

Практика 1

Проектирование и
нормализация БД

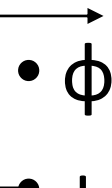
Требования к БД

- Целостность и согласованность
 - Целостность БД — соответствие имеющейся в БД информации её внутренней логике, структуре и явно заданным правилам.
 - Согласованность – см. непротиворечивость.
- Отказоустойчивость
- Защита и разграничение доступа

Задание

Рассмотрим на примере, потом дам аналогичное задание

Этапы проектирования

- инфологическое (концептуальное) проектирование;
 - определение требований к операционной обстановке, в которой будет функционировать информационная система;
 - выбор системы управления базой данных (СУБД) и других инструментальных программных средств;
 - логическое проектирование БД;
 - физическое проектирование БД;
 - + нормализация в рамках разумного
- 

Инфологическое проектирование

– построение семантической модели предметной области, то есть информационной модели наиболее высокого уровня абстракции.

Такая модель создается без ориентации на какую-либо конкретную СУБД и модель данных. Конкретный вид и содержание концептуальной модели базы данных определяется выбранными для этого формальным аппаратом. Обычно используют графические нотации, подобные ER-диаграммам.

Entity-Relationship Diagrams

Какие основные понятия используются в ER-модели?

Entity-Relationship Diagrams

При построении ER-модели используются следующие принципы:

- сущности на диаграмме представляются прямоугольниками;
- каждый прямоугольник может иметь различные визуальные атрибуты;
- каждой сущности должно быть присвоено уникальное имя;
- имена сущностей необходимо задавать в единственном числе;
- связи на диаграмме представляются линиями, идущими от одной сущности (таблицы) к другой;
- каждой связи присваивается уникальное имя.

Логическое проектирование

– создание схемы базы данных на основе конкретной модели базы 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. Преобразование концептуальной модели в логическую модель, как правило, осуществляется по формальным правилам.

На этапе логического проектирования учитывается специфика конкретной модели данных, но может не учитываться специфика конкретной СУБД.

Формальные правила

- Если степень бинарной связи равна 1:1 и класс принадлежности обеих сущностей обязательный, то требуется только одно отношение. Первичным ключом этого отношения может быть ключ любой из этих двух сущностей. В этом случае гарантируется однократное появление каждого значения ключа в любом экземпляре отношения.
- Если степень бинарной связи равна 1:1 и класс одной из сущностей необязательный, то необходимо построение двух отношений, под каждую сущность необходимо выделить одно отношение. Ключ сущности, для которого класс принадлежности является необязательным, добавляется в качестве атрибута в отношение, выделенное для сущности с обязательным классом принадлежности.

Формальные правила

- Если степень бинарной связи равна 1:1 и класс принадлежности ни одной из сущностей не является обязательным, то используется три отношения — по одному для каждой сущности — ключи которых служат в качестве первичных в соответствующих отношениях и одного для связи. Отношение, выделенное для связи, будет иметь по одному ключу сущности от каждой сущности.
- Если степень бинарной связи равна 1:M и класс принадлежности M-связной сущности обязательный, то достаточно использовать два отношения: по одному на каждую сущность, при условии, что ключ сущности служит в качестве первичного ключа для соответствующего отношения. Ключ же односвязной сущности должен быть добавлен как атрибут в отношение, отводимое M-связной сущности.

Формальные правила

- Если степень бинарной связи равна 1:M и класс принадлежности M-связной сущности необязателен, то необходимо использовать три отношения: по одному на сущность и одно для связи. Связь должна иметь среди своих атрибутов ключ сущности от каждой сущности.
- Если степень бинарной связи равна M:M, то для хранения данных необходимо три отношения: по одному на сущность и одно для связи. Ключи сущности входят в связь. Если одна из сущностей вырождена, то два отношения (т.е. достаточно будет двух таблиц).
- В случае трехсторонней связи необходимо использовать четыре отношения: по одному на сущность и одно для связи. Отношение, порождаемое связью, имеет в себе среди атрибутов ключи сущности от каждой сущности.

