

Подписано электронной подписью:

Вержицкий Данил Григорьевич

Должность: Директор КГПИ ФГБОУ ВО «КемГУ»

Дата и время: 2024-02-21 00:00:00

471086fad29a3b30e244c728abc3661ab35c9d50210dcf0e75e03a5b6fdf6436

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Кемеровский государственный университет»
Новокузнецкий институт (филиал)

Факультет информатики, математики и экономики

Кафедра математики, физики и математического моделирования

Ю.С. Гаврилова

СОВРЕМЕННЫЕ ТЕХНОЛОГИИ ПРОГРАММИРОВАНИЯ

SQL

Методические указания к выполнению курсовой работы по дисциплине «Современные технологии программирования SQL» для обучающихся по направлению подготовки

02.03.03 Математическое обеспечение и администрирование информационных систем, профиль «Программное и математическое обеспечение информационных технологий»

Новокузнецк

2020

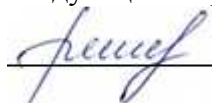
УДК [378.147.88:004.42](072)
ББК 74.484(2Рос-4Кем)я73+32.973-018я73
Г 12

Гаврилова Ю.С.

Г 12 Современные технологии программирования SQL: методические указания к выполнению курсовой работы по дисциплине для студентов факультета информатики, математики и экономики, обучающихся по направлениям подготовки 01.03.02 Прикладная математика и информатика (профиль «Математическое моделирование и информационные технологии»), 02.03.03 Математическое обеспечение и администрирование информационных систем (профиль «Программное и математическое обеспечение информационных технологий») / Ю.С. Гаврилова; Новокузнецкий ин-т (фил.) Кемеров. гос. ун-та. – Новокузнецк : НФИ КемГУ, 2020 – 44 с.

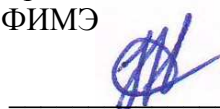
В работе изложены методические указания к выполнению курсовой работы по дисциплине «Современные технологии программирования SQL»: цель, задачи и содержание курсовой работы, структура курсовой работы, требования к ее подготовке и оформлению, процедура защиты, критерии оценивания курсовой работы в балльно-рейтинговой системе, список основной и дополнительной литературы, современных профессиональных баз данных и информационных справочных систем, приложения (титульный лист, задание на курсовую работу, образец введения курсовой работы, примерные темы курсовых работ).

Рекомендовано на заседании
кафедры математики, физики и
математического моделирования
Протокол № 8 от 16.03.2020
Заведующий каф. МФММ



/ Е.В. Решетникова

Утверждено методической комиссией
факультета информатики, математики и
экономики
Протокол № 8 от 18.05.2020
Председатель методической комиссии
ФИМЭ



/ Г.Н. Бойченко

УДК [378.147.88:004.42](072)
ББК 74.484(2Рос-4Кем)я73+32.973-018я73
Г 12

© Гаврилова Юлия Сергеевна
© Федеральное государственное бюджетное
образовательное учреждение высшего
образования «Кемеровский государственный
университет», Новокузнецкий институт
(филиал), 2020

Текст представлен в авторской редакции

СОДЕРЖАНИЕ

ПРЕДИСЛОВИЕ	4
1. ХАРАКТЕРИСТИКА СОДЕРЖАНИЯ КУРСОВОЙ РАБОТЫ	5
1.1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ	6
1.2. РАЗРАБОТКА ПРИЛОЖЕНИЯ БАЗЫ ДАННЫХ	12
1.3. ТЕСТИРОВАНИЕ ПРИЛОЖЕНИЯ БАЗЫ ДАННЫХ	15
2. СТРУКТУРА ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ КУРСОВОЙ РАБОТЫ...	16
3. ОРГАНИЗАЦИОННЫЕ ТРЕБОВАНИЯ К ПОДГОТОВКЕ КУРСОВОЙ РАБОТЫ.....	27
4. ТРЕБОВАНИЯ К ОФОРМЛЕНИЮ КУРСОВОЙ РАБОТЫ.....	30
5. ОФОРМЛЕНИЕ БИБЛИОГРАФИЧЕСКОГО СПИСКА И БИБЛИОГРАФИЧЕСКИХ ССЫЛОК.....	31
6. ЗАЩИТА КУРСОВОЙ РАБОТЫ.....	34
7. КРИТЕРИИ ОЦЕНИВАНИЯ КУРСОВОЙ РАБОТЫ В БАЛЛЬНО-РЕЙТИНГОВОЙ СИСТЕМЕ	35
8. РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА	39
СОВРЕМЕННЫЕ ПРОФЕССИОНАЛЬНЫЕ БАЗЫ ДАННЫХ И СПРАВОЧНЫЕ СИСТЕМЫ.....	40
ПРИЛОЖЕНИЕ 1. Титульный лист	42
ПРИЛОЖЕНИЕ 2. Задание на курсовую работу	43
ПРИЛОЖЕНИЕ 3. Образец введения курсовой работы	44
ПРИЛОЖЕНИЕ 4. Примерные темы курсовых работ по дисциплине «Современные технологии программирования SQL»	46

ПРЕДИСЛОВИЕ

Настоящие методические материалы адресованы студентам, получающим квалификацию бакалавр по направлениям подготовки: 01.03.02 Прикладная математика и информатика, 02.03.03 Математическое обеспечение и администрирование информационных систем и направлены на оказание помощи студентам в написании курсовой работы по дисциплине «Современные технологии программирования SQL».

Курсовая работа по дисциплине «Современные технологии программирования SQL» является важным компонентом профессиональной подготовки будущего специалиста в сфере информационных технологий. Согласно учебным планам основных образовательных программ направлений подготовки 01.03.02 «Прикладная математика и информатика», 02.03.03 «Математическое обеспечение и администрирование информационных систем» курсовая работа выполняется на третьем курсе в шестом семестре.

Целью выполнения курсовой работы является формирование способности студента к самостоятельному анализу предметной области, инфологическому моделированию и использованию современных средств разработки для создания программных продуктов, взаимодействующих с базами данных.

В процессе написания курсовой работы осуществляется закрепление, углубление и обобщение знаний, полученных студентами за время обучения, и применение этих знаний к решению конкретных задач.

1. ХАРАКТЕРИСТИКА СОДЕРЖАНИЯ КУРСОВОЙ РАБОТЫ

Курсовая работа представляет собой вид учебной работы обучающегося, в которой присутствуют элементы самостоятельного научного исследования.

Курсовая работа – это самостоятельный комплексный проект, при выполнении которого студент решает учебно-исследовательские задачи, носящие творческий характер. Поэтому материал для нее не копируется из интернета, а анализируется, систематизируется, приобретая в результате этого характер оригинальности.

При выполнении курсовой работы обучающийся должен научиться проектировать и разрабатывать базы данных, а затем на их основе составлять программы высокого качества, легко модифицируемые и простые в обращении с применением современных информационных технологий.

Курсовая работа носит прикладной характер, направлена на практическое применение изученного материала к решению профессиональных задач. В рамках дисциплины «Современные технологии программирования SQL» обучающиеся изучили современные информационные технологии: современные СУБД и технологии программирования приложений баз данных. Обучающийся может самостоятельно выбрать любую СУБД и технологию программирования из изученных в курсе. Также, по желанию обучающийся может использовать в работе информационные технологии и СУБД, изученные самостоятельно, при условии, что их применение оправдано (дает более эффективное решение задачи).

Во время выполнения курсовой работы обучающийся должен продемонстрировать умение анализировать предметную область, разрабатывать инфологическую и даталогическую модели данных, реализовывать модель в какой-либо современной СУБД, а затем разрабатывать приложение базы данных.

1.1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

Процесс проектирования базы данных представляет собой последовательность переходов от неформального словесного описания информационной структуры предметной области к формализованному описанию объектов предметной области в терминах некоторой модели. Начальным этапом процесса проектирования базы данных является анализ предметной области. На данном этапе необходимо провести словесное описание объектов предметной области и реальных связей, которые присутствуют между описываемыми объектами. Существует 2 подхода к выбору состава и структуры предметной области: функциональный, основанный на движении «от задач», и предметный, в случае, если круг задач заранее не определен.

Чаще всего разработчики комбинируют эти два подхода, ориентируясь на функциональные потребности пользователей и учитывая возможность наращивания новых приложений.

Анализ требований к базе данных и приложению базы данных

Требование – это некоторая функция, которая должна быть включена в создаваемую систему.

Анализ требований пользователей к базе данных предназначен для определения ее информационных потребностей. Обычно решение данной задачи происходит в условиях существования какой-либо информационной системы (на предприятии), поэтому важным является организация взаимодействия между существующими программными продуктами и разрабатываемыми.

При сборе требований уточняются следующие вопросы:

1. Сможет ли разрабатываемое приложение объединить существующие приложения или их будет необходимо переделать для совместной работы с новой системой?
2. Какие данные используются существующими приложениями?

Будет ли разрабатываемое приложение совместно использовать какие-либо из этих данных?

3. Кто будет вносить данные и в каком виде?
4. Как часто будут изменяться данные?
5. Достаточно ли будет для исследуемой предметной области одной базы или потребуется несколько баз данных с различными структурами?
6. Какая информация является наиболее чувствительной к скорости ее извлечения и изменения?

Ответы на данные вопросы помогают наиболее полно учесть все нюансы функциональности, требуемой для разрабатываемой системы, что в свою очередь снизит вероятность возникновения ошибок при проектировании.

Для сбора требований можно воспользоваться одним из следующих способов: интервьюирование, анкетирование, мозговой штурм, сценарии и ролевые игры, создание прототипов, совместная разработка приложений, моделирование, uses case, наблюдение за деятельностью предприятия, изучение документов (особенно тех, которые используются для сбора или представления информации).

