2.3. Представление связей

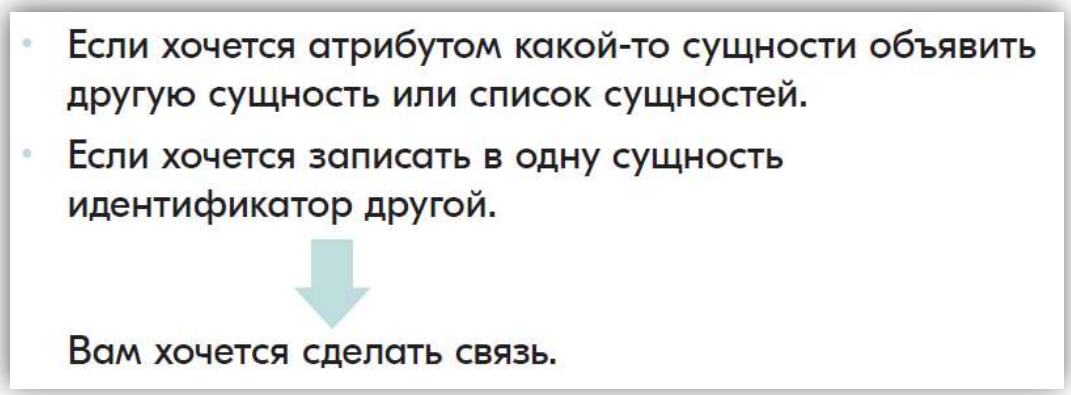
Модель «сущность - связь». Связи

Если бы назначением нашей базы данных было только хранение отдельных сущностей, то структура её была бы очень простой, но наши сущности могут вступать в какие-то взаимоотношения друг с другом. Одной из задач базы данных является отыскание одних сущностей по значениям других, то есть нам нужно устанавливать некоторые связи между сущностями. Поскольку в базе данных могут быть тысячи сущностей, то, чисто теоретически, между ними могут существовать миллионы связей. Именно наличие этих связей определяет сложность построенной модели.

Связь – это отношение между объектами. Также связью можно считать упорядоченный набор сущностей. У связи есть идентификаторы и идентификатором связи являются идентификаторы сущностей, которые в эту связь вступают.

В диаграммах «сущность-связь» имя связи обычно заключается в ромб, а связываемые сущности соединяются линиями, которые могут иметь направленные концы в зависимости от типа связи.

Как увидеть появление связи



Свойства связей

Связи, кроме идентификации связываемых объектов, могут иметь собственные атрибуты.

Подобные связи, также, как и сущности, объединяются в множества.

Кардинальное отличие сущностей от связей в том, что связи не могут существовать отдельно без связываемых объектов, а объекты могут жить своей отдельной жизнью, независимо от связи.

Обычно связи идентифицируются при помощи соединения ключей связываемых сущностей.

Характеристики связей

У связей есть как минимум три характеристики: это ***размерность, мощность*** и ***модальность***.

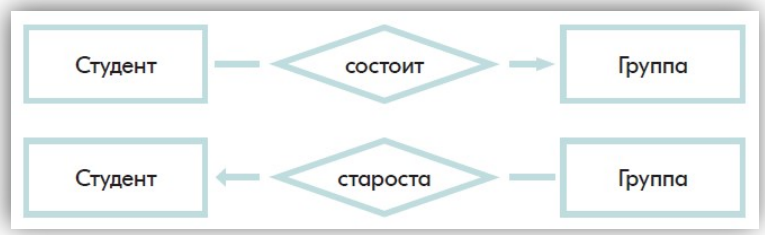
Классификация связей: размерность

Размерность связи определяет количество видов объектов, которые в связи участвуют. Есть наиболее часто используемые связи – ***бинарные*** (между двумя видами объектов), но связи бывают и более сложные, например, ***тернарные*** (если связь образуется тремя видами объектов), ***N-арные*** (в общем случае), также бывают ***рекурсивные*** связи, когда в связь вступают объекты одного и того же вида.

