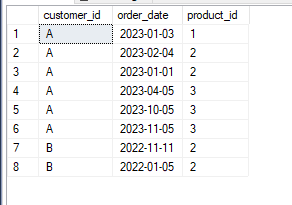
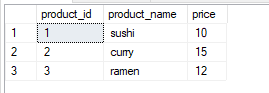
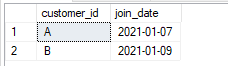
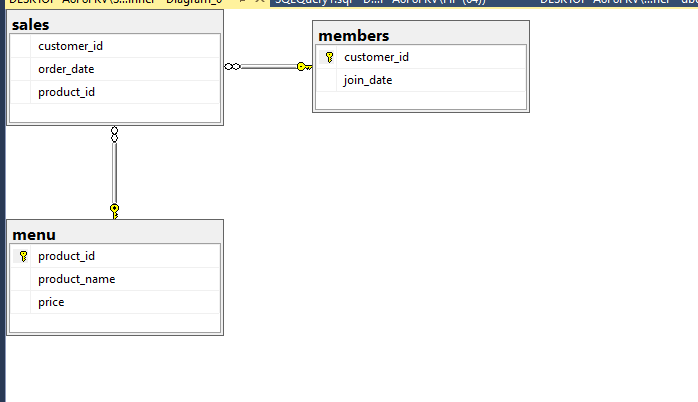
DANNY’S DINNER – CASE STUDY

-BY BHAGYALAKSHMI









Create the tables in SQL server with the needed columns and constraints.

Write appropriate methods in C# to perform the following operations.

DRIVER CODE

using Dannys\_dinner;

using System;

using System.Data.SqlClient;

using System.Text;

class Program

{

public static void Main(string[] args)

{

/\*StringBuilder sb = new StringBuilder("data source=");

Console.WriteLine("Enter Data Source");

sb.Append(Console.ReadLine());

Console.WriteLine("Initial Catalog");

sb.Append(Console.ReadLine());

sb.Append("Integrated security=SSPI");

Console.Write(sb);\*/

Danny d = new Danny();

d.Opencon();

d.CreateTables();

d.InsertRecords();

Console.WriteLine("QUESTION-1 :To Display the total amount each customer spent at the restaurant");

d.Qn1();

Console.WriteLine("QUESTION-2 :To Display the number of days each customer has visited the restaurant");

d.Qn2();

Console.WriteLine("QUESTION-3 :To Display the most purchased item on the menu");

d.Qn3();

Console.WriteLine("QUESTION-4 :To Display the total items and amount spent by each member");

d.Qn4();

Console.WriteLine("QUESTION-5 : If each $1 spent equates to 10 points display the points each customer has earned.");

d.Qn5();

//d.ReadTable();

d.CloseCon();

}

}

STUB CODE

using System;

using System.Collections.Generic;

using System.Data;

using System.Data.SqlClient;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

namespace Dannys\_dinner

{

class Danny

{

SqlConnection conn;

SqlCommand cmd;

SqlDataReader sdr;

SqlDataAdapter da;

DataSet ds;

DataRow dr;

public void Opencon()

{

conn = new SqlConnection("data source=DESKTOP-A8F8FKV\\SQLEXPRESS;initial catalog=dannys\_dinner;integrated security=SSPI");

//"datasource=DESKTOP-A8F8FKV\\SQLEXPRESS;initial catalog=dannys\_dinner;integrated security=SSPI"

try

{

conn.Open();

Console.WriteLine("connection opened");

}

catch(SqlException e) { }

{

Console.WriteLine("Connection not established");

}

}

public void CreateTables()

{

cmd = new SqlCommand("CREATE TABLE menu (product\_id INT primary key,product\_name NVARCHAR(5) not null,price INT not null);", conn);

cmd.ExecuteNonQuery();

Console.WriteLine("Menu table created");

cmd = new SqlCommand("CREATE TABLE members (customer\_id nvarchar(1) primary key,join\_date DATE not null);", conn);

cmd.ExecuteNonQuery();

