



***Правила построения
таблиц БД на основе
диаграмм «сущность -
связь»***

Перечень общих правил генерации таблиц из ER-диаграмм можно получить, опираясь на класс принадлежности и степень отношения как на определяющие факторы. С целью упрощения вывода этих правил будем использовать пример

АВТОР ПИШЕТ КНИГУ

Связь **ПИШЕТ** между сущностями **АВТОР** и **КНИГА** называется **бинарной**, поскольку связывает две сущности. Имеются связи более высокого порядка, существующие между тремя и более сущностями. Но поскольку бинарные связи встречаются наиболее часто, то мы ограничимся рассмотрением только таких связей.

Общий подход к проектированию баз данных с использованием ER-метода состоит прежде всего в построении ER-диаграммы, включающей в себя все важные сущности и связи, той предметной области, для которой проектируется БД.

***Предварительные таблицы для бинарных связей
степени «один к одному»***

Случай 1.




Связь 1 к 1, класс принадлежности обязателен для обеих сущностей, участвующих в связи.

Это гарантирует однократное появление каждого **на** и каждого **нк** в любом экземпляре отношения. Это значит, что таблица никогда не будет содержать ни пустой информации, ни повторяющихся групп избыточных данных. В данном случае в качестве первичного ключа таблицы был выбран ключ сущности **АВТОР**, но совершенно очевидно, что в этом качестве также может быть использован ключ сущности **КНИГА**.

Таблица АВТОР

на	фам автора	тел автора	нк	название кн
A1	Мешков Андрей	333-33-33	K2	Visual C++ и MFC
A2	Тихомиров Юрий	777-77-77	K5	Open GL: Программирование трехмерной графики
A3	Шведов Дмитрий	444-44-44	K6	Программирование в Delphi3



Таким образом, можно сформулировать следующее правило генерации таблиц с учетом существующих связей

