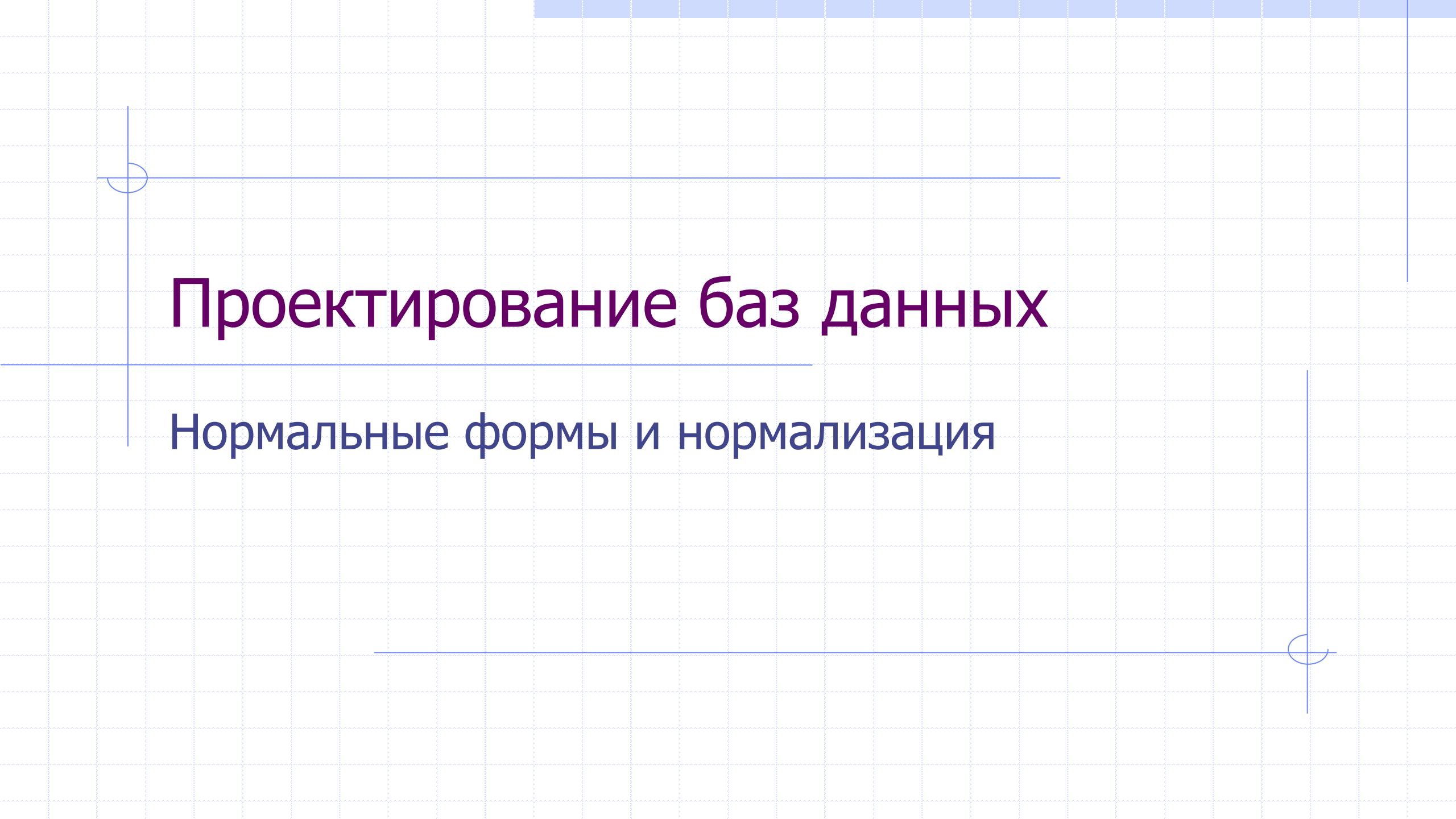
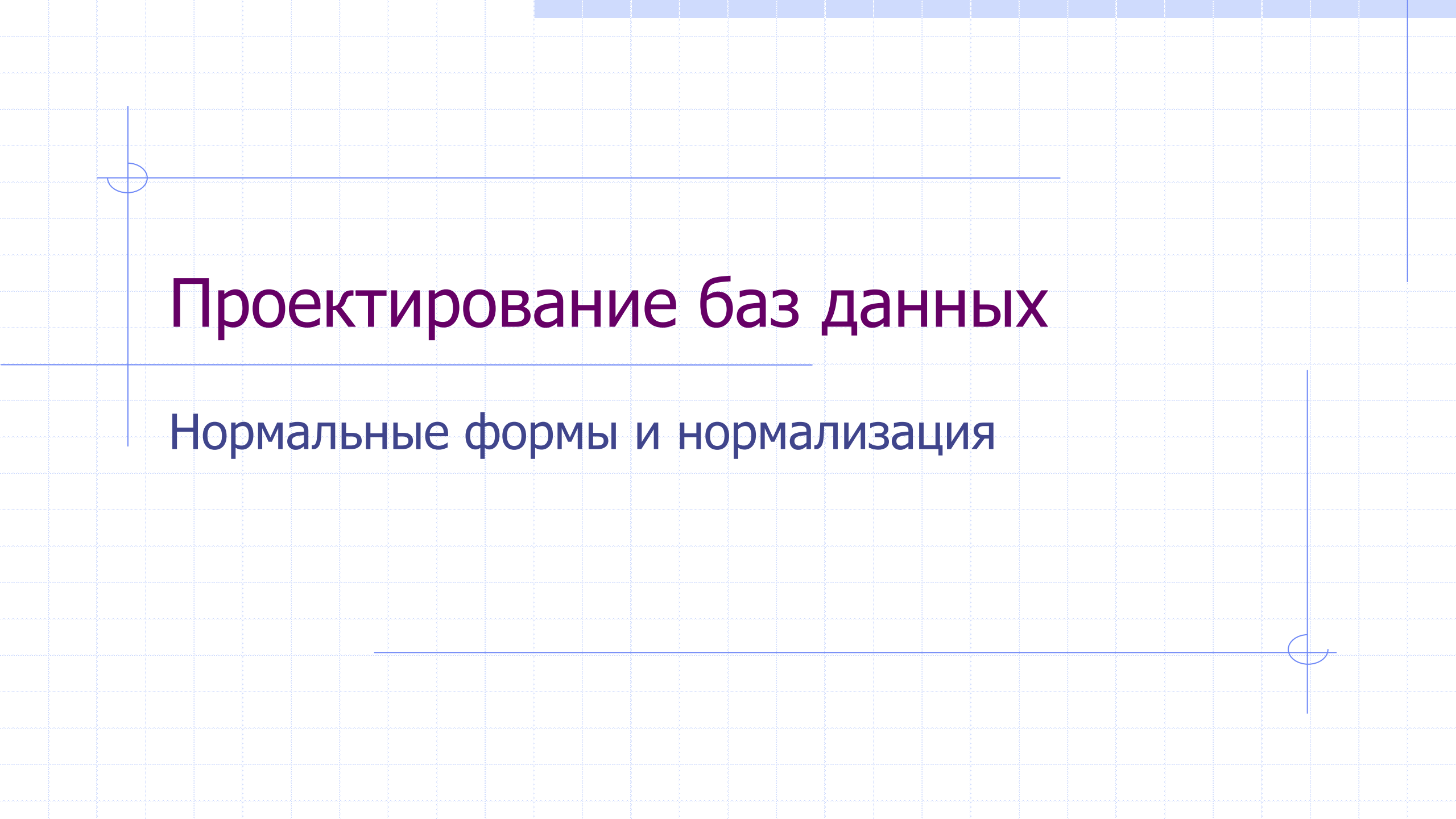




