**А. Э. Можарова**

# МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ ПО ИЗУЧЕНИЮ МОДУЛЯ «ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ С РЕЛЯЦИОННОЙ МОДЕЛЬЮ ДАННЫХ» В КУРСЕ «ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННЫХ СИСТЕМ»

ФГОС ВО направления подготовки 230700 «Прикладная информатика в образовании» определяет компетенции владения методами анализа предметной области (СПК-3 способен ставить и решать прикладные задачи с использованием современных информационно-коммуникационных технологий и СПК-4 способен моделировать и проектировать структуры данных и знаний, прикладные и информационные процессы), проектирования и разработки информационных систем в различных прикладных областях. Данные компетенции формируются в рамках дисциплины «Проектирование информационных систем». Данная дисциплина ориентирует на подготовку к организационно-управленческой, проектно-технологической и аналитической деятельности, способствует решению типовых задач профессиональной деятельности:

В настоящее время информационные системы, включающие базы данных, представляют собой одну из важнейших областей современных компьютерных технологий. С этой сферой связана большая часть современного рынка программных продуктов. Среди общих тенденций в развитии таких систем выделяются процессы интеграции и стандартизации, затрагивающие структуры данных и способы их обработки и интерпретации, системное и прикладное программное обеспечение, средства взаимодействия компонентов баз данных и многое другое.

В процессе изучения курса «Проектирование информационных систем» студенты осваивают технологии PHP и MySQL, создавать динамические Web-сайты. Данные технологии работают на разнообразных платформах и не требуют дополнительных сервисов. На сегодняшний день MySQL является наиболее эффективной системой управления реляционными данными, используемой для создания высококачественных баз данных и запросов к ним.

При организации занятий особое внимание уделяется подбору материала для примеров, а также заданий, необходимых для формирования специальных компетенций.

Каждый модуль курса сопровождается заданиями для самостоятельной работы в целях закрепления на практике теоретического материала.

В первом разделе студенты осваивают реляционную модель данных, формируют простейшие двумерные таблицы, связи (отношения).

**Пример теоретического блока по первому модулю.**

Структурными составляющими таблицы являются записи и поля. Каждая запись содержит информацию об отдельном объекте системы, а каждое поле - это определенные атрибуты объекта. Поля таблицы должны иметь несовпадающие имена.

Вид однотабличной реляционной БД .

| | Поле 1 | Поле 2 | Поле 3 | ... | Поле N |
|---|---|---|---|---|---|
| Запись 1 | | | | | |
| Запись 2 | | | | | |
| Запись 3 | | | | | |
| ... | | | | | |
| Запись m | | | | | |

Каждое поле таблицы имеет определенный тип. Тип - это множество значений, которые поле может принимать, и множество операций, которые можно выполнять над этими значениями. Существуют 4-ре вида основных типа для полей БД: символьный, числовой, логический и дата.

В каждой таблице БД необходимо наличие первичного ключа - так именуют поле или набор полей, однозначно идентифицирующий каждый экземпляр объекта или запись. Значение первичного ключа в таблице БД должно быть уникальным, т.е. в таблице не допускается наличие двух и более записей с одинаковыми значениями первичного ключа. Он должен быть минимально достаточным, а значит, не содержать полей, удаление которых не отразится на его уникальности.

В реляционных базах данных таблицы логически связаны друг с другом, первичные ключи как раз используются для однозначной организации такой связи. Межтабличная связь обеспечивает целостность данных. Связанные таблицы представляют собой единую базу данных, в которой можно создавать новые таблицы, а также запросы и отчеты, содержащие данные из связанных таблиц.

Внешним ключом называется поле таблицы, предназначенное для хранения значения первичного ключа другой таблицы с целью организации связи между этими таблицами.

Реляционные связи между таблицами баз данных

Между двумя или более таблицами базы данных могут существовать отношения подчиненности, которые определяют, что для каждой записи главной таблицы(называемой еще родительской) возможно наличие одной или нескольких записей в подчиненной таблице (называемой еще дочерней).Выделяют три разновидности связи между таблицами базы данных:

- "один-ко-многим";
- "один-к-одному";

- "многие-ко-многим".

Отношение "один-ко-многим"

Отношение "один-ко-многим" имеет место, когда одной записи родительской таблицы может соответствовать несколько записей дочерней. Связь "один-ко-многим" иногда называют связью "многие-к-одному". И в том, и в другом случае сущность связи между таблицами остается неизменной. Связь "один-ко-многим" является самой распространенной для реляционных баз данных.

