

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий  
Кафедра «Прикладная математика и информатика»

02.03.03 МАТЕМАТИЧЕСКОЕ ОБЕСПЕЧЕНИЕ И АДМИНИСТРИРОВА-  
НИЕ ИНФОРМАЦИОННЫХ СИСТЕМ

ТЕХНОЛОГИЯ ПРОГРАММИРОВАНИЯ

### **БАКАЛАВРСКАЯ РАБОТА**

на тему Разработка программного модуля «Электронная зачетная книжка»

Студент \_\_\_\_\_ А.М. Чуриков \_\_\_\_\_

Руководитель \_\_\_\_\_ Н.И. Лиманова \_\_\_\_\_

**Допустить к защите**

Заведующий кафедрой к.т.н., доцент, А.В. Очеповский \_\_\_\_\_

« \_\_\_\_\_ » \_\_\_\_\_ 20 \_\_\_\_\_ г.

Тольятти 2016

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий  
Кафедра «Прикладная математика и информатика»

УТВЕРЖДАЮ  
Зав. кафедрой «Прикладная ма-  
тематика и информатика»  
\_\_\_\_\_ А.В. Очеповский

« \_\_\_\_ » \_\_\_\_\_ 2016 г.

**ЗАДАНИЕ**  
**на выполнение бакалаврской работы**

Студент Чуриков Андрей Михайлович

1. Тема «Разработка программного модуля «Электронная зачетная книжка»
2. Срок сдачи студентом законченной выпускной квалификационной работы 13.06.2016
3. Исходные данные к выпускной квалификационной работе: техническое задание. требование к средству реализации: BIRT, Oracle Database, JavaScript.
4. Содержание выпускной квалификационной работы (перечень подлежащих разработке вопросов, разделов):  
Введение
  1. Глава 1. Методология создания электронной зачетной книжки
  2. Глава 2. Архитектура информационной среды электронной зачетной книжки
  3. Глава 3. Реализация и тестирование электронной зачетной книжки
  4. Разработка программного модуля
  5. Заключение
  5. Приложения
6. Дата выдачи задания «11» января 2016 г.

Руководитель выпускной квалифи-  
кационной работы \_\_\_\_\_

\_\_\_\_\_ Н.И. Лиманова

Задание принял к исполнению \_\_\_\_\_

\_\_\_\_\_ А.М. Чуриков

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий  
Кафедра «Прикладная математика и информатика»

УТВЕРЖДАЮ  
Зав. кафедрой «Прикладная ма-  
тематика и информатика»  
\_\_\_\_\_ А.В. Очеповский

«\_\_\_\_\_» \_\_\_\_\_ 2016 г.

**КАЛЕНДАРНЫЙ ПЛАН**  
**выполнения бакалаврской работы**

Студента Чурикова Андрея Михайловича  
по теме Разработка программного модуля «Электронная зачетная книжка»

Наименование раздела работы	Плановый срок выполнения раздела	Фактический срок выполнения раздела	Отметка о выполнении	Подпись руководителя
Поиск и исследование литературы по теме выпускной квалификационной работы	19.02.2016	19.02.2016	выполнено	
Написание первой главы БР	22.02.2016	22.02.2016	выполнено	
Написание второй главы БР	11.03.2016	11.03.2016	выполнено	
Реализация разработки сервиса	10.04.2015	10.04.2015	выполнено	
Тестирование и отладка сервиса	17.04.2015	17.04.2015	выполнено	
Написание третьей главы	19.04.2016	19.04.2016	выполнено	
Представление выпускной квалификационной работы на кафедру	16.05.2016	16.05.2016	выполнено	
Подготовка доклада и графического материала	20.05.2016	20.05.2016	выполнено	

Предварительная защита	25.05.2016	25.05.2016	выполнено	
Сдача пояснительной записки ВКР	19.06.2016	19.06.2016	выполнено	

Руководитель выпускной квалификационной работы

Н.И. Лиманова

Задание принял к исполнению

А.М. Чуриков

## Аннотация

Тема: Разработка программного модуля «Электронная зачетная книжка»

Актуальность темы данной работы заключается в том, что появится возможность удаленно просматривать успеваемость студента.

Целью данной бакалаврской работы является создание электронной зачётной книжки для портала ТГУ.

Для достижения поставленной цели в ходе работы были поставлены и решены следующие задачи:

- проанализировать подобные сервисы;
- проанализировать и описать технологии для реализации;
- разработать модель базы данных;
- реализовать электронную зачетную книжку;
- провести тестирование.

Работа состоит из введения, трех глав, заключения, списка использованной литературы и приложений.

В работе использованы современные системы и технологии проектирования аналитических систем: технология структурного моделирования, концепция JavaScript, BIRT, MySQL, Oracle Database и др.

Первая глава посвящена анализу предметной области, обзору и обоснованию выбора средств для разработки.

Вторая глава посвящена архитектуре информационной среды электронной зачетной книжки.

Третья глава посвящена реализации и тестированию электронной зачетной книжки.

В заключении сформулированы основные выводы, которые были сделаны в процессе написания бакалаврской работы, описаны результаты практической реализации проекта.

В приложениях представлены фрагменты программного кода и другие дополнительные материалы.

## Оглавление

Введение.....	3
Глава 1 Методология создания электронной зачетной книжки.....	5
1.1 Электронная зачетная книжка .....	5
1.2 Анализ подобных сервисов.....	7
1.3 Форма и внешний вид электронной зачетной книжки.....	9
Глава 2 Архитектура информационной среды электронной зачетной книжки .....	18
2.1 Выбор архитектуры информационной среды и инструментальных средств реализации электронной зачетной книжки .....	18
2.2 Описание модели данных.....	22
2.3 Настройка и подготовка программного обеспечения для реализации электронной зачетной книжки.....	26
Глава 3 Реализация и тестирование электронной зачетной книжки .....	31
3.1 Описание реализации электронной зачетной книжки .....	31
3.2 Тестирование электронной зачетной книжки .....	48
Заключение .....	54
Список использованной литературы.....	55
Приложение А Код таблицы V_STUD_GRADE_BOOK .....	57
Приложение Б Код таблицы V_STUD_COUNT_REPORT.....	62
Приложение В Код таблицы V_STUD_SEMESTERS.....	66
Приложение Г Код таблицы V_STUD_TOTAL_MARKS_CURS .....	67
Приложение Д Код таблицы V_STUD_TOTAL_MARKS_EXAM .....	69
Приложение Е Код таблицы V_STUD_TOTAL_MARKS_FACULT_EXAM ..	71
Приложение Ж Код таблицы V_STUD_TOTAL_MARKS_FACULT_ZACH ..	75
Приложение З Код таблицы V_STUD_TOTAL_MARKS_NIR.....	79
Приложение И Код таблицы V_STUD_TOTAL_MARKS_ZACH .....	83

## Введение

Глобальная информатизация и компьютеризация общества с каждым годом ощущаются все сильнее. Практически во всех сферах нашей жизни, мы можем увидеть влияние тех или иных информационных технологий. Машиностроение, прикладные науки, гуманитарные науки, бухгалтерский учет, документооборот – во всех этих сферах все чаще начинают отказываться от рукописных и печатных документов в пользу электронных.

Преимущества электронного документооборота неоспоримы. Вы можете практически мгновенно переместить нужные бумаги практически по всему земному шару. Раньше нам пришлось бы использовать многочисленную сеть посредников и телеграфы, чтобы утвердить или опровергнуть какую-либо информацию. В современном обществе подобная проблема отсутствует.

Большинство предприятий и учебных заведений переходят на электронный документооборот, где классические документы играют роль лишь вспомогательных инструментов контроля и учета. Не стоит на месте и Тольяттинский Государственный Университет.

Центр Новых Информационных технологий (сокращенно ЦНИТ) подразделение Тольяттинского Государственного Университета, отвечающее за разработку, внедрение и поддержку различных сервисов, необходимых для повышения эффективности обучающего процесса.

Преимущества электронного варианта перед бумажным:

- ее невозможно потерять;
- ее книжку не нужно носить с собой;
- просмотреть электронную зачетку можно практически из любого места, где есть Интернет;
- ее невозможно привести в негодность
- она никогда не "испортится".

Объектом исследования является процесс создания электронной зачетной книжки.

Предметом исследования является электронная зачетная книжка.

Целью ВКР является создание электронной зачётной книжки для портала ТГУ.

Актуальность работы заключается в том, что появиться возможность удаленно просматривать успеваемость студента во время обучения.

Для достижения цели необходимо выполнить следующие задачи:

- проанализировать подобные сервисы;
- проанализировать технологии для реализации электронной зачетной книжки;
- разработать модель базы данных;
- реализовать электронную зачетную книжку;
- провести тестирование.

В ходе выполнения выпускной квалификационной работы будет создан информационный сервис, который предоставляет актуальную информацию об успеваемости студента.

Выпускная квалификационная работа состоит из введения, трех глав, заключения, списка литературы и приложений.

Во введении описывается актуальность проводимого исследования, формируется цель, и ставятся задачи, которые необходимо решить для достижения цели. В первой главе проводится анализ подобных сервисов, а также сформулирована актуальность разработки. Во второй главе проводится разработка и реализация программного продукта. В третьей главе приводится тестирование и анализ реализованного продукта. В заключении приводятся основные выводы по работе, достигнутые в ходе выполнения выпускной квалификационной работы.



## **Глава 1 Методология создания электронной зачетной книжки**

### **1.1 Электронная зачетная книжка**

Электронная зачетная книжка – информационный сервис, который предоставляет актуальную информацию об успеваемости студента. С функционалом, позволяющим ознакомиться с содержащейся информацией. Электронная зачетная книжка студента включает в себя список дисциплин из текущего РУП студента на каждый семестр, разбитый по формам контроля и видам контрольных мероприятий.

В данный момент, большая часть активно-развивающихся высших учебных заведений переходит на электронные версии зачетных книжек. Это может служить доказательством их эффективности в плане понижения затрат и повышения эффективности образования. Поэтому актуальность подобного перехода достаточно велика – во времена технического прогресса и повальной информатизации нельзя стоять на месте, особенно, зная, что внедряемая технология достаточно эффективна. Модернизировать и реставрировать нужно не только здания, но и информационное пространство, иначе в один прекрасный момент можно оказаться в ситуации, когда для здоровой конкуренции у тебя не будет должных сил и инструментов.

В электронной зачетке, как и в обычной зачетке, должны отображаться результаты обучения студента в учебном заведении, полученные студентом в процессе обучения.

Доступ к электронной зачетной книжке планируется организовать через отдельную вкладку в личном кабинете студента на портале ТГУ.

Центр новых информационных технологий (сокращенно ЦНИТ) подразделение Тольяттинского Государственного Университета, отвечающее за разработку, внедрение и поддержку различных сервисов, необходимых для повышения эффективности обучающего процесса студента.

Сообщество молодых квалифицированных специалистов,

использующее свои навыки и умения для создания, внедрения и поддержки множества инновационных сервисов и технологий.

В сферу деятельности ЦНИТ входит:

- проанализировать подобные сервисы;
- проанализировать и описать технологии для реализации;
- управление ИТ-услугами;
- создание и поддержка инфраструктуры, систем, сервисов;
- разработка и модернизация программного обеспечения;
- развитие персонала.

Управление ИТ-услугами в ТГУ обеспечивается автоматизированной системой «ТОР», позволяющей регистрировать запросы пользователей следить за устранением сбоев, запрашивать оценку пользователей о качестве исполнения запросов. Основную нагрузку на первой линии поддержки несут на себе диспетчеры отдела менеджмента качества и оптимизации бизнес-процессов.

Создание основных узлов корпоративной компьютерной сети (ККС) ТГУ было закончено к концу 2003 года. С этого момента и до настоящего времени сетевая инфраструктура постоянно развивается. Ее основу представляет оборудование компании D-Link, а сетевые сервисы реализованы на базе программного обеспечения с открытым кодом. Поддержка и развитие ККС входят в обязанности отдела сетевого и системного администрирования.

Возможности ККС обеспечивают ее пользователям предоставление и поддержку без лимитного канала выхода в Интернет, а также целого ряда базовых ИТ-сервисов, таких как веб-серверы и прокси-серверы, электронная почта, серверы точного времени и резервного копирования, файловый сервер, серверы антивирусной защиты и поддержки пользователей, серверы доменов пользователей.

Также в ведении отдела сетевого и системного администрирования находится поддержка и модернизация базовых ИТ-сервисов, организованных

на базе операционных систем Linux и Microsoft Windows Server.

В мае 2013 года создан отдел сопровождения корпоративной ERP-системы. В обязанности отдела входят сопровождение, поддержка и администрирование системы, используемой в ТГУ, а также проведения работ в рамках её функционала.

ЦНИТ осуществляет разработку программного обеспечения в целях автоматизации основных процессов ТГУ (в первую очередь образовательного и административного), а также различных вспомогательных процессов. Разработкой и модернизацией программного обеспечения занимается отдел разработки информационных систем.

Одной из новых внедренных технологий является электронная зачетная книжка. Преимущества электронного варианта перед бумажным:

- электронную зачетную книжку не нужно носить с собой;
- ее невозможно потерять;
- посмотреть данные, внесенные в нее можно практически из любого места, где есть доступ в Интернет;
- ее невозможно привести в негодность, и она никогда не "испортится".

## **1.2 Анализ подобных сервисов**

В настоящее время некоторые из учебных заведений переходят на более современную систему образования. Качество образования является одной из основных проблем в стране. Министерство образования старается проводить различного рода нововведения в образовательный процесс, чтобы повысить эффективность обучения и задать школьникам и студентам стимул в учебе. Такие нововведения создаются с целью создать у учащихся должное усвоение государственных образовательных стандартов. Стандартная зачетная книжка студента за последнее время также претерпела изменения. Из привычного бумажного вида в некоторых вузах страны она приобрела электронный вид. Со временем все больше вузов переходит на электронные

зачетки. Некоторые из вузов на данный момент полностью отказались от привычных нам, бумажных зачетов в пользу электронного варианта.

Стоит отметить что подобные сервисы сегодня используются не только в высших учебных заведениях, но и в некоторых школьных и средне специальных учебных учреждениях. В электронных зачетках содержится информация о студенте и его успеваемости за все время обучения в учебном заведении. Примеры других электронных зачетов показаны на рисунках 1.1 и 1.2.


Зачётная книжка									
					 Зачётные кредиты: 115.5 Зачётные часы(общие): 4320 Зачётные часы(аудиторные): 1056				
2013/2014 уч. год    2012/2013 уч. год    2011/2012 уч. год									
Дисциплина	Итоговое испытание	Часы		Кредитов	Оценка		Пересдачи	Последняя сдача	Преподаватель
		общ.	аудит.		10 балльная	5 балльная			
<b>2 модуль</b>									
Система высшего образования РФ:политика,право,управление,экономика	Экзамен	216	128	6	8	отлично	0	06-09-12	Т.В. Абакина, А.А. Вавилова
Научно-исследовательский семинар	Зачет	216	76	6	8	отлично	0	04-09-12	И.В. Абакина, К.В. Зиньковский
Методология научных исследований в менеджменте	Экзамен	162	64	4.5	5	удовлетворительно	0	03-09-12	Э.Лемперг-Шепель
Научно-исследовательская практика	Зачет	216	0	6	9	отлично	0	31-07-12	К.В. Зиньковский
Курсовая работа	Экзамен	144	0	4	8	отлично	0	28-06-12	Г.Н. Константинов
Технологии разработки и принятия управленческих решений	Экзамен	162	64	4.5	6	хорошо	0	25-06-12	Н.Б. Филинов
Стратегии в менеджменте: стратегическое управление в вузе	Экзамен	162	64	4.5	8	отлично	0	21-06-12	Г.Н. Константинов
Сравнительный анализ систем высшего образования: прикладные аспекты интернационализации ВУЗов	Зачет	162	64	4.5	8	отлично	0	19-06-12	С.М.Яковлев
<b>1 модуль</b>									
Современные исследования высшего образования	Зачет	162	64	4.5	6	хорошо	0	05-09-12	М. Джонсон

Рисунок 1.1 – Электронная зачетная книжки Высшей школы экономики

2011/2012 учебный год									
Курсовая работа	Экзамен	0	0	5	5	удовлетворительно	0	30.06.12	
Транспортное право	Экзамен	108	40	3	8	отлично	0	29.06.12	Плахин А.А.
Физическая культура	Зачет	54	54	0,25	10	отлично	0	28.06.12	Стрельникова А.А.
Корпоративное право	Зачет	108	40	3	6	хорошо	0	26.06.12	Чеховская С.А.
Правовое регулирование электронного бизнеса в мировой экономике	Зачет	108	40	3	8	отлично	0	25.06.12	Иванов А.Ю.
Юридические лица в международном частном праве	Зачет	108	40	3	8	отлично	0	22.06.12	Иванов А.Ю.
Арбитражный процесс	Экзамен	108	48	3	7	хорошо	0	20.06.12	Бекшьева Д.И.
Конкурентное право	Экзамен	108	48	3	5	удовлетворительно	0	18.06.12	Сушкевич А.Г.
Земельное право	Экзамен	108	48	3	6	хорошо	0	30.03.12	
Коммерческое право	Зачет	108	40	3	8	отлично	0	29.03.12	Плахин А.А.
Российское предпринимательское право	Экзамен	162	72	4,5	5	удовлетворительно	0	28.03.12	Пирогова Е.С.
Иностранный язык (английский)	Экзамен	0	0	3	8	отлично	0	27.03.12	
Иностранный язык (английский)	Экзамен	108	72	3	8	отлично	0	27.03.12	
Криминалистика	Экзамен	108	72	3,5	10	отлично	0	24.03.12	Нестеров А.В.
Структурирование международных сделок	Зачет	162	40	0	8	отлично	0	23.03.12	
Банкротство хозяйствующих субъектов:гражданско-правовые и уголовно-правовые аспекты	Зачет	108	40	3	8	отлично	0	21.03.12	Пирогова Е.С.
Юридическая психология	Зачет	108	44	3	6	хорошо	0	20.03.12	Федотов А.В.
Физическая культура	Зачет	54	54	0,25	10	отлично	0	30.12.11	Стрельникова А.А.
Вексельное право	Зачет	54	20	1,5	8	отлично	0	29.12.11	Вишневский А.А.
Международное частное право	Экзамен	108	72	3	6	хорошо	0	27.12.11	
Строительное право	Зачет	108	40	3	6	хорошо	0	24.12.11	Фогельсон Ю.Б.
Налогообложение бизнеса и иностранных инвестиций в России	Зачет	108	24	0	10	отлично	0	22.12.11	Дмитриева И.
Криминология	Зачет	108	48	3	10	отлично	0	27.10.11	Федотов А.В.

Рисунок 1.2 – Пример электронной зачетной книжки

### 1.3 Форма и внешний вид электронной зачетной книжки

Форма электронной зачетной книжки студента должна соответствовать формам документов, представленным ниже.

<div style="border: 1px solid black; width: 100px; height: 100px; margin: 0 auto;"> <p style="text-align: center;">Фото</p> </div>	<p>Министерство образования и науки РФ ФГБОУ ВПО «Тольяттинский государственный университет»</p> <p><b>ЗАЧЕТНАЯ КНИЖКА № _____</b></p> <p>_____</p> <p>(фамилия, имя, отчество студента (последнее – при наличии))</p> <p>Код, направление подготовки (специальность) _____</p> <p>Структурное подразделение _____</p> <p>Зачислен приказом от _____ 20__ г. № _____</p> <p>Руководитель организации, осуществляющей образовательную деятельность _____</p> <p>Руководитель структурного подразделения _____</p>
--	--

Рисунок 1.3 – Форма 1 электронной зачетной книжки

На 1 форме электронной зачетной книжки отображается основная информация о студенте.

1-й семестр _____ / _____ учебного года		<b>ПЕРВЫЙ</b>			
<b>РЕЗУЛЬТАТЫ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ (ЭКЗАМЕНЫ)</b>					
№ п/п	Наименование дисциплины (модуля), раздела	Общее кол-во час. / з.ед.	Оценка	Дата сдачи экзамена	Фамилия преподавателя
1.	Математика				
2.	Русский язык и культура речи				
3.	Информатика				
4.					
5.					
6.					
7.					
8.					
9.					
10.					

Рисунок 1.4 – Форма 2 электронной зачетной книжки

На 2 форме электронной зачетной книжки отображаются данные о экзаменах студента за конкретный семестр.

<b>КУРС</b>					
(Фамилия, Имя, Отчество студента)					
<b>РЕЗУЛЬТАТЫ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ (ЗАЧЕТЫ)</b>					
№ п/п	Наименование дисциплины (модуля), раздела	Общее кол-во час. / з.ед.	Оценка	Дата сдачи зачета	Фамилия преподавателя
1.	Биология				
2.	Иностранный язык				
3.					
4.					
5.					
6.					
7.					
8.					
9.					
10.					

Количество экзаменов в текущем семестре \_\_\_\_\_; из них сдано \_\_\_\_\_, не сдано \_\_\_\_\_

Количество зачетов в текущем семестре \_\_\_\_\_; из них сдано \_\_\_\_\_, не сдано \_\_\_\_\_

**Рисунок 1.5 – Форма 3 электронной зачетной книжки**

На 3 форме электронной зачетной книжки отображаются данные о зачетах студента за конкретный семестр.

<b>КУРСОВЫЕ</b>		
№ п/п	Наименование дисциплин (- ны), модуля	Тема курсовой работы (проекта)
1.		
2.		
3.		
4.		
5.		
6.		
7.		
8.		
9.		
10.		

**Рисунок 1.6 – Форма 4 электронной зачетной книжки**

<b>РАБОТЫ (ПРОЕКТЫ)</b> _____ (Фамилия, Имя, Отчество студента)			
Семестр	Оценка	Дата сдачи	Фамилия преподавателя

Количество курсовых работ (проектов) \_\_\_\_\_; из них сдано \_\_\_\_\_, не сдано \_\_\_\_\_.

**Рисунок 1.7 – Форма 5 электронной зачетной книжки**

На 4 и 5 форме электронной зачетной книжки отображаются данные о курсовых работах студента.

<b>ПРАКТИКА</b>					
№ п/п	Наименование вида практики	Се-местр	Место проведения практики	В качестве кого работал (должность)	Ф.И.О. руководителя практики от предприятия (организации, учреждения)
1.					
2.					
3.					
4.					
5.					
6.					
7.					
8.					
9.					
10.					

**Рисунок 1.8 – Форма 6 электронной зачетной книжки**





<b>РАБОТА</b>					
(Фамилия, Имя, Отчество студента)					
№ п/п	Вид научно-исследовательской работы	Семестр	Оценка	Дата сдачи	Фамилия преподавателя
10.					
11.					
12.					
13.					
14.					
15.					
16.					
17.					
18.					

Рисунок 1.11 – Форма 9 электронной зачетной книжки

На 8 и 9 форме электронной зачетной книжки отображаются данные о научно-исследовательских работах студента.

<b>ГОСУДАРСТВЕННЫЕ</b>			<b>ЭКЗАМЕНЫ</b>	
			(Фамилия, Имя, Отчество студента)	
№ п/п	Наименование дисциплины (модулей)	Дата сдачи экзамена	Оценка	Фамилии председателя и членов Государственной экзаменационной комиссии
1.				
2.				
3.				
4.				
5.				
6.				
7.				
8.				
9.				
10.				

