

Федеральное государственное бюджетное образовательное учреждение
высшего образования
Национальный исследовательский университет «МЭИ»

Отчет
к курсовому проекту
по дисциплине «Базы данных»
Тема: разработка БД для поддержки процессов работы
отдела кадров ВУЗа

Студент: Бабушкин С.А.
Группа: А-13-16

Преподаватель: доц., к.т.н. Сидорова Н.П.

Москва, 2019

Содержание

Сокращения.....	2
Задание на курсовой проект	3
Введение.....	4
1. Анализ предметной области	5
2. Проектирование базы данных.....	6
2.1 Разработка ER-моделей (модель «сущность-связь»)	6
2.2 Анализ реляционной модели	6
3. Реализация базы данных	8
4. Программная реализация	9
4.1 Объектная модель ADO.NET	9
4.2 Экранные формы интерфейса.....	10
4.3 Примеры работы программы.....	12
Заключение.....	14
Литература	15
Приложение	16
Приложение 1.....	16
Приложение 2.....	18

Сокращения

БД – база данных

ЗНФ – третья нормальная форма

СУБД – система управления базами данных

ОСЭП-общеуниверситетская система электронной почты

Задание на курсовой проект

Цель работы: разработка БД для поддержки процессов работы отдела кадров ВУЗа.

БД должна поддерживать выполнение следующих функций:

- учет сведений о сотрудниках ВУЗа различных категорий (руководители, ППС, вспомогательный персонал);
- формирование списка ППС по кафедрам, у которых истекает срок договора в текущем учебном году;
- изменение данных о ППС (кафедра, должность, уч. степень и звание, награды);

- учет трудовых книжек;
- составление списка дисциплин, которые преподает преподаватель или может преподавать;
- составление списка преподавателей по различным возрастным группам, уч. степеням и др.

Исходные данные для выполнения курсового проекта: среда разработки – Microsoft Windows NT; AllFusion Modeler Suite, СУБД по выбору студента, язык программирования SQL, язык высокого уровня по выбору студента.

Литература

1. Пирогов В.Ю. SQL Server 2005: программирование клиент-серверных приложений. – СПб: БХВ-Петербург, 2006. – 336 с.
2. Н.П. Сидорова. Базы данных. Методические указания к курсовому проектированию. –М.: Изд. дом МЭИ, 2010. – 40 с.

Введение

Целью курсового проекта является разработка модели базы данных (БД) и интерфейсных средств для выполнения указанных в задании функций. БД обеспечивает хранение информации и представляет собой поименованную совокупность данных, организованных по определенным правилам, включающим общие принципы описания, хранения и манипулирования данными. БД является информационной моделью предметной области, в которой решаются поставленные задачи.

Этапы выполнения курсового проекта:

- поведение анализа предметной области;
- проектирование БД;

- реализация БД средствами выбранной СУБД;
- программная реализация интерфейса с БД.

Для анализа предметной области используется CASE-средство AllFusion Process Modeler, который входит в состав пакета AllFusion Modeler Suite. Проектирование моделей БД выполняется с помощью CASE-средства AllFusion ERwin Data Modeler, входящий в состав этого же пакета.

Реализация БД проводится в среде Microsoft SQL Server, в качестве языка реализации был выбран С# в среде Microsoft Visual Studio.

1. Анализ предметной области

Предметная область курсового проекта: процессы работы отдела кадров ВУЗа. У каждого сотрудника есть контракт, по которому он однозначно определяется.

При организации процесса идентификации определены следующие правила:

- каждому договору соответствует один сотрудник
- дисциплину могут преподавать несколько преподавателей

При анализе задачи были выделены информационные объекты предметной области. Состав и характеристика информационных объектов приведена в табл. 1.

Табл. 1. Информационные объекты и их свойства

№	Имя объекта	Описание объекта	Свойства объекта
1	Договор	Содержит номер договора, дату окончания и номер трудовой книжки сотрудника	ИД договора Дата окончания Номер ТК
2	Руководители	Содержит описание характеристик руководителей	ИД руководителя ФИО ИД договора
3	Вспомогательный персонал	Содержит описание характеристик ВП	ИД ВП ФИО ИД договора
4	ППС	Содержит описание характеристик ППС	ИД ППС ФИО Уч. Степень Кафедра Должность Пол Возраст ИД договора

5	Дисциплина	Содержит описание дисциплины	Название Актуальность(преподают ли) ИД ППС
---	------------	------------------------------	--

2. Проектирование базы данных

Проектирование БД предполагает разработку моделей нескольких типов:

- Модель «сущность-связь», которая строится на основе выделенных информационных объектов и определения связей между ними
- Реляционная модель, которая формируется на основе модели «сущность-связь»

2.1 Разработка ER-моделей (модель «сущность-связь»)

ER-модель – модель данных, позволяющая описывать концептуальные схемы предметной области. Данная модель используется при высокоуровневом проектировании БД. С её помощью можно выделить ключевые сущности и обозначить связи, которые могут устанавливаться между этими сущностями.

Документирование модели было выполнено с помощью CASE-средства Erwin Data Modeler и приведено в Приложении 1.

