

# Моделирование реляционных БД

Новакова Н.Е.

# Реляционная модель

Цель моделирования данных состоит в том, чтобы построить формальное представление данных, которое:

- ☐ скрывает неинтересные детали;
- ☐ заменяет значительные по объему компоненты компактными символами;
- ☐ подчеркивает важные факты;
- ☐ способствует пониманию системы в целом.

# Концептуальная, логическая и физическая модели

- ❑ *Концептуальная модель данных* отображает данные с точки зрения их смысла и в контексте пользователя.
- ❑ *Логическая модель данных* — это обобщенная реляционная схема базы данных, которая не зависит от способа ее физической реализации.
- ❑ *Физическая модель данных* — это проект базы данных для выбранного конкретного продукта СУБД и определенной конфигурации компьютера, на котором базу данных предполагается реализовать.

# Концептуальная модель

- ☐ сбор информации о сущностях и связях;
- ☐ отображение сущностей на ER-диаграмме;
- ☐ определение всех потенциальных ключей сущностей;
- ☐ отображение связей на ER-диаграмме;
- ☐ выявление и отображение иерархических связей.

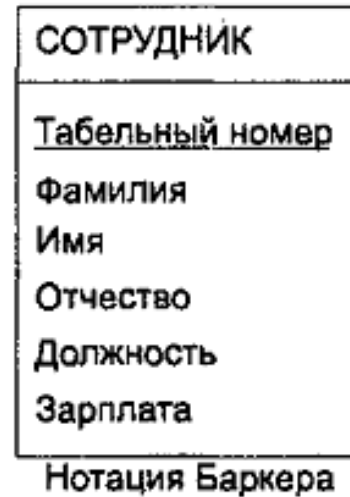
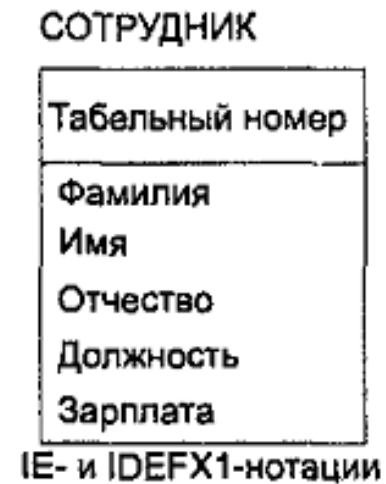
# Логическая модель

- ❑ нормализация в рамках первой нормальной формы и обусловленное ею уточнение перечня ключей сущностей и атрибутов;
- ❑ уточнение полного перечня атрибутов сущностей;
- ❑ агрегация элементов данных в новые сущности, если это необходимо;
- ❑ уточнение и формирование полного перечня возможных ключей, разделение этих ключей на первичные и альтернативные;
- ❑ нормализация с учетом новых ключей;
- ❑ замена связей "многие-ко-многим" ассоциативными сущностями и определение всех атрибутов уточненного списка сущностей;
- ❑ замена иерархических связей структурами, основанными на ограничении внешнего ключа.

# Физическая модель

*Физическая модель данных* — это проект базы для определенной целевой СУБД, который должен содержать полную информацию о содержимом системного каталога базы данных. Способ изображения физической модели — унифицированные графические нотации, принятые в выбранном CASE-

# Нотация ER-диаграмм



# Нотация ER-диаграмм в интерпретации Баркера



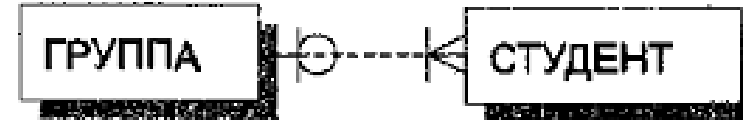


# Другие нотации

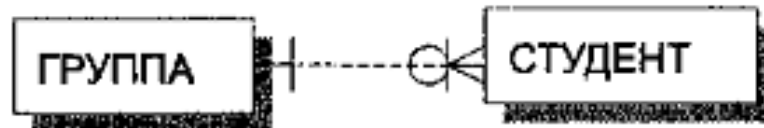
в группе должны быть студенты,  
студент должен быть в группе



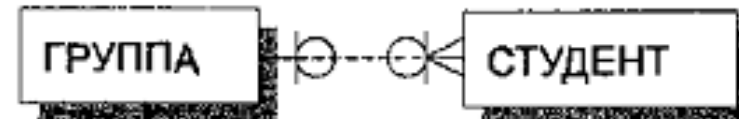
в группе должны быть студенты,  
студент может быть в группе



