

Лабораторная работа №2 (4 ч.)

Концептуальное и логическое проектирование реляционной БД

Модель "Сущность - Связь". Логическое проектирование. Нормализация отношений

1. Для проектирования БД выбрать предметную область (индивидуальное задание).и провести ее анализ.
2. Выполните инфологическое(концептуальное) проектирование базы данных для выбранной предметной области (ER-Модель). (Использовать облачный сервис <https://www.draw.io/>.)
3. Выполните логическое проектирование реляционную БД.
4. Выполнить составление реляционных нормализацию полученных отношений (до 3 НФ).

В процессе проектирования БД необходимо изучить предметную область, сформулировать и проанализировать требования к модели. Определить состав объектов предметной области и атрибутов, их характеризующих. Определить вычисляемые атрибуты, которые можно не хранить в БД;

1. Выделить необходимый набор сущностей, отражающих предметную область и информационные потребности пользователей.
2. Определить необходимый набор атрибутов каждой сущности, выделив идентифицирующие атрибуты.
5. Определить связи между сущностями.
6. Проанализировав структуру связей, исключить избыточные.
7. Определить множественность и условность связей.
8. Дать формулировку связей с точки зрения каждой участвующей сущности.
9. Формализовать связи вида 1:1, 1:M, M:N.
11. Построить ER-диаграмму модели базы данных.
12. Преобразовать ER-диаграмм в реляционные таблицы
13. Провести нормализацию полученных отношений (до 3 НФ).

Изучив по тексту теоретические сведения и примеры разработки ER-модели

!!См. далее по тексту лабораторной работы (с.14) пример проектирования БД компании, которая занимается издательской деятельностью.

I. Теоретические сведения

1.1 Основные определения

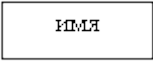

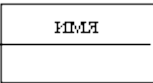




Сущность (Entity) – реальный либо воображаемый объект, имеющий существенное значение для рассматриваемой предметной области. Каждая сущность должна иметь наименование, выраженное существительным в единственном числе. Каждая сущность должна обладать уникальным идентификатором. Каждый экземпляр сущности должен однозначно идентифицироваться и отличаться от всех других экземпляров данного типа сущности.

Атрибут (Attribute) – любая характеристика сущности, значимая для рассматриваемой предметной области и предназначенная для квалификации, идентификации, классификации, количественной характеристики или выражения состояния сущности. Наименование атрибута должно быть выражено существительным в единственном числе.

Связь (Relationship) – поименованная ассоциация между сущностями, значимая для рассматриваемой предметной области.

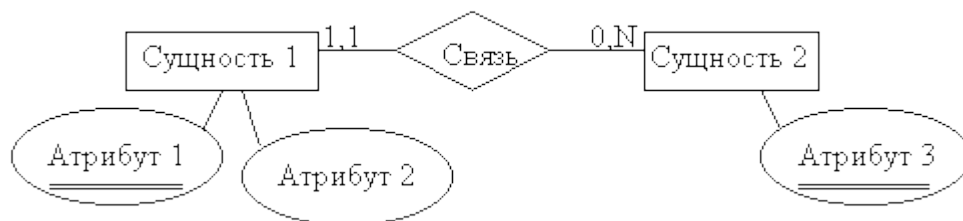
1.2 Обзор нотаций, используемых при построении диаграмм "сущность-связь"

1) Нотация Чена

Элемент диаграммы	Обозначает
	независимая сущность
	зависимая сущность
	родительская сущность в иерархической связи
	Связь
	идентифицирующая связь
	Атрибут
	первичный ключ

	внешний ключ (понятие внешнего ключа вводится в реляционной модели данных)
	многозначный атрибут
	получаемый (наследуемый) атрибут в иерархических связях

Связь соединяется с ассоциируемыми сущностями линиями. Возле каждой сущности на линии, соединяющей ее со связью, цифрами указывается класс принадлежности. Пример:

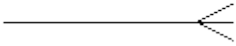
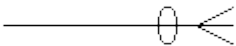



2) Нотация Мартина

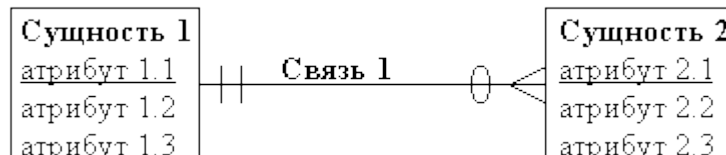
Элемент диаграммы	Обозначает
	независимая сущность
	зависимая сущность
	родительская сущность в иерархической связи

Список атрибутов приводится внутри прямоугольника, обозначающего сущность. Ключевые атрибуты подчеркиваются. Связи изображаются линиями, соединяющими сущности, вид линии в месте соединения с сущностью определяет кардинальность связи:

Обозначение	Кардинальность
	нет
	1,1
	0,1

	M,N
	0,N
	1,N

Имя связи указывается на линии ее обозначающей. Пример:



3) Нотация IDEF1X

Обозначения сущностей:

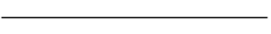
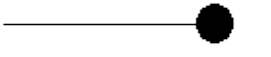
Элемент диаграммы	Обозначает
	независимая сущность
	зависимая сущность

Список атрибутов приводится внутри прямоугольника, обозначающего сущность. Атрибуты, составляющие ключ сущности, группируются в верхней части прямоугольника и отделяются горизонтальной чертой.

