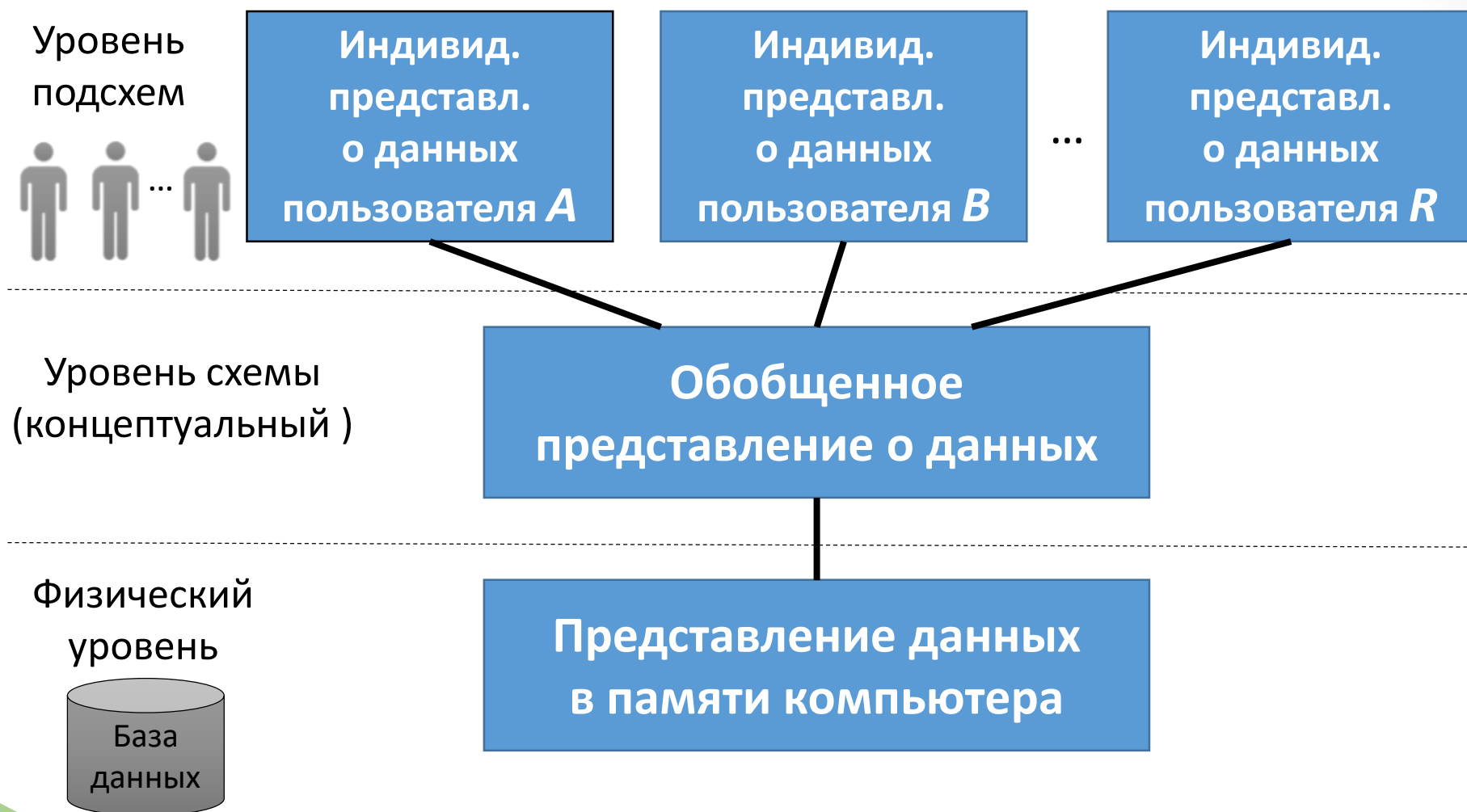


# Проектирование и нормализация баз данных

Технологии баз данных. Лекция 10

# Архитектура систем баз данных ANSI/SPARC



# Проектирование базы данных

## Архитектура ANSI/SPARC

## Этапы проектирования БД

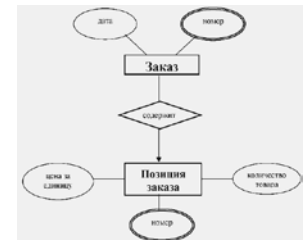
## Представление

Уровень подсхем



Концептуальное проектирование

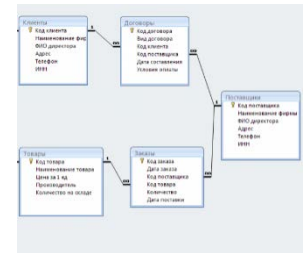
Модель предметной области  
(Н.: ER-диаграмма)



