Модели данных

A2. Нормализация реляционных данных

Московский государственный университет
имени Н.Э. Баумана

Факультет ИБМ

Июль 2024 года

Москва

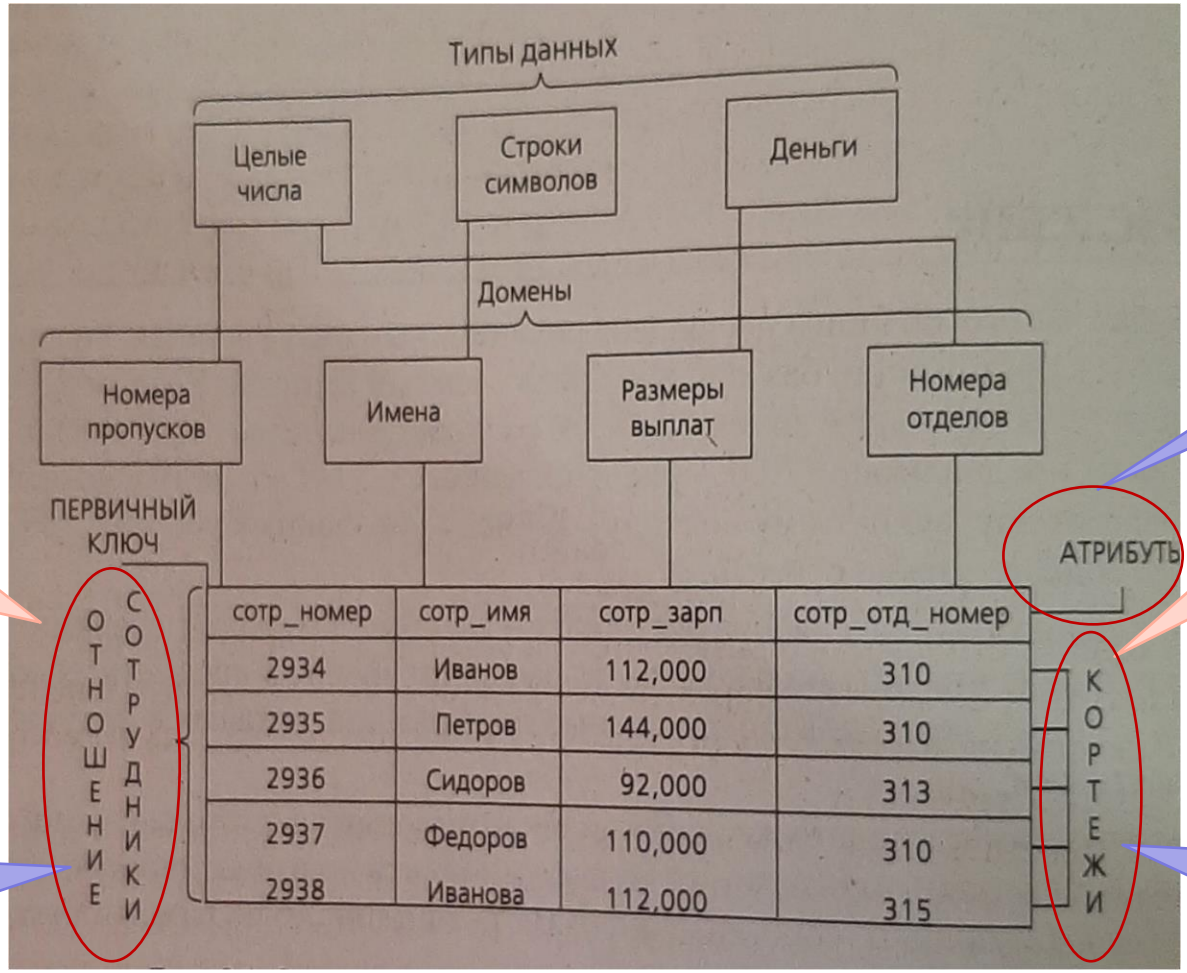
Курс «Моделирование данных»

