

ОСНОВЫ ТЕХНОЛОГИЙ БАЗ ДАННЫХ

Б. А. НОВИКОВ

Санкт-Петербургский государственный университет, JetBrains Research





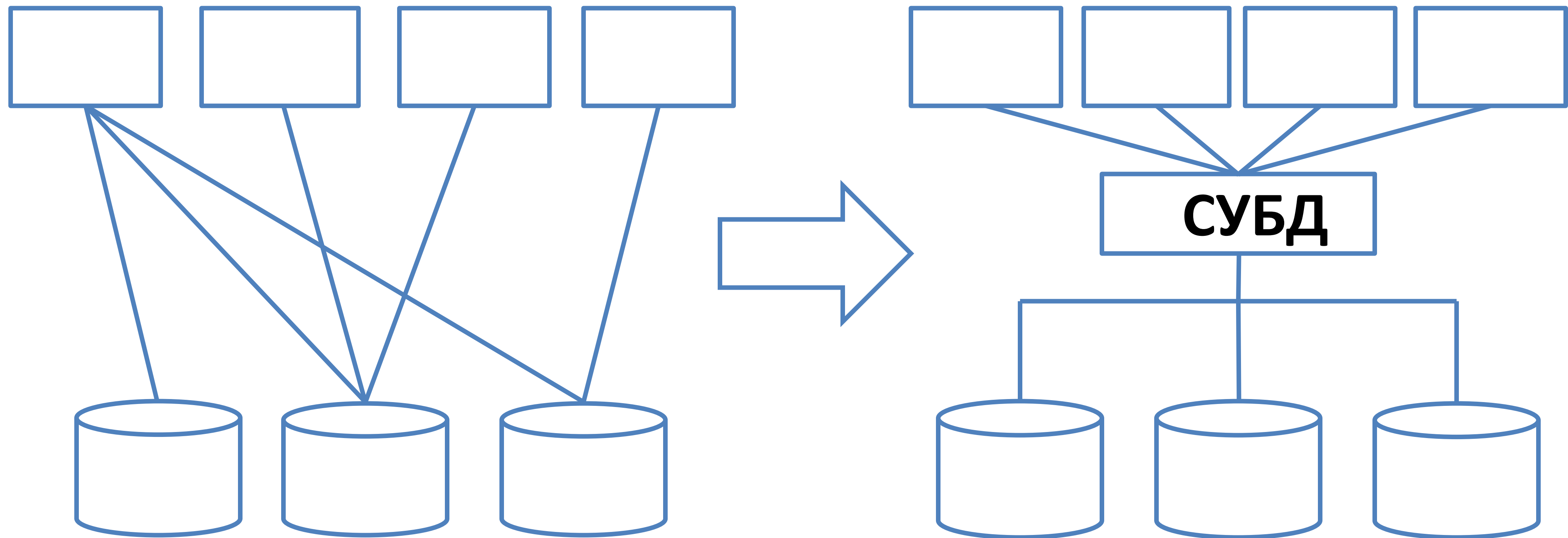
<https://postgrespro.ru/education/university/dbtech>

Предыдущие серии

краткое содержание



Централизация управления данными



Требования к системам управления базами данных (1971)

- Наличие средств описания данных
- Независимость данных и программ
- Высокоуровневый язык запросов
- Оперативный доступ к данным
- Совместное использование данных разными приложениями
- Целостность
- Согласованность
- Надежность
- Безопасность

Демонстрационная база данных PostgreSQL

<https://postgrespro.ru/education/demodb>



Чем отличаются модели данных?

- Методы идентификации объектов внутри и вне модели
- Поиск и связывание объектов
- Массовая или штучная обработка объектов данных

Домены и отношения

- Домен представляет собою множество скалярных значений и может описываться абстрактным типом данных
- Отношением (или предикатом) называется функция (нескольких аргументов), принимающая значения True или False
- Аргументы отношения называются атрибутами
- Каждый атрибут принимает значения в некотором домене
- В реляционной модели атрибуты принято обозначать именами (а не номерами)

Функциональные зависимости и ключи

- Набор атрибутов Y функционально зависит от набора атрибутов X , если для любого значения X существует ровно одно значение Y .

(passenger_id, flight) $\rightarrow$ seat

- Источник зависимостей: законы реального мира, но не конкретное содержание БД
- Тривиальные зависимости: любой атрибут зависит от полного набора атрибутов отношения
- Представляют ограничения целостности в реляционной модели
- Ключи могут использоваться для идентификации кортежей

Теоретическая реляционная модель данных: итоги

- Естественная идентификация по значениям атрибутов
- Идентификация определяется функциональными зависимостями
- Связи вычисляются динамически при выполнении операций соединения
- Ассоциативный доступ к данным по значениям (операция селекции)

**Не все базы данных
реляционные**



Модель данных «сущность-связь»



Модель концептуального уровня

- Peter Pin-Shan Chen. 1976. The entity-relationship model—toward a unified view of data. ACM Trans. Database Syst. 1, 1 (March 1976), 9-36. DOI=<http://dx.doi.org/10.1145/320434.320440>
- Высокоуровневое описание структур данных на основе анализа моделируемой системы
- Отсутствуют операции манипулирования данными
- Необходимо отображение на другую модель данных



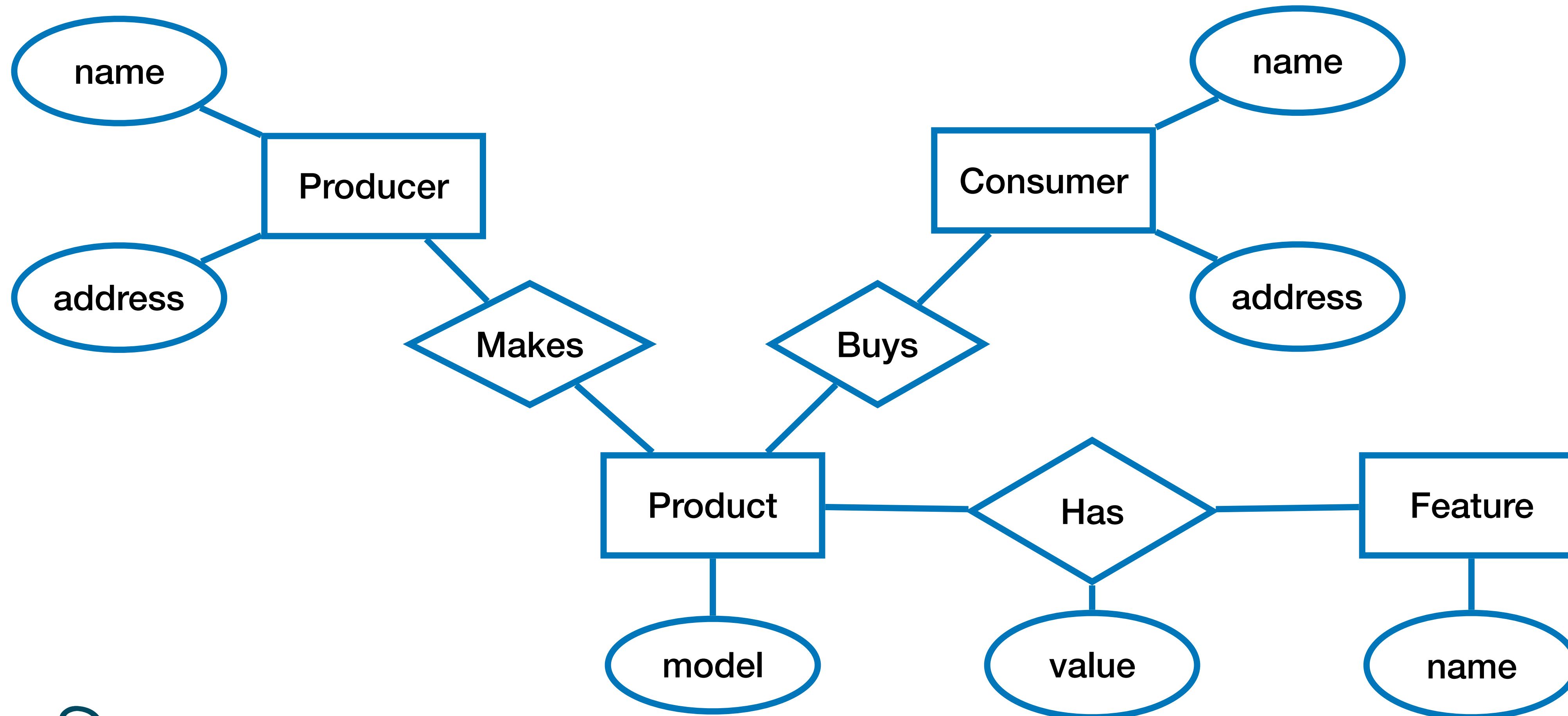
Сущности

- Сущность (entity) — это нечто в реальности, отделимое и отличимое от других сущностей
- Естественная идентификация
- Скалярные именованные атрибуты
- Множества сущностей (entity sets)
- Множества сущностей соответствуют типам в объектных моделях, а сущности — экземплярам

