

Министерство образования и науки Российской Федерации

ФИЛИАЛ ГОСУДАРСТВЕННОГО ОБРАЗОВАТЕЛЬНОГО УЧРЕЖДЕНИЯ
ВЫСШЕГО ОБРАЗОВАНИЯ «БАЙКАЛЬСКИЙ ГОСУДАРСТВЕННЫЙ
УНИВЕРСИТЕТ» В Г. УСТЬ-ИЛИМСКЕ

Кафедра Экономики и менеджмента

«ИНФОРМАЦИОННЫЕ СИСТЕМЫ И ТЕХНОЛОГИИ»

МЕТОДИЧЕСКИЕ УКАЗАНИЯ
ПО ВЫПОЛНЕНИЮ КУРСОВОЙ РАБОТЫ

Направление подготовки 230700 Прикладная информатика

Профиль подготовки: Информационные системы и технологии в управлении

Составитель:
к.п.н., доцент

С.А. Борцова

Усть-Илимск, 2016

Методические указания составлены в соответствии с рабочей программой курса "Информационные системы и технологии" для студентов, обучающихся по направлению бакалавриата 09.03.03 «Прикладная информатика».

Цель курсовой работы состоит в углубленном изучении принципов и приемов проектирования автоматизированных систем, особенностей систем на предприятии и в вузе, и приобретении практических навыков в решении задач на персональных ЭВМ.

Приводится тематика курсовых работ, основные принципиальные положения, практические примеры.

СОДЕРЖАНИЕ

Введение

1. Задание на курсовую работу
2. Содержание пояснительной записки (ПЗ)
 - 2.1. Структура ПЗ
 - 2.2. Порядок выполнения ПЗ курсовой работы
 - 2.2.1. Постановка задачи
 - 2.2.2. Техническое задание на автоматизированную систему
 - 2.2.3. Анализ известных подходов к решению проблемы
 - 2.2.4. Функциональное моделирование деятельности объекта
 - 2.2.5. Инфологическая модель системы.
 - 2.2.6. Физическая база данных
 - 2.2.7. Список используемой литературы
3. Теоретические основы, необходимые для выполнения курсовой работы
 - 3.1. Перечень вопросов, подлежащих изучению
 - 3.2. Построение и анализ диаграмм потоков данных DFD
 - 3.3. Построение словаря данных
 - 3.4. Построение модели «сущность-связь» и ее нормализация

Список рекомендуемой литературы

Приложение А

Приложение В

Приложение С

Приложение D

Приложение E

Приложение F

Приложение G

Приложение H

Введение

Целью курсовой работы является закрепление основ и углубление знаний принципов и приемов проектирования автоматизированных систем для различных предметных областей, а также приобретение практических навыков в решении прикладных задач.

В курсовой работе предлагается разработать проект автоматизированной системы для некоторой предметной области с учетом всех его особенностей. При этом требуется самостоятельно определить структуру и характеристики используемых данных, предусмотреть ряд функций по обработке информации.

При выполнении курсовой работы студент самостоятельно осваивает все этапы проектирования автоматизированной системы от постановки задачи до практической реализации, сопровождающейся документацией. При выполнении курсовой работы студентам необходимо обратить внимание на качество создаваемого продукта.

1. Задание на курсовую работу

Необходимо спроектировать автоматизированную систему некоторой предметной области.

Проект включает в себя постановку задачи с представлением предметной области задачи проектирования, анализ существующих или возможных решений поставленной задачи с кратким обзором литературных источников, алгоритмическую проработку решений, формулировку технического задания (ТЗ) для системы, выбор среды реализации с использованием средств автоматизации проектирования.

В курсовом проекте, как правило, должны быть представлены результаты отладки проектируемых компонент в средствах выбранной CASE-среды.

Практическая реализация физической базы данных предполагает использование Системы управления реляционными базами данных (СУБД) MS ACCESS.

2. Содержание пояснительной записки (ПЗ)

2.1. Структура ПЗ

Пояснительная записка должна содержать все разделы, отражающие этапы проектирования автоматизированных систем предприятия, должны быть выполнены рубрикация текста (см. как это выполнено в данном пособии), пронумерованы страницы, сделаны ссылки на используемую литературу (для ссылок используются квадратные скобки, например [3]) и составлено оглавление.

Основные разделы пояснительной записки:

- Постановка задачи (1-2 стр.)
- Техническое задание на автоматизированную систему (1 - 2 стр.).
- Анализ известных подходов к решению проблемы (3-5 стр.).
- Функциональное моделирование деятельности объекта; (3 - 5 стр.)
- Инфологическая модель системы; (3-5 стр.)
- Физическая база данных; (5-10 стр.)
- Список используемой литературы (1 стр.).

2.2. Порядок выполнения ПЗ курсовой работы

2.2.1. Постановка задачи

В данном разделе следует привести общую характеристику объекта автоматизации; описание организационной структуры объекта автоматизации; характеристика вида деятельности. Определить общую цель работы.

2.2.2. Техническое задание на автоматизированную систему

В процессе разработки технического задания (ТЗ) студент получает практические навыки в составлении ТЗ на программное обеспечение по выданному

заданию на курсовую работу. В ТЗ анализируются требования к программному продукту.

Здесь должны быть перечислены все функции, которые необходимо реализовать в процессе создания программы.

2.2.3. Анализ известных подходов к решению проблемы

Описание существующих программ, методов и средств, позволяющих решить данную проблему.

2.2.4. Функциональное моделирование деятельности объекта

В этом разделе следует привести описание объекта автоматизации и информационных процессов. Описание входных, выходных и нормативно-справочных документов и экранные формы.

Построить диаграмму DFD (не менее 3 уровней детализации). Провести анализ построенных диаграмм.

Описать словарь используемых данных.

2.2.5. Инфологическая модель системы.

В данном разделе описывается логическая структура модели данных. Здесь следует перечислить основные сущности, их структуру и связи между ними. Выделить ключевые атрибуты сущностей. Показать процесс приведения данных к 3 нормальной форме

На основании модели “сущность - связь” сгенерировать проект базы данных.

2.2.6. Физическая база данных

В данном разделе приводится пример реализации базы данных в СУБД MS ACCESS. База данных должна включать таблицы, запросы, формы, посредством которых реализуются все перечисленные в техническом задании функции, а также возможность генерации различных отчетов по системе.

2.2.7. Список используемой литературы

Список литературы приводится по общепринятым правилам для технической литературы. На приведенную литературу должны быть сделаны ссылки в тексте пояснительной записки.

3. Теоретические основы, необходимые для выполнения курсовой работы

3.1. Перечень вопросов, подлежащих изучению

В процессе выполнения работы студент должен разобраться в следующих вопросах:

- построение и анализ диаграмм потоков данных DFD;
- построение словаря данных;
- построение модели «сущность-связь» и ее нормализация;

3.2. Построение и анализ диаграмм потоков данных DFD

DFD – графическая модель представления информации о предметной области.

Диаграммы потоков данных строятся из четырех основных элементов (рис.1):

- внешняя сущность;
- поток данных;
- процесс;
- накопитель данных.

Внешняя сущность представляет на диаграмме модели, организации, системы, находящиеся за пределами объекта разработки (за границами проектирования). Вместе с тем внешние сущности являются источниками или приемниками информации по отношению к рассматриваемой системе и включаются в диаграмму для спецификации ее интерфейса. На диаграмме внешняя сущность изображается прямоугольником, внутри которого записывается ее уникальный идентификатор и имя (см. рис.1, *а*). Уникальный идентификатор образуют литера **Е** и порядковый номер внешней сущности.

Поток данных изображается стрелкой, направление которой совпадает с направлением потока данных. Над стрелкой указывается его имя (см. рис. 1, *б*). Поток данных может рассматриваться как пневмопочта или конвейер, передающий пакеты данных между обрабатывающими их процессами.

Под **процессами** на диаграмме потоков данных понимаются произвольные задачи (вычислительная, задача управления и т. д.), в ходе выполнения которых совершается некоторая обработка информации. На диаграмме процесс обозначается прямоугольником, разбитым на две части (рис. 1, в). В верхней части прямоугольника записывается уникальный идентификатор процесса на диаграмме, в нижней — имя выполняемой функции. Уникальный идентификатор процесса образуют литера **P** и порядковый номер процесса.

В реальных системах довольно часто встречается ситуация, когда информация не может быть обработана сразу, она должна быть накоплена и сохранена для последующей обработки. При описании данных, обработка которых отложена, или данных, обрабатываемых многократно, используются накопители данных.

Накопитель данных изображается на диаграмме прямоугольником с одной открытой стороной, внутри которого указывается уникальный идентификатор и имя накопителя (см. рис. 1, г). Уникальный идентификатор накопителя образуют литера **D** и его порядковый номер.

Правила соединения элементов на диаграмме потоков данных иллюстрирует рис. 2. Процесс может:

- принимать или передавать данные внешней сущности;
- принимать или передавать данные другому процессу;
- считывать или заносить данные в накопитель, другие связи запрещены.

Количество элементов на диаграммах потоков данных, описывающих даже относительно несложные системы, может достигать нескольких сотен. Так как построение и чтение таких диаграмм затруднено, строится их иерархия (рис. 3).

На верхнем уровне иерархии находится так называемая контекстная диаграмма, на которой рассматриваемая система представлена одной вершиной типа процесс. Контекстная диаграмма предназначена для отображения внешних связей системы, поэтому на ней изображаются внешние сущности, а также поступающие к ним и принимаемые от них потоки данных.

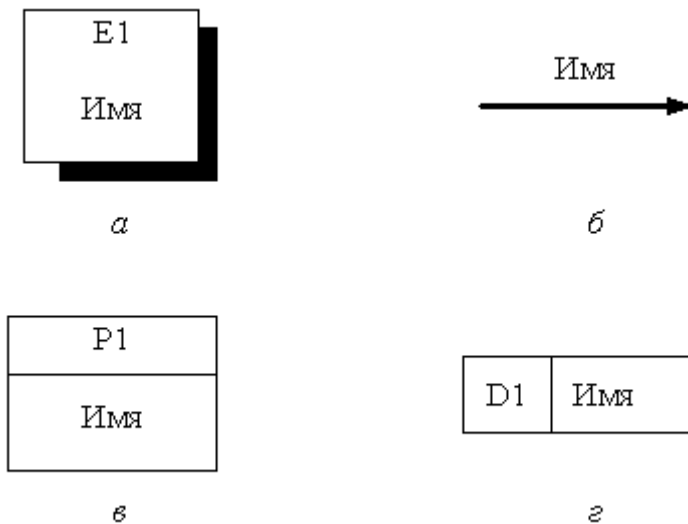


Рис. 1. Элементы диаграмм потоков данных:
a - внешняя сущность; *б* - поток данных; *в* - процесс;
г - накопитель данных

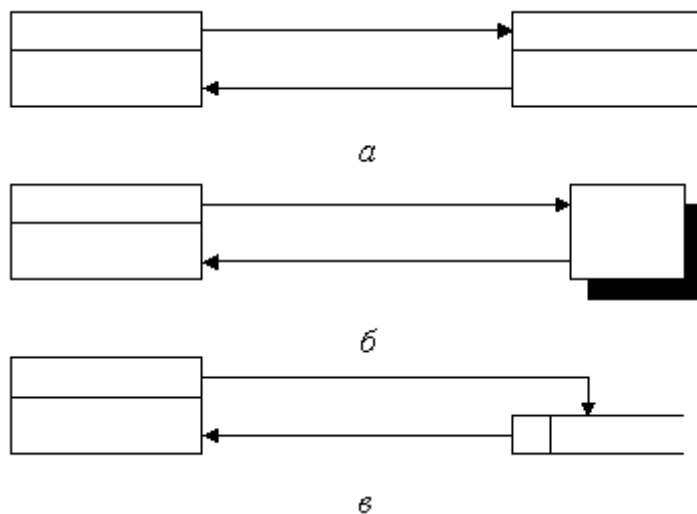


Рис. 2. Правила соединения элементов диаграмм потоков данных:
a - «процесс - процесс»; *б* - «процесс - внешняя сущность»;
в - «процесс - накопитель данных»

На диаграмме следующего уровня иерархии рассматриваемая система разбивается на несколько процессов. Любой процесс данной диаграммы может быть детализирован, т. е. ему может соответствовать диаграмма более низкого уровня

иерархии. Количество иерархических уровней диаграмм ничем не ограничено; процесс декомпозиции имеет смысл прекратить, если описание процесса может быть сделано достаточно компактно с помощью деревьев и таблиц решений или на структурном естественном языке.

При построении иерархии диаграмм потоков данных должны соблюдаться следующие правила:

- 1) внешние сущности и накопители данных при необходимости дублируются на диаграммах более низкого уровня (копии накопителей и внешних сущностей специальным образом помечаются на диаграмме);
- 2) накопители данных размещаются на том уровне, на котором они используются более чем одним процессом (соблюдение данного требования позволяет избежать чрезвычайного усложнения диаграмм верхних уровней иерархии);
- 3) входные и выходные потоки детализируемого процесса должны либо дублироваться, либо уточняться на диаграмме следующего уровня – иначе говоря, входными потоками для диаграммы более низкого уровня могут быть либо входные потоки процесса верхнего уровня, либо их компоненты (с учетом сделанных в словаре описаний структур данных); аналогичное требование справедливо и для выходных потоков диаграммы.

Жестких ограничений на количество элементов, расположенных на диаграмме, не существует; вместе с тем оптимальным числом считается порядка семи процессов. При большем количестве диаграмма становится трудно читаемой, а меньшее число приводит к появлению неоправданно большого количества диаграмм.

ПРАВИЛА ИМЕНОВАНИЯ ЭЛЕМЕНТОВ

1. Процесс именуется глаголом или какой-то глагольной формой, обозначающей действие.
2. Внешняя сущность именуется существительным, определяющим тип компонента.
3. Накопитель данных – название должно отражать хранимую информацию, а не способ реализации накопителя.

4. Поток данных именуется в соответствии с теми данными, которые он переносит.

АНАЛИЗ ДИАГРАММ ПОТОКОВ ДАННЫХ

DFD может быть признана корректной при выполнении некоторых условий. Различают две основные стадии анализа:

1. Статический анализ правильности

Здесь не учитывается логика процессов, он направлен на проверку правильности соединения элементов.

Условия соединения элементов:

- а) внешняя сущность имеет хотя бы один входной или выходной поток данных
- б) процесс имеет хотя бы один входной и выходной поток данных
- в) соответствие потоков данных на диаграмме потокам процесса верхнего уровня
- г) баланс накопителя (описываем ошибочные ситуации)

- ✓ данные, приносимые в накопитель, отсутствуют в накопителе
- ✓ данные, выносимые выходным потоком данных, отсутствуют в накопителе данных
- ✓ данные, хранимые в накопителе, не привносятся входным потоком
- ✓ данные, хранимые в накопителе, не забираются выходным потоком
- ✓ информационный канал не порождает и не поглощает информацию

2. Динамический анализ DFD

Для контекстной диаграммы:

- 1) помечать выходные потоки внешней сущности маркером
- 2) каждый процесс, у которого все входные потоки данных помечены маркером, можно объявить выполненным, а на выходные потоки поместить маркер
- 3) если все процессы выполнены и на всех потоках есть маркеры, то такая диаграмма корректна.

