

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМЕНИ Н.Г.ЧЕРНЫШЕВСКОГО»**

Кафедра математического и компьютерного моделирования

Проектирование и реализация ИС «Абитуриент»

АВТОРФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студентки 4 курса 451 группы

направление 38.03.05 — Бизнес-информатика

механико-математического факультета

Кургановой Златы Михайловны

Научный руководитель
доцент, к.ф. - м.н.

О.М. Ромакина

Зав. кафедрой
зав.каф., д.ф.-м.н., доцент

Ю.А. Блинков

Саратов 2019

Введение. Стремительное развитие информационных технологий в последнее десятилетие обусловлено переходом человечества к модели постиндустриального информационного общества, где определяющим фактором являются теоретические знания, а главными структурами общества выступают места их производства и накопления. Существует несколько критериев, в соответствии с которыми выделяют основные характеристики и маркеры информационного общества, одним из таких является технологический. Ключевым маркером данного критерия выступают информационные технологии, которые широко применяются в производстве, учреждениях, а также системе образования.

В настоящее время использование информационных систем в высших образовательных учреждениях позволяет охватить большинство аспектов их деятельности: от автоматизации отдельно взятых направлений работы до полной автоматизации всей структуры высшего учебного заведения. Вне зависимости от объекта автоматизации в образовательном учреждении такие системы внедряют, преследуя конечную цель — повышение качества образования. Первым звеном данной цепи является качественный отбор абитуриентов.

На протяжении нескольких последних лет в ходе приемной кампании механико-математического факультета Саратовского национального исследовательского государственного университета имени Н. Г. Чернышевского некоторые затруднения доставляли многочисленные проблемы внесения, редактирования, удаления и хранения существенного объема важных сведений об абитуриентах. Используемые в данных целях электронные таблицы «excel» отличались искаженностью хранимых данных, а также возникновением в них множественных ошибок при ручном заполнении, что влекло за собой существенные проблемы при анализе информации. В связи с этим актуальность данной работы заключается в необходимости разработки информационной системы «Абитуриент». Объектом исследования является процесс проведения приемной кампании факультета и дальнейшая работа с полученными данными. Предметом исследования является специально разработанная и спроектированная информационная система «Абитуриент».

Цель бакалаврской работы — разработка информационной системы для оптимизации процесса сбора информации в рамках приемной кампании. Практическая значимость заключается в разработке системы, обеспечивающей эффективное функционирование данного процесса. Для достижения данной цели необходимо выполнить следующие задачи:

1. Проанализировать актуальность данной темы;
2. Рассмотреть внутренние процессы, протекающие в данной области;
3. Осуществить выбор системы управления базами данных;
4. Разработать информационную систему;
5. Создать базу данных для предметной области;
6. Обеспечить перенос данных из файлов Microsoft Excel в созданную базу данных ;
7. Создать графическую оболочку для информационной системы «Абитуриент»;
8. Организовать хранение информации;
9. Осуществить предоставление доступа (для поиска, изменения и т. п.) для определенных групп лиц.

Стоит отметить, что в ходе эксплуатации возможно внесение корректировок в систему в связи с изменениями внутренних процессов ВУЗа, факультета и/или действующих законов.

Цель данной бакалаврской работы будет считаться достигнутой, если в результате выполнения перечисленных выше задач будет спроектирована функционирующая информационная система, предназначенная для эксплуатации в высшем учебном заведении в ходе приемной кампании на факультете.

Основное содержание работы состоит из 2 разделов:

1. Теоретические основы;
2. Реализация практической части.

В первом разделе рассматривается понятие «Информационная система». Приведены основные задачи, которые призваны решить информационные системы.

Информационная система — система, предназначенная для хранения, поиска и обработки информации, включающая соответствующие организацион-

ные ресурсы (человеческие, технические, финансовые и т. д.), которые обеспечивают и распространяют информацию.

