

# Реляционные базы данных

Бессарабов Н.В.

[bes@fpm.kubsu.ru](mailto:bes@fpm.kubsu.ru)

2018 г.

# Цели лекции

В этой лекции начинаем рассматривать реляционную модель, в которой единственным источником данных являются отношения, может быть связанные между собой.

Основная цель лекции – теорема Хиса, позволяющая проверить, “правильно” ли сформированы отношения. Определим операцию декомпозиции – разбиения заполненного или незаполненного отношения на части. Удивительно, но не всегда воссоединение компонент дает исходное отношение. В этом случае декомпозиция называется неполной. На языке алгебры декомпозиция и воссоединение компонент определяются, соответственно, операциями проекции и естественного соединения. Можно было бы определить и другие операции на отношениях, задав реляционную алгебру. Мы это сделаем позже. А сейчас учимся строить “правильные” отношения.

И последнее. Оказывается, функциональные зависимости, определенные на отношениях, дают естественный язык для задания идентификаторов кортежей отношений (ключей) и для формулирования теоремы Хиса о декомпозициях отношений.

Об изучении  
логики в школах и  
высших учебных  
заведениях

Вопрос: Что было  
раньше, курица или  
яйцо?

Ответ: Раньше всё  
было.

Показан титульный  
лист 8-го издания  
учебника для школы.

С.Н.ВИНОГРАДОВ и А.Ф.КУЗЬМИН

---

# ЛОГИКА

УЧЕБНИК  
ДЛЯ СРЕДНЕЙ  
ШКОЛЫ

---

УЧПЕДГИЗ · 1954

# Высказывания и предикаты

- Вспомним, что высказывание это повествовательное предложение, для которого можно дать оценку истинности “Истинно” или “Ложно”. Например, «сегодня хорошая погода»..
- Если для любой собственной части высказывания нельзя определить его истинность, то высказывание называют простым.
- Компоненты сложного высказывания связываются связками “И”, “НЕ”, “ИЛИ”, “ЕСЛИ ... ТО” и другими (“ОНО КОНЕЧНО, ... НО ОПЯТЬ ТАКИ ...”). Житейский смысл этих связок может существенно отличаться от принятого в классической логике.
- Высказывание, содержащее переменные, называется предикатом. Пример: “ $X > 0$ ”. Предикат может быть истинным или ложным в зависимости от значения входящих в него переменных.
- Переход от высказываний в естественном языке к высказываниям на формальных языках не тривиален. Одна из причин – референция, в частности, эллипсис -- намеренный пропуск слов, не искажающих смысл высказывания.

Пример эллипсиса: “зарплата больше 1000, но меньше 2000”.

Точный эквивалент: “зарплата больше 1000 **И ЗАРПЛАТА** меньше 2000”.

# Предикаты, понятия, концепты, функции

- **$n$ -местный ( $n$ -арный) предикат** — можно считать функцией в множество из двух элементов  $\{0,1\}$  или {"Истина", "Ложь"}, определённой на  $n$ -й декартовой степени некоторого множества  $M$  или декартовом произведении  $n$  множеств  $M_1 \times M_2 \times \dots \times M_n$ .
- Предикаты связаны с отношениями, определёнными на том же множестве  $M_1 \times M_2 \times \dots \times M_n$ : если кортеж (элемент этого множества) принадлежит отношению, то предикат на нём вернёт значение "Истина".
- Будем различать понятия и их чётко определённые версии – концепты.
- Предикаты связаны с определениями понятий (более точно, концептов), но не всякое понятие есть предикат.
- Нульместные предикаты определяют константы.
- Одноместные предикаты задают свойства объектов рассуждения. Например, предикат "учится\_в\_КубГУ( $X$ )".
- При  $n > 1$  предикат определяет отношение.
- Поскольку частные случаи функций – предикаты -- принимают два значения, то к ним применимы все операции булевой алгебры: отрицание, импликация, конъюнкция, дизъюнкция и т. д.

Замечание: Не считайте, что связи "предикат - высказывание",

"предикат - отношение", "предикат - определение" тривиальны.

# Отношения и предикаты

В 1970 г. появилась работа Э.Ф.Кодда [1], в которой он применил к описанию баз данных алгебру отношений – **реляционную алгебру**.

Появилась **реляционная модель данных**, представляющая базу данных как набор отношений, может быть связанных.