Рисунок 1.12 – Форма 10 электронной зачетной книжки

На 10 форме электронной зачетной книжки отображаются данные

государственных

экзаменах

студента.

<b>Выпускная квалификационная работа</b>	
(Фамилия, Имя, Отчество студента)	
Форма выпускной квалификационной работы:	_____
Тема	_____
_____	
Руководитель:	_____
	(Фамилия И.О.)
Дата защиты «__» _____ 20__ г.	
Оценка:	_____
Председатель ГЭК	_____
Члены ГЭК:	_____
	_____

Рисунок 1.13 – Форма 11 электронной зачетной книжки

На 11 форме электронной зачетной книжки отображаются данные о ВКР студента.

<b>Решением Государственной экзаменационной комиссии</b>	
от _____ 20__ г. протокол № _____	
студенту _____	(Фамилия, Имя, Отчество студента)
Присвоена квалификация _____	(наименование)
_____	
Председатель ГЭК	_____
Члены ГЭК:	_____
	_____
Выдан диплом _____ № _____ от «__» _____ 20__ г.	

Рисунок 1.14 – Электронная зачетная книжка. Форма 16

В первой главе в результате анализа предметной области было выявлено, что некоторые из учебных заведений переходят на более современную систему образования, стараются внедрять различного рода нововведения, одним из которых и является электронная зачетная книжка, с помощью которой можно просмотреть свои успехи в учебе. Определены

основные требования к информационному сервису.

## **Глава 2 Архитектура информационной среды электронной зачетной книжки**

### **2.1 Выбор архитектуры информационной среды и инструментальных средств реализации электронной зачетной книжки**

Для того, чтобы реализовать электронную зачетную книжку необходимо решить, где будет храниться вся информация об успеваемости студентов.

В Тольяттинском государственном университете уже несколько лет успешно используется система ERP Галактика. Корпорация «Галактика» является одним из ведущих разработчиков ERP-систем в СНГ, предлагая свои решения для различных отраслей экономики. Работа с высшими учебными заведениями занимает особое место в структуре бизнеса корпорации «Галактика» и строится по четырем основным направлениям:

- разработка проектов комплексной информатизации университетов, помощь в построении единого информационного пространства вуза, анализ существующих программно-аппаратных комплексов вузов и выработка рекомендаций по их модернизации;

- разработка проектов комплексной информатизации университетов, помощь в построении единого информационного пространства вуза, анализ существующих программно-аппаратных комплексов вузов и выработка рекомендаций по их модернизации;

- настройка и внедрение в учебных заведениях комплексной информационной системы ERP-класса «Галактика Управление Вузом»;

- внедрение в вузах систем электронного документооборота, создание веб-порталов университетов, настройка систем аналитической отчетности;

- ведение федеральной образовательной программы «Галактика и Вузы», в рамках которой студенты получают знания в области современных

информационных технологий, а вузы получают наборы необходимых методических материалов.

Решение «Галактика Управление Вузом» создано на базе системы ERP Галактика. Это современное комплексное IT-решение для осуществления информационной поддержки эффективного управления учебным процессом и всей финансово-хозяйственной деятельностью высшего учебного заведения. Решение основано на модульном принципе, реализовано под использование одной из СУБД на выбор: Oracle, MS SQL Server, Pervasive.SQL, PostgreSQL.

В ТГУ в качестве базы данных используется СУБД компании Oracle под ОС Linux.

Oracle Database – объектно-реляционная система управления базами данных компании Oracle. База данных Oracle это масштабируемая, производительная, функциональная СУБД для решения широкого спектра задач, доступная под различные архитектуры процессоров и для различных операционных систем. В настоящее время заказчикам предлагается несколько редакций Oracle Database, удовлетворяющих бизнес и информационным потребностям организаций различного уровня и бюджета. Все редакции основаны на единой архитектуре, полностью совместимы друг с другом и отличаются лицензионными ограничениями. Кроме того, существуют дополнительные программные продукты, расширяющие возможности Oracle Database при решении конкретных прикладных задач[1-4].

База данных разделена на логические структурные единицы, называемые табличными пространствами. Они используются для объединения хранимых в них логически связанных структур. Для хранения информации, содержащейся в логических структурах табличного пространства, создается один или несколько файлов данных[6].

Все данные для электронной зачетной книжки будут храниться в БД Oracle.

Также необходимо решить, как именно нужно выводить информацию о успеваемости студента на портале.

В ТГУ начиная с 2009 года успешно используется система корпоративной отчетности BIRT. BIRT – основанный на технологии Eclipse, это продукт с открытым исходным кодом, который предоставляет разработчикам широкие возможности для создания и интегрирования отчетов в свои приложения, в том числе и web-приложения. BIRT имеет 2 основных компонента для разработки: визуальный редактор отчетов для создания отчетов и компонент времени выполнения для генерации отчетов, которые могут быть развернуты в любой среде Java. BIRT также содержит технологию построения графиков, которая полностью интегрирована в дизайнер отчетов и может использоваться автономно для интеграции графиков в приложение[20].

Проект BIRT поддерживается сообществом пользователей в Центре разработчиков BIRT и разработчиками на странице BIRT Project сайта Eclipse.org.

BIRT отчет состоит из 4 основных частей: данных, преобразования, представления и бизнес-логики.

Базы данных могут предоставлять свои данные для составления отчетов BIRT. В BIRT есть поддержка XML, Web Services, JDBC и плоских файлов. Данные из базы данных предоставляются путем написания SQL запросов в базу данных. Для того чтобы подключить BIRT к базе данных Oracle необходимо скачать и установить JDBC драйвер. Скачать этот драйвер можно на официальном сайте Oracle.

JDBC Driver — платформенно-независимый промышленный стандарт взаимодействия Java-приложений с различными СУБД, реализованный в виде пакета `java.sql`, который входит в состав Java SE. Позволяет получать доступ к базе данных и выполнять к ней SQL запросы.

Преобразование данных: BIRT способен легко сортировать, обобщать, группировать и фильтровать полученные данные из базы. BIRT позволяет



выполнять сложные операции, такие как проценты от общей суммы, группировка на суммы, и многое другое. В то время как база данных может выполнять лишь некоторые из этих функций [16-18].

**Бизнес-логика:** данные в базе редко бывают структурированы так, как это необходимо для создания отчета. Многие данные требуют некоторой бизнес-логики для преобразования исходных данных в информацию, полезную и понятную для пользователя. Преобразовать данные в понятный для конечного пользователя вид можно с помощью JavaScript.

**Представление:** после того, как все данные будут готовы, у разработчика появляются широкие возможности для представления данных пользователю. Текст, таблицы, графики, диаграммы и многое другое.

BIRT способен генерировать отчеты в таких форматах как: PDF, Excel, HTML, Power Point, Word. Причем сделать это можно без каких-либо изменений в программном коде проекта [16-18].

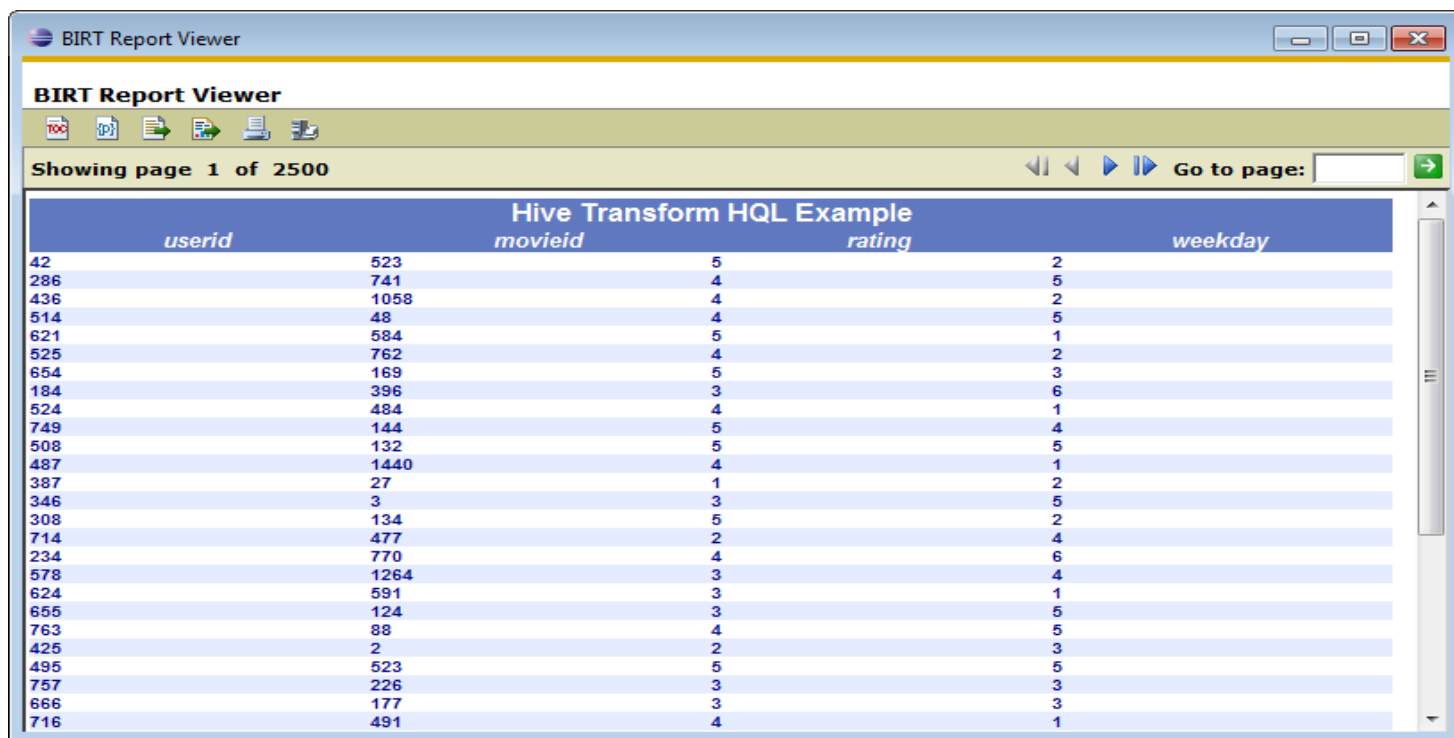
Первый релиз BIRT состоялся в 2004 году. С тех пор он стал одним из самых известных продуктов для создания отчетов. Проект имеет поддержку большого сообщества, есть различные статьи, учебники и примеры программного кода и шаблонов.

BIRT генерирует отчеты в виде web-документа, позволяет динамически адаптировать их к размерам веб-браузера, в котором они отображаются.

BIRT обладает достаточным функционалом для создания сложных отчетов, которые необходимо просматривать в web-среде. Обилие различных материалов для обучения, открытая документация и большое сообщество поддержки облегчают задачу освоения этого инструмента, а сам продукт периодически обновляется, в последние версии добавляют новые функции, что снижает риск его морального устаревания [16-18].

BIRT Viewer – это плагин для Eclipse, который также доступен в виде отдельного приложения, основанный на технологиях AJAX [16-18]. С помощью BIRT Viewer возможно просматривать отчеты с реальными данными. Он поддерживает различные интерактивные функции, такие как

нумерация страниц, содержание отчета, оглавление, и т.д. В настоящее время этот плагин полностью интегрирован в IDE Eclipse. Пример отображения отчета в BIRT Viewer показан на рисунке 2.1.



The screenshot shows the BIRT Report Viewer interface. At the top, it says 'BIRT Report Viewer' and 'Showing page 1 of 2500'. Below this is a table with the title 'Hive Transform HQL Example'. The table has four columns: 'userid', 'movieid', 'rating', and 'weekday'. The data is as follows:

userid	movieid	rating	weekday
42	523	5	2
286	741	4	5
436	1058	4	2
514	48	4	5
621	584	5	1
525	762	4	2
654	169	5	3
184	396	3	6
524	484	4	1
749	144	5	4
508	132	5	5
487	1440	4	1
387	27	1	2
346	3	3	5
308	134	5	2
714	477	2	4
234	770	4	6
578	1264	3	4
624	591	3	1
655	124	3	5
763	88	4	5
425	2	2	3
495	523	5	5
757	226	3	3
666	177	3	3
716	491	4	1

Рисунок 2.1 – Пример отображения отчета в BIRT Viewer

## 2.2 Описание модели данных

В ходе работы было выяснено, что для того, чтобы собрать всю необходимую информацию о студенте и его успеваемости требуется работать с очень большим количеством реляционных таблиц. В следствии этого приходилось бы писать огромные SQL запросы, в которые сложно было бы вносить изменения и искать в них ошибки. Поэтому для удобства работы была создана новая схема, в которой были созданы 9 виртуальных таблиц, в которых хранилась вся необходимая информация. Физическая модель базы данных показана на рисунке 2.2.

V_STUD_COUNT_REPORT	
FIO	varchar2(60)
FCPERSONS	char(16)
EKZ_1	number
ZACH_1	number
EKZ_2	number
ZACH_2	number
EKZ_3	number
ZACH_3	number
EKZ_4	number
ZACH_4	number
EKZ_5	number
ZACH_5	number
EKZ_6	number
ZACH_6	number
EKZ_7	number
ZACH_7	number
EKZ_8	number
ZACH_8	number
EKZ_9	number
ZACH_9	number
EKZ_10	number
ZACH_10	number
EKZ_11	number
ZACH_11	number
EKZ_12	number
ZACH_12	number
FAC_E	number
FAC_Z	number
CURS	number
PRACT_1	number
PRACT_2	number
PRACT_3	number
PRACT_4	number
PRACT_5	number
PRACT_6	number
PRACT_7	number
PRACT_8	number
PRACT_9	number
PRACT_10	number
PRACT_11	number
PRACT_12	number
NIR	number
DATE_GOS	varchar2
DATE_GOS	number
ORDER_GOS	varchar2

V_STUD_GRADE_BOOK	
FCPERSONS	char(16)
USTUD	char(16)
GROUP_NAME	varchar2(200)
FFIO	varchar2(60)
FNMB	varchar2(20)
LOGIN	varchar2(255)
FCODEPROF	varchar2(100)
SPEC	varchar2(100)
INST	varchar2(255)
STGR	varchar2(200)
HEAD	varchar2(65)
BUP	char(16)
ORDER_NUMBER	varchar2(20)
ORD_DATE	date(7)
TERM	float(22, 126)
FYEARENT	number(5)

V_STUD_SEMESTERS	
FCPERSONS	char(16)
FFIO	varchar2(60)
FWSEMESTR	number(5)
FWCOURSE	number(5)
FWYEARED	number(5)
ORD_DATE	date(7)
ORD_NUM	varchar2(20)
ORD_TXT	varchar2(100)
FTERM	varchar2(22, 126)

V_STUD_TOTAL_MARKS_ZACH	
FCPERSONS	char(16)
MARK	varchar2(255)
FWMARK	number(5)
DIS	varchar2(200)
FCDIS	char(16)
FWSEMESTR	number(5)
RUP	char(16)
FWYEARED	number(5)
FWHOURS	number(5)
ZET	number(24, 8)
TIP	varchar2(200)
TEACHER	varchar2(60)
TEM	varchar2(255)
FNUMDOC	varchar2(20)
DATEMARK	varchar2(10)
VED	char(16)
FWCOURSE	number(5)
RN	number

V_STUD_TOTAL_MARKS_FACULT_NIR	
FCPERSONS	char(16)
MARK	varchar2(255)
FWMARK	number(5)
DIS	varchar2(200)
FCDIS	char(16)
FWSEMESTR	number(5)
RUP	char(16)
FWYEARED	number(5)
FWHOURS	number(5)
ZET	number(24, 8)
TIP	varchar2(200)
TEACHER	varchar2(60)
TEM	varchar2(255)
FNUMDOC	varchar2(20)
DATEMARK	varchar2(10)
VED	char(16)
FWCOURSE_E	numeric(5)
RN	number

V_STUD_TOTAL_MARKS_FACULT_ZACH	
FCPERSONS	char(16)
MARK	varchar2(255)
FWMARK	number(5)
DIS	varchar2(200)
FCDIS	char(16)
FWSEMESTR	number(5)
RUP	char(16)
FWYEARED	number(5)
FWHOURS	number(5)
ZET	number(24, 8)
TIP	varchar2(200)
TEACHER	varchar2(60)
TEM	varchar2(255)
FNUMDOC	varchar2(20)
DATEMARK	varchar2(10)
VED	char(16)
FWCOURSE_E	numeric(5)
RN	number

V_STUD_TOTAL_MARKS_EXAM	
FCPERSONS_E	char(16)
MARK_E	varchar2(255)
FWMARK_E	number(5)
DIS_E	varchar2(200)
FCDIS_E	char(16)
FWSEMESTR_E	number(5)
RUP_E	char(16)
FWYEARED_E	number(5)
FWHOURS_E	number(5)
ZET_E	number(24, 8)
TIP_E	varchar2(200)
TEACHER_E	varchar2(60)
TEM_E	varchar2(255)
FNUMDOC_E	varchar2(20)
DATEMARK_E	varchar2(10)
VED_E	char(16)
FWCOURSE_E	number(5)
RN_E	number

V_STUD_TOTAL_MARKS_EXAM	
FCPERSONS_E	char(16)
MARK_E	varchar2(255)
FWMARK_E	number(5)
DIS_E	varchar2(200)
FCDIS_E	char(16)
FWSEMESTR_E	number(5)
RUP_E	char(16)
FWYEARED_E	number(5)
FWHOURS_E	number(5)
ZET_E	number(24, 8)
TIP_E	varchar2(200)
TEACHER_E	varchar2(60)
TEM_E	varchar2(255)
FNUMDOC_E	varchar2(20)
DATEMARK_E	varchar2(10)
VED_E	char(16)
FWCOURSE_E	number(5)
RN_E	number

V_STUD_TOTAL_MARKS_CURS	
FCPERSON_E	char(16)
MARK	varchar2(255)
FWMARK	number(5)
DIS	varchar2(200)
FCDIS	char(16)
FWSEMESTR	number(5)
RUP	char(16)
FWYEARED	number(5)
FWHOURS	number(5)
ZET	number(24, 8)
TIP	varchar2(200)
TEACHER	varchar2(60)
TEM	varchar2(255)
FNUMDOC	varchar2(20)
DATEMARK	varchar2(10)
VED	char(16)
FWCOURSE	number(5)
RN	number

Рисунок 2.2 – Схема базы данных

На рисунке 16 представлена логическая модель базы данных. Обращение к данным осуществляются через обычные SQL запросы.

Стоит отметить что в схеме все данные представлены не в виде обычной, привычной таблицы в SQL базе, а в виде виртуальной таблицы(представлении). В отличие от обычных реляционных таблиц базы данных, представление не является самостоятельной частью хранимых данных в базе данных. Содержимое виртуальной таблицы(представления) динамически вычисляется на основании данных, которые содержатся в реальных таблицах. Изменение данных в реальной таблице базы данных немедленно отражается в содержимом всех представлений, построенных на основании этой таблицы.

Преимущества использования представлений:

– возможность более гибкой настройки прав доступа к данным за счет того, что права даются не на таблицу, а на представление;

- позволяет разделить логику хранения данных и программного обеспечения. Можно менять структуру данных, не затрагивая программный код, для этого нужно лишь создать представления, аналогичные таблицам, к которым раньше обращались приложения;

- удобство в использовании за счет автоматического выполнения таких действий как доступ к определенной части строк или столбцов, получение данных из нескольких таблиц и их преобразование с помощью различных функций;

- представления скрывают от прикладной программы сложность запросов и саму структуру таблиц базы данных. Когда прикладной программе требуется таблица с определённым набором данных, она делает запрос из подготовленного представления. При этом даже если для получения этих данных требуется чрезвычайно сложный запрос, сама программа этого запроса не содержит [1-4].

Принцип работы с представлениями практически идентичен принципу работы с обычными таблицами в базе данных. Данные из представления можно получить с помощью обычного SQL запроса, например, `SELECT * from MY_VIEW_EXAMPLE`. Таким образом имя представления может находиться в SQL-запросе на месте имени таблицы (в предложении FROM). SQL запросы, описывающие работы всех виртуальных таблиц находятся в приложениях А, Б, В, Г, Д, Е, Ж, З, И. Также при использовании таблиц мы не создаем новые записи в базе данных, а значит не увеличиваем ее размер. Поскольку SQL запрос, выбирающий данные представления, зафиксирован на момент его создания, СУБД получает возможность применить к этому запросу оптимизацию, что позволяет уменьшить время выполнения запроса к представлению, по сравнению с прямым выполнением того же запроса из прикладной программы.

Схема является одним из основных объектов базы данных Oracle. В Oracle схема привязывается только к одному пользователю и является логическим набором объектов базы данных [6].

Схема базы данных может включать в себя такие элементы, как:

- проанализировать подобные сервисы;
- проанализировать и описать технологии для реализации;
- последовательности;
- хранимые программы;
- кластеры;
- связи баз данных;
- триггеры;
- библиотеки внешних процедур;
- индексы;
- пакеты;
- хранимые функции и процедуры;
- синонимы;
- представления;
- снимки;
- объектные таблицы;
- объектные типы;
- объектные представления;
- виртуальные таблицы.

На основе полученных результатов, была построена логическая блок схема, которая описывает работу проекта на рисунке 2.3.

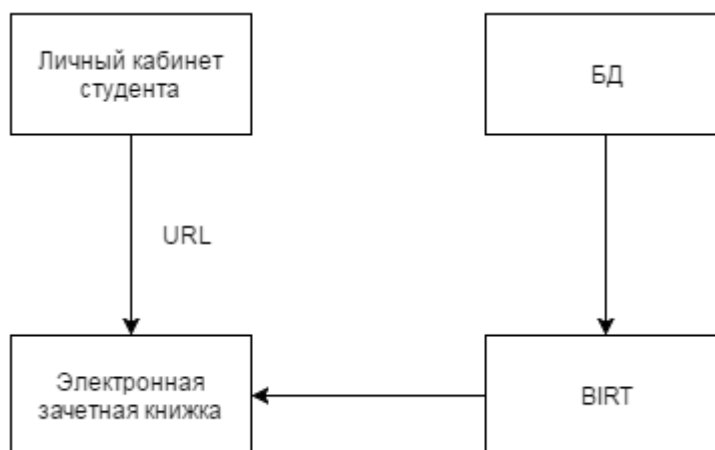


Рисунок 2.3 – Блок схема проекта

### **2.3 Настройка и подготовка программного обеспечения для реализации электронной зачетной книжки**

Для реализации электронной зачетной книжки необходимо скачать и установить IDE Eclipse. Скачать данную IDE можно на официальном сайте Eclipse. После установки Eclipse необходимо создать новый проект. В меню выберите File/New/Other/Bussiness Intelligence and Reporting Tools/Report. Создание проекта показано на рисунке 2.4.

