

Базы данных

Лекция 5.

Проектирование баз данных, ссылочная целостность, нормализация

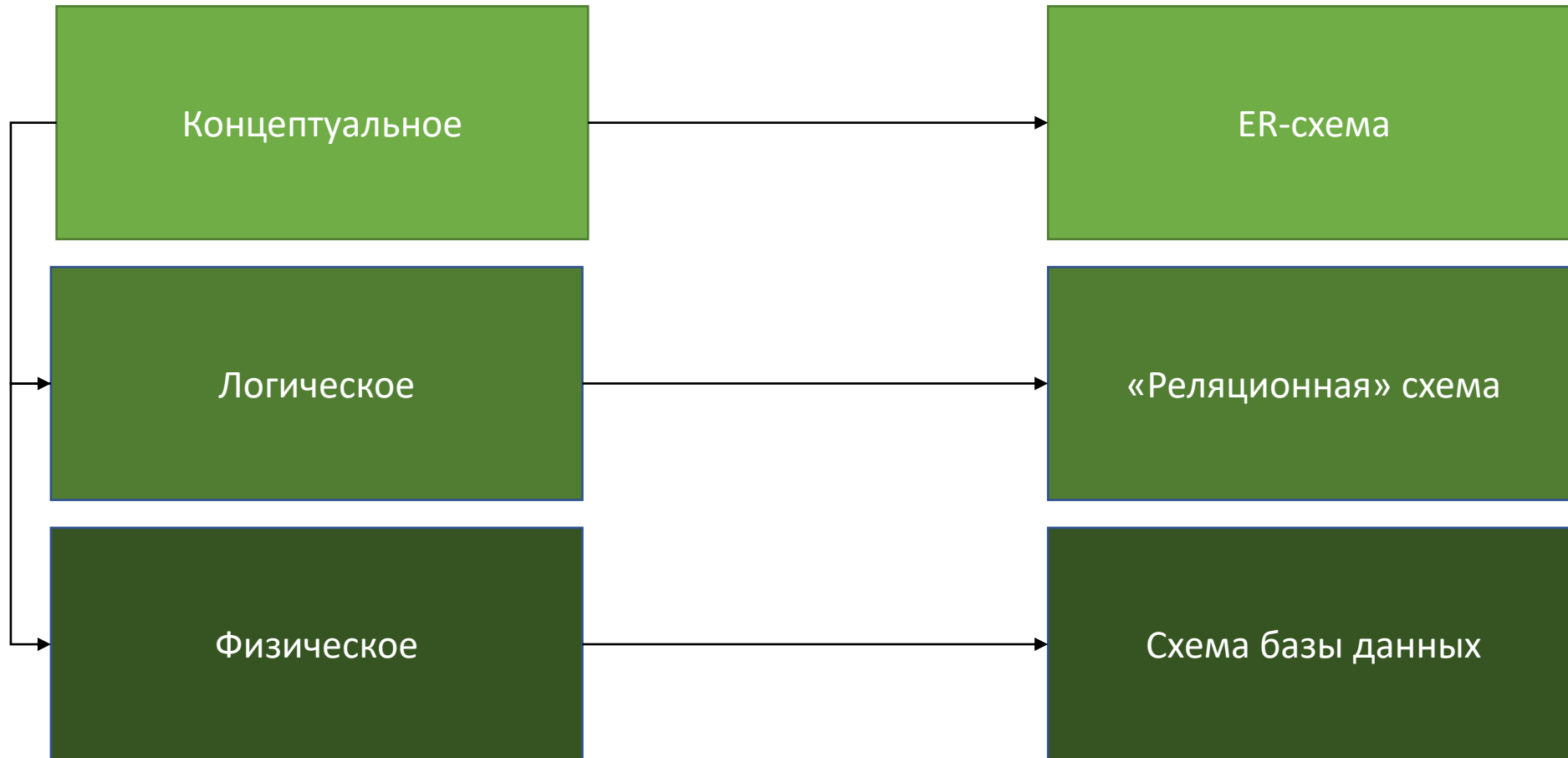
МФТИ, 2024

Игорь Шевченко

[@igorshvch](https://t.me/igorshvch)

I. Проектирование

Этапы проектирования



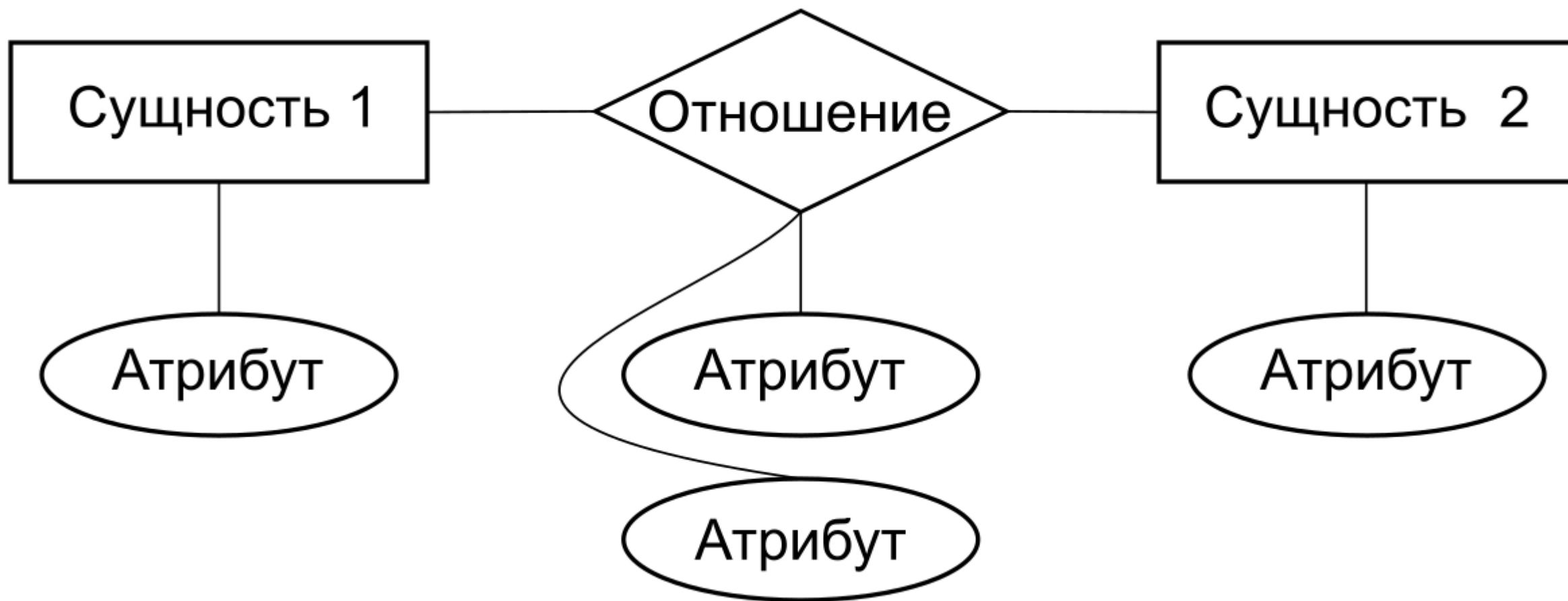
Этапы проектирования: характеристики

- Концептуальное:
 - Выявление наиболее общих свойств предметной области – сущностей и отношений между ними;
- Логическое:
 - Усовершенствование концептуальной схемы
 - Формирование отношений и атрибутов, первичных ключей, установление функциональных зависимостей
- Физическое:
 - Усовершенствование логической схемы
 - Формирование схемы базы данных применительно к конкретной СУБД
 - Определение типов данных, ограничений, формирование технических таблиц (многие-ко-многим, один-к-одному и т.п.)

