МЕТОДИЧЕСКИЕ УКАЗАНИЯ К КУРСОВОЙ РАБОТЕ ПО ДИСЦИПЛИНЕ «УПРАВЛЕНИЕ ДАННЫМИ»

ВВЕДЕНИЕ

В курсовой работе по дисциплине «Управление данными» рассматриваются вопросы концептуального, логического и физического проектирования локальных баз данных. В ходе выполнения курсового проектирования студенты проводят исследовании по выбранной тематике, обзор методов реализации алгоритмической постановки задачи, приобретают углубленные знания по вопросам проектирования баз данных оптимизации логической структуры изучения целевой СУБД и языка SQL, приобретают навыки самостоятельной работы.

1 ЦЕЛИ И ЗАДАЧИ КУРСОВОЙ РАБОТЫ

Основными целями курсовой работы по дисциплине «Управление данными» являются:

– упорядочивание информационных потоков (документооборота) внутри организации; выработка рекомендаций по построению рациональных технологий работы подразделений организации и ее взаимодействие с внешним миром;

– систематизация, закрепление и расширение теоретических знаний по специальности и применение их для решения задач концептуального и логического проектирования базы данных;

– изучение целевой СУБД, CASE-технологий, языка SQL для физического проектирования базы данных;

– выявление степени подготовленности студента к самостоятельной работе, его творческого потенциала.

Основной задачей курсового проектирования является углубление подготовки по дисциплине «Управление данными».

2 СОДЕРЖАНИЕ И ОБЪЕМ КУРСОВОЙ РАБОТЫ

Курсовая работа состоит из пояснительной записки и программного продукта. Пояснительная записка должна быть изложена ясно, четко, сжато, технически грамотно и включать вопросы, входящие в содержание курсовой работы. Пояснительная записка должна оформляться в объеме 27–45 машинописных страниц основного текста на листах бумаги формата А4 и содержать следующий перечень разделов:

Титульный лист
Задание на курсовую работу
СОДЕРЖАНИЕ
ВВЕДЕНИЕ
1 ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ
  1.1 Обзор предметной области
  1.2 Обзор существующего программного обеспечения
  1.3 Концептуальное проектирование базы данных

1.4 Логическое проектирование базы данных
1.5 Физическое проектирование базы данных
2 РАЗРАБОТКА ПРОГРАММНОГО ПРОДУКТА
2.1 Структура программного продукта
2.2 Реализация бизнес-правил
2.3 Руководство программиста
2.4 Руководство пользователя
2.5 Тестирование программного продукта
ЗАКЛЮЧЕНИЕ
СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ
ПРИЛОЖЕНИЕ А Техническое задание на разработку
ПРИЛОЖЕНИЕ Б Образцы документации
ПРИЛОЖЕНИЕ В Примеры отчетов и графиков генерируемые программой

Требования к составу курсовой работы

Минимальные требования к разрабатываемой базе данных и реализуемому программному продукту в зависимости от оценки:

На оценку «отлично»	
Параметр	**значение**
Количество сущностей (таблицы, справочники и т.п)	**11**
Всего Отчетов	**6**
Сложные отчеты (с группировкой или Master–Detail + фильтрация)	**3**
Всего Графиков	**4**
Сложный график (динамически обновляемые данные с учетом фильтрации основного SQL запроса)	**2**
Количество записей в базе данных – всего (основных)	**100 (35)**
SQL запросы на выборку данных, с изменяемыми условиями	**5**
Организация поиска не менее чем на * полях *(для всей БД)* из которых: инкрементный поиск *(для всей БД)*	**3 поля** **1 поле**
Организация фильтрация с использованием не менее * полей *(для всей БД)* из которых с использованием одновременно 2х полей	**7 полей** **2 фильтра**
На оценку «Хорошо»	
Параметр	**значение**
Количество сущностей (таблицы, справочники и т.п)	**9**
Всего Отчетов	**4**
Сложные отчеты (с группировкой или Master–Detail + фильтрация)	**2**
Всего Графиков	**3**
Сложный график (динамически обновляемые данные с учетом фильтрации основного SQL запроса)	**1**
Количество записей в базе данных – всего (основных)	**80 (30)**
SQL запросы на выборку данных, с изменяемыми условиями	**3**
Организация поиска не менее чем на * полях *(для всей БД)* из которых: инкрементный поиск *(для всей БД)*	**2 поля** **1 поле**
Организация фильтрация с использованием не менее * полей *(для всей БД)* из которых с использованием одновременно 2х полей	**5 полей** **1 фильтр**

На оценку «удовлетворительно»	
Параметр	**значение**
Количество сущностей (таблицы, справочники и т.п)	**7**
Всего Отчетов	**3**
Сложные отчеты (с группировкой или Master–Detail + фильтрация)	**2**
Всего Графиков	**2**
Сложный график (динамически обновляемые данные с учетом фильтрации основного SQL запроса)	**1**
Количество записей в базе данных – всего (основных)	**40 (20)**
SQL запросы на выборку данных, с изменяемыми условиями	**2**
Организация поиска не менее чем на * полях *(для всей БД)* из которых: инкрементный поиск *(для всей БД)*	**2 поля** **1 поле**
Организация фильтрация с использованием не менее * полей *(для всей БД)* из которых с использованием одновременно 2х полей	**4 полей** **1 фильтр**

При формировании оценки также влияет способ организации работы с базой данных и выбранная СУБД. Повышающими оценку факторами являются:

– использование многопользовательских СУБД (технология клиент-сервер);

– реализация групповой политики в приложении (в зависимости от типа пользователя предусмотрен разный функционал программного продукта);

– реализация конструкторов отчетов (разветвленная система фильтров и возможность установки режимов сортировки);

Защита курсовой работы

До защиты студент должен получить допуск к защите курсовой работы по результатам проверки преподавателем пояснительной записки и демонстрации программного продукта путем компиляции исходных текстов. Защита включает доклад (4 – 7 минут) о проделанной работе с демонстрацией программного продукта по ходу доклада и ответы на вопросы. По результату защиты ставится оценка.

Для проставления оценки студент должен сдать преподавателю пояснительную записку к курсовой работе и диск (CD или DVD) с программой. На диске должна быть записана следующая информация:

ДИСК (название диска формируется как ***Группа_Фамилия_Инициалы***)

- Программа (Программа со всеми необходимыми файлами. *После копирования программы на жесткий диск компьютера, она должна нормально функционировать*)
  - База данных (*при наличии портабельной версии СУБД, например* MySQL)
- Исходные коды (*Исходные коды программы*)
  - Библиотеки (*Библиотеки необходимые для компиляции программы, не включенные в состав среды разработки*)
  - Компоненты (*Сторонние компоненты необходимые для компиляции*)
- Пояснительная записка (*Пояснительная записка в форматах* ***.doc**, ***.docx**, ***.pdf**)

- Дополнительные материалы (*Материалы необходимые для нормального функционирования программы*)
  - Драйверы
  - Библиотеки
  - Другое

3 ОФОРМЛЕНИЕ КУРСОВОЙ РАБОТЫ

Материал, включаемый в курсовую работу, должен быть обработан и систематизирован. Общими требованиями к тексту курсовой работы являются:

– четкость и логическая последовательность изложения материала;

– убедительность аргументации;

– краткость и точность формулировок, исключающих возможность неоднозначного толкования;

– конкретность изложения результатов работы;

Текст пояснительной записки должен быть тщательно отредактирован, набран на компьютере в текстовом редакторе. Размещается текст на одной стороне листа формата А4 (210×297), при этом соблюдая следующие размеры полей: левое – 25 мм; правое – 15 мм; верхнее – 20 мм; нижнее – 20 мм.

Текст следует печатать шрифтом **Liberation Serif**, параметры стиля основного текста:

Размер шрифта – 14 пунктов.

Расстановка переноса – автоматическая, ширина зоны переноса слов – 0,25.

Выравнивание текста – по ширине.

Абзацный отступ – 1,25 см.

Интервалы до и после абзаца – 0 пт.