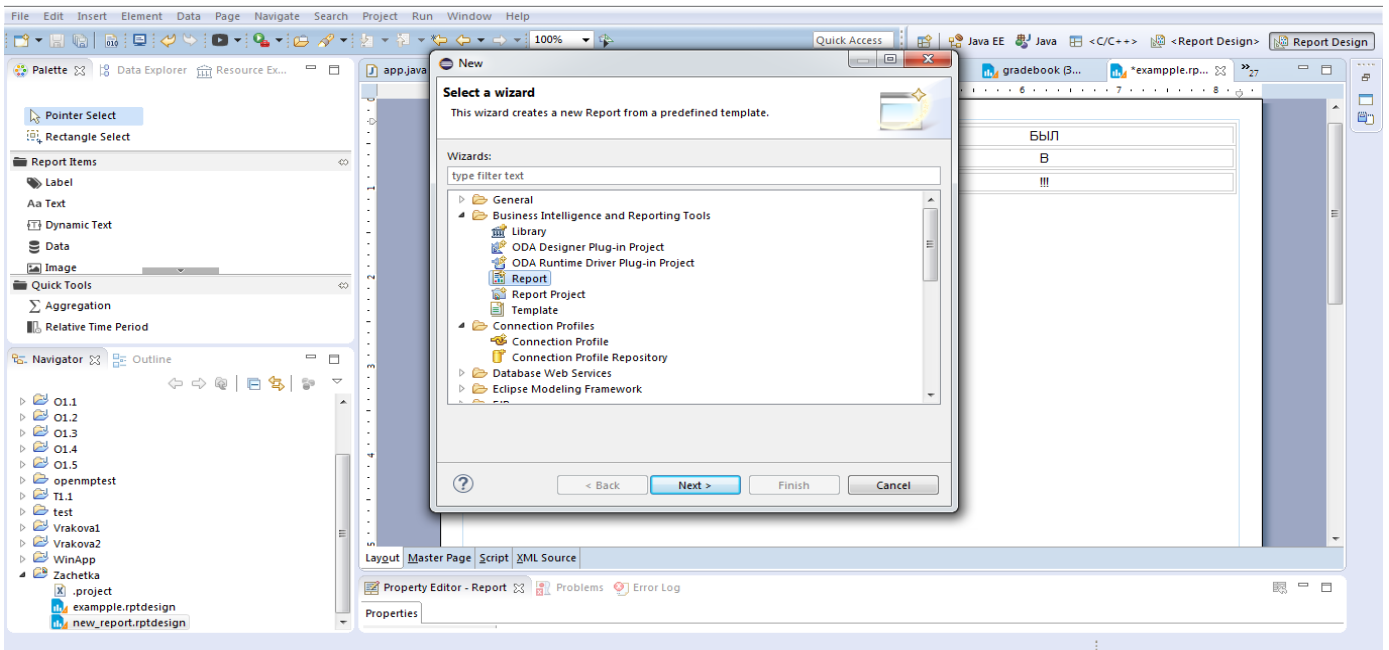


Рисунок 2.4 – Создание проекта

Теперь необходимо подключить созданный проект к БД. Для этого необходимо создать новое подключение, для этого необходимо выполнить следующие действия: Data/New Data Source/JDBC Data Source. Создание подключения к БД показано на рисунке 2.5 и 2.6.

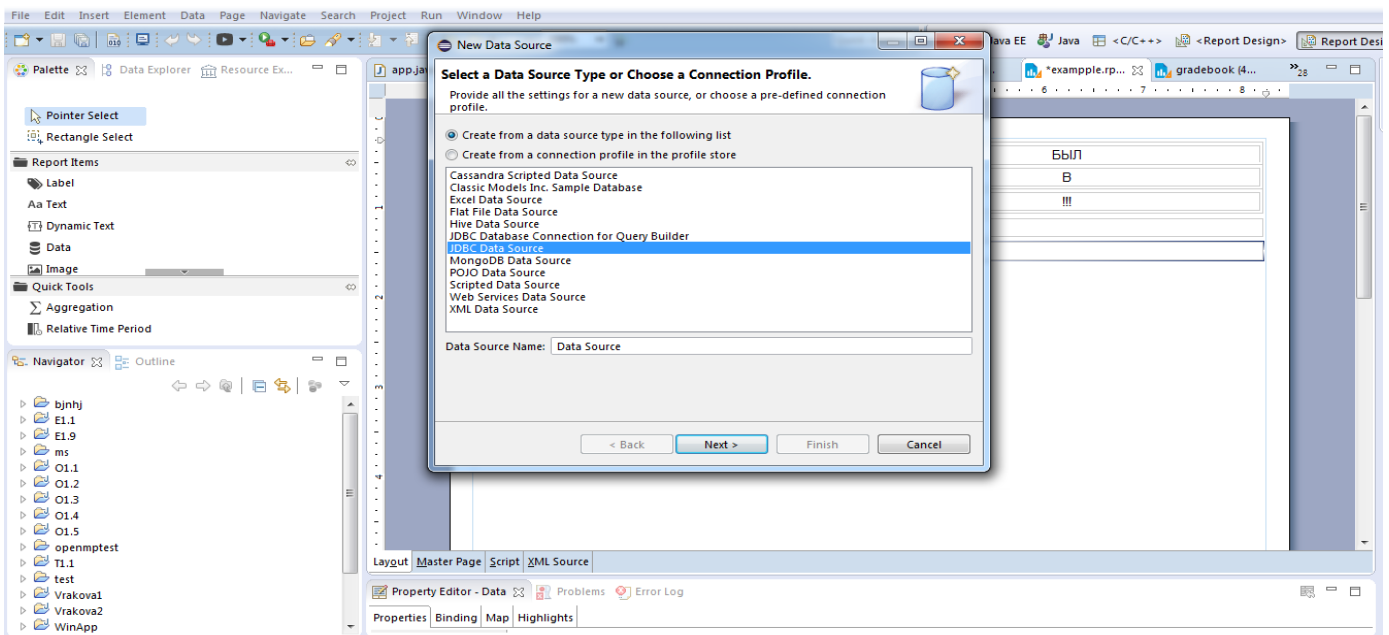


Рисунок 2.5 – Создание подключения к БД

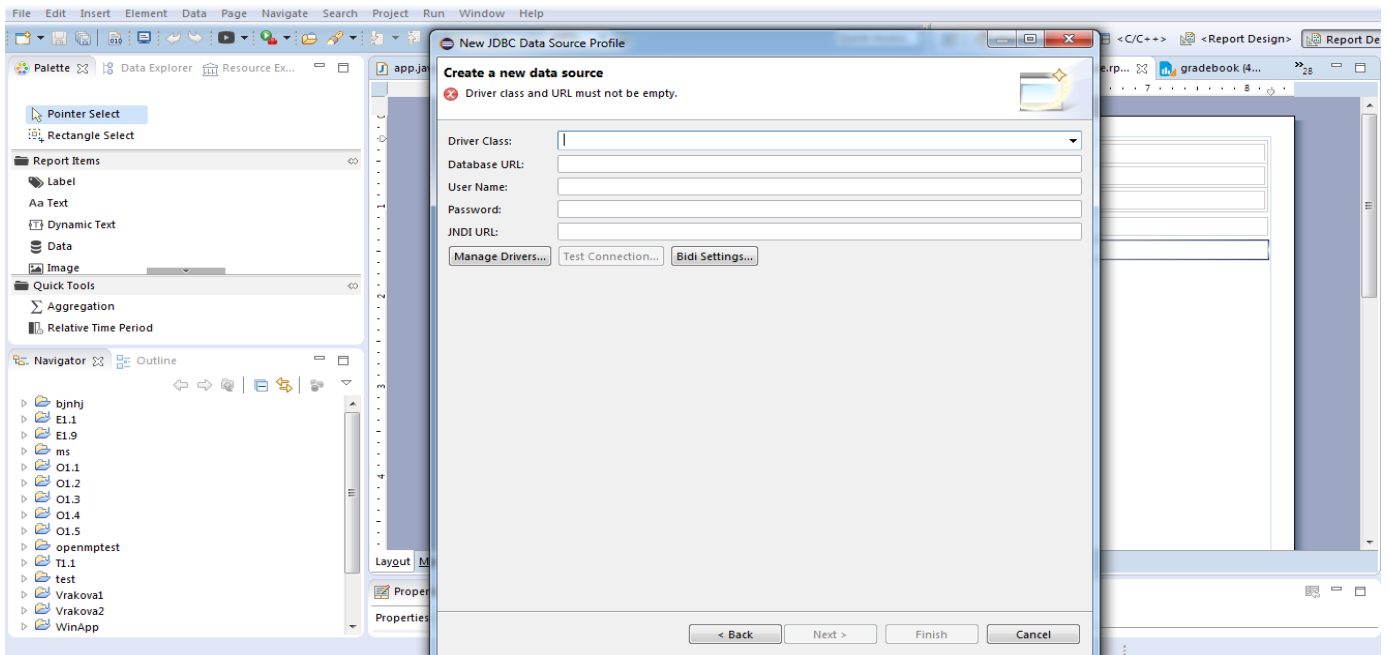
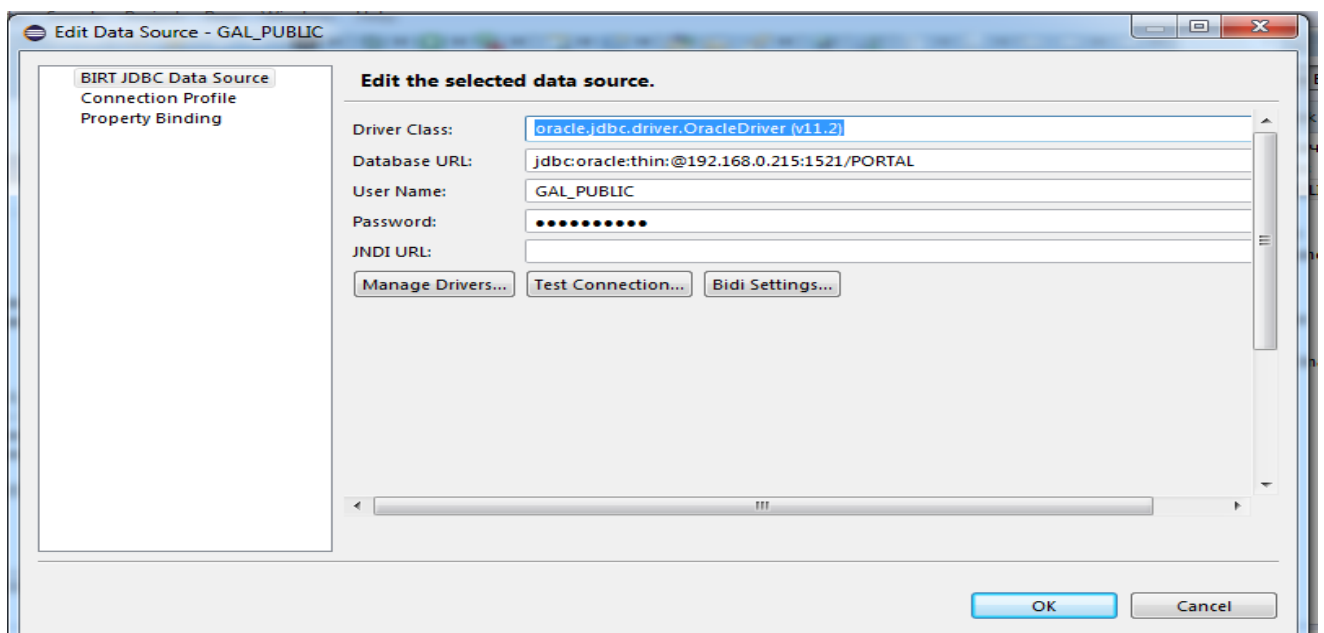


Рисунок 2.6 – Параметры подключения к БД

Но сначала, как уже описывалось выше, необходимо скачать и подключить драйвер JDBC для подключения к БД Oracle. Чтобы подключить драйвер необходимо в открывшемся окне нажать на кнопку **Manage Drivers**, затем на кнопку **Add** и выбрать скачанный драйвер. После этого в окне выбора драйвера для подключения появится драйвер JDBC. Скачать этот драйвер можно на официальном сайте Oracle. Подключение драйвера показано на рисунке 2.7.





## Рисунок 2.7 – Подключение драйвера JDBC

## Рисунок 2.8 – Параметры нового подключения

После подключения драйвера необходимо заполнить все остальные поля для создания подключения к БД. Теперь надо протестировать созданное подключение. Для этого необходимо нажать на кнопку Test Connection. Настройка и проверка подключения к БД показаны на рисунках 2.8 и 2.9

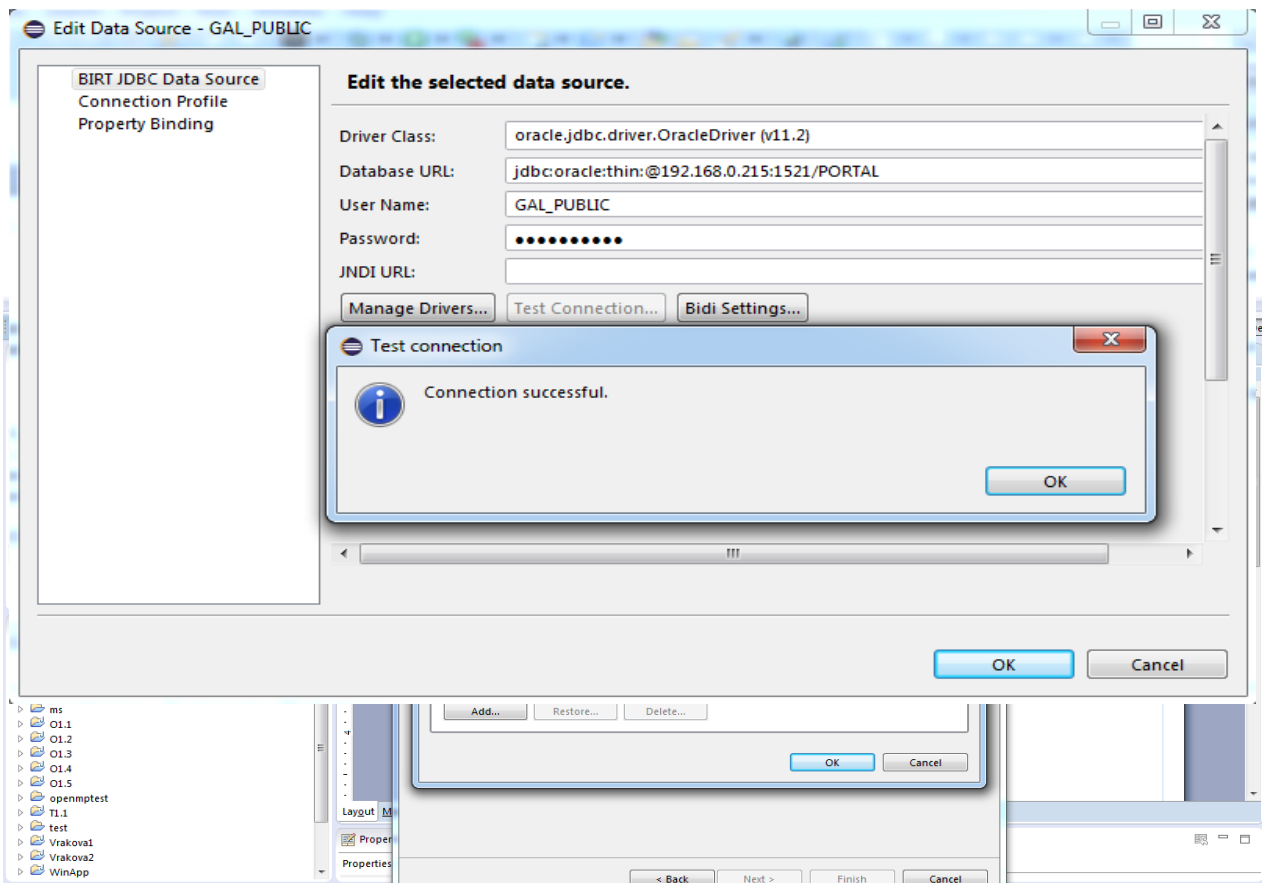


Рисунок 2.9 – Проверка созданного подключения

Подключение к БД прошло успешно. Для того, чтобы работать с БД был установлен Oracle SQL Developer. Скачать его можно также с официального сайта Oracle.

Oracle SQL Developer — интегрированная среда разработки на языках SQL и PL/SQL, с возможностью администрирования баз данных, ориентирована на применение в среде Oracle Database. Работа с программной Oracle SQL Developer показана на рисунке 2.10.

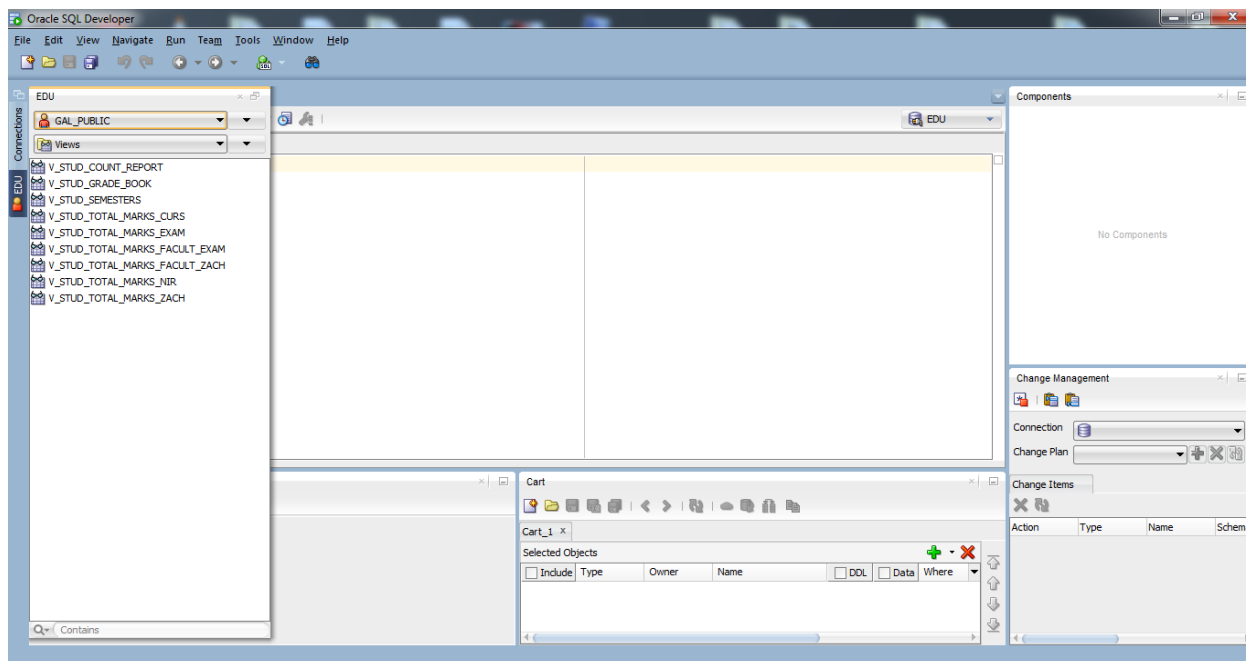


Рисунок 2.10 – Oracle SQL Developer

После этого настройку программного обеспечения можно считать успешно завершенной.

Во второй главе были подробно описаны технологии, с помощью которых планируется создание электронной зачетной книжки. Была произведена и подробно расписана настройка ПО. Был создан начальный проект.

## Глава 3 Реализация и тестирование электронной зачетной книжки

### 3.1 Описание реализации электронной зачетной книжки

В этой главе рассмотрены основные моменты в реализации электронной зачетной книжки: составление интерфейса, описание функций для преобразования информации и запросов в БД. Все действия выполняются в IDE Eclipse, о которой говорилось выше.

Начнем с создания запросов в БД, чтобы получить всю необходимую информацию для 1 формы электронной зачетной книжки. В 1 форме необходимо получить общую информацию о студенте: ФИО, номер группы, номер зачетной книжки, название института, дату зачисления, фотографию студента и т.д. Для того чтобы добавить запрос в БД необходимо создать набор данных (Data Set). Для этого необходимо нажать правой кнопкой по вкладке Data Set и выбрать пункт New Data Set, как показано на рисунке 3.1.

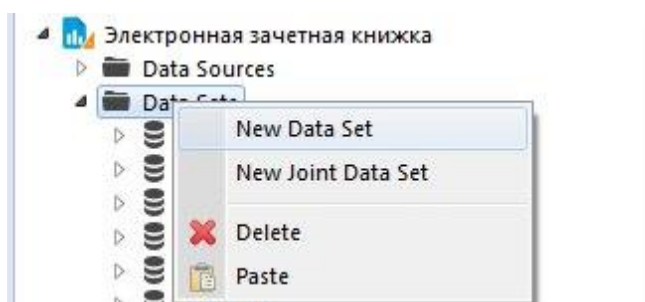


Рисунок 3.1 – Создание набора данных

В появившемся окне нужно выбрать источник данных (подключение к БД) и вписать название, как показано на рисунке 3.2. Так как в этом запросе будет получена основная информация о студенте, то набор данных был назван «GeneralInformation».

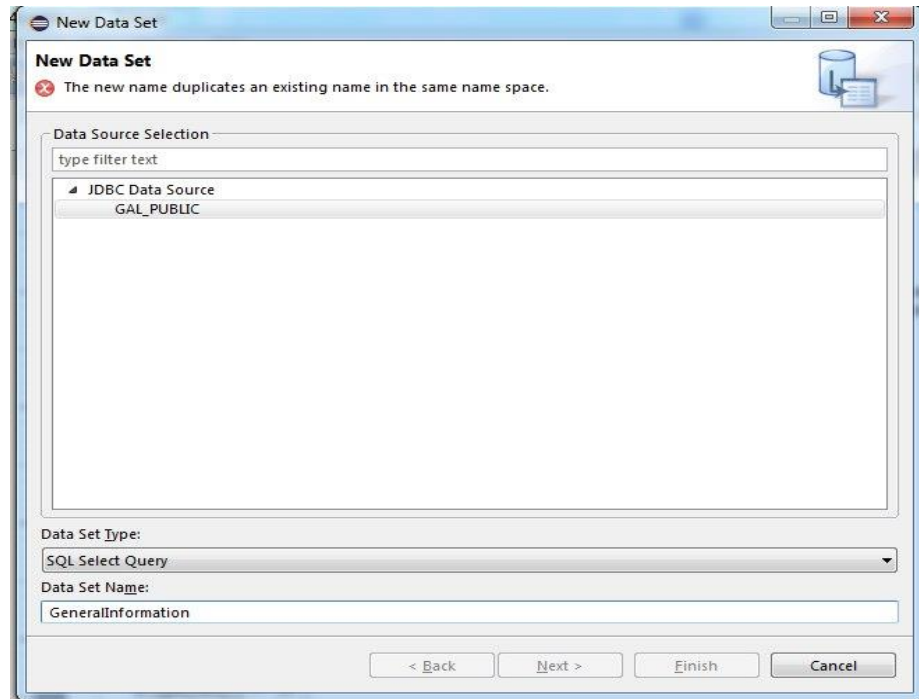


Рисунок 3.2 – Создание набора данных. 2

После создания нового набора данных откроется окно настройки для этого набора. В левой части окна во вкладке Parameters необходимо добавить поле, в котором будет передаваться ID студента, как показано на рисунке 3.3. Поле будет иметь название person и тип значения String. Передавать это поле можно через адресную строку в окне браузера.

