ERM (Entity-Relation Model)

Анализ и проектирование структур данных с использованием CASE-средств

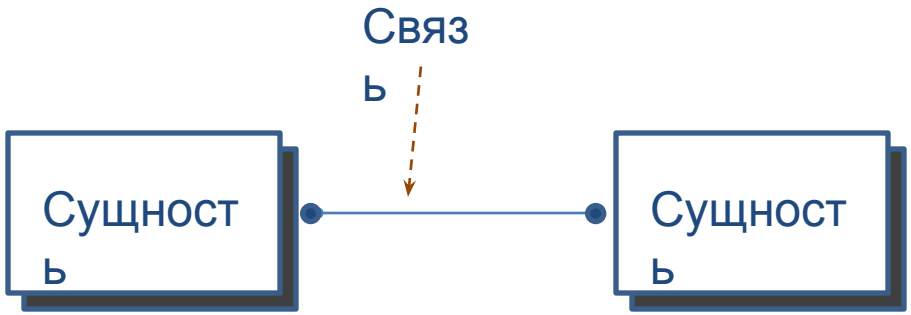
Стандарт IDEF1x. Общие сведения

- Разработан в 1983 году в рамках проекта военного ведомства США "Интегрированные системы информационной поддержки" (ICAM).
- Как методология семантического моделирования данных.
- Стал расширением методологии IDEF1.

IDEF1x. Базовые определения

- Логическая модель.
- Физическая модель.
- Сущность (Entity).
- Атрибут (Attribute).
- Связь (Realtionship).
- Ключ.

Связи



Правила именования связей (Verb Phrase)

- Выражает некоторое бизнес-правило, либо ограничение, например –

  МЕНЕДЖЕР *оформляет* ДОГОВОР,

  ПРЕДМЕТ *изучается* СТУДЕНТОМ.
- Связи подписываются для их идентификации, например,

  1..N (читается «один-ко-многим»),

  M..N (читается «многие-ко-многим»).
- Имя связи указывается в направлении от родительской к дочерней сущности.
- Для отношения «многие-ко-многим» используют имена в обоих направлениях.

Мощность связи (Cardinality)

- Служит для обозначения отношения числа экземпляров родительской сущности к числу экземпляров дочерней.
- *1..1* – один и только один. Символ предусмотрен.
- *1..0 или 1*. Символ *1*.
- *1..const*. Обозначается соответствующим

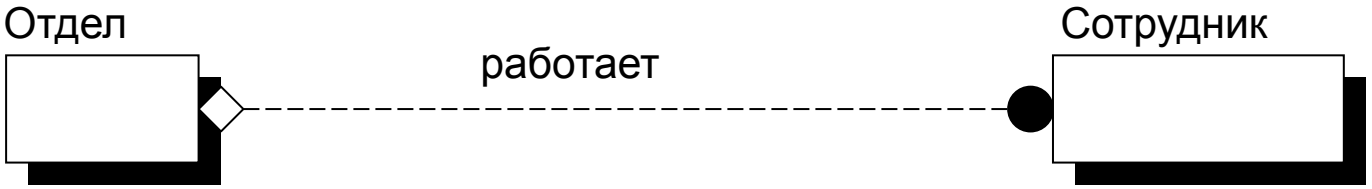
Связи

Идентифицирующая связь.

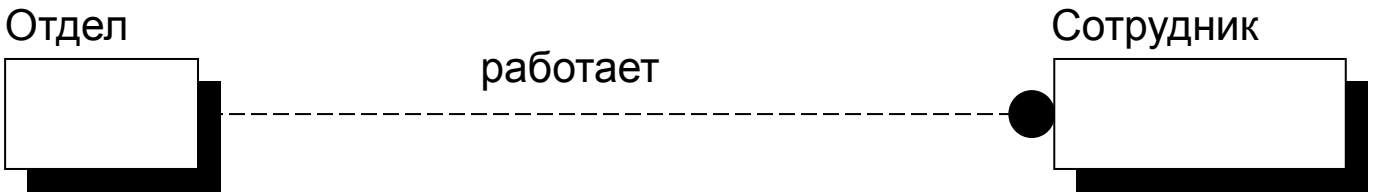
Связь между зависимой (дочерняя сторона связи) *и независимой* (родительская сторона связи) *сущностями*.

Неидентифицирующая связь.

Необязательная



Обязательная



Ключи

- Виды ключей.
  - **Потенциальным ключом** - атрибут или группа атрибутов, однозначно идентифицирующих экземпляр сущности.
  - В общем случае у сущности может быть несколько потенциальных ключей, хотя для целей однозначной идентификации достаточно одного.
  - **Сложный ключ (составной) –** ключ, содержащий более одного атрибута.
  - **Первичный ключ** - один из потенциальных ключей.
  - **Альтернативный ключ** - потенциальный ключ, не ставший первичным.
  - **Суррогатный ключ** - создаётся на основе искусственно созданного атрибута, не имеющего аналогов в предметной области, это – автоинкрементный столбец, уникальность значений которого поддерживается СУБД.

