

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Воронежский государственный технический университет»

Кафедра систем автоматизированного проектирования
и информационных систем

МЕТОДИЧЕСКИЕ УКАЗАНИЯ

к выполнению курсового проекта по дисциплине «Базы данных» для студентов
направлений 09.03.01 «Информатика и вычислительная техника», 09.03.02
«Информационные системы и технологии» очной формы обучения



+



+



Версия 1.1

Составитель: к.т.н. доц. О.Г. Яскевич
УДК 681.38+681.3

Методические указания к выполнению курсового проекта по дисциплине «Базы данных» для студентов направлений 09.03.01 «Информатика и вычислительная техника», 09.03.02 «Информационные системы и технологии» очной формы обучения / ФГБОУ ВО «Воронежский государственный технический университет»; сост. Д.В. Иванов. Воронеж, 2021. – 26 с.

Методические указания содержат краткие теоретические и практические сведения об основных конструкциях языках С# и их практического применения.

Методические указания подготовлены в электронном виде в текстовом редакторе MS Word 2003 и содержатся в файле SQL КП 1.1.pdf.

Рецензент д-р техн. наук, проф. А.А. Рындин

Ответственный за выпуск зав. кафедрой
д-р техн. наук, проф. Я.Е. Львович

Рекомендовано методическим семинаром кафедры САПРИС и методической комиссией ФИТКБ Воронежского государственного технического университета в качестве методических материалов

© ФГБОУ ВО «Воронежский государственный технический университет», 2021

1. ЦЕЛЬ КУРСОВОГО ПРОЕКТА

Целью курсового проекта является освоение методов проектирования баз данных и работы с базами данных в среде конкретной СУБД.

2. ЗАДАНИЕ НА КУРСОВОЙ ПРОЕКТ

Задание содержит название конкретной предметной области, для которой необходимо:

- сформулировать цель проектирования базы данных;
- описать возможного пользователя базы данных;
- определить круг запросов и задач, которые предполагается решать с использованием созданной базы данных;
- построить концептуальную модель;
- сформулировать требования к базе данных;
- построить логическую модель и выполнить её нормализацию;
- осуществить выбор СУБД и технических средств;
- создать спроектированную базу данных в среде выбранной СУБД;
- разработать приложение для реализации запросов и решения задач;
- оценить базу данных с точки зрения возможностей её дальнейшего развития.

3. МЕТОДИЧЕСКИЕ УКАЗАНИЯ К ВЫПОЛНЕНИЮ КУРСОВОГО ПРОЕКТА

В пояснительной записке к курсовому проекту должны быть представлены следующие разделы:

1. Анализ и описание предметной области
2. Проектирование базы данных (концептуальная, логическая и физическая модель данных)
3. Реализация и разработка клиентского приложения

1. АНАЛИЗ И ОПИСАНИЕ ПРЕДМЕТНОЙ ОБЛАСТИ (в пояснительной записке необходимо указать конкретную предметную область: *Анализ и описание системы управления студенческим общежитием ВУЗа*)

Прежде чем приступить к разработке концептуальной модели данных необходимо:

- сформулировать цель проектирования базы данных;
- описать возможного пользователя базы данных;
- определить круг запросов и задач, которые предполагается решать с использованием созданной базы данных.

В качестве примера рассматривается процесс разработки базы данных для реализации функций управления студенческим общежитием. В этом случае **целью**

курсового проектирования **является** разработка информационной подсистемы, реализующей функции управления студенческим общежитием.

Пользователями данной системы могут быть коменданты общежитий, бухгалтер образовательного учреждения, сотрудники образовательного учреждения, непосредственно задействованные в процессе управления студенческим общежитием.

Задачи, которые необходимо решить при использовании базы данных:

- возможность добавления и удаления студентов в базе общежития;
- возможность редактирования информации в базе общежития;
- возможность заселения и выселения студентов из базы общежития;
- возможность отслеживания количества занятых и доступных для заселения мест в конкретной комнате выбранного общежития.

2. ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ (КОНЦЕПТУАЛЬНАЯ, ЛОГИЧЕСКАЯ И ФИЗИЧЕСКАЯ МОДЕЛЬ ДАННЫХ) (в пояснительной записке необходимо указать конкретную предметную область: *Проектирование базы данных для системы управления студенческим общежитием ВУЗа*)

2.1. Создание концептуальной модели данных

Формирование концептуальной модели данных заключается в подготовке модели данных в рамках конкретного объекта.

Для построения концептуальной модели предметной области рекомендуется придерживаться следующей последовательности действий.

1. Определение основных типов сущностей, присутствующих в представлении предметной области приложения. Документирование выделенных типов сущностей.

В рамках сформулированной цели и решения конкретных задач можно выделить следующие типы сущностей:

Студенты

Обо всех студентах очной формы обучения должны храниться следующие данные: номер и дата приказа на зачисление в образовательное учреждение, имя, отчество и фамилия, домашний адрес, дата рождения, пол, студенческая группа, национальность, текущие жилищные условия (обеспечен местом в общежитии/ожидает поселения), все дополнительные комментарии.

Студенческие общежития

Каждое студенческое общежитие имеет название, адрес, номер телефона, а за его состоянием следит комендант общежития. В общежитиях предоставляются только отдельные комнаты, и в базе данных должны храниться сведения о номере комнаты, количество мест в комнате и ежемесячная арендная плата.

Договора аренды

Студент может арендовать комнату в общежитии на разные периоды времени. Новые договора аренды заключаются в начале каждого академического года, при этом минимальный период аренды составляет один семестр, а

максимальный – один академический год. Каждый отдельный договор между студентом и образовательным учреждением имеет уникальное обозначение в виде номера и даты договора аренды. О каждом договоре аренды должны храниться следующие данные: номер договора аренды, продолжительность периода аренды, приказ на зачисление студента в образовательное учреждение, номер места и комнаты, адрес общежития, дата вселения и выселения.

Счета

О каждом счете хранятся следующие данные: номер счета, номер договора аренды, фамилия и номер приказа о зачислении студента, номер места и комнаты в общежитии, дата оплаты счета.

Сотрудники образовательного заведения

Определенная информация хранится также о сотрудниках образовательного учреждения. Она включает табельный номер сотрудника, имя, отчество и фамилия, должность и место работы (например, факультет).

Группы

В образовательном учреждении храниться также ограниченный набор информации о группах, которые обучаются в учреждении, в частности номер и название группы, год начала обучения, имя, отчество и фамилия декана факультета, на котором обучается группа, телефон.

Ближайшие родственники

По возможности в базу данных вносится информация об одном из ближайших родственников студента, которая включает имя, отчество и фамилию, адрес и номер телефона, по которому можно с ним связаться.

2. Определение важнейших типов связей, существующих между сущностями, выделенными на предыдущем этапе. Применение ER-моделирования для наглядного отображения сущностей и связей. Определение отображения кардинальности, которые распространяются на связи различных типов. Выявление дефектов разветвления и разрывов. Проверка того, участвует ли каждая сущность, по меньшей мере, в одной связи.

3. Сопоставление атрибутов с соответствующими типами сущностей или связей. Выявление простых/составных атрибутов, однозначных/многозначных атрибутов и производных атрибутов.