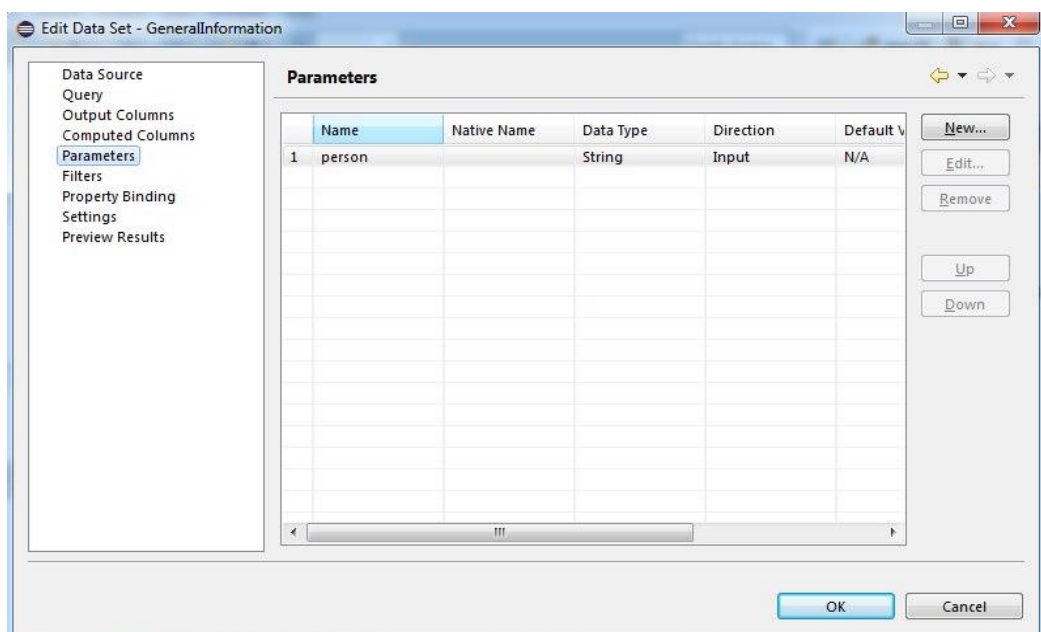


Рисунок 3.3 – Добавления параметра person

После этого во вкладке Query пишется запрос в БД, как показано на рисунке 3.4.

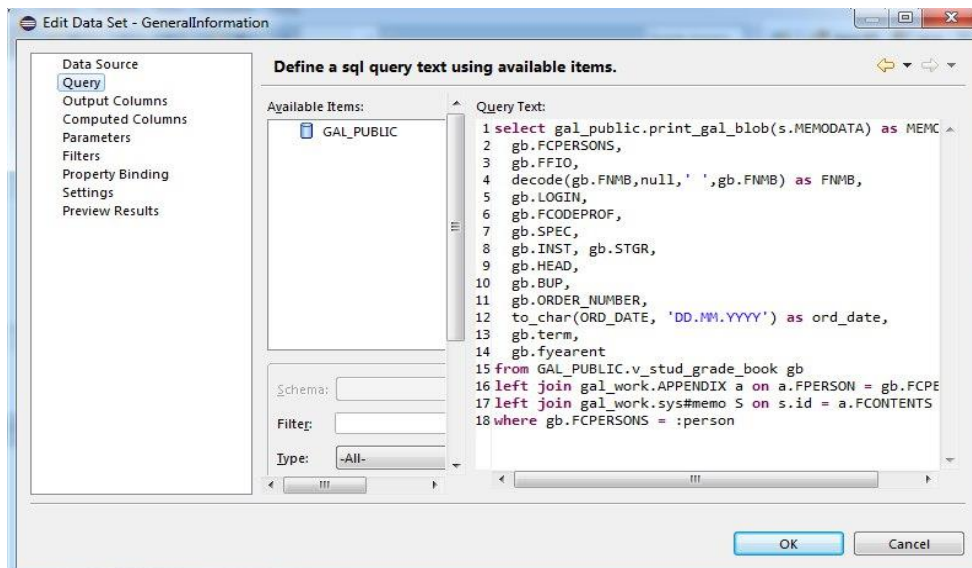


Рисунок 3.4 – Запрос в базу данных

Во вкладке Output Columns можно увидеть и отредактировать имена и типы данных, которые будут получены в результате выполнения запроса в БД, как показано на рисунке 3.5.

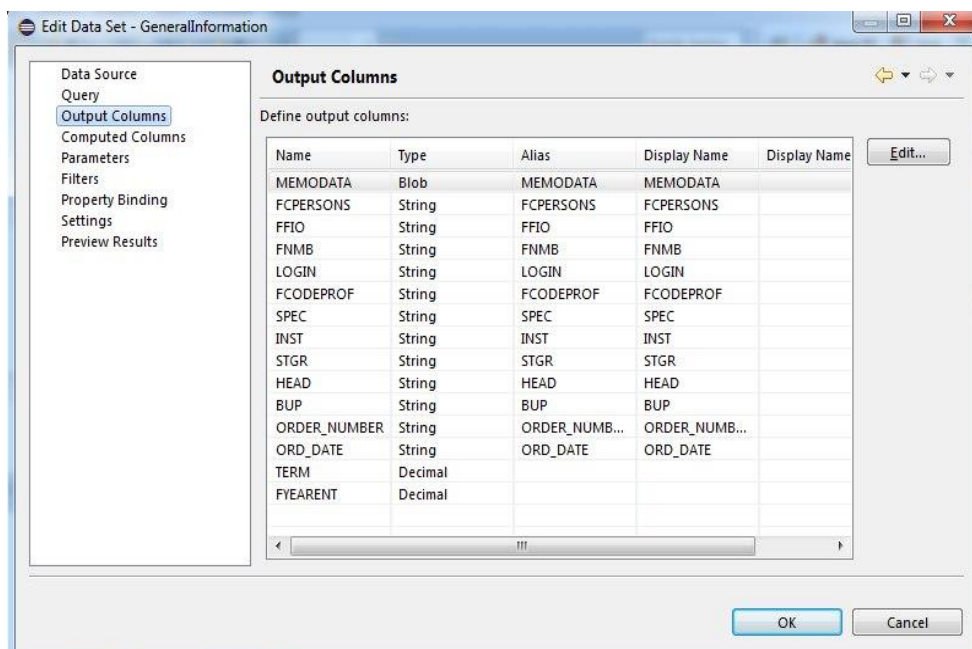


Рисунок 3.5 – Запрос в базу данных

Теперь необходимо реализовать интерфейс для первой формы электронной зачетки. В BIRT существует несколько уже готовых элементов, для реализации интерфейса. Чтобы вставить их в проект, необходимо щелкнуть правой кнопкой мыши, выбрать поле Insert и затем выбрать нужный элемент для вставки, как показано на рисунке 3.6.

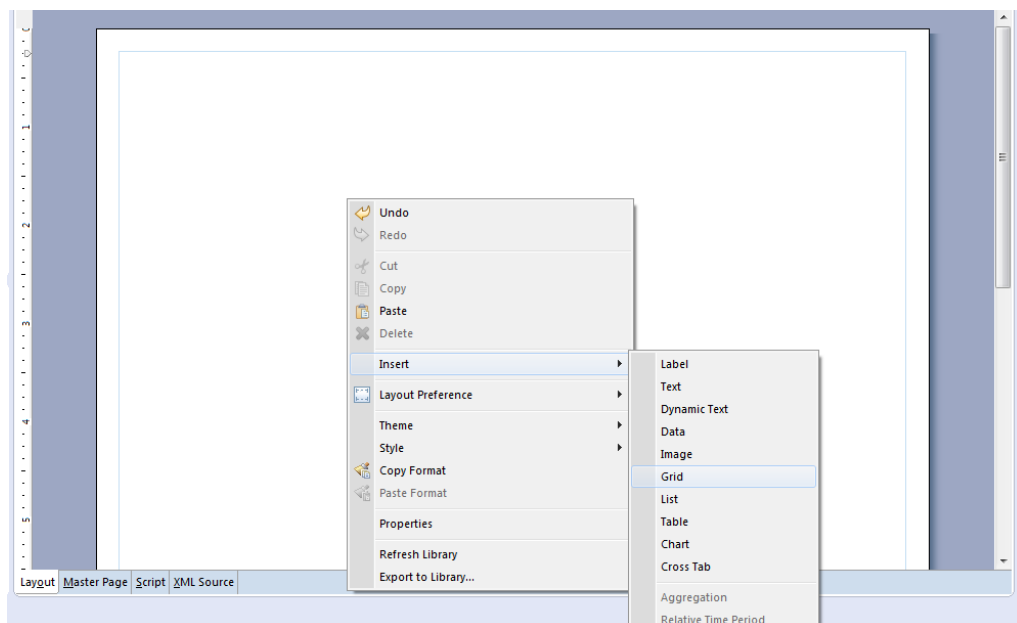


Рисунок 3.6 – Реализация интерфейса

Чтобы создать первую форму необходимо вставить элемент Grid(Сетка). При создании сетки указываем что в ней должно быть 2 столбца.

Все созданные элементы в проекте отображаются в поле Outline в разделе Body, как показано на рисунке 3.7. Таким образом можно легко найти любой ранее созданный элемент.

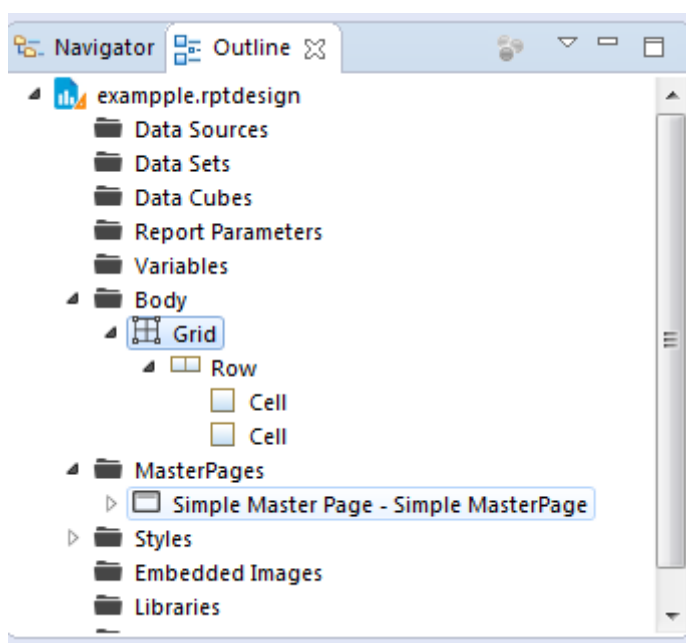


Рисунок 3.7 – Список элементов

У каждого созданного элемента есть набор свойств. Просмотреть и изменить свойства элемента можно выделив элемент и нажав Property Editor (Редактор свойств). Например, через свойства можно задать элементу фиксированную высоту, как показано на рисунке 3.8.

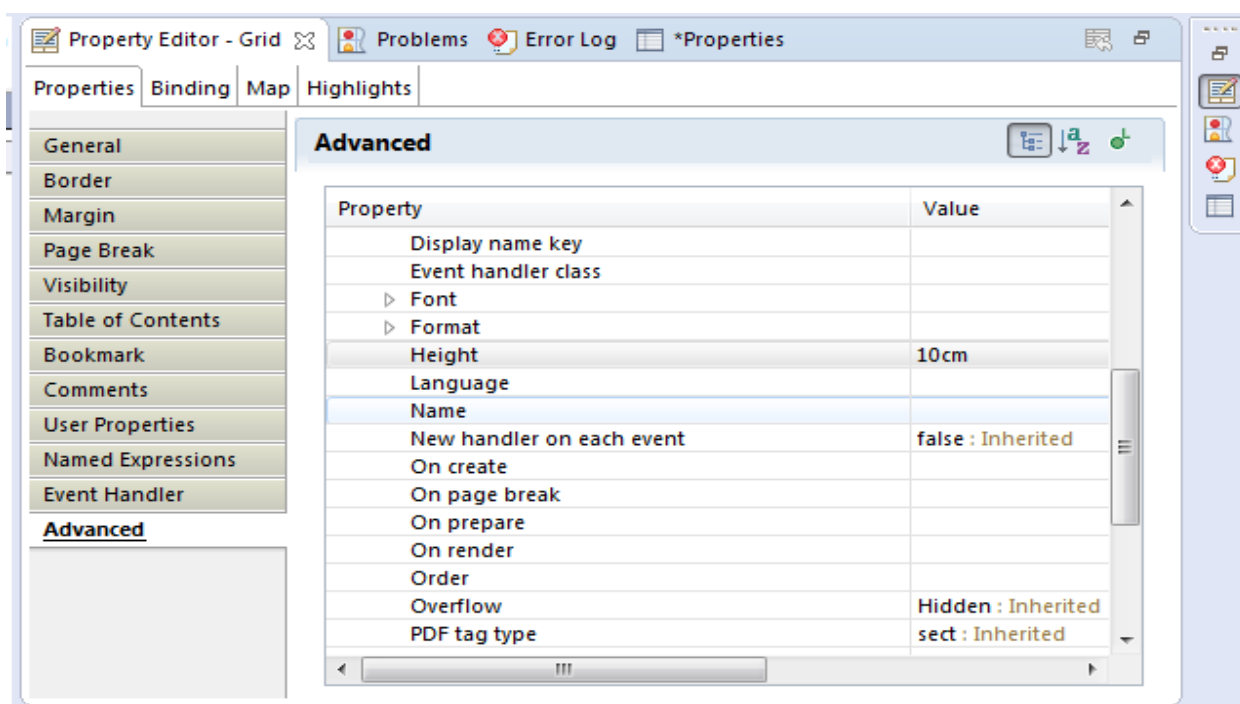


Рисунок 3.8 – Свойства элементов

Теперь элемент Grid имеет фиксированную высоту, то есть высота элемента больше не зависит от количества контента внутри элемента.

Согласно форме в левом столбце должна быть фотография студента, чтобы добавить ее необходимо в левой колонке нажать на правую кнопку мыши/Insert/Image. В появившемся окне необходимо выбрать источник для изображения. Фото студента было получено в созданном ранее запросе в базу данных и находится в поле MEMODATA. Чтобы достать информацию из этого поля в источнике изображение нужно написать `row["MEMODATA"]`, как показано на рисунке 3.9.

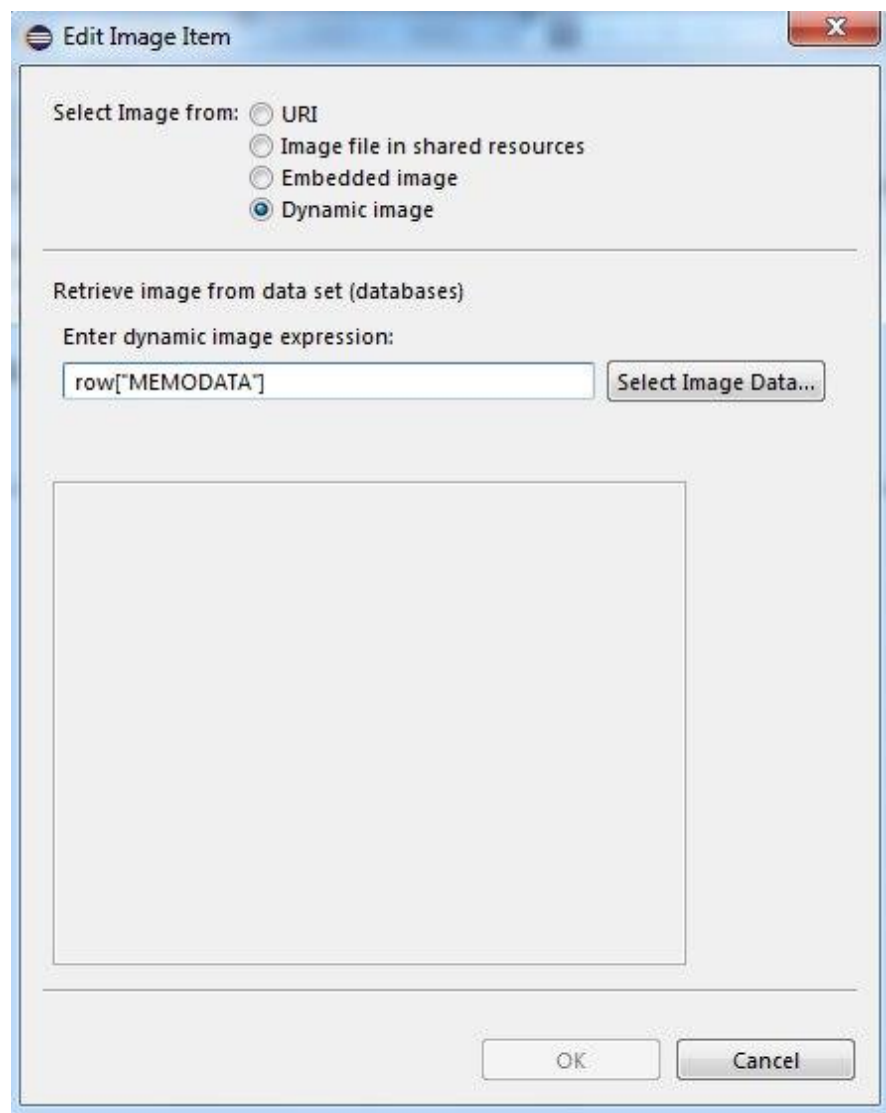


Рисунок 3.9 – Источник изображения



Теперь необходимо задать изображению фиксированную ширину. Это делается через свойства объекта, как уже было показано ранее. Чтобы фото студента отображалось в центре столбца, необходимо изменить некоторые свойства столбца. Свойству Text alignment присвоить значение Center, а свойству Vertical alignment присвоить значение Middle.

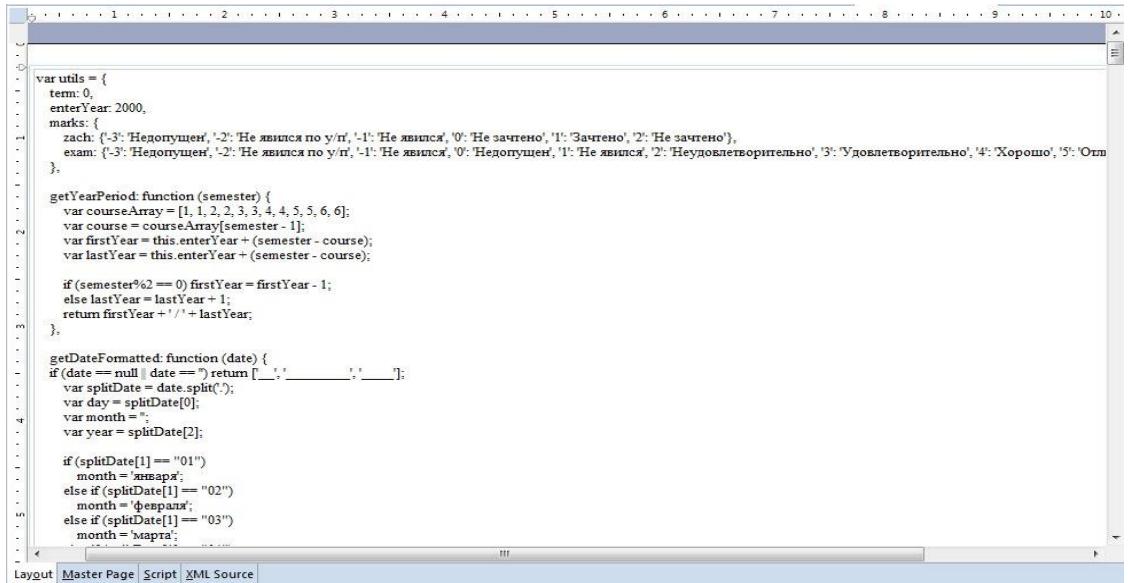
Теперь нужно заполнить информацией правый столбец. Для этого необходимо в правый столбец вставить несколько элементов типа Text. Задать расположение текста можно также с помощью свойств элементов. В первых 2 элементах типа текст будет отображаться обычный текст, который будет одинаков для всех студентов. В третьем поле должен выводиться номер зачетной книжки. Номер зачетной книжки студента хранится в поле FNMB, как показано на рисунке 3.10.

Министерство образования и науки РФ
ФГБОУ ВО «Тольяттинский государственный университет»
<b>'Зачетная книжка № ' + row["FNMB"]</b>

Рисунок 3.10 – Номер зачетной книжки

Однако некоторые данные полученные в ходе запроса в БД требуют некоторого преобразования перед тем, как использовать их. В таких случаях для преобразования данных можно использовать JavaScript. В BIRT писать

JavaScript можно прямо в текстовых полях. Поэтому перед первой формой электронной зачетной книжки было создано текстовое поле, в котором будут описаны все необходимые для преобразования данных функции на языке JavaScript, как показано на рисунке 3.11.



```

var utils = {
  term: 0,
  enterYear: 2000,
  marks: {
    zach: {'-3': 'Недопущен', '-2': 'Не явился по у/п', '-1': 'Не явился', '0': 'Не зачтено', '1': 'Зачтено', '2': 'Не зачтено'},
    exam: {'-3': 'Недопущен', '-2': 'Не явился по у/п', '-1': 'Не явился', '0': 'Недопущен', '1': 'Не явился', '2': 'Неудовлетворительно', '3': 'Удовлетворительно', '4': 'Хорошо', '5': 'Отлично'}
  },
  getYearPeriod: function (semester) {
    var courseArray = [1, 1, 2, 2, 3, 3, 4, 4, 5, 5, 6, 6];
    var course = courseArray[semester - 1];
    var firstYear = this.enterYear + (semester - course);
    var lastYear = this.enterYear + (semester - course);

    if (semester%2 == 0) firstYear = firstYear - 1;
    else lastYear = lastYear + 1;
    return firstYear + '/' + lastYear;
  },
  getDateFormatted: function (date) {
    if (date == null || date == '') return ['_', '_', '_', '_', '_', '_'];
    var splitDate = date.split('.');
    var day = splitDate[0];
    var month = '';
    var year = splitDate[2];

    if (splitDate[1] == "01")
      month = 'января';
    else if (splitDate[1] == "02")
      month = 'февраля';
    else if (splitDate[1] == "03")
      month = 'марта';
  }
}

```

Рисунок 3.11 – JavaScript функции для преобразования данных

Чтобы задать тексту различные шрифты, размер, цвет и т.д. в BIRT используются стили. Чтобы создать стиль нужно щелкнуть правой кнопкой мышь и во вкладке Style выбрать New Style, как показано на рисунке 3.12.