При осуществлении сбора и анализа требований к базе данных и программному приложению в рамках выполнения курсовой работы по дисциплине «Современные технологии программирования SQL» необходимо:

1. выбрать метод сбора требований пользователей к разрабатываемому программному продукту (допускается комбинирование нескольких методов), обосновать выбор;
2. провести сбор требований с помощью выбранного метода;
3. провести анализ собранных требований, сделать выводы.

Построение инфологической модели данных

На этапе анализа предметной области выявляются основные сущности, их атрибуты, домены, связи между сущностями, которые затем нужно представить в виде инфологической модели данных.

Сущность – это реальный или представляемый объект предметной области, информация о котором должна сохраняться и быть доступна.

Атрибут – это поименованная характеристика сущности, определяющая его свойства и принимающая значения из некоторого множества.

Домен – это диапазон допустимых значений, которые может принимать атрибут.

Пример: рассмотрим предметную область «Зоопарк». В данной предметной области есть несколько зоопарков, обладающих следующими характеристиками: название зоопарка, адрес, номер телефона, Ф.И.О. директора. В зоопарке обитают животные, причем разные животные представлены в разных количествах. Каждое животное имеет свое название и среднюю продолжительность жизни, питается каким-то видом продуктов питания в определенных количествах.

Таким образом, можно выделить следующие сущности: зоопарк, животное, еда (рис. 1).

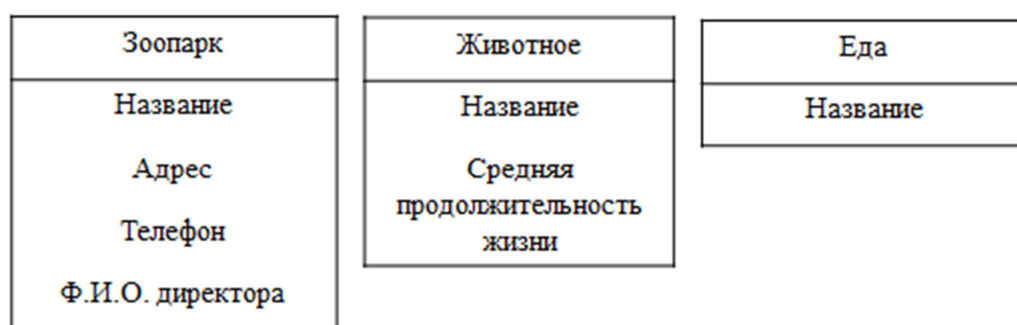


Рисунок 1. Сущности предметной области «Зоопарки»

Обратите внимание, что не все атрибуты, возможные в реальности, следует переносить в базу данных. Необходимость переноса атрибута зависит от того, планируется ли использовать этот атрибут в дальнейшем при решении поставленной задачи. Например, если бы перед нами стояла задача

по учету качества питания животных в зоопарках – необходимо было бы к сущности «Еда» добавить такие атрибуты, как «Калорийность», предусмотреть учет витаминов, микроэлементов и т.п.

Кроме того, необходимо отметить, что под животным подразумевается не конкретный медведь, а совокупность медведей, например, белый медведь, т.е. степень конкретизации предметной области тоже может быть различной и зависит от поставленной задачи. Если нам интересно, какие виды животных обитают в различных зоопарках – можно оставить сущность «Животное» в таком виде, однако если нам необходимо вести учет состояния здоровья конкретного животного, знать его возраст, вес и прочие параметры, то в таблицу «Животное» помещался бы список конкретных особей.

Связь – ассоциирование двух или более сущностей. Связи можно охарактеризовать кардинальностью (мощностью).

Выделяют 3 вида связей:

- «Один-к-одному» – один экземпляр одной сущности связан с единственным экземпляром другой сущности;
- «Один-ко-многим» – один экземпляр одной сущности связан с одним или более экземпляром другой сущности и каждый экземпляр второй сущности связан только с одним экземпляром первой сущности;
- «Многие-ко-многим» – один экземпляр одной сущности связан с одним или более экземпляром другой сущности и каждый экземпляр второй сущности связан с одним или более экземпляром первой сущности.

Инфологическая модель данных – это частично формализованное описание объектов предметной области.

Инфологическая модель строится в соответствии с какой-либо нотацией (графическим способом представления). Выделяют нотации Питера Чена, Crow's Foot (воронья лапка) и нотацию Баркера.

В соответствии с нотацией Crow's Foot (рис. 2) сущности обозначаются прямоугольником, состоящим из двух частей: в верхней расположено название сущности, а в нижней – список ее атрибутов. Название сущности и

название атрибутов, как правило, записываются в именительном падеже, единственном числе.

Связи между сущностями изображаются линиями. Необязательность связи обозначается пустым кружком на конце связи, а обязательность - перпендикулярной линией, перечеркивающей связь.

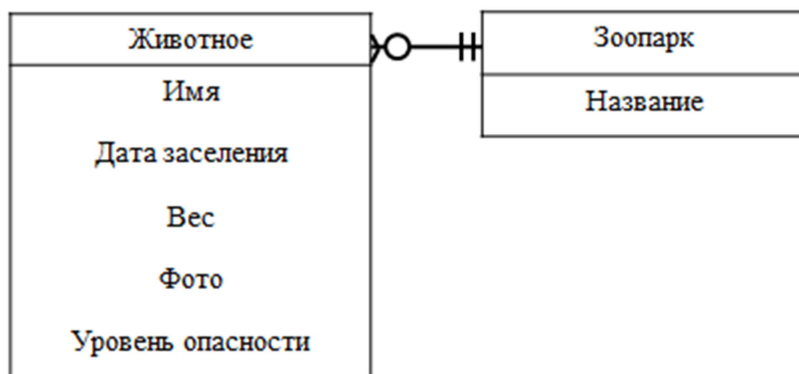


Рисунок 2. Нотация Crow's Foot

Кружок иллюстрирует возможность такой ситуации: зоопарк уже построен, но животных в него еще не заселили. Одинарный перпендикуляр означал бы, что, хотя бы одно животное точно живет в зоопарке (этот вариант подходит, если мы рассматриваем всех существующих в мире животных, часть из которых живет в зоопарке). Двойной перпендикуляр показывает, что каждое животное из этой базы данных должно жить в своем зоопарке.

При построении модели данных в рамках выполнения курсовой работы по дисциплине «Современные технологии программирования SQL» необходимо составить текстовое описание предметной области и инфологическую модель данных в соответствии с нотацией Crow's Foot.

Разработка проекта приложения баз данных

Разработка проекта приложения базы данных должна включать в себя разработку структурной схемы приложения и его модулей, построение диаграммы переходов состояний интерфейса, диаграмм приложения и объектов приложения.

Структурная схема – это совокупность элементарных звеньев объекта и связей между ними, один из видов графической модели.

Диаграмма переходов состояний является графической формой предоставления математической абстракции, используемой для моделирования определенного поведения технических объектов или объектов реального мира. Диаграмма перехода состояний содержит три типа элементов:

- возможные состояния системы (показаны в виде прямоугольников);
- разрешенные состоянием переходы (показаны в виде стрелок, соединяющих пары прямоугольников);
- события или условия, вызывающие каждый переход (показаны в виде текстовых пояснений для каждой стрелки перехода). Текст может пояснять и событие, и соответствующую реакцию системы.

Диаграмма классов используется для описания структуры классов, атрибутов, методов и зависимостей между ними. Атрибуты используются для того чтобы описать свойства для каждого объекта из данного класса, а поведение класса – это услуга (функция), которую класс может выполнить. Классы могут быть отображены по-разному, не обязательно отображать блок «Поведение класса», если класс не имеет собственных функций, а только наследует их от «предка».

Последним этапом в разработке проекта программного приложения является создание **диаграммы объектов** – снимка состояния системы в определенный момент времени, показывающего множество объектов, их состояний и отношений между ними в данный момент. В качестве объектов могут выступать пользователи, инициирующие взаимодействие, классы, обладающие поведением в системе, или программные компоненты.

1.2. РАЗРАБОТКА ПРИЛОЖЕНИЯ БАЗЫ ДАННЫХ

Этап разработки состоит из двух частей: разработку базы данных и самого программного приложения.

Разработка базы данных включает в себя:

1. реализацию даталогической модели в конкретной СУБД (создание таблиц);
2. заполнение полученных таблиц (в количестве, достаточном для демонстрации работы базы данных – примерно 150-200 строк в таблице, реализующей связь вида «многие-ко-многим»);
3. составление на языке SQL запросов к базе данных, необходимых для функционирования приложения;
4. создание объектов базы данных, необходимых для ее оптимальной работы (индексов, курсоров, представлений, хранимых процедур и триггеров).

Реализация даталогической модели базы данных в СУБД

Даталогическая модель базы данных (рис. 3) реализуется средствами конкретной СУБД, подходящей для разработки программного приложения. Даталогическую модель данных можно создать либо в конструкторе, либо путем создания ее описания на языке определения данных.

