

Федеральное агентство связи
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Поволжский государственный университет телекоммуникаций и информатики»

Факультет

Информационных технологий

Направление
(специальность)

Информатика и вычислительная техника

Кафедра

Информатики и вычислительной техники

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА
(ДИПЛОМНАЯ РАБОТА)

Разработка программного модуля "Мониторинг
эффективности вуза"

Утверждаю	зав. кафедрой	д.т.н., проф.	Н. Ф. Бахарева		
	Должность	Уч. степень, звание	Подпись	Дата	Инициалы Фамилия
Руководитель	ассистент		М. А. Марина		
Н. контролер	ассистент		Е.С. Карташевская		
Разработал	ПО-21		А. Э. Кромников		
	Группа		Подпись	Дата	Инициалы Фамилия

ВКР проверку на антиплагиат прошла

Подпись Дата Инициалы Фамилия

Самара 2016

Содержание

Содержание	1
ЗАДАНИЕ	Error! Bookmark not defined.
ОТЗЫВ РУКОВОДИТЕЛЯ	7
ПОКАЗАТЕЛИ КАЧЕСТВА ВКР	9
РЕЦЕНЗИЯ.....	10
Введение	10
1.Исследовательский раздел.....	13
1.1 Анализ предметной области	13
1.2 Необходимость автоматизации составления отчетов по профессорско-преподавательскому составу.....	17
1.3 Анализ аналогов	19
1.4 Обоснование необходимости разработки приложения	22
1.5 Обоснование выбора языка программирования.....	23
1.5 Обоснование выбора языка программирования.....	Error! Bookmark not defined.
1.6 Обоснование выбора СУБД.....	24
1.7 Постановка задачи.....	28
1.8 Определение функциональных требований	29
2. Разработка приложения «Управление ППС ПГУТИ».....	32
2.1 Разработка функциональной схемы	32
2.2 Разработка интерфейса приложения	33
2.3 ER – диаграмма	38
2.4 Укрупненная схема алгоритма приложения	39
2.5 Защита информации	40
3. Технологический раздел	43

3.1 Руководство пользователя	43
3.2 Методика проведения тестирования и результаты	53
3.3 Внедрение приложения «Управление ППС ПГУТИ»	58
Заключение.....	60
Список использованных источников	62
Приложение А – Программный код.....	65
Приложение Б – Презентационный материал	Error! Bookmark not defined.

3 Исходные данные и постановка задачи

4 Перечень подлежащих разработке в ВКР вопросов или краткое
содержание ВКР. Сроки исполнения 30.04.16

5 Перечень графического материала. Сроки исполнения 30.04.16
1) Презентационный материал

6 Дата выдачи задания « 15 » декабря 2015 г.

Кафедра Информатики и вычислительной техники

Утверждаю зав.кафедрой д.т.н., проф. 15.12.15 Н. Ф. Бахарева

Должность	Уч.степень, звание	Подпись	Дата	Инициалы Фамилия
-----------	--------------------	---------	------	------------------

Руководитель ассистент М. А. Марина

Должность	Уч.степень, звание	Подпись	Дата	Инициалы Фамилия
-----------	--------------------	---------	------	------------------

Задание принял
к исполнению ПО-21 15.12.15 А. Э. Кромников

Группа	Подпись	Дата	Инициалы Фамилия
--------	---------	------	------------------

Федеральное агентство связи
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Поволжский государственный университет телекоммуникаций и информатики»

ОТЗЫВ РУКОВОДИТЕЛЯ

Тип ВКР	<u>Дипломная работа</u>
Студента(ки)	<u><i>Кромникова Алексея Эдуардовича</i></u>
Специальность/ направление	<u><i>Информатика и вычислительная техника</i></u>
Тема ВКР	<u><i>Разработка программного модуля "Мониторинг эффективности вуза"</i></u>
Руководитель	<u>М. А. Марина</u>
Ученая степень, звание	<u></u>
Место работы (должность)	<u>ПГУТИ, кафедра. ИВТ, ассистент</u>

АКТУАЛЬНОСТЬ ТЕМЫ

Дипломная работа Кромникова А. Э. посвящена разработке приложения «Управление профессорско-преподавательским составом ПГУТИ». С помощью данного приложения можно значительно сократить временные и трудовые ресурсы учебного отдела при составлении ежегодных отчетов для Министерства образования и науки РФ. Приложение позволяет анализировать информацию на разных временных срезах.

ОЦЕНКА СОДЕРЖАНИЯ РАБОТЫ

(Структура, логика и стиль изложения представленного материала. глубина и степень проработки материала, обоснованность изложенных выводов, использование математического аппарата, использование средств вычислительной техники, макетирование, моделирование, экспериментирование)

В дипломной работе достаточно подробно и ясно описаны такие разделы как: анализ предметной области, анализ аналогов, обоснование выбора языка программирования, обоснование выбора СУБД, разработка функциональной схемы, разработка интерфейса приложения и предоставлены исходные коды приложения.

Также описана разработка системы с момента разработки схемы базы данных, до готового к использованию продукта.

СТЕПЕНЬ ДОСТИЖЕНИЯ ЦЕЛИ И ПРАКТИЧЕСКАЯ ЗНАЧИМОСТЬ

(Полнота раскрытия исследуемой темы, практическая ценность и возможность внедрения)

По результатам работы Кромникова А.Э., был получен готовый программный продукт. Пояснительная записка достаточно подробно описана и содержит весь необходимый материал для обучения персонала. Систему можно считать законченным продуктом и использовать по назначению.

ЗАКЛЮЧЕНИЯ ПО ПРЕДСТАВЛЕННОЙ РАБОТЕ

(Степень самостоятельной работы студента; совокупная оценка труда студента и его квалификация)

При выполнении работы, связанной с тематикой дипломной работы, Кромников А.Э. проявил себя как хороший программист, способный решить самостоятельно поставленные перед ним задачи. В ходе выполнения работы продемонстрирован достаточный уровень знаний и специальной подготовки в профессиональной области.

Дипломная работа соответствует требованиям, предъявляемым к ВКР и заслуживает оценки «отлично».

Руководитель ВКР

Подпись

Дата

М. А. Марина
Инициалы Фамилия

Федеральное агентство связи
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Поволжский государственный университет телекоммуникаций и информатики»

ПОКАЗАТЕЛИ КАЧЕСТВА ВКР

По ВКР студента	<u>Кромникова Алексея Эдуардовича</u>		
На тему	<u>Разработка программного модуля</u> <u>"Мониторинг эффективности вуза"</u>		
1 Работа выполнена:			
- по теме, предложенной студентом	✓		
- по заявке предприятия	✓	<u>ПГУТИ</u>	наименование предприятия
- в области фундаментальных и поисковых научных исследований	□		указать область исследований
2 Результаты ВКР:			
- рекомендованы к опубликованию	□		указать где
- рекомендованы к внедрению	✓	<u>Учебный отдел ПГУТИ</u>	указать где
- внедрены	□		акт внедрения
3 ВКР имеет практическую ценность	✓	<u>Управление ППС ПГУТИ</u>	в чем заключается практическая ценность
4 Использование ЭВМ при выполнении ВКР: (ПО, компьютерное моделирование, компьютерная обработка данных и др.)	✓		
5. ВКР прошла проверку на объем заимствований	✓	<u>% заимствований</u>	

Студент	<u>ПО-21</u>	<u>А. Э. Кромников</u>		
	Группа	Подпись	Дата	Инициалы Фамилия
Руководитель ВКР	<u>ассистент</u>	<u>М. А. Марина</u>		
	Должность	Уч.степень, звание	Подпись	Дата
				Инициалы Фамилия

Федеральное агентство связи
Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Поволжский государственный университет телекоммуникаций и информатики»

РЕЦЕНЗИЯ

Тип ВКР	<i>Дипломная работа</i>
Студента(ки)	<i>Кромникова Алексея Эдуардовича</i>
Специальность/ направление	<i>Информатика и вычислительная техника</i>
Тема ВКР	<i>Разработка программного модуля "Мониторинг эффективности вуза"</i>
Рецензент	<i>Бахарева Надежда Федоровна</i>
Ученая степень, звание	<i>профессор, д.т.н</i>
Место работы (должность)	<i>ПГУТИ кафедра ИВТ, зав. кафедрой</i>

АКТУАЛЬНОСТЬ ТЕМЫ

Дипломная работа Кромникова А. Э. посвящена разработке приложения «Управление профессорско-преподавательским составом ПГУТИ». С помощью данного приложения можно значительно сократить временные и трудовые ресурсы учебного отдела при составлении ежегодных отчетов для Министерства образования и науки РФ. Приложение позволяет анализировать информацию на разных временных срезах.

ОЦЕНКА СОДЕРЖАНИЯ РАБОТЫ

(Структура, логика и стиль изложения представленного материала.глубина и степень проработки материала, обоснованность изложенных выводов, использование математического аппарата, использование средств вычислительной техники, макетирование, моделирование, экспериментирование)

ДОСТОИНСТВА И НЕДОСТАТКИ РАБОТЫ

СТЕПЕНЬ ДОСТИЖЕНИЯ ЦЕЛИ И ПРАКТИЧЕСКАЯ ЗНАЧИМОСТЬ

(Полнота раскрытия исследуемой темы; практическая ценность и возможность внедрения)

ЗАКЛЮЧЕНИЯ ПО ПРЕДСТАВЛЕННОЙ РАБОТЕ

(Совокупная оценка труда студента и его квалификация)

Рецензент

Подпись

Дата

Инициалы Фамилия

Введение

Каждый год объем обрабатываемой информации растет, вынуждая тратить на свою обработку все больше временных и трудовых затрат. В настоящее время большинство задач невозможно решить без применения современных вычислительных систем и программных комплексов. В связи с этим, потребность в автоматизированные программных комплексах, которые способны за малые сроки обрабатывать большой объём данных и предоставлять ее в удобном для пользователя виде, возрастает.

К данным системам можно отнести автоматизированные системы управления, неотъемлемой частью которых, являются реляционные базы данных (БД), которые хранят информацию в виде таблиц. Персонал учебных отделов тратит много времени на составление отчётов из большого объёма информации. Поэтому появляется необходимость внедрить автоматизированную систему управления отчетностью ВУЗа, которая бы автоматизировала часть деятельности учебного отдела. Однако в условиях экономического кризиса государство выделяет не достаточно средств на покупку готового программного обеспечения, которое легко можно было бы внедрить в ВУЗ.

Другим вариантом решения данной проблемы является разработка данного программного продукта, позволяющего упростить и значительно сократить временные и трудовые ресурсы при обработке информации по профессорско-преподавательскому составу (ППС), также позволяющего анализировать информацию на разных временных срезах, что обуславливается **актуальностью** выбранной темы исследования «Разработка программного модуля "Мониторинг эффективности вуза"». Разработка такого вида программного продукта требует изучения специфики работы учебного учреждения.

Объектом исследования для данной дипломной работы является учебный отдел Поволжского государственного университета телекоммуникаций и информатики (ПГУТИ).

Предметом рассмотрения является деятельность учебного отдела ПГУТИ, связанная с формированием ежегодных отчетов и анализом информации по профессорско-преподавательскому составу (ППС).

Целью дипломной работы является проектирование, реализация и последующее внедрение программного продукта для автоматизации деятельности учебного отдела вуза.

Для достижения поставленных целей необходимо решить следующие **задачи:**

1. Изучить теорию управления образовательными учреждениями и вузами.
2. Проанализировать деятельность учебного отдела ПГУТИ, связанную с управлением ППС.
3. Разработать формы отчетов по ППС.
4. Спроектировать информационную систему.
5. Разработать схему БД управлением отчётностью ППС.
6. Реализовать программное обеспечение системы.

В рамках данного дипломного проекта были использованы следующие методы:

- системный анализ;
- сравнительный анализ
- изучение и анализ научной литературы;
- классификация - разделение отдельных изучаемых предметов на группы;

- дедукция - вид умозаключения от абстрактного к конкретному, от общего к частному;
- конструирование и проектирование.
- моделирование;

Практическая значимость данной дипломной работы состоит в том, что позволит сократить время формирования отчетов по ППС на разных временных срезах.

1 Исследовательский раздел

1.1 Анализ предметной области

Предметная область - совокупность связанных между собой функций, задач управления, с помощью которых достигается выполнение поставленных целей, это часть реального мира, представляющая интерес для конкретного исследования [10].

УОУП ПГУТИ является структурным подразделением ВУЗа. Работа учебного отдела построена на основе типовой вузовской документации и регламентируемая соответствующими годовыми планами. Учебный отдел также взаимодействует с другими подразделениями образовательного учреждения. Рассмотрим данную связь на нижеперечисленных процессах:

- УОУП координирует учебный процесс факультетов и кафедр;
- УОУП ПГУТИ своевременно представляет кафедрам и факультетам документацию, необходимую для ведения учебного процесса;
- факультетам и кафедрам необходимо вовремя представить в учебный отдел определенные сведения и учебно-методическую документацию, связанную с учебным процессом и работой ППС ПГУТИ;
- учебный отдел выдаёт разрешение преподавателям кафедр на работу по совместительству и по трудовым соглашениям;
- учебный отдел ежегодно предоставляет отчеты ректорату и Министерству образованию РФ [2].

Итак, можно увидеть активную взаимосвязь УОУП с кафедрами и факультетами учебного заведения, что способствует перемещению информации от одного субъекта образовательного пространства к другому и обеспечению организации внутри учреждения. На рис. 1.1 представим в виде модели.

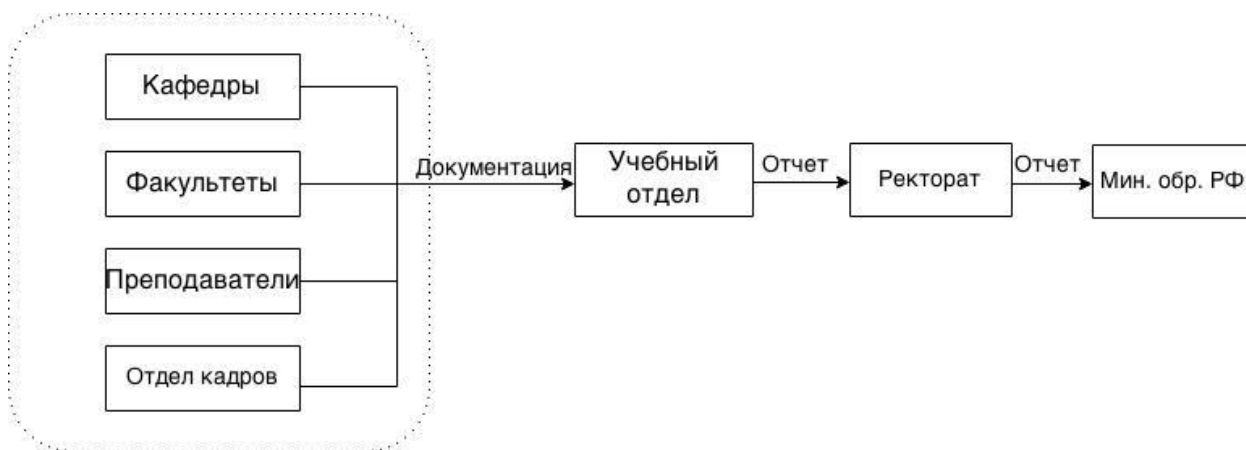


Рис. 1.1 – Модель информационных потоков

Контроль по организации учебного процесса в ВУЗе осуществляется ректоратом, начальником УОУП, деканами, заведующими кафедрами.

В обязанности начальника УОУП входит:

- контроль соответствия расписания учебных занятий учебным планам и нормативным документам организации учебного процесса;

- контроль выполнения графика учебного процесса;

- контроль равномерного и своевременного планирования практических и лабораторных занятий;

- осуществление контроля выполнения преподавателями запланированного объема учебной, организационно-методической, учебно-методической и воспитательной работ;

- осуществление контроля выполнения ППС установленных расписаний учебных занятий, экзаменационных сессий и экзаменов итоговой государственной аттестации;

- организация работы методистов отдела. Отслеживать своевременное выполнение работниками учебно-методического отдела распоряжений ректора;

–разработка и представление на утверждение ректору университета плана работы учебного отдела, обеспечение четкого и своевременного их выполнения;

–подготовка необходимых отчетов, в том числе, ежегодный отчет о численности профессорско-преподавательского состава.

Данная деятельность требует много времени и больших трудозатрат. Благодаря исследованию предметной области для облегчения работы УОУП, для автоматизации было выбрано одно из самых сложных и трудоёмких дел, выполняемых сотрудниками - составление отчетов по управлению отчётностью ППС.

Задача - создать приложение для автоматизации составления отчетов ППС. Для этого требуется рассмотреть все аспекты выбранной области.

Предоставляется два вида отчета для Министерства образования и науки Российской Федерации. Основной, отражающий эффективность вуза - это «Мониторинг по основным направлениям деятельности образовательной организации высшего образования». Второй отчет «Сведения об образовательной организации, осуществляющей образовательную деятельность по образовательным программам высшего образования».

Отчеты предоставляются министерству образования и науки РФ каждый год по состоянию на первое число девятого месяца.

Данные о ППС предоставляются за отчетный год в специальных формах федерального статистического наблюдения [7].

Для наглядности, на рис. 1.2 приведена схема управления отчетами в ПГУТИ.

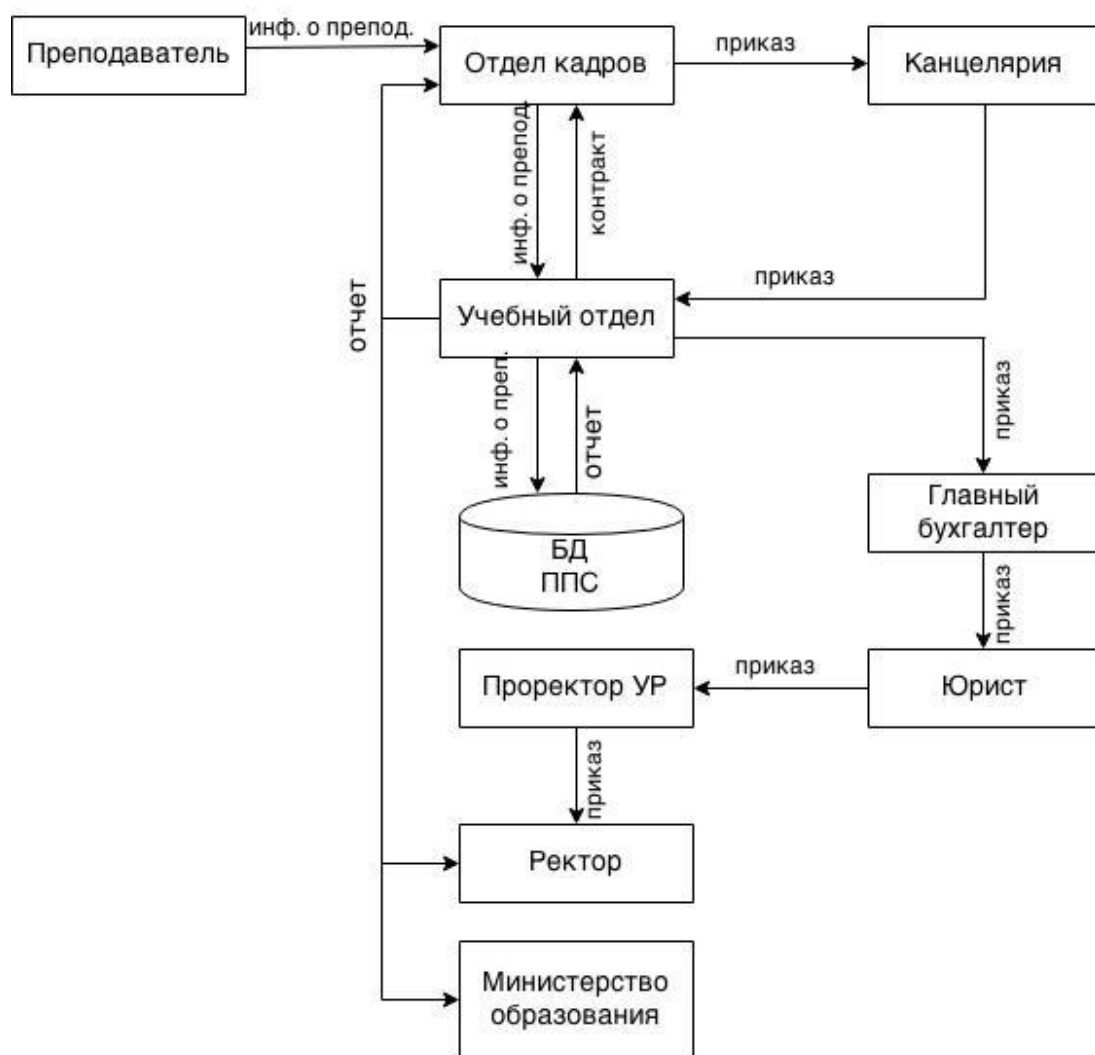


Рис. 1.2 – Схема информационных потоков при формировании отчетов о ППС

1.2 Необходимость автоматизации составления отчетов по профессорско-преподавательскому составу.

Продуктивное управление учебным процессом - одной из важнейших управленческих задач в ВУЗе, которая охватывает большое количество лиц, задействованных в этом процессе - студенты, преподаватели, учебно-вспомогательный и административно-управленческий персонал, и непосредственно влияет на условия их учебы и работы. Учет во внимание всех имеющихся возможностей и их эффективная реализация требуют немалых трудозатрат управленческого персонала. Разумеется, что проведение мероприятий по информатизации процедур управления учебным процессом способно значительно улучшить качество управления и сократить его трудоемкость. В настоящее время существует большое количество систем управления учебным процессом, представляющих собой как индивидуальные, так и коммерческие разработки. Чаще всего они направлены на решение следующих задач:

- Поступление, перевод и отчисление студентов («контингент»);
- Контроль академической успеваемости студентов («сессия»);
- Распределение учебной нагрузки между факультетами, кафедрами и преподавателями;
- Формирование учебных планов;
- Формирование штатного расписания;
- Составление и корректировка расписаний учебных занятий и экзаменов.

Создание информационной системы бесспорно требует значительных затрат. Как правило, из-за отсутствия типовых решений, ВУЗы решают эту задачу, исходя из собственных возможностей. Часто ВУЗ прибегает к разработкам собственных специалистов, либо разработками сторонней организации. Эти пути решения проблемы имеют хорошо известные

преимущества и недостатки: разработка силами своих специалистов обычно затягивается, превышая запланированные сроки, а сторонние организации привязывают типовое решение из другой сферы, недостаточно учитывая вузовскую специфику [11]. Возможно, наилучшим решением является организация временного коллектива разработчиков, состоящих из сотрудников и студентов вуза.

1.3 Анализ аналогов

Прежде чем приступить к разработке реального программного продукта по управлению отчётностью ППС, необходимо изучить существующие аналоги. В процессе изучения, были найдены следующие аналогичные программные продукты:

1. Решение «БИТ: Зарплата и кадры в вузе» – встроенный блок программы "1С:Зарплата и кадры бюджетного учреждения 8", который позволяет автоматизировать расчет заработной платы и ведение кадрового учета с учетом специфики учебного заведения. Предоставляет возможность хранить и обрабатывать информацию по сотруднику: категории персонала, профессиональные квалификационные группы, информация о повышении квалификации, о дополнительных соглашениях трудового договора (рис. 1.3).

N	Номер доп. соглашения	Дата доп. соглашения	Дата начала	Дата окончания
1	00000065	01.06.2010	01.06.2010	31.12.2010

Рис. 1.3 – БИТ: Зарплата и кадры в вузе

Стоимость данного решения от 48000 рублей [14].

2. ТАНДЕМ.Университет Модуль “Кадры”. Предназначен для ведения кадрового учета образовательной организации. Основные функции:

- I. Ведение личных карточек сотрудников: персональные данные (биография), профессиональные навыки, должности, занимаемые работником, ученая степень, ученое звание, трудовая книжка, награды и отличия, научные труды и изобретения, сведения об отпусках и другие данные.
- II. Формирование отчетов: сводки по движению, распределение по должностям, распределение по разрядам, списочная численность, списки по срокам окончания договоров, распределение сотрудников по возрастам (рис. 1.4) [21].

Кадровый ресурс: Логинов Руслан Юрьевич

Личные данные | Кадровый ресурс | Поселение | Приказы

Общая информация | Должности | ППС | Учет листов нетрудоспособности | История должностей сотрудника | Поощрения и награды | Стаж | ПК и ПП

☐ Добавить врачебную специальность
☐ Добавить ученую степень
☐ Добавить ученое звание
☐ Редактировать дополнительные сведения

Врачебные специальности							
№	Врачебная специальность ↓	Дата присуждения	Дата подтверждения	Квалификационная категория	Дата присуждения категории	Дата подтверждения категории	Имеется сертификат специалиста

Ученые степени			
№	Ученая степень ↓	Отрасль науки	Дата присуждения
1	Кандидат филологических наук	ФИЛОЛОГИЧЕСКИЕ	25.06.1999

Ученые звания	
№	Ученое звание ↓
1	Доцент

Дополнительные сведения	
Научные труды	Есть
Изобретения	Нет

Рис. 1.4 – ТАНДЕМ.Университет. Модуль Кадры.

Рассмотрев решения предлагаемые сторонними организациями, можно сказать, что на сегодняшний день, на рынке небольшое количество решений, которые способны автоматизировать кадровый учет сотрудников и сформировать необходимые отчеты по ППС. Можно добавить, что существующие программные комплексы очень дорогостоящие, что может существенно затруднить их внедрение. На основании вышесказанного, будет целесообразнее создать новое приложение для управления профессорско-

преподавательским составом ПГУТИ с возможностью автоматической генерации отчетов.

1.4 Обоснование необходимости разработки приложения

В настоящее время УОУП формирует отчёты с помощью программного пакета MS excel. Информация о профессорско-преподавательском составе в процессе работы заполняется в таблицы, а итоговое количество автоматически подсчитывается с помощью формул в определённых ячейках таблицы. Файл отчёта состоит из трех листов: преподаватели, деканы, кафедры. Распределение ППС по званиям, ученым степеням и условию привлечения к трудовой деятельности производится вручную. На данный момент существует несколько проблем в составлении отчётов по ППС в УОУП ПГУТИ: отсутствие оперативности в обновлении данных, дублирование данных, и как правило постоянная необходимость перепроверки всей базы ППС. Данная система формирования отчётов ППС хоть и справляется с возложенными на нее обязанностями, однако отнимает слишком много рабочего времени у сотрудников учебного отдела.

Очевидным является тот факт, что для дальнейшей более эффективной работы учебному отделу требуется построение более эффективной системы автоматизации отчетов, позволяющей сократить затрачиваемое время.

