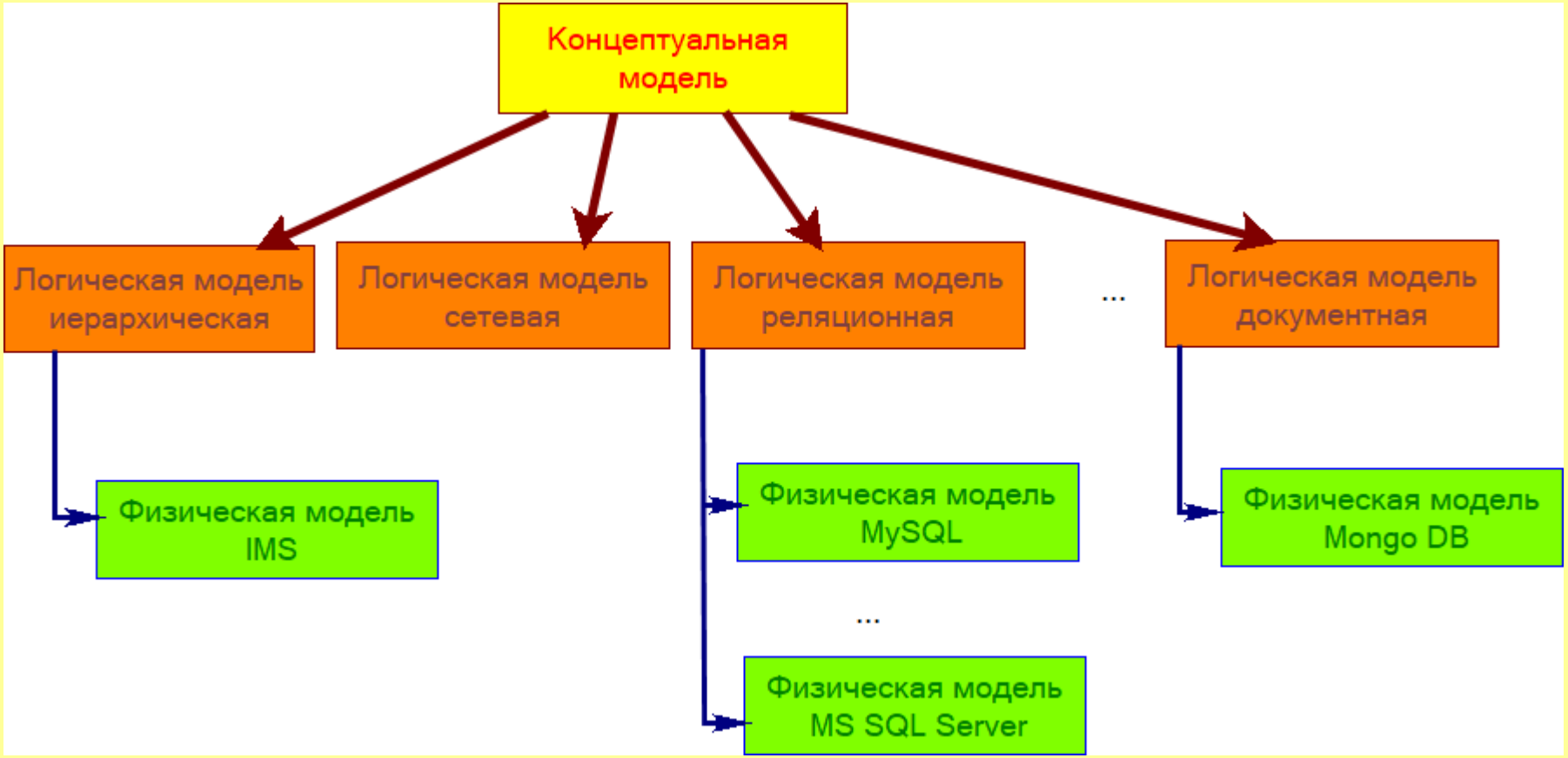
Реляционная логическая модель

Нормализация и денормализация

Модели данных

- *Модель данных*– это абстрактное, независимое, логическое определение структур данных, операторов над данными и прочего, что в совокупности составляет *абстрактную систему*, с которой взаимодействует пользователь.

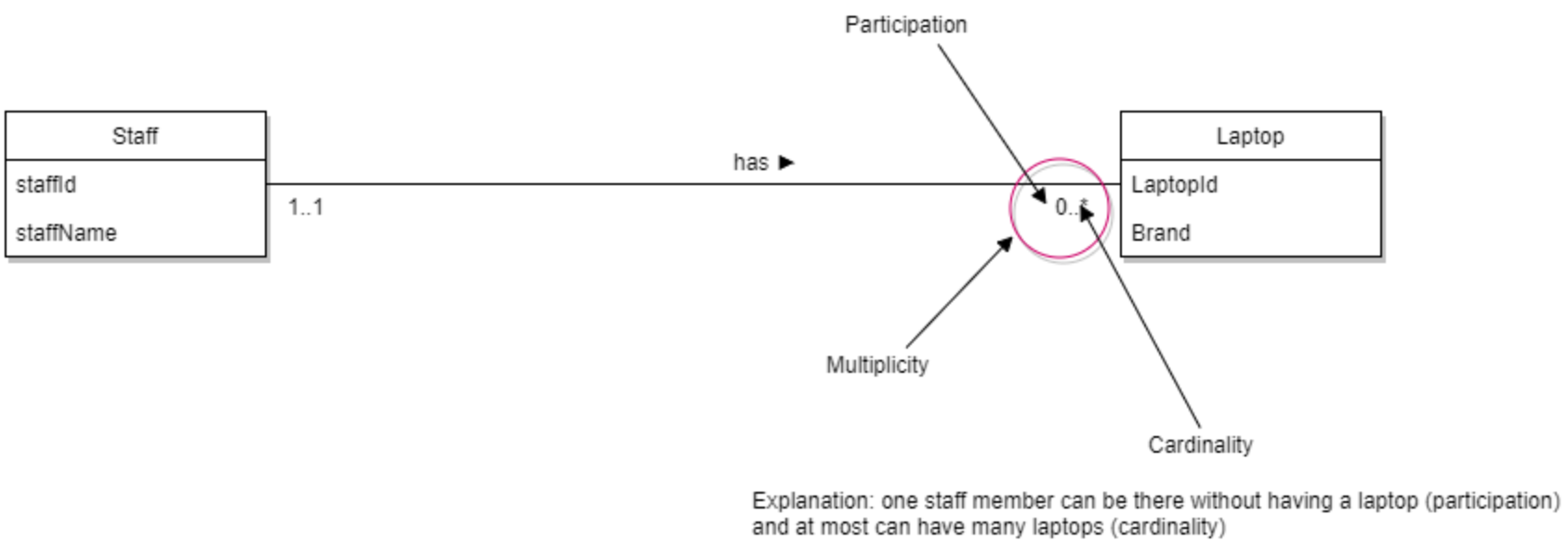


Реляционная таблица (отношение)

- ***Отношение*** представляет собой множество элементов, называемых кортежами.
- **Отношение**— Плоская таблица, состоящая из столбцов и строк
- **Кортеж** — Строка отношения.
- В математике корте́ж — последовательность конечного числа элементов.
- Многие математические объекты формально определяются как кортежи. Например, граф определяется как кортеж (V,E), где V — это набор вершин, а E — подмножество V × V, обозначающее рёбра.
- **Степень.** Степень отношения определяется количеством атрибутов, которое оно содержит.
- **Кардинальность** — Количество кортежей, которое содержится в отношении.
- **Кратностью** (multiplicity) **связи** называется характеристика, указывающая, сколько атрибутов класса сущности с данной ролью может или должно участвовать в каждом экземпляре связи какого-либо вида.
- **Кардинальность** (Cardinality) **связи** Обозначает максимальное количество возможных взаимосвязей, в которых может участвовать определенная сущность (не более)

Кратность и Кардинальность

- Multiplicity = Cardinality + Participation

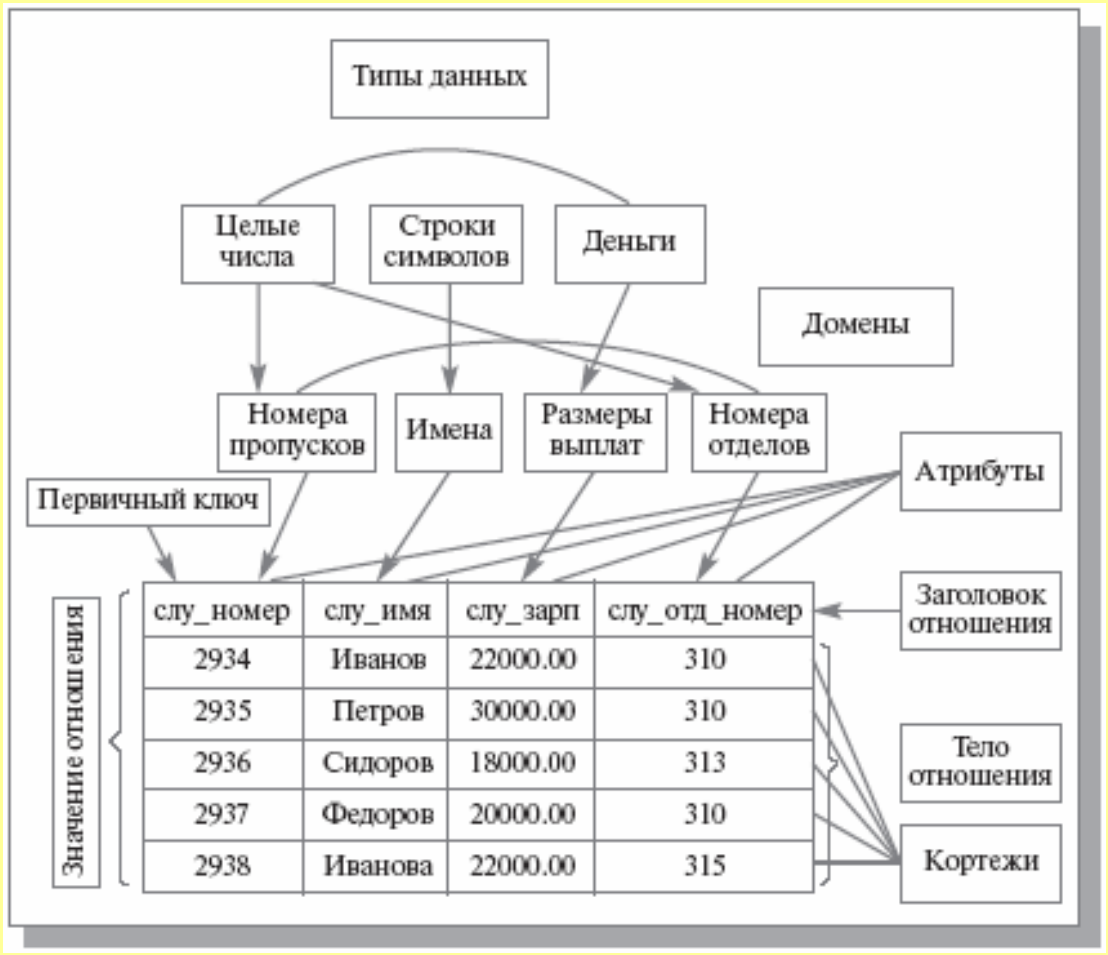


- https://martinfowler.com/bliki/Multiplicity

Альтернативные наименования

Официальные термины	Альтернативный вариант 1	Альтернативный вариант 2
Отношение (Relation**)**	**Таблица (table)**	Файл (file)
Кортеж (tuple)	Строка (row)	Запись (record)
Атрибут (attribute)	Столбец (column)	Поле (field)

Реляционная таблица: что есть что



Свойства реляционной таблицы

1. Таблица представляет собой двумерную структуру, состоящую из строк и столбцов
2. Каждая строка таблицы (кортеж, tuple) представляет собой отдельную сущность внутри набора сущностей
3. Каждый столбец таблицы представляет собой атрибут, и у каждого столбца есть свое имя
4. На каждом пересечении строки и столбца имеется единственное значение
5. Каждая таблица должна иметь атрибут или несколько атрибутов, уникально идентифицирующих каждую строку
6. Все значения в столбце должны отображаться в одинаковом формате. Например, если атрибуту присваивается формат целого, то все значения в столбце, представляющем данный атрибут должны быть целыми
7. Каждый столбец имеет определенный диапазон значений, называемый доменом атрибута
8. Порядок следования строк и столбцов не существенен.

Ключи реляционных баз данных

Тип ключа	Определение
Суперключ (superkey)	Атрибут(или комбинация атрибутов), уникально идентифицирующих каждую сущность в таблице
Потенциальный ключ (candidate key)	Минимальный суперключ. Суперключ, который не содержит подмножества атрибутов, которое само по себе является суперключом.
Первичный ключ (primary key)	Потенциальный ключ, выбранный для уникальной идентификации всех остальных значения атрибутов в любой строке. Не может содержать пустых значений
Вторичный ключ (secondary key)	Атрибут(или комбинация атрибутов), используемый исключительно в целях поиска данных
Внешний ключ (foreign key)	Атрибут(или комбинация атрибутов) в одной таблице, значения которого должны или совпадать со значениями первичного ключа другой таблицы, или быть пустыми

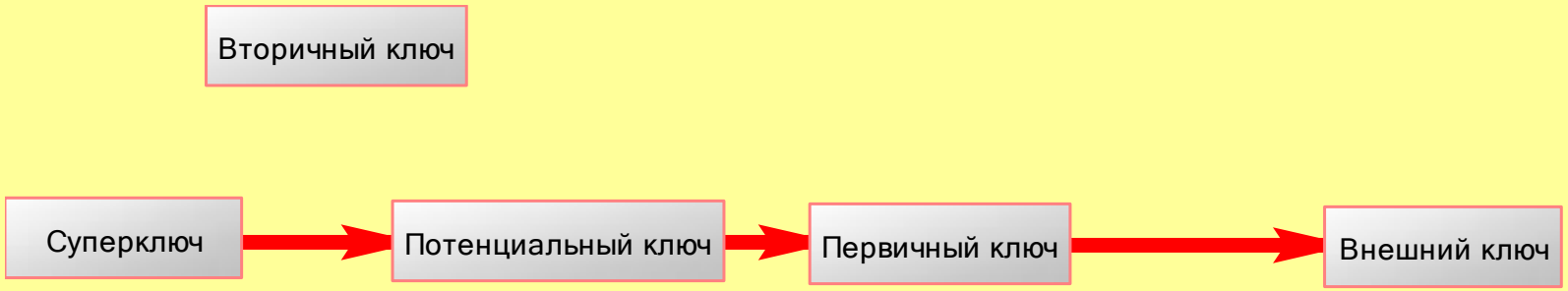
Ключи пример

Человек

id	инн	фамилия	имя	отчество	серия паспорта	номер паспорта	дата рождения

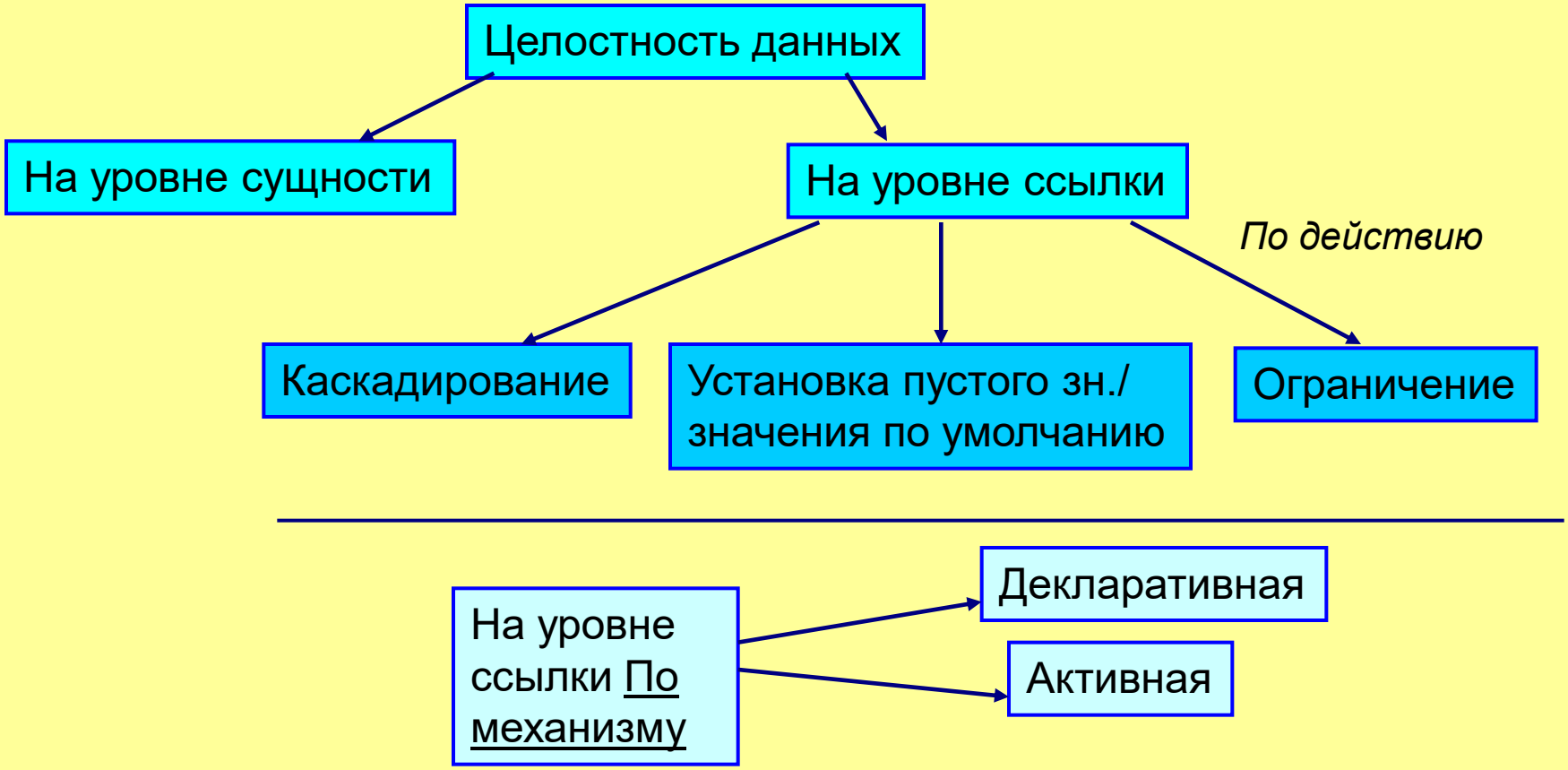
Суперключ	**Потенциальный ключ**	**Первичный ключ**	**внешний ключ**	**вторичный**
id	id	id	id	ф+и+о+ дата рождения
инн	инн			
ф+и+о+дата рождения	ф+и+о+дата рождения			
серия паспорта+ номер паспорта	серия паспорта+ номер паспорта			
ф+и+о+ИНН				
ф+инн				
…				
id+ИНН+ ф+и+о+серия+ номер+ дата рождения				