2.2 Анализ реляционной модели

На основе разработанной ER-модели была построена реляционная модель, представленная на Рис 2.2.1

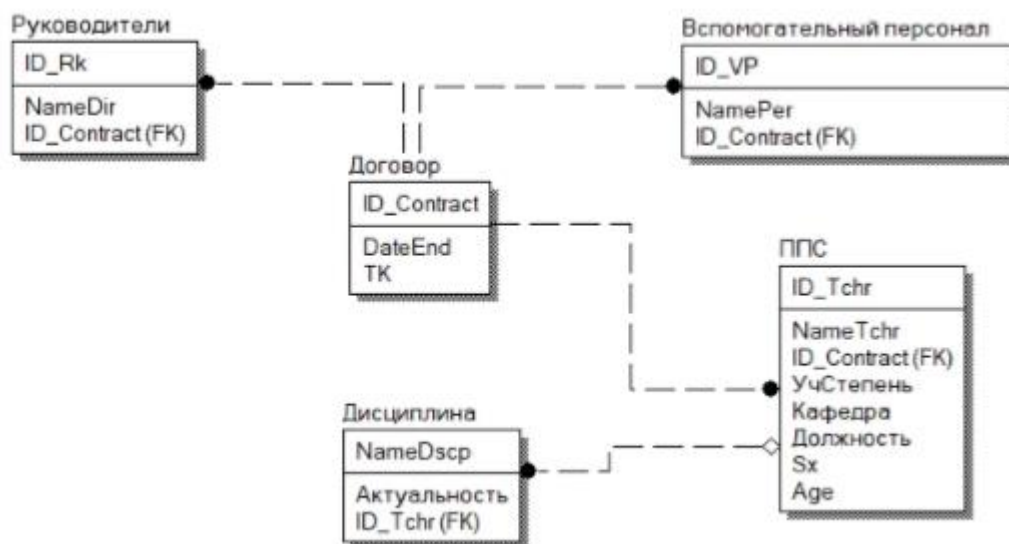


Рис. 2.2.1 Реляционная модель БД

Для получения хороших характеристик работы с БД, она должна находиться в 3НФ. Для проверки этого необходимо обеспечить атомарность атрибутов, выделить функциональные зависимости между атрибутами в каждом отношении и провести их анализ.

1. Отношение **Договор**(ИД_договора, Дата_Оконч, Номер_ТК), ключ отношения – ИД_договора.

Функциональные зависимости:

ИД_договора -> Дата_Оконч

ИД_договора -> Номер_ТК

2. Отношение **Руководители**(ИД_руководителя, ФИО, ИД_договора), ключ отношения – ИД_руководителя.

Функциональные зависимости:

ИД_руководителя -> ФИО

ИД_руководителя -> ИД_договора

3. Отношение **Вспомогательный персонал**(ИД_ВП, ФИО, ИД_договора), ключ отношения – ИД_ВП

Функциональные зависимости:

ИД_ВП -> ФИО

ИД_ВП -> ИД_договора

4. Отношение **ППС**(ИД_ППС, ФИО, Уч_степ, Кафедра, Должность, Пол, Возраст), ключ отношения – ИД_ППС

Функциональные зависимости:

ИД_ППС -> ФИО

ИД_ППС -> Уч_степ

ИД_ППС -> Кафедра

ИД_ППС -> Должность

ИД_ППС -> Пол

ИД_ППС -> Возраст

ИД_ППС -> ИД_договора

5. Отношение **Дисциплина**(Название_дисц, актуальность, ИД_ППС)

Функциональные зависимости:

Название_дисц -> Актуальность

Название_дисц -> ИД_ППС

Других функциональных зависимостей выявлено не было. Обоснование нахождения БД в ЗНФ предполагает проверку ограничений, накладываемых в ЗНФ.

1. Все таблицы БД находятся в 1НФ, их поля содержат атомарные значения
2. Полная функциональная зависимость не ключевых атрибутов от ключа для отношения Договор, Руководители, Вспомогательный персонал, ППС, Дисциплина обеспечивается тем, что ключи в этих отношениях простые.
3. В разработанных отношениях отсутствуют транзитивные зависимости.

Следовательно, БД находится в ЗНФ.

Полное описание разработанной реляционной модели создано с помощью CASE-средства Erwin Data Modeler и приведено в Приложении 2.

3. Реализация базы данных

В качестве СУБД для реализации БД была выбрана СУБД Microsoft SQL Server. Основным используемый язык запросов – Transact-SQL. Ниже приведено описание БД средствами Transact-SQL.

