УЧРЕЖДЕНИЕ ОБРАЗОВАНИЯ

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»

ФИЛИАЛ «МИНСКИЙ РАДИОТЕХНИЧЕСКИЙ КОЛЛЕДЖ»

**УТВЕРЖДаю**

Директор МРК

\_\_\_\_\_\_\_\_\_\_\_\_\_ С.Н. Анкуда

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

Регистрационный №\_\_\_\_\_\_\_\_

**БАЗЫ ДАННЫХ И СИСТЕМЫ УПРАВЛЕНИЯ БАЗАМИ ДАННЫХ**

**МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ**

**по выполнению курсового проекта**

для учащихся специальности

2 40 01 01 «Программное обеспечение информационных технологий»

Минск 2020 г.

**СОСТАВИТЕЛИ:**

Е А. Лазицкас, преподаватель высшей категории дисциплин общепрофессионального и специального циклов,

О.Н. Виничук, преподаватель первой категории дисциплин общепрофессионального и специального циклов

**РЕЦЕНЗЕНТ:**

И.Г. Смолер, преподаватель первой категории дисциплин общепрофессионального и специального циклов

**РЕКОМЕНДОВАНЫ К УТВЕРЖДЕНИЮ:**

Цикловой комиссией «Программное обеспечение информационных технологий» филиала БГУИР «Минский радиотехнический колледж»

Протокол № 6 от 31.01.2020

Заседанием педагогического совета филиала БГУИР «Минский радиотехнический колледж»

Протокол № \_\_\_\_\_от \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Методическая экспертиза \_\_\_\_\_\_\_\_\_\_\_ Т.С. Власенко

**Предисловие**

Написание курсового проекта является завершающим этапом не только изучения дисциплины «Базы данных и системы управления базами данных» (БД и СУБД), но и всего обучения, а также представляет собой синтез знаний по основным специальным дисциплинам.

Курсовой проект – это самостоятельно написанный законченный проект в области БД и СУБД, содержащая совокупность результатов и положений, выдвигаемых учащихся для публичной защиты, имеющая внутреннее единство, что свидетельствует о возможности автора решать конкретные прикладные задачи в процессе разработки приложений баз данных.

Данное методическое пособие написано для учащихся специальности 2-40 01 01 «Программное обеспечение информационных технологий». В методическом пособии описаны цели курсового проектирования, общие требования, методические указания к выполнению курсового проекта и составу пояснительной записки.

**1 Цели курсового проектирования**

Курсовое проектирование имеет следующие цели:

* закрепление, углубление и систематизация теоретических знаний и практических умений, полученных при изучении дисциплины БД и СУБД;
* привитие навыков самостоятельной разработки БД на SQL-сервере;
* закрепление навыков разработки удобного интерфейса пользователя, системы меню, навигации, системы помощи.

**2 Общие требования к курсовому проекту**

**2.1 Тематика курсового проектирования**

Тема проекта должна соответствовать содержанию учебной программы. Выбор темы осуществляется учащимся и преподавателем-руководителем исходя из интересов и личной склонности к определенному виду профессиональной деятельности.

## Примерный перечень тем курсовых проектов

1. Автоматизация рабочего места диспетчера железнодорожного вокзала
2. Автоматизация учета услуг, оказываемых населению жилищно-эксплуатационной службой
3. Автоматизированное рабочее место сотрудника кинотеатра
4. Автоматизация учета товарооборота книжного магазина
5. Автоматизированная кулинарная книга
6. Автоматизация рабочего места менеджера салона мебели
7. Автоматизация рабочего места менеджера гостиницы
8. Автоматизация рабочего места менеджера салона сотовой связи
9. Автоматизация рабочего места менеджера модельного агентства
10. Автоматизация учета сотрудников предприятия
11. Автоматизация рабочего места менеджера салона красоты
12. Автоматизация учета подписок на периодические издания
13. Интернет-магазин по продаже бытовой техники
14. Автоматизация учета операций пункта обмена валют
15. Автоматизация составления расписания
16. Автоматизация рабочего места менеджера ресторана
17. Автоматизация рабочего места заведующего склада
18. Автоматизация учета спортивных показателей
19. Автоматизация рабочего места сотрудника регистратуры
20. Автоматизация рабочего места менеджера туристического агентства
21. Автоматизация учета посещений занятий учащимися школы
22. Автоматизация сбора и хранения информации о спортивных соревнованиях
23. Автоматизация учета перемещения книг в библиотеке
24. Автоматизированная система по начислению коммунальных платежей
25. Автоматизированная информационная система строительных организаций

**2.2 Задание на курсовое проектирование**

Учащиеся получают индивидуальное задание на курсовой проект и заполняют специальный бланк с исходными данными.

В листе задания необходимо указать наименование дисциплины и учебного заведения, сведения о разработчике. Далее формулируется тема курсового проекта и исходные данные. Исходные данные – это данные, которые необходимы для функционирования БД, ими будет заполняться БД (например: сведения о сотрудниках для отдела кадров, сведения о моделях телефонов для менеджера по продажам сотовых телефонов и т. д.).

Заданием оговаривается содержание пояснительной записки и графической части курсового проекта. Перечень разделов и подразделов пояснительной записки должен соответствовать последовательности, указанной в задании. В графической части приводится перечень диаграмм, которые будут спроектированы в рамках курсового проекта.

Задание должно быть утверждено председателем цикловой комиссии и подписано руководителем проекта.

**2.3 Содержание курсового проекта**

Курсовой проект должен содержать:

* пояснительную записку с приложениями;
* графическую часть (не менее 2 листов формата А1);
* рабочее приложение на цифровом носителе.

Пояснительная записка (ПЗ) выполняется , строго по индивидуальному заданию, выданному преподавателем-руководителем курсового проекта. Объем ПЗ должен быть не более 30–40 листов формата А4 машинописного (компьютерного) текста, за исключением таблиц, рисунков, графиков. В пояснительной записке предусмотрено наличие приложений, в которые могут быть вынесены: схема базы данных, экранные формы, фрагменты кода, алгоритм функционирования системы. Вспомогательный материал (программы, диаграммы и т. п.) включается в работу в качестве приложений.

Графическая часть курсового проекта предполагает наличие диаграмм и блок-схем представленных на листах формата А1. Пример оформления диаграмм приведен в приложении Д. Пример оформления блок-схем приведен в приложении Е.

Рекомендуется следующая структура ПЗ курсового проекта:

Пояснительная записка должна содержать следующие части:

* титульный лист;
* задание на курсовой проект;
* содержание;
* введение;
* основной текст включающий:

1 постановку задачи, которая состоит из аналитического обзора существующих аналогов и описания функционального назначения (5-8 стр);

2 проектирование задачи – это модели, положенные в основу проектирования реляционных баз данных (проекта) и (или) теоретические исследования, а также разработка алгоритма, схемы и структуры программы, схемы работы системы, схемы данных, схемы взаимодействия программ, схемы ресурсов системы (согласно ГОСТ 19.701-90) (8-12 стр);

3 программная реализация задачи, описание разработанных программных модулей (8-12 стр):

4 тестирование, экспериментальные исследования и анализ полученных результатов (6-8 стр);

5 руководство пользователя (6-8 стр);

* заключение;
* список литературы;
* приложения;

Указанную последовательность элементов пояснительной записки рекомендуется принять за порядок размещения частей пояснительной записки.