Уровень схемы  
(концептуальный)

Логическое проектирование

Модель данных  
(Н.: реляционная схема БД)



Физический уровень



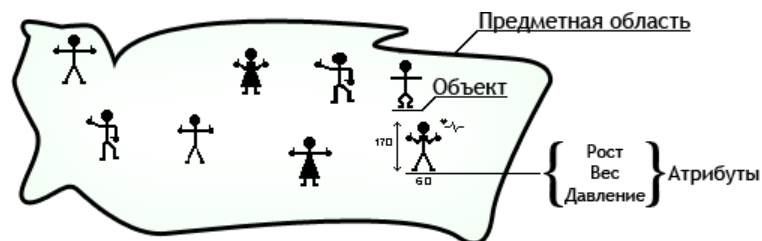
Физическое проектирование

Объекты СУБД  
(Н.: реализация БД в СУБД PostgreSQL)



# Множественность вариантов представления каждого уровня

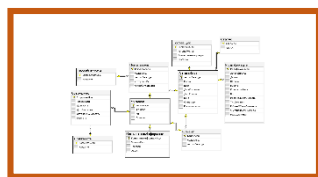
Предметная область:



Альтернативные ER-диаграммы:



Альтернативные реляционные схемы:



Альтернативные СУБД:

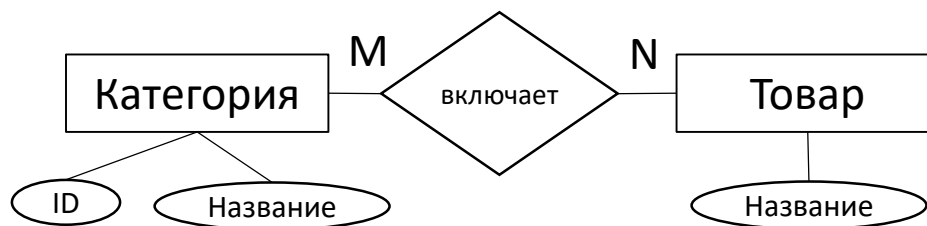


# Логическое проектирование БД

**Важно ли то, как именно мы «разбиваем» ER-диаграмму на таблицы?**

Пример:

Каталог товаров



CatID	CatName	Product1	Product2	Product3
1	бытовая химия	порошок	мыло	сода
2	продукты	сода	-	-
3	спорт. инвентарь	лыжи	палки	-

Проблемы:

1. Данные о товаре повторяются (избыточны).
2. Как добавить еще один товар в каталог «бытовая химия»?
3. Чтобы удалить товар «сода» или изменить на «сода пищевая», нужно просмотреть всю таблицу.

# Аномалии

- **Аномалией** называется такая ситуация в таблице БД, которая приводит к противоречию в БД либо существенно усложняет обработку БД.
- *Причиной аномалии является излишнее дублирование данных в таблице.*
  - 1) **Аномалии обновления** проявляются в том, что изменение одних данных может повлечь просмотр всей таблицы и соответствующее изменение некоторых записей таблицы.

CatID	CatName	Product1	Product2	Product3
1	бытовая химия	порошок	мыло	сода
2	продукты	сода	-	-
3	спорт. инвентарь	лыжи	палки	-

Пример: Чтобы изменить товар «сода» на «сода пищевая», нужно просмотреть всю таблицу.

# Аномалии

- **Аномалией** называется такая ситуация в таблице БД, которая приводит к противоречию в БД либо существенно усложняет обработку БД.
- *Причиной аномалии является излишнее дублирование данных в таблице.*
  - 2) **Аномалии удаления** — при удалении какого либо кортежа из таблицы может пропасть информация, которая не связана напрямую с удаляемой записью.

CatID	CatName	Product1	Country1	Product2	Country2
1	бытовая химия	порошок	Голландия	сода	Россия
2	продукты	сода	Германия	-	-
3	спорт. инвентарь	лыжи	Финляндия	-	-

Пример: Если мы будем удалять товар «сода» только по названию, то получим аномалию удаления – удаляться два разных товара, что приведет к потере целостности БД.