Исходя из вышеизложенного, требуется разработка схемы базы данных, где будет структурированно храниться информация обо всех сотрудниках ППС, работающих в вузе. Данная информация будет представлена в виде таблицы с такими полями, как: паспортные данные, личные данные (пол, возраст, дата рождения, телефон), условия привлечения к трудовой деятельности, ставка, ученая степень, стаж работы.

Данное решение (создание единой базы данных для всех работников учебного отдела) должно позволить осуществить параллельную работу многих пользователей структурного подразделения с базой данных, внесение в нее изменений, получение оперативных промежуточных и ежегодных отчетов.

1.5 Обоснование выбора языка программирования

Для реализации приложения используется язык программирования C#. C# это объектно-ориентированный язык программирования. Разработан в 1998-2001 годах. C# лишен многих недостатков, на которые указывают эксперты, анализируя процесс программирования на других языках. К данным недостаткам можно отнести сложность синтаксиса, недостаточную гибкость и даже иногда чрезмерную свободу выбора, связанную с тем, что за долгую историю язык программирования может накопить в себе достаточно много средств, решающих одну и ту же задачу.

Данный язык создан для разработки на платформе Microsoft.NET Framework, которая “оборудована” сборщиком мусора, что позволяет разработчику не задумываться о проблемах, связанных с выделением и освобождением памяти для переменных программы. В ходе выполнения очистки сборщик мусора отыскивает в управляемой куче объекты, которые более не используются, и освобождает выделенную для них память.

В программу ВУЗа входит курс изучения данного языка программирования, что существенно помогло с выбором языка. Синтаксис C# понятный, компактный и наглядный, поэтому программный код, написанный на C# будет легко читаем и переводим на другие языки программирования [1].

1.6 Обоснование выбора СУБД

Система управления базами данных (СУБД) - это программный механизм, предназначенный для записи, поиска, сортировки, обработки (анализа) и печати информации, содержащейся в базе данных [17].

SQL Server 2008 – СУБД разработанная корпорацией Microsoft, а так, как в качестве языка разработки был выбран С#, то данная СУБД должна легко интегрироваться в приложение и не должно быть конфликтов совместимости. Она позволяет хранить в базе данных информацию, полученную из структурированных, полуструктурированных и неструктурированных источников, таких как музыка и изображения. В SQL Server 2008 имеется большой набор интегрированных служб, расширяющих возможности использования хранимых данных: присутствует возможность составлять запросы, выполнять поиск, проводить синхронизацию, делать отчеты, анализировать данные, создавать резервные копии баз данных. Все данные хранятся на серверах, входящих в состав центра обработки данных. Доступ к этим данным осуществляется с настольных компьютеров и мобильных устройств. Таким образом, осуществляется контроль над данными вне зависимости от их места хранения.

Система SQL Server 2008 позволяет обращаться к данным из любого приложения с помощью необходимого драйвера. Сотрудники, отвечающие за сбор и анализ информации, могут работать с данными, не покидая привычных приложений, которыми они пользуются каждый день, например MicrosoftOffice. MS SQL Server 2008 позволяет создать надежную, производительную, интеллектуальную платформу, отвечающую всем требованиям по работе с данными [8].

Динамическое шифрование всей базы данных, файлов данных и файлов журналов не требует никаких изменений в имеющихся приложениях. Присутствует ряд преимуществ: поиск по зашифрованным данным как по

диапазонам, так и нечетким поиском; поиск защищенных данных у неавторизованных пользователей; использование шифрования без каких-либо изменений в существующих приложениях.

Регулятор ресурсов обеспечивает стабильное и предсказуемое время отклика на запросы конечных пользователей. С его помощью можно определить допустимые границы использования ресурсов и устанавливать приоритеты для разных рабочих нагрузок. Это обеспечивает стабильную производительность одновременно выполняемых задач.

Фиксация планов выполнения запросов позволяет достичь большей стабильности и предсказуемости их выполнения. Стабильные планы запросов не потеряются при замене оборудования, обновлениях сервера и промышленном развертывании.

С помощью технологии сжатия данные хранятся более эффективно и занимают меньше места в хранилище. Также это обеспечивает заметный прирост производительности на задачах, связанных с большим количеством операций чтения/записи, как, например, при работе с хранилищами данных.

В SQL Server 2008 физическая установка на оборудование отделена от конфигурирования, что позволяет самим предприятиям и их компаниям-партнерам создавать рекомендованные к установке конфигурации.

Запросы к базе можно осуществлять с помощью управляемого языка программирования, например, C# или VB.NET, а не командами SQL. Строго типизированные, ориентированные на наборы данных запросы, написанные на языках .NET, могут легко выполняться к ADO.NET (LINQ to SQL), ADO.NET DataSets (LINQ to DataSets), платформе ADO.NET EntityFramework (LINQ to Entities) и к поставщику EntityDataSourceMapping. Новый поставщик LINQ to SQL позволяет через LINQ напрямую обращаться к таблицам и столбцам SQL Server 2008.

Значения NULL больше не занимают физического пространства диска, что позволяет очень эффективно управлять пустыми областями данных и экономно расходовать память. Благодаря разреженным столбцам те объектные модели, которые содержат многочисленные значения NULL, будут храниться в базе данных SQL Server 2005, не требуя значительного места в хранилище.

Благодаря языку запросов LanguageIntegratedQuery (LINQ), реализованному в MSVisualStudio 2008, разработчики могут воспользоваться собственным синтаксисом в сочетании с традиционными языками программирования, такими как C# и VisualBasic (VB), для ссылки на объекты базы данных как на собственные объекты языка и создания запросов к этим объектам. Запросы MSVisualStudio 2008 и LINQ преобразуют код процедур в вызовы базы данных на основе T-SQL, направляемые в SQL Server. Как новая технология .NET, LINQ является частью библиотек Microsoft .NET Framework начиная с версии 3.5.

В LINQ объектная модель представляет источник данных. Затем LINQ ссылается на этот источник данных как на объект DataContext - например, System.Data.Linq.DataContext. Объект DataContext инкапсулирует строку соединения ADO.NET для базы данных, затем используется с набором определений объекта - например, System.Data.Linq.Mapping.TableAttribute - для таблиц, хранимых процедур и функций в базе данных.

СУБД SQL-Server 2008 содержит большой набор интегрированных служб, расширяющих возможности использования данных – использование запросов, выполнение поиска, создание отчетов, скриптов. SQL Server 2008 позволяет обращаться к данным из любого приложения, разработанного с применением технологий Microsoft .NET и VisualStudio, которые в данный момент имеют самое широкое распространение в сфере создания баз данных и приложений к ним, обладают широким спектром различных языков программирования для выполнения поставленных задач и всем необходимым

инструментарием. Удобный интерфейс пользователя при создании, как базы данных, так и пользовательского приложения способствует более эффективному выполнению поставленной задачи. Также, SQL Server 2008 имеет достаточно простой процесс связывания данных и приложения.

1.7 Постановка задачи

2. Разработать приложение для обработки данных ППС и вывода отчётов.
3. Разработать схему базы данных, содержащую основные данные о сотрудниках ППС.
4. Разработать формы отчетов для штатных сотрудников, штатных совместителей, сторонних совместителей.
5. Разработать таблицы для отчетов:
 - А) распределение численности персонала по уровню образования с разделением на штатных сотрудников, внутренних совместителей, сторонних совместителей.
 - Б) Сведения об учёных степенях ППС и научных работников
 - В) Распределение персонала по полу и возрасту
 - Г) Распределение персонала по стажу.
6. Разработать таблицы для промежуточных данных:
 - штатные сотрудники;
 - внутренние совместители;
 - внешние совместители.

1.8 Определение функциональных требований

В приложении должны быть реализованы следующие функции:

– в базе данных должны храниться основные данные о сотруднике ППС.

- ФИО;
- должность;
- пол;
- дата рождения;
- паспортные данные;
- адрес прописки;
- телефон;
- образовательное учреждение;
- ученая степень;
- дата получения степени;
- ученое звание;
- дата получения звания;
- условия привлечения к трудовой деятельности;
- повышение квалификации;
- ставка;
- общий стаж.

– все данные формы должны заполняться по данным первичной учетной документации, имеющейся в УОУП, бухгалтерии, отделе кадров и других подразделениях образовательной организации высшего образования;

– при заполнении формы должна быть обеспечена полнота заполнения и достоверность внесённых статистических данных;

Необходимые условия для отчета по распределению численности персонала по уровню образования с разделением на штатных сотрудников, внутренних совместителей, сторонних совместителей:

– в форме должны приводятся данные о распределении основного персонала вуза по уровню образования и отображаться фактическая численность сотрудников;

– в графе «всего» приводятся сведения о работниках, основным местом которых является отчитываемая образовательная организация.

Штатные сотрудники подразделяются:

1. Профессора.

1.1 Численность ППС с ученой степенью доктора наук.

1.2 Численность ППС с ученой степенью кандидата наук и ученым званием профессора.

1.3 Численность ППС на должности профессора.

2. Доценты.

2.1 Численность ППС с ученой степенью кандидата наук и/или званием доцента.

2.2 Численность ППС доцентов (без ученой степени и/или звания).

3. Численность ППС старших преподавателей.

4. Численность ППС преподавателей и ассистентов.

Сторонние совместители подразделяются:

1. Профессора.

1.1 Численность ППС с ученой степенью доктора наук.

1.2 Численность ППС с ученой степенью кандидата наук и ученым званием профессора.

1.3 Численность ППС на должности профессора.

2. Доценты.

2.1 Численность ППС с ученой степенью кандидата наук и/или званием доцента.

2.2 Численность ППС доцентов (без ученой степени и/или звания).

3. Численность ППС старших преподавателей.

4. Численность ППС преподавателей и ассистентов.

Внешние совместители подразделяются:

1. Профессора.

1.1 Численность ППС с ученой степенью доктора наук.

1.2 Численность ППС с ученой степенью кандидата наук и ученым званием профессора.

1.3 Численность ППС на должности профессора.

2. Доценты.

2.1 Численность ППС с ученой степенью кандидата наук и/или званием доцента.

2.2 Численность ППС доцентов (без ученой степени и/или звания).

3. Численность ППС старших преподавателей.

4. Численность ППС преподавателей и ассистентов.

2 Разработка приложения «Управление ППС ПГУТИ»

2.1 Разработка функциональной схемы

Для представления функций приложения и используемых ресурсов на рис. 2.1 представлена функциональная схема.

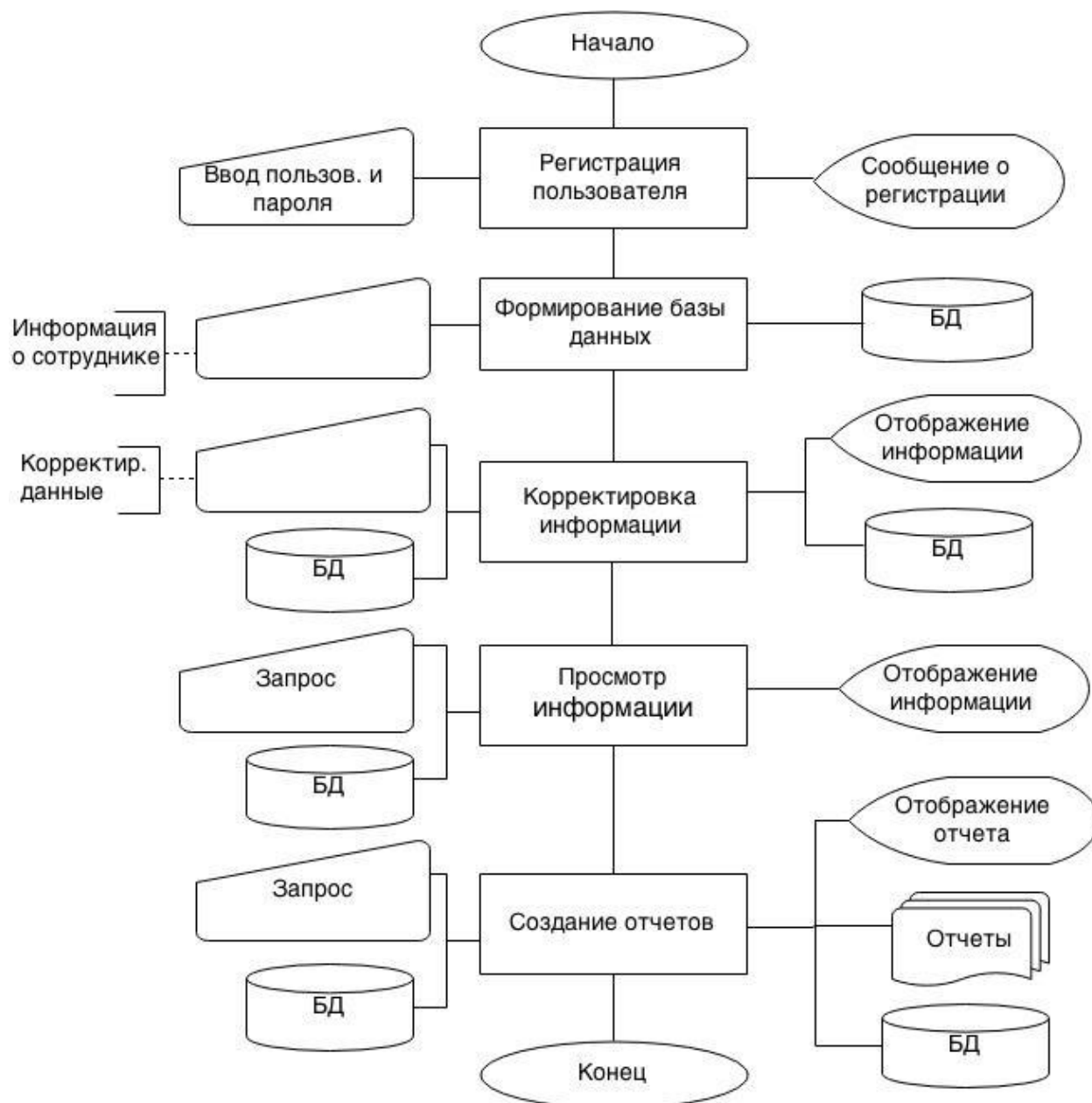


Рис. 2.1 – Функциональная схема

Приложение выполняет следующие функции: авторизации пользователя, формирование базы данных, корректировка информации, просмотр информации, создание отчетов.

2.2 Разработка интерфейса приложения

Интерфейс - система средств и правил, регламентирующая и обеспечивающая взаимодействие нескольких процессов или объектов.

Пользовательский интерфейс - это совокупность информационной модели предметной области, средств и способов взаимодействия пользователя с данной моделью, а также компонентов, обеспечивающих формирование информационной модели в процессе работы системы. Создание качественного и понятного интерфейса требует значительно больше усилий, чем просто соблюдение некоторых инструкций. Оно предполагает реализацию принципа «usability» и соответствующую методологию разработки всего программного продукта. Usability - степени удобства интерфейса программного средства для применения пользователями при достижении определённых целей. Однако usability означает не только улучшенное визуальное руководство или улучшенную иерархию, это также означает и больший контакт с пользователем посредством профессионально сделанного серьёзного дизайна, преподнесения верной информации тогда, когда она нужна.

Для того, чтобы обеспечить комфортную работу с программной системой, разработана схема пользовательского интерфейса (рис. 2.2).

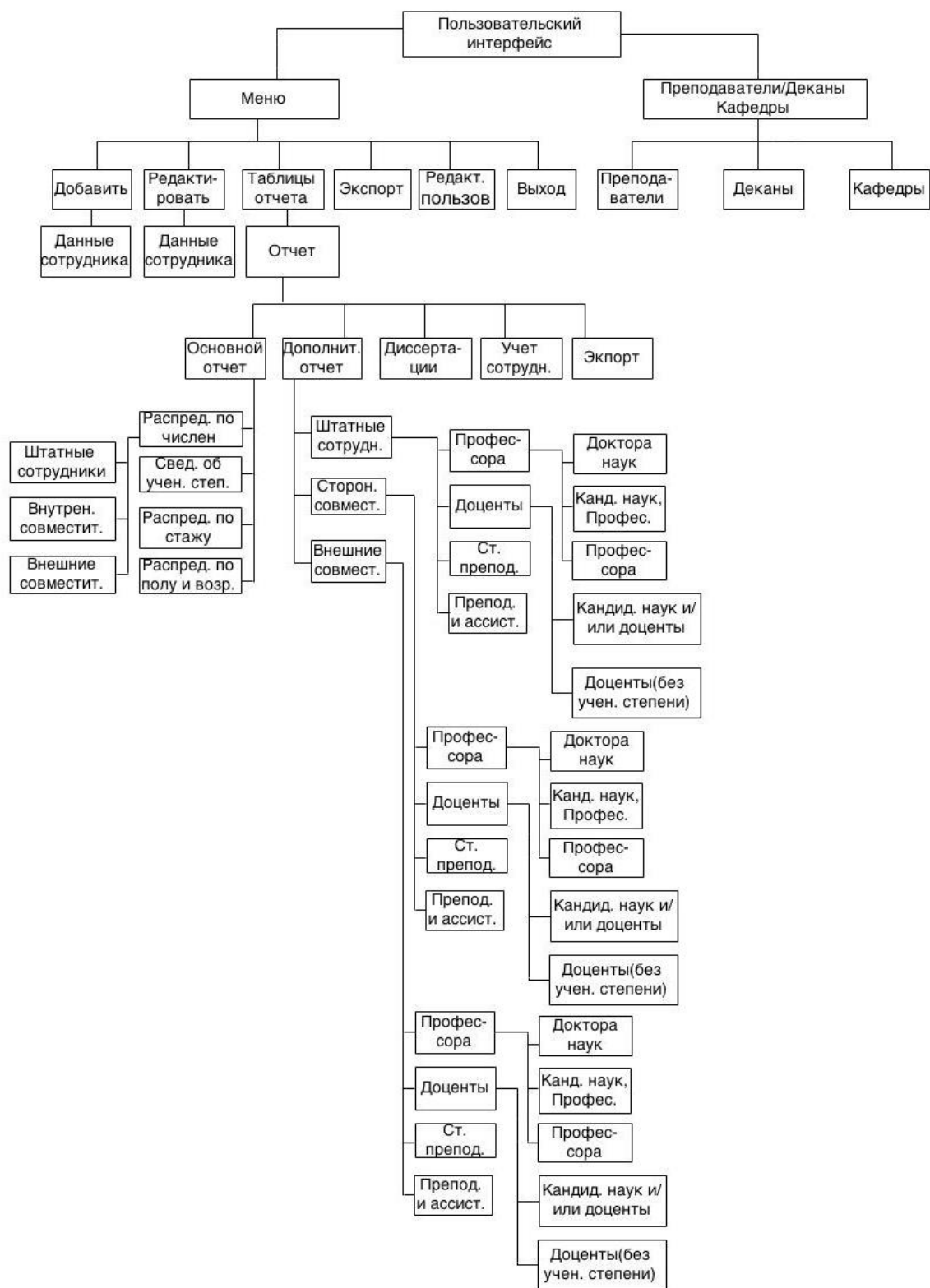


Рис. 2.2 – Схема интерфейса приложения

Один из важнейших показателей качества ПО – удобство использования и понятность для пользователя. Описать качество ПО можно с помощью следующих характеристик:

- Легкость использования;
- Понятность для пользователя;
- Эффективность их решения;
- Сложность решаемых задач проблемной области;
- Простой процесс обучения работе с ПО.

Создание удобного для пользователя интерфейса вся необходимая для работы информация должна быть представлена в максимально структурированном виде. Наиболее важная информация должна находиться всегда перед глазами, но и всю сразу предоставлять не стоит – пользователь должен видеть только ту информацию, которая необходима на данный момент.

Для выполнения такой задачи рассмотрим следующие принципы создания пользовательского интерфейса:

- 1) Принцип простоты. Однотипные операции должны осуществляться с максимальной простотой и минимальными затратами времени. Например, заполнение справочников – несколько пунктов, кнопка сохранить, отменить.
- 2) Принцип обратной связи. Пользователь должен понимать, что он допустил ошибку или все действия были верными и данные были сохранены.
- 3) Принцип толерантности. Интерфейс ПО должен быть продуман и не дать пользователю совершить ошибку при редактировании данных.
- 4) Принцип естественности. Естественный интерфейс такой, который не обязывает пользователя менять привычные для него способы решения задачи. Это, в частности, означает, что сообщения и

результаты, выдаваемые приложением, не должны требовать дополнительных пояснений. Целесообразно также соблюдать терминологию, используемую в данной предметной области.

Качество интерфейса пользователя сложно оценить количественными характеристиками, однако есть возможность получить его объективную оценку на основе приведенных ниже частных показателей:

- время, необходимое установленному пользователю для достижения заданного уровня знаний и навыков по работе с приложением. Например, непрофессиональный пользователь должен освоить команды работы с файлами не более чем за 4 часа;
- сохранение полученных рабочих навыков по истечении периода времени. Например, после недельного перерыва пользователь сможет выполнить определенную последовательность операций за заданное время;
- скорость решения задачи с помощью данного приложения, при этом оценивается не быстроедействие системы, а время, необходимое на решение поставленной задачи. Например, так: пользователь должен обработать за час не менее 20 документов с ошибкой не более 1%;
- субъективная удовлетворенность пользователя при работе с данным ПО и его интерфейсом, которая количественно может быть выражена в процентах или оценкой по n-бальной шкале.

Форма – основная единица пользовательского интерфейса. Все информационные единицы желательно располагать в зависимости от характера решаемой задачи – последовательно, по мере необходимости их использования – если речь идет о заполнении заявок или справочников. Или в порядке их важности для решаемой задачи. Взаимосвязанные единицы должны отображаться на одной форме [15].

В разработанном приложении имеются основные типы диалоговых окон, содержащих разнообразные элементы управления:

- тексты сообщения;
- поля для ввода информации пользователя;
- списки возможных альтернатив для выбора;
- кнопки.

В результате, был получен интерфейс следующего вида (рис. 2.3).

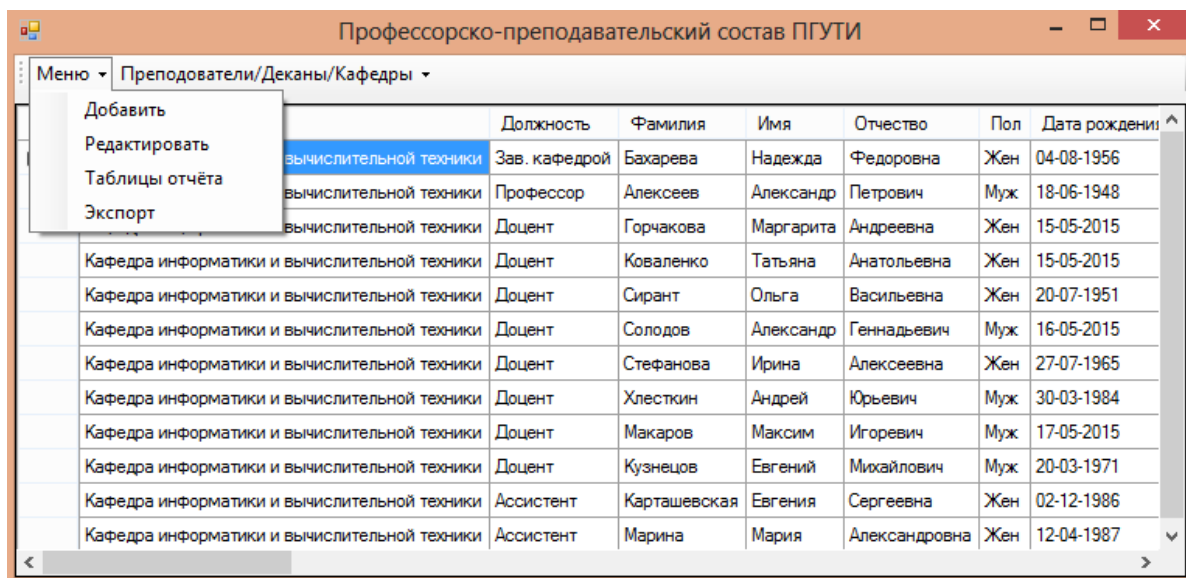


Рис. 2.3 – Интерфейс приложения

2.3 ER – диаграмма

Современные подходы к проектированию реляционных баз данных основываются на использовании ER-модели. С её помощью можно выделить ключевые сущности и обозначить связи, которые могут устанавливаться между этими сущностями [4]. Схема базы данных представлена на рис. 2.4.

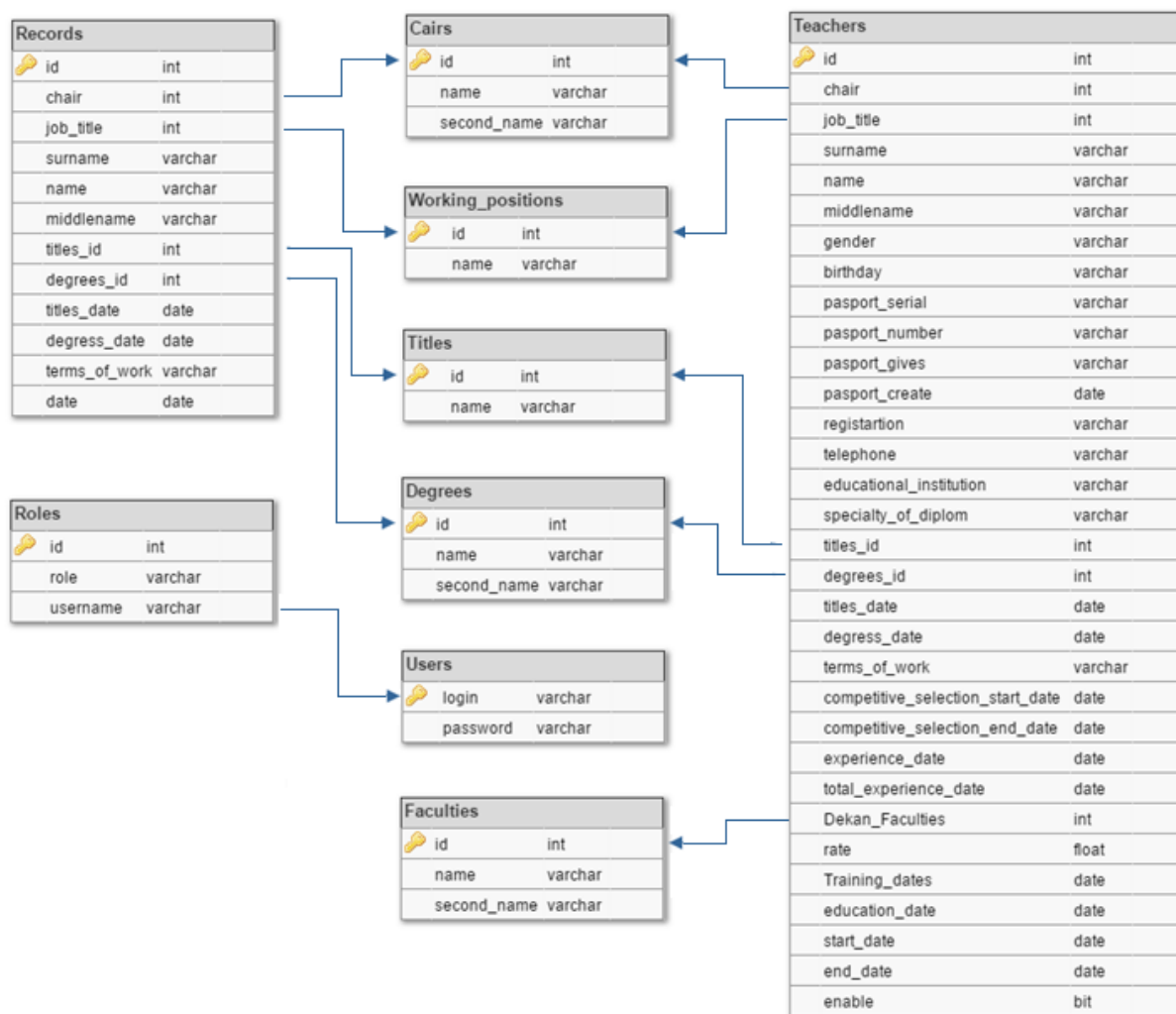


Рис. 2.4 – ER-диаграмм

2.4 Укрупненная схема алгоритма приложения

На рисунке 2.5 представлена укрупненная схема алгоритма.