Проектирование баз данных

Нормальные формы и нормализация



Проектирование с помощью декомпозиции без потерь

- ◆ Разработка отношения в 1НФ
- ◆ Определение возможных ключей, выбор первичного ключа
- ◆ Определение всех ФЗ между атрибутами
- ◆ Проверка, находится ли отношение в НФБК
- ◆ 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

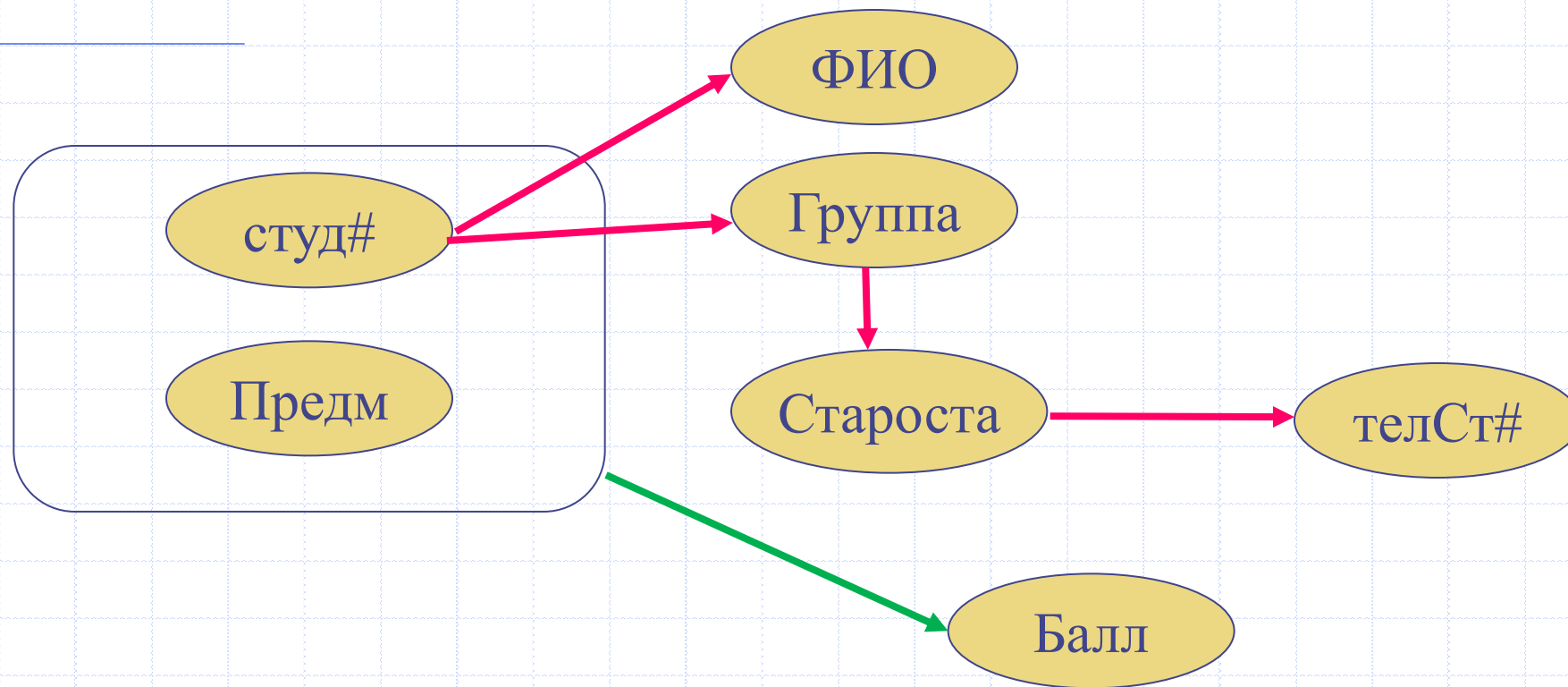
$R1(\underline{A}, B, C, E, \dots)$

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

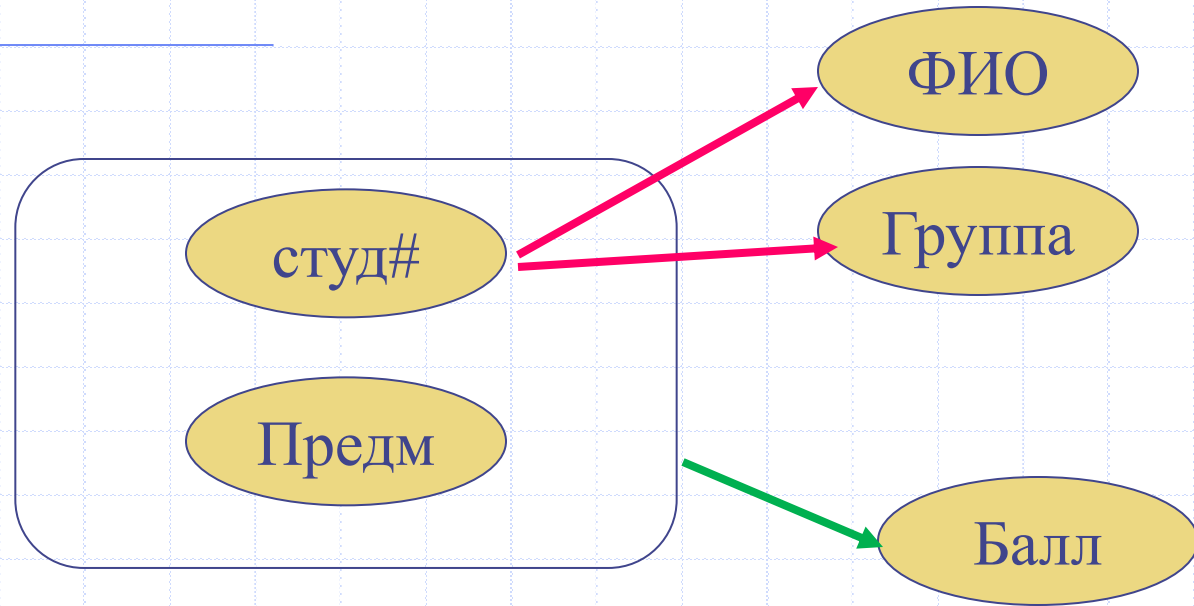
