

# Проектирование баз данных

Продолжение

# Неприводимое множество ФЗ

- ◆ Множество ФЗ называется неприводимым слева, если
  - Зависимые части – одноэлементные множества
  - Каждый детерминант является неприводимым (ни один атрибут не может быть исключен без изменения замыкания)
  - Ни одна ФЗ не может быть исключена, без изменения замыкания
- ◆ Для любого множества ФЗ существует по крайней мере одно эквивалентное множество, которое является неприводимым

# Пример

$R(a,b,c,d)$

$\{a\} \rightarrow \{b,c\}$

$\{b\} \rightarrow \{c\}$

$\{a\} \rightarrow \{b\}$

$\{a,b\} \rightarrow \{c\}$

$\{a,c\} \rightarrow \{d\}$

Построить неприводимое множество ФЗ

# Пример

Используем аксиомы

$$\{a\} \rightarrow \{b,c\} \quad \longrightarrow \quad \begin{array}{l} \{a\} \rightarrow \{b\} \\ \{a\} \rightarrow \{c\} \end{array}$$

$$\{b\} \rightarrow \{c\}$$

$$\{a\} \rightarrow \{b\}$$

$$\{a,b\} \rightarrow \{c\}$$

$$\{a,c\} \rightarrow \{d\}$$

# Пример

Используем аксиомы

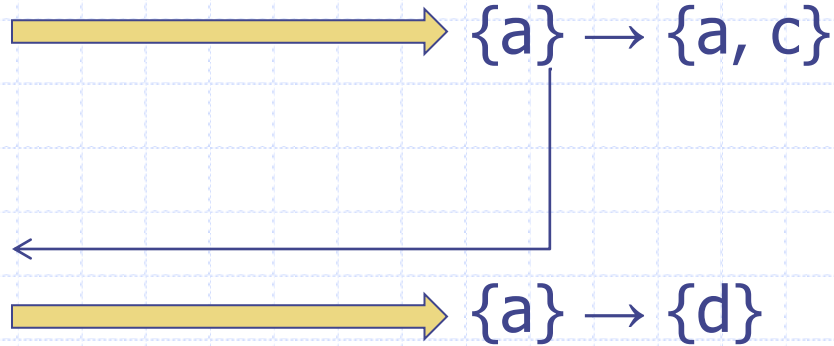
$$\{a\} \rightarrow \{b\}$$

$$\{a\} \rightarrow \{c\}$$

$$\{b\} \rightarrow \{c\}$$

$$\{a,b\} \rightarrow \{c\}$$

$$\{a,c\} \rightarrow \{d\}$$



# Пример

Используем аксиомы

$$\{a\} \rightarrow \{b\}$$

$$\{a\} \rightarrow \{c\}$$

$$\{b\} \rightarrow \{c\}$$

$$\{a,b\} \rightarrow \{c\} \quad \longrightarrow \quad \text{избыточная}$$

$$\{a\} \rightarrow \{d\}$$

# Пример

Используем аксиомы

$$\{a\} \rightarrow \{b\}$$

$$\{a\} \rightarrow \{c\}$$

$$\{b\} \rightarrow \{c\}$$

$$\{a\} \rightarrow \{d\}$$



избыточная  
(транзитивная)

