Assignment - 7

using System;

using System.Collections.Generic;

using System.Configuration;

using System.Linq;

using System.Data.SqlClient;

using System.Text;

using System.Threading.Tasks;

using System.Text.RegularExpressions;

using System.Data;

namespace ADOSample

{

class ExampleQueries

{

string restaurantName;

string address;

long phoneNumber;

string mailId;

string connection = ConfigurationManager.ConnectionStrings["SampleDB"].ConnectionString;

static void Main(string[] args)

{

ExampleQueries exampleQueries = new ExampleQueries();

Console.WriteLine("Enter your choice 1. Insert 2. Update 3. Delete 4. Display 5. Count");

int choice = Convert.ToInt32(Console.ReadLine());

switch(choice)

{

case 1: exampleQueries.InsertCommand();break;

case 2:

exampleQueries.UpdateCommand();break;

case 3: exampleQueries.DeleteRecord();break;

case 4:

{

//exampleQueries.DisplayRecords();

exampleQueries.DisplayUsingDataset();

}

break;

case 5:exampleQueries.Displaycount();break;

}

Console.ReadLine();

}

public void InsertCommand()

{

while (true)

{

try

{

Console.WriteLine("Enter the Restaurant name");

restaurantName = Console.ReadLine();

if (restaurantName != null)

break;

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

}

while (true)

{

try

{

Console.WriteLine("Enter the Restaurant address");

address = Console.ReadLine();

if (address != null)

break;

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

}

while (true)

{

try

{

Console.WriteLine("Enter the phone number");

string number = Console.ReadLine();

Regex regex = new Regex(@"^(\+?\d{1,4}[\s-])?(?!0+\s+,?$)\d{10}\s\*,?$");

if (regex.IsMatch(number))

{

phoneNumber = Convert.ToInt64(number);

break;

}

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

}

while (true)

{

try

{

Console.WriteLine("Enter the mail id");

mailId = Console.ReadLine();

Regex regex = new Regex("^[a-z0-9][-a-z0-9.\_]+@([-a-z0-9]+.)+[a-z]{2,5}$");

if (regex.IsMatch(mailId))

{

break;

}

}

catch (Exception ex)

{

Console.WriteLine(ex.Message);

}

}

using (SqlConnection sqlConnection = new SqlConnection(connection))

{

try

{

string query = "Sp\_InsertRestaurant";

SqlCommand sqlCommand = new SqlCommand(query, sqlConnection);

sqlCommand.CommandType = CommandType.StoredProcedure;

sqlCommand.Parameters.AddWithValue("@RestaurantName", restaurantName);

sqlCommand.Parameters.AddWithValue("@RestaurantAddress", address);

sqlCommand.Parameters.AddWithValue("@PhoneNumber", phoneNumber);

sqlCommand.Parameters.AddWithValue("@MailId", mailId);

sqlConnection.Open();

Console.WriteLine("Number of rows inserted : " + sqlCommand.ExecuteNonQuery());

}

catch (SqlException ex)

{

Console.WriteLine("Error generated details " + ex.Message);

}

}

}

public void UpdateCommand()

{

Console.WriteLine("Enter the choice 1.Update name 2.Update address 3.Update Phone number 4. Update mailId");

int updateChoice = Convert.ToInt32(Console.ReadLine());

switch(updateChoice)

{

case 1:

{

using (SqlConnection sqlConnection = new SqlConnection(connection))

{

string query = "sp\_UpdateRestaurantName";

SqlCommand sqlCommand = new SqlCommand(query, sqlConnection);

sqlCommand.CommandType = CommandType.StoredProcedure;

Console.WriteLine("Enter the restaurant name and restaurant id");

sqlCommand.Parameters.AddWithValue("@RestaurantName", Console.ReadLine());

sqlCommand.Parameters.AddWithValue("@RestaurantId", Convert.ToInt32(Console.ReadLine()));

try

{

sqlConnection.Open();

Console.WriteLine("Updted successfully" + sqlCommand.ExecuteNonQuery());

}

catch (SqlException ex)

{

Console.WriteLine("Error generated" + ex.Message);

}

}

}break;

case 2:

{

using (SqlConnection sqlConnection = new SqlConnection(connection))

{

string query = "sp\_UpdateRestaurantAddress";

SqlCommand sqlCommand = new SqlCommand(query, sqlConnection);

sqlCommand.CommandType = CommandType.StoredProcedure;

Console.WriteLine("Enter the restaurant address and restaurant id");

sqlCommand.Parameters.AddWithValue("@RestaurantAddress", Console.ReadLine());

sqlCommand.Parameters.AddWithValue("@RestaurantId", Convert.ToInt32(Console.ReadLine()));

try

{

sqlConnection.Open();

Console.WriteLine("Updted successfully" + sqlCommand.ExecuteNonQuery());

