Driver Code

using ADO\_Assessment;

using System.Data.SqlClient;

using System.Text;

class program

{

public static void Main(string[] args)

{

StringBuilder cnnstr = new StringBuilder("Data source=");

Console.WriteLine("Enter Data Source");

cnnstr.Append(Console.ReadLine());

cnnstr.Append(";Initial catalog=");

Console.WriteLine("Enter Database Name");

cnnstr.Append(Console.ReadLine());

cnnstr.Append(";Integrated Security=SSPI;");

Console.WriteLine(cnnstr);

bool n = true;

while (n)

{

Console.WriteLine("To exit Enter 0");

Console.WriteLine("Enter the query number to be executed:");

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

if (num == 0)

{

break;

}

switch (num)

{

case 1:

{

Console.WriteLine("qstn1");

Qstn1 qstn1 = new Qstn1(cnnstr.ToString());

qstn1.ReadData();

break;

}

case 2:

{

Console.WriteLine("qstn2");

Qstn2 qstn2 = new Qstn2(cnnstr.ToString());

qstn2.ReadData();

break;

}

case 3:

{

Console.WriteLine("qstn3");

Qstn3 qstn3 = new Qstn3(cnnstr.ToString());

qstn3.ReadData();

break;

}

case 4:

{

Console.WriteLine("qstn4");

Qstn4 qstn4 = new Qstn4(cnnstr.ToString());

qstn4.ReadData();

break;

}

case 5:

{

Console.WriteLine("qstn5");

Qstn5 qstn5 = new Qstn5(cnnstr.ToString());

qstn5.ReadData();

break;

}

default:

{

Console.WriteLine("Enter the correct key");

break;

}

}

}

}

}

QUESTION 1:

using System.Data.SqlClient;

namespace ADO\_Assessment

{

internal class Qstn1 : Base

{

string cnnstr;

public Qstn1(string cnnstr)

{

this.Cnnstr = cnnstr;

}

public string Cnnstr { get => cnnstr; set => cnnstr = value; }

public void ReadData()

{

OpenConn(Cnnstr);

SqlCommand cmd = new SqlCommand("Select S.member\_id,sum(m.cost) from sales S join menu M ON S.product\_id = M.product\_id group by member\_id", Conn);

SqlDataReader sdr;

if (N != 1) {

sdr = cmd.ExecuteReader();

if (!sdr.HasRows)

{

Console.WriteLine("Table is Empty");

}

else

{

Console.WriteLine("Member\_Id Total Cost Spent");

while (sdr.Read())

{

Console.WriteLine($" {sdr[0]} {sdr[1]}");

}

}

CloseConn();

}

else

{

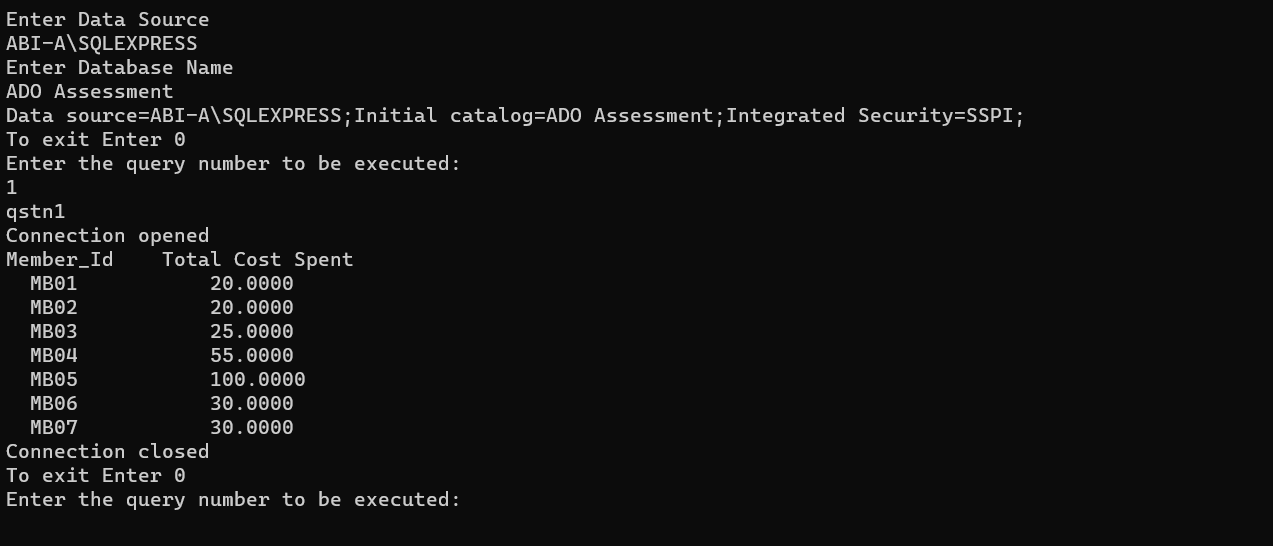
Console.WriteLine("Needs Connection to read data");

