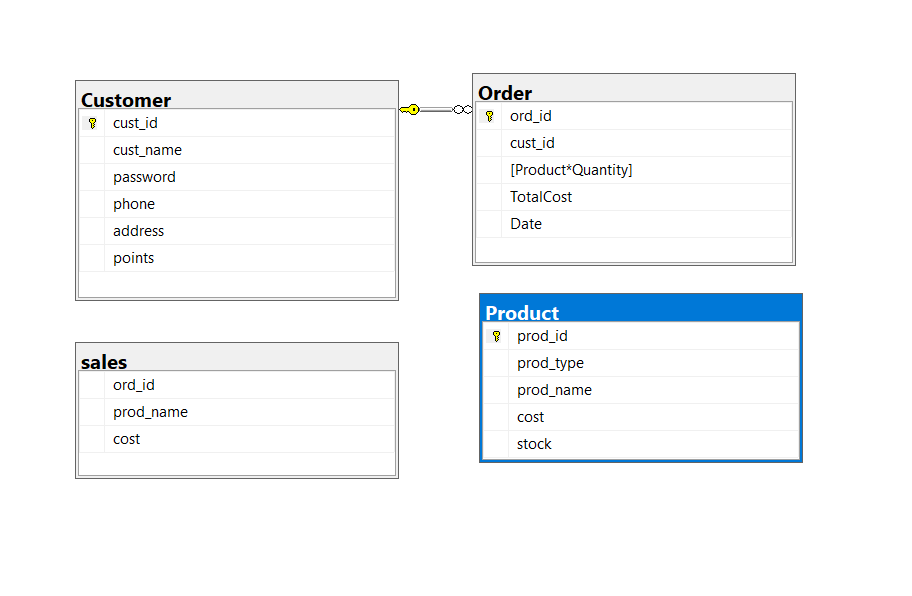
**ONLINE SHOPPING MART**

**SCHEMA:**

****

using csharpADO;

using System.Security.Cryptography.X509Certificates;

class program

{

public static void Main(string[] args)

{

string cnnstr = "Data source=ABI-A\\SQLEXPRESS; database = ado; integrated security = SSPI";

ADO ado = new ADO();

ado.OpenConn(cnnstr);

int n = 0;

bool result = true;

Console.WriteLine("To Exit Enter 1 \n" +

"To Create Account Enter 2\n" +

"To Place A Order Enter 3\n");

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

while (result)

{

if ( value == 1 )

{

ado.CloseConn();

result= false;

}

else if(value == 2 )

{

Console.WriteLine("Enter your name: ");

string name = Console.ReadLine();

Console.WriteLine("Enter the password: ");

string password = Console.ReadLine();

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

long phone\_num = Convert.ToInt64(Console.ReadLine());

Console.WriteLine("Enter address: ");

string address = Console.ReadLine();

ado.create\_account(name ,password, phone\_num , address);

value = 4;

}

else if(value == 3)

{

Console.WriteLine("Before place a order Login your Account: ");

Console.Write("Enter CustomerId: ");

string customerId = Console.ReadLine();

Console.Write("Enter Password: ");

string password = Console.ReadLine();

value = ado.Check(customerId, password);

Console.WriteLine(value);

}

else if(value == 4)

{

ado.ProductTypes();

Console.Write("Enter the Product type: ");

string type = Console.ReadLine();

ado.SearchByType(type);

Console.WriteLine("Enter the Product Id to be purchased: ");

int prod\_id = int.Parse(Console.ReadLine());

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

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

ado.sales(prod\_id , quantity);

Console.WriteLine("Enter 1 to generate bill \n Enter 2 to make another purchase \n");

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

if (choice == 1)

{

value = 5;

}

}

else if (value == 5)

{

ado.Billing();

Console.WriteLine("Enter 1 to exit");

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

}

}

}

}

using System;

using System.Collections.Generic;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Collections.Generic;

using System.Collections;

namespace csharpADO

{

internal class ADO

{

SqlConnection Conn;

string prod\_name;

int cost = 0;

int prod\_id;

public void OpenConn(string cnnstr)

{

Conn = new SqlConnection(cnnstr);

Conn.Open();

Console.WriteLine("Opened");

}

public void create\_account(string name,string password, long phone\_no , string address)

{

SqlCommand cmd = new SqlCommand($"insert into Customer values('{name}','{password}',{phone\_no},'{address}',0)", Conn);

cmd.ExecuteNonQuery();

Console.WriteLine("Account Create Successfully");

}

public int Check(string Id , string password)

{

int sts;

int value;

SqlCommand cmd = new SqlCommand($"Select count(cust\_id) from customer where cust\_id='{Id}' and password='{password}'",Conn);

SqlDataReader sdr;

sdr = cmd.ExecuteReader();

while (sdr.Read())

{

sts = (int)sdr[0];

if (sts == 0)

{

Console.WriteLine("Enter the valid userid password");

value = 3;

sdr.Close();

return value;

}

}

value = 4;

sdr.Close();

return value;

}

public void ProductTypes()

{

SqlCommand cmd = new SqlCommand("Select distinct(prod\_type) from product;",Conn);

SqlDataReader sdr;

sdr = cmd.ExecuteReader();

if (sdr.HasRows)

{

while (sdr.Read())

{

Console.WriteLine(sdr[0]);

}

}

sdr.Close();

}

public void SearchByType(string type)

{

SqlCommand cmd = new SqlCommand($"Select prod\_name, cost, stock , prod\_id from product where prod\_type = '{type}'",Conn);

SqlDataReader sdr;

sdr = cmd.ExecuteReader();

if (sdr.HasRows)