Связи и множества связей

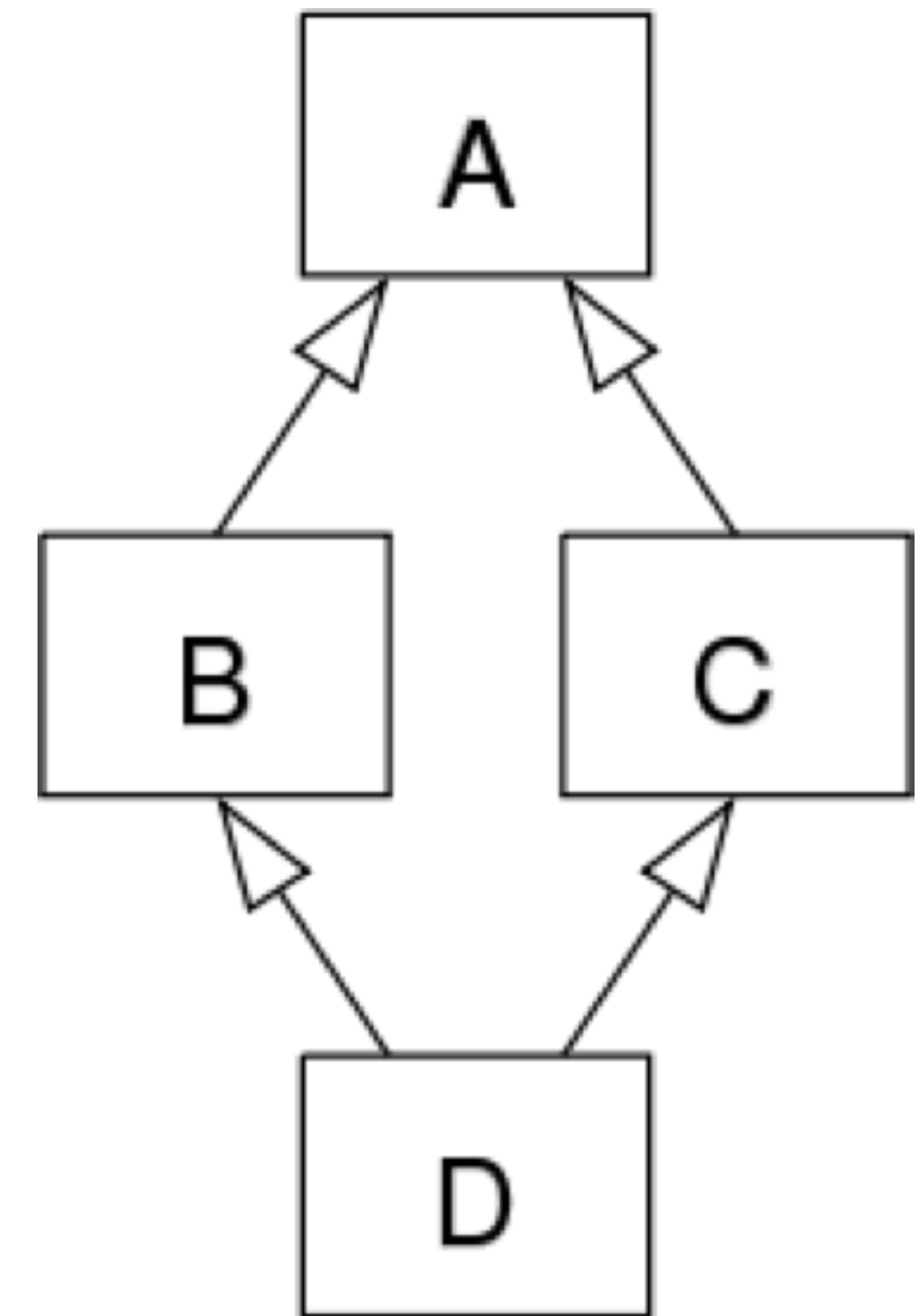
- Связь (relationship) — упорядоченный набор (экземпляров) сущностей
- Связь идентифицируется и набором идентификаторов сущностей, и своим идентификатором
- Связь может иметь атрибуты
- Бинарные связи связывают две сущности

Диаграммы

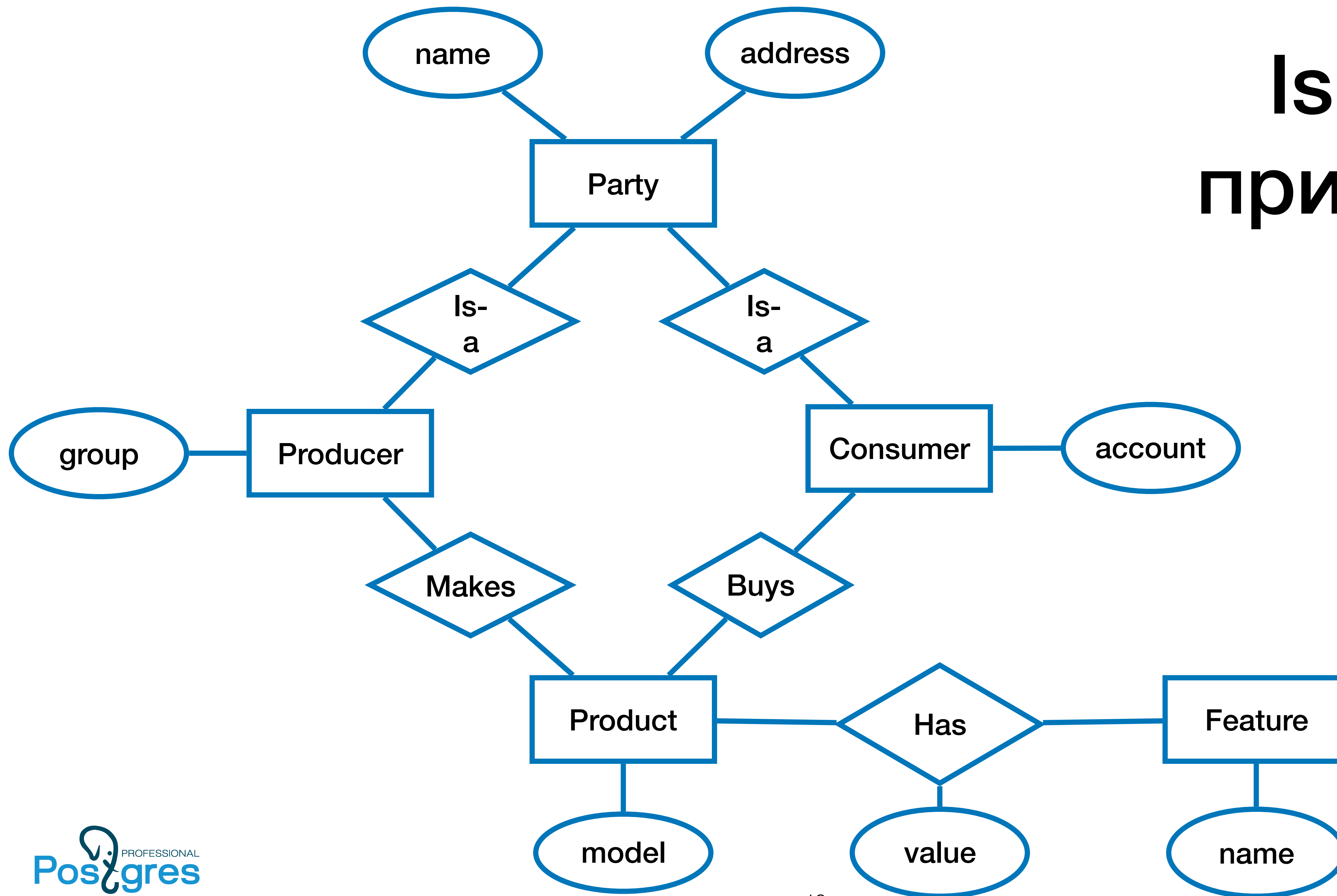


Is-A: вложенность множеств сущностей

- Операции обобщения и специализации: удаление/добавление атрибутов
- Ключ специализации должен совпадать с ключом обобщения
- Множественное наследование возможно, если имеется общее обобщение (ромб наследования)
- Атрибуты наследуются, но не копируются