Дополнение к пункту 2: процесс, убирающий данные из накопителя, может быть выполнен, если существует другой процесс, который помещает данные в накопитель, и он помечен как выполненный.

Для диаграммы, декомпозирующей процесс:

- 1) помечается маркером входные потоки данных процесса P_i , изображенные на декомпозирующей диаграмме
- 2) аналогично для контекстной диаграммы
- 3) если все потоки данных, совпадающие с выходными потоками 1 уровня, имеют маркеры, то диаграмма корректна.

3.3. Построение словаря данных

Понятия, вводимые на диаграммах потоков данных, описываются в словаре данных.

Можно выделить три уровня описания:

1. элемент данных – информационный объект, который в контексте рассматриваемой задачи не имеет смысла подвергать дальнейшей детализации;
2. структура данных – агрегат, состоящий из элементов и/или других структур данных;
3. потоки и накопители данных; потоки – это структуры данных, находящиеся в движении, накопители – статические структуры данных.

Описание элемента данных в общем случае включает:

- имя;
- синонимы;
- связанные элементы данных;
- диапазон и смысл значений;
- длину;
- способ кодирования;
- другую необходимую информацию.

Имя элемента данных выбирается таким образом, чтобы оно было максимально информативно для пользователей (заказчиков) системы.

Синонимы возникают в том случае, когда различные пользователи называют одно и то же понятие по-разному или понятие выступает в различных ролях по отношению к различным пользователям.

Связанные элементы данных возникают в ситуации, когда одна и та же информация представляется различным образом (записывается в различном формате). Например, время может записываться как 6:30PM (формат, принятый в США) или как 18:30; тогда первый элемент мог бы иметь имя «НАЦИОНАЛЬНОЕ ВРЕМЯ», а второй, связанный с ним, – просто «ВРЕМЯ».

Диапазон и смысл значений может указываться двояким образом:

- путем указания минимальной и максимальной границы; например, номер абонента – от 0 до 9999999 (непрерывный элемент данных);
- путем перечисления множества возможных значений; например, день недели – ПН, ВТ, СР, ЧТ, ПТ, СБ, ВС, тип запроса – поиск, актуализация (дискретный элемент данных).

Элемент данных целесообразно определять как дискретный, когда важен смысл значений. Например: типы запросов используются для их классификации; в воскресенье (ВС) система обслуживает ограниченный круг абонентов и т. д.

Длина и способ кодирования элемента данных на начальных этапах разработки, как правило, не задаются.

Описание **структуры данных** представляет собой перечень элементов и структур данных, ее образующих. Любая структура данных может быть построена одним из трех возможных способов (см. таблицу):

- объединение (агрегация);
- альтернатива (выбор);
- повторение (итерация).

Агрегация позволяет построить новый информационный объект посредством объединения других объектов.

Составляющие объекты обязательно входят в структуру объекта верхнего уровня, например «АДРЕС» – это «ИНДЕКС», «ГОРОД», «УЛИЦА», «ДОМ».

Альтернатива позволяет представлять данные с переменной структурой. Так книга может быть сборником (тогда для нее указывается редактор) или монографией (тогда — автор).

Итерация позволяет описывать данные с регулярной структурой: массивы, файлы, картотеки и т. д.

Способы построения структур данных

Способ	Графическое обозначение	Текстовая форма записи	Пример
Агрегация (объединение)		A B C	АДРЕС ИНДЕКС ГОРОД УЛИЦА ДОМ
Выбор (альтернатива)		A { B C }	КНИГА {СБОРНИК МОНОГРАФИЯ }
Итерация (повторение)		A*(N-M) B C	КАТАЛОГ* (1-1000) КАРТОЧКА

(N-минимальное, M-максимальное количество элементов)

Помимо описаний элементов и структур данных в словаре проекта приводятся описания:

- потоков данных;
- накопителей данных;
- процессов;
- внешних сущностей.

Для каждого потока данных указываются:

- имя потока;

- источник потока;
- приемник потока;
- перечень переносимых потоком элементов и структур данных.

Для каждого накопителя указываются :

- имя накопителя;
- входящие потоки;
- выходящие потоки;
- перечень структур и элементов данных, хранимых в накопителе.

Для каждого процесса указываются:

- имя процесса на диаграмме;
- входные потоки;
- выходные потоки;
- краткое описание логики.

Для каждой внешней сущности указываются:

- имя внешней сущности на диаграмме;
- входные потоки данных;
- выходные потоки данных;
- количество экземпляров, если сущность — множественная.

Словарь данных является центральным хранилищем информации о проекте, которая используется всеми аналитиками, проектировщиками и программистами.

На основе словаря данных:

- формируются отчеты (проектная документация);
- организуется анализ полноты, непротиворечивости, связности;
- производится синтез структур данных.

3.4. Построение модели «сущность-связь» и ее нормализация

Модель «сущность–связь» (или ER-модель, от английского «Entity-Relationship») позволяет формализовать структуру и отношения между информационными объектами рассматриваемой предметной области.

При разработке модели предмет или класс предметов рассматривается как некоторая сущность, а элементы данных, которые описывают свойства предметов, – как атрибуты сущностей. Между сущностями устанавливаются связи, представляющие в модели отношения между объектами реального мира.

Базовыми элементами модели являются понятия:

- атрибута;
- сущности;
- связи.

Атрибут – логически неделимый элемент информации, отображающий некоторую количественную или качественную характеристику сущности.

Сущность является агрегацией атрибутов, характеризующих соответствующий ей объект или событие.

Между атрибутом и сущностью нет принципиальной разницы: один и тот же объект может рассматриваться и как сущность, и как атрибут – все определяется контекстом решаемых задач. Если элемент данных по условиям решаемых задач может рассматриваться как неделимый, то он будет представлен в модели атрибутом; в противном случае он должен рассматриваться как самостоятельная сущность.

Связь в модели указывает на наличие отношения между объектами, послужившими прототипами сущностей.

Каждое из понятий: атрибут, сущность, связь – характеризуется набором признаков.

Для сущности задается имя, существительное в единственном числе, уникальное в рамках модели. Оно должно быть кратким и вместе с тем давать исчерпывающее определение объекта.

При графическом изображении модели (на диаграмме) сущность обозначается прямоугольником с закругленными краями, разделенным на три части: в верхней части записывается имя сущности, в средней – имена ключевых атрибутов, в нижней – имена других атрибутов.

Для атрибута, как и для сущности, задается имя, которое также должно быть уникальным в рамках рассматриваемой сущности.

Кроме того, атрибут может быть:

- обязательным или необязательным;
- ключевым или не ключевым.

Атрибут является **обязательным**, если его значение должно быть обязательно задано при создании экземпляра сущности. При графическом изображении модели обязательные атрибуты помечаются символом «*».

Атрибут называется **ключевым**, если его значение используется для однозначной идентификации каждого экземпляра сущности. Ключ сущности может быть составным — образовываться не одним, а несколькими атрибутами данной сущности. Все ключевые атрибуты должны быть обязательными. При графическом изображении модели ключевые атрибуты помечаются символом «#».

Для каждого конца связи задаются:

- имя;
- степень множественности (участие в связи одного или нескольких экземпляров сущности);
- степень обязательности (обязательно или необязательно наличие связи между экземплярами сущностей).

Кроме того, указывается, является ли данная связь ключевой (участвует ли в образовании ключа сущности).

На диаграмме связь изображается ромбом и линиями, соединяющими сущности.

Имя связи должно отражать смысл отношения между объектами.

По степени множественности различают следующие виды связей:

- один к одному (1 : 1);
- многие к одному (М : 1);
- многие ко многим (М : М).

Обязательность связи показывает могут или не могут отдельные экземпляры сущностей находиться вне связи друг с другом. Обязательность для связи задается с двух сторон (как со стороны первой, так и со стороны второй участвующей в связи сущности).

Связь является **обязательной**, если экземпляр сущности на одном конце связи не может быть создан вне связи с экземпляром сущности, находящейся на другом конце связи.

Если экземпляр сущности на одном конце связи может быть создан вне зависимости от существования экземпляров связанной сущности, связь будет **необязательной**.

На диаграмме обязательные связи изображаются сплошными линиями, а необязательные – пунктирными.

Связь с одной из сторон может быть помечена как ключевая. Связь является **ключевой** в том случае, когда собственные атрибуты не позволяют однозначно идентифицировать экземпляры сущности, т. е. различить экземпляры сущности можно только в контексте экземпляра связанной сущности.

Для ключевых связей должны выполняться следующие требования:

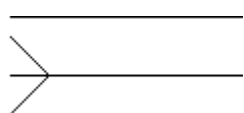
- связь может быть ключевой только с одной из сторон (со стороны одной из связанных сущностей);
- ключевой может быть только обязательная сторона связи;
- в случае связи “многие к одному” связь может быть ключевой только со стороны “многие”.

При графическом изображении ключевая связь помечается словом “key”.

ВАРИАНТЫ ОБОЗНАЧЕНИЯ ОСНОВНЫХ ХАРАКТЕРИСТИК СВЯЗИ

1) Множественность

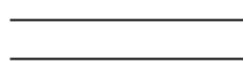
со стороны многие:



а) М

б)

со стороны единственный:

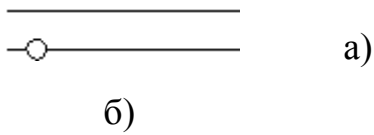


а) 1

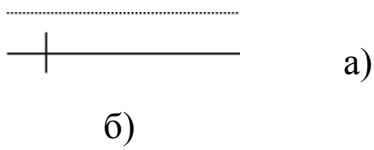
б)

2) Обязательность

обязательная связь:



необязательная связь:



3) Ключевые

а) key



Нормализация данных в ER-моделях

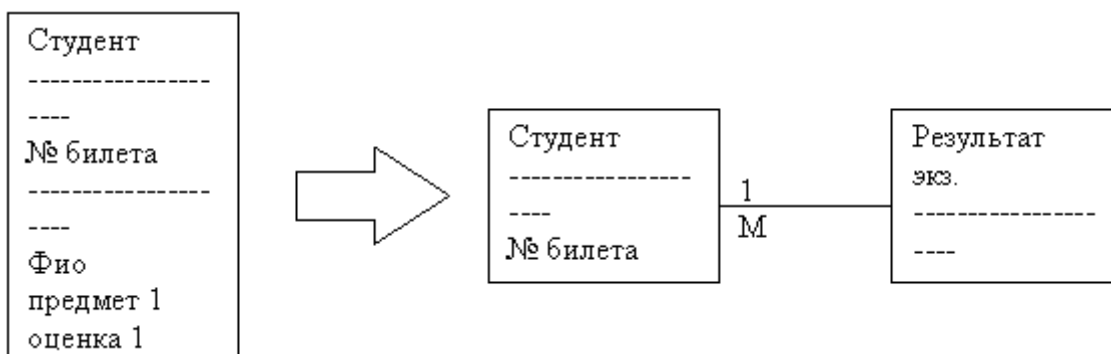
Нормализация данных – операция, которая позволяет обеспечить физическую реализуемость БД и уменьшить объем хранимой информации за счет исключения дублирования и так далее.

1) приведение к первой нормальной форме требуется для обеспечения физической реализуемости в виде реляционной БД.

Сущность находится в первой нормальной форме, если отсутствуют множественные, повторяющиеся атрибуты.

Приведение к первой нормальной форме состоит в исключении множественных и повторяющихся атрибутов с внесением их в новую сущность, с установкой связи 1:M между исходной и вновь образованной сущностью.

Например:



Данной связи недостаточно, нужно добавить ключевую связь для однозначной идентификации оценки за предмет (для студента).

2) Приведение ко второй нормальной форме применяется для уменьшения объема информации, хранимой в базе данных.

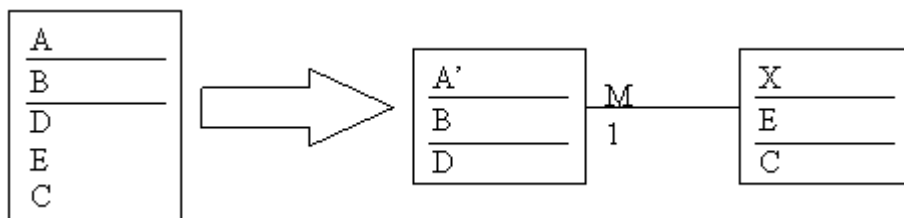
Сущность находится во второй нормальной форме, если она находится в первой нормальной форме, и нет неключевых атрибутов, нефункционально полно зависящих от составного ключа (т.е атрибуты которые используют 1,2 (не все) ключевой атрибут из 3 и более(составной ключевой атрибут)).

Приведение ко второй нормальной форме выполняется для сущности, находящейся в первой нормальной форме, и состоит в исключении неключевых атрибутов, нефункционально полно зависящих от составного ключа путем вынесения этих атрибутов и части составного ключа, от которого они зависят, в новую сущность; атрибуты, от которых зависят выносимые неключевые атрибуты, объявляются ключевыми в новой сущности. Между сущностями устанавливается связь М:1, М со стороны исходной сущности, связь помечается как ключевая.

3) Сущность находится в 3 нормальной форме, если она находится во второй нормальной форме и отсутствуют транзитивные зависимости между атрибутами.

При приведении к 3 нормальной форме атрибут, для которого имеет место транзитивная зависимость и атрибут, от которого он зависит, переносятся в новую сущность. Атрибут, от которого существует зависимость, становится ключевым в новой сущности, между сущностями устанавливается связь М:1, М со стороны исходной сущности.

Пример:



ВЫБОР КЛЮЧА

При выборе ключевых атрибутов следует руководствоваться требованиями эффективности, которые предполагают использование числовых ключей и во вторую очередь – существующими натуральными ключами в этой предметной области.

Натуральный ключ – ключ, который существует для данного понятия, без относительной программной реализации.

Список рекомендуемой литературы

Образец оформления титульного листа курсовой работы

Министерство образования и науки Российской Федерации

Северо-Кавказский горно-металлургический институт
(государственный технологический университет)

Кафедра Автоматизированной обработки информации

КУРСОВАЯ РАБОТА

по дисциплине

Программирование на языках высокого уровня

Выполнил:

Студент _____

Группа _____

Проверил:

Преподаватель: _____

Дата защиты _____

Оценка _____

Пример описания требований к автоматизированной информационной системе библиотеки

Библиотека располагает значительным фондом книг и журналов. Формирование фонда предусматривает:

- 1) заказ книг в соответствии с планами издательств;
- 2) покупку заказанных или имеющихся в продаже изданий;
- 3) подписку периодических журналов по каталогам издательств;
- 4) учет вновь поступивших книг и журналов в алфавитной и предметной картотеках.