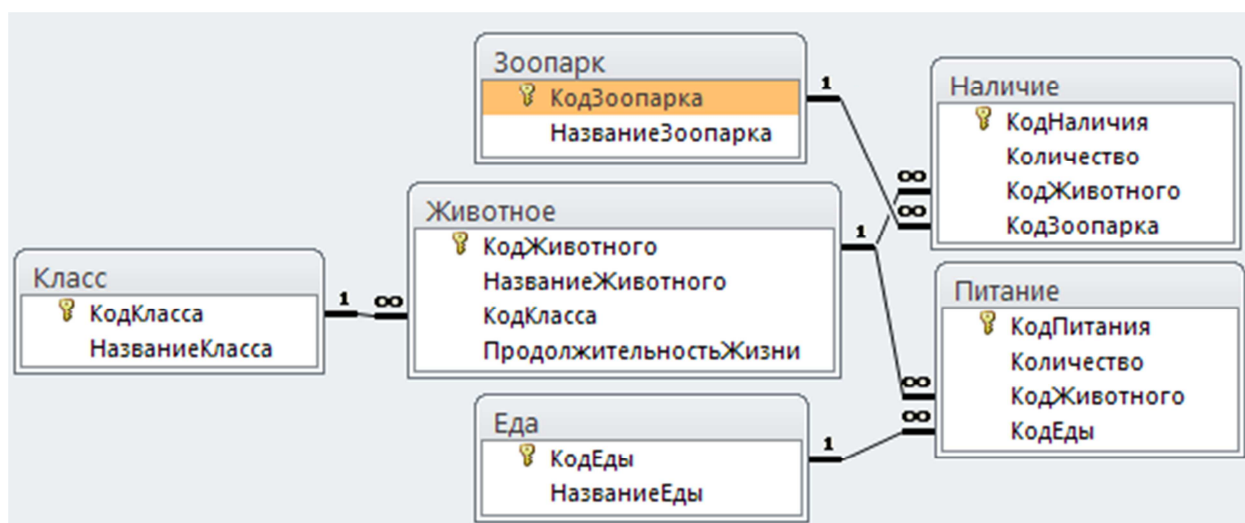


Рисунок 3. Даталогическая модель данных, реализованная в СУБД Microsoft Office Access

Не все СУБД предоставляют разработчику возможность графического представления даталогической модели, поэтому для курсовой работы достаточно составить ее описание на языке определения данных (представить текст SQL-запросов создания таблиц).

Заполнение таблиц базы данных

Заполнение базы данных можно осуществлять в двух режимах: непосредственное внесение в режиме «Таблица» (рис. 4) или с помощью запросов на языке SQL.

Текст запросов выглядит следующим образом:

```
INSERT INTO public."Animals"(  
    "IDAnimals", "NameAnimals", "LifeTimeAVG", "ClassID")  
VALUES (21, "Обезьяна", 50, 6);
```

	IDAnimals [PK] integer	NameAnimals text	LifeTimeAVG integer	ClassID integer
11	12	Рыба-пила	32	2
12	13	Тритон	32	3
13	14	Саламандра	34	3
14	15	Фламинго	27	5
15	16	Страус	59	5
16	17	Черепаша	9000	4
17	18	Лягушка	3	1
18	19	Крокодил-веган	10000	4
19	20	Рыба-клоун	10	2
20	11	Рыба-меч	34	2
21	21	Обезьяна	50	6

Рисунок 4. Заполнение базы данных в режиме «таблица»

Не требуется представлять результат заполнения таблиц в тексте курсовой работы, однако для проверки корректности функционирования

приложения, необходимо наличие в базе данных достаточного количества записей.

Разработка объектов базы данных на языке SQL

Назначение запросов к базе данных и их количество определяются обучающимся самостоятельно, в соответствии с требованиями к функциональным возможностям разрабатываемого программного приложения.

Необходимыми объектами базы данных являются представления, которые выглядят как обычный запрос на выборку, но помогают скрыть от пользователя схему базы данных (например, первичные и внешние ключи таблиц). Это обеспечивает легкость восприятия предоставляемой приложением информации и защищенность информации от случайных ошибок пользователя.

Для оптимальной работы базы данных и программного приложения может быть необходимым создание индексов, курсоров, хранимых процедур или триггеров.

Разработка модулей приложения базы данных

Модули приложения базы данных разрабатываются в соответствии с составленным проектом.

При реализации модулей программного приложения необходимо настроить права доступа, т.е. определить, кто может просматривать данные (например, любой пользователь, любой авторизованный пользователь, или пользователь, который относится к определенной категории), вносить изменения в базу данных и т.д.

Необходимо реализовать заполнение базы данных таким образом, чтобы возможность серьезных ошибок была сведена к минимуму. Например, вместо внешнего ключа таблицы должно подставляться поле из связанной таблицы, а механизм связи должен быть скрыт от пользователей. Если пользователь во время заполнения базы данных должен вносить информацию

определенного вида, например, адрес электронной почты или номер телефона, то оптимальным решением будет организовать предварительную проверку корректности внесенных данных.

1.3. ТЕСТИРОВАНИЕ ПРИЛОЖЕНИЯ БАЗЫ ДАННЫХ

Тестирование приложения базы данных направленно на проверку корректности функционирования самого приложения, а также на корректность заполнения базы данных через приложение.

Основные направления тестирования:

1. тестирование прав доступа (доступ строго разграничен, вносить изменения в базу данных незарегистрированный пользователь не может);
2. тестирование корректности добавления информации в базу данных (данные, которые были добавлены пользователем в программной оболочке, корректно добавились в базу);
3. тестирование корректности выводимых на форму (или в отчет) данных (корректность выполнения запросов на выборку);
4. если в системе предусмотрены триггеры, то необходимо протестировать их корректную работу (снимки экрана до и после наступления события, на которое срабатывает триггер).

2. СТРУКТУРА ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ КУРСОВОЙ РАБОТЫ

Структура пояснительной записки (текста) курсовой работы должна способствовать раскрытию избранной темы. Структурные элементы: титульный лист, оглавление, введение, основная часть, заключение, библиографический список (использованные источники) и, при необходимости, приложения.

Введение курсовой работы в обязательном порядке должно содержать постановку цели и задач исследования, обоснование его актуальности, теоретической и практической значимости. Пример составления и оформления введения для курсовой работы на тему «Разработка базы данных и программного приложения для стоматологической клиники» представлен в Приложении 3.

Актуальность исследования – это степень важности курсовой работы в данный момент времени в конкретной стадии развития выбранной отрасли знаний. Она часто связана с неизученностью (или недостаточной изученностью) темы или возможностью решения определенной задачи практики и охватывает запросы отдельной отрасли науки, региона, части общества.

В качестве **цели** исследования будет обозначен результат, который будет получен в ходе написания курсовой работы. Целью написания курсовой работы является решение конкретной проблемы, обозначенной в теме работы. Поэтому цель следует формулировать в виде достижения результата.

Все темы курсовой работы по дисциплине «Современные технологии программирования SQL» представляют собой задачу разработки базы данных и программного приложения, поэтому начать формулировку цели необходимо со слов: «разработать базу данных и программное приложение» или «разработать базу данных и программный продукт». Цель исследования

формулируется на основе темы курсовой работы, например, если тема курсовой работы «Разработка базы данных и программного приложения для стоматологической клиники», то цель данной работы можно сформулировать следующим образом: «разработать базу данных и программное приложение для стоматологической клиники».

Постановка **задач** исследования по выбранной теме курсовой работы осуществляется согласно поставленной цели. Задачи представляют собой ряд (как правило, четыре или пять) конкретных последовательных этапов достижения поставленной цели.

Курсовая работа должна содержать теоретическую и практическую часть, это должно отражаться и в задачах написания работы. Задачи курсовой работы взаимосвязаны друг с другом, поэтому они располагаются последовательно раскрытию теоретического и практического материала и соответствуют составленному содержанию работы.

Теоретическая и практическая значимость исследования формулируются на основании возможности последующего применения результатов, полученных в курсовой работе.

Основная часть пояснительной записки курсовой работы должна иметь трехчастную структуру, т.е. курсовая работа состоит из трех глав, а каждая глава, в свою очередь, из параграфов.

Рекомендуется придерживаться следующего порядка изложения текста курсовой работы:

1. Анализ предметной области.
 - 1.1 Требования к базе данных и приложению базы данных.
 - 1.2 Информационная модель данных.
 - 1.3 Проект приложения базы данных.
2. Разработка приложения базы данных.
 - 2.1 Информационная модель данных.
 - 2.2 Запросы к базе данных.
 - 2.3 Модули приложения базы данных.

3. Тестирование приложения базы данных.

В качестве примера приведем структуру курсовой работы по теме «Разработка базы данных и программного приложения для стоматологической клиники»:

СОДЕРЖАНИЕ	
Введение.....	3
1 Анализ предметной области	4
1.1. Сбор и анализ требований к базе данных и приложению базы данных	5
1.2. Построение инфологической модели данных.....	8
1.3. Разработка проекта приложения базы данных	11
2 Разработка приложения базы данных	16
2.1. Реализация даталогической модели данных в СУБД PostgreSQL.....	16
2.2. Разработка запросов к базе данных	18
2.3. Разработка модулей приложения базы данных	21
3 Тестирование приложения базы данных	25
Заключение.....	30
Список литературы.....	31

Рассмотрим подробнее вопросы, которые необходимо исследовать в основной части курсовой работы.

Первая глава – это анализ предметной области, в состав которого входит исследование основных сущностей, построение на его основе инфологической модели данных и разработка проекта приложения базы данных.

Пример анализа предметной области и построения инфологической модели: *Анализируя предметную область «Разработка программного*

приложения для обучения детей основам музыкальной литературы», можно выделить следующие сущности:

1. Музыкальный инструмент. Содержит информацию о музыкальных инструментах
2. Категория. Содержит информацию о категориях, к которым относятся музыкальные инструменты.
3. Резонатор. Содержит информацию о резонаторах (способах извлечения звука).
4. Человек. Обобщающая сущность для сущностей «Композитор» и «Мастер».
5. Композитор.
6. Мастер.
7. Страна.
8. Произведение.
9. Жанр.

Для данной предметной области можно составить инфологическую модель (рис 5).

