



**МИНИСТЕРСТВО ОБРАЗОВАНИЯ  
РЕСПУБЛИКИ БЕЛАРУСЬ**

**Белорусский национальный  
технический университет**

---

**Кафедра «Программное обеспечение вычислительной техники  
и автоматизированных систем»**

# **БАЗЫ ДАННЫХ**

**Методические указания  
к выполнению курсовой работы**

**Минск  
БНТУ  
2013**

МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ  
Белорусский национальный технический университет

---

Кафедра «Программное обеспечение вычислительной техники  
и автоматизированных систем»

## БАЗЫ ДАННЫХ

Методические указания  
к выполнению курсовой работы  
для студентов специальности  
1-40 01 01 «Программное обеспечение  
информационных технологий»

Минск  
БНТУ  
2013

УДК 004.65:378.147.091.313(075.8)

ББК 32.973-018.2я7

Б17

С о с т а в и т е л ь

*И. А. Бухвалова*

Р е ц е н з е н т ы :

*А. А. Лобатый, А. А. Москаленко*

Издание содержит общие положения по выполнению курсовой работы (цели курсовой работы, необходимые теоретические сведения, пример проектирования базы данных, описание и оформление разделов курсовой работы). Приводятся правила оформления пояснительной записки.

© Белорусский национальный  
технический университет, 2013

## ОГЛАВЛЕНИЕ

ЦЕЛИ И ЗАДАЧИ РАБОТЫ .....	4
1. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ.....	4
1.1. Общие положения.....	4
1.2. Этапы проектирования базы данных .....	5
1.3. Требования к приложению .....	11
2. ПРИМЕР ПРОЕКТИРОВАНИЯ РЕЛЯЦИОННОЙ БАЗЫ ДАННЫХ.....	12
2.1 Концептуальное проектирование.....	12
2.2. Логическое проектирование реляционной БД.....	15
2.3. Физическое проектирование БД.....	32
3. ВЫПОЛНЕНИЕ КУРСОВОЙ РАБОТЫ .....	34
СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ.....	35

## ЦЕЛИ И ЗАДАЧИ РАБОТЫ

Цель курсовой работы – применение на практике знаний, полученных в процессе изучения дисциплины «Базы данных», и получение практических навыков проектирования базы данных на основе описания предметной области, создания автоматизированных информационных систем (АИС), основанных на базах данных.

## 1. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

### 1.1. Общие положения

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.

Основная цель процесса проектирования БД состоит в получении такого проекта, который удовлетворяет следующим требованиям:

- корректность схемы БД, т. е. база должна быть гомоморфным образом моделируемой предметной области (ПО), где каждому объекту предметной области соответствуют данные в памяти ЭВМ, а каждому процессу – адекватные процедуры обработки данных;
- обеспечение ограничений ;
- эффективность функционирования ;
- защита данных (от аппаратных и программных сбоев и несанкционированного доступа);
- простота и удобство эксплуатации;
- гибкость, т. е. возможность развития и адаптации к изменениям предметной области и/или требований пользователей.

## 1.2. Этапы проектирования базы 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.

### 1.2.1. Концептуальное проектирование

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

Концептуальная модель ПО представляет собой описание структуры и динамики ПО, характера информационных потребностей пользователей в терминах, понятных пользователю и не зависящих от реализации БД. Это описание выражается в терминах не отдельных объектов ПО 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

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.

### 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 ПО. 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 ПО. 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 студента 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 ту же 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. Пример бинарной связи – связь между отделом и сотрудниками, которые в нём работают. Примером тернарной связи является связь типа экзамен между сущностями ДИСЦИПЛИНА, СТУДЕНТ, ПРЕПОДАВАТЕЛЬ. Из последнего примера видно, что связь также может иметь атрибуты (в данном случае это Дата проведения и Оценка). Пример ER–диаграммы с указанием сущностей, их атрибутов и связей приведен на рисунке 1.

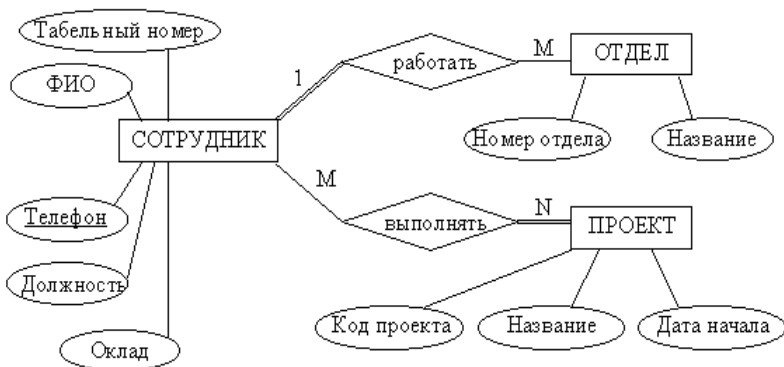


Рисунок 1. Пример ER–диаграммы с однозначными и многозначными атрибутами

### 1.2.2. Логическое проектирование БД

На этапе логического проектирования разрабатывается логическая структура БД, соответствующая логической модели ПО. Решение этой задачи существенно зависит от модели данных, поддерживаемой выбранной СУБД. Будем рассматривать логическое проек-

тирование 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 ограниченной 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)  $A$  называется **детерминантом**.

Полная функциональная зависимость означает, что если атрибут  $B$  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 к 2НФ, нужно исключить из отношения частичную зависимость и поместить ее в новое отношение вместе с копией их детерминанта.

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

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

$A \rightarrow B$  и  $B \rightarrow C$ , то говорят, что атрибут  $C$  транзитивно зависит от атрибута  $A$  через атрибут  $B$ .