# Пример

Неприводимое множество зависимостей

$\{a\} \rightarrow \{b\}$

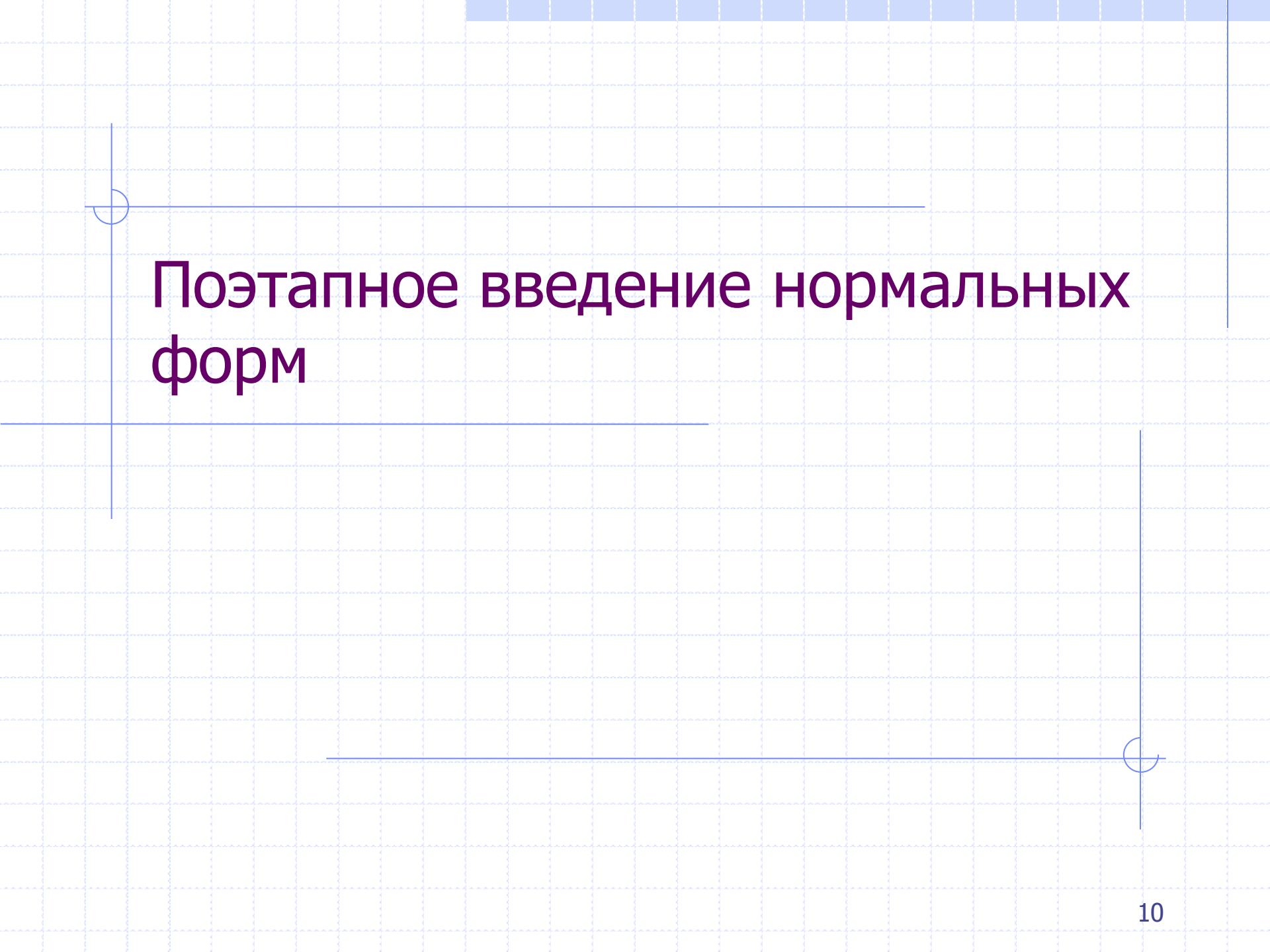
$\{b\} \rightarrow \{c\}$

$\{a\} \rightarrow \{d\}$



# Нормальная форма Бойса-Кодда

- ◆ Переменная отношения находится в НФБК  $t^3$ , когда каждая нетривиальная и неприводимая слева ФЗ имеет в качестве своего детерминанта возможный ключ
- ◆ Для большинства отношений НФБК гарантирует отсутствие аномалий (проблем) вставки, редактирования и удаления



