



Базы данных и  
Информационные системы  
2/15 Концептуальное моделирование ИС

Кузиков Б.О.  
Сумы, СумГУ  
2013



# Модели БД

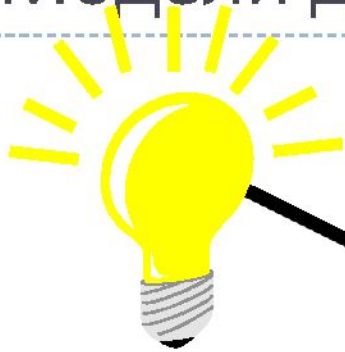
---

- ▶ **Информационная модель** данных отражает потребности системы в данных и связи между данными с точки зрения их потребителей – пользователей.
- ▶ На основе информационной модели в процессе проектирования создаются логическая и физическая модели данных.
- ▶ **Логическая модель** (Сущностная) данных является независимым логическим представлением данных.
- ▶ **Физическая модель** (Табличная) данных содержит определения всех реализуемых объектов в конкретной базе данных для конкретной СУБД.



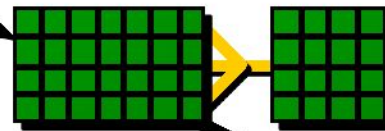
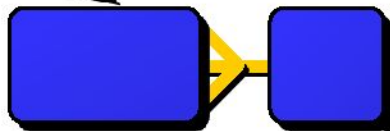
# Модели данных

---

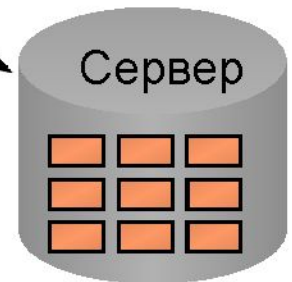


Модель  
системы в  
голове клиента

Сущностная модель  
модели клиента



Табличная модель  
сущностной модели



Таблицы на диске

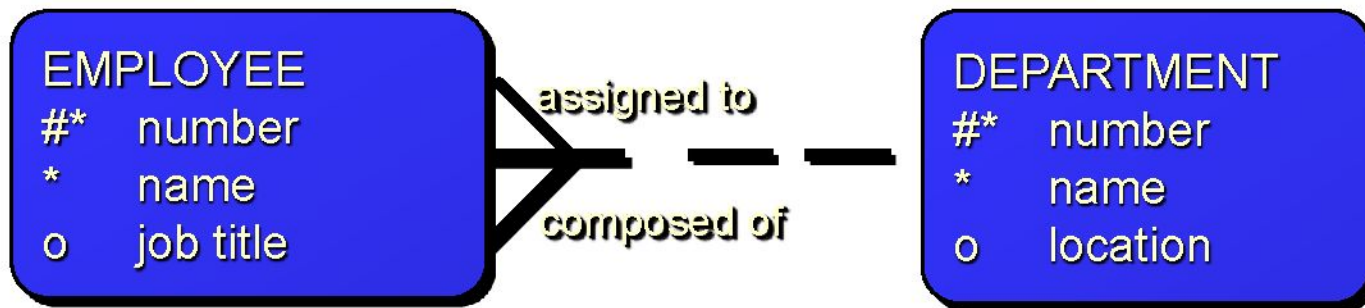




# ER-модель

---

ER-диаграмма создается на основе спецификаций или рассказов



## Сценарий

- “. . . Добавить одного или более сотрудников в отдел . . . .”
- “. . . Некоторые отделы все еще не имеют ассоциированных сотрудников . . . .”





# Сущность

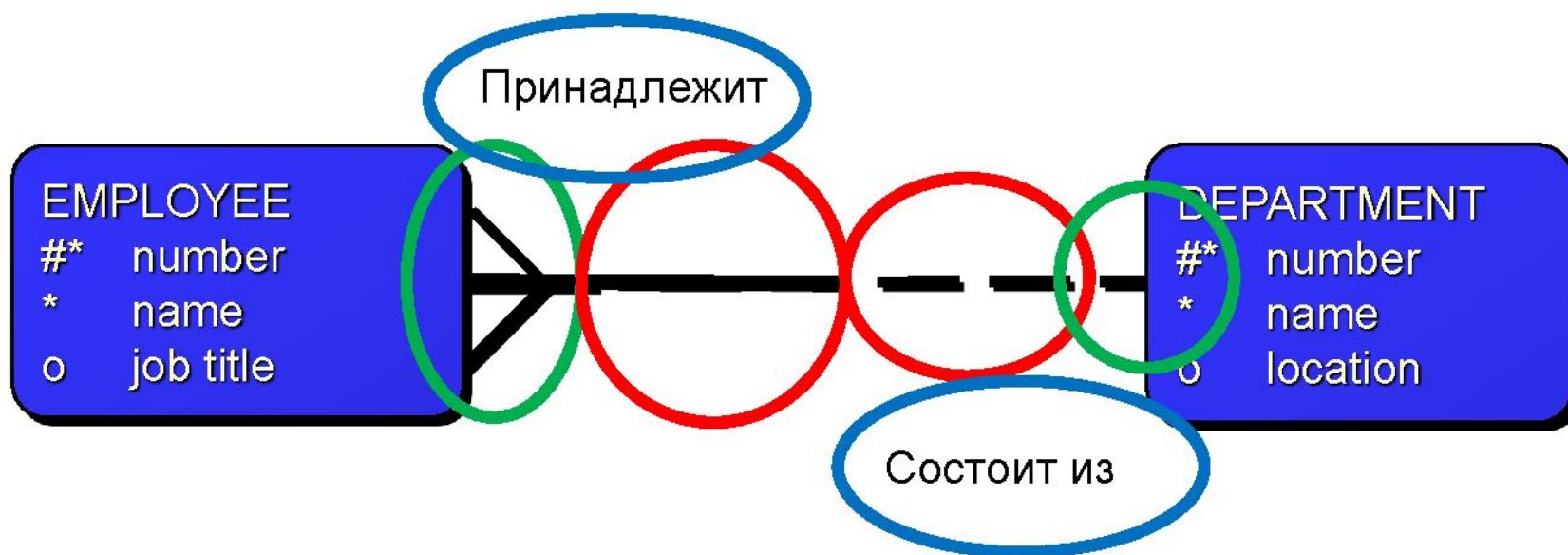
---

- ▶ Сущность – реальны или абстрактный объект информация о котором должны быть собрана или сохранена



