

The background is a light blue grid. A solid blue horizontal line is positioned above the text, and another solid blue horizontal line is positioned below it. A vertical blue line is on the left side, and another is on the right side. At the top-left corner, there is a blue L-shaped line with a small semi-circle at the top end. At the bottom-right corner, there is a blue L-shaped line with a small semi-circle at the bottom end.

# Проектирование баз данных

# Цели проектирования

- ◆ Определение всех необходимых данных
- ◆ Определение необходимого набора таблиц (отношений)
- ◆ Определение ограничений

Результат проектирования – схема базы данных

# Как определить какие данные нужны?

- ◆ Моделируем предметную область
- ◆ Рассматриваем известные данные, связи между ними, процессы (бизнес-процессы)
- ◆ Разделяем на подзадачи
- ◆ Используются UML и case-средства

# Избыточность первичных данных

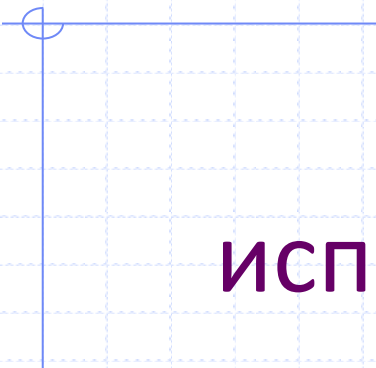
- ◆ Одни и те же данные в разных задачах/подсистемах
- ◆ Дублирование информации в рамках одной задачи
  - Одни и те же данные в разных отношениях
  - Одни и те же данные в одном отношении

# Таблицы (отношения)

- ◆ Часто изначально проектируются на основе уже существующих таблиц
- ◆ В процессе проектирования могут подвергаться преобразованиям
- ◆ Противоречие
  - Разделение одного отношения на несколько более мелких отношений
  - Минимизация количества отношений, чтобы избежать множества join в запросах

# Таблица

- ◆ Все начинается с таблицы (отношения)
- ◆ Целостность таблицы
  - на пересечении строки и столбца стоят атомарные значения
  - все строки различны (существует непустое подмножество атрибутов – возможный ключ, однозначно определяющих каждую строку)



# Проблемы, связанные с использованием минимального количества отношений

# Пример таблицы для базы данных «Тьютор»

Исходная таблица

Это еще не отношение !!!

студ#	ФИО	комн#	тел#	Курс	Предм	Балл
m3215	Иванов И.И.	120DH	2136123	PM16	MA84	87
					AG85	75
					PL83	60
i2345	Петров П.П.	238MP	2345671	IT211	MA84	70
					PL84	85
f1234	Сидоров С.С.	120DH	2136123	AS12	TP12	62



# Пример отношения для базы данных «Тьютор»

студ#	ФИО	комн#	тел#	Курс	Предм	Балл
m3215	Иванов И.И.	120DH	2136123	PM16	MA84	87
m3215	Иванов И.И.	120DH	2136123	PM16	AG85	75
m3215	Иванов И.И.	120DH	2136123	PM16	PL83	60
i2345	Петров П.П.	238MP	2345671	IT211	MA84	70
i2345	Петров П.П.	238MP	2345671	IT211	PL84	85
f1234	Сидоров С.С.	120DH	2136123	AS12	TP12	62

Возможный ключ – {студ#, Курс, Предм}

Чердынцева М.И., ИММиКН ЮФУ

2023

9

# Проблема вставки

студ#	ФИО	КОМН#	тел#	Курс	Предм	Балл
m3215	Иванов И.И.	120DH	2136123	PM16	MA84	87
m3215	Иванов И.И.	120DH	2136123	PM16	AG85	75
m3215	Иванов И.И.	120DH	2136123	PM16	PL83	60
i2345	Петров П.П.	238MP	2345671	IT211	MA84	70
i2345	Петров П.П.	238MP	2345671	IT211	PL84	85
f1234	Сидоров С.С.	120DH	2136123	AS12	TP12	62
m3456	Егоров Е.Е.	121MP	2123341	PM11	PL84	0

```
insert into TUTOR values('m3456','Егоров Е.Е.',  
                          '121MP','2123341','PM11','PL84',0)
```

```
select ФИО from TUTOR where Балл<60
```

# Проблема обновления

студ#	ФИО	комн#	тел#	Курс	Предм	Балл
m3215	Иванов И.И.	120DH	2136123	PM16	MA84	87
m3215	Иванов И.И.	120DH	2136123	PM16	AG85	75
m3215	Иванов И.И.	120DH	2136123	PM16	PL83	60
i2345	Петров П.П.	238MP	2345671	IT211	MA84	70
i2345	Петров П.П.	238MP	2345671	IT211	PL84	85
f1234	Сидоров С.С.	120DH	2177777	AS12	TP12	62

update TUTOR set тел#='2177777' where ФИО='Сидоров С.С.'  
select тел# from TUTOR where комн# = '120DH'

# Проблема удаления

студ#	ФИО	КОМН#	тел#	Курс	Предм	87
m3215	Иванов И.И.	120DH	2136123	PM16	MA84	75
m3215	Иванов И.И.	120DH	2136123	PM16	AG85	60
m3215	Иванов И.И.	120DH	2136123	PM16	PL83	70
i2345	Петров П.П.	238MP	2345671	IT211	MA84	85
i2345	Петров П.П.	238MP	2345671	IT211	PL84	62
f1234	Сидоров С.С.	120DH	2177777	AS12	TP12	87

delete from TUTOR

where ФИО='Сидоров С.С.' and Предм='TP12'

select \* from TUTOR where ФИО='Сидоров С.С.'