# Аномалии

- **Аномалией** называется такая ситуация в таблице БД, которая приводит к противоречию в БД либо существенно усложняет обработку БД.
  - *Причиной аномалии является излишнее дублирование данных в таблице.*
- 3) **Аномалии добавления** возникают, когда информацию в таблицу нельзя поместить, пока она не полная, либо вставка записи требует дополнительного просмотра таблицы.

CatID	CatName	Product1	Country1	Product2	Country2
1	бытовая химия	порошок	Голландия	сода	Россия
2	продукты	сода	Германия	-	-
3	спорт. инвентарь	лыжи	Финляндия	-	-

Пример: Мы не можем поместить в БД продукт, который еще не отнесен ни к одной категории.



# Функциональная зависимость

- Пусть  $R$  – переменная отношения,  $X, Y$  – атрибуты  $R$ . Тогда
  - $Y$  *функционально зависит* от  $X$ , или
  - $X$  *функционально определяет*  $Y$ , или
  - $X$  является *детерминантом*, а  $Y$  является *зависимой* частью
  - $X \rightarrow Y$

если не существует двух различных кортежей, которые в атрибуте  $X$  имеют совпадающие значения, а в атрибуте  $Y$  имеют разные значения.

$X \rightarrow Y$

КодП	ИмяП	Город	Рейтинг
S1	Бендер	Черноморск	15
S2	ИТ-монстр	Москва	29
S3	Разноэкспорт	Черноморск	15
S4	Макрохард	Москва	29
...	...	...	...

$\{\text{Город}\} \rightarrow \{\text{Рейтинг}\}$

«Рейтинг»  
функционально  
зависит от «Город»

# Функциональная зависимость

КодП	ИмяП	Город	Рейтинг
S1	Бендер	Черноморск	15
S2	ИТ-монстр	Москва	29
S3	Разноэкспорт	Черноморск	15
S4	Макрохард	Москва	29
...	...	...	...

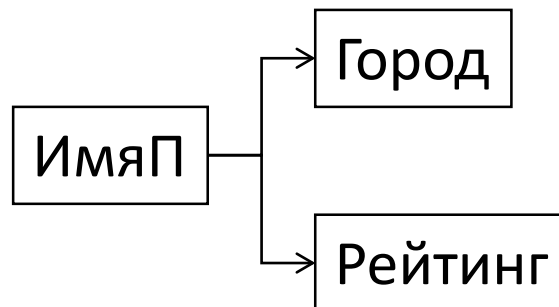
{Город}→{Рейтинг}

- ФЗ подразумевает **все возможные кортежи отношения**, а не только те, которые находятся в отношении в настоящий момент.
- ФЗ **определяется только семантикой предметной области** и не может быть выведена из текущего состояния базы данных.

# Диаграммы ФЗ

КодП	ИмяД	ИмяП	Город	Рейтинг
S1	Болт	Бендер	Черноморск	15
S2	Гайка	Макрохард	Москва	29
S3	Винт	Бендер	Черноморск	15
S4	Шуруп	Макрохард	Москва	29
...		...	...	...

Диаграмма ФЗ {ИмяП}→{Город, Рейтинг}:



# Функциональная зависимость от первичного ключа

- Все атрибуты кортежа в совокупности находятся в функциональной зависимости от первичного ключа по определению первичного ключа.

КодП	ИмяП	Город	Рейтинг
S1	Бендер	Черноморск	15
S2	ИТ-монстр	Москва	29
S3	Разноэкспорт	Черноморск	15
S4	Макрохард	Москва	29
...	...	...	...