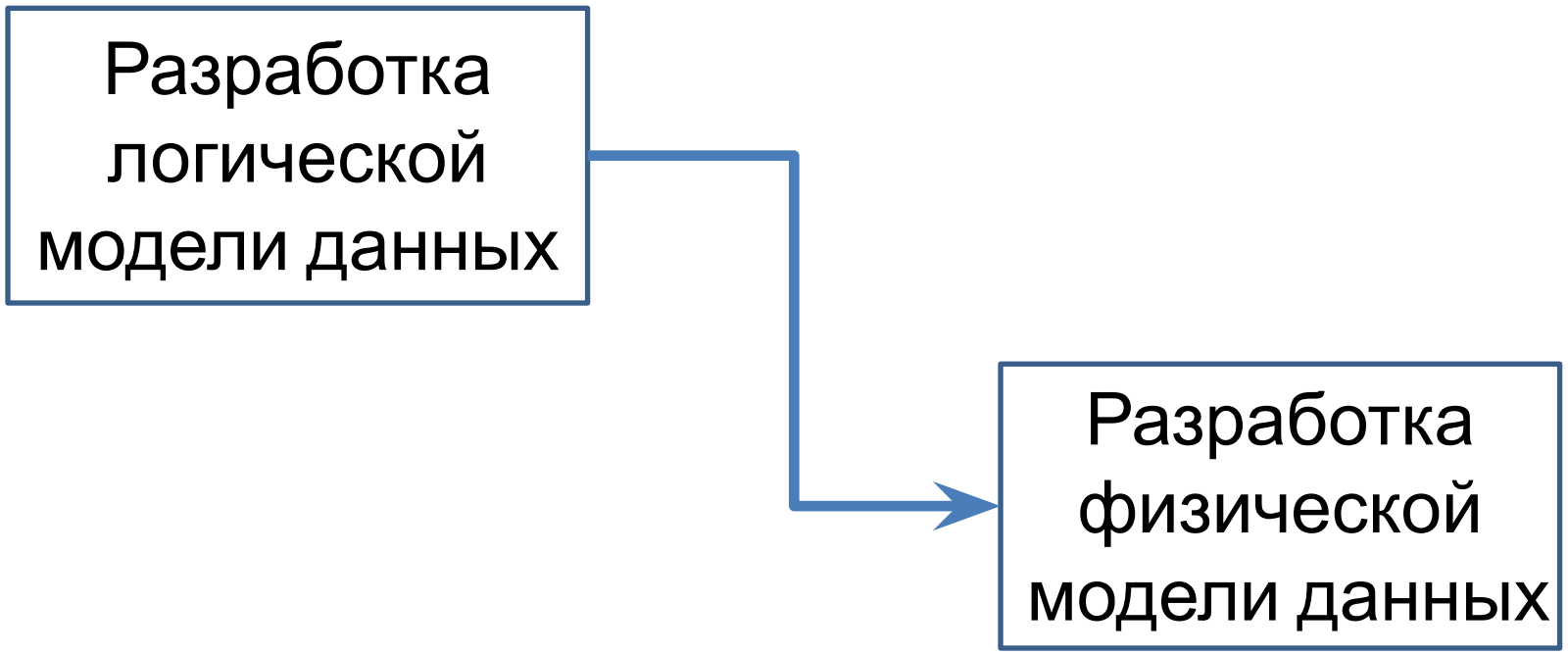
Ключи

- Требования к первичному ключу.
  - Уникальность — Два экземпляра сущности не должны иметь одинаковых значений атрибутов, образующих ключ
  - Компактность — Сложный ключ не должен содержать ни одного атрибута, удаление которого не привело бы к потере уникальности.
  - Простота — При выборе первичного ключа предпочтение отдаётся **более простым** ключам, т.е. ключам, содержащим меньшее количество атрибутов.
  - Полная определенность — Ни один из атрибутов, образующих ключ, не должен содержать значение с предопределённым значением NULL (пустой, неопределённый).
  - Постоянство во времени — Значения атрибутов ключа не должны изменяться на протяжении всего времени его существования.