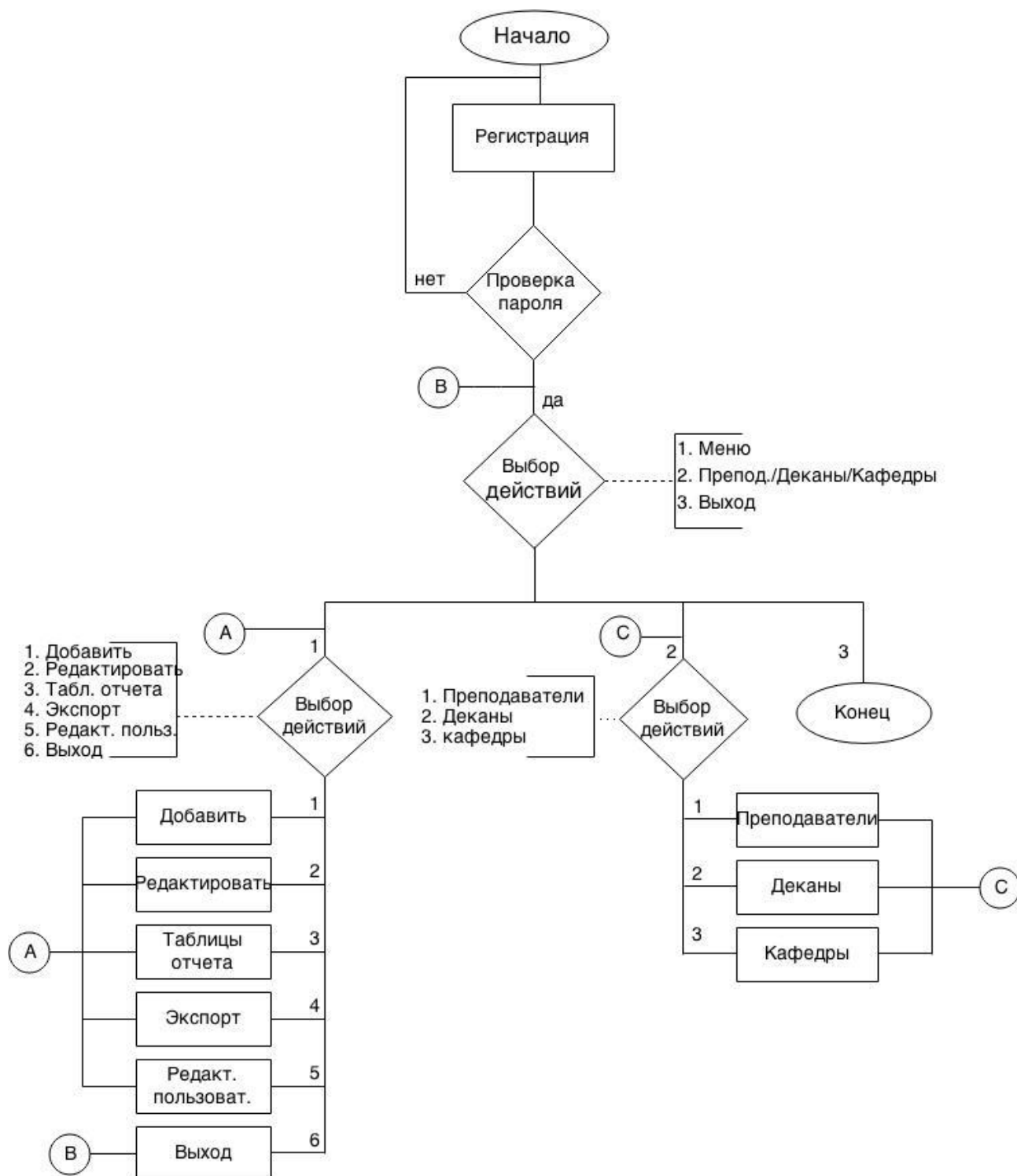


Рис. 2.5 – Укрупненная схема алгоритма

2.4 Защита информации

Защита информации – комплекс мероприятий, направленных на обеспечение важнейших аспектов информационной безопасности, а именно целостности, доступности и конфиденциальности информации и ресурсов, используемых для ввода, хранения, обработки и передачи данных). В используемой версии MS SQL Server 2008 существует 2 режима проверки при определении прав пользователя: интегрированный, называемый Windows NT Authentication Mode (Windows NT Authentication), и смешанный, Mixed Mode (Windows NT Authentication and SQL Server Authentication). Алгоритм проверки аутентификации пользователя представлен на рис 2.6.

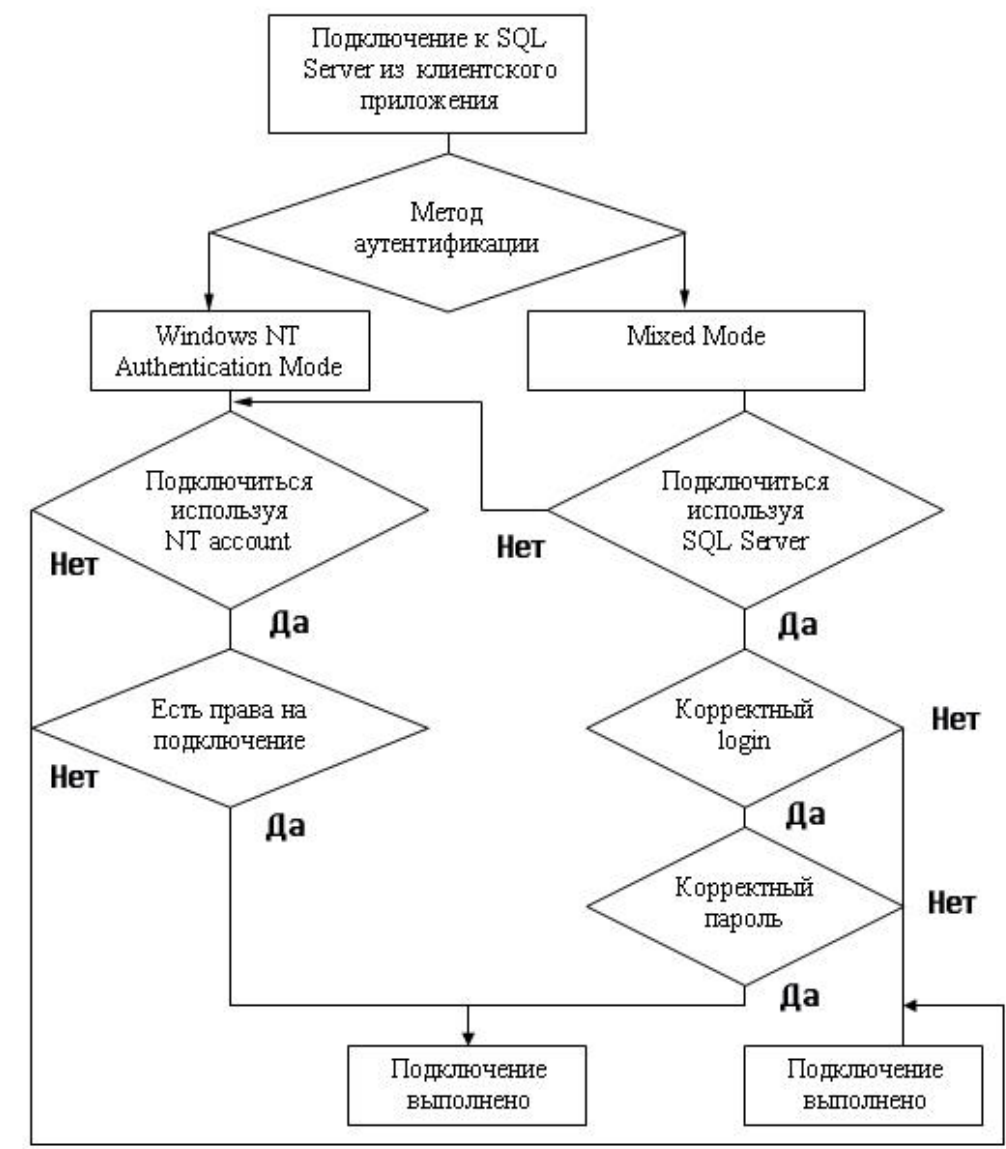


Рис. 2.6 – Алгоритм аутентификации в MS SQL Server 2008

В рамках данной дипломной работы необходимо реализовать защиту базы данных, для хранения и чтения конфиденциальной информации о сотрудниках вуза. Наиболее гибкий и распространенный способ защиты базы данных называется защитой на уровне пользователей. Данный способ требует пользователя идентифицировать себя путём введения логина и пароля. После успешного подключения сервер определяет, какие привилегии имеет данный пользователь при работе с БД и приложением.

Следует отметить главные преимущества защиты на уровне пользователей:

- возможности пользователей ограничены в возможность использования функциями данной системы, что позволяет скрыть часть не нужной информации и пользоваться только необходимыми данному пользователю функциями.
- защищаются конфиденциальные сведения в базе данных.

Пользователи информационной системы. Уровни доступа пользователей

В рамках данной дипломной работы реализованы 2 набора прав (ролей):

- администратор базы данных, который будет иметь полный доступ к базе, в том числе и редактирование самих пользователей;
- пользователь, выполняющий основные действия (добавления, редактирование информации о ППС) и имеет доступ к отчетам.

Разграничение прав доступа к функциям системы взаимодействующими с базой данных представлено в таблице 2.2.

Таблица 2.2

Разграничение прав доступа к функциям базы данных

Функциональные возможности	Администратор базы	Пользователь
Данные сотрудников	RWD	RW
Просмотр/формирование отчетов	+	+
Экспортировать данные	+	+
Редактировать пользователей	RID	-

В таблице использованы следующие обозначения:

- R - read - доступ для чтения;
- W - write - доступ на ввод нового;
- D - delete - доступ на удаление.

3 Технологический раздел

3.1 Руководство пользователя

Краткое описание возможностей

Данная информационная система предназначена для быстрого формирования нужного отчета и вывода его на печать, что способствует оптимизации управления отчетностью вуза по профессорско-преподавательскому составу.

Приложение имеет следующие основные функции:

- добавление и редактирование сотрудников в БД;
- формирование необходимых отчетов ППС;
- экспорт результатов в MS Excel документ.

При работе с отчетностью используется инструмент пользователя Microsoft Office Excel, который предоставляют следующие возможности:

- оформить табличную часть отчетов в соответствии с пожеланиями пользователя;
- вывести данные на печать.

Уровень подготовки пользователя

При работе с данным приложением пользователь должен иметь опыт работы с ОС MS Windows и иметь навыки работы с ПО Microsoft Office Excel.

Назначение и условия применения приложения

Программное обеспечение применяется для автоматизации подготовки ежегодных отчетов по численности и деятельности профессорско-преподавательского состава вуза. Подключение к сети интернет не требуется. Функционал приложения доступен по логину и паролю.

Порядок загрузки данных и приложения

Перед началом работы необходимо выполнить следующие действия:

1. Двойным щелчком мыши открыть приложение PGUTI.exe
2. Появится окно авторизации с запросом на логин и пароль (рис. 3.1).

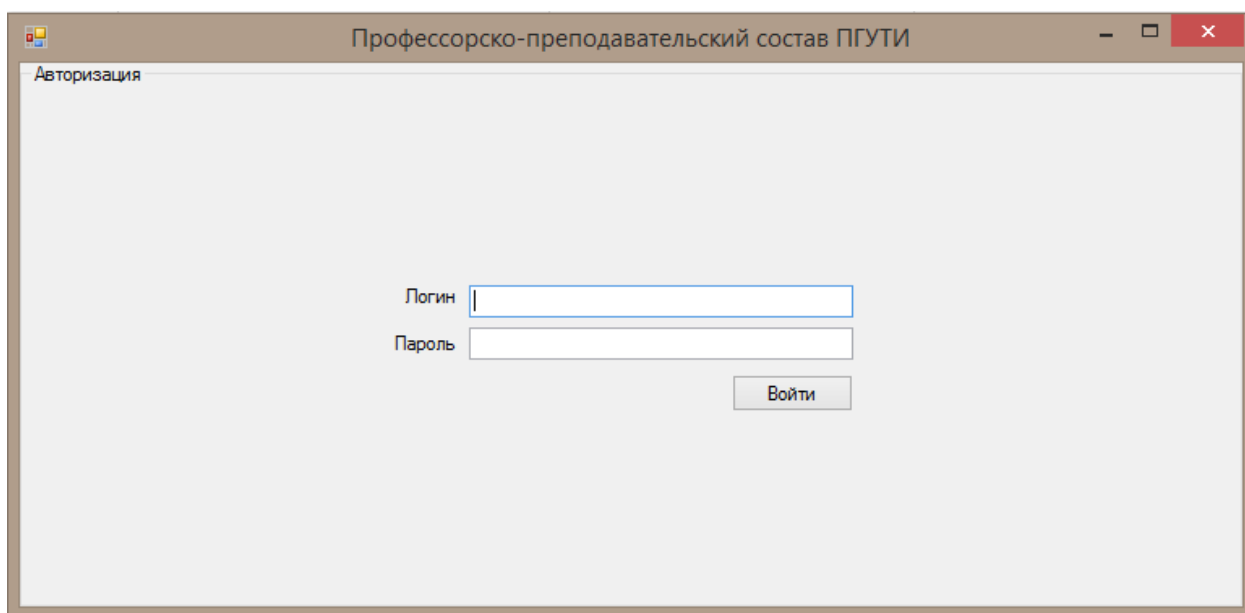


Рис. 3.1 – Окно авторизации

Стандартный для первичного входа в приложение, под ролью администратора, является логин admin, пароль admin. Внимание! Рекомендуется после первого же пользования приложением сменить логин и пароль, во избежание утечки конфиденциальных данных. Для этого необходимо произвести двойной клик по нужной строке (рис. 3.2.1). Появится окно (рис. 3.2.2).

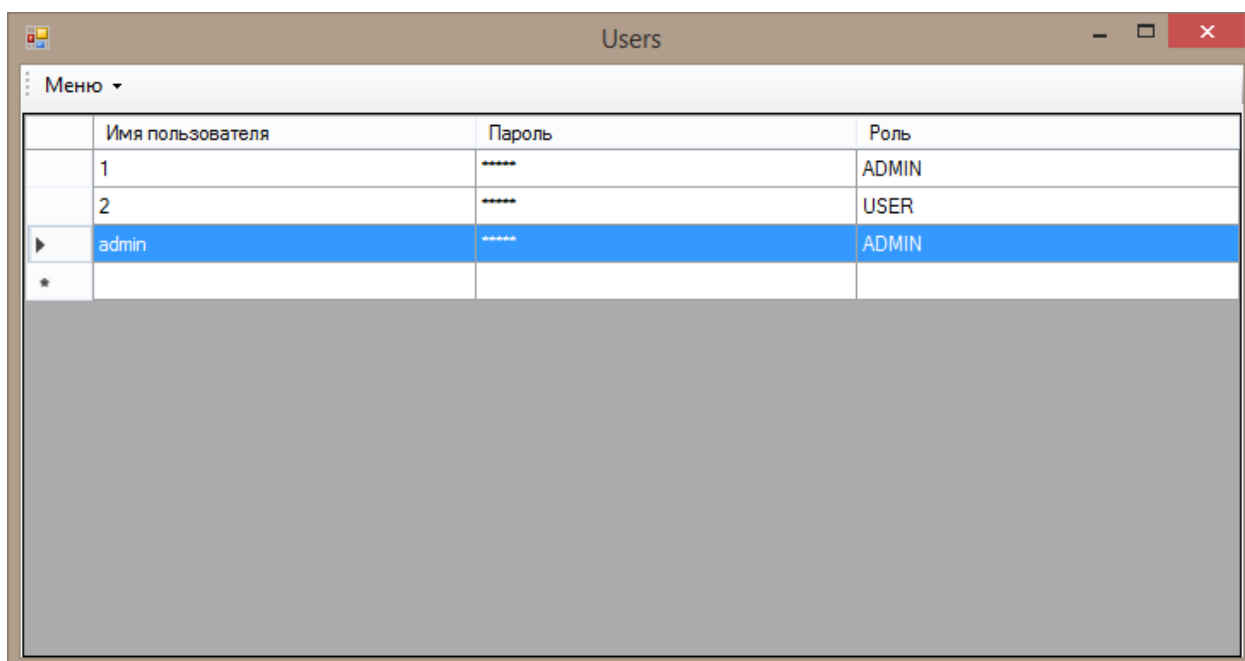


Рис. 3.2.1 – Редактирование пользователей

Users

Редактировать

Логин

Пароль

Роль

Рис. 3.2.2 – Редактирование пользователей

В данном окне необходимо ввести логин и пароль известный только администратору. Если с приложением работает несколько человек, администратору необходимо внести их в базу данных Меню → Добавить. У администратора, при необходимости, существует возможность добавлять, редактировать и удалять пользователей (рис. 3.3).

Users

Меню ▾

- Добавить
- Редактировать
- Удалить

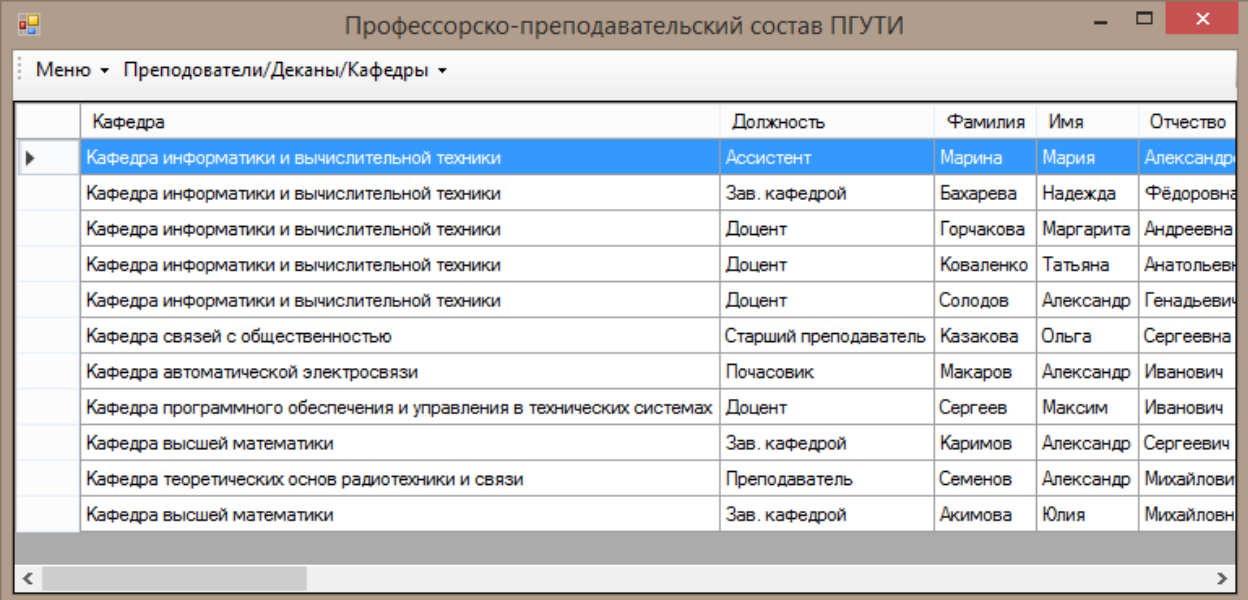
Логин	Пароль	Роль
	*****	ADMIN
	*****	USER
admin	*****	ADMIN

Рис. 3.3 – Добавление новых пользователей

Описание интерфейса

Интерфейс пользователя компьютерного приложения включает:

- Главная форма для отображения основной информации о сотрудниках (рис. 3.4).



Кафедра	Должность	Фамилия	Имя	Отчество
Кафедра информатики и вычислительной техники	Ассистент	Марина	Мария	Александровна
Кафедра информатики и вычислительной техники	Зав. кафедрой	Бахарева	Надежда	Фёдоровна
Кафедра информатики и вычислительной техники	Доцент	Горчакова	Мargarита	Андреевна
Кафедра информатики и вычислительной техники	Доцент	Коваленко	Татьяна	Анатолевна
Кафедра информатики и вычислительной техники	Доцент	Солодов	Александр	Генадьевич
Кафедра связей с общественностью	Старший преподаватель	Казакова	Ольга	Сергеевна
Кафедра автоматической электросвязи	Почасовик	Макаров	Александр	Иванович
Кафедра программного обеспечения и управления в технических системах	Доцент	Сергеев	Максим	Иванович
Кафедра высшей математики	Зав. кафедрой	Каримов	Александр	Сергеевич
Кафедра теоретических основ радиотехники и связи	Преподаватель	Семенов	Александр	Михайлович
Кафедра высшей математики	Зав. кафедрой	Акимова	Юлия	Михайловна

Рис. 3.4 – Главная форма приложения

- Обратная связь между пользователем и приложением (рис. 3.5).

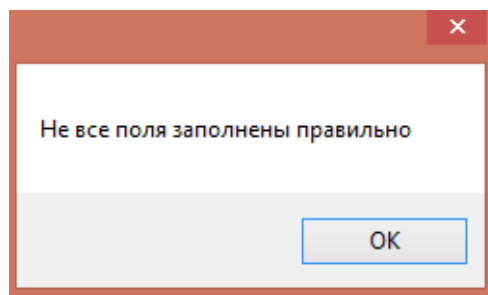


Рис. 3.5 – Обратная связь между пользователем и приложением

Описание операций

Для формирования отчетов необходимо

1. Добавить в базу данных новую запись. Меню → Добавить (рис. 3.6).

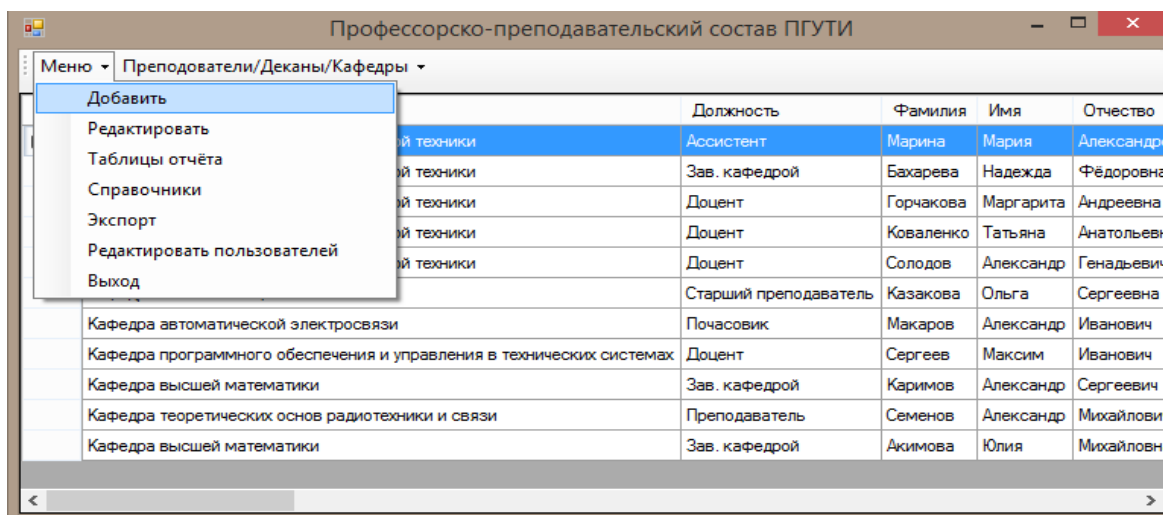


Рис. 3.6 – Добавление нового сотрудника

2. Заполнить данные о сотруднике (рис. 4.6)

Рис. 3.7 – Заполнение данных о сотруднике

В этом окне необходимо ввести всю необходимую информацию о сотруднике. В противном случае приложение выдаст сообщение об ошибке.

3. Отредактировать данные уже имеющегося в базе сотрудника. Двойной клик по строке или выделить строку с данными сотрудника и нажать на кнопку Меню → Редактировать.
4. Выбор из общей базы только преподавателей либо только деканов, либо только кафедры. Открыть в меню Преподаватели/деканы/кафедры и выбрать необходимый пункт (рис. 3.8).