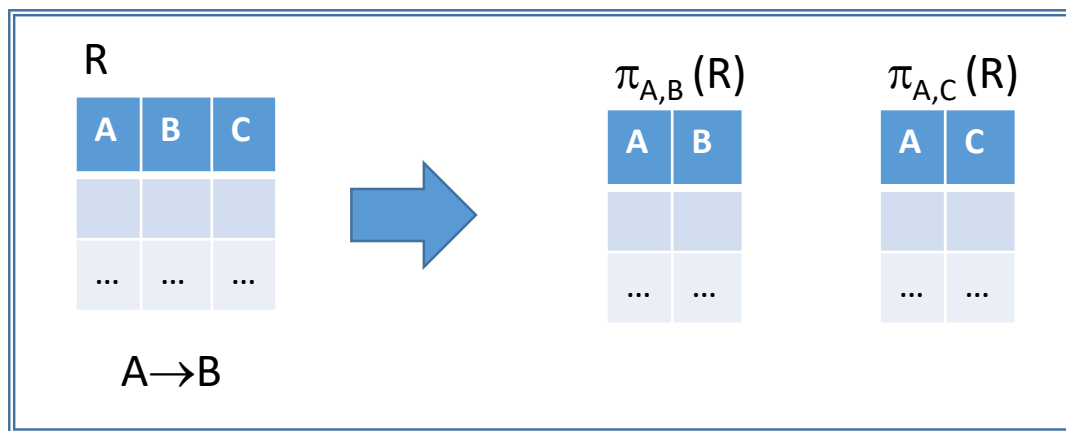
{КодП}→{ИмяП, Город, Рейтинг}

# Количество ФЗ

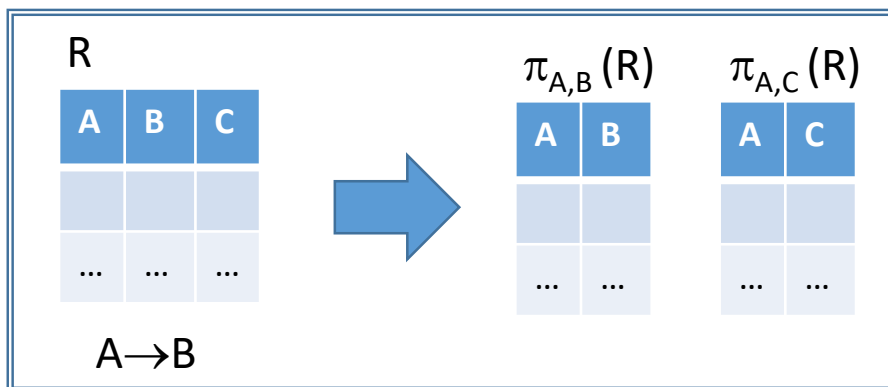
- ФЗ – ограничение целостности, за соблюдением которого должна следить СУБД.
- Меньше ФЗ – легче работа СУБД.
- Как сократить количество ФЗ?

# Теорема Хита (Heath)

- Пусть  $R$  – отношение с заголовком  $A, B, C$  (атрибуты, возможно, составные) и выполняется ФЗ  $A \rightarrow B$ . Тогда  $R = \pi_{A,B}(R) \bowtie \pi_{A,C}(R)$ .
- Теорема Хита дает возможность выполнить декомпозицию отношений без потерь информации:
  - исходное отношение с ФЗ преобразуется в другие отношения, в каждом из которых атрибуты минимально зависят от первичного ключа.
  - атрибут  $B$  минимально зависит от атрибута  $A$ , если выполняется минимальная слева ФЗ  $A \rightarrow B$ .



# Теорема Хита (Heath). Пример



R	C		A	B
	КодП	ИмяП	Город	Рейтинг
S1	Бендер	Черноморск	15	
S2	ИТ-монстр	Москва	29	
S3	Разноэкспорт	Черноморск	15	
S4	Макрохард	Москва	29	
...	...	...	...	...

{Город} → {Рейтинг}



$\pi_{\text{Город, Рейтинг}}(R)$

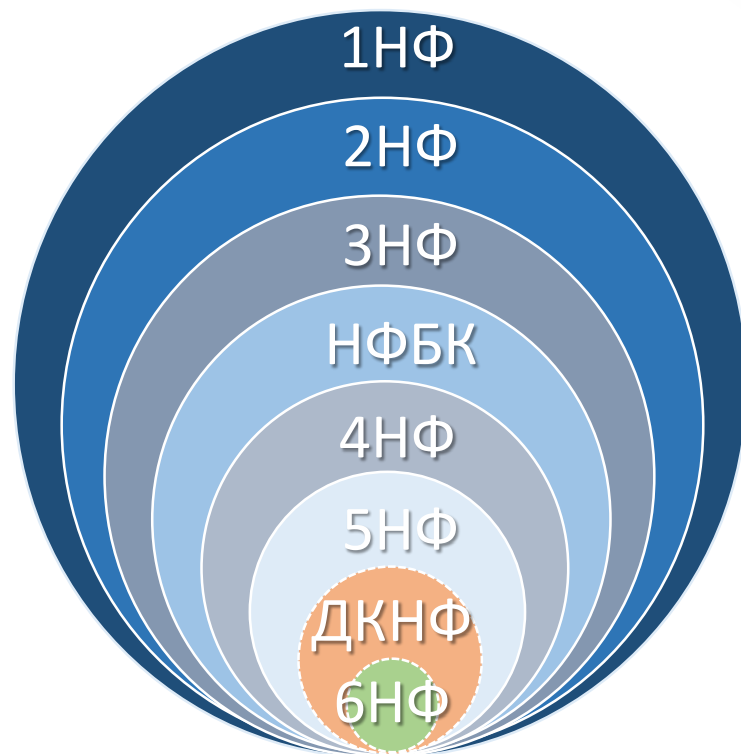
Город	Рейтинг
Черноморск	15
Москва	29
...	...