Схема отношения определяется уже известным нам способом – через задание предиката. Принадлежность кортежа к отношению определяется истинностью описывающего его предиката.

Однако, в реляционной модели принято говорить не о предикатах, а об отношениях и их атрибутах.

Пример: *отдел (ном\_отд, назв\_отд, город)*



После 1970 г. масса важных событий. 1974 г. -- создание языка SQL (Чемберлин, Бойс), 1974-1975 язык QBE (М. Злуф), 1976 г. -- ERD (П. Чен), 1977 – начало фирмы Oracle (тогда Software Development Laboratories, позже *Relational Software, Inc.*)

# Отношения

Итак, рассматриваем отношения как наборы однотипных строк-кортежей. Отношения имеют имя и набор свойств, называемых **атрибутами** отношения.

Пример: Объект “Студент” с атрибутами “Фамилия”, “Имя”, “Отчество”, “Дата\_рождения”, “Студенческая\_группа”.

Схема отношения:

*Студент(Фамилия, Имя, Отчество, Дата\_рождения,  
Студенческая\_группа)*

Каждый студент представляется строкой своих значений атрибутов – **кортежем**.

Пример: (“Иванов”, “Петр”, “Сидорович”, “22.05.82”, 32).

Свойство: В отношении не может быть двух одинаковых кортежей.

Замечание 1: Данные в реляционных базах хранятся **только в отношениях**. Других источников данных не существует.

Замечание 2: Затруднения в задании семантики могут возникнуть, если имеются атрибуты, задающие временн`ые (темпоральные)<sub>7</sub> свойства сущности.

# Свойства отношений

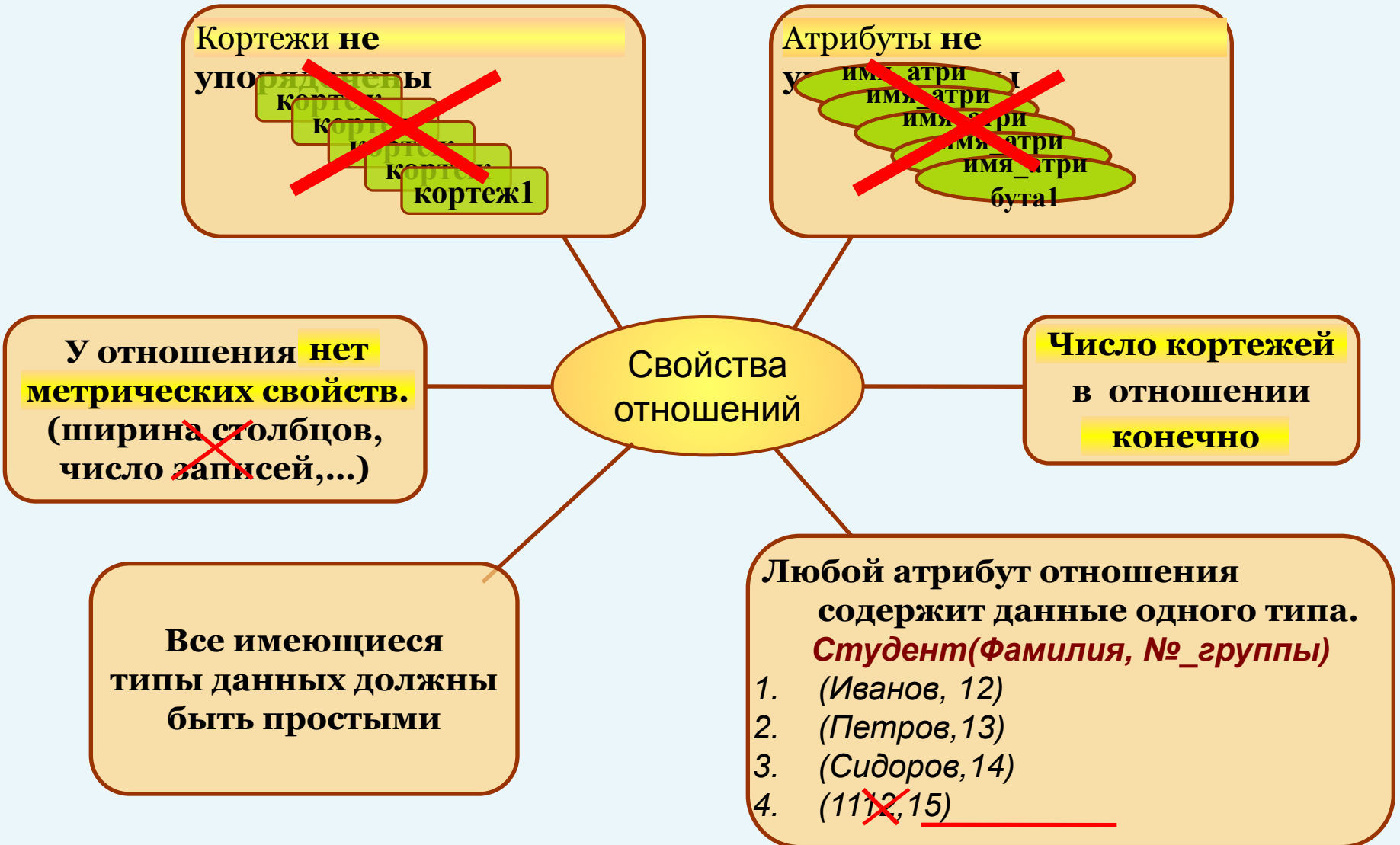
Основные свойства отношений (в реляционной теории):

