

APPLY THE CRUD OPERATIONS FOR PRESCRIPTION COST MANAGEMENT USING SQL SERVER, USE ARRAY LIST

Program.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace PrescriptionCrudOperation
{
    internal class Program
    {
        static void Main(string[] args)
        {
            PrescriptionMenu.Menu();
            Console.ReadKey();
        }
    }
}
```

PrescriptionCost.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace PrescriptionCrudOperation
{
    internal class PrescriptionCost
    {
        public int PrescriptionID { get; set; }
        public string PatientName { get; set; }
        public string Medication { get; set; }
        public double Cost { get; set; }

        public PrescriptionCost(int prescriptionID, string patientName, string medication, double cost)
        {
            PrescriptionID = prescriptionID;
            PatientName = patientName;
            Medication = medication;
            Cost = cost;
        }

        public override string ToString()
        {

```

```

        return $"PrescriptionID: {PrescriptionID}, PatientName:
{PatientName}, Medication: {Medication}, Cost: ${Cost:F2}";
    }
}

```

PrescriptionDAO.cs

```

using System;
using System.Collections.Generic;
using System.Data.SqlClient;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace PrescriptionCrudOperation
{
    internal class PrescriptionDAO
    {
        private string connectionString = "Data
Source=(localdb)\MSSQLLocalDB;Initial Catalog=Week4AssessmentDbCrud;Integrated
Security=True;";

        public void Create(PrescriptionCost prescriptionCost)
        {
            using (SqlConnection conn = new SqlConnection(connectionString))
            {
                string query = "INSERT INTO PrescriptionCost (PrescriptionID,
PatientName, Medication, Cost) VALUES (@PrescriptionID, @PatientName,
@Medication, @Cost)";
                SqlCommand cmd = new SqlCommand(query, conn);
                cmd.Parameters.AddWithValue("@PrescriptionID",
prescriptionCost.PrescriptionID);
                cmd.Parameters.AddWithValue("@PatientName",
prescriptionCost.PatientName);
                cmd.Parameters.AddWithValue("@Medication",
prescriptionCost.Medication);
                cmd.Parameters.AddWithValue("@Cost", prescriptionCost.Cost);

                conn.Open();
                cmd.ExecuteNonQuery();
            }
        }

        public PrescriptionCost Read(int PrescriptionID)
        {
            PrescriptionCost prescriptionCost = null;
            using (SqlConnection conn = new SqlConnection(connectionString))
            {
                string query = "SELECT PrescriptionID, PatientName, Medication,
Cost FROM PrescriptionCost WHERE PrescriptionID = @PrescriptionID";
                SqlCommand cmd = new SqlCommand(query, conn);
                cmd.Parameters.AddWithValue("@PrescriptionID", PrescriptionID);

                conn.Open();
                SqlDataReader reader = cmd.ExecuteReader();
                if (reader.Read())
                {

```

```

        prescriptionCost = new
PrescriptionCost((int)reader["PrescriptionID"],
                  reader["PatientName"].ToString(),
                  reader["Medication"].ToString(),
                  (double)reader["Cost"]);
    }
    }
    return prescriptionCost;
}

// Update a PrescriptionCost
public void Update(PrescriptionCost prescriptionCost)
{
    using (SqlConnection conn = new SqlConnection(connectionString))
    {
        string query = "UPDATE PrescriptionCost SET PatientName =
@PatientName, Medication = @Medication, Cost = @Cost WHERE PrescriptionID =
@PrescriptionID";
        SqlCommand cmd = new SqlCommand(query, conn);
        cmd.Parameters.AddWithValue("@PrescriptionID",
prescriptionCost.PrescriptionID);
        cmd.Parameters.AddWithValue("@PatientName",
prescriptionCost.PatientName);
        cmd.Parameters.AddWithValue("@Medication",
prescriptionCost.Medication);
        cmd.Parameters.AddWithValue("@Cost", prescriptionCost.Cost);

        conn.Open();
        cmd.ExecuteNonQuery();
    }
}

// Delete a PrescriptionCost by prescriptionID
public void Delete(int PrescriptionID)
{
    using (SqlConnection conn = new SqlConnection(connectionString))
    {
        string query = "DELETE FROM PrescriptionCost WHERE PrescriptionID
= @PrescriptionID";
        SqlCommand cmd = new SqlCommand(query, conn);
        cmd.Parameters.AddWithValue("@PrescriptionID", PrescriptionID);

        conn.Open();
        cmd.ExecuteNonQuery();
    }
}

// List all PrescriptionCost
public List<PrescriptionCost> ListAll()
{
    List<PrescriptionCost> prescriptionCosts = new
List<PrescriptionCost>();
    using (SqlConnection conn = new SqlConnection(connectionString))
    {
        string query = "SELECT PrescriptionID, PatientName, Medication,
Cost FROM PrescriptionCost";
        SqlCommand cmd = new SqlCommand(query, conn);

        conn.Open();
        SqlDataReader reader = cmd.ExecuteReader();
        while (reader.Read())
        {