# Соглашения в ER-моделях

---



## Типы связей

---

- ▶ Обязательные (mandatory)
  - ▶ Факультативные (необязательные, optional)
  
  - ▶ Один-к-одному
  - ▶ Один-ко-многим
  - ▶ Многие-ко-многим
  
  - ▶ Идентифицирующая
  - ▶ Неидентифицирующая
  - ▶ *Информационная*
- 





Notation	Information Engineering	Barker Notation	IDEF1X	UML
<b>Multiplicities:</b>				
- Zero or one				
- One only				
- Zero or more				
- One or more				
- Specific range	N/A	N/A	N/A	
<b>Attributes:</b>				
Names	N/A	Attribute Name: Type	attribute-name: Type	attributeName: Type
Primary key/unique identifier	N/A	# Attribute Name		attributeName <<PK>> {order=#}
Foreign key	N/A	N/A	attribute-name (FK)	attributeName <<FK>> {to=tablename}
<b>Associations:</b>				
Labels				
Entity roles	N/A	N/A	N/A	
Subtyping				
Aggregation				
Composition				
Or Constraint		N/A	N/A	
Exclusive Or (XOR) Constraint			N/A	

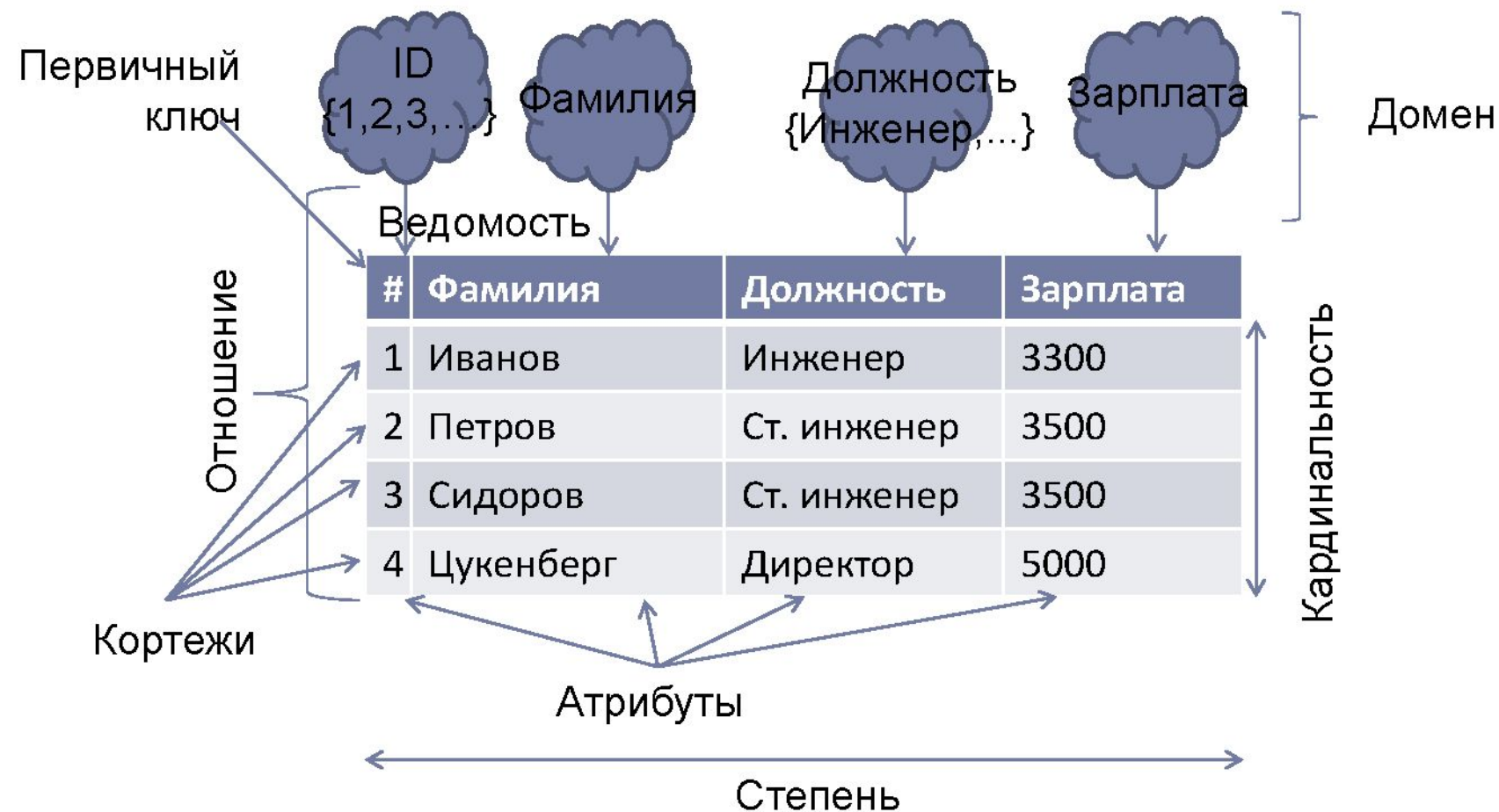
## Проверка модели

---

- ▶ Данные не должны быть противоречивыми, т.е. при выполнении операций включения, удаления и обновления данных их потенциальная противоречивость должна быть сведена к минимуму (требования непротиворечивости данных).
- ▶ Схема отношений базы данных должна быть устойчивой, способной адаптироваться к изменениям при ее расширении дополнительными атрибутами (требование гибкости структуры базы данных).



# Базовые термины





# Просто о сложном

---

<b>Формальный термин</b>	<b>Неформальный эквивалент</b>
Отношение	Таблица
Кортеж	Строка, запись
Кардинальность	Количество строк
Атрибут	Столбец, поле
Степень	Количество столбцов
Первичный ключ	Идентификатор
Домен	Область допустимых значений



# Ключи

---

- ▶ Составной ключ, простой ключ
- ▶ Потенциальный ключ
- ▶ Первичный ключ, альтернативный ключ
- ▶ Внешний ключ



# Primary key

---

- ▶ Каждое реляционное отношение имеет только 1 первичный ключ, все остальные - альтернативные
- ▶ Значение всех атрибутов первичного ключа **не** может быть **не определено**
- ▶ Значение первичного ключа не влияет на расположение кортежей в табличном представлении отношения
- ▶ Первичный ключ не влияет на доступ к атрибутам кортежа





# Внешний ключ

---

- ▶ Значение внешнего ключа всегда ссылается на первичный ключ другого отношения
- ▶ Значения может быть неопределенно и повторяются в рамках одного отношения

Недвижимость

Адрес	№Паспорта
Замостянская 5, к301	АА789
Замостянская 5, к301	ББ788
Харьковская, 30, к1	АА789
Харьковская, 30, к2	
<del>Харьковская, 30, к3</del>	<del>ДД786</del>

Владелец

№ Паспорта	ФИО
АА789	Иванов И.И.
ББ788	Петров П.П.
СС787	Сидоров С.С.