На этапе проектирования сущность Факультет имеет атрибуты: Название, Сотрудник и Телефон. Так как атрибут сотрудник является многозначным, а использование многозначных атрибутов в реляционной СУБД запрещено, то создадим новую сущность Сотрудник, которая будет иметь два атрибута: Табельный номер и Должность.

4. Определение доменов атрибутов в концептуальной модели данных

5. Определение потенциального ключа для каждого типа сущности, если таковых несколько – определение первичного ключа. Определение альтернативных ключей для каждой сущности.

6. Проверка модели на избыточность, т.е. внимательная проверка связей «один к одному» и удаление избыточных связей.

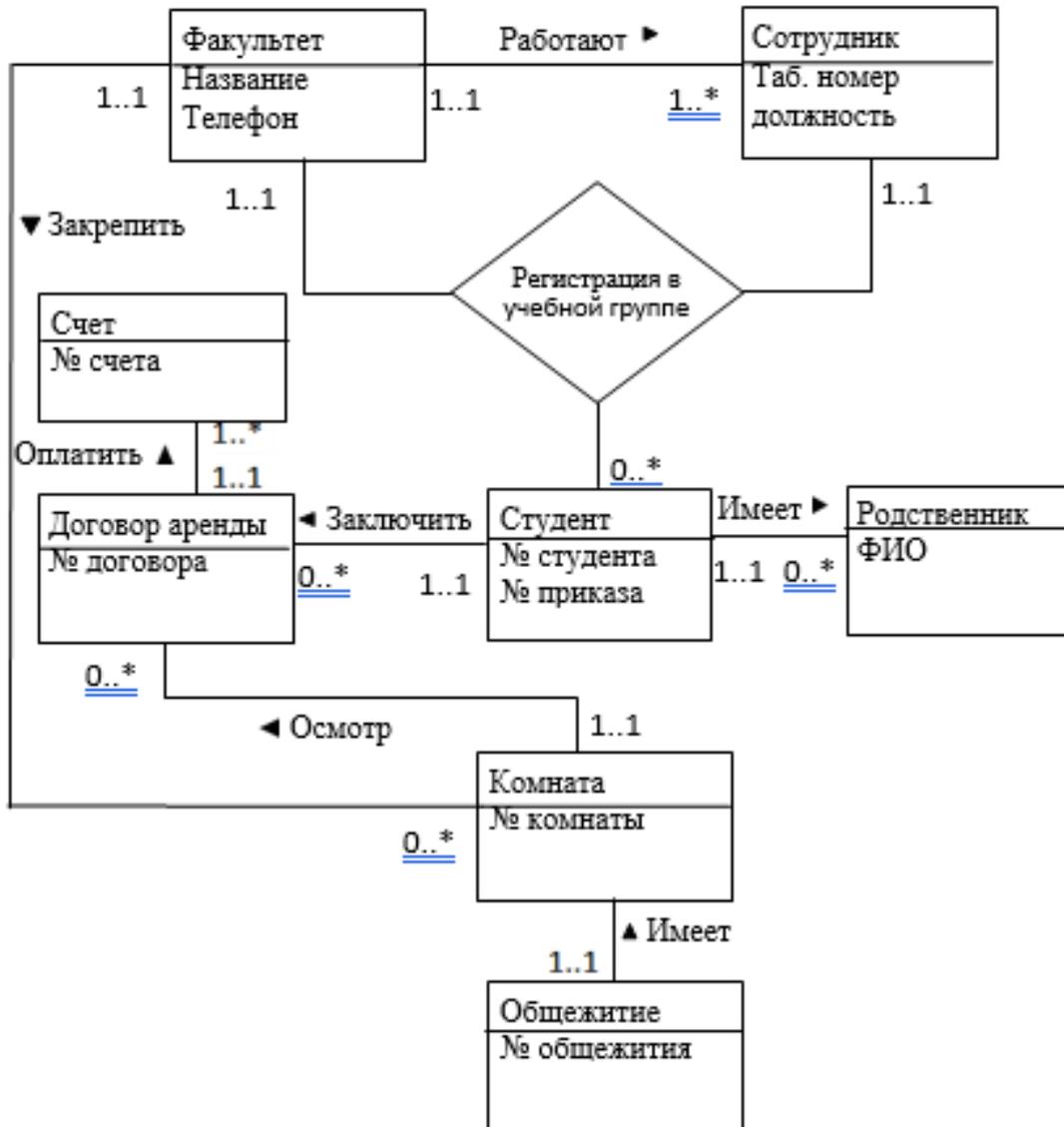


Рис.1 Концептуальная модель данных

7. Проверка того, поддерживает ли концептуальная модель все транзакции, которые требуются для рассматриваемого представления. Для этого выполняется либо описание транзакций, либо использование путей выполнения транзакций.

В рамках рассматриваемого проекта можно выделить следующие транзакции:

1. Предоставить отчет со списком имен комендантов и номеров телефонов каждого студенческого общежития.
2. Составить отчет со списком имен и приказов на зачисление студентов и внести в него сведения о договорах аренды этих студентов.
3. Показать договора аренды, которые охватывают летний семестр.
4. Показать сведения об общей сумме арендной платы, которая была внесена указанным студентом.
5. Составить отчет со сведениями о студентах, которые не оплатили свои счета к указанной дате.
6. Составить отчет со сведениями об именах и приказах на зачисление студентов и включить в него сведения о номерах комнат и мест в конкретном студенческом общежитии.
7. Подготовить отчет и включить в него сведения о студентах, которым в настоящее время еще не предоставлено место в общежитии.
8. Показать общее количество студентов, относящихся к каждой категории.
9. Составить отчет с именами всех студентов, которые не предоставили сведений о своих ближайших родственниках.
10. Показать имя и номер телефона куратора, который закреплен за указанным студентом.
11. Показать общее количество свободных (незаселенных) мест в каждом студенческом общежитии.

2.2. Построение логической модели данных

Для построения логической модели предметной области рекомендуется придерживаться следующей последовательности действий.

1. Доработка концептуальной модели с целью удаления свойств несовместимых с реляционной моделью данных. Удаление двухсторонних связей типа *:*, рекурсивных связей, сложных связей и многозначных атрибутов. (рис. 2)