Ключи


Вторичный ключ
Суперключ
Потенциальный ключ
Первичный ключ
Внешний ключ

Целостность данных

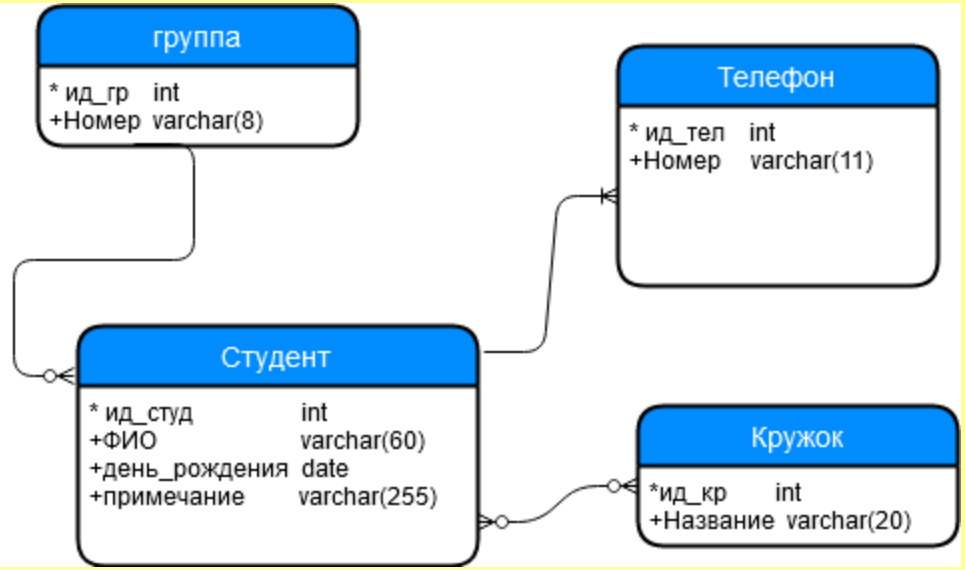
- **Пустое значение.** Указывает, что значение атрибута в настоящий момент неизвестно или неприемлемо для этого кортежа.



Ссылочная целостность

Наименование типа сохранения	Код на SQL	Что происходит при удалении	Что происходит при обновлении
Ограничение	restrict	Не дает удалить данные из родительской таблицы, если с ними связаны какие-либо данные в дочерней (проверка сразу)	Не дает изменить первичный ключ из родительской таблицы, если с ним связаны какие-либо данные в дочерней (проверка сразу)
Ограничение	No action	Не дает удалить данные из родительской таблицы, если с ними связаны какие-либо данные в дочерней (проверка отложена)	Не дает изменить первичный ключ из родительской таблицы, если с ним связаны какие-либо данные в дочерней (проверка отложена)
Каскадирование	cascade	При удалении данных из родительской таблицы, удалятся связанные с ними данные в дочерней	При изменении первичного ключа из родительской таблицы, связанные с ним внешние ключи в дочерней изменятся на такое же значение
Установка	set null	При удалении данных из родительской таблицы, внешние ключи дочерней таблицы связанные с удаляемыми данными получат пустое значение(null)	При изменении первичного ключа из родительской таблицы, внешние ключи дочерней таблицы связанные с изменяемыми ключами получат пустое значение(null)
Установка	set default	При удалении данных из родительской таблицы, внешние ключи дочерней таблицы связанные с удаляемыми данными получат значение по умолчанию, которое должно быть задано в дочерней таблице	При изменении первичного ключа из родительской таблицы, внешние ключи дочерней таблицы связанные с изменяемыми ключами получат значение по умолчанию, которое должно быть задано в дочерней таблице

Концептуальная модель (ER-диаграмма)



Правила построения реляционной логической модели по концептуальной

Сущность/связь	Способ преобразования
Сильная сущность	Создание отношений, которые включают все простые атрибуты
Слабая сущность	Создание отношений, которые включают все простые атрибуты (после преобразования связи с каждой сущностью-владельцем необходимо также определить первичный ключ)
Многозначный атрибут	Создание отношения, представляющего многозначный атрибут, и передача копии первичного ключа сущности-владельца в новое отношение для использования в качестве внешнего ключа

Символьные типы данных

- Char(7)
- Лесопова**л** → “Лесопов”
- Лес → “Лес_ _ _ _”

- Varchar(7)
- Лесопова**л** → “Лесопов”
- Лес → “Лес”

Character Large object (Text)

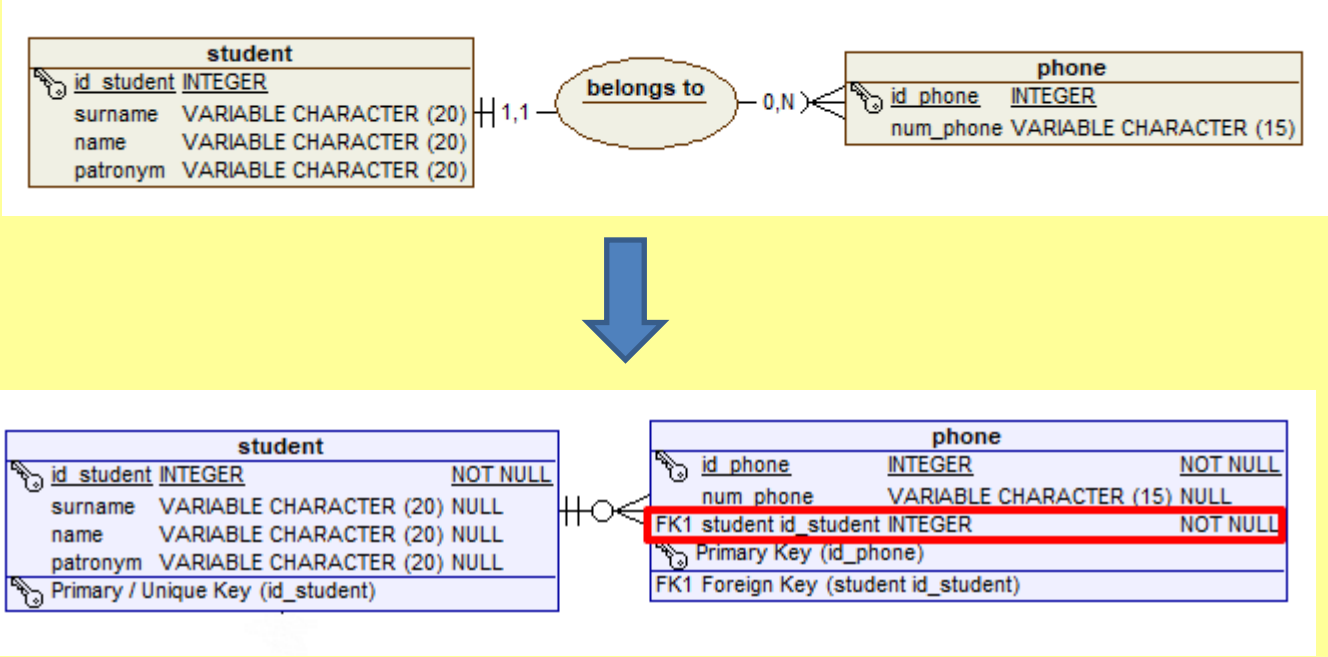
- до 2 ГБ
- Нет значения по умолчанию
- Поиск по шаблону (like) не возможен
- Ограниченная сортировка и индексы

Правила построения реляционной логической модели по концептуальной

Связь	Способ преобразования
Двухсторонняя связь типа 1:*	Передача первичного ключа сущности на сторону "один" для использования в качестве первичного ключа в отношении, соответствующем сущности на стороне "многие". На сторону "многие" передаются также все атрибуты связи
Двухсторонняя связь типа *:*, сложная связь	Создание отношения, представляющего связь, и включение всех атрибутов связи. Передача в новое отношение копии первичного ключа из каждой сущности-владельца для использования в качестве внешних ключей
Двухсторонняя связь типа 1:1:	
обязательное участие обеих сторон	Объединение сущностей в одно отношение
обязательное участие одной стороны	Передача первичного ключа сущности на "необязательную" сторону для использования в качестве внешнего ключа в отношении, представляющем сущность на "обязательной" стороне
необязательное участие обеих сторон	Если отсутствует дополнительная информация, то выбор становится произвольным

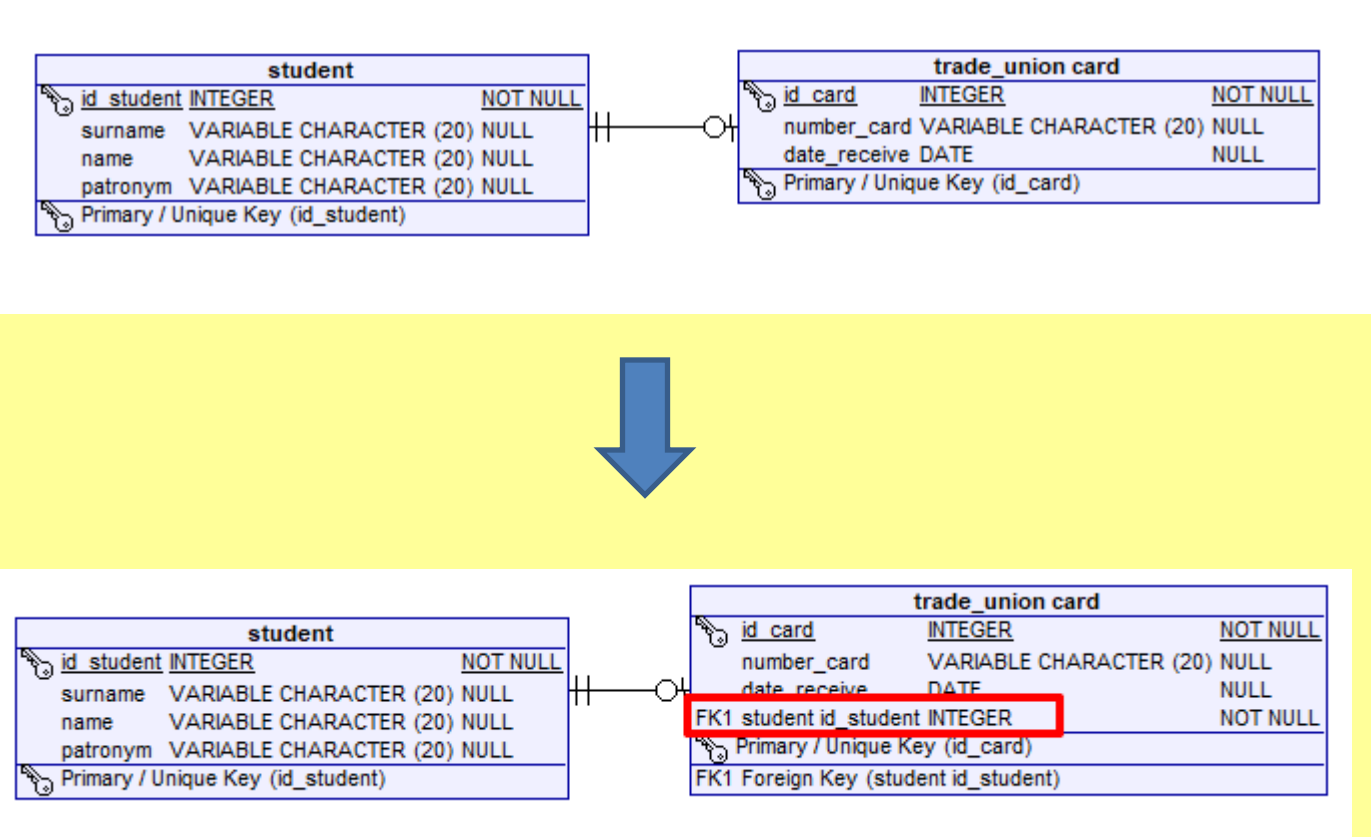
Двухсторонняя связь типа 1:*

Передача первичного ключа сущности на сторону "один" для использования в качестве первичного ключа в отношении, соответствующем сущности на стороне "многие". На сторону "многие" передаются также все атрибуты связи



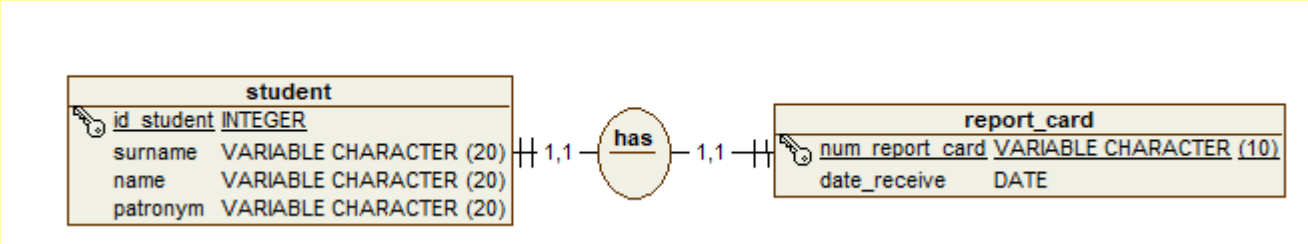
1:1 обязательное участие одной стороны

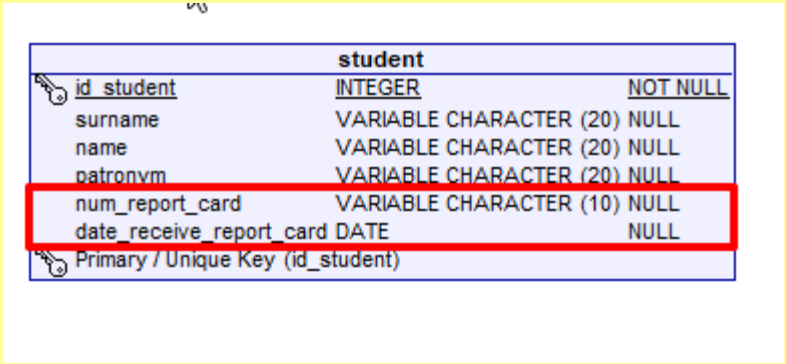
Передача первичного ключа сущности на "необязательную" сторону для использования в качестве внешнего ключа в отношении, представляющем сущность на "обязательной" стороне