Для того чтобы привести отношение к 3НФ, нужно исключить из отношения транзитивную зависимость, поместив ее в 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.

### **1.2.3. 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.

## **1.3 Требования к приложению**

Процесс проектирования базы данных охватывает несколько основных сфер.

Проектирование объектов базы данных (таблицы, представления, индексы, триггеры, хранимые процедуры, функции, пакеты) для представления данных предметной области в базе данных.

Проектирование интерфейса взаимодействия с базой данных (формы, отчеты и т.д.), т.е. проектирование приложений, которые будут сопровождать данные в базе данных и реализовывать вопросы-ответные отношения на этих данных.

Проектирование баз данных под конкретную вычислительную среду или информационную технологию (архитектура "клиент-сервер", параллельные архитектуры, распределенная вычислительная среда).

Проектирование баз данных под назначение системы (интеллектуальный анализ данных, OLAP, OLTP и т.д.).

Приложения работы с базой данных проектируются одновременно с физической схемой базы данных, а не отдельно! Зачастую вычислительная среда задается в качестве входных условий проектирования, но иногда проектирование следует проводить с учетом возможного перехода в будущем на другую аппаратную платформу или технологию.

## **2. ПРИМЕР ПРОЕКТИРОВАНИЯ РЕЛЯЦИОННОЙ БАЗЫ ДАННЫХ**

### **2.1 Концептуальное проектирование**

#### *2.1.1 Анализ предметной области*

В качестве примера возьмем базу данных компании, которая занимается издательской деятельностью.

База данных создаётся для информационного обслуживания редакторов, менеджеров и других сотрудников компании. БД должна содержать данные о сотрудниках компании, книгах, авторах, финансовом состоянии компании и предоставлять возможность получать разнообразные отчёты.

В соответствии с предметной областью система строится с учётом следующих особенностей (требований):

- каждая книга издаётся в рамках контракта;
- книга может быть написана несколькими авторами;
- контракт подписывается одним менеджером и всеми авторами книги;
  - каждый автор может написать несколько книг (по разным контрактам);
  - порядок, в котором авторы указаны на обложке, влияет на размер гонорара;
  - если сотрудник является редактором, то он может работать одновременно над несколькими книгами;
  - у каждой книги может быть несколько редакторов, один из них – ответственный редактор;
  - каждый заказ оформляется на одного заказчика;
  - в заказе на покупку может быть перечислено несколько книг.

Выделим базовые сущности этой предметной области:

1. Сотрудники компании. Атрибуты сотрудников – ФИО, табельный номер, пол, дата рождения, паспортные данные, ИНН, должность, оклад, домашний адрес и телефоны. Для редакторов необходимо хранить сведения о редактируемых книгах; для менеджеров – сведения о подписанных контрактах.

2. Авторы. Атрибуты авторов – ФИО, ИНН (индивидуальный номер налогоплательщика), паспортные данные, домашний адрес, телефоны. Для авторов необходимо хранить сведения о написанных книгах.

3. Книги. Атрибуты книги – авторы, название, тираж, дата выхода, цена одного экземпляра, общие затраты на издание, авторский гонорар.

4. Контракты будем рассматривать как связь между авторами, книгами и менеджерами. Атрибуты контракта – номер, дата подписания и участники.

5. Для отражения финансового положения компании в системе нужно учитывать заказы на книги. Для заказа необходимо хранить номер заказа, заказчика, адрес заказчика, дату поступления заказа, дату

его выполнения, список заказанных книг с указанием количества экземпляров.

ER–диаграмма издательской компании приведена на рисунке 2 (базовые сущности на рисунках выделены полужирным шрифтом).

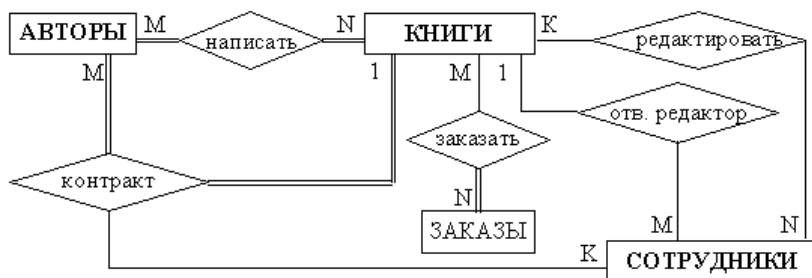


Рисунок 2. ER – диаграмма издательской компании

### 2.1.2. Анализ информационных задач и круга пользователей системы

Система создаётся для обслуживания следующих групп пользователей:

- администрация (дирекция);
- менеджеры;
- редакторы;
- сотрудники компании, обслуживающие заказы.

Определим границы информационной поддержки пользователей:

1) Функциональные возможности:

- ведение БД (запись, чтение, модификация, удаление в архив);
- обеспечение логической непротиворечивости БД;
- обеспечение защиты данных от несанкционированного или случайного доступа (определение прав доступа);
- реализация наиболее часто встречающихся запросов в готовом виде;
- предоставление возможности сформировать произвольный запрос на языке манипулирования данными.

## 2) Готовые запросы:

- получение списка всех текущих проектов (книг, находящихся в печати и в продаже);
- получение списка редакторов, работающих над книгами;
- получение полной информации о книге (проекте);
- получение сведений о конкретном авторе (с перечнем всех книг);
- получение информации о продажах (по одному или по всем проектам);
- определение общей прибыли от продаж по текущим проектам;
- определение размера гонорара автора по конкретному проекту.