- Создание таблиц:

```
CREATE TABLE Contract (  
    ID_Contract Integer NOT NULL PRIMARY KEY IDENTITY,  
    DateEnd varchar(20) NOT NULL,  
    TK varchar(20) NOT NULL  
);
```

```
CREATE TABLE Ruk(  
    ID_Rk Integer NOT NULL PRIMARY KEY IDENTITY,  
    NameDir varchar(20) NOT NULL,  
    ID_Contract INTEGER NOT NULL foreign key references  
Contract(ID_Contract)  
);
```

```
CREATE TABLE VspomPers(  
    ID_VP Integer NOT NULL PRIMARY KEY IDENTITY,  
    NamePer varchar(20) NOT NULL,  
    ID_Contract INTEGER NOT NULL foreign key references  
Contract(ID_Contract)  
);
```

```
CREATE TABLE PPS(  
    ID_Tchr Integer NOT NULL PRIMARY KEY IDENTITY,  
    NameTchr varchar(20) NOT NULL,  
    Degree varchar(20) NOT NULL,  
    Kafedra varchar(20) NOT NULL,  
    Dolzh varchar(20) NOT NULL,  
    Sx char NOT NULL,  
    Age INTEGER NOT NULL,  
    ID_Contract INTEGER NOT NULL foreign key references  
Contract(ID_Contract)  
);
```

```
CREATE TABLE Dscp(  
    NameDscp varchar(20) NOT NULL PRIMARY KEY,  
    Actuality char NOT NULL,  
    ID_Tchr INTEGER NOT NULL foreign key references PPS(ID_Tchr)  
);
```


- Добавление правил целостности:

```
alter table Dscp
```

```
add check (Actuality in ('д', 'н'))
```

```
alter table PPS
```

```
add check (Sx in ('м', 'ж'))
```

4. Программная реализация

Для реализации приложения был выбран язык программирования C# в среде Microsoft Visual Studio.

4.1 Объектная модель ADO.NET

ADO.NET – это набор классов (фреймворк) для работы с базами данных, а также XML файлами. Аббревиатура ADO расшифровывается как ActiveX Data Objects. Данная технология имеет методы и классы для извлечения и обработки данных.

Список .NET приложений, которые используют возможности ADO.NET для различных действий с БД:

- ASP.NET Web Applications
- Console Applications
- Windows Applications.

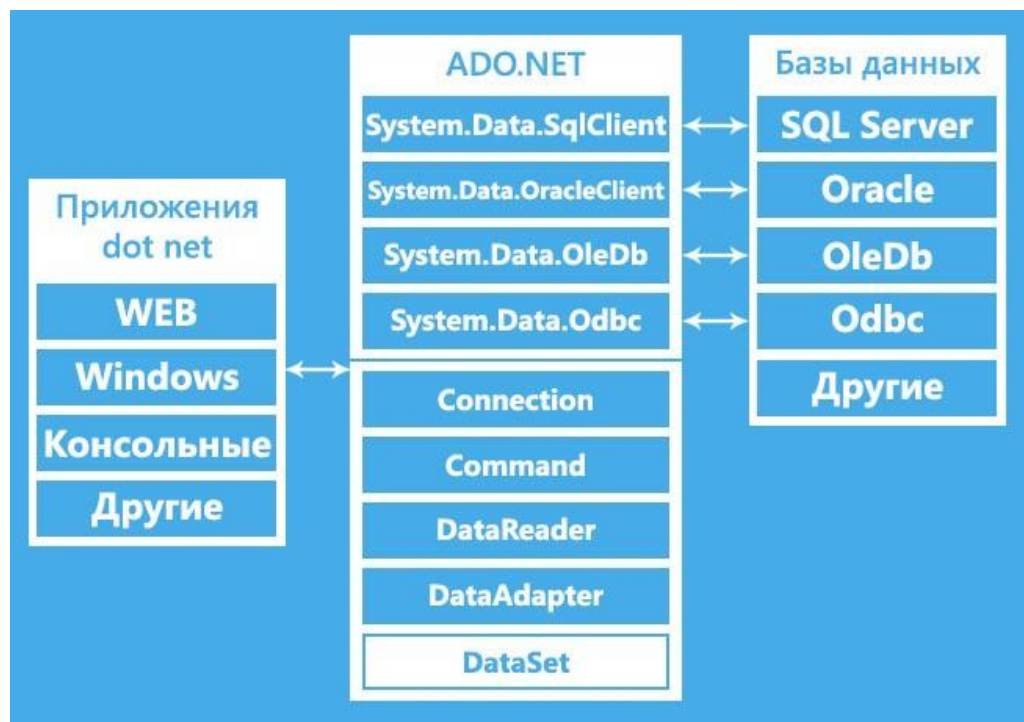


Рис. 4.1.1 Структура ADO.NET

На Рис.4.1.1 видны различные типы приложений (Веб приложения, консольные приложения, приложения для Windows и т.д.), использующие ADO.NET для подсоединения к БД (SQL Server, Oracle, OleDb, ODBC, XML-файлы и так далее).

Классы в ADO.NET:

1. Connection Class

2. Command Class
3. DataReader Class
4. DataAdaptor Class
5. DataSet.Class

Описание классов:

1. Connection Class

Данные классы применяются в ADO.NET для подсоединения к БД.

2. Command Class

Данный класс обеспечивает хранение и выполнение SQL команд. Ниже приведены различные команды, выполняющиеся с помощью данного класса.

- ExecuteReader: Возвращает данные к клиенту в виде строк.
- ExecuteNonQuery: Выполняет команду, изменяющую данные в базе данных.
- ExecuteScalar: Данный класс возвращает только одно значение.
- ExecuteXMLReader: (Только для классов SqlClient) Получает данные из базы данных SQL Server 2000 с помощью XML-потока.