Отношение "один-к-одному"

Отношение "один-к-одному" имеет место, когда одной записи в родительской таблице соответствует одна запись в дочерней. Это отношение встречается намного реже, чем отношение "один-ко-многим". Его используют например чтобы разделить таблицу, содержащую слишком много полей.

Отношение "многие-ко-многим"

Отношение "многие-ко-многим" применяется в следующих случаях:

- одной записи в родительской таблице соответствует более одной записи в дочерней;
- одной записи в дочерней таблице соответствует более одной записи в родительской.

Всякую связь "многие-ко-многим" в реляционной базе данных необходимо заменить на связь "один-ко-многим" (одну или более) с помощью введения дополнительных таблиц.

## 1.2 Проектирование баз данных

Проектирование базы данных (БД) - одна из наиболее сложных и ответственных задач, связанных с созданием информационной системы (ИС). В результате её решения должны быть определены содержание БД, эффективный для всех её будущих пользователей способ организации данных и инструментальные средства управления данными.

Процесс проектирования БД представляет собой последовательность переходов от неформального словесного описания информационной структуры предметной области к формализованному описанию объектов предметной области в терминах некоторой модели. В общем случае можно выделить следующие этапы проектирования:

1. Системный анализ и словесное описание информационных объектов предметной области.

2. Концептуальное (инфологическое) проектирование - построение семантической модели предметной области, то есть информационной модели наиболее высокого уровня абстракции. Такая модель создаётся без ориентации на какую-либо конкретную СУБД и модель данных. Термины «семантическая модель», «концептуальная модель» и «инфологическая модель» являются синонимами. Кроме того, в этом контексте равноправно могут использоваться слова «модель базы данных» и «модель предметной области» (например, «концептуальная модель базы данных» и «концептуальная модель предметной области»), поскольку такая модель является как образом реальности, так и образом проектируемой базы данных для этой реальности. Конкретный вид и содержание концептуальной модели базы данных определяется выбранным для этого формальным аппаратом. Обычно используются графические нотации, подобные ER-диаграммам.

Чаще всего концептуальная модель базы данных включает в себя:

- описание информационных объектов, или понятий предметной области и связей между ними.
- описание ограничений целостности, т.е. требований к допустимым значениям данных и к связям между ними.

3. Даталогическое (логическое) проектирование - создание схемы базы данных на основе конкретной модели данных, например, реляционной модели данных. Для реляционной модели данных даталогическая модель - набор схем отношений, обычно с указанием первичных ключей, а также «связей» между отношениями, представляющих собой внешние ключи.

4. Физическое проектирование - создание схемы базы данных для конкретной СУБД. Специфика конкретной СУБД может включать в себя ограничения на именование объектов базы данных, ограничения на поддерживаемые типы данных и т.п. Кроме того, специфика конкретной СУБД при физическом проектировании включает выбор решений, связанных с физической средой хранения данных (выбор методов управления дисковой памятью, разделение БД по файлам и устройствам, методов доступа к данным), создание индексов и т.д.

Условно процесс проектирования БД можно представить последовательностью выполнения четырех соответствующих этапов.

В процессе изучения первого модуля студенты осваивают технологию системного анализа предметной области, теоретический материал для темы формируется с опорой на изученные дисциплины.

**Пример теоретического блока по теме «Технология системного анализа предметной области»**

Первый этап проектирования БД это системный анализ , то есть подробное словесное описание объектов предметной области и реальных связей, которые присутствуют между описываемыми объектами.

1. Функциональный подход к проектированию БД.

Этот метод является наиболее распространённым. Он реализует принцип "от задач" и применяется в том случае, когда известны функции некоторой группы лиц и комплекса задач, для обслуживания информационных потребностей которых создаётся рассматриваемая БД.

2. Предметный подход к проектированию БД.

Предметный подход применяется в тех случаях, когда у разработчиков есть чёткое представление о самой ПО 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 ПО и наиболее адекватному её отображению в БД с учётом самого широкого спектра информационных запросов к ней.

Чаще всего на практике рекомендуется использовать некоторый компромиссный вариант, который, с одной стороны, ориентирован на конкретные задачи или функциональные потребности пользователей, а с другой стороны, учитывает возможность наращивания новых приложений. Системный анализ должен заканчиваться подробным описанием информации об объектах предметной области, которая требуется для решения конкретных задач и которая должна храниться в БД, формулировкой конкретных задач, которые будут решаться с использованием данной БД с кратким описанием алгоритмов их решения, описанием выходных документов, которые должны генерироваться в системе, описанием входных документов, которые служат основанием для заполнения данными БД.