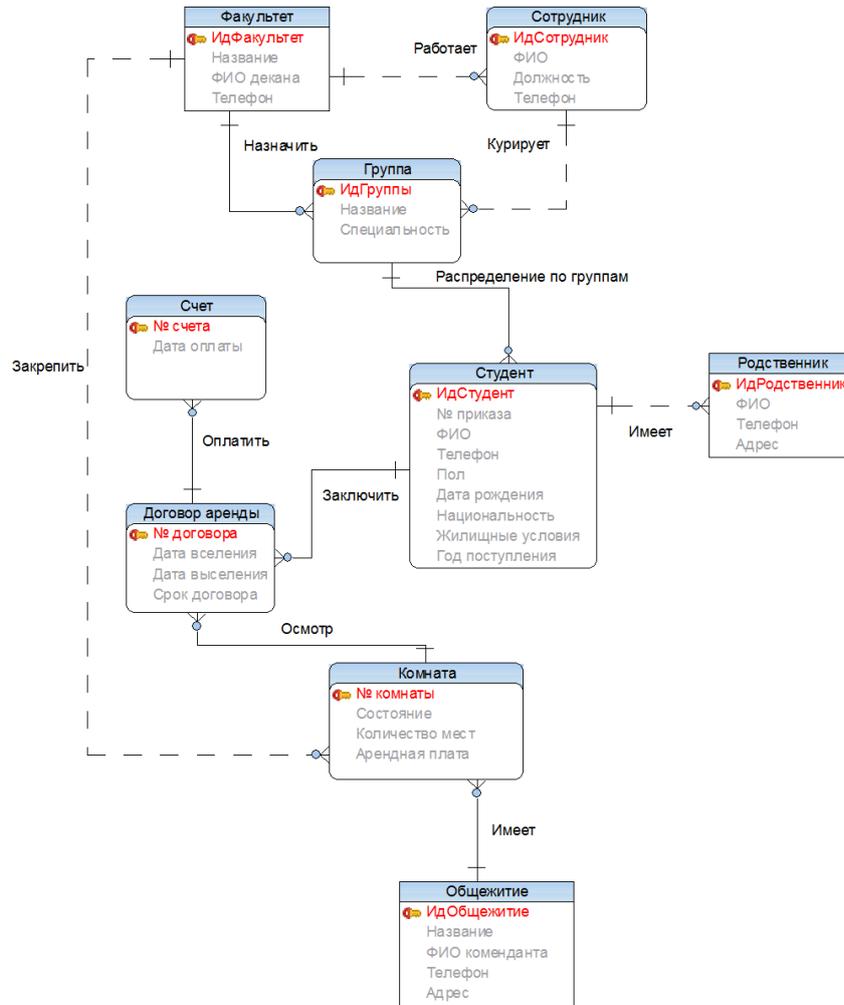


Рис.2 Логическая модель данных

2. Проверка отношений логической модели данных с использованием метода нормализации (третья нормальная форма).

Нормализация – процесс, в результате которого можно избавиться от дефектов проектирования базы данных. В процессе нормализации мы получаем ряд нормальных форм, используя набор определенных правил.

Таблица 10 – Ненормализованная форма

ИД студента	ФИО	Телефон	Пол	Дата рождения	Группа	Факультет	Родственники
1	Иванов Иван Иванович	8908 8581 101	м	01.01.1995	ИС1 21	ФИТ КБ	Иванова Анна Ивановна, Иванов Иван Андреевич

Так как атрибут Родственники содержит список значений, что является недопустимым, то данную таблицу необходимо привести к первой нормальной форме (1НФ).

Таблица находится в 1НФ, если каждое значение атрибута или столбца является атомарным.

В данном случае для приведения к 1НФ достаточно создать новую таблицу, в которой необходимо выделить по одной строке на каждый элемент Родственники.

Таблица 11 – Первая нормальная форма

ИД студента	ФИО	Телефон	Пол	Дата рождения	Группа	Факультет	Родственники
1	Иванов Иван Иванович	8908 8581 101	м	01.01.1995	ИС1 21	ФИТ КБ	Иванова Анна Ивановна
1	Иванов Иван Иванович	8908 8581 101	м	01.01.1995	ИС1 21	ФИТ КБ	Иванов Иван Андреевич

После приведения таблицы к 1НФ необходимо продолжить нормализацию далее, так как в данной форме при добавлении нового родственника придется создавать новую запись с дублированием данных.

Таблица находится во второй нормальной форме (2НФ), если таблица находится в 1НФ и все атрибуты, не являющиеся частью первичного ключа,

полностью функционально зависимы от первичного ключа, т.е. нужно разбить таблицу на такие таблицы, в которых все не ключевые атрибуты будут полностью функционально зависеть от ключа.

В данном случае для приведения ко 2НФ выделим элемент Родственники в отдельную таблицу, которая будет связана с другой таблицей с помощью внешнего ключа ИД студента.

Таблица 12.1 – Вторая нормальная форма таблицы Студент

ИД студента	ФИО	Телефон	Пол	Дата рождения	Группы	Факультет
1	Иванов Иван Иванович	89088581101	м	01.01.1995	ИС121	ФИТКБ

Студент (ИД студента, ФИО, Телефон, Пол, Дата рождения, Группа, Факультет)

Таблица 12.2 – Таблица Родственники

ИД студента	ФИО родственников
1	Иванова Анна Ивановна
1	Иванов Иван Андреевич

Родственники (ИД студента, ФИО родственников)

Нормализацию таблицы 12.1 необходимо продолжить, так как все атрибуты таблицы должны зависеть только от ключа, что является требованием третьей нормальной формы (3НФ).

Таблица находится в 3НФ, если таблица находится во 2НФ и все транзитивные зависимости исключены.

В данном примере таблица 12.1 первичным ключом является ИД студента:

ИД студента -> ФИО, Телефон, Пол, Дата рождения, Группа, Факультет

Но заметим также, что:

ИД студента -> Группа;

ИД студента -> Факультет и Группа -> Факультет.

Обратим внимание, что ИД студента не является первичным ключом, а значит, отношение является транзитивным.

Для устранения транзитивной зависимости в таблице 12.1 мы разбиваем ее на несколько таблиц: студенты, группы, факультеты.

Таблица 13.1 – Третья нормальная форма таблицы Студент

ИД студента	ФИО	Телефон	Пол	Дата рождения	ИД группы
1	Иванов Иван Иванович	89088581101	м	01.01.1995	101

Студент (ИД студента, ФИО, Телефон, Пол, Дата рождения, ИД группы)

Таблица 13.2 – Таблица Группы

ИД группы	Название группы	ИД факультета
101	ИС121	301

Группы (ИД группы, Название группы, ИД факультета)

Таблица 13.3 – Таблица Факультеты

ИД факультета	Название факультета
301	ФИТКБ

Факультет (ИД факультета, Название факультета)

Используя аналогичный алгоритм, нормализуем все таблицы, которые будут задействованы в проекте.

Представленные ниже таблицы 14.1 – 14.9 находятся в 3 НФ.

Таблица 14.1 – Студенты

ИД студента	№ приказа	ФИО	Телефон	Пол	Дата рождения	Национальность	Жилищные условия	Год поступления	ИД группы
1	010121	Иванов Иван Иванович	89088581101	м	01.01.1995	русский	Заселен	2012	201

Студент (ИД студента, № приказа, ФИО, Телефон, Пол, Дата рождения, Национальность, Жилищные условия, Год поступления, ИД группы)

Таблица 14.2 – Группы

ИД группы	Название группы	Специальность	ИД факультета
201	ИС121	Информационные системы	301

Группы (ИД группы, Название группы, Специальность, ИД факультета)

Таблица 14.3 – Факультеты

ИД факультета	Название факультета	ФИО декана	Телефон деканата
301	ФИТКБ	Пасмурнов Сергей Михайлович	2110001

Факультет (ИД факультета, Название факультета, ФИО декана, Телефон деканата)

Таблица 14.4 - Сотрудники

ИД сотрудника	ФИО сотрудника	Должность сотрудника	Телефон сотрудника	ИД факультета
401	Петров Петр Петрович	доцент	230006	301

Сотрудник (ИД сотрудника, ФИО, Должность, Телефон, ИД факультета)