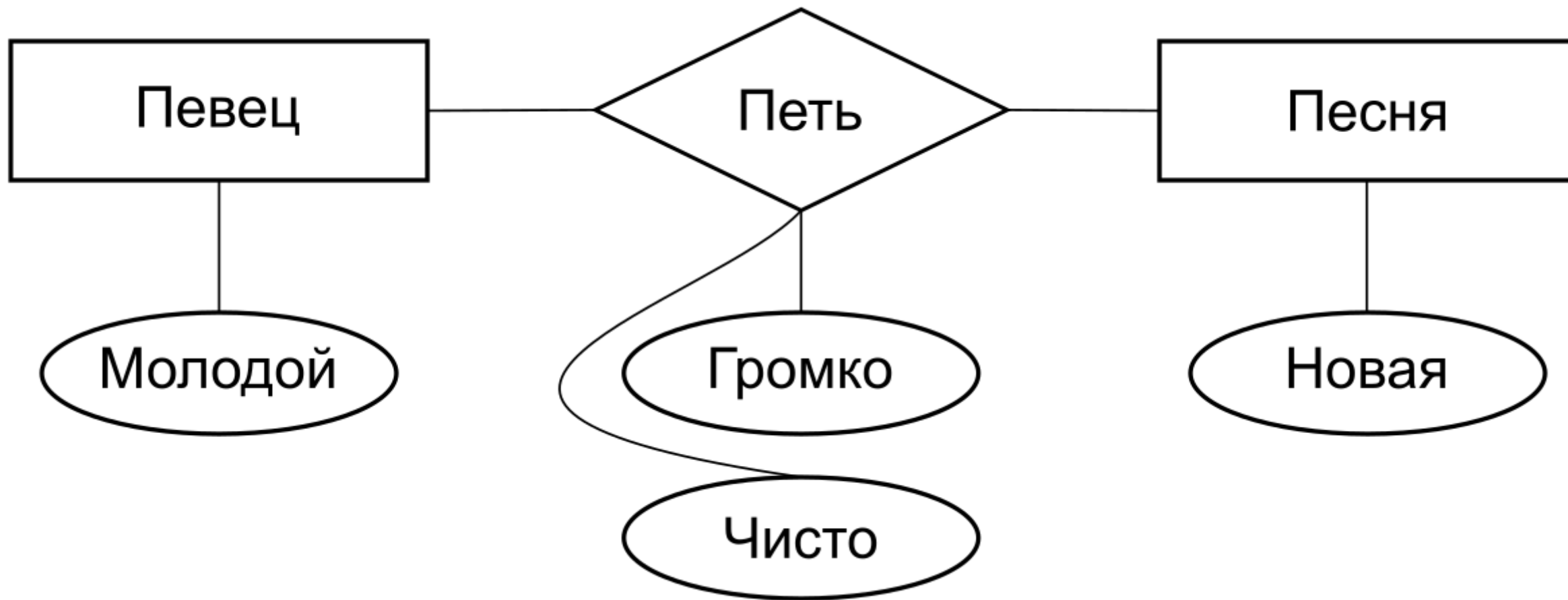
Концептуальное проER-схема: одна схема, две нотации

- FYI - Есть две нотации ER-схемы (ER-диаграммы):
 - Нотация Питера Чена:
 - Прямоугольники – сущности (выражаются существительными)
 - Ромбы – отношения (глаголы)
 - Овалы – атрибуты сущностей, свойства отношений (прилагательные и наречия)
 - Нотация «воронья лапка»:
 - Прямоугольники – сущности
 - Различные дуги между ними – отношения. Графически дуга выражает кардинальность отношения

Пример ER-диаграммы в нотации П. Чена



Пример ER-диаграммы в нотации П. Чена



Нотация «воронья лапка» (crow's foot)