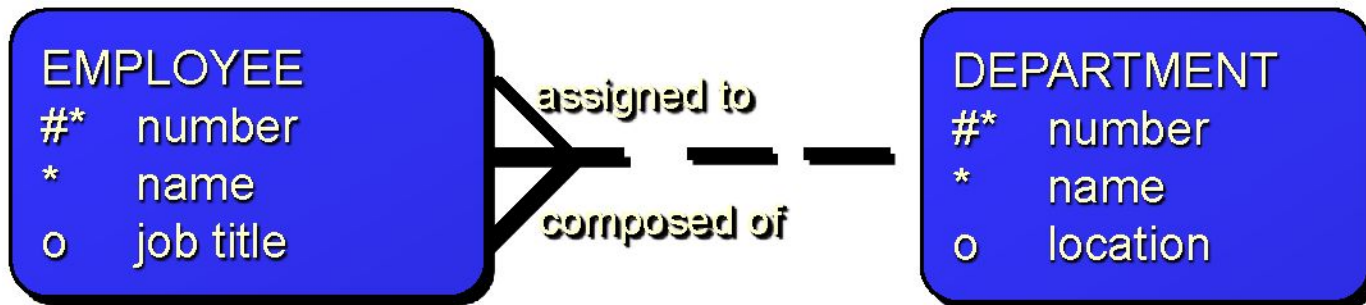
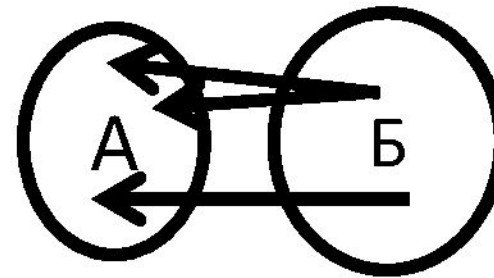
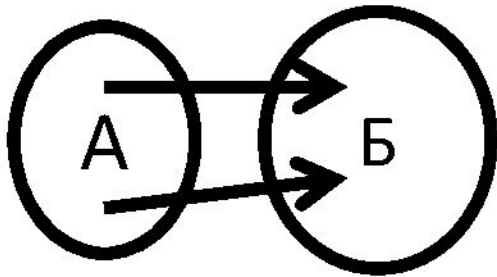


# Отношение

---

- Функциональные

- ▶ Многозначные



## Функциональная зависимость

---

- ▶ Пусть задано отношение  $R$ , которое содержит наборы атрибутов  $A$  и  $B$ . В отношении  $R$  набор атрибутов  $B$  *функционально зависит от  $A$*  и  *$A$  функционально определяет  $B$*  тогда и только тогда, когда в любой момент времени каждому значению проекции  $R[A]$  отвечает ровно одно значение  $R[B]$ .





# Свойства отношений

Властивість	Графічне позначення
1) Транзитивність: якщо $A \rightarrow B$ і $B \rightarrow C$ , то $A \rightarrow C$	
2) Проективність: якщо $B \subseteq A$ , то $A \rightarrow B$	
3) Адитивність (об'єднання): якщо $A \rightarrow B$ і $A \rightarrow C$ , то $A \rightarrow (B, C)$	
4) Рефлексивність: $A \rightarrow A$	
5) Псевдотранзитивність: якщо $A \rightarrow B$ і $(B, C) \rightarrow D$ , то $(A, C) \rightarrow D$	
6) Продовження: якщо $A \rightarrow B$ , то $(A, C) \rightarrow B$ для будь-якого атрибута $C$	
7) Поповнення: якщо $A \rightarrow B$ , то $(A, C) \rightarrow (B, C)$ для будь-якого атрибута $C$	
8) Декомпозиція: якщо $A \rightarrow B$ і $C \subseteq B$ , то $A \rightarrow C$	

# Нормализация

---

- ▶ Построение информационной модели предметной области и логической модели реляционной базы данных - результат решения комбинаторных задач:
  - ▶ группировка атрибутов в отношении предметной области;
  - ▶ распределение атрибутов по отношениям базы данных.
- ▶ Процесс устранения потенциальной противоречивости и избыточности данных в отношениях реляционной базы данных называется нормализацией исходных схем отношений.



# Первая нормальная форма

---

- ▶ Отношение находится в первой нормальной форме (1НФ), если все атрибуты отношения являются простыми (требование атомарности атрибутов в реляционной модели), т.е. не имеют компонентов.



# Первая нормальная форма

---

Пример: Приведение отношения к 1НФ

Отношение SHIPMENT (ОТГРУЗКА) является ненормализованным. Оно содержит повторяющиеся группы, представляющие массив значений, 1st Consignment, 2st Consignment, 3st Consignment (партии грузов).

Shipment
<u>Ship Registration Number</u>
<u>Departure Data</u>
Ship Name
Origin
Destination
1st Consignment
2st Consignment
3st Consignment
Customs Value
Ship Capacity
Customs Declaration

Для такого отношения следует ввести бизнес-правило, требующее, чтобы груз состоял не более чем из трех партий!



# Первая нормальная форма

---

Использование отношения, представленного не в 1НФ, может породить следующие проблемы:

- ▶ если груз аннулируется и строка, связанная с грузом, удаляется из отношения, то вместе с ней удаляются все сведения о партиях груза на борту судна;
- ▶ если на склад прибывает новая партия груза, и она еще не включена в состав груза, подлежащего отправке, то сведения о партии заносить некуда;
- ▶ необходимо вводить ограничение: в грузе не может быть более трех партий.

