

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ
ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО
ОБРАЗОВАНИЯ
ИВАНОВСКИЙ ГОСУДАРСТВЕННЫЙ ХИМИКО-
ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ

КАФЕДРА ЭКОНОМИКИ И ФИНАНСОВ

Курсовая работа
по дисциплине «Базы данных»
на тему: «Разработка базы данных для приемной комиссии»

Студент: Ермакова Н.Е..

Форма обучения: очная

Курс/группа: 2/47

Номер зачетной книжки: 1131033

Руководитель: к.т.н., Константинов Е.С.

ИВАНОВО, 2014 г.

Оглавление

Задание.....	3
Введение.....	4
1.Подготовительные мероприятия.....	5
1.1.Рассмотрение предметной области.....	5
1.2. Выбор инструмента для реализации.....	11
2.Описание процесса.....	12
2.1.Начальный этап создания проекта.....	12
2.2.Реализация структуры базы.....	15
2.3.Структура базы данных.....	17
Выводы.....	22
Список используемой литературы.....	23

Задание

Требуется создать программную систему, предназначенную для автоматизации приемной компании в ВУЗах. Должен быть сформирован XML-документ (далее импортируемый пакет) , содержащий группу данных: информация о приемных кампаниях; сведения об объеме и структуре приема; заявления абитуриентов; списки заявлений абитуриентов, включенных в приказ; списки лиц, рекомендованных к зачислению.

Группа «информация о приемных кампаниях» должна содержать: идентификатор(ИД) приемной кампании; название; год начала; год завершения; формы обучения; ИД статуса; уровни образования; даты приемной кампании.

Группа «сведения об объеме и структуре приема»: объем приема (контрольные цифры); конкурсные группы.

Группа «заявления абитуриентов»: номер заявления; абитуриент; дата регистрации заявления; признак необходимости общежития; статус заявления; конкурсные группы для заявления; элементы конкурсных групп; условия приема, выбранные абитуриентом; общая льгота; результат вступительных испытаний; документы-приложения к заявлению; приоритет заявления; индивидуальные достижения поступающего.

Группа «списки заявлений абитуриентов, включенных в приказ»: сведения о заявлении; ИД направлений подготовки; ИД формы обучения; ИД источника финансирования; ИД уровня образования; этап зачисления; признак приказа для льготников; признак иностранного гражданина, поступающего по межправительственным соглашениям; ИД конкурсной группы.

Группа «списки лиц, рекомендованных к зачислению»: этап зачисления; заявления, включенные в список рекомендованных.

Подробное описание состава и структуры импортируемых данных смотрите в спецификации сервиса импорта .[1]

Введение

В нынешний век информационных технологий всё большую ценность представляет информация. В связи с этим можно констатировать острую необходимость в построении систем обработки различных документов, а точнее, содержащейся в них информации. Таким образом, информационная система требует создания в памяти электронно-вычислительной машины динамически обновляемой модели внешнего мира с использованием единого хранилища - базы данных. [2]

База данных — представленная в объективной форме совокупность самостоятельных материалов (статей, расчётов, нормативных актов иных подобных материалов), систематизированных таким образом, чтобы эти материалы могли быть найдены и обработаны с помощью ЭВМ. [3]

Задачей данной курсовой работы является сформировать программную часть приложения для хранения базы данных. Подобного рода приложения широко востребованы в Высших учебных заведениях (ВУЗ-ах). Главным достоинством такого приложения является полная автоматизация процессов приемной кампании. В качестве основополагающей цели выступает целесообразное и эффективное осуществление своей деятельности, чему и может поспособствовать приложение базы данных конкретной организации.

1. Подготовительные мероприятия

1.1. Рассмотрение предметной области

Предметной областью приложения является сфера образования, а в частности необходимо построить программную систему, предназначенную для приема абитуриентов в ВУЗ.

В состав системы автоматизированного взаимодействия данных входят следующие методы:

1. Метод импорта данных;
2. Метод получения результатов импорта данных;
3. Метод удаления данных;
4. Метод получения результата удаления данных;
5. Метод предоставления сведений о проверках импортированных заявлений;
6. Метод валидации импортируемого пакета;
7. Метод получения информации об образовательном учреждении;
8. Метод получения списка справочников системы;
9. Метод получения списка элементов справочника;
10. Метод, возвращающий тестовые ответы сервиса автоматизированного взаимодействия;
11. Метод импорта одного заявления;
12. Метод проверки одного заявления;
13. Метод получения части сведений по ОУ.

В данной курсовой работе рассмотрим метод импорта данных, так как он является основой будущей программы. Импорт данных из информационных систем образовательных учреждений в Систему осуществляется посредством сервиса импорта данных. Сервис импорта содержит 2 группы методов: вспомогательные(рис.1) и методы обмена данными(рис.2).

