

# КОНЦЕПТУАЛЬНОЕ ПРОЕКТИРОВАНИЕ

# Entity-Relationship model

Модель «сущность-связь» (Entity-Relationship model - ER-модель) — модель данных, позволяющая описывать концептуальные схемы предметной области.

ER-модель используется при высокоуровневом (концептуальном) проектировании баз данных. С её помощью можно выделить ключевые сущности и обозначить связи, которые могут устанавливаться между этими сущностями.

Во время проектирования баз данных происходит преобразование ER-модели в конкретную схему базы данных на основе выбранной модели данных (реляционной, объектной, сетевой или др.).

# Entity-Relationship model

ER-модель представляет собой формальную конструкцию, которая сама по себе не предписывает никаких графических средств её визуализации.

В качестве стандартной графической нотации, с помощью которой можно визуализировать ER-модель, была предложена диаграмма сущность-связь (ER-диаграмма) (англ. entity-relationship diagram, ERD).

Понятия *ER-модель* и *ER-диаграмма* часто ошибочно не различают, хотя для визуализации ER-моделей предложены и другие графические нотации (см. ниже).

Модель «сущность-связь» была предложена в 1976 году Питером Пин-Шен Ченом (англ. Peter Pin-Shen Chen), американским профессором компьютерных наук в университете штата Луизиана.

# Нотации (графические диаграммы)

- Нотация Питера Чена
- Нотация Мартина
- Нотация IDEF1x
- Нотация Баркера

# Нотация Питера Чена (сущность)

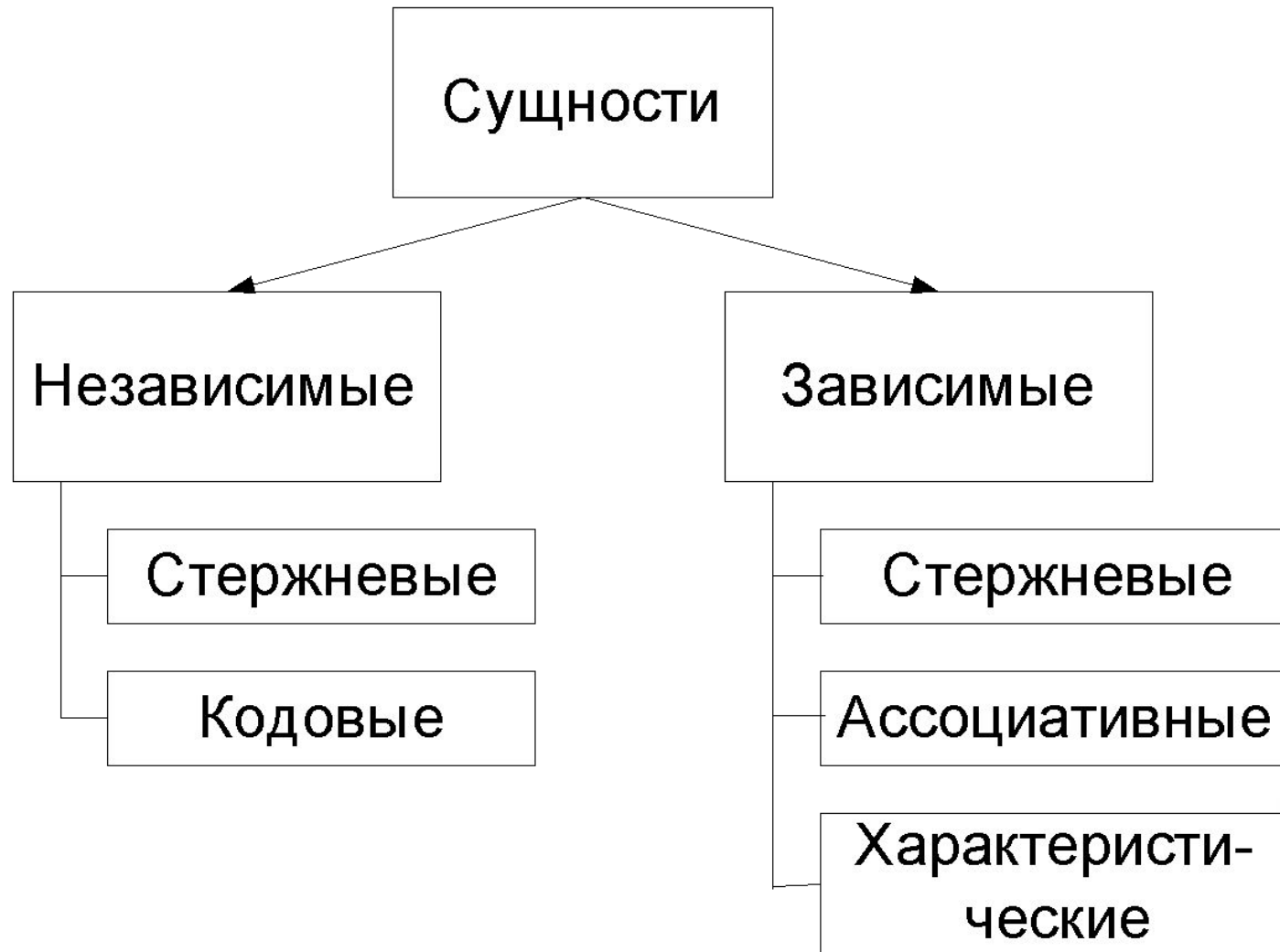
**Сущность** - это физическое представление логической группировки данных. Сущности представляют собой объекты, данные о которых корпорация заинтересована сохранять.