# Поэтапное введение нормальных форм

# 1НФ

- ◆ Отношение находится в первой нормальной форме (1НФ), если каждый его элемент имеет атомарное значение
- ◆ Для отношения в первой нормальной форме может быть определен набор атрибутов (возможный ключ), однозначно определяющий каждый кортеж отношения
- ◆ Среди возможных ключей выбирается один - первичный ключ

# 2НФ

- ◆ Отношение находится во второй нормальной форме, если оно находится в первой нормальной форме и не существует функциональных зависимостей неключевых атрибутов от части возможного ключа

# 3NF

- ◆ Отношение находится в третьей нормальной форме, если оно находится во второй нормальной форме и не существует транзитивных функциональных зависимостей от ключевых атрибутов

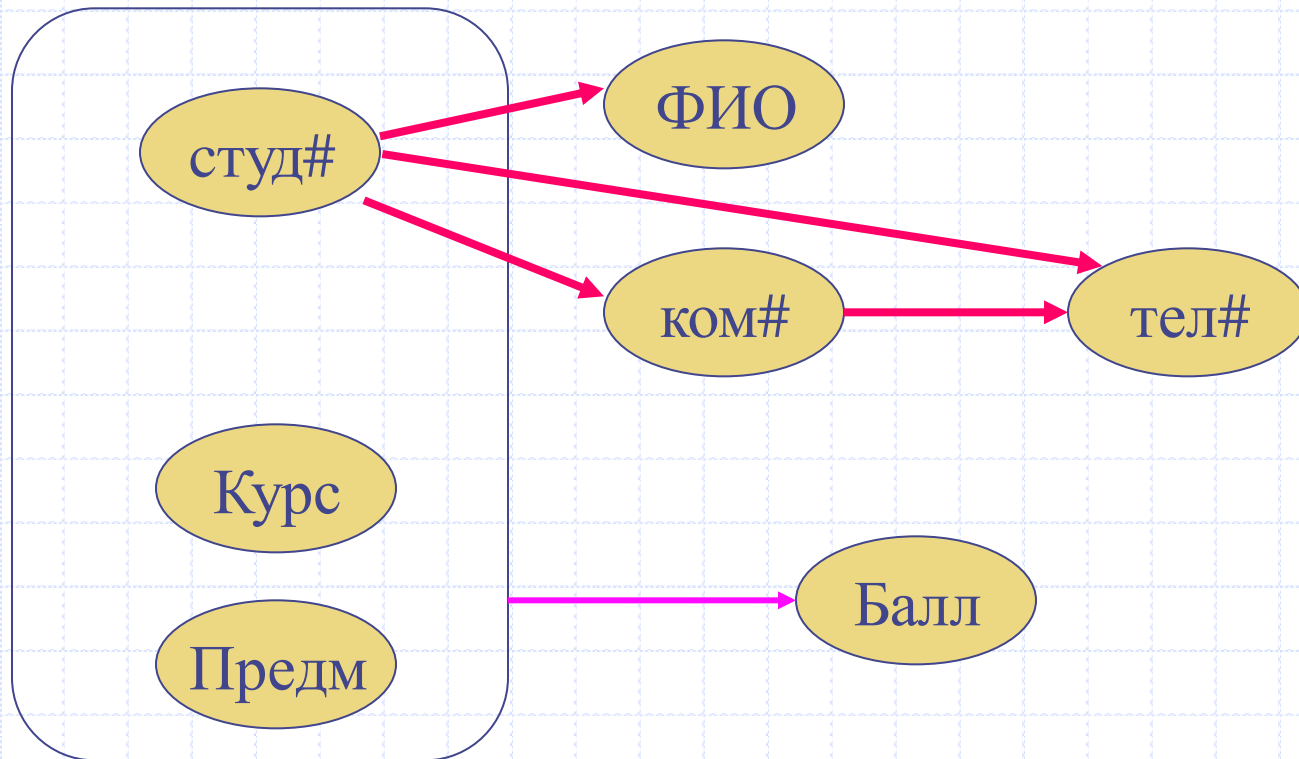
# БКНФ (усиленная ЗНФ)