Shipment
<u>Ship Registration Number</u>
<u>Departure Data</u>
Ship Name
Origin
Destination
1st Consignment
2st Consignment
3st Consignment
Customs Value
Ship Capacity
Customs Declaration



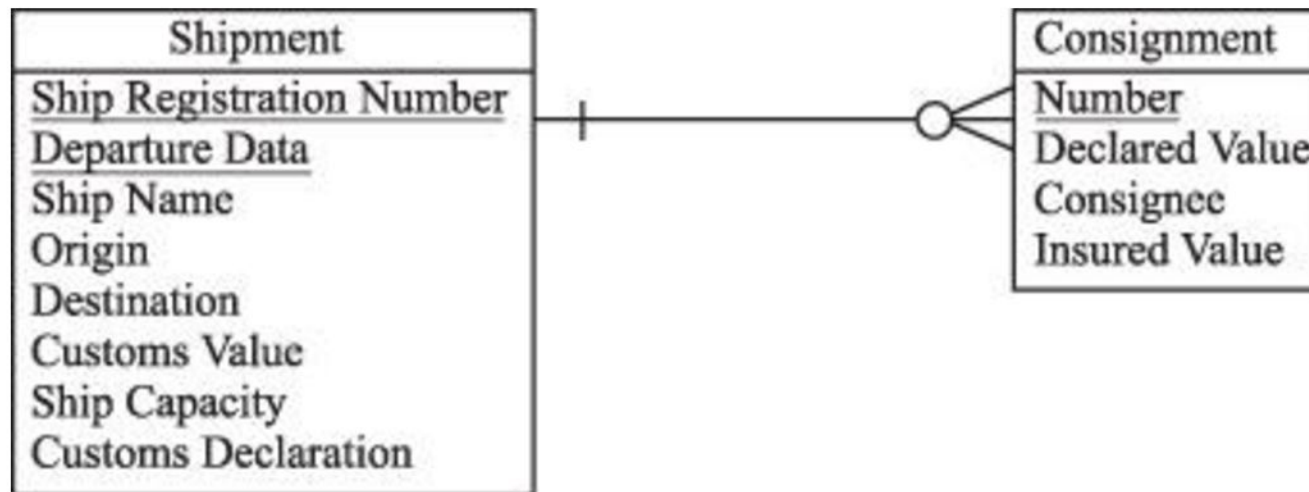


# Первая нормальная форма

---

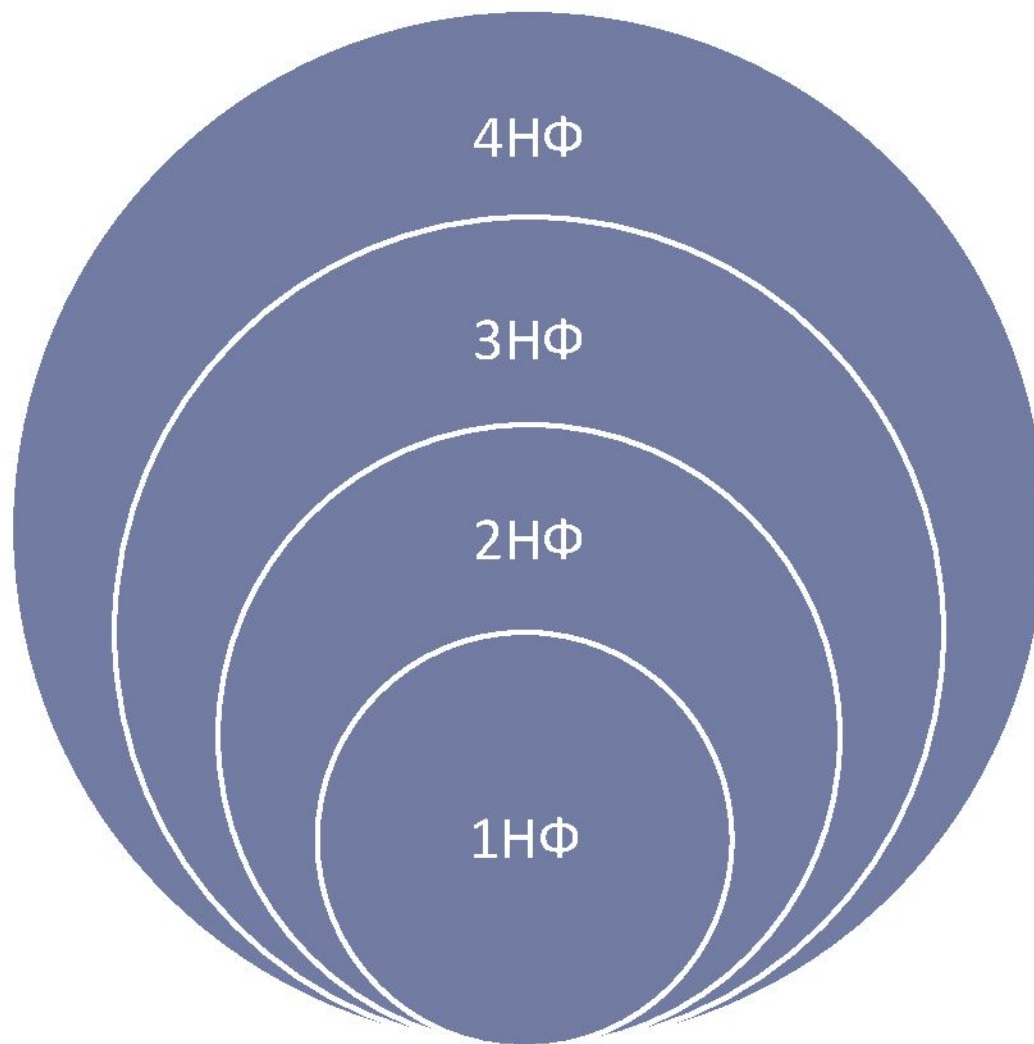
Приведение отношения SHIPMENT к 1НФ заключается в изъятии данных о партиях груза из отношения SHIPMENT и создании для них связанного подчиненного отношения CONSIGNMENT (ПАРТИЯ\_ГРУЗА).

Результат приведения отношения SHIPMENT к 1НФ :



# Вложенность нормальных форм

---



## 2-я Нормальная форма

---

- ▶ **Атрибут отношения считается ключевым если он является элементом какого-либо ключа отношения.**
- ▶ В противном случае атрибут будет считаться *неключевым атрибутом*.
- ▶ Так в отношении (Город, Адрес, Почтовый\_индекс) все атрибуты являются ключевыми, поскольку при заданных ФЗ (*город, адрес*) → *почтовый\_индекс* и *почтовый\_индекс* → *город* ключами являются пары (*город, адрес*) и (*адрес, почтовый\_индекс*).
- ▶ **Отношение находится во второй нормальной форме (2НФ), если оно находится в 1НФ, и все неключевые атрибуты отношения функционально полно зависят от составного ключа отношения.**



## 2-я Нормальная форма

---

Пример: Приведение отношения ко 2НФ.

