

Таким образом, разработана автоматизированная информационная система учета контингента студентов университета с удобным и понятным пользователю интерфейсом. В справочниках вся информация разграничена по блокам, при ошибочных действиях появляются окна с информацией об ошибке. Кроме этого возможен быстрый переход между подсистемами. Использование разработанной автоматизированной информационной системы приведет к повышению производительности и качества труда сотрудников, оперативности и достоверности данных, исключению многих ошибок, предоставлению современной отчетности

ЛИТЕРАТУРА

1. Головач, В. Искусство мыть слона [Электронный ресурс] / В. Головач // Дизайн пользовательского интерфейса. – Режим доступа: <http://uibook2.usethics.ru/uibookII>. – Дата доступа: 06.05.2015.
2. Тидвелл, Дж. Разработка пользовательских интерфейсов / Дж. Тидвелл. – СПб. : Питер, 2008. – 395 с.
3. Шилдт, Г. C# 3.0 : полное руководство / Г. Шилдт. – М. : И.Д. Вильямс, 2010. – 992 с.
4. Windows Forms [Электронный ресурс] / Википедия. – Режим доступа: http://ru.wikipedia.org/wiki/Windows_Forms. – Дата доступа: 26.05.2015.

УДК 004.45.001.63

**ОСОБЕННОСТИ ПРОЕКТИРОВАНИЯ БАЗЫ ДАННЫХ АВТОМАТИЗИРОВАННОЙ
ИНФОРМАЦИОННОЙ СИСТЕМЫ УЧЕТА КОНТИНГЕНТА СТУДЕНТОВ УНИВЕРСИТЕТА****Д.В. ПОПКОВА***(Представлено: М.В. ДЕКАНОВА)*

Рассмотрены необходимые сущности для построения базы данных, с их описанием применения и характеристик. Построена концептуальная схема, выделены необходимые отношения и связи между таблицами. Все таблицы нормализованы и приведены к третьей формальной форме. Обоснован выбор системы управления базами данных для хранения информации о студентах. Описаны способы поддержания целостности базы данных, а также ошибочного и некорректного ввода сведений при заполнении карточек и документации студентов.

Учет и ведение документации в учреждениях образования становится все сложнее [1, 2]. В деканатах высших учебных заведений (ВУЗ) осуществляется работа с личными данными студентов, распределение студентов по группам, заполнение учебных карт студентов, ведение общего списка студенческих групп, учет аттестационных, экзаменационных ведомостей, движения контингента студентов, учет и оформление приказов о зачислении; разработка и исполнение различных видов приказов, распоряжений и справок.

Всей этой вышперечисленной работой занимается один человек – секретарь деканата. Это очень трудоемкая работа, если выполнять ее вручную. К тому же объем информации, который нужно хранить в бумажном варианте, достаточно велик, и поиск нужной информации значительно затрудняется [2]. Создание автоматизированной информационной системы, основанной на современных технологиях сбора, обработки, анализа, передачи и хранения информации, значительно облегчит труд работника деканата.

Важная роль при разработке автоматизированной информационной системы отводится проектированию базы данных, в которой хранится большая по объему информация о какой-либо области человеческих знаний. Важно, что для пользователя эта база представляется, как единое хранилище информации, куда он может обратиться с запросом. В данной работе рассматривается вопрос проектирования базы данных для автоматизированной информационной системы учета контингента студентов университета.

Выделим следующие сущности проектируемой базы данных:

- студент (описывает студента, обучающегося в университете; характеризуется фамилией, именем, отчеством, номером зачетной книжки, условиями обучения, контингентом и группой);
- группа (представляет собой список всех учебных групп студентов факультета; характеризуется названием, специализацией и формой обучения);
- специализация (описывает все имеющиеся в университете специализации; характеризуется кодом, названием и специальностью, за которой закреплена);
- специальность (хранит все имеющиеся в университете специальности; характеризуется кодом, названием, кафедрой, за которой закреплена, может разделяться на специализации);
- кафедра (описывает кафедры, имеющиеся в университете; характеризуется названием и факультетом, к которому относится);
- факультет (хранит информацию о факультетах университета; характеризуется названием и деканом);

- предмет (описывает изучаемые дисциплины; характеризуется названием, видом отчетности, семестром);
- успеваемость (содержит информацию об успеваемости студента; характеризуется предметом, изучаемым студентом, первой и второй аттестацией, а также полученной отметкой);
- документ (описывает всевозможные документы на студентов; характеризуется студентом и оформленным документом);
- вид документов (представляет собой справочники типов документа; характеризуется типом документа);
- приказ, распоряжение, справка (хранит дату формирования документа для всех видов);
- приказ (описывает приказы по всем его видам и основания назначения);
- распоряжение (описывает распоряжения по всем его видам и основания назначения данного документа);
- справка (описывает выдаваемую справку по месту требования);
- контингент (хранит информацию о студентах на момент поступления в университет. Информация следующего характера: гражданство, национальность, адрес проживания и регистрации, иностранный язык, изучаемый ранее, информация о родителях, семейное и социальное положение, место учебы до поступления).