Атрибут (ы) C в отношении $R2$ является первичным ключом, а в $R1$ реализует целостность по связям (ссылочная целостность)

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

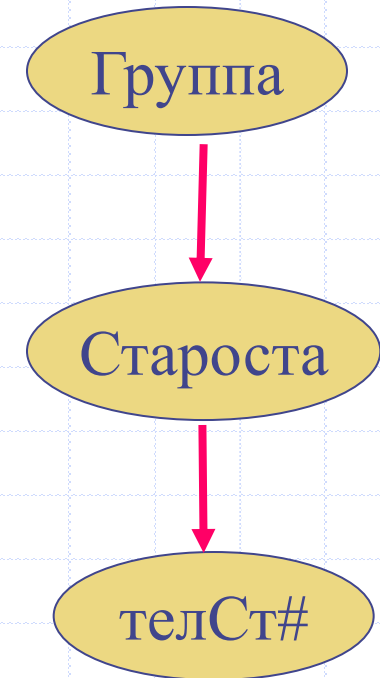
Схема ФЗ для отношения «Руководитель»



Декомпозиция отношения «Руководитель»



R1

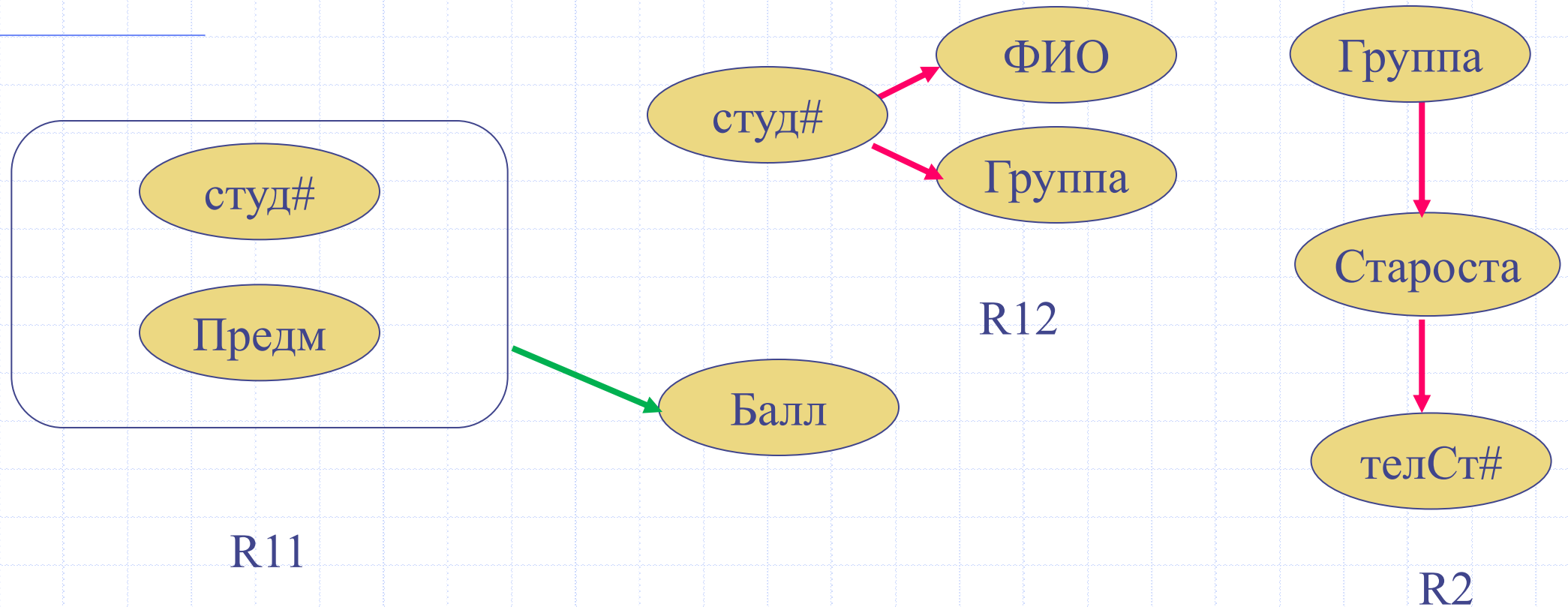


R2

Группа – детерминант, не является возможным ключом
 $Z^+ = \{\text{Группа, Староста, телСт\#}\}$

Избавились от транзитивной зависимости

Декомпозиция отношения «Руководитель»