Межстрочный интервал – одинарный.

В зависимости от текста меняется начертание. Так русские обозначения и греческие символы имеют **прямое** начертание, а латинские символы – **курсив**.

Англоязычные слова и отдельные буквы на английском языке выделяются курсивом как по всему тексту, так и в подрисуночных подписях и названиях таблиц.

В тексте курсовой работы, за исключением формул, таблиц, рисунков, не допускается:

– применять математический знак «минус» (–) перед отрицательными величинами (следует писать слово «минус»);

– применять без числовых значений математические знаки, например > (больше), < (меньше), = (равно), ≥ (больше или равно), ≤ (меньше или равно), ≠ (не равно), а также знаки № (номер), % (процент).

Нумерация страниц в работе должна быть сквозной. На титульном листе и бланке задания на курсовую работу номер страницы не ставится, но включается в общую нумерацию. Первая страница разделов верхнего уровня печатается без отображения нумерации (номер страницы можно скрыть белым прямоугольником помещенным поверх текста). Нумерация страниц проставляется в нижнем колонтитуле по центру. Номер страницы

проставляется без точки, от края до колонтитула – 14 мм, размер колонцифры – 12 п. ко всему тексту.

Титульный лист является первой страницей курсовой работы. Оформление титульного листа **см. приложение 1**.

Задание на курсовую работу является второй страницей работы. Бланк задания заполняется студентом в рукописной форме. **В приложение 2** приведен бланк задания на курсовую работу.

Оформление заголовков

Текст курсовой работы делят на главы (разделы), параграфы (подразделы) и пункты. Каждую главу (раздел) следует начинать с нового листа (страницы).

Заголовки разделов, подразделов и пунктов следует печатать с абзацного отступа без точки в конце, при этом переносы слов в заголовках не допускаются. Параметры стилей для заголовков:

<table>
<tr><td rowspan="2">Свойство стиля</td><td colspan="4">Заголовки</td></tr>
<tr><td>1-ого уровня</td><td>2-ого уровня</td><td>3-го уровня</td><td>4+ уровня</td></tr>
<tr><td>Выравнивание</td><td colspan="4">По левому краю</td></tr>
<tr><td>Отступ слева</td><td colspan="4">1,25 см.</td></tr>
<tr><td>Межстрочный интервал</td><td colspan="4">1 (Одинарный)</td></tr>
<tr><td>Интервал до абзаца</td><td>0 пт.</td><td>16 пт. / 0 пт.*</td><td colspan="2">0 пт.</td></tr>
<tr><td>Интервал после абзаца</td><td>32 пт.</td><td>16 пт.</td><td colspan="2">0 пт.</td></tr>
<tr><td>Дополнительно</td><td colspan="4">Не добавлять интервал между абзацами одного стиля</td></tr>
<tr><td>Регистр</td><td>Прописные буквы</td><td colspan="3">Как в предложениях</td></tr>
<tr><td>Начертание</td><td>Полужирный</td><td>Полужирный</td><td></td><td>Курсив</td></tr>
<tr><td rowspan="2">Дополнительные параметры:
– Разбивка на страницы
– Исключение форматирования</td><td>С новой страницы</td><td colspan="3"></td></tr>
<tr><td colspan="4">Запрет висячих строк
Не отрывать от следующего
Не разрывать абзац
Запретить автоматический перенос слов</td></tr>
</table>

* – значение 0 пт. необходимо задавать в том случае, если заголовок второго уровня идет непосредственно за заголовком первого уровня и текстовый редактор не поддерживает автоматическую подстройку параметров интервала до/после абзаца.

Нумеровать разделы, подразделы и пункты следует арабскими цифрами. Разделы должны иметь порядковую нумерацию в пределах всего текста, за исключением содержания, введения, заключения, списка использованных источников и приложений, которые не нумеруются. При оформлении заголовков нумеровать и выносить в содержание следует только заголовки до третьего уровня включительно.

После номера раздела, подраздела и пункта точка не ставится. В случае если раздел или подраздел имеет только один пункт, то нумеровать и выносить в содержание его не нужно.

Рисунки. Рисунки размещаются непосредственно после текста, в котором они упоминаются впервые, или на следующей странице. На все рисунки должны быть даны ссылки.

Каждому рисунку или таблице присваивается номер по порядку упоминания в тексте каждого раздела отдельно. Первая цифра означает принадлежность рисунка к разделу, а вторая – к записи. В тексте необходимо делать ссылки на все рисунки и таблицы. Параметры стиля для блока рисунка:

Шрифт: **Liberation Serif**

Размер шрифта – 12 пунктов.

Выравнивание – по центру.

Отступ слева – 0 см.

Интервал до абзаца – 6 пт.

Интервал после абзаца – 4 пт.

Межстрочный интервал – одинарный.

Дополнительно – не добавлять интервал между абзацами одного стиля.

Разбивка на страницы – запрет висячих строк, не отрывать от следующего.

Исключение форматирования – запретить автоматический перенос слов.

Под каждым рисунком должна быть подпись и/или пояснительные данные (подрисуночный текст), отделенные от номера рисунка тире. Параметры стиля для подписи рисунка:

Шрифт: **Liberation Serif**

Размер шрифта – 12 пунктов.

Выравнивание – по центру.

Отступ слева – 0 см.

Интервал до абзаца – 0 пт.

Интервал после абзаца – 16 пт.

Межстрочный интервал – одинарный.

Дополнительно – не добавлять интервал между абзацами одного стиля.

Разбивка на страницы – запрет висячих строк.

Исключение форматирования – запретить автоматический перенос слов.

В случае, когда текст подрисуночной подписи не умещается в одну строку, то его следует разбить исходя из смыслового контекста на несколько строк, используя разрыв строки.

Если рисунок располагается на нескольких страницах (например, диаграмма активности), то на следующих страницах пишется номер рисунка и слово «Продолжение». Например, «Рисунок 2.3 – Диаграмма активности алгоритма», на последующих страницах – «Рисунок 2.3 – Продолжение». В случае если схема/диаграмма активности расположены на нескольких страницах, то на линиях разрыва необходимо проставить нумерацию (в окружности). Экранные формы оформляются как рисунки.

При ссылках на рисунки допускается делать ссылки внутри предложения «...представлено на рисунке 2.1». В случаях если для ссылки на рисунки нет явного поясняющего текста, то следует указывать ссылку в круглых скобках. В этом случае ссылка начинается с прописной буквы, например: «Диаграмма активности алгоритма (Рисунок 2.5) характеризует…».

Иллюстрации каждого приложения обозначают отдельной нумерацией арабскими цифрами с добавлением перед цифрой обозначение приложения. Например: Рисунок Б.3.

Для формирование UML диаграмм и ER-модели возможно использование программы **Software Ideas Modeler** (доступна на сайте https://www.softwareideas.net/). В случае использования данной программы при экспорте изображения необходимо установить белый цвет фона.

Таблицы в пояснительной записке должны быть компактными, иметь нумерацию по принципу рисунков, наименование, названия граф, размерность указываемых величин. Параметры базового стиля шрифта в таблице:

Шрифт: **Liberation Serif**

Размер шрифта данных в таблице – 12 пунктов.

Отступ слева – 0 см.

Интервал до абзаца – 0 пт.

Интервал после абзаца – 0 пт.

Межстрочный интервал – одинарный.

Выравнивание – определяется тем, что содержится в таблице. Так возможны варианты: по центру/по центру по левому краю /сверху по левому краю.

Исключение форматирования – запретить автоматический перенос слов.

Название таблицы следует помещать над таблицей слева и оформлять 12 пунктов без абзацного отступа в одну строку с ее номером и названием через тире, без точки в конце. Например, «Таблица 1.2 – Сравнение программных средств». В тексте производится упоминание в виде (табл. 1.2) или таблица 1.2. При разрыве таблицы на несколько страниц в качестве названия таблицы продолжения пишется, например, «Продолжение таблицы 1.2» и размещается по правому краю. В случае если на новую страницу переносится конечная часть таблицы и таблица размещена более чем на двух страницах, то в названии пишется, например, «Окончание таблицы 1.2».