2. Нормализация реляционной модели данных

- Нормализация реляционных моделей данных
- 1-ая нормальная форма
- 2-ая нормальная форма
- 3-ья нормальная форма
- Прочие нормальные формы

- Вопросы для самопроверки

Соотношение понятий реляционного подхода, ER-модели и баз данных



Колонки

Сущность

Экземпляры сущности

Таблица

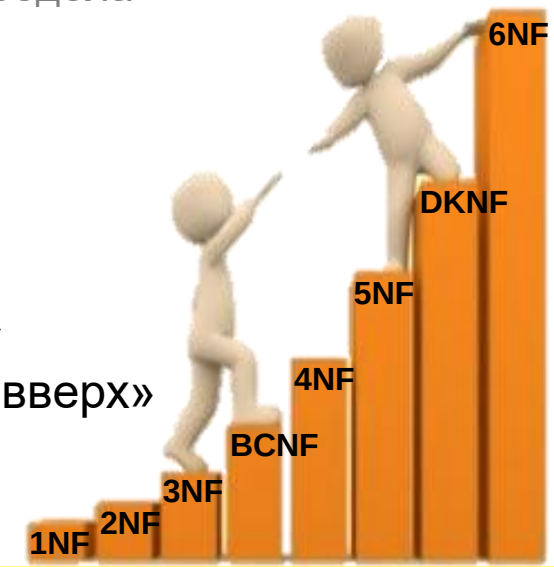
Записи (строки)

С.Д. Кузнецов. Базы данных: языки и модели. 2008 ©

Нормализация отношений

Нормализация отношений – метод моделирования и проектирования реляционных БД путём приведения структуры и состава данных в соответствие с определёнными правилами и требованиями, к эталонному виду.

- Зачем нужна нормализация отношений?
  - снижение избыточности данных
  - устранение аномалий ведения данных
  - повышение гибкости
  - увеличение производительности до определённого предела
  - повышение качества данных
- Нормальные формы – наборы *расширяемых* правил и требований нормализации отношений
  - сейчас существует 8 нормальных форм
  - на практике достаточна нормализация до 3NF
- Контроль ЛМД при проектировании «сверху – вниз»
- Создание ЛКМ / ФМД при проектировании «снизу – вверх»
  - часто приводит к разбиению таблицы без потерь на несколько таблиц с установлением связей



Избыточность и аномалии изменения

Избыточность данных возникает в базах данных, когда одни и те же значения определённой колонки повторяются в разных строках таблицы, что приводит к аномалиям изменения.

При изменении данных (добавлении, обновлении, удалении) может возникнуть их несогласованность.

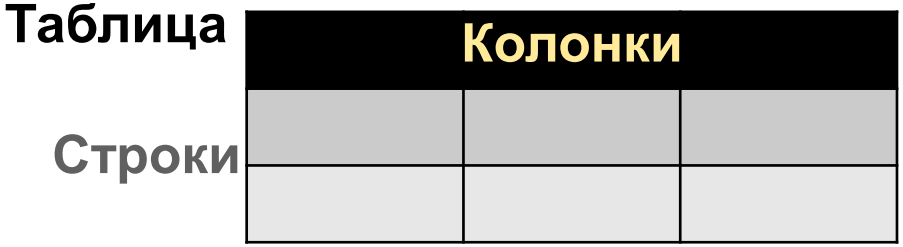
Некоторые называют это нулевой нормальной формой.

Начальные условия для нормализации

Прежде всего, согласно реляционной теории данные должны быть, представлены в виде кортежей и атрибутов отношений, а реляционные базы данных – в виде строк и колонок таблицы.

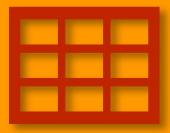
При этом

- Не упорядочены кортежи (строки таблицы)
- Фиксирован состав атрибутов (колонок таблицы)
- Не упорядочены атрибуты (колонки таблицы)



Некоторые называют это нулевой нормальной формой.

Не упорядочены кортежи (строки таблицы)



В реляционной теории отношения состоят из неупорядоченных кортежей, поэтому в реляционных БД записи хранятся в произвольном порядке.