Чтобы построить схему реляционной базы данных необходимо определить связи между сущностями. На концептуальном уровне связи представляют собой простые ассоциации между уровнями. Одним из основных требований к организации базы данных является обеспечение возможности поиска одних сущностей по значениям других, для чего необходимо установить между ними определенные связи [3].

Для реализации информационной системы учета контингента студентов установим все связи между объектами. А именно, рассмотрим всю информационную систему в совокупности и определим взаимное влияние объектов, составляющих систему.

В проектируемой базе данных все таблицы имеют первичные ключи, присутствуют связи двух типов «один ко многим» и «один к одному» [4].

Концептуальная схема базы данных, в первой нормальной форме, представлена на рисунке 1. Определим совокупность отношений, которые составляют базу данных. Данная совокупность отношений будет содержать всю информацию, которая должна храниться в базе данных.

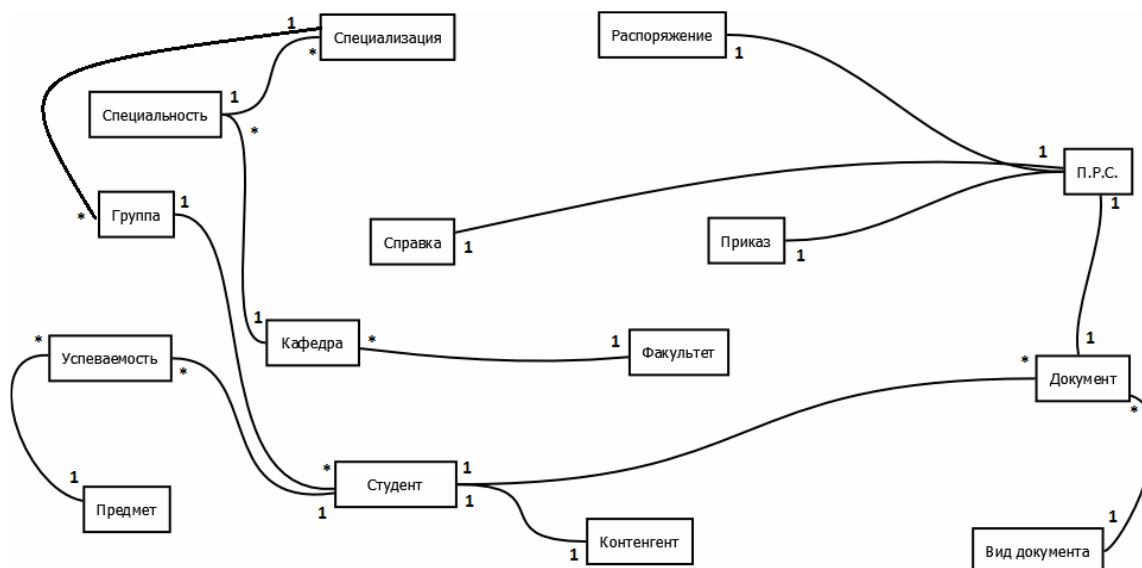


Рис. 1. Концептуальная схема базы данных

На основе полученной концептуальной модели можно определить набор необходимых отношений базы данных. На рисунке 2 представлены отношения для базы данных контингента студентов университета.

В реляционной базе данных каждому объекту и сущности реального мира соответствуют кортежи отношений. И любое отношение должно обладать первичным ключом. Ключ – это минимальный набор атрибутов, по значениям которых можно однозначно найти требуемый экземпляр сущности. Минимальность означает, что исключение из набора любого атрибута не позволяет идентифицировать сущность по оставшимся. Каждое отношение должно обладать хотя бы одним ключом [5].

