

Краткое описание ER–метода проектирования реляционных баз данных (метод, использующий схему «сущность-связь» -«Entity-Relationships»)

Одним из наиболее понятных и практически используемых методов проектирования реляционных баз данных является метод, в основу которого положена модель «сущность-связь». В качестве инструмента используются различные варианты диаграмм.

Ключевыми элементами модели «сущность-связь» являются сущности, их свойства (атрибуты) и связи между объектами.

Сущность определяется как некоторый объект рассматриваемой предметной области, информация о котором должна быть отражена в базе данных.

Этот объект должен иметь **экземпляры** - конкретные представители данной сущности, отличающиеся друг от друга и допускающие однозначную идентификацию.

Взаимоотношения сущностей выражаются **связями** между ними.

Атрибут – это свойство сущности.

Сущность – это, как правило, существительное; связь чаще всего выражается глаголом.

Например, проектируется база данных издательства, предназначенная для хранения информации о книгах и авторах, которые их написали. Тогда два главных объекта (две сущности, информация о которых должна быть отражена в базе данных) – это **книга** и **автор**. Эти сущности содержательно соединены с помощью связи **пишет**. Атрибуты сущности **книга** – это название, количество страниц, тираж, дата выхода сигнального экземпляра, цена и т.д. Атрибуты сущности **автор** – это фамилия, адрес, телефон, №счета и т.д.

Сущности изображаются в виде прямоугольника, атрибуты вписываются внутрь прямоугольника, изображающего сущность:

АВТОР
Фамилия И.О. Адрес Телефон №счета

КНИГА
Название Количество страниц Тираж Дата выхода

Данные определения не являются формальными, однако они приемлемы для использования при проектировании базы данных. Особенность ER-метода заключается в том, что разные проектировщики могут рассматривать одну и ту же предметную область с различных точек зрения, получая в итоге разные наборы сущностей и связей. Определение лучшего из нескольких возможных наборов является вопросом личного предпочтения.

Атрибут или набор атрибутов, используемый для идентификации экземпляра сущности, называется **ключом сущности**. Таким образом, ключ сущности должен быть уникальным для каждого экземпляра этой сущности. Ключ каждой сущности не должен быть избыточным, т.е. удаление любого атрибута из этого набора будет нарушать его уникальность. Ключевые атри-

буты каким-либо образом выделяются на диаграмме (например, подчеркиванием или более жирным шрифтом).

АВТОР
Фамилия И.О. Адрес Телефон №счета

КНИГА
Название Количество страниц Тираж Дата выхода

В нашем примере в качестве ключевого атрибута сущности АВТОР было решено взять фамилию, а в качестве ключевого атрибута сущности КНИГА взять её название. Первое решение заведомо не является бесспорным: возможно появление авторов-однофамильцев (тогда атрибут **Фамилия И.О.** теряет уникальность и не может быть использован в качестве ключа). В принципе, допустимо появление книг с одинаковыми названиями.

Если есть хотя бы минимальное подозрение, что атрибут, выбираемый в качестве ключевого, может потерять свою уникальность, нужно отказаться от его использования в качестве ключа и попытаться подобрать на эту роль другой атрибут.

Если окажется, что ни один содержательный атрибут не может быть использован как ключевой, то существует (по меньшей мере) два способа решения этой проблемы:

- подобрать набор атрибутов, значения которых будут уникальными для каждого экземпляра сущности;
- ввести еще один атрибут, который не будет отражать какое-либо свойство сущности, но будет пригоден в качестве ключевого. Обычно таким атрибутом становится номер экземпляра анализируемой сущности.

Для нашего примера выберем второй способ:

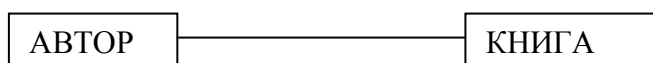
АВТОР
Номер автора Фамилия И.О. Адрес Телефон №счета

КНИГА
Номер книги Название Количество страниц Тираж Дата выхода

Замечание: если при выборе атрибутов какой-либо сущности возникает желание определить атрибут, название которого совпадает с названием другой сущности нашей предметной области, то от такого атрибута надо отказаться. Наличие такого атрибута – это попытка зафиксировать связь между этими сущностями, но анализ связи еще не проведен и, возможно, она должна быть реализована иначе.

Например, при выборе атрибутов сущности Книга очень хочется определить такой атрибут как Автор этой книги. Но у нас есть сущность Автор, поэтому связь сущности Книга с сущностью Автор будет реализована в результате анализа связи между этими сущностями.

Связь между двумя сущностями может быть представлена графически в виде ER-диаграммы:



Важной характеристикой связи является *степень связи*. Возможны следующие степени связи: *один-к-одному*, *один-ко-многим* и *многие-ко-многим*.