Is a: пример



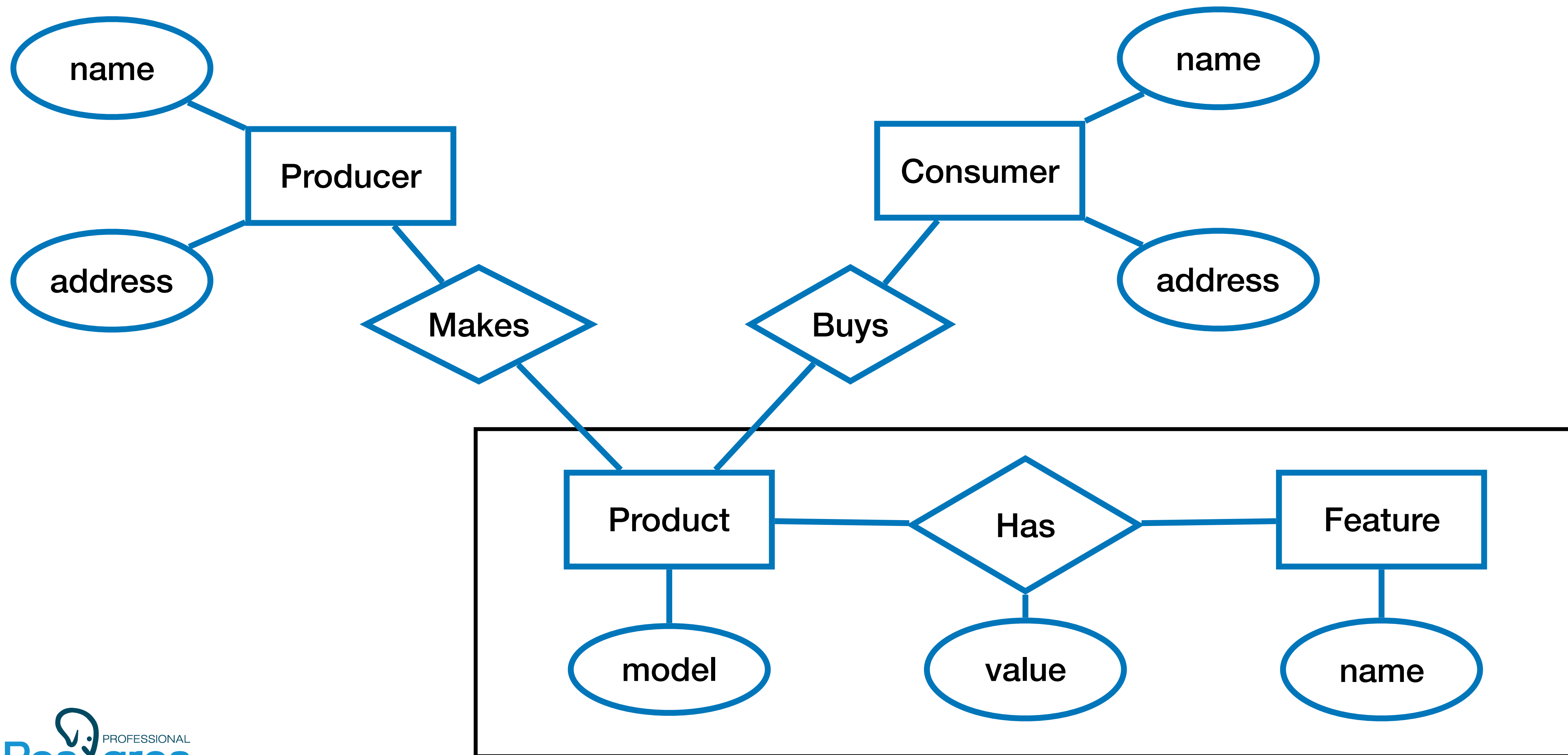
Ограничения целостности

- Задают ограничения на использование связей
- Определяется кратность связи для каждой участвующей сущности
- Бинарные связи: 0:1, 0:n, 1:n, m:n

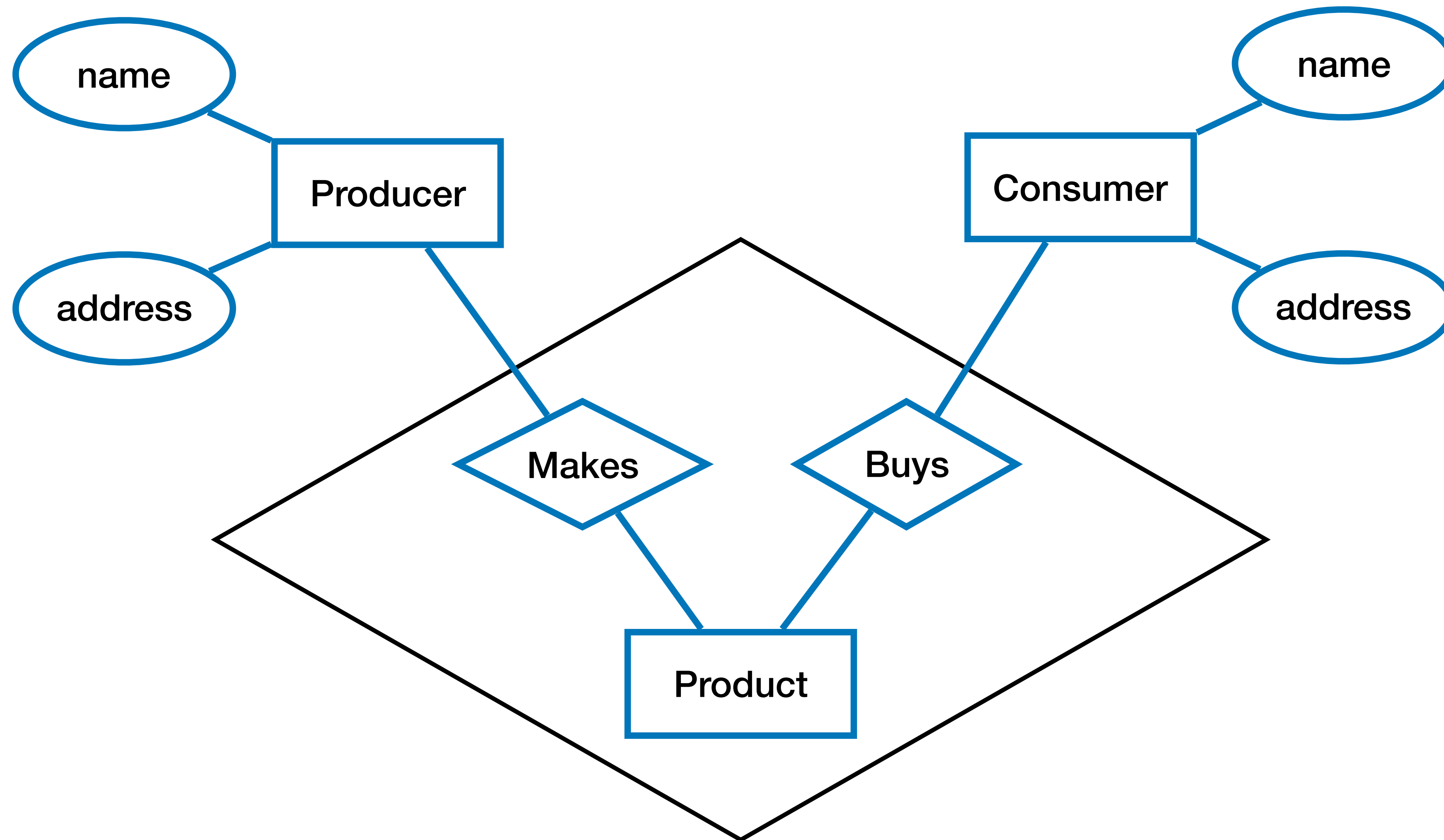
Агрегирование

- Большие диаграммы необходимо структурировать
- Агрегат – группа сущностей и связей, рассматриваемая как единая сущность и/или связь
- Если агрегированная сущность включает связь, она должна включать все сущности, связываемые этой связью

Агрегирование сущностей



Агрегирование связей



Другие средства модели «сущность-связь»

- Слабые сущности: не могут существовать в БД без связи с другими (сильными) сущностями
- Многочисленные расширения, добавляющие объектные возможности
- Модель постоянно развивается
- . . .