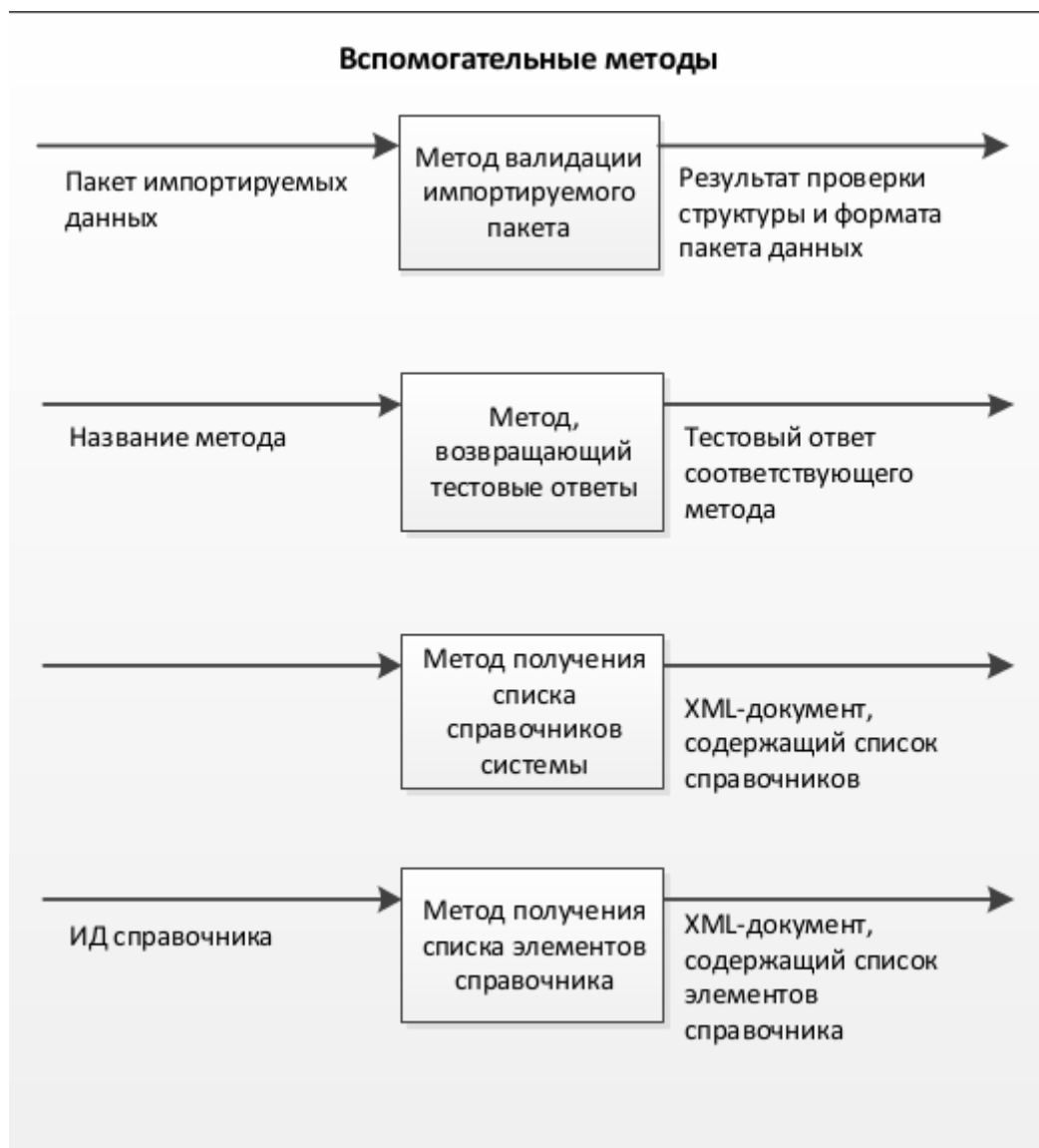


Рис.1 Структура вспомогательных методов обмена данными

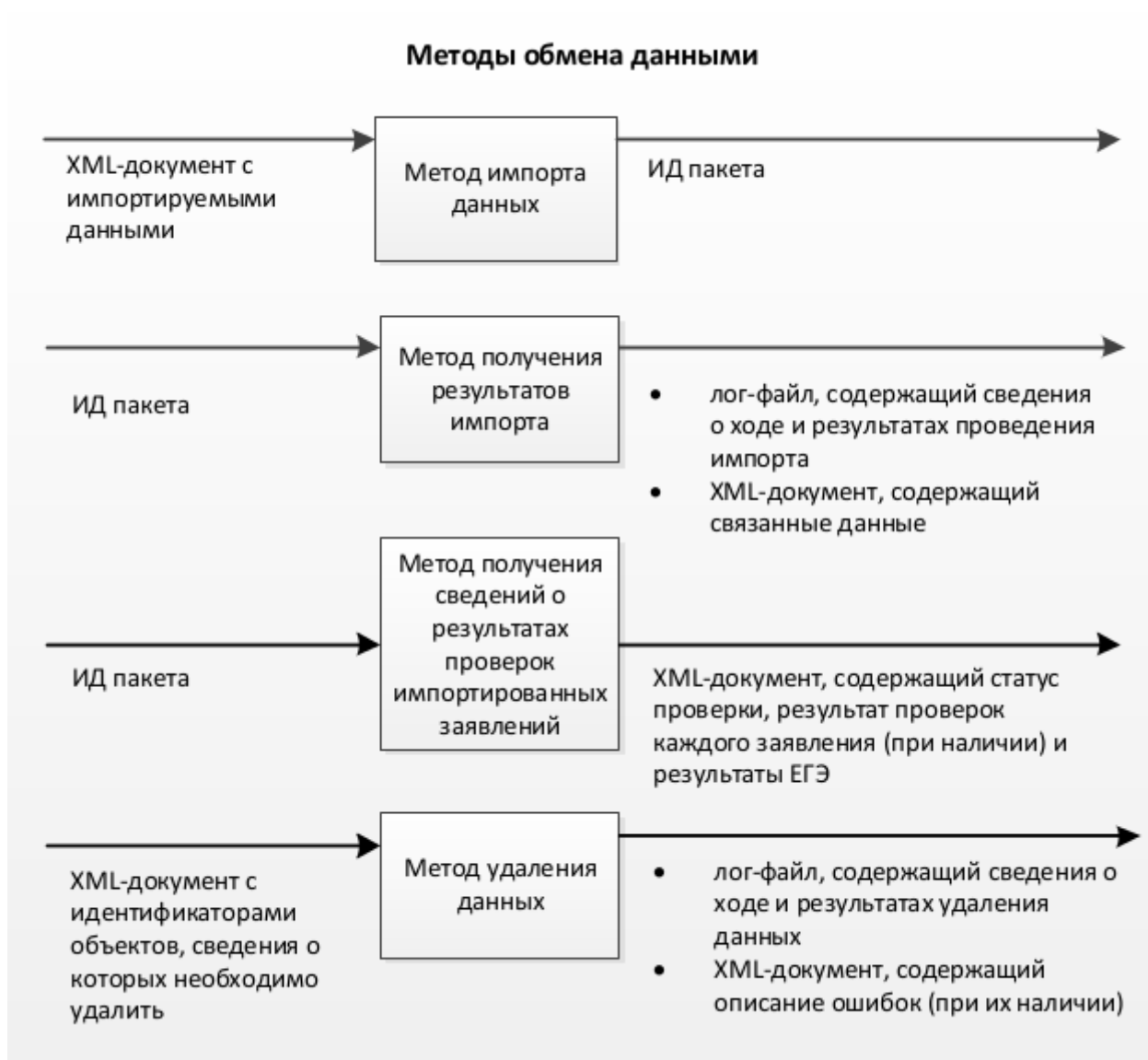
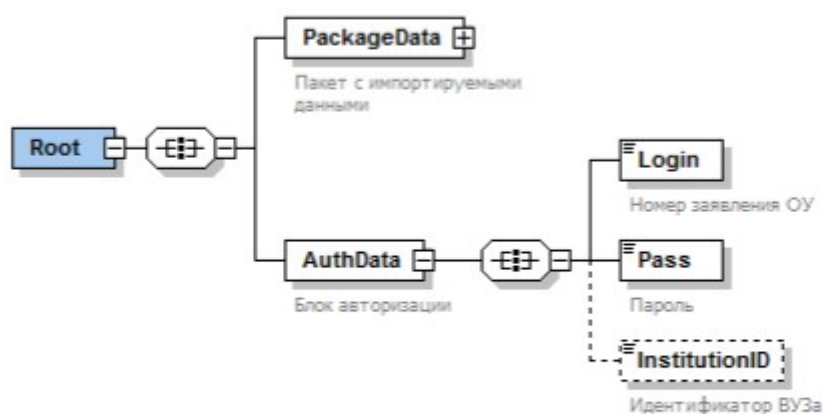


Рис.2 Структура методов обмена данными

Входные данные, в курсовой работе, представлены в виде пакета (XML-файла), содержащие импортируемые данные(рис.3).