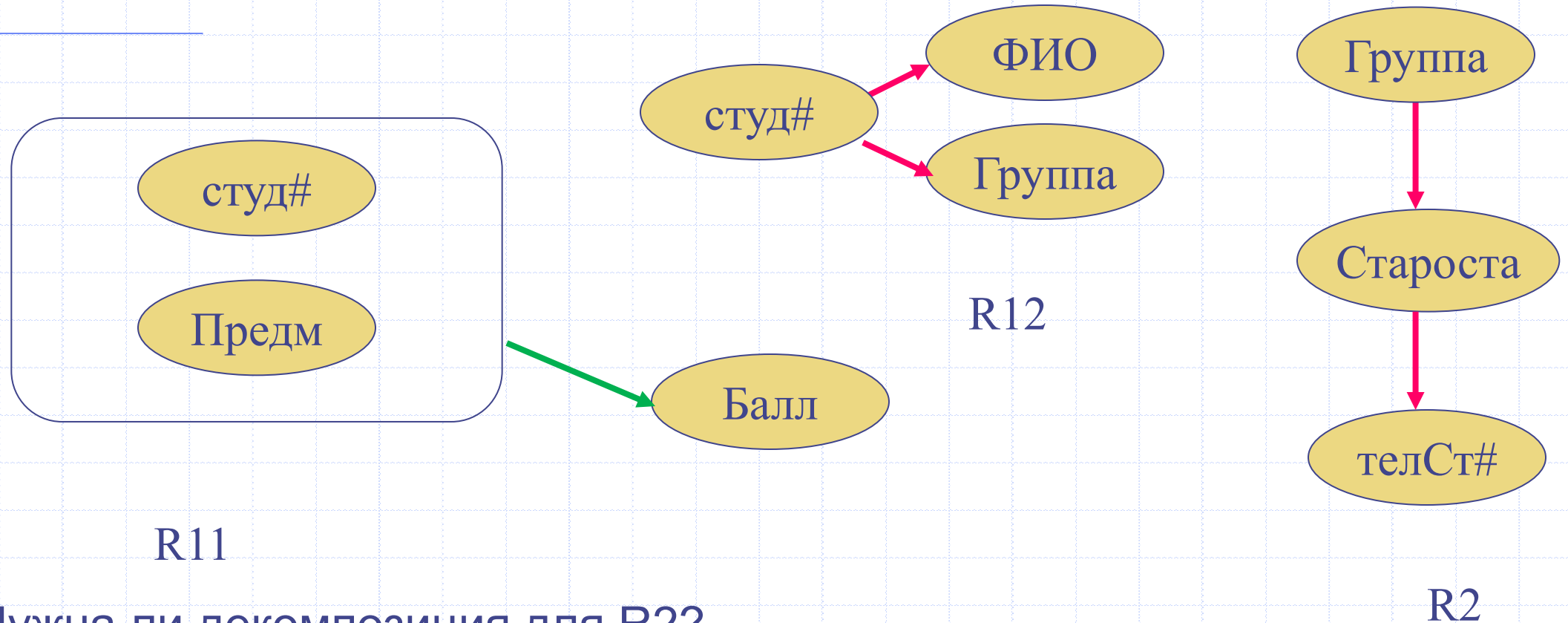


студ# - детерминант в двух ФЗ, не является возможным ключом

$Z^+ = \{\text{студ\#}, \text{ФИО}, \text{Группа}\}$

Исключили зависимость от части ключа

Анализ результата декомпозиции



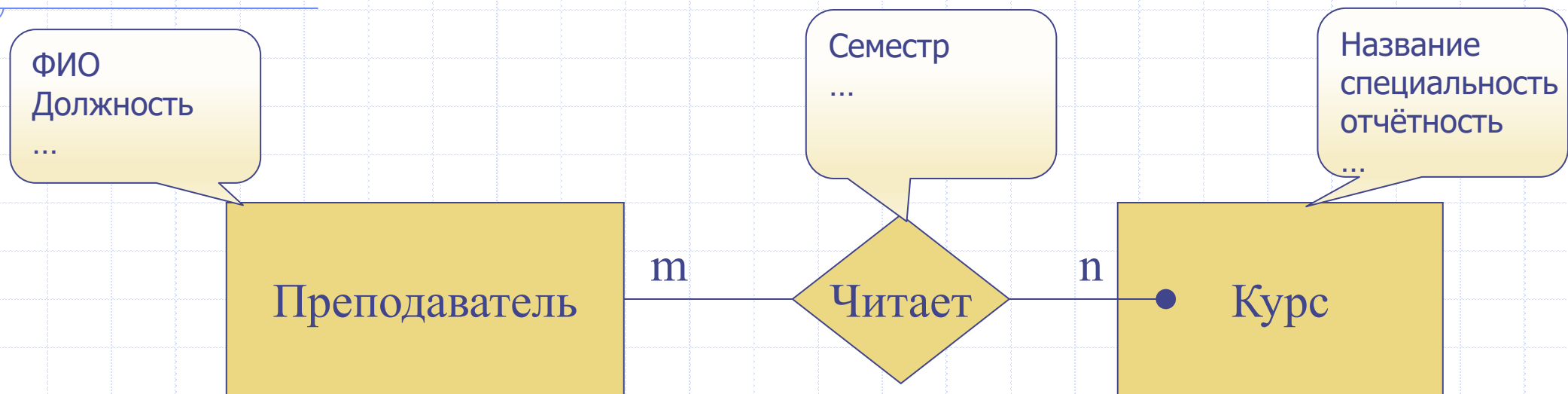
- Нужна ли декомпозиция для R2?
- Получили бы мы такой же результат, если бы проектировали последовательно 1НФ->2НФ->3НФ?

Проектирование на основе модели «Сущность-Связь»