$\pi_{\text{КодП, ИмяП, Город}}(R)$

КодП	ИмяП	Город
S1	Бендер	Черноморск
S2	ИТ-монстр	Москва
S3	Разноэкспорт	Черноморск
S4	Макрохард	Москва
...	...	...

# Нормальные формы

- **Цель:** исключить избыточное дублирование данных, которое является причиной аномалий обновления, удаления и добавления.
- **Нормальная форма** – совокупность ограничений, накладываемых на схему отношения. Ограничения исключают аномалии при работе с базой данных.
- **Нормализация** состоит в получении из исходного набора отношений *нормализованных отношений*, каждое из которых находится в НФ как можно более высокого порядка.





# Определение 1НФ

**Формальное определение:** Переменная отношения R находится в первой нормальной форме (1НФ) тогда и только тогда, когда в любом допустимом значении отношения каждый его кортеж содержит только одно значение для каждого из атрибутов.

**Неформальное определение:**

- 1) Все строки должны быть различными.
- 2) Все элементы внутри ячеек должны быть атомарными (не списками). Другими словами, элемент является атомарным, если его нельзя разделить на части, которые могут использоваться в таблице независимо друг от друга.

**Замечание:**

Любое реляционное отношение находится в 1НФ (по определению реляционного отношения).

# Приведение к 1НФ

- Любое реляционное отношение находится в 1НФ (по определению реляционного отношения).

NON\_1NF\_S\_P\_SP

Приведение к 1НФ

Код_П	Имя_П	Город_П	Рейтинг	Поставка					
S1	Бендер	Черноморск	15	Код_Д	Имя_Д	Город_Д	Вес	Цена	Колич.
				P1	Болт	Париж	15	40	1000
				P5	Шуруп	Москва	29	33	1240
S3	ИТ-монстр	Москва	29	Код_Д	Имя_Д	Город_Д	Вес	Цена	Колич.
				P5	Шуруп	Москва	29	33	7600
				P4	Гвоздь	Москва	10	42	2700

BAD\_S\_P\_SP

Код_П	Имя_П	Город_П	Рейтинг	Код_Д	Имя_Д	Город_Д	Вес	Цена	Колич.
S1	Бендер	Черноморск	15	P1	Болт	Париж	15	40	1000
S3	ИТ-монстр	Москва	29	P5	Шуруп	Москва	29	33	7600
S1	Бендер	Черноморск	15	P2	Гайка	Челябинск	20	24	1240
S3	ИТ-монстр	Москва	29	P4	Гвоздь	Москва	10	42	2700

# Пример отношения не в 1НФ

## Поставки

Код_П	Имя_П	Город_П	Рейтинг	Поставка
S1	Бендер	Черноморск	15	P1, P2, P13
S3	ИТ-монстр	Москва	29	P5, P1

элементы внутри ячеек НЕ атомарные!

# Пример отношения не в 1НФ

## Поставки

Код_П	Имя_П	Город_П	Рейтинг	Поставка
S1	Бендер	Черноморск	15	P1, P2, P13
S3	ИТ-монстр	Москва	29	P5, P1



Приведение к 1НФ

## Поставки

Код_П	Имя_П	Город_П	Рейтинг	Поставка
S1	Бендер	Черноморск	15	P1
S1	Бендер	Черноморск	15	P2
S1	Бендер	Черноморск	15	P13
S3	ИТ-монстр	Москва	29	P5
S3	ИТ-монстр	Москва	29	P1

# Полная функциональная зависимость

- Определение: Функциональная зависимость  $X \rightarrow Y$  называется **полной**, если атрибут  $Y$  не зависит функционально от любого точного подмножества  $X$ .
- Пример не полной зависимости:

## Поставки

КодПоставщика*	Деталь*	Цена
S1	Болт	15
S1	Гайка	29
S3	Винт	77
S2	Болт	15
...	...	...