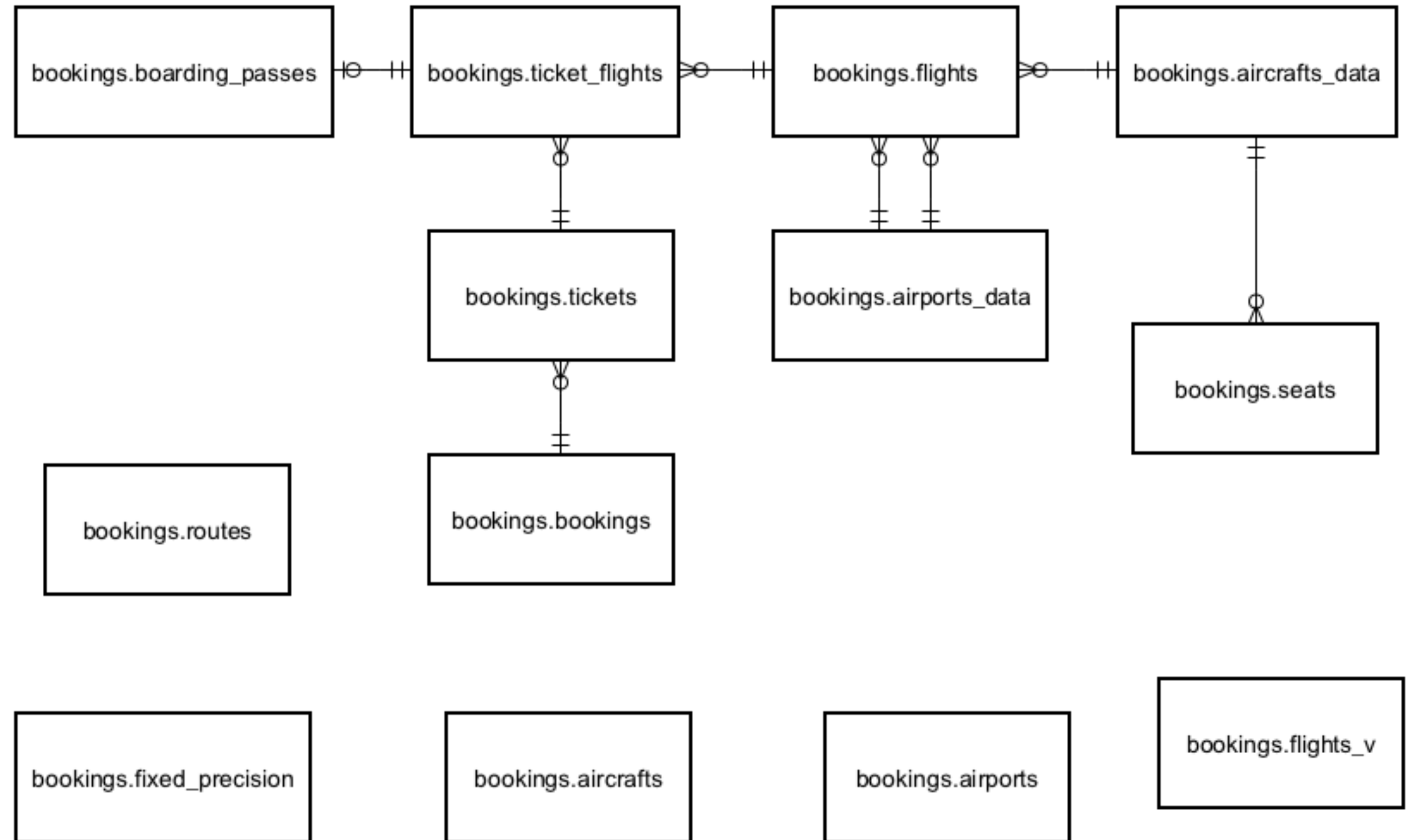
Проектирование в модели «СУЩНОСТЬ-СВЯЗЬ»

- Выбор между атрибутами с слабыми сущностями (адрес, номер телефона, ...)
- Выбор между сущностью и связью
- Бинарные или более сложные связи?
- Агрегирование?
- Ограничения целостности

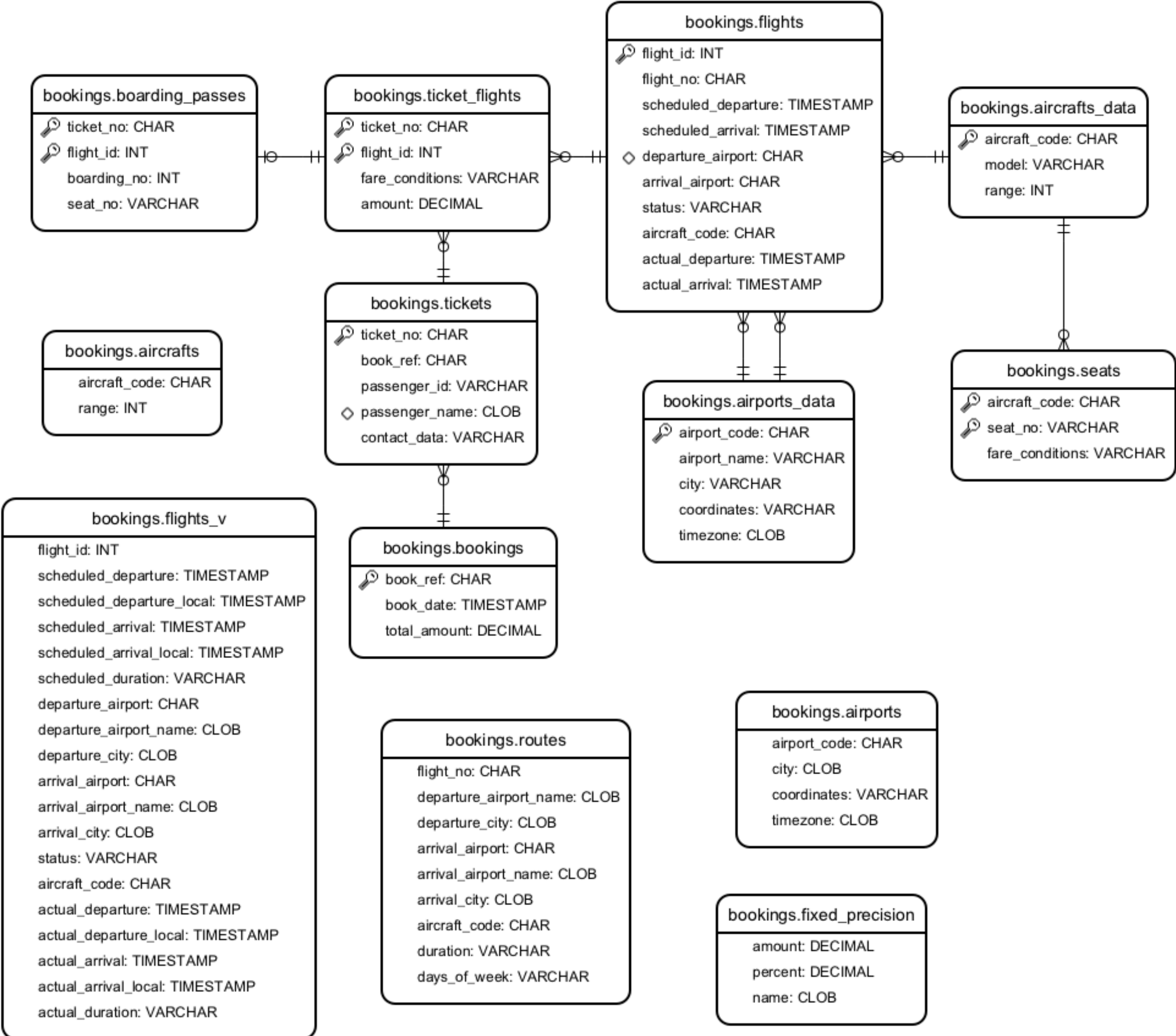
Вариации

- Модель с бинарными связями без атрибутов
 - Используются только бинарные связи
 - Связи в смысле канонической модели представляются сущностями
 - Все графические инструменты реализуют именно этот вариант
- Сокращенная нотация для атрибутов
- Представление информации о структурах хранения