Обозначения связей:

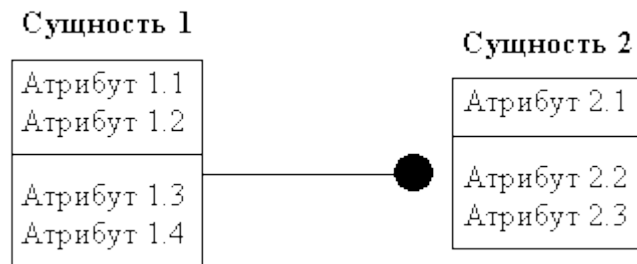
Элемент диаграммы	Обозначает
	идентифицирующая связь
	неидентифицирующая связь>

Обозначение кардинальности связей:

Элемент диаграммы	Обозначает
	1,1
	0,M

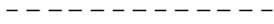

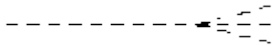
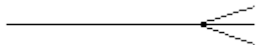
	0,1
	1,M
	точно N (N - произвольное число)

Пример:

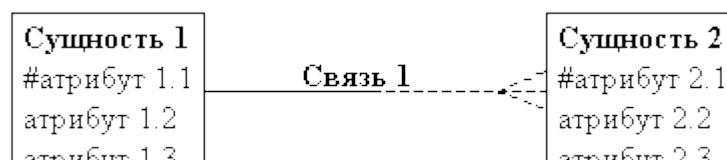


4) Нотация Баркера

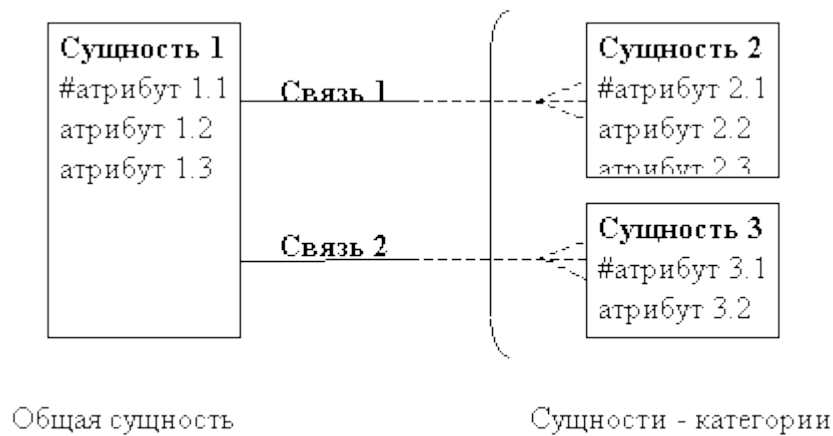
Сущности обозначаются прямоугольниками, внутри которых приводится список атрибутов. Ключевые атрибуты отмечаются символом # (решетка). Связи обозначаются линиями с именами, место соединения связи и сущности определяет кардинальность связи:

Обозначение	Кардинальность
	0,1
	1,1
	0,N
	1,N

Пример:



Для обозначения отношения категоризации вводится элемент "дуга":



II. Примеры разработки простой ER-модели

При разработке ER-моделей мы должны получить следующую информацию о предметной области:

1. Список сущностей предметной области.
2. Список атрибутов сущностей.
3. Описание взаимосвязей между сущностями.

Пример 1. Например, в ходе беседы с менеджером по продажам, выяснилось, что он (менеджер) считает, что проектируемая система должна выполнять следующие действия:

- Хранить информацию о покупателях.
- Печатать накладные на отпущенные товары.
- Следить за наличием товаров на складе.

Выделим все существительные в этих предложениях - это будут потенциальные кандидаты на сущности и атрибуты, и проанализируем их (непонятные термины будем выделять знаком вопроса):

- *Покупатель* - явный кандидат на сущность.
- *Накладная* - явный кандидат на сущность.
- *Товар* - явный кандидат на сущность
- *(?)Склад* - а вообще, сколько складов имеет фирма? Если несколько, то это будет кандидатом на новую сущность.
- *(?)Наличие товара* – это, скорее всего, атрибут, но атрибут какой сущности?

Сразу возникает очевидная связь между сущностями - "покупатели могут покупать много товаров" и "товары могут продаваться многим покупателям". Первый вариант диаграммы выглядит так:



Рис. 7

Задав дополнительные вопросы менеджеру, мы выяснили, что фирма имеет несколько складов. Причем, каждый товар может храниться на нескольких складах и быть проданным с любого склада.

Куда поместить сущности "Накладная" и "Склад" и с чем их связать? Спросим себя, как связаны эти сущности между собой и с сущностями "Покупатель" и "Товар"? Покупатели покупают товары, получая при этом накладные, в которые внесены данные о количестве и цене купленного товара. Каждый покупатель может получить несколько накладных. Каждая накладная обязана выписываться на одного покупателя. Каждая накладная обязана содержать несколько товаров (не бывает пустых накладных). Каждый товар, в свою очередь, может быть продан нескольким покупателям через несколько накладных. Кроме того, каждая накладная должна быть выписана с определенного склада, и с любого склада может быть выписано много накладных. Таким образом, после уточнения, диаграмма будет выглядеть следующим образом:

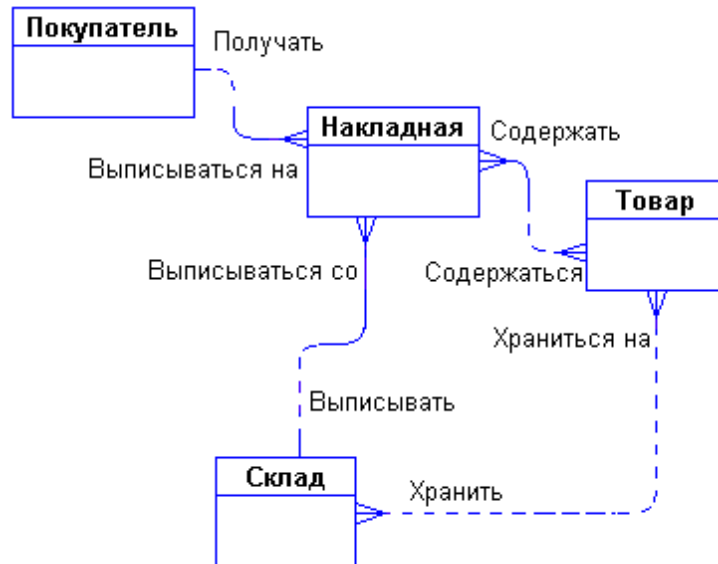


Рис. 8

Атрибуты сущности

- Каждый покупатель является юридическим лицом и имеет наименование, адрес, банковские реквизиты.
- Каждый товар имеет наименование, цену, а также характеризуется единицами измерения.

- Каждая накладная имеет уникальный номер, дату выписки, список товаров с количествами и ценами, а также общую сумму накладной. Накладная выписывается с определенного склада и на определенного покупателя.
- Каждый склад имеет свое наименование.
- Снова выпишем все существительные, которые будут потенциальными атрибутами, и проанализируем их:
- *Юридическое лицо* - термин риторический, мы не работаем с физическими лицами. Не обращаем внимания.
- *Наименование покупателя* - явная характеристика покупателя.
- *Адрес* - явная характеристика покупателя.
- *Банковские реквизиты* - явная характеристика покупателя.
- *Наименование товара* - явная характеристика товара.
- (?) *Цена товара* - похоже, что это характеристика товара. Отличается ли эта характеристика от цены в накладной?
- *Единица измерения* - явная характеристика товара.
- *Номер накладной* - явная уникальная характеристика накладной.
- *Дата накладной* - явная характеристика накладной.
- (?) *Список товаров в накладной* - список не может быть атрибутом. Вероятно, нужно выделить этот список в отдельную сущность.
- (?) *Количество товара в накладной* - это явная характеристика, но характеристика чего? Это характеристика не просто "товара", а "товара в накладной".
- (?) *Цена товара в накладной* - опять же это должна быть не просто характеристика товара, а характеристика товара в накладной. Но цена товара уже встречалась выше - это одно и то же?
- *Сумма накладной* - явная характеристика накладной. Эта характеристика не является независимой. Сумма накладной равна сумме стоимостей всех товаров, входящих в накладную.
- *Наименование склада* - явная характеристика склада.

В ходе дополнительной беседы с менеджером удалось прояснить различные понятия цен. **Оказалось, что каждый товар имеет некоторую текущую цену. Эта цена, по которой товар продается в данный момент. Естественно, что эта цена может меняться со временем.** Цена одного и того же товара в разных накладных, выписанных в разное время, может быть различной. **Таким образом, имеется две цены - цена товара в накладной и текущая цена товара.**

С возникающим понятием "Список товаров в накладной" все довольно ясно. Сущности "Накладная" и "Товар" связаны друг с другом отношением типа много-ко-многом. Такая связь, как мы отмечали ранее, должна быть расщеплена на две связи типа один-ко-многом. Для этого требуется дополнительная сущность.

Этой сущностью и будет сущность "Список товаров в накладной". Связь ее с сущностями "Накладная" и "Товар" характеризуется следующими фразами

- "каждая накладная обязана иметь несколько записей из списка товаров в накладной",

"каждая запись из списка товаров в накладной обязана включаться ровно в одну накладную",

"каждый товар может включаться в несколько записей из списка товаров в накладной",

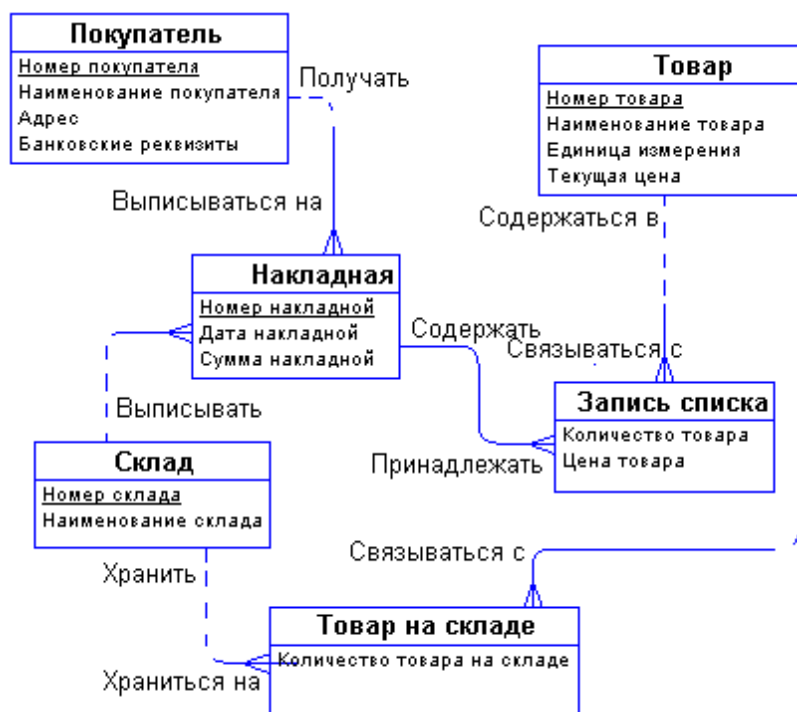
" каждая запись из списка товаров в накладной обязана быть связана ровно с одним товаром".

Атрибуты "Количество товара в накладной" и "Цена товара в накладной" являются атрибутами сущности " Список товаров в накладной".