**3 Методические указания по выполнению курсового проекта**

Курсовой проект представляет собой нормализованную БД и приложение, работающее с БД. Приложение необходимо разработать таким образом, чтобы максимально облегчить работу пользователя с БД. Информация, представленная на формах приложения, должна быть понятна и не носить двусмысленный характер. Все формы оснащаются навигацией и кнопками поиска необходимой информации. Приложение предполагает наличие отчетов, которые позволяют выводить информацию на печать (различные договоры, прайс-листы, чеки покупки или продажи и т. д.). Система помощи является обязательным компонентом для разработанного приложения. Она должна вызываться с любого места приложения, а также легко и наглядно разрешать проблемы пользователя по эксплуатации приложения. Чтобы разгрузить приложение от большого количества кнопок перехода по формам, должна быть организована удобная система меню. Все приложение необходимо организовать в одном стиле и цветовой гамме, приятной для восприятия пользователя.

**В основном тексте пояснительной записки** анализируют существующие решения, определяют пути достижения цели проектирования, составляют технические требования, на основании которых разрабатывают конкретные методики проектирования базы данных, принимают схемотехнические, алгоритмические, программные и конструктивно-технологические решения.

Общими требованиями к основному тексту пояснительной записки являются: четкость и логическая последовательность изложения материала, убедительность аргументации, краткость и ясность формулировок, исключающие неоднозначность толкования, конкретность изложения результатов, доказательств и выводов.

Запрещается переписывание общих сведений из учебников, учебных пособий, монографий, статей и других источников без соответствующей ссылки.

**3.1 Титульный лист**

Пример оформления титульного листа курсового проекта приведен в приложении А.

Тема курсового проекта должна быть сформулирована кратко и точно соответствовать его содержанию. Следует избегать тем, в которых не отражаются в должной мере суть, завершенность работы, нет достаточно ясного определения ее цели и результатов.

**3.2 Лист задания (Бланк индивидуального задания)**

Пример оформления бланка индивидуального задания приведен в приложении Б.

**3.3 Содержание**

Включает в себя заголовки всех частей пояснительной записки, в том числе заголовки разделов и подразделов, приложений.

Расположение заголовков должно точно отражать последовательность и соподчиненность заголовков в пояснительной записке (приложение В).

**3.4 Введение**

Начинают писать на отдельной странице. Оно должно быть кратким и четким. Во введении не должно быть общих мест и отступлений, непосредственно не связанных с разрабатываемой темой. Объем введения должен быть не более двух страниц.

Рекомендуется следующее содержание введения:

* постановка темы курсового проекта;
* краткий анализ достижений в той области, которой посвящена тема курсового проекта;
* цель курсового проектирования;
* задачи курсового проектирования;
* принципы, положенные в основу проектирования, научного исследования, поиска технического решения;
* краткое изложение содержания разделов пояснительной записки с обязательным указанием задач, решению которых они посвящены.

**3.5 Постановка задачи**

Данный раздел должен содержать сравнительный анализ теоретических и эвристических методов решения поставленных задач и существующих по данной тематике технических научных решений (систем, проектов), а также примеры отечественных и зарубежных аналогов. В данном разделе необходимо отразить функциональное назначение курсового проекта.

Раздел должен заканчиваться четкой постановкой общей задачи на курсовое проектирование и формулировкой конкретных детализированных задач, решаемых в курсовом проекте.

**3.5.1 Описание предметной области**

В данном разделе необходимо описать предметную область разрабатываемого приложения.

В широком смысле предметная область проекта представляет собой абстрактную область соединения множества предметов — понятий и материальных объектов в своем единстве составляющих объект управления проектом.

Предметная область – это часть реального мира, данные о которой необходимо отразить в БД. Предметная область определяется темой курсового проекта. Например, в качестве предметной области можно выбрать бухгалтерию какого-либо предприятия, отдел кадров, банк, магазин и т. д.

Необходимо четко выделять в предметной области важные понятия и малозначащие или вообще не значащие данные. Так, если в качестве предметной области выбран учет товаров на складе, то понятия «накладная» и «счет-фактура» являются важными, а информация о наличии у сотрудницы, принимающей накладные, двоих детей для учета товаров неважна. Однако, с точки зрения отдела кадров, данные о наличии детей являются важными. Таким образом, важность данных зависит от выбора предметной области.

**3.5.2 Обзор существующих аналогов**

В данном разделе пояснительной записки необходимо провести сравнительный анализ существующих решений (минимум трех), описать их достоинства и недостатки. Но основании проанализированных аналогов необходимо сформулировать достоинства и недостатки разрабатываемого приложения.

**3.5.3 Функциональное назначение**

В данном разделе необходимо описать функциональное назначение курсового проекта, а именно постановка цели, описание задач для достижения поставленной цели, перечисление функций разрабатываемого приложения.

**3.6 Проектирование реляционных баз данных**

Процесс проектирования включает в себя следующие этапы.

Концептуальное проектирование – это процедура конструирования информационной модели, не зависящей от каких-либо физических условий реализации.

Логическое проектирование – это процесс конструирования информационной модели на основе существующих моделей данных, не зависимо от используемой СУБД и других условий физической реализации.

Физическое проектирование – это процедура создания описания конкретной реализации БД с описанием структуры хранения данных, методов доступа к данным (описание СУБД для создания БД, синтаксис создания БД и таблиц).

**3.6.1 Описание инструментов разработки**

В данном разделе необходимо описать СУБД для создания БД, описать достоинства и недостатки выбранной СУБД, обосновать выбор данной СУБД.

**3.6.2 Концептуальное проектирование**

Основными задачами концептуального проектирования являются определение предметной области системы и формирование взгляда на предметную область с позиций сообщества будущих пользователей БД, т. е. инфологической модели предметной области.

Концептуальная модель предметной области представляет собой описание структуры и динамики предметной области, характера информационных потребностей пользователей в терминах, понятных пользователю и не зависимых от реализации БД. Это описание выражается в терминах не отдельных объектах предметной области и связей между ними, а их типов, связанных с ними ограничений целостности и тех процессов, которые приводят к переходу предметной области из одного состояния в другое.

Рассмотрим основные подходы к созданию концептуальной модели предметной области:

1. Функциональный подход к проектированию БД.

Этот метод реализует принцип "от задач" и применяется тогда, когда известны функции некоторой группы лиц и/или комплекса задач, для обслуживания информационных потребностей которых создаётся рассматриваемая БД.

2. Предметный подход к проектированию БД.

Предметный подход к проектированию БД применяется в тех случаях, когда у разработчиков есть чёткое представление о самой предметной области и о том, какую именно информацию они хотели бы хранить в БД, а структура запросов не определена или определена не полностью. Тогда основное внимание уделяется исследованию предметной области и наиболее адекватному её отображению в БД с учётом самого широкого спектра информационных запросов к ней.

3. Проектирование с использованием метода «сущность-связь»

Метод "сущность–связь" (entity–relation, ER–method) является комбинацией двух предыдущих и обладает достоинствами обоих. Этап инфологического проектирования начинается с моделирования предметной области. Проектировщик разбивает её на ряд локальных областей, каждая из которых (в идеале) включает в себя информацию, достаточную для обеспечения запросов отдельной группы будущих пользователей или решения отдельной задачи (подзадачи). Каждое локальное представление моделируется отдельно, затем они объединяются.