Чердынцева М.И., ИММиКН ЮФУ

2023

12

# Как решить проблемы?

- ◆ Может быть помогут NULL
- ◆ Можно более жестко контролировать операции в приложении
- ◆ Выделить часть информации в отдельную таблицу
  - Какую информацию?
  - Достаточно ли будет еще одной таблицы или нужно больше?
  - Возможно ли будет восстановить исходную таблицу после этого?
- ◆ В основе решения – определение свойств отношения и разбиение на большее количество отношений путем декомпозиции без потерь

# Нормализация путем декомпозиции

- ◆ Декомпозиция отношения на два или более отношений с целью разрешения проблем вставки, обновления и удаления данных
- ◆ Декомпозиция без потерь – операция join всегда позволит восстановить исходную таблицу
- ◆ Нормальная форма – способ классификации таблиц на основе содержащихся в них функциональных зависимостей (functional dependencies, FD)

# Функциональные зависимости

- ◆ Функциональная зависимость устанавливает связь между двумя подмножествами атрибутов внутри одной переменной-отношения
- ◆ Виды функциональных зависимостей
  - которые могут быть определены всегда, такие ФЗ можно считать ограничениями целостности
  - которые могут определяться для конкретного момента времени (для конкретных кортежей), могут приводить к ошибкам в проектировании

# Функциональные зависимости

$F: \{A\} \rightarrow \{B\}$



A и B непересекающиеся множества атрибутов

Если имеется ФЗ  $F: \{X\} \rightarrow \{Y\}$ ,

то X – детерминант, Y – зависимая часть



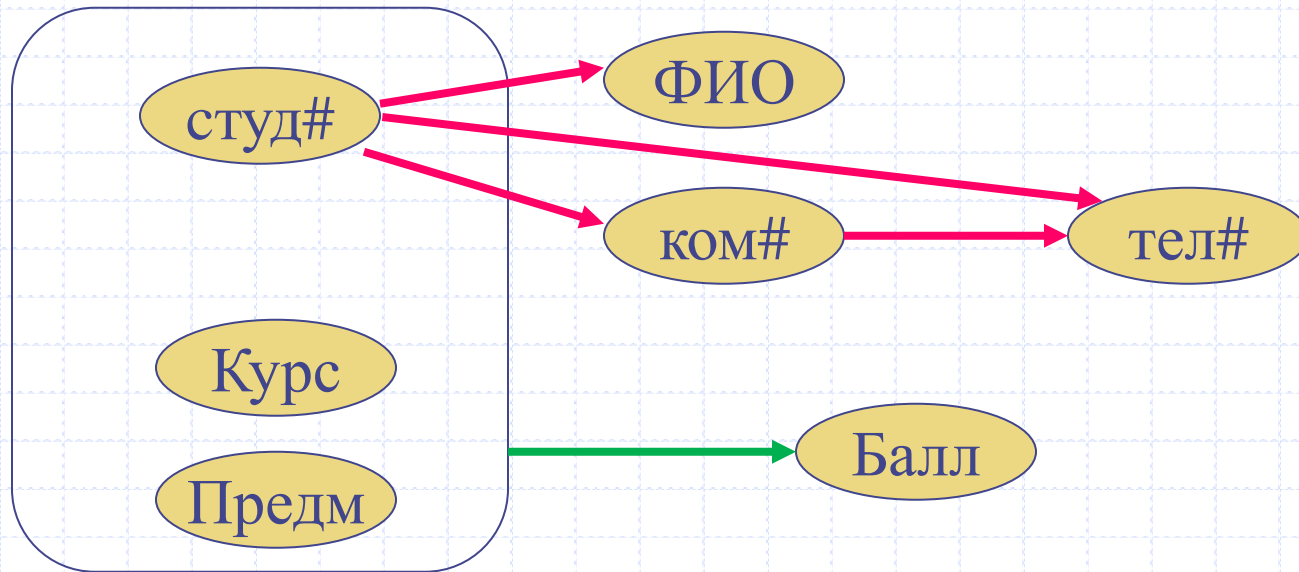
# Возможный ключ

- ◆ Если ФЗ определяет в отношении зависимость всего кортежа от набора атрибутов (т.е. гарантирует основное свойство отношения-уникальность кортежей), тогда детерминант ФЗ является потенциальным или возможным ключом
- ◆ Отношение может иметь несколько потенциальных ключей
- ◆ Отношение всегда имеет один возможный ключ, совпадающий с полным набором атрибутов отношения

# Функциональные зависимости

- ◆ Если  $X$  является потенциальным (возможным) ключом, то все не входящие в  $X$  атрибуты должны быть функционально зависимы от  $X$
- ◆ Если переменная-отношение  $R$  удовлетворяет ФЗ  $F: \{A\} \rightarrow \{B\}$  и  $A$  не является потенциальным ключом, то  $R$  будет характеризоваться некоторой избыточностью, приводящей к проблемам вставки/редактирования/удаления

# Схема ФЗ для отношения «Тьютор»



# Что нужно проверять?

- ◆ Является ли детерминант потенциальным ключом

Пусть  $Z$  – интересующее нас множество атрибутов (проверяемый детерминант)

Обозначим через  $Z^+$  множество атрибутов, которые функционально зависимы от  $Z$  (замыкание множества атрибутов)