Ноль



Много



Один



Ноль или один



Один и только один

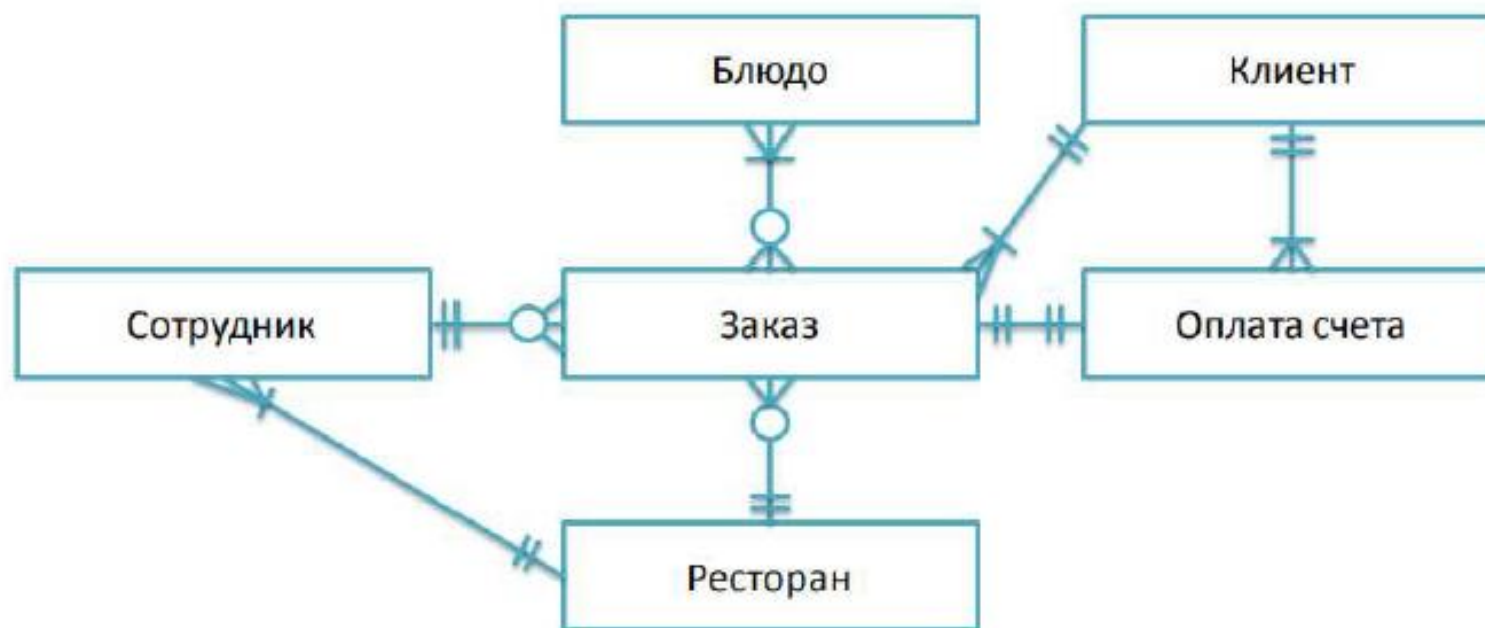


Ноль или много

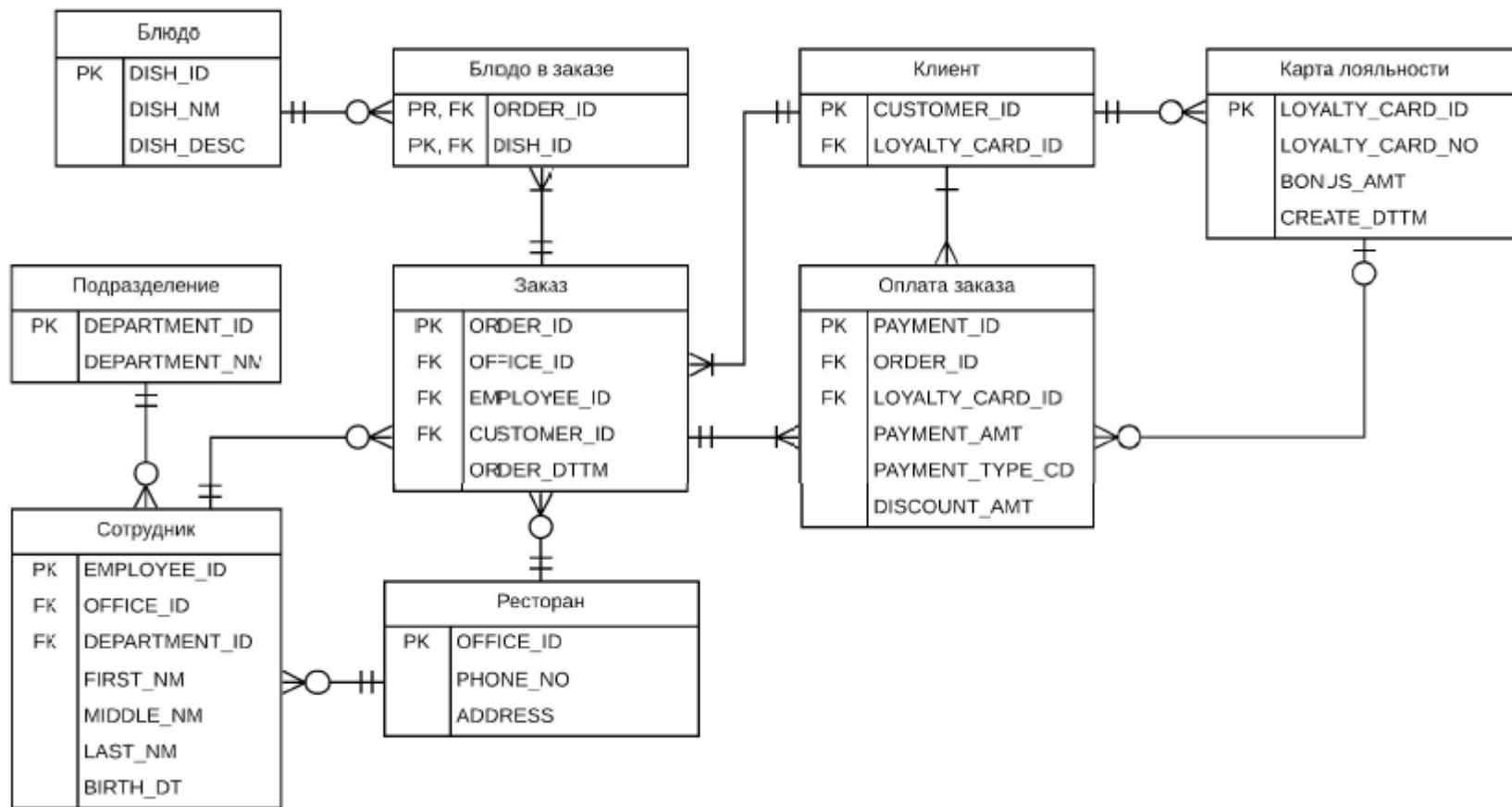


Один или много

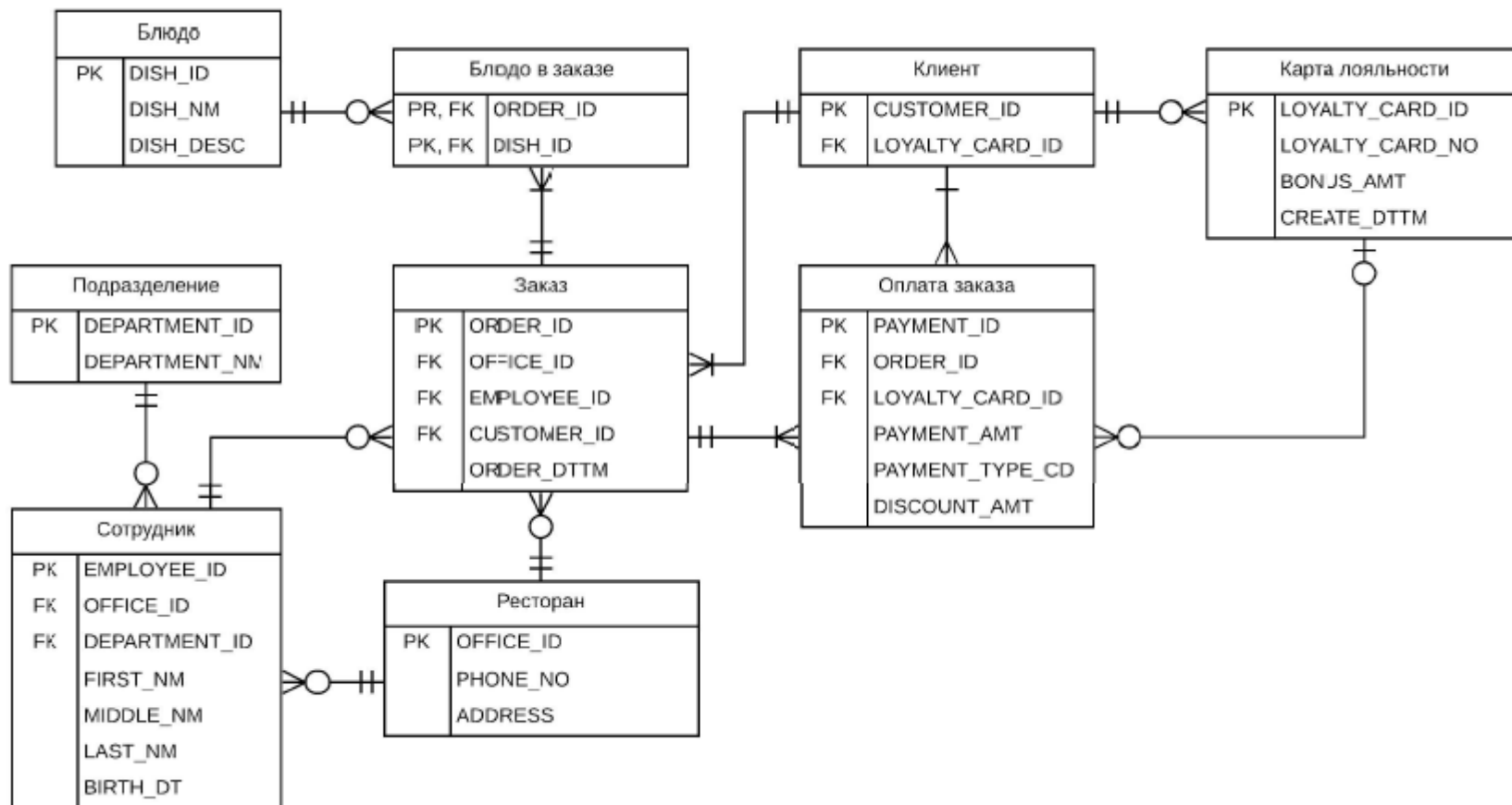
Пример ER в нотации «воронья лапка» (концептуальная модель ресторана)