Параметры стиля названия таблицы:

Шрифт: **Liberation Serif**

Размер шрифта подписи таблицы – 12 пунктов.

Выравнивание – по левому краю (для продолжения – по правому краю)

Отступ слева – 0 см.

Интервал до абзаца – 4 пт.

Интервал после абзаца – 4 пт.

Межстрочный интервал – одинарный.

Дополнительно – не добавлять интервал между абзацами одного стиля.

Разбивка на страницы – запрет висячих строк, не отрывать от следующего, не разрывать абзац.

Исключение форматирования – запретить автоматический перенос слов.

Заголовки столбцов и строк таблицы следует писать с прописной буквы в единственном числе, подзаголовки столбцов – со строчной буквы, если они составляют одно предложение с заголовком, или с прописной буквы, если они имеют самостоятельное значение. В конце заголовков и подзаголовков таблиц точки не ставятся.

Параметры шрифта заголовка столбцов и строк в таблице принимают аналогичные значения базовому стилю и добавляются следующие параметры:

Выравнивание для столбцов – сверху по центру.

Выравнивание для строк – по левому краю.

Разбивка на страницы – запрет висячих строк, не разрывать абзац.

Не допускается разрыв ячеек на страницах. В случае, если таблица расположена на нескольких страницах, то выполняется разрыв таблицы и добавляются идентификаторы заголовков столбцов таблицы на каждой странице в виде числовых обозначений или полного копирования шапки столбцов.

При осуществлении разрыва таблиц, для обозначения того что на следующей странице будет продолжение, следует закрывающую линию таблицы сделать белым цветом:

Таблица 1.1 – Пример первой части

Название 1	Название 2	Название Б

Продолжение таблицы 1.1

Название 1	Название 2	Название Б

После таблицы необходимо оставить пустую строку с одинарным межстрочным интервалом (стиль «Строка после таблицы»). Пункт или раздел не должны заканчиваться таблицей.

Листинги используются в тексте курсовой работы для отображения программного кода на языке программирования. При этом соответствующий блок кода предваряется словом «Листинг» с указанием его порядкового номера с соблюдением общих принципов нумерации (оформляется по аналогии с названием таблицы). Для набора текста программного кода используется следующий стиль «Листинг (фрагмент кода)»:

Шрифт: **Courier New**

Размер шрифта– 12 пунктов.

Выравнивание – по левому краю

Отступ слева – 0 см.

Интервал до абзаца – 4 пт.

Интервал после абзаца – 4 пт.

Межстрочный интервал – одинарный.

Дополнительно – не добавлять интервал между абзацами одного стиля.

В **список использованных источников** допускается включать не только издания, которые были фактически использованы автором, но и названия работ, отвечающих тематике представляемой КР, по которым автор проводил обзор. При написании работы студент обязан давать ссылки на источник, из которого он заимствует материалы. Помимо печатных изданий на русском языке могут использоваться материалы на иностранных языках и электронные ресурсы, при указании которых необходимо соблюдать следующий порядок:

– печатные издания на русском языке;
– печатные издания на иностранных языках;
– электронные ресурсы на русском языке;
– электронные ресурсы на иностранных языках.

Внутри каждого из перечисленных блоков списка использованных источников элементы оформляются в алфавитном порядке по фамилии первого автора либо по названию (при отсутствии авторов).

Ссылка на литературу в тексте пояснительной записки дается в виде цифр в квадратных скобках, где указывается номер книги (журнала) или интернет источника в списке используемых источников и страница для справочного издания.

Если авторов не более трех, то указываются фамилии, имена, отчества авторов, полное название книги или статьи (для статьи наименование журнала и номер журнала), место издания, наименование издательства, год издания, количество страниц.

Если авторов более трех, указываются первые три автора и оборот «и др.». Если книга издана под редакцией авторов, то вначале указывается ее полное библиографическое название. В списке литературы такая книга располагается в алфавитном порядке по своему наименованию.

4 ОСНОВНОЕ СОДЕРЖАНИЕ КУРСОВОЙ РАБОТЫ

Во введении определяется предмет исследования, дается общая характеристика курсовой работы, обоснование актуальности темы, состояние разработки проблемы, о которой пойдет речь в работе, основные цели и задачи, практическая значимость и описания состава курсовой работы. Объем введения ***около 2-3 страниц***.

Основная часть состоит из двух глав. Первая глава, как правило, носит теоретическо-методологический характер. Здесь нужно: проанализировать поставленную задачу (например, выявить область и объект автоматизации); рассмотреть существующие программные средства для решения поставленной задачи и выявить их сильные и слабые стороны; спроектировать базу данных. Содержание второй главы носит практический характер.

Заключение содержит **выводы**, которые должны быть общими по всей работе и отражать действия, выполненные студентом в ходе выполнения курсовой работы. Они должны содержать общие выводы, обобщенное изложение основных проблем, авторскую оценку работы с точки зрения решения задач, поставленных в курсовой работе.

После заключения дается **список использованных источников**. Список организуется в соответствии с требованиями описания произведений печати и электронных изданий. Список использованных источников должен состоять не менее чем из **15** источников.

Приложения помещают после списка использованных источников. Часть материала основных глав, в том числе распечатки фрагментов SQL кода запросов, таблицы, схемы, графики, статистические материалы, графическая часть курсовой работы и т. д. может быть вынесена в приложения.

5. СОДЕРЖАНИЮ РАЗДЕЛА «ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ»

В качестве вводной части приводится абзац про выполняемые действия при проектировании базы данных.

5.1. Обзор (анализ) предметной области

Приводится обзор по предметной области, в рамках которой выполняется курсовая работа, с описанием проблемной ситуации – т.е. текущей ситуации в организации и объектов автоматизации. Для наглядности следует использовать схемы отображения существующих потоков данных, характеризующих работу в исследуемой организации/объекте.

Предметную область можно определить как объект или производственную систему со всем комплексом понятий и знаний о ее функционировании. При исследовании проблемной области необходимы знания о задачах, решаемых в производственной системе, и стоящих перед ней целях. Для наглядности целесообразно использовать схемы отображения существующих потоков данных, характеризующих работу в исследуемой организации/объекте. Отображение процессов, для которых будет осуществляться автоматизация, рекомендуется выполнять с использованием IDEF UML диаграмм. В случае выбора UML диаграмм целесообразно использовать диаграмму потоков.

5.2 Обзор существующего программного обеспечения

Приводится краткий перечень существующего программного обеспечения по предметной области (в зависимости от предметной области 7-***12 наименований***) с описанием решаемых проблемных ситуаций (т.е. с указанием того какие сферы деятельности охватывает программные продукты).

Приводится подробный анализ нескольких существующих систем (**3-5**), наиболее близких к выбранной предметной области, с указанием их функциональных характеристик. Необходимо проанализировать достоинства и недостатки существующих программных продуктов. Желательно построить сравнительную таблицу по функциональным и эксплуатационным показателям существующих программных продуктов.

Информационный поиск основывается на анализе монографий, статей из журналов, реферативных журналов, сообщений с сайтов Internet.

5.3 Концептуальное проектирование базы данных

Концептуальное проектирование базы данных выполняется на основе развернутого технического задания на разработку программного продукта. Начинается с создания концептуальной модели данных, которая является абсолютно независимой от таких деталей реализации, как целевая СУБД, особенности прикладных программ, используемый язык программирования, выбранная вычислительная платформа и т. п.

На данном этапе осуществляется построение моделей деятельности организации. Модель включает в себя структурную функциональную модель деятельности (например, в виде иерархии диаграмм потоков данных), информационную модель (с использованием нотации «сущность – связь»), а

при необходимости событийную (описывающую поведение) модель (с использованием диаграмм переходов состояний).

По результатам анализа деятельности организации следует сформулировать спецификации требований, включающих:

– четкий перечень **требований к данным** (по персоналу, по их обязанностям, по архитектуре организации);

– **требования к транзакциям**, представляющие собой перечень действий, например, по составлению списков, созданию отчетов, поиску объектов, корректировке записей и т. д.