Структура корневого элемента «Root»:



Структура элемента «PackageData» (Пакет с импортируемыми данными):

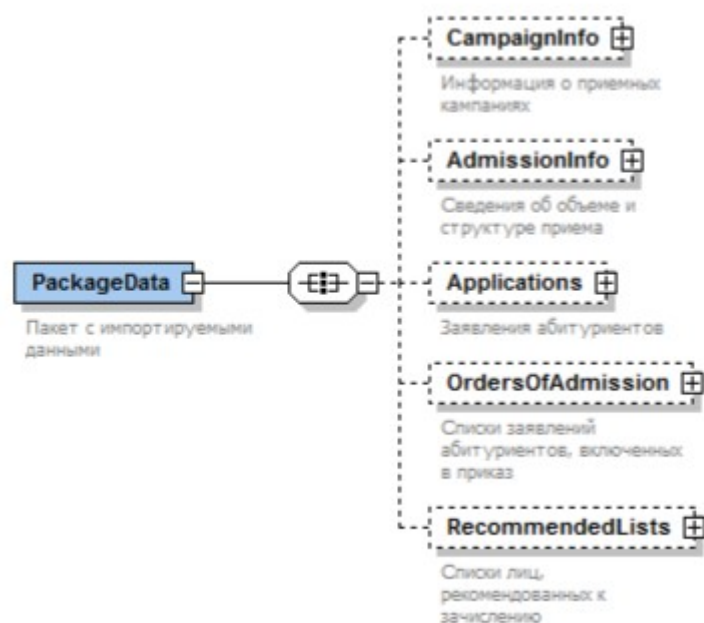
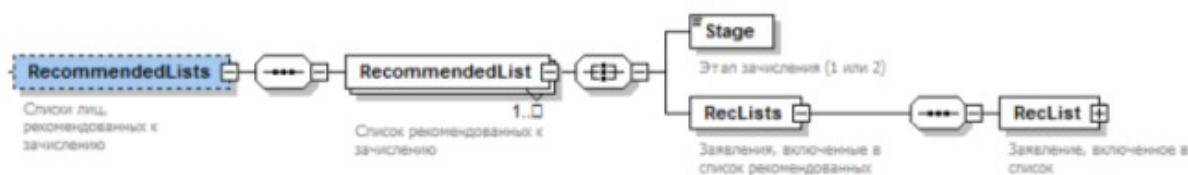


Рис.3 Описание структуры файла, содержащего импортируемые данные

Выходные данные: XML-файл, содержащий идентификатор импортированного пакета(рис.4).

2.1.1.5 Структура элемента «RecommendedLists» (Списки лиц, рекомендованных к зачислению):



Структура элемента «RecList» (Заявление, включенное в список рекомендованных):

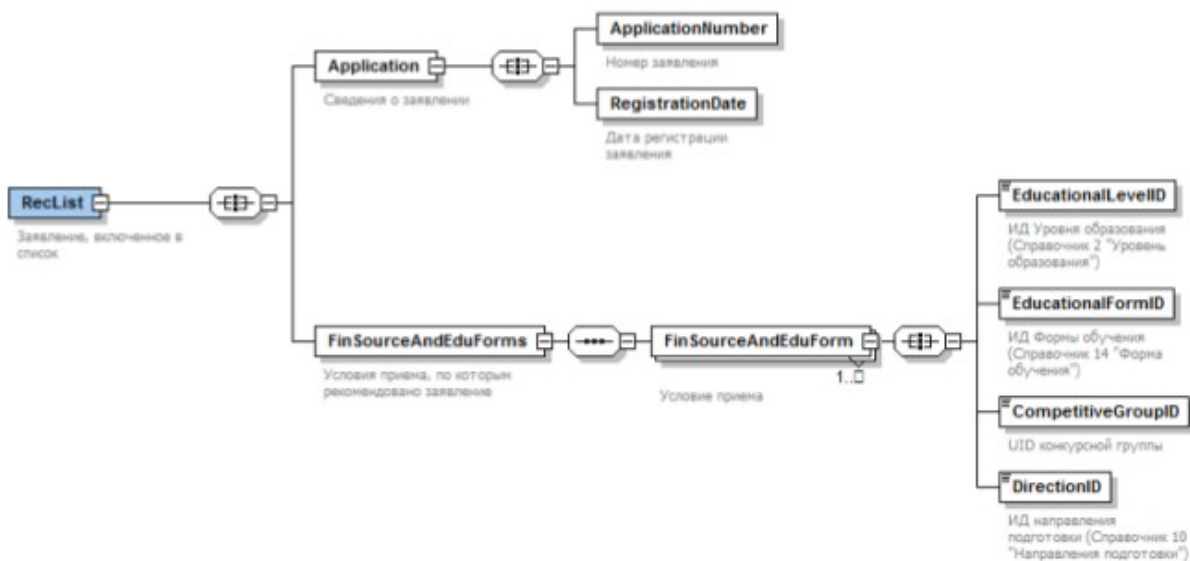


Рис.4 Описание структуры

файла, содержащего идентификатор импортированного пакета

В итоге выделяются следующие основные таблицы:

- информация о приемных кампаниях;
- сведения об объеме и структуре приема;
- заявления абитуриентов;
- списки заявлений абитуриентов, включенных в приказ;
- списки лиц, зарекомендованных к зачислению

со стороны входных данных. А со стороны выходных

- этапы зачисления;
- заявление, включенное в список рекомендованных.

Каждой таблице присваивается свой уникальный идентификационный номер, он же ID и первичный ключ. Некоторые таблицы являются промежуточными,

их задача проводить связь между двумя другими, давая им возможность выступать в нескольких вариациях. Так же стоит отметить, что некоторые таблицы играют роль «связующих звеньев», - они же таблицы-справочники, предназначенные для хранения часто вводимых данных. Например, если взять наш случай, будет логично прописать в таблице «информация о приемных кампаниях» поле «название», но вместо того, чтобы постоянно вводить вручную эти данные, можно завести соответствующие справочники и подставлять данные из них.

1.2. Выбор инструмента для реализации

В качестве инструмента реализации использовалась программа Platypus Platform.