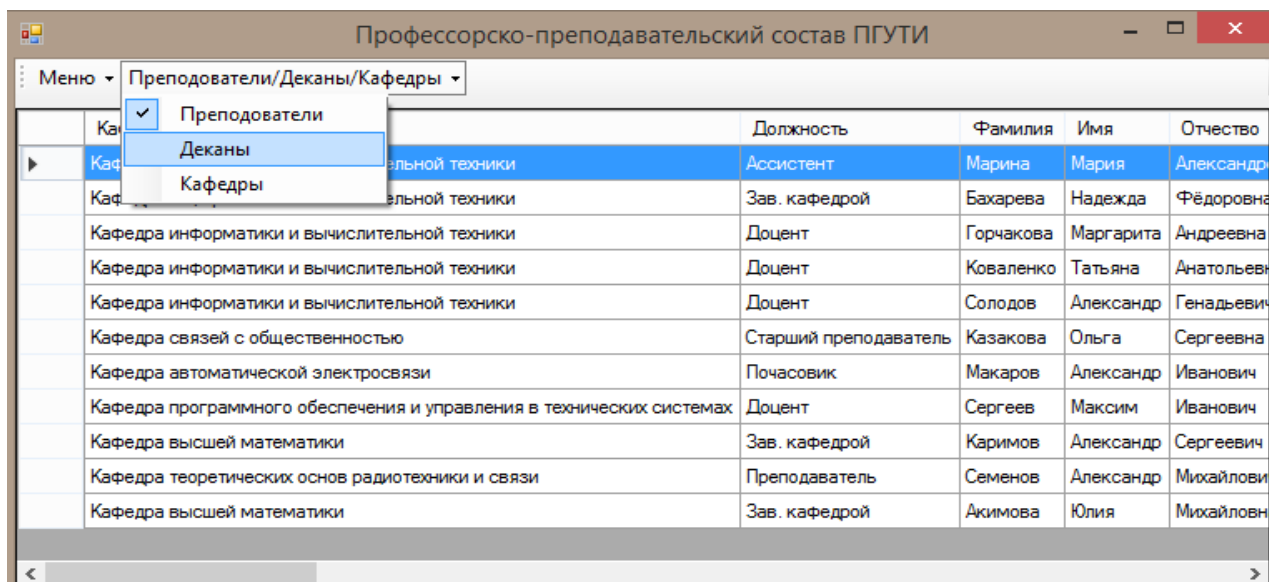


Рис. 3.8 – Выбор необходимого пункта

5. Сформировать отчет. Меню → Таблицы отчетов (рис. 3.9).

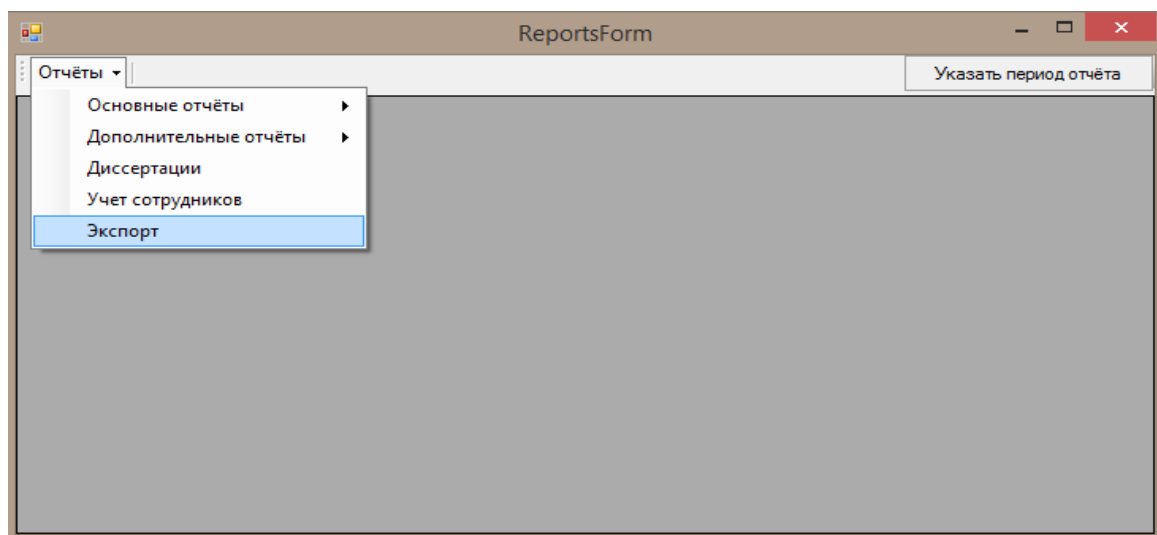


Рис. 3.9 – Окно с выбором отчета

При необходимости можно сформировать отчет за определенный период, для этого необходимо нажать на кнопку - Указать период отчета и выбрать период (рис. 3.10).

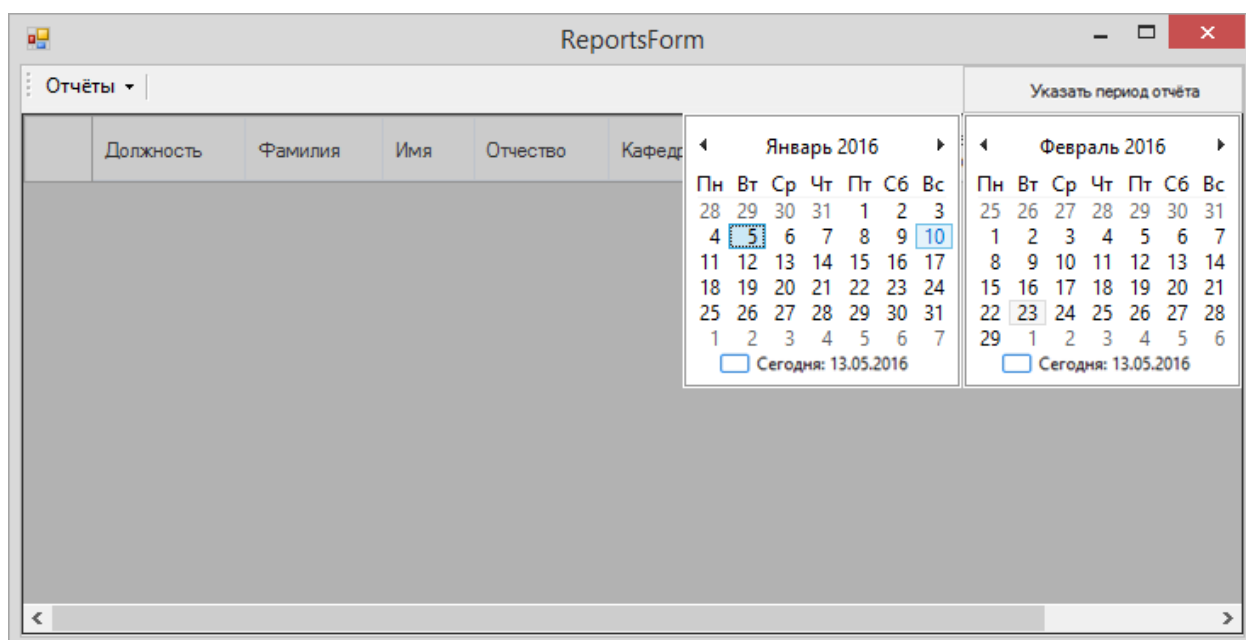


Рис. 3.10 – Выбор периода отчета

Выбираем из списка нужный отчёт (рис. 3.11).

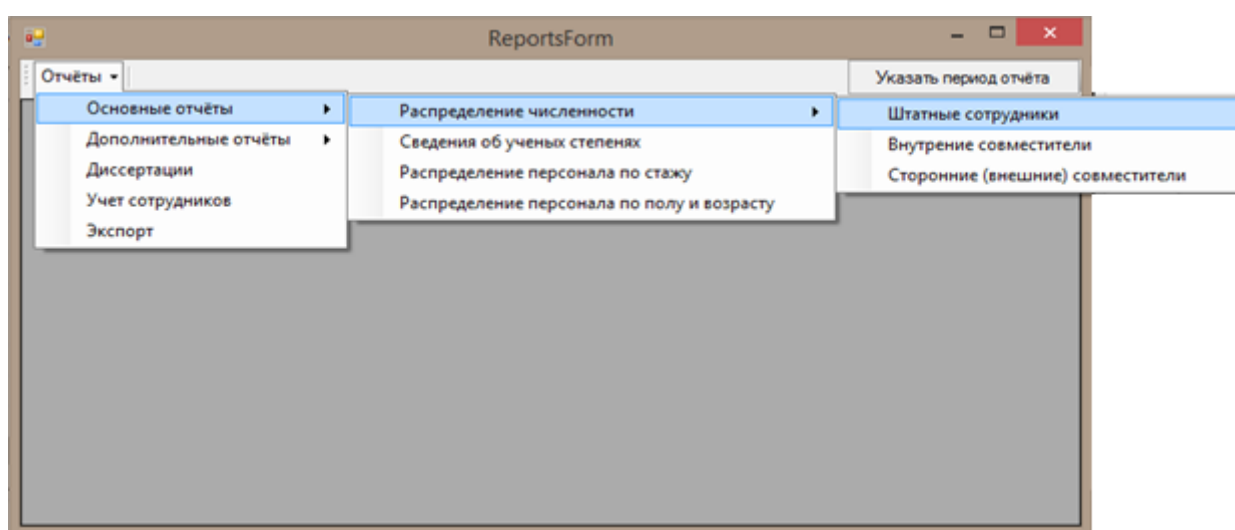


Рис. 3.11 – Выбор нужного отчета

Кликаем по нему и получаем результат (рис. 3.12).

ReportsForm							
Отчёты ▾		Указать период отчёта					
	Всего	имеют высшее образование	ученую степень		ученое звание		женщины
			доктора наук	кандидата наук	профессора	доцента	
1	3	4	5	6	7	8	9
профессорско-преподавательский состав – всего	192	192	30	113	22	71	89
в том числе:							
деканы факультетов	5	5	1	4	1	4	-
заведующие кафедрами	19	19	15	4	13	5	3
профессора	16	16	13	3	8	3	4
доценты	109	109	1	101	-	59	57
старшие преподаватели	31	31	-	1	-	-	19

Рис. 3.12 – Результат отчета

6. Вывести нужный отчет на печать. Таблицы → Экспорт. Отчет открывается в программе MS Excel (рис. 3.13).

Рис. 3.13 – Отчет в программе MS Excel

Нажимаем на иконку печати и печатаем нужный отчет.

7. Заключительные действия. Меню → Выход.

Аварийные ситуации

В случае возникновения ошибки при работе с приложением, не описанных в таблице 3.1, необходимо обращаться к системному администратору.

Таблица 3.1

Разграничение прав доступа к объектам базы данных

Ошибка	Описание ошибки	Рекомендуемые действия пользователя
Неверный логин/пароль	При авторизации в приложении неверно введены входные данные, либо такие данные не зарегистрированы.	Повторить ввод имени пользователя и пароля, однако после второй неудачной попытки авторизации рекомендуется обратиться к администратору.
Не все поля заполнены правильно	При заполнении формы сотрудника были заполнены не все нужные поля.	Внимательно проверить форму ввода данных о сотруднике и заполнить форму недостающими данными. Если проблема не пропала обратиться к администратору.

С целью защиты конфиденциальных данных, на предоставленных скриншотах приведены фиктивные данные.

3.2 Методика проведения тестирования и результаты

Тестирование программы - это этап проверки поведения программы при различных наборах входных данных, в том числе и на заведомо неверных.

Основные принципы организации тестирования:

- неотъемлемой частью каждого теста является описание ожидаемых результатов работы программы, чтобы выявить наличие или отсутствие ошибки;
- следует по возможности избегать тестирования программы ее автором, т.к. присутствует фактор противоречия человеческой психологии, сторонний от разработки программы человек сможет протестировать намного лучше (однако отладка программы эффективнее всего выполняется именно автором программы);
- по тем же соображениям организация - разработчик программного обеспечения не должна проводить все тесты собственноручно (существуют организации, специализирующиеся на тестировании программных комплексов);
- необходим подбор тестов с заранее не верными входными данными;
- следует хранить уже готовые тесты (для повторного тестирования программы после ее модификации или установки у заказчика);
- тестирование проводится даже если есть предположение, что в программе не будут обнаружены ошибки (следует выделить достаточные временные и материальные ресурсы для тестирования);
- следует учитывать так называемый "принцип скопления ошибок": вероятность наличия не обнаруженных ошибок в некоторой части программы прямо пропорциональна числу ошибок, уже обнаруженных в этой части [16].

При тестировании программы были выполнены следующие принципы:

- тест проводился не только для правильных (предусмотренных) входных данных, но и для неправильных (непредусмотренных);
- необходимо досконально изучить результат каждого теста, чтобы не пропустить малозаметную ошибку в программе;
- приложение тестируется не только автором, но и сотрудниками учебного отдела.

Для данной курсовой работы результатом тестирования является сформированные отчеты по управлению ППС вуза. Для защиты конфиденциальных данных в отчётах используются фиктивные данные. Результаты тестирования основных отчетов представлены на рис 3.14 – 3.18.

	Всего	из гр.3 имеют высшее образо- вание	из гр. 4 имеют:				Из гр. 3		из гр.3 работают на												
			ученую степень		ученое звание		жен- щины	освоили допол- нительные про- граммы повыше- ния квалифика- ции и (или) про- фессиональной переподготовки за предыдущий учебный год	0,1 ставки	0,2 ставки	0,25 ставки	0,3 ставки	0,4 ставки	0,5 ставки	0,6 ставки	0,7 ставки	0,75 ставки	0,8 ставки	0,9 ставки	1 ставку	
			доктора наук	кандида- та наук	профес- сора	доцента															
1	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	
профессорско-преподавательский состав – всего (сумма строк 08 – 13)	192	192	30	113	22	71	89	37	1	1	6	1	3	12	1	-	9	-	1	157	
в том числе: деканы факультетов	5	5	1	4	1	4	-	1	-	-	-	-	-	-	-	-	-	-	-	5	
заведующие кафедрами	19	19	15	4	13	5	3	3	-	-	1	-	-	-	-	-	-	-	-	18	
профессора	16	16	13	3	8	3	4	2	1	-	-	-	-	1	-	-	-	-	-	14	
доценты	109	109	1	101	-	59	57	20	-	-	3	1	3	7	-	-	7	-	1	87	
старшие преподаватели	31	31	-	1	-	-	19	4	-	1	-	-	-	3	1	-	1	-	-	25	
преподаватели, ассистенты	12	12	-	-	-	-	6	7	-	-	2	-	-	1	-	-	1	-	-	8	

Рис. 3.14 – Распределение по численности персонала. Штатные сотрудники

	Всего	из гр.3 имеют высшее образо- вание	из гр. 4 имеют:				Из гр. 3		из гр.3 работают на												
			ученую степень		ученое звание		жен- щины	освоили допол- нительные про- граммы повыше- ния квалифика- ции и (или) про- фессиональной переподготовки за предыдущий учебный год	0,1 ставки	0,2 ставки	0,25 ставки	0,3 ставки	0,4 ставки	0,5 ставки	0,6 ставки	0,7 ставки	0,75 ставки	0,8 ставки	0,9 ставки	1 ставку	
			доктора наук	кандида- та наук	профес- сора	доцента															
1	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	
Численность внешних совместителей – всего	47	44	16	20	9	14	16	14	2	5	14	-	1	25	-	-	-	-	-	-	
из них: профессорско-преподавательский состав – всего (сумма строк 03 – 08)	35	35	15	19	8	13	8	14	2	5	14	-	-	14	-	-	-	-	-	-	
в том числе: деканы факультетов	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	
заведующие кафедрами	3	3	3	-	2	-	-	1	-	-	2	-	-	1	-	-	-	-	-	-	
профессора	13	13	12	1	6	5	-	4	1	3	4	-	-	5	-	-	-	-	-	-	
доценты	18	18	-	18	-	8	8	9	1	2	7	-	-	8	-	-	-	-	-	-	
старшие преподаватели	1	1	-	-	-	-	-	-	-	-	1	-	-	-	-	-	-	-	-	-	
преподаватели, ассистенты	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	

Рис. 3.15 – Распределение по численности персонала. Внешние совместители

	Всего	из гр.3 работают на											
		0,1 ставки	0,2 ставки	0,25 ставки	0,3 ставки	0,4 ставки	0,5 ставки	0,6 ставки	0,7 ставки	0,75 ставки	0,8 ставки	0,9 ставки	1 ставку
1	3	4	5	6	7	8	9	10	11	12	13	14	15
Численность работников профессорско-преподавательского состава – всего (без учета внешних и внутренних совместителей)	192	1	1	6	1	3	12	1	-	9	-	1	157
из них: доктор наук	30	1	-	-	-	-	-	-	-	-	-	-	29
кандидаты наук	113	-	-	3	1	3	9	-	-	7	-	1	89
Численность работников профессорско-преподавательского состава – всего (внешние совместители)	35	2	5	14	-	-	14	-	-	-	-	-	-
из них: доктор наук	15	1	3	5	-	-	6	-	-	-	-	-	-
кандидаты наук	19	1	2	8	-	-	8	-	-	-	-	-	-
Численность научных работников – всего (без учета внешних и внутренних совместителей)	8	-	-	-	-	-	-	-	-	-	-	-	8
из них: доктор наук	-	-	-	-	-	-	-	-	-	-	-	-	-
кандидаты наук	-	-	-	-	-	-	-	-	-	-	-	-	-
Численность научных работников – всего (внешние совместители)	3	-	-	-	-	-	3	-	-	-	-	-	-
из них: доктор наук	1	-	-	-	-	-	1	-	-	-	-	-	-
кандидаты наук	1	-	-	-	-	-	1	-	-	-	-	-	-

Рис. 3.16 – Сведения об ученых степенях

	Численность работников, имеющих общий стаж работы, всего (сумма гр.4 – 9)	из гр. 3 – имеют общий стаж работы, лет						Из гр. 3 имеют педагогический стаж работы, всего (сумма гр.11 – 16)	из гр. 10 – имеют педагогический стаж работы, лет					
		до 3	от 3 до 5	от 5 до 10	от 10 до 15	от 15 до 20	20 и более		до 3	от 3 до 5	от 5 до 10	от 10 до 15	от 15 до 20	20 и более
1	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Профессорско-преподавательский состав – всего (сумма строк 06 – 11)	192	4	4	30	36	11	107	191	6	16	43	26	27	73
в том числе: деканы факультетов	5	-	-	1	1	1	2	5	-	-	2	1	-	2
заведующие кафедрами	19	-	-	-	2	1	16	19	-	-	5	1	-	13
профессора	16	-	-	-	2	1	13	16	-	-	-	4	3	9
доценты	109	-	-	20	30	5	54	109	3	10	28	15	15	38
старшие преподаватели	31	1	2	7	-	2	19	31	-	4	6	4	7	10
преподаватели, ассистенты	12	3	2	2	1	1	3	11	3	2	2	1	2	1

Рис. 3.17 – Распределение персонала по стажу

	Всего (сумма гр.4, 6, 8, 10, 12, 14, 16, 18, 20, 22)	Число полных лет по состоянию на 1 января 2016 года																			
		менее 25		25 – 29		30 – 34		35 – 39		40 – 44		45 – 49		50 – 54		55 – 59		60 – 64		65 и более	
		всего	из них женщины	всего	из них женщины	всего	из них женщины	всего	из них женщины	всего	из них женщины	всего	из них женщины	всего	из них женщины	всего	из них женщины	всего	из них женщины		
1	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Профессорско-преподавательский состав – всего (сумма строк 06 – 11)	192	1	-	13	5	38	25	23	9	11	6	14	8	17	11	14	8	20	8	41	9
в том числе: деканы факультетов	5	-	-	-	-	2	-	-	-	1	-	-	-	-	-	-	-	1	-	1	-
заведующие кафедрами	19	-	-	-	-	-	-	2	-	-	-	-	-	3	1	5	2	3	-	6	-
профессора	16	-	-	-	-	-	-	1	-	2	-	-	-	2	1	1	1	2	1	8	1
доценты	109	-	-	7	3	28	19	18	8	4	3	9	5	7	6	6	4	9	4	21	5
старшие преподаватели	31	-	-	3	1	6	5	-	-	4	3	5	3	3	2	1	-	4	2	5	3
преподаватели, ассистенты	12	1	-	3	1	2	1	2	1	-	-	-	-	2	1	1	1	1	1	-	-

Рис. 3.18 – Распределение персонала по полу и возрасту

3.3 Внедрение приложения «Управление ППС ПГУТИ»

Для внедрения современных информационных технологий нужно затратить не мало материальных ресурсов. Требуется, как минимум, анализ экономических последствий (оценки экономической эффективности того или иного шага преобразования системы).

Внедрение автоматизированной системы зависит от многих факторов и является нетривиальной практической задачей. Для внедрения предусмотрена многоэтапная процедура, на каждом шаге которой, осуществляется тесное взаимодействие отделов специалистов исполнителя и заказчика, контроль выполнения предъявляемых требований, тестирование и обучение персонала.

Можно выделить три этапа внедрения информационной системы:

1. Исследование. Изначально проводится исследование предметной области, бизнес процессов и составляется техническое задание.
2. Тестирование. Проходит этап тестирование и исправления найденных ошибок.
3. Внедрение. Внедрение готового продукта и обучение персонала.

На этап исследования, как правило, уделяется наибольшее время. Данный шаг, при верно составленным техническом задании, позволяет допускать наименьшее количество ошибок и сократить процесс разработки. Необходимо определить наиболее важные функции, которые необходимы заказчику и определить, как это повлияет на стоимость конечного программного продукта и время проектирования системы.

На этапе тестирования необходимо контролировать весь жизненный цикл проектирования системы, соответственно со стороны заказчика необходим человек хорошо знакомый с задачами компании и её бизнес процессами.

Множество проектов после внедрения не приносили желаемых результатов из-за нежелания или не умения людей использовать новую

систему, поэтому на стадии внедрения необходимо проводить тренинги и показывать, как использование системы позволит избавиться от рутинных задач и поможет в оптимизации работы.

После внедрения, как правило, систему не используют как основной инструмент. Нужно проанализировать результат от внедрения, решены ли поставленные задачи.

Внедрение считается успешным, только если система справляется с поставленными задачами и позволяет получать выгоду, а именно оптимизирует работу, позволяет выполнять работу быстрее, повышая качество процессов. Необходим постоянный анализ показателей работы системы, а также степень заинтересованности персонала в использовании данной системы.

Процесс внедрения информационной системы занимает как минимум несколько месяцев. На протяжении этого времени важно фокусироваться на целях, которые компания хочет достигнуть, внедряя систему. Логично помнить о возможных рисках и финансовых издержках [38].

Разработанное приложение удовлетворяет поставленным требованиям и, на сегодняшний день, находится на стадии внедрения работниками УОУП при непосредственной поддержке разработчика системы.

Заключение

Информация во все времена была самым важным ресурсом. Достоверная информация позволяет специалистам различных областей осуществлять свою профессиональную деятельность наиболее эффективно. При постоянно возрастающем объеме и роли информации возникает необходимость применения информационных технологий, позволяющих осуществлять ее сбор, структурирование, хранение, поиск, обработку и выдачу в соответствии с требованиями, предъявляемыми пользователями. Актуальность данной проблемы послужила выбором темы дипломной работы: «Разработка ПО “Мониторинг эффективности” по профессорскому преподавательскому составу»

Первая глава посвящена рассмотрению предметной области. Была выявлена связь между всеми подразделениями внутренней структуры университета с учебным отделом. Описан основной принцип формирования отчётов. Разработаны формы (представления) и схема управления отчётами по ППС. Было разработано приложение способное формировать необходимые отчёты за указанный период. Перед разработкой были рассмотрены уже имеющиеся программные продукты. Рассмотренные решения очень дорогостоящие и не соответствуют требованиям, предъявленным учебным отделом ПГУТИ. Было принято решение создать оригинальное приложение более легкое в эксплуатации с использованием базы данных MS SQL Server.

Во второй главе осуществлено описание процесса разработки. Представлены функциональная схема, ег-модель, интерфейс пользователя, укрупненная схема алгоритма. Разработаны схема базы данных, запросы, отчеты и формы. Следствием данной главы является готовое разработанное приложение, которое решает все поставленные задачи:

- добавление, редактирование и удаление информации о сотрудниках;

- основные отчёты: распределение численности персонала по уровню образования с разделением на штатных сотрудников, внутренних совместителей, сторонних совместителей; сведения об учёных степенях ППС и научных работниках; распределение персонала по стажу; распределение персонала по полу и возрасту. Промежуточные отчёты: штатные сотрудники; сторонние совместители; внешние совместители
- разработан графический интерфейс пользователя;
- разработан алгоритм авторизации приложения.

В третьей главе представлено разработанное руководство пользователя. Рассмотрены анализ и результаты тестирования приложения.

Из вышеизложенного можно сделать вывод, что поставленные задачи выполнены, основная цель достигнута. Приложение «Управление ППС ПГУТИ» можно считать разработанным.

Список использованных источников

1. Диго, С.М. Базы данных: проектирование и управление [Текст] – М. : Центр ЕАОИ, 2008. – 171 с.
2. Толстолобов, А.П. SQL в примерах и задачах [Текст] / А.П. Толстолобов, И.Ф. Астахова, В. М Мельников. – М. : Новое знание, 2002. – 176 с.
3. Будущее С#. [Электронный ресурс]. – Режим доступа: <http://habrahabr.ru>, свободный. – Загл. с экрана.
4. Алексеенко, В.А. Организация и ведение учебного процесса в вузе [Текст] / В.А. Алексеенко – ИКАР, 2005. – 493 с.
5. Гагарина, Л.Г. Технология разработки программного обеспечения [Текст] / Л.Г. Гагарина, Е.В. Кокорева, Б.Д. Виснаул – М. : Форум, 2008. – 400 с.
6. Биллиг, В. Основы программирования на С#. [Электронный ресурс]. – Режим доступа: <http://www.intuit.ru>, свободный. – Загл. с экрана.
7. Вендров, А. М. Проектирование программного обеспечения экономических информационных систем [Текст]. – М.: Финансы и статистика, 2005. – 450 с.
8. ГИВЦ Рособразование [Электронный ресурс]. Метод. указ. по заполнению формы №1-Мониторинг. Утв. А.А. Климов; Мин. Обр. и науки РФ – док. в формате pdf, 2015. – 59 с.
9. Албахари, Д. С# 5.0. Справочник. Полное описание языка [Текст] / Д. Албахари, Б. Албахари. – М. : Вильямс, 2014. – 1008 с.
10. ГИВЦ Рособразование [Электронный ресурс]. Федеральное статистическое наблюдение. Приказ Росстата: об утв. формы от 21.08.2014 № 529. – док. в формате pdf – 60 с.
11. Голицына, О.Л. Базы данных [Текст] / О.Л. Голицына, Н.В. Максимов, – М. : Форум, 2003. – 352 с. (про sql)

12. Грекул, В.И. Проектирование информационных систем. [Электронный ресурс]. – Режим доступа: <http://www.intuit.ru>, свободный. – Загл. с экрана.
13. Зафиевский, А.В. Автоматизация управления учебным процессом в вузе // Успехи современного естествознания. – 2010. – №1. – С. 115-117.
14. Зудилова, Т. В. Создание запросов в Microsoft SQL Server 2008 [Текст] / Т.В. Зудилова, Г. Ю. Шмелева. – СПб: НИУ ИТМО, 2013. – 149 с.
15. Иаклаков, С.В. BPWin и ERWin. CASE-средства разработки информационных систем [Текст]. – М.: Диалог-МИФИ, 2005. – 256 с.
16. Кадры бюджетного учреждения. [Электронный ресурс]. – Режим доступа: <http://samara.1cbit.ru>, свободный. – Загл. с экрана.
17. Коннолли, Т., Базы данных [Текст] / Т. Коннолли, К. Бегг. – М. : Вильямс, 2003. – 1440 с.
18. Константайн, Л. Разработка программного обеспечения [Текст] / Л. Константайн, Л. Локвуд – М. : Питер, 2004. – 560 с.
19. Критерии выбора СУБД. [Электронный ресурс]. – Режим доступа: <http://forum.vingrad.ru>, свободный. – Загл. с экрана.
20. Малыхина, М.П. Базы данных: основы, проектирование, использование [Текст]. – М. : Финист, 2004. – 512 с.
21. Маркин А.В. Учебное пособие по SQL [Текст] / А.В. Маркин – Диалог МИФИ, 2008. – 312 с.
22. Модель сущность-связь. [Электронный ресурс]. – Режим доступа: <http://cyberguru.ru>, свободный. – Загл. с экрана.
23. Модуль кадры. [Электронный ресурс]. – Режим доступа: <http://tandemservice.ru>, свободный. – Загл. с экрана.
24. Петкович, Д. Microsoft SQL Server 2008. Руководство для начинающих [Текст]. – М. : БХВ-Петербург, 2009. – 752 с.