## Описание предметной области:

- Поставщик может поставлять различные детали.
- Одна и та же деталь может поставляться разными поставщиками.
- Все поставщики поставляют деталь по одной и той же цене.

{КодПоставщика, Деталь} → {Цена} – по определению первичного ключа

{Деталь} → {Цена} – по описанию предметной области

# Определение 2НФ

**Формальное определение:** Отношение *находится в 2НФ* тогда и только тогда, когда

- 1) оно находится в 1НФ
- 2) каждый неключевой атрибут функционально полно зависит от каждого ключа отношения.

**Неформальное определение для таблицы с единственным ключом:**

- 1) Таблица должна находиться в первой нормальной форме.
- 2) Любое ее поле, не входящее в состав первичного ключа, функционально зависит только от всех атрибутов ключа и не зависит от его отдельных атрибутов.

**Замечание:**

Если таблица с единственным ключом приведена к первой нормальной форме и у нее установлен уникальный id для каждой строки, то она находится во второй нормальной форме.

# Пример отношения не в 2НФ

- Значение второго правила («любое поле таблицы, не входящее в состав первичного ключа, функционально зависит только от всех атрибутов ключа и не зависит от его отдельных атрибутов») можно понять на примере с составным первичным ключом таблицы:

## Поставки

КодПоставщика*	Деталь*	Цена
S1	Болт	15
S1	Гайка	29
S3	Винт	77
S2	Болт	15
...	...	...

{КодПоставщика, Деталь}→{Цена}  
{Деталь}→{Цена}



Эта таблица находится в первой нормальной форме, но не во второй.

# Приведение к 2НФ

## Поставки

КодПоставщика*	Деталь*	Цена
S1	Болт	15
S1	Гайка	29
S3	Винт	77
S2	Болт	15
...	...	...

Необходимо избавиться от функциональной зависимости  $\{Деталь\} \rightarrow \{Цена\}$  по теореме Хита

Приведение к 2НФ



Поставки =

$$= \pi_{Деталь, Цена} (Поставки) \bowtie \pi_{Деталь, КодПоставщика} (Поставки)$$

Деталь*	Цена
Болт	15
Гайка	29
Винт	77
...	...

Деталь*	КодПоставщика*
S1	Болт
S1	Гайка
S3	Винт
S2	Болт
...	...



# Транзитивная функциональная зависимость

- Определение: Если для атрибутов  $X$ ,  $Y$  и  $Z$  отношения  $R$  существуют функциональные зависимости  $X \rightarrow Y$ ,  $Y \rightarrow Z$ , говорят, что **атрибут  $Z$  связан транзитивной зависимостью с атрибутом  $X$  через атрибут  $Y$**  (при этом атрибут  $X$  не должен функционально зависеть ни от атрибута  $Y$ , ни от атрибута  $Z$ ).
- Пример транзитивной зависимости:

Хранение

Поставщик*	Склад	Объем склада
S1	Склад 1	32
S5	Склад 2	32
S3	Склад 5	5
S2	Склад 1	32
...	...	...

Описание предметной области:

- Содержит информацию о поставщиках, получающих товары со складов, и объемах этих складов.
- Каждый поставщик может получать товар только с одного склада.

{Поставщик}  $\rightarrow$  {Склад}

{Склад}  $\not\rightarrow$  {Поставщик}

{Склад}  $\rightarrow$  {Объем склада}

{Объем склада}  $\not\rightarrow$  {Склад}

***Т.О. атрибут «Объем склада» связан транзитивной зависимостью с атрибутом «Поставщик» через атрибут «Склад»***

# Определение 3НФ

**Формальное определение:** Отношение находится в 3НФ тогда и только тогда, когда

- 1) оно находится в 2НФ и
- 2) каждый неключевой атрибут нетранзитивно зависит от первичного ключа.

**Неформальное определение:**

- 1) Таблица находится во второй нормальной форме.
- 2) Любой ее неключевой атрибут функционально зависит только от первичного ключа.