Выбор локального представления зависит от масштабов предметной области. Обычно она разбивается на локальные области таким образом, чтобы каждая из них соответствовала отдельному внешнему приложению и содержала 6 - 7 сущностей.

Сущность – это объект, о котором в системе будет накапливаться информация. Сущности бывают как физически существующие (например, СОТРУДНИК или АВТОМОБИЛЬ), так и абстрактные (например, ЭКЗАМЕН или ДИАГНОЗ).

Для сущностей различают тип сущности и экземпляр. Тип характеризуется именем и списком свойств, а экземпляр – конкретными значениями свойств.

Типы сущностей можно классифицировать как сильные и слабые. Сильные сущности существуют сами по себе, а существование слабых сущностей зависит от существования сильных. Например, читатель библиотеки – сильная сущность, а абонемент этого читателя – слабая, которая зависит от наличия соответствующего читателя. Слабые сущности называют подчинёнными (дочерними), а сильные – базовыми (основными, родительскими).

Для каждой сущности выбираются свойства (атрибуты):

* Идентифицирующие и описательные атрибуты. Идентифицирующие атрибуты имеют уникальное значение для сущностей данного типа и являются потенциальными ключами. Они позволяют однозначно распознавать экземпляры сущности. Из потенциальных ключей выбирается один первичный ключ (ПК). В качестве ПК обычно выбирается потенциальный ключ, по которому чаще происходит обращение к экземплярам записи. Кроме того, ПК должен включать в свой состав минимально необходимое для идентификации количество атрибутов. Остальные атрибуты называются описательными и заключают в себе интересующие свойства сущности.
* Составные и простые атрибуты. Простой атрибут состоит из одного компонента, его значение неделимо. Составной атрибут является комбинацией нескольких компонентов, возможно, принадлежащих разным типам данных (например, ФИО или адрес). Решение о том, использовать составной атрибут или разбивать его на компоненты, зависит от характера его обработки и формата пользовательского представления этого атрибута.
* Однозначные и многозначные атрибуты (могут иметь соответственно одно или много значений для каждого экземпляра сущности).
* Основные и производные атрибуты. Значение основного атрибута не зависит от других атрибутов. Значение производного атрибута вычисляется на основе значений других атрибутов (например, возраст учащегося вычисляется на основе даты его рождения и текущей даты).

Спецификация атрибута состоит из его названия, указания типа данных и описания ограничений целостности – множества значений (или домена), которые может принимать данный атрибут.

Далее осуществляется спецификация связей внутри локального представления. Связи могут иметь различный содержательный смысл (семантику). Различают связи типа «сущность – сущность», «сущность - атрибут» и «атрибут - атрибут» для отношений между атрибутами, которые характеризуют одну и ту же сущность или одну и туже связь типа «сущность – сущность».

Каждая связь характеризуется именем, обязательностью, типом и степенью. Различают факультативные и обязательные связи. Если вновь порождённый объект одного типа оказывается по необходимости связанным с объектом другого типа, то между этими типами объектов существует обязательная связь (обозначается двойной линией). Иначе связь является факультативной.

По типу различают множественные связи «один к одному» (1:1), «один ко многим» (1:N) и «многие ко многим» (M:N).

Степень связи определяется количеством сущностей, которые охвачены данной связью. Рассмотрим диаграмму сущность-связь на примере кулинарной книги. В качестве основных сущностей выделим БЛЮДО, РЕЦЕПТ, СОСТАВ, ПРОДУКТЫ. Каждая из сущностей, в свою очередь, включает в себя определенные атрибуты. Пример ER–диаграммы с указанием сущностей, их атрибутов и связей приведен на рисунке 1.

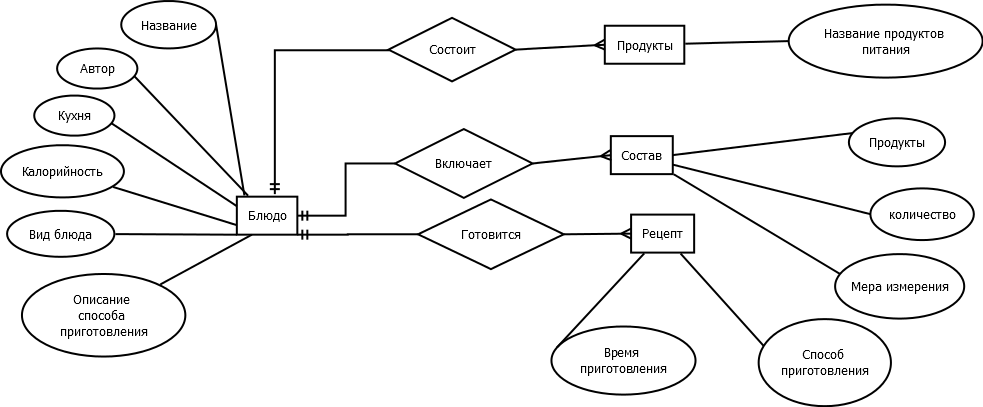


Рисунок 1. Пример ER–диаграммы с однозначными и многозначными атрибутами

**3.6.3 Логическое проектирование БД**

На этапе логического проектирования разрабатывается логическая структура БД, соответствующая логической модели предметной области. Решение этой задачи существенно зависит от модели данных, поддерживаемой выбранной СУБД. Будем рассматривать логическое проектирование БД для реляционной модели данных, так как современные СУБД – реляционные.

Проектирование реляционной базы данных проходит в том же порядке, что и проектирование БД других моделей данных, но имеет свои особенности.

Проектирование схемы БД должно решать задачи минимизации дублирования данных и упрощения процедур их обработки и обновления. При неправильно спроектированной схеме БД могут возникнуть аномалии модификации данных. Они обусловлены отсутствием средств явного представления типов множественных связей между объектами ПО и неразвитостью средств описания ограничений целостности на уровне модели данных.

На этом этапе выполняются следующие действия:

* удаление связей M:N;
* удаление рекурсивных связей;
* удаление связей с атрибутами;
* удаление множественных атрибутов;
* перепроверка связей типа 1:1;
* удаление избыточных связей.

Далее выполняется нормализация отношений. В рамках реляционной модели данных Э. Ф. Коддом (E. F. Codd) был разработан аппарат нормализации отношений и предложен механизм, позволяющий любое отношение преобразовать к третьей нормальной форме.

Нормализация отношений выполняется на основе анализа первичных ключей и существования функциональных зависимостей между атрибутами. Как правило, нормализация выполняется в несколько этапов. Каждый этап соответствует определенной нормальной форме (НФ). При проектировании реляционных баз данных требование первой нормальной формы (1НФ) должны выполняться всегда, остальные по желанию проектировщика. Однако, чтобы исключить аномалии обновления и избыточность данных рекомендуется приводить отношение к третьей нормальной форме 3НФ.

***Требование 1НФ***: все атрибуты должны быть атомарными.

Ненормализованное отношение приводится к 1НФ следующими способами:

* выравнивание таблиц или добавление строк;
* один атрибут или группа атрибутов, которые назначены ключом отношения повторяющейся группы, помещается в отдельные отношения. Во вновь созданных отношениях устанавливаются свои первичные ключи.

***Требование 2НФ***: отношение удовлетворяет 1НФ и каждый атрибут, который не входит в состав первичного ключа, функционально полно зависит от первичного ключа.