- ◆ Отношение находится в нормальной форме Бойса-Кодда, если каждый детерминант отношения является возможным ключом

# Более высокие НФ

- ◆ 4НФ
- ◆ 5НФ
- ◆ ДКНФ (доменно-ключевая)
- ◆ 6НФ

# Схема ФЗ для отношения «Тьютор»





# Проектирование с помощью декомпозиции без потерь

- 1) Разработка отношения в 1НФ
- 2) Определение возможных ключей, выбор первичного ключа
- 3) Определение всех ФЗ между атрибутами
- 4) Проверка, находится ли отношение в НФБК
- 5) Если нет, то разбиение отношения на два новых и повторение для них шагов 2-5

# Правило разбиения отношения для алгоритма декомпозиции без потерь

Пусть отношение  $R(\underline{A}, B, C, D, E, \dots)$  не находится в НФБК

Пусть функциональная зависимость  $C \rightarrow D$  нарушает требование НФБК

Создаем два новых отношения

$R1(\underline{A}, B, C, E, \dots)$

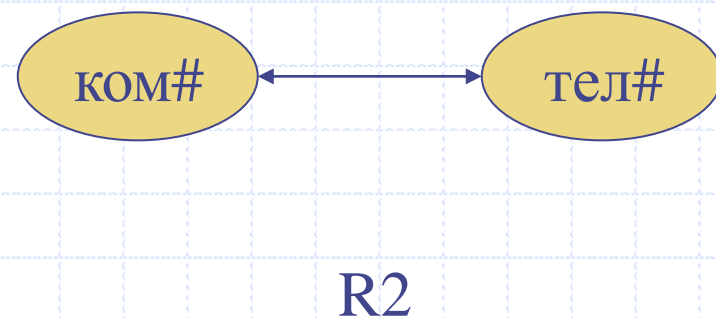
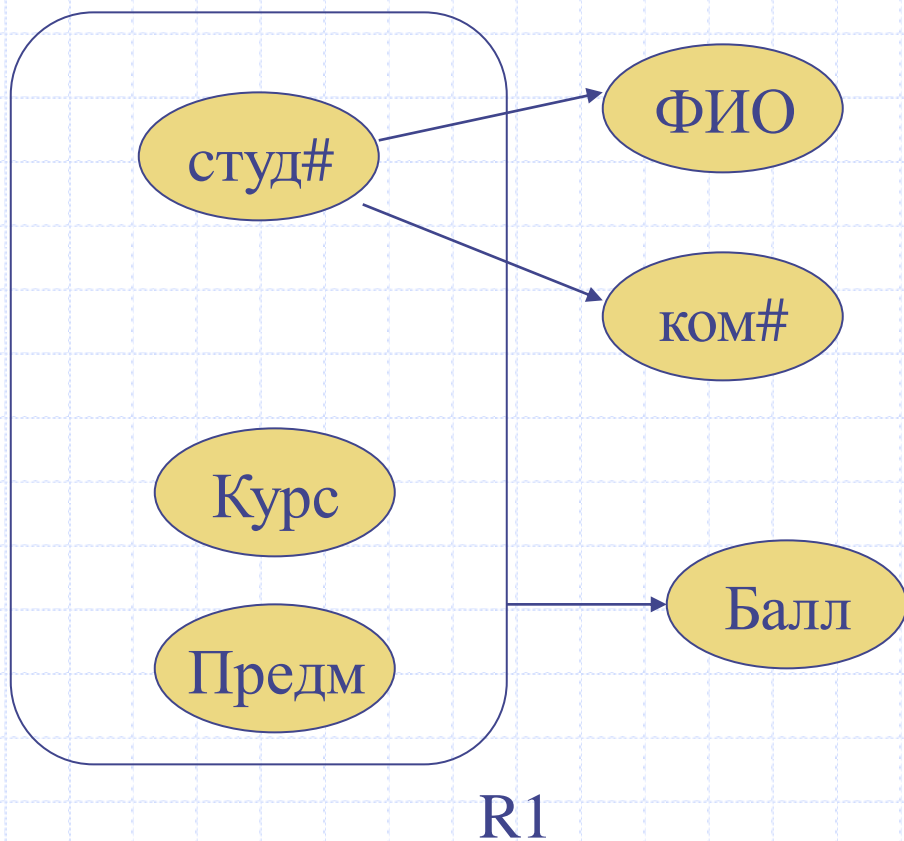
$R2(\underline{C}, D)$

