МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ

«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

Кафедра «Програмна інженерія та інформаційні технології управління»

Звіт з лабораторної роботи №6

з дисципліни «Стек технологій .NET»

Виконав:

Студент групи КН-36а

Рубан Ю. Д.

Перевірила:

Доц. Літвінова Ю. С.

Харків – 2018

**Тема:** Доступ к даним в .NET. Технологія ADO.NET

**Ціль:** Ознайомитися з технологією доступу к даним ADO.NET

**Варіант індивідуального завдання:** Виробниче підприємство з виробницвта побутової техніки

**Опис схеми бази даних:**

Схема бази даних показана на рисунку 1.

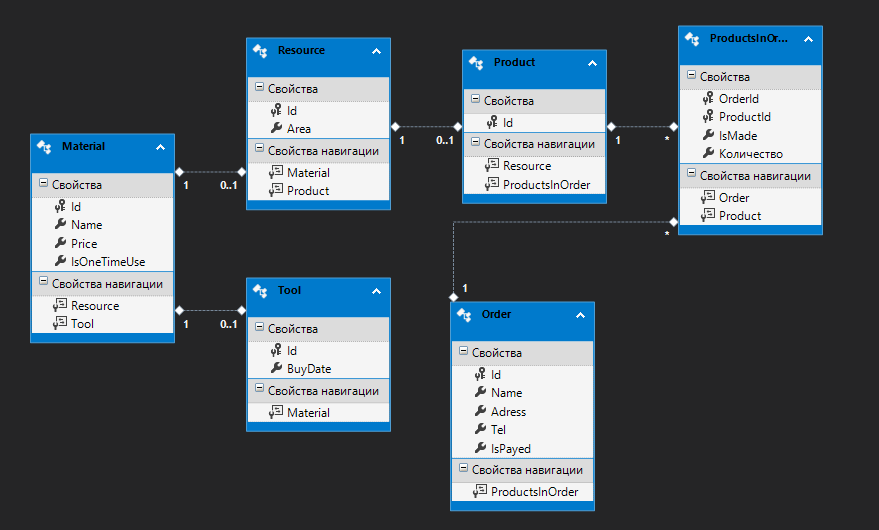


Рисунок 1 – Схема бази даних

Таблиця Order потрібна для зберігання інформації про замовника, відпові на запитання чи оплачена вона. Ця таблиця пов’язана з таблицею ProductsInOrder. Вона показує товари в конкретному замовеленні. Товари відбповідають таблиці Product яка є ресурсом (Resource) і матеріалом (Material). Meterial основна таблиця, яка містить інформацію про конкретний матеріал, який використовується у виробництві.

У ході розробки веб застосунку було використано два підходи взаємодії з базою даних. Перший – взаємодія з базою даних через Entity Framework, а другий через від’єднаний рівень доступу до даних через ADO.NET.

Код, який демонструє перший підхід показаний у лістингу 1, а другий підхід показаний у лістингу 2.

Лістинг 1. Доступ до даних через Entity Framework

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using lab4Calc.Models.Buisness;

using lab4Calc.Models.DBModels;

namespace lab4Calc.Models.Data

{

public static class DBConverter

{

public static Buisness.Product FromDbProduct(DBModels.Product product)

{

return new Buisness.Product()

{

Id = product.Id,

Price = product.Resource.Material.Price,

Area = product.Resource.Area,

Name = product.Resource.Material.Name,

IsOneTimeUse = product.Resource.Material.IsOneTimeUse,

IsMade = false

};

}

public static Buisness.Order FromDBOrder(DBModels.Order o)

{

Stack<Buisness.Product> temp = new Stack<Buisness.Product>();

foreach (var p in o.ProductsInOrder)

{

temp.Push(DBConverter.FromDbProduct(p.Product));

}

OrderInfo info = new OrderInfo();

info.Adress = o.Adress;

info.CustomerName = o.Name;

info.Tel = o.Tel;

return new Buisness.Order(new ProductCollection(temp), info) {Id = o.Id };

}

public static DBModels.Product FromBuisnessProduct( LabBDEntities context, Buisness.Product product)

{

return context.Product.FirstOrDefault(x => x.Id == product.Id);

}

public static DBModels.Order FromBuisnessOrder(LabBDEntities context, Buisness.Order order)

{

DBModels.Order res = new DBModels.Order();

res.Name = order.Info.CustomerName;

res.Tel = order.Info.Tel;

res.Adress = order.Info.Adress;

List<DBModels.ProductsInOrder> products = new List<DBModels.ProductsInOrder>();

foreach(var p in order.Products.Products)

{

products.Add(new ProductsInOrder() { Order = res, Product = FromBuisnessProduct(context,p) });

}

res.ProductsInOrder = products;

return res;