1. Кортежи не упорядочены
2. Атрибуты не упорядочены
3. Число кортежей в отношении конечно
4. Любой атрибут отношения должен содержать данные одного типа
5. Все используемые типы данных должны быть простыми
6. Отношение не обладает метрическими свойствами, такими как ширина столбцов, число кортежей и т.д.

Замечание: В реализациях отсутствие упорядоченности может быть не удобным или не реализуемым, а метрические свойства всегда важны, поскольку они определяют многие свойства приложения работающего с базой данных.



# Свойства отношений



# Плоские (реляционные) таблицы

В реализациях реляционных баз (т. е. на физическом уровне) отношениям соответствуют плоские (реляционные) таблицы.

Определение: **Плоская (реляционная) таблица** – это таблица с одной одноуровневой шапкой и **атомарными значениями** в ячейках таблицы. Выбор ячейки в ней однозначно определяется выбором строки и указанием имени столбца.

Каждый атрибут принимает значения из какого-то множества допустимых значений, называемого **доменом**.

Множество кортежей (записей), образующих отношение естественно представляется строками таблицы.

Поскольку между отношением и представляющей его таблицей существует взаимно однозначное соответствие, имеем право утверждать, что строки реляционных таблиц не повторяются.

Уточнение: Атомарность означает не простоту значения, а то, что значение в базе не разделяется на части (она “не умеет” этого делать).

Пример: в поле “ФИО”, содержащем фамилию, имя и отчество, не возможно выделить ни фамилии, ни имени, ни отчества.

# Пример реляционной таблицы

Название  
таблицы

Столбец DEPTNO, пред-  
ставляющий атрибут  
НОМЕР ОТДЕЛА

DEPT

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW-YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

Это шапка таблицы

Это строка таблицы -  
запись

Домен для DEPTNO  
возможно есть множество  
значений 10, 20, 30, 40, 50, ...

# Состояние отношения “Сотрудники”

Шапка

Столбец

Строка

Таб_номер	Фамилия	Зарплата	Номер_отдела
1	Иванов	1000	10
2	Петров	2000	20
3	Сидоров	3000	10

Состояние  
отноше-  
ния

Определение: Состояние отношения определяется набором входящих в него кортежей.

Замечание: Состояние отношения в реляционной теории не рассматривается.

# Таблица с двумя шапками, верхней и боковой (не реляционная)

## ПРИБЫЛЬ

Отдел/ Год	2000	2001
Производственный	1134	7545
Торговый	0	17555

Создайте эквивалентную  
реляционную таблицу

# Преобразуем гиперкуб в таблицу

Таблица “Прибыль” из предыдущего слайда преобразуется в реляционную таблицу вида

<b>2000_ Производственный</b>	<b>2000_ Торговый</b>	<b>2001_ Производственный</b>	<b>2001_ Торговый</b>
1134	0	7545	17555

Имена столбцов в ней образованы конкатенацией имен исходной таблицы, связанных операцией “И”. Столбцу соответствует некоторое высказывание, например, для “2000\_Торговый” это “Год=2000” & Отдел=“Торговый”.

Таким же образом трансформируются многомерные таблицы (гиперкубы) любой размерности.