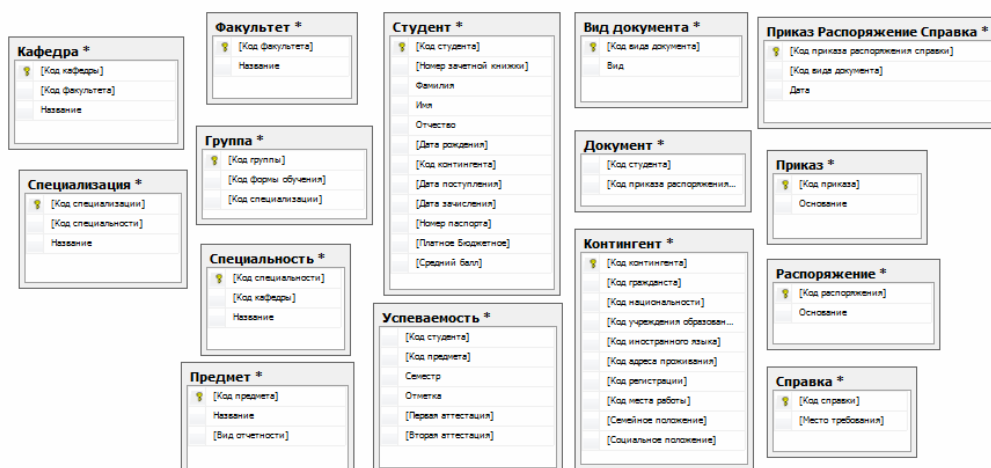


Рис. 2. Набор необходимых отношений базы данных

После определения связей между таблицами, назначения ключей и построения реляционной базы данных, остается привести ее к третьей нормальной форме. Например, имеем таблицу контингент, у которой атрибуты гражданство, национальность, иностранный язык, учреждение образования, место работы, адрес проживания до поступления, адрес регистрации, семейное положение, социальное положение, родственные связи определяют информацию о студенте. Эту таблицу логичнее разбить на несколько отдельных.

Поэтому необходимо привести эту таблицу к третьей нормальной форме. Результатом приведения будут таблицы: место работы, иностранный язык, гражданство, национальность, регистрация, адрес проживания до поступления, учреждение образования (рис. 3).

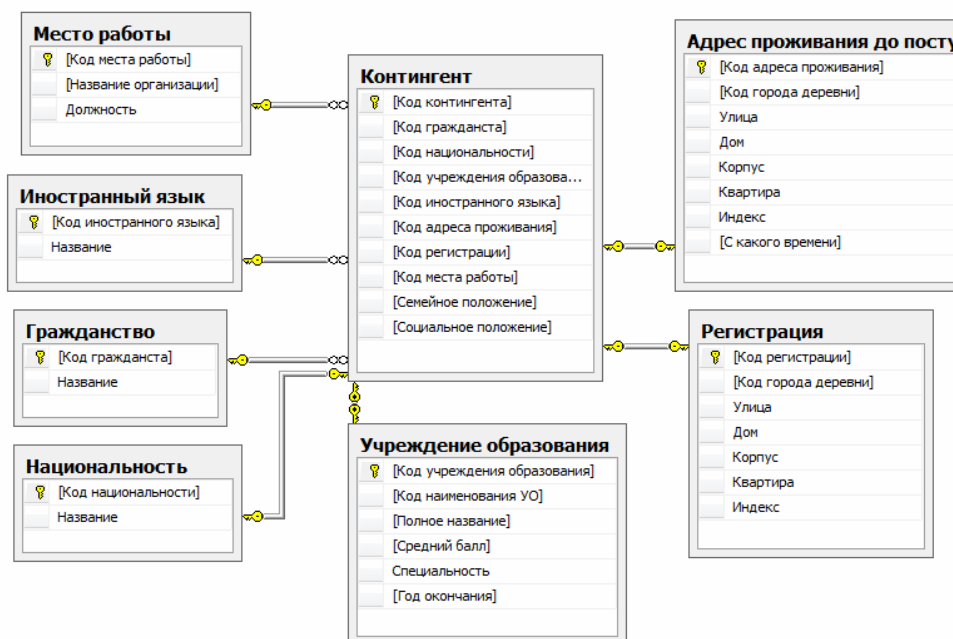


Рис. 3. Результат приведения таблиц к третьей нормальной форме

Процесс нормализации применяем также к таблицам распоряжение, приказ, родственные связи, адрес прописки до поступления, адрес регистрации, группа, учреждение образования. В итоге нормализации таблиц получаем новые таблицы перевод в другую группу – для оформления документа на перевод в другую группу, отчисление, восстановление – для оформления документов на отчисление, либо восстановление, перевод – для оформления документов на перевод, академический отпуск – для оформле-