{

while (sdr.Read())

{

Console.WriteLine($"Product Name: {sdr[0]} \t Product Id: {sdr[3]} \nCost per unit: {sdr[1]} \nAvailable Stocks: {sdr[2]}");

}

}

else

{

Console.WriteLine("No products in the selected Product type");

}

sdr.Close();

}

public void SearchById(int id)

{

SqlCommand cmd = new SqlCommand($"Select prod\_name, cost, stock from product where prod\_id ={id}", Conn);

SqlDataReader sdr;

sdr = cmd.ExecuteReader();

while (sdr.Read())

{

Console.WriteLine($"Product Name: {sdr[0]} \nCost per unit: {sdr[1]} \nAvailable Stocks: {sdr[2]}");

}

sdr.Close();

}

public void sales(int prod\_id, int quantity)

{

this.prod\_id = prod\_id;

SqlCommand cmd = new SqlCommand($"Select prod\_name , cost from product where prod\_id = {prod\_id};", Conn);

SqlDataReader sdr;

sdr = cmd.ExecuteReader();

sdr.Read();

prod\_name = sdr[0].ToString();

Console.WriteLine($"Prod\_name : {prod\_name}");

cost = (int)(decimal)sdr[1];

Console.WriteLine($"cost : {cost}");

sdr.Close();

salesadd(quantity);

}

public void salesadd(int quantity)

{

SqlCommand cmd1 = new SqlCommand($"Insert into sales (prod\_name,cost) values('{prod\_name} \* {quantity}',{cost\*quantity})",Conn);

cmd1.ExecuteNonQuery();

Console.WriteLine("Row in sales Table added");

ProdUpd(quantity);

}

public void ProdUpd(int quantity)

{

SqlCommand cmd = new SqlCommand($"update Product set stock = (select stock from Product where prod\_id = {prod\_id} ) - {quantity} where prod\_id = {prod\_id};",Conn);

cmd.ExecuteNonQuery();

Console.WriteLine("row updated");

}

public void Billing()

{

SqlCommand cmd = new SqlCommand($" Select prod\_name , cost from sales", Conn);

SqlDataReader sdr;

sdr = cmd.ExecuteReader();

if (!sdr.HasRows)

{

while (sdr.Read())

{

Console.WriteLine($"Product \* Quantity : {sdr[0]}, Cost : {sdr[1]}");

}

}

sdr.Close();

BillingFinal();

}

public void BillingFinal()

{

SqlCommand cmd = new SqlCommand($"Select sum(cost) from sales",Conn);

SqlDataReader sdr = cmd.ExecuteReader();

sdr.Read();

Console.WriteLine($"Total Cost : {sdr[0]}");

sdr.Close();

}

public void CloseConn()

{

SqlCommand cmd = new SqlCommand("delete from sales",Conn);

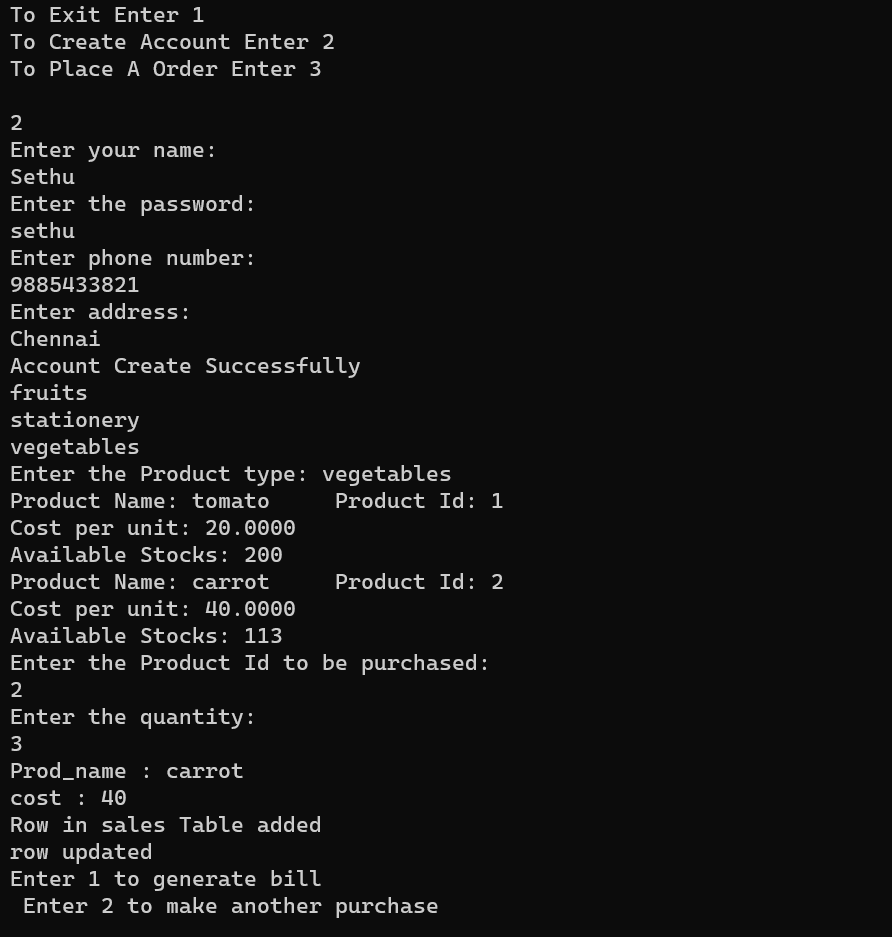
cmd.ExecuteNonQuery();

Conn.Close();

Console.WriteLine("Closed");

}

}

****}