## 2.2. Логическое проектирование реляционной БД

### ***2.2.1. Преобразование ER-диаграммы в схему базы данных***

База данных создаётся на основании схемы базы данных. Для преобразования ER – диаграммы в схему БД приведём уточнённую ER – диаграмму, содержащая атрибуты сущностей (рисунок 3).

Преобразование ER-диаграммы в схему БД выполняется путем сопоставления каждой сущности и каждой связи, имеющей атрибуты, отношения (таблицы БД). При построении схемы БД будем использовать обозначения, представленные на рисунке 4.



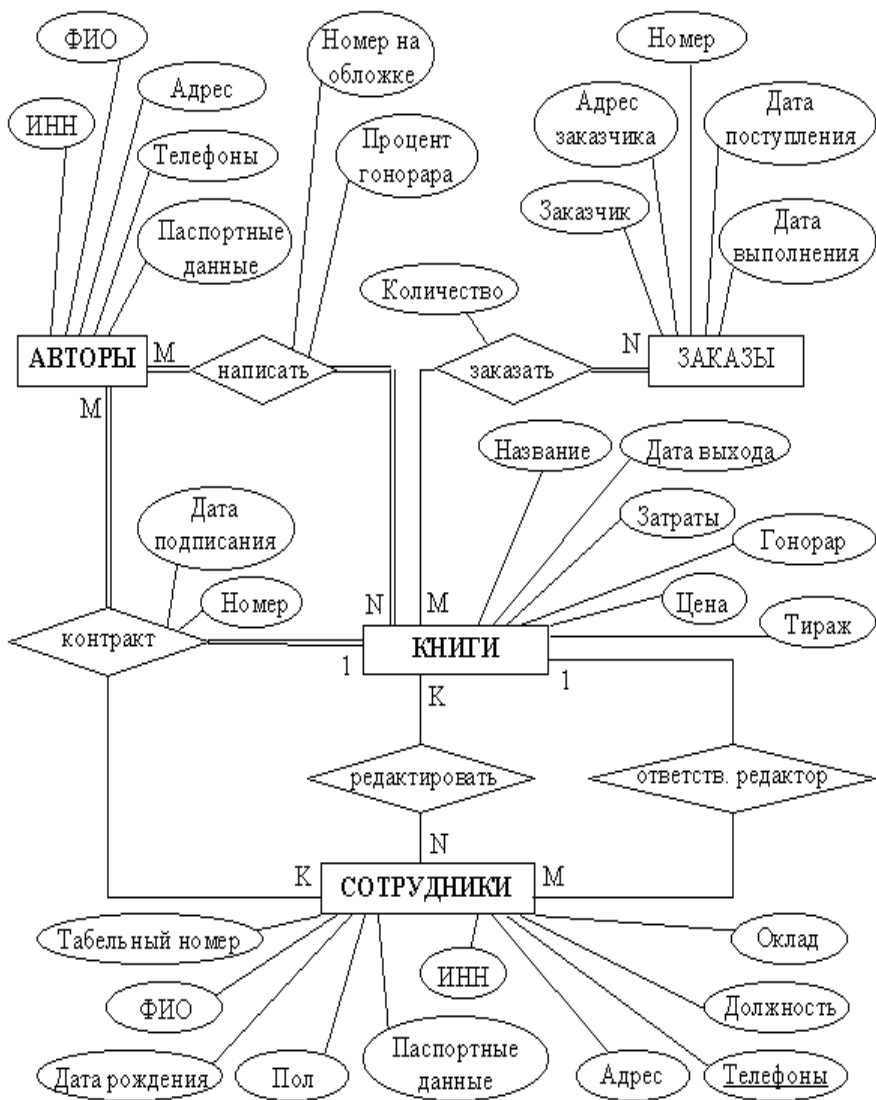


Рисунок 3. Уточнённая ER – диаграмма издательской компании

*Примечание:* многозначные атрибуты на рисунке выделены подчеркиванием.

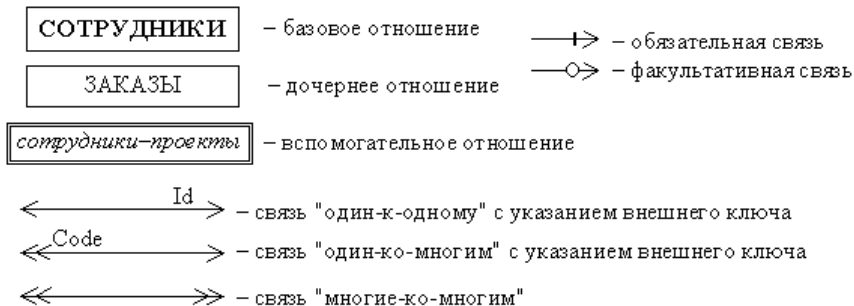


Рисунок 4. Обозначения, используемые на схеме базы данных

Полученная схема реляционной базы данных (РБД) приведена на рисунке 5.