Информационные системы обеспечивают решение несколько основных задач:

- Анализ и прогнозирование потоков разнообразной информации, перемещающихся в обществе. Изучаются потоки документов с целью их минимизации, стандартизации и приспособления для эффективной обработки на вычислительных машинах, а также особенности потоков информации, протекающей через журналы, газеты, радиоканалы, телевизионные каналы и другие каналы распространения информации. Оценивается влияние распространяемой информации на научно-технический прогресс и состояние общества.
- Исследование способов представления и хранения информации, создание специальных языков для формального описания информации различной природы, разработка специальных приемов сжатия и кодирования информации, аннотирования объемных документов и реферирования их. В рамках этого направления развиваются работы по созданию банков данных большого объема, хранящих информацию из различных областей знаний в форме, доступной для вычислительных машин.
- Построение различных процедур и технических средств для их реализации, с помощью которых можно автоматизировать процесс извлечения информации из документов, не предназначенных для вычислительных машин, а ориентированных на восприятие их человеком: Эти исследования тесно связаны с проблемой извлечения смысла (содержания) тех или иных документов при вводе их в банки данных и другие информационные хранилища, ориентированные на компьютеры.
- Создание информационно-поисковых систем, способных воспринимать запросы к информационным хранилищам, сформулированные на естественном языке, а также специальных языках запросов для систем такого типа.
- Создание сетей хранения, обработки и передачи информации, в состав которых входят информационные банки данных, терминалы, обрабатывающие центры и средства связи.

База данных (БД) — представленная в объективной форме совокупность самостоятельных материалов (статей, расчётов, нормативных актов, судебных решений и иных подобных материалов), систематизированных таким образом, чтобы эти материалы могли быть найдены и обработаны с помощью электронной вычислительной машины. В рамках поставленной задачи было принято решение разработать реляционную базу данных для информационной системы «Абитуриент». Реляционная модель данных позволяет представлять информацию о предметной области с помощью взаимосвязанных таблиц. В таких базах данных вся информация сведена в таблицы, строки и столбцы которых называются записями и полями соответственно. Эти таблицы получили название реляций. Записи в таблицах не повторяются. Их уникальность обеспечивается первичным ключом, содержащим набор полей, однозначно определяющих запись. Взятая за основу модель данных имеет ряд положительных характеристик для реализации поставленной задачи данной работы, таких как:

- отображение информации в наиболее простой для пользователя форме,
- возможность лаконичного описания операций над данными; обусловленная развитым математическим аппаратом, взятым за основу,
- поддержка создания языков манипулирования данными не процедурного типа;
- манипулирование данными на уровне выходной БД и возможность их изменения.

PostgreSQL — свободная объектно-реляционная система управления базами данных (СУБД).

Сильными сторонами PostgreSQL считаются:

- высокопроизводительные и надёжные механизмы транзакций и репликации;
- расширяемая система встроенных языков программирования;
- наследование;
- легкая расширяемость.

Основными возможностями выступают:

- А. Функции, являющиеся блоками кода, исполняемыми на сервере, а не на клиенте БД.

- Б. Триггеры, определяемые как функции, иницируются DML-операциями.
 - В. Правила и представления. Механизм правил (от англ. rules) представляет собой механизм создания пользовательских обработчиков не только DML-операций, но и операции выборки.
 - Г. Индексы.
 - Д. Многоверсионность. PostgreSQL поддерживает одновременную модификацию БД несколькими пользователями с помощью механизма MultiVersion Concurrency Control (MVCC).
 - Е. Поддержка большого набора встроенных типов данных.
 - Ж. Возможность расширения пользователем для собственных нужд практически в любом аспекте.
3. Наличие наследования и партицирования.

Исходя из всех вышеперечисленных преимуществ свободной объектно-реляционной системы управления базами данных PostgreSQL, был сделан вывод об практичности и эргономичности разработки и реализации базы данных информационной системы «Абитуриент» посредством применения предоставляемых данной СУБД возможностей. DBdesigner — это инструмент для визуального проектирования баз данных, позволяющий видеть и управлять всеми связями между таблицами. Интерфейс программы интуитивен: таблицы выглядят как отдельные блоки, записи которых связаны линиями со стрелочками (указывающими тип связи) с другими таблицами или блоками. ER-модель (от англ. entity-relationship model, модель «сущность - связь») — модель данных, позволяющая описывать концептуальные схемы предметной области. Она используется при высокоуровневом (концептуальном) проектировании баз данных. С ее помощью можно выделить ключевые сущности и обозначить связи, которые могут устанавливаться с ними. Во время проектирования баз данных происходит преобразование ER-модели в конкретную схему базы данных на основе выбранной модели данных (реляционной, объектной, сетевой или др.). ER-модель представляет собой формальную конструкцию, которая сама по себе не предписывает никаких графических средств ее визуализации. В качестве стандартной графической нотации, с помощью которой можно визуализировать ER-модель, была пред-

ложена диаграмма «сущность-связь» (англ. entity-relationship diagram, ERD, ER-диаграмма).

