МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

КРЕМЕНЧУЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ІМЕНІ МИХАЙЛА ОСТРОГРАДСЬКОГО

НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ЕЛЕКТРИЧНОЇ ІНЖЕНЕРІЇ

ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра автоматизації та інформаційних систем

Звіт з лабораторної роботи №4

З навчальної дисципліни «Організація баз даних та знань»

Виконав: студент групи КН-19-1

Данілочкін Данііл

Перевірив: старший викладач

Васильєв Денис Олегович

КРЕМЕНЧУК 2022

**Тема:** **Создание модели EDM – Entity Data Model**

Завдання: Используя технологию Entity Framework, реализовать приложение для организации работы с БД, соответственно номеру варианта. Для выборки данных предусмотреть реализацию API-интерфейса LINQ to Entities.

**Хід роботи**

Код програми:

namespace lab4

{

public partial class Menu : Form

{

public buildingCompanyEntities BCContext;

public ContractsForm contractsForm;

public SellersForm sellersForm;

public TeamsForm teamsForm;

public UsedMaterialsForm usedMaterialForm;

public WorkScheduleForm workScheduleForm;

public MaterialsInSellersForm materialsInSellersForm;

public EmployeesForm employeesForm;

public WorkSchedule\_TeamsForm workSchedule\_TeamsForm;

public Menu()

{

InitializeComponent();

}

private void openContracts\_Click(object sender, EventArgs e)

{

if (contractsForm == null)

{

contractsForm = new ContractsForm(this);

}

contractsForm.Show();

this.Hide();

}

private void openSellers\_Click(object sender, EventArgs e)

{

if (sellersForm == null)

{

sellersForm = new SellersForm(this);

}

sellersForm.Show();

this.Hide();

}

private void openTeams\_Click(object sender, EventArgs e)

{

if (teamsForm == null)

{

teamsForm = new TeamsForm(this);

}

teamsForm.Show();

this.Hide();

}

private void Menu\_Load(object sender, EventArgs e)

{

BCContext = new buildingCompanyEntities();

}

private void Menu\_FormClosing(object sender, FormClosingEventArgs e)

{

SaveChanges();

BCContext.Dispose();

}

public void SaveChanges()

{

try

{

BCContext.SaveChanges();

this.Refresh();

}

catch (Exception ex)

{

MessageBox.Show(ex.Message);

}

}

}

public partial class ContractsForm : Form

{

public Menu menu;

public ContractsForm(Menu menu)

{

InitializeComponent();

this.menu = menu;

WriteContracts();

}

private void WriteContracts()

{

contractsView.DataSource = menu.BCContext.Contracts.ToList();

contractsView.AllowUserToDeleteRows = false;

contractsView.AutoResizeColumns(DataGridViewAutoSizeColumnsMode.AllCells);

}

private void ContractsForm\_FormClosing(object sender, FormClosingEventArgs e)

{

menu.SaveChanges();

menu.contractsForm = null;

menu.Show();

}

private void openWorkSchedule\_Click(object sender, EventArgs e)

{

if (menu.workScheduleForm == null)

{

menu.workScheduleForm = new WorkScheduleForm(this);

}

menu.workScheduleForm.Show();

this.Hide();

}

private void openUsedMaterials\_Click(object sender, EventArgs e)

{

if (menu.usedMaterialForm == null)

{

menu.usedMaterialForm = new UsedMaterialsForm(this);

}

menu.usedMaterialForm.Show();

this.Hide();

}

}

public partial class UsedMaterialsForm : Form

{

ContractsForm contractsForm;

public UsedMaterialsForm(ContractsForm contractsForm)

{

InitializeComponent();

this.contractsForm = contractsForm;

WriteUsedMaterials();

}

public void WriteUsedMaterials()

{

contractsBox.DisplayMember = "Number";

contractsBox.DataSource = contractsForm.menu.BCContext.Contracts.ToList();

int selectedContract = (int)contractsForm.contractsView[0, contractsForm.contractsView.CurrentCell.RowIndex].Value;

usedMaterialsView.DataSource = contractsForm.menu.BCContext.UsedMaterials.Where(m =>

m.ContractNumber == selectedContract).ToList();

usedMaterialsView.Columns["Contract"].Visible = false;

usedMaterialsView.Columns["MaterialsInSeller"].Visible = false;

}

private void UsedMaterialsForm\_FormClosing(object sender, FormClosingEventArgs e)

{

contractsForm.menu.SaveChanges();

contractsForm.menu.usedMaterialForm = null;

contractsForm.Show();

}

private void contractsBox\_SelectedIndexChanged(object sender, EventArgs e)

{

usedMaterialsView.DataSource = ((Contract)contractsBox.SelectedItem).UsedMaterials.ToList();

}

}

public partial class WorkScheduleForm : Form

{

ContractsForm contractsForm;

public WorkScheduleForm(ContractsForm contractsForm)