1:1 обязательное участие обеих сторон

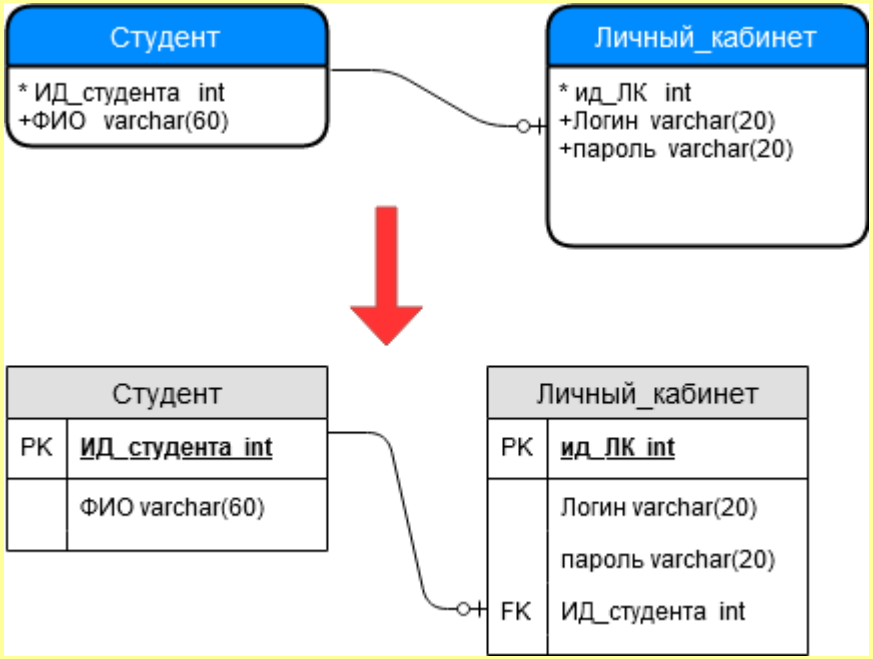
Объединение сущностей в одно отношение



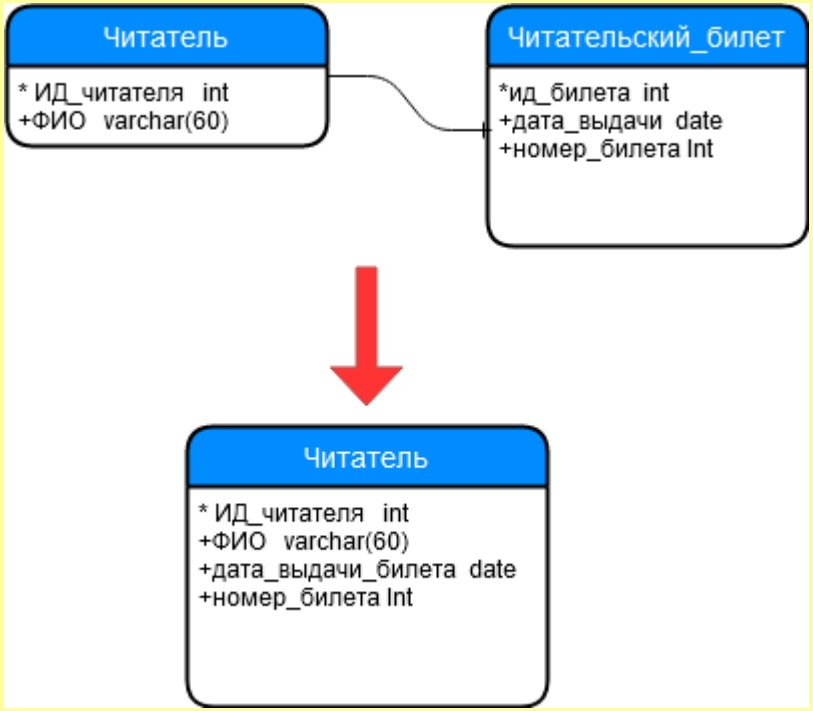


Логическая 1:1

Обязательная одна сторона

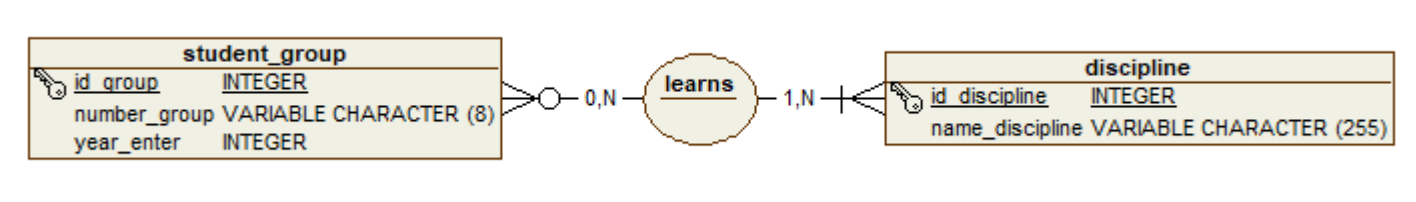


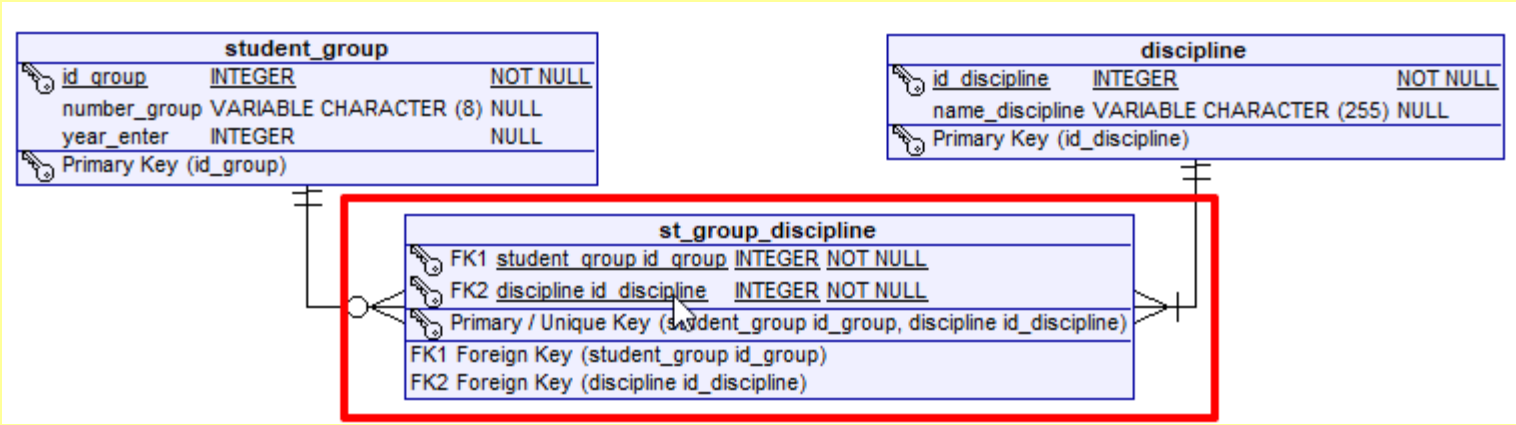
Обязательны обе стороны



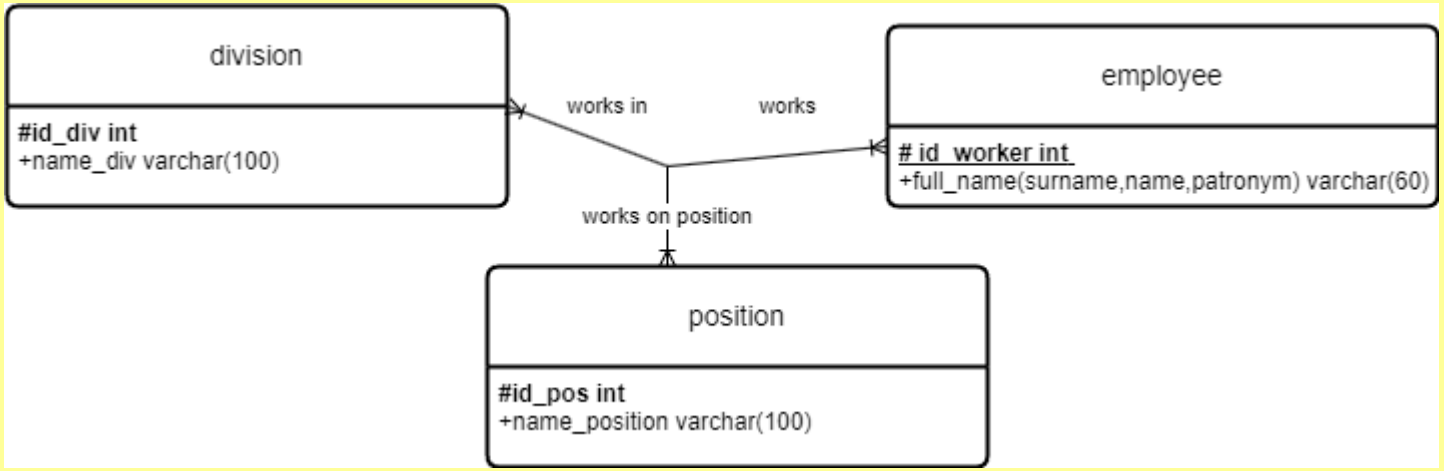
Двухсторонняя связь типа *:*

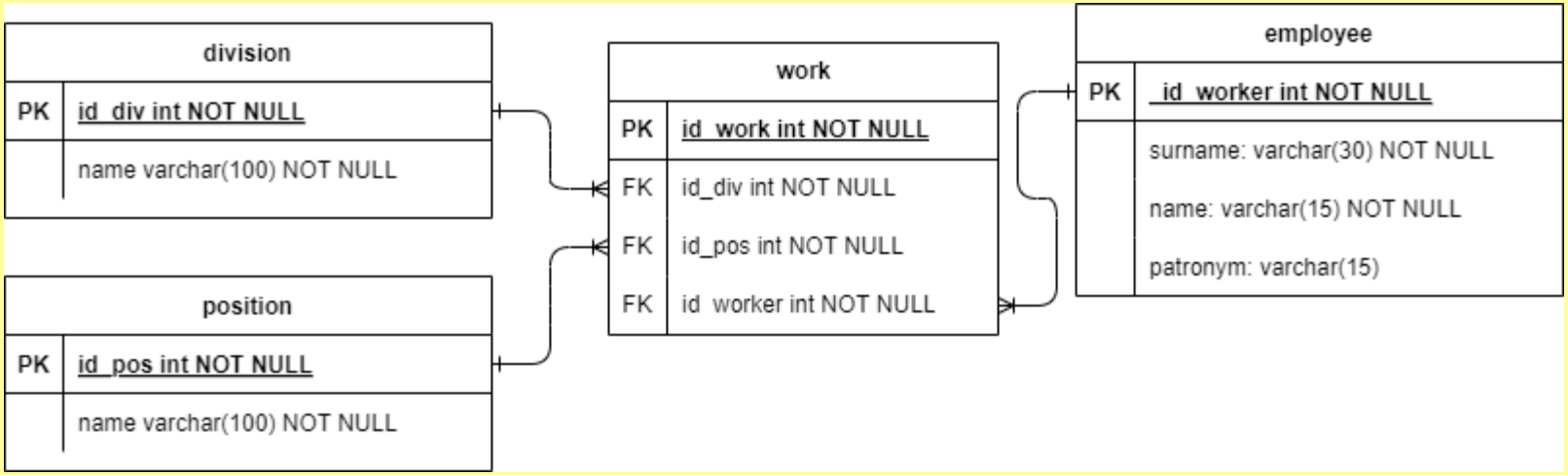
Создание отношения, представляющего связь, и включение всех атрибутов связи. Передача в новое отношение копии первичного ключа из каждой сущности-владельца для использования в качестве внешних ключей





сложная связь (тройная)

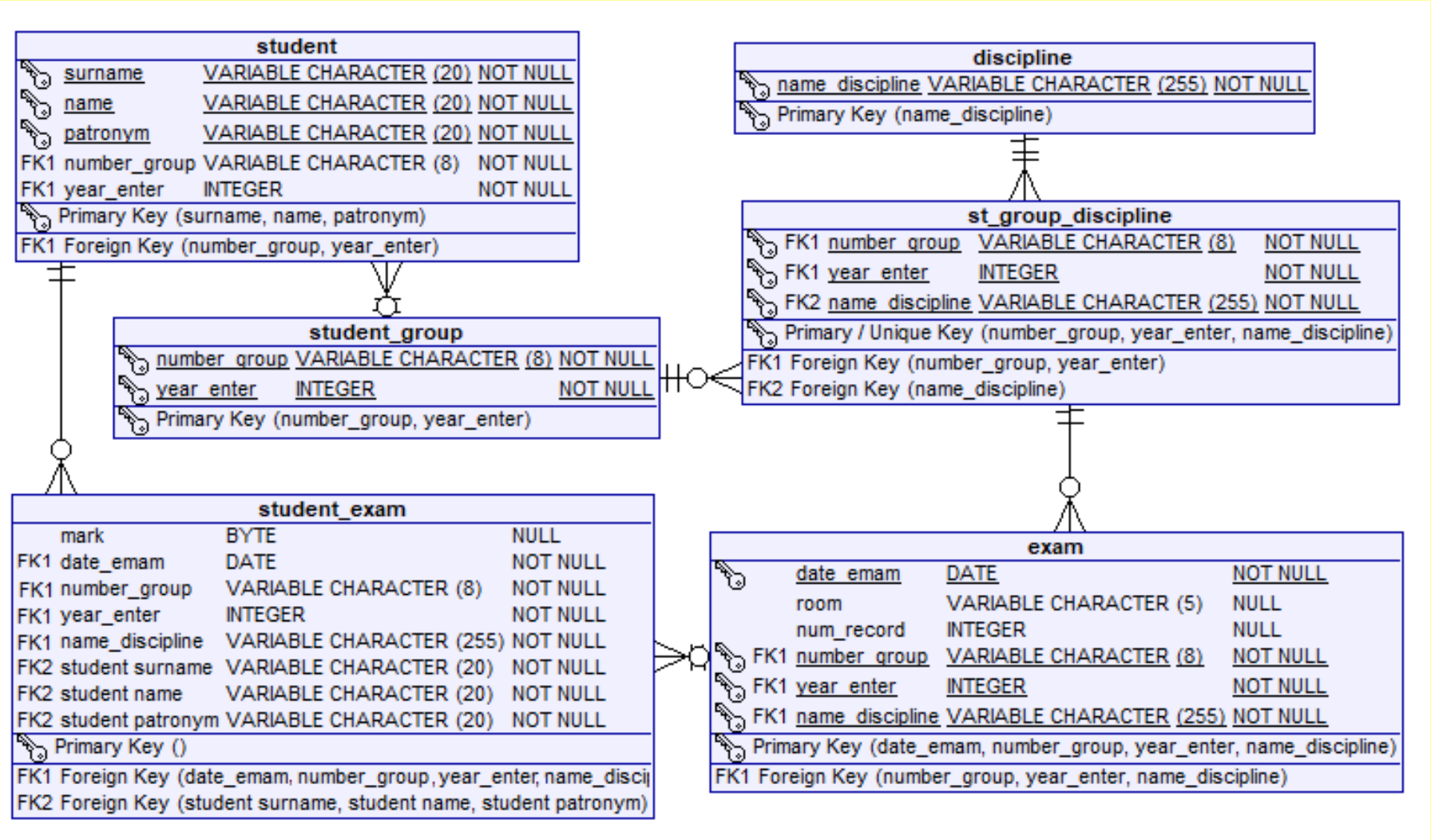




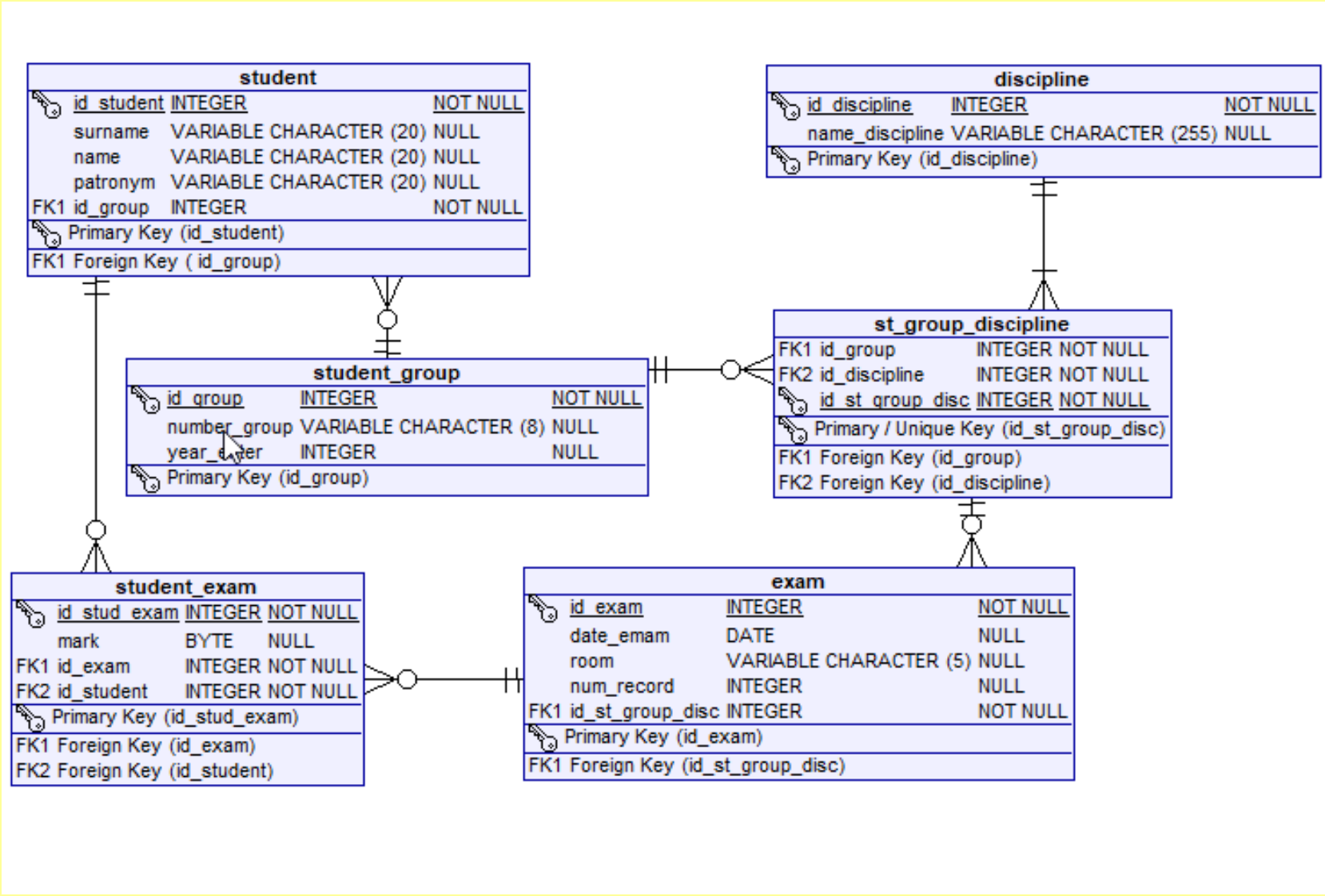
Ключи по происхождению

- **Естественный Ключ (ЕК)** – набор атрибутов описываемой записью сущности, уникально её идентифицирующий (например, номер паспорта для человека)
- **Суррогатный Ключ (СК)** – автоматически сгенерированное поле, никак не связанное с информационным содержанием записи. Обычно в роли СК выступает автоинкрементное поле типа INTEGER.