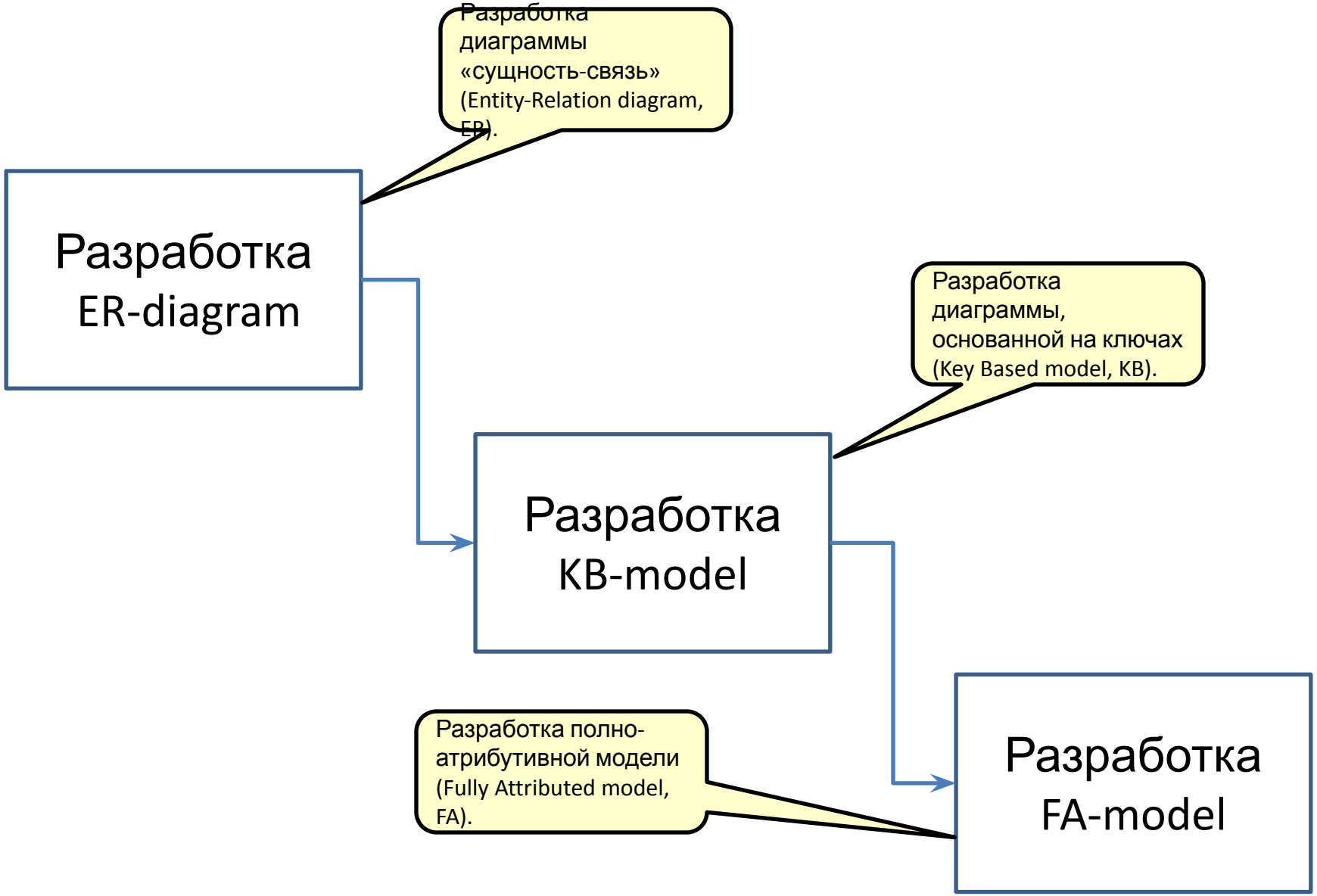
Пример ER диаграммы

Проектирование модели данных на примере Интернет магазина

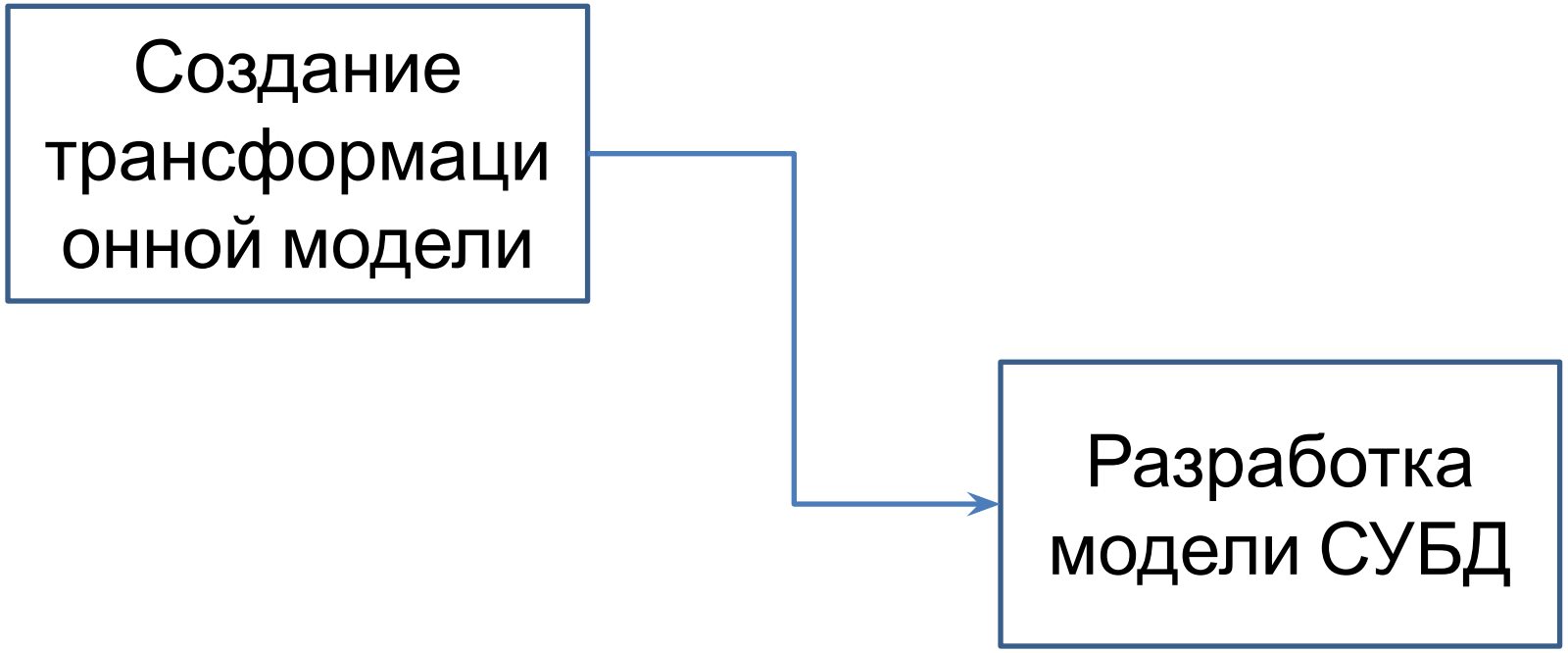