Обслуживание читателей предполагает поиск заказанной литературы и запись соответствующей информации в формуляре читателя. Периодическая отчетность заключается в заполнении необходимых форм.

Требования

к справочно-информационной системе библиотеки

Справочно-информационная система библиотеки должна:

- хранить информацию об имеющихся в фонде библиотеки изданиях;
- обеспечивать возможность поиска необходимого издания: по названию, году издания, фамилии автора, тематике;
- предоставлять возможность дополнения или удаления информации о вновь поступивших или утерянных изданиях;
- давать возможность внесения соответствующих записей о дате выдачи и наименовании книги или журнала в формуляр читателя;
- выполнять в автоматизированном режиме заполнение бланков для заказа книг или подписки журналов;
- заполнять в автоматизированном режиме формы отчетности.

Пример описания разработки диаграмм потоков данных

ДПД начнем с построения контекстной диаграммы, являющейся диаграммой самого верхнего уровня. Она включает три внешние сущности: “ЧИТАТЕЛЬ”, “МАГАЗИН” и “ИЗДАТЕЛЬСТВО”. Вся деятельность библиотеки представим одним процессом БИБЛИОТЕКА. Контекстная диаграмма (см. рис. 4) отображает предварительные границы будущей системы.

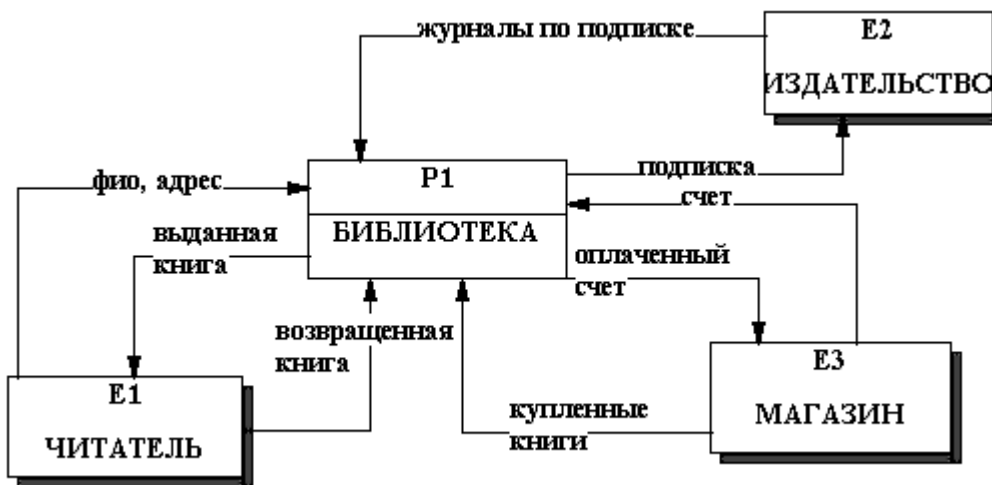


Рис.4. Контекстная диаграмма функционирования Библиотеки

На рисунке 4 проектируемая система представлена одной вершиной типа процесс и указаны потоки, связывающие ее с внешними сущностями: ЧИТАТЕЛЬ, МАГАЗИН, ИЗДАТЕЛЬСТВО.

Нашей следующей задачей является детализация процесса БИБЛИОТЕКА. Диаграмма потоков данных, конкретизирующая функционирование проектируемой системы приведена на рис 5.

Построение диаграммы ведется последовательно – от входа к выходу или наоборот. При этом необходимо начинать с события, побуждающего систему функционировать; в нашем случае — с регистрации читателя при выдаче ему книги. Процесс ВЫДАЧА КНИГ просматривает содержание накопителя КАРТОТЕКА ЧИТАТЕЛЕЙ с целью поиска формуляра читателя. Если читатель впервые обратился в

библиотеку, то выписывается новый формуляр, в котором фиксируются данные о читателе: фамилия, имя, домашний адрес, телефон, паспортные данные.

Читатель формулирует запрос на книгу или журнал, являющийся исходной информацией для процесса ПОИСК ИЗДАНИЯ. Процесс ПОИСК ИЗДАНИЯ просматривает накопитель КАТАЛОГ и в зависимости от результата поиска либо выдает отрицательный ответ, либо передает реквизиты найденного издания процессу ВЫДАЧА КНИГ.

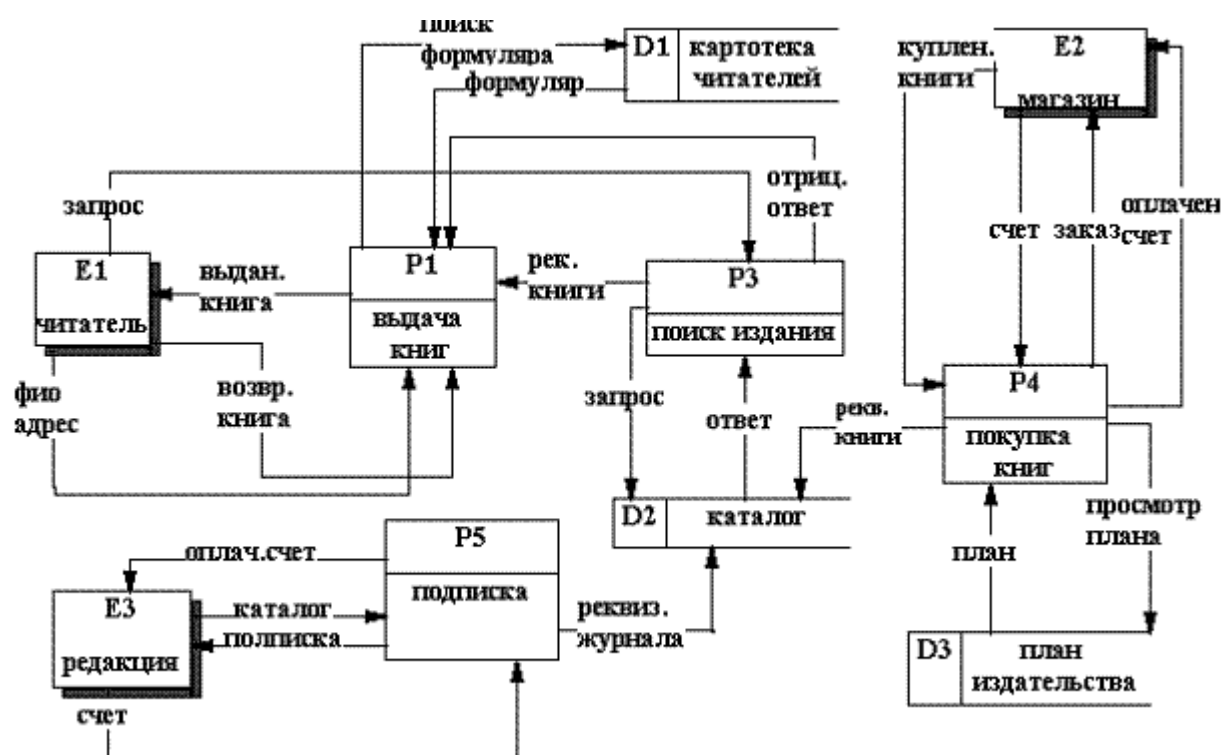


Рис. 5. Диаграмма потоков данных

Процесс ВЫДАЧА КНИГ делает отметку о выданном издании в формуляре читателя. При выдаче или возвращении книги читателем процесс ВЫДАЧА КНИГ фиксирует в накопителе КАТАЛОГ изменение количества хранимых изданий.

Формирование фонда библиотеки происходит при покупке книг и подписке журналов. Процесс ПОДПИСКА формирует список подписных изданий на основании каталогов издательств. Журналы поступают в библиотеку из РЕДАКЦИИ, являющейся внешней сущностью по отношению к системе. Процесс ПОДПИСКА фиксирует

информацию о новых поступлениях в накопителе КАТАЛОГ. Кроме того, процесс фиксирует информацию об оплате подписки.

Диаграммы потоков данных строятся безотносительно к последующей реализации, так, например, сведения о подписных изданиях могут храниться в файле или в виде бланков, что не имеет принципиального значения на данном этапе. Границы автоматизации будут обсуждаться на последующих этапах разработки системы.

Процесс ПОКУПКА КНИГ фиксирует информацию о купленных книгах и вносит ее в накопитель КАТАЛОГ. Покупку книг предваряет заказ их в соответствии с планами издательств, хранящимися в накопителе ПЛАН ИЗДАТЕЛЬСТВ. Информация о выставленных магазином и оплаченных счетах также фиксируется этим процессом.

Дальнейшая детализация диаграммы представлена на рисунках 6 и 7. Книги могут быть куплены не только по заказам в соответствии с планами издательств, но и в свободной продаже. Следовательно, возникает необходимость вынесения формирования заказа на покупку книг в отдельный процесс ЗАКАЗ КНИГ.

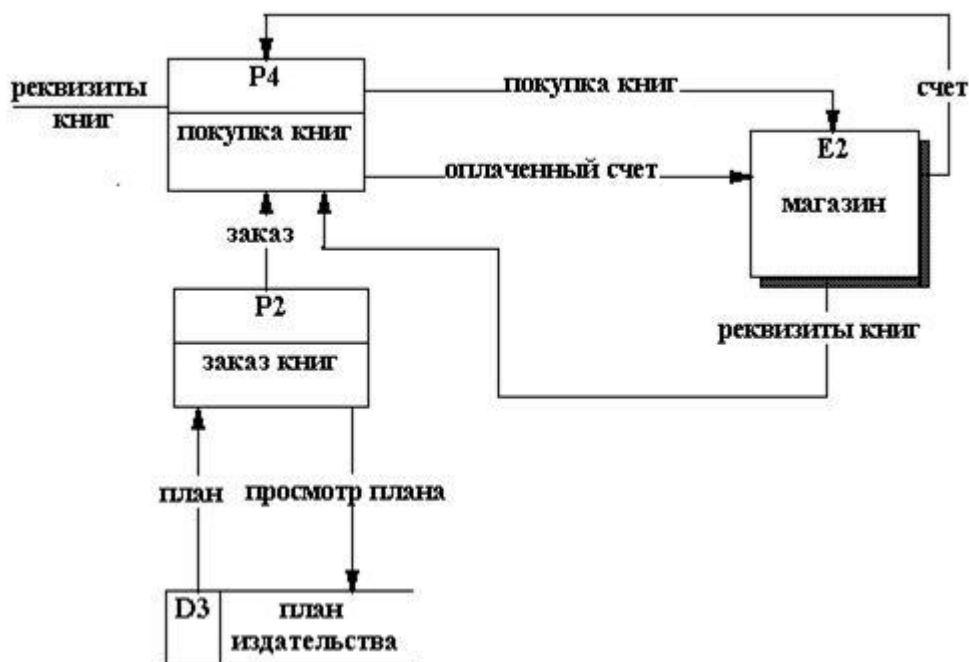


Рис.6. Детализация процесса покупки книг

Рассмотрим более детально процесс ВЫДАЧА КНИГ(см. рис.7).

Процедура выдачи книги или журнала начинается с поиска формуляра читателя в накопителе КАРТОТЕКА ЧИТАТЕЛЕЙ. Этот поиск проводит процесс ОФОРМЛЕНИЕ ЧИТАТЕЛЕЙ.

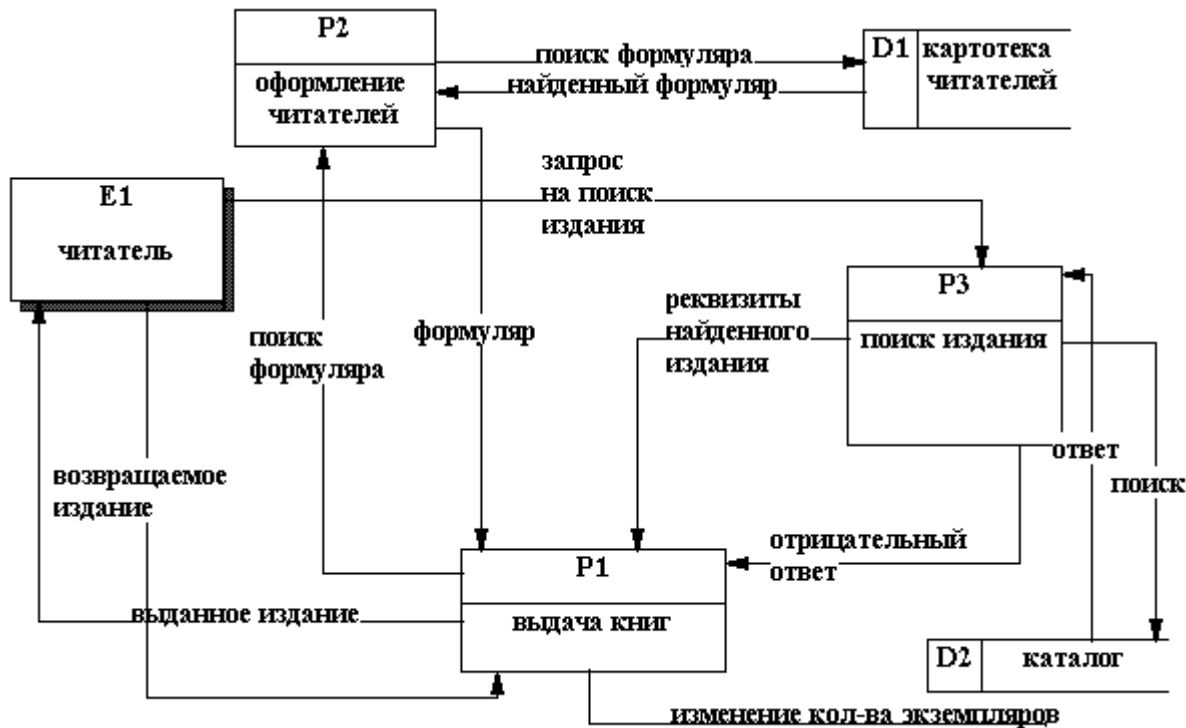


Рис.7. Детализация процесса выдачи книг и журналов

Окончательный вариант диаграммы потоков данных показан на рисунке 8.