Концептуальная ER-модель



Логическая модель





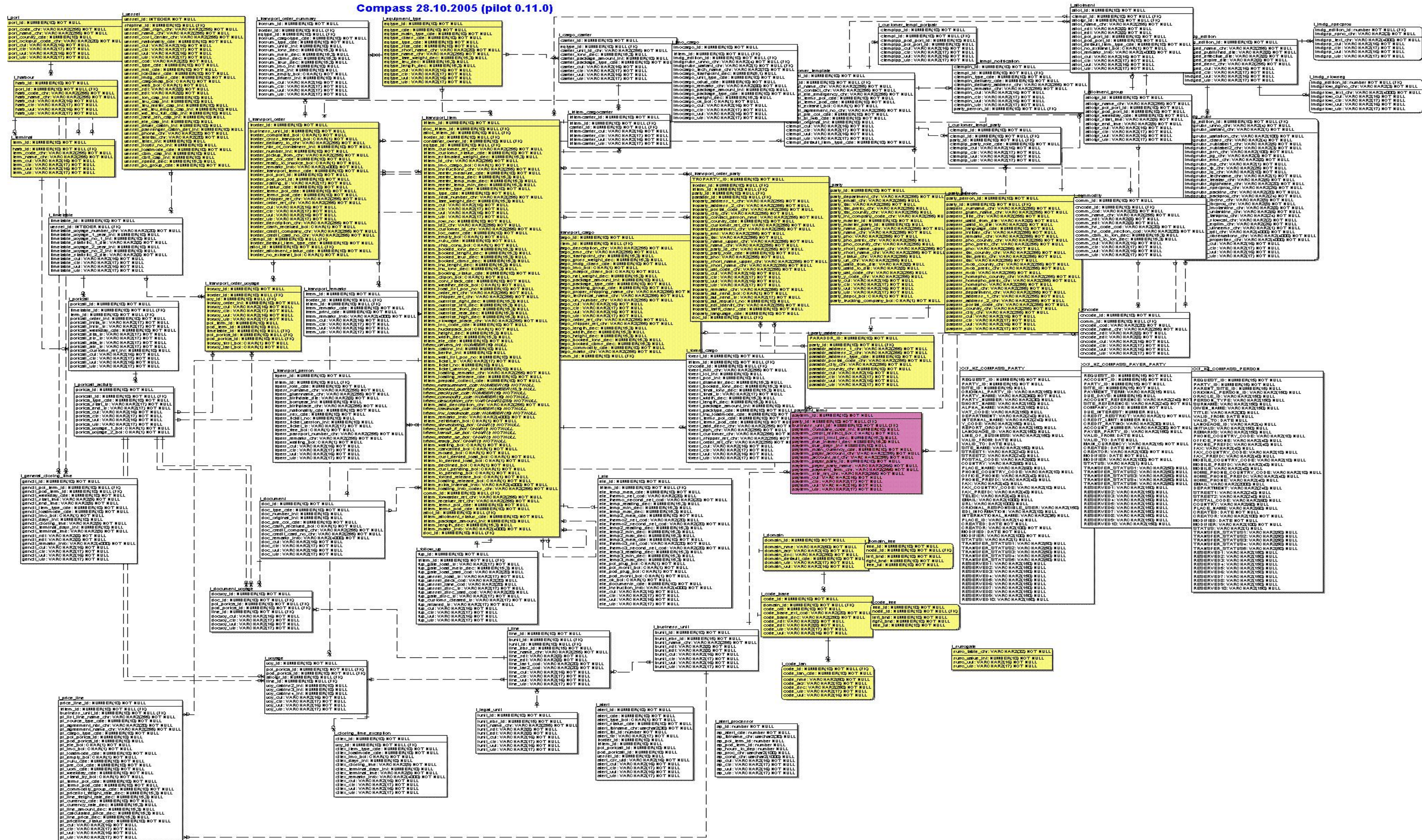


Диаграмма среднего размера

Предикатное представление модели

- Множество сущностей описывается предикатом, аргументы которого соответствуют атрибутам сущностей
- Множество связей описывается предикатом, аргументы которого представляют атрибуты связи и ключи связываемых сущностей
- Ограничения целостности описываются дополнительными предикатами
- Предикатное представление полезно для автоматической обработки больших схем

Пример предикатного представления

`Producer (name, address, group)`

`Consumer (name, address, account)`

`Product (model)`

`Feature (name)`

`Has (product, feature, value)`

`Buys (consumer, product)`

`Makes (producer, product)`

Отображение модели «сущность-связь» в реляционную

- По существу предикатов представление является реляционной схемой
- Идентификаторы сущностей отображаются в первичные ключи
- Остальные атрибуты функционально зависят от идентификаторов
- Если ER-модель спроектирована правильно, то реляционная схема получается в 3NF

Модель данных SQL



Модель SQL

- Системы, основанные на SQL, принято называть реляционными
- Несущественные отличия: отношения = таблицы, кортежи = строки, атрибуты = колонки или столбцы
- Существенные отличия от теоретической реляционной модели:
 - Допускается появление строк с совпадающими значениями всех атрибутов (дубликаты строк)
 - Ограниченный набор доменов
 - Допускаются неопределенные значения атрибутов (NULL)
 - Семантика операций отличается от реляционной
 - Расширенный набор операций

Неопределенные значения

- Упрощают проектирование схемы
- Создают сложности при поиске и обработке
- Отсутствует единая (универсально пригодная) семантика
 - Значение неизвестно
 - Значение отсутствует
 - Значение не имеет смысла
- Различаются ли неопределенные значения одного атрибута для разных объектов?

Дубликаты

- Логически бесполезны
- Для идентификации необходим выход за пределы модели
- Изменяют семантику операций массовой обработки
- Удаление дубликатов — относительно дорогостоящая операция

Объектно-ориентированные модели данных



В любой объектной модели должны быть

- Инкапсуляция
- Идентификация
- Наследование
- Полиморфизм

Манифесты

- Alcolm Atkinson, Francois Bancilhon, David DeWitt, Klaus Dittrich, David Maier, Stanley Zdonik. *The Object-Oriented Database System Manifesto*. Proc. 1st International Conference on Deductive and Object-Oriented Databases, Kyoto, Japan (1989). New York, N.Y.: Elsevier Science (1990).
- Michael Stonebraker, Lawrence A. Rowe, Bruce G. Lindsay, Jim Gray, Michael J. Carey, Michael L. Brodie, Philip A. Bernstein, and David Beech. 1990. *Third-generation database system manifesto*. SIGMOD Rec. 19, 3 (September 1990), 31–44. DOI=<http://dx.doi.org/10.1145/101077.390001>
- Hugh Darwen and C. J. Date. 1995. *The third manifesto*. SIGMOD Rec. 24, 1 (March 1995), 39–49. DOI=<http://dx.doi.org/10.1145/202660.202667>

Объектно-ориентированные СУБД

- Полная реализация всех принципов О-О
- Идентификация по суррогатам
- Навигация по объектным указателям
- Обработка отдельных объектов
- Исполнение методов (кода приложения) на сервере БД
- Тесная интеграция языка описания данных и языка программирования приложений