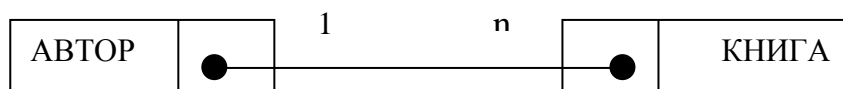
Кроме того, надо выявить *класс принадлежности сущности*, который характеризует обязательность включения каждого экземпляра сущности в связь.

Связь ОДИН-КО -МНОГИМ:

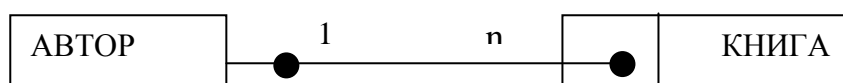
Связь один-ко-многим подразумевает, что один экземпляр сущности, расположенной в левой части диаграммы, связан с несколькими экземплярами сущности, расположенной в правой части диаграммы.

Если все экземпляры сущности должны участвовать в связи, то участие является обязательным, и изображается на ER-диаграмме кружком, помещенным в блок, изображающий сущность. Если не все экземпляры сущности должны участвовать в связи, то участие является необязательным, и кружок на ER-диаграмме располагается вне блока сущности.

Следующая диаграмма отражает связь один-ко-многим сущностей АВТОР – КНИГА, где экземпляры обеих сущностей вступают в обязательную связь:

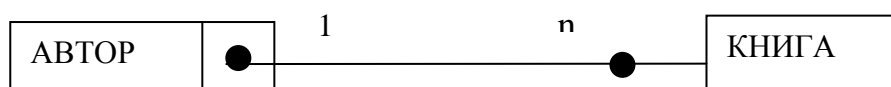


Таким образом, в этой диаграмме отражено правило: «каждую книгу пишет только один автор, но каждый автор может писать несколько книг»; обязательность включения экземпляров говорит о том, что в базе данных не будет информации об авторах, не пишущих ни одной книги, а также информации о книгах, которые пока никто не пишет.

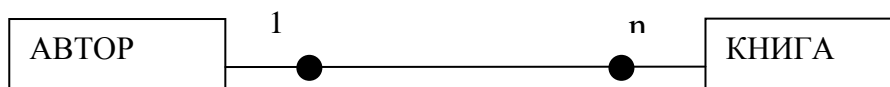


В этой диаграмме отражено правило: «каждую книгу пишет только один автор, каждый автор может писать несколько книг либо не писать их вовсе», т.е. в базе данных допускается наличие авторов, не пишущих ни одной книги, но нет информации о книгах, которые пока никто не пишет.

Для сущностей АВТОР - КНИГА возможны еще два типа связи один-ко-многим, отражающих два оставшихся варианта обязательности включения экземпляров:



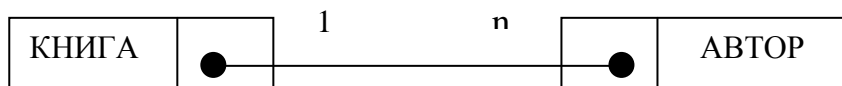
«Каждую книгу может писать не более чем один автор; каждый автор может писать несколько книг (но должен писать хотя бы одну)», т.е. в базе данных не будет информации об авторах, которые не пишут ни одной книги, но допускается хранить информацию о книгах, которые еще никто не пишет.



«Каждую книгу может писать не более чем один автор; каждый автор может писать несколько книг либо не писать их вовсе», т.е. в базе данных допускается наличие авторов, не пишущих сейчас ни одной книги, и наличие книг, которые еще никто не пишет.

Аналогично анализируются и фиксируются все варианты связи один-ко-многим сущностей КНИГА - АВТОР (с учетом обязательности / необязательности участия в связи всех экземпляров этих сущностей). Здесь также возможны четыре различных варианта.

Вот один из возможных вариантов:



В этой диаграмме отражено правило: «каждую книгу пишут несколько авторов, каждый автор должен писать только одну книгу»; обязательность включения экземпляров говорит о том, что в базе данных не будет информации об авторах, не пишущих ни одной книги, а также информации о книгах, которые пока никто не пишет.

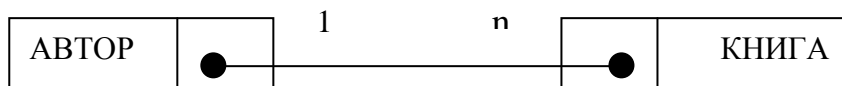
Правила генерации таблиц по ER-диаграмме при связи ОДИН-КО-МНОГИМ

Замечание: в этом случае определяющим фактором является класс принадлежности n-связной сущности; класс принадлежности 1-связной сущности на конечный результат не влияет.