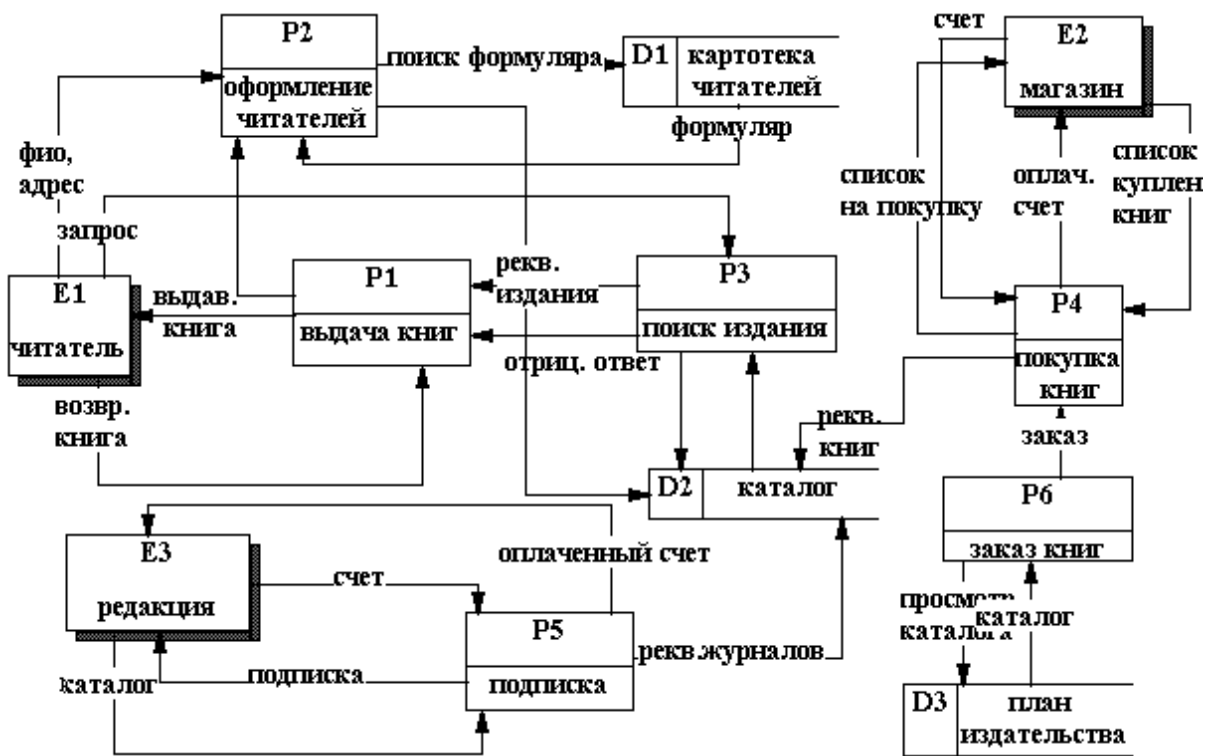


Рис. 8. Окончательный вариант диаграммы потоков

Пример описания словаря данных

Рассмотрим фрагмент словаря данных, созданного для информационной системы библиотеки:

КНИГА

ШИФР КНИГИ
 НАЗВАНИЕ
 ПЕРЕВОДЧИК
 РЕДАКТОР
 ТОМ
 НАЗВАНИЕ ТОМА
 КОЛИЧЕСТВО ЭКЗЕМПЛЯРОВ
 ЭКЗЕМПЛЯРОВ В НАЛИЧИИ
 АННОТАЦИЯ
 НОМЕР СТЕЛЛАЖА
 ЦЕНА

Имя элемента	Формат	Допустимый диапазон значений
Шифр книги	числовой	0-9999999
Название	текстовый	
Переводчик	текстовый	
Редактор	текстовый	
Том	числовой	
Название тома	текстовый	
Количество экземпляров	числовой	
Экземпляров в наличии	числовой	
Аннотация	текстовый	
Номер стеллажа	числовой	1–10
Цена	денежный	

Приведенное описание дает исчерпывающее представление о логической структуре заявки.

Пример модели «сущность - связь»

В нашем примере основными объектами являются:

- книга со всеми ее реквизитами,
- журнал,
- формуляр читателя,
- бланк подписки журналов,
- счет на покупку изданий в магазине.

Для построения информационной модели будущей системы, как можно более полно описывающей информационные особенности предметной области, используем стандартное средство моделирования — аппарат моделей “сущность–связь”, или ER – модель. Это позволит представить информационные потребности в виде, наглядном и удобном для восприятия.

Основными сущностями моделируемой области являются “КНИГА”, “ЖУРНАЛ”, “ЧИТАТЕЛЬ”, “СЧЕТ”, “ПОДПИСКА”, “РЕДАКЦИЯ”. Графическое изображение этих сущностей со всеми присущими им атрибутами представлено на рис. 9.

Для сущностей “КНИГА” и “ЖУРНАЛ” введем дополнительные атрибуты “Код книги” и “Код журнала”, возникающие при индексации каждого экземпляра книги или журнала в картотеке.

КНИГА
Код книги
Шифр книги
Название
Число страниц
Год издания
Переводчик
Редактор
Том
Название тома
Цена
В наличии экземпляров
Число экземпляров
Фамилия автора 1
Дата рождения автора 1
Тематика работ автора 1
...
Фамилия автора N
Дата рождения автора N
Тематика работ автора N

СЧЕТ
Номер счета
Название магазина
Адрес магазина
Расчетный счет магазина
Шифр книги 1
Название книги 1
Автор книги 1
Кол-во экземпляров книги 1
Цена книги 1
...
Шифр книги N
Название книги N
Автор книги N
Кол-во экземпляров книги N
Цена книги N
Общая стоимость заказа

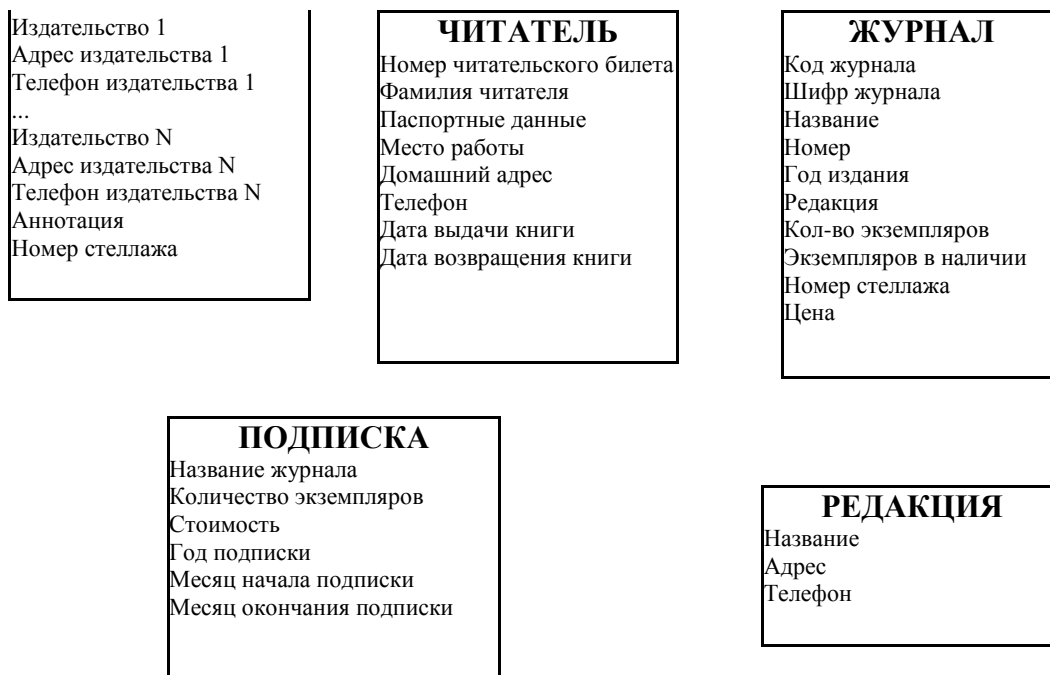


Рис. 9. Основные объекты инфологической модели библиотеки

Отдельные экземпляры каждой сущности должны быть различимы. Для их идентификации используют ключ, представляющий собой один из атрибутов сущности или комбинацию нескольких атрибутов. Для экземпляров сущности “КНИГА” ключевым атрибутом может быть атрибут “Код книги”, так как он уникален для каждой конкретной книги.

Для сущности “ЧИТАТЕЛЬ” в качестве ключевого атрибута можно использовать “Номер читательского билета”, так как среди читателей могут оказаться однофамильцы, а следовательно, атрибут “Фамилия читателя” не может являться

ключевым. Экземпляры сущности “СЧЕТ” могут быть идентифицированы значением атрибута “Номер счета”, “ЖУРНАЛ” — атрибутом “Код журнала”. Экземпляры сущности “РЕДАКЦИЯ” вполне различаются значением атрибута “Название редакции”, а для сущности “ПОДПИСКА” ключом может стать только комбинация атрибутов: “Название журнала”, “Год подписки”, “Месяц начала подписки” и “Месяц окончания подписки”.

Для атрибутов, не являющихся ключевыми, необходимо указать в модели степень их обязательности. Обязательность или необязательность атрибута каждой сущности определяется условиями предметной области.

Для сущности “КНИГА” значения атрибута “Дата рождения автора” являются желательными, но не обязательными, для сущности “ЧИТАТЕЛЬ” такой необязательной информацией является “Домашний телефон”.

Все остальные атрибуты являются обязательными.

Установим связи между сущностями, учитывая следующие закономерности:

- каждый читатель может заказать несколько книг, а отдельные экземпляры одной книги выдаются разным читателям; следовательно, между сущностями “КНИГА” и “ЧИТАТЕЛЬ” устанавливается связь типа М:М (многие ко многим);
- каждый читатель может заказать несколько журналов, а отдельные экземпляры журнала выдаются разным читателям; кроме того, один и тот же экземпляр может в разные моменты времени выдаваться разным читателям; следовательно, между сущностями “ЖУРНАЛ” и “ЧИТАТЕЛЬ” устанавливается связь типа М:М (многие ко многим);
- каждый счет может включать в себя оплату по нескольким книгам; с другой стороны, нельзя исключать случая многократной покупки одной и той же книги; следовательно, между сущностями “КНИГА” и “СЧЕТ” устанавливается связь типа М:М (многие ко многим);

- редакция может выпускать несколько журналов, на которые проводится подписка; в свою очередь, подписка на конкретный журнал, выпускаемый редакцией, проводится только в одной редакции; следовательно, между сущностями “ПОДПИСКА” и “РЕДАКЦИЯ” устанавливается связь типа 1:М (один ко многим, один со стороны сущности “ПОДПИСКА”), связь со стороны сущности “ПОДПИСКА” является ключевой, так как экземпляры сущности “ПОДПИСКА” не могут быть однозначно идентифицированы без связи с соответствующими экземплярами сущности “РЕДАКЦИЯ”;
- подписка на журналы проводится периодически, но на каждое издание выписывается один бланк; следовательно, между сущностями “ЖУРНАЛ” и “ПОДПИСКА” устанавливается связь типа 1:М (один ко многим, один со стороны сущности “ЖУРНАЛ”), связь со стороны сущности “ПОДПИСКА” является ключевой, так как экземпляры сущности “ПОДПИСКА” не могут быть однозначно идентифицированы без связи с соответствующими экземплярами сущности “ЖУРНАЛ”.

Связь отражает в модели отношение между объектами, послужившими прототипами сущностей. При переходе к физической структуре БД обработка связей приводит к копированию атрибутов связанных сущностей из таблицы, создаваемой для одной сущности, в таблицу, создаваемую для другой. Поэтому в эскизе модели, приведенном на рис. 9 после установления связей должны быть устранены “лишние” атрибуты. Так, например, связь между сущностями “КНИГА” и “СЧЕТ” позволяет устранить избыточные атрибуты для сущности “СЧЕТ” : “Шифр книги”, “Название книги”, “Автор книги”, “Издательство” и “Стоимость”, которые являются также атрибутами сущности “КНИГА”; для бланка подписки связь с сущностью “ЖУРНАЛ” позволяет не дублировать название журнала.

С учетом описанных связей и внесенных уточнений модель приобретет вид, показанный на рис. 10.

Связи типа «многие ко многим» (М:М) не могут быть представлены в реальной схеме данных. Преобразуем их в соответствии с правилами.

Экземпляр книги, выданный читателю, переходит из состояния “хранится в библиотеке” в состояние “находится у читателя” и может характеризоваться дополнительными атрибутами. С учетом этого обстоятельства, введем в модель сущность “КНИГА У ЧИТАТЕЛЯ” с атрибутами “Дата выдачи” и “Дата возвращения” и установим соответствующие связи с сущностями “КНИГА” и “ЧИТАТЕЛЬ”. Связи со стороны сущности “КНИГИ У ЧИТАТЕЛЯ” помечаются как ключевые, так как экземпляры этой сущности не могут быть однозначно идентифицированы вне связи с соответствующими экземплярами сущностей “КНИГА” и “ЧИТАТЕЛЬ”. Тип описываемой связи 1:М (многие со стороны сущности “КНИГИ У ЧИТАТЕЛЯ”) представлен на рис. 11.

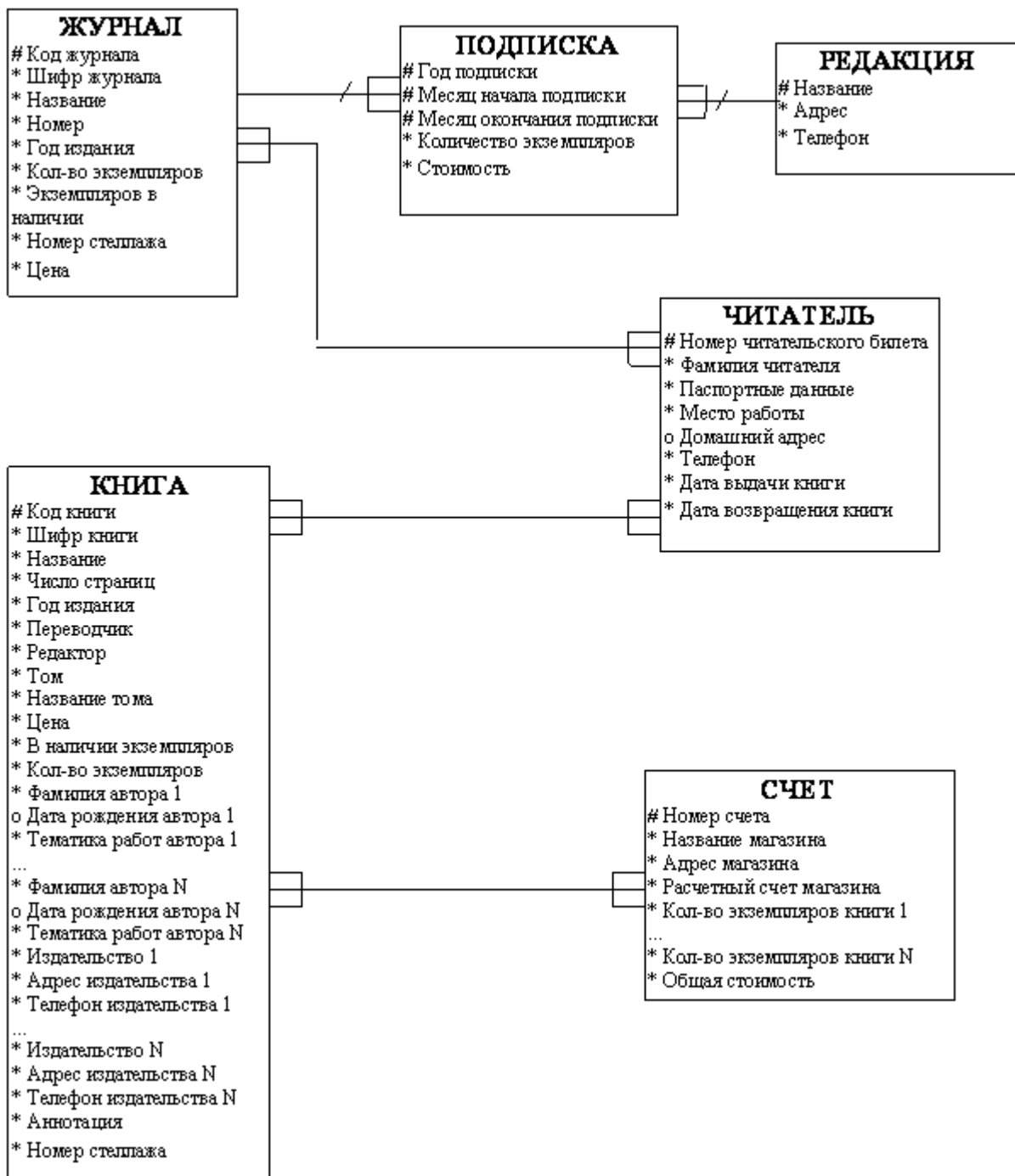


Рис.10. Установление связей между объектами модели библиотеки

Аналогичная ситуация возникает при анализе связи «журнал – читатель». Введение дополнительной сущности “ЖУРНАЛЫ У ЧИТАТЕЛЕЙ” позволит устранить связь “многие ко многим” (рис. 12).