Пример описания предметной области

Пусть требуется разработать информационную систему для автоматизации учета получения и выдачи книг в библиотеке. Система должна предусматривать режимы ведения системного каталога, отражающего перечень областей знаний, по которым имеются книги в библиотеке. Внутри библиотеки области знаний в систематическом каталоге могут иметь уникальный внутренний номер и полное наименование.

Каждая книга может содержать сведения из нескольких областей знаний. Каждая книга в библиотеке может присутствовать в нескольких экземплярах. Каждая книга, хранящаяся в библиотеке, характеризуется следующими параметрами:

- уникальный шифр;
- название;
- фамилии авторов;
- место издания (город);
- издательство;
- год издания;
- количество страниц;
- количество экземпляров;
- номер области знаний;
- название области знаний.

Книги могут иметь одинаковые названия, но они различаются по своему уникальному шифру (ISBN).

В библиотеке ведется картотека читателей.

На каждого читателя в картотеку заносятся следующие сведения:

- фамилия, имя, отчество;
- пол
- домашний адрес;
- телефон;
- дата рождения.

Каждому читателю присваивается уникальный номер читательского билета. Каждый читатель может одновременно держать на руках не более 5 книг. Читатель не должен одновременно держать более одного экземпляра книги одного названия.

Каждая книга в библиотеке может присутствовать в нескольких экземплярах. Каждый экземпляр имеет следующие характеристики:

- уникальный инвентарный номер;
- шифр книги, который совпадает с уникальным шифром из описания книг;
- место размещения в библиотеке.

В случае выдачи экземпляра книги читателю в библиотеке хранится специальный вкладыш, в котором должны быть записаны следующие сведения:

- номер билета читателя, который взял книгу;
- дата выдачи книги;
- дата возврата.

*При работе с системой библиотекарь должен иметь возможность решать следующие задачи:*

1. Принимать новые книги и регистрировать их в библиотеке.

2. Относить книги к одной или к нескольким областям знаний.

3. Проводить каталогизацию книг, то есть назначение новых инвентарных номеров вновь принятым книгам, и, помещая их на полки библиотеки, запоминать место размещения каждого экземпляра.

4. Проводить дополнительную каталогизацию, если поступило несколько экземпляров книги, которая уже есть в библиотеке, при этом информация о книге в предметный каталог не вносится, а каждому новому экземпляру присваивается новый инвентарный номер и для него определяется место на полке библиотеки.

5. Проводить списание старых и не пользующихся спросом книг. Списывать можно только книги, ни один экземпляр которых не находится у читателей. Списание проводится по специальному акту списания, который утверждается администрацией библиотеки.

6. Вести учет выданных книг читателям, при этом предполагается два режима работы: выдача книг читателю и прием от него возвращаемых им книг обратно в библиотеку. При выдаче книг фиксируется, когда и какой экземпляр книги был выдан данному читателю и к какому сроку читатель должен вернуть этот экземпляр книги. При выдаче книг наличие свободного экземпляра и его конкретный номер могут определяться по заданному уникальному шифру книги или инвентарный номер может быть известен заранее. Не требуется вести «историю» чтения книг, то есть требуется отражать только текущее состояние библиотеки.

7. Проводить списание утерянных читателем книг.

8. Проводить закрытие абонемента читателя, то есть уничтожение данных о нем, если читатель хочет выписаться из библиотеки и не является ее должником, то есть за ним не числится ни одной библиотечной книги.

*Читатель должен иметь возможность решать следующие задачи:*

1. Просматривать системный каталог, то есть перечень всех областей знаний, книги по которым есть в библиотеке.

2. По выбранной области знаний получить полный перечень книг, которые числятся в библиотеке.

3. Для выбранной книги получить инвентарный номер свободного экземпляра книги или сообщение о том, что свободных экземпляров книги нет. В случае отсутствия свободных экземпляров книги читатель должен иметь возможность узнать дату ближайшего предполагаемого возврата экземпляра данной книги. Читатель не может узнать данные о том, у кого в настоящий момент экземпляры данной книги находятся на руках (в целях обеспечения личной безопасности держателей требуемой книги).

4. Для выбранного автора получить список книг, которые числятся в библиотеке.

Этот пример показывает, что перед началом разработки необходимо иметь точное представление о том, что же должно выполняться в системе, какие пользователи в ней будут работать, какие задачи будет решать каждый пользователь. Отсутствие четких целей создания БД может свести на нет все усилия разработчиков, и проект БД получится «плохим», неудобным, не соответствующим ни реально моделируемому объекту, ни задачам, которые должны решаться с использованием данной БД.

==========

При изучении темы «Инфологическое проектирование» студенты осваивают ER-моделирование, реализованное в разных средах.