Python — высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода. Синтаксис ядра Python минималистичен. В то же время стандартная библиотека включает большой объём полезных функций.

Python поддерживает несколько парадигм программирования, в том числе объектно-ориентированное. программирования. Дизайн языка Python построен вокруг объектно-ориентированной модели программирования. Реализация ООП в Python является элегантной, мощной и хорошо продуманной, но вместе с тем достаточно специфической по сравнению с другими объектно-ориентированными языками.

Во втором разделе приведено описание выбранной предметной области.

Приемная комиссия университета — это сложный учебно-хозяйственный комплекс с многочисленными внешними и внутренними связями. Большие возможности для совершенствования управления приемной комиссией предоставляет использование вычислительной техники и средств связи. В организационной системе наиболее трудоемкими являются процессы, связанные с обработкой информации — сбор, накопление, преобразование, отображение, хранение, передача и вывод. Ускорить эти процессы и облегчить труд персонала приемной комиссии позволяет информационная подсистема для учета абитуриентов.

Для организации эффективного ведения документации приемной комиссии требуется наличие централизованного хранения информации, а также свободного доступа к ней. Существенной проблемой является необходимость быстрого и результативного поиска необходимой информации среди огромного объема данных. Применение АИС позволяет уменьшить время поиска информации и способствует оптимальному взаимодействию в области создания и контроля прохождения документов.

В ходе проектирования и разработки ИС «Абитуриент», предполагается решение следующих задач:

1. Обеспечение возможности занесения информации об абитуриентах и их результатах ЕГЭ во время работы приемной комиссии;

2. Обеспечение корректировки информации об абитуриентах во время работы приемной комиссии;
3. Обеспечение возможности выборки записей;
4. Обеспечение возможности занесения информации о студентах;
5. Обеспечение возможности корректировки информации о студентах.

Исходные табличные данные содержали информацию об абитуриентах, подавших свои документы. В документах хранились excel-таблицы с данными о факультете, форме обучения, фамилии, имени и отчестве абитуриента, оригинале поданного документа, гражданстве, названии ранее оконченного учебного заведения, районе, категории зачисления, форме оплаты, типе оконченного учебного заведения, годе окончания учебного заведения и изучаемом языке поступающего. Однако предоставленные данные большого объема имели плохо организованную структуру, неудобный формат для пользователя при внесении, редактировании и удалении записей, в ходе чего возникали ошибки и расхождения в таблицах. Подобного рода проблемы существенно затрудняли эффективную работу приемной комиссии факультета. Возникла необходимость преобразования имеющихся данных в удобную и практичную для пользователей форму, содержащую в себе все необходимые аспекты информационной системы «Абитуриент».

На основе имеющейся информации с помощью интерфейса PgAdmin для системы управления базами данных PostgreSQL была создана база данных, содержащая 10 таблиц: «Абитуриент», «Баллы_Абитуриентов», «Предметы», «Специальности», «Гражданство», «Районы», «Учебные_Заведения», «Форма_Оплаты», «Тип_Оконченного_УЗ» и «Конкурсная_Категория».

Таблица «Абитуриент» содержит следующие поля:

- «РК_Абитуриент» типа integer, которому присвоен первичный ключ (от англ. primary key) таблицы;
- «ФИО» типа varchar;
- «Дата_Рождения» типа date;
- «Пол» типа varchar (1) с ограничением на количество символов в строке, равное 1, для удобства внесения информации о поле абитуриента (буква «М» или «Ж»);

- «Оригинал» типа boolean, где при наличии оригинала аттестата полю присваивается значение «1», а при отсутствии «0».

Далее перечислены поля, которым присвоен внешний ключ (от англ. foreign key), другими словами, ссылку полей связанной таблицы на одноименные поля другой. Все поля имеют тип integer:

- «FK_Специальность»;
- «FK_Гражданство»;
- «FK_Район»;
- «FK_УЗ»;
- «FK_Форма_Оплаты»;
- «FK_Тип»;
- «FK_Категория».