Отношение SHIPMENT содержит частичную ФЗ: неключевой атрибут Ship Capacity (грузоподъемность корабля) не зависит от ключевого атрибута Departure Date (даты убытия), а зависит от ключевого атрибута Ship Registration Number (регистрационный номер корабля).

Shipment
<u>Ship Registration Number</u>
<u>Departure Date</u>
Ship Name
Origin
Destination
1st Consignment
2st Consignment
3st Consignment
Customs Value
Ship Capacity
Customs Declaration



## 2-я нормальная форма

---

- ▶ Использование отношения, представленного не во 2НФ, может породить следующие проблемы:
  - ▶ невозможно занести в базу данных название и грузоподъемность корабля, который не доставил еще ни одного груза, - можно только ввести для него фиктивный груз;
  - ▶ если удалить кортеж из отношения Shipment после отправки груза, то потеряются все данные о кораблях, для которых в настоящее время нет груза;
  - ▶ невозможно отразить факт переоборудования корабля и получения им новой грузоподъемности; если переписать все предыдущие кортежи об этом корабле, то получится, что он в прошлом плавал недогруженным или перегруженным.

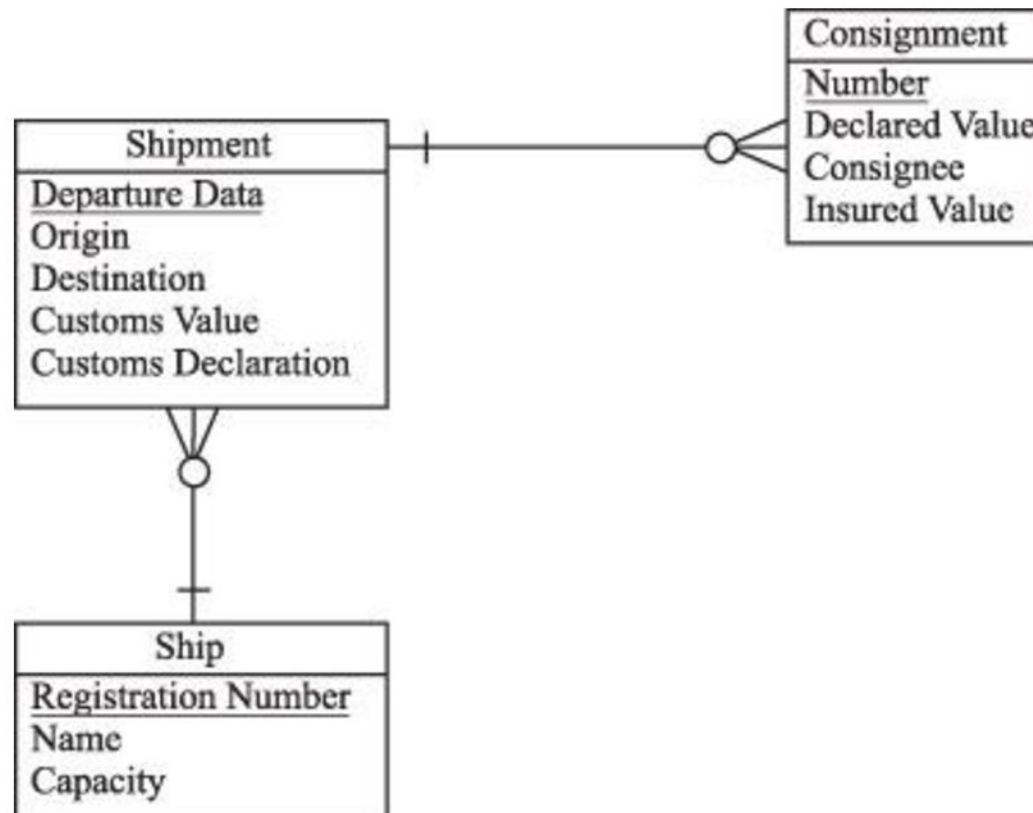




# 2-я нормальная форма

---

Отношения SHIPMENT и создании для нее связанного подчиненного отношения SHIP.



## 3-я Нормальная форма

---

- Отношение находится в третьей нормальной форме (3НФ), если оно находится во 2НФ, и все неключевые атрибуты отношения зависят только от первичного ключа.



## 3-я Нормальная Форма

---

Возможна следующая проблема:

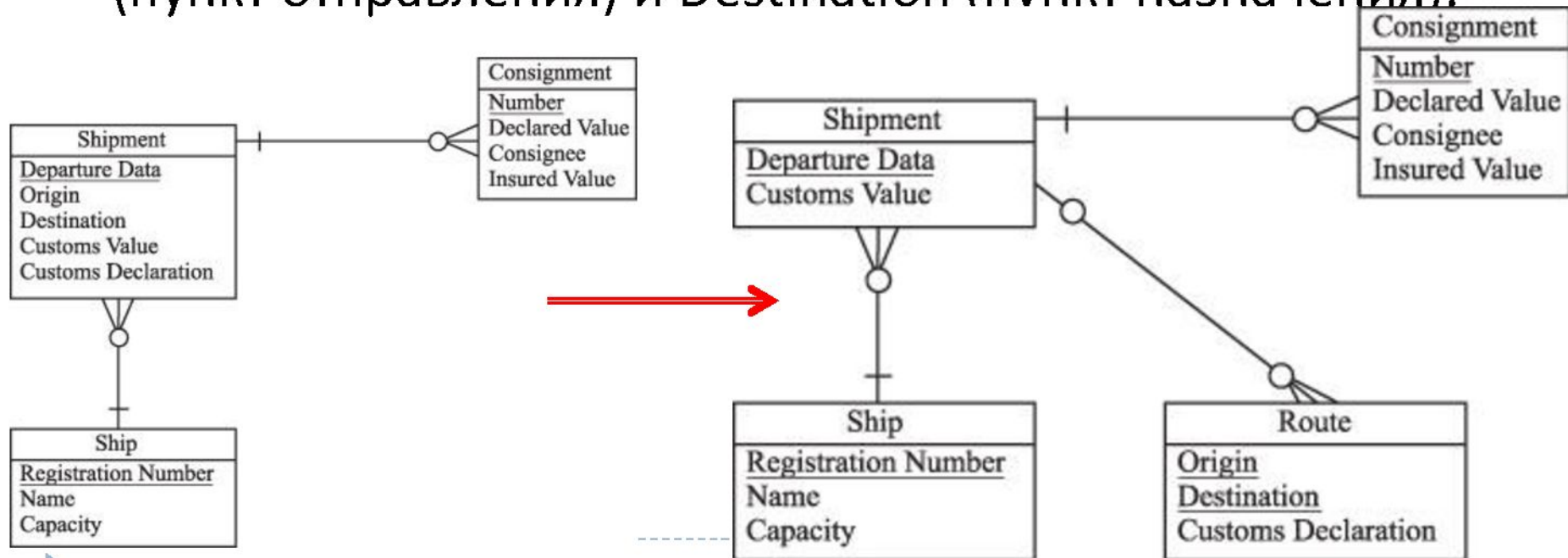
- ▶ Наличие транзитивной зависимости не позволяет связать значения  $Y$  и  $X$ , если не существует значения  $A$ , связанного со значением  $Y$ . Это затрудняет вставку и обновление данных, которые необходимо выполнить сразу для пары связей, а в случае удаления данных приводит к потере связи.