«Гонка атрибутов» естественные ключи

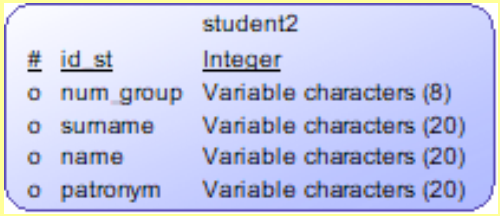


«Гонка атрибутов» суррогатные ключи



Проблемы проектирования

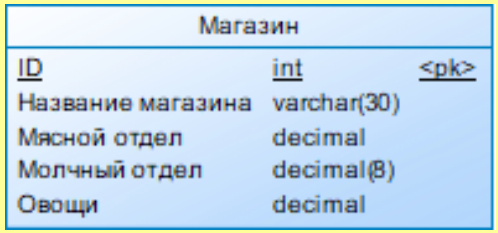
- Аномалии обновления
- Аномалии удаления
- Аномалии вставки
- Использование большого объема памяти под данные

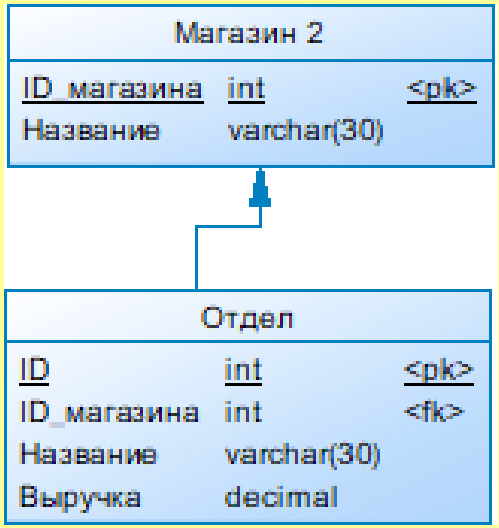


Тип	Место (байты)
TINYINT	1
SMALLINT	2
MEDIUMINT	3
INT	4
BIGINT	8

Длина строки	Место для однобайтовой строки (байты)	Место для двухбайтовой строки (байты)
1	1	2
2	2	4
8	8	16
10	10	20
20	20	40

Как лучше формализовать предметную область ?



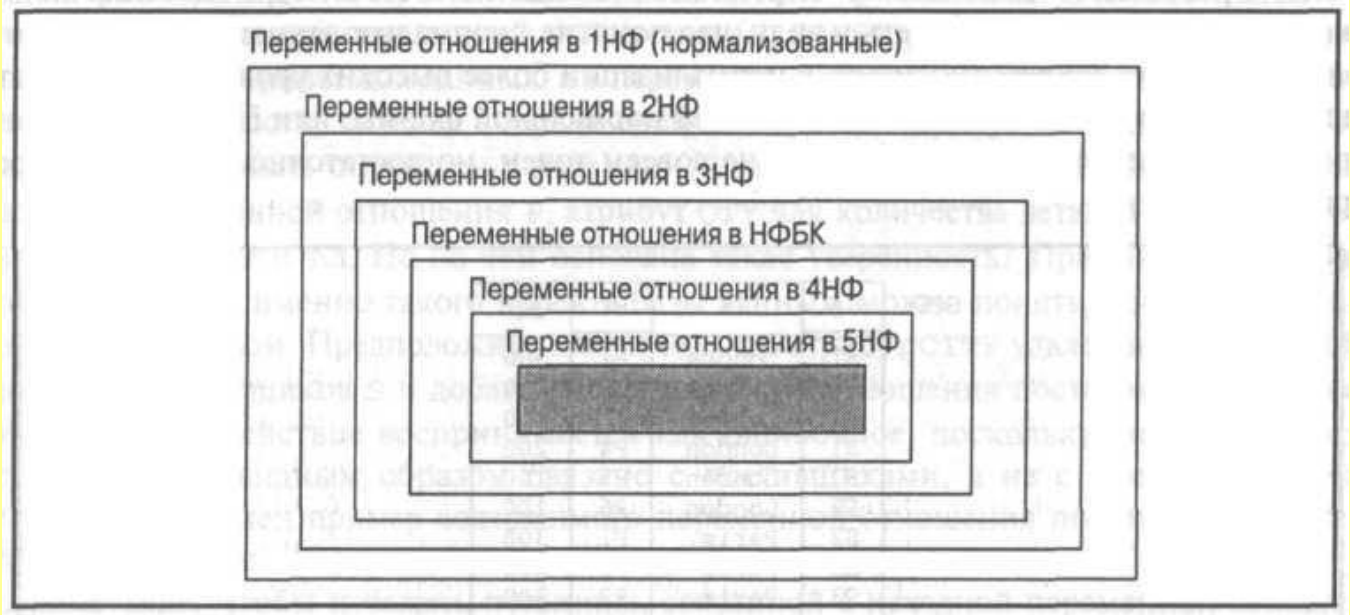


Нормализация

- **Нормализация данных** – это процесс приведения модели к виду, позволяющему получить в дальнейшем структуру базы данных, в которой устранена избыточность хранения и сведены к минимуму аномалии при добавлении, удалении, изменении данных.
- Процесс нормализации проводится поэтапно. На каждом из этапов на структуру базы накладывается некоторое ограничение и в структуре базы выправляется некоторый дефект. Про базу с соответствующими ограничениями говорят, что она находится в одной из **нормальных форм.**

Нормальные формы

1. Первая нормальная форма (1НФ)
2. Вторая нормальная форма (2НФ)
3. Третья нормальная форма (3НФ)
4. Нормальная форма Бойса-Кодда(НФБК)
5. Четвертая нормальная форма (4НФ)
6. Пятая нормальная форма (5НФ)
7. Шестая нормальная форма (6НФ)
8. Доменно-ключевая нормальная форма (DKNF)



Ненормализованное отношение

- **Ненормализованная форма (ННФ).** Таблица, содержащая одну или несколько повторяющихся групп данных.

Первая нормальная форма

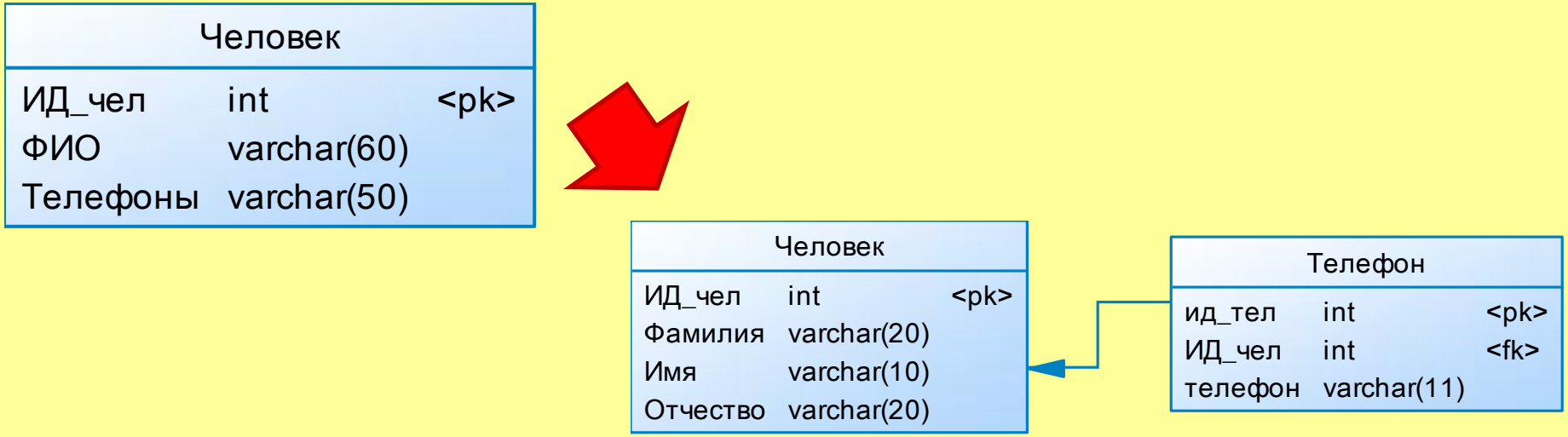
- **Первая нормальная форма (1НФ).** Отношение, в котором на пересечении каждой строки и каждого столбца содержится одно и только одно атомарное значение.

Возможные нарушения 1НФ:

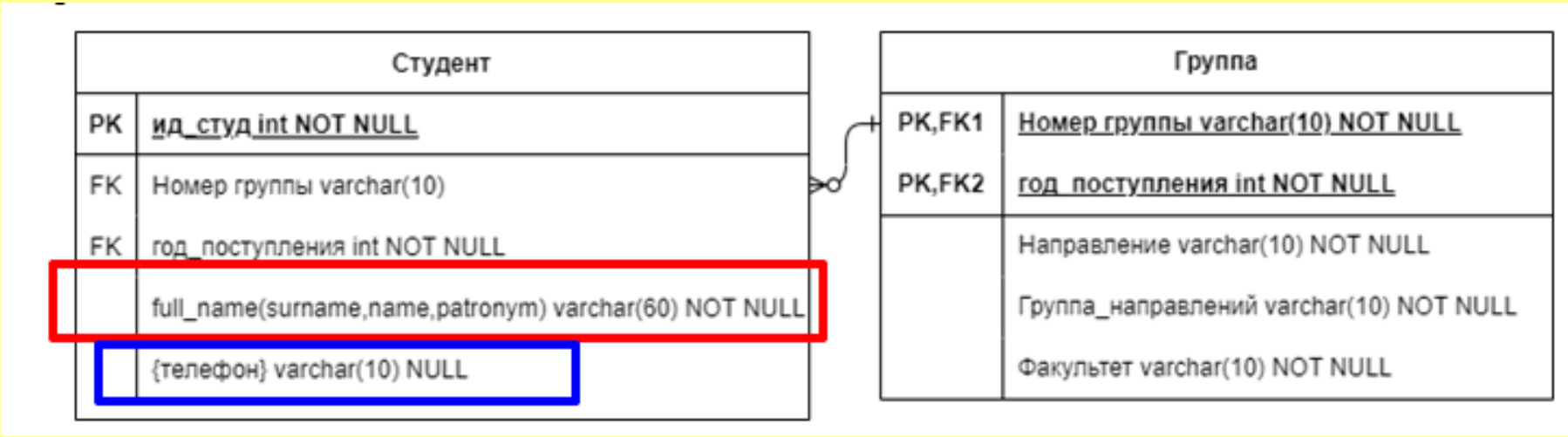
- Значение может быть составным (ФИО)
- многозначным (список номеров телефона)

Как привести к 1 НФ?

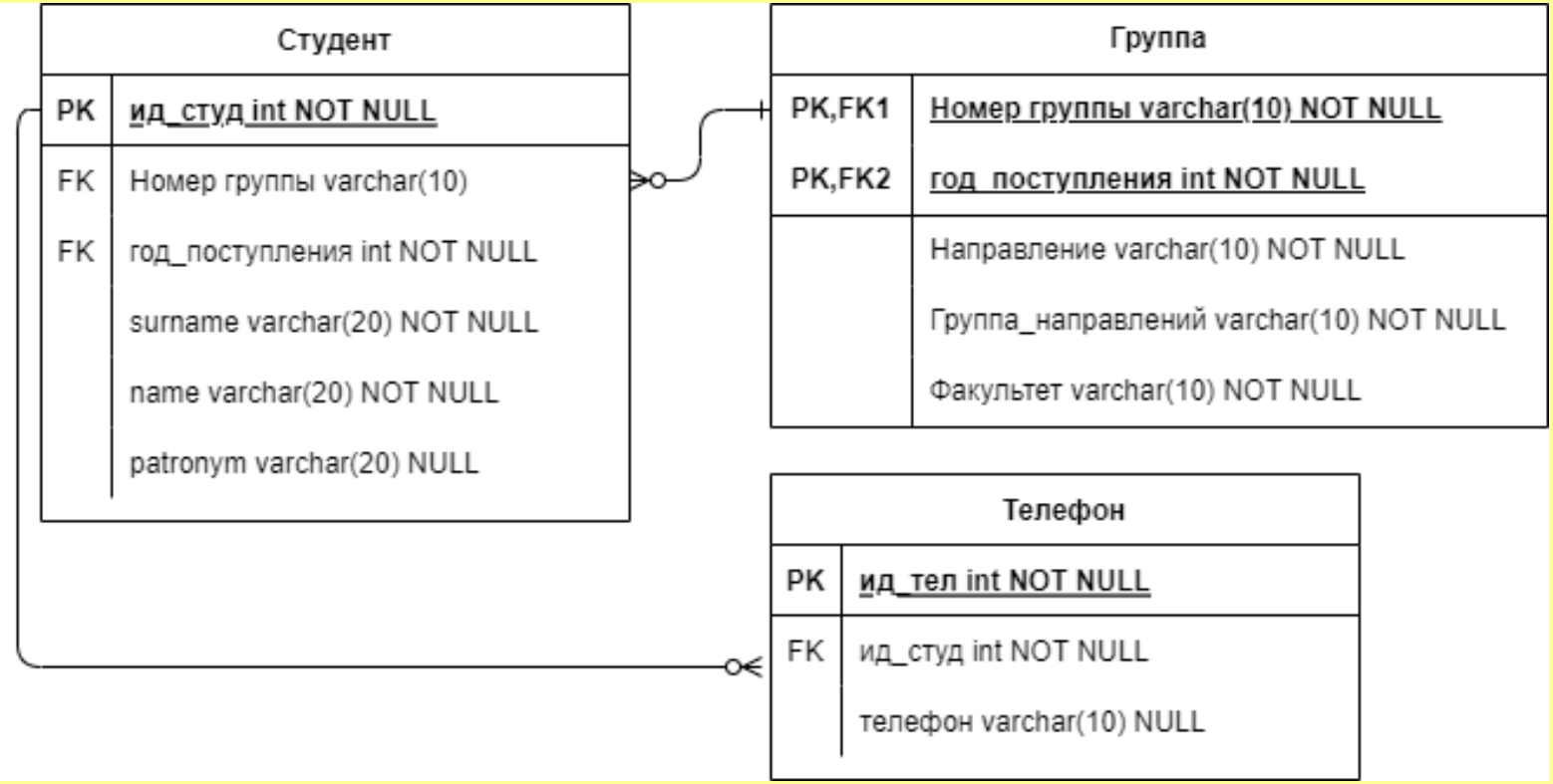
- Составное поле — разделяем на элементы
- Многозначный атрибут — создание отношения (таблицы), представляющего многозначный атрибут , и передача копии первичного ключа сущности-владельца в новое отношение для использования в качестве внешнего ключа