Концептуальная модель данных включает:

– определение типов сущностей;

– определение типов связей;

– предварительное создание диаграммы «сущность – связь».

Сведения о типах сущностей и типах связей следует оформить также в виде таблиц (табл. 5.1 и табл. 5.2).

Таблица 5.1 – Сведения о типах сущностей

№	Имя сущности	Описание	Тип
	. . . . .	. . . . .	. . . . .

Таблица 5.2 –Сведения о типах связей

№	Тип сущности	Тип связи	Тип сущности	Кардинальность
	. . . . .	. . . . .	. . . . .	. . . . .

В данном разделе также приводятся ER-модель и по возможности соответствующая карта транзакций.

5.4 Логическое проектирование базы данных

Логическое проектирование базы данных представляет собой процесс конструирования модели информационной структуры организации, выполняемый в соответствии с выбранной схемой организации информации (например, реляционной). Однако создаваемая логическая модель не зависит от особенностей конкретной СУБД и физических условий реализации.

Действия, необходимые для преобразования концептуальной модели данных в логическую модель данных, включают: удаление связей типа M:N, удаление сложных связей, удаление рекурсивных связей, удаление связей с атрибутами, удаление множественных атрибутов, перепроверка связей типа 1:1 и удаление избыточных связей.

Логическая модель данных может быть проверена с помощью методов нормализации, а также на возможность выполнения всех требуемых транзакций. **Нормализация** используется для общего улучшения характеристик модели, что достигается с помощью введения различных ограничений, позволяющих избежать дублирования данных. Проведение нормализации позволяет получить уверенность в том, что результирующая модель более точно отражает особенности организации, обладает внутренней согласованностью, минимальной избыточностью и максимальной устойчивостью.

Существует два подхода к проверке логической модели на возможность выполнения всех требуемых транзакций:

– исходя из описания каждой транзакции убедиться, что логическая модель позволяет получить всю информацию (сущности, связи и их атрибуты), необходимую для выполнения любой из них;

– непосредственно на ER-диаграммах отобразить все пути доступа к данным, необходимые для выполнения транзакций.

На данном этапе необходимо обосновать **ограничения целостности данных**, представляющие собой такие ограничения, которые вводятся с целью предотвратить помещение в базу данных противоречивых данных. Существует пять типов ограничений целостности: обязательные данные, ограничения для доменов атрибутов, целостность сущностей, ссылочная целостность и требования данной организации (бизнес-правила). Для поддержания ссылочной целостности данных устанавливаются **ограничения на существование**, определяющие условия, при которых потенциальный или внешний ключ может быть вставлен, обновлен или удален.

В пояснительной записке необходимо спроектировать логическую модель, осуществить её проверку с помощью правил нормализации и с помощью транзакций, а также определить требования поддержки целостности данных. Для спроектированной модели привести таблицу с описанием атрибутов и доменов атрибутов.

Таблица 5.3 – Атрибуты и домены атрибутов

№	Сущность	Атрибут сущности	Домен атрибута
	. . . . .	. . . . .	. . . . .
		. . . . .	. . . . .

В случае если логическая модель формируется в программе, где идет обозначение атрибутов с использованием только английского языка, то тогда в структуру таблицы «Атрибуты и домены атрибутов» добавляется колонка «Псевдонимы». При этом в колонке «Сущность» указывается два варианта названия – на русском языке и в скобках на английском. В колонку «Атрибуты сущности» записываются русскоязычные названия атрибутов, а англоязычные в колонку «Псевдонимы». При заполнении таблица «Атрибуты и домены атрибутов» будет иметь следующий вид:

Таблица 5.4 – Атрибуты и домены атрибутов

№	Сущность	Атрибут сущности	Псевдоним атрибута	Домен атрибута
	Русское название (*English name*)	Русское название атрибута	*English name of attribute*	Домен атрибута
		. . . . .	. . . . .	. . . . .

5.5 Физическое проектирование базы данных

В данном разделе приводится краткое обоснование выбора СУБД, в которой будет осуществлена непосредственная реализация базы данных. Помимо этого необходимо пояснить при помощи каких средств будет выполняться физическое проектирование.

Физическое проектирование базы данных представляет собой процесс подготовки описания реализации этой базы во вторичной памяти. Создается описание таблиц базы данных и выбранных для них структур хранения, а также методов доступа, которые будут использоваться для эффективного доступа к данным. Высококачественный проект реализации таблиц базы данных может быть создан только программистами, хорошо знающими все функциональные возможности выбранной целевой СУБД.

Первый этап физического проектирования базы данных состоит в преобразовании логической модели в форму, которая может быть реализована в среде целевой реляционной СУБД. Следующий этап предусматривает выбор структуры файлов и методов доступа, которые будут применены при реализации каждой из таблиц базы данных. Выполнение этого этапа предусматривает проведение анализа транзакций, которые будут выполняться в базе данных. Результаты анализа используются для выбора оптимальной файловой организации таблиц. С целью оптимизации производительности системы могут быть приняты решения о создании вторичных индексов или внесении в данные контролируемой избыточности. Завершается данный этап проведением оценки объема дисковой памяти, необходимой для размещения создаваемой базы данных.

Вторичные индексы представляют собой механизм определения дополнительных ключей для таблиц базы данных, которые могут использоваться для повышения эффективности выборки данных. Однако наличие вторичного индекса создает дополнительную нагрузку при внесении изменений в данные, что должно учитываться при принятии решения о создании этих индексов.

В некоторых случаях может потребоваться отказаться от преимуществ, достигаемых при использовании полностью нормализованных отношений, в пользу обеспечения более высокого уровня производительности системы. Подобные решения следует принимать только тогда, когда удовлетворить требования к производительности системы любыми иными способами невозможно. Практика показывает, что **денормализация** части отношений может оказаться подходящим методом, если уровень производительности системы неудовлетворителен, данные в таблицах базы данных обновляются достаточно редко, а количество выполняемых запросов весьма велико.

База данных представляет собой ценный корпоративный ресурс, поэтому организация ее защиты является одной из важнейших задач разработчиков. Назначение следующего этапа физического проектирования состоит в реализации тех требований к защите данных, которые были установлены на этапе логического проектирования базы данных. Используемые решения могут включать создание пользовательских представлений (просмотров) и организацию механизма контроля над доступом к данным, реализуемого с помощью средств языка SQL.

В пояснительной записке необходимо обосновать выбор целевой СУБД, привести структуру таблиц базы данных (см. табл. 5.4), описать процесс создания базы данных и проектирования таблиц в среде целевой СУБД. Далее

необходимо составить карты выполнения транзакций и проанализировать их выполнение. В результате проведения анализа выполнения транзакций принимается решение о создании вторичных индексов. Анализируется необходимость введения контролируемой избыточности данных (обоснование денормализации). Следует проработать вопросы разработки механизмов защиты данных. При соответствующей тематике рассматривается возможность создания распределенной базы данных. Структура уточненных полей таблиц БД отражается в таблице (табл 5.4), именуется следующим образом – например: Таблица 1.4 – Структура таблиц базы данных. По результатам физического проектирования формируется физическая схема БД, в пояснительной записке её необходимо представить в виде рисунка (изображения).

Таблица 5.4 – Структура таблиц базы данных

Наименование поля	Содержание поля	Тип поля	Размер поля	Значение по умолчанию	Условие на значение	Ключ / индекс
Название таблицы 1 на русском (*English name* 1)						
. . . . .	. . . . .	. . . . .	. . . . .	. . . . .	. . . . .	. . . . .
Название таблицы 2 на русском (*English name* 2)						
. . . . .	. . . . .	. . . . .	. . . . .	. . . . .	. . . . .	. . . . .