**Пример теоретического блока по теме «Инфологическое проектирование».**

Инфологическая модель применяется на втором этапе проектирования БД , то есть после словесного описания предметной области и постановки задачи. Инфологическая модель должна включать такое формализованное описание предметной области, которое будет «читабельно» не только для специалистов по базам данных, но и сторонних людей. Описание должно быть настолько емким, чтобы можно было оценить глубину и корректность проработки проекта БД, и не должно быть привязано к конкретной СУБД.

ER-диаграмма (entity-relationship diagram, диаграмма «сущность-связь») - это графическое представление инфологической модели.

Основные элементы ER-моделей:

- сущности (объекты);

- атрибуты сущностей;
- ключ сущности
- связи между сущностями.

Сущность - это класс однотипных объектов, информация о которых должна быть учтена в модели. Каждая сущность должна иметь наименование, выраженное существительным в единственном числе. Каждая сущность в модели изображается в виде прямоугольника с наименованием.

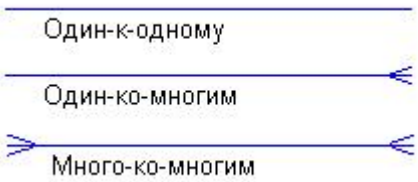
Атрибут сущности - это именованная характеристика, являющаяся некоторым свойством сущности. Наименование атрибута должно быть выражено существительным в единственном числе (возможно, с характеризующими прилагательными). Примерами атрибутов сущности "Книга" могут быть такие атрибуты как "ISBN", "Название", "Автор", "Издательство", "Место издания", "Год издания", "Количество страниц", "Количество экземпляров". Атрибуты изображаются в пределах прямоугольника, определяющего сущность:

Ключ сущности - это неизбыточный набор атрибутов, значения которых в совокупности являются уникальными для каждого экземпляра сущности. Неизбыточность заключается в том, что удаление любого атрибута из ключа нарушается его уникальность. Сущность может иметь несколько различных ключей. Ключевые атрибуты изображаются на диаграмме подчеркиванием:

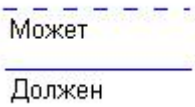
Связь - это некоторая ассоциация между двумя сущностями. Одна сущность может быть связана с другой сущностью или сама с собою.

Связи позволяют по одной сущности находить другие сущности, связанные с нею. Графически связь изображается линией, соединяющей две сущности. (рис ) Каждая связь имеет два конца и одно или два наименования. Наименование обычно выражается в неопределенной глагольной форме: "иметь", "принадлежать" и т.п. Каждое из наименований относится к своему концу связи. Иногда наименования не пишутся ввиду их очевидности.

Каждая связь может иметь один из следующих типов связи:




Каждая связь может иметь одну из двух модальностей связи:




Модальность "может" означает, что экземпляр одной сущности может быть связан с одним или несколькими экземплярами другой сущности, а может быть и не связан ни с одним экземпляром.

Модальность "должен" означает, что экземпляр одной сущности обязан быть связан не менее чем с одним экземпляром другой сущности

Например между сущностями «Читатель» и «Экземпляр» установлена связь «один-ко-многим», и при этом она не обязательная с двух сторон. Читатель в данный момент может не держать ни одной книги на руках, а с другой стороны, данный экземпляр книги может не находиться ни у одного читателя, а просто стоять на полке в библиотеке.

Так как мы выделили три основные сущности «Книга», «Экземпляр» и «Читатель» получилась инфологическая модель, представленная на рисунке 1.