Platypus Platform — программная платформа для построения приложений различного уровня сложности. Основная область применения — создание информационных систем уровня предприятия, однако приложения на базе платформы могут применяться и в других областях, например, государственном управлении.

Назначение платформы Platypus — ускорение разработки приложений и обеспечение при этом максимальной автоматизации рутинных операций. Для типовых задач предоставляются готовые отработанные решения. Платформа скрывает от прикладного разработчика технические детали реализации, давая возможность сконцентрироваться на предметной области.

В Platypus Platform используются два прикладных языка программирования:

- для доступа к данным применяется язык SQL;
- для написания программной логики применяется язык JavaScript .

При создании приложений используются визуальные средства разработки:

- визуальный редактор структуры базы данных (БД) приложения;
- визуальный редактор запросов к базе данных;
- визуальный редактор форм;

Таким образом, можно сказать, что данное приложение вполне подходит для реализации нашей задачи.

2. Описание процесса

2.1. Начальный этап создания проекта

После установки Platypus Platform открываем программу. По истечению процесса запуска появляется стартовое окно с пустыми параметрами (рис. 5).

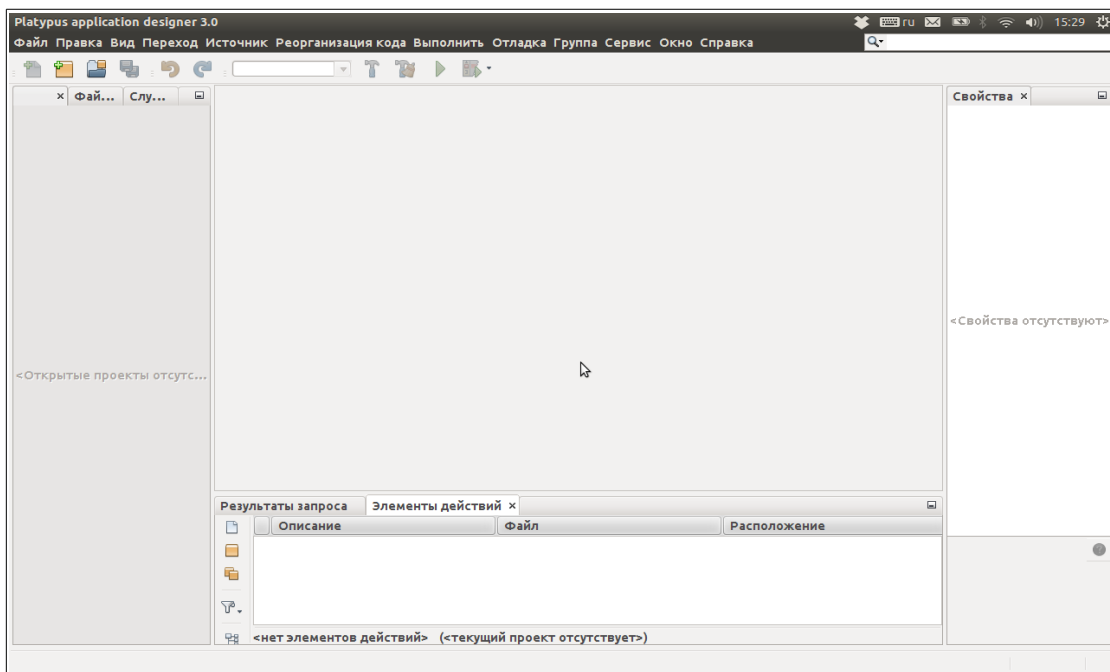


Рис. 5 Рабочее окно Platypus Platform

Необходимо создать новый проект по следующему принципу: выбрать пункт «Файл» верхнего меню, затем «Создать проект». После этого появляется соответствующее окно (рис. 6).

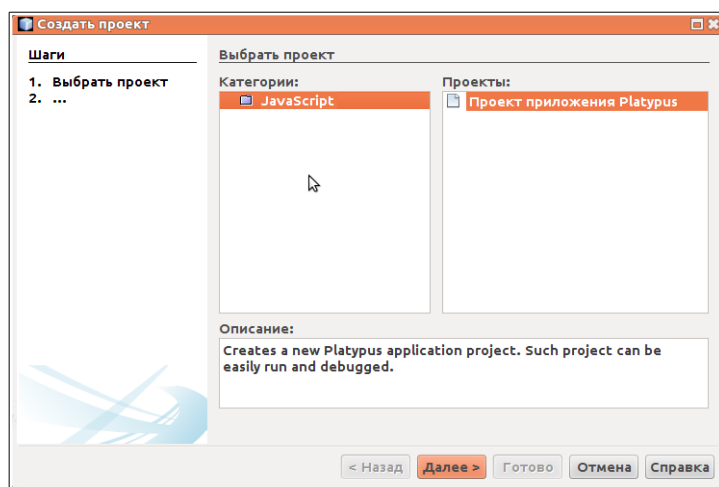


Рис. 6 Создание проекта

Нажимаем «Далее», указываем название нашего проекта и место в

файловой системе, в котором он будет храниться (рис. 7).

Создать проект

Шаги

1. Выбрать проект
- 2. Имя и расположения**
3. Настройки соединения с базой данных

Имя и расположения

Имя папки проекта:

Расположение проекта:

Папка проекта:

Название проекта:

< Назад Далее > Готово Отмена Справка

Рис. 7 Название проекта

После ещё одного нажатия на кнопку «Далее», мы попадаем в область привязки нашего проекта непосредственно к базе данных (рис. 8). Выбираем пункт «Построить JDBC Url» и указываем значения, представленные на рис. 9.

Создать проект

Шаги

1. Выбрать проект
2. Имя и расположения
- 3. Настройки соединения с базой данных**

Настройки соединения с базой данных

JDBC Url:

Имя пользователя базы данных:

Пароль базы данных:

⚠ Проверьте соединение с помощью кнопки "Проверить соединение".

< Назад Далее > Готово Отмена Справка

Рис. 8 Привязка к базе данных

Построить JDBC Url

Server:

Host:

Port:

Database:

Рис. 9 Настройка JDBC Url

Применяем заданные настройки кнопкой «ОК», и всё, что остается — прописать любые имя пользователя и пароль.

Итак, наш проект готов (рис. 10). Однако мы можем заметить, что в самом низу программы прописано следующее: «Не удалось подключиться к базе данных». Без взаимодействия с базой наша дальнейшая работа невозможна, и для того, чтобы все-таки подключиться к базе данных, следует зайти во вкладку «Службы», выбрать в пункте «Базы данных» значение «H2» и кликнуть по нему ПКМ — выбрать «Запустить» (рис. 11).