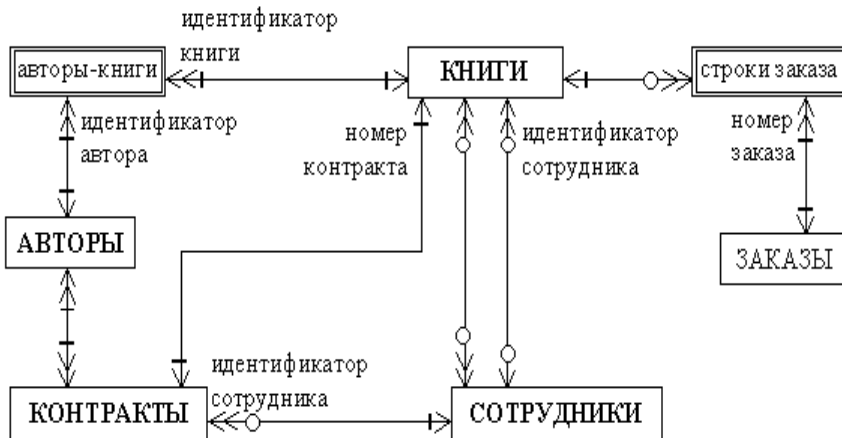


Рисунок 5. Схема РБД, полученная из ER – диаграммы издательской компании

На схеме (рисунок 5) есть связь типа 1:1 – обязательная связь между КНИГАМИ и КОНТРАКТАМИ. Такие отношения следует объединять в одно. Дополнительный эффект от объединения этих отношений – слияние связей авторы–контракты и авторы–книги: ведь в нашем случае контракт заключается именно для написания книги.

Примечание: исключение для связи типа 1:1 составляют ситуации, когда для увеличения производительности системы в отдельную таблицу выделяются редко используемые данные большого объёма.

Связь типа 1:N(один-ко-многим) между отношениями реализуется через внешний ключ. Ключ вводится для того отношения, к которому осуществляется множественная связь (КНИГИ).

Связь «редактировать» между отношениями КНИГИ и СОТРУДНИКИ принадлежит к типу M:N (многие-ко-многим). Этот тип связи реализуется через вспомогательное отношение, которое является соединением первичных ключей соответствующих отношений.

Бинарная связь между отношениями не может быть обязательной для обоих отношений. После объединения сущностей КНИГИ и КОНТРАКТЫ остаётся три связи, обязательные для всех участников: между авторами и книгами и между заказами и строками заказов. Такой тип связи означает, что, например, прежде чем добавить новый заказ в отношение ЗАКАЗЫ, нужно добавить новую строку в отношение СТРОКИ ЗАКАЗА, и наоборот. Поэтому для такой связи необходимо снять с одной стороны условие обязательности. Так как все эти связи будут реализованы с помощью внешнего ключа, снимем условие обязательности связей для отношений, содержащих первичные ключи.

Схема на рисунке 6 содержит цикл «сотрудники–книги–сотрудники». Цикл допустим только в том случае, если связи, входящие в него, независимы друг от друга. Примем для нашей ПО, что ответственный редактор книги может являться также просто редактором этой же книги или не входить в число редакторов. При этом цикл не приводит к нарушению логической целостности данных.

Уточнённая схема РБД издательской компании приведена на рисунке 6.

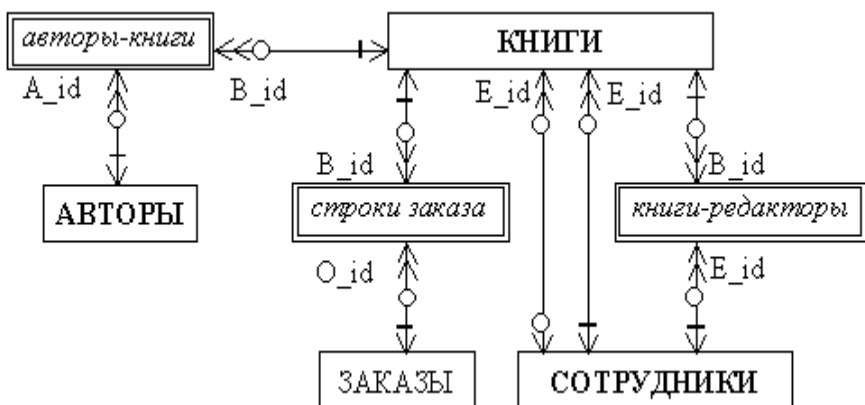


Рисунок.6. Уточнённая схема РБД издательской компании

## 2.2.2. Составление реляционных отношений

Каждое реляционное отношение соответствует одной сущности (объекту ПО) и в него вносятся все атрибуты сущности. Для каждого отношения необходимо определить первичный ключ и внешние ключи (если они есть). В том случае, если базовое отношение не имеет потенциальных ключей, вводится суррогатный первичный ключ, который не несёт смысловой нагрузки и служит только для идентификации записей.

*Примечание:* суррогатный первичный ключ также может вводиться в тех случаях, когда потенциальный ключ имеет большой размер (например, длинная символьная строка) или является составным (не менее трёх атрибутов).

Потенциальными ключами отношения АВТОРЫ являются атрибуты

Паспортные данные и ИНН. Первый хранится как длинная строка, а последний по условиям предметной области не является обязательным. Поэтому для авторов необходимо ввести суррогатный ключ –  $A\_id$ . Книги можно идентифицировать по атрибуту Кон-

тракт: его номер обязателен и уникален. Потенциальные ключи отношения СОТРУДНИКИ – атрибуты ИНН, Паспортные данные, Табельный номер, причём все они обязательные. Табельный номер занимает меньше памяти, чем ИНН, поэтому он и будет первичным ключом. Кортежи отношения ЗАКАЗЫ можно идентифицировать ключом Номер заказа.

Потенциальными ключами вспомогательных отношений являются комбинации первичных ключей соответствующих базовых отношений.

Отношения приведены в таблицах 1 - 7. Для каждого отношения указаны атрибуты с их внутренним названием, типом и длиной. Типы данных обозначаются так: N – числовой, С – символьный, D – дата (последний имеет стандартную длину, зависящую от СУБД, поэтому она не указывается).