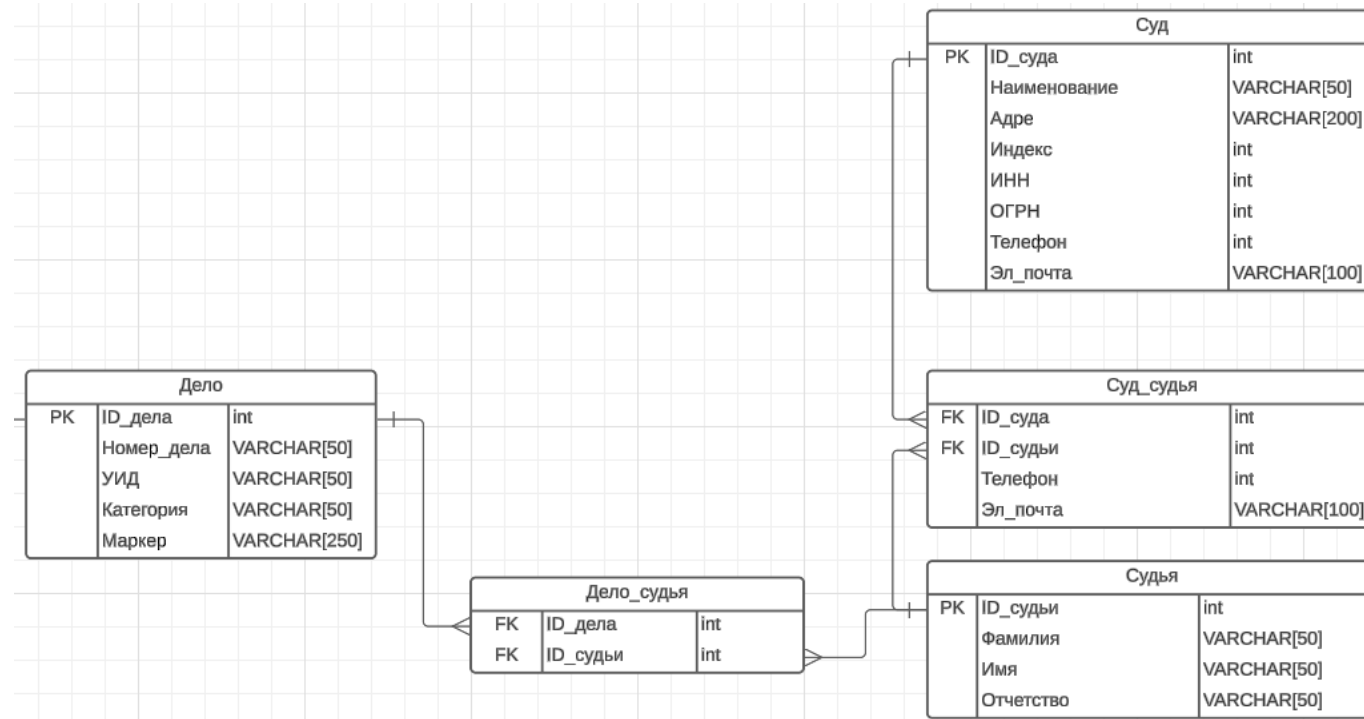
Логическое проектирование: реляционная схема (логическая модель ресторана)



Логическое проектирование: реляционная схема (логическая модель ресторана)



Физическое проектирование: (фрагмент тасктрекера юрдепартамента)



Для чего необходимо проектировать БД?

- Проектирование, помимо очевидных целей определения хранимых данных, также позволяет достичь следующего:
 - Исключить избыточность данных
 - Обеспечить поддержку целостности, и, в частности, ссылочной целостности данных.
- Это важно для поддержания корректности данных при их хранении и обработки
- Если пренебречь качественным проектированием БД, то впоследствии можно столкнуться с различного рода **аномалиями** :
 - добавления,
 - модификации (редактирования),
 - удаления данных
- Для качественного проектирования БД необходимо:
 - Проводить нормализацию данных
 - Использовать средства поддержания (ссылочной) целостности (первичные и внешние ключи, ограничения, проверки)

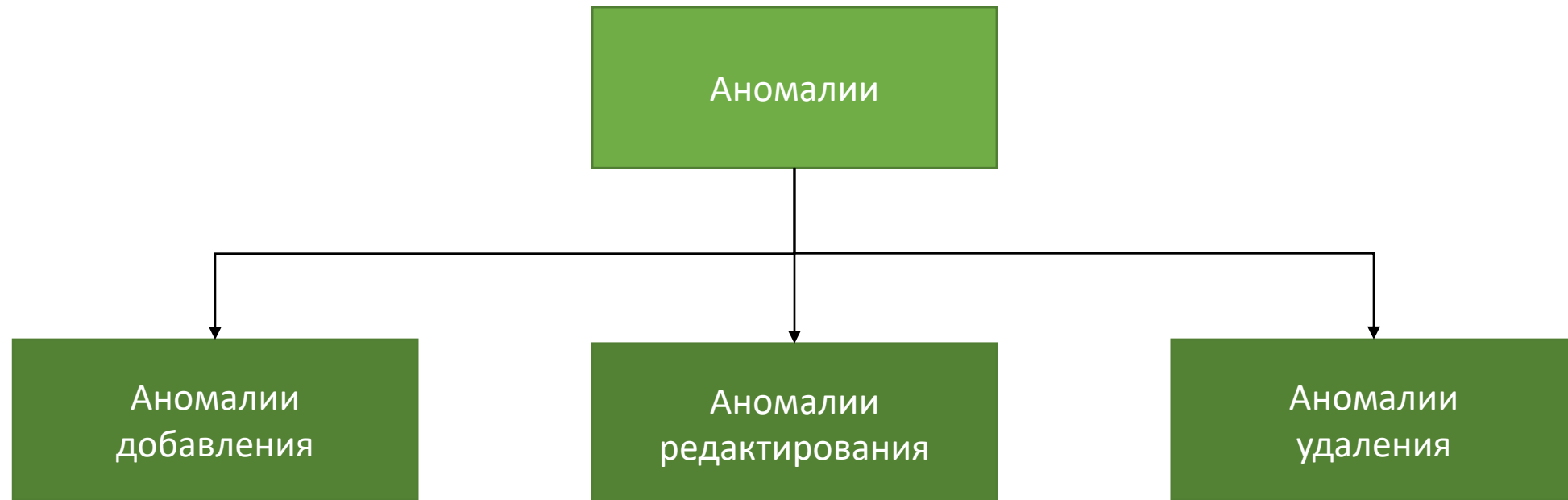
Необходимые в дальнейшем понятия

- **Целостность (сущностей)** – любое отношение должно обладать потенциальным ключом. Поддержание целостности сущностей средствами СУБД: (1) при добавлении записей в таблицу проверяется уникальность их первичных ключей; (2) не допускается изменение значений атрибутов, входящих в первичный ключ
- **Ссылочная целостность** – это необходимое качество реляционной базы данных, заключающееся в отсутствии в любом её отношении внешних ключей, ссылающихся на несуществующие кортежи (по Дж Дейту: база данных не должна содержать каких-либо несогласованных значений внешнего ключа). База данных обладает свойством ссылочной целостности, когда для любой пары связанных внешним ключом отношений в ней условие ссылочной целостности выполняется
- **Избыточность данных** – дублирование данных в базе
- **Аномалия** – ситуация в таблице БД, которая приводит к противоречию в БД либо существенно усложняет обработку БД. Причиной аномалии является (излишнее) дублирование данных в таблице