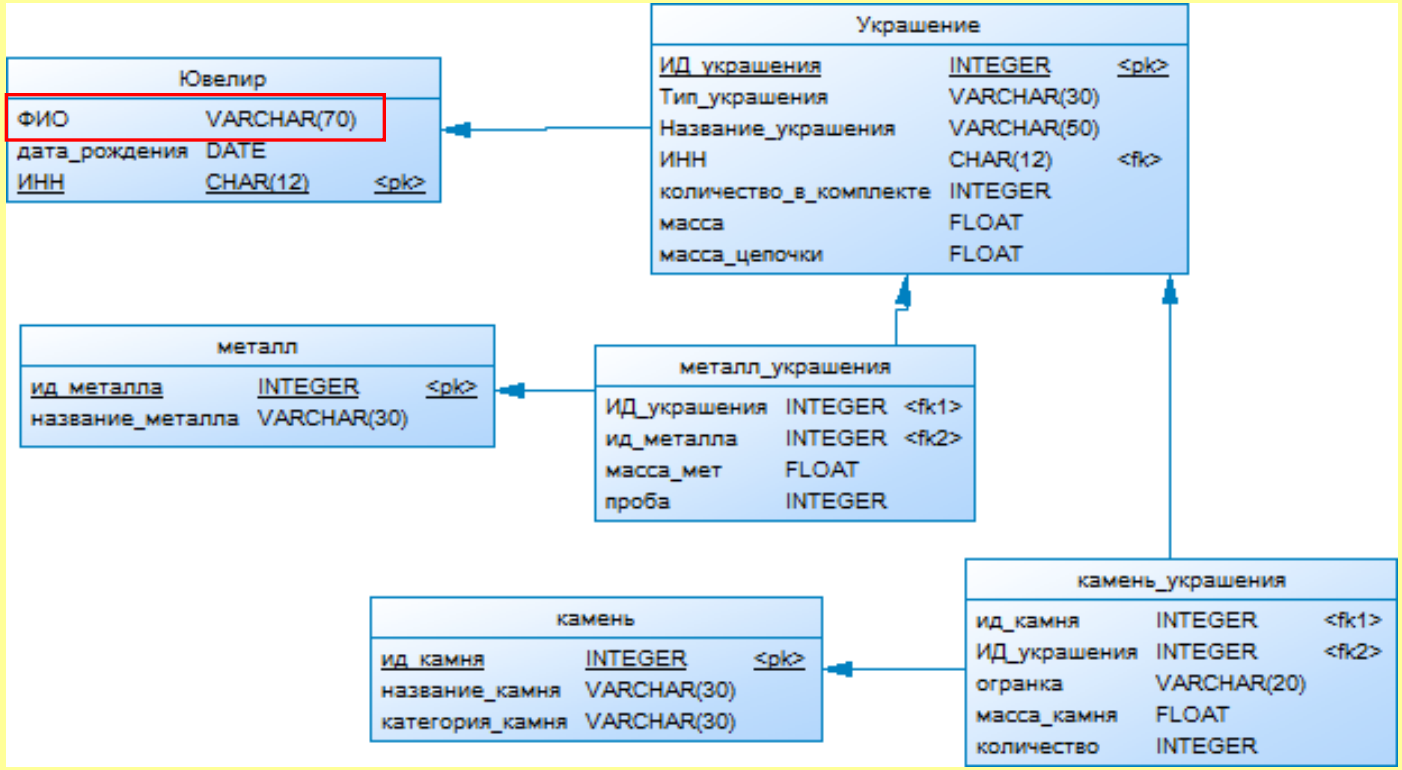
Ненормализованная база данных



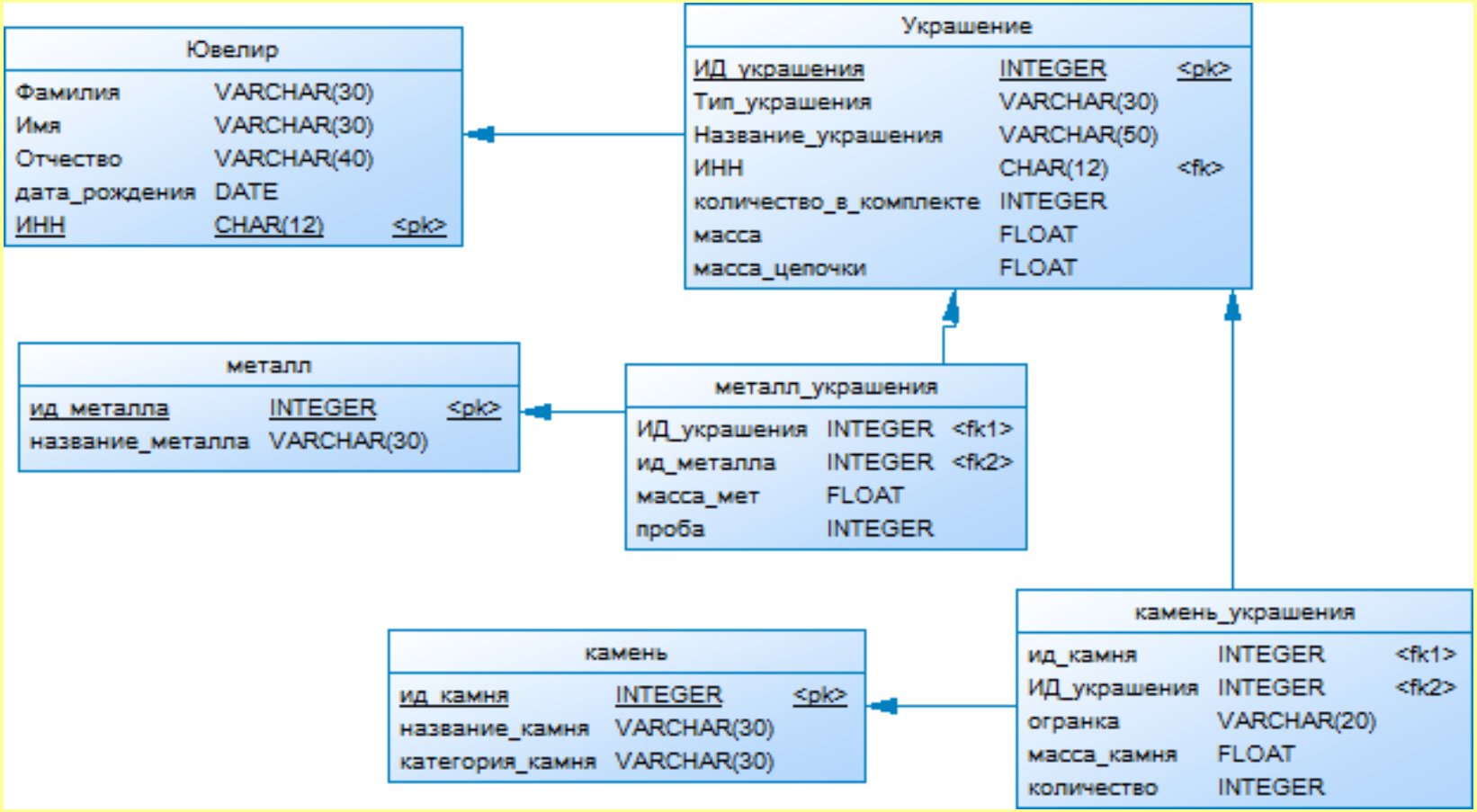
База данных в 1НФ



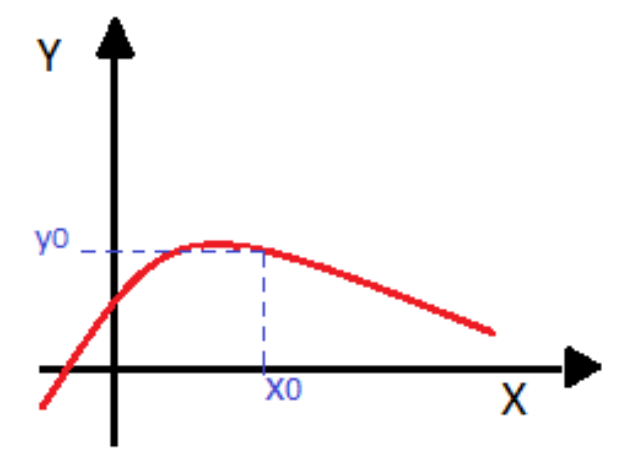
Ненормализованная база данных



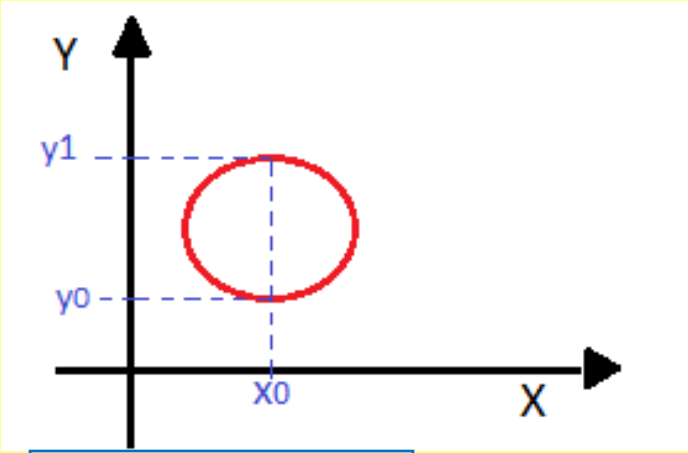
База данных в 1НФ



Функциональная зависимость



Y-функция (Y зависит от x)



Y- НЕ функция

- **Функциональная зависимость.** Описывает связь между атрибутами отношения. Например если в отношении R, содержащем атрибуты A и B, атрибут B функционально зависит от атрибута A (что обозначается как A (B)), то каждое значение атрибута A связано только с одним значением атрибута B. (Причем атрибуты A и B могут состоять из одного или нескольких атрибутов.)

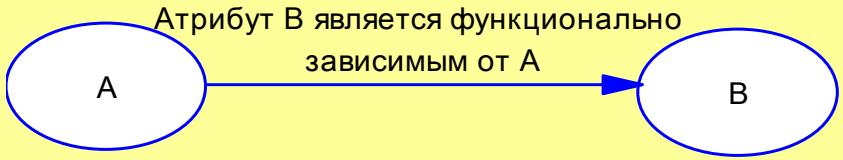


Диаграмма функциональной зависимости

Функциональная зависимость

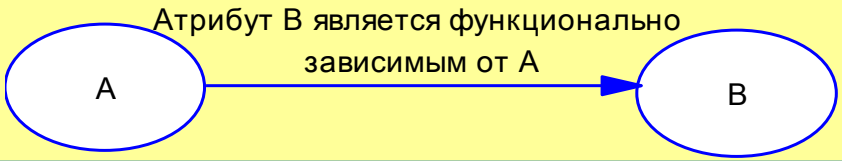


Диаграмма функциональной зависимости
(атрибут A является детерминантом атрибута B)

- **Детерминант.** Детерминантом функциональной зависимости называется атрибут или группа атрибутов, расположенная на диаграмме функциональной зависимости слева от стрелки.
- Функциональная зависимость называется ***тривиальной,*** если она остается справедливой при любых условиях.
- зависимость является тривиальной, если и только если в правой части выражения, определяющего зависимость, приведено подмножество (но необязательно собственное подмножество) множества, которое указано в левой части (детерминанте) выражения
- Функциональная зависимость A $\rightarrow$ B является ***полной*** функциональной зависимостью, если удаление какого-либо атрибута из A приводит к утрате этой зависимости. Функциональная зависимость A $\rightarrow$ B называется ***частичной,*** если в A есть некий атрибут, при удалении которого эта зависимость сохраняется.

Аксиомы Армстронга

- Набор правил вывода, называемый *аксиомами Армстронга,* показывает способы вывода новых функциональных зависимостей из заданных.
- Предположим, что A, B и C — подмножества атрибутов отношения R.

аксиомы Армстронга:

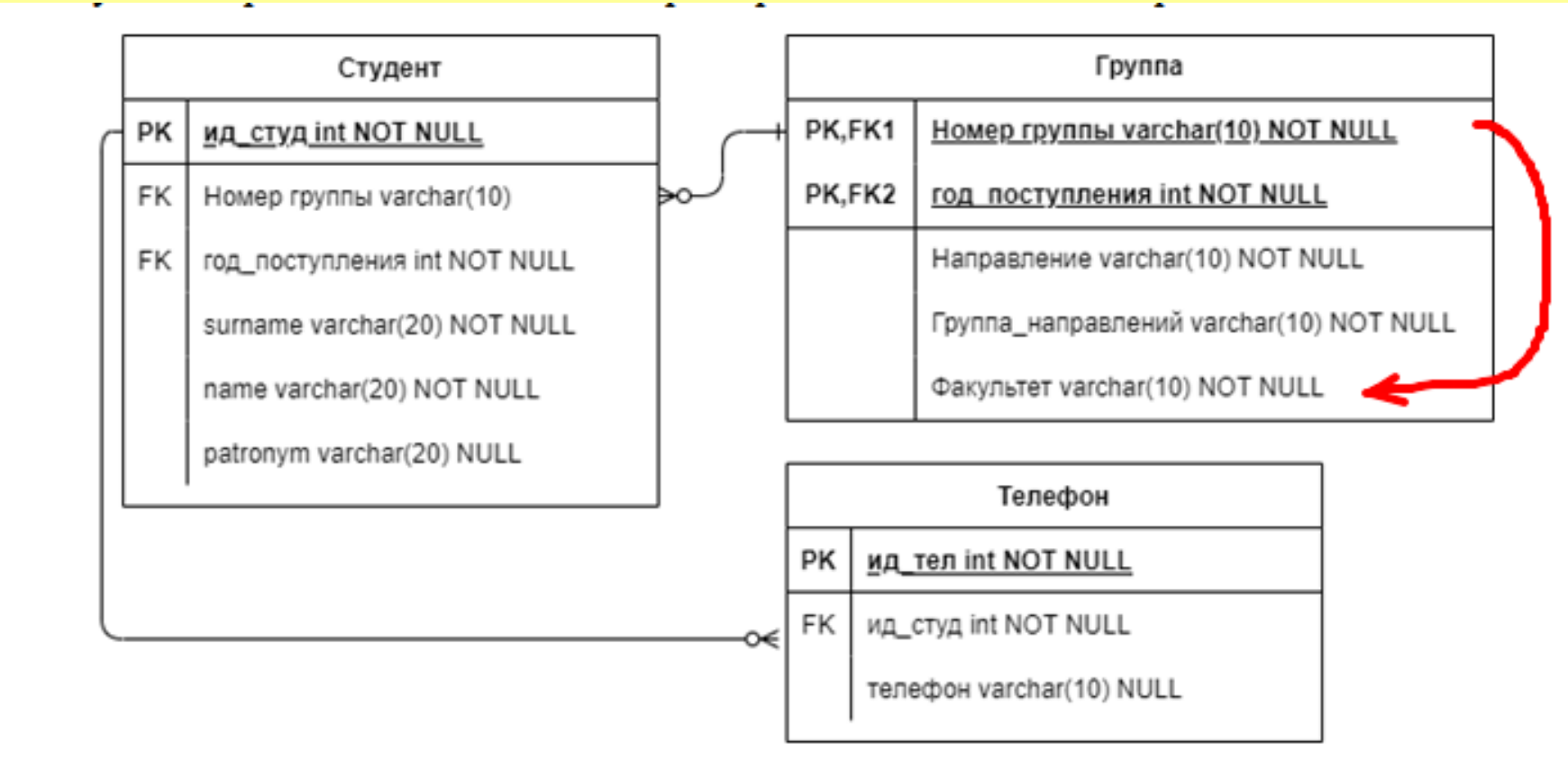
- 1. Рефлексивность. Если B — подмножество A, то $A \rightarrow B$.
- 2. Дополнение. Если $A \rightarrow B$, то $A, C \rightarrow B, C$.
- 3. Транзитивность. Если $A \rightarrow B$ и $B \rightarrow C$, то $A \rightarrow C$.
- 4. Самоопределение. $A \rightarrow A$.
- 5. Декомпозиция. Если $A \rightarrow B, C$, то $A \rightarrow B$ и $A \rightarrow C$.
- 6. Объединение. Если $A \rightarrow B$ и $A \rightarrow C$, то $A \rightarrow B, C$.
- 7. Композиция. Если $A \rightarrow B$ и $C \rightarrow D$, то $A, C \rightarrow B, D$.

Вторая нормальная форма (2НФ)

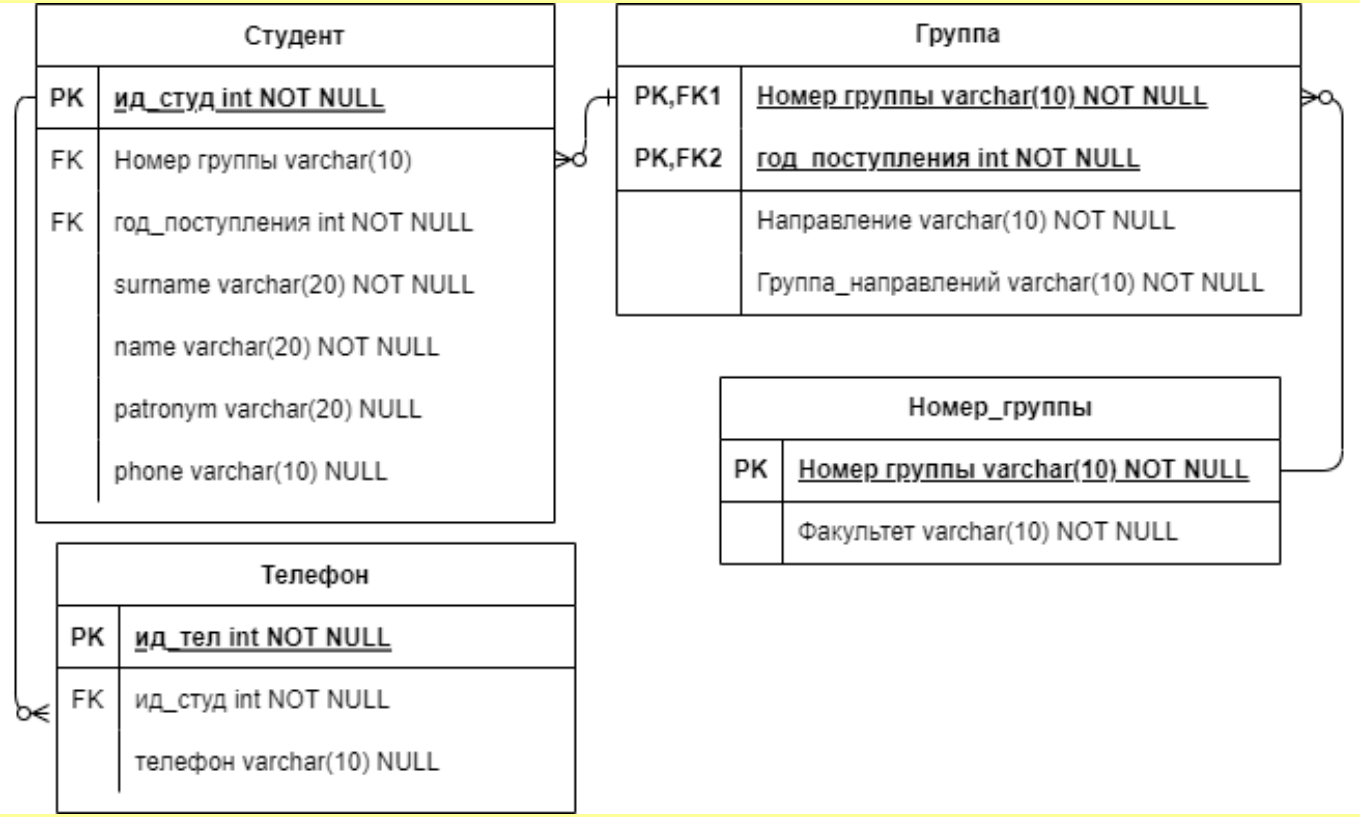
- **Вторая нормальная форма (2НФ) —** Отношение, находящееся в первой нормальной форме, в котором каждый атрибут, отличный от атрибута первичного ключа, является полностью функционально независимым от любого потенциального ключа.
- Вторая нормальная форма (2НФ). Отношение, которое находится в первой нормальной форме и каждый атрибут которого, не входящий в состав первичного ключа, характеризуется полной функциональной зависимостью от этого первичного ключа.

Вторая нормальная форма применяется к отношениям с составными ключами, т.е. к таким отношениям, первичный ключ которых состоит из двух или нескольких атрибутов.
Отношение в 1 НФ с первичным ключом на основе единственного атрибута всегда находится, по крайней мере, в форме 2НФ.