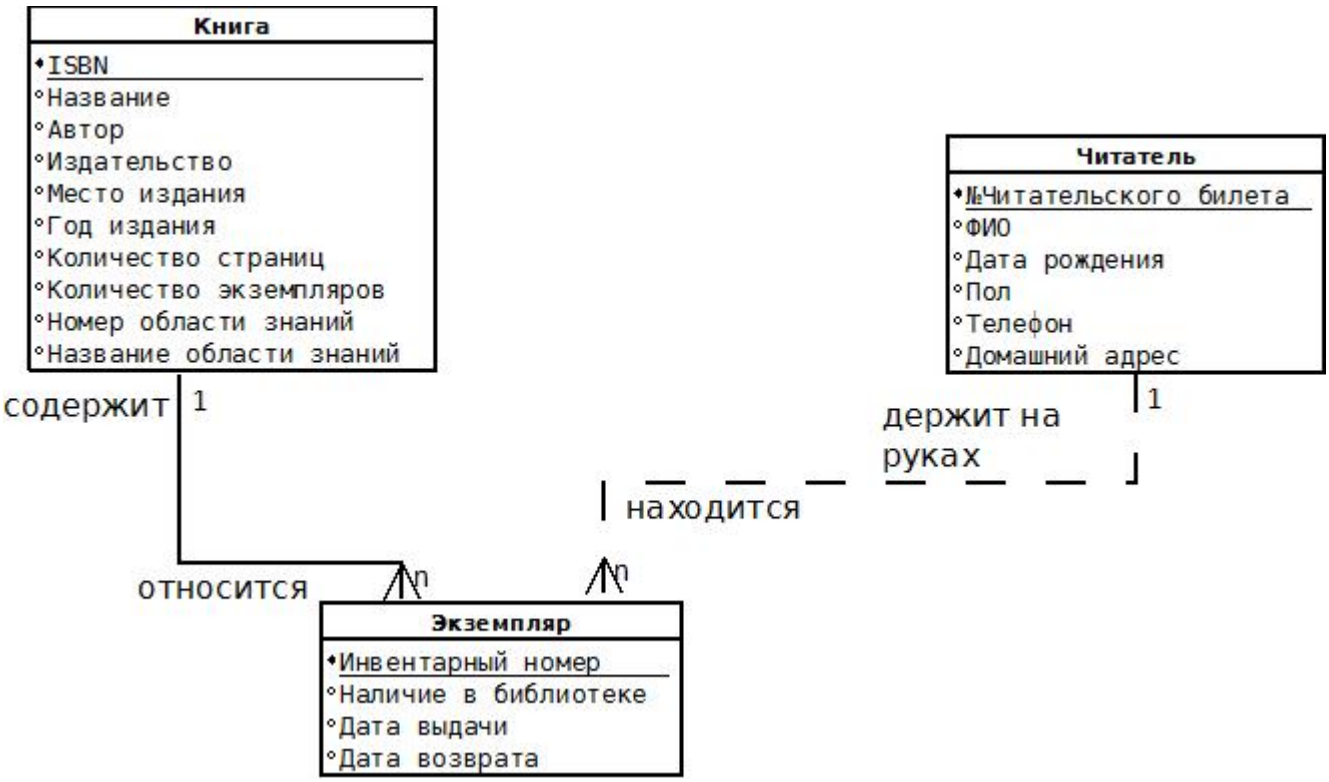



Рисунок 1. ER-диаграмма предметной области «Библиотека»

==========

При изучении темы «Даталогическое проектирование» студенты на основе ER-диаграмм формируют структуру данных для конкретной СУБД, оптимизируют данные.

**Пример теоретического блока по теме «Даталогическое проектирование».**

Следующим шагом является выбор конкретной СУБД и отображение в ее среду спецификаций инфологической модели предметной области. Эту стадию называют даталогическим (логическим) проектированием БД.

Получение реляционной схемы из ER-диаграммы.

1. Каждая простая сущность превращается в таблицу (отношение). Имя сущности становится именем таблицы.

2. Каждый атрибут становится возможным столбцом с тем же именем.

3. Компоненты уникального идентификатора сущности превращаются в первичный ключ.

4. Связи «многие к одному» и «один к одному» становятся внешними ключами. Т.е. создается копия первичного ключа(уникального атрибута) с конца связи «один», и соответствующие столбцы составляют внешний ключ.

В результате получили следующие отношения:

Книга (ISBN, Название, Автор, Издательство, Место издания, Год издания, Количество страниц, Количество экземпляров, Номер области знаний, Название области знаний )

Экземпляр (Инвентарный номер книги, ISBN , Наличие в библиотеке, № Читательского билета, Дата выдачи, Дата возврата)

Читатель (№Читательского билета, Фамилия И.О., Дата рождения, Домашний адрес, Пол, Телефон).

Основная цель логического проектирования базы данных - устранение избыточности и дублирования информации. Для удовлетворения этих требований Э.Коддом предложен аппарат нормализации отношений. База данных считается нормализованной, если ее таблицы представлены как минимум в третьей нормальной форме.

Первая нормальная форма

1НФ требует атомарности данных в таблицах, т.е. данные в таблицах должны быть представлены в виде минимально возможных и далее неделимых кусочков информации.

Вторая нормальная форма

2НФ требует, чтобы данные во всех неключевых столбцах полностью зависели от первичного ключа.

Третья нормальная форма

Третья нормальная форма, известная также как 3НФ, требует от нас, чтобы структура базы данных удовлетворяла требованиям 1НФ и 2НФ. Все неключевые столбцы таблицы зависели от первичного ключа, но были независимы друг от друга.

Отношения «Читатель» и «Экземпляр» находятся в 1-ой нормальной форме, т.к. не имеют сложных атрибутов. Атрибуты таблицы «Читатель» и таблицы «Экземпляр» зависят от первичных ключей (2НФ) и независят друг от друга что соответствует 3НФ.