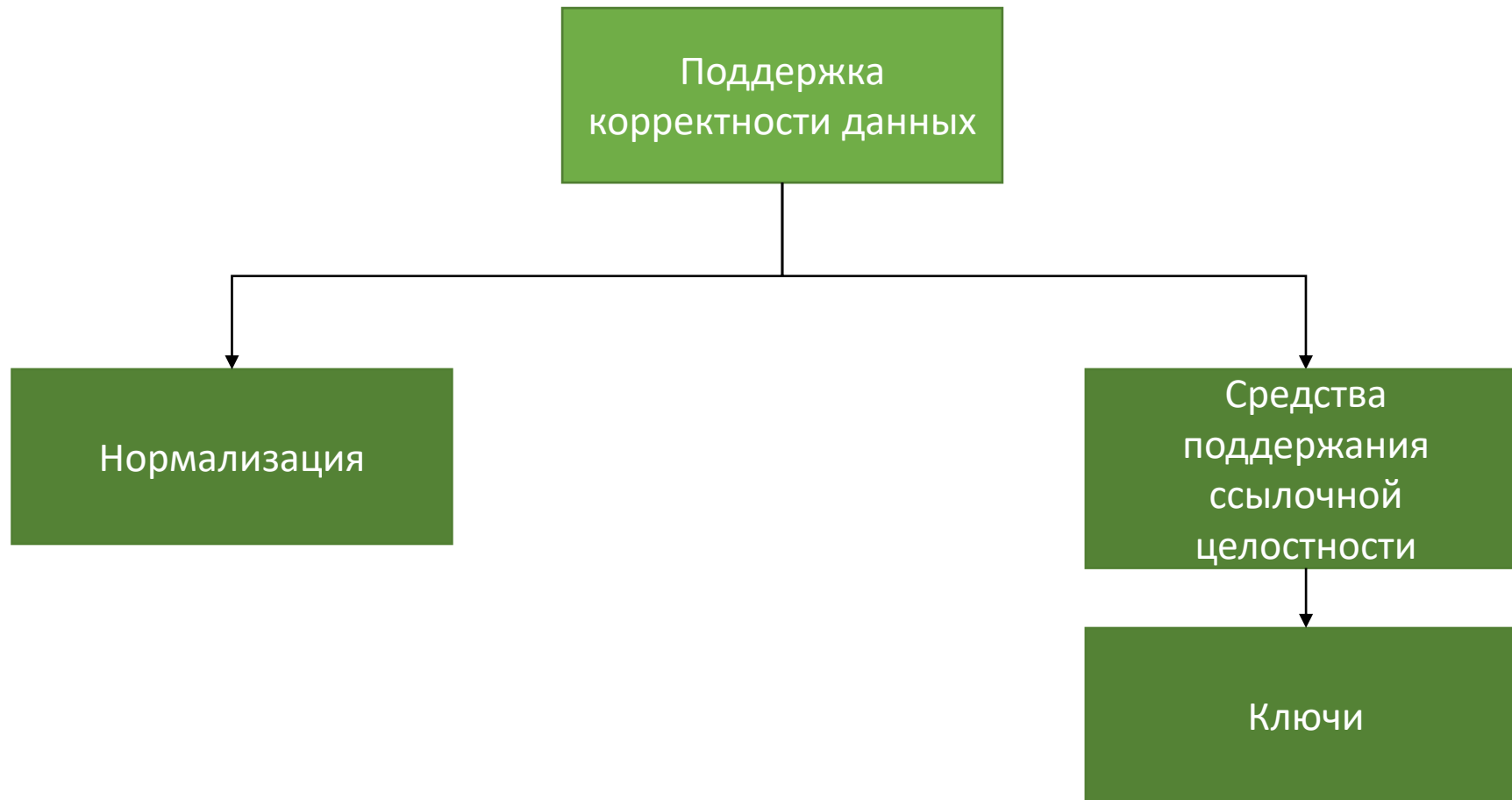
Цели проектирования



Последствия некачественного проектирования



Обеспечение поддержания корректности данных при проектировании



II. Аномалии обновления

Аномалия добавления

- Возникают, когда информацию в таблицу нельзя поместить, пока она не полная, либо вставка записи требует дополнительного просмотра таблицы

Номер поставки (РК)	Название товара (РК)	Цена товара	Количество	Дата поставки	Название поставщика	Адрес поставщика
1	Карандаш	15	10000	12.10.2017	Поставщик_1	Адрес_1
2	Клей	30	1500	03.03.2018	Поставщик_1	Адрес_1
2	Тетрадь	5	10000	03.03.2018	Поставщик_2	Адрес_2
3	Ручка	5	13000	05.03.2018	Поставщик_1	Адрес_1
3	Блокнот	50	20000	05.03.2018	Поставщик_1	Адрес_1
3	Альбом	100	25000	05.03.2018	Поставщик_2	Адрес_2

Аномалия добавления

- Не можем добавить новую запись, если неизвестны значения первичных ключей: Заключили контракт с поставщиком 3: не можем добавить информацию о нем в таблицу, т.к. еще не было поставок

Номер поставки (РК)	Название товара (РК)	Цена товара	Количество	Дата поставки	Название поставщика	Адрес поставщика
1	Карандаш	15	10000	12.10.2017	Поставщик_1	Адрес_1
2	Клей	30	1500	03.03.2018	Поставщик_1	Адрес_1
2	Тетрадь	5	10000	03.03.2018	Поставщик_2	Адрес_2
3	Ручка	5	13000	05.03.2018	Поставщик_1	Адрес_1
3	Блокнот	50	20000	05.03.2018	Поставщик_1	Адрес_1
3	Альбом	100	25000	05.03.2018	Поставщик_2	Адрес_2

Аномалия редактирования (модификации)

- Проявляются в том, что изменение одних данных может повлечь просмотр всей таблицы и соответствующее изменение некоторых записей таблицы

Номер поставки (РК)	Название товара (РК)	Цена товара	Количество	Дата поставки	Название поставщика	Адрес поставщика
1	Карандаш	15	10000	12.10.2017	Поставщик_1	Адрес_1
2	Клей	30	1500	03.03.2018	Поставщик_1	Адрес_1
2	Тетрадь	5	10000	03.03.2018	Поставщик_2	Адрес_2
3	Ручка	5	13000	05.03.2018	Поставщик_1	Адрес_1
3	Блокнот	50	20000	05.03.2018	Поставщик_1	Адрес_1
3	Альбом	100	25000	05.03.2018	Поставщик_2	Адрес_2