Фазы жизненного цикла проектирования



Этапы разработки логической модели данных



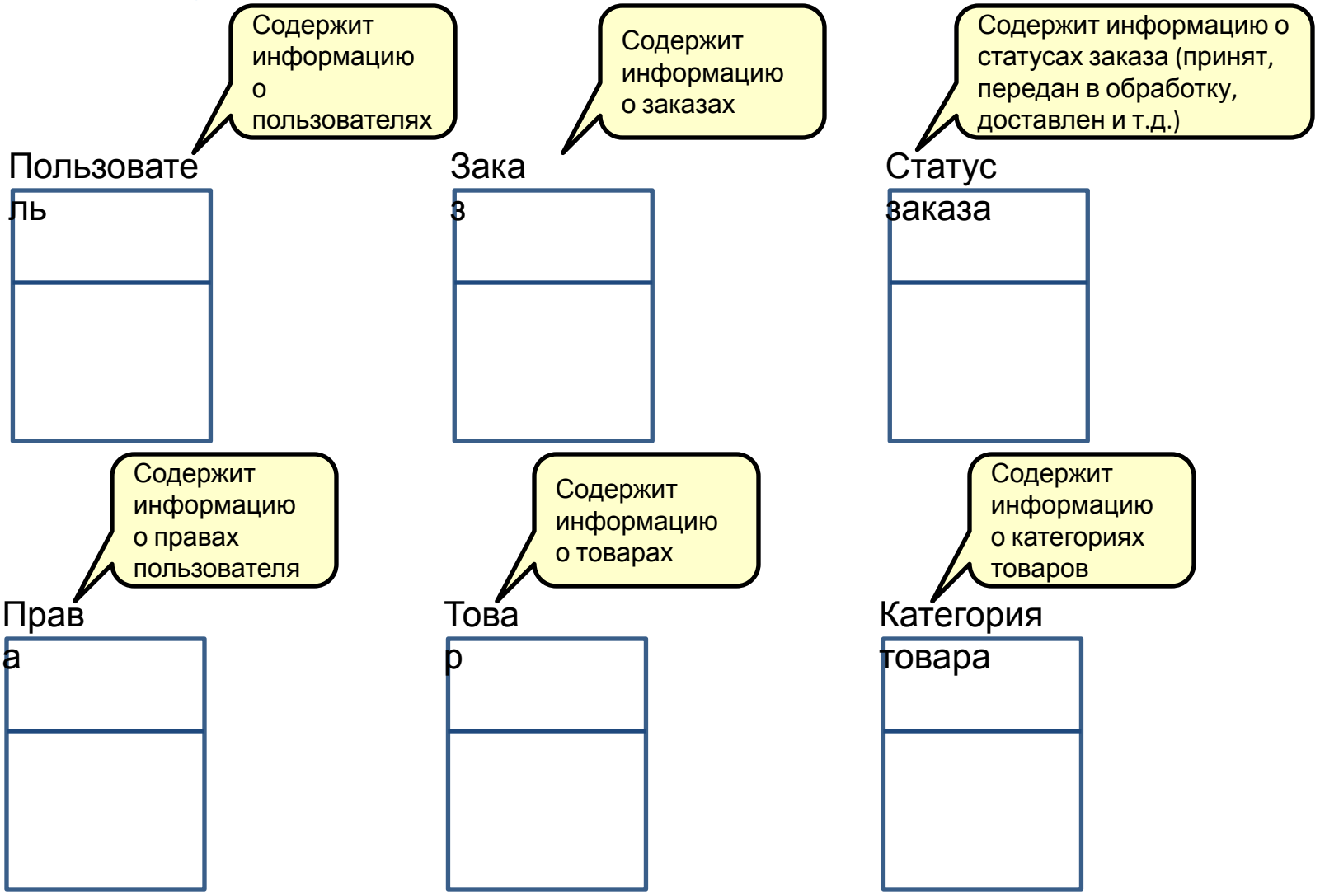
Этапы разработки физической модели данных