Атрибут  $C$  в отношении  $R2$  является первичным ключом, а в  $R1$  реализует целостность по связям (ссылочная целостность)

# Декомпозиция без потерь

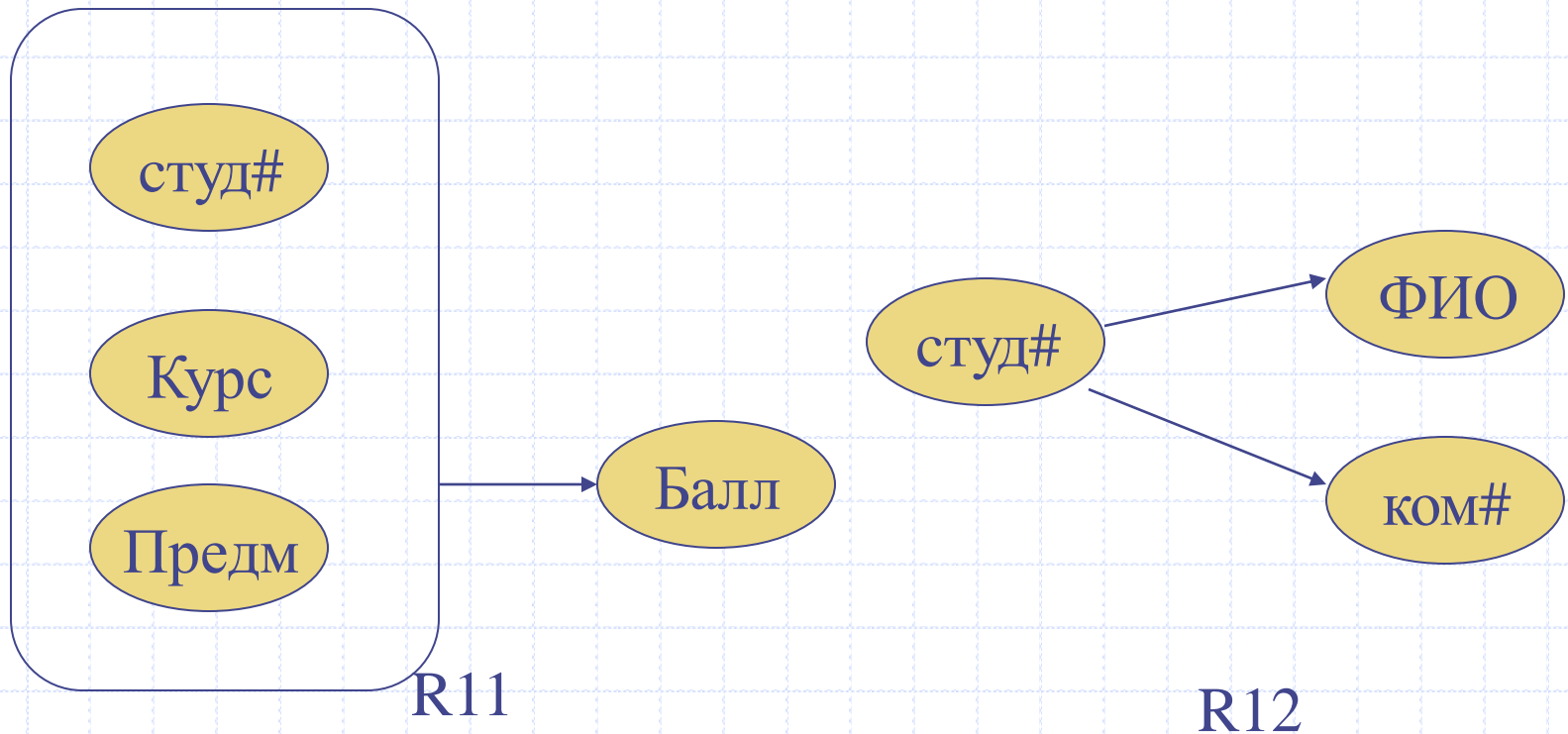
$R = R1 \text{ join } R2 \text{ on } R1.C = R2.C$

