

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РФ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ
БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«ВОРОНЕЖСКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ»

В.Г. Рудалев

РАЗРАБОТКА ПРИЛОЖЕНИЙ БАЗ ДАННЫХ

Часть 2

Учебное пособие

Воронеж
Издательский дом ВГУ
2016

СОДЕРЖАНИЕ

1. Проектирование приложения	4
1.1. Общие принципы	4
1.2. Пример предметной области	4
1.3. Серверная часть приложения.....	7
2. Создание бизнес-слоя	9
2.1. Базовые классы.....	9
2.2. Логика доступа к данным.....	10
3. Пользовательский интерфейс	15
4. Настройка безопасности.....	22
5. Управление транзакциями	25
5.1. Общие принципы	25
5.2. Транзакции на примере сохранения покупательской корзины.....	27
5.3. Пользовательский интерфейс	34
Задания	38
Библиографический список	39
Приложение. классы ado.net	39

После этого нажмите кнопку New Query, загрузите в окно заготовленный скрипт и выполните его. Перед выполнением добавьте в начало скрипта строчку use Themes;

```
use Themes;
```

```
CREATE TABLE KursWork(  
    WorkID int IDENTITY(1,1) NOT NULL,  
    WorkName varchar(Max) NOT NULL,  
    PrepID int NOT NULL,  
    StudID int NULL,  
    Kurs int NOT NULL,  
    CONSTRAINT PK_KursWork PRIMARY KEY (WorkID)  
)
```

```
CREATE TABLE Students(  
    StudID int IDENTITY(1,1) NOT NULL,  
    FIO varchar(Max) NOT NULL,  
    kurs int NOT NULL,  
    CONSTRAINT PK_Students PRIMARY KEY (StudID)  
)
```

```
CREATE TABLE Teachers(  
    PrepID int IDENTITY(1,1) NOT NULL,  
    PrepFIO varchar(Max) NOT NULL,  
    Post varchar(10) NOT NULL,  
    CONSTRAINT PK_Teachers PRIMARY KEY (PrepID)  
)
```

```
ALTER TABLE KursWork ADD CONSTRAINT FK_KursWork_Students FOREIGN KEY(StudID)  
REFERENCES Students (StudID)
```

```
ALTER TABLE KursWork ADD CONSTRAINT FK_KursWork_Teachers FOREIGN KEY(PrepID)  
REFERENCES Teachers (PrepID)
```

Проверьте правильность создания, определив в узле Themes диаграмму базы данных (рис. 3).

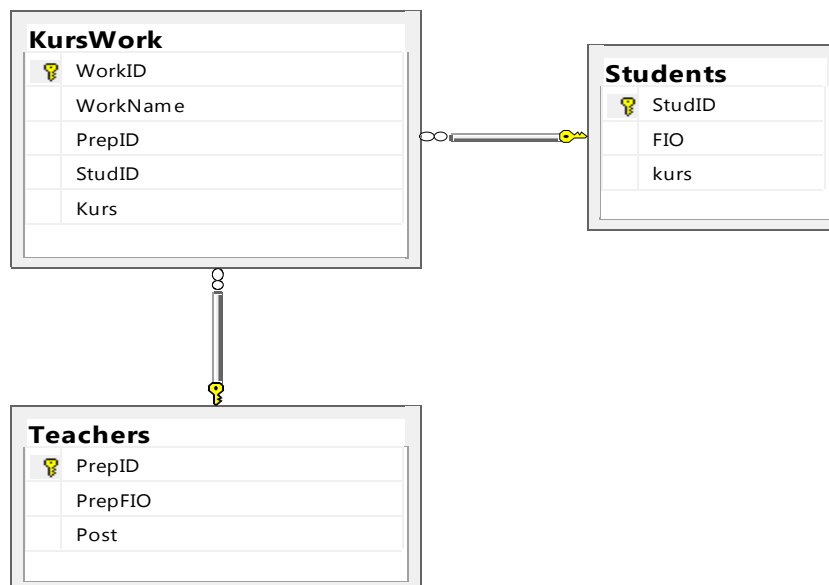


Рис. 3

1.3. Серверная часть приложения

Напишем следующие объекты в БД Themes, составляющие серверную часть приложения. Все нижеследующие скрипты удобнее выполнить в окне запроса New Query для БД Themes.