```

```

        PrescriptionCost prescriptionCost = new
PrescriptionCost((int)reader["PrescriptionID"],
                  reader["PatientName"].ToString(),
                  reader["Medication"].ToString(),
                  (double)reader["Cost"]);
        prescriptionCosts.Add(prescriptionCost);
    }
}
return prescriptionCosts;
}
}
}

```

PrescriptionCostUI.cs

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace PrescriptionCrudOperation
{
    internal class PrescriptionCostUI
    {
        private PrescriptionDAO PrescriptionDAO = new PrescriptionDAO();

        public void CreatePrescription()
        {
            Console.WriteLine("Enter PrescriptionID: ");
            int PrescriptionID = int.Parse(Console.ReadLine());
            Console.WriteLine("Enter PatientName: ");
            string PatientName = Console.ReadLine();
            Console.WriteLine("Enter Medication: ");
            string Medication = Console.ReadLine();
            Console.WriteLine("Enter Cost: ");
            double Cost = double.Parse(Console.ReadLine());

            PrescriptionCost prescriptionCost = new
PrescriptionCost(PrescriptionID, PatientName, Medication, Cost);

            PrescriptionDAO.Create(prescriptionCost);

            Console.WriteLine("Prescription created successfully.");
        }

        public void ReadPrescription()
        {
            Console.WriteLine("Enter PrescriptionID: ");
            int PrescriptionID = int.Parse(Console.ReadLine());

            PrescriptionCost prescriptionCost =
PrescriptionDAO.Read(PrescriptionID);
            if (prescriptionCost != null)
            {
                Console.WriteLine($"PrescriptionID:
{prescriptionCost.PrescriptionID}");
            }
        }
    }
}

```

```

        Console.WriteLine($"PatientName:
{prescriptionCost.PatientName}");
        Console.WriteLine($"Medication: {prescriptionCost.Medication}");
        Console.WriteLine($"Cost: {prescriptionCost.Cost}");
    }
    else
    {
        Console.WriteLine("Prescription not found.");
    }
}

public void UpdatePrescription()
{
    Console.Write("Enter Prescription ID: ");
    int id = int.Parse(Console.ReadLine());

    PrescriptionCost prescriptionCost = PrescriptionDAO.Read(id);
    if (prescriptionCost != null)
    {
        Console.Write("Enter New PatientName: ");
        prescriptionCost.PatientName = Console.ReadLine();
        Console.Write("Enter New Medication: ");
        prescriptionCost.Medication = Console.ReadLine();
        Console.Write("Enter New Cost: ");
        prescriptionCost.Cost = double.Parse(Console.ReadLine());
        PrescriptionDAO.Update(prescriptionCost);
        Console.WriteLine("Prescription updated successfully.");
    }
    else
    {
        Console.WriteLine("Prescription not found.");
    }
}

public void DeletePrescription()
{
    Console.Write("Enter Prescription ID: ");
    int id = int.Parse(Console.ReadLine());

    PrescriptionDAO.Delete(id);
    Console.WriteLine("Prescription deleted successfully.");
}

