Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ

ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

**Кафедра информационных систем и технологий**

**администрирование баз данных и приложений**

**Методические указания к выполнению курсового проекта  
для студентов специальности 1-40 05 01-03   
«Информационные системы и технологии**

**(издательско-полиграфический комплекс)»**

Минск 2018

УДК \_\_\_\_\_\_\_\_\_\_\_

ББК \_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_

Рассмотрены и рекомендованы к изданию редакционно-издательским советом университета

Рецензенты:

кандидат физико-математических наук, доцент, доцент кафедры управления информационными ресурсами Академии управления при Президенте Республики Беларусь, *Н. И. Белодед*;

кандидат технических наук, доцент, доцент кафедры программной инженерии, *Н. В. Пацей*.

**Администрирование баз данных и приложений :**

\_\_\_\_ методические указания к выполнению курсового проекта для студентов специальности 1-40 05 01-03 «Информационные системы и технологии» /сост. : Е. А. Блинова, Л. С. Мороз. – Минск : БГТУ, 2018. – \_\_\_ с.

Пособие предназначено для студентов, выполняющих курсовой проект по дисциплине «Администрирование баз данных и приложений» и содержит план работы, требования к курсовому проекту, описание структуры пояснительной записи и демонстрационный пример. Результатом курсового проекта является разработанная студентом база данных в СУБД Oracle, с применением какой-либо технологии, применяемой в СУБД Oracle 12с и приложение для демонстрации возможностей разработанной БД.

**УДК \_\_\_\_\_\_\_\_\_\_\_**

**ББК \_\_\_\_\_\_\_\_\_\_\_\_\_**

© УО «Белорусский государственный

технологический университет», 2018

### Содержание

[Содержание 3](#_Toc525026444)

[ВВЕДЕНИЕ 5](#_Toc525026445)

[1. ТРЕБОВАНИЯ К курсовому проекту 6](#_Toc525026446)

[1.1 Этапы и сроки выполнения курсового проекта 6](#_Toc525026447)

[1.2 Минимальные требования к курсовому проекту 7](#_Toc525026448)

[1.3 Дополнительные требования к курсовому проекту 7](#_Toc525026449)

[2 ОСНОВНЫЕ РАЗДЕЛЫ ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ 9](#_Toc525026450)

[2.1 Структура пояснительной записки 9](#_Toc525026451)

[2.2 Титульный лист 9](#_Toc525026452)

[2.3 Задание на курсовую работу 9](#_Toc525026453)

[2.4 Содержание пояснительной записки 9](#_Toc525026454)

[2.5 Введение 10](#_Toc525026455)

[2.6 Основная часть пояснительной записки 10](#_Toc525026456)

[2.6.1 Постановка задачи 10](#_Toc525026457)

[2.6.2 Установка и настройка сервера СУБД 11](#_Toc525026458)

[2.6.3 Проектирование инфраструктуры базы данных и порядок авторизации пользователей к базе данных 11](#_Toc525026459)

[2.6.4 Проектирование и реализация объектов базы данных 12](#_Toc525026460)

[2. 6.4.1 Проектирование и реализация таблиц 12](#_Toc525026461)

[2. 6.4.2 Проектирование и реализация представлений 13](#_Toc525026462)

[2. 6.4.3 Проектирование и реализация индексов 13](#_Toc525026463)

[2. 6.4.4 Проектирование и реализация хранимых процедур и функций 13](#_Toc525026464)

[2. 6.4.4 Проектирование и реализация триггеров 14](#_Toc525026465)

[2.6.5 Изучение и применение технологии в созданной базе данных 14](#_Toc525026466)

[2.6.6 Импорт и генерация тестовых данных 15](#_Toc525026467)

[2.6.7 Резервное копирование и восстановление данных 15](#_Toc525026468)

[2.6.8 Разработка демонстрационного приложения 16](#_Toc525026469)

[2.7 Заключение 16](#_Toc525026470)

[2.8 Графический материал 16](#_Toc525026471)

[2.9 Список использованных источников 17](#_Toc525026472)

[2.10 Приложения 17](#_Toc525026473)

[3 ОФОРМЛЕНИЕ ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ 18](#_Toc525026474)

[3.1 Общие требования 18](#_Toc525026475)

[3.2 Структурные элементы записки 18](#_Toc525026476)

[3.3 Нумерация страниц 19](#_Toc525026477)

[3.4 Перечисления 19](#_Toc525026478)

[3.5 Изложение текста 20](#_Toc525026479)

[3.6 Формулы 20](#_Toc525026480)

[3.7 Примечания 20](#_Toc525026481)

[3.8 Рисунки 21](#_Toc525026482)

[3.9 Таблицы 21](#_Toc525026483)

[3.10 Ссылки 22](#_Toc525026484)

[3.11 Приложения 22](#_Toc525026485)

[3.12 Список использованных источников 22](#_Toc525026486)

[4 Примерная тематика курсовых проектов 24](#_Toc525026487)

[4.1 Общие замечания 24](#_Toc525026488)

[4.2 Список областей применения баз данных 24](#_Toc525026489)

[4.3 Примерный список рекомендуемых технологий для самостоятельного изучения 25](#_Toc525026490)

[5 Демонстрационный Пример 26](#_Toc525026491)

[5.1 Постановка задачи 26](#_Toc525026492)

[5.2 Логины и пользователи 27](#_Toc525026493)

[5.3 Проектирование базы данных 27](#_Toc525026494)

[5.4 Процедуры работы с данными 28](#_Toc525026495)

[5.5 Процедуры извлечения и записи данных в XML 28](#_Toc525026496)

[5.6 Технология SQL Server Reporting Services (SSRS) 29](#_Toc525026497)

[5.7 Демонстрационное приложение 30](#_Toc525026498)

[5.8 ПРИЛОЖЕНИЯ 32](#_Toc525026499)

[Приложение А 32](#_Toc525026500)

[Приложение Б 32](#_Toc525026501)

[Приложение В 33](#_Toc525026502)

[Приложение А (обязательное) 35](#_Toc525026503)

[Приложение Б (обязательное) 36](#_Toc525026504)

# ВВЕДЕНИЕ

Целью курсового проекта является освоение навыков проектирования и администрирования базы данных. В процессе выполнения работы студент должен применить теоретические знания, полученные при изучении дисциплины «Администрирование баз данных и приложений», спроектировав базу данных, самостоятельно освоить новые технологии, применяемые в СУБД, и разработать приложение для демонстрации.

Задание предполагает:

* установку и настройку сервера СУБД Oracle 12c,
* проектирование инфраструктуры базы данных,
* создание необходимых объектов,
* загрузку тестовых данных для демонстрации и проведения оптимизации,
* самостоятельное изучение и применение определенной технологии в созданной базе данных,
* самостоятельное изучение и применение резервного копирования и восстановления данных в созданной базе данных,
* разработку небольшого приложения для демонстрации.

Для успешной защиты курсового проекта студент должен:

* установить и настроить сервер СУБД,
* спроектировать инфраструктуру базы данных,
* создать необходимые объекты,
* импортировать или сгенерировать тестовые данные для демонстрации,
* исследовать и при необходимости оптимизировать структуру запросов к БД,
* самостоятельно изучить и применить определенную технологию в созданной базе данных,
* произвести резервное копирование и восстановление данных,
* разработать небольшое приложение для демонстрации,
* подготовить пояснительную записку к курсовому проекту.

Предлагаемое пособие содержит требования к курсовому проекту (раздел 1), структуру и описание содержимого пояснительной записки (раздел 2), правила ее оформления (раздел 3), примерный список тематик областей применения баз данных и технологий (раздел 4) и демонстрационный пример (раздел 5).

# 1. ТРЕБОВАНИЯ К курсовому проекту

## 1.1 Этапы и сроки выполнения курсового проекта

Тематика курсовых проектов представлена отдельным списком (см. раздел 4). Курсовой проект разрабатывается студентами в сроки, предусмотренные графиком учебного процесса. На выполнение курсового проекта в учебном плане предусмотрено 40 часов. Курсовой проект должен быть выполнен в течении 11 учебных недель со следующим распределением объемов выполнения задания (в процентах, с нарастающим итогом).