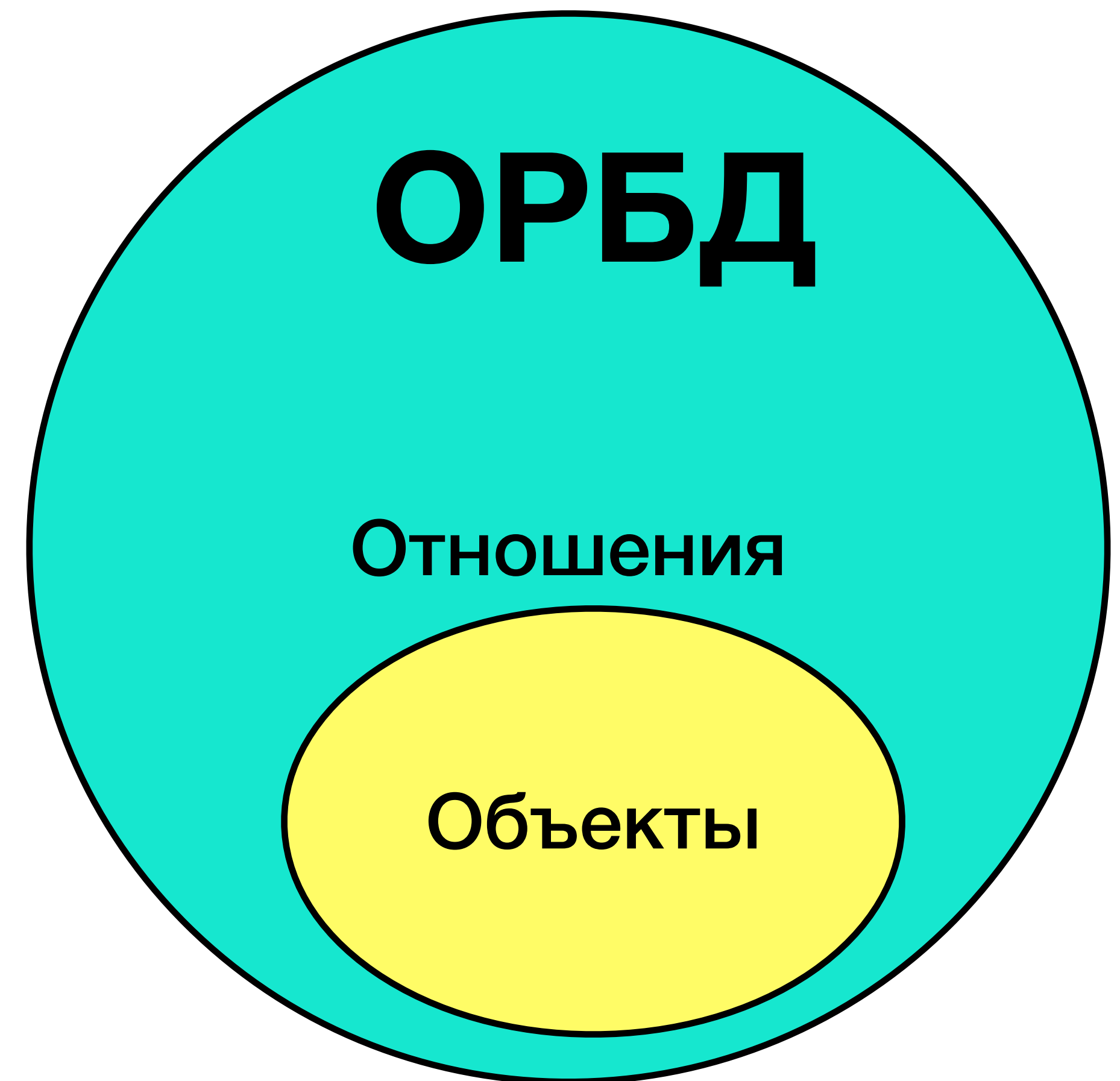
Особенности структур данных в объектно-ориентированных БД

- Пользовательские (абстрактные) типы данных
- Сложные (иерархические) структуры с вложенными коллекциями (NFNF)
- Типы коллекций: множества, сумки, списки, массивы, словари
- Бинарные связи без атрибутов, представленные объектными указателями
- Связи типа 1:m представляются массивами указателей

Ограниченность О-О баз данных

- Инкапсуляция препятствует эффективной работе с очень большими объемами данных
- Тесная интеграция с одним языком программирования мешает использованию других языков разработки приложений
- Организация совместного использования затруднена
- Хорошо работают для ограниченного набора областей применения

Объектно-ориентированные или объектно-реляционные?



Объектно-реляционные БД: расширение модели SQL

- Определение новых типов данных
- Пользовательские функции (методы, операции, ...)
- Типы объектных указателей
- Коллекции: set, bag, list, array
- NFNF-таблицы, операции nest, unnest

Модель данных PostgreSQL



PostgreSQL: определение ТИПОВ ДАННЫХ

- Домены
 - Дополнительные ограничения целостности
 - Не ограничивают преобразование к другим типам
- Составные типы
 - Объединяют несколько атрибутов, возможно, сложных
- Базовые типы
 - Требуют определения всех функций и операторов
 - Основной метод создания новых скалярных типов
- Массивы
 - Основной инструмент для создания вложенных коллекций

PostgreSQL: определение функций, операторов и процедур

- Расширяют возможность SQL
- Полиморфизм
- Не имеют состояния
- Обычные, агрегирующие, триггерные функции

Наследование

Модель сущность-связь

- Связь IS-A
- Каждый экземпляр сущности во всех множествах
- Идентификаторы совпадают
- Атрибуты не дублируются

PostgreSQL

- Наследование таблиц
- Каждая строка только в одной таблице
- Атрибуты обобщения копируются в специализацию