Таблица 14.5 – Родственники

ФИО родственника	Телефон родственника	Адрес родственника	ИД студента
Иванова Анна Ивановна	2350006	Московский проспект 4	101

Родственник (ФИО родственника, Телефон родственника, Адрес родственника, ИД студента)

Таблица 14.6 – Договора аренды

Номер договора	Дата вселения	Дата выселения	Срок договора	ИД студента	Номер комнаты	Номер общежития
0121	2.01.2012	2.06.2016	4	101	5	03

Договор аренды (Номер договора, Дата вселения, Дата выселения, Срок договора, ИД студента, Номер комнаты, Номер общежития)

Таблица 14.7 – Счета

№ счета	Дата оплаты	Номер договора	ИД студента
1121	7.09.2012	0121	101

Счет (№ счета, Дата оплаты, Номер договора, ИД студента)

Таблица 14.8 – Комнаты

Номер комнаты	Состояние	Количество мест	Арендная плата	Номер общежития	ИД факультета
5	Хорошее	2	1000	3	301

Комната (Номер комнаты, Состояние, Количество мест, Арендная плата, Номер общежития, ИД факультета)

Таблица 14.9 – Общежитии

Номер общежития	Название общежития	ФИО коменданта	Телефон общежития	Адрес общежития
3	Общежитие ВГТУ 3	Сидоров Петр Иванович	211011	Московский проспект 174

Общежитие (Номер общежития, Название общежития, ФИО коменданта, Телефон общежития, Адрес общежития)

3. Проверка того, поддерживают ли отношения логической модели данных транзакции, которые требуются в рассматриваемом приложении, для получения гарантий того, что в процессе создания отношений не были допущены ошибки.

4. Выявление ограничений целостности, заданных в рассматриваемом представлении предметной области: определение необходимых данных, установление ограничений для доменов атрибутов, определение целостности сущностей, ссылочной целостности и ограничений предметной области.

5. Определение вероятности внесения каких-либо существенных изменений в созданную модель данных и оценка того, насколько полученная модель позволяет учесть подобные изменения.

2.3. Построение физической модели данных

Физическая модель БД определяет способ размещения данных на носителях (устройствах внешней памяти), а также способ и средства организации эффективного доступа к ним. Поскольку СУБД функционирует в составе и под управлением операционной системы, то организация хранения данных и доступа к ним зависит от принципов и методов управления данными операционной системы.

Способ хранения БД определяется механизмами СУБД автоматически по умолчанию на основе спецификаций концептуальной схемы БД, и внутренняя схема в явном виде в таких системах не используется. Внешние схемы БД обычно конструируются на стадии разработки приложений.

1. Подготовка схемы реляционной базы данных, которая может быть реализована в целевой СУБД на основе логической модели данных. (рис.3)

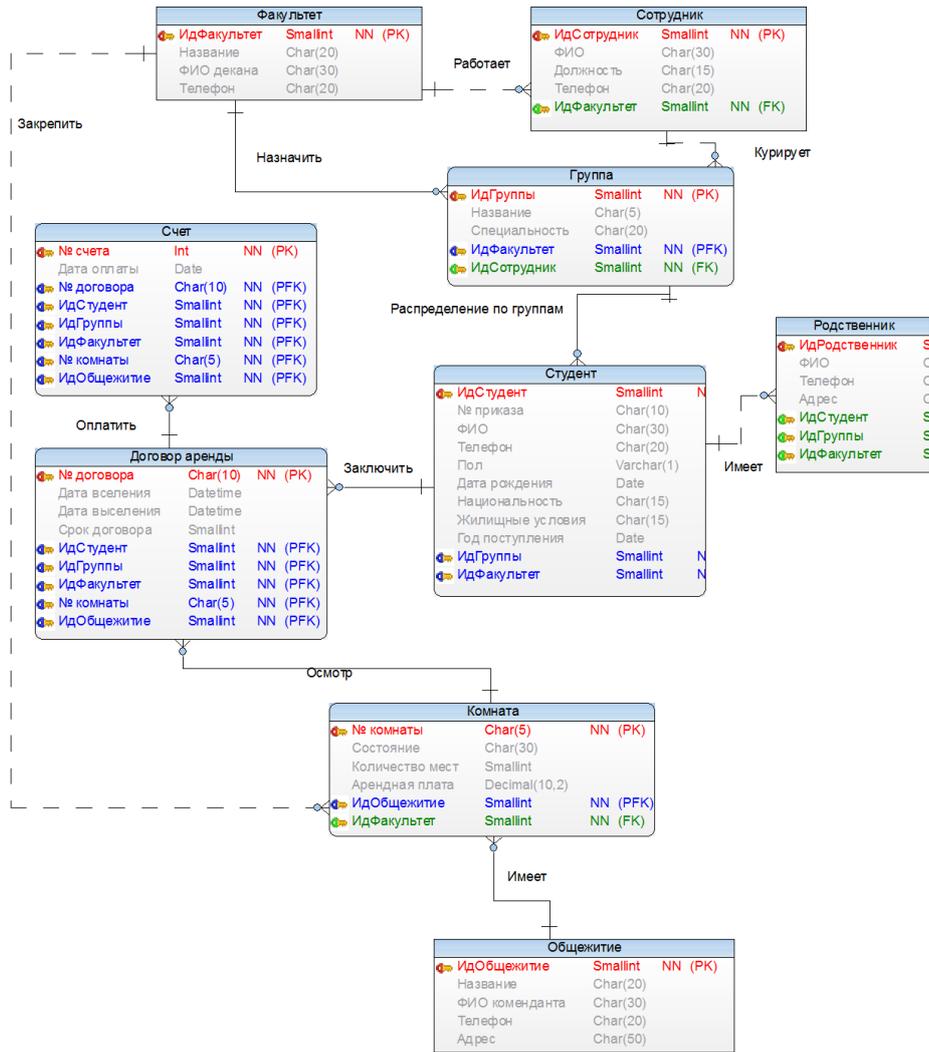


Рис. 3 Физическая модель данных

2. Определение способа представления всех базовых отношений в целевой СУБД.

3. Проектирование ограничений предметной области для целевой СУБД.

3. РАЗРАБОТКА И РЕАЛИЗАЦИЯ КЛИЕНТСКОГО ПРИЛОЖЕНИЯ (в пояснительной записке необходимо указать конкретную предметную область: *Разработка и реализация системы управления студенческим общежитием ВУЗа*)

3.1. Выбор средств реализации программного обеспечения

Данное приложение предназначено для управления студенческим общежитием ВУЗа. Программа позволяет заселять и выселять студентов в общежитии, а также просматривать, редактировать, добавлять и удалять информацию о них. Приложение написано на языке C# в среде разработки – Visual Studio 2013. Отладка производилась в операционной системе MS Windows 7.

3.2. Модульная структура программного обеспечения

Модульная структура разработанного программного средства приведена на рисунке 5.