# Декомпозиция отношения «Тьютор»



Избавились от транзитивной зависимости

# Декомпозиция отношения «Тьютор»



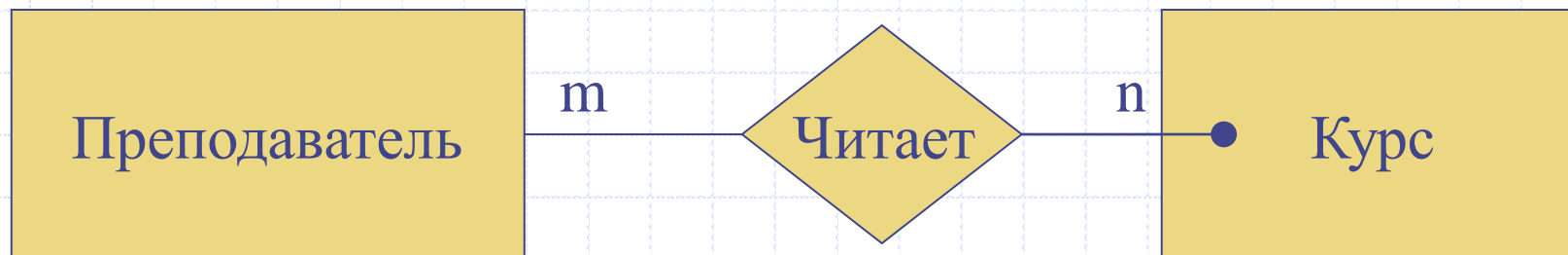
Исключили зависимость от части первичного ключа

# Проектирование на основе модели «Сущность-Связь»

## ◆ Основные понятия

- Сущность
- Атрибут
- Связь
- Степень связи (один к одному, один ко многим, многие ко многим)
- Принадлежность (обязательность) связи для каждой из сущностей