$Z$  является возможным (потенциальным) ключом  $R$ , если  $Z^+$  содержит все атрибуты  $R$

# Замыкание множества атрибутов

## Алгоритм

$$Z^+ = Z$$

повторять пока  $Z^+$  изменяется

для каждой ФЗ  $f:\{X\} \rightarrow \{Y\}$

если  $\{X\} \in Z^+$ , то добавить  $\{Y\}$

$$Z^+ = Z^+ \cup \{Y\}$$

# Пример

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

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

$\{e\} \rightarrow \{c,f\}$

$\{b\} \rightarrow \{e\}$

$\{c,d\} \rightarrow \{e,f\}$

Найти замыкание для множества атрибутов  $\{a,b\}$

# Пример

$$Z^+ = Z = \{a, b\}$$

$$\{a\} \rightarrow \{b, c\} \quad Z^+ = Z^+ \cup \{b, c\} = \{a, b, c\}$$

$$\{e\} \rightarrow \{c, f\}$$

$$\{b\} \rightarrow \{e\} \quad Z^+ = Z^+ \cup \{e\} = \{a, b, c, e\}$$

$$\{c, d\} \rightarrow \{e, f\}$$

$$Z^+ = \{a, b, c, e\}$$

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

$$\{e\} \rightarrow \{c, f\} \quad Z^+ = Z^+ \cup \{c, f\} = \{a, b, c, e, f\}$$

$$\{b\} \rightarrow \{e\}$$

$$\{c, d\} \rightarrow \{e, f\}$$

# Пример

$$Z^+ = \{a, b, c, e, f\}$$

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

$$\{e\} \rightarrow \{c, f\}$$

$$\{b\} \rightarrow \{e\}$$

$$\{c, d\} \rightarrow \{e, f\}$$

замыкание  $Z^+ = \{a, b, c, e, f\}$

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

$$Z^+ \neq R$$

значит,  $\{a, b\}$  не является  
ВОЗМОЖНЫМ КЛЮХОМ

Показать, что возможным  
ключом является

$$\{a, d\}$$



# 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 слева

# 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 Армстронга

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

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

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

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

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



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

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

# Пример

Используем аксиомы

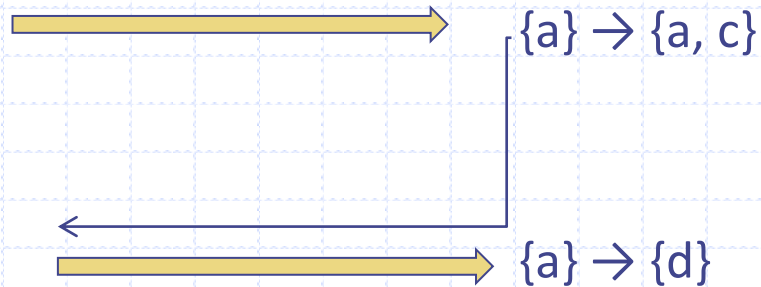
$$\{a\} \rightarrow \{b\}$$

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

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

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

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



# Пример

Используем аксиомы

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

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

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

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

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


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

избыточная

# Пример

Используем аксиомы

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

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

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

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



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

избыточная (транзитивная)

# Пример

Неприводимое множество зависимостей

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

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

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

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

Можно показать, что  $\{a\}$  – потенциальный ключ

# Нормальная форма Бойса-Кодда

- ◆ Переменная отношения находится в НФБК тогда и только тогда, 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



# Нормальная форма

Свойство отношения в реляционной модели данных, характеризующее его с точки зрения избыточности, потенциально приводящей к логически ошибочным результатам выборки или изменения данных.

Нормальная форма определяется как совокупность требований, которым должно удовлетворять отношение.

# Поэтапное введение нормальных форм

1НФ

2НФ

3НФ

НФБК(усиленная 3НФ)

# 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

# НФБК (усиленная ЗНФ)

- ◆ Переменная отношения находится в НФБК тогда и только тогда, когда каждая нетривиальная и неприводимая слева ФЗ имеет в качестве своего детерминанта возможный ключ

# Более высокие НФ

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

# 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

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

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

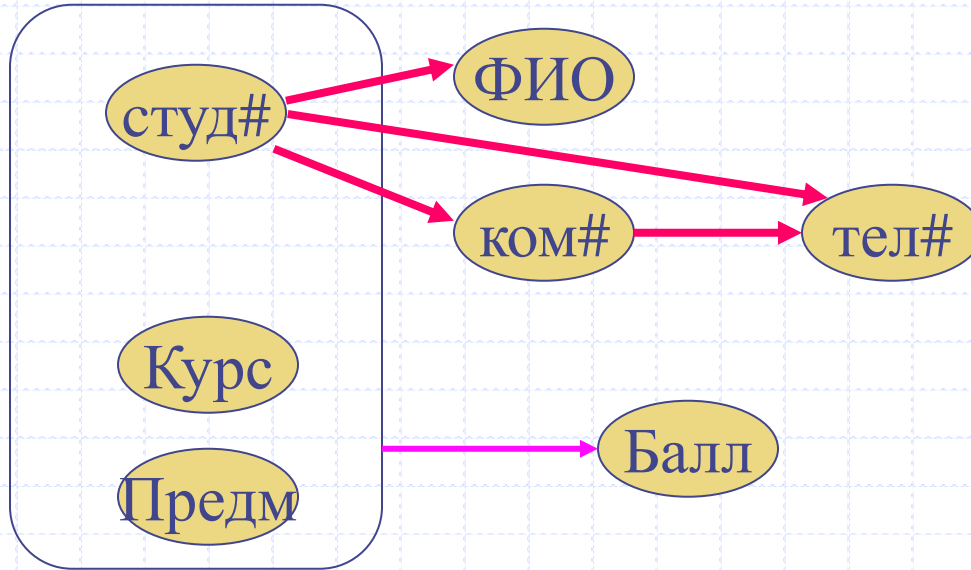
Атрибут C в отношении R2 является первичным ключом, а в R1 реализует целостность по связям (ссылочная целостность)