База данных в 1НФ

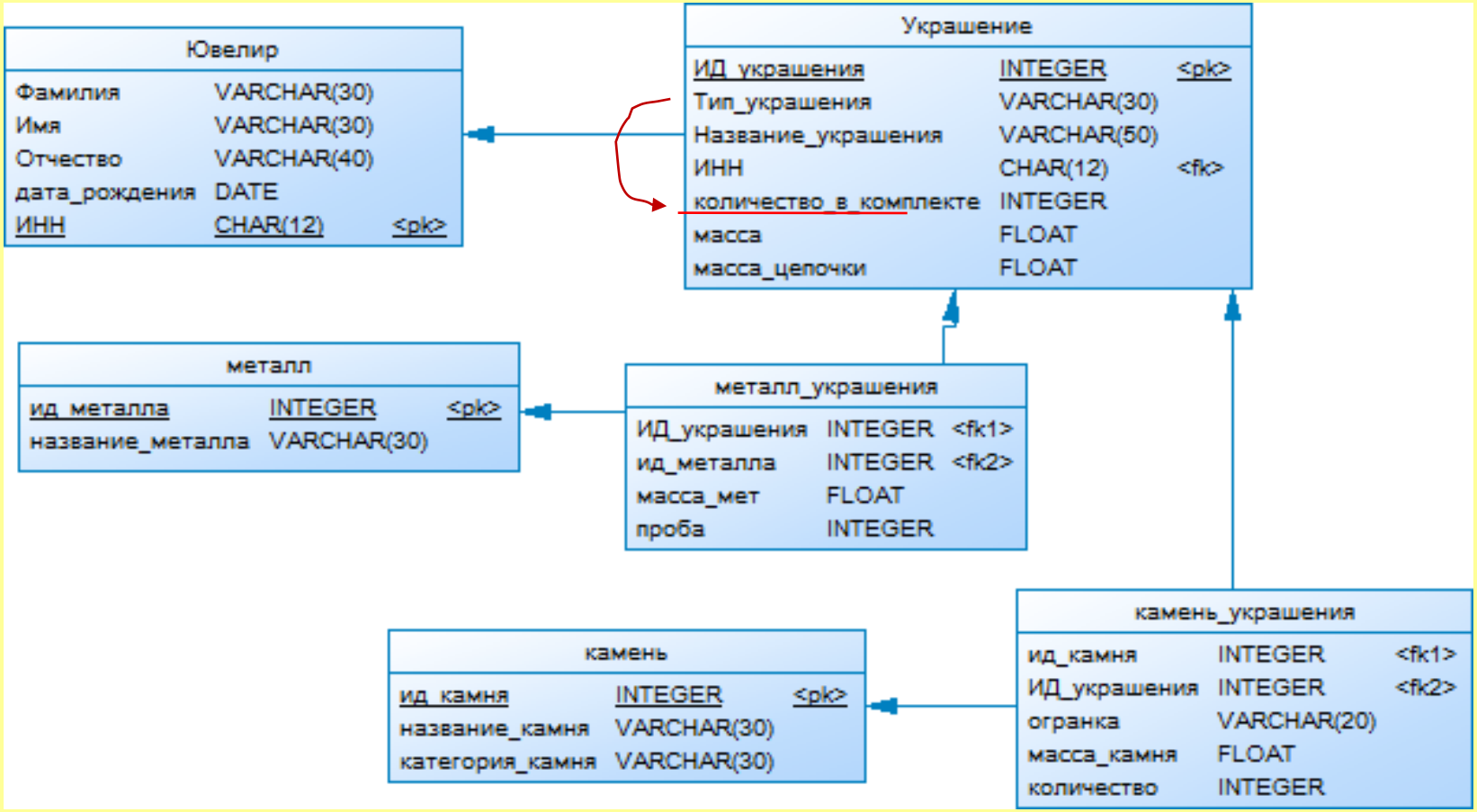


База данных в 2НФ

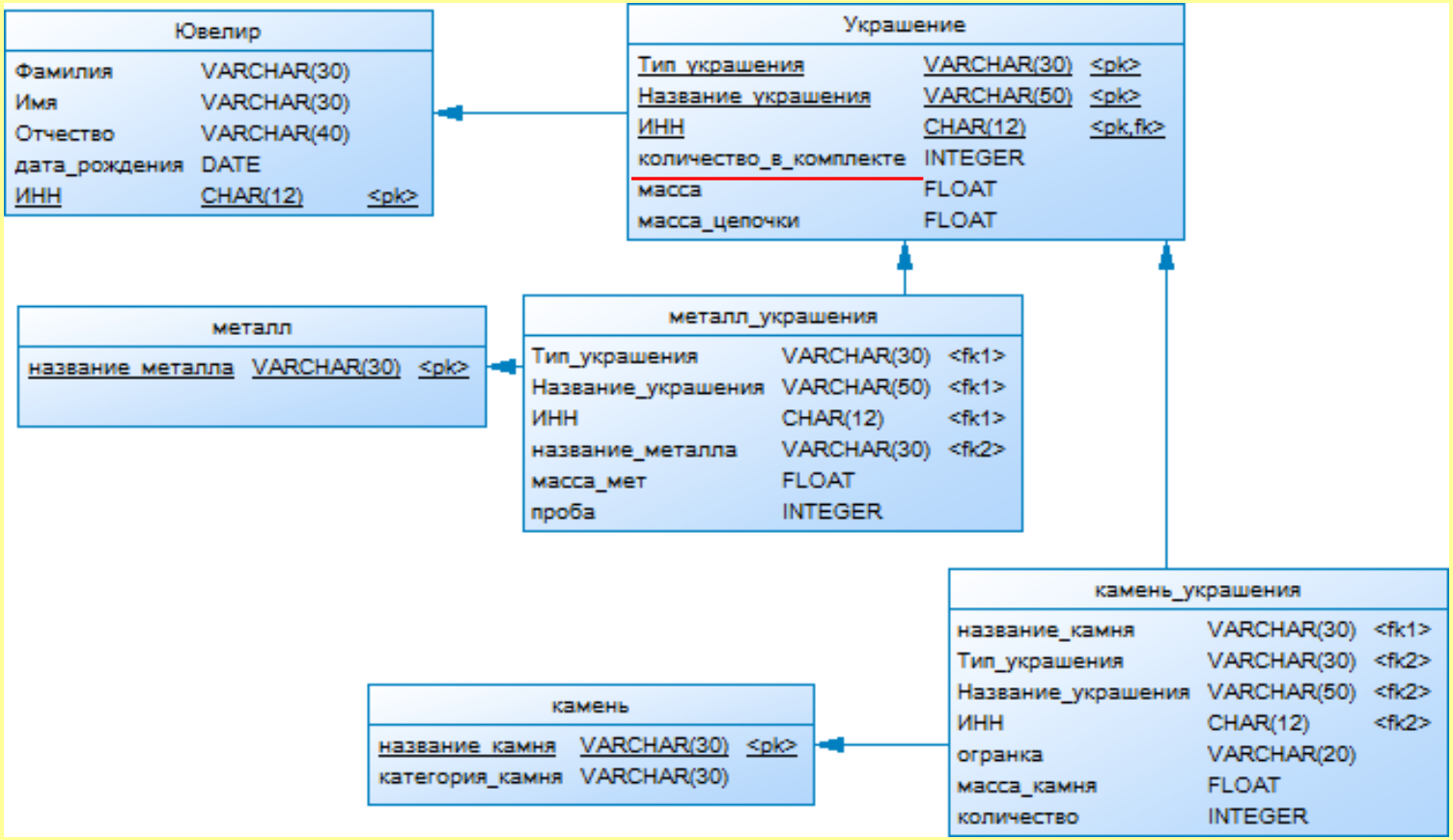


Приведение к 2НФ

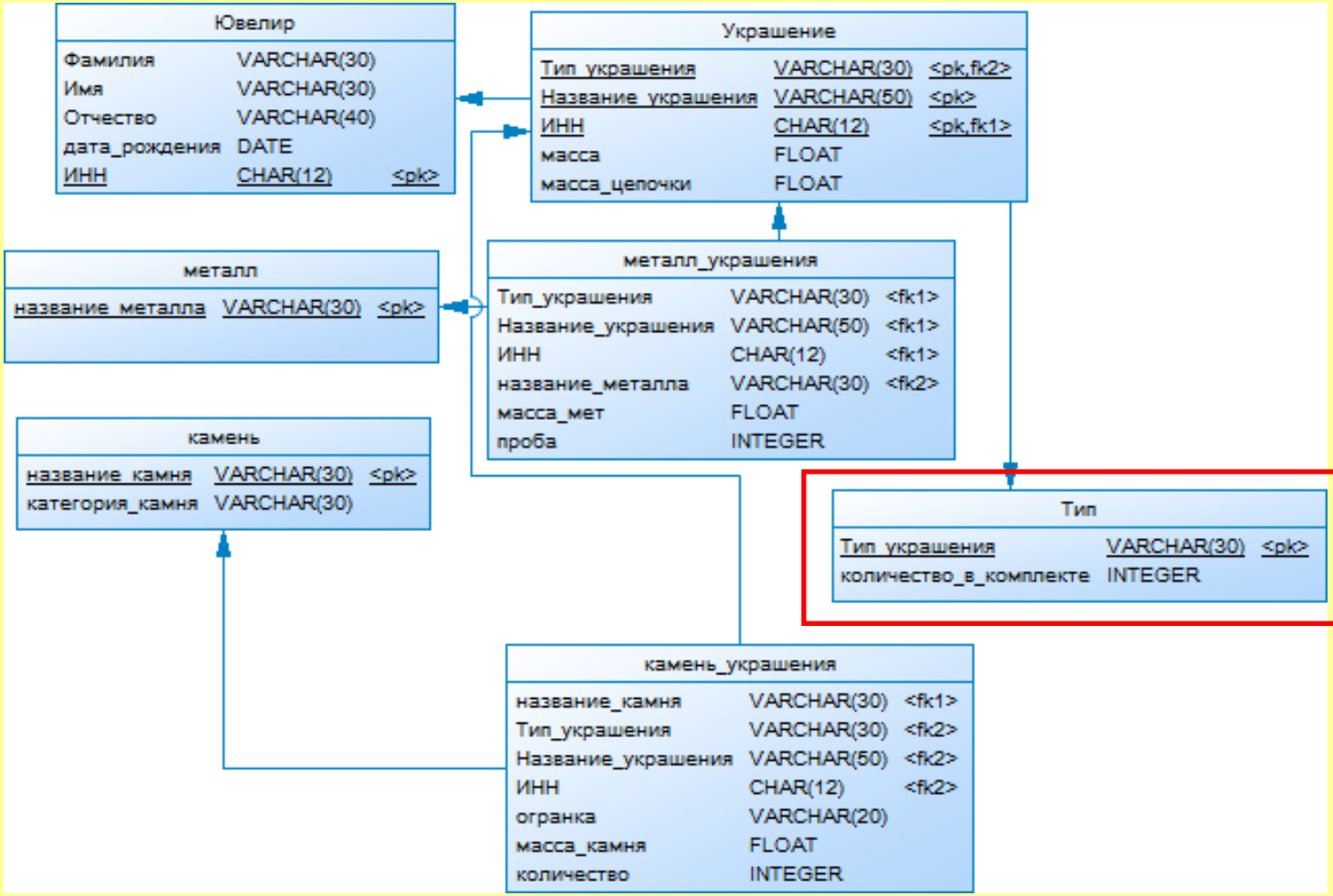
БД в 1НФ



База данных без суррогатных ключей



База данных в 2НФ без суррогатных ключей



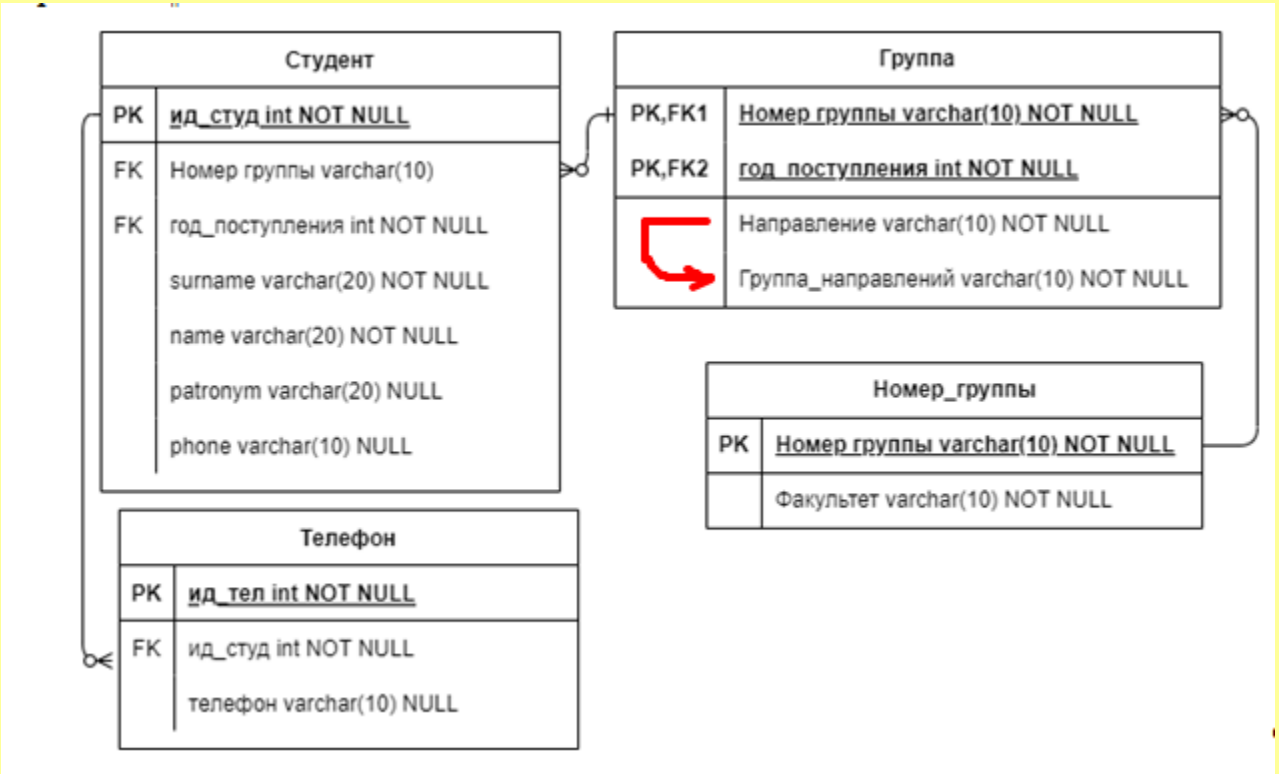
Третья нормальная форма (2НФ)

- **Транзитивная зависимость.** Если для атрибутов A, B и C некоторого отношения существуют зависимости вида
  $A \rightarrow B$ и $B \rightarrow C$,
  это означает, что атрибут C транзитивно зависит от атрибута A через атрибут B (при условии, что атрибут A функционально не зависит ни от атрибута B, ни от атрибута C).
- **Третья нормальная форма (3НФ) —** Отношение, которое находится а первой и во второй нормальных формах и не имеет атрибутов, не входящих в первичный ключ атрибутов, которые находились бы в транзитивной функциональной зависимости от этого первичного ключа.

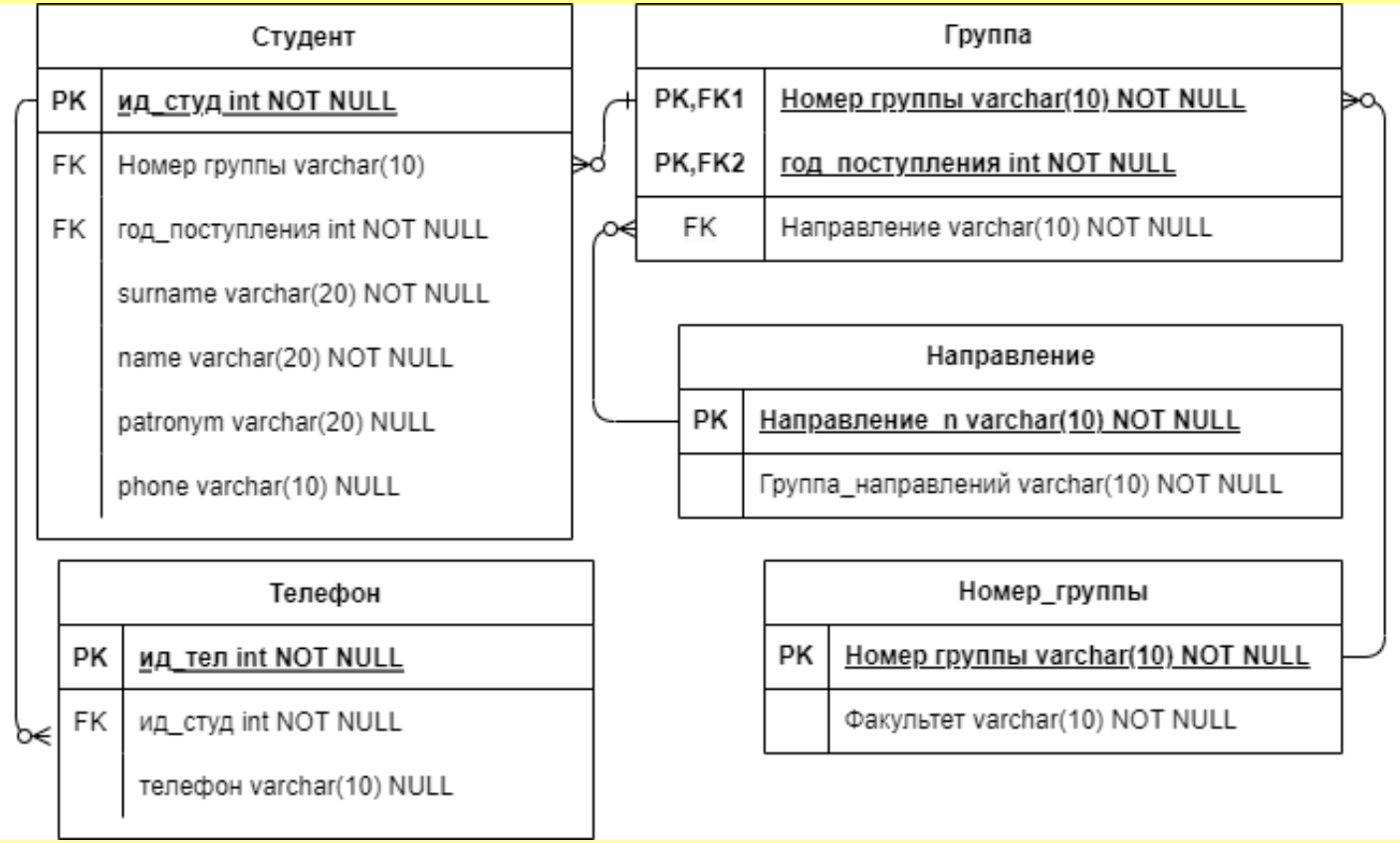
Или если более строго

- **третья нормальная форма (3НФ).** Отношение, находящееся в первой и второй нормальной форме, в котором ни один атрибут, отличный от атрибута первичного ключа, не является транзитивно зависимым ни от одного потенциального ключа.