# 3-я Нормальная форма

## Пример: Приведение отношения к 3НФ

Отношение SHIPMENT содержит транзитивную ФЗ:  
атрибут Customs Declaration (таможенная декларация)  
является по своей сути свойством атрибутов Origin  
(пункт отправления) и Destination (пункт назначения).



# Нормальная форма Бойса-Кодда

---

ЗНФ упрощает решение проблем контроля избыточности данных, интерпретации нуль-значений, контроля за операциями модификации данных, только если в отношениях отсутствуют какие-либо другие ФЗ, в частности *обратные ФЗ неключевого атрибута на один из атрибутов составного первичного ключа или многозначные ФЗ*.

В противном случае вышеперечисленные проблемы остаются неразрешенными. Для устранения таких проблем, связанных с существованием обратных ФЗ неключевых атрибутов на часть составного ключа, была предложена усиленная ЗНФ или *НФ Бойса-Кодда*.





# Нормальная форма Бойса-Кодда

---

***Отношение находится в нормальной форме Бойса-Кодда (НФБК), если оно находится в ЗНФ, и в нем отсутствуют зависимости ключевых атрибутов от неключевых атрибутов.***

Иными словами, НФБК допускает наличие только таких ФЗ, в которых ключ определяет один или более других атрибутов



# Нормальная форма Бойса-Кодда

---

Схема отношения в НФБК обладает теми же достоинствами, что и схема в 3НФ, но устраняет некоторые дополнительные аномалии, не устраняемые 3НФ.

Пример: в отношении (Город, Адрес, Почтовый\_индекс), находящееся в 3НФ, невозможно записать кортеж для города с известным почтовым индексом, если не известен адрес с этим почтовым индексом.

Данное отношение не находится в НФБК, так как имеет место ФЗ *Почтовый\_индекс* → *Город*, а атрибут почтовый\_индекс не является ключом этого отношения.

---



# Нормальная форма Бойса-Кодда

Отношение КОМАНДА		
Член экипажа	команда	Руководитель
Иванов	Наблюдение	Прохоров
Иванов	Питание	Макаров
Петров	Наблюдение	Леонтьев
Модин	Наблюдение	Прохоров
Васин	Питание	Лазарев
Фролов	Обслуживание	Сидоров
Ивлев	Обслуживание	Сидоров

Отношение РУКОВОДИТЕЛЬ_КОМАНДЫ	
Команда	Руководитель
Наблюдение	Прохоров
Питание	Макаров
Наблюдение	Леонтьев
Питание	Лазарев
Обслуживание	Сидоров



# Нормальная форма Бойса-Кодда

---

Результат приведения отношения КОМАНДА к НФБК:



***Следует обратить внимание на возникновение избыточности данных***

---



# Четвертая нормальная форма

---

- ▶ **Отношение находится в четвертой нормальной форме (4НФ), если оно находится в 3НФ или НФБК и все независимые многозначные ФЗ разнесены в отдельные отношения с одним и тем же ключом.**
- ▶ Иными словами, 4НФ применяется при наличии в отношении более чем одной многозначной ФЗ и требует, чтобы отношение не содержало независимых многозначных ФЗ.





# Четвертая нормальная форма

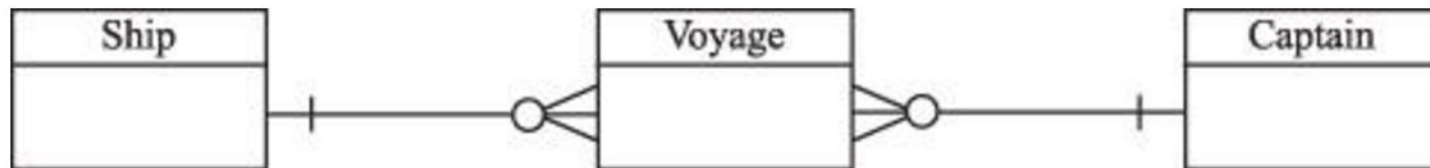
---

- ▶ Пример: Приведение к 4НФ
- ▶ Рассмотрим отношение, содержащее сведения о кораблях (Ship), совершаемых ими рейсах (Voyage) и капитанах (Captain)

Отношение КАПИТАН_КОРАБЛЬ_РЕЙС		
Акбар	Иванов	Санкт-Петербург - Калининград
Акбар	Петров	Санкт-Петербург - Калининград
Акбар	Ивлев	Санкт-Петербург - Калининград
Акбар	Прохоров	Санкт-Петербург - Калининград
Акбар	Лазарев	Санкт-Петербург- Лондон
Акбар	Прохоров	Санкт-Петербург- Лондон
Жучка	Петров	Санкт-Петербург - Марсель
Жучка	Фролов	Санкт-Петербург - Стокгольм
Жучка	Ивлев	Санкт-Петербург - Стокгольм

# Четвертая нормальная форма

---



Отношение находится в НФБК и содержит только многозначные ФЗ.

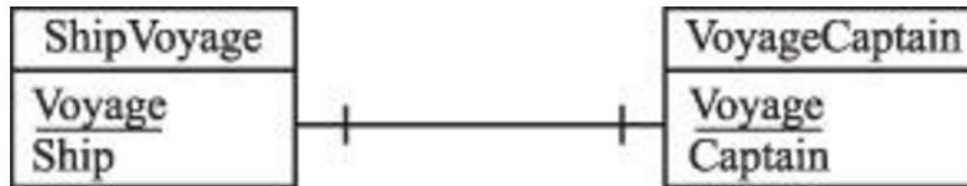
Однако имеет место **аномалия удаления**: если капитан Петров уйдет в отставку и все кортежи о нем будут удалены, то будут потеряны сведения о том, что корабль Жучка совершает рейсы Санкт-Петербург - Марсель. Если добавить новый рейс, то, возможно, придется ввести несколько кортежей в наше отношение.