Обозначения связей



One



Many



One (and only one)



Zero or one



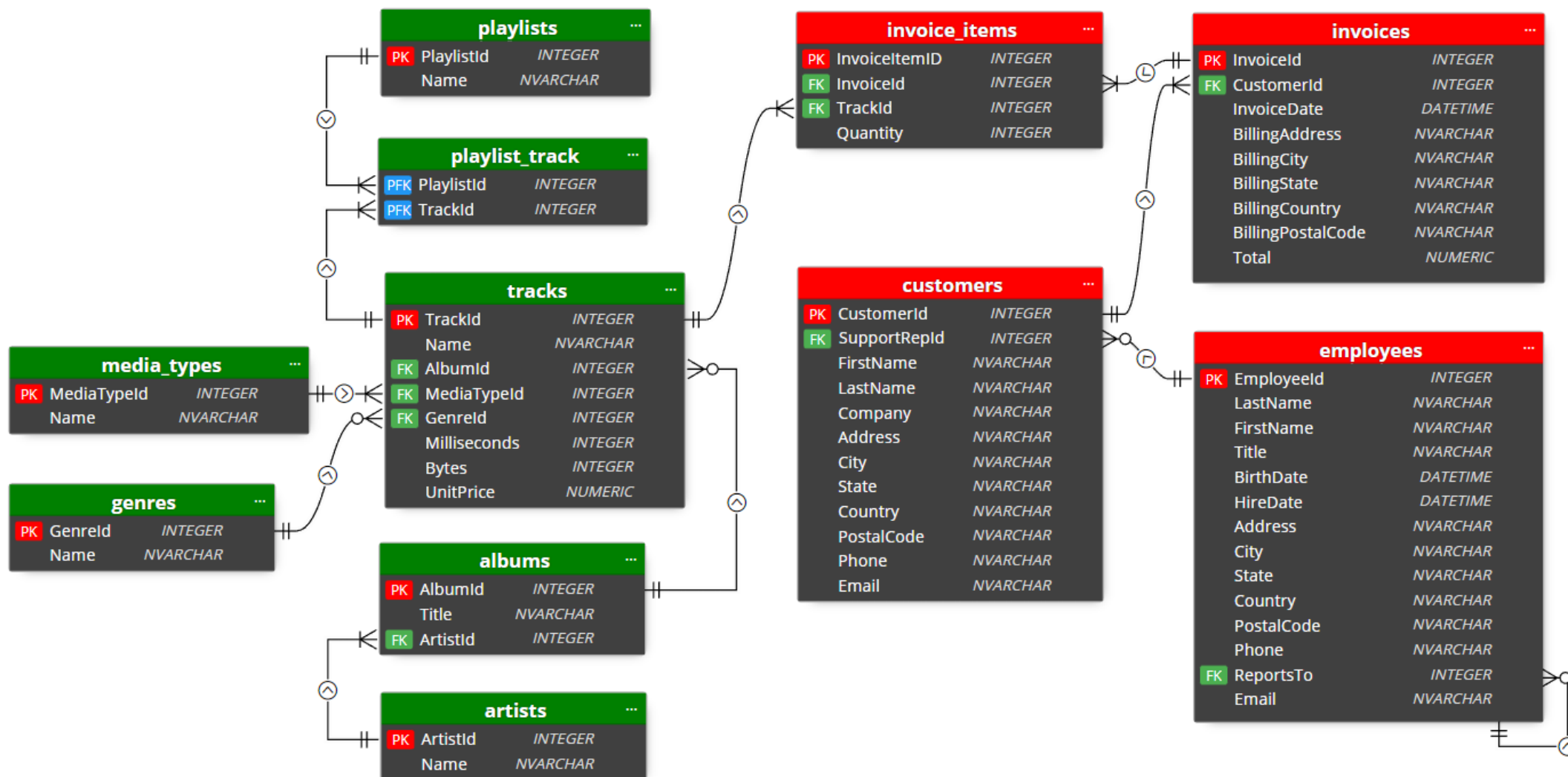
One or many



Zero or many

- Существует несколько различных нотаций
- Чаще на практике применяется нотация Мартина

Пример резултата



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.