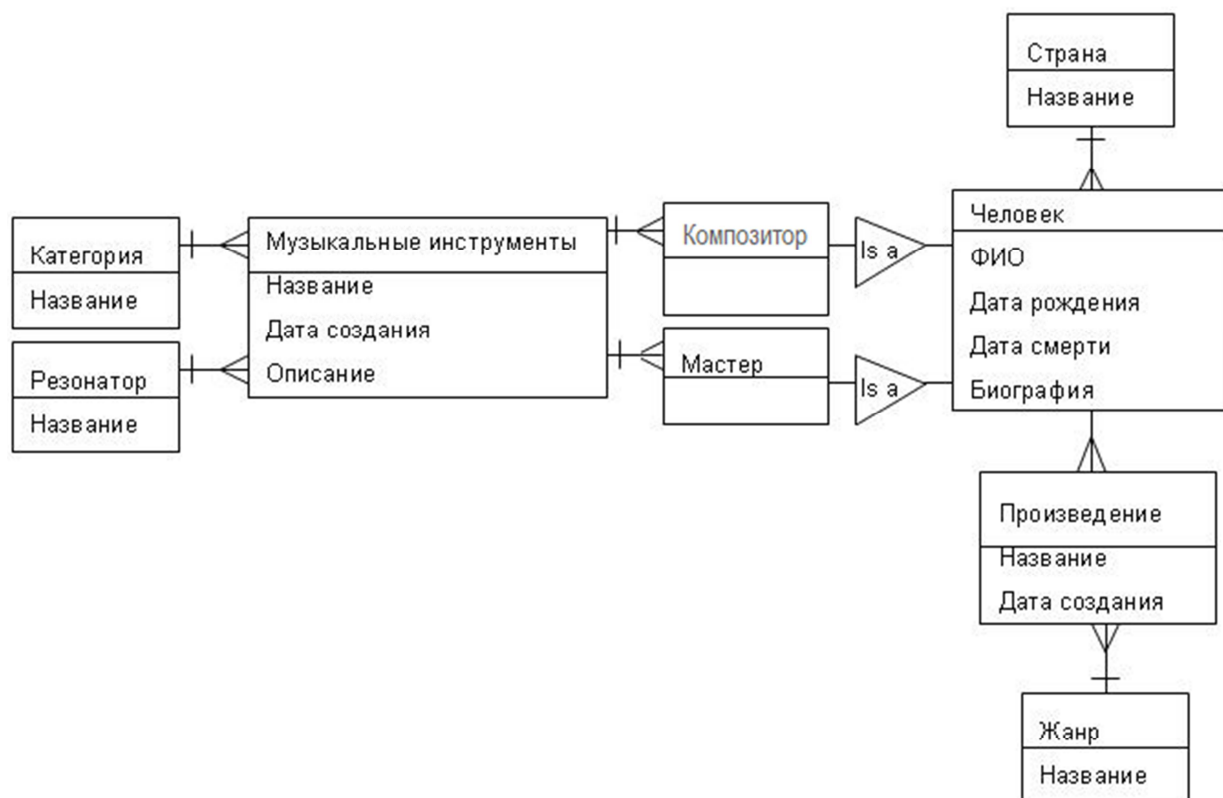


Рисунок 5. Инфологическая модель данных

Оформлять инфологическую модель данных необходимо строго в соответствии с графической нотацией Crow's Foot.

Проект приложения базы данных должен включать в себя структурную схему приложения и его модулей, диаграммы переходов состояний интерфейса, диаграмму приложения и объектов приложения.

Необходимо представить структурную схему разрабатываемого программного приложения, позволяющую определить взаимоотношения между отдельными частями программы и содержание программных сообщений. На основе этой схемы можно построить схему движения информационных потоков, диаграмму взаимодействия классов и т.д.

Пример описания структурной схемы приложения базы данных:
Структурная схема приложения (рис. 6) должна иметь следующие модули:

- 1. модуль интерфейса, предназначенный для того, чтобы пользователь получил общую информацию о приложении. В нем содержится управляющий элемент, предназначенный для перехода в модуль авторизации сотрудника;*
- 2. модуль авторизации, в котором пользователю необходимо ввести логин и пароль для работы в приложении;*
- 3. модуль работы с расписанием, предназначенный для учета часов преподавателей, закрепления преподавателей за группами по дисциплинам и распределения групп и преподавателей по кабинетам. Также модуль должен предоставлять возможность загружать из XML-файла необходимую информацию в таблицу на форме, редактировать ее, сохранять обратно в XML-файл и выводить на печать.*

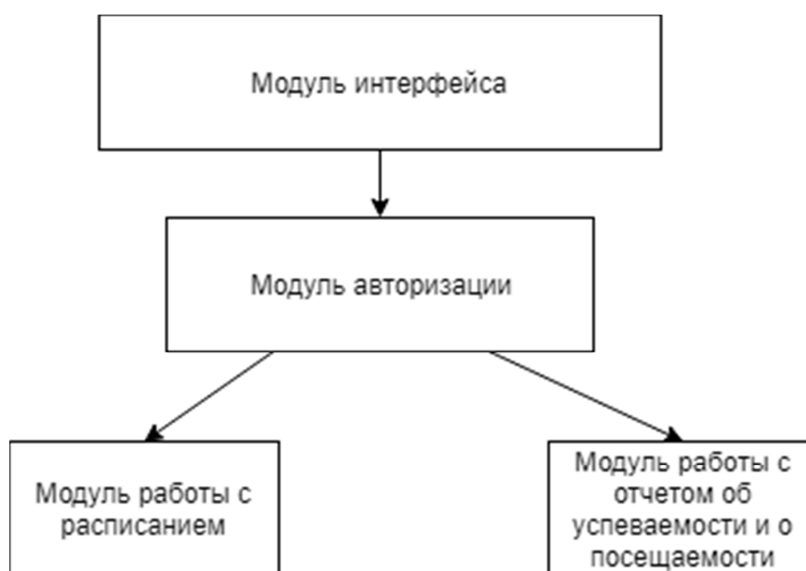


Рисунок 6. Структурная схема приложения

Пример описания структурной схемы одного из модулей приложения базы данных: *Модуль работы с документом о посещаемости и успеваемости позволяет оформлять сводную ведомость по посещаемости за месяц; предоставляет возможность просматривать и редактировать результаты аттестации, загружать из XML-файла необходимую информацию в таблицу на форме, редактировать ее и снова сохранять в XML-файл (рис. 7).*



Рисунок 7 - Структурная схема модуля работы с отчетом о посещаемости и успеваемости

Пример описания диаграммы переходов состояний интерфейса приложения базы данных: Рассмотрим диаграмму переходов состояний интерфейса для модуля работы с расписанием (рис. 8).

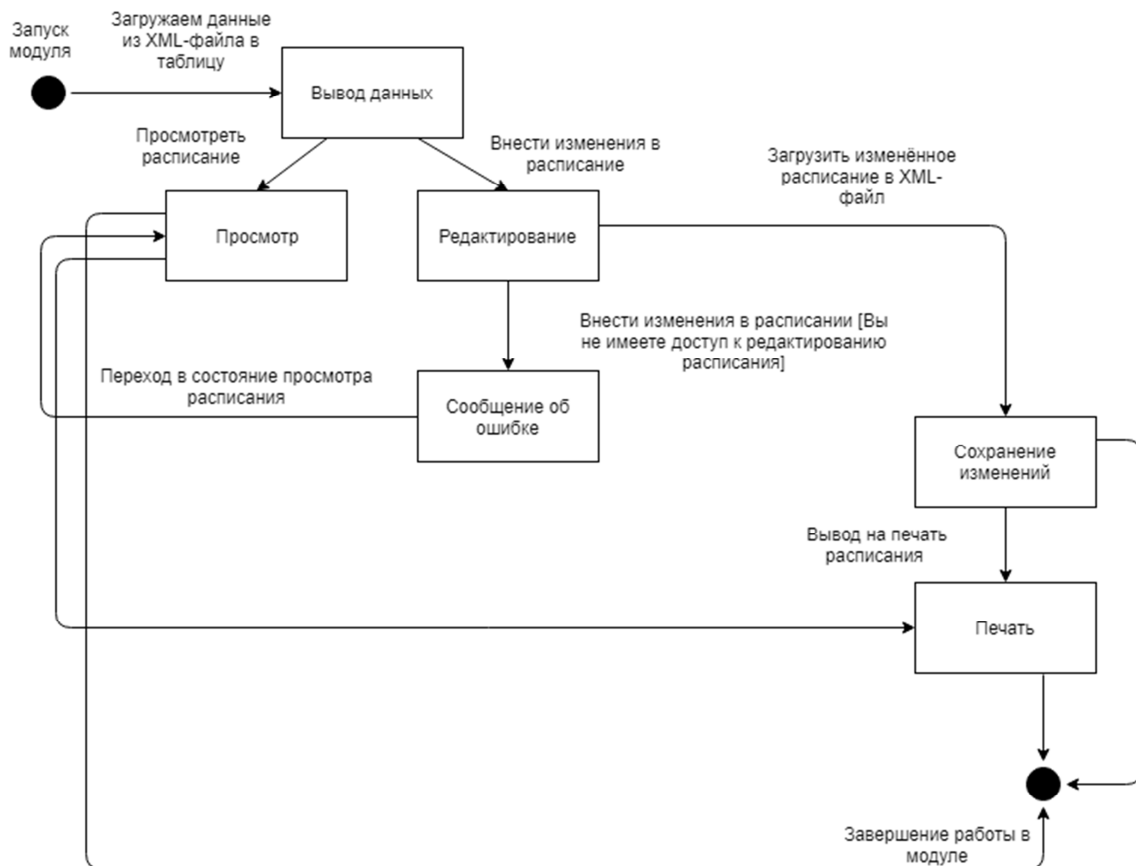


Рисунок 8 - Диаграмма переходов состояний интерфейса для модуля работы с расписанием

Пример оформления диаграммы классов приложения (рис. 9).

