**DATABASE CREATION**

use StoreManagement;

create table ProductData

(

product\_id nvarchar(10) primary key,

product\_name nvarchar(20),

quantity\_available int,

cost money

);

insert into ProductData values

('I100', 'milk', 100, 15),

('I101', 'chips', 50, 20),

('I102', 'eggs', 200, 10),

('I103', 'bread', 70, 35.75),

('I104', 'chocolates', 200, 45.5),

('I105', 'cookies', 300, 15.5),

('E100', 'shampoo', 30, 150),

('E101', 'tooth paste', 50, 100),

('E102', 'cosmetics', 250, 1050),

('E103', 'ice cubes', 50, 15),

('E104', 'vinegar', 50, 100),

('E105', 'hand sanitizers', 45, 75);

Driver Code

using System.Configuration;

class Program:ConfigurationSection

{

public static void Main(string[] args)

{

StoreManagement<object> sm = new StoreManagement<object>();

sm.Login();

}

}

Stub code

using System;

using System.Collections.Generic;

using System.Data.SqlClient;

using System.Data;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace DBConnect

{

internal class StoreManagement<T>

{

SqlConnection conn;

SqlDataAdapter da;

DataSet ds;

public void OpenConn()

{

string cnnstr = "data source=LAPTOP-BMALL679\\SQLEXPRESS;Initial catalog=StoreManagement;integrated security=SSPI";

conn = new SqlConnection(cnnstr);

try

{

conn.Open();

Console.WriteLine("Connection Established");

}

catch (SqlException ex)

{

Console.WriteLine("Connection not established");

}

}

public void Login()

{

Console.WriteLine("Enter 1 for admin login or 2 for customer login");

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

switch (ch)

{

case 1:

string retry = "yes";

while (retry == "yes")

{

Console.WriteLine("Admin Login");

Console.WriteLine("Enter admin userid and password");

string uid = Console.ReadLine();

string pass = Console.ReadLine();

if ((uid == "admin") && (pass == "admin"))

{

AdminManipluation();

retry= "no";

}

else

{

Console.WriteLine("Incorrect userid/password");

}

}

break;

default:

Console.WriteLine("Enter option 1 or 2");

break;

}

}

public void InsertRecord(string product\_id, string product\_name, int quantity\_available, double cost)

{

da = new SqlDataAdapter("select \* from ProductData", conn);

ds = new DataSet();

da.Fill(ds, "sd");

SqlCommandBuilder scb = new SqlCommandBuilder(da);

//ds.Tables["pd"].Columns["product\_id"].Unique = true;

DataRow drow = ds.Tables["sd"].NewRow();

drow[0] = product\_id; drow[1] = product\_name; drow[2] = quantity\_available; drow[3] = cost;

ds.Tables["sd"].Rows.Add(drow);

da.Update(ds, "sd");

Console.WriteLine("Inserted");

conn.Close();

}

public void UpdateRecord<T>(string product\_id, T var)

{

da = new SqlDataAdapter("select \* from ProductData", conn);

ds = new DataSet();

da.Fill(ds, "sd");

SqlCommandBuilder scb = new SqlCommandBuilder(da);

foreach (DataRow dr in ds.Tables["sd"].Rows)

{

if ((dr["product\_id"].ToString()) == product\_id)

{

if (typeof(T) == typeof(string))

dr["product\_name"] = var;

else if (typeof(T) == typeof(int))

dr["quantity\_available"] = var;

else

dr["cost"] = var;

}

}

da.Update(ds, "sd");

Console.WriteLine("Updated");

conn.Close();

}

public void DeleteRecord(string product\_id)

{

da = new SqlDataAdapter("select \* from ProductData", conn);

ds = new DataSet();

da.Fill(ds, "sd");

SqlCommandBuilder scb = new SqlCommandBuilder(da);

foreach (DataRow dr in ds.Tables["sd"].Rows)

{

if (dr["product\_id"].ToString() == product\_id)

{

dr.Delete();

break;

}

}

da.Update(ds, "sd");

Console.WriteLine("Deleted");

conn.Close();

}

public void AdminManipluation()

{

bool exit = false;

while (exit != true)

{

Console.WriteLine("1. Display products \t2. Insert products \t3. Update products \t4. Delete products \t5.exit");

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

switch (ch)

{

case 1:

OpenConn();

da = new SqlDataAdapter("select \* from ProductData", conn);

ds = new DataSet();

da.Fill(ds, "pd");

foreach (DataRow drow in ds.Tables["pd"].Rows)

{

Console.WriteLine(drow["product\_id"].ToString() + drow["product\_name"].ToString() + drow["quantity\_available"] + drow["cost"]);

}

conn.Close();

break;

case 2:

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

string product\_id = Console.ReadLine();

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

string product\_name = Console.ReadLine();

Console.WriteLine("Enter the quantity available");

int quantity\_available = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("Enter the cost");

double cost = Convert.ToDouble(Console.ReadLine());

OpenConn();

InsertRecord(product\_id, product\_name, quantity\_available, cost);

break;

case 3:

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

string product\_id1 = Console.ReadLine();

Console.WriteLine("Enter the field to be updated");

Console.WriteLine("1. Product name 2. Quantity 3. cost");

OpenConn();

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

switch(field)

{

case 1:

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

string prod\_name\_update = Console.ReadLine();

UpdateRecord(product\_id1, prod\_name\_update);

break;

case 2:

Console.WriteLine("Enter the product quantity");

int prod\_quantity\_update = Convert.ToInt32(Console.ReadLine());

UpdateRecord(product\_id1, prod\_quantity\_update);

break;

case 3:

Console.WriteLine("Enter the product cost");

double prod\_cost\_update = Convert.ToDouble(Console.ReadLine());

UpdateRecord(product\_id1, prod\_cost\_update);

break;

}

break;

case 4:

Console.WriteLine("Enter the product id to be deleted");

string product\_id\_del = Console.ReadLine();

OpenConn();

DeleteRecord(product\_id\_del);

break;

case 5:

exit = true;

break;

default:

Console.WriteLine("Give proper value");

exit = true;

break;

}

}

}

}

}