{

InitializeComponent();

this.contractsForm = contractsForm;

WriteWorkSchedule();

}

private void WriteWorkSchedule()

{

contractsBox.DisplayMember = "Number";

contractsBox.DataSource = contractsForm.menu.BCContext.Contracts.ToList();

int selectedContract = (int)contractsForm.contractsView[0, contractsForm.contractsView.CurrentCell.RowIndex].Value;

workScheduleView.DataSource = contractsForm.menu.BCContext.workSchedules.Where(m =>

m.ContractNumber == selectedContract).ToList();

workScheduleView.Columns["Contract"].Visible = false;

workScheduleView.Columns["Team1"].Visible = false;

}

private void WorkScheduleForm\_FormClosing(object sender, FormClosingEventArgs e)

{

contractsForm.menu.SaveChanges();

contractsForm.menu.workScheduleForm = null;

contractsForm.Show();

}

private void contractsBox\_SelectedIndexChanged(object sender, EventArgs e)

{

workScheduleView.DataSource = ((Contract)contractsBox.SelectedItem).workSchedules.ToList();

}

}

public partial class SellersForm : Form

{

public Menu menu;

public SellersForm(Menu menu)

{

InitializeComponent();

this.menu = menu;

WriteSellers();

}

public void WriteSellers()

{

sellersView.DataSource = menu.BCContext.Sellers.ToList();

sellersView.Columns["MaterialsInSellers"].Visible = false;

}

private void SellersForm\_FormClosing(object sender, FormClosingEventArgs e)

{

menu.SaveChanges();

menu.sellersForm = null;

menu.Show();

}

private void openMaterialsInSellers\_Click(object sender, EventArgs e)

{

if (menu.materialsInSellersForm == null)

{

menu.materialsInSellersForm = new MaterialsInSellersForm(this);

}

menu.materialsInSellersForm.Show(this);

this.Hide();

}

}

public partial class MaterialsInSellersForm : Form

{

public SellersForm sellersForm;

public MaterialsInSellersForm(SellersForm sellersForm)

{

InitializeComponent();

this.sellersForm = sellersForm;

WriteMaterialsInSellers();

}

private void WriteMaterialsInSellers()

{

sellersBox.DisplayMember = "SellerName";

sellersBox.DataSource = sellersForm.menu.BCContext.Sellers.ToList();

string selectedSeller = sellersForm.sellersView[0, sellersForm.sellersView.CurrentCell.RowIndex].Value.ToString();

materialsInSellersView.DataSource = sellersForm.menu.BCContext.MaterialsInSellers.Where(m => m.SellerName == selectedSeller).ToList();

materialsInSellersView.Columns["Seller"].Visible = false;

materialsInSellersView.Columns["UsedMaterials"].Visible = false;

}

private void MaterialsInSellersForm\_FormClosing(object sender, FormClosingEventArgs e)

{

sellersForm.menu.SaveChanges();

sellersForm.menu.materialsInSellersForm = null;

sellersForm.Show();

}

private void sellersBox\_SelectedIndexChanged(object sender, EventArgs e)

{

materialsInSellersView.DataSource = ((Seller)sellersBox.SelectedItem).MaterialsInSellers.ToList();

}

}

public partial class TeamsForm : Form

{

public Menu menu;

public TeamsForm(Menu menu)

{

InitializeComponent();

this.menu = menu;

WriteTeams();

}

private void WriteTeams()

{

teamsView.DataSource = menu.BCContext.Teams.ToList();

teamsView.Columns["Employe"].Visible = false;

teamsView.Columns["workSchedules"].Visible = false;

teamsView.AllowUserToDeleteRows = false;

teamsView.AutoResizeColumns(DataGridViewAutoSizeColumnsMode.AllCells);

}

private void TeamsForm\_FormClosing(object sender, FormClosingEventArgs e)

{

menu.SaveChanges();

menu.teamsForm = null;

menu.Show();

}

private void openEmployees\_Click(object sender, EventArgs e)

{

if (menu.employeesForm == null)

{

menu.employeesForm = new EmployeesForm(this);

}

menu.employeesForm.Show();

this.Hide();

}

private void openWorkSchedule\_Click(object sender, EventArgs e)

{

if (menu.workSchedule\_TeamsForm == null)

{

menu.workSchedule\_TeamsForm = new WorkSchedule\_TeamsForm(this);

}

menu.workSchedule\_TeamsForm.Show();

this.Hide();

}

}

public partial class EmployeesForm : Form

{

public TeamsForm teamsForm;

public EmployeesForm(TeamsForm teamsForm)

{

InitializeComponent();

this.teamsForm = teamsForm;

WriteEmployees();

}

private void WriteEmployees()