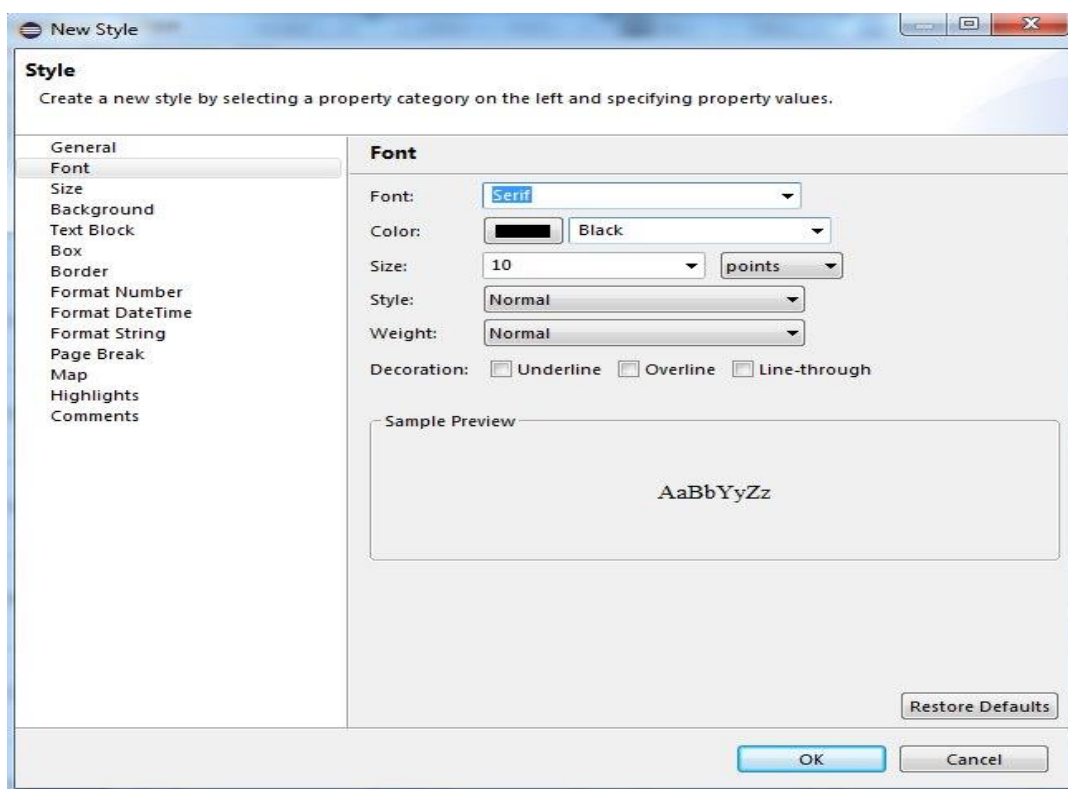


Рисунок 3.12 – Создание стиля

Всего было создано несколько стилей, которые будут использованы в заголовках, в таблицах и т.д. Чтобы применить стиль на элемент необходимо выделить элемент, щелкнуть по нему правой кнопкой, во вкладке Style выбрать Apply Style и выбрать название стиля, который необходимо применить, как показано на рисунке 3.13.

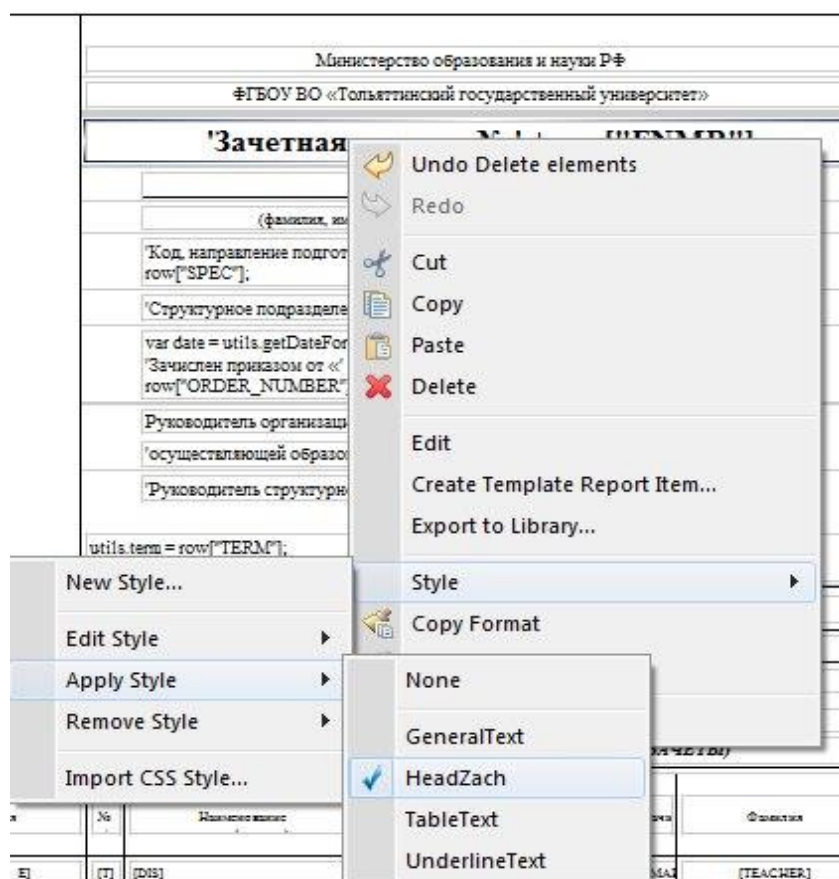


Рисунок 3.13 – Применения стиля

В итоге была создана первая форма электронной зачетной книжки, которая показана на рисунке 3.14.

The screenshot shows a preview window for the first form of an electronic credit book. The form is titled 'Зачетная книжка № ' + row["FNMB"]'. It contains several fields and a code block:
 

```

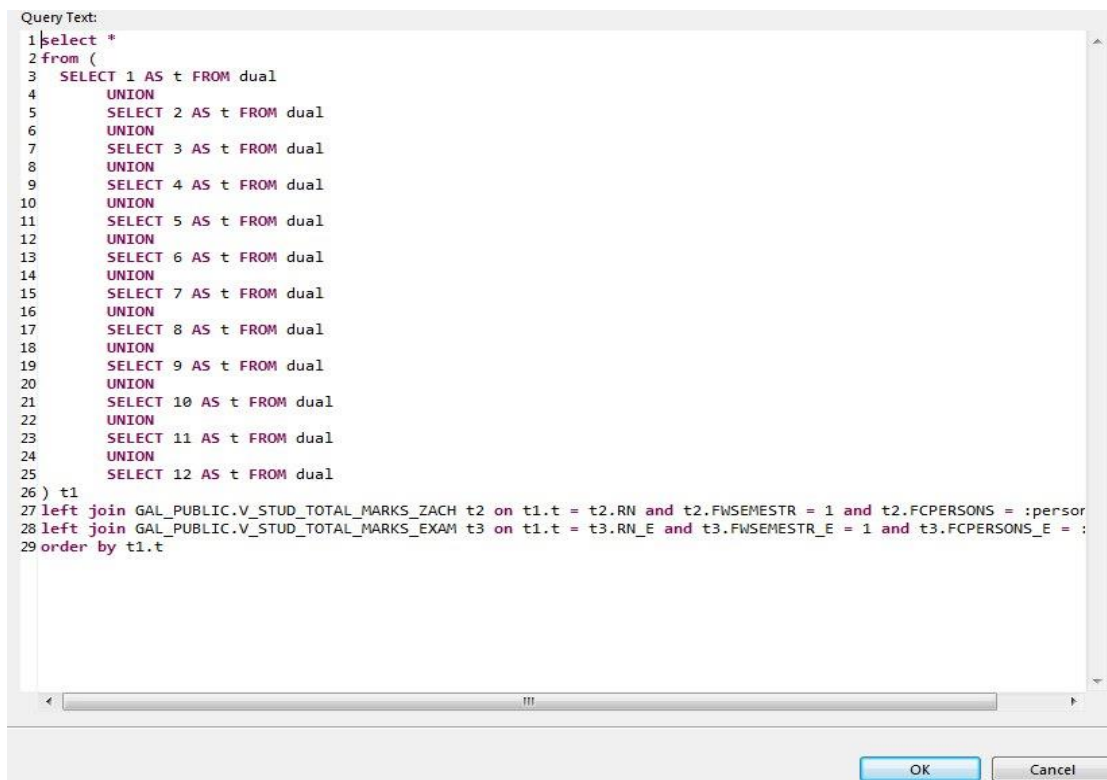
  row["FFIO"]
  (фамилия, ил. отчество студента (последнее - при наличии))
  Код, направление подготовки (специальность) ' + row["FCODEPROF"] + ' ' +
  row["SPEC"];
  'Структурное подразделение ' + row["INST"];
  var date = utils.getDateFormatted(row["ORD_DATE"]);
  'Зачислен приказом от << ' + date[0] + '>> ' + date[1] + ' ' + date[2] + ' г. № ' +
  row["ORDER_NUMBER"];
  Руководитель организации,
  'осуществляющей образовательную деятельность М.М. Криштап'
  'Руководитель структурного подразделения ' + row["HEAD"];
  utils.term = row["TERM"];
  utils.enterYear = row["FYEAARENT"];
  
```

 The form is shown in a preview window with a small red 'X' icon in the top left corner.

Рисунок 3.14 – Применения стиля

В последующих 12 формах будут показываться результаты зачетов и экзаменов студента за время его учебы. Значит необходимо создать 12 новых наборов данных, чтобы получить результаты его учебы за каждый семестр.

Запросы будут подобны друг другу, они будут различаться лишь номерами семестров, как показано на рисунке 3.15. Наборы данных будут называться «maks\_номер семестра».



```
Query Text:
1 |select *
2 |from (
3 |  SELECT 1 AS t FROM dual
4 |    UNION
5 |    SELECT 2 AS t FROM dual
6 |    UNION
7 |    SELECT 3 AS t FROM dual
8 |    UNION
9 |    SELECT 4 AS t FROM dual
10 |   UNION
11 |   SELECT 5 AS t FROM dual
12 |   UNION
13 |   SELECT 6 AS t FROM dual
14 |   UNION
15 |   SELECT 7 AS t FROM dual
16 |   UNION
17 |   SELECT 8 AS t FROM dual
18 |   UNION
19 |   SELECT 9 AS t FROM dual
20 |   UNION
21 |   SELECT 10 AS t FROM dual
22 |   UNION
23 |   SELECT 11 AS t FROM dual
24 |   UNION
25 |   SELECT 12 AS t FROM dual
26 | ) t1
27 | left join GAL_PUBLIC.V_STUD_TOTAL_MARKS_ZACH t2 on t1.t = t2.RN and t2.FWSEMESTR = 1 and t2.FCPERSONS = :persor
28 | left join GAL_PUBLIC.V_STUD_TOTAL_MARKS_EXAM t3 on t1.t = t3.RN_E and t3.FWSEMESTR_E = 1 and t3.FCPERSONS_E = :
29 | order by t1.t
```

Рисунок 3.15 – Запрос для получения успеваемости

Для вывода успеваемости студента за конкретный семестр будет создан элемент Table (Таблица). Этот элемент отличается от элемента Grid тем, что к нему можно привязать конкретный набор данных. Набор данных, который будет привязан к таблице можно указать при создании таблицы. Также в этом окне можно выбрать поля, которые будут отображаться в таблице, как показано на рисунке 3.16.

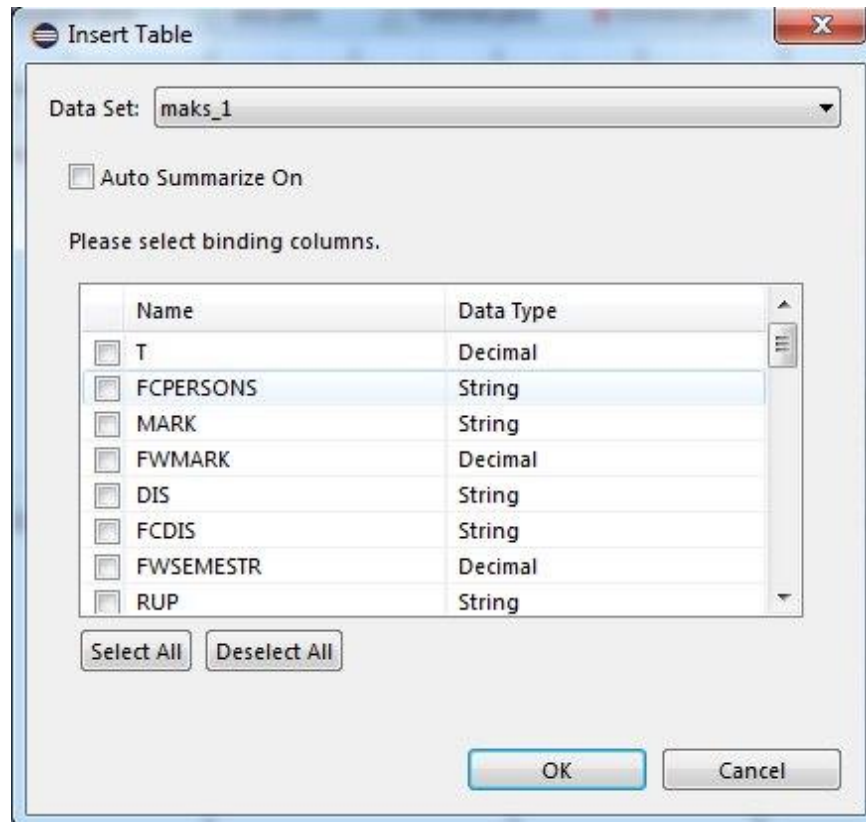


Рисунок 3.16 – Запрос для получения успеваемости

Теперь к таблице привязан набор данных maks\_1. В таблицу также были добавлены текстовые поля, по аналогии с предыдущей формой. Конечный вариант таблицы показан на рисунке 3.17.

row["SEMESTER"]+'-й семестр '+ utils.getYearPeriod(row["SEMESTER"])+' учебного года!					<b>ПЕРВЫЙ</b>					КУРС					[FIO]				
															(Фамилия, Имя, Отчество студента)				
<i>Детали Вывод</i>																			
<b>РЕЗУЛЬТАТЫ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ (ЭКЗАМЕНЫ)</b>										<b>РЕЗУЛЬТАТЫ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ (ЗАЧЕТЫ)</b>									
No	Имя поля	Объем по л-во	Оценка	Дата сдачи	Фамилия	No	Имя поля	Объем по л-во	Оценка	Дата сдачи	Фамилия								
[T]	[DIS_E]	[ZET_E]	[MARK_E]	[DATEMA]	[TEACHER_E]	[T]	[DIS]	[ZET]	[MARK]	[DATEMA]	[TEACHER]								
count_zsch_2_true = row["count_z_2_true"];		[count_				count_ekz_2_true = row["count_e_2_true"];		[count_											

Рисунок 3.17 – Таблица успеваемости за 1 семестр

Аналогичным образом были составлены остальные формы электронной зачетной книжки, которые отображали успеваемость студента за конкретный семестр.

В следующей форме электронной зачетной книжки будут отображаться

факультативные дисциплины. Чтобы получить эти данные необходимо создать новый набор данных, как показано на рисунке 3.18. Набор данных будет называться «marks\_fac».

```

Query Text:
1 SELECT t1.*,
2     t2.*,
3     t3.*
4 FROM (
5     SELECT 1 AS t
6     FROM dual
7     UNION
8     SELECT 2 AS t
9     FROM dual
10    UNION
11    SELECT 3 AS t
12    FROM dual
13    UNION
14    SELECT 4 AS t
15    FROM dual
16    UNION
17    SELECT 5 AS t
18    FROM dual
19    UNION
20    SELECT 6 AS t
21    FROM dual
22    UNION
23    SELECT 7 AS t
24    FROM dual
25    UNION
26    SELECT 8 AS t
27    FROM dual
28    UNION
29    SELECT 9 AS t
30    FROM dual
31    UNION
32    SELECT 10 AS t
33    FROM dual) t1
34 left join gal_public.v_stud_total_marks_facult_zach t2 on t1.t = t2.rn and t2.fcpersons = :person1
35 left join gal_public.v_stud_total_marks_facult_exam t3 on t1.t = t3.rn_e and t3.fcpersons_e = :person2
36 order by t1.t
  
```

Рисунок 3.18 – Запрос данных для факультативных дисциплин

Для отображения этих данных также будет создана таблица, к которой будет привязан только что созданный набор данных. Конечный вариант этой таблицы показан на рисунке 3.19.

ФАКУЛЬТАТИВНЫЕ					ДИСЦИПЛИНЫ					[FIO]	
										(Фамилия, Имя, Отчество студента)	
РЕЗУЛЬТАТЫ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ (ЭКЗАМЕНЫ)					РЕЗУЛЬТАТЫ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ (ЗАЧЕТЫ)						
№	Наименование	Объем лек-ов	Средня	Дата сдачи	Фамилия	№	Наименование	Объем лек-ов	Средня	Дата сдачи	Фамилия
[#]	[DIS_E]	[LET_E]	[MARK_E]	[DATEMA]	[TEACHER_E]	[#]	[DIS]	[LET]	[MARK]	[DATEMA]	[TEACHER]
	count_f_zach_true =										
	row['count_f_z'];										
	count_f_ekz_true =										
	row['count_f_s'];										
	[count_f_z]										
	[count_f_s]										

Рисунок 3.19 – Факультативные дисциплины

В следующей форме отображаются курсовые работы студента. Для



получения данных о курсовых работах студента необходимо создать новый набор данных, как показано на рисунке 3.20. Набор данных будет называться «marks\_kurs».

```

Query Text:
1 SELECT *
2 FROM (
3     SELECT 1 AS t
4     FROM dual
5     UNION
6     SELECT 2 AS t
7     FROM dual
8     UNION
9     SELECT 3 AS t
10    FROM dual
11    UNION
12    SELECT 4 AS t
13    FROM dual
14    UNION
15    SELECT 5 AS t
16    FROM dual
17    UNION
18    SELECT 6 AS t
19    FROM dual
20    UNION
21    SELECT 7 AS t
22    FROM dual
23    UNION
24    SELECT 8 AS t
25    FROM dual
26    UNION
27    SELECT 9 AS t
28    FROM dual
29    UNION
30    SELECT 10 AS t
31    FROM dual) t1
32 left join gal_public.V_STUD_TOTAL_MARKS_CURS t2 on t1.t = t2.rn and t2.fcpersons = :person
33 order by t1.t
  
```

Рисунок 3.20 – Запрос данных для курсовых работ

Для отображения этих данных также будет создана таблица, к которой будет привязан только что созданный набор данных. Конечный вариант этой таблицы показан на рисунке 3.21.

КУРСОВЫЕ			РАБОТЫ (ПРОЕКТЫ)		[FIO]	
(Фамилия, Имя, Отчество студента)						
Маркер: RN						
№	Наименование	Тема курсовой работы (проекта)	Семестр	Оценка	Дата сдачи	Фамилия преподавателя
[#]	[DIS]	[TEMA]	[FW SEMESTR]	[MARK]	[DATEMA]	[TEACHER]
If (com['CURS'] = 0) Количество курсовых работ (проектов) __ не фиксировано __, не связано __; или: Количество курсовых работ (проектов) __ + com['CURS'] + 1; не фиксировано __, не связано __; com_count['course_marks'] + 1; не фиксировано __, не связано __; com_count['course_marks'] + 1;						
[count_curs_true]						

Рисунок 3.21 – Курсовые работы студента

В следующей форме отображаются оценки за прошедшую студентом



практику. Для получения этих данных необходимо создать новый набор данных, как показано на рисунке 3.22. Набор данных будет называться «marks\_pract».

```

Query Text:
1 SELECT *
2 FROM (
3     SELECT 1 AS t
4     FROM dual
5     UNION
6     SELECT 2 AS t
7     FROM dual
8     UNION
9     SELECT 3 AS t
10    FROM dual
11    UNION
12    SELECT 4 AS t
13    FROM dual
14    UNION
15    SELECT 5 AS t
16    FROM dual
17    UNION
18    SELECT 6 AS t
19    FROM dual
20    UNION
21    SELECT 7 AS t
22    FROM dual
23    UNION
24    SELECT 8 AS t
25    FROM dual
26    UNION
27    SELECT 9 AS t
28    FROM dual
29    UNION
30    SELECT 10 AS t
31    FROM dual) t1
32 left join gal_public.V_STUD_TOTAL_MARKS_PRACT t2 on t1.t = t2.rn
33 order by t1.t
  
```

Рисунок 3.22 – Запрос данных для оценок за практику

Для отображения этих данных также будет создана таблица, к которой будет привязан только что созданный набор данных. Конечный вариант этой таблицы показан на рисунке 3.23.

ПРАКТИКА								[FIO]		
								(Фамилия, Имя, Отчество студента)		
Настройка Поля										
№	Наименование	Семестр	Место проведения практики	В качестве кого работал	Ф.И.О. руководителя практики от предприятия	Объем по-час	Ф.И.О. руководителя практики от организации, осуществляющей	Оценки по итогам	Дата проведения	Фамилия лица,
[P]	[PRACT_TYPE]	[SEMESTR]	[PRACT_PLACE]	[POST]	[HEAD_COMPANY]	[HOURS]	[HEAD_EDU]	with galMarkTend	[DATE_ATT]	[FIO_ATT]

Рисунок 3.23 – Оценки за практику

В следующей форме отображаются данные о научно-исследовательских работах. Для получения этих данных необходимо создать

новый набор данных, как показано на рисунке 3.24. Набор данных будет называться «marks\_nir».

```

Query Text:
1 select *
2 from (SELECT 1 AS t FROM dual
3 UNION
4 SELECT 2 AS t FROM dual
5 UNION
6 SELECT 3 AS t FROM dual
7 UNION
8 SELECT 4 AS t FROM dual
9 UNION
10 SELECT 5 AS t FROM dual
11 UNION
12 SELECT 6 AS t FROM dual
13 UNION
14 SELECT 7 AS t FROM dual
15 UNION
16 SELECT 8 AS t FROM dual
17 UNION
18 SELECT 9 AS t FROM dual
19 UNION
20 SELECT 10 AS t FROM dual
21 UNION
22 SELECT 11 AS t FROM dual
23 UNION
24 SELECT 12 AS t FROM dual) t1
25 left join (
26 select rn, fcpersons, dis as type_nir, fwsemestr as semestr, fwmark as mark, datemark as date_mark, teacher f
27 ) t2 on t1.t = t2.rn
28 left join (
29 select rn as rn_t, fcpersons as fcpersons_t, dis as type_nir_t, fwsemestr as semestr_t, fwmark as mark_t, daten
30 ) t3 on t1.t + 12 = t3.rn_t and t3.rn_t > 12
31 order by t1.t
32
  
```

Рисунок 3.24 – Запрос данных для оценок за научно-исследовательские работы

Для отображения этих данных также будет создана таблица, к которой будет привязан только что созданный набор данных. Конечный вариант этой таблицы показан на рисунке 3.25.

НАУЧНО-ИССЛЕДОВАТЕЛЬСКАЯ					РАБОТА					[FIO]	
											(Фамилия, Имя, Отчество студента)
№	Вид научно-	Семестр	Оценка	Дата сдачи	Фамилия преподавателя	№	Вид научно-	Семестр	Оценка	Дата сдачи	Фамилия преподавателя
[T]	[TYPE_NIR]	[SEMESTR]	[MARK]	[DATE_MARK]	[TEACHER]	rn_t	[TYPE_NIR_T]	[SEMESTR]	[MARK_T]	[DATE_MARK_T]	[TEACHER_T]

Рисунок 3.25 – Научно-исследовательские работы

В следующей форме отображаются данные о государственных экзаменах студента. Для получения этих данных необходимо создать новый набор данных, как показано на рисунке 3.26. Набор данных будет называться «marks\_gek».

```

Query Text:
1 SELECT *
2 FROM (
3     SELECT 1 AS t
4     FROM dual
5     UNION
6     SELECT 2 AS t
7     FROM dual
8     UNION
9     SELECT 3 AS t
10    FROM dual
11    UNION
12    SELECT 4 AS t
13    FROM dual
14    UNION
15    SELECT 5 AS t
16    FROM dual
17    UNION
18    SELECT 6 AS t
19    FROM dual
20    UNION
21    SELECT 7 AS t
22    FROM dual
23    UNION
24    SELECT 8 AS t
25    FROM dual
26    UNION
27    SELECT 9 AS t
28    FROM dual
29    UNION
30    SELECT 10 AS t
31    FROM dual) t1
32 left join gal_public.v_fake_gek t2 on t1.t = t2.rn
33 order by t1.t

```

Рисунок 3.26 – Запрос данных для оценок за государственные экзамены

Для отображения этих данных также будет создана таблица, к которой будет привязан только что созданный набор данных. Конечный вариант этой таблицы показан на рисунке 3.27.