Функциональная зависимость описывает связь между атрибутами отношения R(A,B) и обозначается. Атрибут (группа атрибутов) А называется ***детерминантом***.

Полная функциональная зависимость означает, что если атрибут В функционально зависит от первичного ключа, то зависит от полного его значения, а не какого-то подмножества. 2НФ применяется к отношениям с составными ключами.

Для того чтобы привести отношение ко 2НФ, нужно исключить из отношения частичную зависимость и поместить ее в новое отношение вместе с копией их детерминанта.

***Требование 3НФ:*** Отношение находится во 2НФ и каждый неключевой атрибут нетранзитивно зависит от первичного ключа.

Если в отношении R(A, B, C) имеют место следующие функциональные зависимости:

А -> B и B -> C, то говорят, что атрибут С транзитивно зависит от атрибута А через атрибут В.

Для того чтобы привести отношение к 3НФ, нужно исключить из отношения транзитивную зависимость, поместив ее с новое отношение вместе с копией детерминанта.

Процесс нормализации заключается в декомпозиции отношения посредством выполнения последовательных операций проекции.

На этапе логического проектирования необходимо определить требования поддержки целостности данных. Ограничения целостности представляют собой ограничения, которые вводятся с целью предотвращения ввода в базу данных противоречивых данных. Различают следующие пять типов ограничений целостности:

* обязательные данные;
* ограничения для доменов атрибутов;
* целостность сущностей;
* ссылочная целостность;
* требования данного пользователя.

Итогом этого этапа проектирования, должна быть нормализованная схема БД (9-12 таблиц).

Пример схемы базы данных представлен на рисунке 2.

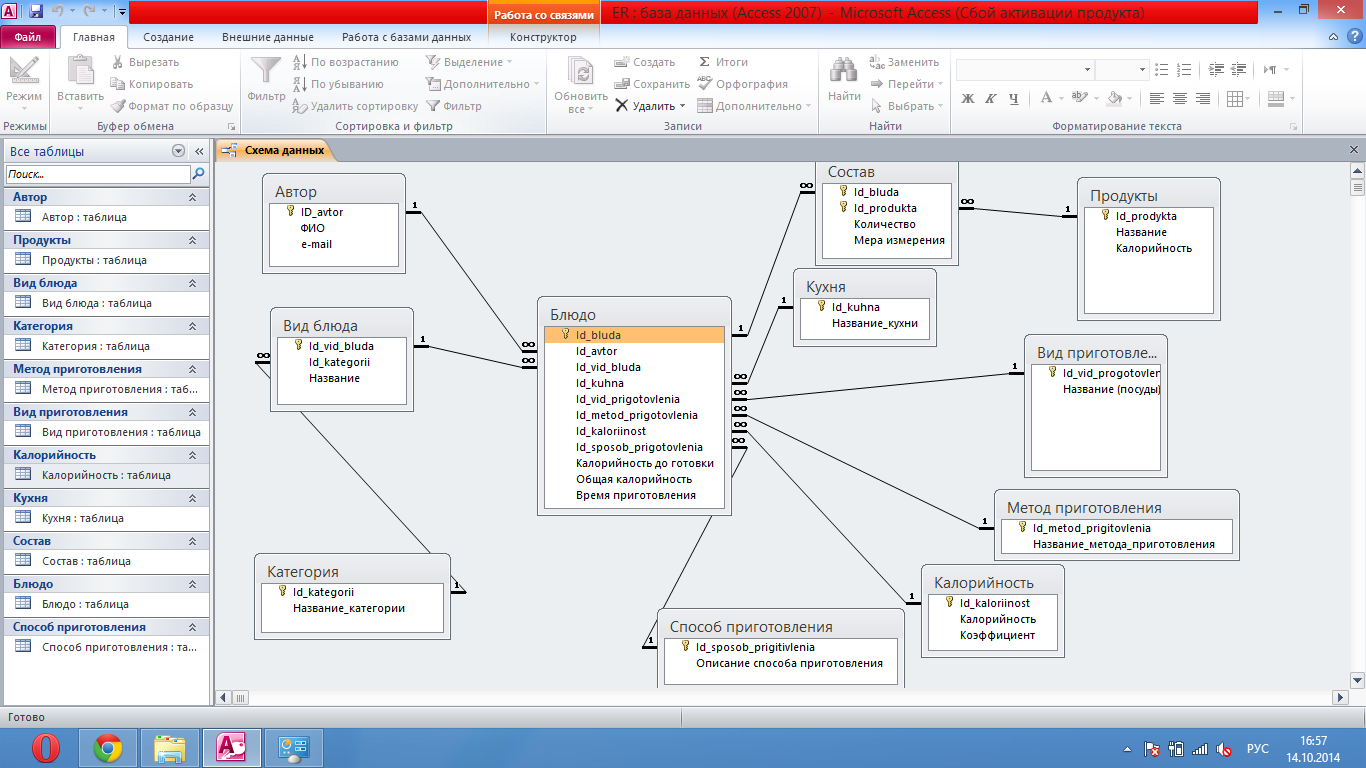


Рисунок 2 – Схема базы данных

**3.6.4 Физическое проектирование БД**

Этап физического проектирования заключается в увязке логической структуры БД и физической среды хранения с целью наиболее эффективного размещения данных, т. е. отображении логической структуры БД в структуру хранения. Решается вопрос размещения хранимых данных в пространстве памяти, выбора эффективных методов доступа к различным компонентам «физической» БД. Результаты этого этапа документируются в форме схемы хранения на языке определения данных (DDL). Принятые на этом этапе решения оказывают определяющее влияние на производительность системы. Между физическим и логическим проектированием существует обратная связь, так как иногда с целью повышения эффективности приходится менять структуру БД. Это возникает, если полная нормализация данных не позволяет достичь требуемой эффективности обработки информации. Принимается решение о денормализации отношений. При этом следует учитывать следующее:

* денормализация усложняет реализацию БД;
* денормализация снижает гибкость системы;
* денормализация ускоряет выборку данных, но снижается скорость обновления данных.

Иногда денормализацию называют оптимизацией исполнения.

Одной из важнейших составляющих проекта базы данных является разработка средств защиты БД. Защита данных имеет два аспекта: защита от сбоев и защита от несанкционированного доступа. Для защиты от сбоев разрабатывается стратегия резервного копирования. Для защиты от несанкционированного доступа каждому пользователю доступ к данным предоставляется только в соответствии с его правами доступа.

Итогом физического проектирования должно быть представление метаданные всех таблиц.