Точно также поступим со связью, соединяющей сущности "Склад" и "Товар". Введем дополнительную сущность "Товар на складе".

Атрибутом этой сущности будет "Количество товара на складе". Таким образом, товар будет числиться на любом складе и количество его на каждом складе будет свое.

Теперь можно внести все это в диаграмму:



Пример 2. Построение диаграммы сущность-связь, предлагаемой предметной области, используя нотацию Чена и нотацию Мартина.

1 Описание сущностей предметной области «Золотая рыбка»

В результате анализа предметной области было выделено четыре сущности:

- 1) исполнитель;
- 2) автор;
- 3) посыльный, доставляющий заявки от автора к исполнителю;
- 4) заявка на исполнение желания;
- 5) желание, представляющее собой озвученную посыльным на берегу моря заявку, которая зарегистрирована исполнителем.

1.1 Сущность «Исполнитель»

Сущность «Исполнитель» характеризуется уникальным личным номером (главный ключ сущности), именем и годом рождения.

Таблица 1 – Атрибуты сущности «Исполнитель»

Атрибут	Описание	Тип, домен
<u>Номер</u>	Уникальный номер Ключевой атрибут.	Целое число
Год рождения	Год рождения	Целое число
Имя	Имя	Текст

1.2 Сущность «Автор»

Сущность «Автор» характеризуется личным номером (главный ключ сущности), фамилией, именем и датой рождения.

Таблица 2 – Атрибуты сущности «Автор»

Атрибут	Описание	Тип, домен
<u>Номер</u>	Уникальный номер автора в информационной системе. Ключевой атрибут.	Целое число
Год рождения	Год рождения автора	Год
Имя	Имя автора	Текст

1.3 Сущность «Посыльный»

Сущность «Посыльный» также характеризуется личным номером (главный ключ сущности), фамилией, именем и датой рождения.

Таблица 3 – Атрибуты сущности «Посыльный»

Атрибут	Описание	Тип, домен
<u>Номер</u>	Уникальный номер Ключевой атрибут.	Целое число
Год рождения	Год рождения автора	Год
Имя	Имя посыльного	Текст

1.4 Сущность «Заявка»

Сущность «Заявка» определяется числовым идентификатором (главный ключ сущности), датой формирования заявки и текстом заявки.

Сущность Заявка слабая, поскольку она не существует без сущностей Автор и Посыльный.

Таблица 4 – Атрибуты сущности «Заявка»

Атрибут	Описание	Тип
Номер	Уникальный номер	Целое число, Ключевой
Дата создания	Дата создания заявки	Дата и время
Текст	Текст заявки	Текст

1.5 Сущность «Желание»

Сущность «Желание» – слабая сущность, которая не существует без заявки. Сущность «Желание» характеризуется датой исполнения. Остальные атрибуты определяются через отношения сущности «Желание» с другими сущностями: «Заявка» и «Посыльный».

Атрибуты сущности приведены в таблице 5.

Таблица 5 – Атрибуты сущности «Желание»

Атрибут	Описание	Тип
Дата исполнения	Дата исполнения желания. Может быть пустым, если желание еще не исполнилось	Дата и время

2 Отношения

2.1 Связь «Автор»-«Заявка»

Автором может формулировать любое количество заявок, но у каждой заявки может быть только один автор: связь бинарная, тип связи – «один ко многим». Связь показана на рисунке 1.



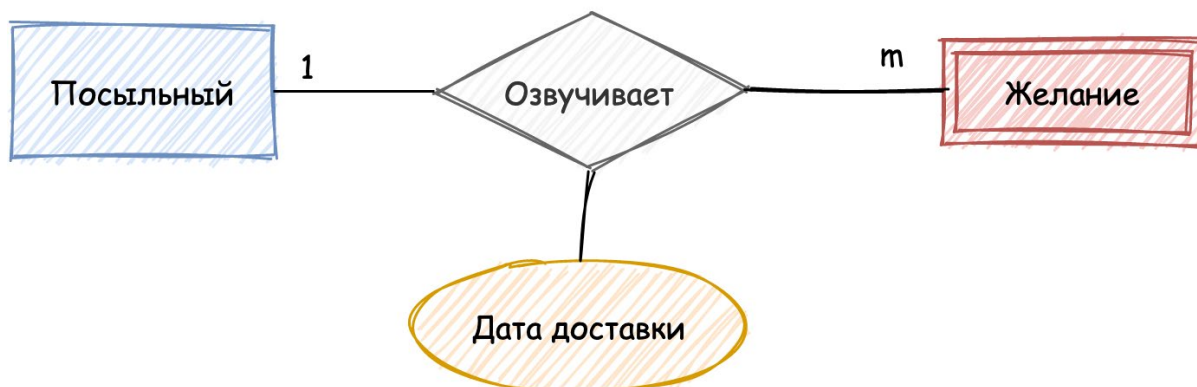
2.2 Связь «Посыльный»-«Заявка»

«Посыльный» может передавать любое количество заявок ($m \geq 0$), но у каждой заявки может быть только один исполнитель: связь бинарная, тип связи – «один ко многим». Связь показана на рисунке 2.



2.3 Связь «Посыльный-Желание»

Посыльный озвучивает заявку Автора и в этот момент заявка становится озвученным Желанием. Желание может быть озвучено только одним Посыльным. Желание без посыльного не существует. Связь имеет атрибут «Дата доставки» или «Дата озвучивания» Желания.



2.4 Связь «Исполнитель-Желание»

Исполнитель исполняет любое количество желаний. У каждого желания может быть только один исполнитель. Связи бинарная, тип «один ко многим». Связь показана на рисунке 4.