3. DataReader Class

DataReader используется для получения данных. Он используется в сочетании с Command Class для выполнения SQL-запроса.

5. DataSet Class

Класс DataSet – сердце ADO.NET, представляющее из себя набор объектов DataTable. Каждый такой объект содержит много объектов DataColumn и DataRow.

4.2 Экранные формы интерфейса

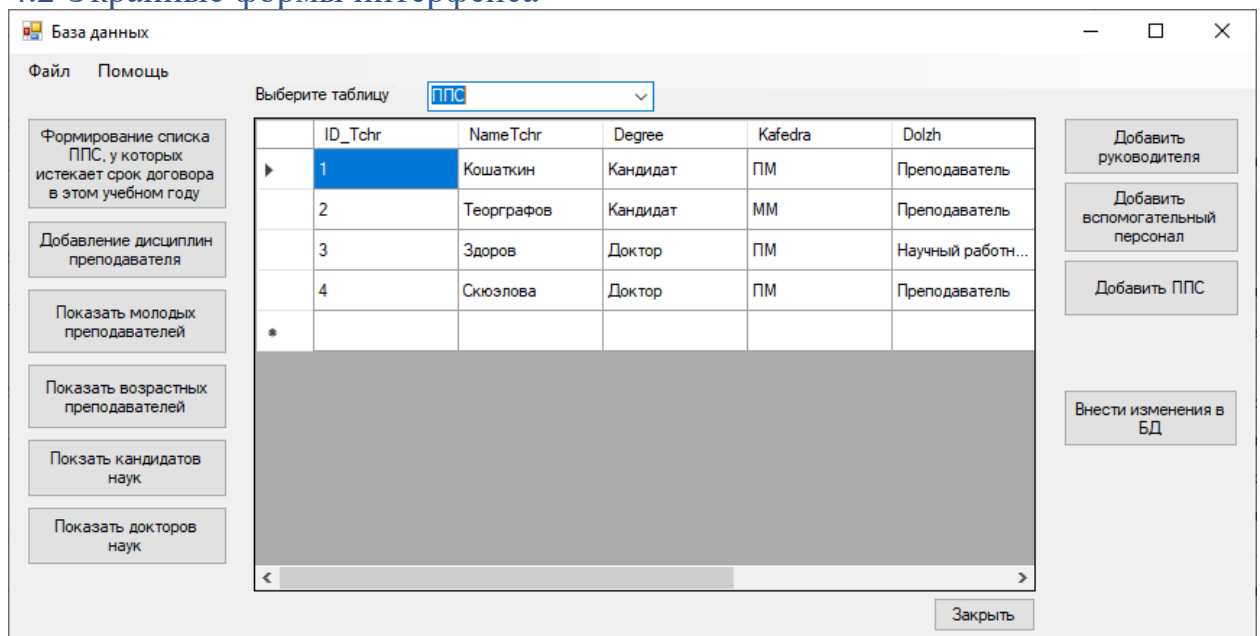


Рис. 4.2.1 Основное окно программы

В основном окне программы можно выбирать созданные в базе данных таблицы (Контракты, Руководители, Вспомогательный персонал, ППС, дисциплины), слева от таблицы расположены кнопки с запросами. Закреть программу можно с помощью кнопки «Закреть» или пункта меню «Файл/Закреть».

При добавлении пользователя в таблицу Ruk, VP открывается форма на Рис.4.2.2. Для корректной работы следует заполнить все поля в форме, иначе появляется сообщение об ошибке с возможностью изменения данных.

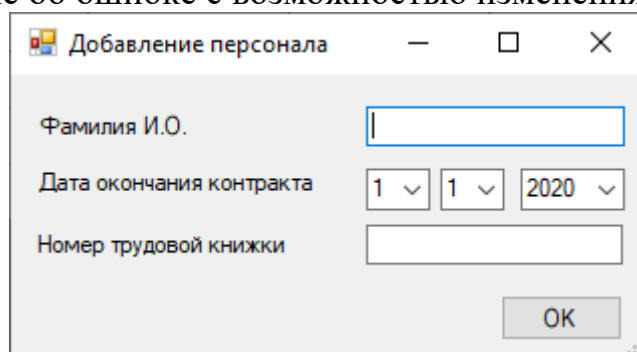


Рис.4.2.2 Добавление пользователя

Для добавления ППС открывается форма на Рис.4.2.3. Необходимо заполнить все поля. Ошибка в заполнении данных в других полях приводит к возникновению сообщения, соответствующего ошибки. Предлагается продолжить или отменить заполнение полей.