Рассмотрим отношение «Книги». Каждая книга может относиться к многим областям знаний , т.е. атрибуты номер области знаний и название области знаний - сложные, а это значит, что нарушена 1НФ. Чтобы привести к 1-ой нормальной форме добавим к ключу еще один атрибут - номер области знаний.

Книга (ISBN, Название, Автор, Издательство, Место издания, Год издания, Количество страниц, Количество экземпляров, Номер области знаний, Название области знаний).

В отношении «Книга» 2-ая нормальная форма нарушена, т.к. есть неключевые атрибуты, зависящие только от части ключа, а не от всего составного ключа. Приведем это отношение ко 2-ой форме, разделив отношение на три отношения по зависимости от ключа или части ключа.

Назовем новые отношения: «Книга», «Область знаний», «Принадлежность книги к области знаний». В результате мы получили 5 отношений: «Книга», «Экземпляр», «Область знаний», «Принадлежность книги к области знаний», «Читатель» .

Даталогическая модель нормализованных отношений представлена на рисунке 2.

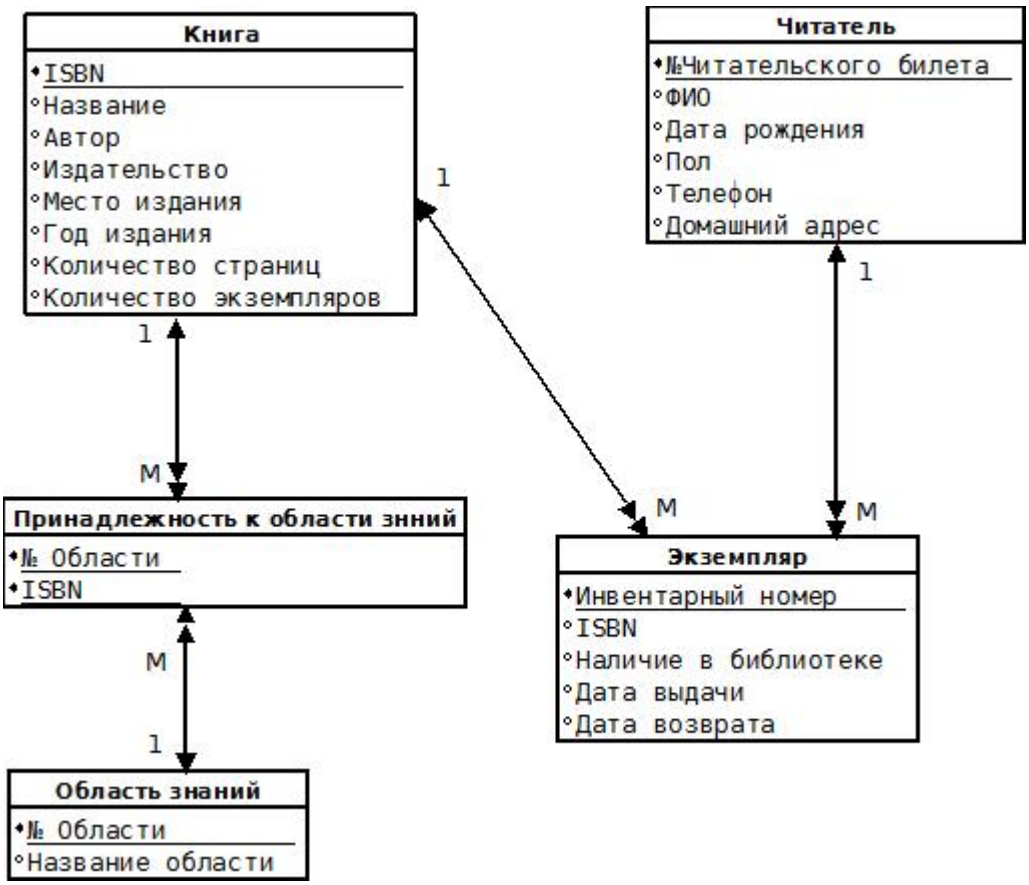



Рисунок 2 Даталогическая модель предметной области «Библиотека»

==========

При изучении темы «Физическое проектирование» студенты осваивают работу в СУБД, например в СУБД OpenOffice.org Base, входящей в пакет свободно распространяемого ПО.

**Пример теоретического блока по теме «Физическое проектирование».**

Основой для физического проектирования является схема БД, полученная на предыдущем этапе. Физическое проектирование заключается в увязке логической структуры БД и физической среды хранения с целью наиболее эффективного размещения данных.