◆ Основные понятия

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

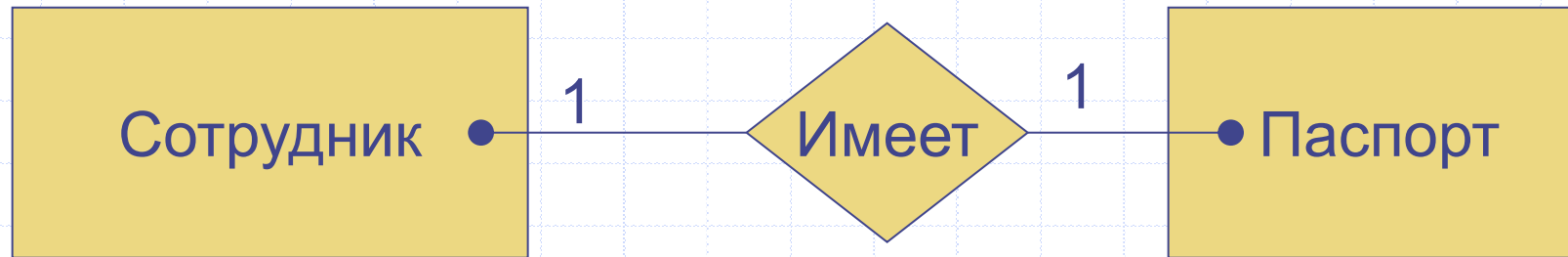
Пример ER-диаграммы бинарной связи



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

Пример

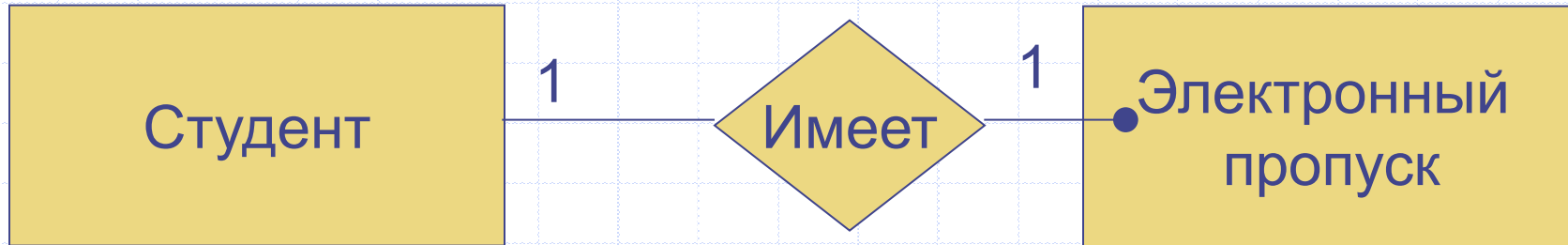


Сотрудник(#карточки, ФИО, датаРожд, #паспорта, дата, выдан, ...)

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 (#зачетки, ФИО, ...)

ЭлКарта(#пропуска, *#зачетки*, 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 (#карточки, ФИО, должность, ...)

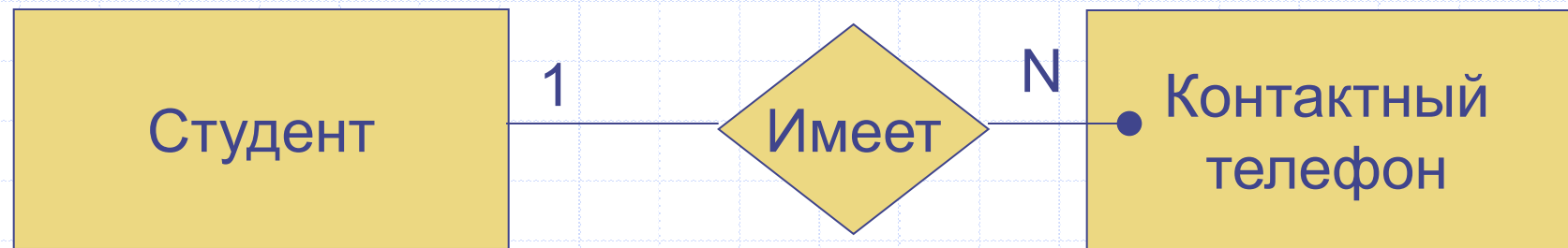
Кафедра (НазвКаф, #комнаты, . . .)

Заведующий (#карточки, 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(#карточки, ФИО, должность, ...)

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

Руководитель(#карточки, КодПр, ...)

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.

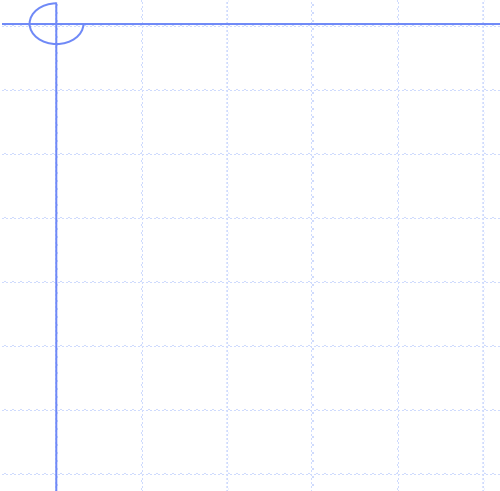
Пример



Сотрудник(#карточки, ФИО, должность, ...)

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

Участвует(#карточки, КодПр, 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, ...)

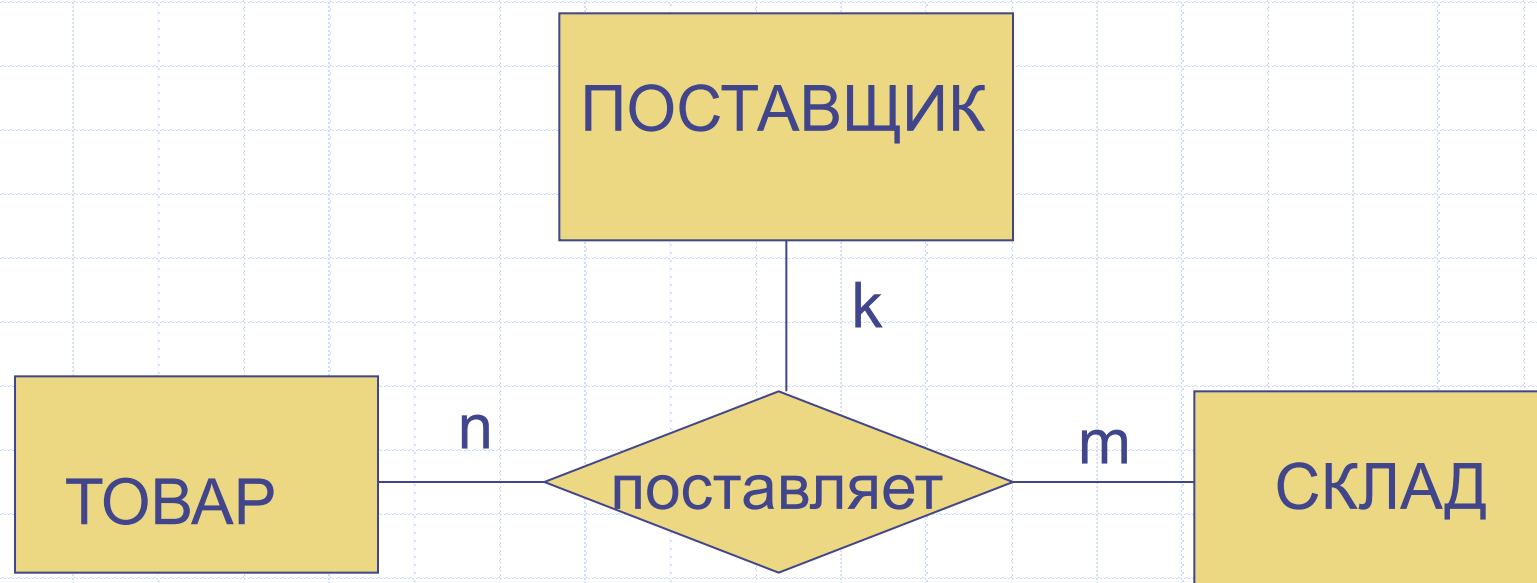
Таблица связи

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*,....)