Аномалия редактирования (модификации)

- Изменение данных одной записи влечет за собой необходимость изменения аналогичных данных еще некоторых записей. Хотим изменить адрес поставщика 1: придется менять адрес во всех строках

Номер поставки (РК)	Название товара (РК)	Цена товара	Количество	Дата поставки	Название поставщика	Адрес поставщика
1	Карандаш	15	10000	12.10.2017	Поставщик_1	Адрес_1
2	Клей	30	1500	03.03.2018	Поставщик_1	Адрес_1
2	Тетрадь	5	10000	03.03.2018	Поставщик_2	Адрес_2
3	Ручка	5	13000	05.03.2018	Поставщик_1	Адрес_1
3	Блокнот	50	20000	05.03.2018	Поставщик_1	Адрес_1
3	Альбом	100	25000	05.03.2018	Поставщик_2	Адрес_2

Аномалия удаления

- При удалении какого либо кортежа из таблицы может пропасть информация, которая не связана на прямую с удаляемой записью

Номер поставки (РК)	Название товара (РК)	Цена товара	Количество	Дата поставки	Название поставщика	Адрес поставщика
1	Карандаш	15	10000	12.10.2017	Поставщик_1	Адрес_1
2	Клей	30	1500	03.03.2018	Поставщик_1	Адрес_1
2	Тетрадь	5	10000	03.03.2018	Поставщик_2	Адрес_2
3	Ручка	5	13000	05.03.2018	Поставщик_1	Адрес_1
3	Блокнот	50	20000	05.03.2018	Поставщик_1	Адрес_1
3	Альбом	100	25000	05.03.2018	Поставщик_2	Адрес_2

Аномалия удаления

- Удаление определенных записей несет потерю информации, которую удалять не хотели. Хотим удалить записи о поставках от поставщика 2: теряем всю информации о поставщике 2, включая его адрес

Номер поставки (РК)	Название товара (РК)	Цена товара	Количество	Дата поставки	Название поставщика	Адрес поставщика
1	Карандаш	15	10000	12.10.2017	Поставщик_1	Адрес_1
2	Клей	30	1500	03.03.2018	Поставщик_1	Адрес_1
2	Тетрадь	5	10000	03.03.2018	Поставщик_2	Адрес_2
3	Ручка	5	13000	05.03.2018	Поставщик_1	Адрес_1
3	Блокнот	50	20000	05.03.2018	Поставщик_1	Адрес_1
3	Альбом	100	25000	05.03.2018	Поставщик_2	Адрес_2

III. Ссылочная целостность, ключи, функциональная зависимость

Средства поддержания ссылочной целостности - обзор

- Некоторые команды и предложения SQL, которые позволяют поддерживать (ссылочную) целостность, мы уже рассмотрели при знакомстве с DDL – это ограничения, задаваемые при создании таблиц командой CREATE TABLE:
 - NOT NULL
 - UNIQUE
 - PRIMARY KEY / FOREIGN KEY
 - CHECK
 - DEFAULT

Средства поддержания ссылочной целостности - обзор

- Кроме перечисленных ограничений можно явно задать определенное поведение СУБД при удалении некоторой строки или изменении некоторого значения во внешней таблице – то есть в таблице, на которую ссылается атрибут, являющийся внешним ключом. (Если при изменении данных строка изменена, но это изменение не затронуло целевые столбцы, никакое действие не производится)

Средства поддержания ссылочной целостности - обзор

- NO ACTION
 - Выдать ошибку, показывающую, что при удалении или изменении записи произойдёт нарушение ограничения внешнего ключа
- RESTRICT
 - Выдать ошибку, показывающую, что при удалении или изменении записи произойдёт нарушение ограничения внешнего ключа
- CASCADE
 - Удалить все строки, ссылающиеся на удаляемую запись, либо поменять значения в ссылающихся столбцах на новые значения во внешних столбцах, в соответствии с операцией
- SET NULL [(имя_столбца [, ...])]
 - Установить во всех ссылающихся столбцах или в указанном подмножестве ссылающихся столбцов значения null. Подмножество столбцов можно указать только для действий ON DELETE
- SET DEFAULT [(имя_столбца [, ...])]
 - Установить во всех ссылающихся столбцах или в указанном подмножестве ссылающихся столбцов значения по умолчанию. Подмножество столбцов можно указать только для действий ON DELETE. (Если значения по умолчанию отличны от NULL, во внешней таблице должна быть строка, соответствующая набору этих значений; в противном случае операция завершится ошибкой.)

Средства поддержания ссылочной целостности – пример синтаксиса

```
CREATE TABLE orders (  
    order_id SERIAL PRIMARY KEY,  
    customer_id INT REFERENCES customers(customer_id) ON DELETE  
    CASCADE,  
    order_date DATE,  
    total_amount DECIMAL  
);
```

Средства поддержания ссылочной целостности - ключи

- По Дж. Дейту:
 - Определение потенциального ключа по сути является просто сокращением для некоторого ограничения на отношении
 - Потенциальные ключи имеют важное значение с точки зрения практики: в частности, в реляционной модели они предоставляют основной механизм адресации на уровне кортежей.
 - Это означает, что единственный гарантируемый системой способ точного определения некоторого конкретного кортежа состоит в использовании определенного значения потенциального ключа.

Ключи:
потенциальный,
первичный,
внешний

- **Потенциальный ключ** - в реляционной модели данных это подмножество атрибутов отношения, удовлетворяющее требованиям:
 - уникальности - нет и не может быть двух кортежей данного отношения, в которых значения этого подмножества атрибутов совпадают (равны)
 - несократимости (минимальности) - в составе потенциального ключа отсутствует меньшее подмножество атрибутов, удовлетворяющее условию уникальности. Иными словами, если из потенциального ключа убрать любой атрибут, он утратит свойство уникальности
- Потенциальный ключ существует всегда, даже если он включает все атрибуты отношения (следует из свойств отношений)
- Потенциальных ключей может быть несколько

Ключи:
потенциальный,
первичный,
внешний

- **Потенциальный ключ** может быть:
 - Простым – если состоит из одного атрибута
 - Составным – если состоит из двух и более атрибутов отношения

Ключи:
потенциальный,
первичный,
внешний

- Первичный ключ (PRIMARY KEY) – это такой потенциальный ключ отношения, который выбран в качестве «основного»
 - Любой потенциальный ключ пригоден на роль первичного

Ключи: потенциальный, первичный, внешний

- Для отражения функциональных зависимостей между кортежами разных отношений используется дублирование первичного ключа одного отношения (родительского) в другое (дочернее). Атрибуты, представляющие собой копии ключей родительских отношений, называются внешними ключами.
- Внешний ключ в отношении R2 – это непустое подмножество FK множества атрибутов этого отношения, такое, что:
 - 1) Существует отношение R1 с потенциальным ключом PK;
 - 2) Каждое значение внешнего ключа FK в текущем значении отношения R2 обязательно совпадает со значением ключа PK некоторого кортежа в текущем значении отношения R1.
- Отношения R1 и R2 необязательно различны.