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

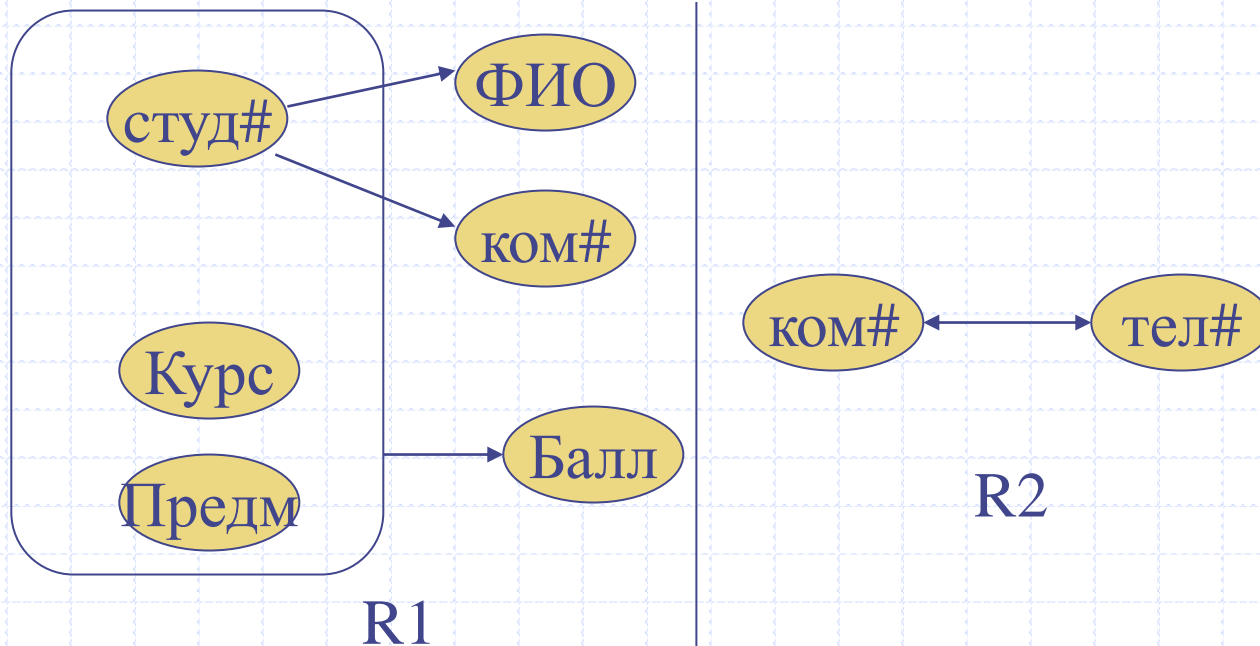
R может быть реализована через представление



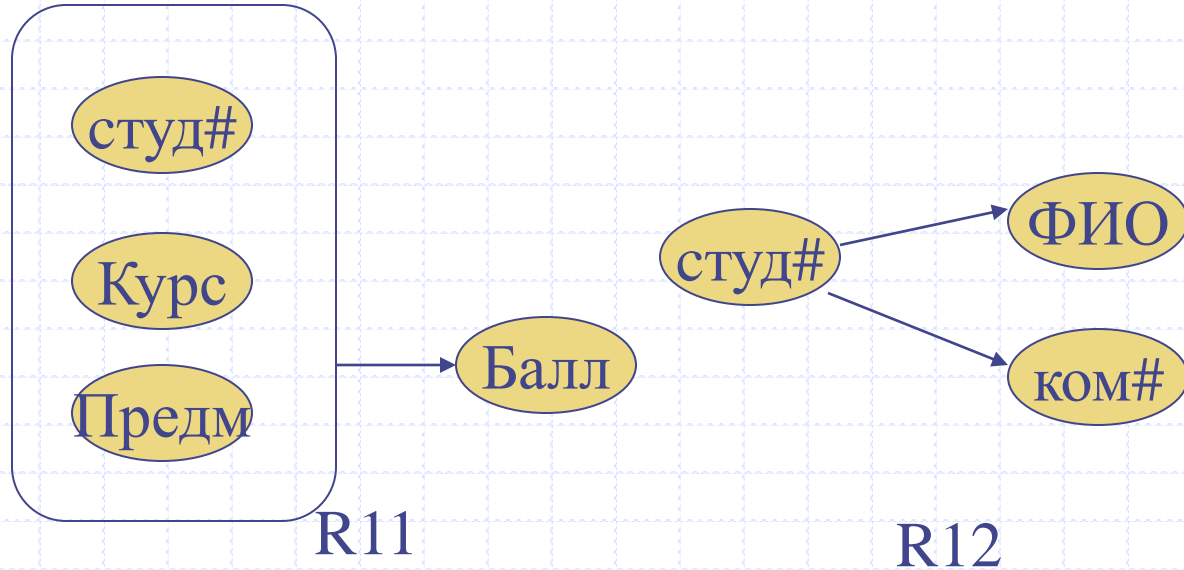
## Схема ФЗ для отношения «Тьютор»