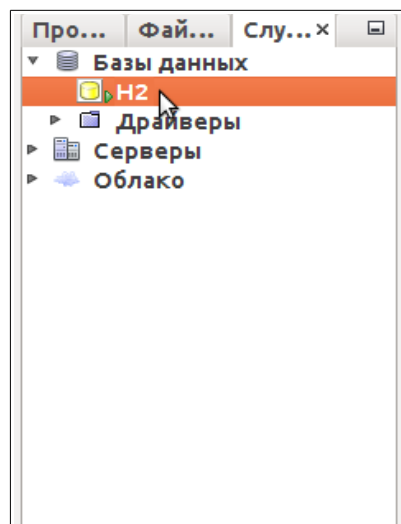
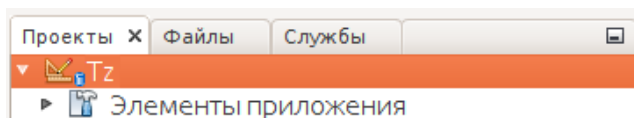


Рис. 10 Отображение проекта

Рис. 11 Запуск базы H2

Возвращаясь во вкладку с нашим проектом, также щелкаем по его названию ПКМ и выбираем «Соединиться с базой данных». После нескольких секунд замечаем в строке состояния оповещение о том, что соединение установлено.

2.2. Реализация структуры базы

Структура базы данных — «фундамент» разработки приложения, потому как и графическая его составляющая, и способность выполнять определенные запросы базируется непосредственно на структуре базы данных. В разделе «Рассмотрение предметной области» были приведены основные схемы-диаграммы базы данных приложения, теперь необходимо «перенести» это в Platypus Platform.

Лишь только специфичная процедура создания структуры базы в данной среде разработки представляет собой определенную сложность. Но, если разобраться, - алгоритм довольно прост: необходимо щелкнуть ПКМ по «Элементам приложения», что внутри левого окна с нашим проектом, выбрать «Новый» и «Диаграмма структуры базы данных» (рис. 12).

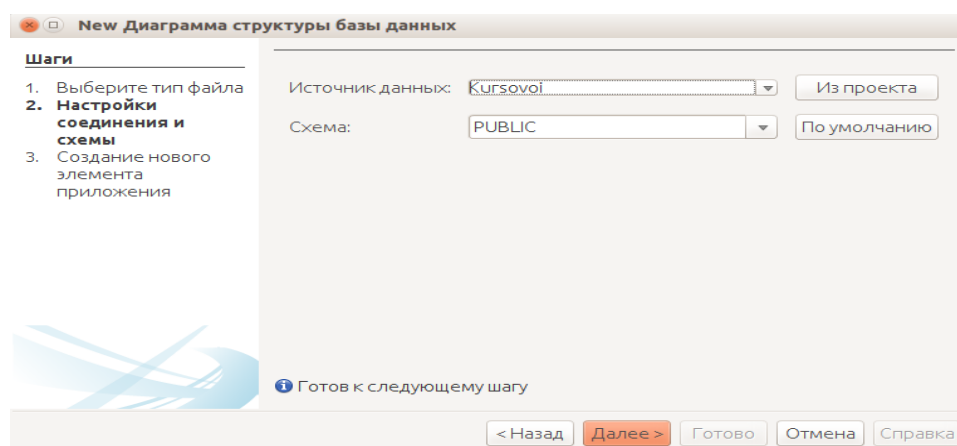


Рис. 12 Создание диаграммы структуры базы данных

Оставляя настройки по умолчанию и выбрав «Далее», остается прописать названия нашей диаграммы и имя проекта (рис. 13).

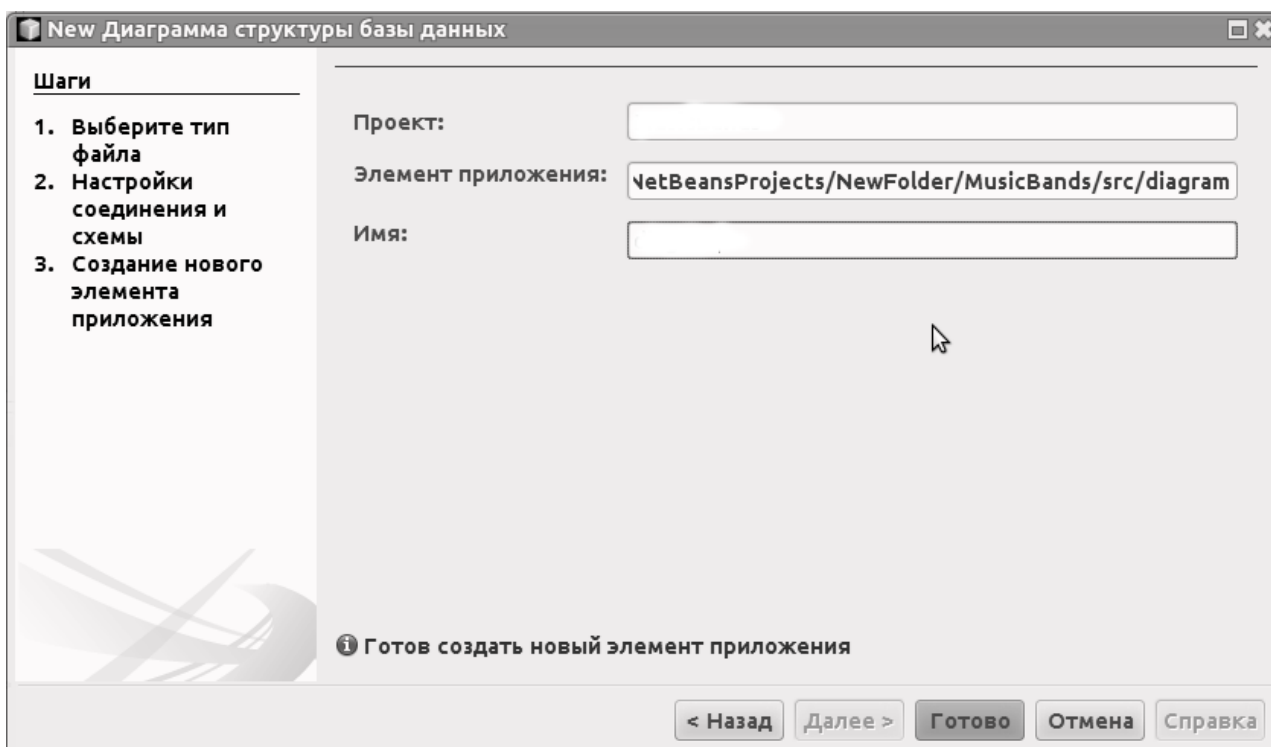


Рис. 13 Название диаграммы

Теперь есть область, в которой можно начать создавать наши таблицы. Замечаем вверху, левее выбора масштаба значок таблицы с плюсом: щелкаем по близлежащему к нему треугольнику и выбираем «Создать таблицу». Появляется окно для ввода названия таблицы. Нажимаем «ОК» и замечаем, что таблица действительно появилась в окне диаграммы структуры. Теперь необходимо добавить в неё поля. Для этого щелкаем по значку звёздочки (это есть «Создать новое поле/параметр»), что располагается правее ранее выбранного значка таблицы, и указываем в данном поле определенные его свойства (рис. 14). Например, прописываем название «Name», подтверждаем его тип (строка), размер, масштаб и возможность иметь значение «ноль».