25. РД 50-34.698-90. Руководство пользователя. Пример оформления [Текст]. – Введ. 1992 – 01 – 01. – М. : Изд-во стандартов, 1991. – 33 с.

Приложение А – Программный код

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Data;
using System.Text;
using System.Threading.Tasks;
using System.Collections;
using System.Configuration;
using System.Data.Common;
using System.Data.SqlClient;

namespace PGUTI
{
    public static class Data
    {
        private static string between(DateTime startDate, DateTime endDate)
        {
            return String.Format(" (((start_date < '{1}' and start_date > '{0}' ) and (end_date is null)) or ((start_date between '{0}' and '{1}') or (end_date between '{0}' and '{1}')) ) ", ReverseDateTime(startDate), ReverseDateTime(endDate));
        }

        private static string connectionString =
            ConfigurationManager.ConnectionStrings["Pguti.Connection"].ConnectionString; //Строка
            подключения к базе , из файла конфигурации

        private static DataSet ds = new DataSet(); //Объект запроса

        public static string ReverseDateTime(DateTime date)
        {
            return date.ToString("yyyy.MM.dd hh:mm:ss");
        }

        private static string[] getPreviousDate() //Получаем предыдущий учебный год
        { //сейчас 2015 год, для примера
            string[] date = new string[2];
            date[0] = DateTime.Now.Year.ToString();
            if (DateTime.Now.Month < 9) //Если месяц меньше 9, тогда нынешний учебный год
            это 2014-2015 , а предыдущий соответственно 2013-2014
            {
                date[0] = "" + (DateTime.Now.Year - 2).ToString() + ".9.1";
                date[1] = "" + (DateTime.Now.Year - 1).ToString() + ".8.31";
            }
            else //Если это не так, тогда текущий учебный год 2015-2016 а предыдущий 2014-
            2015
            {
                date[0] = "" + (DateTime.Now.Year - 1).ToString() + ".9.1";
                date[1] = "" + (DateTime.Now.Year).ToString() + ".8.31";
            }
            return date;
        }

        public static class EditForm
        {
            public static DataSet getTeachers(int id) //Главная таблица
            {
                string result = "select * from dbo.Teachers where id = " + id;
                return ds = NDataAccess.DataAccess.GetDataSet(@result, "Table1",
connectionString);
            }
        }
    }
}
```

Продолжение приложения А

```
}  
}  
  
//Справочники  
public static class Reference  
{  
    //Faculties  
    public static DataSet getFaculties()  
    {  
        string result = "SELECT id,name as 'Полное название',second_name as  
'Сокращённое название' FROM [PGUTI_faculty].[dbo].[Faculties]";  
        return NDataAccess.DataAccess.GetDataSet(@result, "Faculties",  
connectionString);  
    }  
    public static DataSet getFaculties(int id)  
    {  
        string result = "SELECT id,name as 'Полное название',second_name as  
'Сокращённое название' FROM [PGUTI_faculty].[dbo].[Faculties] where id=" + id;  
        return NDataAccess.DataAccess.GetDataSet(@result, "Faculties",  
connectionString);  
    }  
    public static void editFaculties(int id, string name, string secondName)  
    {  
        string result = "UPDATE [PGUTI_faculty].[dbo].[Faculties] SET name='" +  
name + "' , second_name='" + secondName + "' where id=" + id;  
        //return NDataAccess.DataAccess.GetDataSet(@result, "Faculties",  
connectionString);  
        NDataAccess.DataAccess.ExecuteNonQuery2(result, connectionString);  
    }  
    public static void addFaculties(int id, string name, string secondName)  
    {  
        string result = "INSERT INTO [PGUTI_faculty].[dbo].[Faculties]  
(id,name,second_name) VALUES ((select max(id)+1 from  
[PGUTI_faculty].[dbo].[Faculties]),'" + name + "', '" + secondName + "') ";  
        NDataAccess.DataAccess.ExecuteNonQuery2(result, connectionString);  
    }  
    public static void dellFaculties(int id)  
    {  
        string result = "DELETE FROM [dbo].[Faculties] where id=" + id;  
        NDataAccess.DataAccess.ExecuteNonQuery2(result, connectionString);  
    }  
  
    //Cairs  
    public static DataSet getCairs()  
    {  
        string result = "SELECT id,name as 'Полное название',second_name as  
'Сокращённое название' FROM [PGUTI_faculty].[dbo].[Cairs]";  
        return NDataAccess.DataAccess.GetDataSet(@result, "Cairs",  
connectionString);  
    }  
    public static DataSet getCairs(int id)  
    {  
        string result = "SELECT id,name as 'Полное название',second_name as  
'Сокращённое название' FROM [PGUTI_faculty].[dbo].[Cairs] where id=" + id;  
        return NDataAccess.DataAccess.GetDataSet(@result, "Cairs",  
connectionString);  
    }  
}
```

```
public static void editCair(int id, string name, string secondName)
{
```

Продолжение приложения А

```
    string result = "UPDATE [PGUTI_faculty].[dbo].[Cairs] SET name='" + name + "' ,
second_name='" + secondName + "' where id=" + id;
    //return NDataAccess.DataAccess.GetDataSet(@result, "Cairs",
connectionString);
    NDataAccess.DataAccess.ExecuteNonQuery2(result, connectionString);
}
public static void addCair(int id, string name, string secondName)
{
    string result = "INSERT INTO [PGUTI_faculty].[dbo].[Cairs]
(id,name,second_name) VALUES ((select max(id)+1 from [PGUTI_faculty].[dbo].[Cairs]),'" +
name + "',''" + secondName + "') ";
    NDataAccess.DataAccess.ExecuteNonQuery2(result, connectionString);
}
public static void dellCairs(int id)
{
    string result = "DELETE FROM [dbo].[Cairs] where id=" + id;
    NDataAccess.DataAccess.ExecuteNonQuery2(result, connectionString);
}

//Degrees
public static DataSet getDegrees()
{
    string result = "SELECT id,name as 'Полное название',second_name as
'Сокращённое название' FROM [PGUTI_faculty].[dbo].[Degrees]";
    return NDataAccess.DataAccess.GetDataSet(@result, "Degrees",
connectionString);
}
public static DataSet getDegrees(int id)
{
    string result = "SELECT id,name as 'Полное название',second_name as
'Сокращённое название' FROM [PGUTI_faculty].[dbo].[Degrees] where id=" + id;
    return NDataAccess.DataAccess.GetDataSet(@result, "Degrees",
connectionString);
}
public static void editDegrees(int id, string name, string secondName)
{
    string result = "UPDATE [PGUTI_faculty].[dbo].[Degrees] SET name='" +
name + "' , second_name='" + secondName + "' where id=" + id;
    //return NDataAccess.DataAccess.GetDataSet(@result, "Degrees",
connectionString);
    NDataAccess.DataAccess.ExecuteNonQuery2(result, connectionString);
}
public static void addDegrees(int id, string name, string secondName)
{
    string result = "INSERT INTO [PGUTI_faculty].[dbo].[Degrees]
(id,name,second_name) VALUES ((select max(id)+1 from [PGUTI_faculty].[dbo].[Degrees]),'" +
name + "',''" + secondName + "') ";
    NDataAccess.DataAccess.ExecuteNonQuery2(result, connectionString);
}
public static void dellDegrees(int id)
{
    string result = "DELETE FROM [dbo].[Degrees] where id=" + id;
    NDataAccess.DataAccess.ExecuteNonQuery2(result, connectionString);
}

//Titles
public static DataSet getTitles()
{
```

```

        string result = "SELECT id,name as 'Звание' FROM
[PGUTI_faculty].[dbo].[Titles]";

```

Продолжение приложения А

```

return NDataAccess.DataAccess.GetDataSet(@result, "Titles", connectionString);
    }
    public static DataSet getTitles(int id)
    {
        string result = "SELECT id,name as 'Звание' FROM
[PGUTI_faculty].[dbo].[Titles] where id=" + id;
        return NDataAccess.DataAccess.GetDataSet(@result, "Titles",
connectionString);
    }
    public static void editTitles(int id, string name)
    {
        string result = "UPDATE [PGUTI_faculty].[dbo].[Titles] SET name='" + name
+ "' where id=" + id;
        //return NDataAccess.DataAccess.GetDataSet(@result, "Titles",
connectionString);
        NDataAccess.DataAccess.ExecuteNonQuery2(result, connectionString);
    }
    public static void addTitles(int id, string name)
    {
        string result = "INSERT INTO [PGUTI_faculty].[dbo].[Titles] (id,name)
VALUES ((select max(id)+1 from [PGUTI_faculty].[dbo].[Titles]),'" + name + "') ";
        NDataAccess.DataAccess.ExecuteNonQuery2(result, connectionString);
    }
    public static void dellTitles(int id)
    {
        string result = "DELETE FROM [dbo].[Titles] where id=" + id;
        NDataAccess.DataAccess.ExecuteNonQuery2(result, connectionString);
    }

    //WorkingPositions
    public static DataSet getWorkingPositions()
    {
        string result = "SELECT id,name as 'Должность' FROM
[PGUTI_faculty].[dbo].[Working_positions]";
        return NDataAccess.DataAccess.GetDataSet(@result, "WorkingPositions",
connectionString);
    }
    public static DataSet getWorkingPositions(int id)
    {
        string result = "SELECT id,name as 'Должность' FROM
[PGUTI_faculty].[dbo].[Working_positions] where id=" + id;
        return NDataAccess.DataAccess.GetDataSet(@result, "WorkingPositions",
connectionString);
    }
    public static void editWorkingPositions(int id, string name)
    {
        string result = "UPDATE [PGUTI_faculty].[dbo].[Working_positions] SET
name='" + name + "' where id=" + id;
        //return NDataAccess.DataAccess.GetDataSet(@result, "Working_positions",
connectionString);
        NDataAccess.DataAccess.ExecuteNonQuery2(result, connectionString);
    }
    public static void addWorkingPositions(int id, string name)
    {
        string result = "INSERT INTO [PGUTI_faculty].[dbo].[Working_positions]
(id,name) VALUES ((select max(id)+1 from [PGUTI_faculty].[dbo].[Working_positions]),'" +
name + "') ";

```

```

        NDataAccess.DataAccess.ExecuteNonQuery2(result, connectionString);
    }
    public static void dellWorkingPositions(int id)
    {

```

Продолжение приложения А

```

        string result = "DELETE FROM [dbo].[Working_positions] where id="+id;
        NDataAccess.DataAccess.ExecuteNonQuery2(result, connectionString);
    }

}

//ШТАТ
public static class BasicStructure
{
    public static void getGroupBy(string dekan, string by)
    {
        string[] a = null;
        string result = "select COUNT(*) from dbo.Teachers where Dekan_Faculties
" + dekan + " group by " + by;
        ds = NDataAccess.DataAccess.GetDataSet(@result, "Table1",
connectionString);
        string b = "";
        for (int i = 0; i < ds.Tables[0].Rows.Count; i++)
        {
            b = ds.Tables[0].Rows[i].ItemArray[0].ToString();
        }
    }
    public static DataSet getDegreesAndTitle(DateTime startDate, DateTime
endDate)
    {
        string result = "SELECT COUNT(*) as Всего , (select Count(*) from
dbo.Teachers where degrees_id > 4 and degrees_id < 16 and terms_of_work like '%штатный
сотрудник%' and "+between(startDate, endDate)+" as Ученые_степени, (select Count(*)
from dbo.Teachers where degrees_id between 4 and 10 and terms_of_work like '%штатный
сотрудник%' and "+between(startDate, endDate)+" as Кандидаты_наук, (select Count(*)
from dbo.Teachers where degrees_id > 10 and degrees_id < 16 and terms_of_work like
'%штатный сотрудник%' and "+between(startDate, endDate)+" as Доктора_наук, (select
Count(*) from dbo.Teachers where titles_id > 0 and terms_of_work like '%штатный
сотрудник%' and "+between(startDate, endDate)+" as Ученые_звания, (select Count(*)
from dbo.Teachers where titles_id =1 and terms_of_work like '%штатный сотрудник%' and
"+between(startDate, endDate)+" as Профессора, (select Count(*) from dbo.Teachers
where titles_id = 2 and terms_of_work like '%штатный сотрудник%' and "+between(startDate,
endDate)+" as Доценты, (select COUNT(*) from dbo.Teachers where gender like 'Жен' and
terms_of_work like '%штатный сотрудник%' and "+between(startDate, endDate)+" as Женщины
FROM Teachers where terms_of_work like '%штатный сотрудник%' and "+between(startDate,
endDate)+" union all SELECT COUNT(*) as Всего, (select Count(*) from dbo.Teachers
where degrees_id > 4 and degrees_id < 16 and Dekan_Faculties is not null and
terms_of_work like '%штатный сотрудник%' and "+between(startDate, endDate)+" ) as
Ученые_степени, (select Count(*) from dbo.Teachers where degrees_id between 4 and 10
and Dekan_Faculties is not null and terms_of_work like '%штатный сотрудник%' and
"+between(startDate, endDate)+" as Кандидаты_наук, (select Count(*) from dbo.Teachers
where degrees_id > 10 and degrees_id < 16 and Dekan_Faculties is not null and
terms_of_work like '%штатный сотрудник%' and "+between(startDate, endDate)+" as
Доктора_наук, (select Count(*) from dbo.Teachers where titles_id > 0 and
Dekan_Faculties is not null and terms_of_work like '%штатный сотрудник%' and
"+between(startDate, endDate)+" as Ученые_звания, (select Count(*) from dbo.Teachers
where titles_id =1 and Dekan_Faculties is not null and terms_of_work like '%штатный
сотрудник%' and "+between(startDate, endDate)+" as Профессора, (select Count(*) from
dbo.Teachers where titles_id = 2 and Dekan_Faculties is not null and terms_of_work like
'%штатный сотрудник%' and "+between(startDate, endDate)+" as Доценты, (select
COUNT(*) from dbo.Teachers where gender like 'Жен' and terms_of_work like '%штатный
сотрудник%' and "+between(startDate, endDate)+" as Женщины FROM Teachers where
Dekan_Faculties is not null and terms_of_work like '%штатный сотрудник%' and
"+between(startDate, endDate)+" union all SELECT COUNT(*) as Всего, (select

```

```
Count(*) from dbo.Teachers where degrees_id > 4 and degrees_id < 16 and Job_title = 1
and terms of work like '%штатный сотрудник%' and "+between(startDate, endDate)+" as
```

Продолжение приложения А

[illegible]

сотрудник%' and "+between(startDate, endDate)+") as Доценты, (select COUNT(*) from
dbo.Teachers where gender like 'Жен' and Job_title =4 and terms_of_work like '%Штатный

Продолжение приложения А

сотрудник%' and "+between(startDate, endDate)+") as Женщины FROM Teachers where
Job_title = 4 and terms_of_work like '%Штатный сотрудник%' and "+between(startDate,
endDate)+") union all SELECT COUNT(*) as Всего, (select Count(*) from dbo.Teachers
where degrees_id > 4 and degrees_id < 16 and (Job_title = 5 or Job_title = 6) and
terms_of_work like '%Штатный сотрудник%' and "+between(startDate, endDate)+") as
Ученые_степени, (select Count(*) from dbo.Teachers where degrees_id between 4 and 10
and (Job_title = 5 or Job_title = 6) and terms_of_work like '%Штатный сотрудник%' and
"+between(startDate, endDate)+") as Кандидаты_наук, (select Count(*) from dbo.Teachers
where degrees_id > 10 and degrees_id < 16 and (Job_title = 5 or Job_title = 6) and
terms_of_work like '%Штатный сотрудник%' and "+between(startDate, endDate)+") as
Доктора_наук, (select Count(*) from dbo.Teachers where titles_id > 0 and (Job_title =
5 or Job_title = 6) and terms_of_work like '%Штатный сотрудник%' and "+between(startDate,
endDate)+") as Ученые_звания, (select Count(*) from dbo.Teachers where titles_id =1
and (Job_title = 5 or Job_title = 6) and terms_of_work like '%Штатный сотрудник%' and
"+between(startDate, endDate)+") as Профессора, (select Count(*) from dbo.Teachers
where titles_id = 2 and (Job_title = 5 or Job_title = 6) and terms_of_work like '%Штатный
сотрудник%' and "+between(startDate, endDate)+") as Доценты, (select COUNT(*) from
dbo.Teachers where gender like 'Жен' and (Job_title = 5 or Job_title = 6) and
terms_of_work like '%Штатный сотрудник%' and "+between(startDate, endDate)+") as Женщины
FROM Teachers where (Job_title = 5 or Job_title = 6) and terms_of_work like '%Штатный
сотрудник%' and "+between(startDate, endDate)+"";

return NDataAccess.DataAccess.GetDataSet(@result, "Table1",
connectionString);

```

    }
    public static DataSet getRate(DateTime startDate, DateTime endDate)
    {
        string result = "SELECT count(*), (select Count(*) from
        dbo.Teachers where rate = 0.1 and terms_of_work like '%Штатный сотрудник%' and
        "+between(startDate, endDate)+") as '0,1', (select Count(*) from dbo.Teachers where
        rate = 0.2 and terms_of_work like '%Штатный сотрудник%' and "+between(startDate,
        endDate)+") as '0,2', (select Count(*) from dbo.Teachers where rate = 0.3 and
        terms_of_work like '%Штатный сотрудник%' and "+between(startDate, endDate)+") as '0,3',
        (select Count(*) from dbo.Teachers where rate = 0.4 and terms_of_work like '%Штатный
        сотрудник%' and "+between(startDate, endDate)+") as '0,4', (select Count(*) from
        dbo.Teachers where rate = 0.5 and terms_of_work like '%Штатный сотрудник%' and
        "+between(startDate, endDate)+") as '0,5', (select Count(*) from dbo.Teachers where
        rate = 0.6 and terms_of_work like '%Штатный сотрудник%' and "+between(startDate,
        endDate)+") as '0,6', (select Count(*) from dbo.Teachers where rate = 0.7 and
        terms_of_work like '%Штатный сотрудник%' and "+between(startDate, endDate)+") as '0,7',
        (select Count(*) from dbo.Teachers where rate = 0.8 and terms_of_work like '%Штатный
        сотрудник%' and "+between(startDate, endDate)+") as '0,8', (select Count(*) from
        dbo.Teachers where rate = 0.9 and terms_of_work like '%Штатный сотрудник%' and
        "+between(startDate, endDate)+") as '0,9', (select Count(*) from dbo.Teachers where
        rate >0.9 and terms_of_work like '%Штатный сотрудник%' and "+between(startDate,
        endDate)+") as '1' FROM Teachers union all SELECT count(*), (select Count(*)
        from dbo.Teachers where rate = 0.1 and terms_of_work like '%Штатный сотрудник%' and
        "+between(startDate, endDate)+") and Dekan_Faculties is not null) as '0,1', (select
        Count(*) from dbo.Teachers where rate = 0.2 and terms_of_work like '%Штатный сотрудник%'
        and "+between(startDate, endDate)+") and Dekan_Faculties is not null) as '0,2',
        (select Count(*) from dbo.Teachers where rate = 0.3 and terms_of_work like '%Штатный
        сотрудник%' and "+between(startDate, endDate)+") and Dekan_Faculties is not null) as
        '0,3', (select Count(*) from dbo.Teachers where rate = 0.4 and terms_of_work like
        '%Штатный сотрудник%' and "+between(startDate, endDate)+") and Dekan_Faculties is not
        null) as '0,4', (select Count(*) from dbo.Teachers where rate = 0.5 and terms_of_work
        like '%Штатный сотрудник%' and "+between(startDate, endDate)+") and Dekan_Faculties is
        not null) as '0,5', (select Count(*) from dbo.Teachers where rate = 0.6 and
        terms_of_work like '%Штатный сотрудник%' and "+between(startDate, endDate)+") and
        Dekan_Faculties is not null) as '0,6', (select Count(*) from dbo.Teachers where rate =
        0.7 and terms_of_work like '%Штатный сотрудник%' and "+between(startDate, endDate)+") and
        Dekan_Faculties is not null) as '0,7', (select Count(*) from dbo.Teachers where rate =

```

0.8 and terms_of_work like '%Штатный сотрудник%' and "+between(startDate, endDate)+" and Dekan Faculties is not null) as '0,8', (select Count(*) from dbo.Teachers where rate =

Продолжение приложения А

[illegible]

```
terms_of_work like '%Штатный сотрудник%' and "+between(startDate, endDate)+" and
Job_title = 3) as '0,8', (select Count(*) from dbo.Teachers where rate = 0.9 and
```

Продолжение приложения А

```
terms_of_work like '%Штатный сотрудник%' and "+between(startDate, endDate)+" and
Job_title = 3) as '0,9', (select Count(*) from dbo.Teachers where rate >0.9 and
terms_of_work like '%Штатный сотрудник%' and "+between(startDate, endDate)+" and
Job_title = 3) as '1' FROM Teachers where Job_title = 3 union all SELECT
count(*), (select Count(*) from dbo.Teachers where rate = 0.1 and terms_of_work like
'%Штатный сотрудник%' and "+between(startDate, endDate)+" and Job_title = 4) as '0,1',
(select Count(*) from dbo.Teachers where rate = 0.2 and terms_of_work like '%Штатный
сотрудник%' and "+between(startDate, endDate)+" and Job_title = 4) as '0,2', (select
Count(*) from dbo.Teachers where rate = 0.3 and terms_of_work like '%Штатный сотрудник%'
and "+between(startDate, endDate)+" and Job_title = 4) as '0,3', (select Count(*)
from dbo.Teachers where rate = 0.4 and terms_of_work like '%Штатный сотрудник%' and
"+between(startDate, endDate)+" and Job_title = 4) as '0,4', (select Count(*) from
dbo.Teachers where rate = 0.5 and terms_of_work like '%Штатный сотрудник%' and
"+between(startDate, endDate)+" and Job_title = 4) as '0,5', (select Count(*) from
dbo.Teachers where rate = 0.6 and terms_of_work like '%Штатный сотрудник%' and
"+between(startDate, endDate)+" and Job_title = 4) as '0,6', (select Count(*) from
dbo.Teachers where rate = 0.7 and terms_of_work like '%Штатный сотрудник%' and
"+between(startDate, endDate)+" and Job_title = 4) as '0,7', (select Count(*) from
dbo.Teachers where rate = 0.8 and terms_of_work like '%Штатный сотрудник%' and
"+between(startDate, endDate)+" and Job_title = 4) as '0,8', (select Count(*) from
dbo.Teachers where rate = 0.9 and terms_of_work like '%Штатный сотрудник%' and
"+between(startDate, endDate)+" and Job_title = 4) as '0,9', (select Count(*) from
dbo.Teachers where rate >0.9 and terms_of_work like '%Штатный сотрудник%' and
"+between(startDate, endDate)+" and Job_title = 4) as '1' FROM Teachers where
Job_title = 4 union all SELECT count(*), (select Count(*) from dbo.Teachers where
rate = 0.1 and terms_of_work like '%Штатный сотрудник%' and "+between(startDate,
endDate)+" and (Job_title = 5 or Job_title = 6)) as '0,1', (select Count(*) from
dbo.Teachers where rate = 0.2 and terms_of_work like '%Штатный сотрудник%' and
"+between(startDate, endDate)+" and (Job_title = 5 or Job_title = 6)) as '0,2',
(select Count(*) from dbo.Teachers where rate = 0.3 and terms_of_work like '%Штатный
сотрудник%' and "+between(startDate, endDate)+" and (Job_title = 5 or Job_title = 6)) as
'0,3', (select Count(*) from dbo.Teachers where rate = 0.4 and terms_of_work like
'%Штатный сотрудник%' and "+between(startDate, endDate)+" and (Job_title = 5 or
Job_title = 6)) as '0,4', (select Count(*) from dbo.Teachers where rate = 0.5 and
terms_of_work like '%Штатный сотрудник%' and "+between(startDate, endDate)+" and
(Job_title = 5 or Job_title = 6)) as '0,5', (select Count(*) from dbo.Teachers where
rate = 0.6 and terms_of_work like '%Штатный сотрудник%' and "+between(startDate,
endDate)+" and (Job_title = 5 or Job_title = 6)) as '0,6', (select Count(*) from
dbo.Teachers where rate = 0.7 and terms_of_work like '%Штатный сотрудник%' and
"+between(startDate, endDate)+" and (Job_title = 5 or Job_title = 6)) as '0,7',
(select Count(*) from dbo.Teachers where rate = 0.8 and terms_of_work like '%Штатный
сотрудник%' and "+between(startDate, endDate)+" and (Job_title = 5 or Job_title = 6)) as
'0,8', (select Count(*) from dbo.Teachers where rate = 0.9 and terms_of_work like
'%Штатный сотрудник%' and "+between(startDate, endDate)+" and (Job_title = 5 or
Job_title = 6)) as '0,9', (select Count(*) from dbo.Teachers where rate >0.9 and
terms_of_work like '%Штатный сотрудник%' and "+between(startDate, endDate)+" and
(Job_title = 5 or Job_title = 6)) as '1' FROM Teachers where (Job_title = 5 or Job_title
= 6) and terms_of_work like '%Штатный сотрудник%' and "+between(startDate, endDate)";
return NDataAccess.DataAccess.GetDataSet(@result, "Table1",
connectionString);
}
public static DataSet getQualification(DateTime startDate, DateTime endDate)
{
    string[] date = getPreviousDate();
    string result = "SELECT COUNT(Training_dates) as Bcero FROM Teachers
where terms_of_work like '%Штатный сотрудник%' and (Training_dates between " + date[0] +
" and " + date[1] + ") union all SELECT COUNT(Training_dates) FROM Teachers where
Dekan_Faculties is not null and terms_of_work like '%Штатный сотрудник%' and
"+between(startDate, endDate)+" and (Training_dates between " + date[0] + " and " +
date[1] + ") union all SELECT COUNT(Training_dates) as Bcero FROM Teachers where
```