Сущности могут быть вещественными, реальными объектами, или неосязаемыми концептуальными абстракциями. Сущности не предназначены для представления единичного объекта, они представляют набор экземпляров, содержащих информацию, представляющую интерес с точки зрения их уникальности. Сущности для ясности и обеспечения целостности обозначаются существительными в единственном числе.

Существует две основных группы сущностей: **зависимые и независимые**. Независимая сущность не нуждается в информации из другой сущности для идентификации уникального экземпляра. Первичный ключ независимой сущности не включает в себя первичных ключей других сущностей. Зависимая сущность должна привлекать информацию из другой сущности для идентификации уникального экземпляра. Первичный ключ зависимой сущности включает первичные ключи одной или более родительских сущностей.

И зависимые, и независимые сущности можно разделить на несколько типов

# Нотация Питера Чена



# Нотация Питера Чена

- Стержневые сущности - их иногда называют основными или первичными сущностями. Они представляют важные объекты, информацию о которых следует хранить.
- Коды/ссылки/классификаторы - эти сущности содержат строки, определяющие набор значений или область определения для атрибута.
- Ассоциативные сущности - эти сущности используются для разрешения отношений многие-ко-многим.
- Характеристические сущности - эти сущности бывают двух типов: исключаяющие и включающие.

# Нотация Питера Чена (именование сущностей)

Соглашения об именовании сущностей.

- Имя сущности должно быть достаточно описательным. Используйте имена из одного слова, только когда они являются названием широко распространенных концепций. Подумайте об использовании словосочетаний на основе существительных.
- Имя сущности должно быть существительным или словосочетанием на основе существительного в единственном числе. Используйте ПЕРСОНА вместо ПЕРСОНЫ или ЛЮДИ, и КОНТЕЙНЕР - вместо КОНТЕЙНЕРЫ.
- Имя сущности должно быть уникальным. Использование одинаковых имен для сущностей, содержащих различные данные, или разных имен для сущностей, содержащих одинаковые данные, будет без необходимости вводить в заблуждение разработчиков и конечных пользователей.
- Имя сущности должно указывать на данные, которые будут храниться в каждом из экземпляров.
- Имя сущности не должно содержать специальных символов (таких как !, @, #, \$, %, &, \* и тому подобных) или указывать на принадлежность (МОРОЖЕНОЕ ПЕРСОНЫ).
- Имя сущности не должно содержать акронимов или аббревиатур, если только они не являются частью принятых соглашений об именовании.

Даже хороших имен, указывающих пользователю, какую информацию стоит ожидать от сущности, обычно недостаточно. Каждая сущность нуждается в ясном, точном и полном описании или определении, чтобы быть однозначно интерпретируемой в рамках корпорации. Описание сущности должно объяснять смысл сущности и ее значение для корпорации.



# Нотация Питера Чена (обозначение сущностей)

Группы

- Независимая сущность

Студенты

- Зависимая сущность

# Нотация Питера Чена (Атрибут)



**Атрибут** является логическим представлением фактов, данные о которых корпорация заинтересована хранить. Атрибуты представляют данные об объектах (сущностях), которые необходимо иметь корпорации. Атрибуты представляются именами существительными, которые описывают характеристики сущностей.