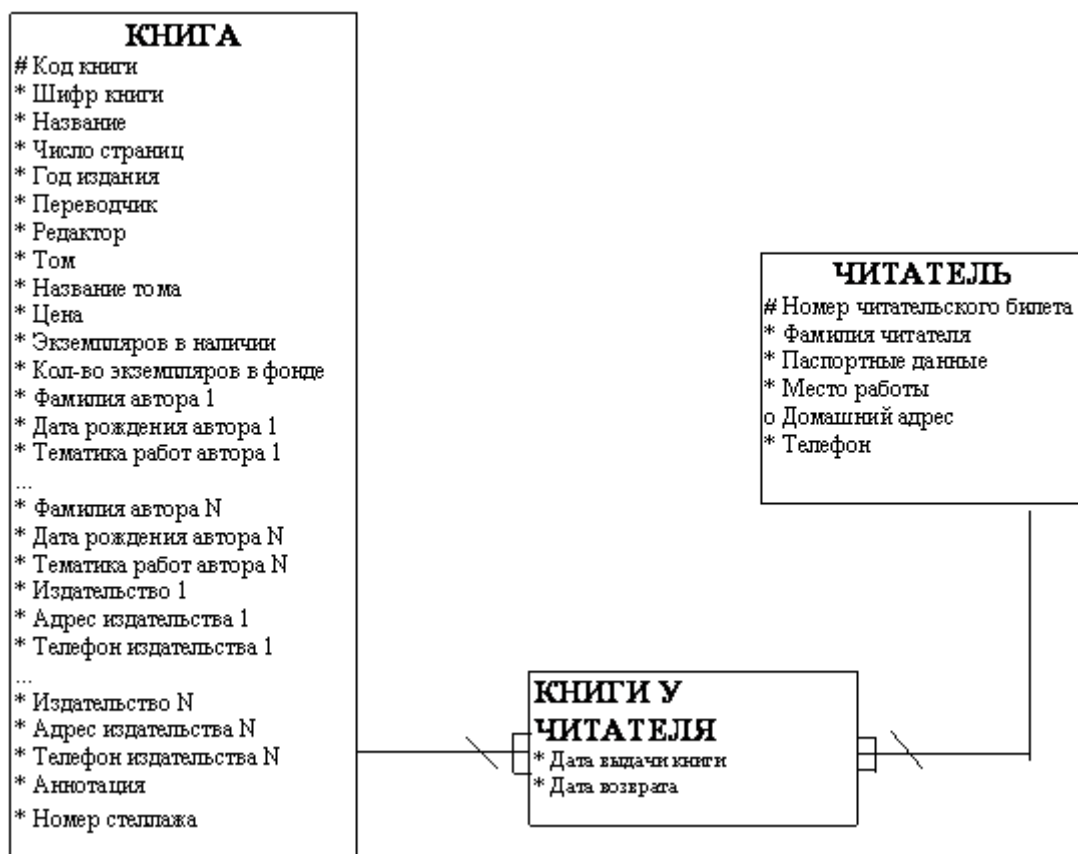


Рис.11 Разрешение связи “многие ко многим” между сущностями “КНИГА” и “ЧИТАТЕЛЬ”

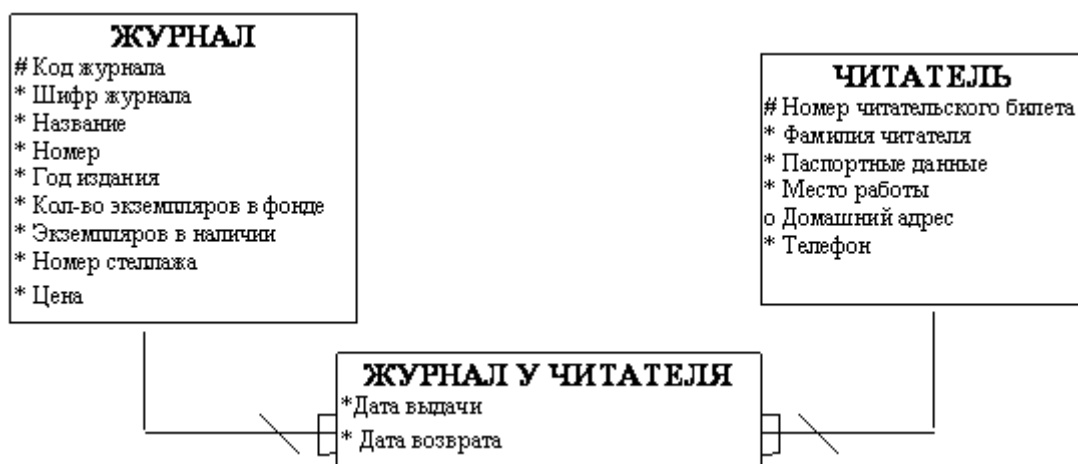


Рис. 12. Разрешение связи “многие ко многим” между сущностями “ЖУРНАЛ” и “ЧИТАТЕЛЬ”

Сущности “ЖУРНАЛ У ЧИТАТЕЛЯ” и “КНИГА У ЧИТАТЕЛЯ” могут быть сведены к одной — “ИЗДАНИЕ У ЧИТАТЕЛЕЙ”, если помимо атрибутов “Дата выдачи” и

“Дата возвращения” сущность “ИЗДАНИЕ У ЧИТАТЕЛЕЙ”, будет иметь атрибут, позволяющий отличать экземпляры книг от журналов, — “Книга или Журнал”.

Но тогда сущность “ИЗДАНИЕ У ЧИТАТЕЛЕЙ” будет иметь две взаимоисключающие связи с сущностями “ЖУРНАЛ” и “КНИГА”.

Ситуация, когда две или более связи от одной и той же сущности могут быть взаимоисключающими, на ER - диаграмме представляется с помощью исключаяющей дуги, пересекающей каждый конец взаимоисключающих связей (рис.13).

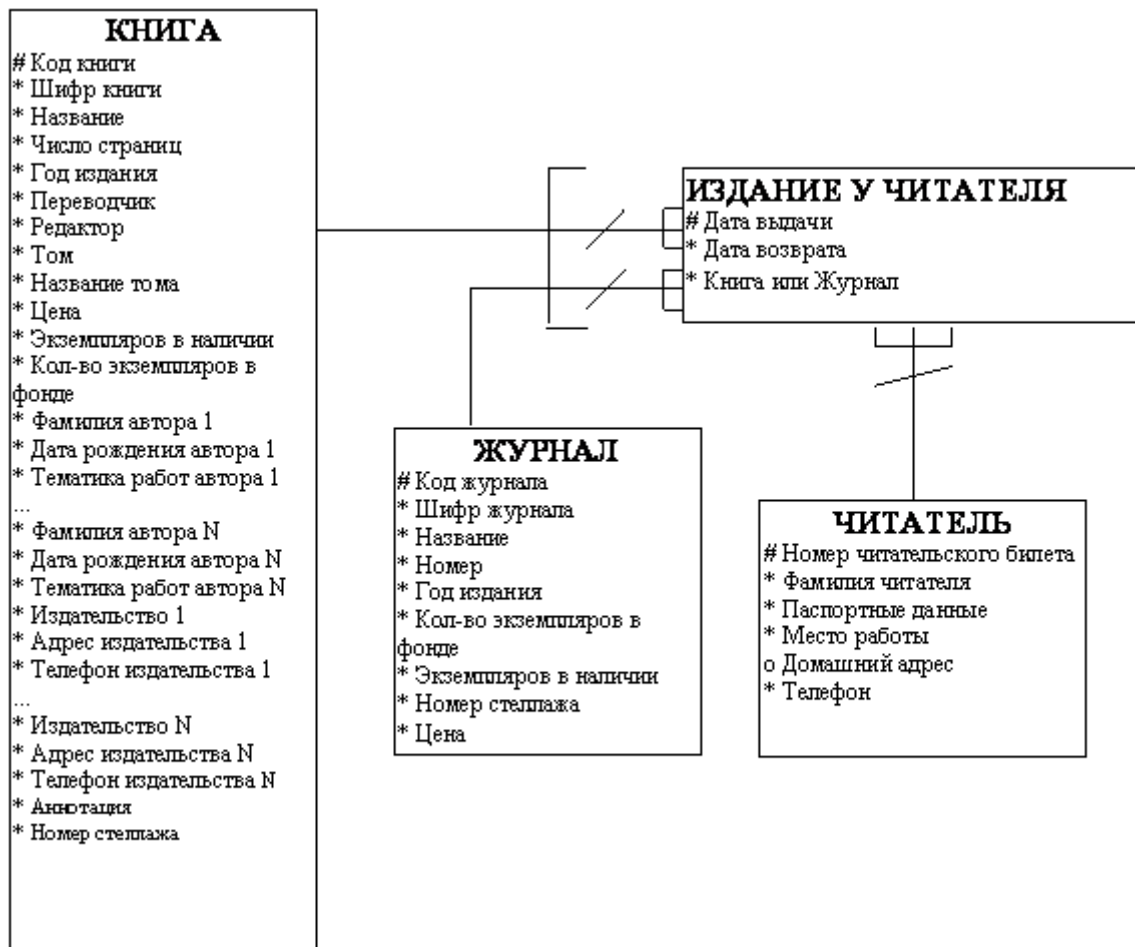


Рис.13. Исключающая дуга для связей между сущностями “КНИГА”, “ЖУРНАЛ” и сущностью “ИЗДАНИЕ У ЧИТАТЕЛЯ”

Между сущностями “КНИГА” и “СЧЕТ” существует связь типа М:М.

Счет представляет собой ряд пунктов, каждый из которых описывает реквизиты одного издания с указанием количества экземпляров; следовательно, для конкретизации отношений «книга-счет» необходимо ввести дополнительную

сущность “ПУНКТ СЧЕТА”, атрибутом которой станет “Количество экземпляров” каждого наименования.

Связь со стороны сущности “Пункт счета” помечается как ключевая, так как экземпляры этой сущности не могут быть однозначно идентифицированы вне связи с соответствующими экземплярами сущностей “КНИГА” и “СЧЕТ” (рис.14).

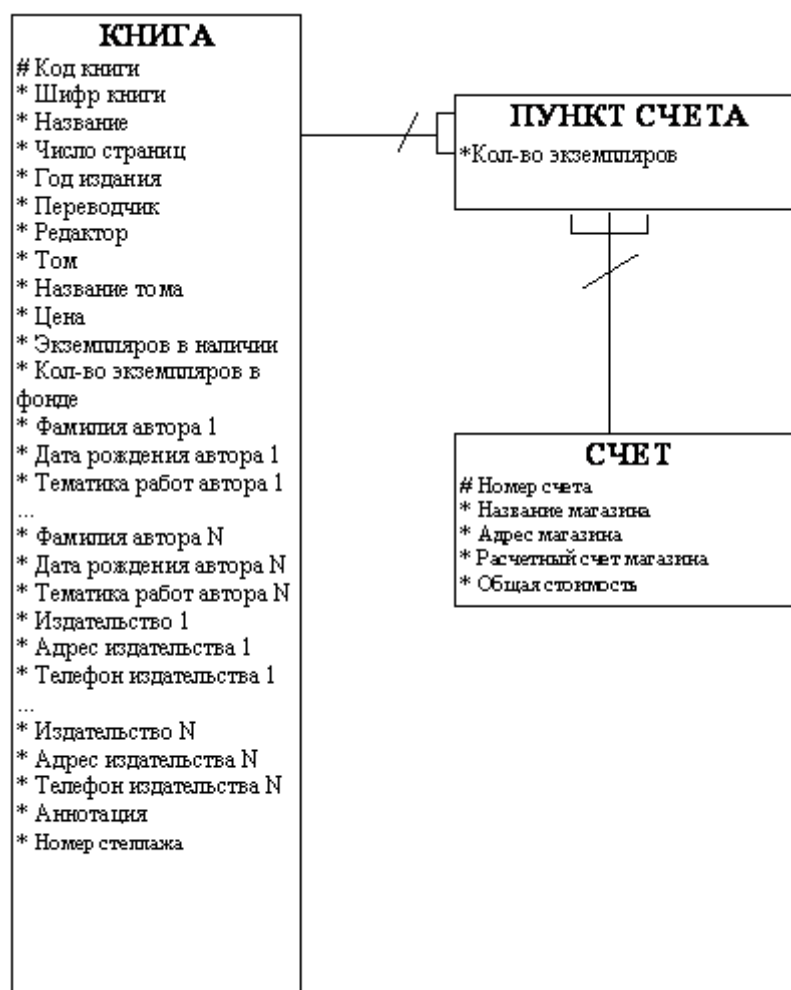


Рис. 14. Разрешение связи “многие ко многим” между сущностями “КНИГА” и “СЧЕТ”

Эскиз модели, представленный на рис.10, необходимо уточнить внесением всех рассмотренных изменений (см. рис. 11-14). Уточненный эскиз модели рассмотрен на рис.15.