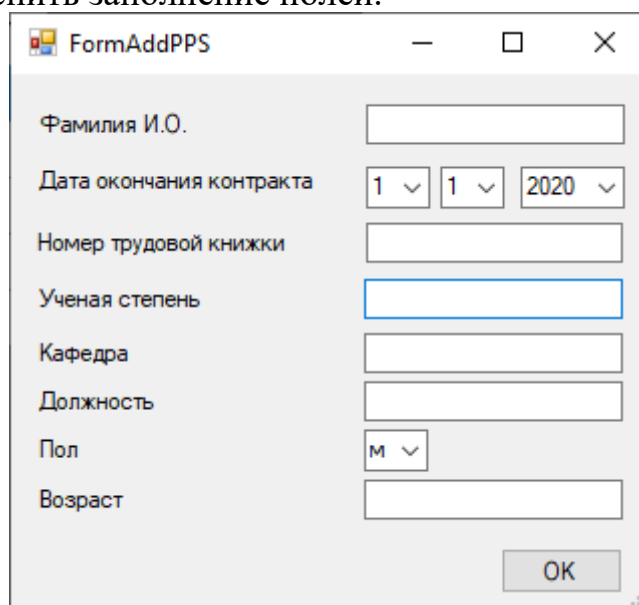


Рис.4.2.3 Добавление ППС

Для добавления дисциплин преподаваемых ППС открывается форма на Рис.4.2.4. Необходимо заполнить все поля. Ошибка в заполнении данных в других полях приводит к возникновению сообщения, соответствующего ошибки. Предлагается продолжить или отменить заполнение полей.

Рис.4.2.4 Добавление дисциплин

Для работы с запросами к базе данных был разработан класс SQL_DB с необходимыми запросами:

- `public static SqlConnection Connect()` – подключение к БД
- `public static DataTable SelectFromTable(string query)` – извлечь данные по запросу query
- `public static void AddRuk()` – добавление пользователя в таблицу Student
 - `public static void AddRuk()` – добавление пользователя в таблицу Ruk
 - `public static void AddVP()` – добавление пользователя в таблицу VspomPers
 - `public static void AddPPS()` – добавление пользователя в таблицу PPS
 - `public static void AddDscp()` – добавление пользователя в таблицу Dscp
 - `public static int GetCountRows(string tabl)` – Получение количества записей в таблице
 - `public static void ModifyDB(DataTable dataTable)` – Сохранение изменений в таблице PPS
 - `public static DataTable SelPPS()` – Формирование списка ППС, у которых истекает срок договора в этом учебном году
 - `public static DataTable SelAge(string sign(<= / >))` – Вывод молодых или возрастных преподавателей
 - `public static DataTable SelAge(string deg('Кандидат' / 'Доктор'))` – Вывод докторов или кандидатов наук