Определим структуры таблиц в среде СУБД OpenOffice.org Base . Дадим названия таблицам и атрибутам, определим типы данных и размерность атрибутов. В таблицах выберем первичные ключи и создадим внешний. О*писания структуры таблицы «Книга» представлен в таблице 1.1*.

Таблица 1.1 Описание структуры таблицы «Книга»

| Название таблицы | Имя поля | Тип данных | Размер поля | Ключ |
|---|---|---|---|---|
| Книга | ISBN | Текстовый | 15 | Первичный |
| | Название | Текстовый | 50 | |
| | Автор | Текстовый | 50 | |
| | Издательство | Текстовый | 20 | |
| | Место издания | Текстовый | 15 | |
| | Год издания | Текстовый | 20 | |
| | Количество страниц | Числовой | | |
| | Количество экземпляров | Числовой | | |

Аналогичным образом определяются структуры остальных таблиц. После создания структуры базы данных необходимо заполнить ее данными и создать связи между таблицами. Используя СУБД OpenOffice.org Base, получилась схема связей между таблицами в базе данных «Библиотека» представлены на Рис.1.3.

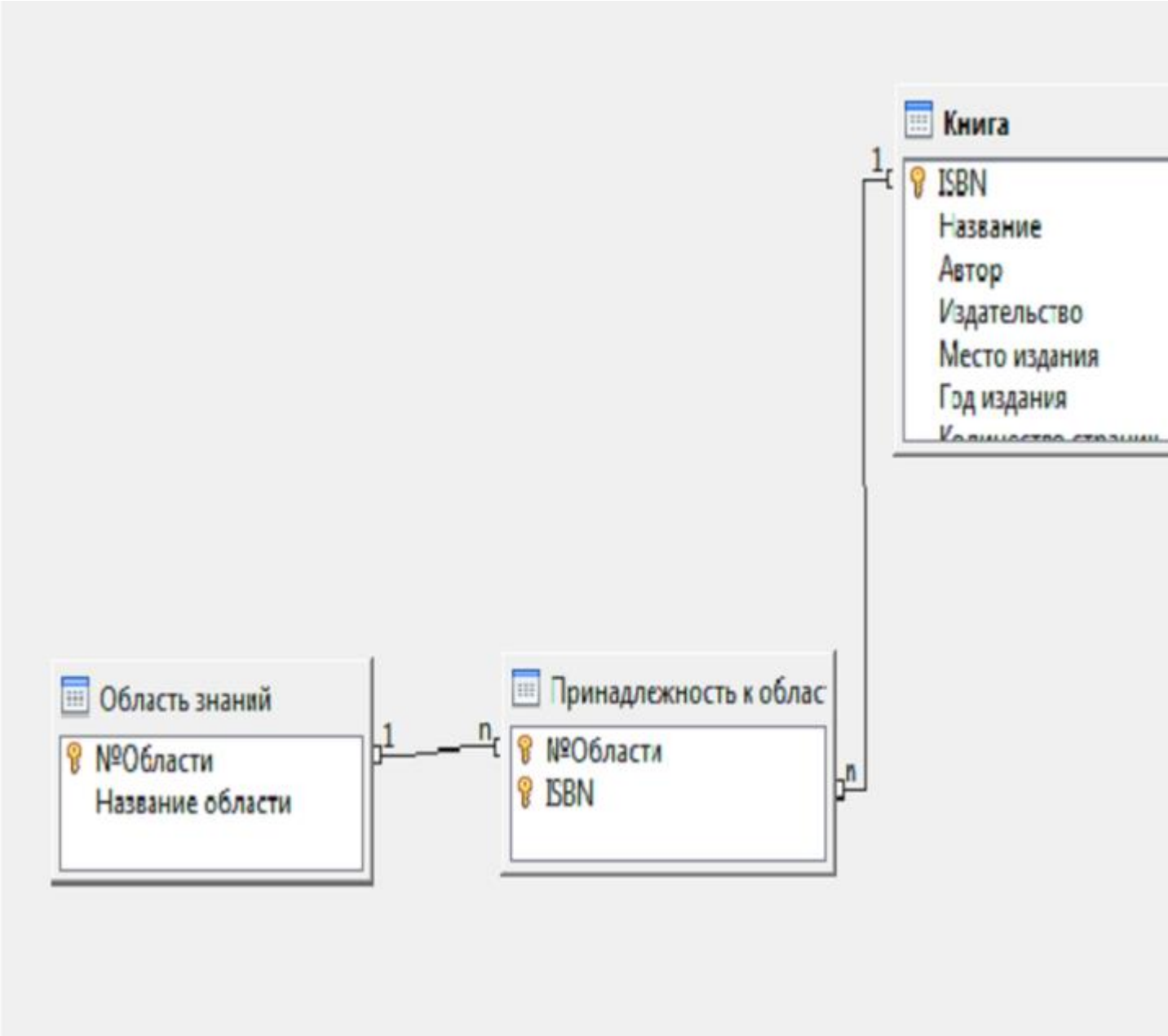



Рисунок 1.3 Схема связей таблиц БД «Библиотека»

==========

Итоговые задания по модулям разработаны в виде индивидуальных проектов, требующих реализации освоенных видов деятельности.