# Пример ER-диаграммы

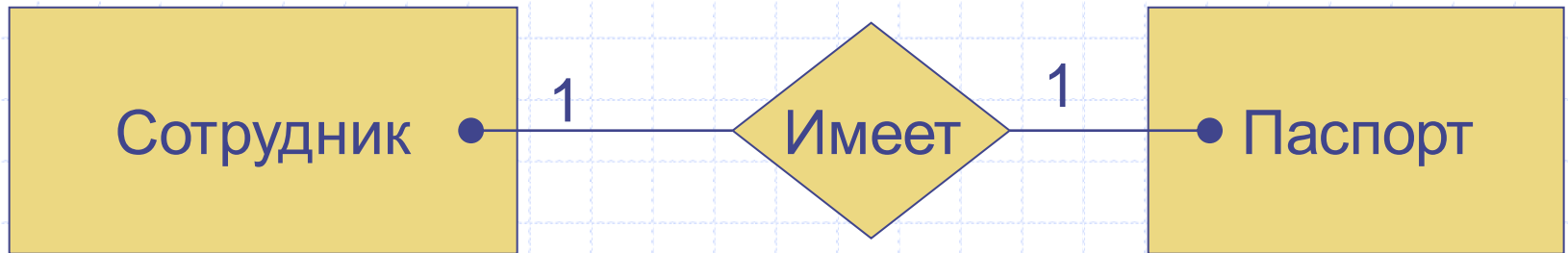


# Правила проектирования для связи 1:1

- ◆ Если степень бинарной связи 1:1 и связи для обеих сущностей являются обязательными, то требуется только одно отношение. Первичным ключом этого отношения может быть первичный ключ любой из сущностей



# Пример



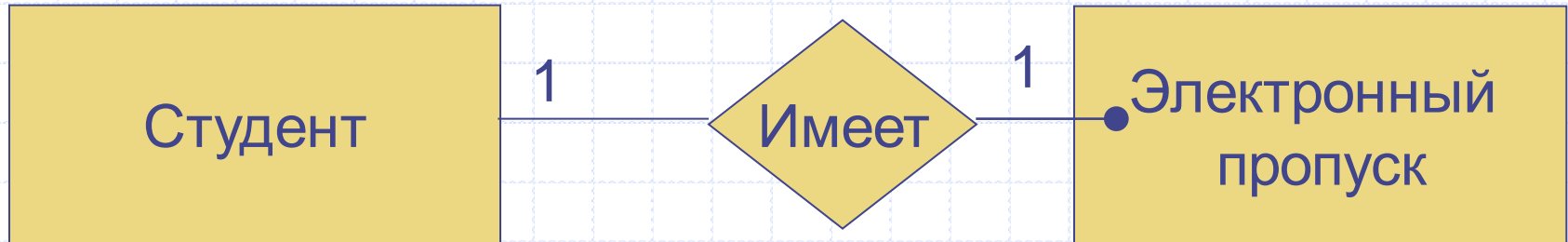
Сотрудник(#пасп, дата, выдан, ФИО,  
пол, датаРожд, ...)

# Правила проектирования для связи 1:1

- ◆ Если степень бинарной связи 1:1 и связь для одной из сущностей является обязательной, а для другой – не обязательной, то необходимо использовать два отношения.

- ◆ Каждая из сущностей представляется отношением, при этом ключ сущности служит ключом отношения.
- ◆ Кроме того, ключ сущности, для которой связь является не обязательной, добавляется в качестве атрибута в отношение, представляющее сущность с обязательной связью.
- ◆ В это же отношение включаются атрибуты связи, если они есть.

# Пример



Студент (#студБилета, ФИО, ....)

ЭлКарта(#пропуска, #студБилета, срокДейств,....)

# Правила проектирования для связи 1:1

- ◆ Если степень бинарной связи 1:1 и связь ни для одной из сущностей не является обязательной, то необходимо использовать три отношения.

- ◆ Каждая из сущностей представляется одним отношением, первичный ключ которого, соответствует ключу сущности.
- ◆ Третье отношение представляет связь. В это отношение в качестве атрибутов включаются ключи обеих сущностей.

# Пример



Сотрудник (ИНН, ФИО должность, ...)

Кафедра (НазвКаф, #комнаты, ...)

Заведующий (ИНН, НазвКаф,  
датаИзбрания, . . .)