4.3 Примеры работы программы

- Добавление нового пользователя

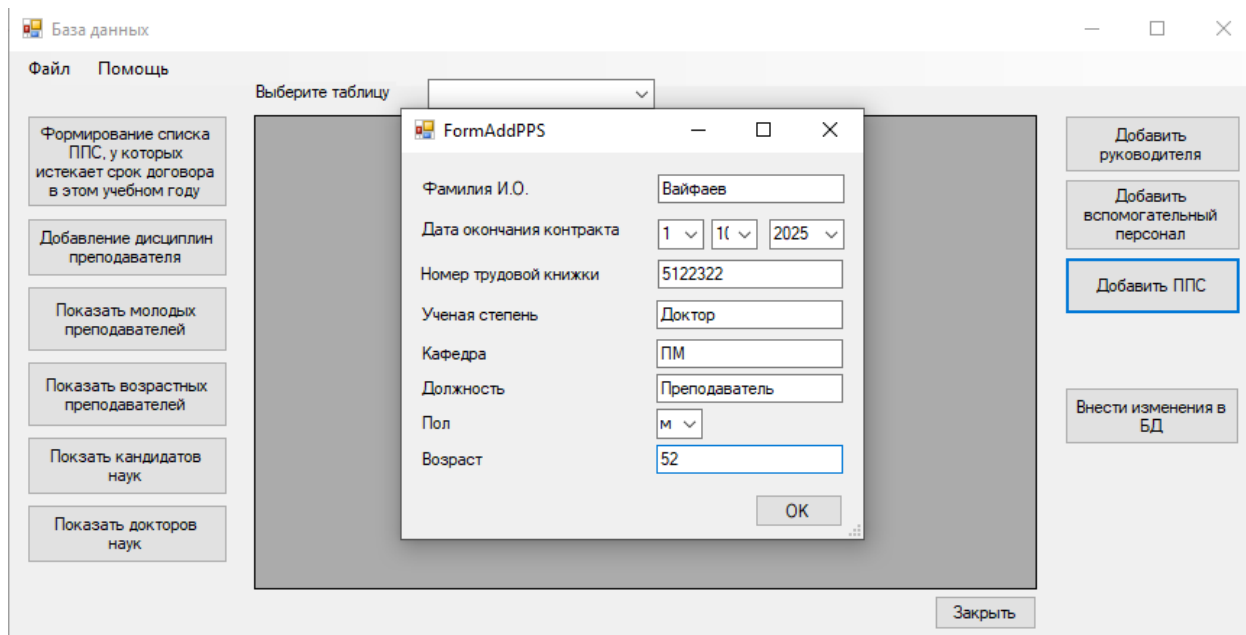


Рис.4.3.1(а) Добавление пользователя

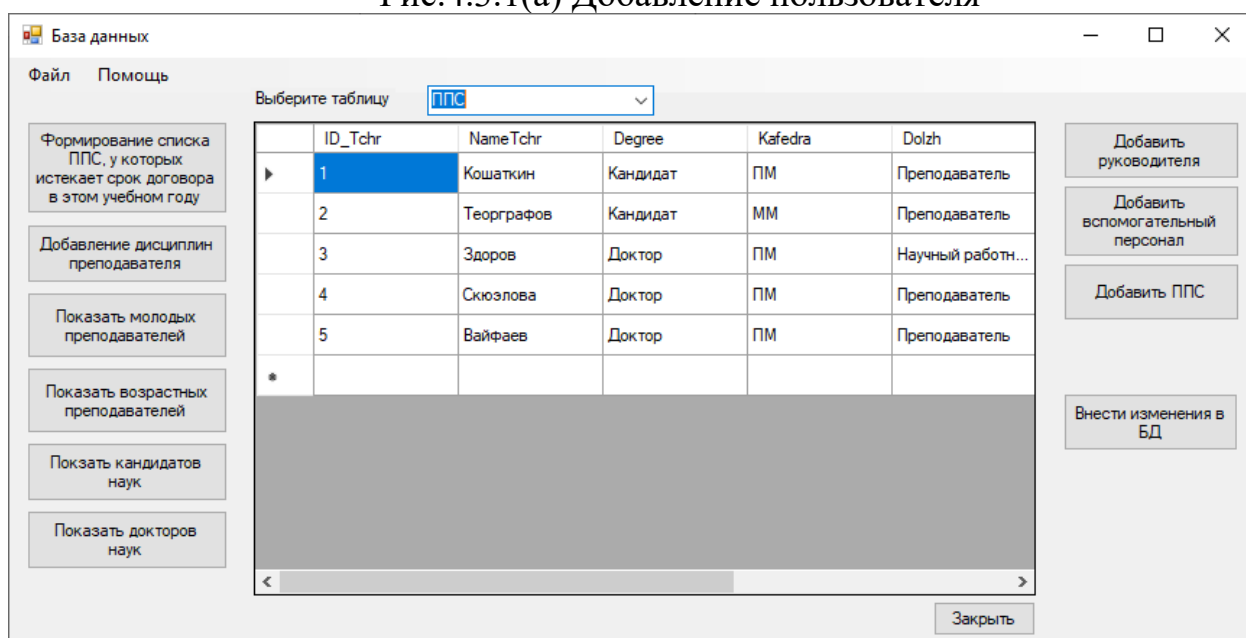


Рис.4.3.1(б) Добавление пользователя

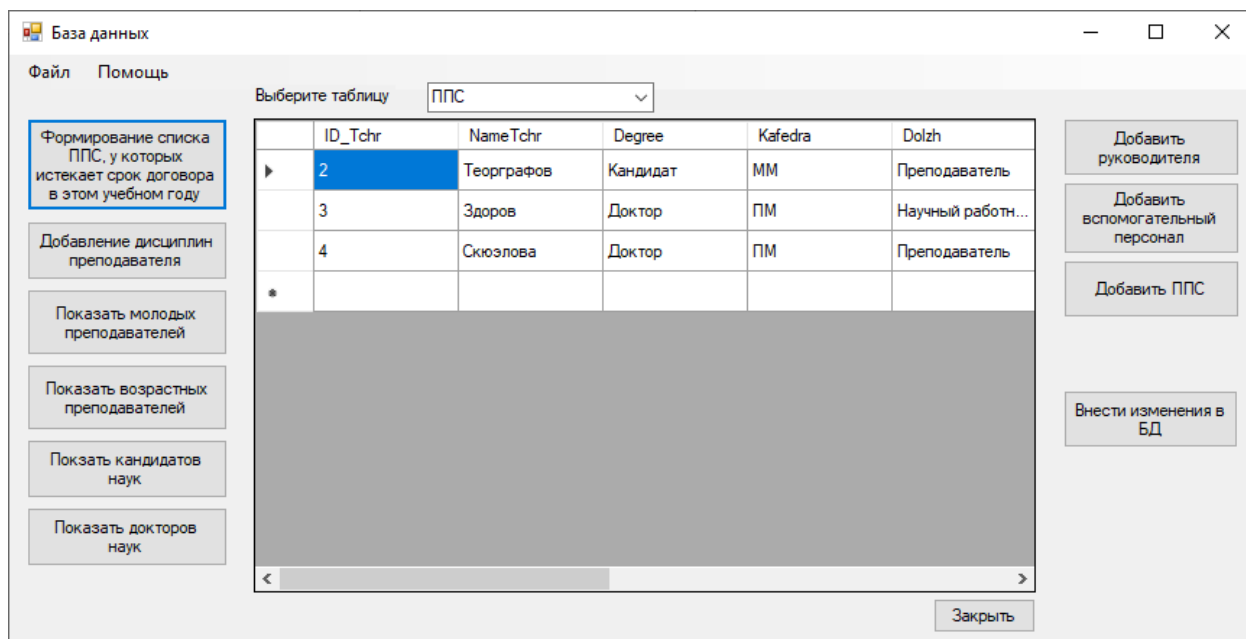


Рис.4.4 Список ППС, срок договора которых истекает в этом году

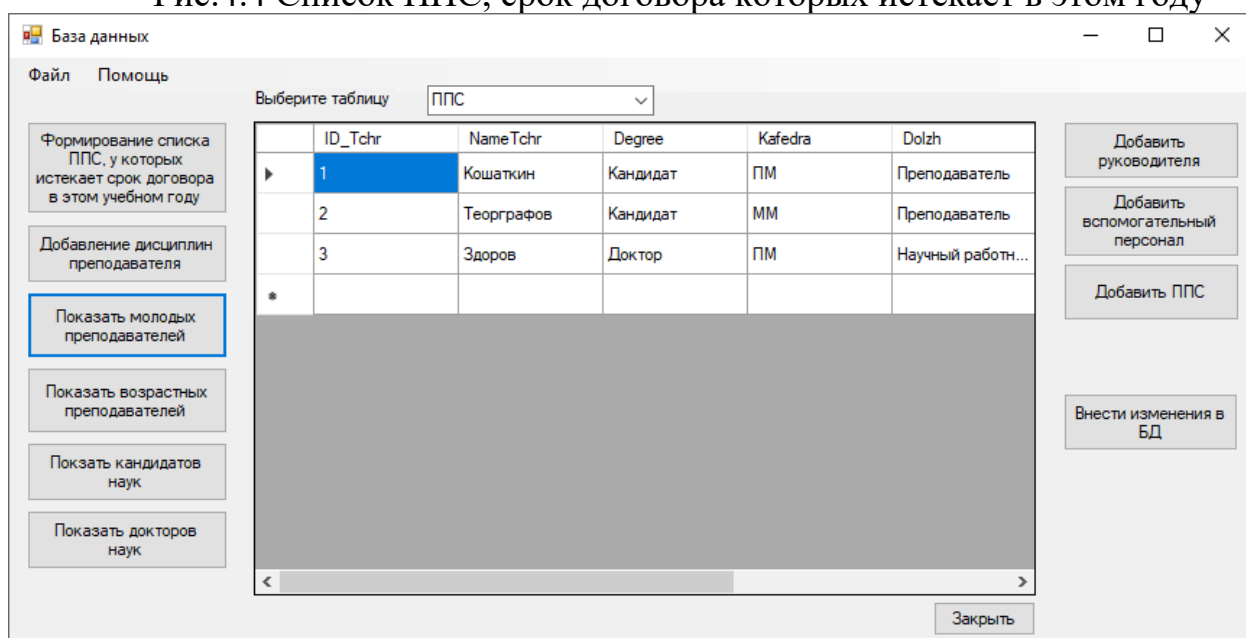


Рис.4.5 Список молодых ППС

Заключение

В результате выполнения курсового проекта была разработана БД для однозначной идентификации пользователей. База данных была разработана в Erwin Data Modeler и реализована в Microsoft SQL Server Management Studio 18 на языке запросов – Transact-SQL. На языке C# в Microsoft Visual Studio были разработаны программные средства интерфейса с БД и приложения.

Литература

1. Н.П. Сидорова. Базы данных. Методическое пособие к курсовому проектированию. – М.: Изд-во МЭИ, 2010 – 40 с.
2. Маран М.М. Программирование на языке С# в среде Microsoft Visual Studio 2005: Учебное пособие по курсу. – М. : Изд-во МЭИ, 2007 – 64
3. Класс SqlConnection [Интернет ресурс]
<https://docs.microsoft.com/ru-ru/dotnet/api/system.data.sqlclient.sqlconnection?view=netcore-3.0>
4. Класс SqlCommand [Интернет ресурс]