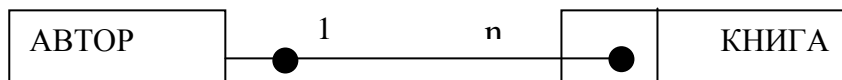
Правило 1: Если класс принадлежности n-связной сущности является обязательным, то достаточно использовать две таблицы (по одной для каждой сущности); ключ каждой сущности служит в качестве первичного ключа соответствующей таблицы. Кроме того, ключ 1-связной сущности должен быть добавлен как атрибут в таблицу, представляющую n-связную сущность.

Такой атрибут, предназначенный для фиксации связей между экземплярами двух сущностей, называется *вторичным ключом*.

Пусть в нашем примере между сущностями АВТОР и КНИГА выявлена такая связь:



либо такая:



Поскольку определяющим фактором является класс принадлежности n-связной сущности, а класс принадлежности 1-связной сущности на конечный результат не влияет, то в обоих случаях количество таблиц и связи между ними будут одинаковыми:

Автор

Номер автора	Фамилия И.О.	Адрес	Телефон	№счета
1	Орлов А.С.	Москва, Лесная 34-1-75	263-67-89	21436587
2	Станов О.Т.	Курск, Новая 23-56	23-45-12	65748392
3	Рыбаков И.И.	Казань, Рыбная 2-34	34-54-12	98765430
4	Туманов П.Р.	Москва, Стасовой 6-2-56	943-45-89	23894567
...

Книга

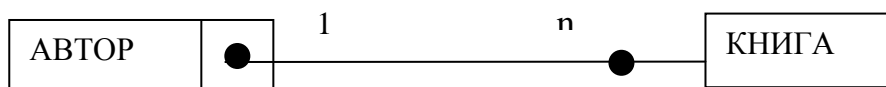
Номер книги	Название	Кол-во страниц	Тираж	Дата выхода	<i>Автор</i>
1	«Городок»	263	50000	15.03.2002	1
2	«Ранним утром»	450	30000	10.09.2003	2
3	«Рыжий»	341	45000	25.05.2002	1
...

Атрибут *Автор* в таблице Книга – вторичный ключ, предназначенный для фиксации связей между экземплярами сущностей Автор и Книга. Значениями этого атрибута являются значения первичного ключа из таблицы Автор.

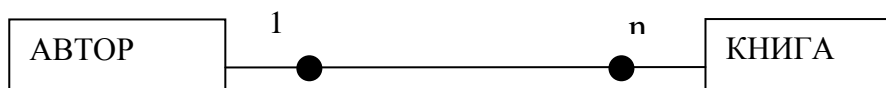
Замечание: здесь и далее вторичные ключи будем выделять жирным курсивом.

Правило 2: Если класс принадлежности n-связной сущности не является обязательным, то необходимо формирование трех таблиц – по одной для каждой сущности (ключ каждой сущности служит в качестве первичного ключа соответствующей таблицы), а также таблицы для связи. Таблица, создаваемая для связи, должна иметь по одному ключу от каждой сущности.

Таким образом, для ситуаций



и



необходимо три таблицы: по одной для каждой сущности и таблица-связка:

Автор

Номер автора	Фамилия И.О.	Адрес	Телефон	№счета
1	Орлов А.С.	Москва, Лесная 34-1-75	263-67-89	21436587
2	Станов О.Т.	Курск, Новая 23-56	23-45-12	65748392
3	Рыбаков И.И.	Казань, Рыбная 2-34	34-54-12	98765430
...

Книга

Номер книги	Название	Кол-во страниц	Тираж	Дата выхода
1	«Городок»	263	50000	15.03.2002
2	«Ранним утром»	450	30000	10.09.2003
3	«Рыжий»	341	45000	25.05.2002
...

План издательства

Номер книги	Номер автора	Верстка
1	3	да
2	1	нет
3	3	нет

Внимание!!! Для контроля того, что у каждой книги есть только один автор (т.е. в поле Номер книги не должно быть повторяющихся значений), при создании таблицы План издательства можно задать уникальность значений в поле Номер книги (в режиме конструктора).

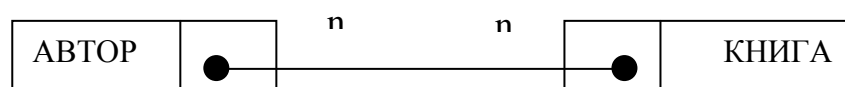
Есть и второй вариант достижения той же цели: в таблице План издательства поле Номер книги (вторичный ключ) можно описать как первичный. Таким образом, будет включен контроль уникальности значений в этом поле.