Коллекции в PostgreSQL

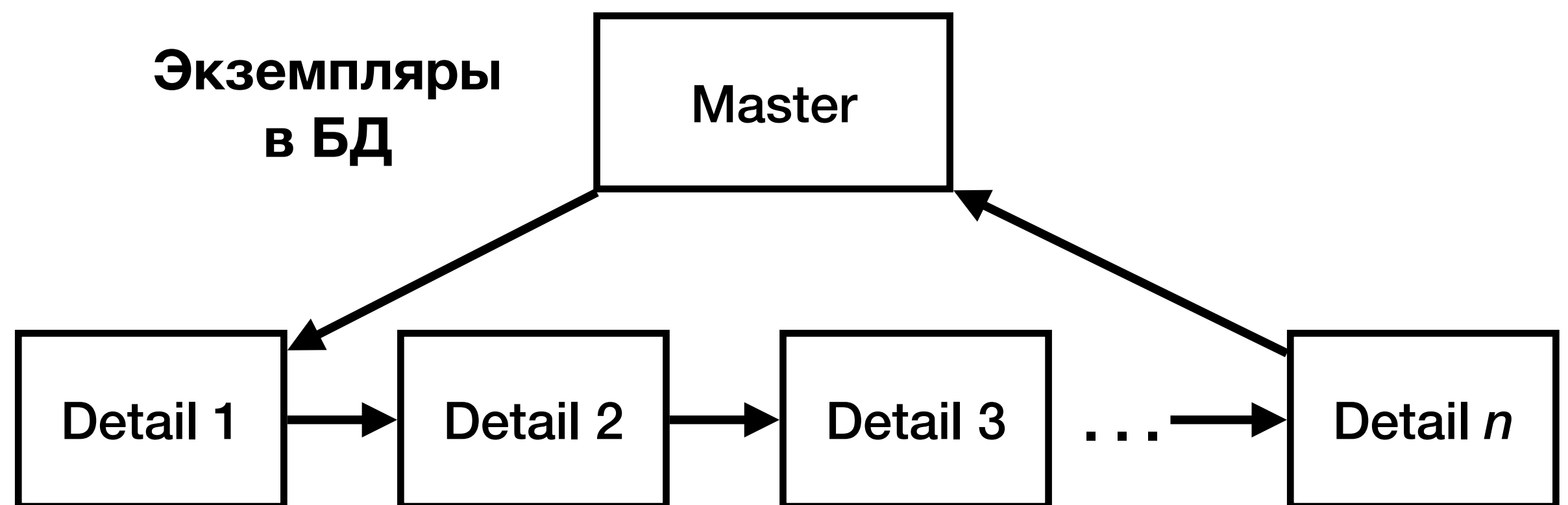
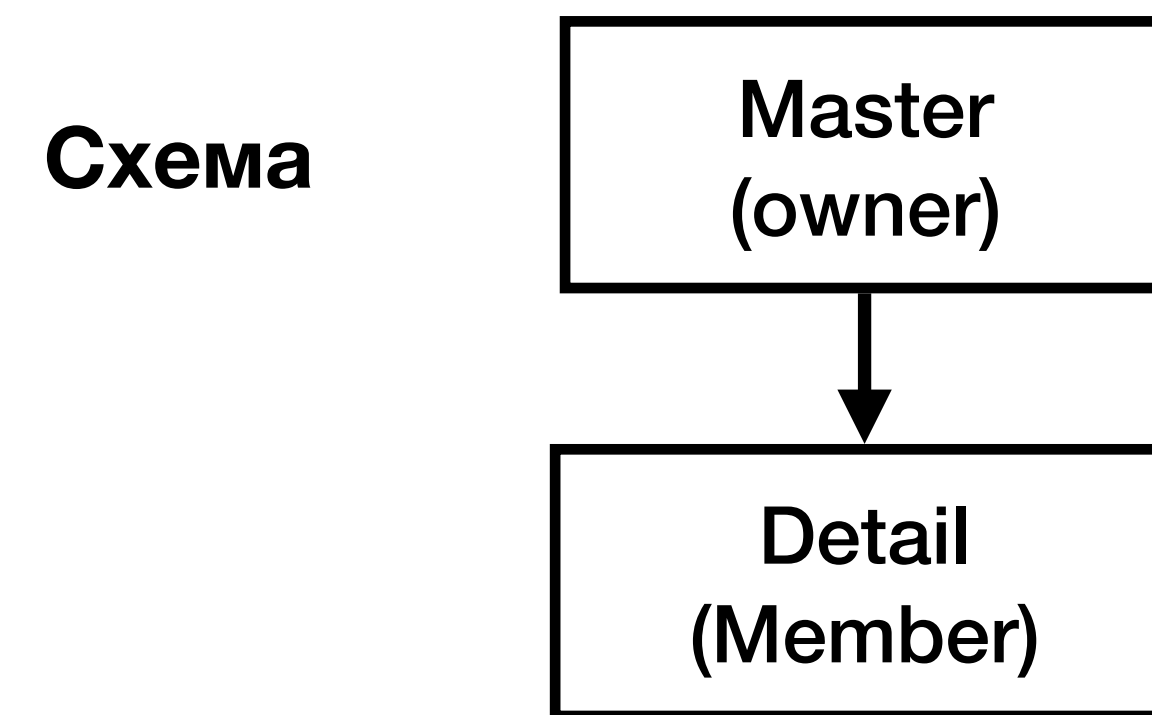
- Функции, возвращающие множества (возможно, с дубликатами)
- Массивы (одномерные и многомерные)
- Вложенные коллекции реализуются на основе массивов и структурных типов данных

Устаревшие модели данных



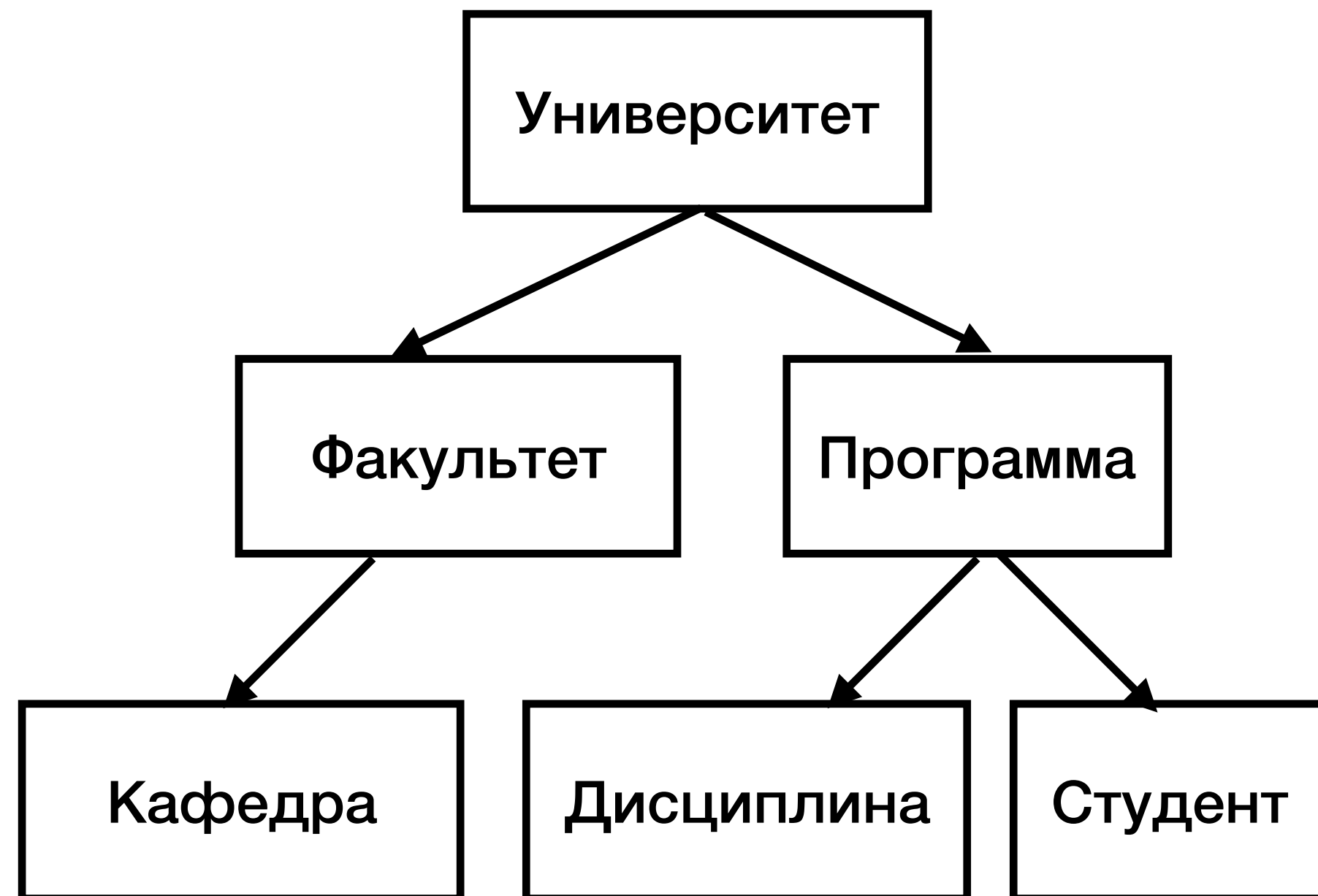
Сетевая модель данных CODASYL (1969)

- Записи и наборы
- Диаграммы Бахмана
- Навигация
- Обработка отдельных записей
- Идентификация по адресу в базе данных