ГОСУДАРСТВЕННЫЕ		ЭКЗАМЕНЫ		[FIO]
(Фамилия, Имя, Отчество студента)				
Набор Ячей				
№	Идентификатор зачетки (номер)	Дата сдачи экзамена	Оценка	Фамилия преподавателя и члена Государственной экзаменационной комиссии
[7]	[IDSK]	[DATE_MARK]	[MARK]	[GEEK_FIO]

Рисунок 3.27 – Государственные экзамены

В последней форме электронной зачетной книжки будет отображаться информация о ВКР студента, а также когда и какая квалификация была присвоена студенту при получении диплома. Для получения этих данных необходимо создать новый набор данных, как показано на рисунке 3.28. Набор данных будет называться «vkr».

```
Query Text:
1 select *
2 from gal_public.v_vkr
```

Рисунок 3.28 – Запрос данных для оценок за государственные экзамены

Для отображения данных о ВКР студента необходимо создать элемент Grid. В левом столбце будет отображаться информация о ВКР студента, в правом столбце будет информация о том, когда и какая квалификация была присвоена студенту, а также список членов ГЭК, как показано на рисунке 3.29

<table border="1"> <tr><td colspan="2" style="text-align: center;">Выпускная квалификационная работа</td></tr> <tr><td colspan="2" style="text-align: center;">[FIO]</td></tr> <tr><td colspan="2" style="text-align: center;">(Фамилия, Имя, Отчество студента)</td></tr> <tr><td>Форма выпускной квалификационной работы:</td><td>[FORM_VKR]</td></tr> <tr><td>Тема:</td><td>[THEME]</td></tr> <tr><td>Руководителя:</td><td>[ADVISER]</td></tr> <tr><td colspan="2" style="text-align: center;">(Фамилия И.О.)</td></tr> <tr><td>Дата защиты:</td><td>var date = utils.getDateFormatted(row["DATE_DEFENCE"]); '&lt;&lt; + date[0] + &gt;&gt;' + date[1] + ' ' + date[2] + ' г.'</td></tr> <tr><td>Оценка:</td><td>[MARK]</td></tr> <tr><td>Председатель ГЭК:</td><td>[HEAD_GEK]</td></tr> <tr><td>Члены ГЭК:</td><td>var str = " if (row["MEMBERS_GEK"] != " &amp;&amp; row["MEMBERS_GEK"] != null) { var array = row["MEMBERS_GEK"].split( ); }; for (var i = 0; i &lt; array.length; i++) str += array[i] + "&lt;br&gt;"; }; str;</td></tr> <tr><td colspan="2">Label</td></tr> </table>	Выпускная квалификационная работа		[FIO]		(Фамилия, Имя, Отчество студента)		Форма выпускной квалификационной работы:	[FORM_VKR]	Тема:	[THEME]	Руководителя:	[ADVISER]	(Фамилия И.О.)		Дата защиты:	var date = utils.getDateFormatted(row["DATE_DEFENCE"]); '<< + date[0] + >>' + date[1] + ' ' + date[2] + ' г.'	Оценка:	[MARK]	Председатель ГЭК:	[HEAD_GEK]	Члены ГЭК:	var str = " if (row["MEMBERS_GEK"] != " && row["MEMBERS_GEK"] != null) { var array = row["MEMBERS_GEK"].split( ); }; for (var i = 0; i < array.length; i++) str += array[i] + " "; }; str;	Label		<table border="1"> <tr><td colspan="2" style="text-align: center;">Решением Государственной экзаменационной комиссии</td></tr> <tr><td colspan="2">var date = utils.getDateFormatted(row["DECISION_DATE"]); 'or &lt;&lt; + date[0] + &gt;&gt;' + date[1] + ' ' + date[2] + ' г. протокол № ' + ((row["DECISION_ORDER"] == null) ? '_____': row["DECISION_ORDER"])</td></tr> <tr><td>студенту</td><td>[FIO]</td></tr> <tr><td colspan="2" style="text-align: center;">(Фамилия, Имя, Отчество студента)</td></tr> <tr><td>Присвоена квалификация:</td><td>[QUAL]</td></tr> <tr><td colspan="2" style="text-align: center;">(наименование)</td></tr> <tr><td>Председатель ГЭК:</td><td>[HEAD_GEK]</td></tr> <tr><td>Члены ГЭК:</td><td>var str = " if (row["MEMBERS_GEK"] != " &amp;&amp; row["MEMBERS_GEK"] != null) { var array = row["MEMBERS_GEK"].split( ); }; for (var i = 0; i &lt; array.length; i++) str += array[i] + "&lt;br&gt;"; }; str;</td></tr> <tr><td colspan="2">var date = utils.getDateFormatted(row["DIPLOM_DATE"]); 'Выдан диплом ' + ((row["DIPLOM_QUAL"] == null) ? '_____': row["DIPLOM_QUAL"] + ' № ' + ((row["DIPLOM_NUM"] == null) ? '_____': row["DIPLOM_NUM"]) + ' or &lt;&lt; + date[0] + &gt;&gt;' + date[1] + ' ' + date[2] + ' г.'</td></tr> </table>	Решением Государственной экзаменационной комиссии		var date = utils.getDateFormatted(row["DECISION_DATE"]); 'or << + date[0] + >>' + date[1] + ' ' + date[2] + ' г. протокол № ' + ((row["DECISION_ORDER"] == null) ? '_____': row["DECISION_ORDER"])		студенту	[FIO]	(Фамилия, Имя, Отчество студента)		Присвоена квалификация:	[QUAL]	(наименование)		Председатель ГЭК:	[HEAD_GEK]	Члены ГЭК:	var str = " if (row["MEMBERS_GEK"] != " && row["MEMBERS_GEK"] != null) { var array = row["MEMBERS_GEK"].split( ); }; for (var i = 0; i < array.length; i++) str += array[i] + " "; }; str;	var date = utils.getDateFormatted(row["DIPLOM_DATE"]); 'Выдан диплом ' + ((row["DIPLOM_QUAL"] == null) ? '_____': row["DIPLOM_QUAL"] + ' № ' + ((row["DIPLOM_NUM"] == null) ? '_____': row["DIPLOM_NUM"]) + ' or << + date[0] + >>' + date[1] + ' ' + date[2] + ' г.'	
Выпускная квалификационная работа																																											
[FIO]																																											
(Фамилия, Имя, Отчество студента)																																											
Форма выпускной квалификационной работы:	[FORM_VKR]																																										
Тема:	[THEME]																																										
Руководителя:	[ADVISER]																																										
(Фамилия И.О.)																																											
Дата защиты:	var date = utils.getDateFormatted(row["DATE_DEFENCE"]); '<< + date[0] + >>' + date[1] + ' ' + date[2] + ' г.'																																										
Оценка:	[MARK]																																										
Председатель ГЭК:	[HEAD_GEK]																																										
Члены ГЭК:	var str = " if (row["MEMBERS_GEK"] != " && row["MEMBERS_GEK"] != null) { var array = row["MEMBERS_GEK"].split( ); }; for (var i = 0; i < array.length; i++) str += array[i] + " "; }; str;																																										
Label																																											
Решением Государственной экзаменационной комиссии																																											
var date = utils.getDateFormatted(row["DECISION_DATE"]); 'or << + date[0] + >>' + date[1] + ' ' + date[2] + ' г. протокол № ' + ((row["DECISION_ORDER"] == null) ? '_____': row["DECISION_ORDER"])																																											
студенту	[FIO]																																										
(Фамилия, Имя, Отчество студента)																																											
Присвоена квалификация:	[QUAL]																																										
(наименование)																																											
Председатель ГЭК:	[HEAD_GEK]																																										
Члены ГЭК:	var str = " if (row["MEMBERS_GEK"] != " && row["MEMBERS_GEK"] != null) { var array = row["MEMBERS_GEK"].split( ); }; for (var i = 0; i < array.length; i++) str += array[i] + " "; }; str;																																										
var date = utils.getDateFormatted(row["DIPLOM_DATE"]); 'Выдан диплом ' + ((row["DIPLOM_QUAL"] == null) ? '_____': row["DIPLOM_QUAL"] + ' № ' + ((row["DIPLOM_NUM"] == null) ? '_____': row["DIPLOM_NUM"]) + ' or << + date[0] + >>' + date[1] + ' ' + date[2] + ' г.'																																											

Рисунок 3.29 – форма ВКР

После этого создание всех форм электронной зачетной книжки можно считать завершенным. Теперь нужно протестировать реализованный продукт.

### 3.2 Тестирование электронной зачетной книжки

Для тестирования электронной зачетной книжки нужно просмотреть

созданный отчет в BIRT Viewer. Для тестирования не нужно отдельно скачивать и подключать этот модуль, он уже интегрирован в IDE Eclipse. Для просмотра отчета нужно нажать на кнопку View Report в верхнем меню. После этого в браузере откроется новая вкладка, в которой будет отображаться отчет.

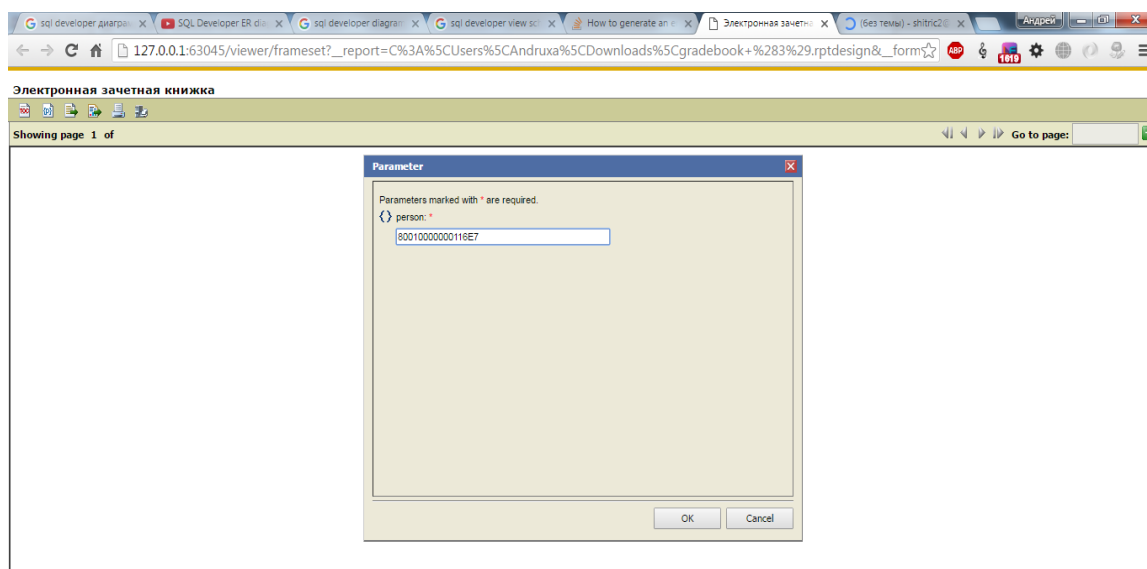


Рисунок 3.30 – Тест 1.

В открывшейся вкладке, которая показана на рисунке 3.30, программа потребует ввести ID студента, так как он не был указан в адресной строке в качестве параметра. После того, как был введен ID студента, необходимо нажать ОК.

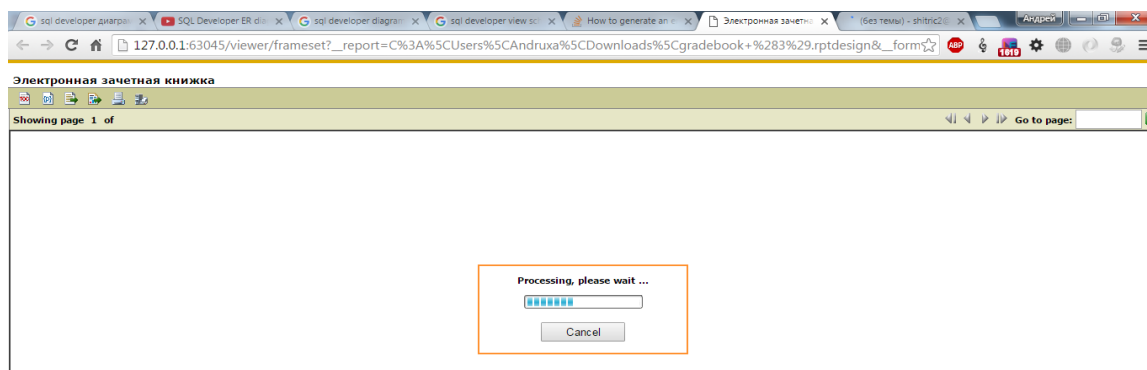


Рисунок 3.31 – Тест 2.

После этого на экране появится окно загрузки, которое показано на рисунке 3.31, в этот момент BIRT загружает все необходимые данные из БД и формирует отчет.

Электронная зачетная книжка

Showing page 1 of 8

Абдураимова Хабиба Комиповна  
(Фамилия, Имя, Отчество студента)

РЕЗУЛЬТАТЫ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ (ЭКЗАМЕНЫ)						РЕЗУЛЬТАТЫ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ (ЗАЧЕТЫ)					
№ п/п	Наименование дисциплины (модуля), раздела	Общее количество час. / эд.	Оценка	Дата сдачи экзамена	Фамилия преподавателя	№ п/п	Наименование дисциплины (модуля), раздела	Общее количество час. / эд.	Оценка	Дата сдачи экзамена	Фамилия преподавателя
1	История	3	Удовлетворительно	31.12.2015		1	Алгебраический логистический язык 1	1	Зачтено	21.12.2015	
2	Экспериментальная геохимия	3	Неуд	31.12.2015		2	Введение в профессию	2	Зачтено	21.12.2015	
3						3	Высшая математика 1	3	Зачтено	21.12.2015	
4						4	Дисциплинарный язык 1	2	Зачтено	21.12.2015	
5						5	Основы информатической культуры	2	Зачтено	28.09.2015	
6						6	Основы электронной техники	4	Зачтено	21.12.2015	
7						7	Первое квалификационное собеседование	2	Зачтено	21.12.2015	
8						8	Русский язык и культура речи	2	Зачтено	28.09.2015	
9						9	Физика: Электродинамика и квантовая физика	2	Зачтено	28.09.2015	
10						10	Физическая культура	2	Зачтено	21.12.2015	
11						11	Экология	2	Зачтено	21.12.2015	
12						12					

Количество экзаменов в текущем семестре 2; из них сдано 1, не сдано 1  
Количество зачетов в текущем семестре 11; из них сдано 9, не сдано 2

Рисунок 3.32 – Тест 3

После того, как BIRT соберет отчет, он будет выведен на экран, как показано на рисунке 3.32. Как видно из рисунка 53 на экране отображаются основные данные о студенте и данные об успеваемости студента.

Теперь в качестве ID студента укажем значение, которого нет в системе, например, null, как показано на рисунке 3.33.

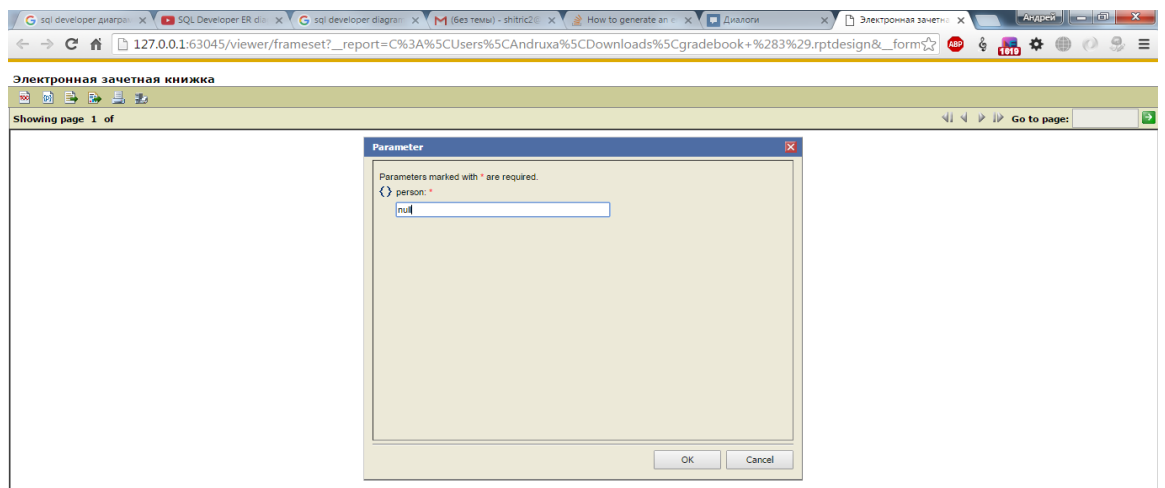


Рисунок 3.33 – Тест 4

Электронная зачетная книжка

Showing page 1 of 4

Go to page:

(Фамилия, имя, отчество студента (последнее – при наличии))  
 Код, направление подготовки (специальность) null null  
 Структурное подразделение null  
 Зачислен приказом от «\_\_» \_\_\_\_\_ г. № null  
 Руководитель организации, осуществляющей образовательную деятельность М.М. Кристал  
 Руководитель структурного подразделения null

ФАКУЛЬТАТИВНЫЕ					ДИСЦИПЛИНЫ						
РЕЗУЛЬТАТЫ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ (ЭКЗАМЕНЫ)					РЕЗУЛЬТАТЫ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ (ЗАЧЕТЫ)						
№ в.п.	Наименование дисциплины (модуля), раздела	Общее количество час. / з.ед.	Оценка	Дата сдачи экзамена	Фамилия преподавателя	№ в.п.	Наименование дисциплины (модуля), раздела	Общее количество час. / з.ед.	Оценка	Дата сдачи экзамена	Фамилия преподавателя
1						1					
2						2					
3						3					
4						4					
5						5					
6						6					
7						7					
8						8					
9						9					
10						10					

Количество экзаменов по факультативным дисциплинам null, из них славо 0, не славо 0  
 Количество зачетов по факультативным дисциплинам null, из них славо 0, не славо 0

Рисунок 3.34 – Тест 5

Как видно из рисунка 3.34 строится и выводится отчет, но на месте тех данных, которые должны были быть получены из БД ничего нет. Из этого следует вывод, что если в качестве ID студента будет указано несуществующее в базе значение, то работа системы нарушена не будет. Также из этого можно сделать вывод, что, если не будут получены некоторые данные из базы, которые должны участвовать в составлении отчета, это не повлияет на вывод остальных данных.

Все предыдущие отчеты выводились в формате HTML, теперь в качестве формата будет указано PDF. Для этого нужно в адресной строке параметру `__format` присвоить значение `pdf`.

Электронная зачетная книжка 1 / 8

Министерство образования и науки РФ  
ФГБОУ ВО «Тюльятинский государственный университет»

**Зачетная книжка № 2015002926**  
Абдурамонова Хабиба Комидовна  
(фамилия, имя, отчество студента (последнее – при наличии))  
Код, направление подготовки (специальность) 27.03.04 Управление в технических системах  
Структурное подразделение Институт энергетки и электротехники  
Зачислен приказом от «07» августа 2015 г. № 3120  
Руководитель организации, осуществляющей образовательную деятельность М.М. Кристалл  
Руководитель структурного подразделения В.А. Шаповалов

1-й семестр 2015 / 2016 учебного года **ПЕРВЫЙ** КУРС Абдурамонова Хабиба Комидовна  
(Фамилия, Имя, Отчество студента)

РЕЗУЛЬТАТЫ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ (ЭКЗАМЕНЫ)					РЕЗУЛЬТАТЫ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ (ЗАЧЕТЫ)						
№ п/п	Наименование дисциплины (модуля), раздела	Общее кол-во час. / з.е.д.	Оценка	Дата сдачи экзамена	Фамилия преподавателя	№ п/п	Наименование дисциплины (модуля), раздела	Общее кол-во час. / з.е.д.	Оценка	Дата сдачи экзамена	Фамилия преподавателя
1	История	3	Удовлетворительно	21.12.2015		1	Академический английский язык I	1		21.12.2015	
2	Начертательная геометрия	3	Неуд	21.12.2015		2	Высшее в профессионально	2	Зачтено	21.12.2015	
3						3	Высшая математика I	3	Зачтено	21.12.2015	
4						4	Иностраный язык I	2	Зачтено	21.12.2015	
5						5	Основы информационной культуры	2	Зачтено	28.09.2015	
6						6	Основы электронной техники	4	Зачтено	21.12.2015	
7						7	Права интеллектуальной собственности	2	Зачтено	21.12.2015	

Рисунок 3.35 – Тест 6

Как видно из рисунка 3.35 отчет выведен в формате PDF. Отчет также может быть выведен в другом формате, например, Word, Excel, Power Point и т.д. Никаких ошибок в работе замечено не было.

Теперь в качестве теста ID студента будет передаваться сразу в адресной строке. Для этого необходимо в адресной строке дописать &person="ID студента", как показано на рисунке 3.36. Через адресную строку также можно передавать другие параметры.



Министерство образования и науки РФ  
ФГБОУ ВО «Тольяттинский государственный университет»

**Зачетная книжка № 2015002926**  
Абдурамонова Хабива Комolloва  
(фамилия, имя, отчество студента (последнее – при наличии))

Код, направление подготовки (специальность) 27.03.04 Управление в технических системах  
Структурное подразделение Институт энергетики и электротехники  
Зачислен приказом от «07» августа 2015 г. № 3120

Руководитель организации, осуществляющей образовательную деятельность М.М. Кристал  
Руководитель структурного подразделения В.А. Шаповалов

1-й семестр 2015 / 2016 учебного года **ПЕРВЫЙ** **КУРС** Абдурамонова Хабива Комolloва  
(Фамилия, Имя, Отчество студента)

РЕЗУЛЬТАТЫ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ (ЭКЗАМЕНЫ)						РЕЗУЛЬТАТЫ ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ (ЗАЧЕТЫ)					
№ п/п	Наименование дисциплины (модуль), раздела	Общее количество час. /з.ед.	Оценка	Дата сдачи экзамена	Фамилия преподавателя	№ п/п	Наименование дисциплины (модуль), раздела	Общее количество час. /з.ед.	Оценка	Дата сдачи экзамена	Фамилия преподавателя
1	История	3	Удовлетворительно	21.12.2015		1	Академической английской язык 1	1		21.12.2015	
2	Направительная геометрия	3	Неуд.	21.12.2015		2	Высшие в профессии	2	Зачтено	21.12.2015	
3						3	Высшая математика 1	3	Зачтено	21.12.2015	
4						4	Инструментальный язык 1	2	Зачтено	21.12.2015	
5						5	Основы информационной культуры	2	Зачтено	28.09.2015	
6						6	Основы электронной техники	4	Зачтено	21.12.2015	
7						7	Право интеллектуальной собственности	2	Зачтено	21.12.2015	
8						8	Русский язык и культура речи	2	Зачтено	28.09.2015	
9						9	Физика: Электродинамика и оптика	2	Зачтено	28.09.2015	
10						10	Физическая культура	2	Зачтено	21.12.2015	

Рисунок 3.36 – Тест 7

Отчет был сформирован без необходимости вводить ID студента в соответствующем окне, как это было показано ранее.

В третьей главы в результате работы была реализована и протестирована электронная зачётная книжка студента, в которой отображается основная информация о студенте и его успеваемость за все время обучения в университете.