Связь МНОГИЕ-КО -МНОГИМ:

Связь многие-ко-многим подразумевает, что один экземпляр сущности, расположенной в левой части диаграммы, связан с несколькими экземплярами сущности, расположенной в правой части диаграммы, и один экземпляр сущности, расположенной в правой части диаграммы, связан с несколькими экземплярами сущности, расположенной в левой части диаграммы.

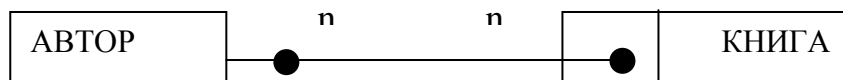
По-прежнему, если все экземпляры сущности должны участвовать в связи, то участие является обязательным, и изображается на ER-диаграмме кружком, помещенным в блок, изображающий сущность. Если не все экземпляры сущности должны участвовать в связи, то участие является необязательным, и кружок на ER-диаграмме располагается вне блока сущности.

Следующая диаграмма отражает связь многие-ко-многим сущностей АВТОР – КНИГА, где экземпляры обеих сущностей вступают в обязательную связь:



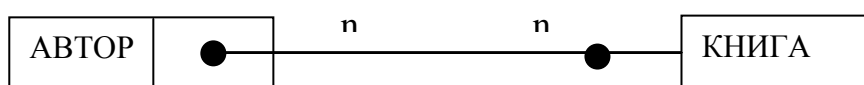
Таким образом, в этой диаграмме отражено правило: «каждую книгу пишут несколько авторов, каждый автор пишет несколько книг»; обязательность включения экземпляров говорит о

том, что в базе данных не будет информации об авторах, не пишущих ни одной книги, а также информации о книгах, которые пока никто не пишет.

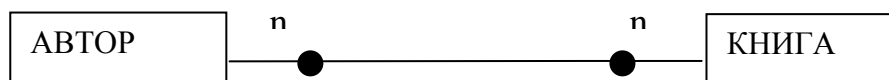


В этой диаграмме отражено правило: «каждую книгу пишут несколько авторов, каждый автор может писать несколько книг либо не писать их вовсе», т.е. в базе данных допускается наличие авторов, не пишущих ни одной книги, но нет информации о книгах, которые пока никто не пишет.

Для сущностей АВТОР - КНИГА возможны еще два типа связи многие-ко-многим, отражающих два оставшихся варианта обязательности включения экземпляров:



«Кажую книгу может писать несколько авторов; каждый автор может писать несколько книг (но должен писать хотя бы одну)», т.е. в базе данных не будет информации об авторах, которые не пишут ни одной книги, но допускается хранить информацию о книгах, которые еще никто не пишет.



«Кажую книгу может писать несколько авторов; каждый автор может писать несколько книг либо не писать их вовсе», т.е. в базе данных допускается наличие авторов, не пишущих сейчас ни одной книги, и наличие книг, которые еще никто не пишет.

Правила генерации таблиц по ER-диаграмме при связи МНОГИЕ-КО-МНОГИМ

Правило 1: В этом случае вне зависимости от класса принадлежности каждой сущности потребуется три таблицы: по одной для каждой сущности (ключ каждой сущности служит в качестве первичного ключа соответствующей таблицы), а также таблицы для связи. Таблица, создаваемая для связи, должна иметь по одному ключу от каждой сущности.

Автор

Номер автора	Фамилия И.О.	Адрес	Телефон	№счета
1	Орлов А.С.	Москва, Лесная 34-1-75	263-67-89	21436587
2	Станов О.Т.	Курск, Новая 23-56	23-45-12	65748392
3	Рыбаков И.И.	Казань, Рыбная 2-34	34-54-12	98765430
...

Книга

Номер книги	Название	Кол-во страниц	Тираж	Дата выхода
1	«Городок»	263	50000	15.03.2002
2	«Ранним утром»	450	30000	10.09.2003
3	«Рыжий»	341	45000	25.05.2002
...

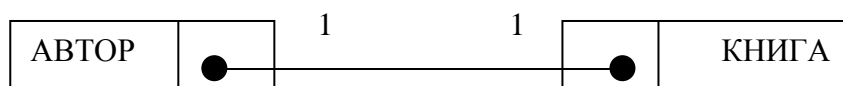
План издательства

Номер книги	Номер автора	Верстка
1	3	да
1	2	нет
2	3	нет
3	1	да
3	2	да

СВЯЗЬ ОДИН-К-ОДНОМУ:

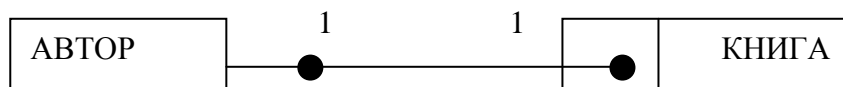
Связь один-к-одному подразумевает, что каждый экземпляр сущности, расположенной как в левой, так и в правой частях диаграммы, связывается не более чем с одним экземпляром сущности, расположенной в противоположной части диаграммы.