Метаданные таблицы «Калорийность» представлены ниже:

CREATE TABLE KALORIINOST

(

ID\_KALORIINOST INT NOT NULL AUTO\_INCREMENT,

KALORIINOST FLOAT

KOEFF FLOAT

PRIMARY KEY (ID\_KALORIINOST)

);

Метаданные таблицы «Категории» представлены ниже:

CREATE TABLE KATEGORII

(

ID\_KATEGORII INT NOT NULL AUTO\_INCREMENT,

NAZVANIE\_KAT VARCHAR(50),

PRIMARY KEY (ID\_KATEGORII)

);

…

Метаданные таблицы «Блюдо» представлены ниже:

CREATE TABLE BLUDO

(

ID\_BLUDA INT NOT NULL AUTO\_INCREMENT,

ID\_AVTOR INT NOT NULL AUTO\_INCREMENT,

ID\_VID\_BLUDA INT NOT NULL AUTO\_INCREMENT,

ID\_KUHNA INT NOT NULL AUTO\_INCREMENT,

ID\_VID\_PRIGOTOVLENIA INT NOT NULL AUTO\_INCREMENT,

ID\_METOD\_PRIGOTOVLENIA INT NOT NULL AUTO\_INCREMENT,

ID\_KALORIINOST INT NOT NULL AUTO\_INCREMENT,

ID\_SPOSOB\_PRIGOTOVLENIA INT NOT NULL AUTO\_INCREMENT,

OBS\_KALORIINOST FLOAT,

VREMYA\_PRIG VARCHAR(50),

KOLICH\_PRODUKTOV INT,

PRIMARY KEY (ID\_BLUDA)

);

ALTER TABLE BLUDO ADD FOREIGN KEY (ID\_AVTOR) REFERENCES AVTOR (ID\_AVTOR);

ALTER TABLE BLUDO ADD FOREIGN KEY (ID\_VID\_BLUDA) REFERENCES VID\_BLUDA (ID\_VID\_BLUDA);

ALTER TABLE BLUDO ADD FOREIGN KEY (ID\_KUHNA) REFERENCES KUHNA (ID\_KUHNA);

ALTER TABLE BLUDO ADD FOREIGN KEY (ID\_VID\_PRIGOTOVLENIA) REFERENCES VID\_PRIGOTOVLENIA (ID\_VID\_PRIGOTOVLENIA);

ALTER TABLE BLUDO ADD FOREIGN KEY (ID\_METOD\_PRIGOTOVLENIA) REFERENCES METOD\_PRIGOTOVLENIA (ID\_METOD\_PRIGOTOVLENIA);

ALTER TABLE BLUDO ADD FOREIGN KEY (ID\_KALORIINOST) REFERENCES KALORIINOST (ID\_KALORIINOST);

ALTER TABLE BLUDO ADD FOREIGN KEY (ID\_SPOSOB\_PRIGOTOVLENIA) REFERENCES SPOSOB\_PRIGOTOVLENIA (ID\_SPOSOB\_PRIGOTOVLENIA);

**3.6.5 Проектирование диаграммы деятельности**

Проектирование диаграммы деятельности является важным этапом при разработке приложения. Диаграмма деятельности (Activity Diagrams) является представлением алгоритмов неких действий (активностей), выполняющихся в системе.

Основные компоненты диаграммы деятельности представлены в таблице 1.

Таблица 1 – Компоненты диаграммы деятельности

|  |  |  |
| --- | --- | --- |
| № | Изображение | Описание |
| 1 | https://www.intuit.ru/EDI/23_04_17_1/1492899714-28128/tutorial/356/objects/4/files/04_02.gif | Начальное состояние |
| 2 | https://www.intuit.ru/EDI/23_04_17_1/1492899714-28128/tutorial/356/objects/4/files/04_02.gif | Конечное состояние |
| 3 |  | Действие (скругленные прямоугольники, указывается начальная форма глагола, например, принять заказ) |
| Продолжение таблицы 1 | | |
| № | Изображение | Описание |
| 4 | https://www.intuit.ru/EDI/23_04_17_1/1492899714-28128/tutorial/356/objects/4/files/04_03.gif | Ветвление (один вход, много выходов) |
| 5 | https://www.intuit.ru/EDI/23_04_17_1/1492899714-28128/tutorial/356/objects/4/files/04_03.gif | Объединение (много входов, один выход) |
| 6 | https://www.intuit.ru/EDI/23_04_17_1/1492899714-28128/tutorial/356/objects/4/files/04_03.gif | Такого компонента быть не может |
| 7 | Диаграмма деятельности UML - Решение | Решение  Условие |
| 8 | Диаграмма деятельности UML - Слияние | Слияние |

Пример диаграммы деятельности представлен на рисунке 3

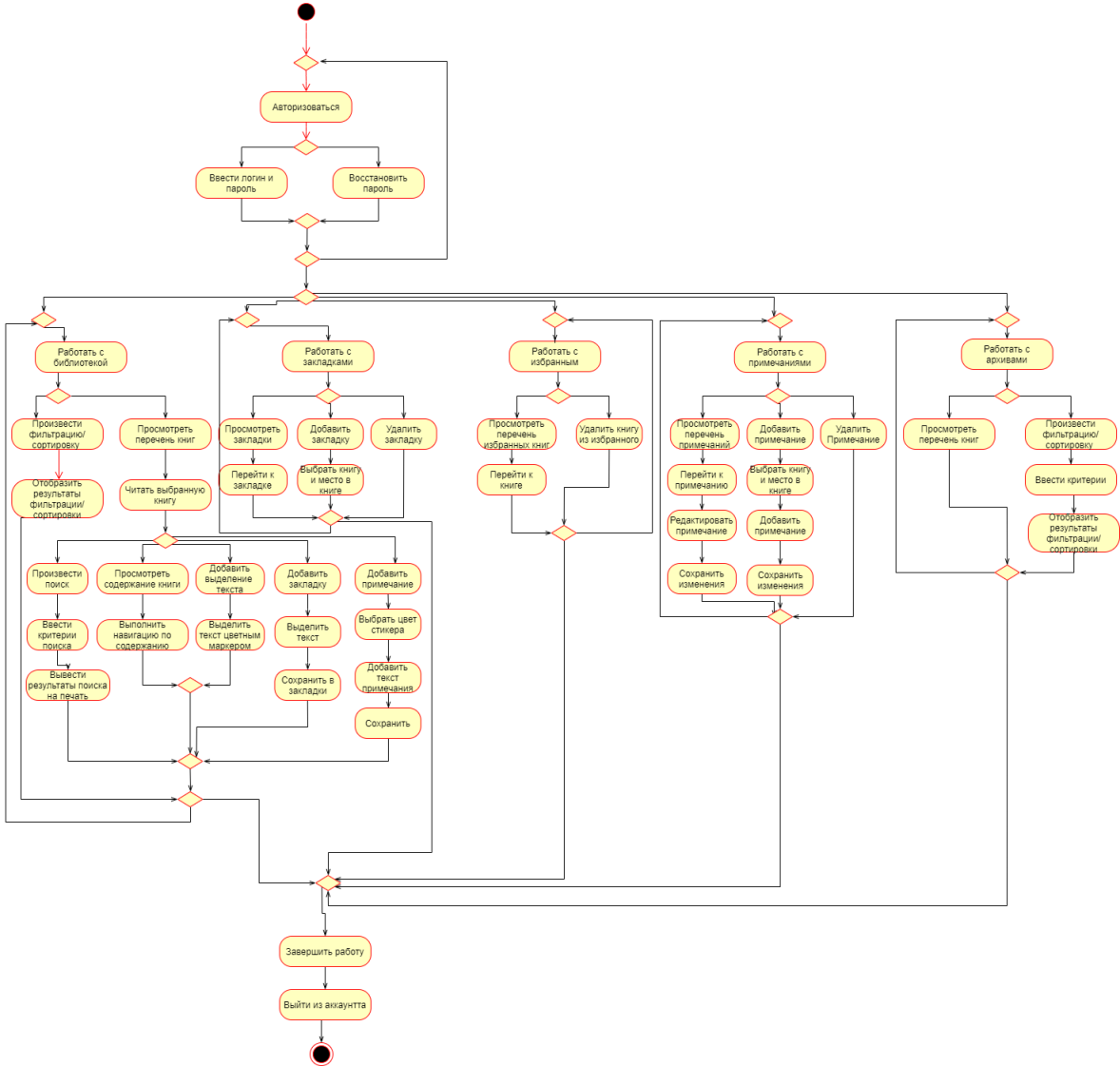


Рисунок 3 – Диаграмма деятельности

**3.7 Программная реализация**

Данный раздел предполагает описание программной реализации курсового проекта, а именно описание инструментов разработки программного средства, описание разработанных компонентов базы данных и описание разработанных модулей приложения.