## **Заключение**

Использование баз данных и информационных систем становится неотъемлемой частью деятельности современного человека и функционирования различных организаций.

В данной выпускной квалификационной работе был выделен объект исследования, проанализирована предметная область, определены цели и задачи работы. Проведен обзор программных средств, позволяющих реализовать электронную зачетную книжку студента.

Анализ программных средств показал, что электронная зачетная книжка будет представлять собой отчет, созданный с помощью инструмента для создания отчетов BIRT. В качестве базы данных используется база данных компании Oracle.

Для удобства работы с данными в базе были созданы 9 виртуальных таблиц, в которых содержатся все необходимые данные об успеваемости студентов.

Была реализована и протестирована электронная зачетная книжка студента.

Таким образом была создана электронная зачетная книжка позволяет удаленно просматривать успеваемость учащегося в личном кабинете студента.

## Список использованной литературы

### *Учебники и учебные пособия*

1. Алапати С. Р. Oracle Database 11g. Руководство администратора баз данных/ Сэм Р. Алапати. – М.: Вильямс, 2010. – 1440 с.
2. Трухнов. SQL. Библия пользователя, 2-е издание/ Борис Трухнов, Алекс Кригель. – Вильямс.; Москва, 2011 – 752 с.
3. Гринвальд Р. Oracle 11g. Основы/ Р. Гринвальд, Р. Стаковьяк, Д. Стерн. – СПб.: Символ-плюс, 2011. – 464 с.
4. Грофф Р. Д. SQL. Полное руководство. 3-е издание/ Р.Д. Грофф, П.Н. Вайнберг, Э. Д. Оппель – М.: Вильямс, 2015. – 959 с.
5. Кайт Т. Oracle для профессионалов: архитектура и методики программирования. 3-е изд./ Т. Кайт, Д. Кун. – М.: Вильямс, 2016. – 960 с.
6. К. Дж. Дейт. SQL и реляционная теория. Как грамотно писать код на SQL/ К. Дж. Дейт. – Символ плюс.; Москва, 2011 – 474 с.
7. Кевин Е.Кляйн. SQL. Справочник, 3-е издание/ Кевин Е.Кляйн, Дэниэл Кляйн, Брэнд Хант. – Символ плюс.; Москва, 2011 – 652 с.
8. Наместников А.М. Построение баз данных в среде Oracle. Практический курс: Учеб. пособие для вузов/ А. М. Наместников – Ульяновск.: УлГТУ, 2011. – 118 с.
9. Санжей М. Секреты Oracle SQL/ М. Санжей – СПб.: Символ-плюс, 2014. – 368 с.
10. Стефанов С. JavaScript Patterns/ С. Стефанов. – СПб.: Символ-плюс, 2011. – 272 с.
11. Трухнов Б. SQL. Библия пользователя, 2 – е издание/ Алекс Кригель, Борис Трухнов. – Вильямс.; Москва, 2011 – 752 с.
12. Флэнаган Д. JavaScript. Подробное руководство, 6-е издание. : учеб. пособие / Д. Флэнаган. – СПб.: Символ-плюс, 2012. – 992 с.
13. Фриман Э. Изучаем программирование на JavaScript/ Э. Фриман, Э.

Робсон – СПб.: Питер, 2016 – 640 с.

*Электронные ресурсы*

14. Eclipse – The Eclipse Foundation open source community website [Электронный ресурс]: // Электрон. дан. – Business Intelligence & Reporting Tools (BIRT) Mastering BIRT – Eclipse – <http://www.eclipse.org/birt/presos/Eclipse%20BIRT%20Project%20Mastering%20BIRT.pdf> (дата обращения 11.03.2016)
15. Intefrace.ru [Электронный ресурс]: // Электрон. дан. – Создание специализированных отчетов с использованием BIRT и Rational Team Concert <http://www.interface.ru/home.asp?artId=28593> (дата обращения 11.03.2016)

*Литература на иностранном языке*

16. Закас Н. Professional JavaScript for Web Developers/ Н. Закас. – СПб.: Питер, 2015. – 960 с.
17. Peh Diana, BIRT: A Field Guide. - 3rd edition / Diana Peh, Nola Hague, Jane Tatchell, Addison-Wesley, 2011
18. Weathersby Jason, Tom Bondur, Iana Chatalbasheva, Integrating and Extending BIRT. - 3rd edition, Addison-Wesley, 2011
19. Vappoo Paul, BIRT by Example. – 1st edition, LULU PR, 2012
20. Powers Shelly, Learning JavaScript, O'Relly, 2011

## Код таблицы V\_STUD\_GRADE\_BOOK

```

CREATE VIEW GAL_PUBLIC.V_STUD_GRADE_BOOK
AS SELECT DISTINCT
    st.fcpersons,
    st.fnrec as ustud,
    st.fsdepartment as group_name,
    ffio,
    fnmb,
    login,
    fcodeprof,
    spec,
    inst,
    stgr,
    head,
    bup,
    ord.order_number,
    ord.ord_date,
    st.term,
    st.fyearent
FROM
    (SELECT
        s.fcpersons,
        s.fnrec,
        s.ffio,
        s.fsdepartment,
        s.fcodeprof,
        s.fspost
AS spec,
        s.fsfaculty
AS inst,
        CASE WHEN s.fsfaculty = 'Институт машиностроения'
            THEN 'Заместитель ректора - директор института машиностроения'
        ELSE CASE WHEN s.fsfaculty = 'Архитектурно-строительный институт'
            THEN 'Заместитель ректора - директор архитектурно-строительного ин-
ститута'
        ELSE
            CASE WHEN s.fsfaculty = 'Институт финансов, экономики и управ-
ления'
                THEN 'Заместитель ректора - директор института финансов, эконо-
мики и управления'
            ELSE
                CASE WHEN s.fsfaculty = 'Институт права'
                    THEN 'Заместитель ректора - директор института права'
                ELSE
                    CASE WHEN s.fsfaculty = 'Институт энергетики и электротех-
ники'
                        THEN 'Заместитель ректора - директор института энергетики
и электротехники'
                    ELSE
                        CASE WHEN s.fsfaculty = 'Гуманитарно-педагогический ин-
ститут'
                            THEN 'Заместитель ректора - директор гуманитарно-
педагогического института'
                        ELSE
                            CASE WHEN s.fsfaculty = 'Институт изобразительного и
декоративно-прикладного искусства'
                                THEN 'Заместитель ректора - директор института изоб-
разительного и декоративно-прикладного искусства'

```

```

ELSE
    CASE WHEN s.fsfaculty = 'Институт физической культуры
и спорта'
        THEN 'Заместитель ректора - директор института фи-
зической культуры и спорта'
        ELSE
            CASE WHEN s.fsfaculty = 'Институт математики, физи-
ки и информационных технологий'
                THEN 'Заместитель ректора - директор института
математики, физики и информационных технологий'
                ELSE
                    CASE WHEN s.fsfaculty = 'Институт химии и инже-
нерной экологии'
                        THEN 'Заместитель ректора - директор института
химии и инженерной экологии' END END END END END END END END AS dir,
gr.fname
stgr,
cur.fnrec
AS bup,
cur.fterm as term,
gr.fyearent,
decode(s.fdisdate, 0, NULL, to_char(gal_work.to_oradate(s.fdisdate),
'DD.MM.YYYY'))
AS disdate
FROM gal_work.u_student s, gal_work.u_studgroup gr,
gal_work.u_curriculum cur, gal_work.u_curr_group cg
WHERE s.fcstgr = gr.fnrec
AND cg.fcstgr = s.fcstgr
AND cur.fnrec = cg.fccurr
AND cur.fwtype = 1
) st
LEFT JOIN
(SELECT
fvstring AS login,
fcrec
FROM gal_work.attrval
WHERE
fcattnam = '80010000000005B9'
AND fwtable = '25001') logs
ON logs.fcrec = st.fcpersons
LEFT JOIN
(SELECT
substr(pf.FFIO, instr(pf.FFIO, ' ') + 1, 1) || '.' || substr(pf.FFIO,
instr(pf.FFIO, ' ', -1) + 1, 1) || ' ' ||
substr(pf.FFIO, 1, instr(pf.FFIO, ' ')) AS head,
dep.prof
FROM GAL_WORK.V_TO_OK_PF2 pf, GAL_WORK.V_TO_STAV stav,
GAL_WORK.V_TO_PF_STAFF dep, GAL_WORK.V_TO_PF_EMPLOY dp
WHERE stav.FSTAFFSTR = dep.FNREC AND stav.FPERSON = pf.FNREC AND
dp.FNREC = pf.FNREC
AND lower(dep.prof) LIKE '%заместитель ректора%') dr
ON st.dir = dr.prof
LEFT JOIN
(SELECT
stud.fcpersons,
pas.fnmb
FROM gal_work.u_student stud
LEFT JOIN gal_work.passports pas ON pas.fperson = stud.fcpersons
INNER JOIN gal_work.catalogs cat ON cat.fnrec = pas.fdocname AND
cat.fname = 'Зачетная книжка') kn
ON kn.fcpersons = st.fcpersons
LEFT JOIN
(SELECT

```

```

ord.person_id,
ord.order_number,
ord.ord_date
FROM      (SELECT DISTINCT
          r.person_id,
          o.order_number,
          o.ord_date
        FROM
          (SELECT
            cdoc.fnrec      AS rpd_id,
            cdoc.fcstr      AS appoint_id,
            t.fnoper        AS rpd_name,
            t.fwtdop,
            t.fcodoper      AS rpd_code,
            cdoc.fperson    AS person_id,
            pdoc.fcdoc      AS order_id,
            fsbottom        AS old_movement
          FROM gal_work.u_typepr t, gal_work.partdoc pdoc,
            (
              SELECT
                fnrec,
                fwrec,
                fsbottom,
                fperson,
                fcstr,
                fcpart,
                fobjtype,
                fobjnrec
              FROM gal_work.contdoc
              WHERE fcpart <> '8000000000000000'
              UNION ALL -- некоторые записи связаны с разделом не напрямую,
                а через промежуточную запись (для группы)
              SELECT
                c1.fnrec,
                c1.fwrec,
                c1.fsbottom,
                c1.fperson,
                c1.fcstr,
                c2.fcpart,
                c1.fobjtype,
                c1.fobjnrec
              FROM gal_work.contdoc c1, gal_work.contdoc c2
              WHERE c1.fcpart = '8000000000000000' AND c1.fcdopref =
c2.fnrec
            ) cdoc
          WHERE t.fcodoper = pdoc.ftypeoper
            AND t.fwtdop = pdoc.fwattr1 AND cdoc.fcpart = pdoc.fnrec
            AND pdoc.fcdoc IN (SELECT fnrec AS order_id
                              FROM gal_work.titledoc
                              WHERE fwstatus = 1
                             )
            AND (lower(t.fnoper) LIKE '%зачисл%' or lower(t.fnoper) like
'%восстан%')) r
        LEFT JOIN
          (SELECT
            fnrec
          AS order_id,
            fdcnmb
          AS order_number,
            gal_work.to_oradate(DECODE(fdodate, 0, NULL, NULL, NULL,
fdodate)) AS ord_date,
            fdodtext
          AS shorttext

```

```

FROM gal_work.titledoc
WHERE fwstatus = 1
) o
ON r.order_id = o.order_id) o JOIN
(SELECT
max(o.ord_date) AS mdate,
r.person_id
FROM
(SELECT
cdoc.fnrec AS rpd_id,
cdoc.fcstr AS appoint_id,
t.fnoper AS rpd_name,
t.fwtdop,
t.fcodoper AS rpd_code,
cdoc.fperson AS person_id,
pdoc.fcdoc AS order_id,
fsbottom AS old_movement
FROM gal_work.u_typepr t, gal_work.partdoc pdoc,
(
SELECT
fnrec,
fwrec,
fsbottom,
fperson,
fcstr,
fcpart,
fobjtype,
fobjnrec
FROM gal_work.contdoc
WHERE fcpart <> '8000000000000000'
UNION ALL -- некоторые записи связаны с разделом не напрямую,
а через промежуточную запись (для группы)
SELECT
c1.fnrec,
c1.fwrec,
c1.fsbottom,
c1.fperson,
c1.fcstr,
c2.fcpart,
c1.fobjtype,
c1.fobjnrec
FROM gal_work.contdoc c1, gal_work.contdoc c2
WHERE c1.fcpart = '8000000000000000' AND c1.fcdopref =
c2.fnrec
) cdoc
WHERE t.fcodoper = pdoc.ftypeoper
AND t.fwtdop = pdoc.fwattr1 AND cdoc.fcpart = pdoc.fnrec
AND pdoc.fcdoc IN (SELECT fnrec AS order_id
FROM gal_work.titledoc
WHERE fwstatus = 1
)
AND (lower(t.fnoper) LIKE '%зачисл%' or lower(t.fnoper) like
'%восстан%')) r
LEFT JOIN
(SELECT
fnrec
AS order_id,
fdocnmb
AS order_number,
gal_work.to_oradate(DECODE(fdocdate, 0, NULL, NULL, NULL,
fdocdate)) AS ord_date,
fdoctext
AS shorttext

```



```
FROM gal_work.titledoc
WHERE fwstatus = 1
) o
ON r.order_id = o.order_i
GROUP BY r.person_id) rpd
ON rpd.person_id = ord.person_id
AND rpd.mdate = ord.ord_date) ord
ON ord.person_id = st.fcpersons
```

## Приложение Б

## Код таблицы V\_STUD\_COUNT\_REPORT

```

CREATE VIEW GAL_PUBLIC.V_STUD_COUNT_REPORT
AS SELECT
  ffio as fio,
  fcpersons,
  sum(EKZ_1)      AS EKZ_1,
  sum(ZACH_1)     AS ZACH_1,
  sum(EKZ_2)      AS EKZ_2,
  sum(ZACH_2)     AS ZACH_2,
  sum(EKZ_3)      AS EKZ_3,
  sum(ZACH_3)     AS ZACH_3,
  sum(EKZ_4)      AS EKZ_4,
  sum(ZACH_4)     AS ZACH_4,
  sum(EKZ_5)      AS EKZ_5,
  sum(ZACH_5)     AS ZACH_5,
  sum(EKZ_6)      AS EKZ_6,
  sum(ZACH_6)     AS ZACH_6,
  sum(EKZ_7)      AS EKZ_7,
  sum(ZACH_7)     AS ZACH_7,
  sum(EKZ_8)      AS EKZ_8,
  sum(ZACH_8)     AS ZACH_8,
  sum(EKZ_9)      AS EKZ_9,
  sum(ZACH_9)     AS ZACH_9,
  sum(EKZ_10)     AS EKZ_10,
  sum(ZACH_10)    AS ZACH_10,
  sum(EKZ_11)     AS EKZ_11,
  sum(ZACH_11)    AS ZACH_11,
  sum(EKZ_12)     AS EKZ_12,
  sum(ZACH_12)    AS ZACH_12,
  sum(FAC_E)      AS FAC_E,
  sum(FAC_Z)      AS FAC_Z,
  sum(CURS)       AS CURS,
  sum(PRACT_1)    AS PRACT_1,
  sum(PRACT_2)    AS PRACT_2,
  sum(PRACT_3)    AS PRACT_3,
  sum(PRACT_4)    AS PRACT_4,
  sum(PRACT_5)    AS PRACT_5,
  sum(PRACT_6)    AS PRACT_6,
  sum(PRACT_7)    AS PRACT_7,
  sum(PRACT_8)    AS PRACT_8,
  sum(PRACT_9)    AS PRACT_9,
  sum(PRACT_10)   AS PRACT_10,
  sum(PRACT_11)   AS PRACT_11,
  sum(PRACT_12)   AS PRACT_12,
  sum(NIR)        AS NIR,
  null AS DATE_GOS,
  null AS ORDER_GOS
FROM (
  SELECT
    t.ffio,
    t.fcpersons,
    CASE WHEN t.FABBR = 'Э' AND t.comp NOT LIKE 'ФТД%' AND fsemester = 1
      THEN 1
    ELSE 0 END AS EKZ_1,
    CASE WHEN t.FABBR = 'З' AND t.comp NOT LIKE 'ФТД%' AND fsemester = 1
      THEN 1
    ELSE 0 END AS ZACH_1,
    CASE WHEN t.FABBR = 'Э' AND t.comp NOT LIKE 'ФТД%' AND fsemester = 2

```

```

THEN 1
ELSE 0 END AS EKZ_2,
CASE WHEN t.FABBR = '3' AND t.comp NOT LIKE 'ФТД%' AND fsemester = 2
THEN 1
ELSE 0 END AS ZACH_2,
CASE WHEN t.FABBR = '3' AND t.comp NOT LIKE 'ФТД%' AND fsemester = 3
THEN 1
ELSE 0 END AS EKZ_3,
CASE WHEN t.FABBR = '3' AND t.comp NOT LIKE 'ФТД%' AND fsemester = 3
THEN 1
ELSE 0 END AS ZACH_3,
CASE WHEN t.FABBR = '3' AND t.comp NOT LIKE 'ФТД%' AND fsemester = 4
THEN 1
ELSE 0 END AS EKZ_4,
CASE WHEN t.FABBR = '3' AND t.comp NOT LIKE 'ФТД%' AND fsemester = 4
THEN 1
ELSE 0 END AS ZACH_4,
CASE WHEN t.FABBR = '3' AND t.comp NOT LIKE 'ФТД%' AND fsemester = 5
THEN 1
ELSE 0 END AS EKZ_5,
CASE WHEN t.FABBR = '3' AND t.comp NOT LIKE 'ФТД%' AND fsemester = 5
THEN 1
ELSE 0 END AS ZACH_5,
CASE WHEN t.FABBR = '3' AND t.comp NOT LIKE 'ФТД%' AND fsemester = 6
THEN 1
ELSE 0 END AS EKZ_6,
CASE WHEN t.FABBR = '3' AND t.comp NOT LIKE 'ФТД%' AND fsemester = 6
THEN 1
ELSE 0 END AS ZACH_6,
CASE WHEN t.FABBR = '3' AND t.comp NOT LIKE 'ФТД%' AND fsemester = 7
THEN 1
ELSE 0 END AS EKZ_7,
CASE WHEN t.FABBR = '3' AND t.comp NOT LIKE 'ФТД%' AND fsemester = 7
THEN 1
ELSE 0 END AS ZACH_7,
CASE WHEN t.FABBR = '3' AND t.comp NOT LIKE 'ФТД%' AND fsemester = 8
THEN 1
ELSE 0 END AS EKZ_8,
CASE WHEN t.FABBR = '3' AND t.comp NOT LIKE 'ФТД%' AND fsemester = 8
THEN 1
ELSE 0 END AS ZACH_8,
CASE WHEN t.FABBR = '3' AND t.comp NOT LIKE 'ФТД%' AND fsemester = 9
THEN 1
ELSE 0 END AS EKZ_9,
CASE WHEN t.FABBR = '3' AND t.comp NOT LIKE 'ФТД%' AND fsemester = 9
THEN 1
ELSE 0 END AS ZACH_9,
CASE WHEN t.FABBR = '3' AND t.comp NOT LIKE 'ФТД%' AND fsemester = 10
THEN 1
ELSE 0 END AS EKZ_10,
CASE WHEN t.FABBR = '3' AND t.comp NOT LIKE 'ФТД%' AND fsemester = 10
THEN 1
ELSE 0 END AS ZACH_10,
CASE WHEN t.FABBR = '3' AND t.comp NOT LIKE 'ФТД%' AND fsemester = 11
THEN 1
ELSE 0 END AS EKZ_11,
CASE WHEN t.FABBR = '3' AND t.comp NOT LIKE 'ФТД%' AND fsemester = 11
THEN 1
ELSE 0 END AS ZACH_11,
CASE WHEN t.FABBR = '3' AND t.comp NOT LIKE 'ФТД%' AND fsemester = 12
THEN 1
ELSE 0 END AS EKZ_12,
CASE WHEN t.FABBR = '3' AND t.comp NOT LIKE 'ФТД%' AND fsemester = 12

```

```

THEN 1
ELSE 0 END AS ZACH_12,
CASE WHEN t.FABBR = '3' AND t.comp NOT LIKE 'ФТД%' AND t.comp LIKE 'ФТД%'
THEN 1
ELSE 0 END AS FAC_E,
CASE WHEN t.FABBR = '3' AND t.comp LIKE 'ФТД%'
THEN 1
ELSE 0 END AS FAC_Z,
CASE WHEN t.FABBR IN ('KP', 'KП')
THEN 1
ELSE 0 END AS CURS,
0 AS PRACT_1,
0 AS PRACT_2,
0 AS PRACT_3,
0 AS PRACT_4,
0 AS PRACT_5,
0 AS PRACT_6,
0 AS PRACT_7,
0 AS PRACT_8,
0 AS PRACT_9,
0 AS PRACT_10,
0 AS PRACT_11,
0 AS PRACT_12,
CASE WHEN nir = 1
THEN 1
ELSE 0 END AS NIR,
to_char(sysdate, 'DD.MM.YYYY') AS DATE_GOS,
227 AS ORDER_GOS
FROM (SELECT
s.ffio,
s.fcpersons,
fsemester,
CASE
WHEN cd."FSADDFLD[1]" LIKE '%ДБ%'
THEN substr(cd."FSADDFLD[1]", 1, length(cd."FSADDFLD[1]") - 2)
ELSE cd."FSADDFLD[1]" END AS comp,
tw.fname,
tw.FABBR,
CASE
WHEN comp.FABBR LIKE '%HMP%'
THEN 1
ELSE 0 END AS nir
FROM gal_work.u_student s
INNER JOIN gal_work.PERSONS p ON s.fcpersons = p.FNREC
INNER JOIN gal_work.APPOINTMENTS a ON s.fcpersons = a.FPERSON AND
p.FAPPOINTCUR = a.fnrec
INNER JOIN gal_work.U_CURR_DIS cd ON cd.FCCURR = a.FCDOPINF
INNER JOIN gal_work.U_CURR_DISCONTENT cdc ON cdc.FCCURR_DIS =
cd.fnrec
INNER JOIN gal_work.U_TYPEWORK tw ON tw.fnrec = cdc.FCTYPEWORK
INNER JOIN GAL_WORK.U_CURRICULUM c ON c.fnrec = a.FCDOPINF
INNER JOIN gal_work.CATALOGS cat ON cat.fnrec = c.FCFACULTY
INNER JOIN GAL_WORK.U_CURR_SEMESTER cs ON cs.FNREC = cdc.FCSEMESTER
INNER JOIN gal_work.U_DISCIPLINE d ON cd.FCDIS = d.fnrec
INNER JOIN gal_work.u_componentdis comp on cd.fccomponent =
comp.fnrec
WHERE cd.fcdis != '80010000000018B4'
GROUP BY s.ffio,
s.fcpersons,
fsemester,
CASE
WHEN cd."FSADDFLD[1]" LIKE '%ДБ%'
THEN substr(cd."FSADDFLD[1]", 1, length(cd."FSADDFLD[1]") - 2)

```

```
ELSE cd."FSADDFLD[1]" END,  
tw.fname,  
tw.FABBR,  
CASE  
WHEN comp.FABBR LIKE '%HIP%'  
  THEN 1  
ELSE 0 END) t  
)  
GROUP BY ffio,  
  fcpersons
```