Вопрос: Чем плох этот вариант? Может быть лучше так:

<b>Отдел</b>	<b>Год</b>	<b>Прибыль</b>
Производственный	2000	1134
Производственный	2001	7545
Торговый	2000	0
Торговый	2001	17555

# Операции над отношениями

В этой лекции будут рассмотрены только операции проекции и естественного соединения, необходимые для рассмотрения декомпозиции отношения на его проекции и восстановления исходных соотношений.

Все остальные операции и определение реляционной алгебры даны в следующей лекции.

Замечание: Обратите внимание на то, что операции действуют на содержимое отношений и преобразуют их схемы. Результат операции есть новое отношение.





# Естественное соединение

Пусть отношения  $r_1$  и  $r_2$  имеют схемы

$R_1(A_1, \dots, A_k, B_1, \dots, B_n)$

и

$R_2(A_1, \dots, A_k, C_1, \dots, C_m)$ .

Тогда **естественное соединение** (join) отношений

$r_1$  и  $r_2$  есть отношение  $r_3$  со схемой

$R_3(A_1, \dots, A_k, B_1, \dots, B_n, C_1, \dots, C_m)$

в котором каждая запись (экземпляр) получена конкатенацией каждой записи из  $r_1$  с теми записями из  $r_2$ , у которых совпадают значения в общих атрибутах  $A_1, \dots, A_k$ .

# Пример естественного соединения

$r_1$ :

B	A
$b_1$	3
$b_2$	7
$b_3$	4
$b_4$	4

$r_2$ :

A	C
3	$c_1$
3	$c_2$
4	$c_3$

$r_1 \text{ join } r_2$ :

A	B	C
3	$b_1$	$c_1$
3	$b_1$	$c_2$
4	$b_3$	$c_3$
4	$b_4$	$c_3$

Обозначения:  $\text{join}(r_1, r_2)$  или  $\text{join}_{=A}(r_1, r_2)$  или  $r_1 \text{ join } r_2$

# Декомпозиция отношения

Определение: Полная декомпозиция отношения это набор его проекций, соединение которых идентично отношению.

**Существуют неполные декомпозиции!!!!**

# Пример полной декомпозиции

Исходное отношение

r:

A	B	C
a <sub>1</sub>	b <sub>1</sub>	c <sub>1</sub>
a <sub>1</sub>	b <sub>2</sub>	c <sub>2</sub>
a <sub>2</sub>	b <sub>3</sub>	c <sub>1</sub>
a <sub>2</sub>	b <sub>2</sub>	c <sub>2</sub>

Соединение этих проекций даст исходное отношение.

Проверьте это сами!

$\text{proj}_{\{A,B\}}(r)$ :

A	B
a <sub>1</sub>	b <sub>1</sub>
a <sub>1</sub>	b <sub>2</sub>
a <sub>2</sub>	b <sub>3</sub>
a <sub>2</sub>	b <sub>2</sub>

$\text{proj}_{\{B,C\}}(r)$ :

B	C
b <sub>1</sub>	c <sub>1</sub>
b <sub>2</sub>	c <sub>2</sub>
b <sub>3</sub>	c <sub>1</sub>

# Неполная декомпозиция. Присоединенные записи.

Соединение  
проекций  $r_1, r_2$

того же

отношения  $r$

создает присоединенные записи

$$r_1 = \text{proj}_{\{A,C\}}(r):$$

A	C
a <sub>1</sub>	c <sub>1</sub>
a <sub>1</sub>	c <sub>2</sub>
a <sub>2</sub>	c <sub>1</sub>
a <sub>2</sub>	c <sub>2</sub>

$$r_2 = \text{proj}_{\{B,C\}}(r):$$

B	C
b <sub>1</sub>	c <sub>1</sub>
b <sub>2</sub>	c <sub>2</sub>
b <sub>3</sub>	c <sub>1</sub>

$$\text{join}_{\{C\}}(r_1, r_2):$$

A	B	C
a <sub>1</sub>	b <sub>1</sub>	c <sub>1</sub>
a <sub>1</sub>	b <sub>3</sub>	c <sub>1</sub>
a <sub>1</sub>	b <sub>2</sub>	c <sub>2</sub>
a <sub>2</sub>	b <sub>1</sub>	c <sub>1</sub>
a <sub>2</sub>	b <sub>3</sub>	c <sub>1</sub>
a <sub>2</sub>	b <sub>2</sub>	c <sub>2</sub>



присоединенные записи

# Первичный ключ (1/2)

Определение (Первичный ключ отношения в реляционной алгебре): Атрибут или набор атрибутов, значения которых позволяют однозначно выбрать кортеж отношения, называется **первичным ключом**.