Таблица 1 – Схема отношения СОТРУДНИКИ (Employees)

Содержание поля	Имя поля	Тип, длина	Примечания
Табельный номер	E_ID	N(4)	первичный ключ
Фамилия, имя, отчество	E_NAME	C(50)	обязательное поле
Дата рождения	E_BORN	D	
Пол	E_SEX	C(1)	обязательное поле
Паспортные данные	E_PASSP	C(50)	обязательное поле
ИНН	E_INN	N(12)	обязательное уникальное поле
Должность	E_POST	C(30)	обязательное поле
Оклад	E_SALARY	N(8,2)	обязательное поле

Продолжение табл. 1

Адрес	E_ADDR	C(50)	
Телефоны	E_TEL	C(30)	многозначное поле

Таблица 2. – Схема отношения КНИГИ (Books)

Содержание поля	Имя поля	Тип, длина	Примечания
Номер контракта	B_CONTRACT	N(6)	первичный ключ
Дата подписания контракта	B_DATE	D	обязательное поле
Менеджер	B_MAN	N(4)	внешний ключ (к Employees)
Название книги	B_TITLE	N(40)	обязательное поле
Цена	B_PRICE	N(6,2)	цена экземпляра книги
Затраты	B_ADVANCE	N(10,2)	общая сумма затрат на книгу
Авторский гонорар	B_FEE	N(8,2)	общая сумма гонорара
Дата выхода	B_PUBL	D	
Тираж	B_CIRCUL	N(5)	
Ответственный редактор	B_EDIT	N(4)	внешний ключ (к Employees)

Таблица 3 - Схема отношения АВТОРЫ (Authors)

Содержание поля	Имя поля	Тип, длина	Примечания
Код автора	A_ID	N(4)	суррогатный первичный ключ

Продолжение табл.3.

Фамилия, имя, отчество	A_NAME	C(50)	обязательное поле
Паспортные данные	A_PASSP	C(50)	обязательное поле
ИНН	A_INN	N(12)	уникальное поле
Адрес	A_ADDR	C(50)	обязательное поле
Телефоны	A_TEL	C(30)	многозначное поле

Таблица 4 - Схема отношения ЗАКАЗЫ (Orders)

Содержание поля	Имя поля	Тип, длина	Примечания
Номер заказа	O_ID	N(6)	первичный ключ
Заказчик	O_COMPAN Y	C(40)	обязательное поле
Дата поступления заказа	O_DATE	D	обязательное поле
Адрес заказчика	O_ADDR	C(50)	обязательное поле
Дата выполнения заказа	O_READY	D	

Таблица 5 - Схема отношения КНИГИ–АВТОРЫ (Titles)

Содержание поля	Имя поля	Тип, длина	Примечания
Код книги (№ контракта)	B_ID	N(6)	внешний ключ (к Books)

Продолжение табл.5.

Код автора	A_ID	N(4)	внешний ключ (к Authors)
Номер в списке	A_NO	N(1)	обязательное поле
Гонорар	A_FEE	N(3)	процент от общего гонорара

Таблица 6 - Схема отношения КНИГИ–РЕДАКТОРЫ (Editors)

Содержание поля	Имя поля	Тип, длина	Примечания
Код книги (№ контракта)	B_ID	N(6)	внешний ключ (к Books)
Код редактора	E_ID	N(4)	внешний ключ (к Employees)

Таблица 7 - Схема отношения СТРОКИ ЗАКАЗА (Items)

Содержание поля	Имя поля	Тип, длина	Примечания
Номер заказа	O_ID	N(6)	внешний ключ (к Orders)
Код книги (№ контракта)	B_ID	N(6)	внешний ключ (к Books)
Количество	B_COUNT	N(4)	обязательное поле

### ***2.2.3. Нормализация полученных отношений (до 3НФ)***

1НФ. Для приведения таблиц к 1НФ требуется составить прямоугольные таблицы (один атрибут – один столбец) и разбить сложные атрибуты на простые, а многозначные атрибуты вынести в отдельные отношения.



*Примечание.* В реальных БД сложные атрибуты разбиваются на простые, если:

- а) этого требует внешнее представление данных;
- б) в запросах поиск может осуществляться по отдельной части атрибута.

Разделим атрибуты Фамилия, имя, отчество на два атрибута Фамилия и Имя, отчество и Паспортные данные на атрибуты Номер паспорта (уникальный), Дата выдачи и Кем выдан.

Многозначный атрибут Телефоны для сотрудников компании следует сначала разделить на два – Домашние телефоны и Рабочие телефоны. (Для авторов мы не будем различать домашние и рабочие телефоны). Затем нужно создать отдельные отношения с (нерабочими) телефонами для сотрудников (ТЕЛЕФОНЫ СОТРУДНИКОВ) и для авторов (ТЕЛЕФОНЫ АВТОРОВ).

Атрибут Рабочие телефоны отношения СОТРУДНИКИ имеет неоднородные значения. Один из номеров телефонов – основной – определяется рабочим местом сотрудника (рассматриваются только стационарные телефоны). Наличие других номеров зависит от того, есть ли в том же помещении (комнате) другие сотрудники, имеющие стационарные телефоны. Можно добавить в отношение СОТРУДНИКИ атрибут Номер комнаты, а в атрибуте Рабочие телефоны хранить номер того телефона, который стоит на рабочем месте сотрудника. Дополнительные номера телефонов можно будет вычислить из других кортежей с таким же номером комнаты. Но в случае увольнения сотрудника мы потеряем сведения о номере рабочего телефона.

Поэтому создадим новое отношение КОМНАТЫ и включим в него атрибуты Номер комнаты и Телефон. Так как в комнате может не быть телефона, первичный ключ нового отношения не определен (ПК не может содержать null-значения), но на этих атрибутах можно определить составной уникальный ключ. Связь между отношениями СОТРУДНИКИ и КОМНАТЫ реализуем через составной внешний ключ (Номер комнаты, Телефон). Значение внешнего ключа для каждого сотрудника будем брать из того кортежа, в котором хранится основной рабочий телефон этого сотрудника.