# Четвертая нормальная форма

---

Приведение отношения к 4НФ заключается в выделении для каждой многозначной ФЗ своего отношения:



# Выводы (свойства нормальных форм)

---

- 1НФ - все атрибуты отношения простые;
- 2НФ - отношение находится в 1НФ и не содержит частичных ФЗ;
- 3НФ - отношение находится во 2НФ и не содержит транзитивных ФЗ от ключа;
- НФБК - отношение находится в 3НФ и не содержит ФЗ ключей от неключевых атрибутов;
- 4НФ, применяется при наличии более чем одной многозначной ФЗ - отношение находится в НФБК или 3НФ и не содержит независимых многозначных ФЗ;
- (Домашнее чтение) 5НФ - отношение находится в 4НФ и не содержит ФЗ по соединению.



# Алгоритм нормализации

---

Предусловие:

- ▶ Все сущности должны быть уникально идентифицированы комбинацией атрибутов и/или связей

1НФ:

- ▶ Необходимо удалить все атрибуты, которые повторяются

2НФ:

- ▶ Необходимо удалить все атрибуты зависящие от части составного идентификатора

3НФ:

- ▶ Если атрибут зависит от другого неключевого атрибута, их нужно преобразовать в новую сущность
- 



## ERB в структуру таблиц

---

1. Каждая простая сущность становится таблицей.
2. Каждый атрибут становится столбцом таблицы. Обязательным атрибутам устанавливают ограничение `not null`.
3. Идентификаторы сущностей становятся ключами. Один из идентификаторов становится первичным ключом.
4. Связи «один-к-одному» и «один-ко-многим» преобразуются во внешние ключи.





## Домашнее чтение

---

- ▶ Контрольный список вопросов к сущностям для ER-диаграмм
- ▶ Контрольный список вопросов к атрибутам для ER-диаграмм
- ▶ «Аномалии»
- ▶ 5 Нормальная форма
- ▶ Таблица обозначения ключевых элементов в разных нотациях (слайд 6)
- ▶ Статья «ER: диаграммы сущность – связь» – о различных подходах к моделированию систем



# Контрольный список вопросов к сущностям

---

- ▶ Отражает ли имя сущности суть данного объекта?
- ▶ Нет ли пересечения с другими сущностями?
- ▶ Имеются ли хотя бы два атрибута?
- ▶ Всего атрибутов не более восьми?
- ▶ Есть ли синонимы/омонимы данной сущности?
- ▶ Сущность определена полностью?
- ▶ Есть ли уникальный идентификатор?
- ▶ Имеется ли хотя бы одна связь?
- ▶ Существует ли хотя бы одна функция по созданию, поиску, корректировке, удалению, архивированию и использованию значения сущности?
- ▶ Ведется ли история изменений?
- ▶ Имеет ли место соответствие принципам нормализации данных?
- ▶ Нет ли такой же сущности в другой прикладной системе, возможно, под другим именем?
- ▶ Не имеет ли сущность слишком общий смысл?
- ▶ Достаточен ли уровень обобщения, воплощенный в ней?



# Контрольный список вопросов к атрибутам

---

- ▶ Является ли наименование атрибута существительным единственного числа, отражающим суть обозначаемого атрибутом свойства?
- ▶ Не включает ли в себя наименование атрибута имя сущности (этого быть не должно)?
- ▶ Имеет ли атрибут только одно значение в каждый момент времени?
- ▶ Отсутствуют ли повторяющиеся значения (или группы)?
- ▶ Описаны ли формат, длина, допустимые значения, алгоритм получения и т.п.?
- ▶ Не может ли этот атрибут быть пропущенной сущностью, которая пригодилась бы для другой прикладной системы (уже существующей или предполагаемой)?
- ▶ Не может ли он быть пропущенной связью?
- ▶ Нет ли где-нибудь ссылки на атрибут как на "особенность проекта", которая при переходе на прикладной уровень должна исчезнуть?
- ▶ Есть ли необходимость в истории изменений?
- ▶ Зависит ли его значение только от данной сущности?
- ▶ Если значение атрибута является обязательным, всегда ли оно известно?
- ▶ Есть ли необходимость в создании домена для этого и ему подобных атрибутов?
- ▶ Зависит ли его значение только от какой-то части уникального идентификатора?
- ▶ Зависит ли его значение от значений некоторых атрибутов, не включенных в уникальный идентификатор?



## Аномалии вставки (INSERT)

---

- ▶ Пример: Отношение **сотрудник\_проект** нельзя вставить данные о проекте, над которым пока не работает ни один сотрудник.
- ▶ Причина аномалии - хранение в одном отношении разнородной информации (и о сотрудниках, и о проектах, и о работах по проекту).
- ▶ Вывод - логическая модель данных неадекватна модели предметной области. База данных, основанная на такой модели, будет работать неправильно.





# Аномалии обновления (UPDATE)

---

- ▶ Фамилии сотрудников, наименования проектов, номера телефонов повторяются во многих кортежах отношения. Поэтому если сотрудник меняет фамилию, или проект меняет наименование, или меняется номер телефона, то такие изменения необходимо *одновременно* выполнить во всех местах, где эта фамилия, наименование или номер телефона встречаются, иначе отношение станет некорректным (например, один и тот же проект в разных кортежах будет называться по-разному). Таким образом, обновление базы данных одним действием реализовать невозможно. Для поддержания отношения в целостном состоянии необходимо написать триггер, который при обновлении одной записи корректно исправлял бы данные и в других местах.
- ▶ Причина аномалии - избыточность данных, также порожденная тем, что в одном отношении хранится разнородная информация.
- ▶ Вывод - увеличивается сложность разработки базы данных. База данных, основанная на такой модели, будет работать правильно только при наличии дополнительного программного кода в виде триггеров.