3 Диаграмма «Сущность-связь»

3.1 Нотация Чена

На рисунке 5 приведена диаграмма «Сущность-связь» предметной области в нотации Чена.

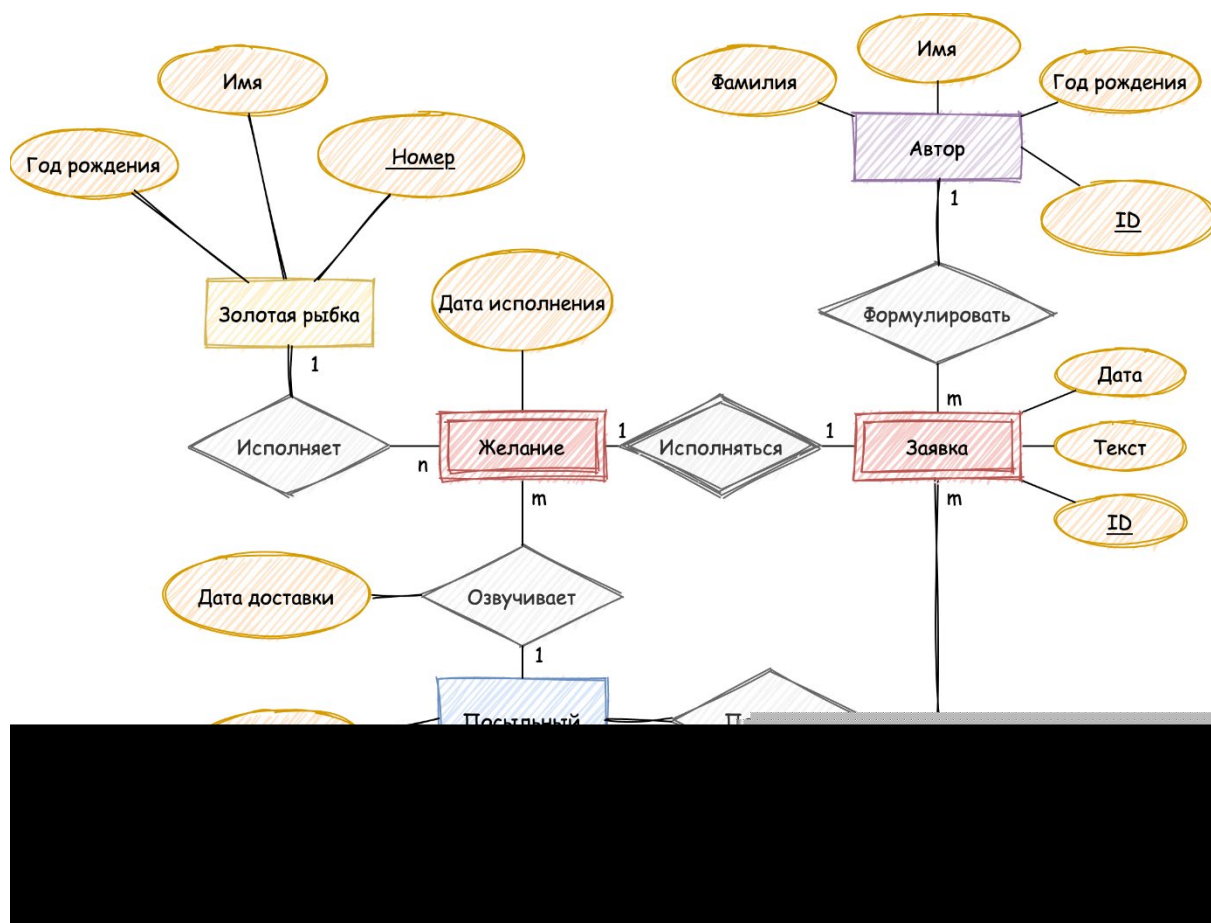


Рисунок 1 – Диаграмма сущность-связь предметной области в нотации Чена

3.2 Нотация Мартина



Рисунок 2 – Диаграмма сущность-связь предметной области в нотации Мартина

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

1. Инфологическое проектирование

Анализ предметной области

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:

- ✓ 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.

Для инфологического проектирования воспользуемся методом «сущность-связь». Для того, чтобы представить, как устроена предметная область нужно задать множество объектов реального мира (главная проблема что считать объектом).

Объект – семантическое понятие, которое может быть полезно при обсуждении устройств реального мира.

Сущность реального мира – объекты – не обязательно материальны – важно понятие существенно и различимо для других.

Между объектами могут возникать **связи трех видов:**

- ✓ один к одному 1 : 1 (пациент: место в палате);
- ✓ один к многим 1 : n и многие к одному n : 1;
- ✓ многие ко многим n : n (пациент : хирург).

Выделим базовые сущности этой предметной области:

- ✓ **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.
- ✓ **Авторы.** *Атрибуты авторов* – ФИО, ИНН (индивидуальный номер налогоплательщика), паспортные данные, домашний адрес, телефоны. Для авторов необходимо хранить сведения о написанных книгах.
- ✓ **Книги.** *Атрибуты книги* – авторы, название, тираж, дата выхода, цена одного экземпляра, общие затраты на издание, авторский гонорар.
- ✓ **Контракты** будем рассматривать как связь между авторами, книгами и менеджерами. *Атрибуты контракта* – номер, дата подписания и участники.

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 ниже (базовые сущности на рисунках выделены полужирным шрифтом).