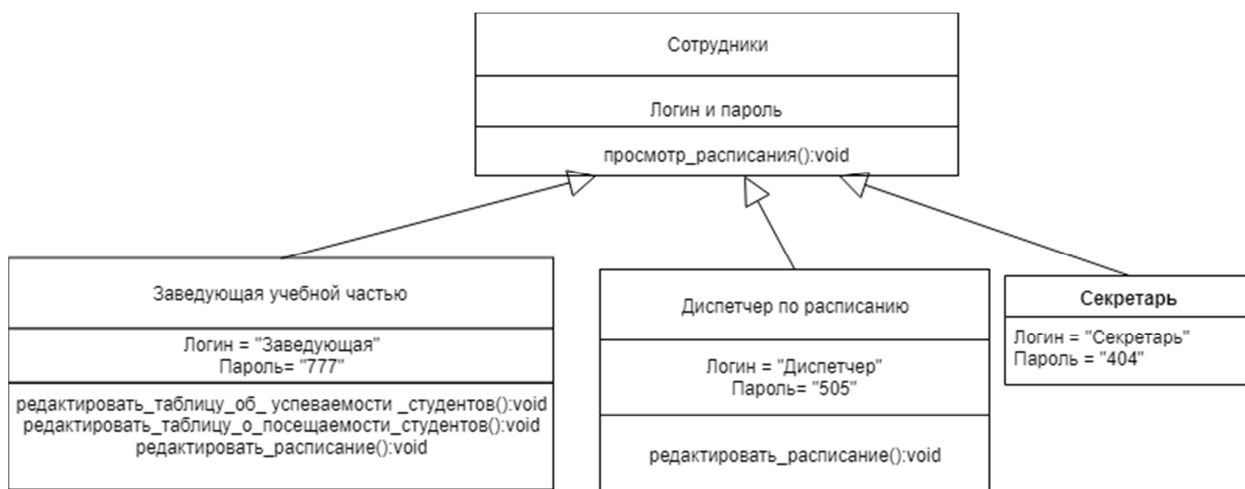


Рисунок 9 - Диаграмма классов приложения

Пример оформления диаграммы объектов приложения (рис. 10).

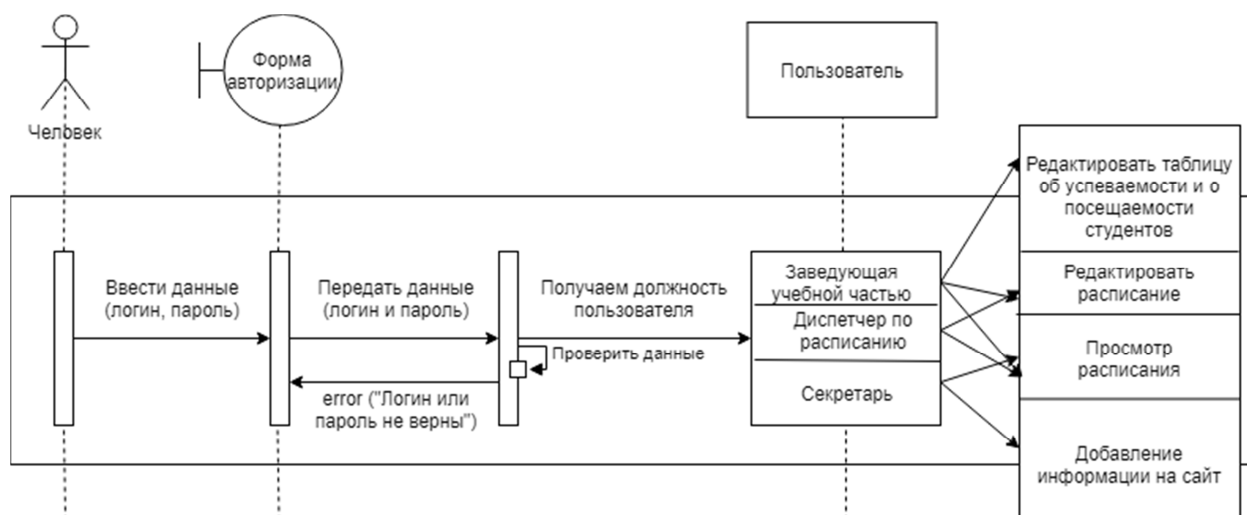


Рисунок 10 - Диаграмма объектов приложения

4. Вторая глава имеет практическую направленность, а именно: разработка даталогической модели данных, разработка программного приложения. Таким образом, в данной главе обучающемуся необходимо представить результат разработки базы данных (текст SQL-запросов создания таблиц) и снимки экрана разработанного приложения. Программная реализация составленного алгоритма представляет собой самостоятельную работу студента.

Пример оформления результатов разработки базы данных: *Текст SQL-запроса на создание таблицы person в системе управления базами данных PostgreSQL выглядит следующим образом:*

```

CREATE TABLE person(
  idperson integer NOT NULL,
  f character varying(50),
  i character varying(50),
  o character varying(50),
  birthdayyear integer,
  idgroup integer NOT NULL,
  avg_exam real,
  CONSTRAINT person_pkey PRIMARY KEY (idperson),

```

*CONSTRAINT person_idgroup_fkey FOREIGN KEY (idgroup)
REFERENCES grupp (idgroup)*

MATCH SIMPLE ON UPDATE NO ACTION ON DELETE NO ACTION)

Результатом выполнения данного запроса является таблица с колонками idperson, f, i, o, birthdayyear, idgroup, avg_exam real (рис. 11).

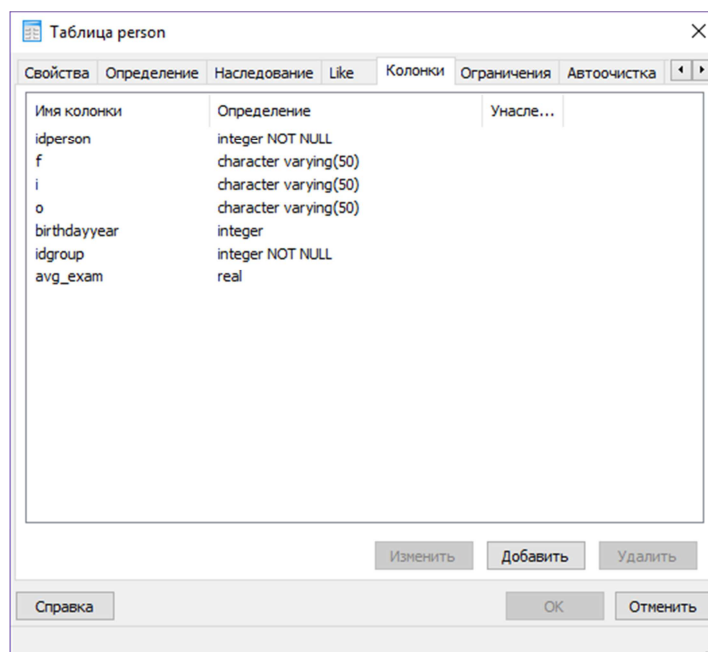


Рисунок 11 – Результат создания таблицы person

Обучающийся должен представить текст SQL-запросов на выборку, которые он использует для получения информации из базы данных.

Если во время разработки базы данных были созданы представления, триггеры, курсоры или другие объекты, SQL-запросы их создания также должны быть представлены в тексте курсовой работы.

Обучающийся может приводить фрагменты кода, если они представляют какую-либо научную или практическую значимость, или если ему необходимо сделать акцент на этой части разработки.

Однако, необходимо учесть, что стоит рационально подойти к размещению в тексте пояснительной записки фрагментов кода, копировать все строки программы в работы не требуется.

Результат разработки оформляется в виде снимков экрана, сделанных во время функционирования приложения (рис. 12).