в группе могут быть студенты,  
студент должен быть в группе



в группе могут быть студенты,  
студент может быть в группе



# Пример ER-диаграммы для нотации IDEF1X

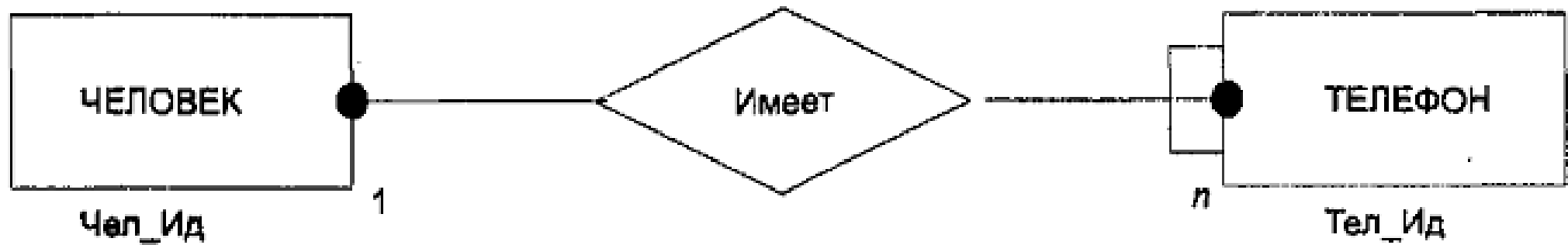
в группе могут быть студенты,  
студент должен быть в группе;  
в группе могут быть студенты,  
студент может быть в группе



в группе должны быть студенты,  
студент может быть в группе;  
в группе должны быть студенты,  
студент должен быть в группе



# ER-диаграммы в нотации Чена



# Идентифицирующие и неидентифицирующие связи

*Неидентифицирующая связь* имеет место, когда в связь вступают две самостоятельные сущности. Сама связь осуществляется путем так называемой *миграции ключевых атрибутов* родительской сущности в область неключевых атрибутов дочерней сущности. Мигрирующие атрибуты в дочерней сущности подчиняются определению внешнего ключа. Обозначается неидентифицирующая связь прерывистой линией.

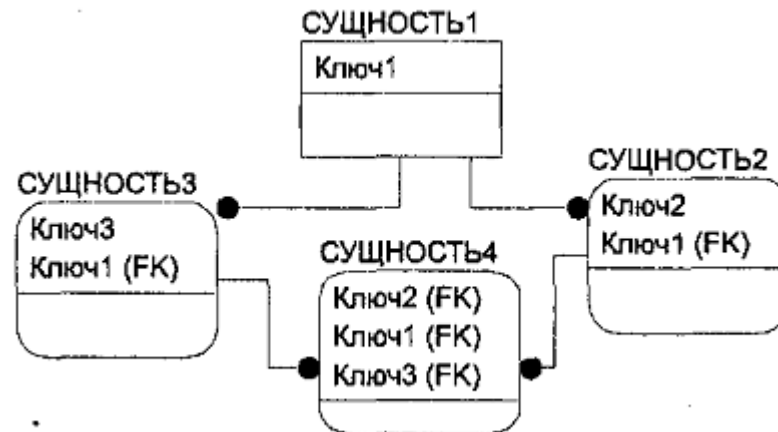
*Идентифицирующая связь* является особым случаем обязательной связи. Если сущность *A* связана с сущностью *B* идентифицирующей связью, то существование экземпляра *B* невозможно и не имеет смысла без существования экземпляра *A*. Сущность *B* в этом случае называют *зависимой* или *слабой* сущностью. Существование экземпляра такой сущности всегда зависит от существования экземпляра другой сущности.

Идентифицирующая связь практически всегда носит характер "один-ко-многим".

# Сильные и слабые сущности

Экземпляр *сильной*, или *независимой*, сущности существует сам по себе, независимо от всех других сущностей.

На диаграммах слабые сущности обозначаются прямоугольниками с закругленными краями, в то время как прямоугольник, изображающий сильные сущности, имеет прямые углы. Так принято в нотациях IDEF1X и IE. При этом идентифицирующая связь обозначается сплошной линией, а неидентифицирующая — прерывистой.



# Сущности

- ❑ *сущность* — это имеющее смысл в данной предметной области, существующее в действительности или воображаемое явление или объект, информацию о котором нужно сохранять или выяснять (Р. Баркер);
- ❑ *сущность* — это информационный контейнер в первой нормальной форме, который сохраняет фиксированный набор атрибутов и может существовать в числе экземпляров от нуля до нескольких, являясь реляционной абстракцией некоторой концепции реального мира.

# Экземпляр и ключ сущности

*Экземпляр сущности* (entity instance) — это конкретный представитель данной сущности. Экземпляры сущностей должны быть *различимы*, т. е. сущности должны иметь некоторые свойства, уникальные для каждого экземпляра этой сущности.