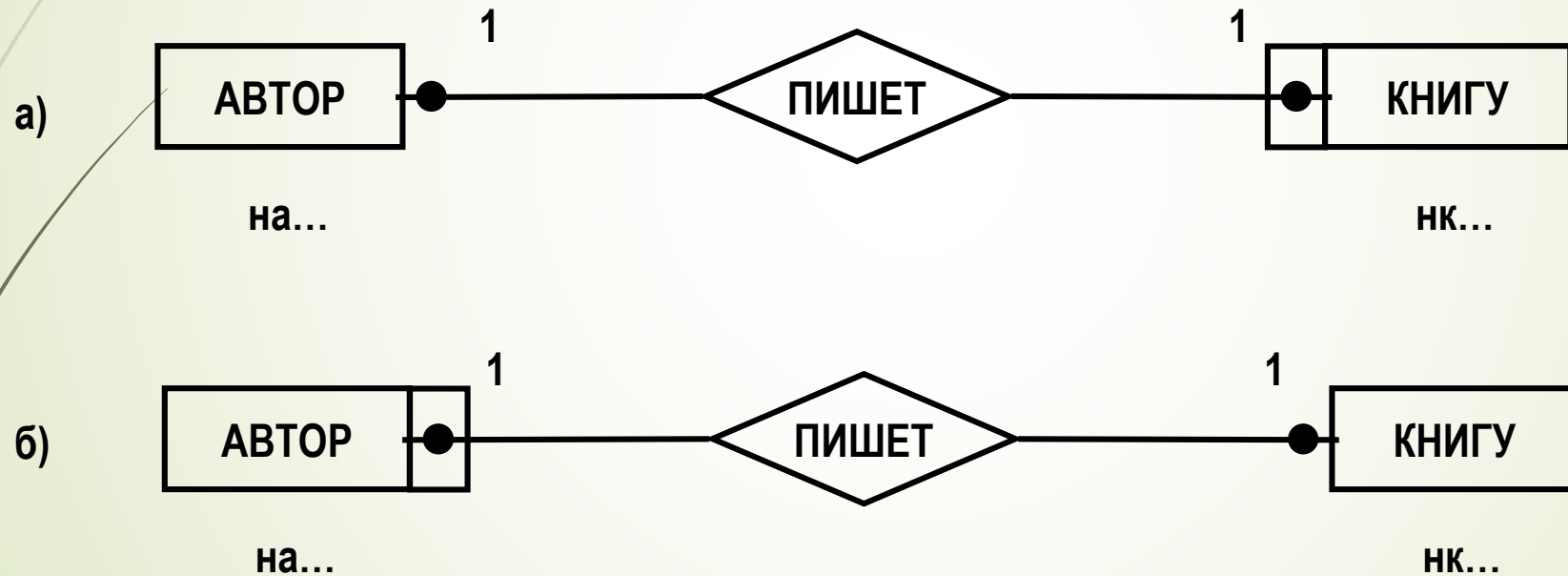
Правило 1

Если степень бинарной связи равна 1:1, и класс принадлежности обеих сущностей является обязательным, то требуется только одна таблица. Первичным ключом этой таблицы может быть ключ любой из двух сущностей.

Случай 2.

Теперь рассмотрим такой случай, когда степень связи 1:1 и класс принадлежности одной сущности является обязательным, а другой – необязательным.

Этим условиям удовлетворяют ER-диаграммы



Приведем пример таблицы АВТОР для случая а)

Таблица АВТОР

на	фам автора	тел автора	нк	название кн
A1	Мешков Андрей	333-33-33	K2	Visual C++ и MFC
A2	Тихомиров Юрий	777-77-77	K5	Open GL: Программирование трехмерной графики
A3	Шведов Дмитрий	444-44-44	K6	Программирование в Delphi3
-	-	-	K1	Табличный процессор Excel

Пробелы появляются в тех строках, содержащих информацию о книгах, которые не пишет ни один из авторов.

Чтобы исключить пробелы (пустые ячейки) требуется две таблицы вместо одной. Кроме того, чтобы не потерять информацию о том, какой автор написал какую книгу, то таблицу АВТОР необходимо дополнить атрибутом, являющимся ключом сущности, класс принадлежности которой является не обязательным.

Таким образом получаем две таблицы АВТОР и КНИ ГА

Таблица АВТОР

на	фам автора	тел автора	нк
A1	Мешков Андрей	333-33-33	K2
A2	Тихомиров Юрий	777-77-77	K5
Ф3	Шведов Дмитрий	444-44-44	K6

Таблица КНИГА

нк	название кн
K2	Visual C++ и MFC
K5	Open GL: Программирование трехмерной графики
K6	Программирование в Delphi3
K1	Табличный процессор Excel

Итак, теперь можем сформулировать следующее правило для данного случая.



Правило 2

Если степень бинарной связи равна 1:1, и класс принадлежности одной сущности является обязательным, а другой – необязательным, то необходимо построение двух таблиц. Под каждую сущность необходимо выделить одну таблицу. При этом первичный ключ сущности должен служить первичным ключом для соответствующей таблицы. Кроме того, ключ сущности, для которой класс принадлежности является необязательным, добавляется в качестве атрибута в таблицу, выделенную для сущности с обязательным классом принадлежности.

Случай 3.

Далее рассмотрим последний случай, соответствующий степени связи 1:1 и класс принадлежности ни одной из сущностей не является обязательным. Данный случай описывается ER-диаграммой



а) использование одной таблицы

Таблица АВТОР

на	фам автора	тел автора	нк	название кн
A1	Мешков Андрей	333-33-33	K2	Visual C++ и MFC
A2	Тихомиров Юрий	777-77-77	K5	Open GL: Программирование трехмерной графики
A3	Шведов Дмитрий	444-44-44	K6	Программирование в Delphi3
A4	Каминский Юрий	111-11-11	-	-
-	-	-	K1	Табличный процессор Excel

б) использование двух таблиц

Таблица АВТОР

на	фам автора	тел автора	нк
A1	Мешков Андрей	333-33-33	K2
A2	Тихомиров Юрий	777-77-77	K5
A3	Шведов Дмитрий	444-44-44	K6
A4	Каминский Юрий	111-11-11	-

Таблица КНИГА

нк	название кн	A1
K2	Visual C++ и MFC	A2
K5	Open GL: Программирование трехмерной графики	A3
K6	Программирование в Delphi3	A4
K1	Табличный процессор Excel	-

Опять имеются пустые ячейки. Чтобы избавиться от пустых ячеек, нужно две таблицы для каждой сущности и одну таблицу для связи.