Как прежде, если все экземпляры сущности должны участвовать в связи, то участие называется обязательным, и изображается на ER-диаграмме кружком, помещенным в блок, изображающий сущность (при словесной формулировке такой связи обычно используется глагол «должен»):



В этой диаграмме отражено правило: «каждый автор должен писать книгу, причем только одну, и каждую книгу должен писать только один автор», т.е. в базе данных не будет информации об авторах, не пишущих ни одной книги, а также информации о книгах, которые пока никто не пишет.

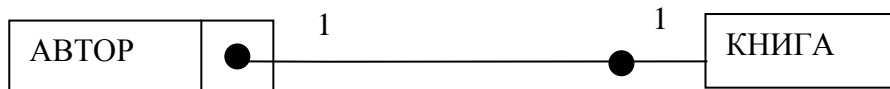
Если не все экземпляры сущности должны участвовать в связи, то участие называется необязательным, и кружок на ER-диаграмме располагается вне блока сущности (при словесной формулировке такой связи обычно используется глагол «может»):



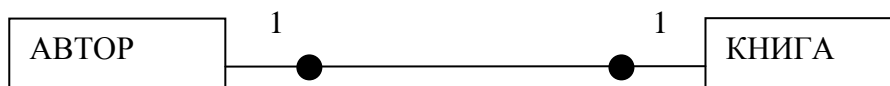
В этой диаграмме отражено правило: «каждый автор может писать не более одной книги, и каждую книгу должен писать только один автор», т.е. в базе данных допускается наличие ав-

торов, не пишущих ни одной книги, но нет информации о книгах, которые пока никто не пишет.

Для сущностей АВТОР - КНИГА возможны еще два типа связи один-к-одному, отражающих два оставшихся варианта обязательности включения экземпляров:



«Каждый автор должен писать книгу, причем только одну, и каждую книгу пишет не более чем один автор (один или никто)», т.е. в базе данных не будет храниться информация об авторах, которые в данный момент не пишут ни одной книги, но допускаются книги, которые еще никто не пишет.



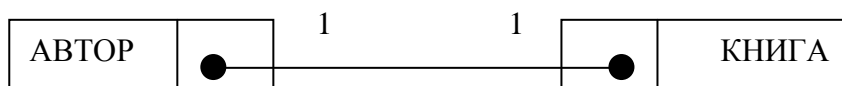
«Каждый автор пишет не более одной книги, и каждая книга пишется не более чем одним автором», т.е. в базе данных допускается наличие авторов, не пишущих сейчас ни одной книги, и наличие книг, которые еще никто не пишет.

Каждая диаграмма представляет некоторый набор правил, принятых в данной предметной области (например, в издательстве) по поводу того, какого рода информация должна храниться в базе данных.

Правила генерации таблиц по ER-диаграмме при связи ОДИН-К-ОДНОМУ

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

Пусть в нашем примере между сущностями АВТОР и КНИГА выявлена такая связь:



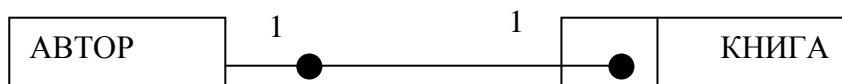
Тогда в базе данных будет только одна таблица, отображающая свойства этих сущностей:

План издательства

Номер	Название	Кол-во стр.	Тираж	Дата	Фамилия автора	Адрес	Телефон	№счета
1	«Городок»	263	50000	15.03.02	Орлов А.С.	Москва	345-67-89	25348217632
2	«Ранним утром»	450	30000	10.09.03	Станов О.Т.	Курск	34-23-78	56487392028
...

Правило 2: Если класс принадлежности одной сущности является обязательным, а другой – необязательным, то необходимо построение двух таблиц. Под каждую сущность необходимо выделить по таблице. При этом первичные ключи каждой из сущностей должны быть ключами соответствующих таблиц. Кроме того, ключ сущности, для которой класс принадлежности является необязательным, добавляется в качестве атрибута в таблицу, созданную для сущности с обязательным классом принадлежности.

Пусть в нашем примере между сущностями АВТОР и КНИГА выявлена такая связь:



В этом случае в базу данных будет включено две таблицы - по одной для каждой сущности. Поскольку класс принадлежности сущности АВТОР является необязательным, в таблицу КНИГА добавляется еще один атрибут (вторичный ключ) – Автор, значениями которого будут значения ключевого атрибута таблицы АВТОР.