В зависимости от особенностей выполненного проекта основную часть излагают в виде текста или сочетания текста, иллюстраций и таблиц.

**3.7.1 Описание разработанных компонентов базы данных**

В данном разделе пояснительной записки необходимо описать разработанные компоненты базы данных. В частности, подробное описание схемы базы данных, ограничений целостности. Кроме подробного описания таблиц в данном разделе приводится описание триггеров, обзоров и хранимых процедур.

**3.7.1 Описание разработанных модулей приложения**

В данном разделе пояснительной записки необходимо описать функции, процедуры, методы, классы разработанного приложения. Необходимо кратко охарактеризовать суть и назначение.

**3.8 Тестирование**

Должны быть разработаны способы проверки правильности и работоспособности отдельных функций и программной системы в целом (тест-кейсы).

Спланированы и проведены экспериментальные исследования, отражающие функционирование разработанной программной системы.

Анализ результатов должен содержать обобщение и оценку результатов исследований, включающие оценку полноты решения поставленной задачи и предложения по дальнейшим направлениям работ, оценку достоверности полученных результатов и их сравнение с аналогичными результатами получаемыми другими системами, обоснование необходимости проведения дополнительных исследований.

Тест-кейс можно представить в виде таблицы, таблица 2.

Таблица 2 – Тест-кейс

|  |  |  |  |
| --- | --- | --- | --- |
| Название  операции | Входной  параметр | Ожидаемый  результат | Фактический  результат |
| Запуск | Запуск файла  start.exe | Вход в систему | Вход в систему |

**3.9 Руководство пользователя**

Руководство пользователя должно содержать описание технических и программных средств, необходимых для успешной работы системы, и последовательность действий, выполняемых пользователем.

**3.9.1 Описание процесса установки и запуска приложения**

В данном разделе необходимо описать процесс установки и запуска приложения. Если требуется наличие сторонних программ для полнофункциональной работы приложения, также необходимо описать способ установки, подключение дополнительных модулей.

**3.10 Список использованных источников**

Данный раздел должен содержать перечень источников, цитируемых и изученных при написании курсового проекта, которые следует располагать по алфавиту авторов или заглавий (в случае четырех авторов и более). Сведения об источниках необходимо привести в соответствии с требованиями ГОСТ 7.1–2003 и приложением Г.

**3.11 Приложения**

Приложения оформляются как продолжение работы на последующих ее страницах, располагая их в порядке появления ссылок на них в тексте. В приложения следует включать вспомогательный материал, необходимый для полноты восприятия, – таблицы вспомогательных цифровых данных, описание алгоритмов и программ задач, решаемых на ЭВМ, разработанных в процессе выполнения работы, иллюстрации вспомогательного характера (все формы приложения).

## 3.12 Защита курсового проекта

Защита курсового проекта осуществляется в соответствии с графиком курсового проектирования. Для сдачи курсового проекта учащийся должен представить пояснительную записку, графическую часть, диск, содержащий выполняемый файл, а также все программные (исходные) файлы и файлы графических документов. К защите учащийся допускается с презентацией к курсовому проекту.

**Рекомендуемая литература**

***Нормативные документы***

1 ГОСТ 2.104–2006 ЕСКД. Основные надписи. Общие положения. – Взамен ГОСТ 2.104–68 ; введ. 2006–08–29. – М. : Изд-во стандартов ; Минск : Госстандарт РБ, 2006.

2 ГОСТ 2.105–95 ЕСКД. Общие требования к текстовым документам. – Взамен ГОСТ 2.105–79, ГОСТ 2.906–71 ; введ. 1997–01–01. – М. : Стандартинформ, 2005.

3 ГОСТ 7.1–2003 Система стандартов по информационному, библиотечному и издательскому делу. Библиографическая запись. Библиографическое описание. Общие требования и правила составления. – Введ. 2004–11–01. – Минск : Госстандарт РБ, 2004.

4 ГОСТ 19.701–90 ЕСПД. Схемы алгоритмов, программ, данных и систем. Обозначения условные и правила выполнения. – Введ. 1992–01–01 ; переизд. – М. : Стандартинформ, 2005.

***Основная***

5 Бекаревич, Ю.Самоучитель MS Office Access 2016 / Ю.Бекаревич – СПб.: БХВ-Петербург, 2017

7 Дюбуа, П. MySQL : пер. с англ. / П. Дюбуа. – 3-е изд. – М. : Вильямс, 2007. – 1168 с.

8 Ковязин, А.Н. MySQL по максимуму / А.Н.Ковязин, С.М.Востриков – СПб.: Питер, 2018.

9 Колисниченко, Д. Н. PHP 5/6 и MySQL 6. Разработка Web-приложений / Д. Н. Колисниченко. – СПб. : БХВ–Петер-бург, 2011.

10 Лазицкас, Е.А. Базы данных и системы управления базами данных / Е.А.Лазицкас, И.Н. Загуменникова, П.Г. Гилевский – РИПО, 2018.

11 Леоненков, А. В. Самоучитель UML / А. В. Леоненков. – СПб. : БХВ–Петербург, 2002. – 304 с.

12 Никсон, Р.LearningPHP, MySQL&JavaScript: WithjQuery, CSS&HTML5 (LearningPhp, Mysql, Javascript, Css&Html5) / Р.Никсон – O’Reilly, 2015.

13 Прохоренок, Н.А. HTML, JavaScript, PHP и MySQL Джентельменский набор Web-мастера / Н.А.Прохоренок – СПб.: БХВ-Петербург, 2016.

14 Рихтер, Дж.CLR via C#. Программирование на платформе Microsoft .NET Framework 4.5 на языке C#. 4-е изд. / Дж. Рихтер – СПб.: Питер, 2018.

.

***Дополнительная***

15 Баркер, Скотт Ф. Профессиональное программирование в Microsoft Access 2002 / С. Ф. Баркер. – М. : Вильямс, 2002. – 272 c.

16 Диго, С. М. Базы данных: проектирование и использование / С. М. Диго. – М. : Финансы и статистика, 2005. – 590 с.

17 Кузнецов, С. Д. Основы баз данных / С. Д. Кузнецов. – М. : Интернет-Университет информационных технологий, 2005. – 484 с.

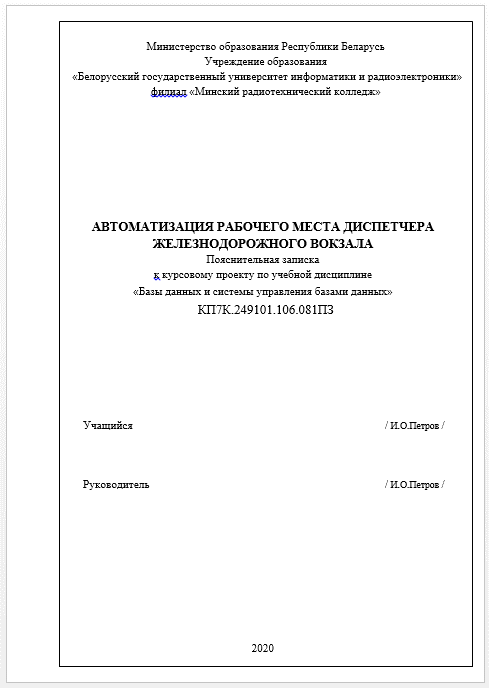
18 Марков, А. С. Базы данных. Введение в теорию и методо-логию / А. С. Марков, К. Ю. Лисовский. – М. : Финансы и статистика, 2004. – 510 с.