- Кортежи (строки) не упорядочены по значению к-л атрибута
- Но строки можно сортировать по атрибутам при выборке

- Нельзя выбрать *n*-ую строку из таблицы
- Но можно выдать *n* строк из результата запроса, в частности одну

Нельзя полагаться на физический порядок строк в реляционной БД.

Фиксирован состав атрибутов (колонок таблицы)

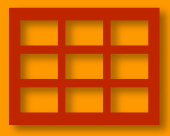
В реляционной теории фиксирован состав атрибутов отношения, поэтому в реляционной БД фиксирован состав колонок таблицы.

- В процессе работы с БД у таблиц фиксирован состав колонок
- Имена колонок – свойства сущности, а не значения свойств
- Изменения данных, например, по времени, разворачивайте по строкам
- Не разворачивайте изменения по колонкам

- Но состав колонок можно изменить при реорганизации БД
- Можно резервировать колонки с необязательными значениями
- Можно применить универсальные модели данных, напр., EVA

Реляционная модель не поддерживает переменный состав атрибутов.

Не упорядочены атрибуты (колонки таблицы)

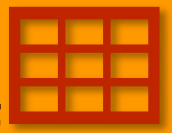


В реляционной теории состав атрибутов отношения не упорядочен, поэтому в реляционной БД явный порядок колонок не поддерживается.

- Расположение колонок таблицы скрыто от пользователя
- Описание из словаря БД будет отсортировано по именам колонок
- Данные выбираются по именам колонок из таблицы
- Нельзя выбрать *n*-ую колонку из таблицы
- Но можно сослаться на *n*-ое поле в рамках запроса

Нельзя полагаться на порядок колонок в реляционной таблице.

1-ая нормальная форма

1NF 

Отношение (таблица) находится в 1NF, когда оно соответствуют начальным условиям для нормализации реляционных данных, и в нём:

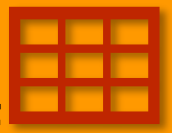
- нет кортежей (строк) с дублирующим содержимым
- семантически однозначные атрибуты (колонки) с атомарными, однотипными значениями

и накладывается ряд дополнительных требований к атрибутам, которые влияют на фильтрацию и агрегирование значений:

- одинаковая детальность значений (первичные данные и итоги)
- одинаковые единицы измерения (км/час или миль/час)
- согласованные позиции справочников
- определённый формат значений (тыс. руб. или руб. коп.)

Всё внимание к соблюдению реляционных принципов.

Нет кортежей (строк) с дублирующим содержимым

1NF 

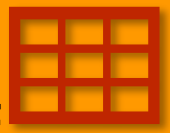
Каждый кортеж (строка таблицы) должен содержать уникальное значение.

Табельный номер	Фамилия, имя, отчество сотрудника	Подразделение
1234	Иванов Илья Степанович	Отдел 75
~~1234~~	~~Илья Степанович Иванов~~	~~Отдел 75~~
1235	Иванов Пётр Семёнович	Отдел 76
~~1235~~	~~Иванов Пётр Семёнович~~	~~Отдел 76~~

Реальная дедубликация сложнее, относится к управлению качеством данных.

Семантически однозначные атрибуты с атомарными, однотипными значениями

1NF



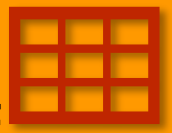
Нарушение этих требований 1NF приводит:

1) **перегруженные атрибуты –** составные и многозначные атрибуты (например: несколько разнотипных значений или список)
2) **наложенные атрибуты –** семантически неоднозначные, возможно однотипные значения (например: наименование организации или ФИО)

- Приемлемые при презентации/ визуализации данных (отчёты, панели мониторинга, слайды), но могут восприниматься как **дефекты качества данных** при сборе/ обработке/ анализе данных
- Специальный признак трактовки значений нужен для наложенных атрибутов

Это может провоцировать создание неудачных структур БД.

Что делать с перегруженными атрибутами?