*Ключ сущности* — это *неизбыточный* набор атрибутов, значения которых в совокупности являются *уникальными* для каждого экземпляра сущности. *Неизбыточность* заключается в том, что удаление любого атрибута из ключа нарушает его уникальность.

---

# Определение ключа сущности

Поскольку в реляционной модели сущность в конце концов преобразуется в отношение, то определение ключа сущности практически аналогично определению ключа отношения. Напомним, что ключи бывают атомарными и составными. Атомарный ключ состоит из единственного атрибута. Составной ключ представляет собой набор атрибутов.

Наборы ключевых атрибутов сущностей определяются на самых ранних стадиях проектирования базы данных исходя из объективных связей между данными. Содержимое того или иного экземпляра сущности на конкретный момент времени не является основанием для определения в качестве ключа произвольных наборов, не совпадающих по значениям экземпляров атрибутов.



# Переход к физической модели

По окончании проектирования при переходе к физической модели сущности должны быть определены как таблицы, первичные и альтернативные ключи сущностей — как уникальные индексы, атрибуты сущностей — как поля этих же таблиц, а экземпляры сущностей — как записи с данными в соответствующих таблицах. Как правило, опытный бизнес-аналитик сразу, еще на стадиях концептуального и логического моделирования мыслит именно этими категориями.

Основное правило для сущностей следующее: каждый экземпляр сущности должен уникальным и четко определяемым образом отличаться от любого другого экземпляра этой сущности и от любых экземпляров других сущностей.

# Атрибуты

*Атрибут сущности* — это именованная характеристика, являющаяся некоторым свойством сущности. Наименование атрибута должно быть выражено существительным в единственном числе (возможно, с характеризующими прилагательными).

*Атрибутом* называется базовая единица информации об экземпляре сущности, которая этот экземпляр качественно или количественно оценивает, идентифицирует, классифицирует или выражает его состояние. *Атрибут* — это свойство сущности, которое зависит только от этой сущности и ни от чего другого и может иметь только одно значение в данный момент времени.

# Проектирование атрибутов

При проектировании атрибутов полезно задавать такие вопросы.

- ☐ Какие данные о сущности мы хотим хранить?
- ☐ Какие свойства есть у экземпляра этой сущности, даже если вокруг нее больше ничего нет?
- ☐ Есть ли у экземпляра этой сущности только один экземпляр этой вещи?
- ☐ Может ли изменяться описанная атрибутом характеристика сущности с течением времени?

# Правила проектирования атрибутов

- ☐ *Атрибут должен описывать именно свою сущность.*
- ☐ *Не стоит объединять в один атрибут разнородные данные.*
- ☐ *Не рекомендуется создавать атрибуты, которые могут иметь разный смысл в зависимости от своего конкретного значения.*
- ☐ *Не стоит задавать "особенных" диапазонов значений и правил трактовки данных.*

# Обязательные и необязательные атрибуты

Атрибуты бывают *обязательными*, т. е. такими, которые не могут принимать NULL-значений, и *необязательными*, если NULL-значения для них допустимы. Согласно правилу целостности сущностей NULL-значения недопустимы для атрибутов, выбранных в качестве первичных ключей сущности.

# СВЯЗИ

- ❑ *связь* является логическим соотношением между сущностями. Каждая связь должна именоваться глаголом или глагольной фразой (Relationship Verb Phrase);
- ❑ *связь* — это некоторая ассоциация между несколькими сущностями. Другими словами, связь представляет собой соединение между двумя или более сущностями и обычно выражается глаголом;
- ❑ *связь* — это отношение или правило, существующее между сущностями, непротиворечивое, неизменяемое выражение некоторого правила предметной области или логический объект, который может быть прикреплен к другой сущности;
- ❑ *связь* — это именованная, значащая ассоциация между двумя сущностями или сущности с самой собой (Р. Баркер).

# Связи между сущностями

Одна сущность может быть связана с другими сущностями или сама с собой. Приведем следующие примеры описания связей:

- ☐ студент *входит* в группу;
- ☐ студент *может* являться старостой группы;
- ☐ потоку, *состоящему из нескольких* групп, *читается* лекция по определенному предмету;
- ☐ курс *может* читаться только после другого курса;
- ☐ сотрудник *может* иметь несколько детей или *не иметь* их вовсе.

# Бинарные связи

Связь между двумя сущностями называется *бинарной*. Это основной тип связи, с помощью которого в конечном счете выражаются связи другого типа. Для каждого из двух концов бинарной связи задаются основные *характеристики* связи:

- ❑ *название*, которое описывает семантику связи, с точки зрения одной из сущностей;
- ❑ *кардинальное число (степень, мощность)*, которое описывает количество возможных связей для каждой из сущностей-участниц;
- ❑ *степень участия сущности в связи (обязательность связи, модальность связи, класс принадлежности)*.