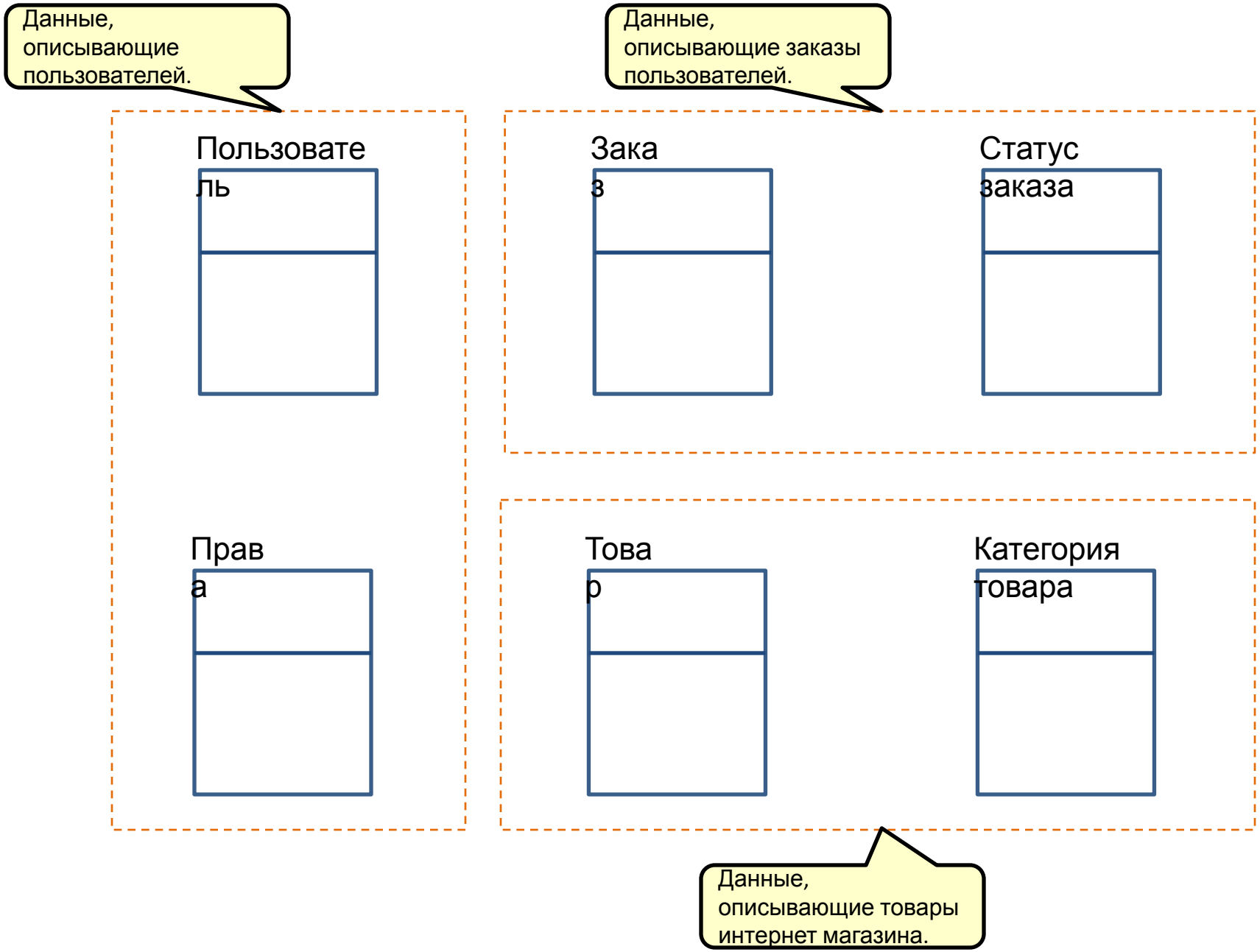
Разработка диаграммы «сущность-связь» (ER)

- ***Последовательность разработки***
  - Выделение сущностей предметной области (ПО)
  - Описание сущностей
  - Формирование тематических областей данных
  - Определение ключей и основных атрибутов сущностей
  - Определение связей
  - Описание связей

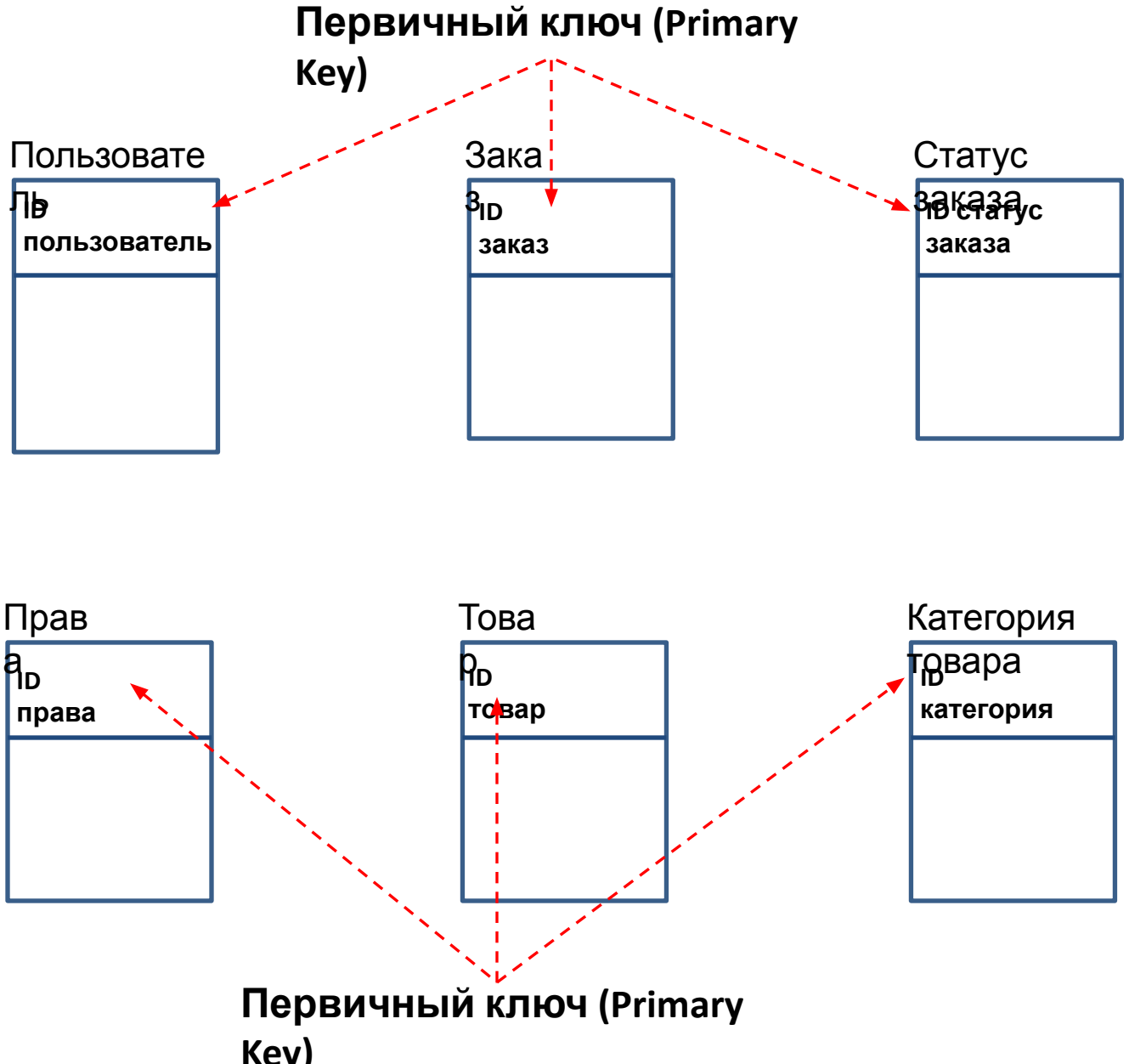
Выделение сущностей предметной области (ПО)