Автор

Номер автора	Фамилия И.О.	Адрес	Телефон	№счета
1	Орлов А.С.	Москва, Лесная 34-1-75	263-67-89	21436587
2	Станов О.Т.	Курск, Новая 23-56	23-45-12	65748392
3	Рыбаков И.И.	Казань, Рыбная 2-34	34-54-12	98765430
4	Туманов П.Р.	Москва, Стасовой 6-2-56	943-45-89	23894567
...

Книга

Номер книги	Название	Кол-во страниц	Тираж	Дата выхода	Автор
1	«Городок»	263	50000	15.03.2002	1
2	«Ранним утром»	450	30000	10.09.2003	2
3	«Рыжий»	341	45000	25.05.2002	4
...

Внимание!!! Для того чтобы контролировать то, что каждый автор связан только с одной книгой (т.е. в таблице Книга в поле Автор не должно быть повторяющихся значений), при создании таблицы Книга необходимо задать уникальность значений в поле Автор (в режиме конструктора таблиц).

Правило 3: Если 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 соответствующих таблиц, и одну таблицу для связи. Таблица, создаваемая для связи, должна иметь по одному ключу от каждой сущности, т.е. содержать два вторичных ключа.

Автор

Номер автора	Фамилия И.О.	Адрес	Телефон	№счета
1	Орлов А.С.	Москва, Лесная 34-1-75	263-67-89	21436587
2	Станов О.Т.	Курск, Новая 23-56	23-45-12	65748392
3	Рыбаков И.И.	Казань, Рыбная 2-34	34-54-12	98765430
...

Книга

Номер книги	Название	Кол-во страниц	Тираж	Дата выхода
1	«Городок»	263	50000	15.03.2002
2	«Ранним утром»	450	30000	10.09.2003
3	«Рыжий»	341	45000	25.05.2002
...

План издательства

<i>Номер книги</i>	<i>Номер автора</i>	<i>Верстка</i>
1	3	да
2	1	нет
3	2	нет

Внимание!!! Для контроля того, что у каждой книги есть только один автор, и каждый автор связан только с одной книгой (т.е. в полях Номер книги и Номер автора не должно быть повторяющихся значений), при создании таблицы План издательства необходимо задать уникальность значений в полях Номер книги и Номер автора.

ПРИМЕР ПРИМЕНЕНИЯ МЕТОДА

Проектирование БД «Детали»

Постановка задачи:

Необходимо разработать проект базы данных и реализовать её в СУБД Access.

В этой базе заказчик хотел бы хранить информацию

- о типах деталей, с которыми будет работать заказчик (гайки, шайбы, болты, винты и т.п.)
- о характеристиках каждого поставляемого изделия (вес, металл, диаметр и т.п.)
- о поставщиках деталей
- о транспортных фирмах, осуществляющих доставку деталей

Некоторые условия, существенные для проектирования базы данных:

- каждый поставщик может поставлять несколько различных изделий
- одно и то же изделие может поставляться разными поставщиками
- возможна поставка одного и того же типа изделия, но с различными характеристиками
- каждая транспортная фирма может поставлять детали от разных поставщиков.

Замечание: если при проектировании базы данных разработчику придется принимать дополнительные соглашения об информации, которая должна храниться в базе данных, о связях между сущностями и свойствах этих связей, то эти соглашения должны быть описаны в отчете.

Этапы проектирования базы данных:

1. определение сущностей предметной области - источников данных, которые должны быть включены в базу данных
2. определение атрибутов каждой сущности, их первичных ключей
3. выявление связей между сущностями
4. определение степени каждой выделенной связи (один-к-одному, один-ко-многим, многие-ко-многим) и класса принадлежности каждой сущности, который характеризует обязательность включения каждого экземпляра сущности в связь
5. построение ER-диаграмм, отображающих выявленные связи
6. формирование таблиц базы данных по ER-диаграммам:
 - определение нужного количества таблиц
 - определение первичных и вторичных ключей таблиц

В ходе проектирования может возникнуть потребность возвращаться к уже пройденным этапам, пересматривать ранее принятые решения, т.е. проектирование БД – это итерационный процесс.

Определение сущностей, их атрибутов и первичных ключей

Список сущностей анализируемой предметной области: **Типы деталей, Детали, Поставщики, Транспортные фирмы.**

Сущности изображаются в виде прямоугольника, атрибуты вписываются внутрь прямоугольника, изображающего сущность:

Типы деталей
Наименование Изображение Описание

Детали
Название Вес Диаметр Металл Цвет

Транспортные фирмы
Название Страна Город Адрес Факс Новости Тип транспорта Менеджер

Поставщики
ФИО Страна Город Адрес Телефон Надежность

Атрибут или набор атрибутов, используемый для идентификации экземпляра сущности, называется *ключом сущности*. Ключевые атрибуты (*первичные ключи*) каким-либо образом выделяются на диаграмме (например, подчеркиванием или более жирным шрифтом).