Ключ состоящий из одного атрибута называют **простым**.

Ключ, состоящий из двух или более атрибутов называют **составным** или **конкатенированным**.

Обратим внимание на то, что ключи в отношениях (логический уровень) и соответствующих им реляционных таблицах находятся во взаимно однозначном соответствии.

Определение (суррогатный ключ): **Суррогатным** называется ключ, не имеющий прототипа в предметной области.

Обычно его значения генерируются приложением.

# Первичный ключ (2/2)

Утверждение: Из того, что кортежи отношений не повторяются следует, что любое отношение имеет ключ.

Заметим, что в ключ могут войти все атрибуты отношения.

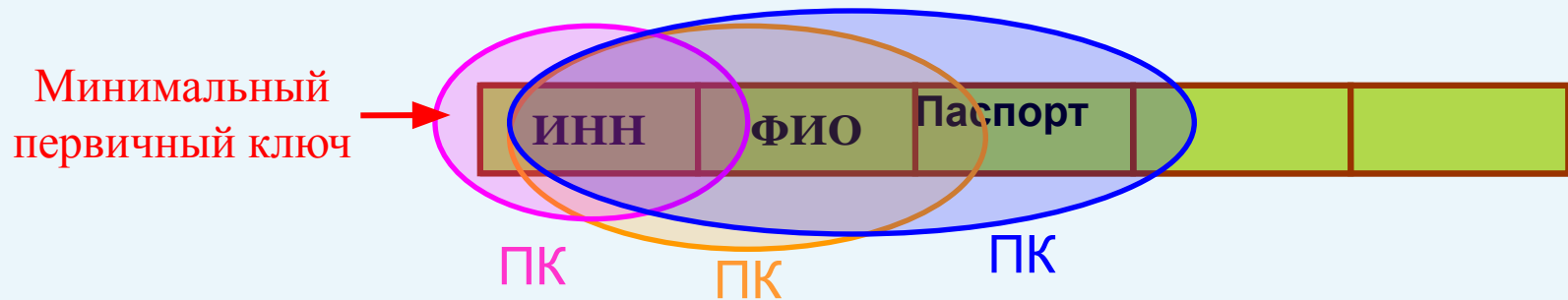
Утверждение: Пополнение ключа еще одним (не ключевым) атрибутом есть ключ.

Определение: Если удаление одного атрибута лишает ключевой набор атрибутов статуса ключа, то такой ключ называют **минимальным**.

Замечание 1 (Важно!!): **Первичным ключом** называют в действительности **минимальный первичный ключ**.

Замечание 2: Операция проекция может иметь результатом отношение в котором нет первичного ключа.

# Говорим “первичный ключ”, а подразумеваем “минимальный первичный ключ”



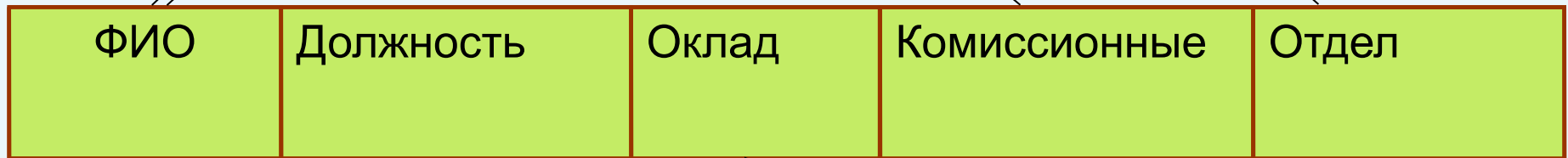
Замечание: Изучая базы данных следует понимать, что все приводимые в лекциях примеры по необходимости сделаны простыми. Всегда следует ограничиваться контекстом примера, и не пытаться рассуждать “как в жизни”, или “как может быть”. Так в нашем отношении можно спросить: “А как быть с теми, у кого нет ИНН?”. Так вот, в примере предполагается, что ИНН есть у всех. При проектировании реальных баз данных необходимые ориентиры дает спецификация модели бизнеса.