На примере

Возьмем базу данных компании, которая занимается издательской деятельностью.

Проведем 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.

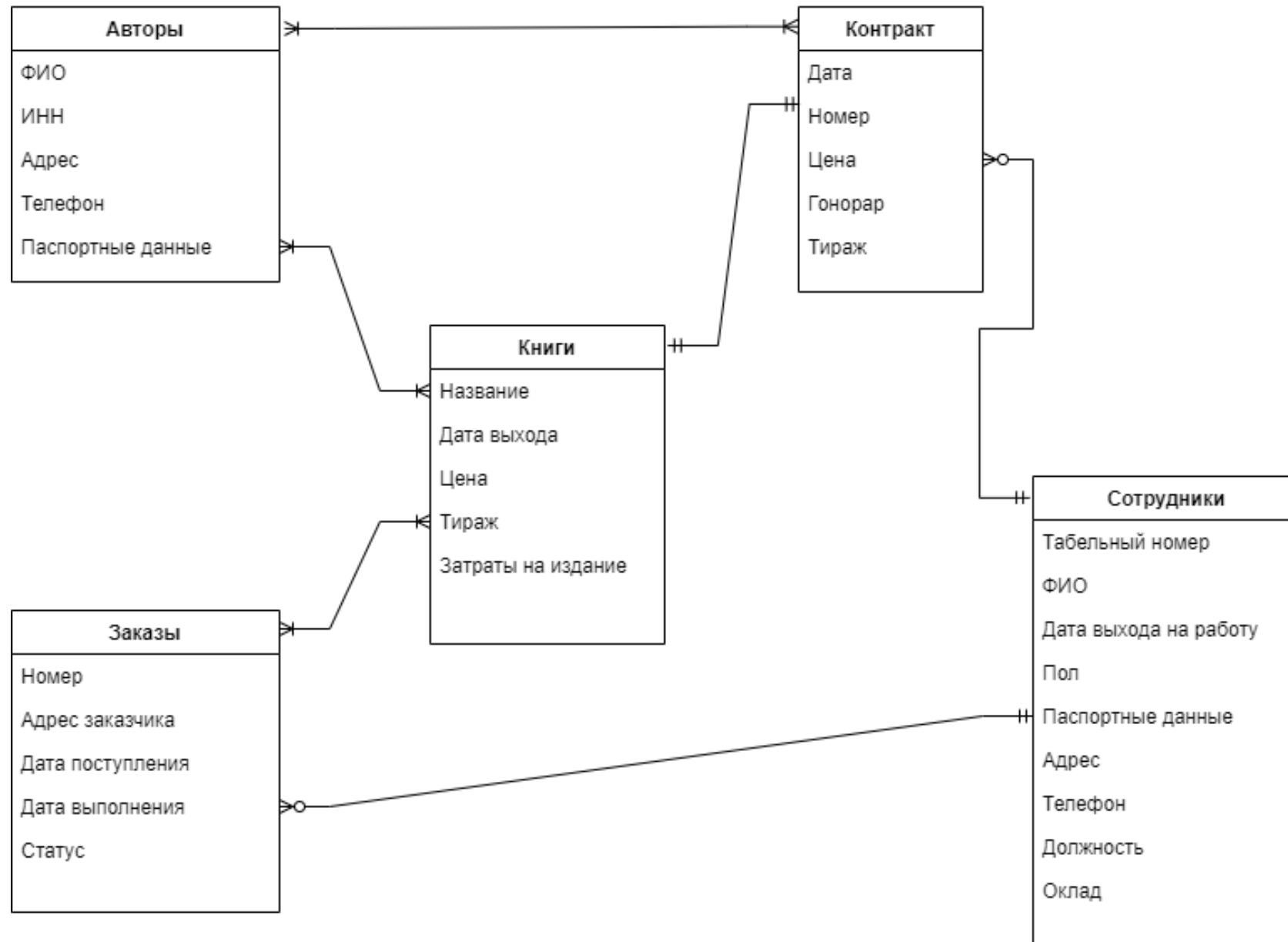
Выделим базовые сущности этой предметной области:

- 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 вместе с книгами, т.к. каждая книга связана непосредственно с одним контрактом.
- Заказы на книги. Необходимо хранить номер заказа, заказчика, адрес заказчика, дату поступления заказа, дату выполнения, список заказанных книг с указанием количества экземпляров.