public void ListAllPrescription()
{
    List<PrescriptionCost> prescriptions = PrescriptionDAO.ListAll();
    foreach (PrescriptionCost prescriptionCost in prescriptions)
    {
        Console.WriteLine($"ID: {prescriptionCost.PrescriptionID},
PatientName: {prescriptionCost.PatientName}, Medication:
{prescriptionCost.Medication}, Cost: {prescriptionCost.Cost}");
    }
}
}
}

```

PrescriptionMenu.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace PrescriptionCrudOperation
{
    internal class PrescriptionMenu
    {
        public static void Menu()
        {
            PrescriptionCostUI ui = new PrescriptionCostUI();
            bool running = true;

            while (running)
            {
                Console.WriteLine("\nPrescription Management System");
                Console.WriteLine("1. Create Prescription");
                Console.WriteLine("2. Read Prescription");
                Console.WriteLine("3. Update Prescription");
                Console.WriteLine("4. Delete Prescription");
                Console.WriteLine("5. List All Prescription");
                Console.WriteLine("6. Exit");

                Console.Write("Choose an option: ");
                string choice = Console.ReadLine();

                switch (choice)
                {
                    case "1":
                        ui.CreatePrescription();
                        break;
                    case "2":
                        ui.ReadPrescription();
                        break;
                    case "3":
                        ui.UpdatePrescription();
                        break;
                    case "4":
                        ui.DeletePrescription();
                        break;
                    case "5":
                        ui.ListAllPrescription();
                        break;
                    case "6":
                        running = false;
                        Console.WriteLine("Exiting...");
                        break;
                    default:
                        Console.WriteLine("Invalid choice. Please try again.");
                        break;
                }
            }
        }
    }
}
```

SQL QUERY

```
CREATE DATABASE Week4AssessmentDbCrud;
```

```
USE Week4AssessmentDbCrud;
```

```
CREATE TABLE PrescriptionCost (  
    PrescriptionID INT PRIMARY KEY,  
    PatientName NVARCHAR(100),  
    Medication NVARCHAR(100),  
    Cost FLOAT  
);
```

```
INSERT INTO PrescriptionCost  
(PrescriptionID, PatientName, Medication, Cost) VALUES  
(1, 'Rahul', 'Dolo 650', '120'),  
(2, 'Girish', 'Vicks', '56'),  
(3, 'Abijith', 'Halls', '30');
```

```
SELECT * FROM PrescriptionCost;
```

OUTPUT

C:\WINDOWS\system32\cmd.exe

```
6. Exit  
Choose an option: 1  
Enter PrescriptionID: 8  
Enter PatientName: Anugrah  
Enter Medication: Acetaminophen  
Enter Cost: 150  
Prescription created successfully.
```

Prescription Management System

```
1. Create Prescription  
2. Read Prescription  
3. Update Prescription  
4. Delete Prescription  
5. List All Prescription  
6. Exit
```

```
Choose an option: 2  
Enter PrescriptionID: 8  
PrescriptionID: 8  
PatientName: Anugrah  
Medication: Acetaminophen  
Cost: 150
```

Prescription Management System

```
1. Create Prescription  
2. Read Prescription  
3. Update Prescription  
4. Delete Prescription  
5. List All Prescription  
6. Exit
```

```
Choose an option:
```

cmd C:\WINDOWS\system32\cmd.exe

Prescription Management System

1. Create Prescription
2. Read Prescription
3. Update Prescription
4. Delete Prescription
5. List All Prescription
6. Exit

Choose an option: 5

ID: 1, PatientName: Rahul, Medication: Dolo 650, Cost: 120

ID: 3, PatientName: Abijith, Medication: Halls, Cost: 30

ID: 5, PatientName: Gopika, Medication: Citrizine, Cost: 20

ID: 6, PatientName: anurag, Medication: Paracetamol, Cost: 50

ID: 8, PatientName: Anugrah, Medication: Acetaminophen, Cost: 150

Prescription Management System

1. Create Prescription
2. Read Prescription
3. Update Prescription
4. Delete Prescription
5. List All Prescription
6. Exit

Choose an option: 4

Enter Prescription ID: 8

Prescription deleted successfully.

Prescription Management System

1. Create Prescription
2. Read Prescription
3. Update Prescription
4. Delete Prescription
5. List All Prescription
6. Exit

Choose an option: 5

ID: 1, PatientName: Rahul, Medication: Dolo 650, Cost: 120

ID: 3, PatientName: Abijith, Medication: Halls, Cost: 30

ID: 5, PatientName: Gopika, Medication: Citrizine, Cost: 20

ID: 6, PatientName: anurag, Medication: Paracetamol, Cost: 50

cmd C:\WINDOWS\system32\cmd.exe

Prescription Management System

1. Create Prescription
2. Read Prescription
3. Update Prescription
4. Delete Prescription
5. List All Prescription
6. Exit

Choose an option: 3

Enter Prescription ID: 5

Enter New PatientName: Gopika V

Enter New Medication: Citrizine

Enter New Cost: 50

Prescription updated successfully.