Таблица АВТОР

на	фам автора	тел автора
A1	Мешков Андрей	333-33-33
A2	Тихомиров Юрий	777-77-77
A3	Шведов Дмитрий	444-44-44
A4	Каминский Юрий	111-11-11

Таблица КНИГА

нк	название кн
K2	Visual C++ и MFC
K5	Open GL: Программирование трехмерной графики
K6	Программирование в Delphi3
K1	Табличный процессор Excel

Таблица ПИШЕТ

на	нк
A1	K2
A2	K5
A3	K6

Отсюда вытекает следующее правило формирования таблиц.



Правило 3

Если степень бинарной связи равна 1:1, и класс принадлежности ни одной из сущностей не является обязательным, то необходимо использовать три таблицы: по одной для каждой сущности, ключи которых служат в качестве первичных ключей в соответствующих таблицах, и одной для связи. Среди своих атрибутов таблица, выделяемая для связи, будет иметь по одному ключу сущности от каждой сущности.

Предварительные таблицы для бинарных связей степени «один ко многим»

Так же как и для связи степени 1:1 фактором влияющим на выбор правила преобразования исходной таблицы является класс принадлежности n-связной сущности, а класс принадлежности 1-связной сущности в обоих случаях не влияет на конечный результат.

Рассмотрим для примера такую таблицу КНИГА

Таблица КНИГА

нк	название кн	на	фам автора	тел автора
K1	Visual C++ и MFC, т.1	A1	Мешков Андрей	333-33-33
K2	Open GL: Программирование трехмерной графики	A2	Тихомиров Юрий	777-77-77
K3	Microsoft Office	A3	Каминский Юрий	111-11-11
K4	Visual C++ и MFC, т.2	A1	Мешков Андрей	333-33-33
K5	Visual C++ 6	A2	Тихомиров Юрий	777-77-77
K6	Программирование в Delphi 3	A4	Шведов Дмитрий	444-44-44
-	-	A5	Никитин Павел	222-22-22

Случай 4.

Этот случай характерен тем, что степень связи 1:n, и класс принадлежности сущности КНИГА является обязательным. Пробел в таблице соответствует тому, что один из авторов не пишет книгу, и появляется дублирование данных, если автор пишет более одной книги.

Данная таблица соответствует следующей диаграмме ER-типа



Решить проблему пустых ячеек и дублирования данных можно, разбив таблицу на две по такому правилу:

Правило 4

Если степень бинарной связи 1:n, и класс принадлежности n-связной сущности является обязательным, то достаточным является использование двух таблиц (по одной для каждой сущности), при условии, что ключ каждой сущности служит в качестве первичного ключа для соответствующей таблицы. Помимо этого, ключ 1-связной сущности должен быть добавлен как атрибут в таблицу, отводимую n-связной сущности.

Результатом применения этого правила будут две таблицы.

Таблица КНИГА

нк	название кн	на
K1	Visual C++ и MFC, т.1	A1
K2	Open GL: Программирование трехмерной графики	A2
K3	Microsoft Office	A3
K4	Visual C++ и MFC, т.2	A1
K5	Visual C++ 6	A2
K6	Программирование в Delphi 3	A4

Таблица АВТОР

на	фам автора	тел автора
A1	Мешков Андрей	333-33-33
A2	Тихомиров Юрий	777-77-77
A3	Каменски1Юрий	111-11-11
A4	Шведов Дмитрий	444-44-44
A5	Никитин Павел	222-22-22

Случай 5.

Теперь рассмотрим вариант связи степени 1:n с необязательным классом принадлежности обеих сущностей. Тут просматривается три проблемы: 1) пробелы возникают в полях обеих сущностей; 2) кроме того, когда автор пишет более одной книги, повторяются данные об авторе.