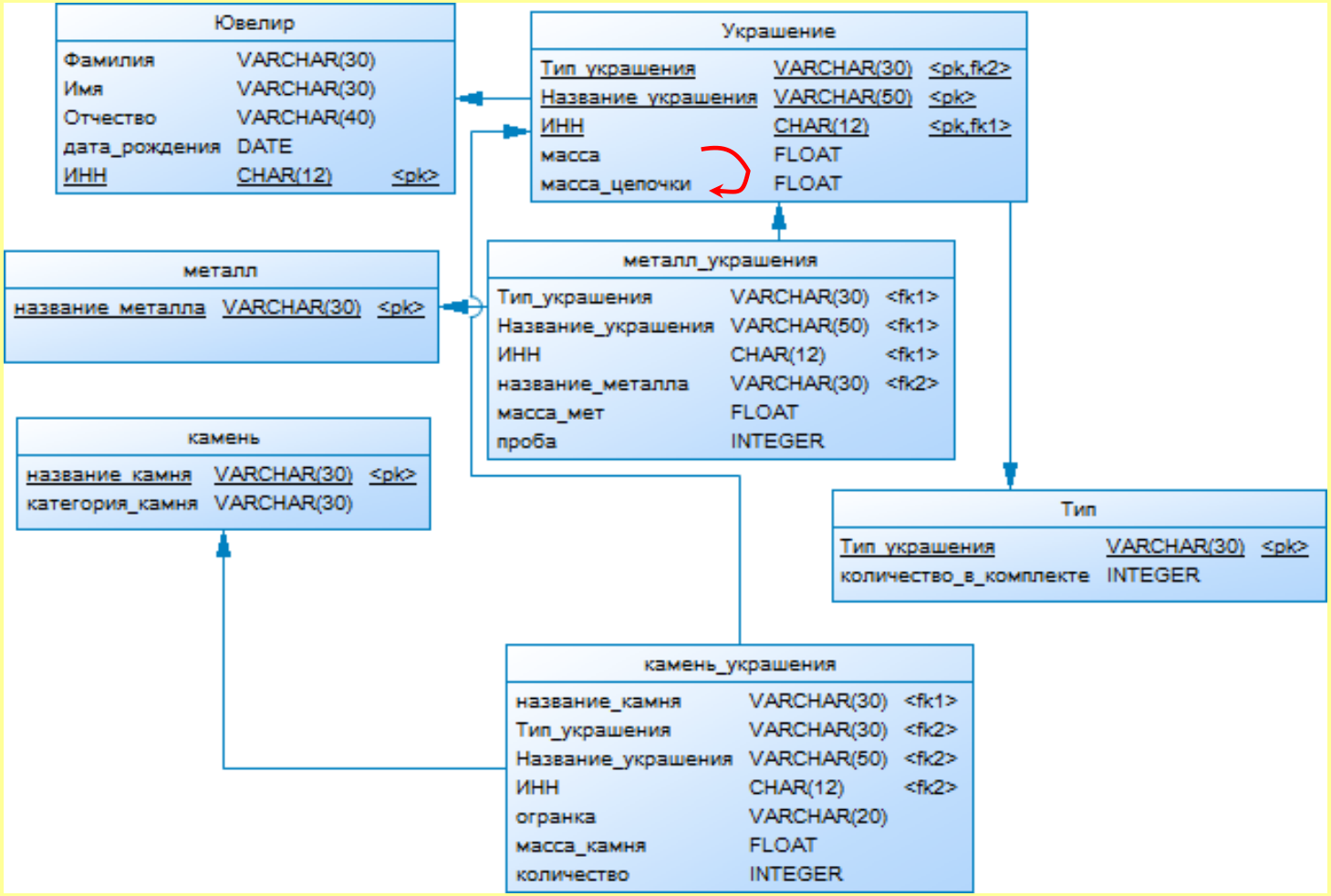
База данных в 2НФ



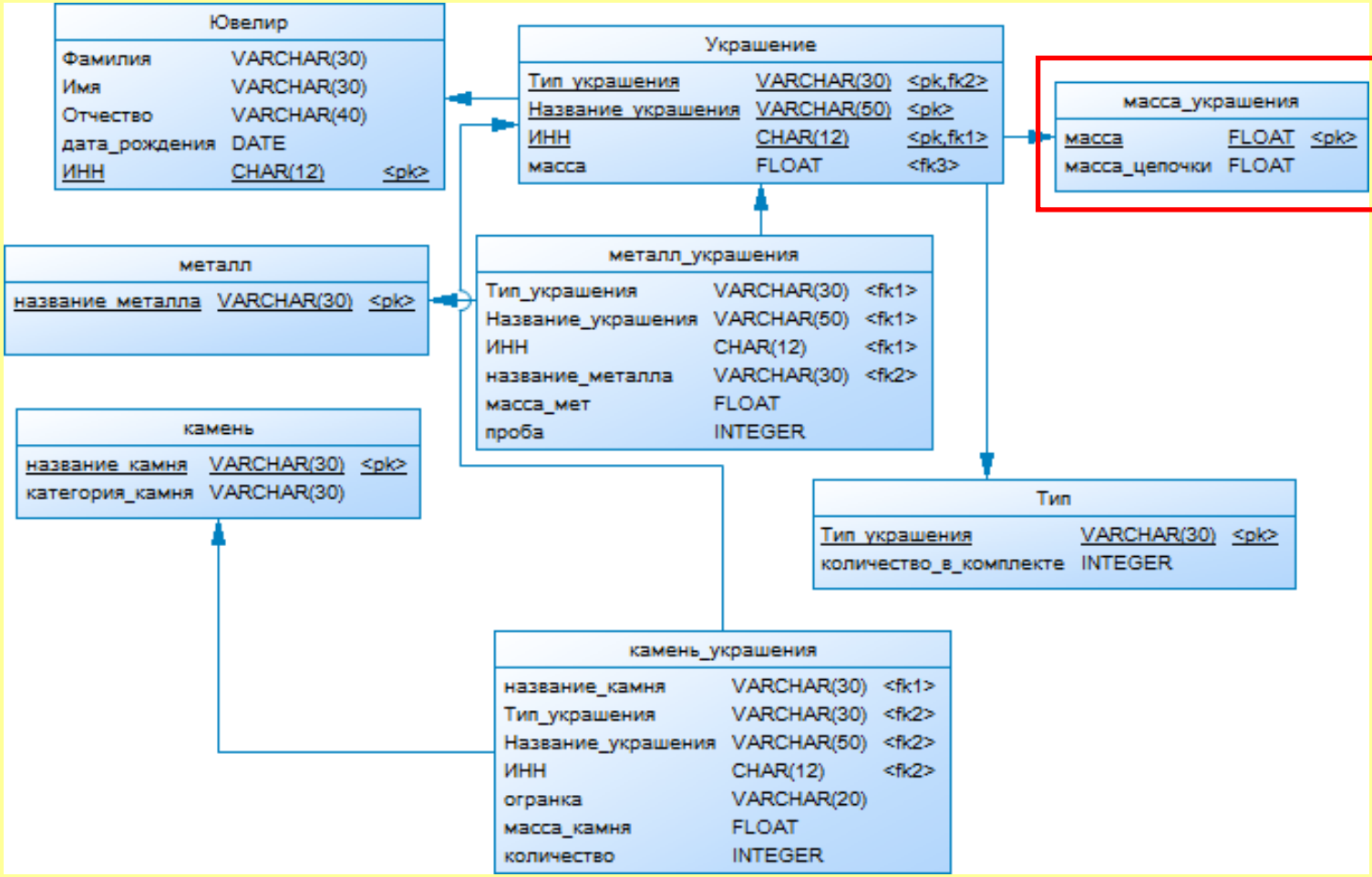
База данных в 3НФ



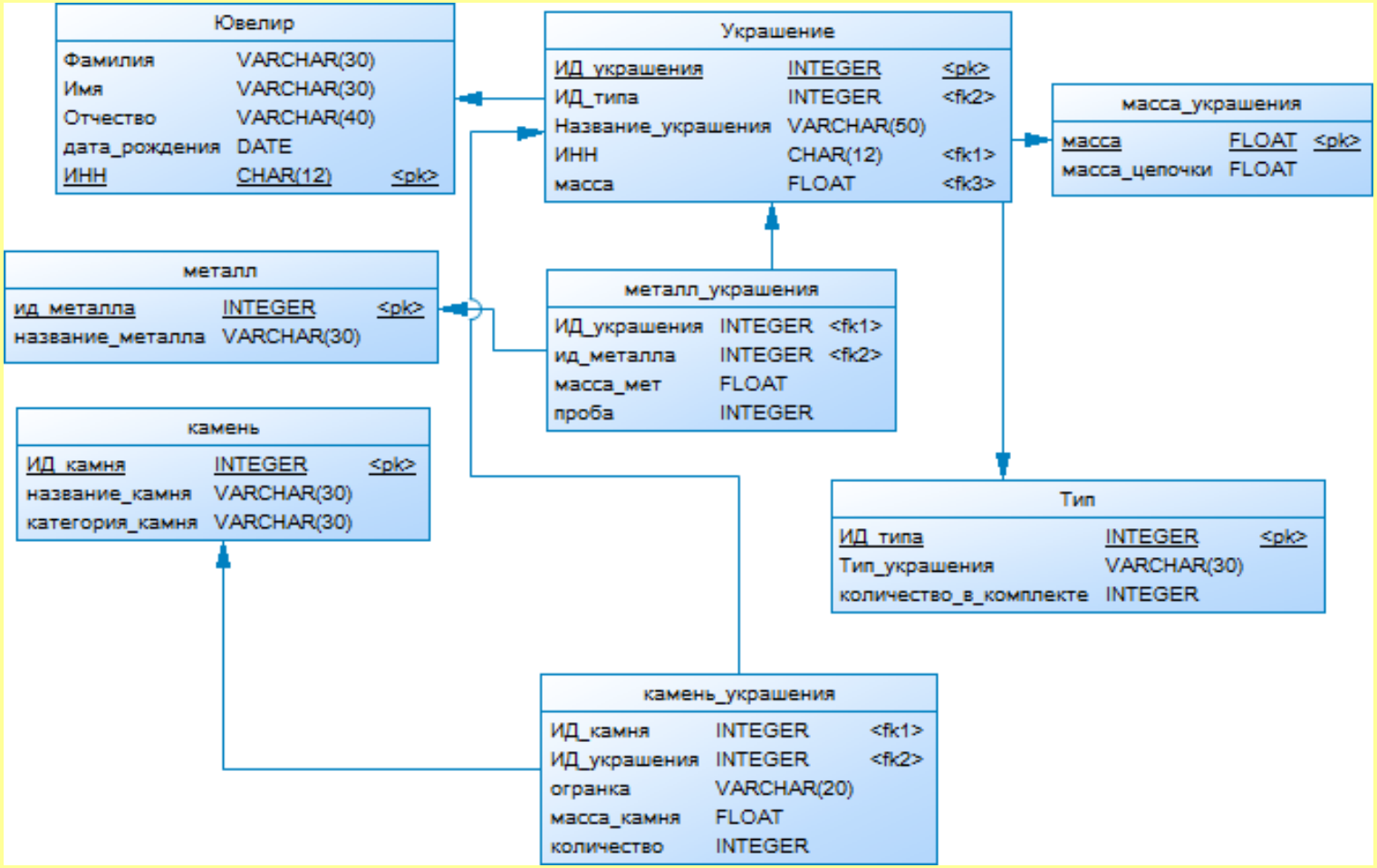
База данных в 2НФ без суррогатных ключей



База данных в 3НФ без суррогатных ключей



БД в 3НФ


Ювелир
Фамилия VARCHAR(30)
Имя VARCHAR(30)
Отчество VARCHAR(40)
дата_рождения DATE
ИНН CHAR(12) <pk>
Украшение
ИД_украшения INTEGER <pk>
ИД_типа INTEGER <fk2>
Название_украшения VARCHAR(50)
ИНН CHAR(12) <fk1>
масса FLOAT <fk3>
масса_украшения
масса FLOAT <pk>
масса_цепочки FLOAT
металл
ид_металла INTEGER <pk>
название_металла VARCHAR(30)
металл_украшения
ИД_украшения INTEGER <fk1>
ид_металла INTEGER <fk2>
масса_мет FLOAT
проба INTEGER
камень
ИД_камня INTEGER <pk>
название_камня VARCHAR(30)
категория_камня VARCHAR(30)
Тип
ИД_типа INTEGER <pk>
Тип_украшения VARCHAR(30)
количество_в_комплекте INTEGER
камень_украшения
ИД_камня INTEGER <fk1>
ИД_украшения INTEGER <fk2>
огранка VARCHAR(20)
масса_камня FLOAT
количество INTEGER

Нормальная форма Бойса-Кодда (НФБК)

- **Нормальная форма Бойса-Кодда (НФБК).** Отношение находится в НФБК тогда и только тогда, когда каждый его детерминант является потенциальным ключом.
- Различие между ЗНФ и НФБК заключается в том, что функциональная зависимость $A \rightarrow B$ допускается в отношении ЗНФ, если атрибут B является первичным ключом, а атрибут A не обязательно является потенциальным ключом. Тогда как в отношении НФБК эта зависимость допускается *только* тогда, когда атрибут A является потенциальным ключом.

Нормальная форма Бойса-Кодда (НФБК)

Отношение ClientInterview

clientNo	interviewDate	interviewTime	staffNo	roomNo
CR76	13-May-19	10:30	SG5	G101
CR56	13-May-19	12:00	SG5	G101
CR74	13-May-19	12:00	SG37	G102
CR56	l-Jul-19	10:30	SG5	G102

три потенциальных ключа:

- (clientNo, interviewDate) ,
- (staffNo, interviewDate, interviewTime)
- (roomNo, interviewDate, interviewTime).

Первичный ключ

(clientNo, interviewDate).

Функциональные зависимости

clientNo	interviewDate	interviewTime	staffNo	roomNo
CR76	13-May-19	10:30	SG5	G101
CR56	13-May-19	12:00	SG5	G101
CR74	13-May-19	12:00	SG37	G102
CR56	l-Jul-19	10:30	SG5	G102

Обозначение	Зависимость	Описание
Ф31	clientNo, interviewDate → interviewTime, staffNo, roomNo	Первичный ключ
Ф32	staffNo, interviewDate, interviewTime → clientNo	Потенциальный ключ
Ф33	roomNo, interviewDate, interviewTime → staffNo,clientNo	Потенциальный ключ
Ф34	staffNo, interviewDate → roomNo	

Приведение к НФБК

clientNo	interviewDate	interviewTime	staffNo
CR76	13-May-19	10:30	SG5
CR56	13-May-19	12:00	SG5
CR74	13-May-19	12:00	SG37
CR56	l-Jul-19	10:30	SG5

interviewDate	staffNo	roomNo
13-May-19	SG5	G101
13-May-19	SG5	G101
13-May-19	SG37	G102
l-Jul-19	SG5	G102

- Обратите внимание на то, что в примере при создании двух новых отношений НФБК на основе исходного отношения ClientInterview "утрачивается" следующая функциональная зависимость: roomNo, interviewDate, interviewTime $\rightarrow$ staffNo, clientNo (зависимость Ф33), поскольку детерминант этой зависимости больше не будет находиться в том же отношении, что и определяемые им атрибуты.

Четвертая нормальная форма (4НФ)

- Однако в результате теоретических исследований был выявлен еще один тип зависимости — *многозначная зависимость* (Multi-Valued Dependency — MVD)
- **Многозначная зависимость** . Представляет такую зависимость между атрибутами отношения (например, A, B и C), что каждое значение A представляет собой множество значений для B и множество значений для C. Однако множества значений для в и с не зависят друг от друга.

$A \rightarrow B$

$A \rightarrow C$

- Многозначная зависимость может быть дополнительно определена как *тривиальная* или *нетривиальная.* Например, многозначная зависимость $A \rightarrow$ B некоторого отношения R определяется как тривиальная, если атрибут B является подмножеством атрибута A или A и B = R. И наоборот, многозначная зависимость определяется как нетривиальная, если ни то ни другое условие не выполняется. Тривиальная многозначная зависимость (МЗЗ) не накладывает никаких ограничений на данное отношение, а нетривиальная — накладывает.