2НФ. В нашем случае составные первичные ключи имеют отношения СТРОКИ ЗАКАЗА, КНИГИ-АВТОРЫ и КНИГИ-

РЕДАКТОРЫ. Неключевые атрибуты этих отношений функционально полно зависят от первичных ключей.

ЗНФ. В отношении ЗАКАЗЫ атрибут Адрес заказчика зависит от атрибута Заказчик, а не от первичного ключа, поэтому адрес следует вынести в отдельное отношение ЗАКАЗЧИКИ. Но при этом первичным ключом нового отношения станет атрибут Заказчик, т.е. длинная символьная строка. Целесообразнее перенести в новое отношение атрибуты Заказчик и Адрес заказчика и ввести для него суррогатный ПК. Так как каждый заказчик может сделать несколько заказов, связь между отношениями ЗАКАЗЧИКИ и ЗАКАЗЫ будет 1:n и суррогатный ПК станет внешним ключом для отношения ЗАКАЗЫ.

В отношении СОТРУДНИКИ атрибут Оклад зависит от атрибута Должность. Поступим с этой транзитивной зависимостью так же, как в предыдущем случае: создадим новое отношение ДОЛЖНОСТИ, перенесём в него атрибуты Должность и Оклад и введём суррогатный первичный ключ.

В отношениях СОТРУДНИКИ и АВТОРЫ атрибуты Дата выдачи и Кем выдан зависят от атрибута Номер паспорта, а не от первичного ключа. Но если мы выделим их в отдельное отношение, то получившиеся связи будут иметь тип 1:1. Следовательно, декомпозиция нецелесообразна.

В реальных базах данных после нормализации может проводиться денормализация. Она проводится с одной целью – повышение производительности БД. Рассмотрим некоторые запросы к нашей базе данных.

Например, запрос на получение списка телефонов авторов или домашних телефонов сотрудников потребует в нормализованной БД соединения отношений. Пользователю безразлична форма представления этого списка: номера телефонов через запятую или в столбец. Поэтому мы откажемся от создания отдельных отношений с номерами телефонов, и вернёмся к варианту с многозначными полями. (Это не касается рабочих телефонов сотрудников).

Другой запрос: как определяется, можно ли выполнить очередной заказ? Для каждой позиции заказа нужно просуммировать количество книг по выполненным заказам, получить остаток (тираж минус полученная сумма) и сравнить остаток с объёмом заказа. Такой расчёт может потребовать много времени, поэтому предлагает-

ся добавить в отношении КНИГИ производный атрибут Остаток тиража. Значение этого атрибута должно автоматически пересчитываться при установлении даты выполнения заказа.

После проведённых преобразований схема БД выглядит так (рисунок 7):

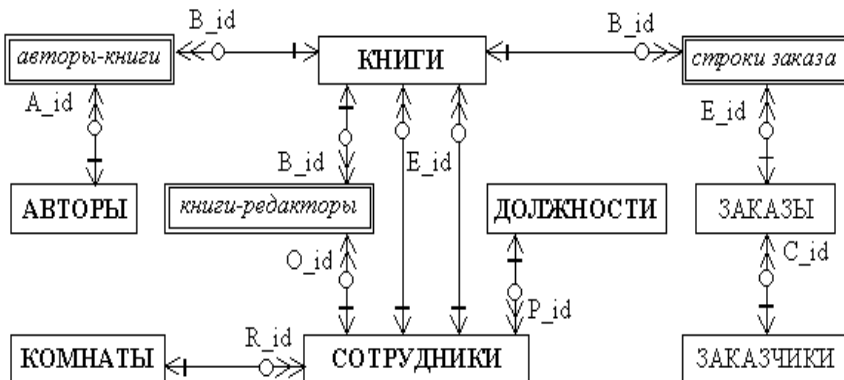


Рисунок 7 – Окончательная схема РБД издательской компании

Окончательные схемы отношений базы данных с указанием ключей и других ограничений целостности приведены в таблицах 8–17.

Таблица 8 – Схема отношения ДОЛЖНОСТИ (Posts)

Содержание поля	Имя поля	Тип, длина	Примечания
Код должности	P_ID	N(3)	суррогатный первичный ключ
Название должности	P_POST	C(30)	обязательное поле
Оклад	P_SAL	N(8,2)	обязательное поле

Таблица 9 - Схема отношения КОМНАТЫ (Rooms)

Содержание поля	Имя поля	Тип, длина	Примечания
Номер комнаты	R_NO	N(3)	обязательное поле
Номер телефона	R_TEL	C(10)	

Таблица 10- Схема отношения СОТРУДНИКИ (Employees)

Содержание поля	Имя поля	Тип, длина	Примечания
Табельный номер	E_ID	N(4)	первичный ключ
Фамилия	E_FNAME	C(20)	обязательное поле
Имя, отчество	E_LNAME	C(30)	обязательное поле
Дата рождения	E_BORN	D	
Пол	E_SEX	C(1)	обязательное поле
Код должности	E_POST	N(3)	внешний ключ (к Posts)
Номер комнаты	E_ROOM	N(3)	составной внешний ключ(к Rooms)
Номер телефона	E_TEL	C(10)	
ИНН	E_INN	C(12)	обязательное поле
Номер паспорта	E_PASSP	C(12)	обязательное поле
Кем выдан паспорт	E_ORG	C(30)	обязательное поле

Продолжение таблицы 10.

