Qn: CRUD for Employee

DB:

create table tblEmployee(

id int identity(1,1) primary key,

name varchar(20),

age int

)

create proc proc\_GetAllEmployees

as

begin

select \* from tblEmployee

end

create proc proc\_AddNewEmployee (@name varchar(20), @age int)

as

begin

insert into tblEmployee (name, age) values (@name, @age)

end

create proc proc\_UpdateEmployeeAge(@id int, @age int)

as

begin

update tblEmployee set age = @age where id = @id

end

create proc proc\_RemoveEmployeeById(@id int)

as

begin

DELETE FROM tblEmployee WHERE id = @id;

end

Code:

class Employee

{

public int Id { get; set; }

public string Name { get; set; }

public int Age { get; set; }

public override string ToString()

{

return "Id: " + Id

+ "\nName: " + Name

+ "\nAge: " + Age;

}

}

public Employee GetEmployeeDetails()

{

Employee employee = new Employee();

Console.WriteLine("Please enter your preferred name");

employee.Name = Console.ReadLine();

Console.WriteLine("Please enter your age");

employee.Age = Convert.ToInt32(Console.ReadLine());

return employee;

}

public void AddEmployee()

{

Employee employee = GetEmployeeDetails();

cmdInsert = new SqlCommand("proc\_AddNewEmployee", conn);

cmdInsert.CommandType = CommandType.StoredProcedure;

try

{

cmdInsert.Parameters.AddWithValue("@name", employee.Name);//together

cmdInsert.Parameters.AddWithValue("@age", employee.Age);

conn.Open();

int result = cmdInsert.ExecuteNonQuery();

if (result > 0)

Console.WriteLine("Employee is added");

}

catch (SqlException se)

{

Console.WriteLine("Could not insert");

Console.WriteLine(se.Message);

Console.WriteLine("---------------------------------------");

Console.WriteLine(se.StackTrace);

Console.WriteLine("---------------------------------------");

}

catch (Exception e)

{

Console.WriteLine("Could not insert-some error");

Console.WriteLine(e.Message);

Console.WriteLine("---------------------------------------");

Console.WriteLine(e.StackTrace);

Console.WriteLine("---------------------------------------");

}

finally

{

conn.Close();

}

}

public void RetrieveAllEmployees()

{

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

cmdSelect = new SqlCommand("proc\_GetAllEmployees", conn);

cmdSelect.CommandType = CommandType.StoredProcedure;

try

{

conn.Open();

IDataReader reader = cmdSelect.ExecuteReader();

while (reader.Read())

{

Employee employee = new Employee();

employee.Id = Convert.ToInt32(reader[0].ToString());

employee.Name = reader[1].ToString();

employee.Age = Convert.ToInt32(reader[2].ToString());

employees.Add(employee);

}

PrintAllEmployees(employees);

}

catch (SqlException se)

{

Console.WriteLine("Could not retrieve employees");

Console.WriteLine(se.Message);

Console.WriteLine("---------------------------------------");

Console.WriteLine(se.StackTrace);

Console.WriteLine("---------------------------------------");

}

catch (Exception e)

{

Console.WriteLine("Could not select-some error");

Console.WriteLine(e.Message);

Console.WriteLine("---------------------------------------");

Console.WriteLine(e.StackTrace);

Console.WriteLine("---------------------------------------");

}

finally

{

conn.Close();

}

}

public Employee GetEmployeeAgeForUpdate()

{

Employee employee = new Employee();

Console.WriteLine("Please enter your employee id");

employee.Id = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Please enter your age to change to");

employee.Age = Convert.ToInt32(Console.ReadLine());

return employee;

}

public void UpdateEmployeeAge()

{

Employee employee = GetEmployeeAgeForUpdate();

cmdUpdate = new SqlCommand("proc\_UpdateEmployeeAge", conn);

cmdUpdate.CommandType = CommandType.StoredProcedure;

try

{

cmdUpdate.Parameters.AddWithValue("@id", employee.Id);

cmdUpdate.Parameters.AddWithValue("@age", employee.Age);

conn.Open();

int result = cmdUpdate.ExecuteNonQuery();

if (result > 0)

Console.WriteLine("Employee age is updated");

}

catch (SqlException se)

{

Console.WriteLine("Could not update employee age");

Console.WriteLine(se.Message);

Console.WriteLine("---------------------------------------");

Console.WriteLine(se.StackTrace);

Console.WriteLine("---------------------------------------");

}

catch (Exception e)

{

Console.WriteLine("Could not update-some error");

Console.WriteLine(e.Message);

Console.WriteLine("---------------------------------------");

Console.WriteLine(e.StackTrace);

Console.WriteLine("---------------------------------------");

}

finally

{

conn.Close();

}

}

public Employee GetEmployeeIDToRemove()

{

Employee employee = new Employee();

Console.WriteLine("Please enter employee id to remove");

employee.Id = Convert.ToInt32(Console.ReadLine());

return employee;

}

public void RemoveEmployee()

{

Employee employee = GetEmployeeIDToRemove();

cmdDelete = new SqlCommand("proc\_RemoveEmployeeById", conn);

cmdDelete.CommandType = CommandType.StoredProcedure;

try

{

cmdDelete.Parameters.AddWithValue("@id", employee.Id);

conn.Open();

int result = cmdDelete.ExecuteNonQuery();

if (result > 0)

Console.WriteLine("Employee has been deleted");

}

catch (SqlException se)

{

Console.WriteLine("Could not delete employee");

Console.WriteLine(se.Message);

Console.WriteLine("---------------------------------------");

Console.WriteLine(se.StackTrace);

Console.WriteLine("---------------------------------------");

}

catch (Exception e)

{

Console.WriteLine("Could not delete-some error");

Console.WriteLine(e.Message);

Console.WriteLine("---------------------------------------");

Console.WriteLine(e.StackTrace);

Console.WriteLine("---------------------------------------");

}

finally

{

conn.Close();

}

}