# Лабораторная работа №6: реляционная БД

## Цель

Опробовать работу с реляционной БД под управлением СУБД Microsoft SQL Server из .NET.

## Задание

Данная работа является продолжением лабораторной работы №4.

1. Создать БД с таблицами: счета, платежные документы, платежи.
2. Объявить интерфейс уровня доступа к данным: интерфейс должен содержать методы из класса доступа к данным лабораторной работы №4.
3. Сделанный в рамках лабораторной работы №4 класс работы с данными изменить так, чтобы он реализовывал интерфейс из пункта 2.
4. Добавить в уровень доступа к данным класс работы с данными, работающий с БД. Класс должен читать счета и платёжные документы из соответствующих таблиц, и записывать платежи в таблицу платежей.
5. В программе должен быть реализован отдельный класс-фабрика, который содержит всего один метод, возвращающий класс для работы с данными. В зависимости от настроек метод должен возвращать либо класс для работы с файлами, либо класс для работы с БД. Пример определения класса:

class RepositoryFactory {

public static IRepository GetRepository ( ) { … }

}

Выполнение перечисленных выше требований оценивается на «Хорошо». Для получения оценки «Отлично» необходимо все настройки разместить в файле конфигурации приложения (app.config). А именно, следующие настройки:

* Имена файлов;
* Строку подключения к СУБД;
* Тип класса для работы с данными: файлы или СУБД.

## Демонстрация работы

Предполагается, что данные в БД точно такие же, какие в файлах для проверки лабораторной работы №4.

1. Настроить программу на работу с файлами;
2. Запустить программу с тестовыми файлами;
3. Продемонстрировать файл, получившийся в результате;
4. Изменить настройки программы на работу с БД;
5. Запустить программу. Результат её работы в таблице БД должен в точности совпадать с результатом работы программы с файлами.

## Вспомогательные материалы

### Создание БД

Для создания к БД необходимо подключиться к СУБД Microsoft SQL Server, запустив SQL Server Management Studio. После этого необходимо создать запрос, нажав соответствующую кнопку в панели инструментов. В открывшемся окне необходимо вписать и запустить запрос (кнопка F5):

CREATE DATABASE Lab6Db

После этого необходимо перейти в созданную БД запросом

USE Lab6Db

После этого необходимо создать таблицы. Это выполняется следующими запросами:

CREATE TABLE dbo.PayDocs (

Id INT PRIMARY KEY IDENTITY,

Client NVARCHAR(250) NOT NULL,

Date DATETIME NOT NULL,

Number NVARCHAR(120) NOT NULL,

Sum DECIMAL(38,2) NOT NULL )

GO

CREATE TABLE dbo.Bills (

Id INT PRIMARY KEY IDENTITY,

Client NVARCHAR(250) NOT NULL,

Date DATETIME NOT NULL,

Number NVARCHAR(120) NOT NULL,

Sum DECIMAL(38,2) NOT NULL )

GO

CREATE TABLE dbo.Payments (

Id INT PRIMARY KEY IDENTITY,

PayDocId INT NOT NULL,

BillId INT NOT NULL,

Sum DECIMAL(38,2) NOT NULL,

CONSTRAINT FK\_Payments\_PayDoc FOREIGN KEY (PayDocId) REFERENCES dbo.PayDocs(Id),

CONSTRAINT FK\_Payments\_Bill FOREIGN KEY (BillId) REFERENCES dbo.Bills(Id) )

GO

После этого надо записать счета и платежные документы в таблицу. Для этого необходимо самостоятельно подготовить запросы для добавления данных. Пример добавления платежного документа:

INSERT INTO dbo.PayDocs ( Client, Date, Number, Sum )

VALUES ( 'Газпромавтоматика', '20160530', 'FT-006789', 1000.45 )

Пример запроса на выборку данных из таблицы:

SELECT \* FROM dbo.PayDocs

### Чтение и запись данных БД из .NET

Для работы с СУБД необходимо открыть соединение. Пример открытия соединения (использовать пространство имён System.Data и System.Data.SqlClient):

using ( var cnn = new SqlConnection (

@"Data Source=localhost\sqlexpress;Database=Lab5Db;User ID=sa;Password=mypwd123" ) ) {

cnn.Open ( );

}

Для чтения данных из БД можно воспользоваться классом SqlDataAdapter:

var bills = new DataTable ( );

var adapter = new SqlDataAdapter ( "SELECT \* FROM dbo.Bills", cnn );

adapter.Fill ( bills );

Для записи данных необходимо создать команду, параметры к ней, после чего выполнить команду. Например:

var insertBill = new SqlCommand ( @"INSERT INTO dbo.PayDocs ( Client, Date, Number, Sum ) VALUES ( @client, @date, @number, @sum )", cnn );

insertBill.Parameters.AddWithValue ( "@client", "Газпромавтоматика" );

insertBill.Parameters.AddWithValue ( "@date", new DateTime ( 2016, 05, 30 ) );

insertBill.Parameters.AddWithValue ( "@number", "FT-006789" );

insertBill.Parameters.AddWithValue ( "@sum", 1000.45 );

insertBill.ExecuteNonQuery ( );

### Полезные классы и методы .NET

1. Чтение пользовательских настроек из секции appSettings:

string tp = System.Configuration.ConfigurationManager.AppSettings["TransferType"];

1. Чтение строки подключения:

string cs = System.Configuration.ConfigurationManager

.ConnectionStrings["Default"].ConnectionString;