salary);

// com.Parameters.Add(p1);

// com.Parameters.Add(p2);

// com.Parameters.Add(p3);

// com.Parameters.Add(p4);

// con.Open();

// com.ExecuteNonQuery();

// con.Close();

// }

// }

//}

static void Insert()

{

int id;

string name;

string dept;

int salary;

Console.WriteLine("ENter ID");

id = Convert.ToByte(Console.ReadLine());

Console.WriteLine("ENter Name");

name = Console.ReadLine();

Console.WriteLine("ENter Dept");

dept = Console.ReadLine();

Console.WriteLine("Enter Salary");

salary = int.Parse(Console.ReadLine());

using (con = new SqlConnection(GetConnectionString()))

{

using (com = new SqlCommand("InsertEmployee", con))

{

com.CommandType = System.Data.CommandType.StoredProcedure;

SqlParameter p1 = new SqlParameter("@id", id);

SqlParameter p2 = new SqlParameter("@name", name);

SqlParameter p3 = new SqlParameter("@department", dept);

SqlParameter p4 = new SqlParameter("@salary", salary);

SqlParameter res = new SqlParameter();

res.Direction = System.Data.ParameterDirection.ReturnValue;

com.Parameters.Add(p1);

com.Parameters.Add(p2);

com.Parameters.Add(p3);

com.Parameters.Add(p4);

com.Parameters.Add(res);

con.Open();

com.ExecuteNonQuery();

int flag = (int)res.Value;

if(flag==0)

{

Console.WriteLine("Record with this ID already exists");

}

else

{

Console.WriteLine("Record inserted successfully");

}

con.Close();

}

}

}

static void UpdateEmployee()

{

int id;

string dept;

int salary;

Console.WriteLine("ENter ID whose Record to be updated");

id = Convert.ToByte(Console.ReadLine());

Console.WriteLine("ENter Dept");

dept = Console.ReadLine();

Console.WriteLine("Enter Salary");

salary = int.Parse(Console.ReadLine());

using (con = new SqlConnection(GetConnectionString()))

{

using (com = new SqlCommand("UpdateEmployee", con))

{

com.CommandType = System.Data.CommandType.StoredProcedure;

SqlParameter p1 = new SqlParameter("@id", id);

SqlParameter p3 = new SqlParameter("@department", dept);

SqlParameter p4 = new SqlParameter("@salary", salary);

com.Parameters.Add(p1);

com.Parameters.Add(p3);

com.Parameters.Add(p4);

con.Open();

com.ExecuteNonQuery();

con.Close();

}

}

}

static void DeleteEmployee()

{

int id;

Console.WriteLine("ENter ID whose Record to be Deleted");

id = Convert.ToByte(Console.ReadLine());

using (con = new SqlConnection(GetConnectionString()))

{

using (com = new SqlCommand("DeleteEmployee", con))

{

com.CommandType = System.Data.CommandType.StoredProcedure;

SqlParameter p1 = new SqlParameter("@id", id);

com.Parameters.Add(p1);

con.Open();

com.ExecuteNonQuery();

con.Close();

}

}

}

static void GetEmployeeDetailsByID()

{

int id;

string name;

string dept;

int salary;

Console.WriteLine("ENter ID");

id = Convert.ToByte(Console.ReadLine());

using (con = new SqlConnection(GetConnectionString()))

{

using (com = new SqlCommand("GetEmployeeDetailsByID", con))

{

com.CommandType = System.Data.CommandType.StoredProcedure;

SqlParameter p1 = new SqlParameter("@id", id);

SqlParameter p2 = new SqlParameter();

p2.ParameterName = "name";

p2.Direction = System.Data.ParameterDirection.Output;

p2.SqlDbType = System.Data.SqlDbType.VarChar;

p2.Size = 20;

SqlParameter p3 = new SqlParameter("@department", System.Data.SqlDbType.VarChar, 20);

p3.Direction = System.Data.ParameterDirection.Output;

SqlParameter p4 = new SqlParameter("@salary", System.Data.SqlDbType.Int);

p4.Direction = System.Data.ParameterDirection.Output;

com.Parameters.Add(p1);

com.Parameters.Add(p2);

com.Parameters.Add(p3);

com.Parameters.Add(p4);

con.Open();

com.ExecuteNonQuery();

name = p2.Value.ToString();

dept = p3.Value.ToString();

salary = (int)p4.Value;

Console.WriteLine("Details are {0} {1} {2} {3}", id, name, dept, salary);

}

}

}

static void Main()

{

// GetEmployee();

// UpdateEmployee();

// DeleteEmployee();

// Insert();

GetEmployeeDetailsByID();

}

}

}