## Аномалии удаления (DELETE)

---

- ▶ При удалении некоторых данных может произойти потеря другой информации. Например, если закрыть проект "Космос" и удалить все строки, в которых он встречается, то будут потеряны все данные о сотруднике Петрове. Если удалить сотрудника Сидорова, то будет потеряна информация о том, что в отделе номер 2 находится телефон 33-22-11. Если по проекту временно прекращены работы, то при удалении данных о работах по этому проекту будут удалены и данные о самом проекте (наименование проекта). При этом если был сотрудник, который работал только над этим проектом, то будут потеряны и данные об этом сотруднике.
  - ▶ Причина аномалии - хранение в одном отношении разнородной информации (и о сотрудниках, и о проектах, и о работах по проекту).
  - ▶ Вывод - логическая модель данных неадекватна модели предметной области. База данных, основанная на такой модели, будет работать неправильно.
- 





# Пятая нормальная форма

---

- ▶ Нормализация отношений выполнялась путем разложения (декомпозиции) схем отношений.
- ▶ Очевидно, что при таком подходе должен соблюдаться принцип обратимости: соединение проекций должно приводить к исходным отношениям.
- ▶ Это предполагает отсутствие потери кортежей; появление ранее не существовавших кортежей; сохранение ФЗ (семантика взаимосвязей между данными не должна нарушаться).



# Пятая нормальная форма

---

- ▶ Декомпозиция схем отношений не всегда гарантирует обратимость. Это обстоятельство связано с существованием класса ФЗ по соединению.
- ▶ *Если отношение удовлетворяет ФЗ по соединению, то оно может быть восстановлено по своим проекциям.*
- ▶ Отношения, содержащие более трех МФЗ, требуют особого внимания при построении *логической модели реляционной базы данных*. Также 4НФ не устраняет избыточность данных полностью, поэтому требуется дальнейшая декомпозиция схем отношений.



# Пятая нормальная форма

---

- ▶ *Отношение находится в пятой нормальной форме (5НФ), если оно находится в 4НФ и удовлетворяет зависимости по соединению относительно своих проекций.*
- ▶ 5НФ называют также нормальной формой с проецированием соединений. Она используется для разрешения трех и более отношений, которые связаны более чем тремя ФЗ по типу «многие-ко-многим»

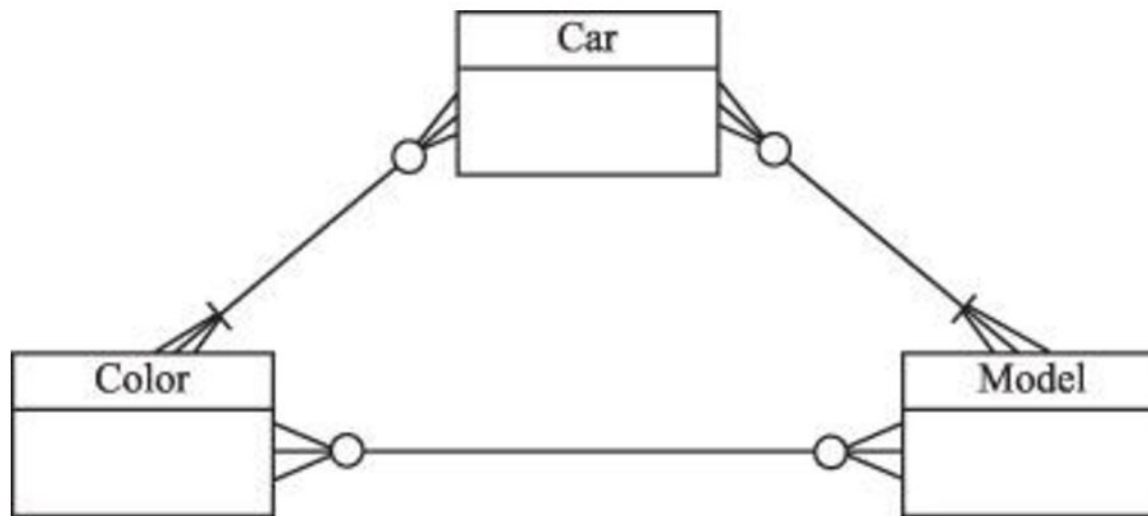


# Пятая нормальная форма

---

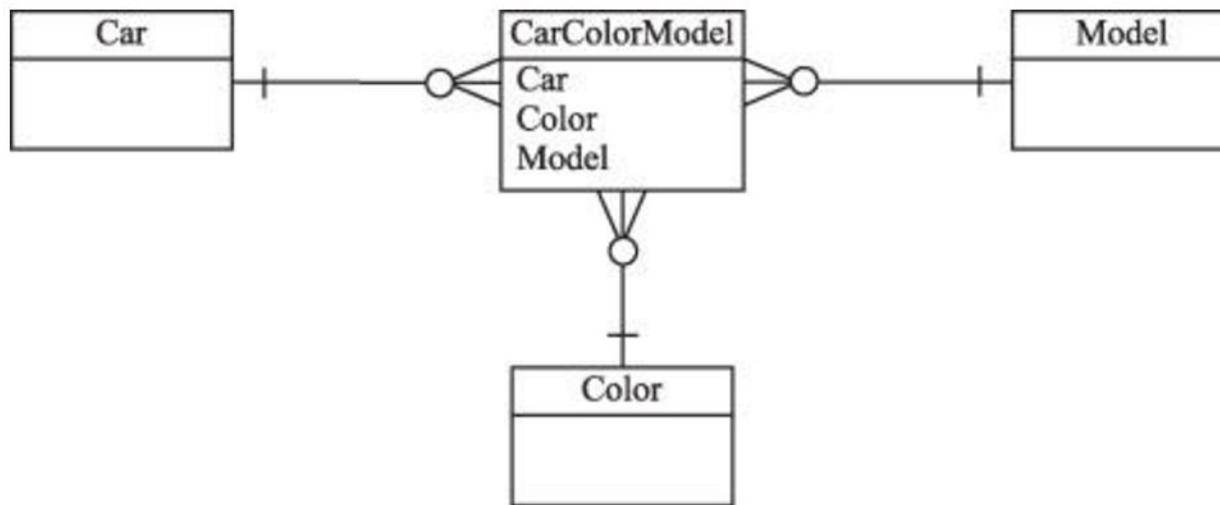
Пример: Приведение к 5НФ

Рассмотрим отношение с несколькими многозначными зависимостями:



# Пятая нормальная форма

Приведение отношения к 5НФ заключается во введении еще одного отношения, связывающего три исходных отношения:



Процедура приведения отношения, содержащего многозначные ФЗ, к 5НФ состоит в построении связывающего отношения, позволяющего исключить появление в соединениях ложных кортежей.

