**NAME: Mahasri M**

**CASE STUDY 1**

**SQL:**

use adoass;

create table menu (p\_id varchar(1),p\_name varchar(10),price smallint,

CONSTRAINT PK\_PID PRIMARY KEY("p\_id"));

insert into menu values('A','sushi',100);

insert into menu values('B','curry',300);

insert into menu values('C','ramen',500);

select \* from menu;

create table sales (cus\_id int, p\_id varchar(1), purchase\_date date,amt int,

CONSTRAINT FK\_CUS\_ID FOREIGN KEY("cus\_id")

REFERENCES "dbo"."members" ("cus\_id"),

CONSTRAINT FK\_PID FOREIGN KEY("p\_id")

REFERENCES "dbo"."menu"("p\_id"),

);

insert into sales values(1,'A','2023/03/26',100);

insert into sales values(2,'B','2023/02/10',300);

insert into sales values(3,'C','2023/01/12',500);

insert into sales values(4,'B','2023/01/10',300);

insert into sales values(1,'A','2023/03/03',100);

insert into sales values(3,'C','2023/02/11',500);

insert into sales values(4,'A','2023/04/13',100);

select \* from sales;

create table members(cus\_id int primary key, c\_name varchar(5),date\_of\_join date, points\_earned tinyint);

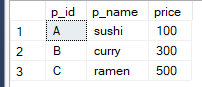
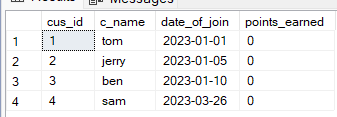
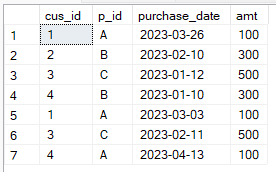
insert into members values(1,'tom','2023/01/01',0);

insert into members values(2,'jerry','2023/01/05',0);

insert into members values(3,'ben','2023/01/10',0);

insert into members values(4,'sam','2023/03/26',0);

select \* from members;

****

**C# code:**

using System;

using System.Collections.Generic;

using System.ComponentModel.Design;

using System.Data;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace DBConnect

{

internal class danny

{

SqlConnection conn;

SqlCommand table1, table2, table3;

SqlDataAdapter da;

DataSet ds;

public void OpenConn()

{

string cnnstr = "data source=LAPTOP-EE8FKNGK\\SQLEXPRESS; Initial Catalog = adoass; Integrated Security = SSPI";

conn = new SqlConnection(cnnstr);

try

{

conn.Open();

Console.WriteLine("DB Connected..");

}

catch (SqlException ex)

{

Console.WriteLine("Connection not established");

}

}

public void ReadingSales()

{

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

ds = new DataSet();

da.Fill(ds, "Sales");

Console.WriteLine("\nSales data:\n");

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

{

Console.WriteLine(dr[0].ToString() +" "+ dr[1].ToString()+" " + dr[2].ToString()+" " + dr[3].ToString());

}

}

public void ReadingMenu()

{

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

ds = new DataSet();

da.Fill(ds, "Menu");

Console.WriteLine("\nMenu data:\n");

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

{

Console.WriteLine(dr[0].ToString() + " " + dr[1].ToString() + " " + dr[2].ToString());

}

}

public void ReadingMembers()

{

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

ds = new DataSet();

da.Fill(ds, "Members");

Console.WriteLine("\nMembers data:\n");

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

{

Console.WriteLine(dr[0].ToString() + " " + dr[1].ToString() + " " + dr[2].ToString()+" " + dr[3].ToString());

}

}

public void fetch()

{

Console.WriteLine("\nSelect any numbers between 1-5:\n");

Console.WriteLine("1. Display the total amount each customer spent at the restaurant?\n" +

"2. Display the number of days each customer has visited the restaurant?\n" +

"3. Display the most purchased item on the menu\n" +

"4. Display the total items and amount spent by each member?\n"+

"5. If each $1 spent equates to 10 points display the points each customer has earned.\n");

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

if(number ==1) {

OpenConn();

Q1();

}

else if(number ==2)

{

OpenConn();

Q2();

}

else if (number ==3)

{

OpenConn();

Q3();

}

else if(number == 4)

{

OpenConn();

Q4();

}

else if(number == 5)

{

OpenConn();

Q5();

}

else

{

Console.WriteLine("Enter a valid number:");

}

}

public void Q1()