Типы деталей
Наименование Изображение Описание

Поставщики
Код поставщика ФИО Страна Город Адрес Телефон Надежность

Транспортные фирмы
Код фирмы Название Страна Город Адрес Факс Новости Тип транспорта Менеджер

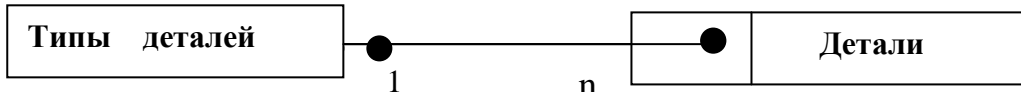
Детали
Код детали Название Вес Диаметр Металл Цвет

Для сущности Типы деталей в качестве первичного ключа можно взять атрибут Наименование, т.к. его значение уникально для каждого экземпляра этой сущности, следовательно, этот атрибут можно использовать для идентификации экземпляра сущности.

Для сущностей Детали, Поставщики и Транспортные фирмы в качестве первичного ключа придется вводить дополнительный атрибут – номер экземпляра сущности, поскольку ни один из содержательных атрибутов не является уникальным.

Определение связей между сущностями, степени этих связей и классов принадлежности, их фиксация с помощью ER-диаграмм, формирование таблиц базы данных по ER-диаграммам

Сущности Типы деталей – Детали



В этой диаграмме отражена связь: «каждая деталь – это деталь одного определенного типа; возможно существование нескольких деталей одного типа, но с разными характеристиками; в базе данных допускается информация о типах деталей, которые пока не представлены конкретными деталями, но «бестиповых» деталей не бывает».

В связи Типы деталей <=> Детали степень связи «один-ко-многим», n-связная сущность имеет обязательный класс принадлежности => **в соответствии с ER-методом достаточно использовать ДВЕ таблицы (по одной для каждой сущности); ключ каждой сущности служит в качестве первичного ключа соответствующей таблицы. Кроме того, ключ 1-связной сущности (Тип детали) должен быть добавлен как атрибут в таблицу, представляющую n-связную сущность (в таблицу Детали).**

Но у нас в таблице Детали уже есть такой атрибут – *Название* (он и будет вторичным ключом, соответствующим первичному ключу **Наименование**).

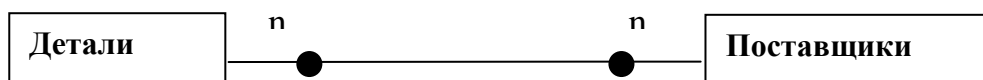
Типы деталей

Наименование	Изображение	Описание
Гайка
Шайба
Гвоздь
...

Детали

Код детали	<i>Название</i>	Вес	Диаметр	Металл	Цвет
1	Гайка	20	50	Сталь	Серый
2	Шайба	50	30	Сплав №1	Черный
3	Гайка	31	45	Латунь	Желтый
...

Сущности Детали - Поставщики



В этой диаграмме отражено правило: «каждую деталь может поставлять несколько поставщиков; каждый поставщик может поставлять несколько разных деталей; в базе данных допускается наличие поставщиков, которые в данный момент еще/уже ничего не поставляют, и наличие информации о деталях, которые еще/уже никто не поставляет».

В связи Детали \Leftrightarrow Поставщики степень связи «многие-ко-многим». В этом случае классы принадлежности сущностей не влияют на количество и структуру соответствующих таблиц \Rightarrow необходимо формирование трех таблиц – по одной для каждой сущности (ключ каждой сущности служит в качестве первичного ключа соответствующей таблицы), а также таблицы для связи. Таблица, создаваемая для связи, должна иметь по одному ключу от каждой сущности.

Поставщики

Код пост	Фамилия И.О.	Страна	Город	Адрес	Телефон	Надежность
1	Орлов А.С.	Россия	Москва	Лесная 34-1-75	263-67-89	10
2	Станов О.Т.	Россия	Курск	Новая 23-56	23-45-12	35
3	Рыбаков И.И.	Украина	Ровно	Рыбная 2-34	34-54-12	15
...

Детали

Код детали	Название	Вес	Диаметр	Металл	Цвет
1	Гайка	20	50	Сталь	Серый
2	Шайба	50	30	Сплав №1	Черный
3	Гайка	31	45	Латунь	Желтый
...

Поставки

Кто	Что	Сколько	Цена изделия	Цена доставки	Дата доставки	Оформлено
1	1	3000	234,56р.	4,56р.	29.10.03	да
2	3	4000	254,90р.	2,90р.	5.12.03	да
1	3	23000	294,00р.	4,00р.	12.01.04	нет
3	2	1200	136,58р.	6,58р.	20.11.03	да
2	2	45000	504,77р.	5,77р.	15.11.03	да

Таким образом, таблицы Поставщики и Детали предназначены для хранения значений атрибутов экземпляров соответствующих сущностей, а таблица Поставки – для реализации связи между этими сущностями.