6 ТРЕБОВАНИЯ К СОДЕРЖАНИЮ РАЗДЕЛА «РАЗРАБОТКА ПРОГРАММНОГО ПРОДУКТА»

6.1 Структура программного продукта

Приводится подробное описание меню разработанного программного продукта и строится структура взаимодействия форм. Структура меню/вкладок и взаимодействия форм или модулей – оформляется как рисунок (как правило, в виде иерархического дерева). При необходимости строится модульная структура программного продукта с кратким описанием всех функциональных модулей. В случае если в программе используется групповые политики (выделение ролей для пользователей) целесообразно отображения схем меню и взаимодействия форм в зависимости от типа пользователя.

Для описания функциональных возможностей доступных пользователям разрабатываемого программного продукта формируется UML диаграмма прецедентов.

6.2 Реализация бизнес-правил

При наличии нескольких групповых политик необходимо привести их описание, при этом следует привести UML диаграммы прецедентов. Для наиболее значимых бизнес-процессов реализуемых в разрабатываемом приложении базы данных необходимо привести UML диаграммы активности.

Следует привести блок-схемы реализации бизнес-правил, оформленные в соответствии с ГОСТ 19.003-80 (возможно использование UML диаграмм). Каждый блок и каждая связь блок-схемы должны быть упомянуты при

описании хотя бы один раз. Необходимо подробно описать имеющиеся запросы, представить планы их выполнения и обосновать их структуру.

В данном разделе также следует привести структуру просмотров для различных категорий пользователей, рассмотреть вопросы архивации/разархивации и парольной защиты (при наличии таковых в программном продукте). Также необходимо отобразить фрагменты SQL кода (или кода программы) для основных запросов (**3-7 запросов).**

Для конструкторов формирующих запросы в динамическом режиме необходимо раскрыть логику их работы, которую также необходимо представить в виде схемы

6.3 Руководство программиста

Руководство программиста оформляется в соответствии с ГОСТ 19.504-79. Структура и оформление документа устанавливаются в соответствии с ГОСТ 19.105-78. Составление информационной части (аннотации и содержания) является обязательным.

Руководство программиста должно содержать следующие разделы: назначение и условия применения программы; характеристики программы; обращение к программе; описание потоков; входные и выходные данные; сообщения.

В зависимости от особенностей документа допускается объединять отдельные разделы или вводить новые.

В разделе "Назначение и условия применения программы" должны быть указаны назначение и функции, выполняемые программой; условия, необходимые для выполнения программы (объем оперативной памяти, требования к составу и параметрам периферийных устройств, требования к программному обеспечению т.п.).

В разделе "Характеристики программы" должно быть приведено описание основных характеристик и особенностей программы (временные характеристики, режим работы, средства контроля правильности выполнения и самовосстанавливаемости программы и т.п.).

В разделе "Обращение к программе" должно быть приведено описание процедур вызова программы (способы передачи управления и параметров данных и др.).

В разделе "Описание потоков" должно быть приведено описание организации взаимодействия форм с компонентами базы данных и не визуальными модулями программного обеспечения.

В разделе "Входные и выходные данные" должно быть приведено описание организации используемой входной и выходной информации и, при необходимости, ее кодирования.

В разделе "Сообщения" должны быть указаны тексты сообщений, выдаваемых программисту или оператору в ходе выполнения программы, описание их содержания и действия, которые необходимо предпринять по этим сообщениям.

В приложении к руководству программиста могут быть приведены дополнительные материалы (примеры, иллюстрации, таблицы, графики и т.п.).

В руководстве программиста следует указать, какая используется среда разработки, драйверы соединения с базой данных, а также средства проектирования шаблонов отчетов и т.п. При использовании сторонних компонентов и программ, их также необходимо описать.

6.4 Руководство пользователя

Руководство пользователя оформляется в соответствии с ГОСТ 19.505-79. Структура и оформление документа устанавливаются в соответствии с ГОСТ 19.105-78. Составление информационной части (аннотации и содержания) является обязательным. Руководство пользователя должно содержать следующие разделы: назначение программы; условия выполнения программы; выполнение программы.

В зависимости от особенностей документа допускается объединять отдельные разделы или вводить новые.

В разделе "Назначение программы" должны быть указаны сведения о назначении программы и информация, достаточная для понимания функций программы и ее эксплуатации.

В разделе "Условия выполнения программы" должны быть указаны условия, необходимые для выполнения программы (минимальный и (или) максимальный состав аппаратных и программных средств и т.п.).

В разделе "Выполнение программы" должны быть указана последовательность действий оператора, обеспечивающих загрузку, запуск, выполнение и завершение программы, приведено описание функций, формата и возможных вариантов команд, с помощью которых оператор осуществляет загрузку и управляет выполнением программы, а также ответы программы на эти команды.

Допускается содержание разделов иллюстрировать поясняющими примерами, таблицами, схемами, графиками. В приложения к руководству оператора допускается включать различные материалы, которые нецелесообразно включать в разделы руководства.

6.5 Тестирование программного продукта

Необходимо привести сведения о:

– работоспособности программного продукта;

– возможности выполнения всех заданных в техническом задании транзакций;

– корректности реализации бизнес-правил;

– поведении программного продукта при ошибочных действиях пользователя (неправильный ввод данных, внештатное прерывание работы программного продукта, попытка обращения к недоступным для него данным и т. п.).

Тестирование должно проводиться при наличии достаточно большого объема информации в базе данных. Результаты тестирования оформлять в виде таблицы имеющей следующую структуру (табл. 6.1).

Таблица 6.1 – Результаты тестирования

№	Цель испытания	Объект	Результат
	. . . . .	. . . . .	. . . . .

7 ТРЕБОВАНИЯ К СОДЕРЖАНИЮ ПРИЛОЖЕНИЯ «ТЕХНИЧЕСКОЕ ЗАДАНИЕ НА РАЗРАБОТКУ ПРОГРАММНОГО ПРОДУКТА»

Техническое задание на разработку программного продукта выполняется на основе задания на курсовую работу, выданного преподавателем, и анализа деятельности организации, включающего следующие основные положения:

– выявление требований, предъявляемых к будущей системе;

– определение перечня целевых задач организации;

– определение потоков документооборота;

– анализ распределения функций по подразделениям и сотрудникам;

– определение перечня применяемых средств автоматизации.

В качестве исходных документов служит нормативно-справочная документация, документы отчетности, приказы, распоряжения и т. п.

Развернутое техническое задание согласовывается с руководителем курсовой работы через 1–2 недели после выдачи задания на курсовое проектирование.

Техническое задание оформляется в соответствии с ГОСТ 19.201-78. Он устанавливает процесс построения и оформления технического задания на разработку программы или программного изделия для вычислительных машин, комплексов и систем независимо от их назначения и области применения.

Техническое задание должно содержать следующие разделы: введение; основания для разработки; назначение разработки; требования к программе или программному изделию; стадии и этапы разработки; в техническое задание допускается включать приложения.

В зависимости от особенностей программы или программного изделия допускается уточнять содержание разделов, вводить новые разделы или объединять отдельные из них.

В разделе "Введение" указывают наименование, краткую характеристику области применения программы или программного изделия и объекта, в котором используют программу или программное изделие.

В разделе "Основания для разработки" должны быть указаны: документ (документы), на основании которых ведется разработка; организация, утвердившая этот документ, и дата его утверждения; наименование и (или) условное обозначение темы разработки.

В разделе "Назначение разработки" должно быть указано функциональное и эксплуатационное назначение программы или программного изделия.

Раздел "Требования к программе или программному изделию" должен содержать следующие подразделы: требования к функциональным характеристикам; требования к надежности; условия эксплуатации; требования к составу и параметрам технических средств; требования к информационной и программной совместимости; специальные требования.

В подразделе "Требования к функциональным характеристикам" должны быть указаны требования к составу выполняемых функций, организации входных и выходных данных, временным характеристикам и т.д.

В подразделе "Требования к надежности" должны быть указаны требования к обеспечению надежного функционирования (обеспечение устойчивого функционирования, контроль входной и выходной информации, время восстановления после отказа и т.п.).

В подразделе "Условия эксплуатации" должны быть указаны условия эксплуатации, при которых должны обеспечиваться заданные характеристики, а также вид обслуживания, необходимое количество и квалификация персонала.