1NF 

Перегруженные атрибуты фактически являются вложенными таблицами с неявной структурой, похожи на строку формата CSV (с разделителями):

2023-10-04; №1234; Б

+7(495)123-45-67; +7(499) 987-65-43

1. Если значения перегруженных атрибутов не подлежат контролю/ обработке/ анализу, скорее являются комментариями, то нужно оставить их, но оговорить синтаксис списка.

2. Если перегруженные атрибуты подлежат контролю/ обработке/ анализу, то нужно:

- для составных атрибутов создать дополнительные атрибуты
- для многозначных атрибутов разнести значения по разным записям или выделить в отдельную таблицу или создать дополнительные атрибуты

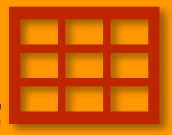
Идент.		Дата протокола	Номер протокола	Индекс решения
101		2023-10-04	№1234	Б
…		…	…	…

Идент.	Номер телефона
123	+7(495)123-45-67
123	+7(499) 987-65-43

Идент.		Номер телефона 1	Номер телефона 2
123		+7(495)123-45-67	+7(499) 987-65-43

Пример наложенных атрибутов

1NF



Тип клиента	Полное наименование юридического лица / Фамилия, имя, отчество физического лица, индивидуального предпринимателя	ИНН (TIN) юридического лица / индивидуального предпринимателя // Код документа, удостоверяющего физическое лицо	ОГРН юридического лица / ОГРН ИП / Серия, номер документа, удостоверяющего физическое лицо
ЮЛ	Полное наименование юридического лица - резидента	ИНН юридического лица - резидента	ОГРН юридического лица - резидента согласно ЕГРЮЛ
ЮЛ	Полное наименование юридического лица - нерезидента	Идентиф. номер налогоплательщика - иностранной организации (или его аналог) или Регистрационный номер в стране регистрации	
ИП	Фамилия, имя, отчество индивидуального предпринимателя	ИНН индивидуального предпринимателя - резидента	ОГРН индивидуального предпринимателя - резидента согласно ЕГРИП
ФЛ	Фамилия, имя, отчество физического лица	Код документа, удостоверяющего физическое лицо -	Серия, номер документа, удостоверяющего физическое лицо

Недостатки:

1) резидентность субъекта явно не определена, нет указания страны регистрации
2) значение TIN или регистрационного номера нерезидента может совпасть с ИНН юрлица, физлица или ИП
3) неявный признак нерезидента юрлица – отсутствие значения атрибута «ОГРН…»

Что делать с наложенными атрибутами?

1NF

1. Если требуется в примере наличие хотя бы одного идентификатора субъекта, то можно оставить наложенный атрибут со значением идентификатора и признаками (атрибутами), которые позволяют определить вид частного идентификатора, тип субъекта и его резидентность.

Страна регистрации субъекта	Тип субъекта	Вид частного идентификатора субъекта	Значение частного идентификатора
РФ	ЮЛ	ОГРН	1234567890123
РФ	ИП	ОГРНИП	983456789012345
РФ	ФЛ	ИНН	987654321012
США	ЮЛ	TIN	5467341289

При таком подходе остаётся возможность контролировать, обрабатывать / анализировать значения идентификаторов. Можно также сгенерировать уникальный идентификатор.

2. Если в примере важны и другие или все атрибуты субъекта, то следует их разместить в отдельные таблицы с идентификатором субъекта. Важно типизировать такие представления в рамках всей предметной области.

2-ая нормальная форма

Таблица (отношение) находится в 2NF, когда

- *таблица находится в 1NF*
- *таблица имеет ключ – уникальный идентификатор*
- *все неключевые колонки (атрибуты) зависят от полного ключа (если он составной)*

Если таблица имеет простой первичный ключ (естественный или суррогатный), то она находится в 2NF.

Если таблица имеет составной первичный ключ, но не имеет неключевых атрибутов, то она находится в 2NF.

Если таблица имеет составной первичный ключ и неключевые атрибуты от всех ключей, то она находится в 2NF.

В центре внимания 2NF находится первичный ключ таблицы.