Свойство ключей \rightarrow функциональная зависимость

- Ключи позволяют формировать функциональные зависимости между отношениями
- Определение функциональной зависимости (по Дж. Дейту):
Пусть r является отношением, а X и Y — произвольными подмножествами множества атрибутов отношения r . **Тогда Y функционально зависит от X** , что в символическом виде записывается как
 $X \rightarrow Y$
(читается либо как " X функционально определяет Y ", либо как " X стрелка Y ") тогда и только тогда, когда каждое значение множества X отношения r связано точно с одним значением множества Y отношения r . Иначе говоря, если два кортежа отношения r совпадают по значению X , они совпадают и по значению Y (\Leftrightarrow не существует двух различных кортежей, которые в атрибуте X имеют совпадающие значения, а в атрибуте Y имеют разные значения).
- Левая часть записи функциональной зависимости называется **детерминантом**, правая — **зависимой частью**.
- **Функциональная зависимость — это связь типа "многие к одному" между двумя множествами атрибутов заданной переменной отношения**

Примеры функциональной зависимости внутри одной таблицы

Город	Поставщик	Продукт	Количество
Москва	ООО "Ромашка"	Конфеты	100
Санкт-Петербург	ООО "Василек"	Шоколад	300
Казань	ООО "Ландыш"	Мороженое	400
Москва	ООО "Ромашка"	Конфеты	200
Санкт-Петербург	ООО "Подорожник"	Шоколад	300

Примеры функциональной зависимости внутри одной таблицы

Поставщик	Город	Продукт	Количество
ООО "Ромашка"	Москва	Конфеты	1000
ООО "Василек"	Санкт-Петербург	Шоколад	3000
ООО "Ландыш"	Казань	Мороженое	4000
ООО "Ромашка"	Москва	Конфеты	2000
ООО "Подорожни"	Краснодар	Шоколад	3000
ООО "Подорожни"	Краснодар	Шоколад	3000

$X \rightarrow Y$

$\{\text{Город}\} \rightarrow \{\text{Продукт}\}$

$\{\text{Поставщик}\} \rightarrow \{\text{Город}\}$

$\{\text{Поставщик}\} \rightarrow \{\text{Продукт}\}$

$\{\text{Количество}\} \rightarrow \{\text{Продукт}\}$

$\{\text{Количество}\} \rightarrow \{\text{Город}\}$

$\{\text{Город, Поставщик}\} \rightarrow \{\text{Продукт}\}$

$\{\text{Поставщик, Продукт}\} \rightarrow \{\text{Город}\}$

Связь между ключами и функциональными зависимостями

- Рассмотрим измененную версию предыдущей таблицы:

Поставщик	Город	Продукт
ООО "Ромашка"	Москва	Конфеты
ООО "Василек"	Санкт-Петербург	Шоколад
ООО "Ландыш"	Казань	Мороженое
ООО "Ромашка"	Москва	Конфеты
ООО "Подорожни"	Краснодар	Шоколад
ООО "Подорожни"	Краснодар	Шоколад

Связь между ключами и функциональными зависимостями

Поставщик	Город	Продукт
ООО "Ромашка"	Москва	Конфеты
ООО "Василек"	Санкт-Петербург	Шоколад
ООО "Ландыш"	Казань	Мороженое
ООО "Ромашка"	Москва	Конфеты
ООО "Подорожни"	Краснодар	Шоколад
ООО "Подорожни"	Краснодар	Шоколад

$X \rightarrow Y$

$\{\text{Поставщик}\} \rightarrow \{\text{Город}\}$

$\{\text{Город}\} \rightarrow \{\text{Продукт}\}$

Связь между ключами и функциональными зависимостями

Поставщик	Город	Продукт
ООО "Ромашка"	Москва	Конфеты
ООО "Василек"	Санкт-Петербург	Шоколад
ООО "Ландыш"	Казань	Мороженое
ООО "Ромашка"	Москва	Конфеты
ООО "Подорожни"	Краснодар	Шоколад
ООО "Подорожни"	Краснодар	Шоколад

Очевидно, что данные в приведенной таблице избыточны – информация о поставщиках, городах и продуктах дублируется в различных строках

Связь между ключами и функциональными зависимостями

Поставщик	Город	Продукт
ООО "Ромашка"	Москва	Конфеты
ООО "Василек"	Санкт-Петербург	Шоколад
ООО "Ландыш"	Казань	Мороженое
ООО "Ромашка"	Москва	Конфеты
ООО "Подорожни"	Краснодар	Шоколад
ООО "Подорожни"	Краснодар	Шоколад

Вопрос: каким образом мы можем избавиться от дублирования?

Ответ: **теорема Хита (Heath)**

Связь между ключами и функциональными зависимостями

- **Теорема Хита**. Пусть $R\{A, B, C\}$ является отношением, где A, B и C — множества атрибутов. Если R удовлетворяет функциональной зависимости $A \rightarrow B$, то R равна соединению ее проекций по атрибутам $\{A, B\}$ и $\{A, C\}$ ($R = \{A, B\} \bowtie \{A, C\}$).
- По отношению к таблице выше примем, что $A = \{\text{Поставщик}\}$, $B = \{\text{Город}\}$, $C = \{\text{Продукт}\}$.
- Тогда по теореме Хита один из вариантов декомпозиции исходного отношения следующий: $\{\text{Поставщик}, \text{Город}\}$ и $\{\text{Поставщик}, \text{Продукт}\}$.

Связь между ключами и функциональными зависимостями

Поставщик	Город	Продукт
ООО "Ромашка"	Москва	Конфеты
ООО "Василек"	Санкт-Петербург	Шоколад
ООО "Ландыш"	Казань	Мороженое
ООО "Ромашка"	Москва	Конфеты
ООО "Подорожни"	Краснодар	Шоколад
ООО "Подорожни"	Краснодар	Шоколад

Поставщик	Город
ООО "Ромашка"	Москва
ООО "Василек"	Санкт-Петербург
ООО "Ландыш"	Казань
ООО "Подорожни"	Краснодар

Поставщик	Продукт
ООО "Ромашка"	Конфеты
ООО "Василек"	Шоколад
ООО "Ландыш"	Мороженое
ООО "Подорожни"	Шоколад

Связь между ключами и функциональными зависимостями

- В рассматриваемом случае атрибут {Поставщик} может быть выбран в качестве первичного ключа в отношении {Поставщик, Город}, а в отношении {Поставщик, Продукт} этот атрибут может иметь ограничение внешнего ключа.
- Обобщив понятие функциональной зависимости, можно (не строго) сказать, что первичный ключ определяет функциональную зависимость содержащей его таблицы по отношению к таблице, содержащей внешний ключ. То есть при равенстве значений «атрибутов – внешних ключей» они всегда равны значению «атрибута – первичного ключа».