}

catch (SqlException ex)

{

Console.WriteLine("Error generated" + ex.Message);

}

}

}break;

case 3:

{

using (SqlConnection sqlConnection = new SqlConnection(connection))

{

string query = "sp\_UpdatePhoneNUmber";

SqlCommand sqlCommand = new SqlCommand(query, sqlConnection);

sqlCommand.CommandType = CommandType.StoredProcedure;

Console.WriteLine("Enter the phone number and restaurant id");

sqlCommand.Parameters.AddWithValue("@PhoneNumber", Console.ReadLine());

sqlCommand.Parameters.AddWithValue("@RestaurantId", Convert.ToInt32(Console.ReadLine()));

try

{

sqlConnection.Open();

Console.WriteLine("Updted successfully" + sqlCommand.ExecuteNonQuery());

}

catch (SqlException ex)

{

Console.WriteLine("Error generated" + ex.Message);

}

}

}break;

case 4:

{

using (SqlConnection sqlConnection = new SqlConnection(connection))

{

string query = "sp\_UpdateMailId";

SqlCommand sqlCommand = new SqlCommand(query, sqlConnection);

sqlCommand.CommandType = CommandType.StoredProcedure;

Console.WriteLine("Enter the restaurant mail id and restaurant id");

sqlCommand.Parameters.AddWithValue("@RestaurantAddress", Console.ReadLine());

sqlCommand.Parameters.AddWithValue("@RestaurantId", Convert.ToInt32(Console.ReadLine()));

try

{

sqlConnection.Open();

Console.WriteLine("Updted successfully" + sqlCommand.ExecuteNonQuery());

}

catch (SqlException ex)

{

Console.WriteLine("Error generated" + ex.Message);

}

}break;

}

}

}

public void DeleteRecord()

{

using (SqlConnection sqlConnection = new SqlConnection(connection))

{

string query = "sp\_DeleteRestaurant";

SqlCommand sqlCommand = new SqlCommand(query, sqlConnection);

sqlCommand.CommandType = CommandType.StoredProcedure;

Console.WriteLine("Enter the restaurant id to delete");

sqlCommand.Parameters.AddWithValue("@RestaurantID", Convert.ToInt32(Console.ReadLine()));

try

{

sqlConnection.Open();

Console.WriteLine("Row deleted successfully" + sqlCommand.ExecuteNonQuery());

}

catch (SqlException ex)

{

Console.WriteLine("Error generated" + ex.Message);

}

}

}

public void DisplayRecords()

{

using (SqlConnection sqlConnection = new SqlConnection(connection))

{

string query = "select \* from Restaurant";

SqlCommand sqlCommand = new SqlCommand(query, sqlConnection);

try

{

sqlConnection.Open();

SqlDataReader sqlDataReader = sqlCommand.ExecuteReader();

if (sqlDataReader.HasRows)

{

while (sqlDataReader.Read())

{

Console.WriteLine("{0}\t{1}\t{2}\t{3}\t{4}",sqlDataReader.GetInt32(0),sqlDataReader.GetString(1), sqlDataReader.GetString(2), sqlDataReader.GetInt64(3), sqlDataReader.GetString(4));

}

}

sqlDataReader.Close();

}

catch (SqlException ex)

{

Console.WriteLine("Error generated" + ex.Message);

}

}

}

public void DisplayUsingDataset()

{

using (SqlConnection sqlConnection = new SqlConnection(connection))

{

try

{

sqlConnection.Open();

string query = "sp\_Display";

SqlDataAdapter sqlDataAdapter = new SqlDataAdapter(query, sqlConnection);

DataSet dataSet = new DataSet();

sqlDataAdapter.Fill(dataSet);

foreach (DataTable dataTable in dataSet.Tables)

{

foreach (DataColumn dataColumn in dataTable.Columns)

{

Console.Write(dataColumn.ColumnName + "\t");

}

Console.WriteLine();

foreach (DataRow dataRow in dataTable.Rows)

{

Console.WriteLine(dataRow["RestaurantID"] + "\t" + dataRow["RestaurantName"] + "\t" + dataRow["RestaurantAddress"] + "\t" + dataRow["PhoneNumber"] + "\t" + dataRow["MailId"]);

}

}

}

catch (SqlException ex)

{

Console.WriteLine("Error generated" + ex.Message);

}

}

}

public void Displaycount()

{

using (SqlConnection sqlConnection = new SqlConnection(connection))

{

try

{

sqlConnection.Open();

SqlCommand sqlCommand = new SqlCommand("sp\_Count", sqlConnection);

sqlCommand.CommandType = CommandType.StoredProcedure;

Console.WriteLine("Total restaurants : " + sqlCommand.ExecuteScalar());

}

catch (SqlException ex)

{

Console.WriteLine("Error generated" + ex.Message);

}

}

}

}

}