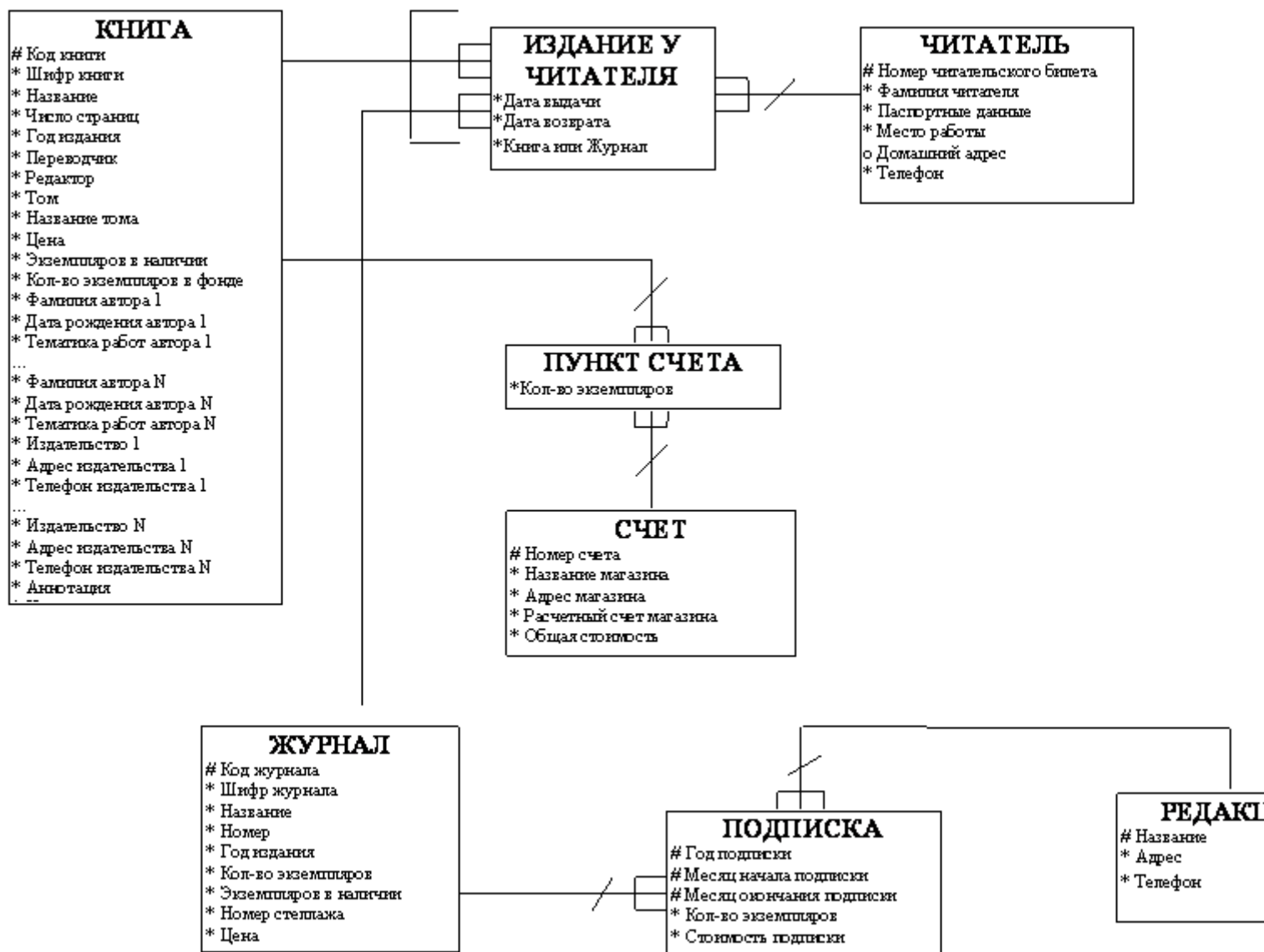


Рис. 15. Эскиз инфологической модели библиотеки с устраненными связями “многие ко многим”

Процесс нормализации модели позволяет устранить избыточность хранения информации и аномалии при добавлении, удалении, изменении данных.

В процессе нормализации модель должна быть последовательно приведена к 1НФ, 2НФ, 3НФ (нормальным формам).

Приведение к 1НФ состоит в исключении множественных или повторяющихся атрибутов.

Все сущности модели, кроме одной – “КНИГА”, соответствуют 1НФ, так как не имеют повторяющихся или множественных атрибутов, а сущность “КНИГА” должна быть приведена к 1НФ.

Книга может быть написана несколькими авторами, поэтому сущность “КНИГА” имеет повторяющиеся атрибуты “Фамилия автора”, “Дата рождения” и “Тематика работ автора”.

Повторяющиеся атрибуты выделяются в самостоятельную сущность “АВТОР”, атрибутами которой становятся “Фамилия автора”, “Дата рождения” и “Тематика работ” (рис. 16).

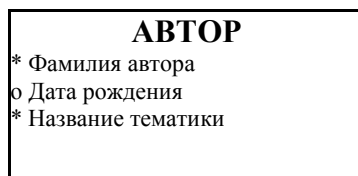


Рис. 16. Приведение к 1НФ сущности “КНИГА” – выделение понятия “АВТОР” в самостоятельную сущность

Для идентификации экземпляров сущности “АВТОР” одного атрибута “Фамилия” недостаточно, так как среди авторов могут оказаться однофамильцы.

Обсудив с библиографами вопрос идентификации авторов, приходим к выводу, что однофамильцы внутри одной тематики встречаются сравнительно редко. Следовательно, ключ сущности “АВТОР” можно определить как комбинацию двух атрибутов “Фамилия автора” и “Шифр тематики ” (рис. 17).

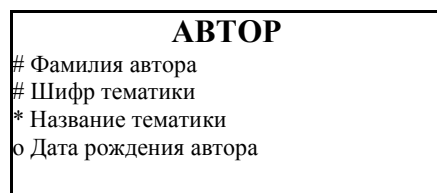


Рис. 17. Определение ключевых атрибутов сущности "Автор"

Между сущностями "КНИГА" и "АВТОР" устанавливается связь М:М, так как книга может быть написана группой авторов, а каждый автор может иметь более одной книги (рис. 18).

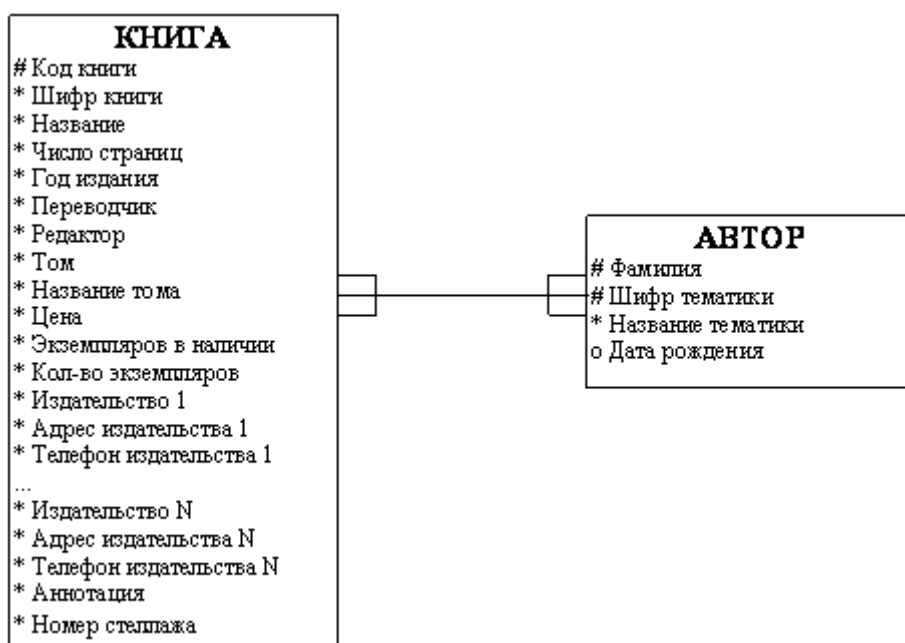


Рис. 18. Определение связи между сущностями

Для устранения такой общей множественной связи необходимо ввести дополнительную межсекционную сущность "АВТОР КНИГИ" (рис. 19).

Связь со стороны сущности "АВТОР КНИГИ" помечается как ключевая, так как экземпляры этой сущности идентифицируются только в связи с соответствующими экземплярами сущностей "КНИГА" и "АВТОР".

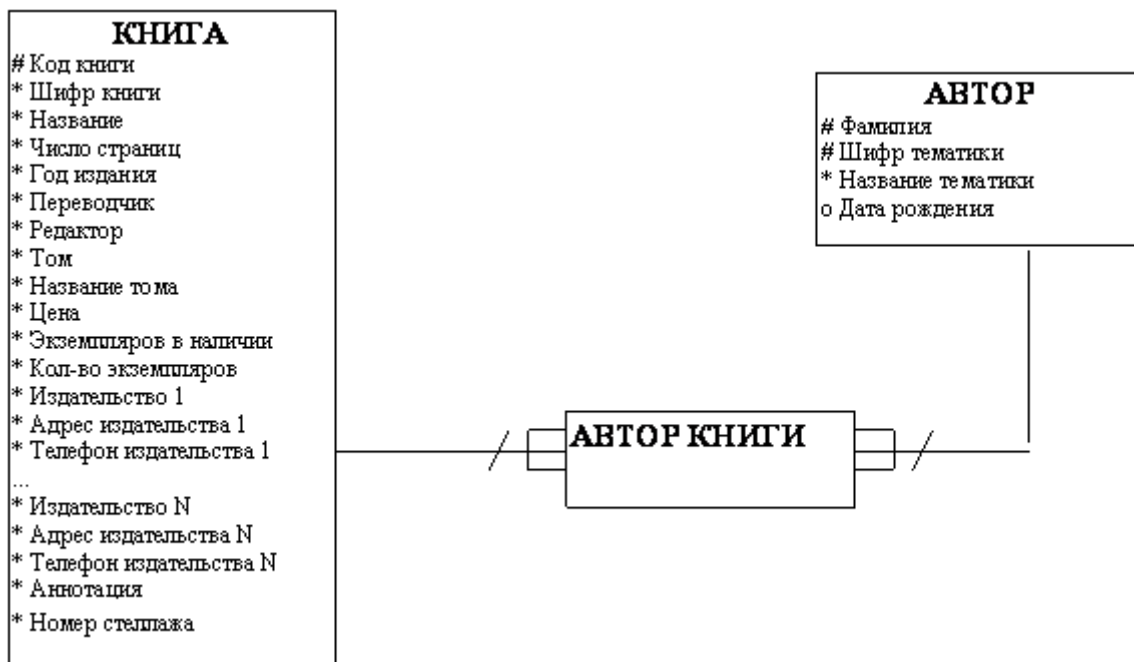


Рис. 19. Разрешение связи “многие ко многим” между сущностями “КНИГА” и “АВТОР”

Группа повторяющихся атрибутов “Издательство”, “Адрес издательства” и “Телефон издательства” выносится в самостоятельную сущность “Издательство”. Между сущностями устанавливается связь “многие ко многим” (M:M), так как книга может быть выпущена несколькими издательствами, с другой стороны, издательство выпускает множество различных книг (рис. 20).

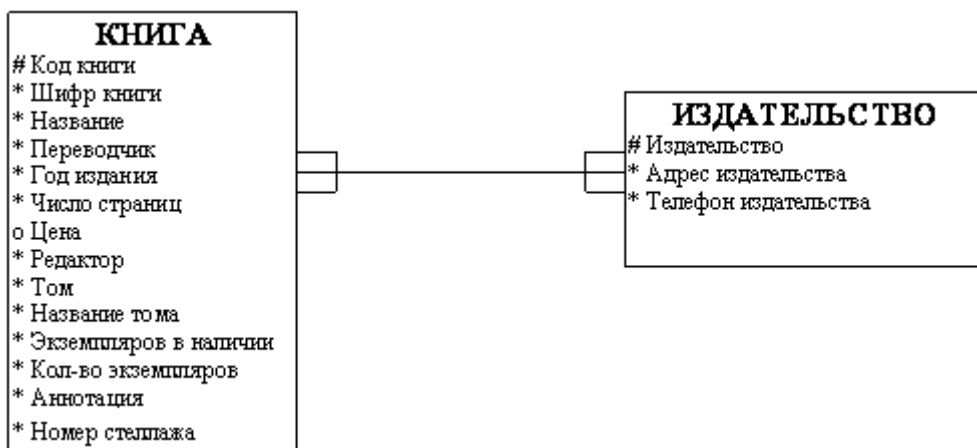


Рис. 20. Приведение к 1НФ сущности “КНИГА” – выделение понятия “ИЗДАТЕЛЬСТВО” в самостоятельную сущность

Для устранения множественной связи введем дополнительную сущность “ИЗДАНИЕ”, атрибутами которой становятся: “Число страниц”, “Год издания” и “Цена”; атрибут “Год издания” является ключевым (рис. 21).

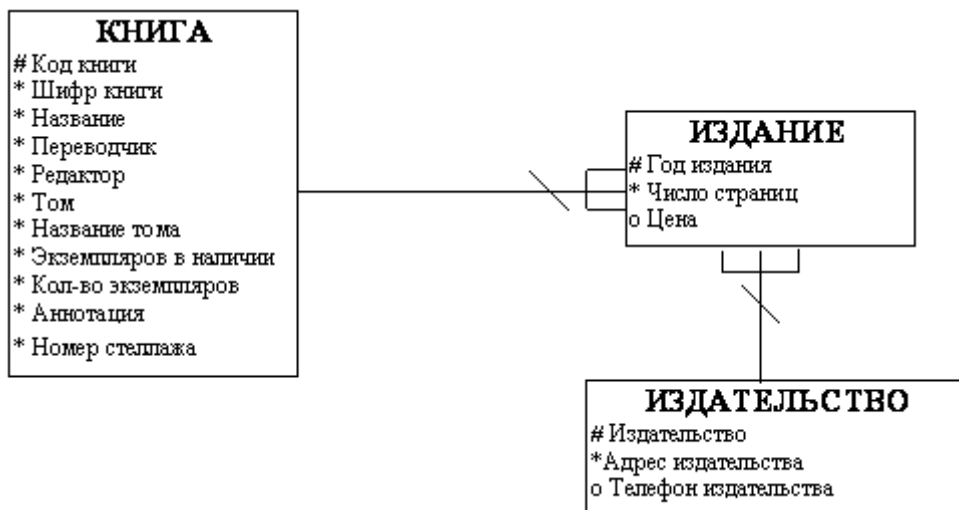


Рис. 21. Разрешение связи “многие ко многим” между сущностями “КНИГА” и “ИЗДАТЕЛЬСТВО”

Для всех сущностей модели были устранены повторяющиеся атрибуты, а следовательно, модель соответствует 1НФ (см. рис. 22).

Приведение модели ко 2НФ предполагает выделение атрибутов, которые удовлетворяют требованиям функционально полной зависимости от уникального идентификатора сущности, являющегося составным ключом.

В нашей модели сущность “АВТОР” имеет составной ключ (атрибуты “Фамилия автора” и “Шифр тематики”). Функциональная зависимость является полной, если не ключевой атрибут функционально зависит от составного ключа, но не зависит от любого подмножества атрибутов, образующих ключ. Для всех не ключевых атрибутов сущности “АВТОР” прослеживается функциональная зависимость не только от составного ключа, но и от отдельной его составляющей – атрибута “Фамилия автора”.