# Функциональные зависимости на отношениях

Функциональные зависимости и зависимости связанные с обобщениями функций играют важную роль в теории реляционных баз данных. В частности, ключи определяются через функции.

Некоторые ФЗ  
определенные  
первичным  
ключом



ПК

Зависимость не от ключа, если  
оклад определяется только должностью

# Теорема Хиса (Heath)

Устанавливает связь между функциональными зависимостями в схеме отношения и способом его полной декомпозиции.

**Теорема Хиса:** Пусть в отношении  $r$  со схемой  $R(S)$ , где  $S$  – полный набор атрибутов отношения, выделены три набора атрибутов  $A, B, C$ , таких что  $A \cap B = \emptyset$ ,  $A \cap C = \emptyset$ ,  $B \cap C = \emptyset$ ,  $A \cup B \cup C = S$ .

Тогда, если существует функциональная зависимость действующая из набора  $B$  в  $C$ , то проекции  $\text{proj}_{\{A,B\}}(r)$ ,  $\text{proj}_{\{B,C\}}(r)$  образуют полную декомпозицию отношения  $r$ .

**Замечание:** обратите внимание на то, что проекция  $\text{proj}_{\{B,C\}}(r)$  содержит аргумент и значение функции, а вторая проекция – только аргумент.

# Теорема Хиса –доказательство (1/2)

Введем вспомогательное отношение

$$r_1 = \text{proj}_{\{A,B\}}(r) \text{ join } \text{proj}_{\{B,C\}}(r) \quad (1)$$

как соединение двух проекций отношения  $r$ .

Покажем, что  $r_1 \equiv r$ .

1) Выберем произвольную запись (кортеж)

$\langle a, b, c \rangle$  из  $r$ .

Так как,

$$\langle a, b \rangle \in \text{proj}_{\{A,B\}}(r),$$

а

$$\langle b, c \rangle \in \text{proj}_{\{B,C\}}(r),$$

то по свойству операции соединения

$\langle a, b, c \rangle \in r_1$ . Следовательно,

$$r \subseteq r_1 \quad (2)$$



# Теорема Хиса –доказательство (2/2)

Идём от  $r_1$  к  $r$

2) Выберем произвольный кортеж  $\langle a', b', c' \rangle \in r_1$ .