Для формирования отчета по самостоятельной работе студентам рекомендуется использовать текстовый редактор Writer. Для составления схем и диаграмм использовать программу Dia. А для создания базы данных рекомендуется использовать СУБД Base.

**Пример задания для самостоятельной работы.**

1. Узнать у преподавателя тему для проектирования базы данных.

2. На основе анализа материалов Интернета составить описание предметной области.

3. Построить ER-диаграмму своей предметной области.

4. Сделать даталогическое проектирование БД т.е. перевести ER-диаграмму в реляционную схему и нормализовать отношения (таблицы).

5. Определить структуры таблиц и их связи, первичные и вешние ключи Реализовать базу данных в СУБД.

6. Заполнить таблицы, спроектировать запросы, формы и отчеты.

## ЛИТЕРАТУРА

1. Можаров М.С., Коткин С.Д. О развитии содержательной линии «моделирование и формализация» в школьном курсе «информатика и икт»//Информатика и образование. 2010. № 4. С. 95-99.

2. Богомолова О. Б. Web-конструирование на HTML. / О.Б. Богомолова - М.: Бином, 2008 - 192 с.

3. Водолазкий В.В. Эффективная работа PHP 4./ В. В. Водолазкий . - СПб. : Питер, 2002 . - 416 с.

4. Гилмор В. PHP 4. Учебный курс. /Гилмор В. - СПб.: Питер, 2001. - 352 с.

5. Голицына О.Л., Максимов Н.В. Информационные технологии: Уч. для вузов - 2 изд., перераб. и доп. - М.: ФОРУМ: ИНФРА , 2008 г. - 608с.:

6. Косентино К. PHP. Web-профессионалам : пер. с англ. / К. Косентино . - Киев : БХВ, 2001 . - 208 с

7. Петюшкин А. HTML в Web-дизайне. /А. Петюшкин-СПб.: БХВ-Петербург, 2004. - 400 с.

8. Прохоренок Н.А. HTML, JavaScript, PHP и MySQL. Джельтельменский набор Web-мастера. / Н.А. Прохоренок - 2-е ., перераб. И доп. - Спб.: БХВ-Петербург, 2009. - 880 с.

9. Райордан Р. Основы реляционных баз данных. /Пер, с англ. - М.: Издательско-торговый дом "Русская Редакция", 2001. - 384 с.

10. Томсон Л., Веллинг Л. Разработка Web-приложений на PHP и MySQL: Пер. с англ. - 2-е изд., испр. - СПб: ООО «ДиаСофтЮП», 2003. - 672 с.

11. Туманов В.Е. Основы проектирования реляционных баз данных./ В. Е. Туманов - М.: Бином, 2007- 420 с.

12. Ульман Л. MySQL Руководство по изучению языка. / Пер. с англ. - М.:ДМК Пресс, 2004. - 352 с.

13. Ульман Л. Основы программирования на PHP. / Пер. с англ. - М.:ДМК Пресс, 2001. - 288 с.

14. Фролов А.В., Фролов Г.В. Базы данных в Интернете: практическое руководство по созданию Web-приложений с базами данных./ А. В. Фролов, Г.В. Фролов - М.: Издательско-торговый дом "Русская Редакция", 2000. - 448 с.

15. Фрост Р., Дей Д., Ван Слайк К. Базы данных. Проектирование и разработка./ Пер. с англ. - М: НТ Пресс, 2007. - 592 с.

16. Проектирование информационных систем [Электронный ресурс]. URL: http://www.intuit.ru/department/database/rdbdev/1/ .

17. Проектирование баз данных [Электронный ресурс]. URL: http://www.mstu.edu.ru/study/materials/zelenkov/ch_5_1.html .

18. Документация phpMyAdmin [Электронный ресурс]. URL: http://hostinfo.ru/articles/461 .

19. Синтаксис SQL [Электронный ресурс]. URL: http://citforum.ru/database/sql_kg/index.shtml

20. Уроки PHP [Электронный ресурс]. URL: http://wcode.ru/php/ .