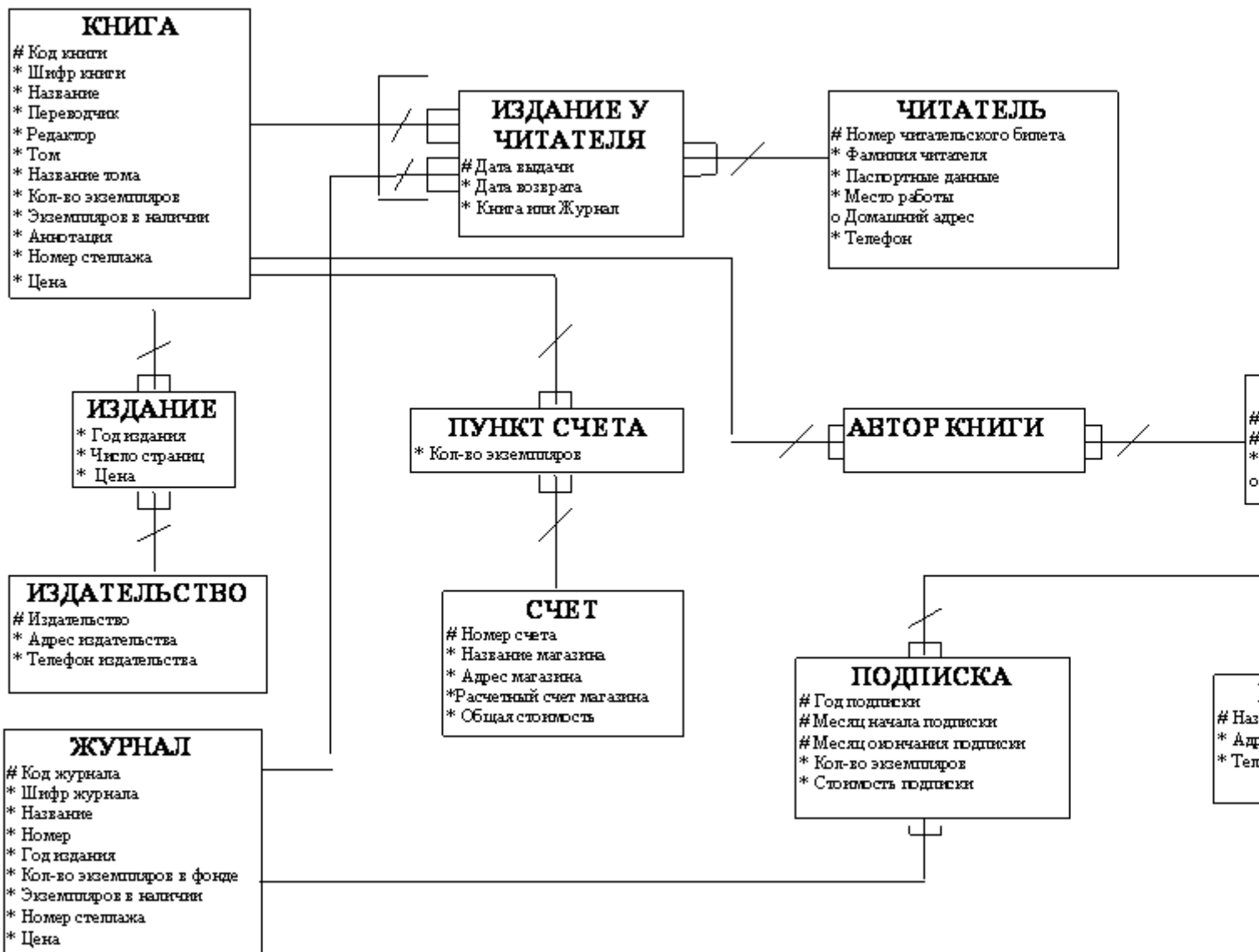


Рис. 22. Эскиз инфологической модели библиотеки, приведенной к 1НФ

Для приведения сущности ко 2НФ необходимо проделать следующие шаги:

- атрибуты, зависящие от части составного ключа и саму эту часть вынести в отдельную сущность, исключив из исходной;
- ключом новой сущности установить часть ключа исходной, от которой имеет место функционально неполная зависимость;
- между новой и исходной сущностями провести связь типа 1:М, ключевую со стороны М (рис. 23).

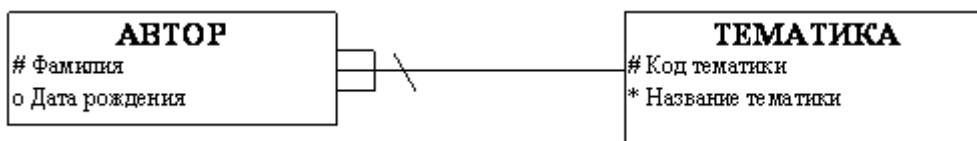


Рис. 23. Приведение сущности “АВТОР” к 2НФ

Ключевым атрибутом сущности “ТЕМАТИКА” становится атрибут “Код тематики”(см. рис. 23).

В качестве ключевого атрибута сущности “АВТОР” удобнее использовать атрибут “Код автора”, который возникает при сортировке книг одной тематики по фамилиям авторов (рис. 24).

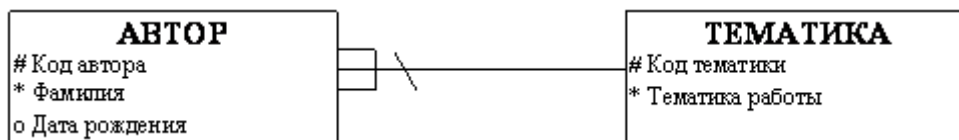


Рис. 24. Замена ключа сущности “АВТОР” на уникальный код

Сущность “ПОДПИСКА” соответствует 2НФ, так как не ключевые атрибуты “Стоимость подписки” и “Количество экземпляров” функционально полно зависят от составного ключа.

Модель, удовлетворяющая требованиям второй нормальной формы (2НФ), приведена на(рис. 25).

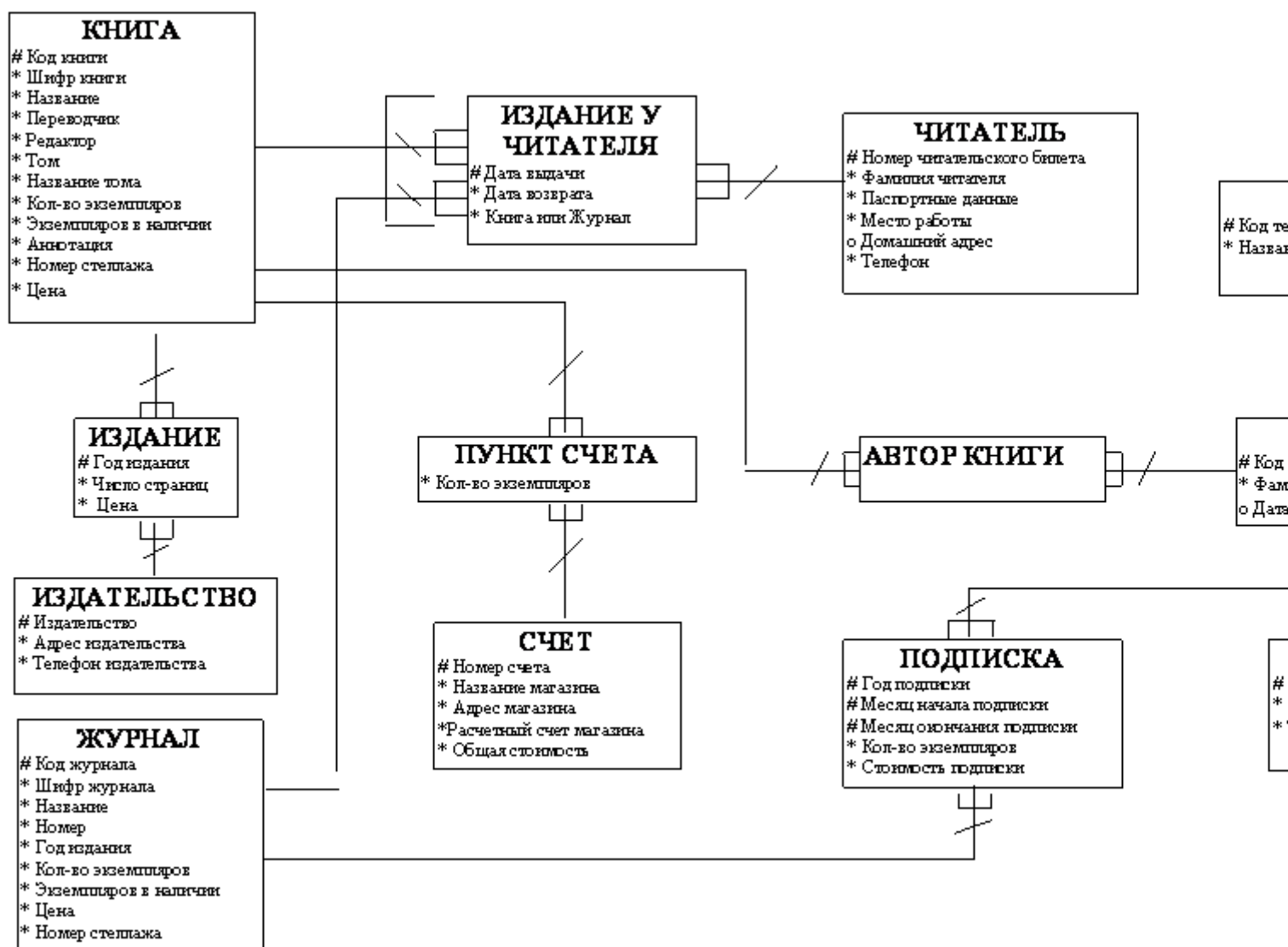


Рис. 25. Эскиз инфологической модели библиотеки, приведенной к 2НФ

Приведение к 3НФ состоит в исключении транзитивных зависимостей атрибутов, не являющихся частью ключа.

В сущностях “ЗАКАЗ” и “СЧЕТ” атрибут “Адрес магазина” транзитивно зависит от не ключевого атрибута “Название магазина”. В соответствии с правилами приведения к 3НФ выделим атрибуты “Название магазина” и “Адрес магазина” в самостоятельную сущность с именем “МАГАЗИН”. Ключевым атрибутом новой сущности является атрибут “Название магазина”, так как от него была обнаружена транзитивная зависимость. Между новой и исходными сущностями устанавливается связь типа 1:M (рис. 26).

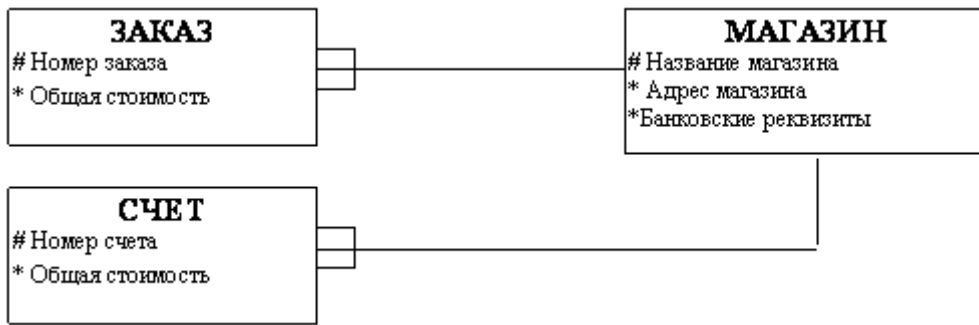


Рис. 26. Исключение транзитивной зависимости

В сущности “МАГАЗИН” атрибут “Банковские реквизиты” должен быть уточнен. В расчетном счете указываются атрибуты: название банка, адрес банка, расчетный счет магазина в банке, с указанием МФО и расчетного центра (рис. 27).

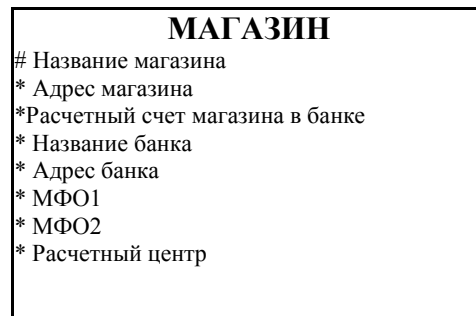


Рис. 27. Уточнение составного атрибута “Банковские реквизиты”
в сущности “МАГАЗИН”

Атрибуты “Расчетный счет магазина”, “Адрес банка”, “МФО 1”, “МФО 2” и “Расчетный центр” имеют транзитивную зависимость от не ключевого атрибута “Название банка”; следовательно, их необходимо вынести в отдельную сущность “БАНК”, ключевым атрибутом которой становится атрибут “Название банка”, от которого была обнаружена транзитивная зависимость (рис. 28).



Рис. 28. Приведение к 3НФ сущности “МАГАЗИН”

До сих пор мы не касались вопроса заказа книг в соответствии с планами издательств.

Реальные объекты: бланк заказа и перечень книг, предполагаемых к выпуску, представим в виде сущностей со всеми присущими им характеристиками.

Между сущностями установим связь типа “многие ко многим”, так как в заказе могут быть перечислены книги, относящиеся к различным издательствам, а план выпуска состоит из перечня изданий (см. рис. 29).

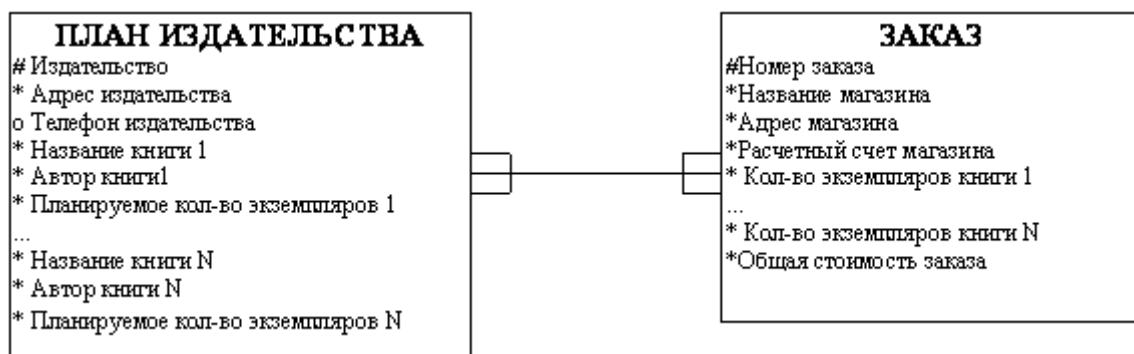


Рис. 29. Сущности “ПЛАН ИЗДАТЕЛЬСТВА” и “ЗАКАЗ”

и связь между ними

Анализ сущностей “ЗАКАЗ” и “ПЛАН ИЗДАТЕЛЬСТВА” показывает необходимость создания сущности “ВЫПУСКАЕМАЯ КНИГА” с атрибутами: “Шифр книги”, “Название книги”, “Автор книги”.