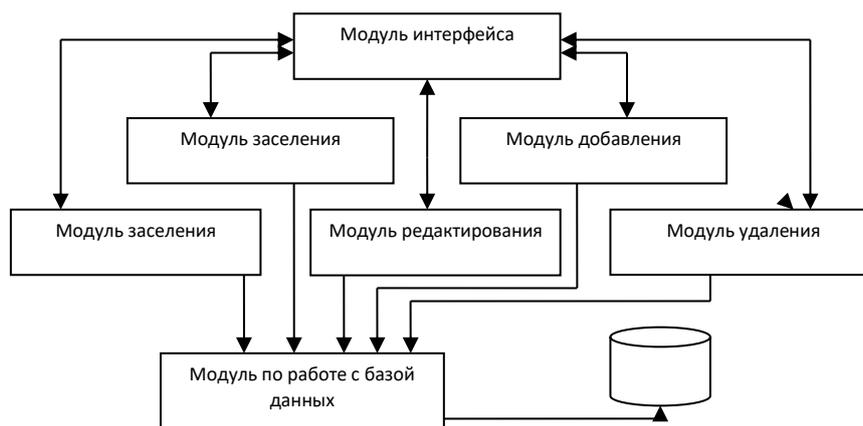


Рисунок 4 - Модульно-иерархическая структура

Далее представлено описание используемых программных модулей:

модуль интерфейса – обеспечивает взаимосвязь между модулями и организует пользовательский интерфейс;

модуль добавления – реализует функцию добавления новой записи о студенте в базу данных;

модуль удаления – реализует функцию удаления выбранной записи о студенте из базы данных;

модуль редактирования – реализует функции изменения информации базы данных;

модуль заселения – реализует функцию заселения студента;

модуль выселения – реализует функцию выселения студента;

модуль по работе с базой данных – реализует соединение с базой данных, а также множество запросов к базе данных.

3.3. Пример работы программного обеспечения

После авторизации в программе в качестве пользователя откроется окно «Контроль общежития».

Первая вкладка «Заселить» предназначена для заселения студента в общежитие. Необходимо выбрать факультет, группу и студента, а также указывать общежитие комнату и срок заключения договора.

Рисунок 6 – Окно пользовательского режима с вкладкой «Заселить»

Запрос на выборку всех факультетов:

```
"SELECT IdFuc,Name FROM faculties"
```

Запрос на выборку групп относительно факультета:

```
"SELECT IdGr,Name FROM groups WHERE IdFuc=" + facId + """
```

Запрос на выборку доступных для заселения студентов относительно группы:

```
"SELECT IdSt,Name FROM students WHERE IdGr=" + groupId + "" and
```

```
Housing="\Не заселен\"
```

Запрос на выборку общежитий:

```
"SELECT IdHos,Name FROM hostel"
```

Выбор доступной для заселения комнаты относительно общежития:

```
"SELECT RoomNumber,Seats FROM rooms WHERE IdHos=" + hosId + """
```

После выбора всех полей необходимо нажать на кнопку «Заселить».

Обработка кнопки «Заселить» происходит следующим образом. Изначально реализуется добавление нового договора:

```
string command = "INSERT INTO contracts VALUES(";
```

```
command += "" + idNewContract.ToString() + ",";
```

```
command += "" + DateTime.Now.ToString("yyyy-MM-dd H:mm:ss")
```

```
+ ",";
```

```
command += "NULL,";  
command += "" + year + ",";"  
command += "" + stId + ",";"  
command += "" + groupId + ",";"  
command += "" + facId + ",";"  
command += "" + roomId + ",";"  
command += "" + hostId + ")";
```

Далее выполняется запрос на проверку существования счета у студента:

```
"SELECT * FROM accounts WHERE IdSt="" + stId + """
```

Если счет не найден, то выполняем запрос на добавление нового счета:

```
command = "INSERT INTO accounts VALUES(";
```

```
command += "" + idNewAcc.ToString() + ",";"  
command += "NULL,";"  
command += "" + idNewContract.ToString() + ",";"  
command += "" + stId + ",";"  
command += "" + groupId + ",";"  
command += "" + facId + ",";"  
command += "" + roomId + ",";"  
command += "" + hostId + ")";
```

Устанавливаем связь между счетом студента и его договором:

```
"UPDATE accounts SET ContractNumber="" + idNewContract + " WHERE  
idSt="" + stId + """
```

Устанавливаем значение «Заселен»:

```
"UPDATE students SET Housing='Заселен' WHERE idSt="" + stId + """
```

Вкладка «Оплата» предназначена для подтверждения оплаты аренды общежития на указанный ранее срок. Для начала необходимо выбрать студента из списка.

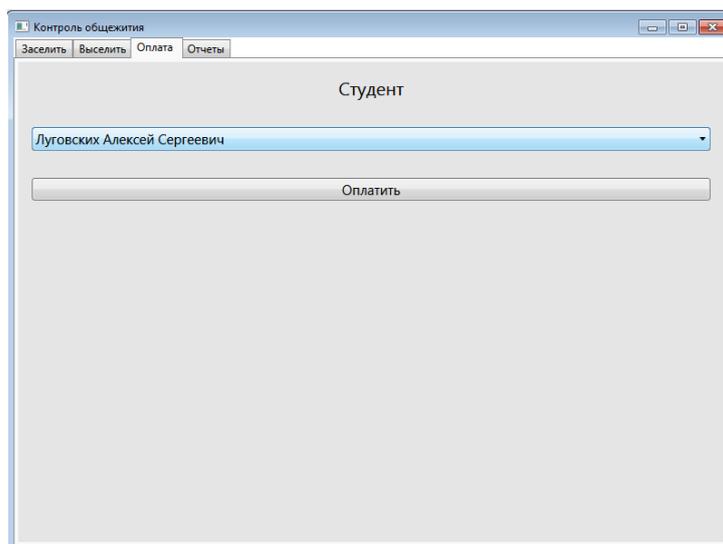


Рисунок 8 – Окно пользовательского режима с вкладкой «Оплатить»

Запрос на выборку студентов, заселенных в общежитие:

```
"SELECT IdSt,Name FROM students WHERE Housing='Заселен'"
```

Для подтверждения оплаты необходимо нажать на кнопку «Оплатить».

Обработка кнопки «Оплатить» происходит следующим образом:

```
"SELECT ContractNumber FROM contracts WHERE IdSt=" + SelectedStud.ID +  
" AND DateEviction is NULL"
```

```
"UPDATE accounts SET PaymentDate=" + DateTime.Now.ToString("yyyy-MM-  
dd H:mm:ss") + " WHERE ContractNumber=" + conNumber);
```

Вкладка «Выселить» предназначена для выселения из общежития выбранного студента. Изначально необходимо выбрать общежитие, номер комнаты и студента.