Атрибуты формируют логические группы, описывающие каждый экземпляр сущности. Конкретным экземпляром атрибута является значение.

Корректная модель атрибута обладает следующими признаками:

- Значение атрибута представляет интерес для корпорации.
- В логической модели присутствует единственный экземпляр атрибута.
- Атрибут имеет логический тип данных и область определения.
- Значение атрибута определяется как требуемое или необязательное.
- Атрибут имеет имя и описание.
- Для каждого экземпляра сущности может использоваться только одно значение.

# Нотация Питера Чена (Атрибут)



Каждый из присутствующих в модели атрибутов должен служить удовлетворению одного или нескольких требований к информации. Модель должна содержать только те атрибуты, которые необходимы для представления фактов, интересующих корпорацию в рамках рассматриваемой предметной области.

Атрибут должен присутствовать в ER-модели в единственном экземпляре. "Один факт в одном месте" (Дейт, 1986).

Природа атрибутов может быть различной.

# Нотация Питера Чена (Атрибут)



- **Простой / составной**  
(фамилия / адрес)
- **Множественный / единичный**  
(специальность / таб. номер)
- **Базовый / производный**  
(год рождения / возраст)
- **Статический / динамический**  
(таб. номер / адрес)
- **Условный**  
(ученая степень)
- **ключевой или неключевой**  
(идентификатор / имя)

# Нотация Питера Чена (Атрибут)



- Атрибут может быть *множественным* или *единичным* – т.е. атрибут, задающий свойство, может одновременно иметь несколько значений или, соответственно, только одно.
- Атрибут может быть *простым* (не подлежащих дальнейшему делению с точки зрения прикладных задач) или *составным* – если его значение составляется из значений простых свойств.
- Атрибут может быть *базовым* или *производным*.
- Атрибут может быть *условным*, если наличие некоторого свойства для всех экземпляров сущности не является обязательным.
- Значения атрибутов могут быть постоянными - *статическими*, или *динамическими*, т.е. меняться со временем. Атрибут может быть *неопределенным*, если он является динамическим, но его текущее значение еще не задано.
- Атрибут может рассматриваться как *ключевой*, если его значение уникально. *Неключевыми* являются атрибуты, значения которых зависят от значений первичного ключа или составного первичного ключа.

# Нотация Питера Чена (Первичный ключ)



Для идентификации конкретного экземпляра сущности необходимо определить первичный ключ.

**Первичным ключом** служит атрибут или набор атрибутов, уникально идентифицирующий единственный экземпляр сущности. Значения ключевых атрибутов не зависят от значений никаких других атрибутов.

Первичный ключ должен быть статическим и неразрушаемым.

Процесс выявления первичных ключей начинается с определения всех потенциально ключевых атрибутов, называемых кандидатами в ключи. *Кандидатом* в ключи может быть и один атрибут, и комбинация нескольких атрибутов. Если кандидатов в ключи не существует, или кандидатом является составной ключ, который слишком велик и громоздок, следует рассмотреть возможность использования искусственного уникального идентификатора. Ключи, заимствованные из родительской сущности, называются внешними ключами.

# Нотация Питера Чена (Выявление первичных ключей)

Номер  
паспорта

- *Кандидатом в ключи* является атрибут или группа атрибутов, идентифицирующих конкретный экземпляр сущности. Кандидат в ключи представляет механизм определения потенциальных первичных ключей для идентификации конкретных экземпляров сущности. Кандидат в ключи, который не выбран в качестве первичного ключа, еще называют альтернативным ключом. *Альтернативный ключ* - это атрибут или группа атрибутов, которые могут быть использованы при индексировании.
- *Составной ключ*. Для составных ключей каждая составляющая ключа должна иметь значение для каждого экземпляра. Ни одна часть ключа не должна быть неопределенной (NULL). Все части ключа являются обязательными и не могут быть опущены.
- *Искусственные первичные ключи* - это атрибуты, созданные с единственной целью идентификации конкретных экземпляров сущности. Искусственный первичный ключ часто формируется простой последовательной нумерацией каждого из экземпляров сущности. Искусственные первичные ключи, гарантированно обладают особенностями хороших первичных ключей.
- *Внешние ключи*. Когда первичный ключ одной сущности мигрирует в другую таблицу, он называется внешним ключом. Внешние ключи "связывают" сущности, представляя отношения между ними. Внешние ключи могут быть как идентифицирующими, так и неидентифицирующими. *Идентифицирующие внешние ключи* становятся частью первичного ключа в той сущности, в которую они мигрировали. *Неидентифицирующие внешние ключи* становятся не ключевыми атрибутами.