# Декомпозиция отношения «Тьютор»



## Декомпозиция отношения «Тьютор»

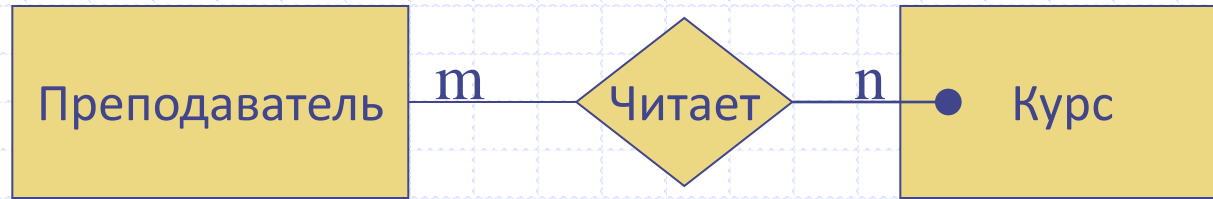


# Проектирование на основе модели «Сущность-Связь»

## ◆ Основные понятия

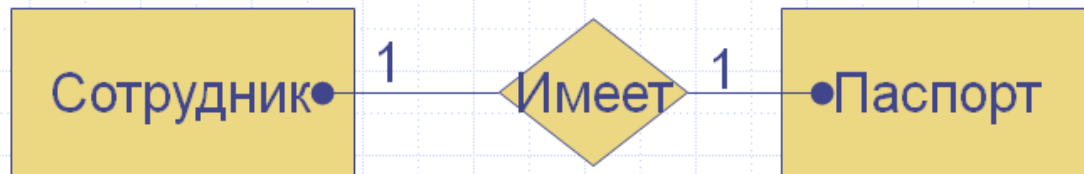
- Сущность
- Атрибут
- Связь
- Степень связи (один к одному, один ко многим, многие ко многим)
- Принадлежность (обязательность) связи для каждой из сущностей

# Пример ER-диаграммы для бинарной связи



# 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

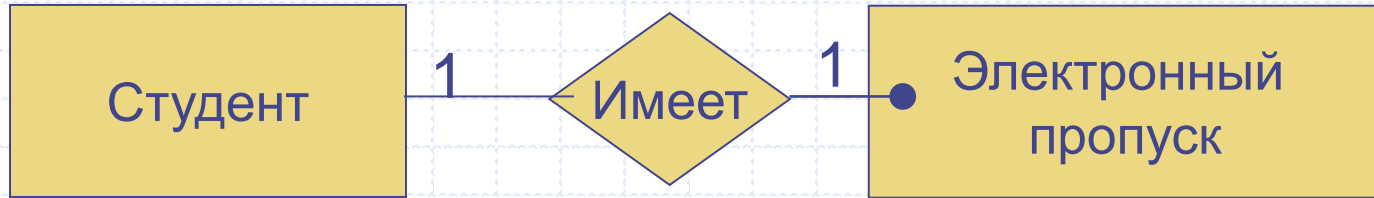


Сотрудник(#пасп, дата, выдан, ФИО, пол, датаРожд, ...)

# 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, ...)

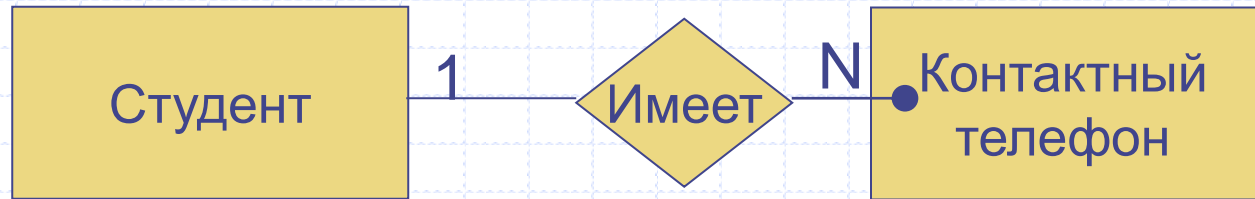
Кафедра (НазвКаф, #комнаты, ...)

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. ...)



# Примеры разработки схемы БД

# Учет товара на складе



ТОВАР (номенклатура, тип, ....)

СКЛАД (название, город, тип....)

ТОВАР\_НА\_СКЛАДЕ (номенклатура, название, количество)

# Справочники

ТОВАР (код товара, номенклатура, тип, ....)

СКЛАД (код склада, название, *код\_города*,.....)

ГОРОД (код города, название, ...)