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Неделя | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 |
| % | 5 | 15 | 25 | 35 | 45 | 55 | 65 | 75 | 85 | 95 | 100 |

Предполагается, что в качестве СУБД выбирается Oracle 12c, однако, по согласованию с преподавателем, СУБД может быть выбрана и другая. Необходимо также область решения задачи, совместно с преподавателем сформулировать решаемые бизнес-задачи и выбрать технологию для самостоятельного изучения. В задании должны быть перечислены бизнес-задачи, для решения которых создается база данных. После согласования задания должен быть составлен подписанный студентом и преподавателем лист задания (в двух экземплярах) для подписи и утверждения у заведующего кафедрой.

Для консультаций по выполнению проекта установлено специальное время консультаций. В процессе консультаций у преподавателя могут возникнуть замечания, которые следует устранить до защиты работы. Во время проведения текущих аттестаций отмечается объем выполненного задания.

Законченный курсовой проект должен быть представлен для защиты на кафедру за месяц до начала сессии. При защите курсового проекта предоставляются пояснительная записка к курсовому проекту и электронный носитель CD/DVD, содержащий пояснительную записку, SQL-скрипты, файлы базы данных и приложения. За месяц до сессии на кафедре вывешивается список дат защиты курсовых проектов. Для защиты курсового проекта необходимо записаться на удобную дату.

Курсовые проекты, претендующие на высокую оценку, направляется на открытую защиту – на защите таких курсовых проектов присутствуют все желающие: студенты и преподаватели.

## 1.2 Минимальные требования к курсовому проекту

Для получения положительной оценки за курсовую работу необходимо выполнить следующее:

* выбрать область решения задачи и технологию,
* установить и настроить сервер СУБД,
* спроектировать инфраструктуру базы данных,
* разработать порядок авторизации в базе данных,
* создать объекты базы данных,
* импортировать и/или сгенерировать тестовые данные для демонстрации,
* самостоятельно изучить и применить определенную технологию в созданной базе данных,
* произвести резервное копирование и восстановление данных,
* подготовить пояснительную записку к курсовому проекту.

Требования к технологии:

* самостоятельно изучить и применить выбранную технологию в созданной базе данных,
* описать в пояснительной записке особенности применения технологии.

Требования по оптимизации:

* исследовать и при необходимости оптимизировать структуру запросов к БД,
* создать необходимые объекты для ускорения поиска в базе данных.

## 1.3 Дополнительные требования к курсовому проекту

Для повышения оценки за курсовой проект необходимо учитывать следующее:

* выбрать сложную бизнес-задачу (по согласованию с преподавателем),
* создать необходимые и достаточные объекты базы данных,
* создать план резервного копирования и восстановления данных,
* разработать полноценное приложение для демонстрации работы,
* подготовить пояснительную записку к курсовому проекту на высоком уровне.

Требования к технологии:

* самостоятельно изучить и применить выбранную технологию в созданной базе данных с использованием в приложении,
* описать в пояснительной записке особенности применения технологии в базе данных и в приложении.

# 2 ОСНОВНЫЕ РАЗДЕЛЫ ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ

Общий объем пояснительной записки курсового проекта должен составлять примерно 50-60 страниц текста, включая приложения.

## 2.1 Структура пояснительной записки

Пояснительная записка состоит из:

* титульного листа;
* задания на курсовой проект;
* содержания;
* введения;
* основной части пояснительной записки;
* заключения;
* списка использованных источников;
* графического материала;
* приложений.

## 2.2 Титульный лист

Титульный лист является первой страницей пояснительной записки. Номер страницы на титульном листе не указывается. Пример оформления приведен в приложении А.

## 2.3 Задание на курсовую работу

Задание на курсовую работу формулируется преподавателем при участии студента и включает:

* тему работы;
* исходные данные работы;
* срок сдачи работы;
* содержание пояснительной записки;
* перечень графического материала;
* календарный план выполнения работы.

Пример задания на курсовое проектирование приведен в приложении Б.

## 2.4 Содержание пояснительной записки

В содержании указываются все разделы и подразделы пояснительной записки и номера их начальных страниц.

## 2.5 Введение

Введение – это небольшой обзор курсового проекта. Во введении следует указать цель выполнения курсового проекта, сформулировать задачи для достижения цели, кратко описать содержание пояснительной записки со ссылкой на её разделы.

## 2.6 Основная часть пояснительной записки

Основная часть пояснительной записки должна включать следующие разделы и подразделы.

## 2.6.1 Постановка задачи

В настоящем разделе необходимо обосновать выбор области решения бизнес-задачи, СУБД и технологии, в ней применяющейся.

Примерные тематики областей баз данных приведены в разделе 4.

При формулировании задания должно быть подробно указано, какие именно бизнес-задачи сможет решить использование базы данных. Например, в случае автоматизации магазина необходимо указать в задании:

* процедуры поставки товара на склад магазина;
* процедуры возврата товара;
* процедуры оплаты товара, в том числе со скидкой;
* процедура получения наличия остатков товара на складе;
* анализ свойств покупателей: частота покупок, наличие дорогостоящих покупок и т.д.;
* управление работой продавцов (смены, ответственность за определенные участки и др.)

Список решаемых бизнес-задач должен быть согласован с преподавателем. Чем больше бизнес-задач могут быть решены при помощи вашей базы данных, тем лучше, и, соответственно, выше оценка. Если какая-то из указанных бизнес-задач не была решена, работа может быть признана невыполненной. Если какая-то из указанных была решена не полностью, работа может быть признана выполненной не полностью, и пропорционально снижена оценка.

Предполагается, что будет использоваться СУБД Microsoft SQL Server версии от 2012 или Oracle 12c. Однако, по согласованию с преподавателем, можно использовать и другую реляционную СУБД, а в случае сравнения применения реляционного и не реляционного решения будут использованы оба типа СУБД. При выборе СУБД следует руководствоваться в том числе и областью решения задачи, например, если ваша задача предполагает сравнение пространственных областей, то в выбранной СУБД должен быть удобный и достоверный механизм работы с пространственными данными.

При выборе технологии необходимо учитывать, целесообразно ли применение данной технологии при решении именно такой задачи. Например, задачи, предполагающие создание общей базы данных для распределенного предприятия, может быть использована технология репликации. Примерный список технологий СУБД приведен в разделе 4.

В пояснительной записке следует пояснить выбор СУБД и применяемой в ней технологии для решения выбранной бизнес-задачи. Необходимо также описать существующие аналогичные решения. Достаточно трех решений при их наличии. Ссылки на аналогичные решения необходимо указать в списке использованных источников.

## 2.6.2 Установка и настройка сервера СУБД

Необходимо установить и настроить сервер выбранной СУБД и среду выполнения приложения на университетском сервере (за деталями обратитесь к преподавателю), в облачной инфраструктуре или на личном ноутбуке (по выбору). В пояснительной записке требуется указать, какой сервер (сетевой, локальный, облачный) используется.

В случае, если выбранная СУБД не СУБД Microsoft SQL Server версии от 2012 или Oracle 12c, то, по согласованию с преподавателем, необходимо указать краткие этапы установки и настройки сервера СУБД.

## 2.6.3 Проектирование инфраструктуры базы данных и порядок авторизации пользователей к базе данных

Физическая реализация базы данных может существенно повлиять на способность базы данных удовлетворять требованиям к хранению и производительности. Проектирование физической реализации базы данных включает планирование файловых групп, использование секционирования для управления большими таблицами и возможное использование сжатия. В пояснительной записке требуется указать размещение базы данных, ее файлов и файловых групп, а также определить порядок аутентификации и авторизации пользователей, в частности для SQL Server:

* аутентификацию подключений к СУБД;
* авторизацию логинов на доступ к базам данных.

Создать необходимые роли на уровне сервера и базы данных. Авторизовать пользователей на доступ к ресурсам, в том числе:

* авторизовать пользователей на доступ к объектам;
* авторизовать пользователей на выполнение кода;
* настроить права на уровне схем.