Связь со стороны сущности “ВЫПУСКАЕМАЯ КНИГА” помечается как ключевая, так как экземпляры этой сущности не могут быть однозначно идентифицированы без связи с соответствующими экземплярами сущностей “ЗАКАЗ” и “ПЛАН ИЗДАТЕЛЬСТВА”(рис. 30).

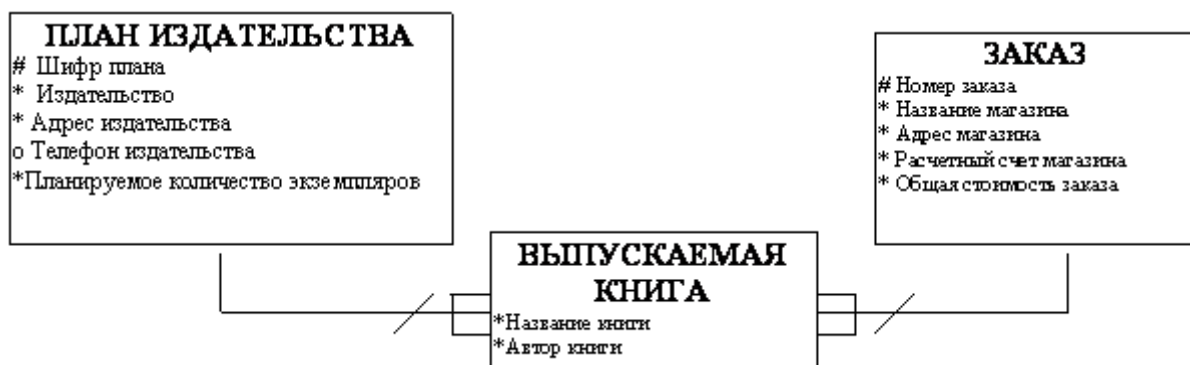


Рис. 30. Разрешение связи “многие ко многим”

ПЛАН ИЗДАТЕЛЬСТВА

Шифр плана
* Издательство
* Адрес издательства
o Телефон издательства
* Планируемое количество экземпляров

ПЛАН ИЗДАТЕЛЬСТВА

Шифр плана
* Издательство
* Адрес издательства
o Телефон издательства
* Планируемое количество
экземпляров

Атрибуты “Телефон издательства” и “Адрес издательства” сущности “ПЛАН ИЗДАТЕЛЬСТВА” транзитивно зависят от атрибута “Название издательства”. В соответствии с правилами приведения к ЗНФ необходимо выделить эти атрибуты в самостоятельную сущность. Ключевым атрибутом новой сущности становится атрибут, от которого была обнаружена транзитивная зависимость.

Наличие в модели сущности “ИЗДАТЕЛЬСТВО”, ключевым атрибутом которой является атрибут “Название издательства”, позволяет исключить транзитивную зависимость атрибутов сущности “ПЛАН ИЗДАТЕЛЬСТВА”. Между сущностями устанавливается связь типа 1:М, что отражает тот факт, что издательство имеет несколько тематических планов. Связь помечается как ключевая со стороны сущности “ПЛАН ИЗДАТЕЛЬСТВА”, так как экземпляры сущности “ПЛАН ИЗДАТЕЛЬСТВА” идентифицируются только в связи с экземплярами сущности “ИЗДАТЕЛЬСТВО” (рис.31).

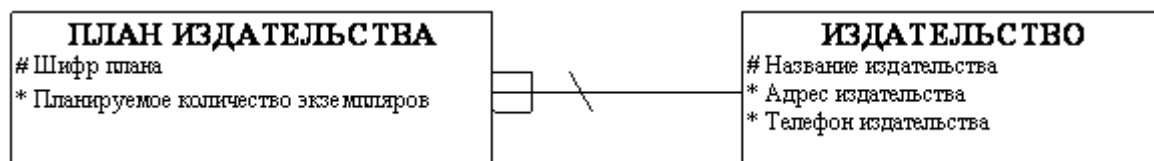


Рис. 31. Исключение транзитивной зависимости

Таким образом, для всех сущностей модели транзитивные зависимости не ключевых атрибутов исключены, а следовательно, модель соответствует 3НФ (рис. 32).

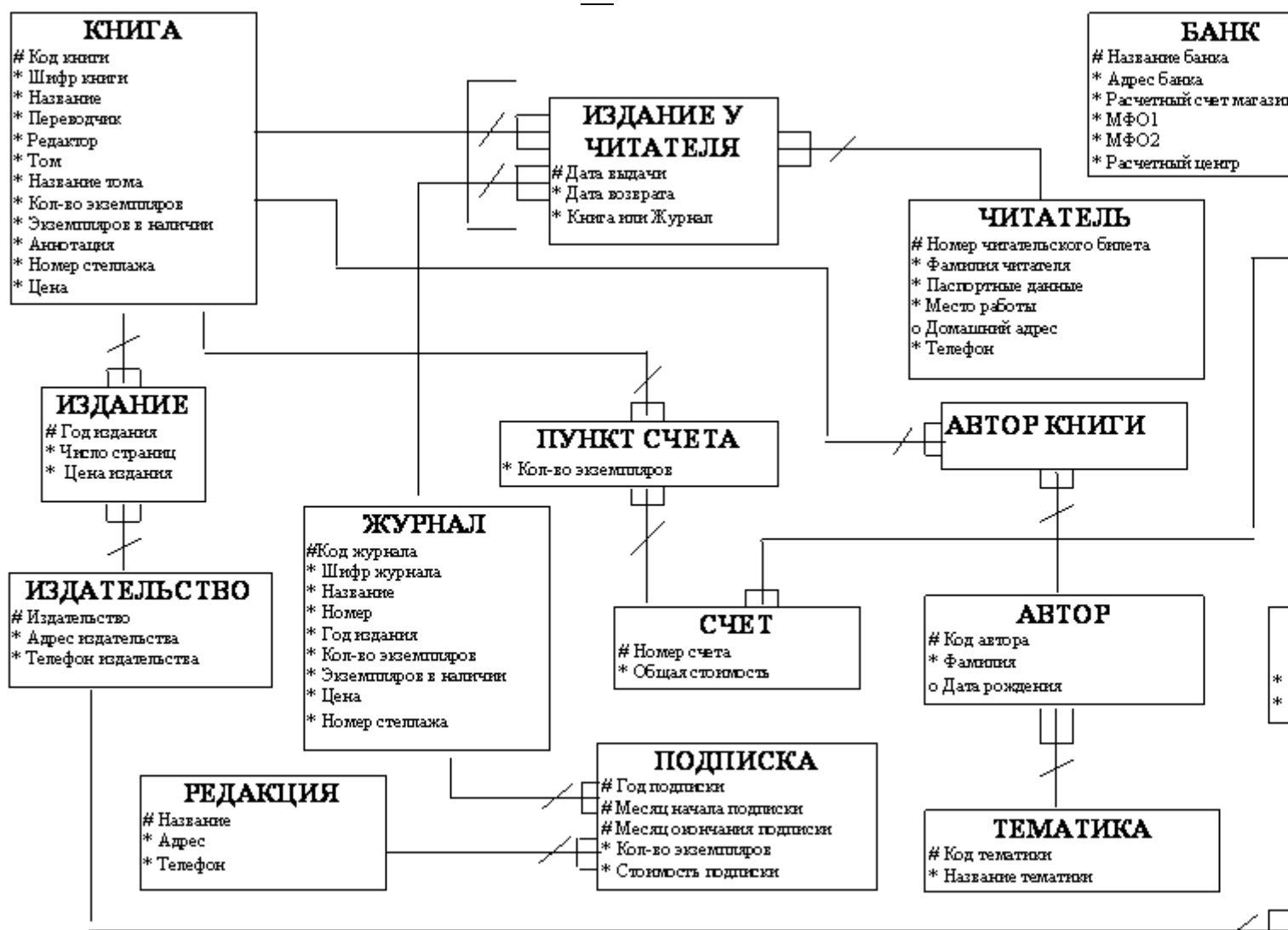


Рис. 32. Инфологическая модель библиотеки, удовлетворяющая требованиям 3НФ

Пример физической структуры базы данных

На основании модели “сущность - связь” генерируется предварительный проект базы данных в соответствии со следующими правилами:

- Каждая сущность преобразуется в таблицу, имя сущности становится именем таблицы.
- Атрибуты сущностей преобразуются в колонки таблицы, имена атрибутов становятся именами колонок таблицы.
- Ключевые атрибуты становятся первичными ключами таблицы.
- Если для сущности была определена ключевая связь, то первичный ключ связанной сущности копируется в таблицу и объединяется с ключом рассматриваемой сущности.
- Связи M:1 и 1:1 приводят к копированию ключей сущности, находящейся на другом конце связи. Если связь M:1, то ключи сущности, находящейся на конце “один”, копируются в таблицу для сущности, находящейся на конце “многие”.

В соответствии с перечисленными требованиями произведем генерацию таблиц реляционной базы данных для модели “Библиотека”.

КНИГА

Код книги	Шифр книги	Название	Переводч.	Редактор	Том	Название	Кол-во экземпляров	Экземпляр в наличии	Аннот.	Номер стеллажа	Цена

ИЗДАНИЕ

Код книги	Издательство	Год издания	Число страниц	Цена издания

ИЗДАТЕЛЬСТВО

Издательство	Адрес издательства	Телефон издательства

ЖУРНАЛ

Код журнала	Шифр журнала	Название	Номер	Год издания	Кол-во экземпляров	Экземплы в наличии	Цена	Номер стеллажа

ПОДПИСКА

Год подписки	Месяц начала подписки	Месяц окончания подписки	Код журнала	Название редакции	Кол-во экземпляров	Стоимость подписки	Шифр журнала

РЕДАКЦИЯ

Название редакции	Адрес редакции	Телефон редакции

ИЗДАНИЕ У ЧИТАТЕЛЯ

Номер читательского билета	Код книги	Код журнала	Дата выдачи	Дата возвращения	Книга или журнал

ЧИТАТЕЛЬ

Номер читательского билета	Фамилия читателя	Паспортные данные	Место работы	Домашний адрес	Телефон

АВТОР КНИГИ

Код книги	Код автора

АВТОР

Код автора	Код тематики	Фамилия автора	Дата рождения

ТЕМАТИКА

Код тематики	Название тематики

ПУНКТ СЧЕТА

Шифр книги	Номер счета	Количество экземпляров

СЧЕТ

Номер счета	Общая стоимость	Название магазина

БАНК

Название банка	Адрес банка	Расчетный счет магазина	МФО 1	МФО 2	Расчетный центр

МАГАЗИН

Название магазина	Адрес магазина	Телефон магазина

ЗАКАЗ

Номер заказа	Общая стоимость заказа	Название магазина

ВЫПУСКАЕМАЯ КНИГА

Шифр плана издательства	Номер заказа	Название книги	Автор книги

ПЛАН ИЗДАТЕЛЬСТВА

Шифр плана	Издательство	Планируемое количество экземпляров

Экранные формы для поиска информации и отчетов

Пример запроса для поиска в базе данных списка книг по конкретной тематике

Список книг по конкретной тематике, имеющихся в библиотеке : запрос на выборку

```

    graph LR
      Книга[Книга] ---|1-∞| Автор_книги[Автор книги]
      Автор_книги ---|∞-1| Автор[Автор]
      Автор ---|∞-1| Тематика[Тематика]
  
```

Книга

- Код книги
- Шифр книги
- Название книги
- Переводчик
- Редактор
- Том
- Название тома
- Количество экземпляров
- Экземпляры в наличии
- Аннотация
- Номер стеллажа
- Цена

Автор книги

- Код книги
- Код автора
- Код тематики

Автор

- Код автора
- Код тематики
- Фамилия автора

Тематика

- Код тематики
- Название темат

Поле:	Фамилия автора	Название тематик	Название книги		
Имя таблицы:	Автор	Тематика	Книга		
Сортировка:					
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Условие отбора:		[Тематика]			
или:					

Пример запроса для формирования списка читателей

Формуляр читателя : запрос на выборку

```

    graph LR
      Читатель[Читатель] ---|1-∞| Издание_у_читателя[Издание у читателя]
      Издание_у_читателя ---|∞-1| Книга[Книга]
  
```

Читатель

- Номер читательского билета
- Фамилия читателя
- Паспортные данные
- Место работы

Издание у читателя

- Номер читательского билета
- Код книги
- Код журнала
- Дата выдачи

Книга

- Код книги
- Шифр книги
- Название книги
- Переводчик

Поле:	Номер читательск	Фамилия читателя	Название кни	Дата выдачи	Дата возвращен
Имя таблицы:	Читатель	Читатель	Книга	Издание у читат	Издание у читате
Сортировка:					
Вывод на экран:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Условие отбора:		[Введите фамилию читателя]			
или:					

Пример отчета «Список читателей библиотеки»

<hr/> <i>Номер читательского билета</i>		1
<i>Фамилия читателя</i>	Анисимова Р.Л.	
<i>Паспортные данные</i>	X 1 - АК № 567865	
<i>Место работы</i>	ЗАО "Техника"	
<i>Домашний адрес</i>	СПб., Невский пр. 45 к	
<i>Телефон</i>	2138976	
<hr/> <i>Номер читательского билета</i>		2
<i>Фамилия читателя</i>	Белов В.В.	
<i>Паспортные данные</i>	XVI - АВ № 543421	
<i>Место работы</i>	ГОИ	
<i>Домашний адрес</i>	СПб., ул. Моховая, д.8	
<i>Телефон</i>	5123457	
<hr/> <i>Номер читательского билета</i>		3
<i>Фамилия читателя</i>	Карпов С.Г.	
<i>Паспортные данные</i>	IX - АК № 567876	
<i>Место работы</i>	ТОО "Электрон"	
<i>Домашний адрес</i>	СПб., Бухарестская ул	
<i>Телефон</i>	1234543	

Примерный перечень заданий

1. Автоматизированная система «Учебный процесс».
2. Автоматизированная система «Абитуриент».
3. Автоматизированная система «Кафедра».
4. Автоматизированная система «Обслуживание читателей в библиотеке».
5. Автоматизированная система «Служба занятости».
6. Автоматизированная система «Поликлиника».
7. Автоматизированная система «Регистрация участников конференции».
8. Автоматизированная система «Книга почтой».
9. Автоматизированная система «Гостиничный комплекс».
10. Автоматизированная система деканата для проведения сессии.
11. Автоматизированная система «Деловые контакты фирмы».
12. Автоматизированная система «Продажа-покупка акций».
13. Автоматизированная система «Обменный пункт валют».
14. Автоматизированная система «Туристическое агентство».
15. Автоматизированная система «Отдел сбыта предприятия».
16. Автоматизированная система «Складское хозяйство предприятия».
17. Автоматизированная система «Репертуар театров города».
18. Автоматизированная система «Выставочные залы города».
19. Автоматизированная система «Экскурсионное бюро».
20. Автоматизированная система «Отдел кадров предприятия».
21. Автоматизированная система «Агентство по продаже недвижимости».
22. Автоматизированная система «Страховая фирма».