В подразделе "Требования к составу и параметрам технических средств" указывают необходимый состав технических средств с указанием их основных технических характеристик.

В подразделе "Требования к информационной и программной совместимости" должны быть указаны требования к информационным структурам на входе и выходе и методам решения, исходным кодам, языкам программирования и программным средствам, используемым программой. При необходимости должны обеспечиваться защита информации и программ.

В разделе "Стадии и этапы разработки" устанавливают необходимые стадии разработки, этапы и содержание работ (перечень программных документов, которые должны быть разработаны, согласованы и утверждены), а также, как правило, сроки разработки и определяют исполнителей.

В приложениях к техническому заданию, при необходимости, приводят: перечень научно-исследовательских и других работ, обосновывающих разработку; схемы алгоритмов, таблицы, описания, обоснования, расчеты и другие документы, которые могут быть использованы при разработке; другие источники разработки.

Пример структуры и оформления технического задания на разработку программного продукта представлен в **приложении 6**.

8 ПРИМЕРНЫЙ ПЕРЕЧЕНЬ ТЕМ КУРСОВЫХ РАБОТ

Приведем примерный перечень тем курсовых работ по дисциплине «Управление данными»:

1. Разработка АРМ сервисной службы компании **X.**
2. Разработка приложения базы данных «Отдел кадров предприятия».
3. Приложение для учета посещаемости **X** «Журнал посещений **X**».
4. Разработка приложения базы данных «Бухгалтерия предприятия».
5. Проектирование информационной системы отдела повышения квалификации.
6. Система складского учета предприятия X.
7. АРМ начальника вычислительного центра.
8. Разработка приложения базы данных «Учебный процесс».
9. Проектирование базы данных контрольно-пропускного пункта предприятия **X.**
10. Программная система учета успеваемости студентов.
11. Разработка информационно-справочной системы «Библиотека».
12. Система автоматизации проведения инвентаризации оборудования в организации **X.**
13. Приложение базы данных «Отдел снабжения».
14. Разработка АРМ оператора энергосбыта.
15. Разработка приложения базы данных «Городской отдел образования».
16. Информационно-справочная система спортивного клуба **X.**
17. Разработка приложения базы данных «Банковская деятельность».
18. Проектирование информационной системы агентства недвижимости.
19. Приложение для ведения учета заказов транспортной компании.
20. Проектирование базы данных туристического агентства **X.**
21. Система учета выполнения работ сервисной службы **X.**
22. Проектирование базы данных «Оптовая база».
23. Информационно-справочная система «Магазин **X**».
24. Система автоматизации оформления продажи авиабилетов.
25. Разработка информационной системы маршрутов железнодорожного транспорта.
26. Система учета движения транспорта в Таксопарке.
27. Разработка приложения базы данных «Типография».
28. Разработка информационно-справочной системы спорткомплекса **X.**
29. Приложения базы данных «Перепись населения».
30. АРМ заведующего аспирантурой.
31. Приложения базы данных «Кинотеатр».
32. Система ведения учета посещения занятий студентами «Журнал группы».
33. Проектирование базы данных каталогизатора медиа файлов.
34. Разработка приложения базы данных «Пенсионный фонд».
35. АРМ заведующего кафедрой.
36. Разработка приложения базы данных «Страховая компания».

37. Каталогизатор медиафайлов домашней коллекции **X.**
38. Разработка приложения базы данных «Налоговая инспекция».
39. АРМ библиотекаря университетской библиотеки.
40. Информационная система рекламного агентства.
41. Система информационного сопровождения клиентов автосервиса.
42. Разработка системы автоматизации документооборота службы ремонта.
43. Разработка приложения базы данных «Охранное агентство».
44. Проектирование базы данных компьютерного салона.
45. Разработка приложения для бронирования номеров в отеле.
46. Приложение базы данных для музыкальной студии.
47. Разработка приложения базы данных «Спортивный комплекс».
48. Разработка информационно-справочной системы по налогообложению.
49. Разработка юридической информационно-справочной системы по законодательству.
50. Разработка информационно-справочной системы по охране труда.
51. АРМ менеджера по продажам магазина **X.**
52. Разработка АРМ сотрудника стоматологической клиники.
53. Разработка АРМ оператора банка.
54. Проектирование АРМ оператора ТСЖ (УК).
55. Разработка приложения базы данных медиа оператора **X.**
56. Информационная система учёта объектов недвижимости агентства **X.**
57. Разработка АРМ сотрудника отдела логистики компании X.
58. База данных салона видеопроката **X.**
59. Автоматизированная система учёта спортивных мероприятий по **T**
60. Информационно-справочная система по предприятиям города

Приложение 1

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
**«Сибирский государственный университет науки и технологий
имени академика М.Ф. Решетнева»**

Институт информатики и телекоммуникаций

Кафедра информатики и вычислительной техники

КУРСОВАЯ РАБОТА

Управление данными

Первая строка названия темы

вторая строка названия темы

Руководитель _____ _____ А.Г. Зотин
подпись, дата инициалы, фамилия

Обучающийся БИС **-01, **1239*** _____ И.И. Иванов
номер группы, зачетной книжки подпись, дата инициалы, фамилия

Красноярск 20** г.

Приложение 2

Институт информатики и телекоммуникаций

Кафедра информатики и вычислительной техники

ЗАДАНИЕ

на курсовую работу по дисциплине *Управление данными*

обучающемуся *Иванову Ивану Ивановичу*

Группа *БИС **-01* Форма обучения *очная*

Тема работы: *Первая строка названия темы*

вторая строка названия темы (строчки темы заполняются отдельно)

Срок сдачи курсовой работы ***.06.20***

Перечень вопросов, подлежащих разработке при написании теоретической части:

*Анализ процессов *** и выявление аспектов автоматизации; обзор программных продуктов для***; концептуальное проектирование ***; логическое проектирование базы данных; разработка бизнес-правил.*

приведен пример заполнения, в зависимости от темы формулировки будут меняться

Перечень вопросов, подлежащих разработке при написании практической части:

*Физическое проектирование базы данных; разработка структуры **; реализация бизнес-правил; разработка **** ; ****

*разработка отчетов и графиков; формирование руководств программиста и пользователя; тестирование разрабонно** ****

приведен пример заполнения, в зависимости от темы формулировки будут меняться

Дата выдачи задания: ***.02.20***

Руководитель *Зотин А.Г., доцент кафедры ИВТ*

(подпись)

Задание принял к исполнению (дата) _____

(подпись обучающегося, И.О. Фамилия)

Приложение 3

КРАТКОЕ ПОЯСНЕНИЕ СОДЕРЖАНИЯ

Титульный лист

Задание на курсовой проект

Бланк заполняется от руки

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Краткое введение в предметную область, постановка цели и задач, описание состава курсовой работы

1. ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ

Вводный абзац или два про выполняемые действия при проектировании БД

1.1. Обзор предметной области

Приводится обзор предметной области с описанием проблемной ситуации (для наглядности следует использовать схемы отображения существующих потоков данных, характеризующих работу в исследуемой организации/объекте) и выявляются группы пользователей разрабатываемого продукта.

1.2. Обзор существующего программного обеспечения

Рассмотреть минимум 3 программы, выполняющие схожие действия согласно поставленной цели. Отразить функциональные возможности желательно выявить недостатки и т.п.

1.3 Концептуальное проектирование базы данных

Краткое описание сути выполняемого концептуального проектирования и заполнений таблиц 1, 2 и создание ER-модели

1.4 Логическое проектирование базы данных

Описание логического проектирования, заполнений таблицы 3 на основе выбранной модели, формирование логической модели

1.5 Физическое проектирование базы данных

Описание процесса физического проектирования БД и формирование таблицы с описанием сущностей в зависимости от выбранной СУБД (пример табл. 4)

2. РАЗРАБОТКА ПРОГРАММНОГО ПРОДУКТА

2.1 Структура программного продукта

Описать структуру программы (взаимодействия форм модулей, организации меню и т.п.)