19 Харрингтон, Дж. Л. Проектирование реляционных баз дан-ных / Дж. Л. Харрингтон. – М. : Лори, 2006. – 230 с.

**Приложение А**

(справочное)

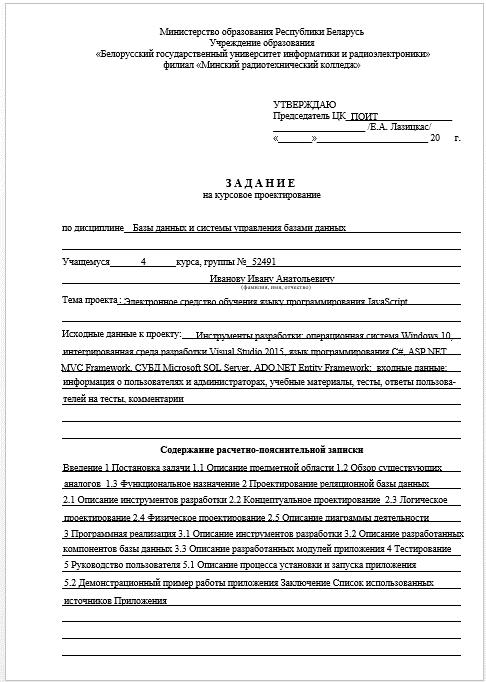
**Пример оформления титульного листа**

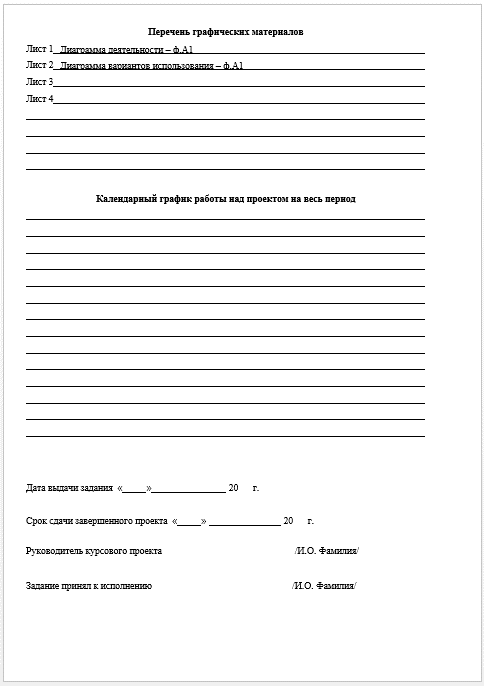


**Приложение Б**

(справочное)

**Пример оформления заданияна курсовое проектирование**



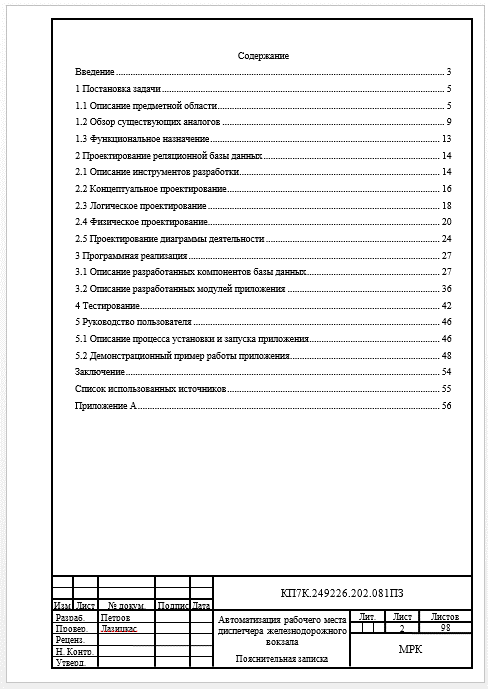


**Приложение В**

(справочное)

#### Пример оформления содержания

#### пояснительной записки

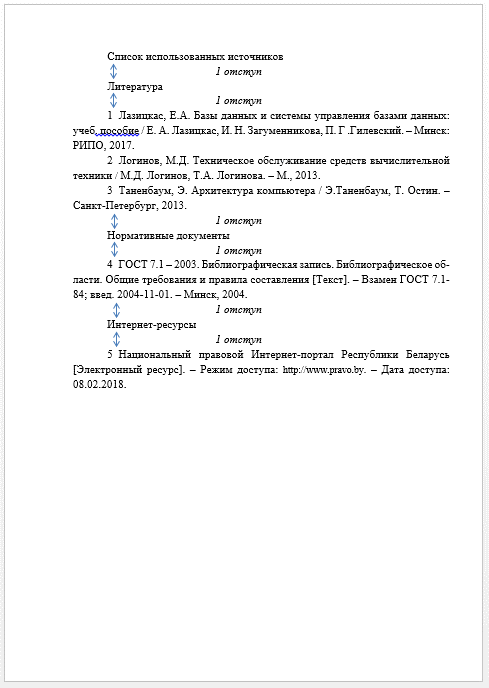


**Приложение Г**

(справочное)