3-ья нормальная форма

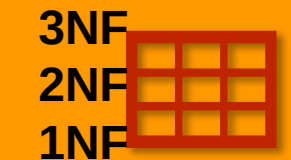


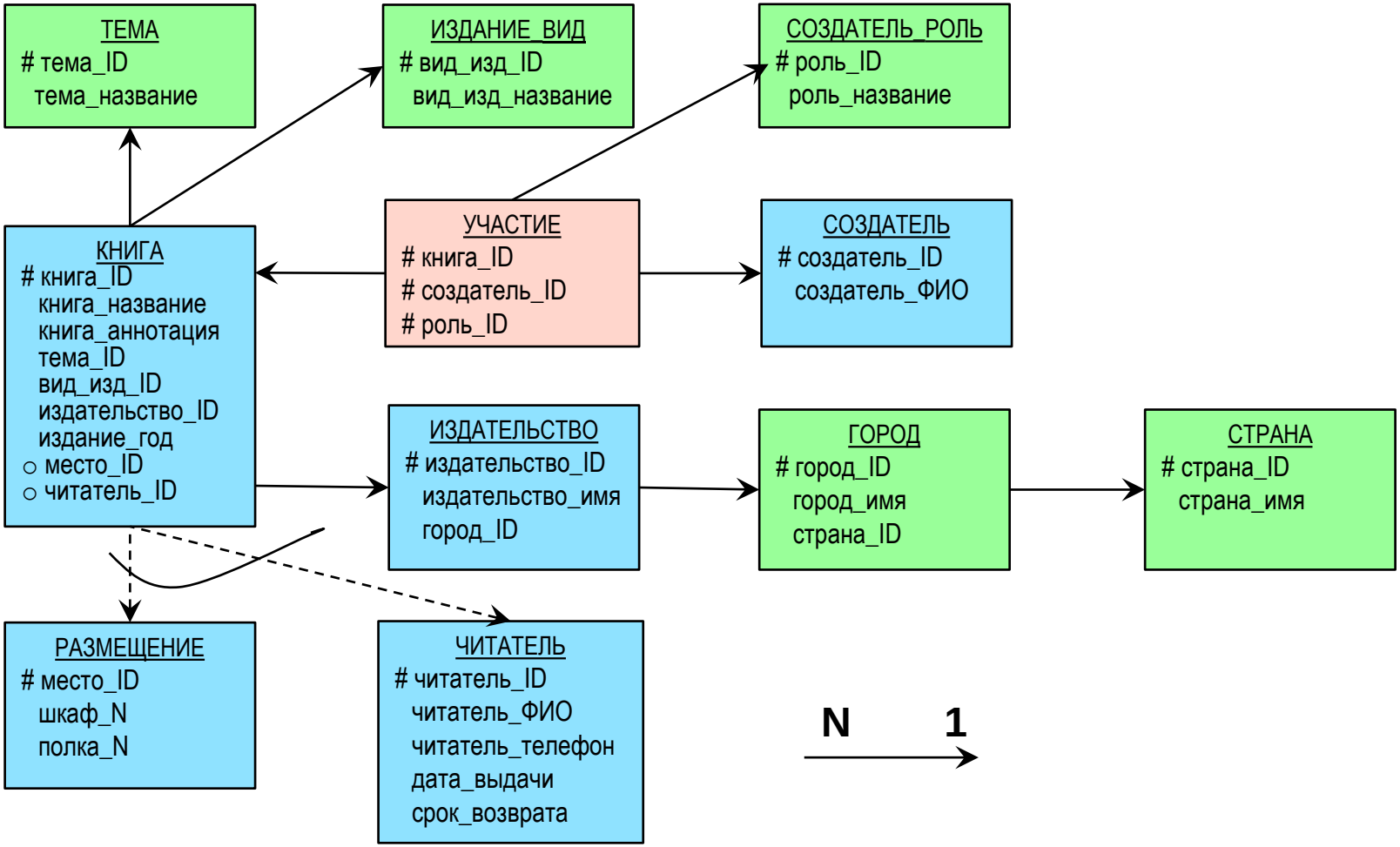
Таблица (отношение) находится в 3NF, когда

- *таблица находится в 2NF*
- *отсутствует транзитивная зависимость, когда неключевые атрибуты зависят от значений других неключевых атрибутов.*

Основной механизм – декомпозиция таблиц.
Способствует выделению справочников-классификаторов и реестров, в которых каждый класс или субъект (объект) содержится однократно, что сокращает избыточность данных и устраняет аномалии изменений.