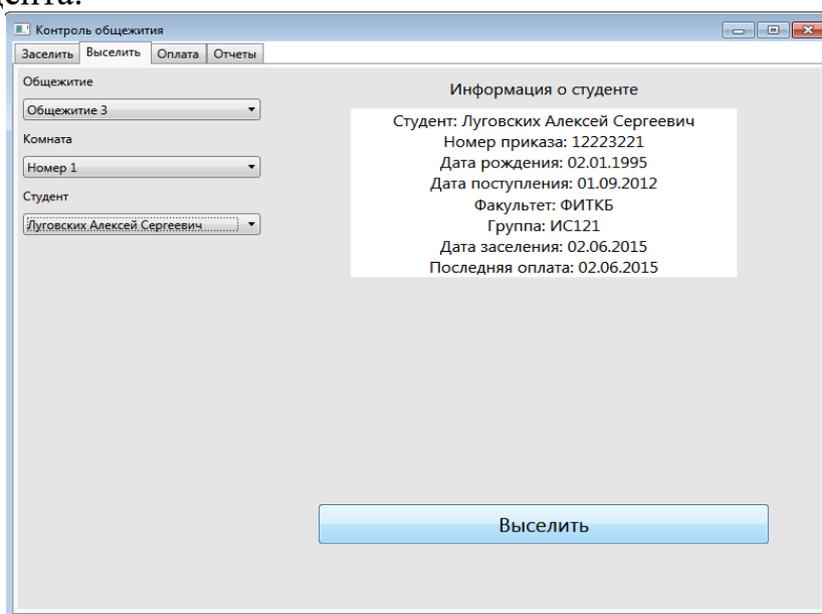


Рисунок 10 – Окно пользовательского режима с вкладкой «Выселить»

Запрос на выборку всех факультетов:

```
"SELECT IdHos,Name FROM hostel"
```

Запрос на выборку комнаты относительно общежития с проверкой на заселение:

```
"SELECT RoomNumber FROM rooms WHERE IdHos=" + hostelId + ""
```

```
"SELECT ContractNumber FROM contracts WHERE RoomNumber=" +  
dr[0].ToString() + " AND DateEviction is NULL";
```

Запрос на выборку доступных для выселения студентов относительно комнаты с проверкой оплаты:

```
"SELECT IdSt FROM contracts WHERE RoomNumber=" + roomId + " AND  
DateEviction is NULL"
```

```
"SELECT * FROM accounts WHERE IdSt=" + dr[0].ToString() + " AND  
PaymentDate is NULL"
```

Для подтверждения операции выселения необходимо нажать на кнопку «Выселить».

Обработка кнопки «Оплатить» происходит следующим образом:

```
"UPDATE students SET Housing='Не заселен' WHERE idSt=" + student.Id + "";
```

```
"UPDATE contracts SET DateEviction=" + DateTime.Now.ToString("yyyy-MM-  
dd H:mm:ss") + " WHERE idSt=" + student.Id + " AND DateEviction is NULL";
```

На следующей вкладке «Отчеты» можем наблюдать основные транзакции, представленные в данном проекте. Для просмотра отчета необходимо нажать на кнопку с интересующей вас транзакцией или также ввести исходные данные, если это необходимо для его формирования. Далее можно нажать на кнопку «Отчет» для получения отчета с помощью FastReport.Net.

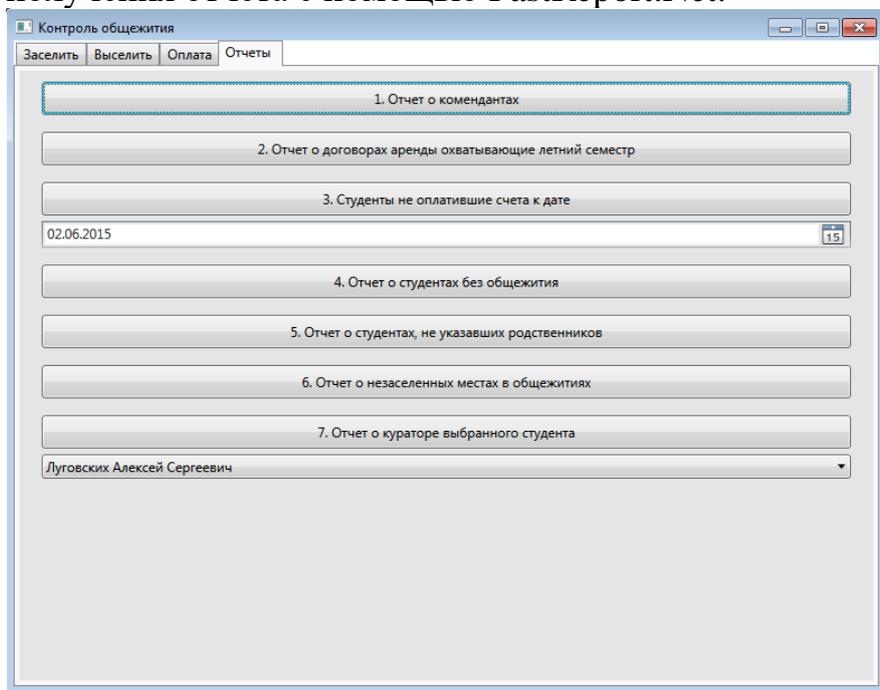


Рисунок 12 – Окно пользовательского режима с вкладкой «Отчеты»

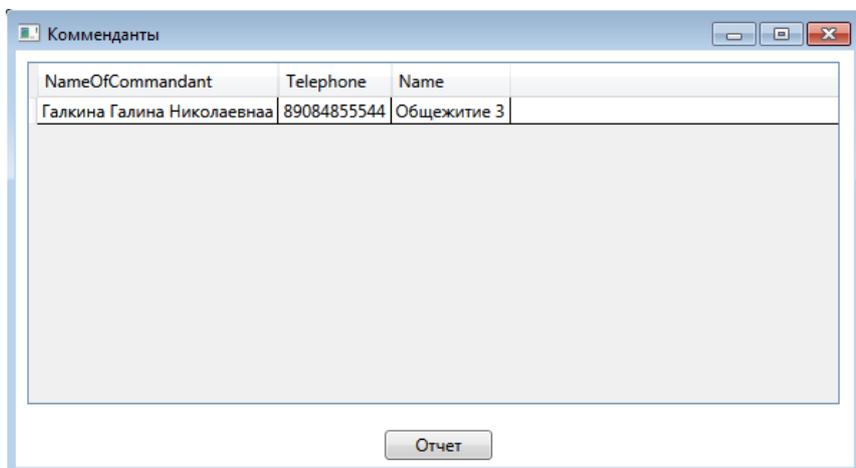


Рисунок 13 – Результат отчета о комендантах

Запрос на выбор имен комендантов и номеров телефонов каждого студенческого общежития:

```
"SELECT NameOfCommandant,Telephone,Name FROM hostel";
```

ContractNumber	DateAvailable	PeriodOfContract	StudentName	Group	Faculty	RoomNumber	Hostel
5	26.05.2015 23:24:37	4	Кочеткова Юлия Александровна	ИС121	ФИТКБ	1	Общежитие 3
6	26.05.2015 23:25:56	1	Литвинов Павел Иванович	АП121	ФИТКБ	12	Общежитие 3

Рисунок 14 – Результат о договорах аренды, охватывающих летний семестр

Запрос на выбор договоров аренды, которые охватывают летний семестр:

```
"SELECT ContractNumber,DateAvailable,PeriodOfContract," +
      "(SELECT      NAME      FROM      students      where
students.IdSt=contracts.IdSt) as 'StudentName'," +
      "(SELECT      NAME      FROM      groups      where
groups.IdGr=contracts.IdGr) as 'Group'," +
      "(SELECT      NAME      FROM      faculties      where
faculties.IdFuc=contracts.IdFuc) as 'Faculty'," +
      "RoomNumber," +
      "(SELECT      NAME      FROM      hostel      where
hostel.IdHos=contracts.IdHos) as 'Hostel' " +
      " FROM contracts WHERE DateEviction is NULL";
```