```
Job_title = 1 and terms_of_work like '%штатный сотрудник%' and "+between(startDate,
endDate)+"and (Training_dates between " + date[0] + " and " + date[1] + ") union all
```

Продолжение приложения А

```
SELECT COUNT(Training_dates) as Всего FROM Teachers where Job_title = 2 and
terms_of_work like '%штатный сотрудник%' and "+between(startDate, endDate)+" and
(Training_dates between " + date[0] + " and " + date[1] + ") union all SELECT
COUNT(Training_dates) as Всего FROM Teachers where Job_title = 3 and terms_of_work
like '%штатный сотрудник%' and "+between(startDate, endDate)+"and (Training_dates between
" + date[0] + " and " + date[1] + ") union all SELECT COUNT(Training_dates) as Всего FROM
Teachers where Job_title = 4 and terms_of_work like '%штатный сотрудник%' and
"+between(startDate, endDate)+"and (Training_dates between " + date[0] + " and " +
date[1] + ") union all SELECT COUNT(Training_dates) as Всего FROM Teachers where
(Job_title = 5 or Job_title = 6) and terms_of_work like '%штатный сотрудник%' and
"+between(startDate, endDate)+" and (Training_dates between " + date[0] + " and " +
date[1] + ")";
return NDataAccess.DataAccess.GetDataSet(@result, "Table1",
connectionString);
}

//Проф
public static DataSet getProfessors(DateTime startDate, DateTime
endDate)//TODO надо развернуть дату, везде =( startDate.ToString("ddMMyyy");
{
    string result = "select
f.surname,f.name,f.middlename,w.name,c.second_name,s.name,a.second_name,f.rate,convert(VA
RCHAR(10),f.birthday,105) from dbo.Teachers as f left join dbo.Cairs as c on c.id=f.Cairs
left join dbo.Working_positions as w on w.id=f.Job_title left join dbo.Degrees as a on
a.id = f.degrees_id left join dbo.Titles as s on s.id = f.titles_id where f.terms_of_work
like '%штатный сотрудник%' and f.Job_title =2 and "+between(startDate, endDate)+" order
by rate";
    return NDataAccess.DataAccess.GetDataSet(@result, "Table1",
connectionString);
}
public static DataSet getProfessorsDoc(DateTime startDate, DateTime endDate)
{
    string result = "select
f.surname,f.name,f.middlename,w.name,c.second_name,s.name,a.second_name,f.rate,convert(VA
RCHAR(10),f.birthday,105) from dbo.Teachers as f left join dbo.Cairs as c on c.id=f.Cairs
left join dbo.Working_positions as w on w.id=f.Job_title left join dbo.Degrees as a on
a.id = f.degrees_id left join dbo.Titles as s on s.id = f.titles_id where f.terms_of_work
like '%штатный сотрудник%' and f.Job_title =2 and f.degrees_id>10 and f.degrees_id<16 and
"+between(startDate, endDate)+" order by rate";
    return NDataAccess.DataAccess.GetDataSet(@result, "Table1",
connectionString);
}
public static DataSet getProfessorsCand(DateTime startDate, DateTime endDate)
{
    string result = "select
f.surname,f.name,f.middlename,w.name,c.second_name,s.name,a.second_name,f.rate,convert(VA
RCHAR(10),f.birthday,105) from dbo.Teachers as f left join dbo.Cairs as c on c.id=f.Cairs
left join dbo.Working_positions as w on w.id=f.Job_title left join dbo.Degrees as a on
a.id = f.degrees_id left join dbo.Titles as s on s.id = f.titles_id where
f.terms_of_work like '%штатный сотрудник%' and f.Job_title =2 and f.degrees_id>4 and
f.degrees_id<11 and f.titles_id=1 and "+between(startDate, endDate)+" order by rate";
    return NDataAccess.DataAccess.GetDataSet(@result, "Table1",
connectionString);
}
//Доценты
public static DataSet getDocentsCandAndOrDocent(DateTime startDate, DateTime
endDate)
{
    string result = "select
f.surname,f.name,f.middlename,w.name,c.second_name,s.name,a.second_name,f.rate,convert(VA
```



```
RCHAR(10),f.birthday,105) from dbo.Teachers as f left join dbo.Cairs as c on c.id=f.Cairs
left join dbo.Working_positions as w on w.id=f.Job_title left join dbo.Degrees as a on
```

Продолжение приложения А

```
a.id = f.degrees_id left join dbo.Titles as s on s.id = f.titles_id where f.terms_of_work
like '%Штатный сотрудник%' and f.Job_title =3 and ((f.degrees_id>4 and f.degrees_id<11)
or (f.titles_id=2) or(f.degrees_id>4 and f.degrees_id<11 and f.titles_id=2) ) and
"+between(startDate, endDate)+" order by rate";
        return NDataAccess.DataAccess.GetDataSet(@result, "Table1",
connectionString);
    }
    public static DataSet getDocentsNull(DateTime startDate, DateTime endDate)
    {
        string result = "select
f.surname,f.name,f.middlename,w.name,c.second_name,s.name,a.second_name,f.rate,convert(VA
RCHAR(10),f.birthday,105) from dbo.Teachers as f left join dbo.Cairs as c on c.id=f.Cairs
left join dbo.Working_positions as w on w.id=f.Job_title left join dbo.Degrees as a on
a.id = f.degrees_id left join dbo.Titles as s on s.id = f.titles_id where f.terms_of_work
like '%Штатный сотрудник%' and f.Job_title =3 and ((f.degrees_id=16) or (f.titles_id=3)
or(f.degrees_id=16 and f.titles_id=3) ) and "+between(startDate, endDate)+" order by
rate";
        return NDataAccess.DataAccess.GetDataSet(@result, "Table1",
connectionString);
    }
    //Старшие преп
    public static DataSet getSenior(DateTime startDate, DateTime endDate)
    {
        string result = "select
f.surname,f.name,f.middlename,w.name,c.second_name,s.name,a.second_name,f.rate,convert(VA
RCHAR(10),f.birthday,105) from dbo.Teachers as f left join dbo.Cairs as c on c.id=f.Cairs
left join dbo.Working_positions as w on w.id=f.Job_title left join dbo.Degrees as a on
a.id = f.degrees_id left join dbo.Titles as s on s.id = f.titles_id where f.terms_of_work
like '%Штатный сотрудник%' and f.Job_title =4 and "+between(startDate, endDate)+" order
by rate";
        return NDataAccess.DataAccess.GetDataSet(@result, "Table1",
connectionString);
    }
    //Ассистенты
    public static DataSet getAssist(DateTime startDate, DateTime endDate)
    {
        string result = "select
f.surname,f.name,f.middlename,w.name,c.second_name,s.name,a.second_name,f.rate,convert(VA
RCHAR(10),f.birthday,105) from dbo.Teachers as f left join dbo.Cairs as c on c.id=f.Cairs
left join dbo.Working_positions as w on w.id=f.Job_title left join dbo.Degrees as a on
a.id = f.degrees_id left join dbo.Titles as s on s.id = f.titles_id where f.terms_of_work
like '%Штатный сотрудник%' and (f.Job_title =5 or f.Job_title=6) and "+between(startDate,
endDate)+" order by rate";
        return NDataAccess.DataAccess.GetDataSet(@result, "Table1",
connectionString);
    }
    //Внутр
    public static class InternalStructure
    {
        public static DataSet getDegreesAndTitle(DateTime startDate, DateTime
endDate)
        {
            string result = "SELECT    COUNT(*) as Всего ,    (select Count(*) from
dbo.Teachers where degrees_id > 4  and degrees_id < 16 and terms_of_work like '%Штатные
совместители %' and "+between(startDate, endDate)+" ) as Ученые_степени,    (select
Count(*) from dbo.Teachers where degrees_id between 4 and 10 and terms_of_work like
'%Штатные совместители %' and "+between(startDate, endDate)+" ) as Кандидаты_наук,
(select Count(*) from dbo.Teachers where degrees_id > 10  and degrees_id < 16 and
terms_of_work like '%Штатные совместители %' and "+between(startDate, endDate)+" ) as
```

Доктора наук, (select Count(*) from dbo.Teachers where titles_id > 0 and terms_of_work like '%штатные совместители %' and "+between(startDate, endDate)+") as Ученые звания,

Продолжение приложения А

[illegible]

```
and Job_title = 3 and terms_of_work like '%Штатные совместители %' and
"+between(startDate, endDate)+") as Ученые_степени, (select Count(*) from dbo.Teachers
```

Продолжение приложения А

```
where degrees_id between 4 and 10 and Job_title = 3 and terms_of_work like '%Штатные
совместители %' and "+between(startDate, endDate)+") as Кандидаты_наук, (select
Count(*) from dbo.Teachers where degrees_id > 10 and degrees_id < 16 and Job_title = 3
and terms_of_work like '%Штатные совместители %' and "+between(startDate, endDate)+") as
Доктора_наук, (select Count(*) from dbo.Teachers where titles_id > 0 and Job_title = 3
and terms_of_work like '%Штатные совместители %' and "+between(startDate, endDate)+") as
Ученые_звания, (select Count(*) from dbo.Teachers where titles_id =1 and Job_title = 3
and terms_of_work like '%Штатные совместители %' and "+between(startDate, endDate)+") as
Профессора, (select Count(*) from dbo.Teachers where titles_id = 2 and Job_title = 3
and terms_of_work like '%Штатные совместители %' and "+between(startDate, endDate)+") as
Доценты, (select COUNT(*) from dbo.Teachers where gender like 'Жен' and Job_title = 3
and terms_of_work like '%Штатные совместители %' and "+between(startDate, endDate)+")as
Женщины FROM Teachers where Job_title = 3 and terms_of_work like '%Штатные
совместители %' and "+between(startDate, endDate)+") union all SELECT COUNT(*) as
Всего, (select Count(*) from dbo.Teachers where degrees_id > 4 and degrees_id < 16
and Job_title = 4 and terms_of_work like '%Штатные совместители %' and
"+between(startDate, endDate)+") as Ученые_степени, (select Count(*) from dbo.Teachers
where degrees_id between 4 and 10 and Job_title = 4 and terms_of_work like '%Штатные
совместители %' and "+between(startDate, endDate)+") as Кандидаты_наук, (select
Count(*) from dbo.Teachers where degrees_id > 10 and degrees_id < 16 and Job_title = 4
and terms_of_work like '%Штатные совместители %' and "+between(startDate, endDate)+") as
Доктора_наук, (select Count(*) from dbo.Teachers where titles_id > 0 and Job_title = 4
and terms_of_work like '%Штатные совместители %' and "+between(startDate, endDate)+") as
Ученые_звания, (select Count(*) from dbo.Teachers where titles_id =1 and Job_title = 4
and terms_of_work like '%Штатные совместители %' and "+between(startDate, endDate)+") as
Профессора, (select Count(*) from dbo.Teachers where titles_id = 2 and Job_title = 4
and terms_of_work like '%Штатные совместители %' and "+between(startDate, endDate)+") as
Доценты, (select COUNT(*) from dbo.Teachers where gender like 'Жен' and Job_title =4
and terms_of_work like '%Штатные совместители %' and "+between(startDate, endDate)+")as
Женщины FROM Teachers where Job_title = 4 and terms_of_work like '%Штатные
совместители %' and "+between(startDate, endDate)+") union all SELECT COUNT(*) as
Всего, (select Count(*) from dbo.Teachers where degrees_id > 4 and degrees_id < 16
and (Job_title = 5 or Job_title = 6) and terms_of_work like '%Штатные совместители %' and
"+between(startDate, endDate)+") as Ученые_степени, (select Count(*) from dbo.Teachers
where degrees_id between 4 and 10 and (Job_title = 5 or Job_title = 6) and terms_of_work
like '%Штатные совместители %' and "+between(startDate, endDate)+") as Кандидаты_наук,
(select Count(*) from dbo.Teachers where degrees_id > 10 and degrees_id < 16 and
(Job_title = 5 or Job_title = 6) and terms_of_work like '%Штатные совместители %' and
"+between(startDate, endDate)+") as Доктора_наук, (select Count(*) from dbo.Teachers
where titles_id > 0 and (Job_title = 5 or Job_title = 6) and terms_of_work like '%Штатные
совместители %' and "+between(startDate, endDate)+") as Ученые_звания, (select
Count(*) from dbo.Teachers where titles_id =1 and (Job_title = 5 or Job_title = 6) and
terms_of_work like '%Штатные совместители %' and "+between(startDate, endDate)+") as
Профессора, (select Count(*) from dbo.Teachers where titles_id = 2 and (Job_title = 5
or Job_title = 6) and terms_of_work like '%Штатные совместители %' and
"+between(startDate, endDate)+") as Доценты, (select COUNT(*) from dbo.Teachers where
gender like 'Жен' and (Job_title = 5 or Job_title = 6) and terms_of_work like '%Штатные
совместители %' and "+between(startDate, endDate)+")as Женщины FROM Teachers where
(Job_title = 5 or Job_title = 6) and terms_of_work like '%Штатные совместители %' and
"+between(startDate, endDate)+"";
return NDataAccess.DataAccess.GetDataSet(@result, "Table1",
connectionString);
}
public static DataSet getRate(DateTime startDate, DateTime endDate)
{
    string result = "SELECT count(*), (select Count(*) from
dbo.Teachers where rate = 0.1 and terms_of_work like '%Штатные совместители %' and
"+between(startDate, endDate)+") as '0,1', (select Count(*) from dbo.Teachers where
rate = 0.2 and terms_of_work like '%Штатные совместители %' and "+between(startDate,
endDate)+") as '0,2', (select Count(*) from dbo.Teachers where rate = 0.3 and
```

```
terms_of_work like '%штатные совместители %' and "+between(startDate, endDate)+" as
'0,3', (select Count(*) from dbo.Teachers where rate = 0.4 and terms_of_work like
```

Продолжение приложения А

[illegible]


```

"+between(startDate, endDate)+" and Job_title = 2) as '0,3',      (select Count(*) from
dbo.Teachers where rate = 0.4 and terms_of work like '%штатные совместители %' and

```

Продолжение приложения А

[illegible]

```
terms_of_work like '%Штатные совместители %' and "+between(startDate, endDate)+" and
(Job_title = 5 or Job_title = 6)) as '0,3', (select Count(*) from dbo.Teachers where
```

Продолжение приложения А

```
rate = 0.4 and terms_of_work like '%Штатные совместители %' and "+between(startDate,
endDate)+" and (Job_title = 5 or Job_title = 6)) as '0,4', (select Count(*) from
dbo.Teachers where rate = 0.5 and terms_of_work like '%Штатные совместители %' and
"+between(startDate, endDate)+" and (Job_title = 5 or Job_title = 6)) as '0,5',
(select Count(*) from dbo.Teachers where rate = 0.6 and terms_of_work like '%Штатные
совместители %' and "+between(startDate, endDate)+" and (Job_title = 5 or Job_title =
6)) as '0,6', (select Count(*) from dbo.Teachers where rate = 0.7 and terms_of_work
like '%Штатные совместители %' and "+between(startDate, endDate)+" and (Job_title = 5 or
Job_title = 6)) as '0,7', (select Count(*) from dbo.Teachers where rate = 0.8 and
terms_of_work like '%Штатные совместители %' and "+between(startDate, endDate)+" and
(Job_title = 5 or Job_title = 6)) as '0,8', (select Count(*) from dbo.Teachers where
rate = 0.9 and terms_of_work like '%Штатные совместители %' and "+between(startDate,
endDate)+" and (Job_title = 5 or Job_title = 6)) as '0,9', (select Count(*) from
dbo.Teachers where rate >0.9 and terms_of_work like '%Штатные совместители %' and
"+between(startDate, endDate)+" and (Job_title = 5 or Job_title = 6)) as '1' FROM
Teachers where (Job_title = 5 or Job_title = 6) and terms_of_work like '%Штатные
совместители %' and "+between(startDate, endDate)"";
return NDataAccess.DataAccess.GetDataSet(@result, "Table1",
connectionString);
}//////////INT
public static DataSet getQualification(DateTime startDate, DateTime endDate)
{
    string[] date = getPreviousDate();
    string result = "SELECT COUNT(Training_dates) as Bcero FROM Teachers
where terms_of_work like '%Штатные совместители %' and (Training_dates between " +
date[0] + " and " + date[1] + ") union all SELECT COUNT(Training_dates) FROM Teachers
where Dekan_Faculties is not null and terms_of_work like '%Штатные совместители %' and
"+between(startDate, endDate)+" and (Training_dates between " + date[0] + " and " +
date[1] + ") union all SELECT COUNT(Training_dates) as Bcero FROM Teachers where
Job_title = 1 and terms_of_work like '%Штатные совместители %' and "+between(startDate,
endDate)+"and (Training_dates between " + date[0] + " and " + date[1] + ") union all
SELECT COUNT(Training_dates) as Bcero FROM Teachers where Job_title = 2 and
terms_of_work like '%Штатные совместители %' and "+between(startDate, endDate)+" and
(Training_dates between " + date[0] + " and " + date[1] + ") union all SELECT
COUNT(Training_dates) as Bcero FROM Teachers where Job_title = 3 and terms_of_work
like '%Штатные совместители %' and "+between(startDate, endDate)+"and (Training_dates
between " + date[0] + " and " + date[1] + ") union all SELECT COUNT(Training_dates) as
Bcero FROM Teachers where Job_title = 4 and terms_of_work like '%Штатные совместители %'
and "+between(startDate, endDate)+"and (Training_dates between " + date[0] + " and " +
date[1] + ") union all SELECT COUNT(Training_dates) as Bcero FROM Teachers where
(Job_title = 5 or Job_title = 6) and terms_of_work like '%Штатные совместители %' and
"+between(startDate, endDate)+"and (Training_dates between " + date[0] + " and " +
date[1] + ")";
return NDataAccess.DataAccess.GetDataSet(@result, "Table1",
connectionString);
}

//Профф
public static DataSet getProfessors(DateTime startDate, DateTime endDate)
{
    string result = "select
f.surname,f.name,f.middlename,w.name,c.second_name,s.name,a.second_name,f.rate,convert(VA
RCHAR(10),f.birthday,105) from dbo.Teachers as f left join dbo.Cairs as c on c.id=f.Cairs
left join dbo.Working_positions as w on w.id=f.Job_title left join dbo.Degrees as a on
a.id = f.degrees_id left join dbo.Titles as s on s.id = f.titles_id where f.terms_of_work
like '%Штатные совместители %' and f.Job_title =2 and "+between(startDate, endDate)+"
order by rate";
return NDataAccess.DataAccess.GetDataSet(@result, "Table1",
connectionString);
}
```

```
public static DataSet getProfessorsDoc(DateTime startDate, DateTime endDate)
{
```

Продолжение приложения А

```
    string result = "select
f.surname,f.name,f.middlename,w.name,c.second_name,s.name,a.second_name,f.rate,convert(VA
RCHAR(10),f.birthday,105) from dbo.Teachers as f left join dbo.Cairs as c on c.id=f.Cairs
left join dbo.Working_positions as w on w.id=f.Job_title left join dbo.Degrees as a on
a.id = f.degrees_id left join dbo.Titles as s on s.id = f.titles_id where f.terms_of_work
like '%штатные совместители %' and f.Job_title =2 and f.degrees_id>10 and f.degrees_id<16
and "+between(startDate, endDate)+" order by rate";
    return NDataAccess.DataAccess.GetDataSet(@result, "Table1",
connectionString);
}
    public static DataSet getProfessorsCand(DateTime startDate, DateTime endDate)
    {
        string result = "select
f.surname,f.name,f.middlename,w.name,c.second_name,s.name,a.second_name,f.rate,convert(VA
RCHAR(10),f.birthday,105) from dbo.Teachers as f left join dbo.Cairs as c on c.id=f.Cairs
left join dbo.Working_positions as w on w.id=f.Job_title left join dbo.Degrees as a on
a.id = f.degrees_id left join dbo.Titles as s on s.id = f.titles_id where
f.terms_of_work like '%штатные совместители %' and f.Job_title =2 and f.degrees_id>4 and
f.degrees_id<11 and f.titles_id=1 and "+between(startDate, endDate)+" order by rate";
        return NDataAccess.DataAccess.GetDataSet(@result, "Table1",
connectionString);
    }
    //Доценты
    public static DataSet getDocentsCandAndOrDocent(DateTime startDate, DateTime
endDate)
    {
        string result = "select
f.surname,f.name,f.middlename,w.name,c.second_name,s.name,a.second_name,f.rate,convert(VA
RCHAR(10),f.birthday,105) from dbo.Teachers as f left join dbo.Cairs as c on c.id=f.Cairs
left join dbo.Working_positions as w on w.id=f.Job_title left join dbo.Degrees as a on
a.id = f.degrees_id left join dbo.Titles as s on s.id = f.titles_id where f.terms_of_work
like '%штатные совместители %' and f.Job_title =3 and ((f.degrees_id>4 and
f.degrees_id<11) or (f.titles_id=2) or(f.degrees_id>4 and f.degrees_id<11 and
f.titles_id=2) ) and "+between(startDate, endDate)+" order by rate";
        return NDataAccess.DataAccess.GetDataSet(@result, "Table1",
connectionString);
    }
    public static DataSet getDocentsNull(DateTime startDate, DateTime endDate)
    {
        string result = "select
f.surname,f.name,f.middlename,w.name,c.second_name,s.name,a.second_name,f.rate,convert(VA
RCHAR(10),f.birthday,105) from dbo.Teachers as f left join dbo.Cairs as c on c.id=f.Cairs
left join dbo.Working_positions as w on w.id=f.Job_title left join dbo.Degrees as a on
a.id = f.degrees_id left join dbo.Titles as s on s.id = f.titles_id where f.terms_of_work
like '%штатные совместители %' and f.Job_title =3 and ((f.degrees_id=16) or
(f.titles_id=3) or(f.degrees_id=16 and f.titles_id=3) ) and "+between(startDate,
endDate)+" order by rate";
        return NDataAccess.DataAccess.GetDataSet(@result, "Table1",
connectionString);
    }
    //Старшие преп
    public static DataSet getSenior(DateTime startDate, DateTime endDate)
    {
        string result = "select
f.surname,f.name,f.middlename,w.name,c.second_name,s.name,a.second_name,f.rate,convert(VA
RCHAR(10),f.birthday,105) from dbo.Teachers as f left join dbo.Cairs as c on c.id=f.Cairs
left join dbo.Working_positions as w on w.id=f.Job_title left join dbo.Degrees as a on
a.id = f.degrees_id left join dbo.Titles as s on s.id = f.titles_id where f.terms_of_work
like '%штатные совместители %' and f.Job_title =4 and "+between(startDate, endDate)+"
order by rate";
```

```

        return NDataAccess.DataAccess.GetDataSet(@result, "Table1",
connectionString);

```

Продолжение приложения А

```

}

//Ассистенты
public static DataSet getAssist(DateTime startDate, DateTime endDate)
{
    string result = "select
f.surname,f.name,f.middlename,w.name,c.second_name,s.name,a.second_name,f.rate,convert(VA
RCHAR(10),f.birthday,105) from dbo.Teachers as f left join dbo.Cairs as c on c.id=f.Cairs
left join dbo.Working_positions as w on w.id=f.Job_title left join dbo.Degrees as a on
a.id = f.degrees_id left join dbo.Titles as s on s.id = f.titles_id where f.terms_of_work
like '%Штатные совместители %' and (f.Job_title =5 or f.Job_title=6) and
"+between(startDate, endDate)+" order by rate";
    return NDataAccess.DataAccess.GetDataSet(@result, "Table1",
connectionString);
}
} //          %Штатные совместители %
//Внеш
public static class ExternalStructure
{
    public static DataSet getDegreesAndTitle(DateTime startDate, DateTime
endDate)
    {
        string result = "SELECT    COUNT(*) as Vcero ,    (select Count(*) from
dbo.Teachers where degrees_id > 4  and degrees_id < 16 and terms_of_work like '%Сторонние
(внешние) совместители%' and "+between(startDate, endDate)+" ) as Ученые_степени,
(select Count(*) from dbo.Teachers where degrees_id between 4 and 10 and terms_of_work
like '%Сторонние (внешние) совместители%' and "+between(startDate, endDate)+" ) as
Кандидаты_наук,    (select Count(*) from dbo.Teachers where degrees_id > 10  and
degrees_id < 16 and terms_of_work like '%Сторонние (внешние) совместители%' and
"+between(startDate, endDate)+" ) as Доктора_наук,    (select Count(*) from dbo.Teachers
where titles_id > 0 and terms_of_work like '%Сторонние (внешние) совместители%' and
"+between(startDate, endDate)+" ) as Ученые_звания,    (select Count(*) from dbo.Teachers
where titles_id =1 and terms_of_work like '%Сторонние (внешние) совместители%' and
"+between(startDate, endDate)+" ) as Профессора,    (select Count(*) from dbo.Teachers
where titles_id = 2 and terms_of_work like '%Сторонние (внешние) совместители%' and
"+between(startDate, endDate)+" ) as Доценты,    (select COUNT(*) from dbo.Teachers where
gender like 'Жен' and terms_of_work like '%Сторонние (внешние) совместители%' and
"+between(startDate, endDate)+" )as Женщины FROM    Teachers where terms_of_work like
'%Сторонние (внешние) совместители%' and "+between(startDate, endDate)+" union all
SELECT    COUNT(*) as Vcero,    (select Count(*) from dbo.Teachers where degrees_id > 4
and degrees_id < 16 and Dekan_Faculties is not null and terms_of_work like '%Сторонние
(внешние) совместители%' and "+between(startDate, endDate)+" ) as Ученые_степени,
(select Count(*) from dbo.Teachers where degrees_id between 4 and 10 and Dekan_Faculties
is not null and terms_of_work like '%Сторонние (внешние) совместители%' and
"+between(startDate, endDate)+" ) as Кандидаты_наук,    (select Count(*) from dbo.Teachers
where degrees_id > 10  and degrees_id < 16 and Dekan_Faculties is not null and
terms_of_work like '%Сторонние (внешние) совместители%' and "+between(startDate,
endDate)+" ) as Доктора_наук,    (select Count(*) from dbo.Teachers where titles_id > 0
and Dekan_Faculties is not null and terms_of_work like '%Сторонние (внешние)
совместители%' and "+between(startDate, endDate)+" ) as Ученые_звания,    (select Count(*)
from dbo.Teachers where titles_id =1 and Dekan_Faculties is not null and terms_of_work
like '%Сторонние (внешние) совместители%' and "+between(startDate, endDate)+" ) as
Профессора,    (select Count(*) from dbo.Teachers where titles_id = 2 and Dekan_Faculties
is not null and terms_of_work like '%Сторонние (внешние) совместители%' and
"+between(startDate, endDate)+" ) as Доценты,    (select COUNT(*) from dbo.Teachers where
gender like 'Жен' and terms_of_work like '%Сторонние (внешние) совместители%' and
"+between(startDate, endDate)+" )as Женщины FROM    Teachers where Dekan_Faculties is not
null and terms_of_work like '%Сторонние (внешние) совместители%' and "+between(startDate,
endDate)+" union all SELECT    COUNT(*) as Vcero,    (select Count(*) from dbo.Teachers
where degrees_id > 4  and degrees_id < 16 and Job_title = 1 and terms_of_work like
'%Сторонние (внешние) совместители%' and "+between(startDate, endDate)+" ) as

```


Ученые_степени, (select Count(*) from dbo.Teachers where degrees_id between 4 and 10 and Job title = 1 and terms of work like '%Сторонние (внешние) совместители%' and

Продолжение приложения А

[illegible]