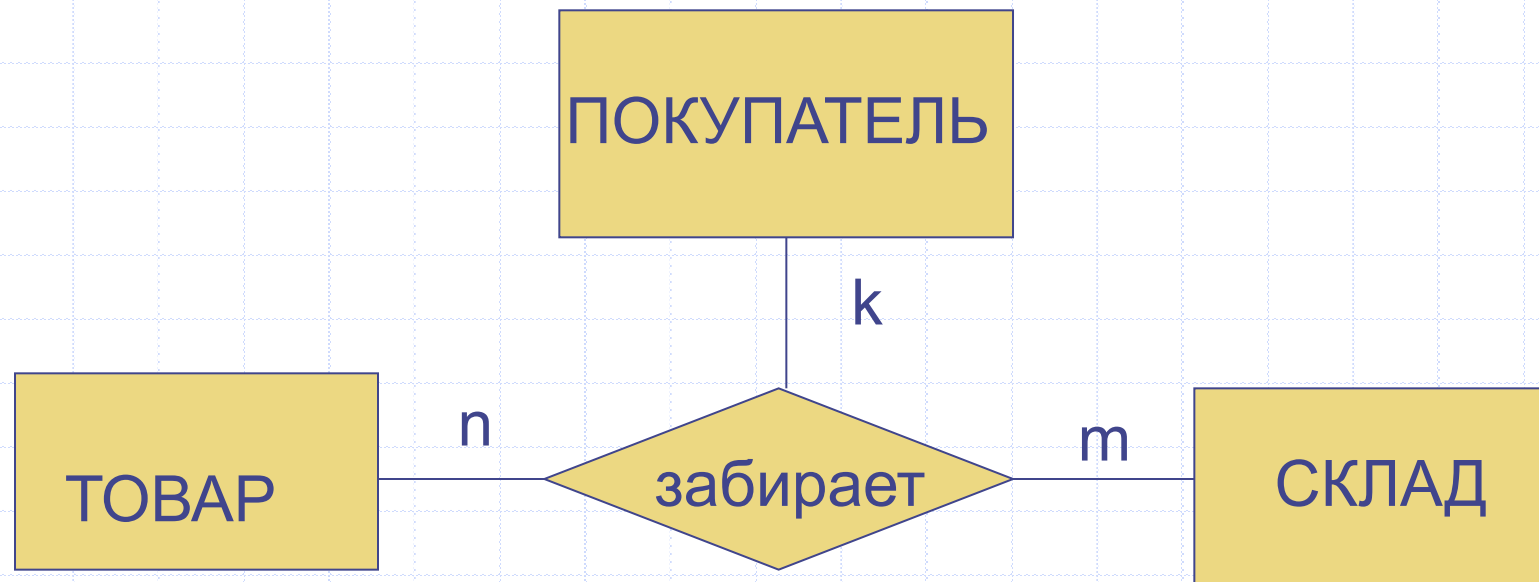
Добавляем справочник

ПОСТАВЩИК(код поставщика, название, *код_города*, . . .)

Таблица связи

ПОСТАВКА (код поставки, *код_товара*, *код_поставщика*, *код_склада*,
дата_поставки, количество, цена)

Роли

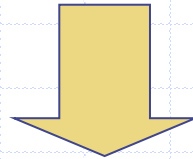


Роли

ПОКУПАТЕЛЬ(код покупателя, название, *код_города*, . . .)

ПОКУПКА (код покупки, *код_товара*, *код_покупателя*, *код_склада*,
дата_покупки, количество, цена)

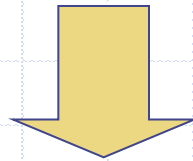
Обобщаем сущности «ПОСТАВЩИК» и «ПОКУПАТЕЛЬ»



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)

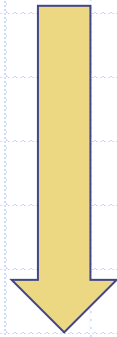
ОПЕРАЦИЯ (код операции, код товара, код контрагента, код склада, дата_операции, тип операции(приход/расход), количество, цена)

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)



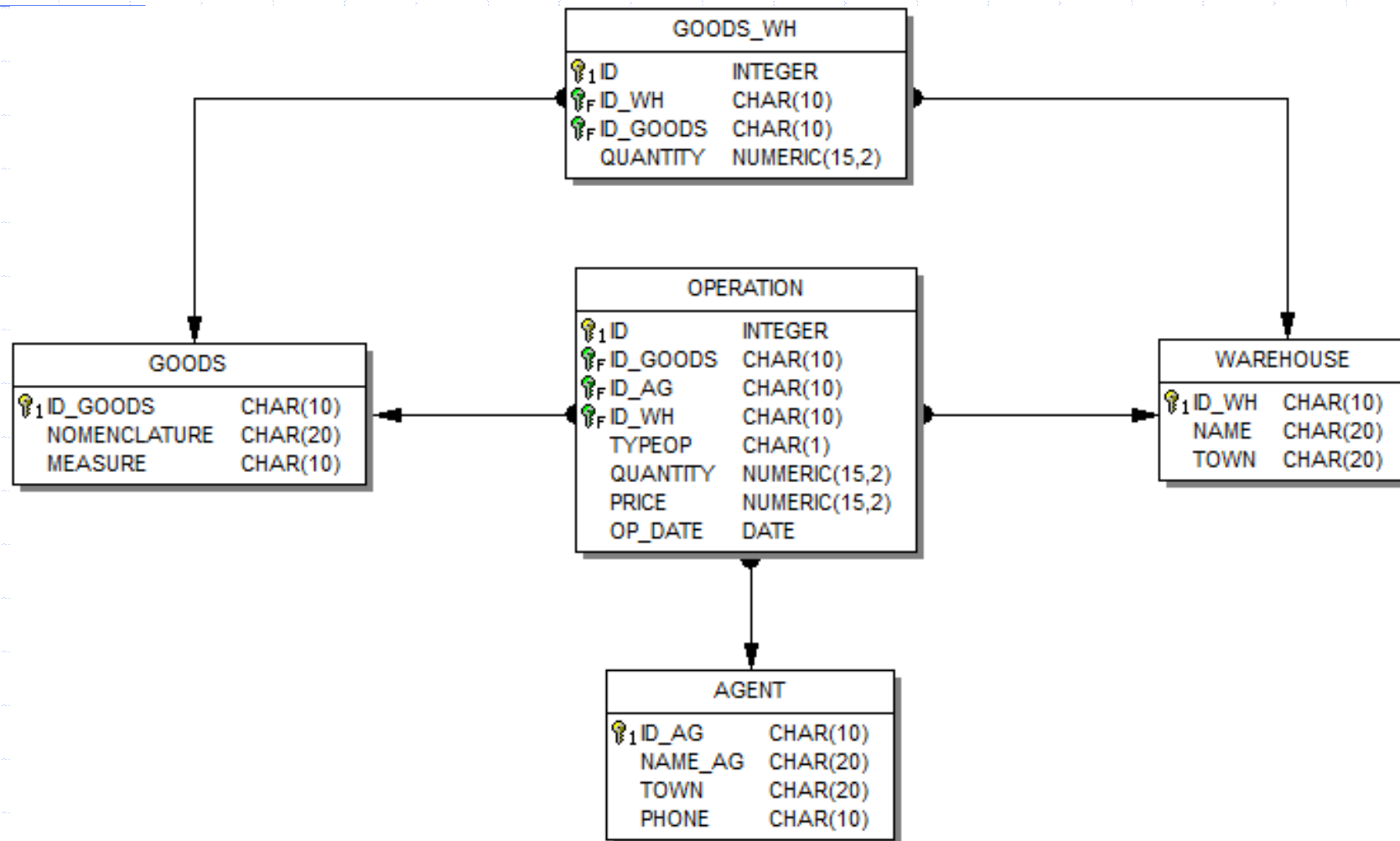
проверка (расход)