Кроме того, в пояснительной записке требуется указать логины, роли и пользователей сервера и базы данных, с указанием необходимых и достаточных привилегий.

## 2.6.4 Проектирование и реализация объектов базы данных

Для решения бизнес-задач, сформулированных в техническом задании, необходимо создать необходимые объекты базы данных.

В простейшем случае это:

* таблицы,
* представления,
* индексы,
* хранимые процедуры и функции, определенные пользователем,
* триггеры.

Схема базы (логическая? Физическая?) данных должна быть приведена в разделе графического материала.

## 2. 6.4.1 Проектирование и реализация таблиц

В системе управления реляционными базами данных пользовательские и системные данные хранятся в таблицах. Каждая таблица состоит из набора строк, которые описывают сущности, и набора столбцов, которые содержат атрибуты объекта. Проектирование таблиц — одна из наиболее важных задач, которые выполняет разработчик базы данных, поскольку неправильный дизайн таблиц приводит к невозможности эффективного запроса данных.

Необходимо выполнить проектирование таблиц с указанием правильных типов данных и ограничений целостности, в том числе первичных и внешних ключей.

В пояснительной записке должны быть перечислены все таблицы с указанием типов данных и ограничений целостности и кратко указано, данные о каких объектах находятся в этой таблице.

## 2. 6.4.2 Проектирование и реализация представлений

Представление — это поименованный SELECT запрос, который хранится и может использоваться в других запросах точно так же, как и таблица. Представления упрощают дизайн базы данных, формируя слой абстракции и скрывая сложность соединения таблиц. Представления также являются способом защиты данных, предоставляя пользователям разрешения использовать требуемые данные, при этом не позволяя манипулировать исходными объектами, содержащими весь набор данных.

Необходимо выполнить проектирование и реализацию представлений. В пояснительной записке должны быть перечислены все созданные представления и указано их назначение.

## 2. 6.4.3 Проектирование и реализация индексов

Индекс представляет собой набор страниц, связанных с таблицей. Индексы используются для улучшения производительности запросов или обеспечения уникальности.

Необходимо выполнить проектирование и реализацию индексов, выбирая наиболее подходящие для каждого конкретного случая.

В пояснительной записке должны быть перечислены все созданные индексы и кратко указан их тип и назначение.

## 2. 6.4.4 Проектирование и реализация хранимых процедур и функций

Процедуры и функции — это подпрограммы, которые используются для инкапсуляции часто выполняемой логики. Вместо того чтобы повторять логику подпрограммы во многих местах, код может вызывать процедуру или функцию. Это делает код более удобным для обслуживания и легче для отладки. Процедуры также являются способом организации удобного доступа к данным из приложения, предоставляя приложению вызывать процедуру для использования требуемых данных, при этом не позволяя манипулировать исходными объектами, содержащими весь набор данных.

Необходимо выполнить проектирование и реализацию хранимых процедур с параметрами:

* для реализации бизнес-логики задачи,
* для вставки данных,
* для обновления данных,
* для удаления данных,
* для выборки данных.

Выборка данных подразумевает, что может быть получен как один объект, так и их набор. В результате выполнения выборки при помощи процедуры должен быть получен набор данных, не требующий дополнительной обработки, например, каких-либо вычислений или сортировки.

В процедурах должна быть реализована обработка ошибок. При необходимости должна быть использована обработка бизнес-логики в явной транзакции и установлен необходимый уровень изоляции транзакций.

Также необходимо выполнить проектирование и реализацию функций, определенных пользователем, для реализации бизнес-логики задачи. По необходимости могут быть использованы как скалярные, так и табличные функции.

В пояснительной записке должны быть перечислены все процедуры и функции с указанием входных и выходных параметров (при их наличии) и возвращаемых значений с указанием типов данных и указано, что именно выполняет каждая хранимая процедура или функция, определенная пользователем.

## 2. 6.4.4 Проектирование и реализация триггеров

Триггеры языка манипуляции данными (DML) — это мощные инструменты, которые можно использовать для обеспечения доменной целостности, контроля сущности реляционных данных и бизнес-логики. Обеспечение целостности помогает создавать надежные приложения.

Необходимо выполнить проектирование и реализацию необходимых триггеров для поддержания доменной целостности, контроля сущности реляционных данных и реализации бизнес-логики базы данных. Рекомендуется создать триггеры аудита. При необходимости можно использовать DDL триггеры.

В пояснительной записке должны быть перечислены все триггеры и указано, что именно выполняет каждый триггер и по какому условию.

## 2.6.5 Изучение и применение технологии в созданной базе данных

Необходимо изучить и подробно описать избранную технологию для СУБД. В записке должно быть подробно отражено следующее:

* Формальное описание назначения технологии;
* Привести примеры приложений с использованием данной технологии;
* Каким образом используется эта технология в базе данных;
* Листинги процедур, использующих эту технологию, приводятся в приложении.

## 2.6.6 Импорт и генерация тестовых данных

Для демонстрации функционирования базы данных и оптимизации запросов к ней необходимо произвести импорт и экспорт данных. Данные могут импортироваться и/или дополнительно генерироваться, причем импорт и экспорт может производиться для данных следующих форматов:

* табличных данных;
* данных в формате XML или JSON;
* данных в текстовом формате;
* с использованием команд BCP и BULK INSERT;
* путем развертывания приложений уровня данных (Data-Tier Applications).

Рекомендуется создать процедуры импорта и экспорта данных и привести их в пояснительной записке. Листинги процедур привести в приложении.

Также данные могут быть сгенерированы любым удобным для вас способом. Для проведения оптимизации запросов в базе данных необходимо, чтобы количество строк в таблицах, к которым выполняются запросы, было не менее 100 000 строк. Для оптимизации запросов в базе данных необходимо провести исследование SELECT -запросов, и создать необходимые индексы и/или переформулировать запросы. В пояснительной записке необходимо описать планы выполнения запросов до оптимизации, предложенные меры и состояние после оптимизации.

## 2.6.7 Резервное копирование и восстановление данных

Необходимо описать схему резервного копирования и восстановления данных с учетом решаемой бизнес-задачи.

В самом простом случае необходимо выполнить полное резервное копирование данных и восстановление из резервной копии. В пояснительной записке необходимо описать скрипты создания резервной копии и восстановления из нее. При использовании специализированных утилит — указать порядок использования утилиты.

Для получения более высоких оценок необходимо создать план резервного копирования и восстановления данных. Следует учитывать, различные аспекты потери данных: сбой хранения, сбой при передаче, умышленные или неумышленные действия третьих лиц. Также необходимо учитывать режим работы с данными: требуется ли режим работы 24/7, сколько времени может составлять допустимый простой и на какой момент времени должны быть восстановлены данные. Все эти аспекты должны быть описаны в пояснительной записке.

Во время защиты может потребоваться продемонстрировать потерю данных и их восстановление.

## 2.6.8 Разработка демонстрационного приложения

Для демонстрации работы необходимо представить результаты курсового проекта в приложении.

Основная цель разработки демонстрационного приложения — показать все стороны решения задачи и выбранную технологию. Среду разработки, язык, структуру и внешний вид приложения студент определяет самостоятельно.

Доступ к данным базы данных осуществляется только при помощи вызова соответствующих процедур и/или функций.

В пояснительной записке приводится краткое описание приложения. Структура, UML диаграмма, основные классы могут быть вынесены в приложение.

## 2.7 Заключение

В заключении формулируются краткие выводы по результатам выполненной работы. Приводятся количественные и качественные характеристики реализации базы данных: решенные бизнес-задачи, результаты применения избранной технологии в базе данных и т.п.

## 2.8 Графический материал

Графический материал выполняется на листе формата А4 по ГОСТ 2.301 и представляет собой логическую схему базы данных.

## 2.9 Список использованных источников

Необходимо перечислить книги, статьи, электронные ресурсы, которые были использованы при выполнении работы. Информация о правилах оформления этого списка приведена в пункте 3.12. Список использованных источников должен содержать не менее пяти наименований.