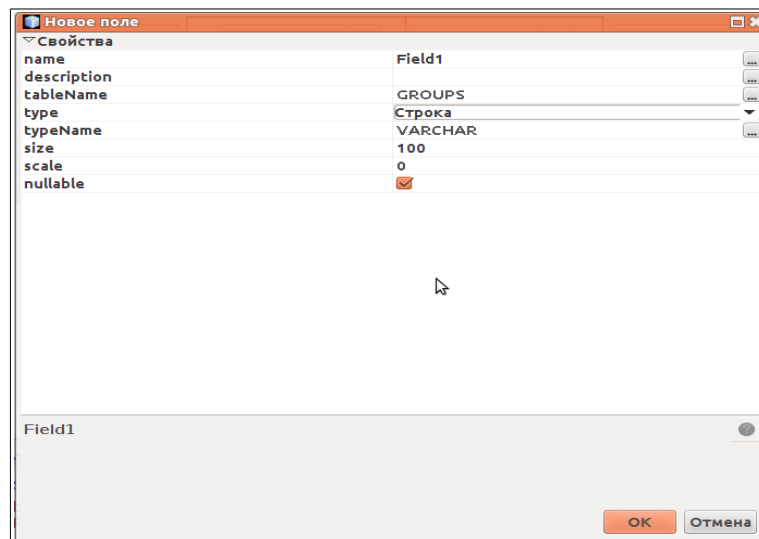


Рис. 14 Создание нового поля

Аналогичным образом добавляем остальные другие таблицы со всеми необходимыми полями.

Остается лишь связать первичные и внешние ключи таблиц. Для этого необходимо просто «схватить» нужное поле (внешний ключ) и перенести на то, на которое оно должно ссылаться (первичный ключ).

2.3. Структура базы данных

Всё курсовая работа состоит из шести диаграмм структуры базы данных(рис.15).Её исходники расположены https://github.com/ISUCT/DBCOURSEWORK_2014/tree/Ermakova_N_E.

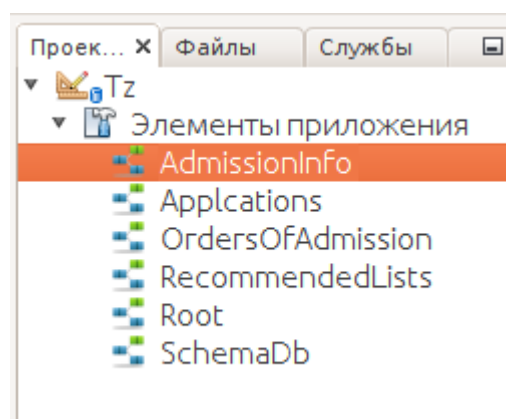


Рис.15Диаграммы

Самая главная, в которой взаимосвязаны все диаграммы - это Root (корневой элемент)(рис.16). В ней содержится как входные, так и выходные

данные.

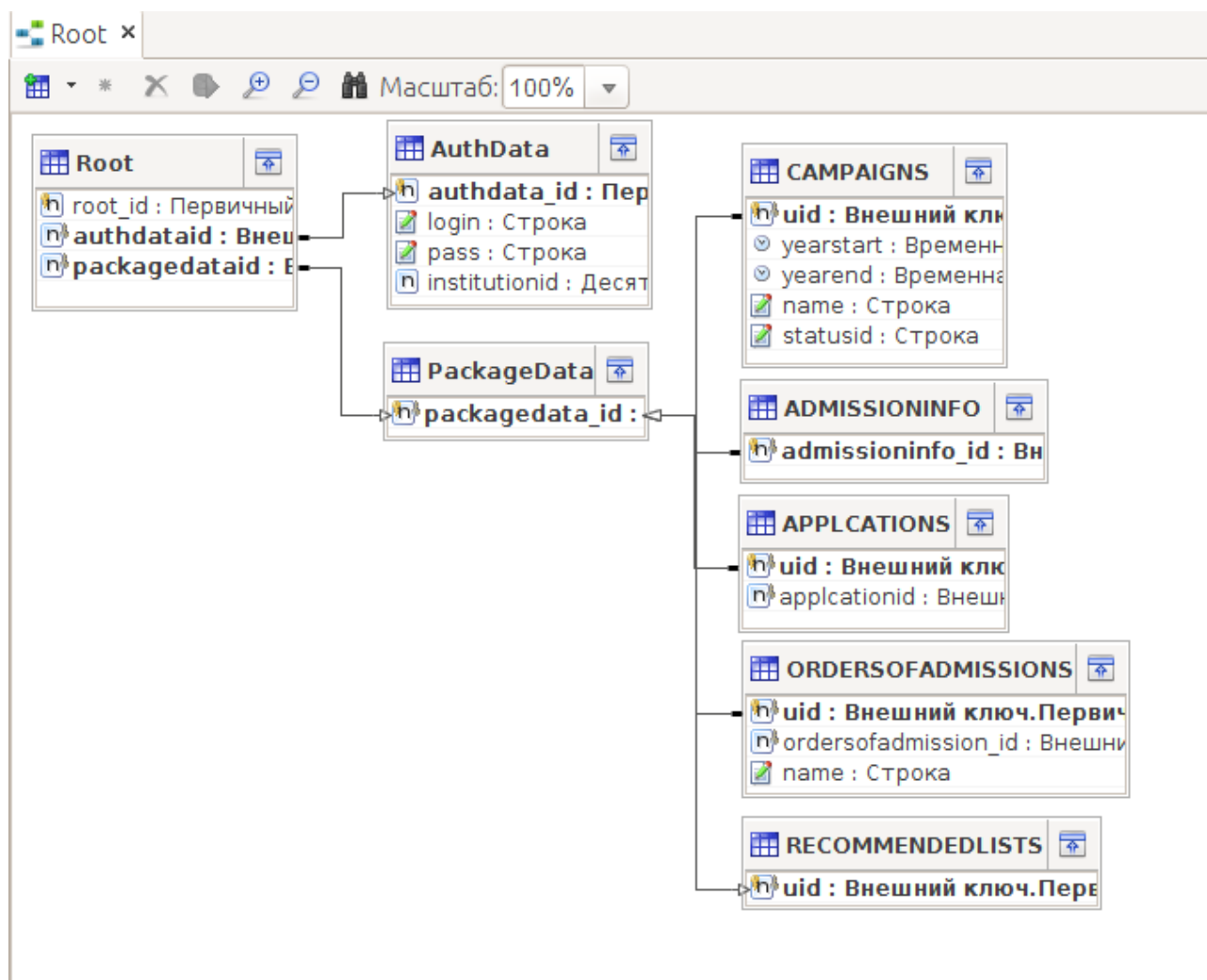


Рис.16 Структура Root

Схема «Campaign» (информация о приемных компаниях) в ней содержится вся та структура, и даже больше, которая описана в техническом задании (рис.17)