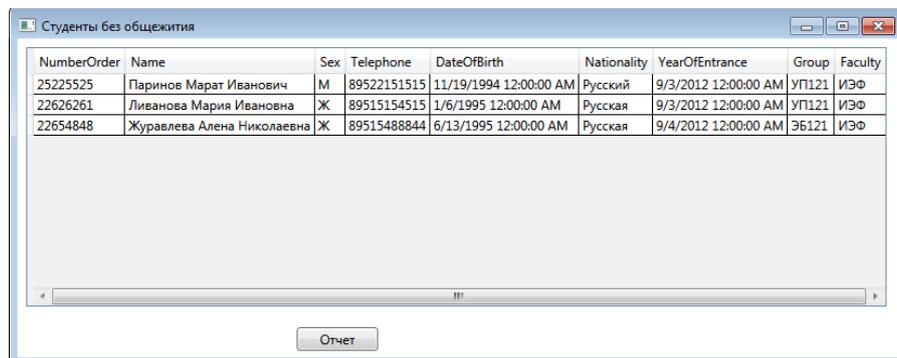
StudentName	Group	Faculty	RoomNumber	Hostel	AccountNumber	PaymentDate	ContractNumber
Луговских Алексей Сергеевич	ИС121	ФИТКБ	1	Общежитие 3	1	6/3/2015 12:00:00 AM	8

Рисунок 15 – Результат отчета о неоплаченных счетах к 2.06.2015

Запрос на выбор сведений о студентах, которые не оплатили свои счета к указанной дате:

```
"SELECT " +
      "(SELECT      NAME      FROM      students      where
students.IdSt=accounts.IdSt) as 'StudentName'," +
      "(SELECT      NAME      FROM      groups      where
groups.IdGr=accounts.IdGr) as 'Group'," +
      "(SELECT      NAME      FROM      faculties      where
faculties.IdFuc=accounts.IdFuc) as 'Faculty'," +
      "RoomNumber," +
      "(SELECT      NAME      FROM      hostel      where
hostel.IdHos=accounts.IdHos) as 'Hostel'," +
```

```
"AccountNumber,PaymentDate,ContractNumber FROM
accounts WHERE PaymentDate > "" + date.ToString("yyyy-MM-dd H:mm:ss") + "" OR
PaymentDate is NULL";
```

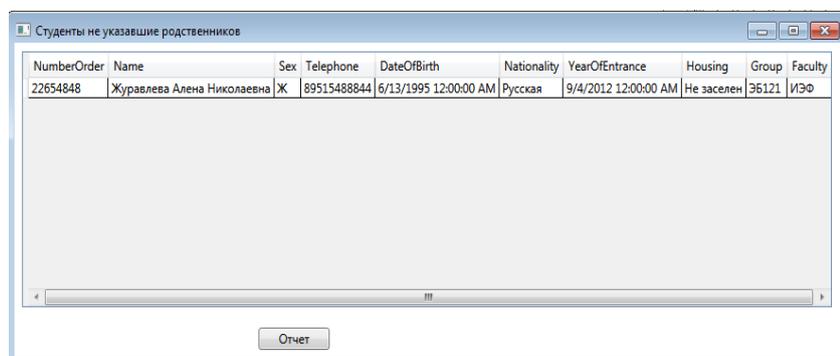


NumberOrder	Name	Sex	Telephone	DateOfBirth	Nationality	YearOfEntrance	Group	Faculty
25225525	Паринов Марат Иванович	М	89522151515	11/19/1994 12:00:00 AM	Русский	9/3/2012 12:00:00 AM	УП121	ИЭФ
22626261	Ливанова Мария Ивановна	Ж	89515154515	1/6/1995 12:00:00 AM	Русская	9/3/2012 12:00:00 AM	УП121	ИЭФ
22654848	Журавлева Алена Николаевна	Ж	89515488844	6/13/1995 12:00:00 AM	Русская	9/4/2012 12:00:00 AM	ЭБ121	ИЭФ

Рисунок 16 – Результат отчета о студентах без общежития

Запрос на выбор сведений о студентах, которым в настоящее время еще не предоставлено место в общежитии:

```
"SELECT
NumberOrder,Name,Sex,Telephone,DateOfBirth,Nationality,YearOfEntrance," +
"(SELECT NAME FROM groups where
groups.IdGr=students.IdGr) as 'Group'," +
"(SELECT NAME FROM faculties where
faculties.IdFuc=students.IdFuc) as 'Faculty' " +
"FROM students WHERE Housing='Не заселен'";
```



NumberOrder	Name	Sex	Telephone	DateOfBirth	Nationality	YearOfEntrance	Housing	Group	Faculty
22654848	Журавлева Алена Николаевна	Ж	89515488844	6/13/1995 12:00:00 AM	Русская	9/4/2012 12:00:00 AM	Не заселен	ЭБ121	ИЭФ

Рисунок 17 – Результат отчета о студентах, не указавших родственников

Запрос на выбор имен всех студентов, которые не предоставили сведений о своих ближайших родственниках:

```
"SELECT
NumberOrder,Name,Sex,Telephone,DateOfBirth,Nationality,YearOfEntrance,Housing,
" +
"(SELECT NAME FROM groups where
groups.IdGr=students.IdGr) as 'Group'," +
"(SELECT NAME FROM faculties where
faculties.IdFuc=students.IdFuc) as 'Faculty' " +
```

"FROM students WHERE (SELECT COUNT(*) FROM parants WHERE parants.IdSt=students.IdSt)=0";

Hostel	Clear Seats
Общежитие 3	9

Рисунок 18 – Результат отчета о количестве свободных мест в общежитии

Запрос на выбор общежитий и количества свободных мест в каждом:

```
string command = "SELECT RoomNumber,Seats FROM rooms WHERE IdHos=" + hosId + "";
```

```
DataTable dt = SelectExecution(command);
```

```
int num = 0;
```

```
foreach (DataRow dr in dt.Rows)
```

```
{
```

```
string checkCommand = "SELECT ContractNumber FROM contracts WHERE RoomNumber=" + dr[0].ToString() + " AND DateEviction is NULL";
```

```
var checkDT = SelectExecution(checkCommand);
```

```
int numOfPeopleInRoom = checkDT.Rows.Count;
```

```
num += (Convert.ToInt32(dr[1].ToString()) - numOfPeopleInRoom);
```

```
}
```

Name	Telephone
Королев Евгений Николаевич	89084578965

Рисунок 19 – Результат отчета о кураторе выбранного студента

Запрос на выбор имени и номера телефона куратора, который закреплен за указанным студентом:

```
command = "SELECT IdGr FROM students WHERE IdSt=" + (comboReport10.SelectedItem as ComboBoxMyItem).ID;
```

```
grId =
MySQLController.SelectExecution(command).Rows[0][0].ToString();

command = "SELECT IdEmp FROM groups WHERE IdGr=" + grId;
empId =
MySQLController.SelectExecution(command).Rows[0][0].ToString();

command = "SELECT Name,Telephone FROM employees WHERE IdEmp="
+ empId;
```

Если же авторизация в программе была произведена в качестве администратора системы, откроется окно, представленное на рисунке 20.