## 2.10 Приложения

Приложения содержат объемные материалы или материалы справочного характер, такие как схема базы данных, блок-схемы алгоритмов решения бизнес-задач, тексты SQL-скриптов, UML -диаграмму приложения, краткое описания классов приложения и т. д.

# 3 ОФОРМЛЕНИЕ ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ

## 3.1 Общие требования

При оформлении пояснительной записки следует руководствоваться требованиями СТП-001-2010 БГТУ. Дипломные проекты (работы).

Пояснительная записка к курсовому проекту оформляется в соответствии с правилами оформления текстовых документов, изложенными в ГОСТ 2.105–95 Общие требования к текстовым документам, и с правилами оформления курсовых проектов, изложенными в данном методическом указании. Текст записки должен быть напечатан на одной стороне листа формата А4. Используемый шрифт – Times New Roman, размер шрифта – 14, межстрочный интервал – одинарный. Цвет шрифта – черный. Абзацный отступ – 1.25 см. В тексте после знаков препинания обязательно ставится пробел. Нельзя сокращать слова (кроме сокращений, установленных правилами орфографии). Графический материал оформляется на листах формата А3 с рамкой.

Текст следует печатать, соблюдая поля: правое – 10±1 мм; верхнее – 20±1 мм; левое – 23±1 мм; нижнее – 15±1 мм. При наличии на листе рамки и основной надписи по форме 2 расстояние между верхней границей основной надписи с последней строкой текста, если лист полностью заполняется текстом, должно составлять 10–15 мм.

## 3.2 Структурные элементы записки

Структурные элементы записки: «Содержание», «Введение», «Заключение», «Список использованных источников», «Графический материал», а также каждый из основных разделов и каждое из приложений следует начинать с нового листа.

Заголовки элементов текста «Содержание», «Введение», «Заключение», «Список использованных источников», «Графический материал» следует записывать в начале соответствующих страниц строчными буквами, первая буква – прописная; шрифт – полужирный. Расположение – симметрично тексту и отделяются от него интервалом в 18 пт.

Текст основной части делят на разделы, разделы – на подразделы, подразделы – на пункты и т.д. Разделы нумеруются арабскими цифрами, точка в конце не ставится; подразделы нумеруются в пределах раздела. Номер раздела отделяется от номера подраздела точкой. В конце номера подраздела точка не ставится. За номером раздела или подраздела следует его название, записанное с прописной буквы без точки в конце. Переносы слов в заголовках не допускаются. Если заголовок состоит из двух предложений, их разделяют точкой. Номер и название раздела или подраздела записывается с абзацного отступа, шрифт – полужирный, отделяется от текста интервалом в 18 пт.

## 3.3 Нумерация страниц

Нумерация страниц пояснительной записки сквозная. На титульном листе, первой странице пояснительной записки, номер не указывается. Номер проставляется арабской цифрой без точки в правом верхнем углу страницы.

Графический материал, размещенный на листе формата А3, учитывается как одна страница.

## 3.4 Перечисления

В тексте пояснительной записки могут быть использованы перечисления. Пункты перечисления записывают после двоеточия в виде списка, каждый с абзацного отступа. Перед каждым пунктом ненумерованного списка перечисления следует ставить тире. При необходимости ссылки в тексте на один или несколько пунктов перечисления, перечисление оформляют в виде маркированного списка, каждый пункт начинают со строчной буквы русского алфавита (за исключением ё, з, о, г, ь, й, ы, ъ) с проставленной после нее круглой скобкой. Для дальнейшей детализации перечислений (сложные перечисления) необходимо использовать арабские цифры с проставленными после них круглыми скобками. Запись подчиненных пунктов сложного перечисления выполняют с абзацными отступами по отношению к основному.

Пример выполнения простого перечисления.

Перечислим объекты базы данных, используемые в настоящей курсовому проекту:

* таблицы,
* представления,
* индексы,
* хранимые процедуры и функции, определенные пользователем,
* триггеры.

## 3.5 Изложение текста

Текст пояснительной записки должен быть кратким, четким и не допускать различных толкований. В тексте пояснительной записки не допускается:

* применять обороты разговорной речи;
* применять для одного и того же понятия различные термины;
* применять сокращения слов, кроме установленных правилами орфографии русского языка.

Перечень допускаемых сокращений русских слов установлен в ГОСТ 2.316 и ГОСТ 7.12, белорусских – в СТБ 7.12.

Если в пояснительной записке принята особая система сокращения слов или наименований, то в ней должен быть приведен перечень принятых сокращений.

## 3.6 Формулы

Формулы располагаются в тексте в отдельных строках, по центру строки. Формулы нумеруются внутри раздела (формат: номер раздела, порядковый номер формулы через точку). Номера формул записываются на уровне формулы в круглых скобках справа в конце строки. Пояснения символов и числовых коэффициентов, входящих в формулу, должны быть приведены непосредственно под формулой. Пояснения должны начинаться со слова «где» и далее следует описание каждого символа в той последовательности, в которой они приведены в формуле по одному в строке.

Размер шрифта символов в формулах и уравнениях должен соответствовать размеру основного шрифта текста. Размер индексов при основных символах в формулах и уравнениях – 9 п.

Ссылки на формулы, ранее приведенные в тексте записки, а также на формулы в приложениях необходимо выполнять с использованием их номера, например: «…по формуле (2.8)…».

## 3.7 Примечания

Примечания следует применить в пояснительной записке, если необходимы пояснения по содержанию текста, таблиц или иллюстраций.

Примечания необходимо помещать непосредственно после текстового материала (рекомендуется в конце пункта, подпункта), таблицы или графического материала, к которым они относятся. Если примечание одно, то после слова «Примечание» следует ставить тире, а за ним с прописной буквы – его текст. Одно примечание не нумеруется. Номер примечания от его текста точкой не отделяют. Примечание к таблице необходимо помещать в конце таблицы над обозначающей ее окончание чертой.

Текст примечаний рекомендуется печатать шрифтом размером 12 п.

## 3.8 Рисунки

Все рисунки (чертежи, схемы, графики, структурные схемы и др. кроме таблиц) должны располагаться непосредственно после ссылки на них в тексте. Рисунок располагается так, чтобы его удобно было смотреть без поворота листа или с поворотом по часовой стрелке. Рисунки нумеруются арабскими цифрами сквозной нумерацией внутри раздела и располагаются с абзацного отступа. Рисунки каждого приложения состоят из обозначения приложения, точки и сквозной нумерации, выполненной арабскими цифрами. Вначале располагается сам рисунок, затем подрисуночный текст в виде: Рисунок, номер, наименование рисунка. Например: Рисунок 6.3 Результат резервного копирования.

Рисунок отделяют от текста интервалом 14 п.

Не допускается отрыв (перенос со страницы на страницу) рисунка и подрисуночного текста.

## 3.9 Таблицы

Таблицы нумеруются внутри раздела. Заголовок таблицы оформляют в виде: Таблица, номер, наименование таблицы. Заголовок таблицы выравнивают по левому краю таблицы. Если таблица выходит за размер листа, то ее делят на части. При переносе части таблицы на другой лист заголовок помещают только над первой частью, над остальными частями пишут слова «Продолжение таблицы» с указанием номера таблицы. Шапку таблицы при переносе части таблицы повторяют. Заголовки строк и столбцов пишут с прописной буквы.

Таблицу располагают в записке непосредственно после текста, в котором она упоминается.

Таблицу следует отделять от текста интервалом 12 п. Допускается выполнять таблицы, размещая их вдоль длинной стороны листа таким образом, чтобы таблица читалась при повороте листа на 90° по часовой стрелке.

Таблицы слева, справа и снизу ограничивают линиями.

## 3.10 Ссылки

Ссылки на разделы, подразделы, перечисления, таблицы, иллюстрации, формулы и приложения пояснительной записки следует выполнять по следующим примерам:

* «…структура базы данных, описанная в разделе 2…»;
* «…по пункту б) перечисления…»;
* «…символы приведены в таблице 1.1…»;
* «…изображено на рисунке 3.8…»;
* «…текст процедуры представлен представлены в приложении Д…».