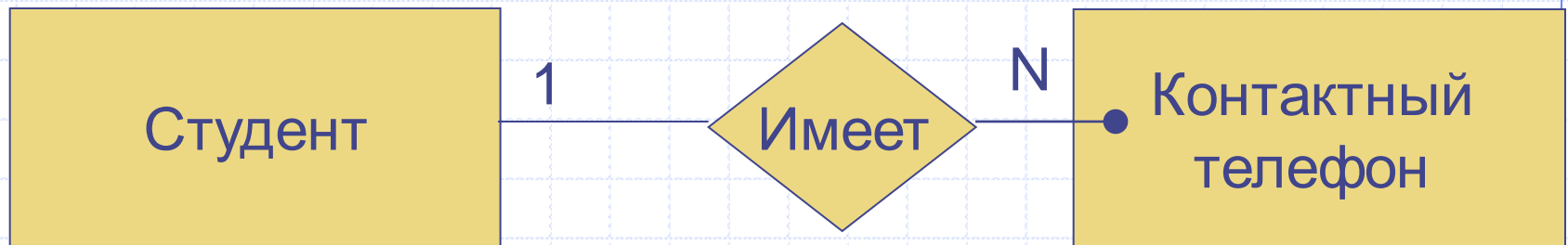
# Правила проектирования для связи 1:N

- ◆ Если степень бинарной связи 1:N и множественная связь является обязательной, то достаточно использовать два отношения.



- ◆ Каждая из сущностей представляется отношением, при этом ключ сущности служит ключом отношения.
- ◆ Кроме того, ключ сущности, имеющей степень связи 1, добавляется в качестве атрибута в отношение, представляющее сущность с множественной связью.
- ◆ В это же отношение включаются атрибуты связи, если они есть.

# Пример



Студент (#студБилета, ФИО, ....)

КонтТел(#тел, **#студБилета**, тип...)

# Правила проектирования для связи 1:N

- ◆ Если степень бинарной связи 1:N и множественная связь не является обязательной, то необходимо использовать три отношения.

- ◆ Каждая из сущностей представляется одним отношением, первичный ключ которого, соответствует ключу сущности.
- ◆ Третье отношение представляет связь.
- ◆ В это отношение в качестве атрибутов включаются ключи обеих сущностей.

# Пример



Сотрудник(ИНН, ФИО, должность, ...)

Проект (НазвПр, тематика, срок, бюджет, ...)

Руководитель(ИНН, НазвПр, ...)

# Правила проектирования для связи N:M

- ◆ Если степень бинарной связи N:M, то необходимо использовать три отношения.
- ◆ Каждая из сущностей представляется одним отношением, первичный ключ которого, соответствует ключу сущности.
- ◆ Третье отношение представляет связь. В это отношение в качестве атрибутов включаются ключи обеих сущностей.

# Пример



Сотрудник(ИНН, ФИО, должность, ...)

Проект (НазвПр, тематика, срок, бюджет, ...)

Участвует(ИНН, НазвПр, период,  
должность...)

# Перекрывающиеся ключи

$R(s, j, t)$

s - студент

j - предмет

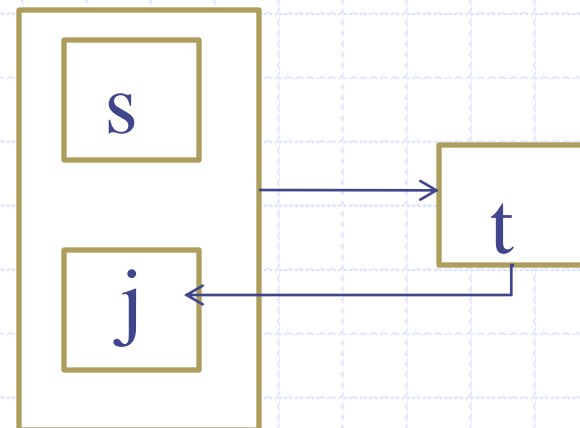
t - преподаватель

Ключи:  $\{s, j\}$  и  $\{s, t\}$

t – детерминант, но не  
ключ

$\{s, j\} \rightarrow \{t\}$

$\{t\} \rightarrow \{j\}$





# Проекция

$R1(s,t) \ R2(t,j)$

в результате потеряна ФЗ  $\{s,j\} \rightarrow \{t\}$

Возможное решение

$R1(t, j) \ R2(s, t, j)$  и упрощение через суррогатный ключ