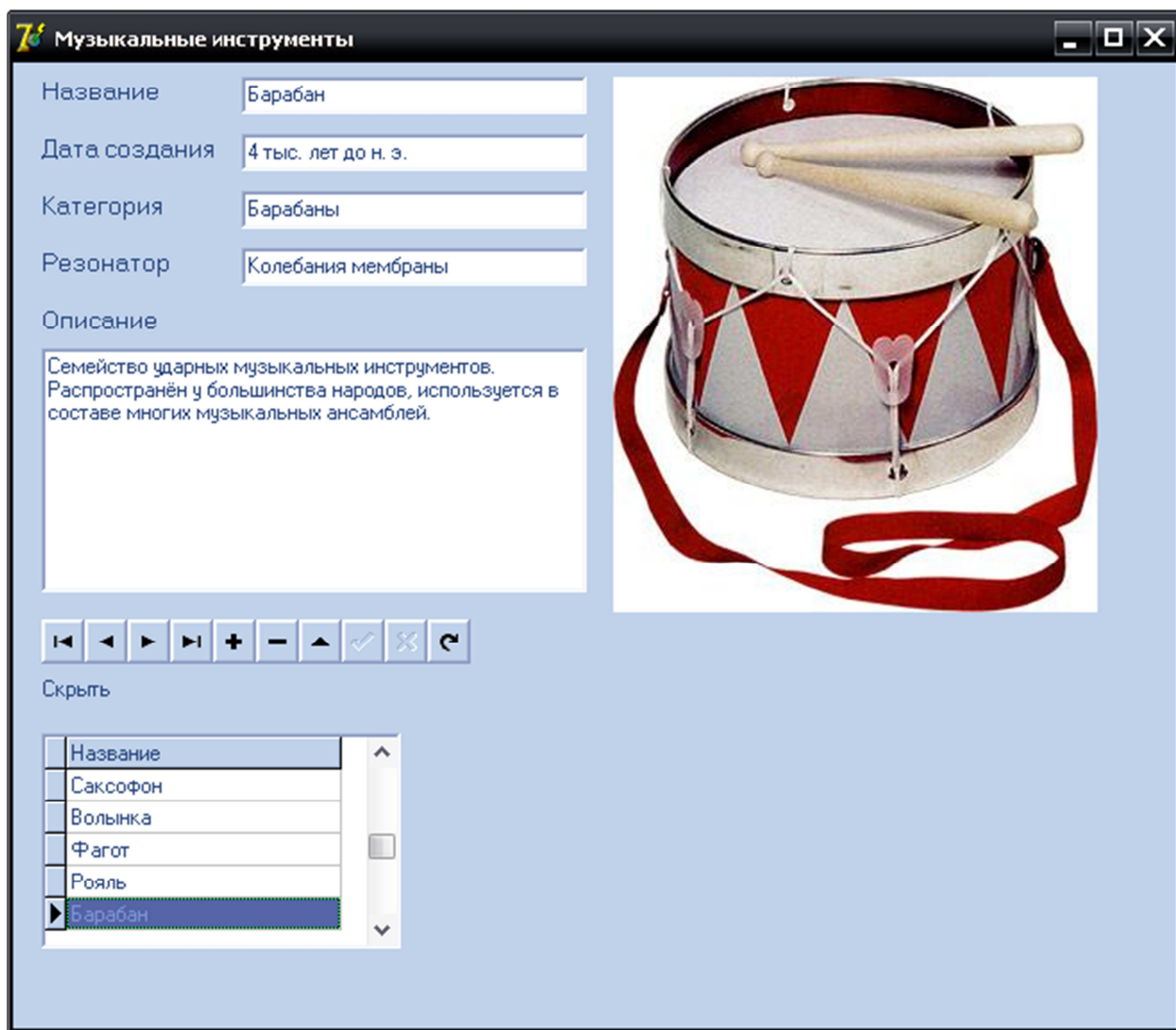


Рисунок 12. Форма приложения базы данных

Выбор конкретных модулей, которые следует показать, осуществляется обучающимся самостоятельно, однако, стоит соблюсти 2 условия:

- наиболее полно продемонстрировать функциональные возможности приложения;
- не допустить избыточности информации.

При необходимости обучающийся может проконсультироваться с преподавателем по поводу графического материала. Необходимо представить окно приложения, в котором осуществляется заполнение базы данных, и окно вывода информации из базы данных на экран.

5. Третья глава представляет собой тестирование приложения базы данных. Тестирование приложения предназначено не только для проверки

корректности работы самого приложения, но и корректности заполнения базы данных. В частности, при взаимодействии пользователей с представлениями важно, чтобы база данных адекватно реагировала на внесенные изменения.

Результат тестирования необходимо представить в виде снимков окон программы с внесенными данными и снимков окна СУБД с соответствующими строками.

6. В заключении обобщаются наблюдения, сделанные во время работы и формулируются основные выводы, отражающие наиболее значимые результаты работы, предлагаются рекомендации относительно возможностей использования материалов и результатов работы. Кроме того, обучающийся может указать направление дальнейших исследований в рамках развития данной задачи.

При составлении текста пояснительной записки следует помнить, что:

- необходимо избегать логических ошибок, например, нельзя давать одинаковое название курсовой работе и одному из ее параграфов;
- содержание работы необходимо иллюстрировать наглядными материалами: таблицами, рисунками, схемами, графиками и т.п.

3. ОРГАНИЗАЦИОННЫЕ ТРЕБОВАНИЯ К ПОДГОТОВКЕ КУРСОВОЙ РАБОТЫ

1. Для направлений подготовки 01.03.02 Прикладная математика и информатика (профиль «Математическое моделирование и информационные технологии»), 02.03.03 Математическое обеспечение и администрирование информационных систем (профиль «Программное и математическое обеспечение информационных технологий») в начале шестого семестра кафедра предлагает студентам третьего курса темы курсовых работ по дисциплине «Современные технологии программирования SQL». Студент может предложить и свою тему, но ее необходимо обосновать, согласовать с руководителем и зарегистрировать на кафедре. Общее руководство и контроль за ходом выполнения курсовой работы осуществляет руководитель курсовой работы.

2. После выбора темы студенту следует встретиться с руководителем курсовой работы для обсуждения задачи. Руководитель выясняет степень подготовленности студента к работе над темой, рекомендует необходимую литературу и дает консультации о порядке выполнения работы. Для самоорганизации студента и своевременного контроля за ходом работы можно составить график ее выполнения.

График выполнения курсовой работы по дисциплине «Современные технологии программирования SQL»

студента (ки) группы _____

шифр группы _____

ФИО _____

№ п/п	Этапы работы	Планируема я дата выполнения	Фактическая дата	Отметка научного руководи теля
1	2	3	4	5
1.	Выбор темы, ее регистрация на кафедре, обсуждение целей и задач работы, графика ее выполнения с руководителем курсовой работы. Подбор библиографических			

	источников по теме, согласование с руководителем.			
2.	Подготовка чернового варианта теоретической части работы и представление его руководителю работы. Работа над замечаниями руководителя курсовой работы, исправление недочетов и повторное представление.			
3.	Выполнение практической части работы: разработка инфологической модели данных. Представление руководителю. Исправление недочетов и повторное представление руководителю.			
4.	Выполнение практической части работы: реализация модели данных в СУБД, разработка приложения базы данных. Представление руководителю завершенной разработки. Исправление недочетов и повторное представление руководителю.			
5.	Подготовка чернового варианта практической части работы и представление его руководителю. Работа над замечаниями руководителя. Написание введения и заключения, оформление библиографического списка, титульного листа работы			
6.	Представление руководителю чернового варианта всей работы. Проверка на некорректные заимствования. Исправление недочетов. Подготовка чистового варианта работы и представление его руководителю.			
7.	Защита курсовой работы			

Руководитель курсовой работы:_____

ФИО

Дата «_____» _____2020____

3. После беседы с руководителем студент приступает к подбору литературы, ее анализу и составлению плана работы. Кроме рекомендуемой руководителем литературы, желательно начать поиск дополнительной литературы по данной теме. Изучая литературу, полезно делать необходимые выписки (конспектировать), закладки, заметки.

4. Желательно, чтобы в процессе написания курсовой работы студент периодически встречался с руководителем, показывал ему законченные части работы, обсуждал свои идеи и наработки. Это поможет детализировать процесс написания курсовой работы, предостережет от ложных направлений, даст дополнительный импульс к творческому поиску. Руководитель свои рекомендации по выполнению курсовой работы дает лишь после того, как он убедился, что студент достаточно изучил проблему и понял ее сущность.

5. Опыт показывает, что даже при самом добросовестном отношении студента, первый вариант курсовой работы потребует существенных исправлений и доработки. Перечитав и выправив черновик, студент передает его руководителю курсовой работы, который принимает решение либо о допуске работы к защите, либо о доработке и устранении имеющихся недочетов и ошибок.

6. Чтобы студент мог успешно защитить курсовую работу, он должен хорошо владеть ее содержанием, уметь доказывать и обосновывать полученные результаты.

4. ТРЕБОВАНИЯ К ОФОРМЛЕНИЮ КУРСОВОЙ РАБОТЫ

1. Объем курсовой работы зависит от темы. Студенту рекомендуется быть максимально кратким, обходиться без повторов, опускать совершенно очевидные выкладки, расчеты и пояснения. Опыт показывает, что объем курсовой работы колеблется в пределах 20-35 страниц машинописного текста.

2. Структурные элементы курсовой работы описаны ранее.

3. Тексты курсовой работы оформляются на одной стороне листа бумаги стандартного размера А4 (210×297 мм). Каждый лист вкладывается в отдельный файл (мультифору). Все файлы должны быть соединены вместе (сшиты в папке – скоросшивателе).

4. Оформлять курсовую работу необходимо в соответствии с Правилами оформления учебных работ студентов¹.

5. Курсовая работа должна содержать самостоятельные выводы и авторский текст не менее 70%. Проверка на наличие некорректных зависимостей осуществляется руководителем курсовой работы с помощью системы antiplagiat.ru во время предоставления обучающимся на проверку черновой работы. Обучающемуся рекомендуется самостоятельно заранее проверять свою работу по мере написания текста.

¹ Правила оформления учебных работ студентов : учебно-методическое пособие / И.А. Жибинова, А.Е. Аракелян, О.В. Соколова, Ю.Н. Соина-Кутищева. – Новокузнецк : НФИ КемГУ, 2018. – 124 с. – Текст : непосредственный

5. ОФОРМЛЕНИЕ БИБЛИОГРАФИЧЕСКОГО СПИСКА И БИБЛИОГРАФИЧЕСКИХ ССЫЛОК

Особое место среди методов исследования занимают методы изучения литературы по проблеме данной работы. Изучение литературы служит средством изучения истории и причин возникновения проблемы, ее современного состояния. Работа с литературой включает несколько методов.

Укажем наиболее важные из них.

1. Составление библиографии, то есть списка литературы, использованной автором при написании курсовой работы.
2. Реферирование - сжатое изложение основного содержания одной или нескольких работ по общей теме.
3. Конспектирование - детальное изложение главных положений и концептуальных идей работы.
4. Аннотирование - краткое, предельно сжатое изложение основного содержания литературных источников.
5. Цитирование - дословная запись высказываний, выражений автора, а также приведение в тексте курсовой работы фактических и статистических данных, содержащихся в литературных источниках.