Из определения отношения  $r_1$  (1) следует, что

$$\langle a', b' \rangle \in \text{proj}_{\{A,B\}}(r) \quad (3)$$

$$\langle b', c' \rangle \in \text{proj}_{\{B,C\}}(r) \quad (4)$$

Значит в  $r$  существуют записи

$$(3) \Rightarrow \langle a', b', c \rangle \in r \quad (5)$$

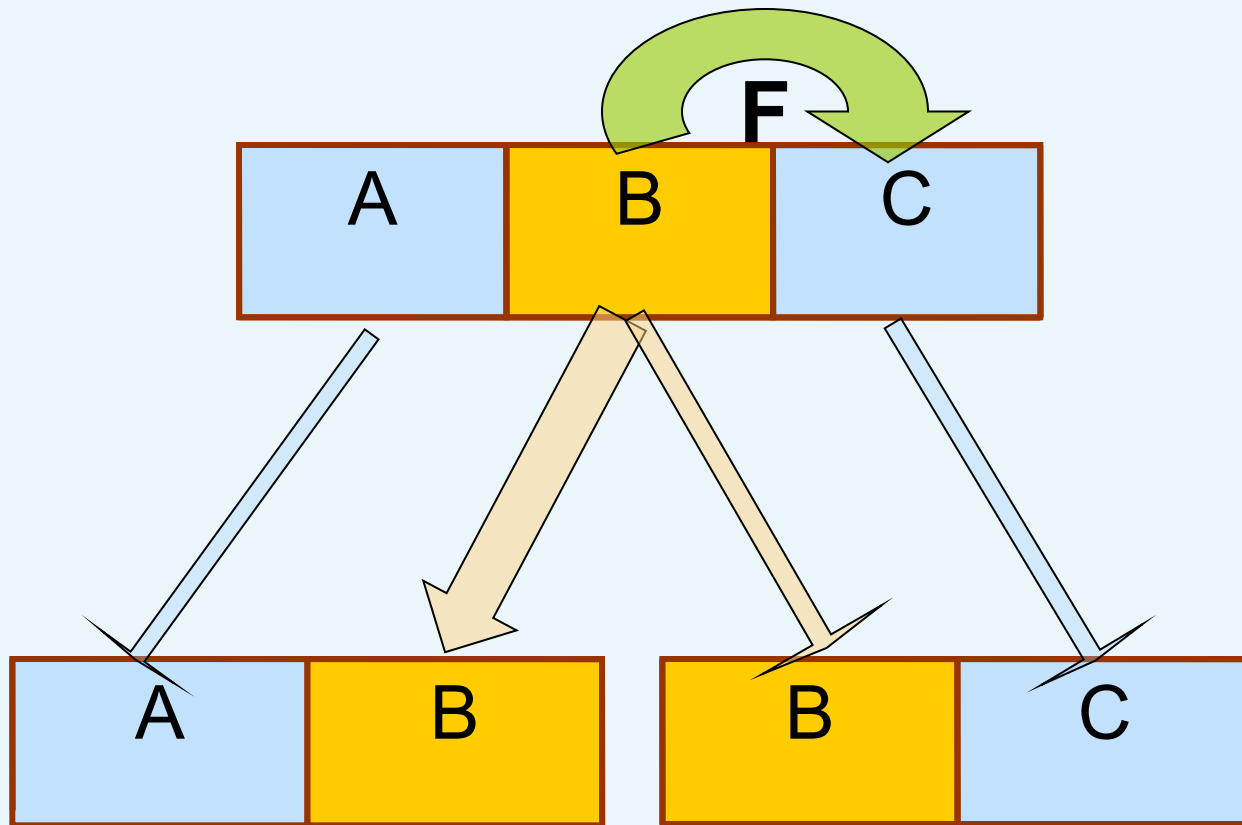
$$(4) \Rightarrow \langle a, b', c' \rangle \in r \quad (6)$$

Но  $C$  функционально зависит от  $B$ . Поэтому в кортежах (5) и (6)  $c = c'$ , а значит запись  $\langle a', b', c' \rangle$  входит в отношение  $r$ , иначе говоря

$$r_1 \subseteq r \quad (7)$$

3) Сопоставляя (2) и (7), получаем  $r \equiv r_1$ . Иначе говоря, декомпозиция (1) полная.

# Мнемоническое изображение декомпозиции по теореме Хиса



# Пример к теореме Хиса

В теореме Хиса не предполагается, что на атрибутах могут быть определены какие-то структуры.

Пример: Список закупаемой партии товаров

Название РК	Кол-во РК	Вес единицы товара	Единица изм.веса	Вес всего
-------------	-----------	--------------------	------------------	-----------

Особенности отношения:

- Первичный ключ образуют атрибуты “Название” и “Количество”.
- Атрибут “Вес единицы товара” нельзя использовать без уточняющего атрибута “Единица измерения”; они образуют неделимую группу.
- Атрибут “Вес всего” вычисляемый, определяется через атрибуты “Количество” и “Вес единицы товара”, и также связан с атрибутом “Единица измерения”.

Поэтому на рассматриваемом отношении существуют:

- Функция из первичного ключа (атрибуты “Название” и “Количество”) в агрегат атрибутов (“Вес единицы товара”, “Единица измерения”);
- Функция из атрибутов “Количество”, “Вес единицы товара” в атрибут “Вес всего”.

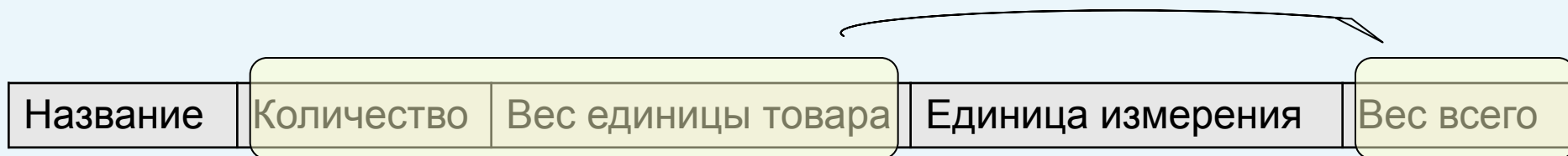
Заметим, что в примере перечислен не полный список функций.

# Уточнение теоремы Хиса

Не все упомянутые структуры могут быть использованы в теореме Хиса.