Последнее поле

- «Год_Поступления» типа varchar(4) с ограничением на количество символов в строке, равное 4, т.е. количеству цифр в году (например, 2015).

Таблицы:

- «Специальности»,
- «Гражданство»,
- «Районы»,
- «Учебные_Заведения»,
- «Форма_Оплаты»,
- «Тип_Оконченного_УЗ»,
- «Предметы»,
- «Конкурсная_Категория»

имеют одинаковую по построению структуру: первое поле типа integer содержит первичный ключ:

- «РК_Специальность»,
- «РК_Гражданство»,
- «РК_Район»,
- «РК_УЗ»,
- «РК_Форма»,
- «РК_Тип»,
- «РК_Предмет»,

- «РК_Категория»,

соответственно. Первичный ключ выступает атрибутом, однозначно идентифицирующим экземпляр сущности. Благодаря первичному и внешнему ключу между таблицами образовывается связь. Связь — это ассоциирование двух или более сущностей. Связь между объектом и его свойством может быть различной. Объект может обладать только одним значением какого-то свойства. Например, договор об оплате можно заключить только на одного абитуриента. Для других свойств возможно существование одновременно нескольких значений у одного объекта. Так, к примеру, одну специальность могут выбрать множество студентов. Вторым полем во всех таблицах является «Наименование» типа varchar. В данном поле могут быть перечислены данные, относящиеся к конкретной таблице.

Таблица «Баллы_Абитуриентов» содержит поля:

- «РК_Баллы_Абитуриентов» - присвоен первичный ключ;
- «FK_Абитуриент» - присвоен внешний ключ со ссылкой на таблицу «Абитуриент»;
- «FK_Предмет» - присвоен внешний ключ со ссылкой на таблицу «Предметы»;
- «Балл» - обычное поле, позволяющее отследить баллы абитуриента по каждому предмету.

Всем вышеперечисленным полям присвоен тип integer, так как все записи содержат числовые значения. В соответствии с приведенными выше таблицами также была построена модель данных с помощью ранее упомянутого инструмента DBdesigner.

Далее было необходимо обеспечить перенос данных из файлов, содержащих таблицы «excel», в спроектированную базу данных. Осуществить данное требование возможно с помощью создания программы-парсера на языке программирования Python. Были импортированы следующие модули: «xlrd» для чтения и работы с «xls»-файлами; «glob», предоставляющий функцию для получения списка файлов на основе заданного шаблона; «os», содержащий множество функций для работы с операционной системой. Подключенная следующим шагом библиотека «psycopg 2» является самой популярной для PostgreSQL в языке программирования Python. Регулярные выражения,

встроены в Python и доступны при помощи модуля «re». Благодаря регулярным выражениям можно выяснить соответствует ли данная строка шаблону или совпадает ли шаблон где-нибудь с этой строкой. Далее импортируется модуль «configparser», который используется для создания и работы с файлами конфигурации.

В последующем разрабатывается основной алгоритм парсинга табличных данных с помощью которого сначала происходит их извлечение из конкретных столбцов, которые должны быть отражены в базе данных в соответствии с требованиями, устанавливается связь между созданной базой данных и парсером, обозначаются основные таблицы, указываются внешние и первичные ключи. Для данных каждого столбца указывается путь для «перехода» в соответствующую таблицу базы данных. Происходит конфигурация данных и они импортируются из файлов формата «xls» в базу данных. Ее заполнение можно отследить в интерфейсе PgAdmin для управления СУБД PostgreSQL.

Для реализации графического интерфейса пользователя был использован модуль или так называемый набор «привязок» PyQt5.

PyQt — набор «привязок» графического фреймворка Qt для языка программирования Python, выполненный в виде расширения Python.

PyQt включает в себя Qt Designer (Qt Creator) — дизайнер графического интерфейса пользователя. Программа pyuic генерирует Python код из файлов, созданных в Qt Designer. Это делает PyQt очень полезным инструментом для быстрого прототипирования. Кроме того, можно добавлять новые графические элементы управления, написанные на Python, в Qt Designer.

Qt Designer — кроссплатформенная свободная среда для разработки графических интерфейсов (GUI) программ, использующих библиотеку Qt; входит в состав Qt framework.