Представления:

Показ студентов, не записавшихся на курсовые работы.

```

CREATE VIEW StudFree
AS
  SELECT    StudID, FIO, kurs
  FROM      Students AS s
  WHERE (StudID NOT IN (SELECT StudID FROM KursWork
                        WHERE (StudID IS NOT NULL)))
  
```

Показ свободных тем, для которых в таблице KursWork столбец StudID имеет значение NULL .

```

CREATE VIEW ViewThemesFree
AS
  SELECT WorkID, PrepID, WorkName, Kurs
  FROM KursWork
  WHERE (StudID IS NULL)
  
```

Показ преподавателей.

```

create view ViewTeachers
as
  select PrepID, PrepFIO, Post from Teachers
  
```

Показ всех тем с информацией о преподавателях и студентах.

Данное представление показывает для всех преподавателей темы курсовых работ вместе с записанными студентами. Так как используется левое внешнее соединение, то показываются преподаватели, чьи темы не разобраны.

```
CREATE VIEW ViewThemes
AS
    SELECT Teachers.PrepFIO AS Teacher, Teachers.Post
    AS Position, KursWork.WorkName AS Theme,
    KursWork.Kurs, Students.FIO AS Student
    FROM Teachers
    LEFT OUTER JOIN KursWork
    ON Teachers.PrepID = KursWork.PrepID
    LEFT OUTER JOIN Students
    ON KursWork.StudID = Students.StudID
```

Хранимые процедуры:

Процедура захвата темы.

```
CREATE PROCEDURE CaptureTheme
    @ThemeID int,
    @StudID int

-- Входные параметры – идентификаторы темы и студента
AS
    -- Объявляем вспомогательные переменные
    declare @kurs as int, @kurstheme as int
    -- Находим курс студента и курс курсовой работы
    select @Kurs=kurs from students where StudID=@StudID
    select @kurstheme=kurs from kurswork
        where workid = @ThemeID
    -- Проверка, не был ли он уже записан на курсовую
    -- работу на своем курсе
    if NOT Exists
        (select * from KursWork where (kurs=@kurs) and
        (StudID=StudID) ) and (@KursTheme=@Kurs)
        update dbo.KursWork Set StudID=@StudID where
            WorkID=@ThemeID
    else
        RaisError ('Ошибка. Разрешается брать только одну тему на своем курсе!', 16, 1)
```

Добавление студента

```
CREATE PROCEDURE InsertStudent
    @FIO varchar(20),
    @kurs int,
    @StudID int OUTPUT
AS
    INSERT INTO Students (FIO, Kurs) VALUES (@FIO, @kurs);
    SET @StudID = @@IDENTITY
```

Задание. Хранимые процедуры для добавления преподавателей InsertTeacher и заполнения первоначального списка тем и др. реализуйте самостоятельно.

2. Создание бизнес-слоя

2.1. Базовые классы

Запустите MS Visual Studio. Создайте новый проект категории «Приложение Windows Forms». Назовите проект Themes. В контекстном меню проекта найдите «Добавить класс» и в открывшемся шаблоне опишите класс Student.

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

namespace themes
{
    public class Student
    {
        public int StudID { set; get; }
        public string FIO { set; get; }
        public int Kurs { set; get; }
        public Student () { }
        public Student (int studID, string fio, int kurs)
        {
            StudID = studID;
            FIO = fio;
            Kurs = kurs;
        }
    }
}
```

Здесь StudID, FIO, Kurs – публичные свойства, совпадающие по именам со столбцами соответствующей таблицы. Присутствуют два конструктора. Первый (без аргументов) будет создавать объект, второй дополнительно инициализировать его поля.

Аналогично поступим с классами для таблиц преподавателей и тем. Называть классы следует в единственном числе, так как класс будет описывать одну строку таблицы.

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

namespace themes
```

```

{
    public class Teacher
    {
        public int PrepID { set; get; }
        public string FIO { set; get; }
        public string Post { set; get; }
        public Teacher () { }
        public Teacher (int prepID, string fio, string post)
        {
            PrepID = prepID;
            FIO = fio;
            Post = post;
        }
    }
}

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

namespace themes
{
    public class KursWork
    {
        public int WorkID { get; set; }
        public string WorkName { get; set; }
        public int PrepID { get; set; }
        public int StudID { get; set; }
        public int Kurs { get; set; }
        public KursWork(int workID, string workName, int prepID, int studID,int
kurs)
        {
            WorkID = workID;
            WorkName = workName;
            PrepID = prepID;
            StudID = studID;
            Kurs = kurs;
        }
    }
}

```

2.2. Логика доступа к данным

Добавим в проект класс ThemesDB, описывающий логику работы с темами курсовых работ.

Чтобы все классы были доступны, не забудьте указать пространства имен

```

using System.Data;
using System.Data.SqlClient;
using System.Configuration;

```