На рисунке ниже отмечена функция из атрибутов “Количество”, “Вес единицы товара” в атрибут “Вес всего”.

Однако, выделенная по Хису сущность с атрибутами “Количество”, “Вес единицы товара” и “Вес всего” бессмысленна. В ней не понятно, к каким товарам относятся приведенные данные.



## Уточнение теоремы Хиса:

**” Теорему Хиса следует использовать только если сущности полученные в результате декомпозиции осмысленны в какой-то семантике”**

# Заключение

Что Вы должны освоить, прослушав эту лекцию:

1. Понятие “отношение”, схема отношения, свойства отношений, их состояния, связи с предикатами, связи с сущностями предметной области
2. Реляционные таблицы и их сопоставление с отношениями в модели сущность-связь
3. Операции проекции и естественного соединения
4. Полная, неполная декомпозиция и присоединенные записи
5. Функциональные зависимости на отношениях, их роль в реляционной теории
6. Первичные ключи, их виды и свойства
7. Теорема Хиса (ЭТО ГЛАВНЫЙ РАЗДЕЛ ЛЕКЦИИ!!)
8. Важность учёта семантики при работе с теоремой Хиса



# Литература

1. E.F. Codd, A Relational Model of Data for Large Shared Databanks, Communications of the ACM, June 1970, с. 377-387
3. К. Дж. Дейт, Введение в системы баз данных. М.: Издательский дом “Вильямс”, 2005. – 1328 с.

# Основные понятия



# Словарь студента (1/3)

- ❑ **Атомарное значение** предполагается в рамках модели данных не разделяемым на части (но может быть разделяемо вне базы).
- ❑ **Атрибут.** Отношение описывается схемой, в которую входят имя, набор атрибутов (свойств), их типы и ограничения целостности.
- ❑ **Декомпозиция** отношения – это набор проекций отношения или процесс получения такого набора; декомпозиция неполна, если соединение проекций дает присоединенные записи (которых не было в исходном отношении).
- ❑ **Естественное соединение** это соединение кортежей двух отношений у которых равны значения в парах соответствующих атрибутов.
- ❑ **Ключ первичный (ПК)** – атрибут или набор атрибутов, значения которых позволяют однозначно выбрать кортеж отношения.
  - **простой** ПК - ключ состоящий из одного атрибута;
  - **составной (конкатенированный)** ПК содержит два или более атрибутов;
  - **суррогатный** ПК – ключ, не имеющий прототипа в предметной области; обычно состоит из одного атрибута;
  - **минимальный** ПК; если удаление одного атрибута лишает набор атрибутов статуса ключа, то ключ называют **минимальным**;Принято термином “ключ” называть только минимальный ключ.<sup>35</sup>

# Словарь студента (2/3)

- ❑ **Кортеж** – упорядоченная совокупность значений атрибутов отношения; в отношении не может быть двух одинаковых кортежей.
- ❑ **Отношение** - это набор однотипных кортежей.
- ❑ **Плоская таблица** - это таблица, с одной одноуровневой шапкой и атомарными значениями в ячейках таблицы; таблице базы данных соответствует отношение реляционной модели.
- ❑ **Полная декомпозиция** отношения – это набор его проекций, соединение которых идентично исходному отношению.
- ❑ **Проекция** - это набор операций выбора подмножества столбцов отношений (в заполненном отношении).
- ❑ **Присоединенные записи** – записи, отсутствующие в исходном отношении, но получающиеся в результате соединения проекций этого отношения.
- ❑ **Реляционная алгебра** -- определена в следующей лекции.
- ❑ **Реляционная модель данных** – модель данных, представляющая базу данных как набор отношений, может быть связанных.

# Словарь студента (3/3)

- ❑ **Свойства отношений** в реляционной теории
  - Кортежи не упорядочены.
  - Атрибуты не упорядочены.
  - Число кортежей в отношении конечно.
  - Любой атрибут отношения должен содержать данные одного типа.
  - Все используемые типы данных должны быть простыми.
  - Отношение не обладает метрическими свойствами, такими как ширина столбцов, число записей и т.д.
- ❑ **Теорема Хиса** - устанавливает связь между функциональными зависимостями в схеме отношения и способом его полной декомпозиции.
- ❑ **Функциональная зависимость** – зависимость одних атрибутов от других. В частности, первичный ключ определяет функции из ключа в любой неключевой (и ключевой) столбец.