Дата выдачи паспорта	E_PDATE	D	обязательное поле
Адрес	E_ADDR	C(50)	

Таблица 11 - Схема отношения ЗАКАЗЧИКИ (Customers)

Содержание поля	Имя поля	Тип, длина	Примечания
Код заказчика	C_ID	N(4)	суррогатный первичный ключ
Заказчик	C_NAME	C(30)	обязательное поле
Адрес заказчика	C_ADDR	C(50)	обязательное поле

Таблица 12 - Схема отношения АВТОРЫ (Authors)

Содержание поля	Имя поля	Тип, длина	Примечания
Код автора	A_ID	N(4)	суррогатный ключ
Фамилия	A_FNAME	C(20)	обязательное поле
Имя, отчество	A_LNAME	C(30)	обязательное поле
ИНН	A_INN	C(12)	
Номер паспорта	A_PASSP	C(12)	обязательное поле
Кем выдан паспорт	A_ORG	C(30)	обязательное поле
Дата выдачи паспорта	A_PDATE	D	обязательное поле
Адрес	A_ADDR	C(50)	обязательное поле
Телефоны	A_TEL	C(30)	многозначное

Таблица 13 - Схема отношения КНИГИ (Books)

Содержание поля	Имя поля	Тип, длина	Примечания
Номер контракта	B_CONTRAC T	N(6)	первичный ключ
Дата подписания контракта	B_DATE	D	обязательное поле
Менеджер	B_MAN	N(4)	внешний ключ (к Employees)
Название книги	B_TITLE	N(40)	обязательное поле
Цена	B_PRICE	N(6,2)	цена экземпляра книги
Затраты	B_ADVANCE	N(10,2)	общая сумма затрат на книгу
Гонорар	B_FEE	N(8,2)	общая сумма гонорара
Дата выхода	B_PUBL	D	
Тираж	B_CIRCUL	N(5)	
Ответственный редактор	B_EDIT	N(4)	внешний ключ (к Employees)
Остаток тиража	B_REST	N(5)	производное поле

Таблица 14 - Схема отношения ЗАКАЗЫ (Orders)

Содержание поля	Имя поля	Тип, длина	Примечания
Номер заказа	O_ID	N(6)	первичный ключ
Код заказчика	O_COMPANY	N(4)	внешний ключ (к Customers)

Продолжение таблицы 14.

Дата поступления заказа	O_DATE	D	обязательное поле
Дата выполнения заказа	O_READY	D	

Таблица 15 - Схема отношения КНИГИ–АВТОРЫ (Titles)

Содержание поля	Имя поля	Тип, длина	Примечания
Код книги (№ контракта)	B_ID	N(6)	внешний ключ (к Books)
Код автора	A_ID	N(4)	внешний ключ (к Authors)
Номер в списке	A_NO	N(1)	обязательное поле
Гонорар	A_FEE	N(3)	процент от общего гонорара

Таблица 16 - Схема отношения СТРОКИ ЗАКАЗА (Items)

Содержание поля	Имя поля	Тип, длина	Примечания
Номер заказа	O_ID	N(6)	внешний ключ (к Orders)
Код книги (№ контракта)	B_ID	N(6)	внешний ключ (к Books)
Количество	B_COUNT	N(4)	обязательное

Таблица 17. Схема отношения КНИГИ–РЕДАКТОРЫ (Editors)

Содержание поля	Имя поля	Тип, длина	Примечания
Код книги (№ контракта)	B_ID	N(6)	внешний ключ (к Books)
Код редактора	E_ID	N(4)	внешний ключ (к Employ-

			ees)
--	--	--	------

#### ***2.2.4. Определение дополнительных ограничений целостности***

Перечислим ограничения целостности, которые не указаны в таблицах 8–17.

1. Значения всех числовых атрибутов – больше 0 (или null, если атрибут необязателен).
2. Область значений атрибута Sex отношения EMPLOYEES – символы 'м' и 'ж'.
3. Отношение ROOMS не имеет первичного ключа, но комбинация значений (R\_no, Tel) уникальна.
4. В отношении TITLES порядковые номера авторов на обложке одной книги должны идти подряд, начиная с 1.
5. В отношении TITLES сумма процентов гонорара по одной книге равна 100.

Ограничения (4,5) нельзя реализовать в схеме отношения. Такие ограничения называются процедурными и будут реализованы через хранимые процедуры, триггеры.

#### ***2.2.5. Описание групп пользователей и прав доступа***

Опишем для каждой группы пользователей права доступа к каждой таблице и к каждому полю (атрибуту).

Администратор БД: имеет доступ ко всем данным (по записи), может изменять структуру базы данных и связи между отношениями. Устанавливает права доступа для всех остальных групп.

Представители администрации компании: имеют доступ по чтению ко всем данным и доступ по записи к отношениям POSTS, ROOMS и EMPLOYEES.

Менеджеры: имеет доступ по чтению ко всем данным, кроме отношения POSTS. Имеют доступ по записи к отношениям AUTHORS, CUSTOMERS, BOOKS, EDITORS, TITLES, ORDERS, ITEMS.

Редакторы: имеют доступ по чтению к следующим отношениям:



AUTHORS, кроме полей A\_passp, A\_org, A\_pdate, A\_INN(паспортные данные и ИНН).

BOOKS, кроме полей B\_advance, B\_fee (затраты и гонорар).

EDITORS.

TITLES.

Сотрудники, принимающие и выполняющие заказы: имеют доступ по записи к отношениям CUSTOMERS, ORDERS, ITEMS и по чтению к полям B\_title, B\_circul, B\_priceи B\_rest отношения BOOKS (название, тираж, цена, непроданный остаток тиража).