## Код таблицы V\_STUD\_SEMESTERS

```

DBMS_METADATA attempting internal generator.
CREATE VIEW GAL_PUBLIC.V_STUD_SEMESTERS
AS SELECT
  cur.fcpersons,
  cur.ffio,
  sem.fwsemestr,
  cur.fwcourse,
  cur.fwyeared,
  rpd.rupdate AS ord_date,
  rpd.ord_num,
  rpd.ord_txt,
  cur.fterm
FROM (SELECT
  st.fcpersons,
  st.ffio,
  cur.fyeared AS fwyeared,
  cur.fcouse AS fwcourse,
  cur.fnrec AS rup,
  bup.fterm
  FROM gal_work.u_student st, gal_work.u_curr_group gr,
  gal_work.u_curriculum cur, gal_work.u_curriculum bup
  WHERE st.fcstgr = gr.fcstgr
  AND cur.fcparent = gr.fccurr
  and bup.fnrec = cur.fcparent
  AND cur.fwtype = 2) cur
LEFT JOIN (SELECT DISTINCT
  rup,
  fwyeared,
  fwsemestr
  FROM v_stud_vedomosti) sem
ON cur.rup = sem.rup AND sem.fwyeared = cur.fwyeared
LEFT JOIN
(SELECT
  person_id,
  ord_num,
  fdocyar AS fwyeared,
  ord_date AS rupdate,
  ord_txt
  FROM view_rpd rpd, (SELECT
  fnrec
AS order_id,
  fdocnmb
AS ord_num,
  fdocxt
AS ord_txt,
  fdocyar,
  gal_work.to_oradate(decode(fdodate, 0, NULL,
NULL, NULL, fdodate)) AS ord_date
  FROM gal_work.titledoc) o
  WHERE rpd.order_id = o.order_id
  AND (rpd_name LIKE '%Перевод на следующий курс%' OR rpd_name LIKE
'Перевод на старший курс%') AND
  person_id <> '8000000000000000') rpd
ON cur.fcpersons = rpd.person_id
AND rpd.fwyeared = cur.fwyeared

```

### Код таблицы V\_STUD\_TOTAL\_MARKS\_CURS

```

CREATE VIEW GAL_PUBLIC.V_STUD_TOTAL_MARKS_CURS

AS select
st.fcpersons,st.mark,st.fwmark,st.dis,st.fcdis,st.fwsemestr,st.rup,st.fwyear
d,st.fwhours,st.zet,st.tip,

st.teacher,st.tem,st.fnumdoc,st.datemark,st.ved, st.fwcourse,row_number()
over (PARTITION BY st.FCPERSONS, st.fwsemestr order by st.dis) as rn from

(select st.fcpersons,st.fwmark,c.fname as
mark,st.dis,st.fcdis,st.fwsemestr,st.rup,st.fwyear,d,st.fwhours,st.zet,st.tip
,

st.teacher,tem.tem,st.fnumdoc,st.datemark,st.ved,st.num, st.fwcourse

from (select

m.fcpersons,m.fwmark,m.fcmark,dis.fname as dis,l.fwsemestr,l.fccur as
rup,l.fwyear,d,l.fwhours,l."FDADDFLD[1]" as zet,

tip.fname as tip,substr(per.FFIO,1,instr(per.FFIO,' ')) as teacher,l.fnumdoc
,st.fwcourse,

decode(m.fdatemark, 0, NULL,
to_char(gal_work.to_oradate(m.fdatemark),'DD.MM.YYYY')) as date-
mark,l.fwtype,l.fcdis,l.fcparent as ved,

row_number() over (partition by st.fcpersons,l.fcdis,l.fwsemestr order by de-
code(m.fdatemark, 0, NULL,
to_char(gal_work.to_oradate(m.fdatemark),'DD.MM.YYYY')) desc) as num

from gal_work.u_discipline dis,gal_work.u_student st,gal_work.u_marks
m,gal_work.u_list l

left join gal_work.persons per on l.fcexaminer=per.fnrec

left join gal_work.u_typework tip on l.fctypework=tip.fnrec

where m.fcpersons=st.fcpersons

and l.fcstgr=st.fcstgr

and m.fclist=l.fnrec

and l.fcdis=dis.fnrec

and fwmark= (select max(fwmark) from gal_work.u_marks ma,gal_work.u_list
li,gal_work.u_discipline disc,gal_work.u_student stu

where ma.fcpersons=stu.fcpersons

and ma.fclist=li.fnrec

and li.fcdis=disc.fnrec

and li.fcdis=l.fcdis

```

```
and stu.fcpersons=st.fcpersons
group by fcdis) ) st
left join
      (select fname,fnrec from gal_work.catalogs) c
      on c.fnrec=st.fcmark
left join
      (select fsname as tem,fcdis,fcauthor,fwtypedoc from gal_work.u_db_diploma)
      tem
      on
      st.fcdis=tem.fcdis and
      st.fwtype=tem.fwtypedoc and
      st.fcpersons=tem.fcauthor
where num=1
and tip in ('Курсовые работы','Курсовой проект') ) st
```



## Приложение Д

**Код таблицы V\_STUD\_TOTAL\_MARKS\_EXAM**

```

CREATE VIEW GAL_PUBLIC.V_STUD_TOTAL_MARKS_EXAM
AS SELECT
  st.fcpersons as fcpersons_e,
  st.mark as mark_e,
  st.fwmark as fwmark_e,
  st.dis as dis_e,
  st.fcdis as fcdis_e,
  st.fwsemestr as fwsemestr_e,
  st.rup as rup_e,
  st.fweared as fweared_e,
  st.fwhours as fwhours_e,
  st.zet as zet_e,
  st.tip as tip_e,
  st.teacher as teacher_e,
  st.tem as tem_e,
  st.fnumdoc as fnumdoc_e,
  st.datemark as datemark_e,
  st.ved as ved_e,
  st.fwcourse as fwcourse_e,
  row_number() over (PARTITION BY st.FCPERSONS, st.fwsemestr order by
st.dis) as rn_e
FROM
  (SELECT
    st.fcpersons,
    c.fname as mark,
    st.fwmark,
    st.dis,
    st.fcdis,
    st.fwsemestr,
    st.rup,
    st.fweared,
    st.fwhours,
    st.zet,
    st.tip,
    st.teacher,
    tem.tem,
    st.fnumdoc,
    st.datemark,
    st.ved,
    st.num,
    st.fwcourse
  FROM (SELECT
    m.fcpersons,
    m.fcmark,
    m.fwmark,
    dis.fname
  AS dis,
    l.fwsemestr,
    l.fccur
  AS rup,
    l.fweared,
    l.fwhours,
    l."FDADDFLD[1]"
  AS zet,
    tip.fname
  AS tip,
    substr(per.FFIO, 1, instr(per.FFIO,

```

```

' '))
AS teacher,
    l.fnumdoc,
    st.fwcourse,
    decode(m.fdatemark, 0, NULL,
to_char(gal_work.to_oradate(m.fdatemark),
                                                'DD.MM.YYYY'))
AS datemark,
    l.fwtype,
    l.fcdis,
    l.fcparent
AS ved,
    row_number()
    OVER (PARTITION BY st.fcpersons, l.fcdis, l.fwsemestr
    ORDER BY decode(m.fdatemark, 0, NULL,
to_char(gal_work.to_oradate(m.fdatemark),
                                                'DD.MM.YYYY'))
DESC)
AS num
FROM gal_work.u_discipline dis, gal_work.u_student st,
gal_work.u_marks m, gal_work.u_list l
LEFT JOIN gal_work.persons per ON l.fcexaminer = per.fnrec
LEFT JOIN gal_work.u_typework tip ON l.fctypework = tip.fnrec
WHERE m.fcpersons = st.fcpersons
AND l.fcstgr=st.fcstgr
AND m.fclist = l.fnrec
AND l.fcdis = dis.fnrec
AND fwmark = (SELECT max(fwmark)
FROM gal_work.u_marks ma, gal_work.u_list li,
gal_work.u_discipline disc,
gal_work.u_student stu
WHERE ma.fcpersons = stu.fcpersons
AND ma.fclist = li.fnrec
AND li.fcdis = disc.fnrec
AND li.fcdis = l.fcdis
AND stu.fcpersons = st.fcpersons
GROUP BY fcdis)
) st
left join
(select fname,fnrec from gal_work.catalogs) c
on c.fnrec=st.fcmark
LEFT JOIN
(SELECT
fccurr,
fcdis,
"FSADDFLD[1]" as comp
FROM gal_work.u_curr_dis) ucd
on ucd.fcdis=st.fcdis
AND ucd.fccurr=st.rup
LEFT JOIN
(SELECT
fsname AS tem,
fcdis,
fcauthor,
fwtypedoc
FROM gal_work.u_db_diploma) tem
ON
st.fcdis = tem.fcdis AND
st.fwtype = tem.fwtypedoc AND
st.fcpersons = tem.fcauthor
WHERE num = 1
AND tip = 'Экзамены'
AND ucd.comp not like 'ФТД%'
) st

```

**Приложение Е****Код таблицы V\_STUD\_TOTAL\_MARKS\_FACULT\_EXAM**

```
CREATE VIEW GAL_PUBLIC.V_STUD_TOTAL_MARKS_FACULT_EXAM
AS SELECT
    st.fcpersons as fcpersons_e,
    st.mark as mark_e,
    st.fwmark as fwmark_e,
    st.dis as dis_e,
    st.fcdis as fcdis_e,
    st.fwsemestr as fwsemestr_e,
    st.rup as rup_e,
    st.fweared as fweared_e,
    st.fwhours as fwhours_e,
    st.zet as zet_e,
    st.tip as tip_e,
    st.teacher as teacher_e,
    st.tem as tem_e,
    st.fnumdoc as fnumdoc_e,
    st.datemark as datemark_e,
    st.ved as ved_e,
    st.fwcourse as fwcourse_e,
    row_number() over (PARTITION BY st.FCPERSONS, st.fwsemestr order by
st.dis) as rn_e
FROM
    (SELECT
        st.fcpersons,
        c.fname as mark,
        st.fwmark,
        st.dis,
        st.fcdis,
        st.fwsemestr,
```

```

st.rup,
st.fweared,
st.fwhours,
st.zet,
st.tip,
st.teacher,
tem.tem,
st.fnumdoc,
st.datemark,
st.ved,
st.num,
st.fwcourse
FROM (SELECT
      m.fcpersons,
      m.fcmark,
      m.fwmark,
      dis.fname
AS dis,
      l.fwsemestr,
      l.fccur
AS rup,
      l.fweared,
      l.fwhours,
      l."FDADDFLD[1]"
AS zet,
      tip.fname
AS tip,
      substr(per.FFIO, 1, instr(per.FFIO,
                                ' '))
AS teacher,
      l.fnumdoc,
      st.fwcourse,
      decode(m.fdatemark, 0, NULL,
to_char(gal_work.to_oradate(m.fdatemark),

```

```

AS datemark,
    l.fwtype,
    l.fcdis,
    l.fcparent
AS ved,
    row_number()
    OVER (PARTITION BY st.fcpersons, l.fcdis, l.fwsemestr
        ORDER BY decode(m.fdatemark, 0, NULL,
to_char(gal_work.to_oradate(m.fdatemark),
                                                'DD.MM.YYYY'))
DESC)
    AS num
    FROM gal_work.u_discipline dis, gal_work.u_student st,
gal_work.u_marks m, gal_work.u_list l
    LEFT JOIN gal_work.persons per ON l.fcexaminer = per.fnrec
    LEFT JOIN gal_work.u_typework tip ON l.fctypework = tip.fnrec
WHERE m.fcpersons = st.fcpersons
and l.fcstgr=st.fcstgr
    AND m.fclist = l.fnrec
    AND l.fcdis = dis.fnrec
    AND fwmark = (SELECT max(fwmark)
    FROM gal_work.u_marks ma, gal_work.u_list li,
gal_work.u_discipline disc,
    gal_work.u_student stu
    WHERE ma.fcpersons = stu.fcpersons
    AND ma.fclist = li.fnrec
    AND li.fcdis = disc.fnrec
    AND li.fcdis = l.fcdis
    AND stu.fcpersons = st.fcpersons
    GROUP BY fcdis)
) st
left join
    (select fname,fnrec from gal_work.catalogs) c
on c.fnrec=st.fcmark

```

```
LEFT JOIN
(SELECT
  fcurr,
  fcdis,
  "FSADDFLD[1]" as comp
FROM gal_work.u_curr_dis) ucd
on ucd.fcdis=st.fcdis
AND ucd.fcurr=st.rup
LEFT JOIN
(SELECT
  fsname AS tem,
  fcdis,
  fcauthor,
  fwtypedoc
FROM gal_work.u_db_diploma) tem
ON
  st.fcdis = tem.fcdis AND
  st.fwtype = tem.fwtypedoc AND
  st.fcpersons = tem.fcauthor
WHERE num = 1
  AND tip = 'Экзамены'
  AND ucd.comp like 'ФТД%'
  ) st
```

**Приложение Ж****Код таблицы V\_STUD\_TOTAL\_MARKS\_FACULT\_ZACH**

```
CREATE VIEW GAL_PUBLIC.V_STUD_TOTAL_MARKS_FACULT_ZACH
AS SELECT
    st.fcpersons,
    st.mark,
    st.fwmark,
    st.dis,
    st.fcdis,
    st.fwsemestr,
    st.rup,
    st.fweared,
    st.fwhours,
    st.zet,
    st.tip,
    st.teacher,
    st.tem,
    st.fnumdoc,
    st.datemark,
    st.ved,
    st.fwcourse,
    row_number() over (PARTITION BY st.FCPERSONS, st.fwsemestr order by
st.dis) as rn
FROM
    (SELECT
        st.fcpersons,
        c.fname as mark,
        st.fwmark,
        st.dis,
        st.fcdis,
        st.fwsemestr,
```

```

    st.rup,
    st.fweared,
    st.fwhours,
    st.zet,
    st.tip,
    st.teacher,
    tem.tem,
    st.fnumdoc,
    st.datemark,
    st.ved,
    st.num,
    st.fwcourse
FROM (SELECT
        m.fcpersons,
        m.fcmark,
        m.fwmark,
        dis.fname
AS dis,
        l.fwsemestr,
        l.fccur
AS rup,
        l.fweared,
        l.fwhours,
        l."FDADDFLD[1]"
AS zet,
        tip.fname
AS tip,
        substr(per.FFIO, 1, instr(per.FFIO,
                                ' '))
AS teacher,
        l.fnumdoc,
        st.fwcourse,
        decode(m.fdatemark, 0, NULL,
to_char(gal_work.to_oradate(m.fdatemark),

```



```

AS datemark,
    l.fwtype,
    l.fcdis,
    l.fcparent
AS ved,
    row_number()
    OVER (PARTITION BY st.fcpersons, l.fcdis, l.fwsemestr
        ORDER BY decode(m.fdatemark, 0, NULL,
to_char(gal_work.to_oradate(m.fdatemark),
                                                    'DD.MM.YYYY'))
DESC)
    AS num
    FROM gal_work.u_discipline dis, gal_work.u_student st,
gal_work.u_marks m, gal_work.u_list l
    LEFT JOIN gal_work.persons per ON l.fcexaminer = per.fnrec
    LEFT JOIN gal_work.u_typework tip ON l.fctypework = tip.fnrec
WHERE m.fcpersons = st.fcpersons
and l.fcstgr=st.fcstgr
    AND m.fclist = l.fnrec
    AND l.fcdis = dis.fnrec
    AND fwmark = (SELECT max(fwmark)
    FROM gal_work.u_marks ma, gal_work.u_list li,
gal_work.u_discipline disc,
    gal_work.u_student stu
    WHERE ma.fcpersons = stu.fcpersons
    AND ma.fclist = li.fnrec
    AND li.fcdis = disc.fnrec
    AND li.fcdis = l.fcdis
    AND stu.fcpersons = st.fcpersons
    GROUP BY fcdis)
) st
left join
    (select fname,fnrec from gal_work.catalogs) c
on c.fnrec=st.fcmark

```

```
LEFT JOIN
(SELECT
  fcurr,
  fcdis,
  "FSADDFLD[1]" as comp
FROM gal_work.u_curr_dis) ucd
on ucd.fcdis=st.fcdis
AND ucd.fcurr=st.rup
LEFT JOIN
(SELECT
  fsname AS tem,
  fcdis,
  fcauthor,
  fwtypedoc
FROM gal_work.u_db_diploma) tem
ON
  st.fcdis = tem.fcdis AND
  st.fwtype = tem.fwtypedoc AND
  st.fcpersons = tem.fcauthor
WHERE num = 1
  AND tip = 'Зачёты'
  AND ucd.comp like 'ФТД%') st
```

**Приложение 3****Код таблицы V\_STUD\_TOTAL\_MARKS\_NIR**

```
CREATE VIEW GAL_PUBLIC.V_STUD_TOTAL_MARKS_NIR
AS SELECT
  st.fcpersons,
  st.mark,
  st.fwmark,
  st.dis,
  st.fcdis,
  st.fwsemestr,
  st.rup,
  st.fwyearned,
  st.fwhours,
  st.zet,
  st.tip,
  st.teacher,
  st.tem,
  st.fnumdoc,
  st.datemark,
  st.ved,
  st.fwcourse,
  row_number() over (PARTITION BY st.FCPERSONS, st.fwsemestr order by st.dis) as rn
FROM
  (SELECT
    st.fcpersons,
    c.fname as mark,
    st.fwmark,
    st.dis,
```

```

st.fcdis,
st.fwsemestr,
st.rup,
st.fwyeaded,
st.fwhours,
st.zet,
st.tip,
st.teacher,
tem.tem,
st.fnumdoc,
st.datemark,
st.ved,
st.num,
st.fwcourse,
row_number()
OVER (PARTITION BY st.FCPERSONS, st.fwsemestr
ORDER BY st.dis) AS rn
FROM (SELECT
m.fcpersons,
m.fcmark,
m.fwmark,
dis.fname AS dis,
l.fwsemestr,
l.fccur AS rup,
l.fwyeaded,
l.fwhours,
l."FDADDFLD[1]" AS zet,
tip.FABBR AS tip,

```

```

substr(per.FFIO, 1, instr(per.FFIO,
                                ' '))                AS teacher,
l.fnumdoc,
st.fwcourse,
decode(m.fdatemark, 0, NULL, to_char(gal_work.to_oradate(m.fdatemark),
                                'DD.MM.YYYY'))        AS datemark,
l.fwtype,
l.fcdis,
l.fcparent                AS ved,
row_number()
OVER (PARTITION BY st.fcpersons, l.fcdis, l.fwsemestr
      ORDER BY decode(m.fdatemark, 0, NULL, to_char(gal_work.to_oradate(m.fdatemark),
'DD.MM.YYYY')) DESC) AS num
FROM gal_work.u_discipline dis, gal_work.u_student st, gal_work.u_marks m, gal_work.u_list l
LEFT JOIN gal_work.persons per ON l.fcexaminer = per.fnrec
LEFT JOIN gal_work.u_curr_dis ucd ON l.fcdis = ucd.fcdis AND l.fccur = ucd.fccurr
LEFT JOIN gal_work.u_componentdis tip ON ucd.fccomponent = tip.fnrec AND tip.FABBR IS NOT
NULL
WHERE m.fcpersons = st.fcpersons
and l.fcstgr=st.fcstgr
AND m.fclist = l.fnrec
AND l.fcdis = dis.fnrec
AND fwmark = (SELECT max(fwmark)
              FROM gal_work.u_marks ma, gal_work.u_list li, gal_work.u_discipline disc,
              gal_work.u_student stu
              WHERE ma.fcpersons = stu.fcpersons
              AND ma.fclist = li.fnrec
              AND li.fcdis = disc.fnrec
              AND li.fcdis = l.fcdis

```

```
        AND stu.fcpersons = st.fcpersons
        GROUP BY fcdis)
) st
left join
(select fname,fnrec from gal_work.catalogs) c
on c.fnrec=st.fcmark
LEFT JOIN
(SELECT
  fsname AS tem,
  fcdis,
  fcauthor,
  fwtypedoc
FROM gal_work.u_db_diploma) tem
ON
  st.fcdis = tem.fcdis AND
  st.fwtype = tem.fwtypedoc AND
  st.fcpersons = tem.fcauthor
WHERE num = 1
  AND tip LIKE '%НИР%') st
```

**Приложение И****Код таблицы V\_STUD\_TOTAL\_MARKS\_ZACH**

```
CREATE VIEW GAL_PUBLIC.V_STUD_TOTAL_MARKS_ZACH
```

```
AS SELECT
```

```
    st.fcpersons,
```

```
    st.mark,
```

```
    st.fwmark,
```

```
    st.dis,
```

```
    st.fcdis,
```

```
    st.fwsemestr,
```

```
    st.rup,
```

```
    st.fwyearned,
```

```
    st.fwhours,
```

```
    st.zet,
```

```
    st.tip,
```

```
    st.teacher,
```

```
    st.tem,
```

```
    st.fnumdoc,
```

```
    st.datemark,
```

```
    st.ved,
```

```
    st.fwcourse,
```

```
    row_number() over (PARTITION BY st.FCPERSONS, st.fwsemestr order by st.dis) as rn
```

```
FROM
```

```
(SELECT
```

```
    st.fcpersons,
```

```
    c.fname as mark,
```

```
    st.fwmark,
```





```

l.fnumdoc,
st.fwcourse,
decode(m.fdatemark, 0, NULL, to_char(gal_work.to_oradate(m.fdatemark),
                                     'DD.MM.YYYY'))          AS datemark,
l.fwtype,
l.fcdis,
l.fcparent          AS ved,
row_number()
OVER (PARTITION BY st.fcpersons, l.fcdis, l.fwsemestr
      ORDER BY decode(m.fdatemark, 0, NULL, to_char(gal_work.to_oradate(m.fdatemark),
                                                     'DD.MM.YYYY')) DESC)          AS num
FROM gal_work.u_discipline dis, gal_work.u_student st, gal_work.u_marks m, gal_work.u_list l
LEFT JOIN gal_work.persons per ON l.fcexaminer = per.fnrec
LEFT JOIN gal_work.u_typework tip ON l.fctypework = tip.fnrec
WHERE m.fcpersons = st.fcpersons
and l.fcstgr=st.fcstgr
AND m.fclist = l.fnrec
AND l.fcdis = dis.fnrec
AND fwmark = (SELECT max(fwmark)
              FROM gal_work.u_marks ma, gal_work.u_list li, gal_work.u_discipline disc,
              gal_work.u_student stu
              WHERE ma.fcpersons = stu.fcpersons
              AND ma.fclist = li.fnrec
              AND li.fcdis = disc.fnrec
              AND li.fcdis = l.fcdis
              AND stu.fcpersons = st.fcpersons
              GROUP BY fcdis)
) st

```

```
left join
  (select fname,fnrec from gal_work.catalogs) c
  on c.fnrec=st.fcmark
LEFT JOIN  (SELECT
fcurr,
fcdis,
"FSADDFLD[1]" as comp
FROM gal_work.u_curr_dis) ucd
on ucd.fcdis=st.fcdis
AND ucd.fcurr=st.rup
LEFT JOIN
(SELECT
  fname AS tem,
  fcdis,
  fcauthor,
  fwtypedoc
FROM gal_work.u_db_diploma) tem
ON
  st.fcdis = tem.fcdis AND
  st.fwtype = tem.fwtypedoc AND
  st.fcpersons = tem.fcauthor
WHERE num = 1
  AND tip = 'Зачёты'
  AND ucd.comp not like 'ФТД%' st
```