#### Пример оформления

#### списка использованных источников

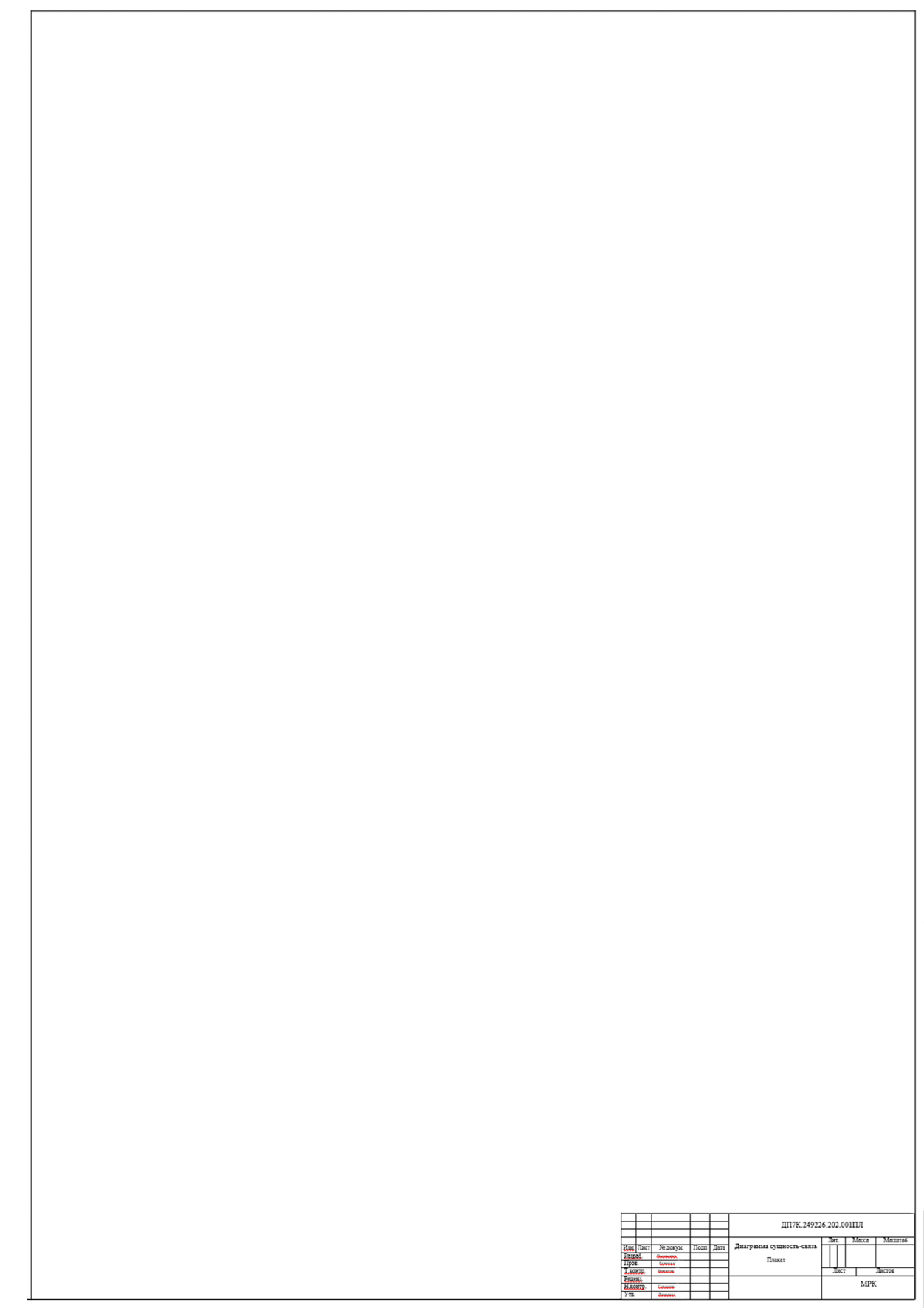


**Приложение Д**

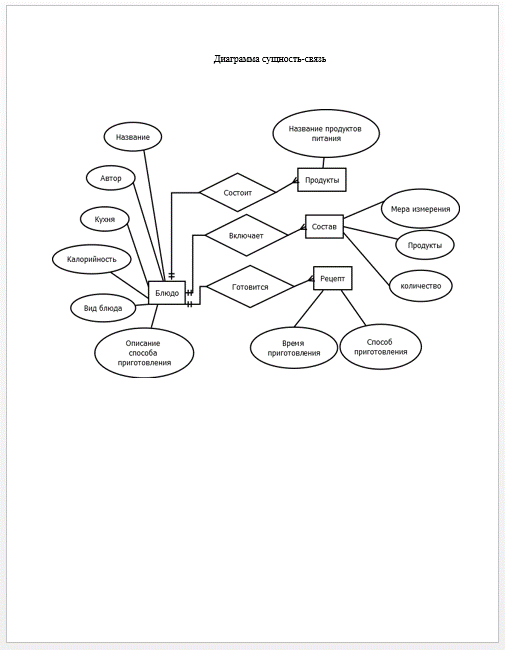
(справочное)

#### Пример оформления диаграмм

Оборотная сторона

****

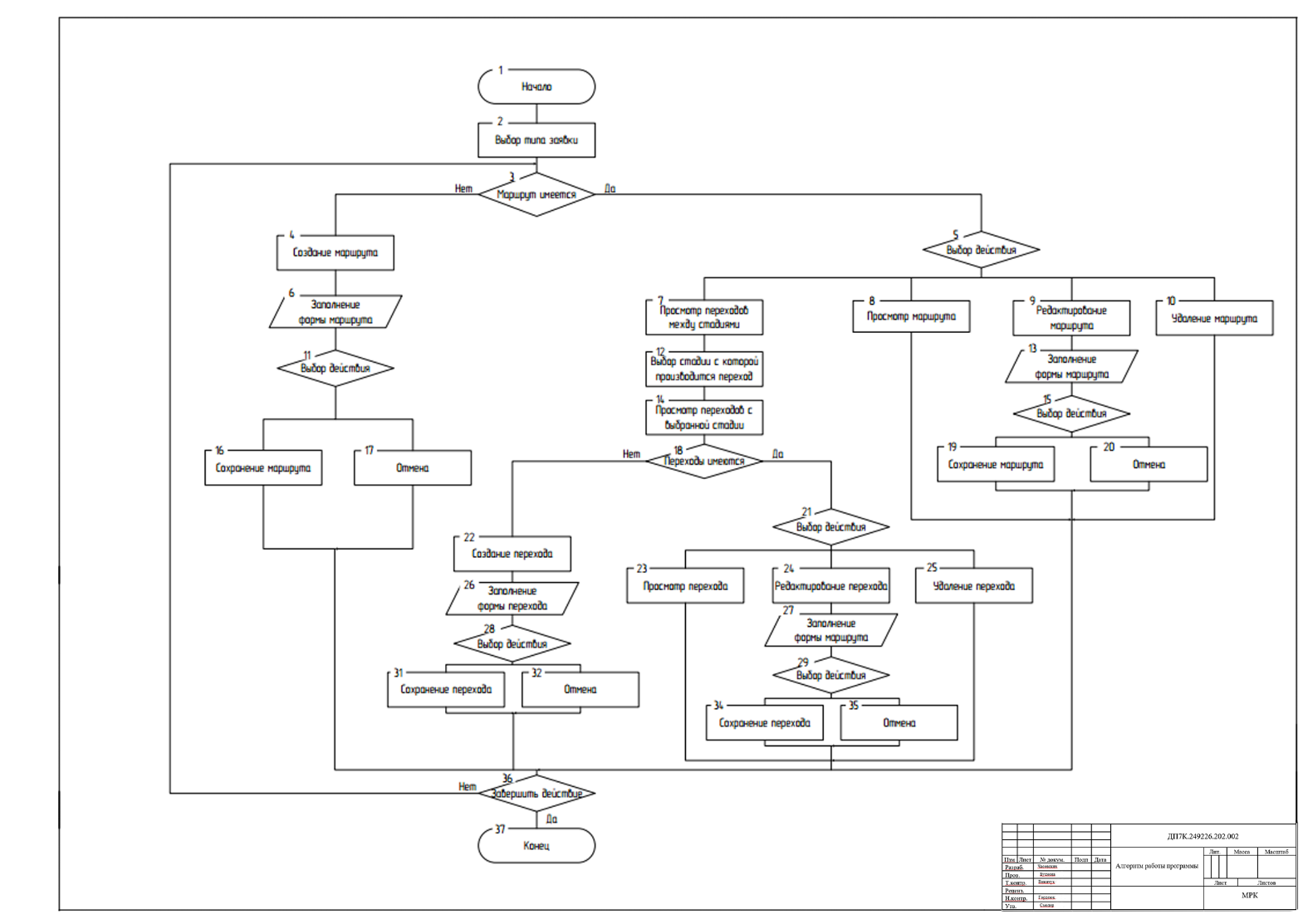
Лицевая сторона



**Приложение Е**

(справочное)

#### Пример оформления схемы алгоритма

****