{

SqlCommand q1 = new SqlCommand("select C\_name, SUM(amt) from Members join Sales on Members.Cus\_id=Sales.Cus\_id group by C\_name;", conn);

if (conn != null)

{

SqlDataReader s = q1.ExecuteReader();

Console.WriteLine("Total amount each customer spent at the restaurant:\n");

while (s.Read())

{

Console.WriteLine(s[0] + " " + s[1]+ " "+"rupees");

}

}

conn.Close();

}

public void Q2()

{

SqlCommand q2 = new SqlCommand("select C\_name, count(distinct purchase\_date) from Sales join Members on Members.Cus\_id=Sales.Cus\_id group by C\_name;", conn);

if (conn != null)

{

SqlDataReader s = q2.ExecuteReader();

Console.WriteLine("Number of days each customer has visited the restaurant:\n");

while (s.Read())

{

Console.WriteLine(s[0] + " " + s[1]+" "+"days");

}

}

conn.Close();

}

public void Q3()

{

SqlCommand q3 = new SqlCommand("select Top 1 P\_name, sum(amt) from Sales join Menu on Menu.P\_id=Sales.P\_id group by P\_name ;", conn);

if (conn != null)

{

SqlDataReader s = q3.ExecuteReader();

Console.WriteLine("Most purchased item on the menu:\n");

while (s.Read())

{

Console.WriteLine(s[0] + " " + s[1]+" "+"rupees");

}

}

conn.Close();

}

public void Q4()

{

SqlCommand q4 = new SqlCommand("select C\_name,count(p\_id), SUM(amt) FROM Members JOIN Sales ON Members.Cus\_id = Sales.Cus\_id group by C\_name;", conn);

if (conn != null)

{

SqlDataReader s = q4.ExecuteReader();

Console.WriteLine("Total items and amount spent by each member:\n");

while (s.Read())

{

Console.WriteLine(s[0] + " " + s[1]+"items and"+" " + s[2]+" "+"rupees");

}

}

conn.Close();

}

public void Q5()

{

SqlCommand q5 = new SqlCommand("select C\_name, Sum(amt \* 10) from Sales join Members on Sales.Cus\_id=Members.Cus\_id group by C\_name;", conn);

if (conn != null)

{

SqlDataReader s = q5.ExecuteReader();

Console.WriteLine("Total points each customer has earned:\n");

while (s.Read())

{

Console.WriteLine(s[0] + " " + s[1] + "points");

}

}

conn.Close();

}

}

}

**DRIVER CODE:**

public static void Main(string[] args)

{

danny danny = new danny();

danny.OpenConn();

danny.ReadingMenu();