```
" + between(startDate, endDate) + ") as Ученые_звания, (select Count(*) from dbo.Teachers
where titles_id = 1 and Job_title = 4 and terms_of_work like '%Сторонние (внешние)
```

Продолжение приложения А

```
совместители%' and " + between(startDate, endDate) + ") as Профессора, (select Count(*)
from dbo.Teachers where titles_id = 2 and Job_title = 4 and terms_of_work like
'%Сторонние (внешние) совместители%' and " + between(startDate, endDate) + ") as Доценты,
(select COUNT(*) from dbo.Teachers where gender like 'Жен' and Job_title = 4 and
terms_of_work like '%Сторонние (внешние) совместители%' and " + between(startDate,
endDate) + ") as Женщины FROM Teachers where Job_title = 4 and terms_of_work like
'%Сторонние (внешние) совместители%' and " + between(startDate, endDate) + " union all SELECT
COUNT(*) as Всего, (select Count(*) from dbo.Teachers where degrees_id > 4 and
degrees_id < 16 and (Job_title = 5 or Job_title = 6) and terms_of_work like '%Сторонние
(внешние) совместители%' and " + between(startDate, endDate) + ") as Ученые_степени,
(select Count(*) from dbo.Teachers where degrees_id between 4 and 10 and (Job_title = 5
or Job_title = 6) and terms_of_work like '%Сторонние (внешние) совместители%' and
" + between(startDate, endDate) + ") as Кандидаты_наук, (select Count(*) from dbo.Teachers
where degrees_id > 10 and degrees_id < 16 and (Job_title = 5 or Job_title = 6) and
terms_of_work like '%Сторонние (внешние) совместители%' and " + between(startDate,
endDate) + ") as Доктора_наук, (select Count(*) from dbo.Teachers where titles_id > 0
and (Job_title = 5 or Job_title = 6) and terms_of_work like '%Сторонние (внешние)
совместители%' and " + between(startDate, endDate) + ") as Ученые_звания, (select Count(*)
from dbo.Teachers where titles_id = 1 and (Job_title = 5 or Job_title = 6) and
terms_of_work like '%Сторонние (внешние) совместители%' and " + between(startDate,
endDate) + ") as Профессора, (select Count(*) from dbo.Teachers where titles_id = 2 and
(Job_title = 5 or Job_title = 6) and terms_of_work like '%Сторонние (внешние)
совместители%' and " + between(startDate, endDate) + ") as Доценты, (select COUNT(*) from
dbo.Teachers where gender like 'Жен' and (Job_title = 5 or Job_title = 6) and
terms_of_work like '%Сторонние (внешние) совместители%' and " + between(startDate,
endDate) + ") as Женщины FROM Teachers where (Job_title = 5 or Job_title = 6) and
terms_of_work like '%Сторонние (внешние) совместители%' and " + between(startDate,
endDate) + """;
```

```
return NDataAccess.DataAccess.GetDataSet(@result, "Table1",
connectionString);
}
public static DataSet getRate(DateTime startDate, DateTime endDate)
{
string result = "SELECT count(*), (select Count(*) from
dbo.Teachers where rate = 0.1 and terms_of_work like '%Сторонние (внешние) совместители%'
and " + between(startDate, endDate) + ") as '0,1', (select Count(*) from dbo.Teachers
where rate = 0.2 and terms_of_work like '%Сторонние (внешние) совместители%' and
" + between(startDate, endDate) + ") as '0,2', (select Count(*) from dbo.Teachers where
rate = 0.3 and terms_of_work like '%Сторонние (внешние) совместители%' and
" + between(startDate, endDate) + ") as '0,3', (select Count(*) from dbo.Teachers where
rate = 0.4 and terms_of_work like '%Сторонние (внешние) совместители%' and
" + between(startDate, endDate) + ") as '0,4', (select Count(*) from dbo.Teachers where
rate = 0.5 and terms_of_work like '%Сторонние (внешние) совместители%' and
" + between(startDate, endDate) + ") as '0,5', (select Count(*) from dbo.Teachers where
rate = 0.6 and terms_of_work like '%Сторонние (внешние) совместители%' and
" + between(startDate, endDate) + ") as '0,6', (select Count(*) from dbo.Teachers where
rate = 0.7 and terms_of_work like '%Сторонние (внешние) совместители%' and
" + between(startDate, endDate) + ") as '0,7', (select Count(*) from dbo.Teachers where
rate = 0.8 and terms_of_work like '%Сторонние (внешние) совместители%' and
" + between(startDate, endDate) + ") as '0,8', (select Count(*) from dbo.Teachers where
rate = 0.9 and terms_of_work like '%Сторонние (внешние) совместители%' and
" + between(startDate, endDate) + ") as '0,9', (select Count(*) from dbo.Teachers where
rate > 0.9 and terms_of_work like '%Сторонние (внешние) совместители%' and
" + between(startDate, endDate) + ") as '1' FROM Teachers union all SELECT count(*),
(select Count(*) from dbo.Teachers where rate = 0.1 and terms_of_work like '%Сторонние
(внешние) совместители%' and " + between(startDate, endDate) + " and Dekan_Faculties is not
null) as '0,1', (select Count(*) from dbo.Teachers where rate = 0.2 and terms_of_work
like '%Сторонние (внешние) совместители%' and " + between(startDate, endDate) + " and
Dekan_Faculties is not null) as '0,2', (select Count(*) from dbo.Teachers where rate =
0.3 and terms_of_work like '%Сторонние (внешние) совместители%' and " + between(startDate,
```

```
endDate)+ " and Dekan_Faculties is not null) as '0,3', (select Count(*) from
dbo.Teachers where rate = 0.4 and terms of work like '%Сторонние (внешние) совместители%'
```

Продолжение приложения А

[illegible]

```
dbo.Teachers where rate = 0.1 and terms_of_work like '%Сторонние (внешние) совместители%'
and "+between(startDate, endDate)+" and Job title = 3) as '0,1', (select Count(*)
```

Продолжение приложения А

[illegible]


```
Job_title = 6)) as '0,8', (select Count(*) from dbo.Teachers where rate = 0.9 and
terms_of_work like '%Сторонние (внешние) совместители%' and "+between(startDate,
```

Продолжение приложения А

```
endDate)+" and (Job_title = 5 or Job_title = 6)) as '0,9', (select Count(*) from
dbo.Teachers where rate >0.9 and terms_of_work like '%Сторонние (внешние) совместители%'
and "+between(startDate, endDate)+" and (Job_title = 5 or Job_title = 6)) as '1' FROM
Teachers where (Job_title = 5 or Job_title = 6) and terms_of_work like '%Сторонние
(внешние) совместители%' and "+between(startDate, endDate)++";
return NDataAccess.DataAccess.GetDataSet(@result, "Table1",
connectionString);
}////EXT
public static DataSet getQualification(DateTime startDate, DateTime endDate)
{
    string[] date = getPreviousDate();
    string result = "SELECT COUNT(Training_dates) as Bcero FROM Teachers
where terms_of_work like '%Сторонние (внешние) совместители%' and (Training_dates between
" + date[0] + " and " + date[1] + ") union all SELECT COUNT(Training_dates) FROM Teachers
where Dekan_Faculties is not null and terms_of_work like '%Сторонние (внешние)
совместители%' and "+between(startDate, endDate)+" and (Training_dates between " +
date[0] + " and " + date[1] + ") union all SELECT COUNT(Training_dates) as Bcero FROM
Teachers where Job_title = 1 and terms_of_work like '%Сторонние (внешние) совместители%'
and "+between(startDate, endDate)+"and (Training_dates between " + date[0] + " and " +
date[1] + ") union all SELECT COUNT(Training_dates) as Bcero FROM Teachers where
Job_title = 2 and terms_of_work like '%Сторонние (внешние) совместители%' and
"+between(startDate, endDate)+" and (Training_dates between " + date[0] + " and " +
date[1] + ") union all SELECT COUNT(Training_dates) as Bcero FROM Teachers where
Job_title = 3 and terms_of_work like '%Сторонние (внешние) совместители%' and
"+between(startDate, endDate)+"and (Training_dates between " + date[0] + " and " +
date[1] + ") union all SELECT COUNT(Training_dates) as Bcero FROM Teachers where
Job_title = 4 and terms_of_work like '%Сторонние (внешние) совместители%' and
"+between(startDate, endDate)+"and (Training_dates between " + date[0] + " and " +
date[1] + ") union all SELECT COUNT(Training_dates) as Bcero FROM Teachers where
(Job_title = 5 or Job_title = 6) and terms_of_work like '%Сторонние (внешние)
совместители%' and "+between(startDate, endDate)+"and (Training_dates between " + date[0]
+ " and " + date[1] + ")";
return NDataAccess.DataAccess.GetDataSet(@result, "Table1",
connectionString);
}

//Профф
public static DataSet getProfessors(DateTime startDate, DateTime endDate)
{
    string result = "select
f.surname,f.name,f.middlename,w.name,c.second_name,s.name,a.second_name,f.rate,convert(VA
RCHAR(10),f.birthday,105) from dbo.Teachers as f left join dbo.Cairs as c on c.id=f.Cairs
left join dbo.Working_positions as w on w.id=f.Job_title left join dbo.Degrees as a on
a.id = f.degrees_id left join dbo.Titles as s on s.id = f.titles_id where f.terms_of_work
like '%Сторонние (внешние) совместители%' and f.Job_title =2 and "+between(startDate,
endDate)+" order by rate";
return NDataAccess.DataAccess.GetDataSet(@result, "Table1",
connectionString);
}
public static DataSet getProfessorsDoc(DateTime startDate, DateTime endDate)
{
    string result = "select
f.surname,f.name,f.middlename,w.name,c.second_name,s.name,a.second_name,f.rate,convert(VA
RCHAR(10),f.birthday,105) from dbo.Teachers as f left join dbo.Cairs as c on c.id=f.Cairs
left join dbo.Working_positions as w on w.id=f.Job_title left join dbo.Degrees as a on
a.id = f.degrees_id left join dbo.Titles as s on s.id = f.titles_id where f.terms_of_work
like '%Сторонние (внешние) совместители%' and f.Job_title =2 and f.degrees_id>10 and
f.degrees_id<16 and "+between(startDate, endDate)+" order by rate";
```

```

        return NDataAccess.DataAccess.GetDataSet(@result, "Table1",
connectionString);
    }

```

Продолжение приложения А

```

    public static DataSet getProfessorsCand(DateTime startDate, DateTime endDate)
    {
        string result = "select
f.surname,f.name,f.middlename,w.name,c.second_name,s.name,a.second_name,f.rate,convert(VA
RCHAR(10),f.birthday,105) from dbo.Teachers as f left join dbo.Cairs as c on c.id=f.Cairs
left join dbo.Working_positions as w on w.id=f.Job_title left join dbo.Degrees as a on
a.id = f.degrees_id left join dbo.Titles as s on s.id = f.titles_id where
f.terms_of_work like '%Сторонние (внешние) совместители%' and f.Job_title =2 and
f.degrees_id>4 and f.degrees_id<11 and f.titles_id=1 and "+between(startDate, endDate)+"
order by rate";
        return NDataAccess.DataAccess.GetDataSet(@result, "Table1",
connectionString);
    }
    //Доценты
    public static DataSet getDocentsCandAndOrDocent(DateTime startDate, DateTime
endDate)
    {
        string result = "select
f.surname,f.name,f.middlename,w.name,c.second_name,s.name,a.second_name,f.rate,convert(VA
RCHAR(10),f.birthday,105) from dbo.Teachers as f left join dbo.Cairs as c on c.id=f.Cairs
left join dbo.Working_positions as w on w.id=f.Job_title left join dbo.Degrees as a on
a.id = f.degrees_id left join dbo.Titles as s on s.id = f.titles_id where f.terms_of_work
like '%Сторонние (внешние) совместители%' and f.Job_title =3 and ((f.degrees_id>4 and
f.degrees_id<11) or (f.titles_id=2) or(f.degrees_id>4 and f.degrees_id<11 and
f.titles_id=2) ) and "+between(startDate, endDate)+" order by rate";
        return NDataAccess.DataAccess.GetDataSet(@result, "Table1",
connectionString);
    }
    public static DataSet getDocentsNull(DateTime startDate, DateTime endDate)
    {
        string result = "select
f.surname,f.name,f.middlename,w.name,c.second_name,s.name,a.second_name,f.rate,convert(VA
RCHAR(10),f.birthday,105) from dbo.Teachers as f left join dbo.Cairs as c on c.id=f.Cairs
left join dbo.Working_positions as w on w.id=f.Job_title left join dbo.Degrees as a on
a.id = f.degrees_id left join dbo.Titles as s on s.id = f.titles_id where f.terms_of_work
like '%Сторонние (внешние) совместители%' and f.Job_title =3 and ((f.degrees_id=16) or
(f.titles_id=3) or(f.degrees_id=16 and f.titles_id=3) ) and "+between(startDate,
endDate)+" order by rate";
        return NDataAccess.DataAccess.GetDataSet(@result, "Table1",
connectionString);
    }
    //Старшие преп
    public static DataSet getSenior(DateTime startDate, DateTime endDate)
    {
        string result = "select
f.surname,f.name,f.middlename,w.name,c.second_name,s.name,a.second_name,f.rate,convert(VA
RCHAR(10),f.birthday,105) from dbo.Teachers as f left join dbo.Cairs as c on c.id=f.Cairs
left join dbo.Working_positions as w on w.id=f.Job_title left join dbo.Degrees as a on
a.id = f.degrees_id left join dbo.Titles as s on s.id = f.titles_id where f.terms_of_work
like '%Сторонние (внешние) совместители%' and f.Job_title =4 and "+between(startDate,
endDate)+" order by rate";
        return NDataAccess.DataAccess.GetDataSet(@result, "Table1",
connectionString);
    }
    //Ассистенты
    public static DataSet getAssist(DateTime startDate, DateTime endDate)
    {

```

```

        string result = "select
f.surname,f.name,f.middlename,w.name,c.second_name,s.name,a.second_name,f.rate,convert(VA
RCHAR(10),f.birthday,105) from dbo.Teachers as f left join dbo.Cairs as c on c.id=f.Cairs
left join dbo.Working_positions as w on w.id=f.Job_title left join dbo.Degrees as a on

```

Продолжение приложения А

```

a.id = f.degrees_id left join dboTitles as s on s.id = f.titles_id where f.terms_of_work
like '%Сторонние (внешние) совместители%' and (f.Job_title =5 or f.Job_title=6) and
"+between(startDate, endDate)+" order by rate";
        return NDataAccess.DataAccess.GetDataSet(@result, "Table1",
connectionString);
    }
}

public static class Dissertation
{
    public static DataSet Show(DateTime startDate, DateTime endDate)
    {
        string result = "select f.id,w.name as Должность,f.surname as
Фамилия,f.name as Имя,f.middlename as Отчество,c.second_name as 'Кафедра',d.name as
'ученое звание',titles_date as 'Дата получения ученого звания',a.second_name as 'ученая
степень',f.degress_date as 'Дата получения ученой степени',terms_of_work as 'Условия
привлечения к труд. деят.' from Teachers as f left join Working_positions as w on
w.id=f.Job_title left join Titles as d on d.id=f.titles_id left join Degrees as a on
a.id=f.degrees_id left join Cairs as c on c.id = f.Cairs where " + between(startDate,
endDate) + " ";//where DATEDIFF(MONTH,degress_date,'" + date + "')=0 ";//and
f.Dekan_Faculties is null
        return NDataAccess.DataAccess.GetDataSet(@result, "Table1",
connectionString);
    }
}

public static class Record
{
    public static void Add(string result)
    {
        NDataAccess.DataAccess.ExecuteNonQuery2(result, connectionString);
    }
    public static void Add(int id, bool del, DateTime experienceDate)
    {
        Object[] array = GetDataTeachers(id).Tables[0].Rows[0].ItemArray;//Объект
строки запроса
        string[] cairsandjob = new string[2], date = new string[2];//массивы для
даты, названия кафедры и должности
        if (array[6].ToString() == "3") date[0] = "null";//Если нету ученой
степени то даты её получения тоже нету
        else date[0] = "'" + array[7].ToString() + "'";
        if (array[8].ToString() == "16") date[1] = "null";//Если нету ученого
звания то даты его получения тоже нету
        else date[1] = "'" + array[9].ToString() + "'";
        string result = "";
        if (del)//Удаляем?
        {
            string[] termworks = array[10].ToString().Split('-');//разбиваем
строку для получения из неё данных и изменения самой строки
            if (array[13].ToString() == "")//Если не декан
                result = "INSERT INTO dbo.Records
(id,Cairs,Job_title,surname,name,middlename,titles_id,titles_date,degrees_id,degress_date
,terms_of_work,date)" +
                "VALUES(" + (int.Parse(array[0]) + 1) + "," + (int)array[1] + "," +
+ (int)array[2] + "','" + array[3] + "','" + array[4] + "','" + array[5] + "','" +
(int)array[6] + "','" + date[0] + "','" + (int)array[8] + "','" + date[1] + "','" + termworks[0]
+ "-" + termworks[1] + "-Уволился-" + experienceDate + "'),'," + experienceDate + "');"

```

```

        else result = "INSERT INTO dbo.Records
(id,surname,name,middlename,titles_id,titles_date,degrees_id,degress_date,terms_of_work,d
ate)" +
        "VALUES(" + (int.Parse(Count()) + 1) + "," + array[3] + "','" +
array[4] + "','" + array[5] + "','" + (int)array[6] + "," + date[0] + "," + (int)array[8]

```

Продолжение приложения А

```

+ "," + date[1] + "','" + termworks[0] + "-" + termworks[1] + "-Уволился-(" +
experienceDate + ")','" + experienceDate + "')";
    }
    else
    { //нет
        if (array[13].ToString() == "") //Если не декан
            result = "INSERT INTO dbo.Records
(id,Cairs,Job_title,surname,name,middlename,titles_id,titles_date,degrees_id,degress_date
,terms_of_work,date)" +
            "VALUES(" + (int.Parse(Count()) + 1) + "," + (int)array[1] + "," +
+ (int)array[2] + "','" + array[3] + "','" + array[4] + "','" + array[5] + "','" +
(int)array[6] + "," + date[0] + "," + (int)array[8] + "," + date[1] + "','" + array[10] +
+ "','" + experienceDate + "')";
        else result = "INSERT INTO dbo.Records
(id,surname,name,middlename,titles_id,titles_date,degrees_id,degress_date,terms_of_work,d
ate)" +
        "VALUES(" + (int.Parse(Count()) + 1) + "','" + array[3] + "','" +
array[4] + "','" + array[5] + "','" + (int)array[6] + "," + date[0] + "," + (int)array[8]
+ "," + date[1] + "','" + array[10] + "','" + experienceDate + "')";
    }
    NDataAccess.DataAccess.ExecuteNonQuery2(result, connectionString);
}
public static string Count()
{
    ds = NDataAccess.DataAccess.GetDataSet(@"select max(id) as count from
dbo.Records", "Table1", connectionString);
    string result = ds.Tables[0].Rows[0].ItemArray[0].ToString();
    if (result == "") result = "0";
    return (result);
}
public static DataSet Show(DateTime startDate, DateTime endDate)
{
    string result = "select f.id,c.second_name as 'Кафедра',w.name as
Должность,f.surname as Фамилия,f.name as Имя,f.middlename as Отчество,d.name as 'ученое
звание',titles_date as 'Дата получения ученого звания',a.second_name as 'ученая
степень',f.degress_date as 'Дата получения ученой степени',terms_of_work as 'Условия
привлечения к труд. деят.' from Records as f left join Working_positions as w on
w.id=f.Job_title left join Titles as d on d.id=f.titles_id left join Degrees as a on
a.id=f.degrees_id left join Cairs as c on c.id = f.Cairs where date > '" +
ReverseDateTime(startDate) + "'"; //where DATEDIFF(MONTH,date,GETDATE())=0 ";
    return NDataAccess.DataAccess.GetDataSet(@result, "Table1",
connectionString);
}
public static DataSet GetDataTeachers(int id)
{
    string result = "select
id,Cairs,Job_title,surname,name,middlename,titles_id,titles_date,degrees_id,degress_date,
terms_of_work,rate,experience_date,Dekan_Faculties,education_date from Teachers where id
= " + id ;
    return NDataAccess.DataAccess.GetDataSet(@result, "Table1",
connectionString);
}
}

public static class DegreesStructure
{

```

```

        private static string SplitAndCheck(double value)//Метод для коррекции
введённого значения ставки и даты
    {
        string[] rates;
        string result = ""; ;

```

Продолжение приложения А

```

        rates = value.ToString().Split(new Char[] { '.', ',' });//Разбираем строку по
разделителю ('.'или',)
        if (rates.Length > 1)
        {
            result = (rates[0] + "." + rates[1]);//если стоит разделитель, значит
собираем строку с точкой для запроса в базу (с запятой запрос не будет работать, поэтому
нужна точка)
        }
        else
        {
            result = rates[0];//если без разделителя , тогда так и оставляем
        }
        return result;
    }

```

```

    private static string queryWithTitles(DateTime startDate, DateTime endDate,
string emploe)
    {

```

```

        StringBuilder sb = new StringBuilder();

        sb.Append("SELECT COUNT(*) as Bcero");
        double a = 0;
        for (int i = 1; i < 10; i++)
        {
            a += 0.1;
            sb.Append(String.Format(",(select Count(*) from dbo.Teachers where
rate = "+SplitAndCheck(a)+" and terms_of_work like '{0}' and {1} ) as ' " +
a.ToString("0.0") + "' ", emploe, between(startDate, endDate)));
        }
        sb.Append(String.Format(",(select Count(*) from dbo.Teachers where rate
>0.9 and terms_of_work like '{0}' and {1} ) as '1' ",emploe, between(startDate,
endDate)));
        sb.Append(String.Format(" from dbo.Teachers where terms_of_work like
'{0}' and {1}", emploe, between(startDate, endDate)));

        return sb.ToString();
    }

```

```

    private static string queryWithTitles(DateTime startDate, DateTime endDate,
string emploe,string title)
    {

```

```

        StringBuilder sb = new StringBuilder();

        sb.Append("SELECT COUNT(*) as Bcero");
        double a = 0;
        for (int i = 1; i < 10; i++)
        {
            a += 0.1;
            sb.Append(String.Format(",(select Count(*) from dbo.Teachers where
rate = " + SplitAndCheck(a) + " and terms_of_work like '{0}' and {1} and {2} ) as ' " +
a.ToString("0.0") + "' ", emploe, title, between(startDate, endDate)));
        }
        sb.Append(String.Format(",(select Count(*) from dbo.Teachers where rate
>0.9 and terms_of_work like '{0}' and {1} and {2} ) as '1' ", emploe,title,
between(startDate, endDate)));
    }

```

```

        sb.Append(String.Format(" from dbo.Teachers where terms_of_work like
'{0}' and {1} and {2}", emploe,title, between(startDate, endDate)));

        return sb.ToString();
    }

```

Продолжение приложения А

```

public static DataSet getRate(DateTime startDate, DateTime endDate)
{
    StringBuilder sb = new StringBuilder();

    string emploe = "%Штатный сотрудник%";
    sb.Append(queryWithTitles(startDate, endDate, emploe));
    sb.Append(" union all ");
    sb.Append(queryWithTitles(startDate, endDate, emploe, "(degrees_id
between 10 and 16)"));
    sb.Append(" union all ");
    sb.Append(queryWithTitles(startDate, endDate, emploe, "(degrees_id
between 4 and 11)"));
    sb.Append(" union all ");

    emploe = "%Сторонние (внешние) совместители%";
    sb.Append(queryWithTitles(startDate, endDate, emploe));
    sb.Append(" union all ");
    sb.Append(queryWithTitles(startDate, endDate, emploe, "(degrees_id
between 10 and 16)"));
    sb.Append(" union all ");
    sb.Append(queryWithTitles(startDate, endDate, emploe, "(degrees_id
between 4 and 11)"));
    sb.Append(" union all ");

    emploe = "%Штатные совместители%";
    sb.Append(queryWithTitles(startDate, endDate, emploe));
    sb.Append(" union all ");
    sb.Append(queryWithTitles(startDate, endDate, emploe, "(degrees_id
between 10 and 16)"));
    sb.Append(" union all ");
    sb.Append(queryWithTitles(startDate, endDate, emploe, "(degrees_id
between 4 and 11)"));

    return NDataAccess.DataAccess.GetDataSet(sb.ToString(), "Table1",
connectionString);
}
} //Сведения об ученых степенях профессорско-преподавательского состава и научных
работников

public static class ExperienceStructure
{
    private static string getExperience(string a,string date)
    {
        return " DATEDIFF(YY," + date + ",CAST(Getdate() AS Date)) " + a + " ";
    }
    private static string getExperience(int a, int b,string date)
    {
        return " DATEDIFF(YY," + date + ",CAST(Getdate() AS Date))>" + (a - 1) +
" and DATEDIFF(YY," + date + ",CAST(Getdate() AS Date))< " + b + " ";
    }
    private static string queryWithTitles(DateTime startDate, DateTime
endDate,string date)
    {

```