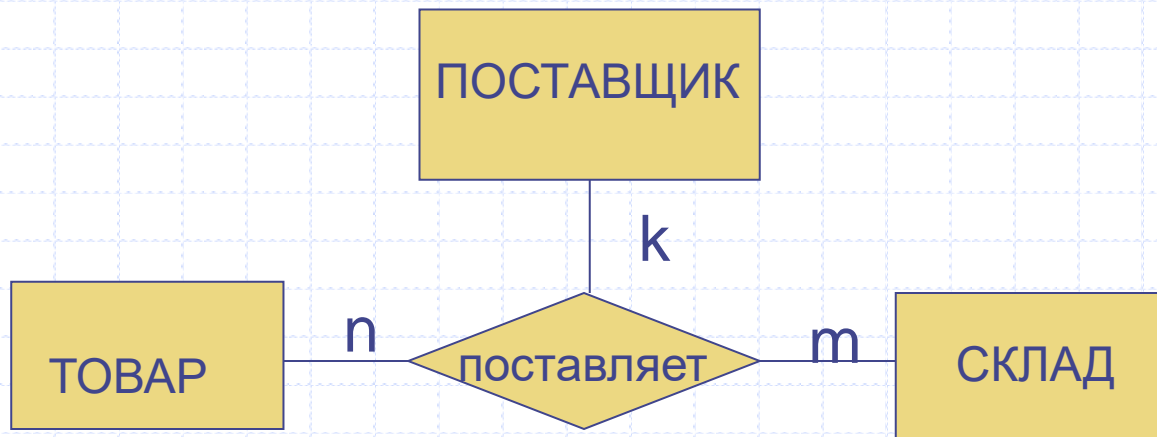
## Таблица связи

ТОВАР\_НА\_СКЛАДЕ (*код\_склада*, *код\_товара*, количество)

## Упрощение первичного ключа

ТОВАР\_НА\_СКЛАДЕ (код остатков, *код\_склада*, *код\_товара*, количество)

# Случай не бинарной связи



ТОВАР (код товара, номенклатура, тип, ....)

СКЛАД (код склада, название, *код\_города*,....)

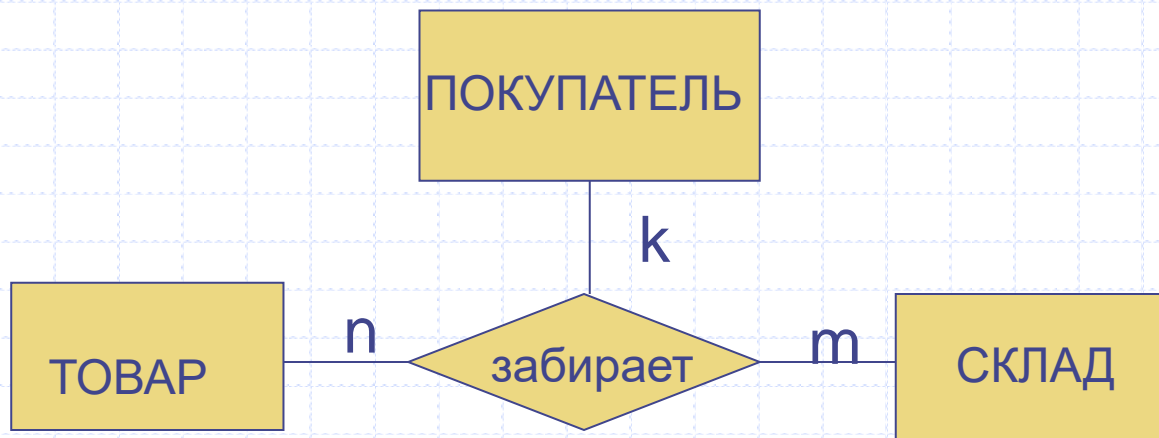
### Добавляем справочник

ПОСТАВЩИК(код поставщика, название, *код\_города*, ...)

### Таблица связи

ПОСТАВКА (код поставки, *код\_товара*, *код\_поставщика*, *код\_склада*,  
дата\_поставки, количество, цена)

# Роли

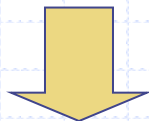


# Роли

ПОКУПАТЕЛЬ (код покупателя, название, код\_города, . . .)

ПОКУПКА (код покупки, код\_товара, код\_покупателя, код\_склада, дата\_покупки, количество, цена)

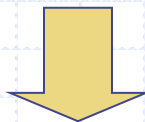
Обобщаем сущности «ПОСТАВЩИК» и «ПОКУПАТЕЛЬ»



КОНТРАГЕНТ (код контрагента, название, код\_города, . . .)

# Роли

Обобщаем связи «ПОСТАВКА» и «ПОКУПКА»



ОПЕРАЦИЯ (код операции, *код\_товара*, *код\_контрагента*, *код\_склада*,  
дата\_операции, *тип операции(приход/расход)*,  
количество, цена)



# Полный список таблиц

ТОВАР (код товара, номенклатура, ед.измерения)

СКЛАД (код склада, название, город)

ТОВАР\_НА\_СКЛАДЕ (код склада, код товара, количество)

КОНТРАГЕНТ (код контрагента, название, город, телефон)

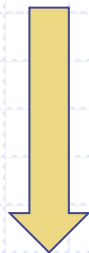
ОПЕРАЦИЯ (код операции, код товара, код контрагента, код склада, дата\_операции, тип операции(приход/расход), количество, цена)

*Для упрощения избавились от таблицы-справочника городов*

# Взаимосвязи

ОПЕРАЦИЯ (код операции, *код\_товара*, *код\_контрагента*, *код\_склада*,  
дата\_операции, тип операции(приход/расход), количество, цена)

изменение (приход)