Пример: бинарная связь

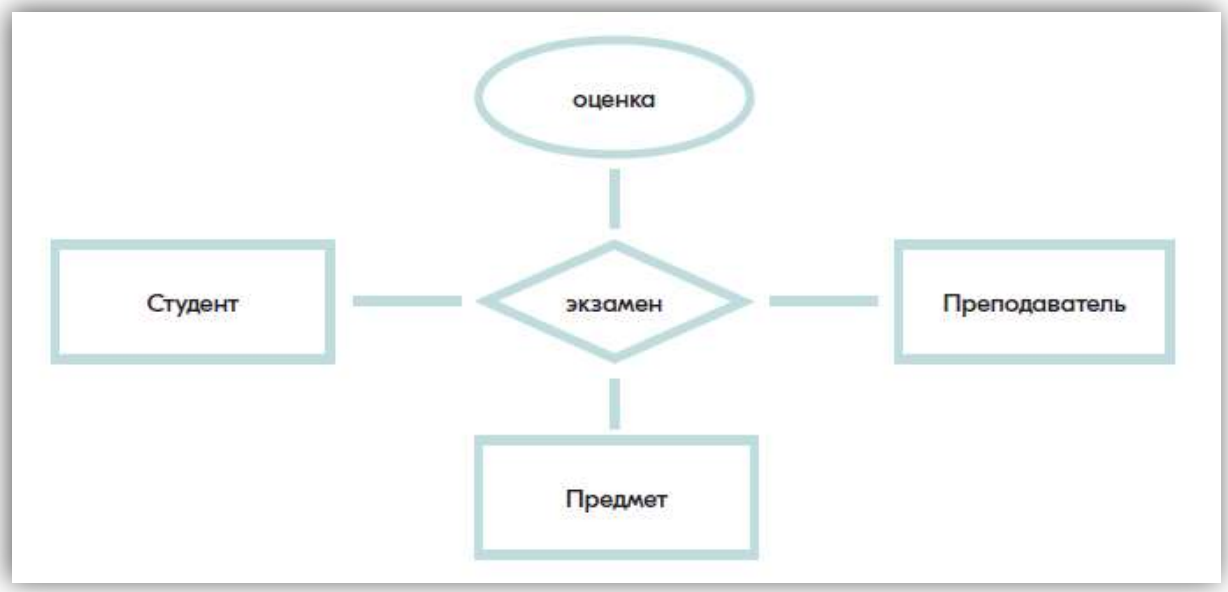
Связь между студентом и группой можно изобразить как связь «состоит», причем можно указать направление связи от студента к группе, указывая, что студент закреплен за одной группой.

Между одними и теми же видами сущностей может быть более одной связи, например, студент может состоять в группе, а может являться старостой группы.



Пример: тернарная связь

Для того, чтобы провести экзамен, нужно взаимодействие трех сущностей: это будет студент, который сдает экзамен; предмет, по которому проходит экзамен и преподаватель, который этот экзамен проводит.



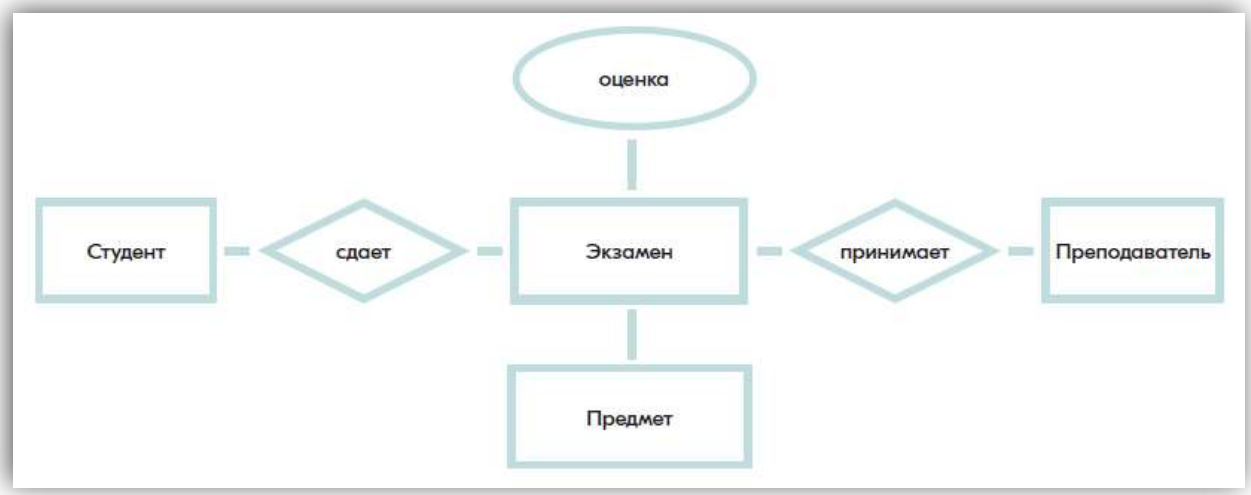
Также у данной тернарной связи будет свой собственный атрибут – оценка. Такой атрибут появляется у связи, потому что оценка не является свойством студента, не является свойством предмета, и тем более не является свойством преподавателя. Это будет именно свойство конкретной связи, которую мы назвали «экзаменом».

Преобразование тернарной связи в бинарную

В рамках модели «сущность-связь» может быть представлены не только тернарные, но и более сложные многомерные связи. Но такие связи очень сложно реализуются потом в контексте выбранной СУБД. Для того, чтобы облегчить ситуацию, всегда можно перейти от многомерной связи к бинарной.

Пример

В прошлом примере мы использовали тернарную связь «экзамен», но вместо этой связи можем ввести соединяющие множество сущности, то есть вместо связи «экзамен» мы можем ввести в базу данных сущность, которая будет называться «экзамен», и эта сущность будет вступать в бинарные связи с другими сущностями: с сущностью «студент», который будет экзамен сдавать; с сущностью «преподаватель», который будет экзамен принимать и с сущностью «предмет», который будет на этот экзамен назначен.