<https://docs.microsoft.com/ru-ru/dotnet/api/system.data.sqlclient.sqlcommand?view=netcore-3.0>

Приложение

Приложение 1

Entity

Entity		
Name	Type	Definition
Вспомогательный персонал	Independent	Сведения о вспомогательном персонале ВУЗа
Дисциплина	Independent	Дисциплины преподаваемые ППС
Договор	Independent	Сведения о заключенном договоре
ППС	Independent	Сведения о ППС ВУЗа
Руководители	Independent	Сведения о руководителях ВУЗа

Табл. Прил.1.1 Entity

Attribute

Attribute		
Name	Definition	Datatype
ID_Rk	Идентификатор руководителя	INTEGER
NameDir	ФИ директора	VARCHAR(20)
ID_Contract	Номер трудового контракта	INTEGER
ID_Contract	Номер трудового контракта	INTEGER
DateEnd	Срок договора	DATE
TK	Номер трудовой книжки	CHAR(18)
ID_VP	Идентификатор ВП	INTEGER
NamePer	ФИ вспомогательного персонала	VARCHAR(20)
ID_Contract	Номер трудового контракта	INTEGER
ID_Tchr	Идентификатор преподавателя	CHAR(18)
NameTchr	ФИ преподавателя	VARCHAR(20)
ID_Contract	Номер трудового контракта	INTEGER
УчСтепень	Ученая степень	VARCHAR(20)
Кафедра	Кафедра	VARCHAR(20)
Должность	Должность	VARCHAR(20)
Sx	Пол	VARCHAR(20)
Age	Возраст	INTEGER
NameDscr	Название дисциплины	CHAR(18)
Актуальность	Преподает ли эту дисциплину("д"/"н")	VARCHAR(20)
ID_Tchr	Идентификатор преподавателя	CHAR(18)