# Нотация Питера Чена (Распространенные ошибки, связанные с первичным ключом)



Распространенными ошибками, связанными с первичным ключом, являются:

- 1. Не уникальность: первичный ключ не является уникальным для каждого из экземпляров.
- 2. Требуемое значение/неопределенность: первичный ключ не имеет значения для некоторых из экземпляров.

Хороший первичный ключ будет обладать следующими признаками:

- Значение гарантирует уникальность для каждого из экземпляров
- Значение не имеет скрытого смысла
- Область определения значений будет оставаться постоянной с течением времени
- Значения существуют для каждого из экземпляров сущности




# Нотация Питера Чена (Соглашения об именовании атрибутов)

Номер  
паспорта

Соглашения об именовании атрибутов.

- Имя атрибута должно быть достаточно описательным. Желательно использовать словосочетания на основе существительных в форме объект/ модификатор/ класс. Форма *объект/модификатор/класс* - широко распространенное в отрасли соглашение об именовании атрибутов. Это соглашение побуждает использовать имена атрибутов, состоящие из трех частей. Часть объект иногда называют субъектом или основным словом. В качестве объекта обычно используется имя сущности.
- По возможности имя атрибута должно включать имя сущности. Используйте "Имя для персоны" вместо просто "Имя".
- Имя атрибута должно указывать на значения конкретных экземпляров атрибута. Использование одинаковых имен для атрибутов, содержащих различные данные, или разных имен для атрибутов, содержащих одинаковые данные, будет без необходимости вводить в заблуждение разработчиков и конечных пользователей.
- Имя атрибута должно использовать язык бизнеса вместо языка технических описаний.
- Имя атрибута не должно содержать специальных символов (таких как !, @, #, \$, %, &, \* и тому подобных) или указывать на принадлежность (Имя, принадлежащее персоне).
- Имя атрибута не должно содержать акронимов или аббревиатур, если только они не являются частью принятых соглашений именовании.

# Нотация Питера Чена

-  Фамилия - Атрибут
-  Номер  
паспорта - Первичный ключ

# Нотация Питера Чена (Отношения)

- **Отношения** представляют взаимосвязи между объектами, о которых корпорация заинтересована хранить данные. **Отношения** выражаются глаголами или глагольными фразами, которые описывают взаимосвязь.
- **Отношение** - это ассоциация или "связь" между двумя сущностями. Отношение представляется в модели линией, соединяющей две сущности и глагольной конструкцией, которая описывает, как две сущности зависят друг от друга.