Таблица соответствующая данному случаю может иметь вид

Таблица КНИГА

нк	название кн	на	фам автора	тел автора
K1	Visual C++ и MFC, т.1	A1	Мешков Андрей	333-33-33
K2	Open GL: Программирование трехмерной графики	A2	Тихомиров Юрий	777-77-77
K3	Microsoft Office	A3	Каминский Юрий	111-11-11
K4	Visual C++ и MFC, т.2	A1	Мешков Андрей	333-33-33
K5	Visual C++ 6	-	-	-
K6	Программирование в Delphi 3	A4	Шведов Дмитрий	444-44-44
-	-	A5	Никитин Павел	222-22-22

Приведенная таблица КНИГА соответствует диаграмме ER-типа



Если ее разбить на две таблицы по правилу 4 исчезают проблемы кроме пробелов

Таблица АВТОР

на	фам автора	тел автора
A1	Мешков Андрей	333-33-33
A2	Тихомиров Юрий	777-77-77
A3	Каменски1Юрий	111-11-11
A4	Шведов Дмитрий	444-44-44
A5	Никитин Павел	222-22-22

Таблица КНИГА

нк	название кн	на
K1	Visual C++ и MFC, т.1	A1
K2	Open GL: Программирование трехмерной графики	A2
K3	Microsoft Office	A3
K4	Visual C++ и MFC, т.2	A1
K5	Visual C++ 6	-
K6	Программирование в Delphi 3	A4

Все проблемы удается решить, используя при таблицы:

Таблица КНИГА

нк	название кн
K1	Visual C++ и MFC, т.1
K2	Open GL: Программирование трехмерной графики
K3	Microsoft Office
K4	Visual C++ и MFC, т.2
K5	Visual C++ 6
K6	Программирование в Delphi 3

Таблица АВТОР

на	фам автора	тел автора
A1	Мешков Андрей	333-33-33
A2	Тихомиров Юрий	777-77-77
A3	Каменски1Юрий	111-11-11
A4	Шведов Дмитрий	444-44-44
A5	Никитин Павел	222-22-22

Таблица ПИШЕТ

на	нк
A1	K1
A2	K2
A3	K3
A1	K4
A4	K6



Набор этих таблиц получается в результате применения следующего правила.

Правило 5.

Если степень бинарной связи равна 1:n и класс принадлежности n-связной сущности является необязательным, то необходимо формирование трех таблиц- по одной для каждой сущности, причем ключ каждой сущности служит первичным ключом соответствующей таблицы, и одной таблицы для связи. Связь должна иметь среди своих атрибутов ключ сущности для каждой сущности.

В варианте, который описывается диаграммой ER-типа



проблем с избыточным дублированием не возникает.

Предварительные таблицы для бинарных связей степени «многие ко многим»

Для степени связи $m:n$ потребуется для хранения данных три таблицы вне зависимости от класса принадлежности обеих сущностей. При использовании одной или двух таблиц неизбежно возникновение пробелов и/или повторяющихся данных. Поэтому для всех вариантов связи степени $m:n$ для сведения к трем таблицам предлагается правило.

Правило 6

Если степень бинарной связи $m:n$, то для хранения данных необходимы три таблицы: по одной для каждой сущности, причем ключ каждой сущности используется в качестве первичного ключа соответствующей таблицы, и одной таблицы для связи. Последняя таблица должна иметь в числе своих атрибутов ключ сущности для каждой сущности.

Таблица КНИГА

нк	название кн	на	фам автора	тел автора
K1	Visual C++ и MFC, т.1	A3	Тихомиров Юрий	777-77-77
K2	Visual C++ и MFC, т.2	A3	Тихомиров Юрий	777-77-77
K3	Microsoft Office	A4	Пашков Сергей	333-33-33
K4	Программирование на C++	-	-	-
K5	Библиотека Qt 4	A4	Пашков Сергей	333-33-33
K6	Работа в Linux	A4	Пашков Сергей	333-33-33
K2	Visual C++ и MFC, т.2	A1	Митрохин Петр	222-22-22
-	-	A2	Федоров Павел	111-11-11
K5	Библиотека Qt 4	A6	Артемьев Евгений	444-44-44
-	-	A5	Петров Артем	555-55-55

Применяя правило 6 приходим к набору таблиц

Таблица КНИГА

НК	название кн
К1	Visual C++ и MFC, т.1
К2	Visual C++ и MFC, т.1
К3	Microsoft Office
К4	Программирование на C++
К5	Библиотека Qt 4
К6	Работа в Linux