В соответствии с предметной областью система строится с учётом следующих особенностей:

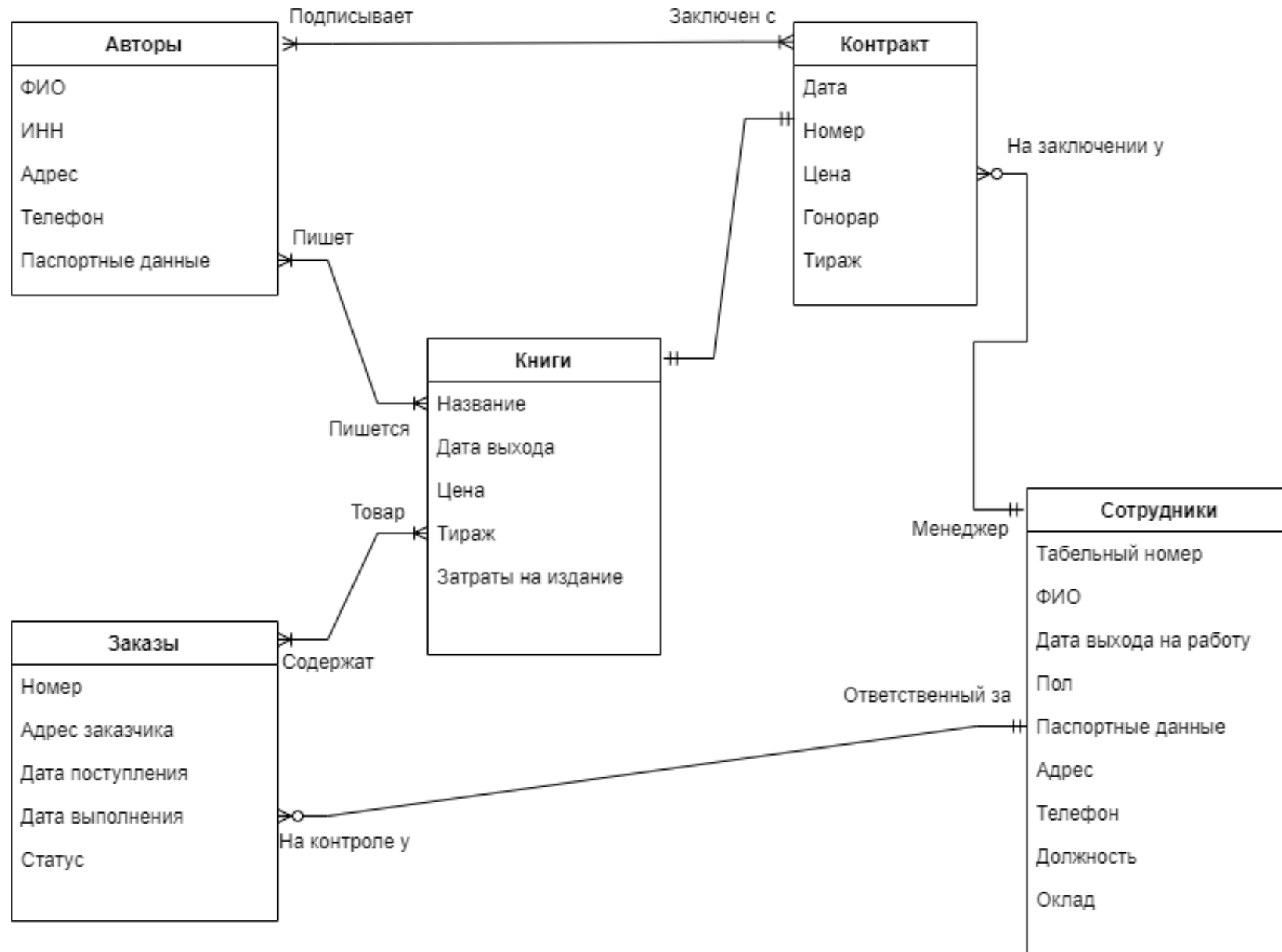
- каждая книга издаётся в рамках контракта;
- книга может быть написана несколькими авторами;
- контракт подписывается одним менеджером и всеми авторами книги;
- каждый автор может написать несколько книг (по разным контрактам);
- если сотрудник является редактором, то он может работать одновременно над несколькими книгами;
- у каждой книги может быть несколько редакторов;
- каждый заказ оформляется на одного заказчика;
- в заказе на покупку может быть перечислено несколько книг.