Ссылку на литературный источник выполняют с указанием порядкового номера источника, под которым он внесен в «Список использованных источников» пояснительной записки. Номер ссылки проставляется арабскими цифрами в квадратных скобках, например, [1], [1,2,5].

Пример ссылки на источники:

* «…согласно п. 3.4 стандарта [7] …».

## 3.11 Приложения

Каждое приложение начинается с нового листа с указанием наверху посередине страницы слова «Приложение» с первой прописной буквы и его обозначения.

Приложения по ГОСТ 2.105 обозначаются заглавными буквами русского алфавита, начиная с А, за исключением букв Ё, З, Й, О, Ч, Ь, Ы, Ъ. После слова «Приложение» пишется буква, идентифицирующая его последовательность. Если в документе одно приложение, оно обозначается «Приложение А».

Приложения должны иметь заголовки, которые записывают симметрично тексту с прописной буквы в следующей строке.

Затем через одну пустую строчку следует текст приложения.

## 3.12 Список использованных источников

Источники – это книги, учебники, статьи из научных журналов и Интернета и т.д., использованные при выполнении курсового проекта. Источники в списке располагаются в порядке ссылок в тексте записки или по алфавиту, нумеруются арабскими цифрами без точки и печатаются с абзацного отступа, при этом дается библиографическое описание каждого источника в соответствии с ГОСТ 7.1, ГОСТ 7.12.

Общий шаблон описания книги, у которой не более трех авторов: ФИО автора, название книги, точка, тире, город, двоеточие, издательство, запятая, год издания, точка, тире, количество страниц, буква «с», точка.

Название города дается целиком, допустимы только сокращения «М.» (Москва) и «СПб.» (Санкт-Петербург); название издательства – без кавычек. Если у книги один, два или три автора, то вначале указывается фамилия, потом – инициалы.

Примеры оформления списка используемых источников приведены ниже.

Список использованных источников

1. Дейт К. Дж. Введение в системы баз данных. М.: Издательский дом Вильямс. — 2005. — 1328с.

2. Microsoft SQL Server 2012. Реализация и обслуживание. Учебный курс Microsoft / Пер. с англ. — М.: Русская редакция, СПб.: Питер, 2014. — 798 с.

3. Проектирование и реализация баз данных Microsoft SQL Server. Учебный курс Microsoft/пер. с англ. — 4-е изд. — М.: Русская редакция; СПб.: Питер, 2012. —512с.

4. Коннолли Т., Бегг К. Базы данных. Проектирование, реализация и сопровождение. Теория и практика. 3-е издание.: Пер. с англ. — М.: Вильямс, 2003. — 1440 с.

# 4 Примерная тематика курсовых проектов

## 4.1 Общие замечания

Для написания курсового проекта вам потребуется выбрать СУБД, тематику задачи и технологию для самостоятельного изучения.

Предполагается, что для реализации данной курсового проекта вы выберете СУБД Microsoft SQL Server версии 2012 и выше или Oracle 12c. Возможен выбор других СУБД по согласованию с преподавателем.

Обязательно проконсультируйтесь с преподавателем по выбору СУБД, задачи и технологии. Для решения некоторых задач потребуются, возможно, специфические технологии.

## 4.2 Список областей применения баз данных

Разработать базу данных для:

* авиакомпании;
* железнодорожного вокзала;
* агентства недвижимости;
* финансово-консалтингового агентства;
* службы занятости;
* промышленного предприятия;
* страховой компании;
* структуры законодательной власти;
* сети санаторно-курортных учреждений;
* сети кинотеатров;
* сети супермаркетов;
* службы доставки;
* гостиничного комплекса;
* сети предприятий общественного питания;
* сети образовательных учреждений разного уровня (школы, колледжи, училища, вузы и т.д.);
* культурно-развлекательного учреждения;
* газетно-журнального издания;
* театра;
* системы тестирования;
* туристического агентства;
* сети сотовой связи;
* электронной площадки по предоставлению услуг;
* аукционной площадки;
* расписания занятий в ВУЗе;
* планирования совместный действий (поездок и т.д.);
* медицинского центра;
* физкультурно-оздоровительного центра;
* магазина автомобилей;
* магазина средств связи;
* магазина бытовой техники и т.д.

## 4.3 Примерный список рекомендуемых технологий для самостоятельного изучения

Если не указано иное, то предполагается, что технология может быть реализована для выбранной СУБД (MS SQL Server, Oracle, PostgreSQL), если это возможно. Предлагаются следующие технологии для самостоятельного изучения и применения в курсовому проекту:

* Секционирование данных в базе данных;
* Установка и применение кластера в СУБД;
* Использование SQL Server Profiler;
* Применение Reporting Services для визуализации данных в СУБД MS SQL Server;
* Применение Analysis Services в СУБД MS SQL Server;
* Применение Integration Services в СУБД MS SQL Server;
* Применение Master Data Services в СУБД MS SQL Server;
* Применение Data Quality Services в СУБД MS SQL Server;
* Миграция данных и объектов из одной СУБД в другую;
* Разработка системы резервного копирования и восстановления БД;
* Применение мультимедийных типов данных в базе данных;
* Применение пространственных типов данных в базе данных;
* Обработка графов в базе данных;
* Шифрование и маскирование в БД;
* Применение полнотекстового поиска в БД;
* Репликация данных между серверами СУБД;
* Настройка системы безопасности сервера СУБД;
* Применение In-Memory – технологий;
* Применение средств мониторинга состояния СУБД;
* Применение облачной версии СУБД;
* Сравнение SQL и NoSQL решений.

# 5 Демонстрационный Пример

В данном демонстрационном примере приводятся выдержки из пояснительной записки к курсовому проекту студентки 3 курса. Пример намеренно сокращен и предназначен только для справки. Количество объектов базы данных, ролей и пользователей указано не полностью. Некоторые разделы намеренно отсутствуют. Приложения с SQL-скриптами не приводятся. При выполнении курсового проекта вы должны руководствоваться своей постановкой задачи.

## 5.1 Постановка задачи

Для выполнения курсового проекта было спроектировано и реализовано приложение ControlDiet, предметной областью которой является система управления режимом питания. Составной частью приложения является база данных, разработанная в СУБД Microsoft SQL Server 2012. Данная РСУБД используется для работы с базами данных размером от персональных до крупных баз данных масштаба предприятия.

Одной из главных задач данной курсового проекта является освоение технологии SQL Server Reporting Services, применяемой для построения различных отчетов. Данная технология была выбрана для работы по причине того, что разрабатываемая база данных хранит статистические данные об изменении пользователей, их приемах питания, по которым удобнее всего создавать отчеты, чтобы просматривать процессы изменения, статистические данные, информацию о конкретных объектах и ее преобразование и обработку для более приятного и удобного восприятия.

На сегодняшний день существует большое количество программных средств, помогающих людям организовывать свой ежедневный рацион, следить за нормами потребления калорий, белков, жиров и углеводов, распределять физическую нагрузку и т.д.

На популярных мобильных платформах есть два приложения «Здоровье» у Apple и «Google Fit» у Google, которые могут собирать информацию из сторонних приложений, а также позволяют им обмениваться этой информацией между собой.

MyFitnessPal — одно из самых многофункциональных приложений для поддержания здорового образа жизни. В нем есть простой и удобный счетчик калорий с большой базой данных о продуктах, содержащей более 4 млн. наименований. Приложение в состоянии отслеживать пройденное расстояние, а также в него можно загружать программы тренировки.

## 5.2 Логины и пользователи

Для базы данных ControlDiet было разработано два типа пользователей. Один из них обычного пользователя приложения, а второй для администратора приложения. Пользователь может просматривать и редактировать только свои данные, администратор может просматривать и редактировать данные всех пользователей, имеет доступ к представлениям, содержащим объединенную информацию по пользователям.

Для авторизации пользователей на сервере СУБД и для подключения к базе данных были разработаны логины.

Далее в рамках базы данных ControlDiet были созданы пользователи, использующие данные логины. Для каждого из пользователей создается роль, включающая необходимые и достаточные разрешения.