Таблица АВТОР

на	фам автора	тел автора
А1	Митрохин Петр	222-22-22
А2	Федоров Павел	111-11-11
А3	Тихомиров Юрий	777-77-77
А4	Пашков Сергей	333-33-33
А5	Петров Артем	555-55-55
А6	Артемьев Евгений	444-44-44

Таблица ПИШЕТ

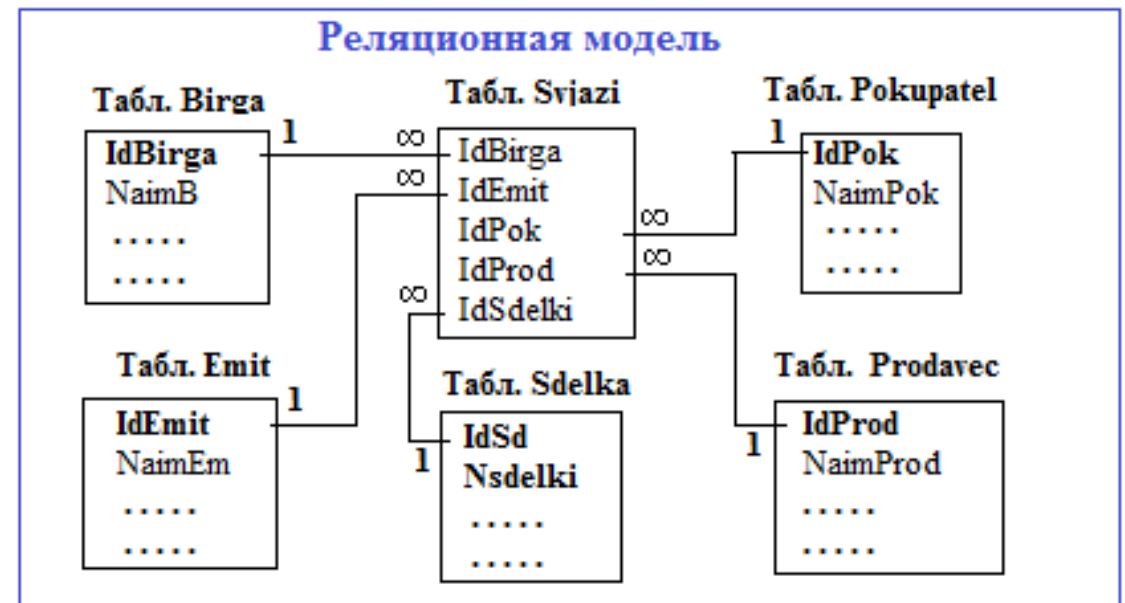
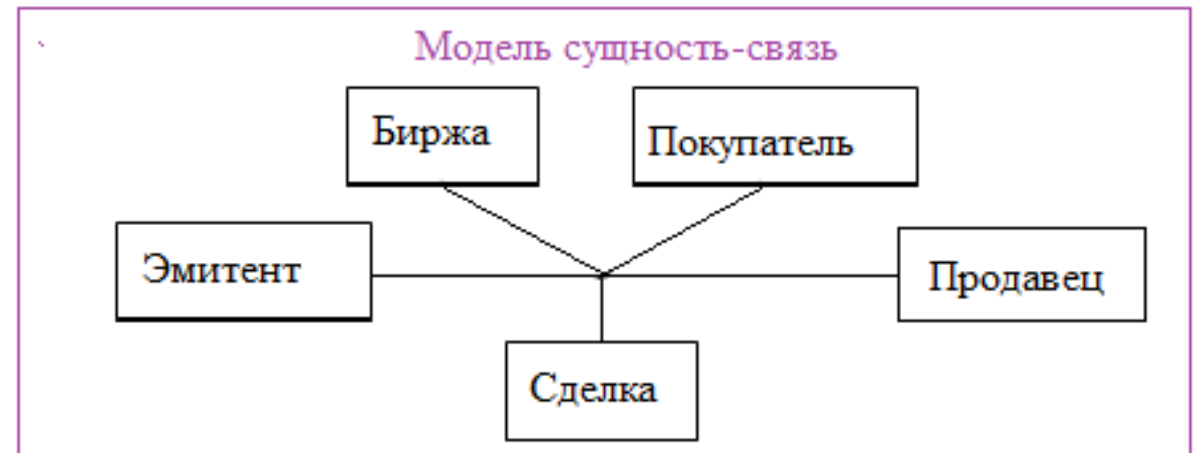
на	НК
А1	К2
А3	К1
А3	К2
А4	К3
А4	К5
А4	К6

Правило №7

При наличии многосторонней связи необходимо создать свою таблицу для каждой сущности, первичным ключом которой будет ключ этой сущности.