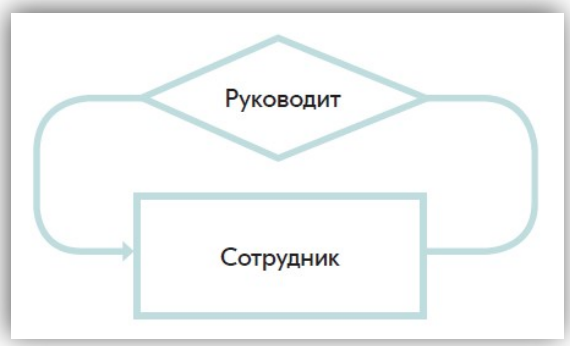


Таким образом мы можем перейти от сложных многомерных связей к бинарным.

Пример: рекурсивная связь

Предположим, что у нас хранится информация о сотрудниках, и один из сотрудников является руководителем.

Используя рекурсивные связи, мы можем выстраивать сложные иерархические структуры.

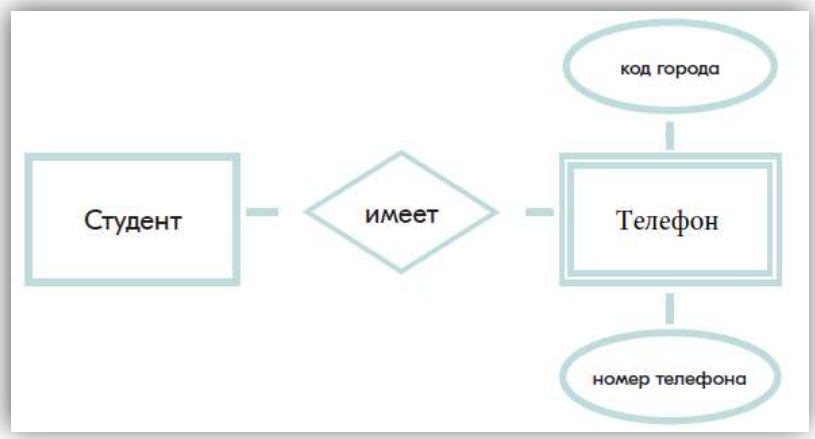


Слабые (зависимые) сущности и связи

Когда мы говорили о сущностях, которые имеют идентификаторы, мы описывали независимые наборы сущностей, но в нашей базе данных могут встречаться и зависимые сущности.

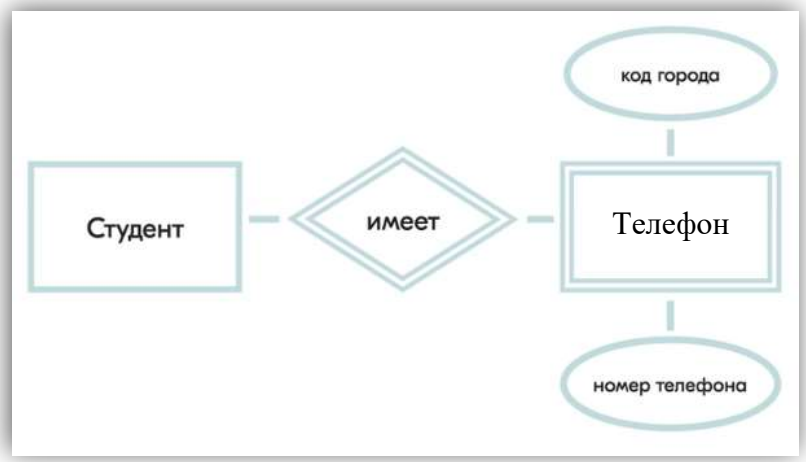
Пример

Мы хотим хранить информацию о студентах. Мы знаем, что у студента есть фамилия, имя, отчество, номер зачетки, который является ключом, адрес и телефон.



Посмотрим внимательнее на телефоны: у кого-то из студентов может совсем не быть телефона, а у кого-то их может быть несколько, но мы говорили, что значение каждого атрибута – атомарно. Возможный вариант – вынести телефон в отдельную сущность, которую мы так и назовем «телефон». В таком виде у сущности «телефон» идентификатора скорее всего не будет, потому что если мы приводим, например, рабочий телефон или домашний телефон. то у нескольких людей может быть один и тот же телефон. И выдумывать какой-то отдельный специальный ключ не стоит, потому что у нас просто есть возможность создать зависимую сущность.

Мы называем сущность зависимой тогда, когда она не может существовать отдельно без связи с другой независимой сущностью. Чтобы подчеркнуть слабость таких сущностей, мы окружаем их название двойной линией, и связи, в которые такие сущности вступают с независимыми сущностями, также будут заключены в двойные ромбы.



Такое описание зависимых сущностей позволяет нам не выдумывать специальный суррогатный ключ, который системе не нужен.