SQL-скрипты создания логинов, ролей, пользователей и назначения разрешений приведены в приложении Б.

## 5.3 Проектирование базы данных

Для хранения и предоставления данных пользователю, была разработана база данных, которая содержит в себе восемь таблиц, 7 из которых связаны друг с другом внешними ключами, а одна таблица типа audit. Схема базы данных представлена в приложении А.

Таблица Products содержит информацию о продуктах питания: ID, название, категорию, калорийность, БЖУ.Таблица Dishes хранит информацию о готовых блюдах, а именно: его имя и соответствующий ID.

Таблица Users содержит информацию о зарегистрированных пользователях приложения по управлению режимом питания: их возраст, вес, рост, цель, физическую нагрузку, а также норму калорий, высчитываемую из приведенных выше данных. History – это таблица для хранения информации об изменении веса и параметров отдельно взятых пользователях. Там хранится имя пользователя и дата замеров, таких как вес объем грудной клетки, талии и бедер.

Таблица Meals и таблица Food представляют собой хранение приемов пищи пользователей. Это осуществляется следующим образом: в таблицу Meals записывается прием питания конкретного пользователя и время данного приема пищи. А в таблицу Food записывается информация из чего состоял данных прием пищи. Так как пользователь может указать продукт или блюдо, в качестве пищи, данная таблица осуществляет хранения продуктов питания, если было указано блюдо, то оно раскладывается на соответствующие продукты питания и в ID\_Dish – указывается ID разложенного блюда, если это был обычных продукт питания, то в ID\_Dish записывается «нулевое блюдо», ID, которого равно нулю. В данном случае, нормализация базы данных намеренно нарушается для удобства обращения к данным.

Таблица Audit. хранит данные пользователей перед их изменением, для возможности отследить или восстановить какие-либо изменения, связанные с пользователями.

SQL-скрипты создания таблиц приведены в приложении Б.

## 5.4 Процедуры работы с данными

Для реализации доступа к данным были разработаны следующие хранимые процедуры и функции:

|  |  |  |
| --- | --- | --- |
| № п/п | Наименование процедуры или функции | Назначение |
| 7 | UpdateUser | Выполняется транзакция вставки данных в таблицу Audit для хранения информации об изменении данных и обновление данных пользователя |
|  |  |  |

*Таблица 5.1*

SQL-скрипты создания процедур и функций приведены в приложении В.

## 5.5 Процедуры извлечения и записи данных в XML

Для реализации импорта и экспорта данных из формата XML были разработаны следующие хранимые процедуры:

|  |  |  |
| --- | --- | --- |
| № п/п | Наименование процедуры или функции | Назначение |
| 1 | insertProductFromXML | Выполняется добавление новых данных в таблицу Product из xml-файла. Файл создан заранее и путь к нему передается в качестве параметра данной процедуры |
|  |  |  |

*Таблица 5.2*

SQL-скрипты создания процедур и функций приведены в приложении В.

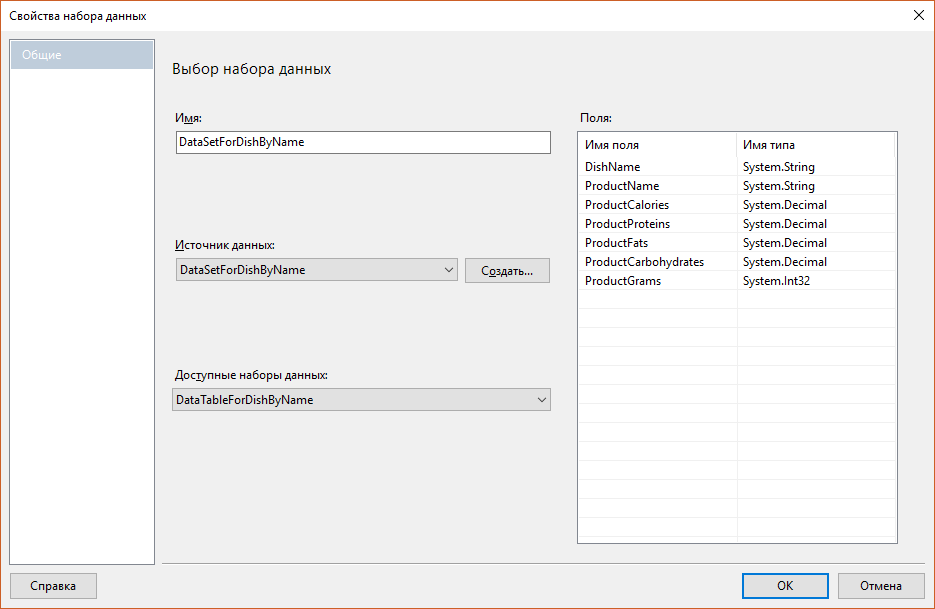
## 5.6 Технология SQL Server Reporting Services (SSRS)

SQL Server Reporting Services (SSRS) – это набор служб для разработки, построения, доставки и просмотра отчетов. С помощью этих служб можно создавать табличные, интерактивные, графические и другие более сложные отчеты с использованием диаграмм и других отчетных элементов.

Для обзора технологии приведено описание создания внедренного отчета SSRS, содержащего информацию о выбранном блюде, о его составе и соотношении белков, жиров и углеводов.

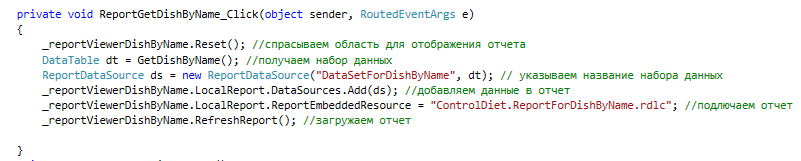
Для использования Reporting Services непосредственно из приложения необходимо в ссылки проекта подключить «Microsoft.Report Viewer.WinForms», после чего в определении окна MainWindow добавить ссылки на пространства имен xmlns:rv="clr-namespace:Microsoft.Reporting.WinForms; assembly= Microsoft.ReportViewer.WinForms". После чего в XAML создать объект WindowsFormsHost, в котором будет отображаться отчет.

Далее необходимо получить набор данных, содержащий сведения о калорийности продукта и содержании БЖУ. После этого в проект добавляем новый отчет, а в него источник данных, как изображено на рисунке 5.1:



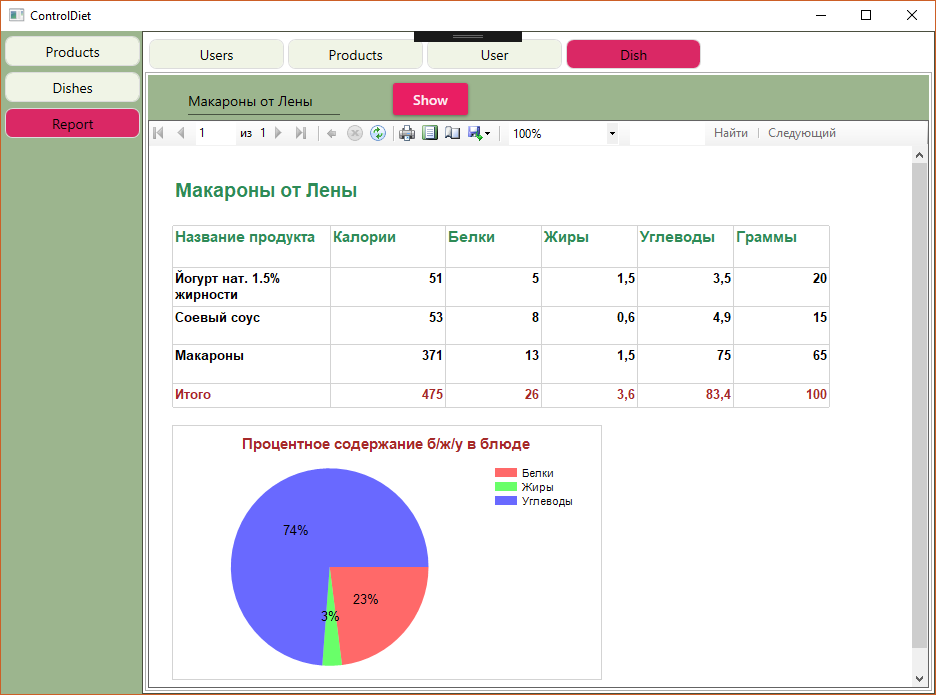
*Рисунок 5.1 – Выбор набора данных*

Далее отчет форматируется, добавляются стили, итоги, диаграммы. Последним этапом является добавление данного отчета в WPF приложение. Для этого используется функция, изображенная на рисунке 5.2:



*Рисунок 5.2 – Функция загрузки отчета*

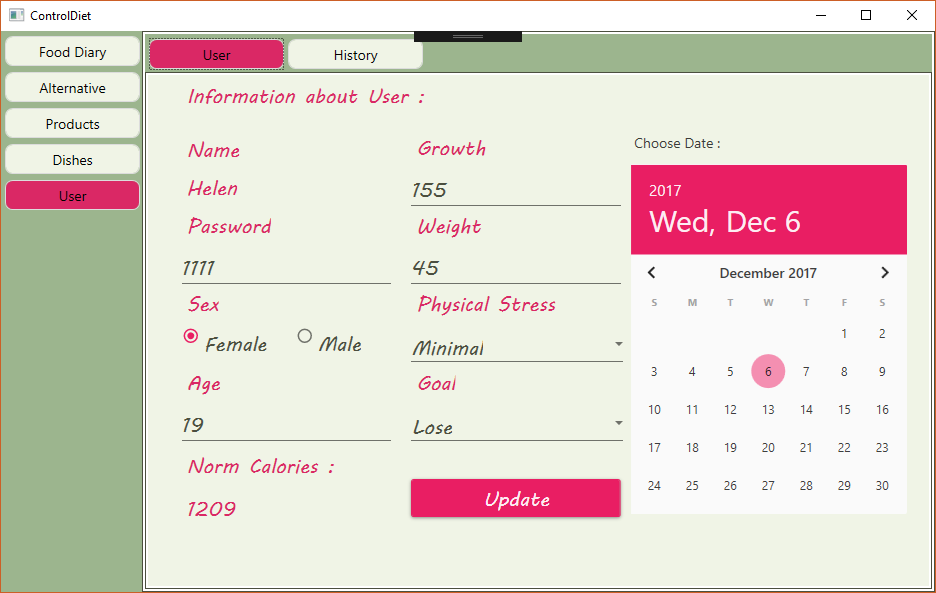
Итогом работы является отчет, представленный на рисунке 5.3:



*Рисунок 5.3 – Полученный отчет*

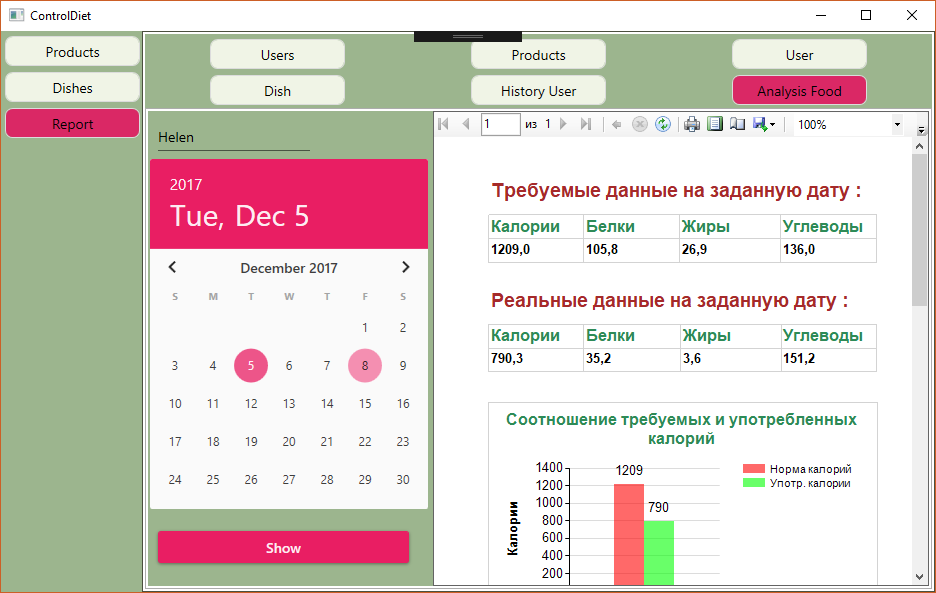
## 5.7 Демонстрационное приложение

На вкладке «User» пользователь может просмотреть информацию о себе, а также изменить ее.



*Рисунок 5.4 – Параметры пользователя*

По норме калорий высчитываются нормальные количества грамм белков, жиров и углеводов на тот период, а также подводится итог на заданный день, а именно суммируется вся калорийность и все б/ж/у. Далее производится построение диаграмм для наглядного отображения в разнице требуемых калорий и б/ж/у к употребленным. Пример отчета представлен на рисунке 5.5.

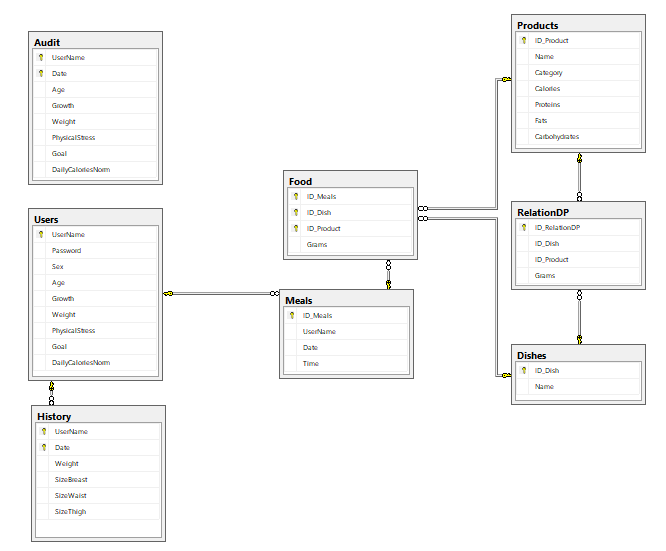


*Рисунок 5.5 – Анализ питания*

## 5.8 Приложения

Приложение А

Схема базы данных



Приложение Б

Логины, роли и пользователи

--connection login to user

--задаем имя пользователя и пароль

CREATE LOGIN [usercd] WITH PASSWORD=N'1111',

--задаем базу данных и язык

DEFAULT\_DATABASE=[ControlDiet], DEFAULT\_LANGUAGE=[Русский],

--указываем политики для истечения время срока пароля и действия пароля

CHECK\_EXPIRATION=OFF, CHECK\_POLICY=ON

GO

ALTER LOGIN [usercd] ENABLE

GO

--connection login to admin

CREATE LOGIN [admincd] WITH PASSWORD=N'admincd',

DEFAULT\_DATABASE=[ControlDiet], DEFAULT\_LANGUAGE=[Русский],

CHECK\_EXPIRATION=OFF, CHECK\_POLICY=ON

GO

ALTER LOGIN [admincd] ENABLE

GO

--создаем пользователей в рамках ControlDiet

CREATE USER [user] FOR LOGIN [usercd];

GO

CREATE USER [admin] FOR LOGIN [admincd];

GO

use ControlDiet;

--создаем роль для пользователей

CREATE ROLE [ControlDiet\_user];

--грант на выполнение [SelectUser]

GRANT EXECUTE ON [dbo].[SelectUser] TO [ControlDiet\_user];

--грант на выполнение [AddUser]

GRANT EXECUTE ON [dbo].[AddUser] TO [ControlDiet\_user];

--грант на выполнение [AddMeals]

GRANT EXECUTE ON [dbo].[AddMeals] TO [ControlDiet\_user];

--грант на выполнение [AddFood]

GRANT EXECUTE ON [dbo].[AddFood] TO [ControlDiet\_user];