Всё внимание в 3NF привлечено к неключевым атрибутам.

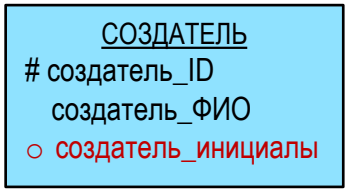
Пример. Логическая модель данных «Домашняя библиотека»



Пример. Проверка соответствия ЛМД требованиям 1NF

В модели присутствует составной семантически неоднозначный атрибут создатель_ФИО, в котором могут вместо имени, отчества содержаться инициалы.

На ER-диаграмме добавить создатель_инициалы с необязательным значением.



В таблице описания ЛМД

исправить создатель_ФИО: *Фамилия, имя и отчество создателя*
добавить *создатель_инициалы: Инициалы создателя (нет)*

Других несоответствий требованиям 1NF не было обнаружено.

Пример. Проверка соответствия ЛМД требованиям 2NF

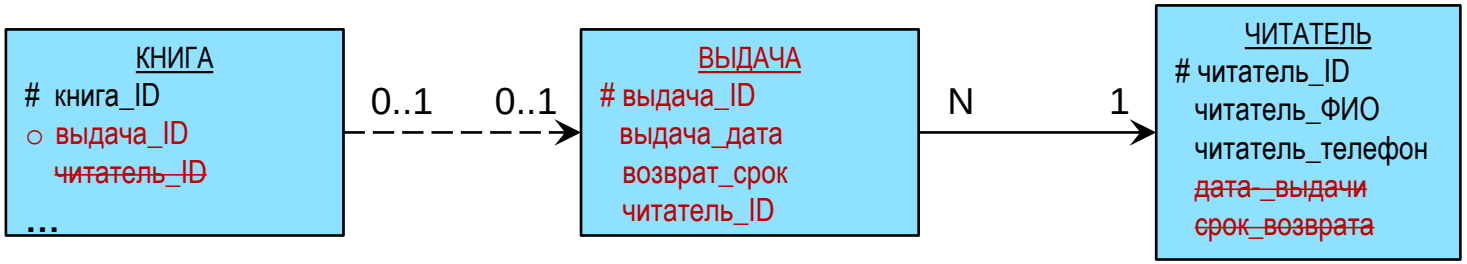
Сущности (таблицы) КНИГА, ТЕМА, ИЗДАНИЕ_ВИД, СОЗДАТЕЛЬ, СОЗДАТЕЛЬ_РОЛЬ, ИЗДАТЕЛЬСТВО, ГОРОД, СТРАНА, РАЗМЕЩЕНИЕ, ЧИТАТЕЛЬ имеют простые первичные ключи (суррогатные) и, значит, находятся в 2NF.

Сущность УЧАСТИЕ имеет составной первичный ключ из ключевых атрибутов книга_ID, создатель_ID, роль_ID, но не имеет неключевых атрибутов, поэтому находится в 2NF.

Пример. Проверка соответствия ЛМД требованиям 3NF

В модели в сущности ЧИТАТЕЛЬ присутствуют неключевые атрибуты дата_выдачи, срок_возврата, которые связаны между собой, но слабо связаны с читателем. Реквизиты читателя повторяются с каждой взятой книгой – избыточность и аномалии изменений.