ER-диаграмма



Чего не хватает на схеме?

ER-диаграмма

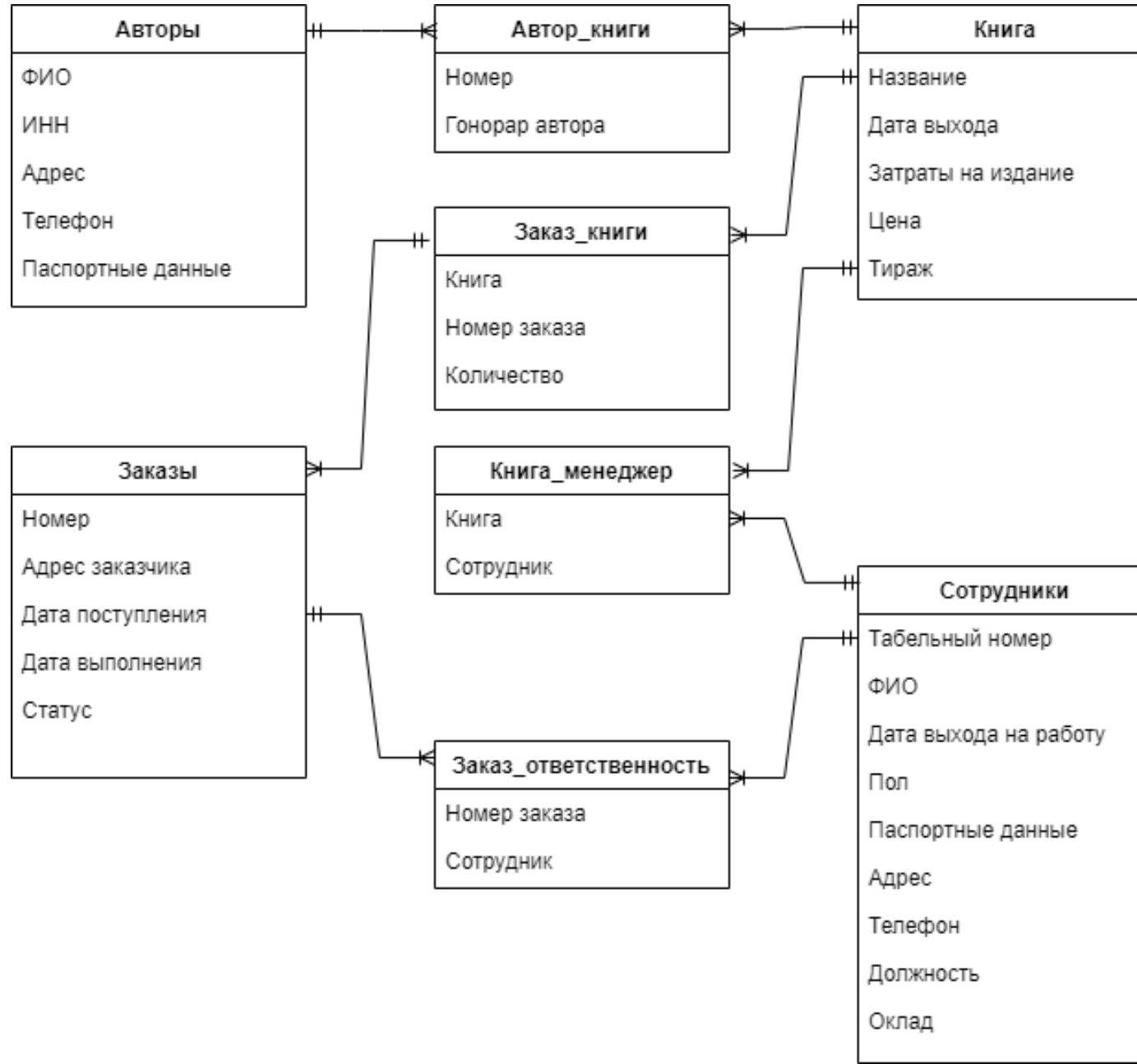


Преобразование ER–диаграммы в схему базы данных

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

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

Дополненная ER-диаграмма



Что осталось до логической модели?

Что осталось до логической модели?

- Первичные и внешние ключи

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 (например ID записи).

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).

Что ещё осталось до логической модели?

- Первичные и внешние ключи
- Типы данных
- Возможно доп. информация (not null, unique и т.д.)

Где это все строить?

Где угодно:

- Microsoft Visio
- Lucidchart
- Draw.io

Специализированное ПО:

- ERwin
- Dbeaver
- Luna Modeler (www.datensen.com)

Нормализация

– это процесс организации данных в базе данных, включающий создание таблиц и установление отношений между ними в соответствии с правилами, которые обеспечивают защиту данных и делают базу данных более гибкой, устраняя избыточность и несогласованные зависимости.

Нормализация

Отношение находится в 1НФ, если все его атрибуты являются простыми, все используемые домены должны содержать только скалярные значения. Не должно быть повторений строк в таблице.

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.

В реальных БД сложные атрибуты разбиваются на простые, если:

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

Нормализация

Отношение находится во 2НФ, если оно находится в 1НФ и каждый не ключевой атрибут неприводимо зависит от первичного ключа.

Неприводимость означает, что в составе потенциального ключа нет подмножества атрибутов, от которого можно вывести функциональную зависимость.