2.2 Реализация бизнес-правил

Дать теоретические сведения и описать наиболее значимые бизнес-правила (до 5-ти основных правил, в зависимости от сложности процесса). Для пользователей системы привести UML диаграмму прецедентов

2.3 Руководство программиста

Привести краткое описание средств используемых для разработки ПО и необходимые для его компиляции, а также характеристики программы. Отобразить описание входных/выходных данных, и описание потоков по организации взаимодействия с БД.

2.4 Руководство пользователя

2.5 Тестирование программного продукта

ЗАКЛЮЧЕНИЕ

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

ПРИЛОЖЕНИЕ А Техническое задание на разработку

ПРИЛОЖЕНИЕ Б Образцы документации

Содержит бланки и формы, используемые в предметной области (которые были взяты за основу при проектировании базы данных)

ПРИЛОЖЕНИЕ В Примеры отчетов и графиков генерируемые программой

Содержит примеры отчетов и графиков, сгенерированных разработанным программным продуктом

Приложение 4

ПРИМЕРЫ ЗАПОЛНЕНИЯ ТАБЛИЦ И ОФОРМЛЕНИЯ МОДЕЛЕЙ

Таблица 1.1 – Сведения о типах сущностей проекта X

№	Имя сущности	Описание	Тип
1	Товары	Содержит описание товаров	слабый
2	Виды товара	Содержит перечень возможных типов для товаров	сильный
3	Продажи	Содержит информацию о совершаемых сделках по продажи товара	слабый
4	Сотрудники	Содержит информацию о сотрудниках организации	сильный
5	Способы оплаты	Содержит описание способов оплаты	сильный

Таблица 1.2 – Сведения о типах связей проекта X

№	Тип сущности	Тип связи	Тип сущности	Кардинальность
1	Товары	Бывают	Типы товара	M:1
2	Продажи	Учитывают	Товары	M:N
3	Сотрудники	Оформляют	Продажи	1:M
4	Продажи	Оплачиваются	Способы оплаты	M:1

Рисунок 1.1 – Пример ER-модели проекта X

Примеры заполнения таблиц в зависимости от формы логической модели: русскоязычная (табл. 1.3) или англоязычная (табл. 1.4)

Таблица 1.3 – Атрибуты и домены атрибутов (для русскоязычного варианта)

№	Сущность	Атрибут сущности	Домен атрибута
1	Товары	ID Товара	Числовой
		Название Товара	Текстовый
		ID Типа товара	Числовой
		Цена	Числовой
2	Типы товара	ID Типа товара	Числовой
		Название типа товара	Текстовый
3	Продажи	ID Продажа	Числовой
		Дата	Дата
		ID Сотрудника	Числовой
		ID Способа оплаты	Числовой
4	Составы продаж	ID Состава продажи	Числовой
		ID Продажа	Числовой
		ID Товара	Числовой
		Количество	Числовой

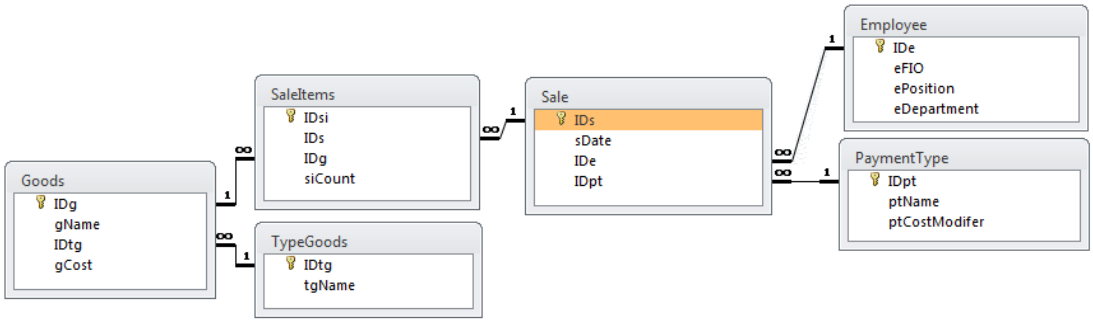
Продолжение таблицы 1.3

№	Сущность	Атрибут сущности	Домен атрибута
5	Сотрудники	ID Сотрудника	Числовой
		ФИО	Текстовый
		Должность	Текстовый
		Отдел торговли	Текстовый
6	Способы оплаты	ID Способа оплаты	Числовой
		Название способа оплаты	Текстовый
		Скидка или наценка	Числовой

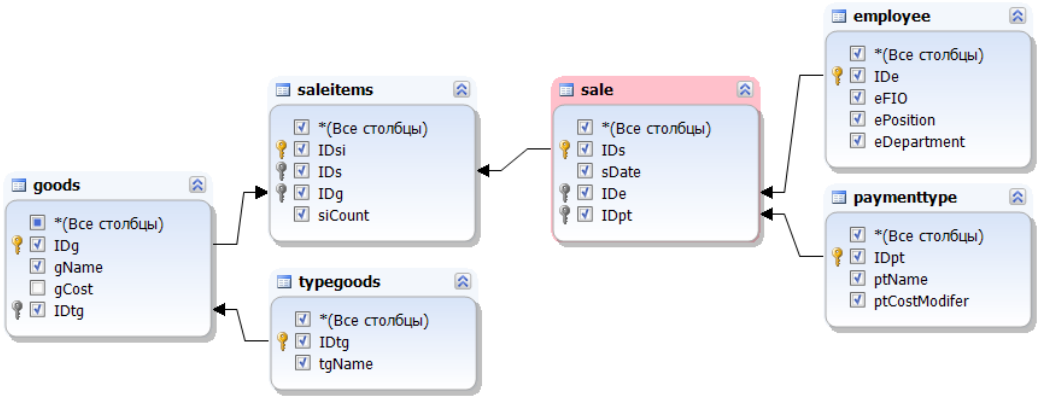
Рисунок 1.2 – Пример логической модели (русскоязычный вариант)

Таблица 1.4 – Атрибуты и домены атрибутов (для англоязычного варианта)

№	Сущность	Атрибут сущности	Псевдоним атрибута	Домен атрибута
1	Товары (*Goods*)	ID Товара	*IDg*	Числовой
		Название Товара	*gName*	Текстовый
		ID Типа товара	*IDtg*	Числовой
		Цена	*gCost*	Числовой
2	Типы товара (*TypeGoods*)	ID Типа товара	*IDtg*	Числовой
		Название типа товара	*tgName*	Текстовый
3	Продажи (*Sale*)	ID Продажа	*IDs*	Числовой
		Дата	*sDate*	Дата
		ID Сотрудника	*IDe*	Числовой
		ID Способа оплаты	*IDpt*	Числовой
4	Составы продаж (*SaleItems*)	ID Состава продажи	*IDsi*	Числовой
		ID Продажа	*IDs*	Числовой
		ID Товара	*IDg*	Числовой
		Количество	*siCount*	Числовой
5	Сотрудники (*Employee*)	ID Сотрудника	*IDe*	Числовой
		ФИО	*eFIO*	Текстовый
		Должность	*ePosition*	Текстовый
		Отдел торговли	*eDepartment*	Текстовый
6	Способы оплаты (*PaymentType*)	ID Способа оплаты	*IDpt*	Числовой
		Название способа оплаты	*ptName*	Текстовый
		Скидка или наценка	*ptCostModifer*	Числовой