}

}

}

}



QUESTION 2:

using System;

using System.Collections.Generic;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ADO\_Assessment

{

internal class Qstn2 : Base

{

string cnnstr;

public Qstn2(string cnnstr)

{

this.Cnnstr = cnnstr;

}

public string Cnnstr { get => cnnstr; set => cnnstr = value; }

public void ReadData()

{

OpenConn(Cnnstr);

SqlCommand cmd = new SqlCommand("Select member\_id , count(distinct(date)) from sales group by member\_id;", Conn);

SqlDataReader sdr;

if (N != 1)

{

sdr = cmd.ExecuteReader();

if (!sdr.HasRows)

{

Console.WriteLine("Table is Empty");

}

else

{

Console.WriteLine("Member Id No of Visited Days");

while (sdr.Read())

{

Console.WriteLine($"{sdr[0]} {sdr[1]}");

}

}

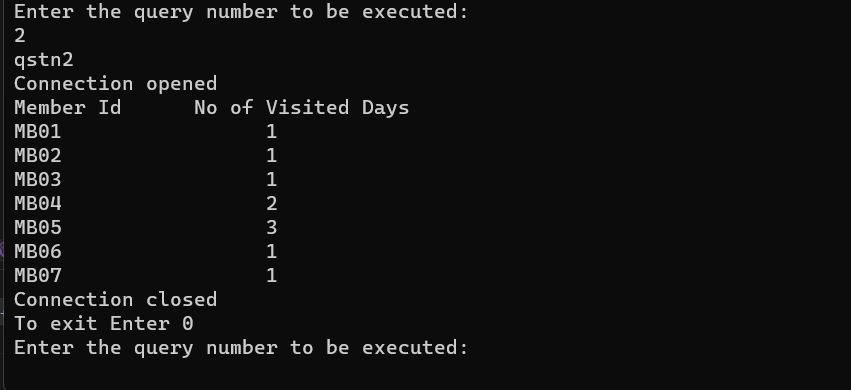
}

CloseConn();

}

}

}



QUESTION 3:

using System;

using System.Collections.Generic;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ADO\_Assessment

{

internal class Qstn3 : Base

{

string cnnstr;

public Qstn3(string cnnstr)

{

this.Cnnstr = cnnstr;

}

public string Cnnstr { get => cnnstr; set => cnnstr = value; }

public void ReadData()

{

OpenConn(Cnnstr);

SqlCommand cmd = new SqlCommand("Select top 1 product\_id , count(product\_id) As Count from sales group by product\_id order by product\_id desc;", Conn);

SqlDataReader sdr;

if (N != 1)

{

sdr = cmd.ExecuteReader();

if (!sdr.HasRows)

{

Console.WriteLine("Table is Empty");

}

else

{

Console.WriteLine("Product Id Amount Spent");

while (sdr.Read())

{

Console.WriteLine($"{sdr[0]} {sdr[1]}");

}

}

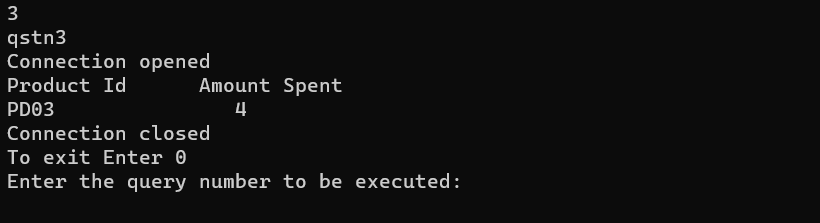
}

CloseConn();

}

}

}



QUESTION 4:

using System;

using System.Collections.Generic;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ADO\_Assessment

{

internal class Qstn4 : Base

{

string cnnstr;

public Qstn4(string cnnstr)

{

this.Cnnstr = cnnstr;

}

public string Cnnstr { get => cnnstr; set => cnnstr = value; }

public void ReadData()

{

OpenConn(Cnnstr);

SqlCommand cmd = new SqlCommand("Select S.member\_id , count(S.product\_id) As No\_of\_products, SUM(M.cost) As Amount from Sales S join Menu M ON S.product\_id = M.product\_id group by member\_id;", Conn);

SqlDataReader sdr;

if (N != 1)

{

sdr = cmd.ExecuteReader();

if (!sdr.HasRows)

{

Console.WriteLine("Table is Empty");

}

else

{

Console.WriteLine("Member Id Count of Product Total Cost");

while (sdr.Read())

{

Console.WriteLine($"{sdr[0]} {sdr[1]} {sdr[2]}");