```

        StringBuilder sb = new StringBuilder();

        sb.Append("SELECT COUNT(*) as Bcero");
        sb.Append(String.Format(",(select COUNT(*) from dbo.Teachers where {0} and {1}) as '3'", between(startDate, endDate), getExperience("< 3", date)));

```

Продолжение приложения А

```

        sb.Append(String.Format(",(select COUNT(*) from dbo.Teachers where {0} and {1}) as '3-5'", between(startDate, endDate), getExperience(3, 5, date)));
        sb.Append(String.Format(",(select COUNT(*) from dbo.Teachers where {0} and {1}) as '5-10'", between(startDate, endDate), getExperience(5, 10, date)));
        sb.Append(String.Format(",(select COUNT(*) from dbo.Teachers where {0} and {1}) as '10-15'", between(startDate, endDate), getExperience(10, 15, date)));
        sb.Append(String.Format(",(select COUNT(*) from dbo.Teachers where {0} and {1}) as '15-20'", between(startDate, endDate), getExperience(15, 20, date)));
        sb.Append(String.Format(",(select COUNT(*) from dbo.Teachers where {0} and {1}) as '20+'", between(startDate, endDate), getExperience("> 19", date)));
        sb.Append(String.Format(" from dbo.Teachers where {0} ", between(startDate, endDate)));

        return sb.ToString();
    }

    private static string queryWithTitles(DateTime startDate, DateTime endDate, string title, string date)
    {
        StringBuilder sb = new StringBuilder();

        sb.Append("SELECT COUNT(*) as Bcero");
        sb.Append(String.Format(",(select COUNT(*) from dbo.Teachers where {0} and {1} and {2}) as '3'", between(startDate, endDate), getExperience("< 3", date), title));
        sb.Append(String.Format(",(select COUNT(*) from dbo.Teachers where {0} and {1} and {2}) as '3-5'", between(startDate, endDate), getExperience(3, 5, date), title));
        sb.Append(String.Format(",(select COUNT(*) from dbo.Teachers where {0} and {1} and {2}) as '5-10'", between(startDate, endDate), getExperience(5, 10, date), title));
        sb.Append(String.Format(",(select COUNT(*) from dbo.Teachers where {0} and {1} and {2}) as '10-15'", between(startDate, endDate), getExperience(10, 15, date), title));
        sb.Append(String.Format(",(select COUNT(*) from dbo.Teachers where {0} and {1} and {2}) as '15-20'", between(startDate, endDate), getExperience(15, 20, date), title));
        sb.Append(String.Format(",(select COUNT(*) from dbo.Teachers where {0} and {1} and {2}) as '20+'", between(startDate, endDate), getExperience("> 19", date), title));
        sb.Append(String.Format(" from dbo.Teachers where {0} and {1}", between(startDate, endDate), title));

        return sb.ToString();
    }

    public static DataSet getExperience(DateTime startDate, DateTime endDate)
    {
        StringBuilder sb = new StringBuilder();
        sb.Append(queryWithTitles(startDate, endDate, "experience_date"));
        sb.Append(" union all ");
        sb.Append(queryWithTitles(startDate, endDate, "Dekan_Faculties is not null ", "experience_date"));
        for (int i = 1; i < 5; i++)
        {
            sb.Append(" union all ");

```



```

        sb.Append(queryWithTitles(startDate, endDate, "Job_title = " + i,
"experience_date"));
    }
    sb.Append(" union all ");
    sb.Append(queryWithTitles(startDate, endDate, "(Job_title = 5 or
Job_title = 6)", "experience_date"));

```

Продолжение приложения А

```

        return NDataAccess.DataAccess.GetDataSet(@sb.ToString(), "Table1",
connectionString);
    }
    public static DataSet getTotalExperience(DateTime startDate, DateTime
endDate)
    {
        StringBuilder sb = new StringBuilder();
        sb.Append(queryWithTitles(startDate, endDate, "total_experience_date"));
        sb.Append(" union all ");
        sb.Append(queryWithTitles(startDate, endDate, " Dekan_Faculties is not
null ", "total_experience_date"));
        for (int i = 1; i < 5; i++)
        {
            sb.Append(" union all ");
            sb.Append(queryWithTitles(startDate, endDate, "Job_title = " + i,
"total_experience_date"));
        }
        sb.Append(" union all ");
        sb.Append(queryWithTitles(startDate, endDate, "(Job_title = 5 or
Job_title = 6)", "total_experience_date"));
        return NDataAccess.DataAccess.GetDataSet(@sb.ToString(), "Table1",
connectionString);
    }
}
//Распределение персонала по стажу работы

public static class AgeStructure
{
    private static string birthdayFirst(int YY)
    {
        return " DATEDIFF(YY,birthday,CAST(Getdate() AS Date))< "+YY;
    }

    private static string birthdayBetween(int YY)
    {
        return String.Format(" DATEDIFF(YY,birthday,CAST(Getdate() AS Date))>{0}
and DATEDIFF(YY,birthday,CAST(Getdate() AS Date))< {1}", YY - 1, YY + 5);
    }
    private static string birthdayBetweenLast(int YY)
    {
        return String.Format(" DATEDIFF(YY,birthday,CAST(Getdate() AS
Date))>{0}", YY - 1);
    }

    private static string allTitles(DateTime startDate, DateTime endDate)
    {
        StringBuilder sb = new StringBuilder();

        sb.Append("SELECT COUNT(*) as Bcero");
        sb.Append(String.Format(",(select COUNT(*) from dbo.Teachers where {0}
and {1}) as 'менее 25'", birthdayFirst(25), between(startDate, endDate)));
        sb.Append(String.Format(",(select COUNT(*) from dbo.Teachers where {0}
and {1}) as 'менее 25-29'", birthdayBetween(25), between(startDate, endDate)));
        sb.Append(String.Format(",(select COUNT(*) from dbo.Teachers where {0}
and {1}) as 'менее 30-34'", birthdayBetween(30), between(startDate, endDate)));
    }
}

```



```

        sb.Append(String.Format("(select COUNT(*) from dbo.Teachers where {0}
and {1}) as 'менее 35-39'", birthdayBetween(35), between(startDate, endDate)));
        sb.Append(String.Format("(select COUNT(*) from dbo.Teachers where {0}
and {1}) as 'менее 40-44'", birthdayBetween(40), between(startDate, endDate)));
        sb.Append(String.Format("(select COUNT(*) from dbo.Teachers where {0}
and {1}) as 'менее 45-49'", birthdayBetween(45), between(startDate, endDate)));

```

Продолжение приложения А

```

        sb.Append(String.Format("(select COUNT(*) from dbo.Teachers where {0} and {1}) as
'менее 50-54'", birthdayBetween(50), between(startDate, endDate)));
        sb.Append(String.Format("(select COUNT(*) from dbo.Teachers where {0}
and {1}) as 'менее 55-59'", birthdayBetween(55), between(startDate, endDate)));
        sb.Append(String.Format("(select COUNT(*) from dbo.Teachers where {0}
and {1}) as 'менее 60-64'", birthdayBetween(60), between(startDate, endDate)));
        sb.Append(String.Format("(select COUNT(*) from dbo.Teachers where {0}
and {1}) as '65 и более'", birthdayBetweenLast(65), between(startDate, endDate)));
        sb.Append(String.Format(" from dbo.Teachers where {0}",
between(startDate, endDate)));

        return sb.ToString();
    }

    private static string allTitlesWoomen(DateTime startDate, DateTime endDate)
    {
        StringBuilder sb = new StringBuilder();

        sb.Append("SELECT COUNT(*) as Bcero");
        sb.Append(String.Format("(select COUNT(*) from dbo.Teachers where {0}
and {1} and gender like 'Жен') as 'менее 25'", birthdayFirst(25), between(startDate,
endDate)));
        sb.Append(String.Format("(select COUNT(*) from dbo.Teachers where {0}
and {1} and gender like 'Жен') as 'менее 25-29'", birthdayBetween(25),
between(startDate, endDate)));
        sb.Append(String.Format("(select COUNT(*) from dbo.Teachers where {0}
and {1} and gender like 'Жен') as 'менее 30-34'", birthdayBetween(30),
between(startDate, endDate)));
        sb.Append(String.Format("(select COUNT(*) from dbo.Teachers where {0}
and {1} and gender like 'Жен') as 'менее 35-39'", birthdayBetween(35),
between(startDate, endDate)));
        sb.Append(String.Format("(select COUNT(*) from dbo.Teachers where {0}
and {1} and gender like 'Жен') as 'менее 40-44'", birthdayBetween(40),
between(startDate, endDate)));
        sb.Append(String.Format("(select COUNT(*) from dbo.Teachers where {0}
and {1} and gender like 'Жен') as 'менее 45-49'", birthdayBetween(45),
between(startDate, endDate)));
        sb.Append(String.Format("(select COUNT(*) from dbo.Teachers where {0}
and {1} and gender like 'Жен') as 'менее 50-54'", birthdayBetween(50),
between(startDate, endDate)));
        sb.Append(String.Format("(select COUNT(*) from dbo.Teachers where {0}
and {1} and gender like 'Жен') as 'менее 55-59'", birthdayBetween(55),
between(startDate, endDate)));
        sb.Append(String.Format("(select COUNT(*) from dbo.Teachers where {0}
and {1} and gender like 'Жен') as 'менее 60-64'", birthdayBetween(60),
between(startDate, endDate)));
        sb.Append(String.Format("(select COUNT(*) from dbo.Teachers where {0}
and {1} and gender like 'Жен') as '65 и более'", birthdayBetweenLast(65),
between(startDate, endDate)));
        sb.Append(String.Format(" from dbo.Teachers where {0}",
between(startDate, endDate)));

        return sb.ToString();
    }
}

```

```

        private static string queryWithTitles(DateTime startDate, DateTime
endDate, string title)
        {
            StringBuilder sb = new StringBuilder();

            sb.Append("SELECT COUNT(*) as Bcero");

```

Продолжение приложения А

```

            sb.Append(String.Format(",(select COUNT(*) from dbo.Teachers where {0} and {1} and {2})
as 'менее 25'", birthdayFirst(25), between(startDate, endDate), title));
            sb.Append(String.Format(",(select COUNT(*) from dbo.Teachers where {0}
and {1} and {2}) as 'менее 25-29'", birthdayBetween(25), between(startDate, endDate),
title));
            sb.Append(String.Format(",(select COUNT(*) from dbo.Teachers where {0}
and {1} and {2}) as 'менее 30-34'", birthdayBetween(30), between(startDate, endDate),
title));
            sb.Append(String.Format(",(select COUNT(*) from dbo.Teachers where {0}
and {1} and {2}) as 'менее 35-39'", birthdayBetween(35), between(startDate, endDate),
title));
            sb.Append(String.Format(",(select COUNT(*) from dbo.Teachers where {0}
and {1} and {2}) as 'менее 40-44'", birthdayBetween(40), between(startDate, endDate),
title));
            sb.Append(String.Format(",(select COUNT(*) from dbo.Teachers where {0}
and {1} and {2}) as 'менее 45-49'", birthdayBetween(45), between(startDate, endDate),
title));
            sb.Append(String.Format(",(select COUNT(*) from dbo.Teachers where {0}
and {1} and {2}) as 'менее 50-54'", birthdayBetween(50), between(startDate, endDate),
title));
            sb.Append(String.Format(",(select COUNT(*) from dbo.Teachers where {0}
and {1} and {2}) as 'менее 55-59'", birthdayBetween(55), between(startDate, endDate),
title));
            sb.Append(String.Format(",(select COUNT(*) from dbo.Teachers where {0}
and {1} and {2}) as 'менее 60-64'", birthdayBetween(60), between(startDate, endDate),
title));
            sb.Append(String.Format(",(select COUNT(*) from dbo.Teachers where {0}
and {1} and {2}) as '65 и более'", birthdayBetweenLast(65), between(startDate, endDate),
title));
            sb.Append(String.Format(" from dbo.Teachers where {0} and {1}",
between(startDate, endDate), title));

            return sb.ToString();
        }

        public static DataSet getAge(DateTime startDate, DateTime endDate)
        {
            StringBuilder sb = new StringBuilder();

            sb.Append(allTitles(startDate, endDate));
            sb.Append(" union all ");
            sb.Append(queryWithTitles(startDate, endDate, " Dekan_Faculties is not
null "));
            for (int i = 1; i < 5; i++)
            {
                sb.Append(" union all ");
                sb.Append(queryWithTitles(startDate, endDate, "Job_title = " + i));
            }
            sb.Append(" union all ");
            sb.Append(queryWithTitles(startDate, endDate, "(Job_title = 5 or
Job_title = 6)"));

            return NDataAccess.DataAccess.GetDataSet(@sb.ToString(), "Table1",
connectionString);

```

```

    }
    public static DataSet getAgeFemale(DateTime startDate, DateTime endDate)
    {
        StringBuilder sb = new StringBuilder();
        sb.Append(allTitlesWoomen(startDate, endDate));
        sb.Append(" union all ");
    }

```

Продолжение приложения А

```

        sb.Append(queryWithTitles(startDate, endDate, " (Dekan_Faculties is not null) and
(gender like 'Жен') "));
        for (int i = 1; i < 5; i++)
        {
            sb.Append(" union all ");
            sb.Append(queryWithTitles(startDate, endDate, "(gender like 'Жен')
and Job_title = " + i));
        }
        sb.Append(" union all ");
        sb.Append(queryWithTitles(startDate, endDate, " (gender like 'Жен') and
(Job_title = 5 or Job_title = 6)"));
        return NDataAccess.DataAccess.GetDataSet(sb.ToString(), "Table1",
connectionString);
    }
}
//Распределение персонала по полу и возрасту

public static class Teachers
{
    public static DataSet getCair(int cair)
    {
        //string result = "select f.id,w.name as Должность,f.surname as
Фамилия,f.name as Имя,f.middlename as Отчество,birthday as 'Дата рождения',telephone as
Телефон,d.name as 'ученое звание',titles_date as 'Дата получения ученого
звания',a.second_name as 'ученая степень',degress_date as 'Дата получения ученой
степени',terms_of_work as 'Условия привлечения к труд. деят.' from Teachers as f ,
Working_positions as w,Titles as d,Degrees as a where f.Job_title = w.id and
f.titles_id=d.id and f.degrees_id = a.id and Cair = " + cair;
        string result = "select f.id,w.name as Должность,f.surname as
Фамилия,f.name as Имя,f.middlename as Отчество,convert(VARCHAR(10),birthday,105) as 'Дата
рождения',d.name as 'ученое звание',convert(VARCHAR(10),titles_date,105) as 'Дата
получения ученого звания',a.second_name as 'ученая
степень',convert(VARCHAR(10),degress_date,105) as 'Дата получения ученой
степени',terms_of_work as 'Условия привлечения к труд. деят.' from Teachers as f left
join Working_positions as w on w.id=f.Job_title left join Titles as d on d.id =
f.titles_id left join Degrees as a on a.id = f.degrees_id where Cair = " + cair + "
and f.enable > 0";
        return NDataAccess.DataAccess.GetDataSet(@result, "Table1",
connectionString);
    }
    public static DataSet getTeachers()//Главная таблица
    {
        //string result = "select f.id,c.name as Кафедра,w.name as
Должность,f.surname as Фамилия,f.name as Имя,f.middlename as Отчество,gender as
Пол,birthday as 'Дата рождения',passport_serial as 'Серия паспорта',passport_number as
'Номер паспорта',passport_gives as 'Кем выдан',passport_create as 'Когда
выдан',registration as 'Прописка',telephone as Телефон,educational_institution as
'Образовательное учреждение',specialty_of_diplom as 'Специальность по диплому',d.name as
'ученое звание',degress_date as 'Дата получения ученого звания',a.second_name as 'ученая
степень',titles_date as 'Дата получения ученой степени',terms_of_work as 'Условия
привлечения к труд. деят.',competitive_selection_start_date as 'Начало кон.
отб',competitive_selection_end_date as 'Окончание кон. отб',Training_dates as 'Повышения
квалификации',rate as 'Ставка',total_experience_date as 'Стаж',experience_date as 'Дата,
педагогический стаж' from Teachers as f , cair as c, Working_positions as w,Titles as
d,Degrees as a where f.cair = c.id and f.Job_title = w.id and f.titles_id=d.id and
f.degrees_id = a.id";
    }
}

```

```

        string result = "select f.id,c.name as Кафедра,w.name as
        Должность,f.surname as Фамилия,f.name as Имя,f.middlename as Отчество,gender as
        Пол,convert(VARCHAR(10),birthday,105) as 'Дата рождения',passport_serial as 'Серия
        паспорта',passport_number as 'Номер паспорта',passport_gives as 'Кем
        выдан',convert(VARCHAR(10),passport_create,105) as 'Когда выдан',registration as
        'Прописка',telephone as Телефон,educational_institution as 'Образовательное
        учреждение',specialty_of_diplom as 'Специальность по диплому', d.name as 'ученое

```

Продолжение приложения А

```

звание',convert(VARCHAR(10),titles_date,105) as 'Дата получения ученого
звания',a.second_name as 'ученая степень',convert(VARCHAR(10),degress_date,105) as 'Дата
получения ученой степени',terms_of_work as 'Условия привлечения к труд.
деят.',convert(VARCHAR(10),competitive_selection_start_date,105) as 'Начало кон.
отб',convert(VARCHAR(10),competitive_selection_end_date,105) as 'Окончание кон.
отб.',convert(VARCHAR(10),Training_dates,105) as 'Повышения квалификации',rate as
'Ставка',convert(VARCHAR(10),total_experience_date,105) as
'Стаж',convert(VARCHAR(10),experience_date,105) as 'Дата, педагогический стаж'
,convert(VARCHAR(10),education_date,105) as 'Дата трудоустройства' from Teachers as f
left join cairs as c on c.id=f.Cairs left join Working_positions as w on
w.id=f.Job_title left join Titles as d on d.id=f.titles_id left join Degrees as a on a.id
= f.degrees_id where f.Cairs is not null and f.enable > 0";
        return NDataAccess.DataAccess.GetDataSet(@result, "Table1",
connectionString);
    }
    public static DataSet getTeachersDekans()//Главная таблица
    {
        //string result = "select f.id,dek.second_name as Факультет,f.surname as
        Фамилия,f.name as Имя,f.middlename as Отчество,gender as Пол,birthday as 'Дата
        рождения',passport_serial as 'Серия паспорта',passport_number as 'Номер
        паспорта',passport_gives as 'Кем выдан',passport_create as 'Когда выдан',registration as
        'Прописка',telephone as Телефон,educational_institution as 'Образовательное
        учреждение',specialty_of_diplom as 'Специальность по диплому',d.name as 'ученое
        звание',degress_date as 'Дата получения ученого звания',a.second_name as 'ученая
        степень',convert(VARCHAR(10),titles_date,105) as 'Дата получения ученой
        степени',terms_of_work as 'Условия привлечения к труд.
        деят.',competitive_selection_start_date as 'Начало кон.
        отб',competitive_selection_end_date as 'Окончание кон. отб.',Training_dates as 'Повышения
        квалификации',rate as 'Ставка',total_experience_date as 'Стаж',experience_date as 'Дата,
        педагогический стаж' from Teachers as f , Faculties as dek,Titles as d,Degrees as a where
        f.Dekan_Faculties = dek.id and f.titles_id=d.id and f.degrees_id = a.id";
        string result = "select f.id,dek.second_name as Факультет,f.surname as
        Фамилия,f.name as Имя,f.middlename as Отчество,gender as
        Пол,convert(VARCHAR(10),birthday,105) as 'Дата рождения',passport_serial as 'Серия
        паспорта',passport_number as 'Номер паспорта',passport_gives as 'Кем
        выдан',convert(VARCHAR(10),passport_create,105) as 'Когда выдан',registration as
        'Прописка',telephone as Телефон,educational_institution as 'Образовательное
        учреждение',specialty_of_diplom as 'Специальность по диплому',d.name as 'ученое
        звание',convert(VARCHAR(10),titles_date,105) as 'Дата получения ученого
        звания',a.second_name as 'ученая степень',convert(VARCHAR(10),degress_date,105) as 'Дата
        получения ученой степени',terms_of_work as 'Условия привлечения к труд.
        деят.',convert(VARCHAR(10),competitive_selection_start_date,105) as 'Начало кон.
        отб',convert(VARCHAR(10),competitive_selection_end_date,105) as 'Окончание кон.
        отб.',convert(VARCHAR(10),Training_dates,105) as 'Повышения квалификации',rate as
        'Ставка',convert(VARCHAR(10),total_experience_date,105) as
        'Стаж',convert(VARCHAR(10),experience_date,105) as 'Дата, педагогический стаж' from
        Teachers as f left join dbo.Faculties as dek on dek.id = f.Dekan_Faculties left join
        Titles as d on d.id=f.titles_id left join Degrees as a on a.id = f.degrees_id where
        f.Cairs is null and f.enable > 0";
        return NDataAccess.DataAccess.GetDataSet(@result, "Table1",
connectionString);
    }
    public static DataSet getSecondNameCairs()//Сокращенные кафедры
    {

```

```

        return NDataAccess.DataAccess.GetDataSet(@"select second_name from
dbo.Cairs", "Table1", connectionString);
    }
    public static DataSet getJobTitles()//Должность
    {
        return NDataAccess.DataAccess.GetDataSet(@"select name from
dbo.Working_positions", "Table1", connectionString);
    }

```

Продолжение приложения А

```

    }

    public static DataSet getScientificDegrees()//Ученые степени
    {
        return NDataAccess.DataAccess.GetDataSet(@"select name from dbo.Titles",
"Table1", connectionString);
    }

    public static DataSet getAcademicTitle()//Ученые звания
    {
        return NDataAccess.DataAccess.GetDataSet(@"select second_name from
dbo.Degrees", "Table1", connectionString);
    }

    public static string getCountRows()//Количество строк в таблице
    {
        ds = NDataAccess.DataAccess.GetDataSet(@"select max(id) as count from
dbo.Teachers", "Table1", connectionString);
        string result = ds.Tables[0].Rows[0].ItemArray[0].ToString();
        if(result=="")result= "0";
        return (result);
    }

    public static void insertPerson(string values)//Добавление преподавателя
    {
        string head =
"id,Cairs,Job_title,surname,name,middlename,gender,birthday,passport_serial,passport_numbe
r,passport_gives,passport_create,registration,telephone,educational_institution,specialt
y_of_diplom,titles_id,titles_date,degrees_id,degress_date,terms_of_work,competitive_selec
tion_start_date,competitive_selection_end_date,rate,experience_date,Training_dates,total_
experience_date,education_date,enable,start_date";
        string result = "INSERT INTO dbo.Teachers (" + head + ") values " +
values;
        NDataAccess.DataAccess.ExecuteNonQuery2(result, connectionString);
    }

    public static void updatePerson(string values, int id)//Добавление
преподавателя
    {
        string result = "UPDATE dbo.Teachers SET " + values + " where id = " +
id;
        NDataAccess.DataAccess.ExecuteNonQuery2(result, connectionString);
    }

    public static void updateEndDate(string endDate, int id)//Добавление
преподавателя
    {
        string result = "UPDATE dbo.Teachers SET end_date='" + endDate + "' where
id = " + id;
        NDataAccess.DataAccess.ExecuteNonQuery2(result, connectionString);
    }

    public static void insertDecan(string values)//Добавление преподавателя
    {
        string head =
"id,surname,name,middlename,gender,birthday,passport_serial,passport_number,passport_give
s,passport_create,registration,telephone,educational_institution,specialty_of_diplom,titl
es_id,titles_date,degrees_id,degress_date,terms_of_work,competitive_selection_start_date,

```



```

        {
            NDataAccess.DataAccess.ExecuteNonQuery2(@"INSERT INTO Users
(login,password) VALUES ('" + login + "', '" + Cryptography.GetHashCodeString(pass) + "')",
connectionString);
            int id = 0;

```

Продолжение приложения А

```

        ds = NDataAccess.DataAccess.GetDataSet(@"Select Max(id) from Roles", "Table1",
connectionString);
        if (ds.Tables[0].Rows[0].ItemArray[0].ToString() == "") id = 1;
        else id = (int)ds.Tables[0].Rows[0].ItemArray[0];
        id++;

        NDataAccess.DataAccess.ExecuteNonQuery2(@"INSERT INTO Roles
(id,role,login) VALUES ('" + id + "', '" + role + "', '" + login + "')",
connectionString);

        return true;
    }
    else
    {
        return false;
    }
}
public static void update(string oldLogin, string login, string pass, string
role)
{
    NDataAccess.DataAccess.ExecuteNonQuery2(@"UPDATE Users SET login='" +
login + "',password='" + Cryptography.GetHashCodeString(pass) + "' where login='" +
oldLogin+"'", connectionString);
    NDataAccess.DataAccess.ExecuteNonQuery2(@"UPDATE Roles SET login='" +
login + "',role='" + role + "' where login='" + oldLogin+"'", connectionString);
}

public static void dell(string login)
{
    NDataAccess.DataAccess.ExecuteNonQuery2("DELETE FROM Users WHERE login
='" + login + "'", connectionString);
    NDataAccess.DataAccess.ExecuteNonQuery2("DELETE FROM Roles WHERE login
='" + login + "'", connectionString);
}

public static User user(string login)
{
    ds= NDataAccess.DataAccess.GetDataSet(@"SELECT u.login as 'Имя
пользователя',u.password as 'Пароль',r.role as 'Роль' FROM Users as u join Roles as r on
r.login=u.login where u.login='"+login+"'", "Table1", connectionString);
    User user = new User();
    user.login = ds.Tables[0].Rows[0].ItemArray[0].ToString();
    user.password = ds.Tables[0].Rows[0].ItemArray[1].ToString();
    user.role = ds.Tables[0].Rows[0].ItemArray[2].ToString();
    return user;
}

public static DataSet users()
{
    return NDataAccess.DataAccess.GetDataSet(@"SELECT u.login as 'Имя
пользователя','*****' as 'Пароль',r.role as 'Роль' FROM Users as u join Roles as r on
r.login=u.login", "Table1", connectionString);
}

public static string role(string login, string pass)
{

```

```

        ds = NDataAccess.DataAccess.GetDataSet(@"SELECT r.role,u.login,u.password
FROM Users as u join Roles as r on r.login=u.login where u.login='" + login + "' and
u.password='" + Cryptography.getHashString(pass) + "'", "Table1", connectionString);
        if (ds.Tables[0].Rows.Count>0)
            return ds.Tables[0].Rows[0].ItemArray[0].ToString();
        else return "role";
    }
}

```