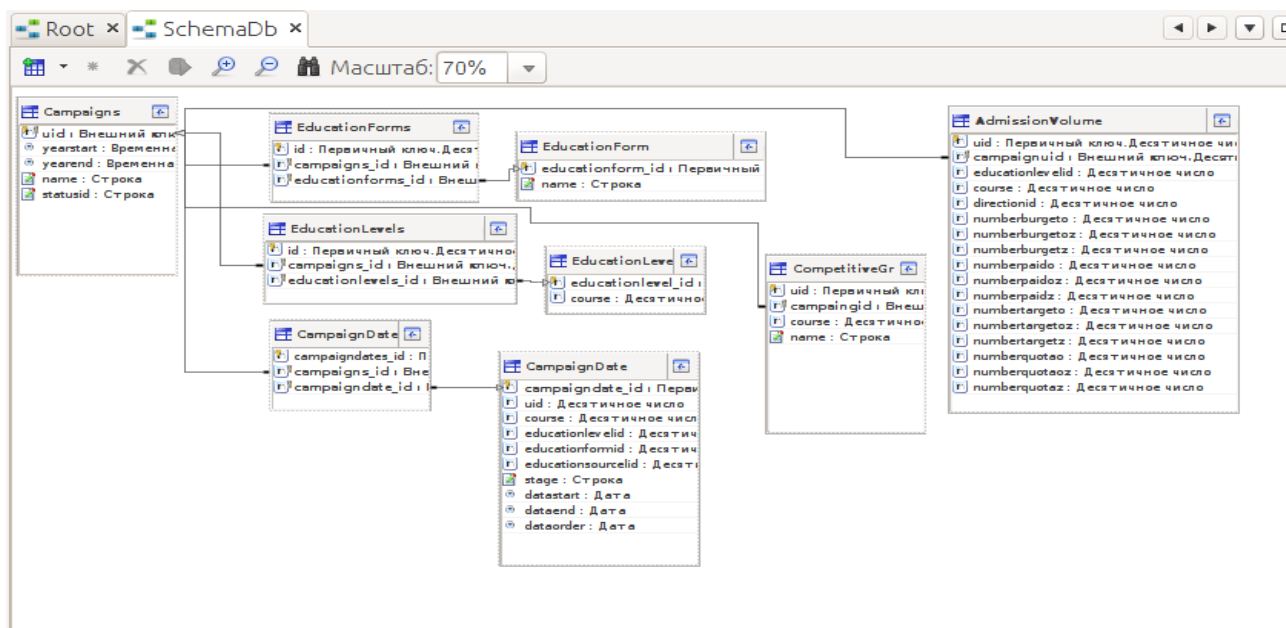


Рис.17 Структура «Campaign»

Диаграмма «AdmissionInfo» (Сведения об объеме и структуре приема) содержит все заданные поля и выглядит так (рис.18)

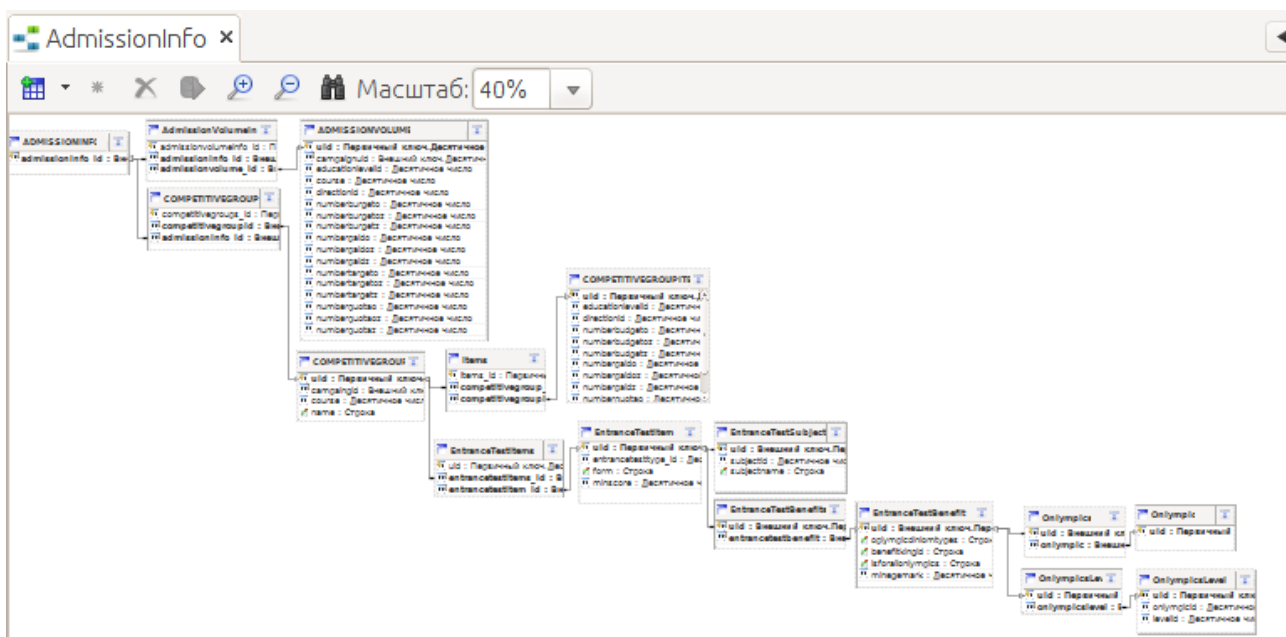


Рис.18 Структура «AdmissionInfo»

Структура элемента «Applications» (Заявления абитуриентов) по объему и значению считается одной из самых важных (рис.19)