ТОВАР_НА_СКЛАДЕ (*код_склада*, *код_товара*, количество)

Можно реализовать процедурой или триггером

ER - диаграмма



Скрипт создания схемы БД

```

/*****
/****                                ****/
/****                                ****/

CREATE TABLE AGENT (
    ID_AG  CHAR(10) NOT NULL,
    NAME_AG CHAR(20) NOT NULL,
    TOWN   CHAR(20),
    PHONE  CHAR(10)
);

...
CREATE TABLE GOODS_WH (
    ID      INTEGER NOT NULL,
    ID_WH   CHAR(10) NOT NULL,
    ID_GOODS CHAR(10) NOT NULL,
    QUANTITY NUMERIC(15,2)
);

...

```

Скрипт создания схемы БД

```

/*****
/****                                ****/
/****                                ****/
/****                                ****/
ALTER TABLE AGENT ADD CONSTRAINT PK_AGENT PRIMARY KEY (ID_AG);
...

/*****
/****                                ****/
/****                                ****/
/****                                ****/
ALTER TABLE OPERATION ADD CONSTRAINT FK_OP_1 FOREIGN KEY (ID_GOODS)
REFERENCES GOODS (ID_GOODS);
...

```

Заполнение БД

```
INSERT INTO WAREHOUSE (ID_WH, NAME, TOWN) VALUES ('C1',  
                                                    'Склад 1', 'Ростов');  
INSERT INTO WAREHOUSE (ID_WH, NAME, TOWN) VALUES ('C2',  
                                                    'Склад на Шолохова', 'Ростов');  
INSERT INTO WAREHOUSE (ID_WH, NAME, TOWN) VALUES ('C3',  
                                                    'Склад 2', 'Батайск');  
INSERT INTO WAREHOUSE (ID_WH, NAME, TOWN) VALUES ('C4',  
                                                    'Склад 3', 'Азов');  
INSERT INTO WAREHOUSE (ID_WH, NAME, TOWN) VALUES ('C5',  
                                                    'Склад ИП ГХР', 'Азов');
```

```
COMMIT WORK;
```

Документы в схеме БД

Накладная (приходная/расходная)

Новая товарная накладная

* Номер По договору

* Дата создания По счету

* Имя клиента  Основание Номер

ИНН: 5256063521; КПП: 775001002; АСХ:
26002332250091003124; Банк: ОАО
"ВОСТОЧЕБАНК"

Добавить НДС ☐

Наименование	Ед. изм.	К-во	Цена	Сумма	
Кресло-мешок "Lite Puff"	шт	4	2300	9200	✖
Кресло-мешок "Big Puff"	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)

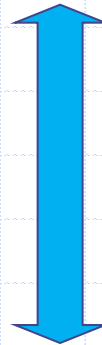
Взаимосвязи

НАКЛАДНАЯ (код накладной, номер, дата, тип, *код контрагента, код склада, ..., сумма*)
СТРОКА_НАКЛАДНОЙ(код строки, *код накладной, код товара, количество, цена*)