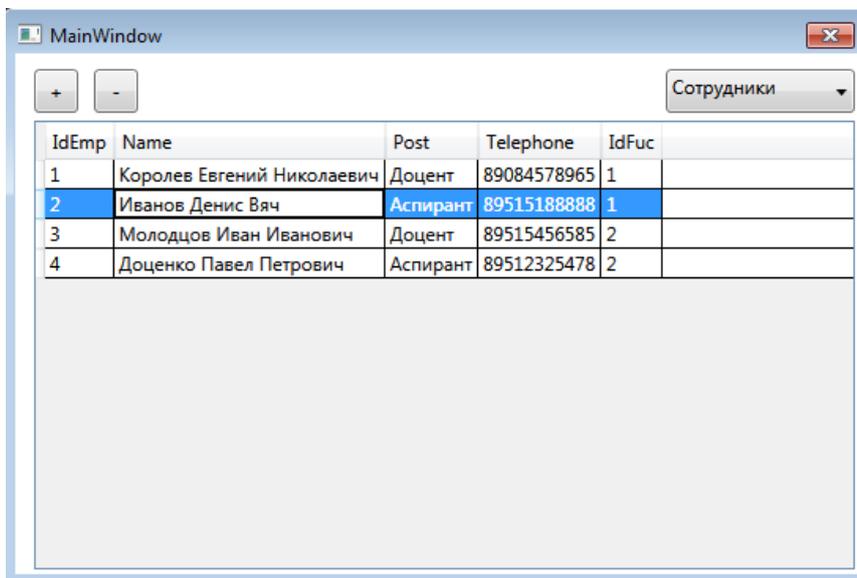


Рисунок 20 – Окно в интерфейсе администратора

В данном интерфейсе администратор может просматривать все таблицы базы данных. Для этого необходимо выбрать нужную таблицу в правом верхнем углу.

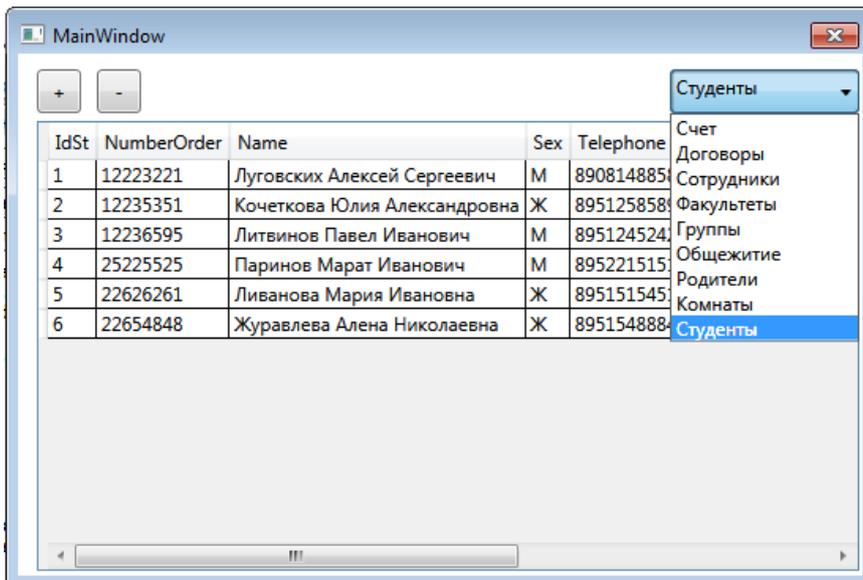


Рисунок 22 – Выбор нужной таблицы в интерфейсе администратора

Функция просмотра таблиц в базе:

```
string table = MainCB.SelectedValue.ToString();
selectedDataTable = MySqlController.SelectExecution("SELECT *
FROM " + table);
MainGrid.ItemsSource = selectedDataTable.DefaultView;
selectTable = table;
```

В интерфейсе администратора существует возможность редактирования информации в полях базы, так как реализована возможность обновления конкретного поля в конкретной таблице с форматированием конкретных данных:

```
command = "UPDATE " + selectTable + " SET ";

string setStr = "";
if (editRow[indexOfEditCell].ToString() == "")
    setStr += selectedDataTable.Columns[indexOfEditCell].Caption +
"=NULL";
else
    if (editRow[indexOfEditCell] is DateTime)
        setStr += selectedDataTable.Columns[indexOfEditCell].Caption +
"=" + ((DateTime)editRow[indexOfEditCell]).ToString("yyyy-MM-dd H:mm:ss") + " ";
    else
        if (editRow[indexOfEditCell] is Decimal)
            setStr += selectedDataTable.Columns[indexOfEditCell].Caption +
"=" + editRow[indexOfEditCell].ToString().Replace(',', '.') + " ";
        else
            setStr += selectedDataTable.Columns[indexOfEditCell].Caption +
"=" + editRow[indexOfEditCell].ToString() + " ";
command += setStr + " WHERE " +
selectedDataTable.Columns[0].Caption + "=" + info[0] + " ";
```

В данном интерфейсе реализована возможность добавления нового студента в базу. Для этого нужно выбрать таблицу «Студенты» в правом верхнем углу окна и нажать на кнопку с изображением «+». После данной операции откроется новое окно с полями для заполнения данных о новом студенте (рис. 22).

Рисунок 22 – Окно заполнения данных о новом студенте

После выбора всех полей необходимо нажать на кнопку «Добавить».

Обработка кнопки «Добавить» происходит следующим образом:

```
command = "INSERT INTO students
(NumberOrder,Name,Sex,Telephone,DateOfBirth,Nationality,Housing,YearOfEntrance
,IdGr,IdFuc) values(";
```

```
command += "" + ContractNumber + ",";
command += "" + FIO + ",";
command += "" + Sex.Substring(("System.Windows.Controls.ComboBoxItem: ").Length,1) + ",";
command += "" + Telephone + ",";
command += "" + BirthDate.ToString("yyyy-MM-dd H:mm:ss") + ",";
command += "" + Nationality + ",";
command += "" + "Не заселен" + ",";
command += "" + DateEntrance.ToString("yyyy-MM-dd H:mm:ss") + ",";
command += "" + SelectedGroup.ID + ",";
command += "" + SelectedFuc.ID + """;
```

В интерфейсе администратора существует возможность удаления студента из базы. Для этого необходимо в таблице «Студенты» навести курсор на выбранного студента, и нажать на кнопку с изображением «-». После чего положительно ответив на вопрос об удалении в новом всплывающем окне (рис. 24) студент будет удален из базы.

Обработка удаления записи происходит следующим образом:

```
"DELETE FROM " + selectTable + " WHERE " +
selectedDataTable.Columns[0].Caption + "=" + row[0].ToString());
```

Результаты проделанной работы должны быть представлены в виде пояснительной записки и приложения, работу которого необходимо продемонстрировать непосредственно на компьютере.

Оформление пояснительной записки должно быть выполнено в соответствии со стандартом предприятия, изложенными в методических указаниях по оформлению курсовых работ, курсовых и дипломных проектов.

При выполнении курсового проекта необходимо использовать материал лекций и рекомендуемую учебно-методическую литературу.

5. РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

1. Веллинг Л. MySQL. Учебное пособие.: Пер. с англ. М.: Издательский дом: «Вильямс», 2005.
2. Дейт Дж.К. Введение в системы баз данных, 1980
3. Советов Б.Я. Базы данных, 2005
4. СТП 62-2007