Отношение находится в 3НФ, если оно находится во 2НФ, и отсутствуют транзитивные функциональные зависимости неключевых атрибутов от ключевых.

Задание

БД – «Структура института (только бакалавриат)».

Задача: хранение информации по студентам, преподавателям и предметам *с её периодическим редактированием*.

Есть кафедры, дисциплины, преподаватели, студенты (организованные по группам), ведомости.

По кафедрам имеется следующая информация: название кафедры, кто заведующий, направления подготовки, номер корпуса, номер кабинета, телефон.

Для каждого преподавателя известно: ФИО, должность, кафедра, дата рождения, год поступления на работу, стаж на этом месте, пол, адрес, телефон, ставка, какие дисциплины преподает.

Задание

По студентам известно: номер студенческого, ФИО, курс, группа, направление подготовки, дата рождения, пол, адрес, телефон, форма обучения (бюджетное/платное).

У дисциплин известны название, кол-во часов, кто ведёт (может быть несколько преподавателей), вид итогового контроля. Так, например, у предмета «мат. основы БД» осенний и весенний семестры стоит различать.

В ведомости указаны: дисциплина, преподаватель, студент, оценка, дата.

Задание

Также нужно реализовать следующие механизмы:

- У студентов старших курсов есть научные руководители (нужно уметь определять, кто у кого научник, какая тема диплома)
- В группах есть старосты (нужно уметь узнавать старосту по номеру группы или наоборот)
- Необязательно: придумайте, как реализовать следующее ограничение – оценку в ведомость по дисциплине может ставить только преподаватель, который её ведет (необязательно у этого студента)

Задание

1. Выполнить анализ предметной области (частично сделано)
2. Построить инфологическую модель базы данных
3. Построить даталогическую модель базы данных
4. Нормализация до 3НФ (доп. вопрос – полезна ли она?)