Qt Designer позволяет создавать графические интерфейсы пользователя при помощи ряда инструментов. Существует панель инструментов «Панель виджетов», в которой доступны для использования элементы интерфейса — виджеты, такие как, например, «выпадающий список» ComboBox, «поле ввода» QLineEdit, «кнопка» QPushButton и многие другие. Каждый виджет имеет свой набор свойств, определяемый соответствующим ему классом библиотеки Qt. Свойства виджета могут быть изменены при помощи «Редакто-

ра свойств». Для каждого класса свойств виджета существует свой специализированный редактор. Характерной особенностью Qt Designer является поддержка визуального редактирования сигналов и слотов. Так, например, можно связать сигнал, генерируемый по переключению состояния виджета CheckBox со слотом, отвечающим за доступность другого виджета.

Разработанный интерфейс сохраняется в файл с расширением ui, который подключается к создаваемой программе с помощью специальных методов библиотеки Qt. Этот файл имеет xml-формат, и может, в случае необходимости, редактироваться в любом текстовом редакторе.

Далее в работе приведены результаты реализации программного кода для создания графического интерфейса пользователя для информационной системы «Абитуриент» с использованием вышеописанных инструментов. Представлены главная страница, предоставляющая возможность просмотра всех записей об абитуриентах с фильтрацией по наименованию учебного заведения, форма для добавления новой записи об абитуриенте, форма просмотра специальностей на факультете, форма добавления новой записи о специальностях. По аналогии с перечисленными выше вкладками интерфейса также был написан код для создания визуальных форм просмотра и занесения новых записей для следующих таблиц базы данных: «Гражданство», «Районы», «Учебные заведения», «Тип учебного заведения», «Форма оплаты», «Конкурсная категория», «Предметы», «Баллы». Визуальное представление соответствующих форм было отражено в приложении.

Заключение. Современный этап развития, определяемый ростом научно-технического прогресса, характеризуется стремительным увеличением объема информационных потоков и окончательным признанием информации основным ресурсом развития современного общества. Информация в настоящее время выступает необходимым и незаменимым ресурсом.

Общеизвестным является тот факт, что качественная информация позволяет специалистам различных областей осуществлять свою профессиональную деятельность целенаправленно и эффективно. Поэтому в сложившихся условиях роста объема и роли такой информации возникает необходимость применения информационных технологий, позволяющих осуществлять

ее сбор, структурирование, хранение, поиск, обработку и выдачу в соответствии с требованиями, предъявляемыми пользователями.

В ходе выполнения данной бакалаврской работы была решена проблема хранения, редактирования, внесения и удаления информации об абитуриентах механико-математического факультета Саратовского государственного научного исследовательского университета имени Н. Г. Чернышевского. Исходная проблема была решена посредством анализа данных и детального описания предметной области «Абитуриент», на основе которой в СУБД PostgreSQL была создана база данных с целью структурирования информации из электронных таблиц, устранения возможности ошибок и опечаток при ручном вводе данных, а также улучшения ведения учета абитуриентов. При создании алгоритма парсинга данных электронных таблиц из Microsoft Office Excel были изучены методы использования высокоуровневого языка программирования Python в разрезе объектно-ориентированного программирования с применением расширенного пакета библиотек и модулей, которые обеспечили качественную и быструю обработку текстовых данных. Также с помощью определённых пакетов и модулей был создан графический интерфейс, наполненный всем необходимым функционалом для удобства и оптимизации работы членов приемной комиссии. Для лучшего понимания архитектуры базы данных «Абитуриент» обычным пользователем был использован инструмент для визуального проектирования баз данных, позволяющего видеть и управлять всеми связями между таблицами, — DBdesigner. Визуальное представление таблиц обеспечивает лучшее понимание структуры разработанной базы данных, связей между полями и таблицами, что, в свою очередь помогает пользователю лучше ориентироваться в информационной системе, легко осваивать процесс чтения, создания, удаления, качественного и количественного отбора, конфигурации и быстрого, экономичного по времени оперирования бизнес-данными с помощью компьютерных технологий в ходе их анализа.

Таким образом, можно сделать выводы, что:

1. Цели и задачи поставленные в бакалаврской работе были выполнены;
2. Разработанную систему можно улучшать и менять ее функционал в соответствии с требованиями внутренней и внешней среды.

В ходе проделанной работы удалось более детально изучить процесс проектирования и реализации информационных систем, разобраться в тонкостях создания и управления базами данных.