# Приведение к 3НФ

## Хранение

Поставщик*	Склад	Объем склада
S1	Склад 1	32
S5	Склад 2	32
S3	Склад 5	5
S2	Склад 1	32
...	...	...

Необходимо по теореме Хита избавиться от транзитивной функциональной зависимости:  
 $\{\text{Поставщик}\} \rightarrow \{\text{Склад}\}$   
 $\{\text{Склад}\} \rightarrow \{\text{Объем склада}\}$

Избавляемся от ФЗ в обратном порядке: сначала от  $\{\text{Склад}\} \rightarrow \{\text{Объем склада}\}$ , затем от  $\{\text{Поставщик}\} \rightarrow \{\text{Склад}\}$

Приведение к 3НФ



Хранение =

$$= \pi_{\text{Склад, Объем склада}}(\text{Хранение}) \bowtie \pi_{\text{Склад, Поставщик}}(\text{Хранение})$$

Поставщик*	Склад
S1	Склад 1
S5	Склад 2
S3	Склад 5
S2	Склад 1
...	...

Склад*	Объем склада
Склад 1	32
Склад 2	32
Склад 5	5
...	...

# Пример 1

## Сотрудники Отделы Проекты

Код_Сотр*	Зарплата_Сотр	Код_Отдела	Код_Проекта*	Задание_Сотр
001	20 000	2	П_4	тестирование
001	20 000	2	П_1	составление ТЗ
002	50 000	1	П_4	программирование
003	50 000	1	П_4	сопровождение
004	20 000	2	П_3	программирование
002	50 000	1	П_1	тестирование

### Описание предметной области:

- Один сотрудник работает над разными проектами.
- Сотрудник работает только в одном отделе.
- Сотрудники из одного отдела получают одинаковую зарплату.
- В рамках одного проекта разные сотрудники выполняют разные задания.

# Пример 1

Сотрудники Отделы Проекты

Код_Сотр*	Зарплата_Сотр	Код_Отдела	Код_Проекта*	Задание_Сотр
-----------	---------------	------------	--------------	--------------

**Описание предметной области:**

- Один сотрудник работает над разными проектами.
- Сотрудник работает только в одном отделе.
- Сотрудники из одного отдела получают одинаковую зарплату.
- В рамках одного проекта разные сотрудники выполняют разные задания.

**Функциональные зависимости:**

- Код\_Сотр  $\rightarrow$  Код\_Отдела
  - Код\_Отдела  $\rightarrow$  Зарплата\_Сотр
  - {Код\_Сотр, Код\_Проекта}  $\rightarrow$  Задание\_Сотр
  - {Код\_Сотр, Код\_Проекта}  $\rightarrow$  {Зарплата\_Сотр, Код\_Отдела, Задание\_Сотр}
- Транзитивная зависимость  
Код\_Сотр  $\rightarrow$  Зарплата\_Сотр*

# Пример 1

Сотрудники Отделы Проекты

Код_Сотр*	Зарплата_Сотр	Код_Отдела	Код_Проекта*	Задание_Сотр
-----------	---------------	------------	--------------	--------------

**Функциональные зависимости:**

- Код\_Сотр → Код\_Отдела
  - Код\_Отдела → Зарплата\_Сотр
  - {Код\_Сотр, Код\_Проекта} → Задание\_Сотр
  - {Код\_Сотр, Код\_Проекта} → {Зарплата\_Сотр, Код\_Отдела, Задание\_Сотр}
- Транзитивная зависимость  
Код\_Сотр → Зарплата\_Сотр*

**Какие здесь аномалии?**

- Обновления
- Удаления
- Добавления

# Пример 1

Сотрудники Отделы Проекты

Код_Сотр*	Зарплата_Сотр	Код_Отдела	Код_Проекта*	Задание_Сотр
-----------	---------------	------------	--------------	--------------

**Функциональные зависимости:**

- Код\_Сотр → Код\_Отдела
  - Код\_Отдела → Зарплата\_Сотр
  - {Код\_Сотр, Код\_Проекта} → Задание\_Сотр
  - {Код\_Сотр, Код\_Проекта} → {Зарплата\_Сотр, Код\_Отдела, Задание\_Сотр}
- Транзитивная зависимость*  
*Код\_Сотр → Зарплата\_Сотр*

**Находится ли данная БД в 1НФ?**

- Да, т.к. все значения атомарные и нет дублирующихся строк.