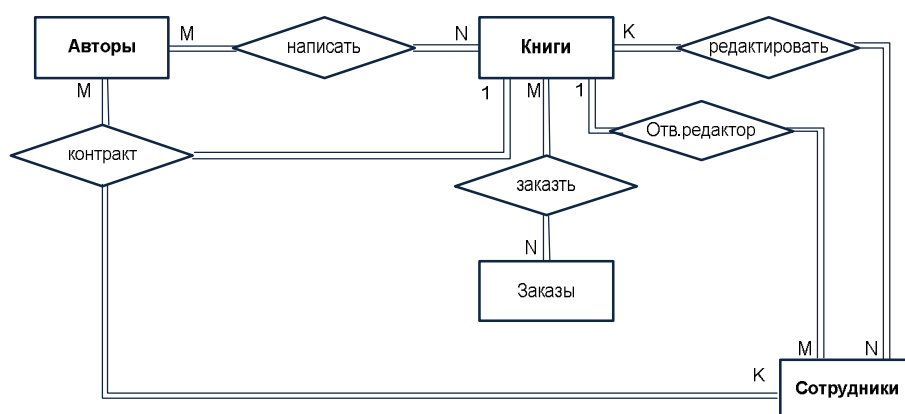
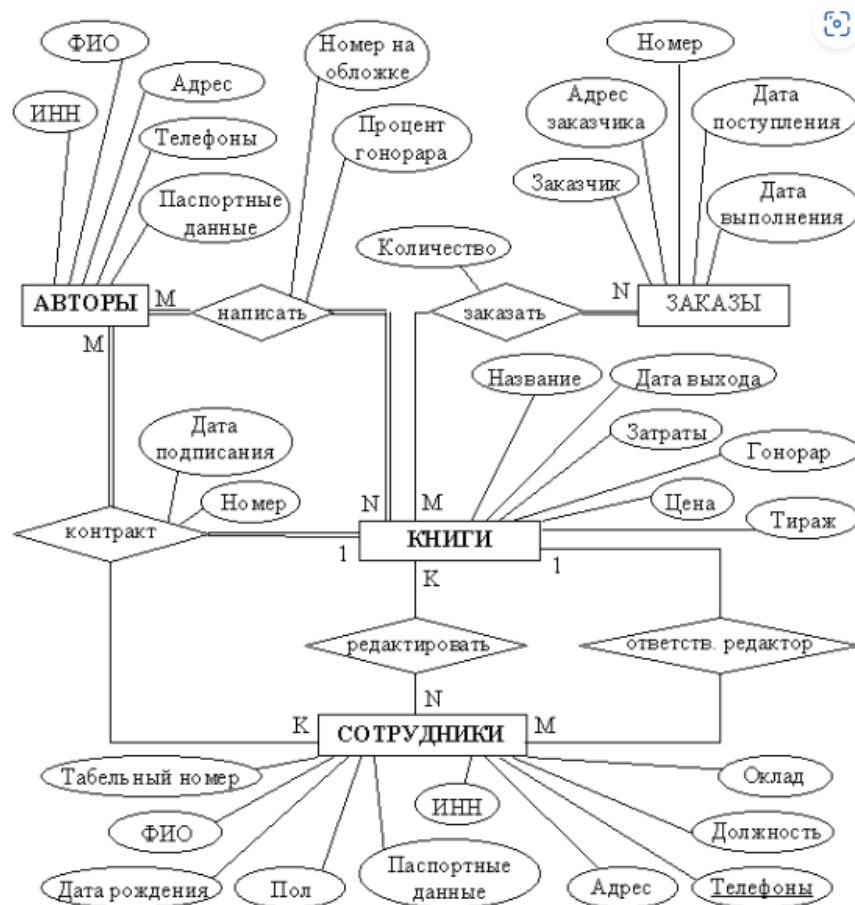


Рис. 1.1 – ER–диаграмма издательской компании

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.