{

teamsBox.DisplayMember = "TeamName";

teamsBox.DataSource = teamsForm.menu.BCContext.Teams.ToList();

string selectedTeam = teamsForm.teamsView[0, teamsForm.teamsView.CurrentCell.RowIndex].Value.ToString();

employeesView.DataSource = teamsForm.menu.BCContext.Employes.Where(e => e.Team == selectedTeam).ToList();

employeesView.Columns["Teams"].Visible = false;

}

private void teamsBox\_SelectedIndexChanged(object sender, EventArgs e)

{

employeesView.DataSource = teamsForm.menu.BCContext.Employes.Where(t => t.Team == ((Team)teamsBox.SelectedItem).TeamName).ToList();

}

private void EmployeesForm\_FormClosing(object sender, FormClosingEventArgs e)

{

teamsForm.menu.SaveChanges();

teamsForm.menu.employeesForm = null;

teamsForm.Show();

}

}

public partial class WorkSchedule\_TeamsForm : Form

{

TeamsForm teamsForm;

public WorkSchedule\_TeamsForm(TeamsForm teamsForm)

{

InitializeComponent();

this.teamsForm = teamsForm;

WriteWorkSchedule();

}

private void WriteWorkSchedule()

{

teamsBox.DisplayMember = "TeamName";

teamsBox.DataSource = teamsForm.menu.BCContext.Teams.ToList();

string selectedTeam = teamsForm.teamsView[0, teamsForm.teamsView.CurrentCell.RowIndex].Value.ToString();

workScheduleView.DataSource = teamsForm.menu.BCContext.workSchedules.Where(s => s.Team == selectedTeam).ToList();

workScheduleView.Columns["Team"].Visible = false;

workScheduleView.Columns["Team1"].Visible = false;

workScheduleView.Columns["Contract"].Visible = false;

}

private void teamsBox\_SelectedIndexChanged(object sender, EventArgs e)

{

workScheduleView.DataSource = ((Team)teamsBox.SelectedItem).workSchedules.ToList();

}

private void WorkSchedule\_TeamsForm\_FormClosing(object sender, FormClosingEventArgs e)

{

teamsForm.menu.SaveChanges();

teamsForm.menu.workSchedule\_TeamsForm = null;

teamsForm.Show();

}

}

}

Зображення, що містить текст

Автоматично згенерований опис

Рисунок 1 – вікно програми

Зображення, що містить стіл

Автоматично згенерований опис

Рисунок 2 – вікно програми

Зображення, що містить стіл

Автоматично згенерований опис

Рисунок 3 – вікно програми

Зображення, що містить текст

Автоматично згенерований опис

Рисунок 4 – вікно програми

Зображення, що містить стіл

Автоматично згенерований опис

Рисунок 5 – вікно програми

Зображення, що містить текст

Автоматично згенерований опис

Рисунок 6 – вікно програми

Зображення, що містить стіл

Автоматично згенерований опис

Рисунок 7 – вікно програми

Зображення, що містить стіл

Автоматично згенерований опис

Рисунок 8 – вікно програми

Зображення, що містить стіл

Автоматично згенерований опис

Рисунок 9 – вікно програми

**Висновок:** в результаті виконання лабораторної роботи було вивчено і протестовано на практиці знання з використання Entity Data Model для роботи Windows Forms додатку з базою даних.

**Контрольні питання**

1. Что представляет платформа Entity Framework какие возможности она предоставляет?

Платформа Entity Framework - набор технологий ADO.NET, обеспечивающих разработку приложений, связанных с обработкой данных. Позволяет работать с концептуальной моделью даных, не занимаясь реляционной схемой хранения.

1. Какие способы выполнения запросов к концептуальной модели и возврата объектов реализует платформа Entity Framework?

- LINQ to Entities.

- Entity SQL. Независимый от хранилища диалект SQL, который работает непосредственно с сущностями в концептуальной модели.

3. Какой поставщик данных включает платформа Entity Framework?

EntityClient

1. Что представляет собой модель EDM?

Entity Data Model – концептуальная модель данных основаная на классах, которая может быть превращена в физическую модель реляционной системы хранения.

1. Какие положения необходимо учитывать при создании EDM модели?

• Все сущности должны иметь ключи.

• Для таблицы, которая представляет связь типа «многие ко многим» между двумя таблицами в базе данных, может не оказаться эквивалентной сущности в концептуальной схеме. В таком случае вместо нее будет ассоциация типа «многие ко многим».

• В версии .NET Framework 3.5 с пакетом обновления 1 (SP1) средства модели Entity Data Model поддерживают только автоматическое создание модели EDM на основе существующего источника данных.

1. Какое предназначение обьекта EntityConnection?

Связь Entity Framework с поставщиком данных ADO.NET.

1. Какое предназначение класса ObjectContext при создании EDM модели?

Средства Entity Framework создают класс, производный от ObjectContext, который представляет контейнер сущностей, определенный в концептуальной модели. Класс ObjectContext поддерживает запросы к модели EDM для создания, обновления и удаления объектов сущностей, а также для получения сущностей в виде объектов.