# Пример 1

Сотрудники Отделы Проекты

Код_Сотр*	Зарплата_Сотр	Код_Отдела	Код_Проекта*	Задание_Сотр
-----------	---------------	------------	--------------	--------------

**Функциональные зависимости:**

- Код\_Сотр → Код\_Отдела
  - Код\_Отдела → Зарплата\_Сотр
  - {Код\_Сотр, Код\_Проекта} → Задание\_Сотр
  - {Код\_Сотр, Код\_Проекта} → {Зарплата\_Сотр, Код\_Отдела, Задание\_Сотр}
- Транзитивная зависимость  
Код\_Сотр → Зарплата\_Сотр*

**Находится ли данная БД в 2НФ?**

- Нет, т.к. существуют функциональные зависимости неключевого атрибута от части первичного ключа: Код\_Сотр → Код\_Отдела, Код\_Сотр → Зарплата\_Сотр.



# Пример 1

## СотрудникиОтделыПроекты

Код_Сотр*	Зарплата_Сотр	Код_Отдела	Код_Проекта*	Задание_Сотр
-----------	---------------	------------	--------------	--------------

Необходимо избавиться от транзитивной функциональной зависимости, которая включает в себя функциональные зависимости неключевого атрибута от части первичного ключа:

- Код\_Сотр  $\rightarrow$  Код\_Отдела
  - Код\_Отдела  $\rightarrow$  Зарплата\_Сотр
- } Транзитивная зависимость  
 $\text{Код\_Сотр} \rightarrow \text{Зарплата\_Сотр}$

### Приведение к 2НФ:

СотрудникиОтделыПроекты =

$$= \pi_{\text{Код\_Отдела, Зарплата\_Сотр}} (\text{СотрудникиОтделыПроекты}) \bowtie$$

$$\bowtie \pi_{\text{Код\_Отдела, Код\_Сотр, Код\_Проекта, Задание\_Сотр}} (\text{СотрудникиОтделыПроекты})$$

R (Код\_Отдела, Код\_Сотр, Код\_Проекта, Задание\_Сотр) =

$$= \pi_{\text{Код\_Сотр, Код\_Отдела}} (R) \bowtie$$

$$\bowtie \pi_{\text{Код\_Сотр, Код\_Проекта, Задание\_Сотр}} (R)$$

# Пример 1

## СотрудникиОтделы

Код_Сотр*	Код_Отдела
001	2
002	1
003	1
004	2

## Отделы

Код_Отдела*	Зарплата_Сотр
2	20 000
1	50 000

## СотрудникиПроекты

Код_Сотр*	Код_Проекта*	Задание_Сотр
001	П_4	тестирование
001	П_1	составление ТЗ
002	П_4	программирование
003	П_4	сопровождение
004	П_3	программирование
002	П_1	тестирование

### Находится ли данная БД в ЗНФ?

- Да, т.к. нет транзитивных функциональных зависимостей.

## Пример 2

№ зачетки*	Фамилия	Имя	Отчество	Предмет*	Оценка
100	Иванов	Сергей	Иванович	Информатика	3
...	...	...	...	...	...
1100	Петров	Сергей	Андреевич	Информатика	5
...	...	...	...	...	...
100	Иванов	Сергей	Иванович	Базы данных	5
...	...	...	...	...	...

Описание предметной области:

- Одна зачетная книжка выдается только одному студенту
- Один студент по одному предмету получает только одну оценку

# Пример 3

## S\_P\_SP

Имя_П*	Город_П	Рейтинг	Код_Д*	Имя_Д	Город_Д	Вес	Цена	Кол.
Бендер	Черноморск	15	P1	Болт	Париж	15	40	1000
ИТ-монстр	Москва	29	P2	Шуруп	Москва	29	33	7600
Бендер	Черноморск	15	P2	Шуруп	Москва	29	33	1240
ИТ-монстр	Москва	29	P3	Гвоздь	Москва	10	42	2700
ИТ-монстр	Москва	29	P4	Болт	Омск	14	31	1000

Описание предметной области:

- Разные поставщики могут быть из одного города
- Рейтинг поставщика определяется по городу поставщика
- Детали с одинаковым названием могут выпускаться в разных городах
- Детали с одинаковым названием могут иметь разный вес
- Детали с одинаковым названием могут иметь разную цену