Еще одно правило работы с литературой - использование библиографического списка в тексте работы: источник, внесенный в список, хотя бы один раз должен быть назван в тексте. И, наоборот, любой источник, на который автор ссылается в тексте работы, должен быть внесен в библиографический список.

При написании курсовой работы наиболее приемлемым является алфавитный способ группировки материалов, включенных в список использованных источников. И использованные источники при этом располагаются в общем алфавитном порядке фамилий авторов или заглавий книг, статей (если автор не указан).

Обращаясь к цитированию, необходимо соблюдать следующие правила:

нельзя вырывать фразы из текста; искажать его произвольными сокращениями; цитату необходимо брать в кавычки и т.п. Следует также обратить внимание на точное указание источников цитат.

Ссылки на использованную литературу в тексте указываются в квадратных скобках порядковым номером по списку источников, например, [23]. Если автор делает ссылку сразу на несколько источников, то в скобках через запятую указываются их номера в списке, например, [23, 25, 38]. Если автор приводит цитату из первоисточника, то в скобках после номера источника обязательно указывается страница, например, [34. С. 216].

Список литературы оформляется в соответствии с требованиями **ГОСТ Р 7.0.100-2018**.

*Примеры библиографического описания информационных источников
по ГОСТ Р 7.0.100-2018*

электронные издания:

1. Маркин, А.В. Построение запросов и программирование на SQL: учебное пособие / А.В. Маркин. – Москва : «Диалог-МИФИ», 2014. – 384 с. – ISBN 978-5-86404-227-4. - URL: http://biblioclub.ru/index.php?page=book_view_red&book_id=89077 (дата обращения: 14.01.2019). – Текст : электронный..

2. Разработка приложений на C# с использованием СУБД PostgreSQL: учебное пособие / И.А. Васюткина, Г.В. Трошина, М.И. Бычков, С.А. Менжулин. – Новосибирск : Изд-во НГТУ, 2015. – 143 с. – ISBN 978-5-7782-2699-9. - URL: http://biblioclub.ru/index.php?page=book_view_red&book_id=438432 (дата обращения: 14.01.2019). – Текст : электронный.

сайты в сети «Интернет»:

CITForum.ru : on-line библиотека свободно доступных материалов по информационным технологиям на русском языке : сайт. – 2001 – URL: <http://citforum.ru> (дата обращения: 22.03.2020). – Текст: электронный.

eLIBRARY.RU : научная электронная библиотека : сайт. – Москва, 2000 - . – URL: <http://www.elibrary.ru> (дата обращения: 22.03.2020). – Режим доступа: для зарегистрир. пользлвателей. – Текст: электронный.

Единое окно доступа к образовательным ресурсам : сайт. – Москва, 2005 - . – URL: <http://window.edu.ru/> (дата обращения: 22.03.2020). –Текст: электронный.

Курсовая работа с неправильно или небрежно оформленным библиографическим списком к защите не допускается.

6. ЗАЩИТА КУРСОВОЙ РАБОТЫ

1. Курсовая работа должна быть защищена не позднее начала экзаменационной сессии.

2. Руководитель курсовой работы, ознакомившись с окончательным вариантом курсовой работы, принимает решение о допуске к защите курсовой работы.

3. Во время защиты обучающийся кратко (3 - 5 минут) докладывает основные результаты своей работы, сопровождая результат наглядным материалом (презентация), а затем отвечает на вопросы преподавателя.

В докладе обучающийся озвучивает цель и задачи исследования; используемые при разработке алгоритмические и программные решения; основные результаты разработки; выводы по работе.

4. Преподаватель оценивает, насколько успешно достигнута цель и решены задачи, поставленные в курсовой работе; качество доклада и ответов на вопросы; степень самостоятельности студента при выполнении исследования (анализе предметной области, инфологическом моделировании); полнота проведенной работы (разработка базы данных, наполнение ее информацией; реализация программного приложения базы данных). Отметка за курсовую работу выставляется согласно приведенной в методических указаниях балльно-рейтинговой системе и таблице перевода баллов в буквенный эквивалент.

5. После этого руководитель курсовой работы заносит отметку в зачетную книжку студента и в соответствующую ведомость.

Курсовая работа хранится на кафедре вплоть до завершения студентом учебы в университете.

7. КРИТЕРИИ ОЦЕНИВАНИЯ КУРСОВОЙ РАБОТЫ В БАЛЛЬНО-РЕЙТИНГОВОЙ СИСТЕМЕ

Баллы по курсовой работе выставляются обучающемуся за два вида деятельности:

- выполнение исследования и оформление пояснительной записки (80 баллов) (таблица 1);
- защита курсовой работы (20 баллов) (таблица 2).

Таблица 1 – Распределение баллов по этапам выполнения исследования

Этапы выполнения исследования	Оценка в аттестации	Баллы
Этап 1. Анализ предметной области. 1.1 Анализ требований к базе данных и приложению базы данных 1.2 Построение инфологической модели данных 1.3 Разработка проекта приложения баз данных	Баллы за этап 1.1: 2 балла (собраны требования хотя бы к одному объекту: к базе данных или приложению базы данных, проведен анализ требований, сделаны выводы, недочеты по работе не критичны) 3 балла (самостоятельно собраны требования к базе данных и приложению базы данных, анализ требований проведен в полном объеме, выводы корректны, но не полны) 4 балла (самостоятельно собраны требования к базе данных и приложению базы данных, в полном объеме проведен анализ требований, сделаны корректные выводы) Баллы за этап 1.2: 6 баллов (инфологическая модель данных построена, соответствует предметной области, однако в ней не учтены требования к базе данных или к программному приложению) 7 баллов (самостоятельно построена инфологическая модель данных, соответствующая предметной области и требованиям из части 1.1, имеются некоторые неточности, или недочеты) 8 баллов (самостоятельно построена инфологическая модель данных,	14-20

	соответствующая предметной области и требованиям из части 1.1) Баллы за этап 1.3: 6 баллов (проект приложения базы данных разработан, но не в полном объеме (не все диаграммы и схемы построены)) 7 баллов (проект приложения базы данных разработан в полном объеме, однако наблюдаются некоторые неточности или недочеты) 8 баллов (проект приложения базы данных разработан в полном объеме)	
Этап 2. Разработка приложения баз данных 2.1 Реализация даталогической модели базы данных в СУБД 2.2 Разработка запросов к базе данных на языке SQL 2.3 Разработка модулей приложения баз данных	Баллы за этап 1.1: 5 баллов (даталогическая модель базы данных реализована в СУБД не в полном объеме (не все таблицы инфологической модели перенесены в базу данных)) 7 баллов (даталогическая модель базы данных реализована в СУБД самостоятельно и в полном объеме, но наблюдаются некоторые неточности или недочеты) 10 баллов (даталогическая модель базы данных реализована в СУБД самостоятельно и в полном объеме) Баллы за этап 1.2: 5 баллов (запросы к базе данных разработаны не в полном объеме (не все требуемые функциональные возможности приложения реализованы)) 7 баллов (запросы к базе данных разработаны в объеме, предусмотренном требованиями к базе данных, однако наблюдаются некоторые неточности или недочеты) 10 баллов (запросы к базе данных разработаны в объеме, предусмотренном требованиями к базе данных) Баллы за этап 1.3: 13 баллов (модули приложения базы данных	23-40

	разработаны, но не в полном объеме (не соответствует разработанному проекту) 17 баллов (самостоятельно и в полном объеме разработаны модули приложения базы данных, однако наблюдаются некоторые недочеты) 20 баллов (самостоятельно и в полном объеме разработаны модули приложения базы данных)	
Этап 3. Тестирование приложения баз данных	Баллы за этап 3: 14 баллов (проведено тестирование хотя бы одного объекта: базы данных или программного приложения) 17 баллов (проведено тестирование базы данных и программного приложения не в полном объеме или с недочетами) 20 баллов (проведено тестирование базы данных и программного приложения в полном объеме)	14-20
Итого за выполнение курсовой работы		51-80

Таблица 2 – Распределение баллов за защиту курсовой работы

Виды и результаты учебной работы	Оценка в аттестации	Баллы
Устное выступление об основных результатах, полученных во время выполнения курсовой работы (5-7 минут)	6 баллов – пороговое значение; 10 баллов – максимальное значение.	6 - 10
Сопровождение устного выступления наглядным материалом (презентация)	2 балла – пороговое значение; 5 баллов – максимальное значение.	2 - 5
Ответы на вопросы по теме курсовой работы	2 балла – пороговое значение; 5 баллов – максимальное значение.	2 - 5
Итого по промежуточной аттестации (защита курсовой работы)		10-20

Набранные обучающимся баллы переводятся в оценку и буквенный эквивалент согласно Положению о балльно-рейтинговой системе оценки деятельности студентов КемГУ от 30.12.2016г. (таблица 3).

Таблица 3 - Перевод баллов из 100-балльной шкалы в числовой и
буквенный эквивалент

Сумма баллов для дисциплины	Оценка	Буквенный эквивалент
86 - 100	5	отлично
66 - 85	4	хорошо
51 - 65	3	удовлетворительно
0 - 50	2	неудовлетворительно

8. РЕКОМЕНДУЕМАЯ ЛИТЕРАТУРА

Основная учебная литература

1. Маркин, А.В. Построение запросов и программирование на SQL: учебное пособие / А.В. Маркин. – Москва : «Диалог-МИФИ», 2014. – 384 с. – ISBN 978-5-86404-227-4. - URL: http://biblioclub.ru/index.php?page=book_view_red&book_id=89077 (дата обращения: 14.01.2019). – Текст : электронный.