IV. Нормализация

Что такое нормализация и для чего она нужна

- 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
- Приведение БД к нормальной форме – нормализация
- Каждая следующая форма включает в себя ограничения предыдущих

Что такое нормализация и для чего она нужна

- Нормализация предназначена:
 - Минимизация логической избыточности
 - Уменьшение потенциальной противоречивости
- Нормализация не предназначена:
 - Уменьшение / увеличение производительности БД
 - Уменьшение / увеличение физического объема БД

Что такое нормализация и для чего она нужна

Нормализация нужна:

- Исключение некоторых типов избыточности
- Устранение некоторых аномалий обновления
- Для разработки проекта БД, который является:
 - Качественным представлением реального мира
 - Интуитивно понятен
 - Легко расширяем в дальнейшем
- Упрощения процедуры применения необходимых ограничений целостности

Что такое нормализация и для чего она нужна

- Нормализация БД производится за счет декомпозиции отношения (см. выше теорему Хита).
- Декомпозиция – разложение исходной переменной отношения на несколько эквивалентных. Декомпозиция обратна соединению
- **Декомпозиция называется декомпозицией без потерь или правильной , если она обратима**

Перечень нормальных форм

- Первая нормальная форма (1NF)
- Вторая нормальная форма (2NF)
- Третья нормальная форма (3 NF)
- Нормальная форма Бойса-Кодда (BCNF)
- Четвертая нормальная форма (4 NF)
- Пятая нормальная форма / Нормальная форма проекции соединения (5 NF / PJNF)
- Доменно-ключевая нормальная форма (DKNF)
- Шестая нормальная форма (6NF)

1NF

- Переменная отношения находится в **первой нормальной форме** тогда и только тогда, когда значения всех атрибутов отношения атомарны. То есть отношение находится в 1 NF , если все его атрибуты являются простыми. Все используемые домены содержат только скалярные значения.
- Неформальное определение:
 - Все строки должны быть различными
 - Все элементы внутри ячеек должны быть атомарными, т.е. не должны быть списками. Другими словами, элемент является атомарным, если его нельзя разделить на части, которые могут использовать в таблице независимо друг от друга
- Любое реляционное отношение находится в 1НФ на основании свойств реляционного отношения.

1NF

Город	Улица
Москва	Моховая, Воздвиженка, Неглинная
Санкт-Петербург	Невский проспект, Миллионная, Рубинштейна
Казань	Кремлевская, Баумана, Пушкина

1NF

Город	Улица
Москва	Моховая, Воздвиженка, Неглинная
Санкт-Петербург	Невский проспект, Миллионная, Рубинштейна
Казань	Кремлевская, Баумана, Пушкина



Город	Улица
Москва	Моховая
Москва	Воздвиженка
Москва	Неглинная
Санкт-Петербург	Невский проспект
Санкт-Петербург	Миллионная
Санкт-Петербург	Рубинштейна
Казань	Кремлевская
Казань	Баумана
Казань	Пушкина

1NF

Город	Улица
Москва	Моховая
Москва	Воздвиженка
Москва	Неглинная
Санкт-Петербург	Невский проспект
Санкт-Петербург	Миллионная
Санкт-Петербург	Рубинштейна
Казань	Кремлевская
Казань	Баумана
Казань	Пушкина

*Обратите внимание: нормализация проводится не с точки зрения типа данных (строка = массив символов), а с точки зрения семантики данных. В данном случае «Невский проспект» – атомарное значение, как и любая строка в ячейке приведенной таблице. Отдельный символ (буква, пробел), а в указанном случае и отдельное слов самостоятельного значения в представленной модели не имеют

2NF

- Переменная отношения находится **во второй нормальной форме** тогда и только тогда, когда она находится в первой нормальной форме, и каждый неключевой атрибут минимально функционально зависит от потенциального ключа
- Повторим: функциональная зависимость между множествами атрибутов X и Y означает, что для любого допустимого набора кортежей в данном отношении верно следующее: если два кортежа совпадают по значению X , то они совпадают по значению Y
- **Минимальная функциональная зависимость** означает, что в составе первичного ключа отсутствует меньшее подмножество атрибутов, от которого можно также вывести данную функциональную зависимость

2NF

- Неформальное определение для таблицы с единственным ключом:
 - Таблица должна находиться в первой нормальной форме
 - Любое ее поле, не входящее в состав первичного ключа, функционально зависит только от всех атрибутов ключа и не зависит от его отдельных атрибутов
- Если таблица с единственным ключом приведена к первой нормальной форме и у нее установлен уникальный id для каждой строки, то она находится во второй нормальной форме.

2NF

Название группы	Название CD-диска	Название песни	Автор слов	Композитор
Scorpions	World Wide Live	Countdown	Klaus Meine	Matthias Jabs
Scorpions	World Wide Live	Coming Home	Rudolf Schenker	Klaus Meine
Scorpions	World Wide Live	Blackout	Rudolf Schenker	Klaus Meine
Scorpions	Blackout	Blackout	Rudolf Schenker	Klaus Meine
The Big City	Blackout	Blackout	Rudolf Schenker	Klaus Meine

2NF

Название группы	Название CD-диска	Название песни	Автор слов	Композитор
Scorpions	World Wide Live	Countdown	Klaus Meine	Matthias Jabs
Scorpions	World Wide Live	Coming Home	Rudolf Schenker	Klaus Meine
Scorpions	World Wide Live	Blackout	Rudolf Schenker	Klaus Meine
Scorpions	Blackout	Blackout	Rudolf Schenker	Klaus Meine
The Big City	Blackout	Blackout	Rudolf Schenker	Klaus Meine

Данная таблица находится в 1NF, но не во 2NF, т.к. автор слов и композитор зависят только от полей «Название группы» и «Название песни», т.е. от того, что песня включена на другой CD-диск, значения этих полей никак не изменятся

2NF

Название группы	Название CD-диска	Название песни	Автор слов	Композитор
Scorpions	World Wide Live	Countdown	Klaus Meine	Matthias Jabs
Scorpions	World Wide Live	Coming Home	Rudolf Schenker	Klaus Meine
Scorpions	World Wide Live	Blackout	Rudolf Schenker	Klaus Meine
Scorpions	Blackout	Blackout	Rudolf Schenker	Klaus Meine
The Big City	Blackout	Blackout	Rudolf Schenker	Klaus Meine

Данная таблица находится в 1NF, но не во 2NF, т.к. автор слов и композитор зависят только от полей «Название группы» и «Название песни», т.е. от того, что песня включена на другой CD-диск, значения этих полей никак не изменятся

2NF

Название группы	Название CD-диска	Название песни
Scorpions	World Wide Live	Countdown
Scorpions	World Wide Live	Coming Home
Scorpions	World Wide Live	Blackout
Scorpions	Blackout	Blackout
The Big City	Blackout	Blackout

Название группы	Название песни	Автор слов	Композитор
Scorpions	Countdown	Klaus Meine	Matthias Jabs
Scorpions	Coming Home	Rudolf Schenker	Klaus Meine
Scorpions	Blackout	Rudolf Schenker	Klaus Meine
Scorpions	Blackout	Rudolf Schenker	Klaus Meine
The Big City	Blackout	Rudolf Schenker	Klaus Meine

2NF – объяснение применительно к теореме Хита

- Нам нужны три группы атрибутов и ФЗ между двумя из них:
- Если $A \rightarrow B$, то $R = \{AB\} \bowtie \{AC\}$, следовательно если
 - ФЗ: {Название группы, Название песни} \rightarrow {Автор слов, Композитор}
- То:
 - A – {Название группы, Название песни}
 - B – {Автор слов, Композитор}
 - C – {Название CD-диска}
- Тогда по теореме Хита:
 - Если $A \rightarrow B$, то $R = \{AB\} \bowtie \{AC\}$ или
 $R = \{\text{Название группы, Название песни, Автор слов, Композитор}\} \bowtie$
 $\bowtie \{\text{Название группы, Название песни, Название CD-диска}\}$