--грант на выполнение [UpdateUser]

GRANT EXECUTE ON [dbo].[UpdateUser] TO [ControlDiet\_user];

--грант на выполнение [AddHistory]

GRANT EXECUTE ON [dbo].[AddHistory] TO [ControlDiet\_user];

--присваем данную роль пользователю

sp\_addrolemember 'ControlDiet\_user', 'user';

Приложение В

Хранимые процедуры и функции, определенные пользователем

--Update User

create procedure UpdateUser -- Создание процедуры

@UserName varchar(20), -- Имя пользователя (параметр)

@Password int, -- Пароль (параметр)

@Sex char(1), -- Пол (параметр)

@Age int, -- Возраст (параметр)

@Growth int, -- Рост (параметр)

@Weight int, -- Вес (параметр)

@PhysicalStress real, -- Физическая нагрузка (параметр)

@Goal real, -- Цель (параметр)

@DailyCaloriesNorm int, -- Норма калорий (параметр)

@Date Date -- Дата изменения (параметр)

as begin -- Начало процедуры

begin tran -- Начало транзации

-- Вставка данных в таблицу Audit для хранения истории данных пользователя

insert into Audit( UserName, [Date], Age, Growth, [Weight], PhysicalStress, Goal, DailyCaloriesNorm)

values ( @UserName, @Date, @Age, @Growth, @Weight, @PhysicalStress, @Goal, @DailyCaloriesNorm);

-- Обновления данных пользователя по переданным параметрам

update Users set UserName=@UserName, [Password]=@Password, Sex=@Sex, Age=@Age, Growth=@Growth,

[Weight]=@Weight, PhysicalStress=@PhysicalStress,

Goal=@Goal, DailyCaloriesNorm=@DailyCaloriesNorm

where UserName = @UserName;

commit -- Сохранения транзации

end; -- Конец процедуры

-- PROC insertProductFromXML

CREATE PROC insertProductFromXML

@path nvarchar(256) --передаем путь

AS

begin

SET NOCOUNT ON --отклюаем вывод кол-ва обработанных строк

SET XACT\_ABORT ON --откат транзакции в случае ошибки

BEGIN TRAN

declare @results table (x xml) --объявляем таблицу с полем xml

--считывание данные с удаленного источника (xml-файла)

--BULK - загрузка большого набора данных,

--SINGLE\_BLOB - возвращает содержимое в виде 1строки 1столбца бинарных данных

declare @sql nvarchar(300)=

'SELECT

CAST(REPLACE(CAST(x AS VARCHAR(MAX)), ''encoding="utf-16"'', ''encoding="utf-8"'') AS XML)

FROM OPENROWSET(BULK '''+@path+''', SINGLE\_BLOB) AS T(x)';

INSERT INTO @results EXEC (@sql)--выполняем и вставлем во временную таблицу

declare @xml XML = (SELECT TOP 1 x from @results); -- записываем xml в xml-переменную

INSERT INTO Products( Name, Сategory, Calories, Proteins, Fats, Carbohydrates)

SELECT

C3.value('Name[1]', 'varchar(30)') AS Name, --создаем именование

C3.value('Сategory[1]', 'varchar(30)') AS Сategory,

C3.value('Calories[1]', 'float') AS Calories,

C3.value('Proteins[1]', 'float') AS Proteins,

C3.value('Fats[1]', 'float') AS Fats,

C3.value('Carbohydrates[1]', 'float') AS Carbohydrates

FROM @xml.nodes('products/product') AS T3(C3) --форирование результирующих строк из метода nodes

COMMIT;

end;

### Приложение А (обязательное)

Образец оформления титульного листа

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет Информационных Технологий

Кафедра Информационных систем и технологий

Специальность 1-40 01 01 Программное обеспечение информационных технологий

Специализация Программирование интернет-приложений

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

**К КУРСОВОМУ ПРОЕКТУ НА ТЕМУ:**

Выполнил студент Фамилия Имя Отчество

(Ф.И.О. студента)

Руководитель проекта Фамилия Имя Отчество

(учен. степень, звание, должность, подпись, Ф.И.О. руководителя)

Заведующий кафедрой к.т.н., доц. Смелов В.В.

(учен. степень, звание, должность, подпись, Ф.И.О.)

Консультанты Фамилия Имя Отчество

(учен. степень, звание, должность, подпись, Ф.И.О.)

Курсовой проект защищен с оценкой

Минск 201\_

### Приложение Б (обязательное)

Образец оформления листа задания

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ

Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ УНИВЕРСИТЕТ»

Факультет информационных технологий   
Кафедра информационных систем и технологий

Утверждаю

Заведующий кафедрой

\_\_\_\_\_\_\_\_\_\_\_\_ В.В. Смелов

подпись инициалы и фамилия

«\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_201\_г.

ЗАДАНИЕ

к курсовому проекту по дисциплине "Базы данных"

Специальность 1-40 01 01 Программное обеспечение информационных технологий

Группа

Студент

Тема

1. Срок сдачи студентом законченной работы: «\_\_\_\_» декабря 201\_\_г.

2. Требования к работе:

2.1. Функционально должны быть выполнены следующие бизнес-задачи:

2.2. Прочие требования.

1. База данных должна быть спроектирована в СУБД \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.
2. Доступ к данным должен осуществляться только через соответствующие процедуры.
3. Должен быть проведен импорт данных из файлов формата \_\_\_\_\_\_, экспорт данных в формат \_\_\_\_\_\_.
4. Необходимо произвести протестировать производительность базы данных (на таблицах, содержащих не менее 100 000 строк) и внести изменения в структуру в случае необходимости.
5. Необходимо при решении бизнес-задачи применить технологию базы данных согласно выбранной теме.
6. Листинги скриптов должны содержать комментарии.

3. Содержание расчетно-пояснительной записки

Введение

Постановка задачи (выбор СУБД, область решения бизнес-задачи, используемая технология, аналоги)

Установка и настройка сервера СУБД

Проектирование инфраструктуры базы данных (пользователи, роли, файлы и пр.)

Проектирование и реализация необходимых объектов базы данных (таблицы, представления, индексы, хранимые процедуры и функции, триггеры и пр.)

Описание применения технологии в созданной базе данных

Импорт и генерация тестовых данных

Резервное копирование и восстановление данных

Демонстрационное приложение

Заключение

Список используемых источников

Приложения

4. Форма представления выполненного курсового проекта:

1. Теоретическая часть курсового проекта должна быть представлена в формате MS Word согласно \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_.
2. Необходимые схемы, диаграммы и рисунки допускается делать в MS Office Visio или PrintScreen.
3. Листинги скриптов представляются в приложении.
4. К записке необходимо приложить CD (DVD), который должен содержать: пояснительную записку, SQL-скрипты, файлы базы данных и приложение.

Календарный план

|  |  |  |
| --- | --- | --- |
| № п/п | Наименование  этапов курсового проекта | Срок выполнения этапов |
| 1 | Аналитический обзор литературы по теме работы |  |
| 2 | Проектирование инфраструктуры и модели базы данных. |  |
| 3 | Проектирование и реализация объектов базы данных |  |
| 4 | Применение технологии в базе данных |  |
| 5 | Импорт и экспорт данных, тестирование производительности |  |
| 6 | Резервное копирование и восстановление данных базы данных |  |
| 7 | Разработка демонстрационного приложения |  |
| 8 | Оформление пояснительной записки |  |

5. Дата выдачи задания «\_\_\_\_\_» сентября 201\_\_ г.

Руководитель

подпись инициалы и фамилия преподавателя

Задание принял к

подпись инициалы и фамилия студента

**Базы данных**

Методические указания к выполнению курсового проекта

Составители: Блинова Евгения Александровна, Мороз Леонарда Станиславовна

Редактор

Компьютерная верстка

Корректор

Издатель:

УО «Белорусский государственный технологический университет»

Свидетельство о государственной регистрации издателя,

Изготовителя, распространителя печатных изданий

№ 1/227 от 20.03.2014.

Ул. Свердлова, 13а, 220006, г. Минск.