}

}

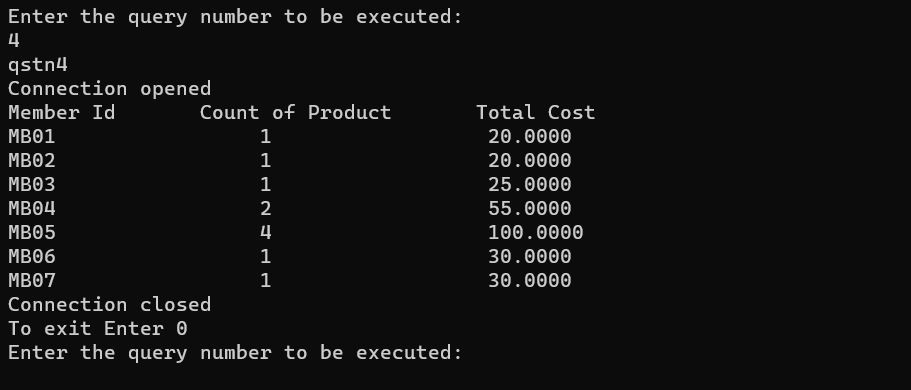
}

CloseConn();

}

}

}



QUESTION 5:

using System;

using System.Collections.Generic;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ADO\_Assessment

{

internal class Qstn5 : Base

{

string cnnstr;

public Qstn5(string cnnstr)

{

this.Cnnstr = cnnstr;

}

public string Cnnstr { get => cnnstr; set => cnnstr = value; }

public void ReadData()

{

OpenConn(Cnnstr);

SqlCommand cmd = new SqlCommand("Select S.member\_id , sum(cost)\*10 As Points from sales S join menu M on S.product\_id = M.product\_id group by S.member\_id;", Conn);

SqlDataReader sdr;

if (N != 1)

{

sdr = cmd.ExecuteReader();

if (!sdr.HasRows)

{

Console.WriteLine("Table is Empty");

}

else

{

Console.WriteLine("Member Id Points");

while (sdr.Read())

{

Console.WriteLine($"{sdr[0]} {sdr[1]}");

}

}

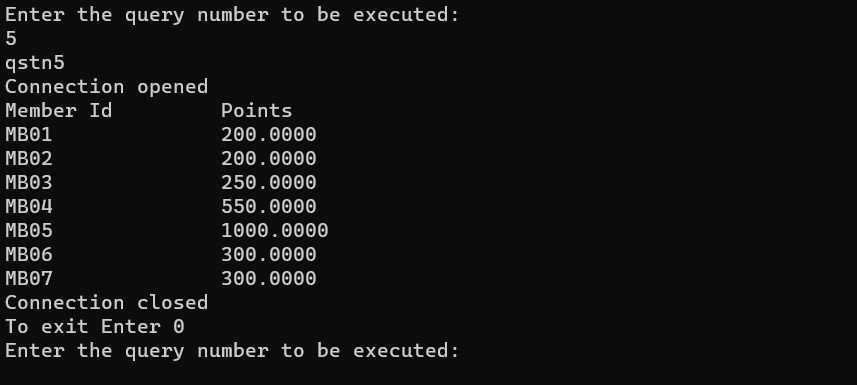
}

CloseConn();

}

}

}



Base Class:

using System;

using System.Collections.Generic;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace ADO\_Assessment

{

internal class Base

{

SqlConnection conn;

int n = 0;

public SqlConnection Conn { get => conn; set => conn = value; }

public int N { get => n; set => n = value; }

public void OpenConn(string cnnstr)

{

Conn = new SqlConnection(cnnstr);

try

{

Conn.Open();

Console.WriteLine("Connection opened");

}

catch (SqlException ex)

{

N = 1;

Console.WriteLine(ex.Message);

Console.WriteLine("Connection Not Established");

}

}

public void CloseConn()

{

Conn.Close();

Console.WriteLine("Connection closed");

}

public void CreateTable()

{

SqlCommand cmd = new SqlCommand("Create table menu (product\_id nvarchar(10) primary key,product\_name nvarchar(15) not null,cost money not null);", conn);

SqlCommand cmd1 = new SqlCommand("create table members (member\_id nvarchar(10) primary key,name nvarchar(40),address nvarchar(40));", conn);

SqlCommand cmd2 = new SqlCommand("create table sales (sales\_id nvarchar(10) primary key,member\_id nvarchar(10), product\_id nvarchar(10) FOREIGN KEY(member\_id) REFERENCES members(member\_id),FOREIGN KEY(product\_id) REFERENCES menu(product\_id))", conn);

}

public void InsertData()

{

SqlCommand cmd = new SqlCommand("insert into Members Values('MB01','Sathyan','krishnagiri');", conn);

SqlCommand cmd1 = new SqlCommand("insert into Menu values('PD01','Sushi',20.0000);", conn);

SqlCommand cmd2 = new SqlCommand("insert into Sales Values('S01','MB01','PD01','2021-01-02');", conn);

}

}

}