Таблица Поставки содержит два вторичных ключа: поле **Кто** соответствует первичному ключу таблицы Поставщики, поле **Что** - первичному ключу таблицы Детали.

Кроме того, в таблицу-связку Поставки были добавлены дополнительные атрибуты, характеризующие свойства каждой поставки: Сколько, Цена изделия, Цена доставки, Дата доставки, Оформлено. Очевидно, что это важная информация для анализируемой предметной области, но она не могла появиться в качестве атрибутов ни в одной из рассмотренных ранее сущностей. Например, информация о количестве поставляемых деталей (атрибут Сколько) не может быть атрибутом сущности Детали, т.к. одну и ту же деталь разные поставщики могут по-

ставлять в разных количествах. Этот атрибут не может храниться и в таблице Поставщики как атрибут конкретного поставщика, поскольку объемы поставок различных деталей одним поставщиком могут быть разными. А появление такого атрибута в таблице Поставки логично: конкретный поставщик конкретную деталь поставляет в определенном количестве.

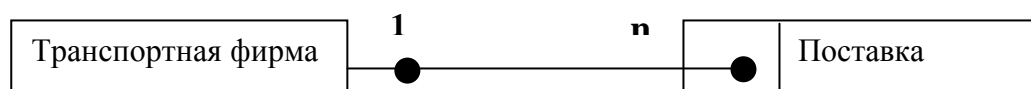
Сущности Транспортные фирмы - Детали - Поставщики

Анализ связи сущности Транспортные фирмы с другими сущностями нашей предметной области требует особого внимания.

Ясно, что транспортные фирмы связаны с поставкой конкретных деталей, но если мы зафиксируем эту связь, то потеряем информацию о том, **какой поставщик** поставляет эту деталь с помощью этой транспортной фирмы.

Транспортные фирмы также связаны с поставщиками, но если мы зафиксируем эту связь, то потеряем информацию о том, **какую деталь** поставляет этот поставщик с помощью этой транспортной фирмы.

Для того чтобы зафиксировать связь всех трех сущностей (Деталь - Поставщик - Транспортная фирма), логичнее всего рассмотреть связь между Поставками и Транспортными фирмами. Таблица-связка Поставки уже фиксирует связь между деталями и поставщиками, храня информацию о поставке конкретной детали конкретным поставщиком. Поэтому, если мы свяжем каждую поставку с транспортной фирмой, то получим содержательную связь трех сущностей.



В этой диаграмме отражено правило: «каждую поставку реализует одна транспортная фирма; каждая транспортная фирма может осуществлять несколько поставок; в базе данных допускается наличие транспортных фирм, которые в данный момент еще/уже ничего не поставляют, но каждая поставка реализуется некоторой транспортной фирмой».

В связи Транспортная фирма \Leftrightarrow Поставка степень связи «один-ко-многим», n-связная сущность имеет обязательный класс принадлежности \Rightarrow **в соответствии с ER-методом достаточно использовать две таблицы (по одной для каждой сущности). Кроме того, ключ 1-связной сущности (Транспортной фирмы) должен быть добавлен как атрибут в таблицу, представляющую n-связную сущность (в таблицу Поставки).**

Поставки

Кто	Что	Транспорт	Сколько	Цена изделия	Цена доставки	Дата доставки	Оформлено
1	1	5	3000	234,56р.	4,56р.	29.10.03	да
2	3	3	4000	254,90р.	2,90р.	5.12.03	да
1	3	4	23000	294,00р.	4,00р.	12.01.04	нет
3	2	1	1200	136,58р.	6,58р.	20.11.03	да
2	2	5	45000	504,77р.	5,77р.	15.11.03	да

Транспортная фирма

Код фирмы	Название	Страна	Город	...	Тип транспорта	Менеджер
1	Меридиан	Россия	Москва		Hyundai Porter	Соков И.Г.
2	Транс-Вектор	Белоруссия	Гродно		Газель	Рудин А.Р.

3	Rus Trans Broker	Россия	Москва		Foton Ollin	Боков Ф.А.
4	Коммивояжер	Латвия	Рига		Mercedes Atego	Липов А.А.
5	ТехноГрад	Украина	Донецк		Бычок	Седов П.Н.

В результате в таблице Поставки появился еще один вторичный ключ *Транспорт*, соответствующий первичному ключу таблицы Транспортные фирмы.

Ситуация, когда таблица-связка начинает играть роль сущности предметной области и вступает в связь с ее другими сущностями, довольно часто встречается при разработке проекта базы данных.

Таким образом, проектирование базы данных «Детали» завершено.