Выделение тематических областей данных



Разработка диаграммы, основанной на ключах (Key Based model, KB).



Разработка полно-атрибутивной модели (Fully Attributed model, FA)

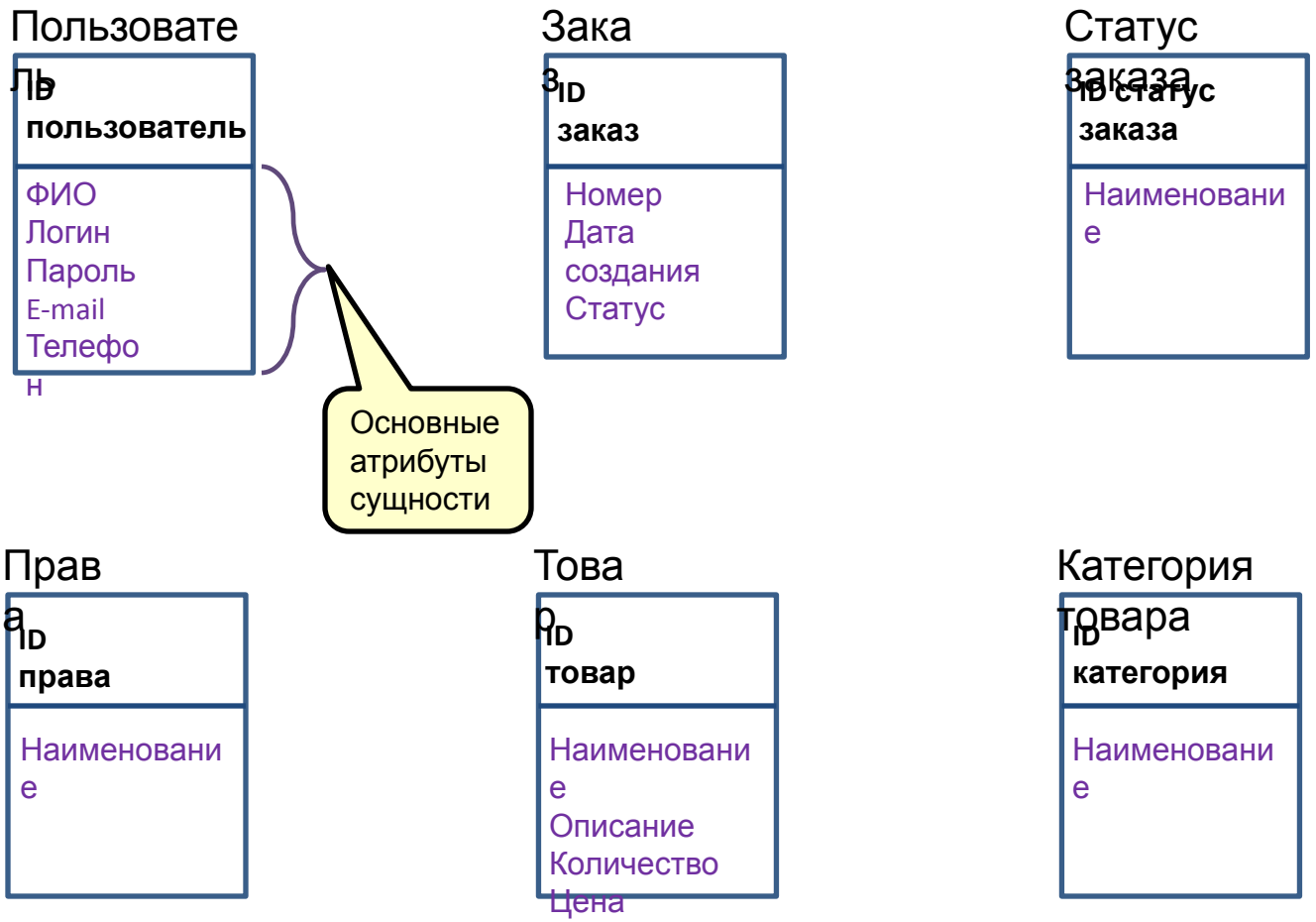
Порядок разработки FA Model

- Формирование полных наборов атрибутов каждой из сущностей.
- Использование доменов
- Имя связи в направлении предок – потомок (только для связей типа N..N)
- Нормализация

Рекомендации по приведению модели данных к нормальному виду

- Для приведения модели к **первой нормальной форме** удалите повторяющиеся атрибуты, либо группы атрибутов сущности
- Для приведения модели ко **второй нормальной форме** удалите атрибуты, зависящие только от части уникального идентификатора
- Для приведения модели к **третьей нормальной форме** удалите атрибуты, значения которых зависят от атрибутов, не входящих в уникальный идентификатор.