а



б

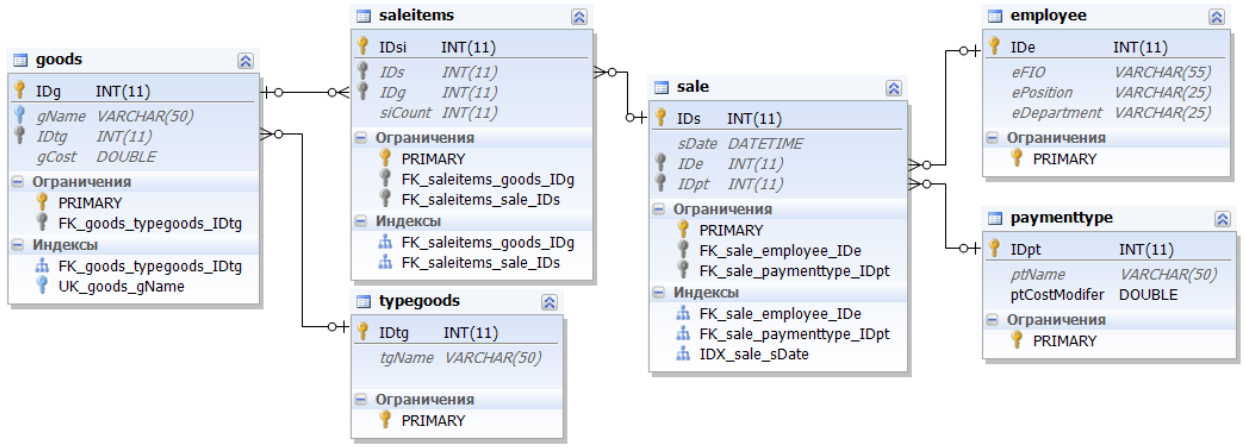
Рисунок 1.3 – Пример логической модели (англоязычный вариант)
а) сделано в *Microsoft Access*; б) сделано *в dbForge Studio for MySQL*.

Таблица 1.6 – Пример заполнения таблицы «Структура таблиц базы данных» **MySQL**

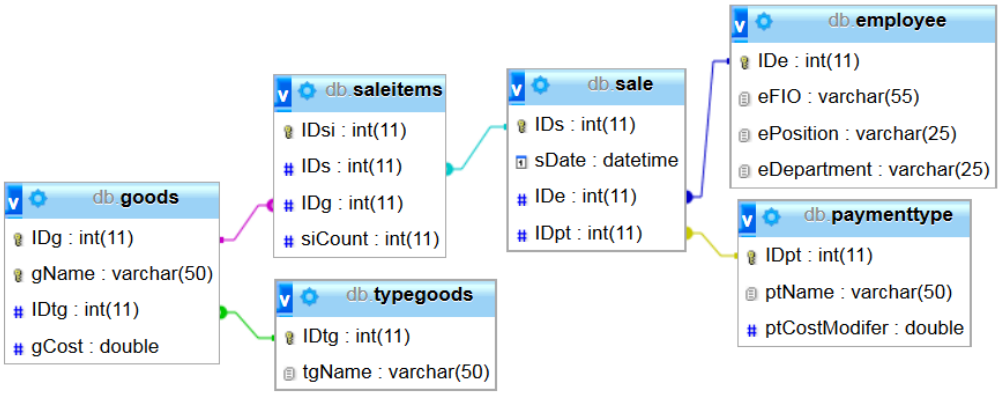
Наименование поля	Содержание поля	Тип поля	Размер поля	Значение по умолчанию	Условие на значение	Ключ / индекс
Товары (*Goods*)						
IDg	Идентификатор товара	*INT* (Автоинк.)	11			ПК, Индекс
gName	Название товара	*VARCHAR*	50		Уникальное NOT NULL	Индекс
IDtg	Идентификатор типа товара	*INT*	11			ВК, Индекс
gCost	Цена товара	*DOUBLE*		10,0	>=10	
Типы товара (*TypeGoods*)						
IDtg	Идентификатор типа товара	*INT* (Автоинк.)	11			ПК, Индекс
tgName	Название типа товара	*VARCHAR*	50			

В столбце «Ключ / индекс» используются следующие обозначения:

- ПК – Первичный ключ
- ВК – Внешний ключ
- Индекс – Для поля установлен Индекс (один индекс может быть для нескольких полей)



а)



б)

Рисунок 1.4 – Пример физической модели проекта X для СУБД *MySQL* 5.5 а) *сделано в dbForge Studio for MySQL*; б) *сделано в phpMyAdmin*

Приложение 5

ПРИМЕР СТРУКТУРЫ И ОФОРМЛЕНИЯ ВВЕДЕНИЯ КУРСОВОЙ РАБОТЫ

ВВЕДЕНИЕ

Актуальность. На сегодняшний день … .

…

Таким образом, … является актуальной задачей.

Цель и задачи. Целью курсовой работы является … .

Для достижения поставленной цели необходимо решить следующие задачи:

– выполнить анализ *** (согласно предметной области);

– определить *** автоматизации и группы пользователей;

– осуществить обзор программного обеспечения в рассматриваемой предметной области;

– ……

– ……

– ……

– ……

– ……

– ……

– ……

осуществить программную реализацию …;

провести тестирование разработанной ***.

Структура работы. Пояснительная записка к курсовой работе состоит из введения, двух глав, заключения и списка использованных источников из … наименований. Изложена на … страницах и содержит … рисунков и … таблиц.

В первой главе курсовой работы приводится … , отражен … .

Во второй главе … .

В заключении … .

*Слова, написанные курсивом во введении, и расположенные в начале абзаца должны присутствовать.

Приложение 6

ПРИМЕР СТРУКТУРЫ И ЧАСТИЧНОГО НАПОЛНЕНИЯ ТЕХНИЧЕСКОГО ЗАДАНИЯ

ПРИЛОЖЕНИЕ А

ТЕХНИЧЕСКОЕ ЗАДАНИЕ НА РАЗРАБОТКУ

Введение

В настоящее время … . В связи с этим разработка … .

Общие положения.

Полное наименование: *****.

Краткое наименование: ****.

Вид разработки: ******.

Разработчик: *****.

Основание для разработки

Разработка программного продукта ведется на основании учебного плана СибГУ им. М.Ф. Решетнева по направлению подготовки 09.03.02 Информационные системы и технологии, направленности «Информационные системы и технологии в управлении».

Плановые сроки начала и окончания выполнения работы

Плановый срок начала работ – **.02.20**

Плановый срок окончания работ – **.06.20**

Назначение разработки

Разрабатываемый программный продукт предназначен для … .

Требования к программе

Требования к функциональным характеристикам

В данном блоке раскрываются функциональные требования, ниже приведены пример фрагментов написания требований. Также можно требования писать обычным текстом, но в таком ключе, чтобы было понятно какие функциональные возможности будут у разработки

В программе должно быть предусмотрено …

Программа должна:

– обеспечить пользователю возможность … ;
– позволять … ;
– осуществлять … ;
– … ;
– … .

Помимо этого для соответствия указанному назначению, состав выполняемых функций должен включать следующее:

– формирование … ;
– генерация … ;
– … .

Ввод данных должен выполняться в программе … . Вывод информации должен выполняться в виде … .

Создаваемые отчеты в программе должны предоставлять следующую информацию:

– … ;
– … ;
– … .

Требования к надежности

В программе должен быть обеспечен контроль за … . Возникновение каких-либо внутренних ошибок не должно приводить к утере данных. Программа должна быть … .

Для защиты от несанкционированного использования должно быть предусмотрено …

Условия эксплуатации

Приложение рассчитано на эксплуатацию пользователями … .

Требования к составу и параметрам технических средств

– процессор ** с частотой не ниже … ГГц.;
– оперативная память не меньше … Гб.;
– не менее … Мб свободного места на жестком диске;
– … .

Требования к информационной и программной совместимости

– операционная система Windows 7/8/10;
– … .

Состав и содержание работ по выполнению курсовой работы

Таблица А.1 – Календарный план-график выполнения стадий и этапов разработки

Наименование работ	Сроки выполнения
Анализ *** предметной области	
Обзор … программного обеспечения …	
Концептуальное проектирование базы данных	
Разработка бизнес-правил	
Логическое проектирование базы данных	
Выбор целевой СУБД	
Физическое проектирование базы данных	
Разработка структуры …	
***	
***	
***	
Разработка пользовательского интерфейса	
Программная реализация ***	
Разработка отчетов и графиков	
Тестирование и отладка ***	
Оформление пояснительной записки	
Предоставление преподавателю всех необходимых для защиты материалов	**.06.202*