исполнение

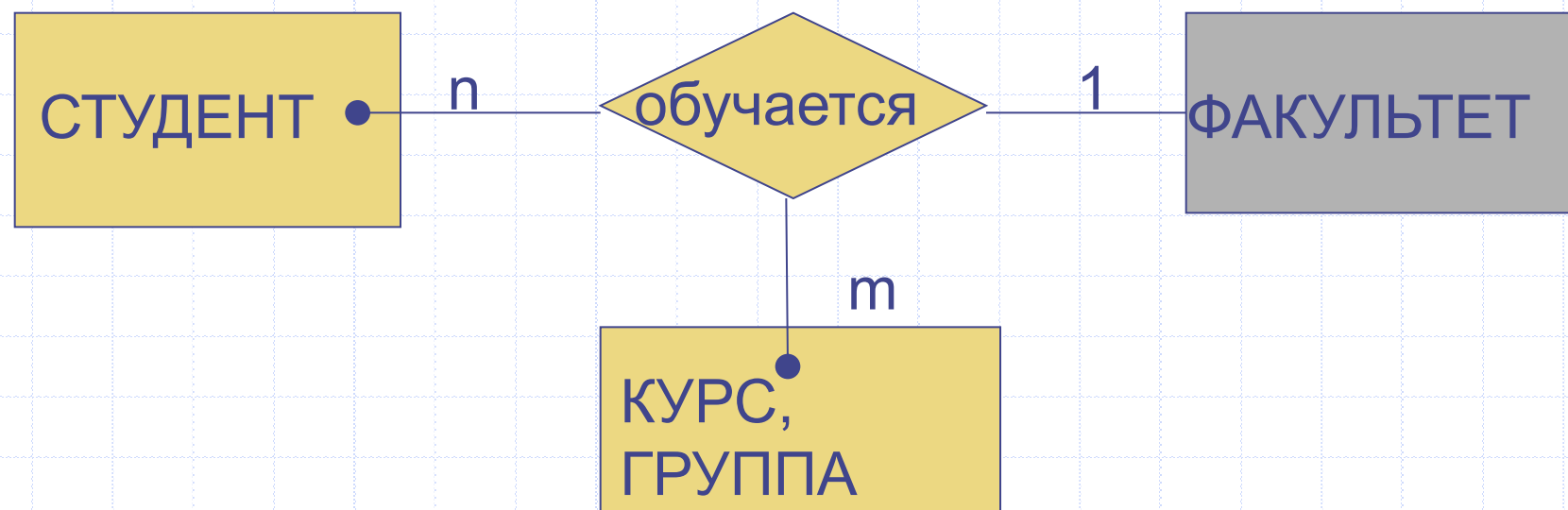


контроль



ОПЕРАЦИЯ (код операции, *код товара, код контрагента, код склада, дата операции, тип операции(приход/расход), количество, цена, код накладной, код строки накладной*)

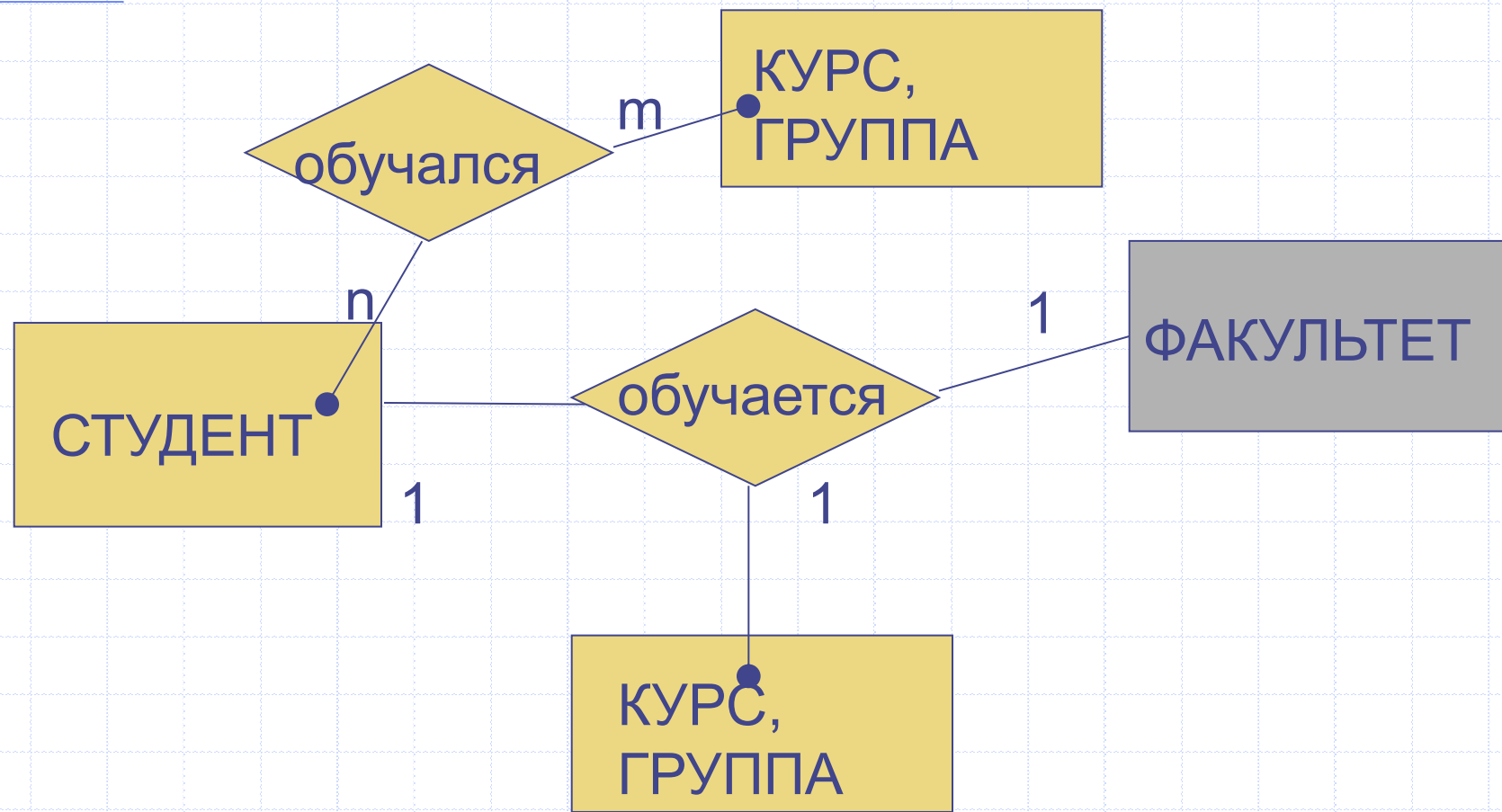
Исторические данные



СТУДЕНТ(код студента, ФИО, *код факультета*, вид_обучения, личные данные,...)

ОБУЧЕНИЕ_ИСТОРИЯ (код, *код студента*, год, курс, группа, семестр, статус)

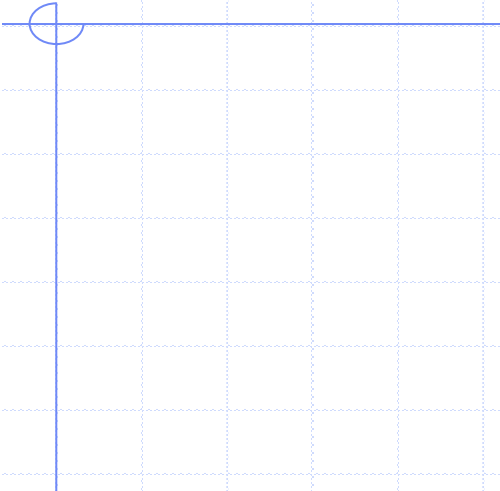
Трудно получать текущие сведения, их можно выделить в отдельную сущность



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

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 через замыкание множества атрибутов)

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

Декомпозиция модели

$R(\underline{a}, \underline{b}, c, d)$

$R_a(\underline{a}, d)$ a – первичный ключ

$R_1(\underline{a}, \underline{b}, c)$ $\{a, b\}$ – первичный ключ
 a – внешний ключ

$R = R_a \text{ JOIN } R_1 \text{ USING } (a)$