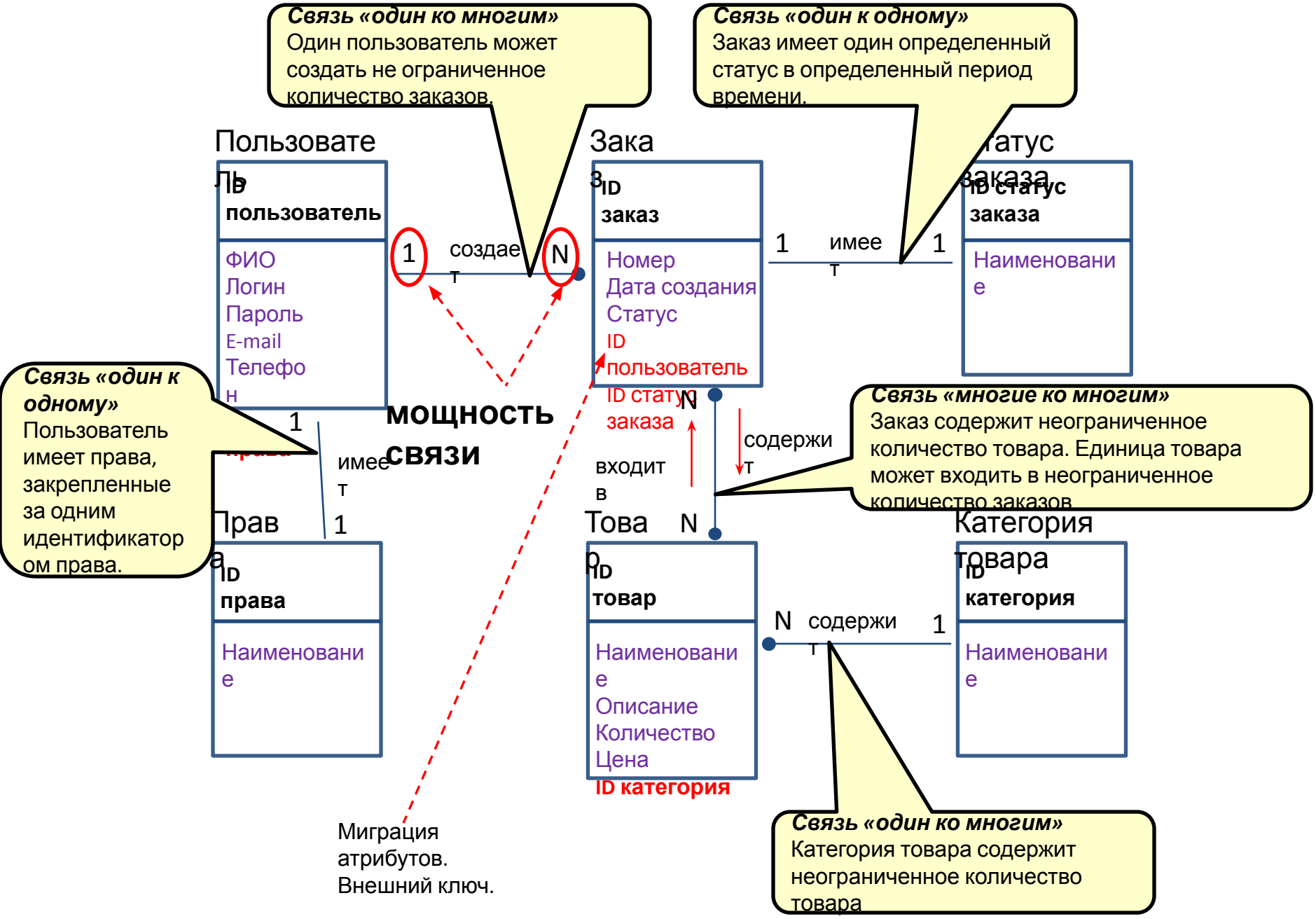
Разработка полно-атрибутивной модели (Fully Attributed model, FA)



Определение связей между сущностями ПО

- ***Порядок определения связей***
  - Классификация связи
  - Имя связи в направлении потомок – предок
  - Имя связи в направлении предок – потомок (только для связей типа N..N)
  - Мощность связи
  - Определение связи (по аналогии с определением сущности)