Console.WriteLine("Members table created");

cmd = new SqlCommand("CREATE TABLE sales (customer\_id nVARCHAR(1) foreign key references members,order\_date DATE not null,product\_id INT foreign key references menu);", conn);

cmd.ExecuteNonQuery();

Console.WriteLine("Sales table created");

}

public void InsertRecords()

{

cmd = new SqlCommand("INSERT INTO menu VALUES ('1', 'sushi', '400'), ('2', 'curry', '150'), ('3', 'ramen', '500');",conn);

cmd.ExecuteNonQuery();

Console.WriteLine("Values inserted into MENU table");

cmd = new SqlCommand("INSERT INTO members (customer\_id, join\_date)VALUES ('A', '2001-01-07'), ('B', '2001-01-09');",conn);

cmd.ExecuteNonQuery();

Console.WriteLine("Values inserted into MEMBERS table");

cmd = new SqlCommand("INSERT INTO sales (customer\_id, order\_date, product\_id)VALUES ('A', '2023/01/03', '1'), ('A', '2023/02/04', '2'), ('A', '2023/01/01', '2'), ('A', '2023/04/05', '3'), ('A', '2023/10/5', '3'), ('A', '2023/11/05', '3'), ('B', '2022/11/11', '2'), ('B', '2022/1/5', '2');", conn);

cmd.ExecuteNonQuery();

Console.WriteLine("Values inserted into SALES table");

}

public void ReadTable()

{

cmd = new SqlCommand("select \* from menu;", conn);

sdr= cmd.ExecuteReader();

while(sdr.Read())

{

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

}

}

public void Qn1()

{

cmd = new SqlCommand("SELECT sales.customer\_id, SUM(menu.price) AS total\_spent FROM sales JOIN menu ON sales.product\_id = menu.product\_id GROUP BY customer\_id ORDER BY customer\_id;",conn);

if (conn!=null)

{

sdr = cmd.ExecuteReader();

if (!sdr.HasRows)

{

Console.WriteLine("Data is empty");

}

else

{

while (sdr.Read())

{

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

}

}

}

else { Console.WriteLine("NO connection"); }

}

public void Qn2()

{

cmd = new SqlCommand("SELECT customer\_id,COUNT (DISTINCT order\_date) AS visited\_days FROM sales GROUP BY customer\_id;", conn);

if (conn != null)

{

sdr = cmd.ExecuteReader();

if (!sdr.HasRows)

{

Console.WriteLine("Data is empty");

}

else

{

while (sdr.Read())

{

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

}

}

}

else

{

Console.WriteLine("No connection");

}

}

public void Qn3()

{

conn.Open();

da= new SqlDataAdapter("select Top 1 m.product\_name, count(s.product\_id) from menu m join sales s on m.product\_id = s.product\_id group by m.product\_name order by count(s.product\_id) desc;",conn);

ds = new DataSet();

da.Fill(ds,"qn3");

foreach(DataRow dr in ds.Tables["qn3"].Rows)

{

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

}

conn.Close();

}

public void Qn4()

{

conn.Open();

da = new SqlDataAdapter("select s.customer\_id, count(s.product\_id), sum(m.price) from sales s join menu m on m.product\_id = s.product\_id join members mem on mem.Customer\_id = s.customer\_id where s.order\_date > mem.join\_date group by s.customer\_id", conn);

da.Fill(ds, "qn4");

foreach (DataRow dr in ds.Tables["qn4"].Rows)

{

Console.WriteLine(dr["customer\_id"].ToString() + " " + dr[1].ToString()+" " + dr[2].ToString());

}

conn.Close();

}

public void Qn5()

{

conn.Open();

da = new SqlDataAdapter("select s.customer\_id, sum(price\*10) from sales s join menu m on m.product\_id = s.product\_id group by s.customer\_id", conn);

da.Fill(ds, "qn5");

foreach (DataRow dr in ds.Tables["qn5"].Rows)

{

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

}

conn.Close();

}

public void CloseCon()

{

conn.Close();

Console.WriteLine("Connection closed");

Console.ReadKey();

}

}

}