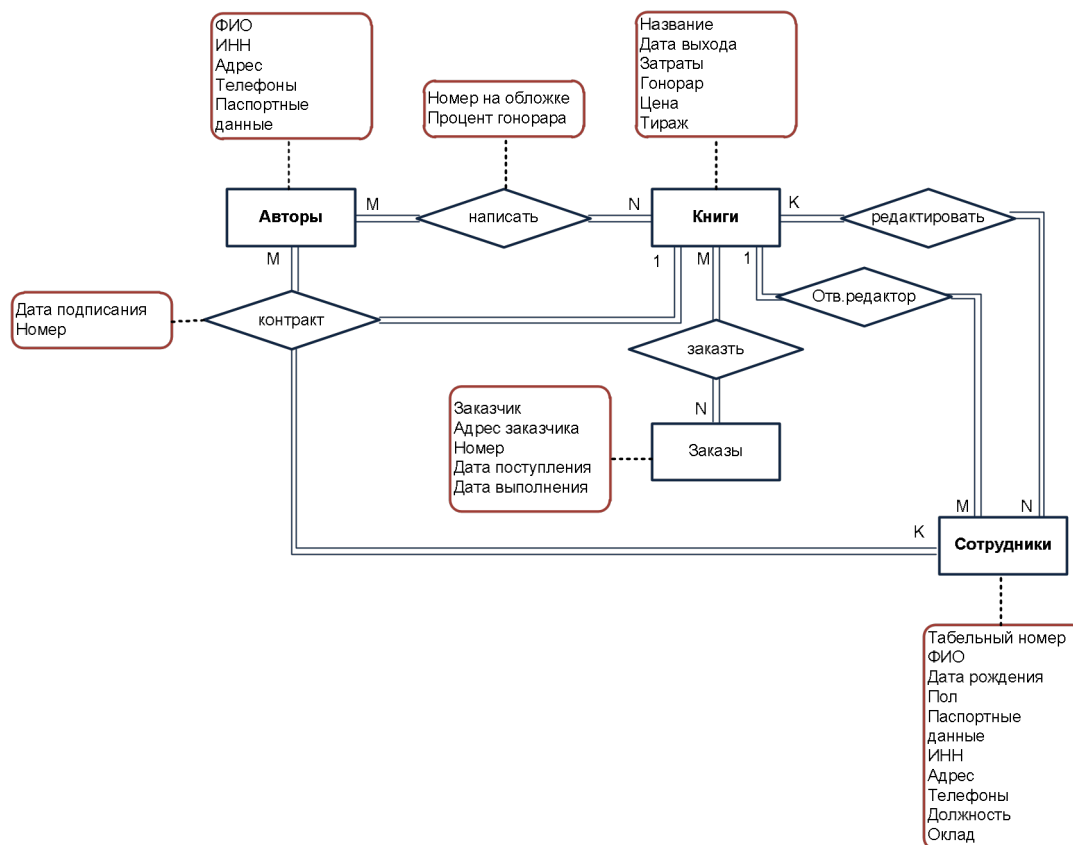


Рис.1.2 – Уточнённая ER–диаграмма издательской компании

2. Логическое проектирование реляционной БД

Преобразование ER–диаграммы в схему БД выполняется путем сопоставления каждой сущности и каждой связи, имеющей атрибуты, отношения (таблицы БД).

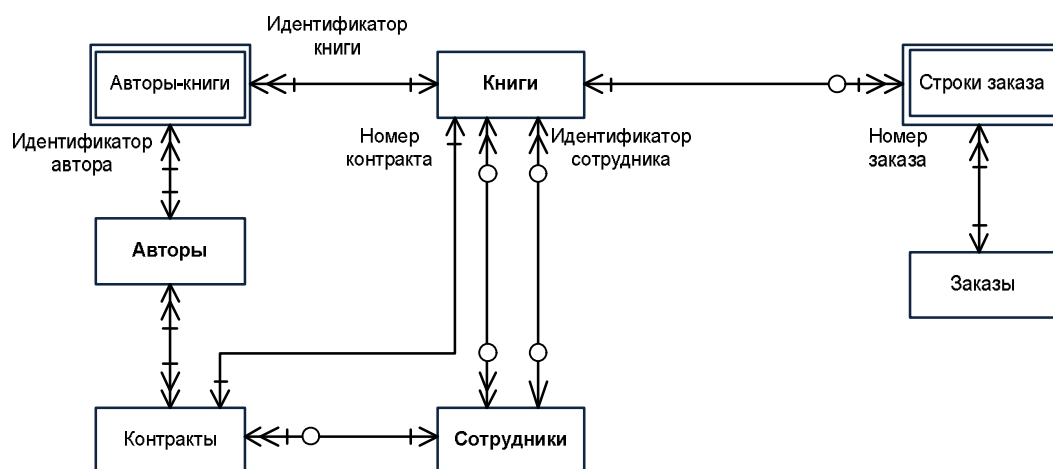


Схема реляционной БД, полученная из ER–диаграммы издательской компании

На схеме 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.

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 n:m (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.

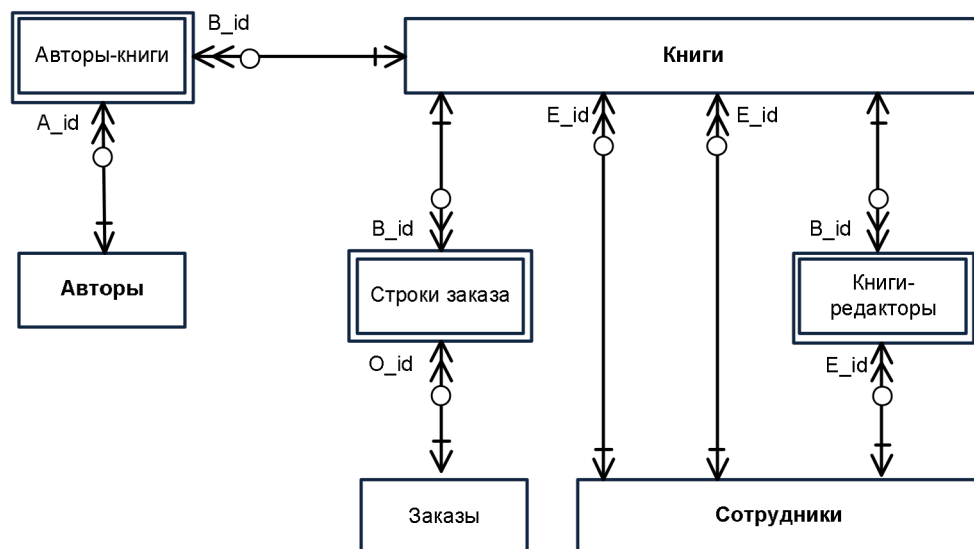


Рис. 1.5 – Уточнённая схема реляционной БД издательской компании

Схема на рисунке 1.5 содержит цикл «сотрудники–книги–сотрудники». Цикл допустим только в том случае, если связи, входящие в него, независимы друг от друга. Примем для нашей предметной области, 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.

Составление реляционных отношений

Каждое реляционное отношение соответствует одной сущности (объекту предметной области) 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 **Контракт**: 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 таблице 1.1-1.7. Для каждого отношения указаны атрибуты с их внутренним названием, типом и длиной. Типы данных обозначаются так: **N** – числовой, **C** – символьный, **D** – дата (последний имеет стандартную длину, зависящую от СУБД, поэтому она не указывается).