- **Четвертая нормальная форма (4НФ)** Отношение в нормальной форме Бойса-Кодда, которое не содержит нетривиальных многозначных зависимостей.

Четвертая нормальная форма (4НФ)

branchNo	sName	oName
вооз	Ann Beech	Carol Farrel
вооз	David Ford	Carol Farrel
вооз	Ann Beech	Tina Murphy
вооз	David Ford	Tina Murphy

- представленное в таблице отношение BranchStaffOwner содержит имена сотрудников (sName) и владельцев недвижимости (oName) определенного отделения компании (branchNo).
- в данном отношении существует многозначная зависимость, так как в нем содержатся две независимые связи типа 1:*
- branchNo —> sName
- branchNo —> oName

Приведение к 4 НФ

branchNo	**sName**	**oName**
B003	Ann Beech	Carol Farrel
B003	David Ford	Carol Farrel
B003	Ann Beech	Tina Murphy
B003	David Ford	Tina Murphy

branchNo	sName
B003	Ann Beech
B003	David Ford

branchNo	oName
B003	Carol Farrel
B003	Tina Murphy

Пятая нормальная форма (5НФ)

- **Зависимость соединения без потерь.** Свойство декомпозиции, которое гарантирует отсутствие фиктивных строк при восстановлении первоначального отношения с помощью операции естественного соединения.
- **Пятая нормальная форма (5НФ).** Отношение без зависимостей соединения.
- Пятая нормальная форма (5НФ), которая также называется *проективно-соединительной нормальной формой,* или ПСНФ (Project-Join Normal Form -PJNF), означает, что отношение в такой форме не имеет зависимостей соединения

Пятая нормальная форма (5НФ)

КлассПредметУчитель

Класс	Предмет	Учитель
9А	Алгебра	Иванов
9А	Физика	Петров

Недопустимое состояние

Класс	Предмет	Учитель
9А	Алгебра	Иванов
9А	Физика	Петров
9Б	Алгебра	Петров

Допустимое состояние

Класс	Предмет	Учитель
9А	Алгебра	Иванов
9А	Физика	Петров
9Б	Алгебра	Петров
9А	Алгебра	Петров

- Это отношение описывает школьные классы(Класс), у которых будут вестись определенные предметы (Предмет), которые ведут учителя (Учитель) у этих классов. Кроме того, если для какого-то класса (C) требуется некоторый предмет (D), некий учитель(T) занимается ведением такого предмета (D), и этот учитель (T) уже вел хотя бы один предмет у класса (C), то этому же учителю (T) также будет поручено вести необходимый предмет (D) у класса (C).
- Если для объекта недвижимости класса 9А требуется предмет «Алгебра» (согласно данным в строке 1), ведением у 9А занимается учитель Петров (согласно данным в строке 2), учитель Петров занимается ведением предмета «Алгебра» (согласно данным в строке 3), Тогда учитель Петров должен также вести предмет «Алгебра» у 9А класса. Этот пример наглядно иллюстрирует циклический характер ограничения, которое распространяется на отношение КлассПредметУчитель.

Приведение к 5НФ

КлассПредметУчитель

Класс	Предмет	Учитель
9А	Алгебра	Иванов
9А	Физика	Петров

Недопустимое состояние

Класс	Предмет	Учитель
9А	Алгебра	Иванов
9А	Физика	Петров
9Б	Алгебра	Петров

Допустимое состояние

Класс	Предмет	Учитель
9А	Алгебра	Иванов
9А	Физика	Петров
9Б	Алгебра	Петров
9А	Алгебра	Петров

Класс	Предмет
9А	Алгебра
9А	Физика
9Б	Алгебра

Класс	Учитель
9А	Иванов
9А	Петров
9Б	Петров

Предмет	Учитель
Алгебра	Иванов
Физика	Петров
Алгебра	Петров

Доменно-ключевая нормальная форма (DKNF)

- Каждое ограничение в связях между таблицами должно зависеть только от ограничений ключа и ограничений домена, где домен представляет набор допустимых значений для столбца.
- Эта форма предотвращает добавление недопустимых данных путем установки ограничения на уровне отношений между таблицами, но не на уровне таблиц или столбцов

Ограничения

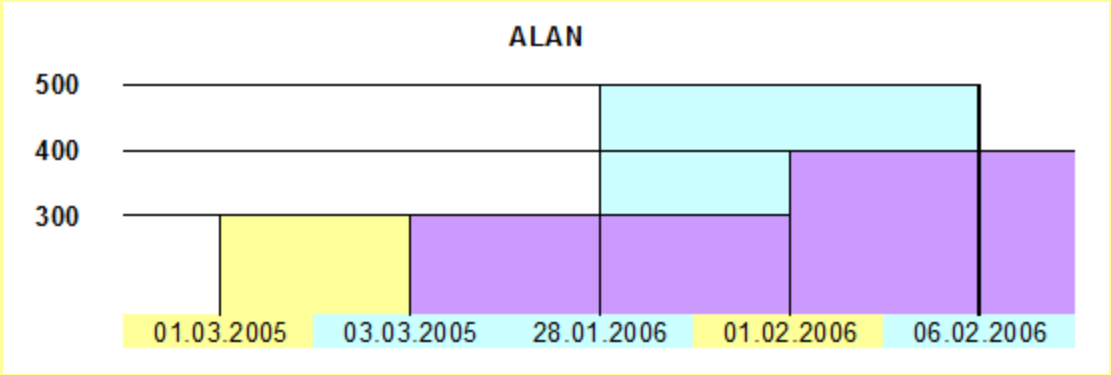
- **Ограничение домена** – это ограничение, предписывающее использование для определенного атрибута значений только из некоторого заданного домена (набора значений).
- **Ограничение ключа** – это ограничение, утверждающее, что некоторый атрибут или комбинация атрибутов представляет собой потенциальный ключ.
- Таким образом, **требование доменно-ключевой** нормальной формы заключается в том, чтобы каждое наложенное ограничение на таблицу являлось логическим следствием ограничений доменов и ограничений ключей, которые накладываются на данную таблицу.
- Таблица, находящаяся в доменно-ключевой нормальной форме, обязательно находится в 5NF

Темпоральные(временные) данные

- В темпоральных БД каждый кортеж содержит информацию о состоянии моделируемого объекта, а также о времени, когда эта информация была записана в БД. Такое размытие информации об одном логическом объекте по нескольким кортежам было названо *вертикальной темпоральной аномалией*
- **Действительное (модельное) время** показывает период в прошлом, настоящем или будущем, когда факт являлся истинным в моделируемом мире
- **Транзакционное время** показывает период в прошлом или настоящем, когда данная запись была представлена в базе данных.

Линии времени

Табельный номер	Зарплата	действительное время	транзакционное время
ALAN	500	с 1 января 2006	с 20 декабря 2005
BOB	300	с 1 марта 2005 по 31 января 2006	с 3 марта 2005 по 27 января 2006
BOB	500	с 1 февраля 2006	с 28 января 2006 по 5 февраля 2006
BOB	400	с 1 февраля 2006	с 6 февраля 2006



Проблемы разработки системы с темпоральными данными

- Темпоральные данные нужны для анализа ситуации
- Актуальные данные должны быть доступны всегда без снижения производительности
- Хранение темпоральных данных увеличивает сложность запросов, снижает их производительность и требует больше места
- Оперативное управление требует большой скорости манипулирования данными

Модели темпоральных данных

- Модель Р. Снодграса
  R = (A1, ..., An, Ts, Te, Vs, Ve)
- Модель К. Дженсена
  *R = (A1, ..., An, Vs, Ve, T, Op). Op::=*I|D
- Модель Дж. Бен-Зви
  R = (A1, ..., An, Tes, Trs, Tee, Tre, Td)
- Модель С. Гадия
  R = {([Ts, Te][Vs, Ve]A1)}, ..., {([Ts, Te][Vs, Ve]An)}
- Модель Е. МакКензи
  это последовательность состояний в модельном времени, проиндексированная транзакционным временем *R = (VR, T), VR = (A1V1, ..., AnVn)*

Модель Р. Снодграса

- *R(A1, ...,An, T), где A1, ..., An — набор атрибутов, T — битемпоральный атри*бут.
- Тогда *R можно представить в виде R = (A1, ..., An, Ts, Te, Vs, Ve),*
- где *Ts, Te, Vs, Ve — атомарные темпоральные атрибуты, содержащие*
- дату начала и окончания транзакционного и модельного времени.

Модель К. Дженсена

- *R = (A1, ..., An, Vs, Ve, T, Op). Op::=*I|D
- кортежи доступны только для чтения.
- Битемпоральное отношение *R с набором атрибутов A1, ..., An может быть* представлено в следующем виде:
  R = (A1, ..., An, Vs, Ve, T, Op).
- *Как и в* модели Снодграса, атрибуты
- *Vs и Ve хранят даты начала и окончан*ия актуальности факта в моделируемой реальности соответственно,
- атрибут *T — информацию о времени внесения кортежа в журнал из*менений.
- Запросы на создание и удаление кортежей обозначаются в атрибуте *Op соответствующими символами I (вставка) и D (удале*ние). Модификации данных представляют собой пару запросов (удаление и создание записи) с одинаковым атрибутом *T.*

Модель Дж. Бен-Зви

- *R = (A1, ..., An, Tes, Trs, Tee, Tre, Td)*
- Пусть битемпоральное отношение *R состоит из* набора атрибутов *A1, ..., An, T,*
- *где T — темпоральный атрибут, определ*енный на множестве битемпоральных элементов.
- Тогда *R может быть* представлено следующим образом: *R = (A1, ..., An, Tes, Trs, Tee, Tre, Td),*
- где *Tes — атрибут времени, когда значение атрибута кортежа* становится актуальным;
- *Trs — атрибут, хранящий информацию о том, к*огда атрибут *Tes был сохранен в БД;*
- *Tre — атрибут, хранящий инфор*мацию о том, когда факт перестает быть актуальным в моделируемой реальности;
- *Tee — атрибут времени, когда Tre было зафиксировано в* БД;
- *Td — атрибут, указывающий на время, когда запись была логичес*ки удалена из БД.

Модель С. Гадия

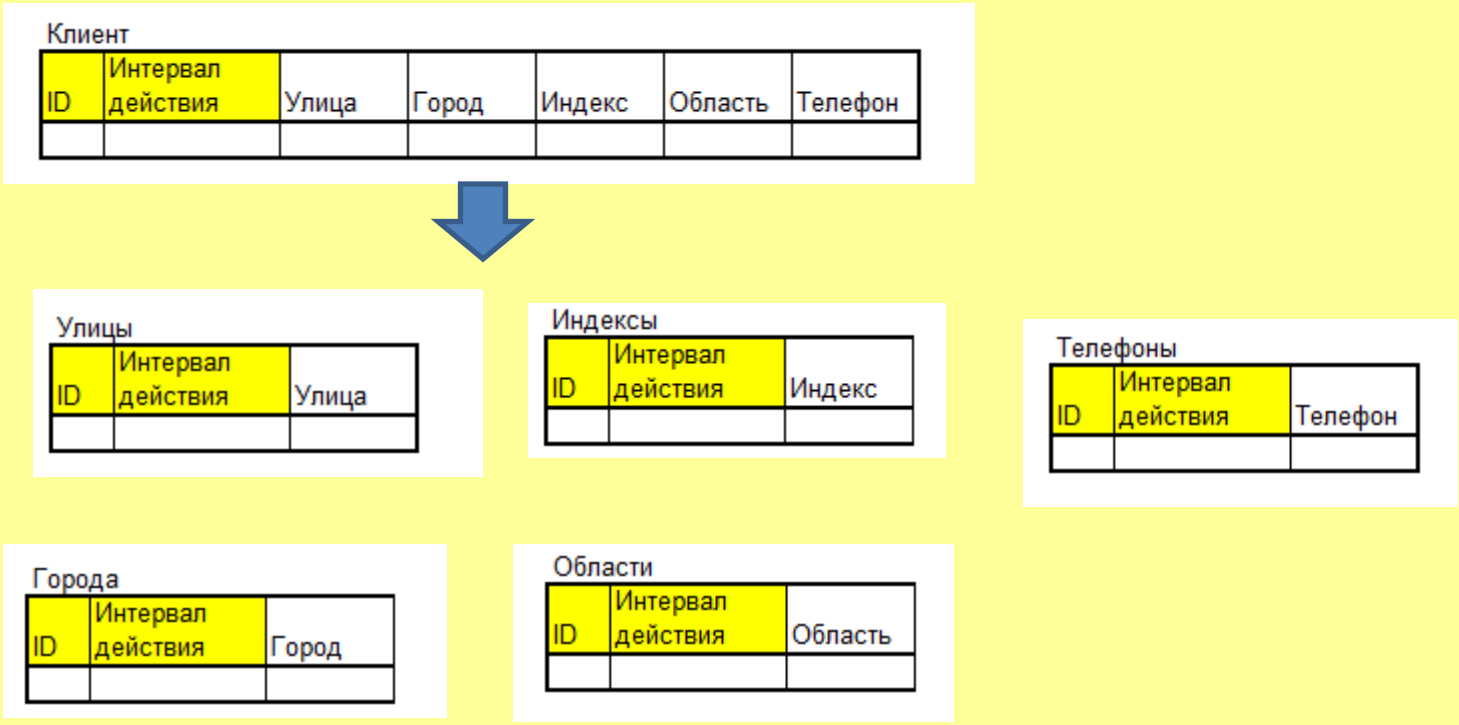
- *R = {([Ts, Te][Vs, Ve]A1)}, …, {([Ts, Te][Vs, Ve]An)}*
- Данный подход предполагает наличие битемпоральных меток у каждого из атрибутов кортежа, что обеспечивает возможность более гибкого моделирования реальности.
- Дано би темпоральное отношение *R(A1, ..., An, T), где T — атрибут,* определенный на множестве битемпоральных элементов. Тогда битемпоральное отношение *R может быть представлено в виде отношений,* где каждый из атрибутов имеет свою темпоральную метку:
  R = {([Ts, Te][Vs, Ve]A1)}, …, {([Ts, Te][Vs, Ve]An)}.
- Кортеж состоит из *n элементов. Каждый элемент представляет* собой тройку значений: транзакционное время [*Ts, Te], модельное* время [*Vs, Ve] и значение атрибута Ai.*

Модель Е. МакКензи

- *R = (VR, T), VR = (A1V1, ..., AnVn)*
- В данной модели битемпоральное отношение — это последовательность состояний в модельном времени, проиндексированная транзакционным временем. В кортежах с модельным временем атрибуты имеют свои темпоральные метки.
- Битемпоральное отношение *R с набором атрибутов A1, …, An может быть* представлено в виде отношения, в котором каждый атрибут помечается временной меткой: *R = (VR, T),*
- *где VR — отношение в модельном* времени;
- *T — транзакционное время.*
- *Схема состояний отношения мо*дельного времени имеет вид
- *VR = (A1V1, ..., AnVn),*
- *где A1, ..., An* — набор атрибутов;
- *V1, Vn — атрибут модельного времени, каждый из* которых соответствует атрибутам *A1, ..., An и обозначает время акту*альности его значения в моделируемой реальности.

6 нормальная форма

- Переменная отношения находится в шестой нормальной форме тогда и только тогда, когда она удовлетворяет всем нетривиальным зависимостям соединения. Из определения следует, что переменная находится в 6НФ тогда и только тогда, когда она неприводима, то есть не может быть подвергнута дальнейшей декомпозиции без потерь.



Денормализация

- **нормализация** переменной отношения R означает ее замену множеством таких проекций Rl, R2, ..., Rn, что результатом обратного соединения проекций Rl, R2, ..., Rn обязательно будет значение R. **Конечной целью нормализации является сокращение степени избыточности данных** за счет приведения проекций Rl, R2, ..., Rn к максимально высокому уровню нормализации.
- Теперь можно перейти к определению понятия **денормализации.** Пусть R1, R2, Rn является множеством переменных отношения. Тогда **денормализацией** этих переменных отношения называется такая замена переменных отношения их соединением R, что для всех возможных значений i (где i = 1, ..., n) выполнение проекции R по атрибутам Ri обязательно снова приводит к созданию значений Ri. Конечной целью денормализации является увеличение степени избыточности данных за счет приведения переменной отношения R к более низкому уровню нормализации по сравнению с исходными переменными отношения Rl, R2, ..., Rn.
- **Цель.** Определение необходимости ввода контролируемой избыточности за счет ослабления условий нормализации для повышения производительности системы.

Проблемы денормализации

- начиная денормализацию, трудно сказать, когда ее следует прекратить.
- проблемы избыточности и аномалиями обновления, которые возникают из-за того, что приходится иметь дело с не полностью нормализованными переменными отношения.
- Когда речь идет о том, что денормализация "способствует достижению высокой производительности", фактически подразумевается, что она способствует достижению высокой *производительности некоторых конкретных приложений.* Любая выбранная физическая структура, которая прекрасно подходит для одних приложений с точки зрения их производительности, может оказаться совершенно непригодной для других.

возможность применения денормализации

1. Объединение таблиц со связями типа "один к одному" (1:1).
2. Дублирование неключевых атрибутов в связях "один ко многим" (1:*) для уменьшения количества соединений.
3. Дублирование атрибутов внешнего ключа в связях "один ко многим" (1:*) для уменьшения количества соединений.
4. Дублирование атрибутов в связях "многие ко многим" (1:*) для уменьшения количества соединений.
5. Введение повторяющихся групп полей.
6. Объединение справочных таблиц с базовыми таблицами.
7. Создание таблиц из данных, содержащихся в других таблицах.

Особенности денормализации

- денормализация может усложнить физическую реализацию системы;
- денормализация часто приводит к снижению гибкости;
- денормализация может ускорить чтение данных, но при этом замедлить обновление записей.