Табл. Прил.1.2 Attribute

Relationship		
Relationship		
Name	Cardinality	Definition
R/3	One-to-Zero-One-or-More	Договор вспомогательного персонала
R/5	One-to-Zero-One-or-More	Договор руководителя
R/2	One-to-Zero-One-or-More	Договор ИПС
R/12	Zero-or-One-to-Zero-One-or-More	ИПС предполагает дисциплину
Parent Entity(s) of "R/3" Relationship		
Name		
Договор		
Child Entity(s) of "R/3" Relationship		
Name		
Вспомогательный персонал		
Parent Entity(s) of "R/5" Relationship		
Name		
Договор		
Child Entity(s) of "R/5" Relationship		
Name		
Руководители		
Parent Entity(s) of "R/2" Relationship		
Name		
Договор		
Child Entity(s) of "R/2" Relationship		
Name		
ИПС		
Parent Entity(s) of "R/12" Relationship		
Name		
ИПС		
Child Entity(s) of "R/12" Relationship		
Name		
Дисциплина		

Табл. Прил.1.3 Relationship

Приложение 2

Column					
Column					
Name	Datatype	Null Option	Comment	Is PK	Is FK
Ид	INTEGER	NOT NULL	Идентификатор студента	Yes	No
Фамилия	VARCHAR2(20)	NOT NULL	Фамилия студента	No	No
Имя	VARCHAR2(20)	NOT NULL	Имя студента	No	No
Отчество	VARCHAR2(20)	NOT NULL	Отчество студента	No	No
Почта_ОСЭП	VARCHAR2(20)	NOT NULL	Почта в общеуниверситетской системе электронной почты	No	No
Место_обучения	VARCHAR2(20)	NOT NULL	Группа, в которой обучается студент	No	No
Контакт_Ид	CHAR(18)	NOT NULL	Идентификатор контакта	Yes	No
Авторизация	VARCHAR2(1)	NOT NULL	Был ли студент авторизован в системе	No	No
Ид	INTEGER	NOT NULL	Идентификатор студента	No	Yes
Тип_Контакта_Ид	INTEGER	NULL	Идентификатор типа контакта	No	Yes
Инф_Ид	CHAR(18)	NOT NULL	Идентификатор информации о студента	Yes	No
icrc	VARCHAR2(1)	NOT NULL	Является ли студент участником кружка (y/n)	No	No
email	VARCHAR2(20)	NULL	Почта	No	No
Телефон	VARCHAR2(11)	NULL	Телефон	No	No
Ид	INTEGER	NOT NULL	Идентификатор студента	No	Yes
Тип	VARCHAR2(20)	NULL	Тип контакта (Discord, vk, email, telegram, phone)	No	No
Данные	VARCHAR2(20)	NULL	Валидирующие данные для типа контакта	No	No
Тип_Контакта_Ид	INTEGER	NOT NULL	Идентификатор типа контакта	Yes	No

Табл. Прил.2.1 Column

Relationship

Relationship			
FK Name	Type	Null Option	Comment
R_7	Non-identifying	No Nulls	Дополнительная информация о студенте
R_5	Non-identifying	No Nulls	Студент может иметь (или не иметь, в случае, если он ещё не авторизован) дополнительную информацию о его авторизации
R_11	Non-identifying	Nulls Allowed	Дополнительная информация о студенте

Parent Table(s) of "R_7" Relationship	
Name	Comment
Студент	Сущность, определяющая студентов в университете

Child Table(s) of "R_7" Relationship	
Name	Comment
Контакт	Представляет дополнительные сведения о студенте: Был ли он авторизован и доступ к его дополнительным контактам

Parent Table(s) of "R_5" Relationship	
Name	Comment
Студент	Сущность, определяющая студентов в университете

Child Table(s) of "R_5" Relationship	
Name	Comment
Студент_Инф	Содержит информацию о студенте, является ли он участником кружка, его дополнительный номер и телефон

Parent Table(s) of "R_11" Relationship	
Name	Comment
Тип_контакта	Дополнительные данные о контакте. Типом является Discord, vk, email, telegram, phone, а Данным соответствующее ему валидирующее значение

Child Table(s) of "R_11" Relationship	
Name	Comment
Контакт	Представляет дополнительные сведения о студенте: Был ли он авторизован и доступ к его дополнительным контактам

Табл. Прил 2.2 Relationship

Validation

Validation		
Name	Type	Quote
Validation Rule 254	Valid Values List	Yes
Validation Rule 259	Valid Values List	Yes

Server(s) of "Validation Rule 254" Validation	
Rule	
%AttFieldName IN ('y', 'n')	

Server(s) of "Validation Rule 259" Validation	
Rule	
%AttFieldName IN ('Discord', 'vk', 'email', 'telegram', 'phone')	

Табл. Прил 2.3 Validation