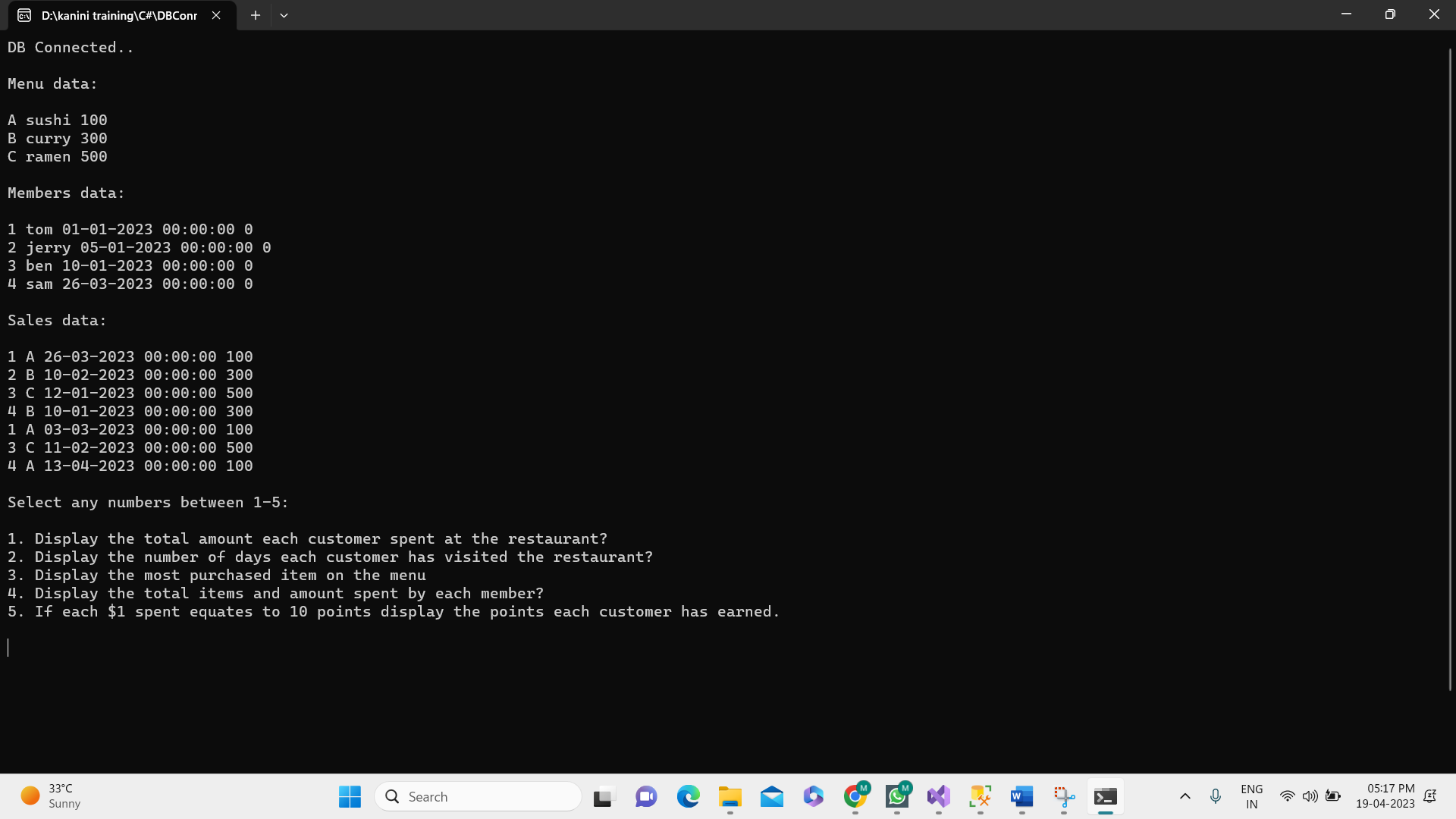
danny.ReadingMembers();

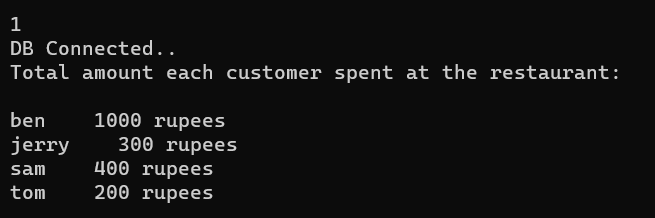
danny.ReadingSales();

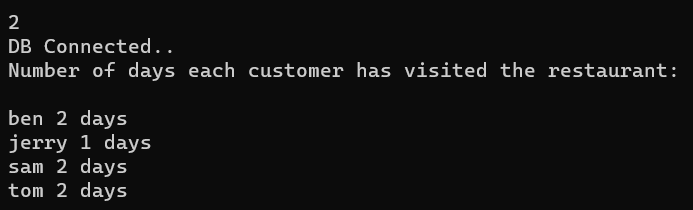
danny.fetch();

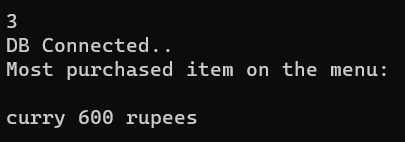
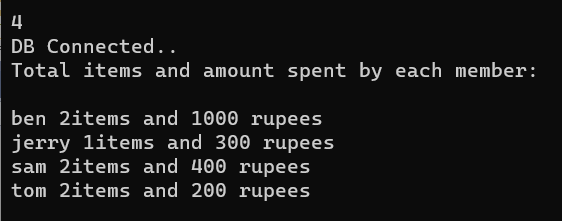
}

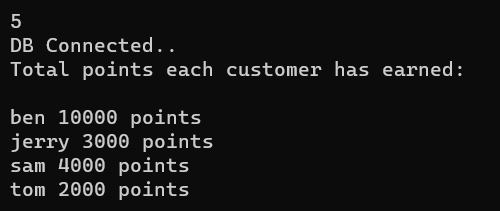
**OUTPUT:**



****

****

** **

****