ния документов на академический отпуск, смена студенческого билета, зачетной книжки – для оформления документов на смену студенческого билета, либо зачетной книжки, начисление стипендии – для оформления документов на начисление стипендии, смена фамилии – для оформления документов на смену фамилии, выпуск – для оформления документов на получение квалификации, наименование учреждения образования – для хранения всех видов учреждений образования, информация о родственниках – для хранения информации о каждом родственнике студента, город, деревня – для хранения информации о городах и деревнях, район – для хранения информации о районах, область – для хранения информации об областях, страна – для хранения информации о странах, состав группы – функциональная таблица для формирования групп.

Для организации хранения данных была выбрана система управления базами данных (СУБД) MySQL Server 5.5, так как она является удобной в использовании, многопоточной, свободно распространяемой и мультиплатформенной [6].

Для добавления, изменения и удаления данных написаны пользовательские функции и процедуры, многие из которых организуют поддержание целостности системы. Самое важное ограничение целостности на уровне отдельных полей – это тип данных. Механизм баз данных MySQL обеспечивает богатый спектр типов данных.

Целостность системы поддерживается также при помощи триггеров и ограничений в виде Check [5, 7]. Созданы триггеры, которые проверяют регистр букв в именах, автоматически формируют название группы, полное и сокращенное имя студента, проверяют отметки студента (помимо цифр от одного до десяти, можно вводить «зачтено»/«не зачтено») и т.д. Для безошибочной работы с данными вводим следующие ограничения: проверка форм обучения на имеющиеся в базе, проверка семестров (нельзя ввести число меньше единицы и больше 12), проверка всевозможных дат (например, чтобы дата рождения не была больше текущей или, чтобы студенту не превышало 60), проверка среднего балла на вхождение в отрезок [4.0, 10.0], проверка типа учреждения образования (выбор только из существующих в базе) и т.д.

В результате проделанной работы была спроектирована база данных, при помощи которой можно создать автоматизированную информационную систему направленную на сокращение временных затрат работника деканата, занимающегося ведением различной документации и учетом контингента, минимизировать появление ошибок при составлении документов и автоматизировать формирование отчетов.

ЛИТЕРАТУРА

1. Деканат [Электронный ресурс] // Википедия. – Режим доступа: <https://ru.wikipedia.org/wiki/Декан/>. – Дата доступа: 10.04.2015.
2. Что такое деканат в университете? [Электронный ресурс] // Студенческая жизнь. – Режим доступа: <http://life-students.ru/chto-takoe-dekanat-v-universitete/>. – Дата доступа: 10.04.2015.
3. Малыхина, М. Базы данных: основы, проектирование, использование / М. Малыхина. – М. : ВНУ, 2004. – 512 с.
4. Дейт, К.Д. Введение в системы баз данных / К.Д. Дейт. – М. : Вильямс, 2005. – 1328 с.
5. Боуман, Дж. Практическое руководство по SQL / Дж. Боуман, С. Эмерсон, М. Дарновски. – Изд. 4-е. – М. : Вильямс, 2002. – 352 с.
6. Кузнецов, М.В. MySQL 5 / М.В. Кузнецов, И.В. Симдянов. – СПб. : БХВ-Петербург, 2010. – 1024 с.
7. Грофф, Дж. Полное руководство по SQL / Дж. Грофф, П. Вайнберг. – М. : ВНУ, 2001. – 816 с.

УДК 004

РАЗРАБОТКА ФОРМАЛЬНОГО ЯЗЫКА ОПИСАНИЯ ПОВЕДЕНИЯ АГЕНТА НА БАЗЕ НЕЧЕТКОЙ ЛОГИКИ

В.А. ПЛЯСОВ

(Представлено: канд. техн. наук, доц. Д.О. ГЛУХОВ)

Рассмотрены основные моменты создания формального языка для искусственного интеллекта на базе нечеткой логики.

Нечеткая логика – это логика, в которой мы можем оперировать не только с лог. «0» или лог. «1», а так же со значениями в интервале [0;1]. Рассмотрим классический пример: расстояние от машины до препятствия в обыкновенной логике: мы бы оперировали такими значениями: 1(150м) – далеко, 0(0м) – близко, а нечеткая логика подразумевает нечеткие понятия, такие как очень близко, близко, средняя, далеко и т.д., такие понятия характерны для мышления человека (рис. 1.) [1].