### 2.3. Физическое проектирование БД

Эта фаза проектирования базы данных выполняется исходя из требований выбранной целевой СУБД. В нашем случае – MS SQL Server 2005 (2008)

Приведём фрагмент описания схемы БД на DDL:

**Отношение POSTS (должности):**

```
create table posts (p_id numeric(3) primary key,p_post varchar(30) not null,
```

```
p_salary numeric(8,2) not null check(p_salary > 0));
```

**Отношение ROOMS (комнаты):**

```
create table rooms (r_no numeric(3) primary key,r_tel varchar(10),unique(r_no, r_tel));
```

**Отношение EMPLOYEES (сотрудники):**

```
create table employees (e_id numeric(4) primary key,e_fname varchar(20) not null,e_lname varchar(30) not null,e_born date,e_sex char(1) not null check(e_sex in ('ж','м')),e_post numeric(3) references posts,e_room numeric(3),e_tel varchar(10),e_inn char(12) not null,e_passp char(12) not null,e_org varchar(30) not null,e_pdate date not null,e_addr varchar(50),foreign key(e_room,e_tel) references rooms(r_no,r_tel));
```

Другие отношения описываются аналогично.

Права доступа пользователей, описанные в п. 2.4.4, предоставляются с помощью команды GRANT. Рассмотрим для примера права сотрудника компании user1, который принимает и обслуживает

заказы. Права доступа к отношениям CUSTOMERS, ORDERS, ITEMS могут быть описаны следующим образом:

```
grant insert, update on customers to user1;  
grant insert, update, delete on orders to user1;  
grant insert, update, delete on items to user1;
```

Для реализации частичного доступа к отношению BOOKS следует создать соответствующее представление и предоставить доступ к этому представлению:

```
create view goods (id, title, circul, price, rest)  
as select b_contract, b_title, b_circul, b_price, b_rest  
from books;  
grant select on goods to user1;
```

Приведём примеры нескольких готовых запросов:

- **Список всех текущих проектов (книг, находящихся в печати и в продаже):**

```
create view edits as  
select b_contract, b_title, b_date  
from books  
where b_rest is null or b_rest > 0;
```

- **Список редакторов, работающих над книгами:**

```
create view edits (title, fname, lname) as  
select b_title, e_fname, e_lname /*ответственный редактор*/  
from books, employees e  
where b_edit=e_id and  
(b_publ is null or b_publ > sysdate);  
union /*sysdate – текущая дата*/  
select b_title, a_fname, a_lname  
from books, employees e, editors d  
where b.b_contract=d.b_id and d.e_id=a.e_id and  
(b_publ is null or b_publ > sysdate)  
order by 1;
```

- **Определение общей прибыли от продаж по текущим проектам:**

```
create view edits (title, total) as  
select b_title, (circul–rest)*price–advance  
from books  
where b_rest is null or b_rest > 0;
```

### 3. ВЫПОЛНЕНИЕ КУРСОВОЙ РАБОТЫ

Курсовая работа выполняется по индивидуальному заданию, которое утверждается заведующим кафедрой.

Пояснительная записка должна отражать ход выполнения курсовой работы и содержать три основные части:

- теоретическая часть – проектирование базы данных,
- практическая часть – реализация базы данных и разработка приложения,
- графическая часть – диаграммы, схемы, указанные в задании должны быть представлены на листах формата А4 с рамкой и штампом.

Образец оформления графического материала приводится в [3].

К пояснительной записке прилагается электронный носитель, на который записывается расчетно-пояснительная записка и разработанное программное обеспечение.

Реализация базы данных выполняется с помощью выбранной СУБД и среды программирования, включающей функции работы с БД. Минимальная реализация системы подразумевает создание базы данных и запросов на SQL, осуществляющих выполнение тех функций, которые оговорены в задании. Дополнительные требования указываются непосредственно в задании на курсовую работу. В том случае, если система реализуется не полностью, например, отсутствуют некоторые ограничения целостности или функциональные возможности, это должно быть указано в пояснительной записке.

## СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ

1. Коннолли Т., Бегг К., Страчан А. Базы данных: проектирование, реализация, сопровождение. Теория и практика, 2-е изд. : пер. с англ. : уч. пос. – М.: Изд. дом "Вильямс", 2000. – 1120 с.
2. Системы управления базами данных : лабораторный практикум для студентов специальностей 1 40 01 01 «Программное обеспечение информационных технологий», 1 40 01 02 «Информационные системы и технологии»/ сост. И. А. Бухвалова. – Минск: БНТУ, 2006 – 56 с.
3. Дипломное проектирование: методические указания к выполнению дипломного проекта для студентов специальностей 1 40 01 01 Программное обеспечение информационных технологий», 1 40 01 02 «Информационные системы и технологии»/ сост. О. В. Бугай, И. А. Бухвалова., А. Т. Ковальков и др. – Минск: БНТУ, 2006 – 28с.

Учебное издание

## **БАЗЫ ДАННЫХ**

Методические указания  
к выполнению курсовой работы  
для студентов специальности  
1-40 01 01 «Программное обеспечение  
информационных технологий»

**С о с т а в и т е л ь**  
**БУХВАЛОВА** Ирина Антоновна

Подписано в печать 05.02.2013. Формат 60×84 <sup>1</sup>/<sub>16</sub>. Бумага офсетная. Ризография.  
Усл. печ. л. 2,09. Уч.-изд. л. 1,64. Тираж 100. Заказ 1129.

Издатель и полиграфическое исполнение: Белорусский национальный технический университет. ЛИ № 02330/0494349 от 16.03.2009. Пр. Независимости, 65. 220013, г. Минск.