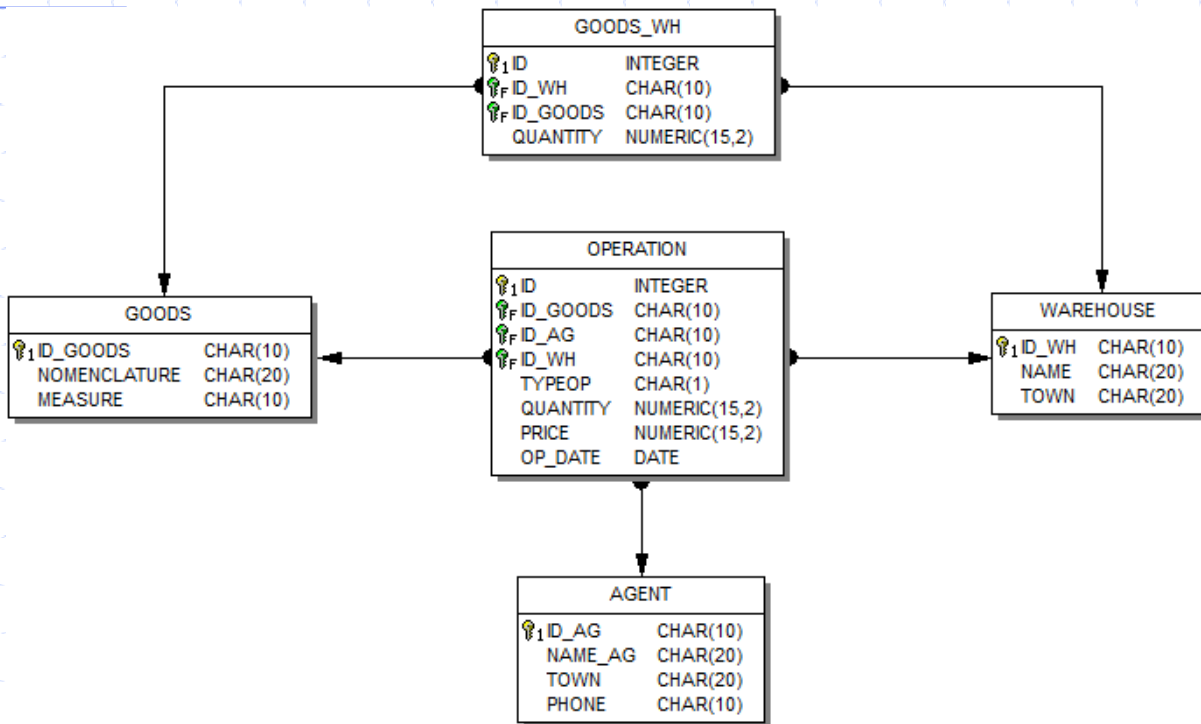
проверка/изменение (расход)



ТОВАР\_НА\_СКЛАДЕ (*код склада*, *код товара*, количество)

Можно реализовать процедурой или триггером

# ER - диаграмма




# Документы в схеме БД

## Накладная (приходная/расходная)

Новая товарная накладная

\* Номер  По договору


\* Дата создания  По счету

\* Имя клиента    
ИНН: 5256063521; КПП: 775001002; ЯСХ:  
26002332250091003124; Банк: ОАО  
\"ВОСТОСИБТРАНСКОМБАНК\"

Основание  Номер

Добавить НДС

Наименование	Ед. изм.	К-во	Цена	Сумма	
Кресло-мешок \"Lite Plus\"	шт	4	2300	9200	✗
Кресло-мешок \"Big Plus\"	шт	2	4000	8000	✗

 Добавить новую строку

Итого: 17 200 руб.

## Накладная (приходная/расходная)



НАКЛАДНАЯ (код накладной, номер, дата, тип, *код контрагента*,  
код\_склада, ..., сумма)

СТРОКА\_НАКЛАДНОЙ(код строки, *код накладной*, *код товара*,  
количество, цена)

# Взаимосвязи

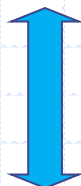
НАКЛАДНАЯ (код накладной, номер, дата, тип, *код\_контрагента, код\_склада*, ..., сумма)

СТРОКА\_НАКЛАДНОЙ(код строки, *код\_накладной, код\_товара*, количество, цена)

исполнение

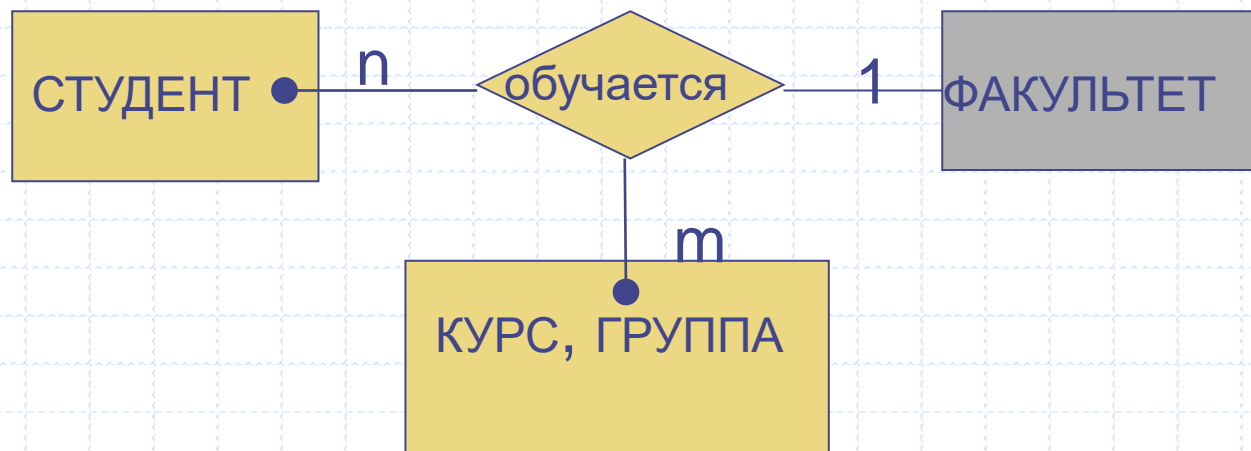


контроль



ОПЕРАЦИЯ (код операции, *код\_товара, код\_контрагента, код\_склада*, дата\_операции, тип операции(приход/расход), количество, цена, *код\_строки\_накладной*)