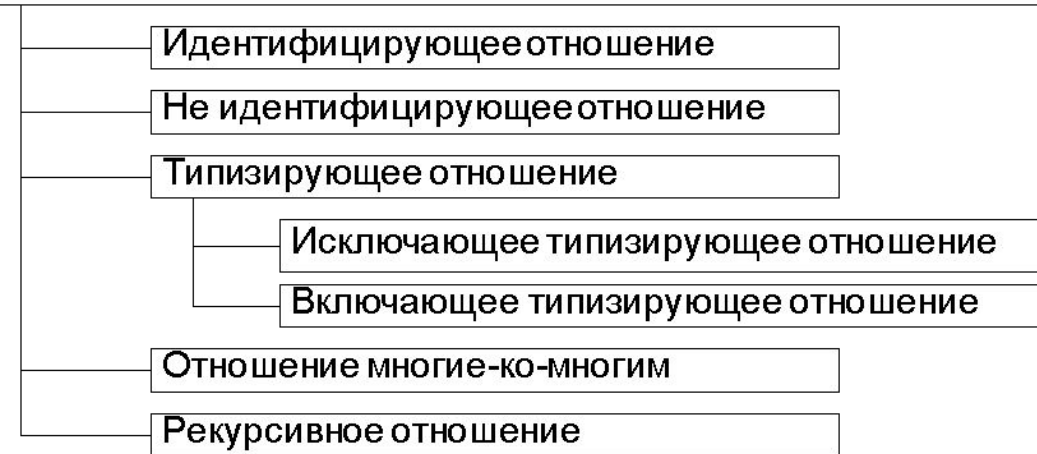
# Свойства отношений

**Степень** отношения - число сущностей, ассоциированных с отношением

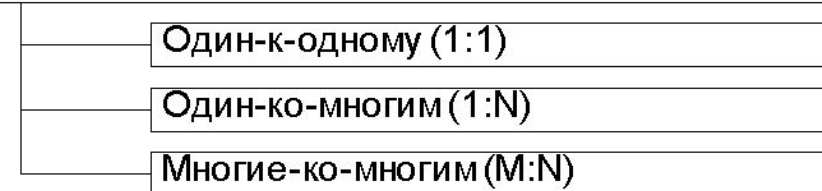


**Направленность** отношения - указание исходной сущности в отношении

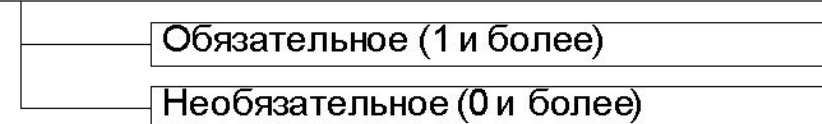
**Тип** отношения



**Количество элементов** - максимальное число экземпляров одной сущности, которые могут быть связаны с экземплярами другой сущности



**Обязательность отношения** - минимальное число экземпляров сущности, которые должны участвовать в отношении



# Свойства отношений

- **Степень** отношения представляет собой число сущностей, ассоциированных с отношением. В основном, отношения имеют степень единица (унарные отношения) или двойка (бинарные отношения). *Унарные*, или *рекурсивные* отношения представляют случаи, когда экземпляр сущности связан с другим экземпляром той же самой сущности. Бинарные отношения представляют случаи, когда одна сущность связана с другой. *Бинарные* отношения отражают наиболее распространенные взаимосвязи, присутствующие в реальном мире.
- **Направленность** отношения указывает на исходную сущность в отношении. Сущность, из которой отношение исходит, называется *родительской сущностью*. Сущность, в которой отношение заканчивается, называется *подчиненной сущностью*.

# Свойства отношений

Отношение между двумя сущностями, или сущности самой с собой, может принадлежать к одному из следующих **типов**:

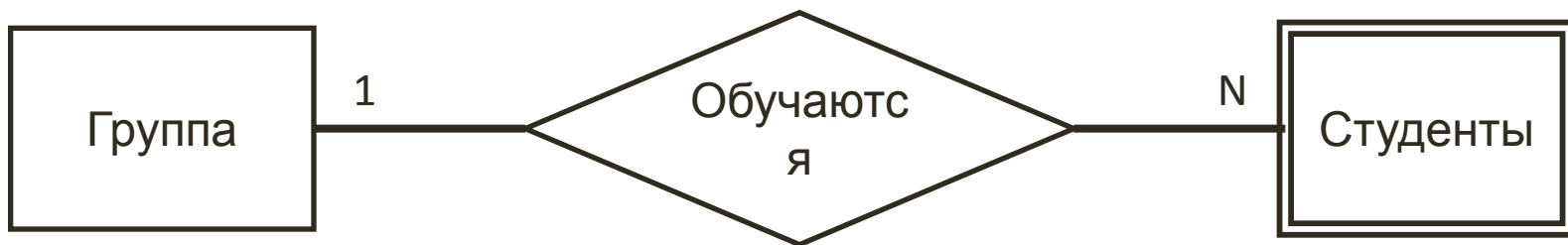
- *Идентифицирующим* является отношение между двумя сущностями, в котором каждый экземпляр подчиненной сущности идентифицируется значениями атрибутов родительской сущности.
- *Неидентифицирующим* является отношение между двумя сущностями, в котором каждый экземпляр подчиненной сущности не зависит от значений атрибутов родительской сущности.
- *Типизирующими* являются отношения между родительской и одной или более подчиненными сущностями.
- *Исключающие типизирующие* отношения указывают, что только одна подчиненная сущность идентифицируется родительской сущностью.
- *Включающее типизирующее отношение* указывает, что экземпляром родительской сущности могут определяться более одной вторичной сущности.
- Отношения *многие-ко-многим* возникают там, где один экземпляр одной сущности связан с несколькими экземплярами другой, и один экземпляр этой другой сущности также связан с несколькими экземплярами первой сущности. Эти отношения также называют неспецифическими.
- *Рекурсивное отношение* - это неидентифицирующее отношение между двумя сущностями, которое указывает, что экземпляр сущности может быть связан с другим экземпляром той же самой сущности. При рекурсивном отношении родительская и подчиненная сущности совпадают.