}

}

public class DBProductRepository : IProductRepository

{

private LabBDEntities context = new LabBDEntities();

private List<Buisness.Product> products = new List<Buisness.Product>();

public DBProductRepository()

{

Init();

}

private void Init()

{

foreach (var p in context.Product)

{

products.Add(DBConverter.FromDbProduct(p));

}

}

public IEnumerable<Buisness.Product> Products { get { return products; } }

public Buisness.Product GetById(int id)

{

return products.FirstOrDefault(x => x.Id == id);

}

public void Remove(Buisness.Product product)

{

products.Remove(product);

context.Product.Remove(context.Product.FirstOrDefault(x => x.Id == product.Id));

context.SaveChanges();

}

public void Update(Buisness.Product product)

{

Buisness.Product p = GetById(product.Id);

if (p == null)

{

p = new Buisness.Product() { Price = product.Price, Area = product.Area, IsMade = product.IsMade, IsOneTimeUse = product.IsOneTimeUse, Name = product.Name };

context.Product.Add(DBConverter.FromBuisnessProduct(context,p));

products.Add(p);

context.SaveChanges();

}

else

{

p.IsMade = product.IsMade;

p.IsOneTimeUse = product.IsOneTimeUse;

p.Name = product.Name;

p.Price = product.Price;

p.Area = product.Area;

DBModels.Product t = context.Product.FirstOrDefault(x => x.Id == product.Id);

t.Resource.Area = p.Area;

t.Resource.Material.Price = p.Price;

t.Resource.Material.Name = p.Name;

t.Resource.Material.IsOneTimeUse = p.IsOneTimeUse;

context.SaveChanges();

}

}

}

}

Лістинг 2. Доступ до даних за допомогою ADO.NET

using System;

using System.Collections.Generic;

using System.Linq;

using System.Web;

using lab4Calc.Models.Simple;

using lab4Calc.Models.Simple.NewsDataSetTableAdapters;

using static lab4Calc.Models.Simple.NewsDataSet;

namespace lab4Calc.Models.Data

{

public class DBNewsRepository : INewsRepository

{

private NewsDataSet ds = new NewsDataSet();

private TitleTableAdapter titleAdapter = new TitleTableAdapter();

private AuthorTableAdapter authorAdapter = new AuthorTableAdapter();

private Author FromAuthorRow(AuthorRow authorRow)

{

return new Author() { Id = authorRow.Id, Name = authorRow.Name };

}

private NewsItem FromTitleRow(TitleRow titleRow)

{

return new NewsItem() { Id = titleRow.Id, AuthorId = titleRow.AuthorId, Date = titleRow.Date, Rating = titleRow.Rating, Text = titleRow.Text, Title = titleRow.Title };

}

public DBNewsRepository()

{

titleAdapter.Fill(ds.Title);

authorAdapter.Fill(ds.Author);

}

public IEnumerable<NewsItem> Items

{

get

{

Stack<NewsItem> items = new Stack<NewsItem>();

foreach(var r in ds.Title)

{

items.Push(FromTitleRow(r));

}

return items;

}

}

public IEnumerable<Author> Authors

{

get

{

Stack<Author> items = new Stack<Author>();

foreach (var r in ds.Author)

{

items.Push(FromAuthorRow(r));

}

return items;

}

}

public Author GetByAuthorId(int AuthorId)

{

var row = ds.Author.FirstOrDefault(x => x.Id == AuthorId);

if(row!=null)

return FromAuthorRow(row);

return null;

}

public NewsItem GetByTitleId(int TitleId)

{

var row = ds.Title.FirstOrDefault(x => x.Id == TitleId);

if (row != null)

return FromTitleRow(row);

return null;

}

public void RemoveArticle(int TitleId)

{

titleAdapter.Delete(TitleId);

titleAdapter.Fill(ds.Title);

ds.AcceptChanges();

}

public void RemoveAuthor(int AuthorId)

{

authorAdapter.Delete(AuthorId);

authorAdapter.Fill(ds.Author);

ds.AcceptChanges();

}

public void UpdateArticle(NewsItem item)

{

NewsItem temp = GetByTitleId(item.Id);

if(temp == null)

{

titleAdapter.Insert(item.Title, item.Text, item.Rating, item.AuthorId, item.Date);

}

else

{

titleAdapter.Update(item.Title, item.Text, item.Rating, item.AuthorId, item.Date, item.Id);

}

titleAdapter.Fill(ds.Title);

ds.AcceptChanges();

}

public void UpdateAuthor(Author author)

{

Author temp = GetByAuthorId(author.Id);

if (temp == null)

{

authorAdapter.Insert(author.Name);

}

else

{

authorAdapter.Update(author.Name,author.Id);

}

authorAdapter.Fill(ds.Author);

ds.AcceptChanges();

}

}

}

**Висновки:**

У даній лабораторній роботі було ознайомлено з технологією доступу до даних за допомогою ADO.NET