# Исторические данные

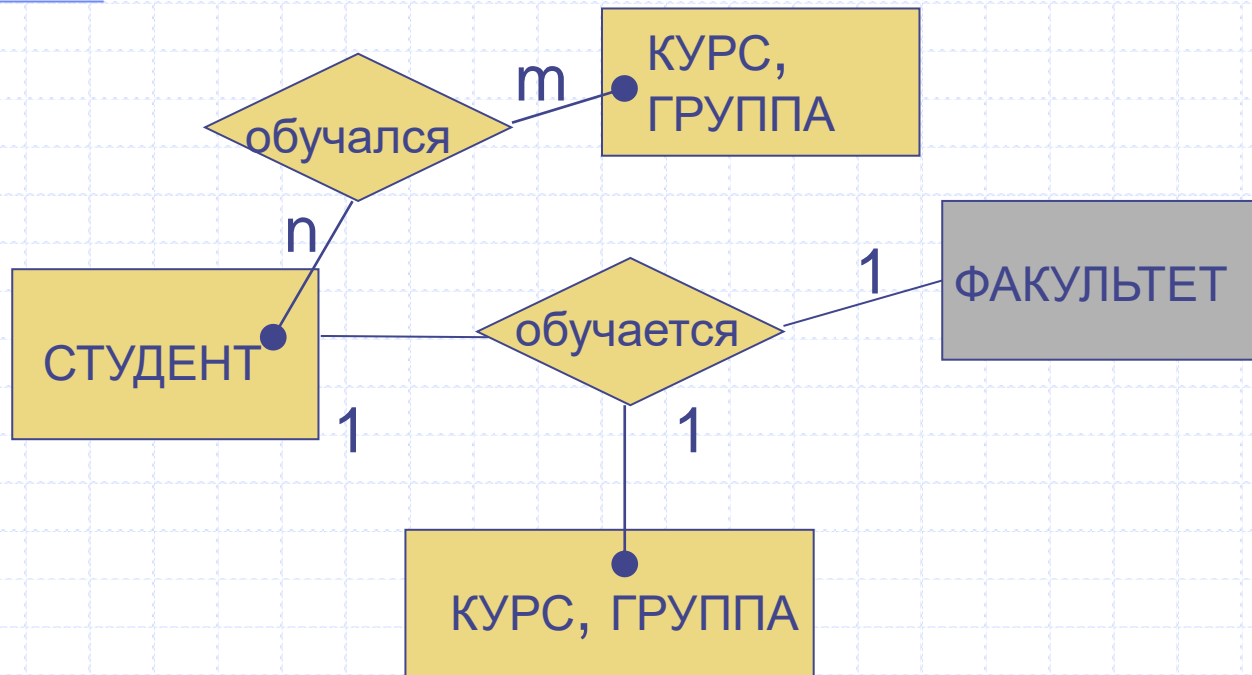


СТУДЕНТ(код студента, ФИО, *код\_факультета*, вид\_обучения, личные данные,...)

ОБУЧЕНИЕ\_ИСТОРИЯ (код, *код\_студента*, год, курс, группа, семестр, статус)

Трудно получать текущие сведения, их можно выделить в отдельную сущность





СТУДЕНТ(код студента, ФИО, *код\_факультета*, вид\_обучения,  
личные данные,...)

ОБУЧЕНИЕ(код, *код\_студента*, курс, группа, статус)

ОБУЧЕНИЕ\_ИСТОРИЯ (код, *код\_студента*, год, курс, группа, семестр,  
статус, дата\_изменения)



# ПРИМЕРЫ НОРМАЛИЗАЦИИ

Чердынцева М.И., ИММиКН  
ЮФУ, 2023

# Проверка отношения (таблицы)

Расписание (день недели, время, группа, предмет, аудитория, количество мест, оборудование)

ФЗ

{день недели, время, группа} → {предмет}

{день недели, время, группа} → {аудитория}

{аудитория} → {количество мест, оборудование}

аудитория – детерминант, но не ключ

# Декомпозиция

Аудитория (аудитория#, количество мест, оборудование)

Расписание\_изм (день недели, время, группа, предмет, аудитория#)

Расписание = Расписание\_изм JOIN Аудитория  
USING (аудитория#)

# Модель отношения

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

F:

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

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

$\{a, b\}$  – потенциальный ключ (можно показать)

$\{a\} \rightarrow \{d\}$  детерминант не является возможным ключом

# Декомпозиция модели

R (a, b, c, d)

Ra (a, d) a – первичный ключ

R1 (a, b, c) {a, b} – первичный ключ  
a – внешний ключ

R = Ra JOIN R1 USING (a)