2. Разработка приложений на C# с использованием СУБД PostgreSQL: учебное пособие / И.А. Васюткина, Г.В. Трошина, М.И. Бычков, С.А. Менжулин. – Новосибирск : Изд-во НГТУ, 2015. – 143 с. – ISBN 978-5-7782-2699-9. - URL: http://biblioclub.ru/index.php?page=book_view_red&book_id=438432 (дата обращения: 14.01.2019). – Текст : электронный.

3. Волк, В.К. Базы данных. Проектирование, программирование, управление и администрирование : учебник / В.К. Волк. – Санкт-Петербург : Лань, 2020. – 244 с. – ISBN 978-5-8114-4189-1. - URL: <https://e.lanbook.com/reader/book/126933> (дата обращения: 14.01.2019). – Текст : электронный.

Дополнительная литература

1. Маркин, А.В. Построение запросов и программирование на SQL : учебное пособие / А.В. Маркин. – Москва : Издательство «Диалог-МИФИ», 2014. – 384 с. – ISBN 978-5-86404-227-4. – URL: http://biblioclub.ru/index.php?page=book_view_red&book_id=89077. – (дата обращения: 22.03.2020). – Текст : электронный.

2. СУБД: язык SQL в примерах и задачах : учебное пособие / И.Ф. Астахова, В.М. Мельников, А.П. Толстобров, В.В. Фертиков. – Москва : ФИЗМАТЛИТ, 2009. – 168 с. – ISBN 978-5-9221-0816-4.- URL: http://biblioclub.ru/index.php?page=book_view_red&book_id=76768. – (дата обращения: 22.03.2020). – Текст : электронный.

3. Гудов, А.М. Базы данных и системы управления базами данных. Программирование на языке PL/SQL: учебное пособие /А.М. Гудов, С.Ю. Завозкин, Т.С. Рейн. – Кемерово : ГОУ ВПО «Кемеровский государственный университет», 2010. – 134 с. – ISBN 978-5-8353-1005-0. – URL: http://biblioclub.ru/index.php?page=book_view_red&book_id=232497 (дата обращения: 14.01.2019). – Текст : электронный.

Литература для оформления курсовой работы

Правила оформления учебных работ студентов : учебно-методическое пособие / И.А. Жибинова, А.Е. Аракелян, О.В. Соколова, Ю.Н. Соина-Кутищева. – Новокузнецк : НФИ КемГУ, 2018. – 124 с. – Текст : непосредственный

СОВРЕМЕННЫЕ ПРОФЕССИОНАЛЬНЫЕ БАЗЫ ДАННЫХ И СПРАВОЧНЫЕ СИСТЕМЫ

CITForum.ru : on-line библиотека свободно доступных материалов по информационным технологиям на русском языке : сайт. – 2001 – URL: <http://citforum.ru> (дата обращения: 22.03.2020). – Текст: электронный.

eLIBRARY.RU : научная электронная библиотека : сайт. – Москва, 2000 - . – URL: <http://www.elibrary.ru> (дата обращения: 22.03.2020). – Режим

доступа: для зарегистрир. пользлвателей. – Текст: электронный.

Единое окно доступа к образовательным ресурсам : сайт. – Москва, 2005 - . – URL: <http://window.edu.ru/> (дата обращения: 22.03.2020). –Текст: электронный.

ПРИЛОЖЕНИЕ 1. Титульный лист
МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Новокузнецкий институт (филиал)
федерального государственного бюджетного образовательного
учреждения высшего образования
«Кемеровский государственный университет»

Факультет информатики, математики и экономики
Кафедра математики, физики и математического моделирования

Иванов Иван Иванович
гр. ПМИ-19-1

РАЗРАБОТКА БАЗЫ ДАННЫХ И ПРОГРАММНОГО ПРИЛОЖЕНИЯ ДЛЯ
СТОМАТОЛОГИЧЕСКОЙ КЛИНИКИ

Курсовая работа
по дисциплине «Современные технологии программирования SQL»

по направлению подготовки 01.03.02 Прикладная математика и информатика
направленность (профиль) подготовки «МАТЕМАТИЧЕСКОЕ МОДЕЛИРОВАНИЕ И
ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ»

Проверил:
канд. техн. наук, доцент
Е.В. Решетникова

Общий балл: _____

Оценка: _____

_____ подпись

«____» _____ 20____ г.

Новокузнецк 20_____

ПРИЛОЖЕНИЕ 2. Задание на курсовую работу

**Новокузнецкий институт (филиал)
федерального государственного бюджетного образовательного
учреждения высшего образования
«Кемеровский государственный университет»**

Факультет информатики, математики и экономики

Кафедра математики, физики и математического моделирования

УТВЕРЖДАЮ

заведующий кафедрой

_____ Е.В. Решетникова

«____» _____

ЗАДАНИЕ

на курсовую работу

студенту гр. ПМИ-19-1 Иванову Ивану Ивановичу

1. Тема курсовой работы «Разработка базы данных и программного приложения для стоматологической клиники» утверждена распоряжением декана _____.

2. Исходные данные: печатные и электронные учебные издания, ресурсы сети интернет.

Содержание работы (перечень подлежащих разработке вопросов):

Подобрать и проанализировать научную литературу по теме «разработка базы данных и программного приложения для стоматологической клиники»; исследовать предметную область; собрать и проанализировать требования к базе данных и программному приложению; построить инфологическую модель данных; реализовать базу данных с помощью современной СУБД; разработать программное приложение.

3. Задание выдано: «____» _____ 20__ г.

4. Срок сдачи курсовой работы: «____» _____ 20__ г.

Руководитель канд. техн. наук, доцент _____ Е.В. Решетникова

Студент гр. ПМИ-19-1 _____ И.И. Иванов

ПРИЛОЖЕНИЕ 3. Образец введения курсовой работы

**Тема: Разработка базы данных и программного приложения для
стоматологической клиники**

Введение

Актуальность исследования

Большинство программных средств функционирует на основе баз данных. Хранение информации в них является довольно надежным способом, защищающим данные от потери или повреждения. Общепринятые требования к разработке базы данных позволяют упростить их дальнейшую поддержку. Также достаточно важной является задача разработки программного приложения базы данных на основе требований к функциональным и нефункциональным характеристикам. На основании вышеизложенного можно считать актуальной задачу разработки базы данных и программного приложения, предназначенных для эффективной организации информационных процессов стоматологической клиники.

Цель исследования: разработать базу данных и программное приложение для стоматологической клиники.

Задачи исследования:

1. анализ предметной области решения задачи;
2. сбор и анализ требований к базе данных и приложению базы данных;
3. построение инфологической модели данных;
4. реализация инфологической модели данных средствами современной СУБД;
5. разработка программного приложения;
6. тестирование разработанного программного приложения.

Теоретическая значимость исследования заключается в сборе и систематизации сведений о предметной области, анализе полученных данных.

Практическая значимость исследования заключается в том, что выводы и результаты курсовой работы (разработанная инфологическая модель данных и программное приложение) могут быть использованы на практике при решении аналогичных задач.

ПРИЛОЖЕНИЕ 4. Примерные темы курсовых работ по дисциплине «Современные технологии программирования SQL»

1. Разработка базы данных и программного приложения для поликлиники.
2. Разработка базы данных и программного приложения для страховой компании.
3. Разработка базы данных и программного приложения для службы доставки.
4. Разработка базы данных и программного приложения для железнодорожного вокзала.
5. Разработка базы данных и программного приложения для центра дополнительного образования.
6. Разработка базы данных и программного приложения для стоматологической клиники.
7. Разработка базы данных и программного приложения для библиотеки.
8. Разработка базы данных и программного приложения для продуктового магазина.
9. Разработка базы данных и программного приложения для фитнес-клуба.
10. Разработка базы данных и программного приложения для микрофинансовой организации, занимающейся выдачами кредитов.
11. Разработка базы данных и программного приложения для рекламного агентства.
12. Разработка базы данных и программного приложения для онлайн-сервиса продажи билетов.
13. Разработка базы данных и программного приложения для парикмахерской.
14. Разработка базы данных и программного приложения для

аэропорта.

15. Разработка базы данных и программного приложения для учета дорожно-транспортных происшествий.

16. Разработка базы данных и программного приложения для учета успеваемости студентов.

17. Разработка базы данных и программного приложения для обувного магазина.

18. Разработка базы данных и программного приложения для охранного предприятия.

19. Разработка базы данных и программного приложения для турагентства.

20. Разработка базы данных и программного приложения для магазина музыкальных товаров.

21. Разработка базы данных и программного приложения для учета заявок в отделе технической поддержки пользователей.

22. Разработка базы данных и программного приложения для кинотеатра.

23. Разработка базы данных и программного приложения для ресторана.

24. Разработка базы данных и программного приложения для составления расписания.

25. Разработка базы данных и программного приложения для агентства недвижимости.

26. Разработка базы данных и программного приложения для рекрутингового агентства.

27. Разработка базы данных и программного приложения для гостиницы.

28. Разработка базы данных и программного приложения для оператора каршеринга.

29. Разработка базы данных и программного приложения для

почтового отделения.

30. Разработка базы данных и программного приложения для автовокзала.

31. Разработка базы данных и программного приложения для кадрового учета.

32. Разработка базы данных и программного приложения для автосервиса.

33. Разработка базы данных и программного приложения для книжного магазина.

34. Разработка базы данных и программного приложения для детского оздоровительного лагеря.

35. Разработка базы данных и программного приложения для автосервиса.

36. Разработка базы данных и программного приложения для аптеки.

37. Разработка базы данных и программного приложения для троллейбусного депо.

38. Разработка базы данных и программного приложения для афиши событий города.

39. Разработка базы данных и программного приложения для магазина бытовой техники.

40. Разработка базы данных и программного приложения для таксопарка.