Таблица 1.1 – Схема отношения СОТРУДНИКИ (Employees)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечания</i>
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
Адрес	E_ADDR	C(50)	
Телефоны	E_TEL	C(30)	многозначное поле

Таблица 1.2 – Схема отношения КНИГИ (Books)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечания</i>
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 1.3 – Схема отношения АВТОРЫ (Authors)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечания</i>
Код автора	A_ID	N(4)	суррогатный первичный ключ
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 1.4 – Схема отношения ЗАКАЗЫ (Orders)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечания</i>
Номер заказа	O_ID	N(6)	первичный ключ
Заказчик	O_COMPANY	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 1.5 – Схема отношения КНИГИ–АВТОРЫ (Titles)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечания</i>
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 1.6 – 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 1.7 – 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

Нормализация полученных отношений (до 4НФ)

✓ **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:

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

Разделим атрибуты «Фамилия, имя, отчество» на три атрибута «Фамилия», «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. Не ключевые атрибуты этих отношений функционально полно зависят от первичных ключей.

3НФ

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

После проведённых преобразований схема БД выглядит так (рисунок 1.6):

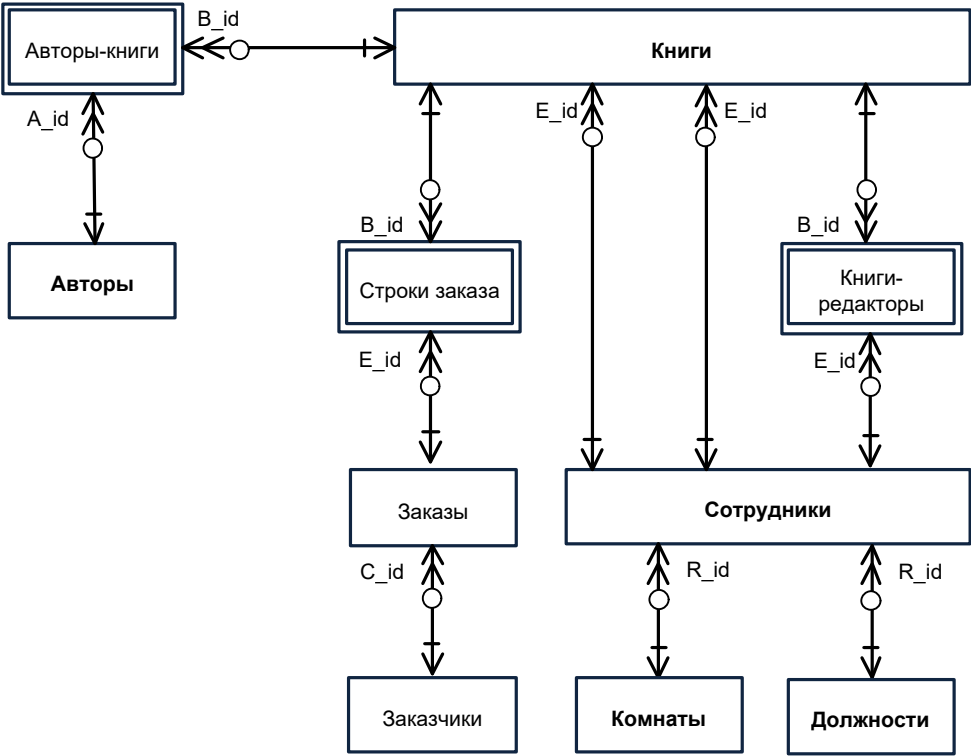


Рис. 1.6 – Окончательная схема РБД издательской компании

Окончательные схемы отношений базы данных с указанием ключей и других ограничений целостности приведены в табл. 1.8–1.17.

Таблица 1.8 – 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 1.9 – Схема отношения КОМНАТЫ (Rooms)

Содержание поля	Имя поля	Тип, длина	Примечания
-----------------	----------	------------	------------

Номер комнаты	R_NO	N(3)	составной первичный ключ
Номер телефона	R_TEL	C(10)	

Таблица 1.10 – 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
Дата выдачи паспорта	E_PDATE	D	обязательное поле
Адрес	E_ADDR	C(50)	

Таблица 1.11 – 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 1.12 – 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 поле

Таблица 1.13 – 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)
Остаток тиража	B_REST	N(5)	производное поле

Таблица 1.14 – Схема отношения ЗАКАЗЫ (Orders)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечания</i>
Номер заказа	O_ID	N(6)	первичный ключ
Код заказчика	O_COMPANY	N(4)	внешний ключ (к Customers)
Дата поступления заказа	O_DATE	D	обязательное поле
Дата выполнения заказа	O_READY	D	

Таблица 1.15 – Схема отношения КНИГИ–АВТОРЫ (Titles)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечания</i>
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 1.16 – Схема отношения СТРОКИ ЗАКАЗА (Items)

<i>Содержание поля</i>	<i>Имя поля</i>	<i>Тип, длина</i>	<i>Примечания</i>
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

Таблица 1.17 – Схема отношения КНИГИ–РЕДАКТОРЫ (Editors)

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

1.4.4. Определение дополнительных ограничений целостности

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

- ✓ Значения всех числовых атрибутов – больше 0 (или null, если атрибут необязателен).
- ✓ Область значений атрибута Sex отношения EMPLOYEES – символы «М» и «Ж».
- ✓ Отношение ROOMS не имеет первичного ключа, но комбинация значений (R_no, Tel) уникальна.
- ✓ В отношении TITLES порядковые номера авторов на обложке одной книги должны идти подряд, начиная с 1.
- ✓ В отношении TITLES сумма процентов гонорара по одной книге равна 100.

Ограничения (4,5) нельзя реализовать в схеме отношения. В реальных БД подобные ограничения целостности реализуются программно (через внешнее приложение или специальную процедуру контроля данных).