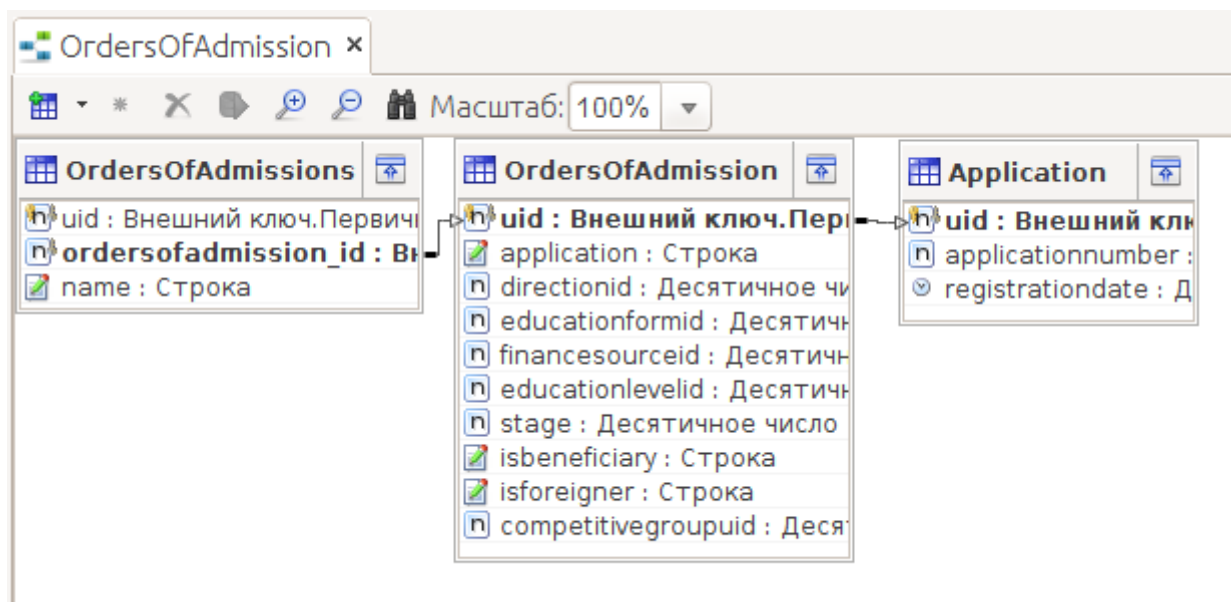
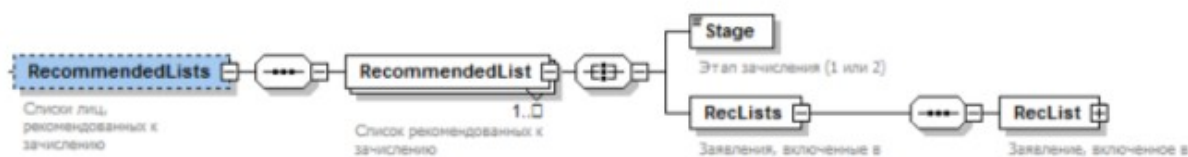


Рис.20 Структура элемента «OrdersOfAdmission»

Структура элемента «RecommendedLists» (Списки лиц, рекомендованных к зачислению) относится к выходным данным (рис.21)

2.1.1.5 Структура элемента «RecommendedLists» (Списки лиц, рекомендованных к зачислению):



Структура элемента «RecList» (Заявление, включенное в список рекомендованных):

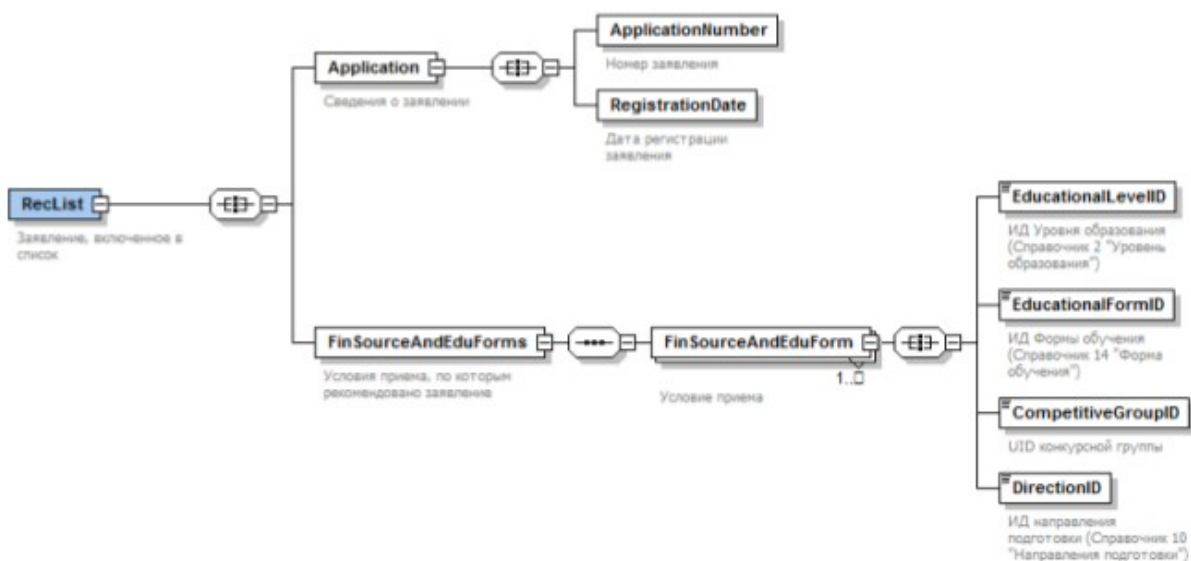


Рис.21 Структура элемента «RecommendedLists»

Выводы

Таким образом, в результате проделанной работы был создан пакет с импортными данными он является основой будущей программы. В дальнейшем программа приобретет вид, который будет понятен и прост для любого пользователя.

Главной нашей задачей было создать программную часть приложения для хранения базы данных. В качестве основного требования моделировалась ситуация, когда обычный человек, имеющий слабое представление о том, как работает данное реализованное приложение, что представляют из себя SQL-запросы и как выполняется код, написанный на JavaScript, мог с легкостью воспользоваться функционалом получившейся программы и повысить эффективность своей деятельности за счет упрощения решения задач, имеющих в своей основе работу с большим количеством разного рода информации.

Список используемой литературы

- 1 Нормативная спецификация
- 2 Зеленков, Ю. А. Введение в базы данных : учеб. пособие [электронный ресурс] <http://www.mstu.edu.ru/study/materials/zelenkov/intro.html>
- 3 Сайт «Википедия» [электронный ресурс] http://ru.wikipedia.org/База_данных