# Мощности связей

Как известно, бинарная связь бывает "один-к-одному" (1:1), "один-ко-многим" (1 :  $n$ ) или "много-ко-многим" ( $n$  :  $m$ ), что задается опцией, называемой *мощность связи*, или *кардинальность связи* (cardinality). Кроме того, связь характеризуется таким параметром, как модальность, который может принимать всего два значения: обязательная и необязательная.

Связи на диаграмме обозначаются линиями, соединяющими сущности. Концы этих линий выделяются различными символами, задающими степень связи и ее обязательность.

# Наименования связей

Если речь идет о бинарных связях, то каждая связь имеет два конца и одно или два наименования. Наименование обычно выражается в неопределенной глагольной форме: "иметь", "принадлежать" и т. п. Каждое из наименований относится к своему концу связи. Иногда наименования не пишутся ввиду их очевидности.

Всегда удобно давать не одно, а два названия связи с двух сторон линии, т. е. с точки зрения сущностей, по обе стороны связи, например, студент *находится* в группе, группа *состоит* из студентов.

# Параметры и классификация бинарных связей

Бинарные связи — это связи, в которые вступают ровно две сущности. Важнейшее свойство связи — кардинальное число. Синонимами этого понятия являются мощность связи или степень связи. Оно описывает максимально возможное количество связей для каждого из экземпляров сущностей, участвующих в связи. В зависимости от значения кардинального числа различают бинарные связи следующих типов:

- связь типа *"один-к-одному"* означает, что один экземпляр первой сущности связан не более чем с одним экземпляром второй сущности и, наоборот, один экземпляр второй сущности связан не более чем с одним экземпляром первой сущности;

# Параметры и классификация бинарных связей (продолжение)

- связь типа "*один-ко-многим*" означает, что один экземпляр первой сущности связан с несколькими экземплярами второй сущности, но при этом один экземпляр второй сущности связан не более чем с одним экземпляром первой сущности. Сущность со стороны "один" (в нашем случае вторая сущность) называется *родительской*, а со стороны "многих" (в нашем случае первая сущность) — *оочернеи*.
- связь типа "*многие-ко-многим*" означает, что каждый экземпляр первой сущности может быть связан с несколькими экземплярами второй сущности, и каждый экземпляр второй сущности может быть связан с несколькими экземплярами первой сущности. Связь типа "многие-ко-многим" должна быть заменена двумя связями типа "один-ко-многим" путем создания промежуточной сущности.

# Обязательность связи

Вторым параметром связи является ее обязательность, называемая также *модальностью связи*, или *классом принадлежности*. Каждая связь может иметь одну из двух модальностей:

- ❑ модальность "*может*" соответствует необязательному классу принадлежности и означает, что любой экземпляр сущности с таким типом модальности может быть связан с одним или несколькими экземплярами другой сущности или не связан ни с одним экземпляром другой сущности;
- ❑ модальность "*должен*" соответствует обязательному классу принадлежности и означает, что каждый экземпляр сущности *обязан быть связан не менее чем с одним* экземпляром другой сущности.

# Основные этапы логического проектирования по методологии «сущность-связь»

- ❑ детально изучить предметную область в целях составления ее словесного описания и конкретизации необходимых бизнес-правил. Сформулировать функциональные требования к базе данных, выявив основные запросы и транзакции;
- ❑ выявить сущности, которые необходимо включить в модель, и определить первичные ключи каждой из выявленных сущностей;
- ❑ выявить и описать смысл, тип и возможные значения всех представляющих интерес атрибутов для каждой сущности. Определить домены для каждого атрибута;
- ❑ определить типы связей между выявленными сущностями;

# Основные этапы логического проектирования по методологии «сущность-связь» (продолжение)

- ❑ построить графическое описание функциональных зависимостей между атрибутами и выявить минимальное покрытие;
- ❑ для построенной ER-диаграммы следует получить предварительные отношения, пользуясь правилами их генерации, и разместить в них выявленные атрибуты сущностей;
- ❑ выполнить нормализацию модели, используя минимальное покрытие;
- ❑ принять обоснованные решения по введению в модель контролируемой избыточности и способов ее контроля;

# Основные этапы логического проектирования по методологии «сущность-связь» (продолжение)

- ❑ получить окончательное проектное решение в виде наборов отношений с присвоенными им смысловыми именами, указанными ключами и перечнем атрибутов для каждого отношения. Проверить, все ли атрибуты нашли свое место;
- ❑ убедиться в возможности реализации необходимой функциональности базы данных на полученном проектном решении. После этого можно переходить к работе с физической моделью базы данных и генерировать ее системный каталог.