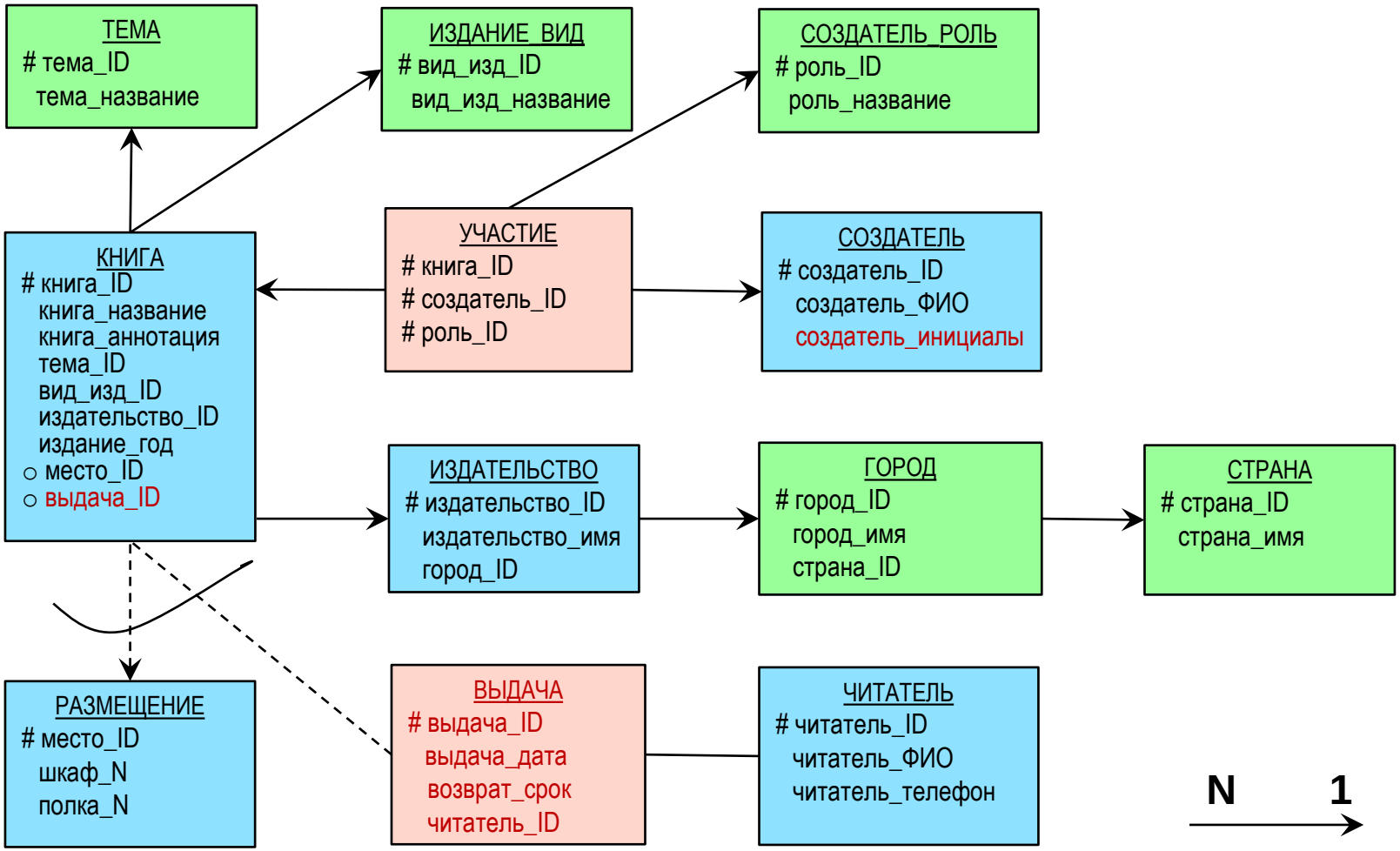
Следует декомпозировать сущность ЧИТАТЕЛЬ на две сущности ВЫДАЧА и ЧИТАТЕЛЬ на ER-диаграмме:



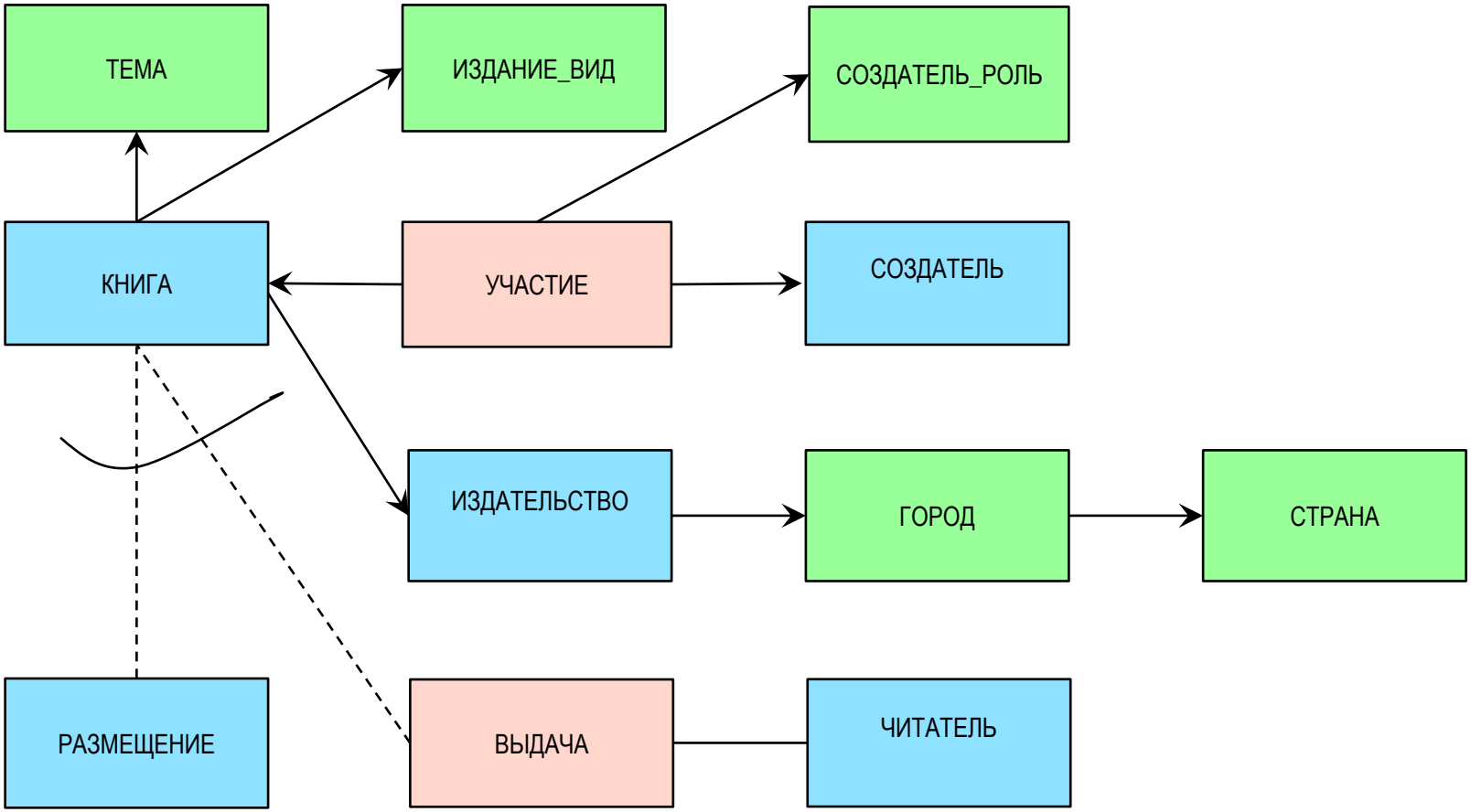
Изменены имена атрибутов дата_выдачи, срок_возврата на имена выдача_дата, возврат_срок согласно выбранным правилам именования. Соответственно нужно изменить таблицу описания КМД и ЛМД.

Других несоответствий 2NF не было обнаружено.

Пример. Обновлённая ER-диаграмма ЛМД «Домашняя библиотека»



Пример. Обновлённая ER-диаграмма КМД «Домашняя библиотека»



Терпения и удачи всем, кто связан с моделированием данных

Спасибо за внимание!

Валерий Иванович Артемьев

МГТУ имени Н.Э. Баумана, кафедра ИУ-5

Банк России
Департамент данных, проектов и процессов

Тел.: +7(495) 753-96-25
e-mail: viart@bmstu.ru