Кроме того, необходимо создать еще одну таблицу связи, которая будет содержать первичные ключи остальных таблиц в качестве своих атрибутов.

То есть для *n-сторонней связи* необходимо создать *n+1 таблиц*.



Итак...

Один-к-одному



Используется редко



Используется редко



Используется крайне редко
и почти всегда ошибочно



ОДИН-КО-МНОГИМ



Многие к одному (обязательная)



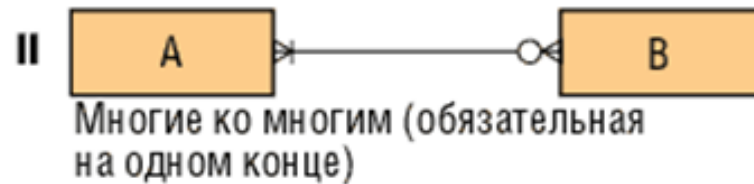
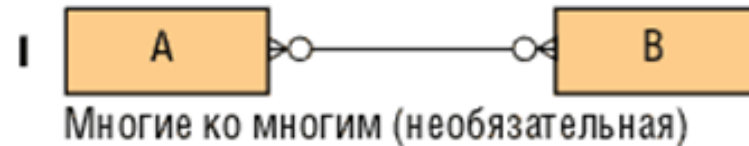
Многие к одному (обязательная
на одном конце)



Многие к одному (необязательная)

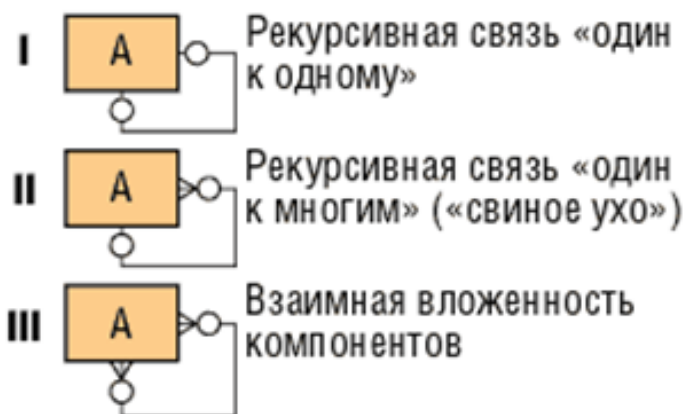
- ▶ I - достаточно сильная конструкция, предполагающая, что вхождение сущности B не может быть создано без одновременного создания по меньшей мере одного связанного с ним вхождения сущности A. **Чаще всего это неверная связь.**
- ▶ II - это наиболее часто встречающаяся форма связи. Она предполагает, что каждое и любое вхождение сущности A может существовать только в контексте одного (и только одного) вхождения сущности B. В свою очередь, вхождения B могут существовать как в связи с вхождениями A, так и без нее.
- ▶ III - Как A, так и B могут существовать без связи между ними.

Многие-ко-многим



- ▶ I - такая конструкция часто имеет место в начале этапа анализа и означает связь - либо понятую не до конца и требующую дополнительного разрешения, либо отражающую простое коллективное отношение - двунаправленный список.
- ▶ II - применяется редко. Такие связи всегда подлежат дальнейшей детализации.

Рекурсивные связи



- ▶ I - редко, но имеет место. Отражает связи альтернативного типа.
- ▶ II - достаточно часто применяется для описания иерархий с любым числом уровней.
- ▶ III - имеет место на ранних этапах. Часто отражает структуру "перечня материалов" (взаимная вложенность компонентов). Пример: каждый КОМПОНЕНТ может состоять из одного и более (других) КОМПОНЕНТОВ и каждый КОМПОНЕНТ может использоваться в одном и более (других) КОМПОНЕНТОВ.

Пример: логическая модель данных предприятия по сборке компьютеров и ноутбуков