# Свойства отношений

**Количество элементов отношения.** Количество элементов отношения задает максимальное число экземпляров одной сущности, которые могут быть связаны с экземплярами другой сущности. Количество элементов определяется для обеих сторон отношения - для исходной и завершающей сущностей. Количество элементов часто выражается как **один** или **много**:

- **Один-к-одному** (1:1) - один и только один экземпляр сущности связан с одним и только одним экземпляром другой сущности.
- **Один-ко-многим** (1:N) - один и только один экземпляр родительской сущности связан со многими экземплярами подчиненной сущности.
- **Многие-ко-многим** (M:N) - много экземпляров одной сущности связаны с многими экземплярами другой сущности (также называется неспецифическим отношением).

# Свойства отношений





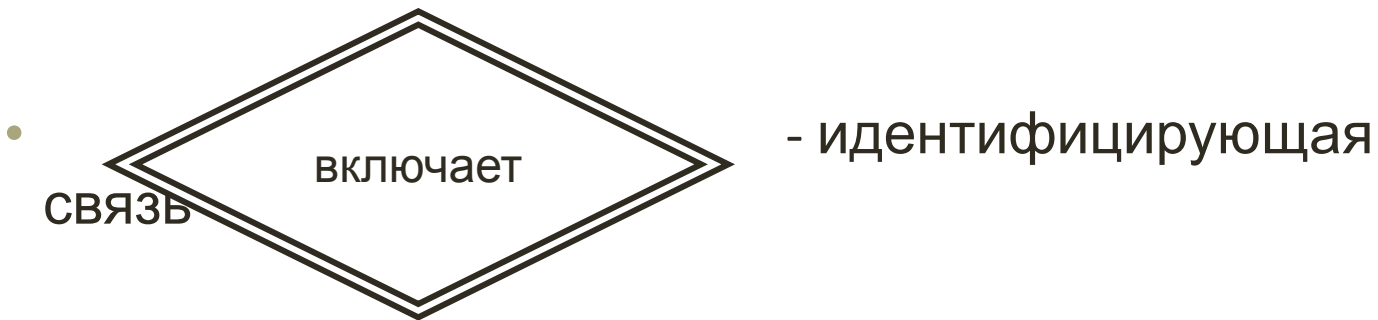
# Свойства отношений

**Обязательность отношения.** Обязательность отношения определяет, должны ли экземпляры сущности участвовать в отношении. Значение обязательности равно нулю в том случае, если экземпляр сущности не обязателен или не требуется, и равно единице, если наличие сущности требуется или обязательно.