Определение связей между сущностями ПО

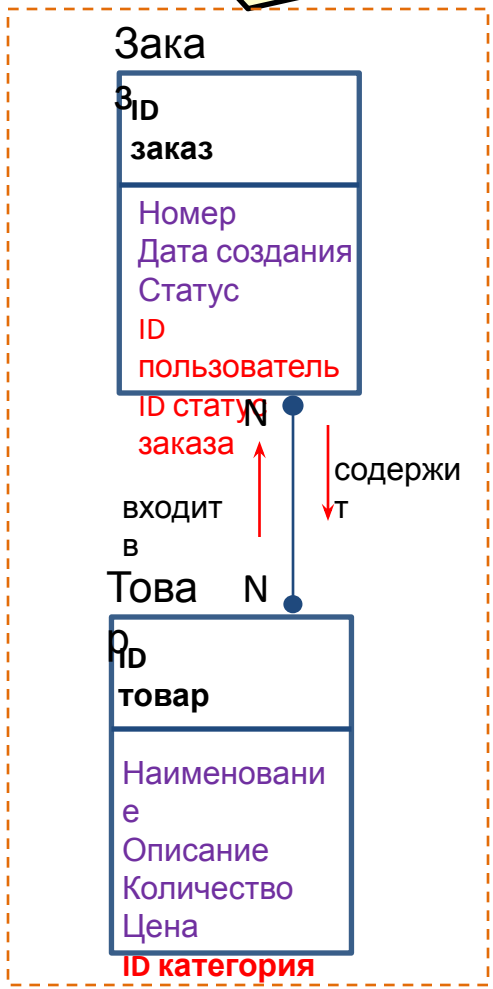


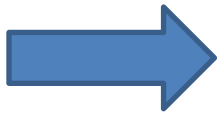
Миграция атрибутов

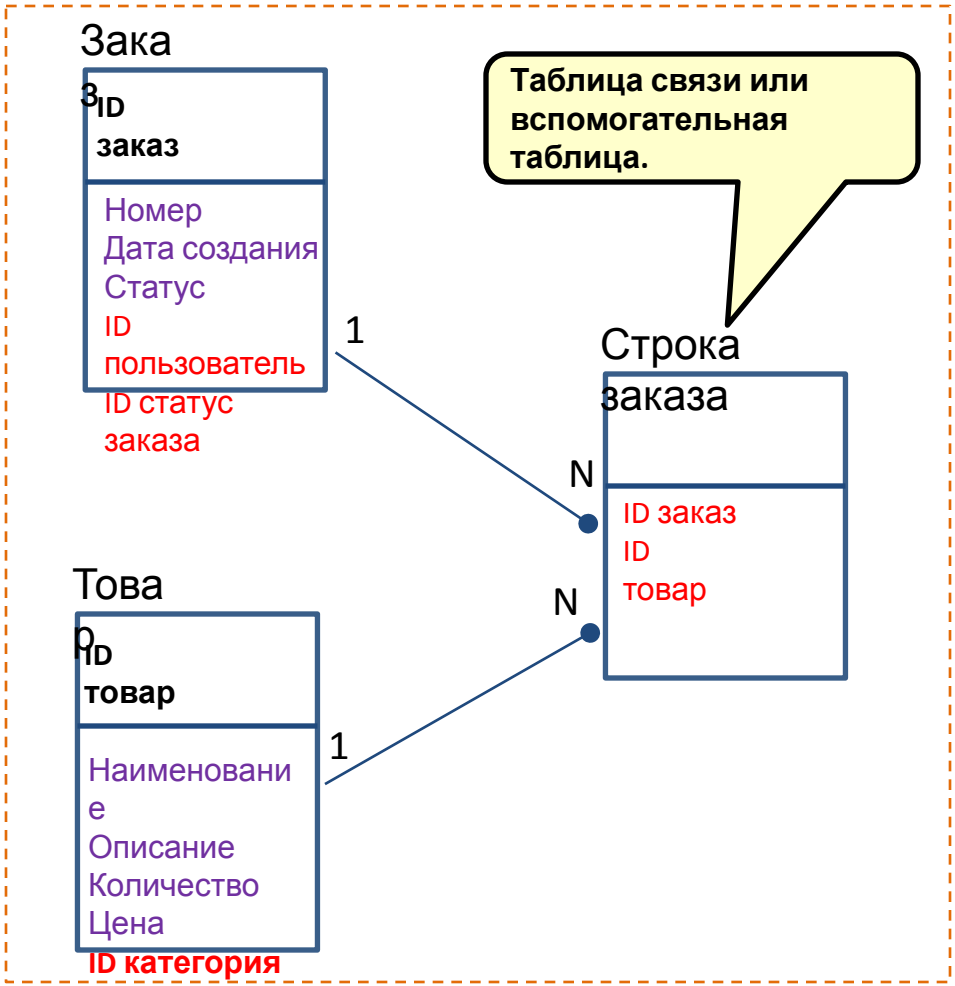
- При формировании связи между двумя сущностями атрибуты ПК родительской сущности мигрируют (копируются) в дочернюю сущность.
- Процесс миграции характерен как для идентифицирующей, так и к неидентифицирующей связи, однако его реализация различна.
- Атрибуты, мигрировавшие по идентифицирующей связи, входят в ПК дочерней сущности, а атрибуты, мигрировавшие по неидентифицирующей связи – нет.
- В любом случае, совокупность атрибутов, появившаяся в дочерней сущности путём миграции, образует так называемый ***внешний ключ*** (FK)

Связь «многие ко многим» в схеме СУБД

Рассмотрим связь «многие ко многим» в приведенном примере.
Заказ может содержать большое количество товара и каждый товар, в свою очередь, может входить в несколько заказов.
Для практической реализации данной связи в схеме СУБД необходимо **ввести таблицу связи**, где каждому *ID заказа* будет соответствовать несколько *ID товара*.







Создание трансформационной модели

- Создаётся CASE-средством автоматически.
- Содержит всю информацию, необходимую для генерации схемы данных в реляционной СУБД, поддерживающей стандарт SQL.
- Инвариантна к конкретной реализации.

Разработка модели СУБД

- Для перехода к **модели СУБД** необходимо выбрать в интерфейсе CASE-средства одну из доступных СУБД.

- Переход к физической модели не является необратимым: её можно будет трансформировать в другую физическую модель.

- Однако при этом все настройки, специфичные для данной СУБД, будут утеряны.