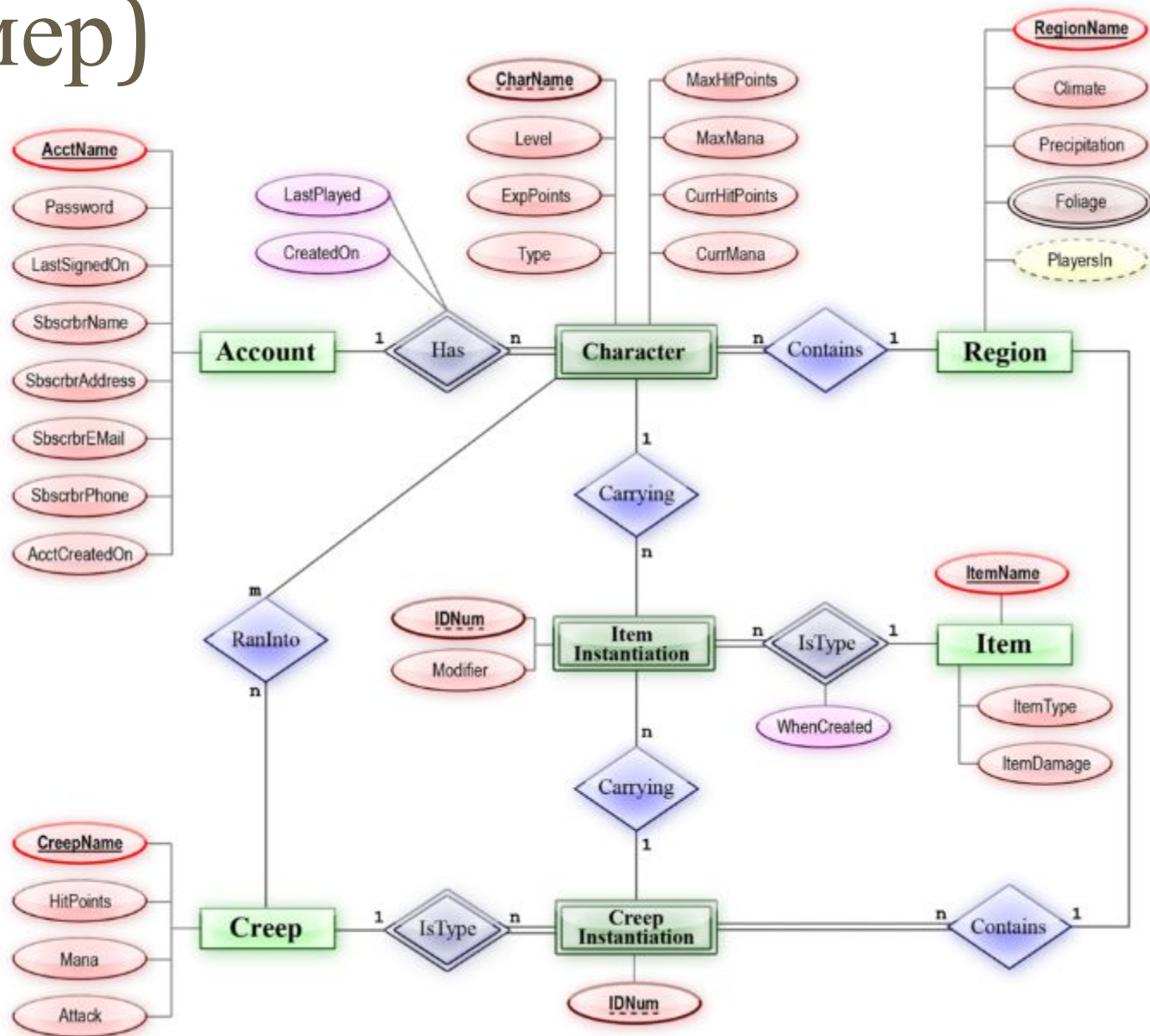
Распространенные ошибки, связанные с отношениями:

- 1. Назначение имени в форме глагольной конструкции, направленности, степени, типа, количества элементов и обязательности данному отношению должно подтвердить его корректность.
- 2. Как показывает опыт, реальные отношения один-к-одному очень редки и еще реже они реализуются на практике.
- 3. При определении необязательности отношения и определении того, требуется ли наличие экземпляра одной сущности для существования экземпляра другой, важно убедиться, что требуемый экземпляр всегда доступен.

# Отношение (связь)



# Нотация Питера Чена (пример)



# Нотация Мартина

# Нотация Мартина

Группы

- сущность

Студенты

- Зависимая сущность

# Нотация Мартина

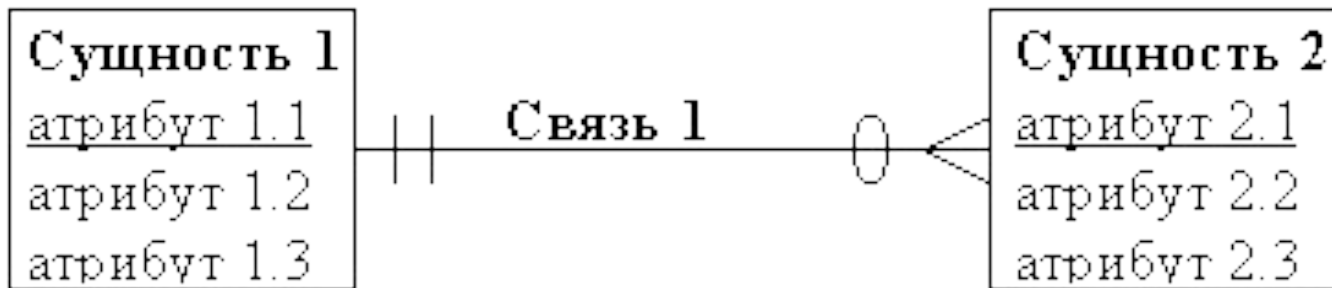
- Список атрибутов приводится внутри прямоугольника, обозначающего сущность. Ключевые атрибуты подчеркиваются. Связи изображаются линиями, соединяющими сущности, вид линии в месте соединения с сущностью определяет кардинальность связи:

# Нотация Мартина

Обозначение	Кардинальность
	нет
	1,1
	0,1
	M,N
	0,N
	1,N

# Нотация Мартина (имя связи)

- Имя связи указывается на линии ее обозначающей. Пример:



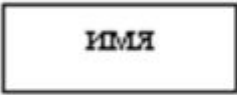
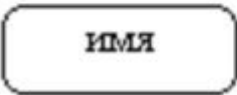


# Нотация IDEF1x

# Нотация IDEF1x



## Обозначение

сущностей

Элемент диаграммы	Обозначает
	независимая сущность
	зависимая сущность

## Обозначение

связей

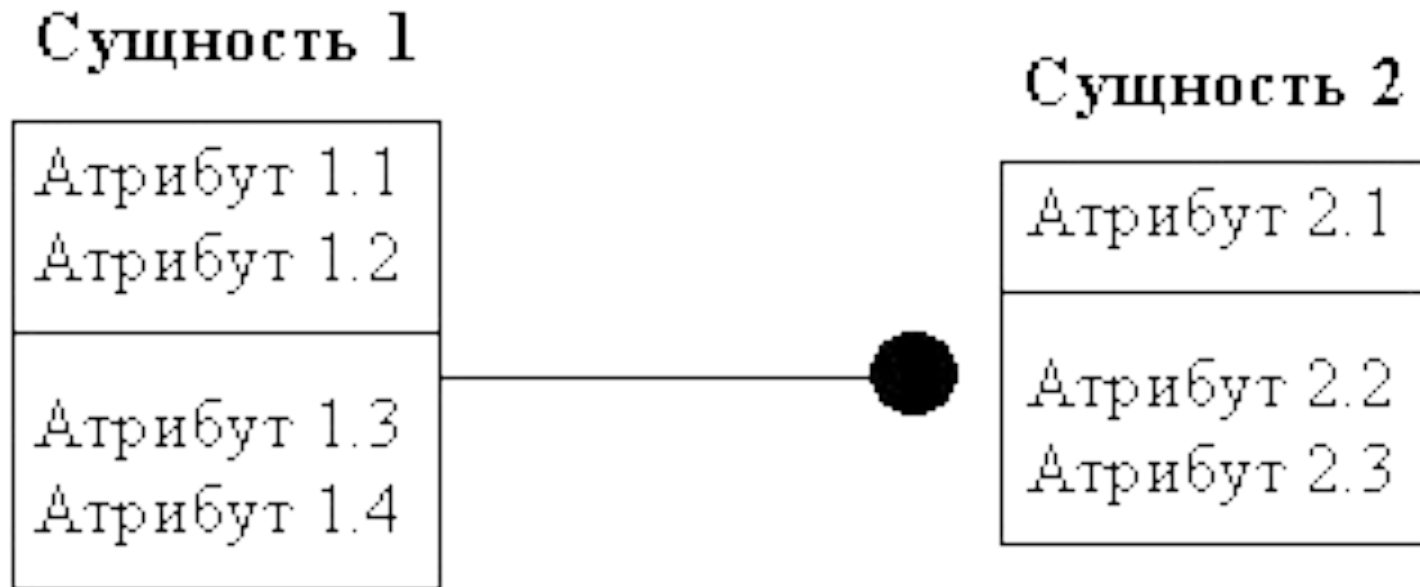
Элемент диаграммы	Обозначает
	идентифицирующая связь
	неидентифицирующая связь>

# Нотация IDEF1x

(обозначение кардинальности связей)

Элемент диаграммы	Обозначает
	1,1
	0,М
	0,1
	1,М
	точно $N$ ( $N$ - произвольное число)

# Нотация IDEF1x (пример)



# Нотация Баркера

# Нотация Баркера

- Сущности обозначаются прямоугольниками, внутри которых приводится список атрибутов. Ключевые атрибуты отмечаются символом # (решетка). Связи обозначаются линиями с именами, место соединения связи и сущности определяет кардинальность связи.

# Нотация Баркера (кардинальность связи)

Обозначение	Кардинальность
	0,1
	1,1
	0,N
	1,N

# Нотация Баркера (пример)