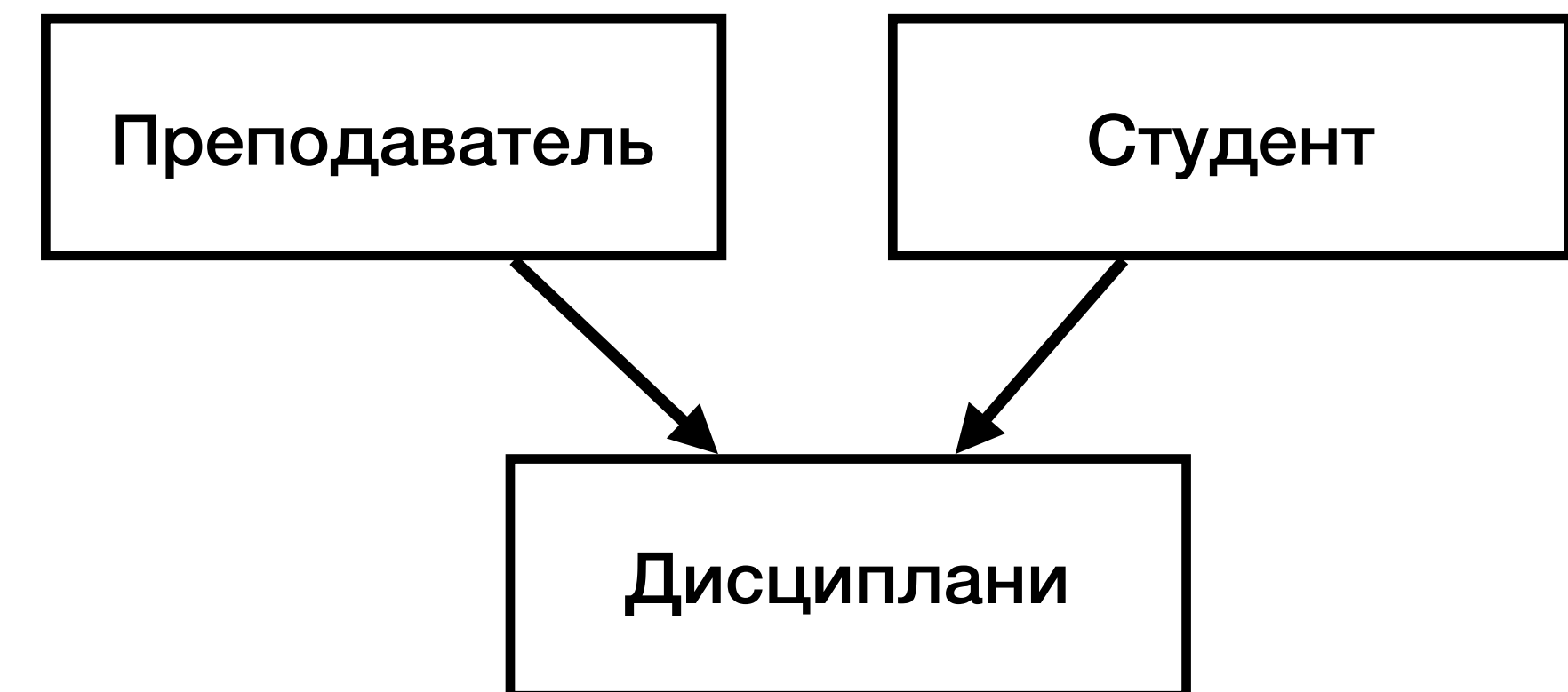


Построение сложных структур в сетевой модели данных

Иерархия



Сетевая структура

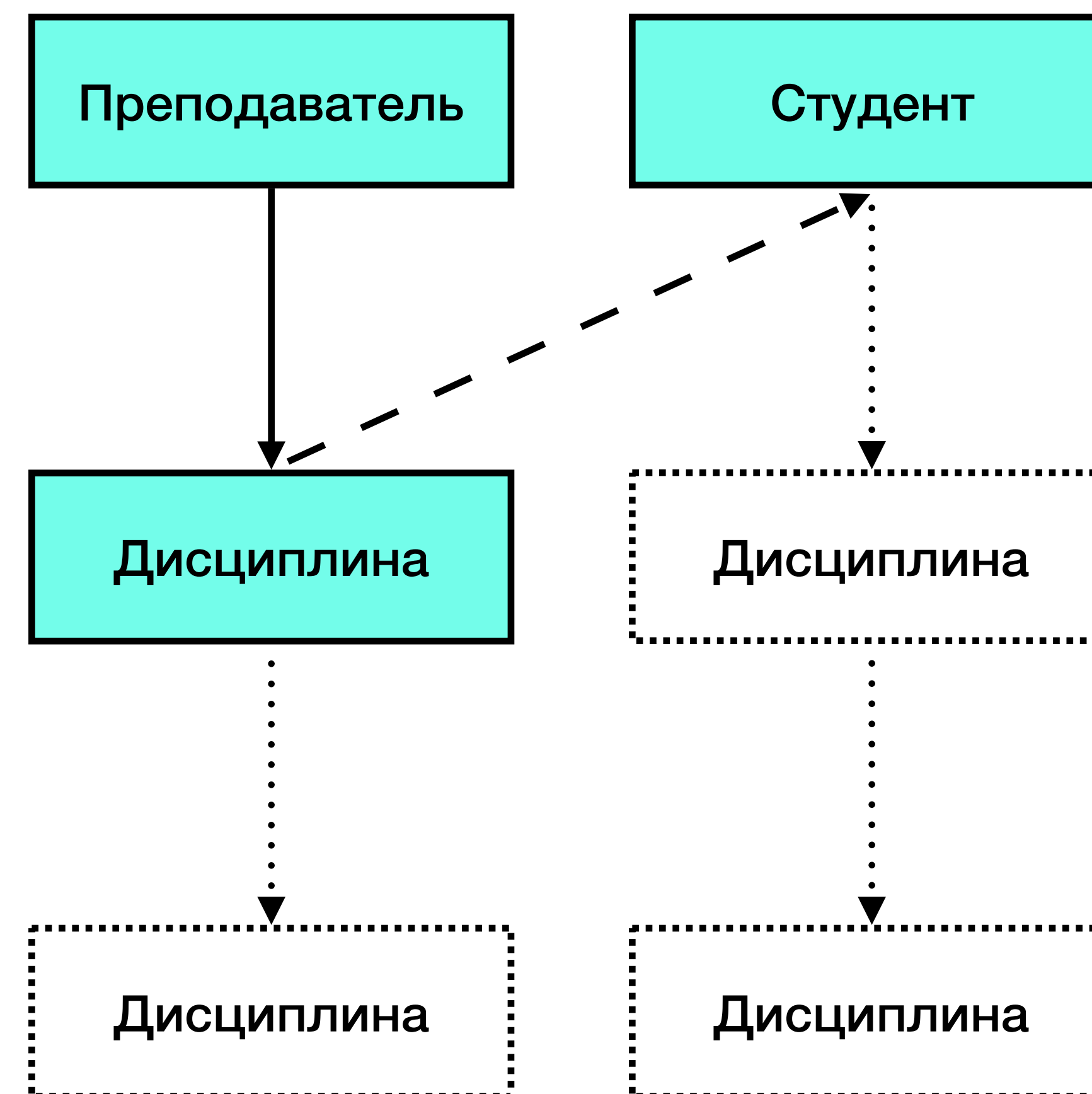


Операции и ограничения целостности в сетевой модели данных

- Текущая запись для каждого типа записей
- Переход по набору от владельца к члену, к следующей, к предыдущей, к владельцу, ...
- Добавление, изменение и удаление записей
- Включение и исключение из набора
- Обязательное членство в наборе
- Автоматическое включение, исключение, удаление

Иерархическая модель данных

- Реализована в системе IMS/360 и IMS/370
- Доминировала на протяжении двух десятилетий
- Тщательная проработка вопросов эффективности
- Иерархические подсхемы, сетевая схема
- Навигация и поиск в пределах иерархии
- Обработка отдельных записей



Отображение сетевой модели данных в модель SQL

- Записи сетевой модели представляются строками в таблицах SQL
- Поля записей соответствуют атрибутам
- Необходимо ввести суррогаты для идентификации записей
- Для представления наборов вводятся дополнительные атрибуты, содержащие идентификатор владельца
- Ограничения целостности отображаются в ограничения целостности SQL

Слабоструктурированные модели данных



Semistructured: слабоструктурированные или полуструктурированные?

- Задачи интеграции неоднородных источников данных
- Ориентация только на чтение данных
- Допустимость отклонений от схемы или работа с неизвестной схемой
- Допустимость пропусков и ошибок
- Модели малого веса (без схемы) и тяжелого веса (со схемой)
- Стирается граница между схемой и данными
- Описывающие или предписывающие схемы?
- Ранние модели (lore) основаны на представлении ориентированных графов



**Hector
Garcia-Molina**

XML

- LaTeX - SGML - HTML - XML
- Самоописывающее представление документов
- Средство передачи данных, но не хранения
- Документы без схемы, со схемой DTD и XSD
- Гибкие возможности определения ограничений на структуру и содержание документов
- Иерархические структуры (вспоминаем устаревшие модели)
- Native XML databases

JSON

- Формат для передачи данных
- Более экономное (по сравнению с XML) представление структуры
- Стандарт схем появился значительно позже, чем JSON
- Иерархическая структура данных

Отображения в модель SQL

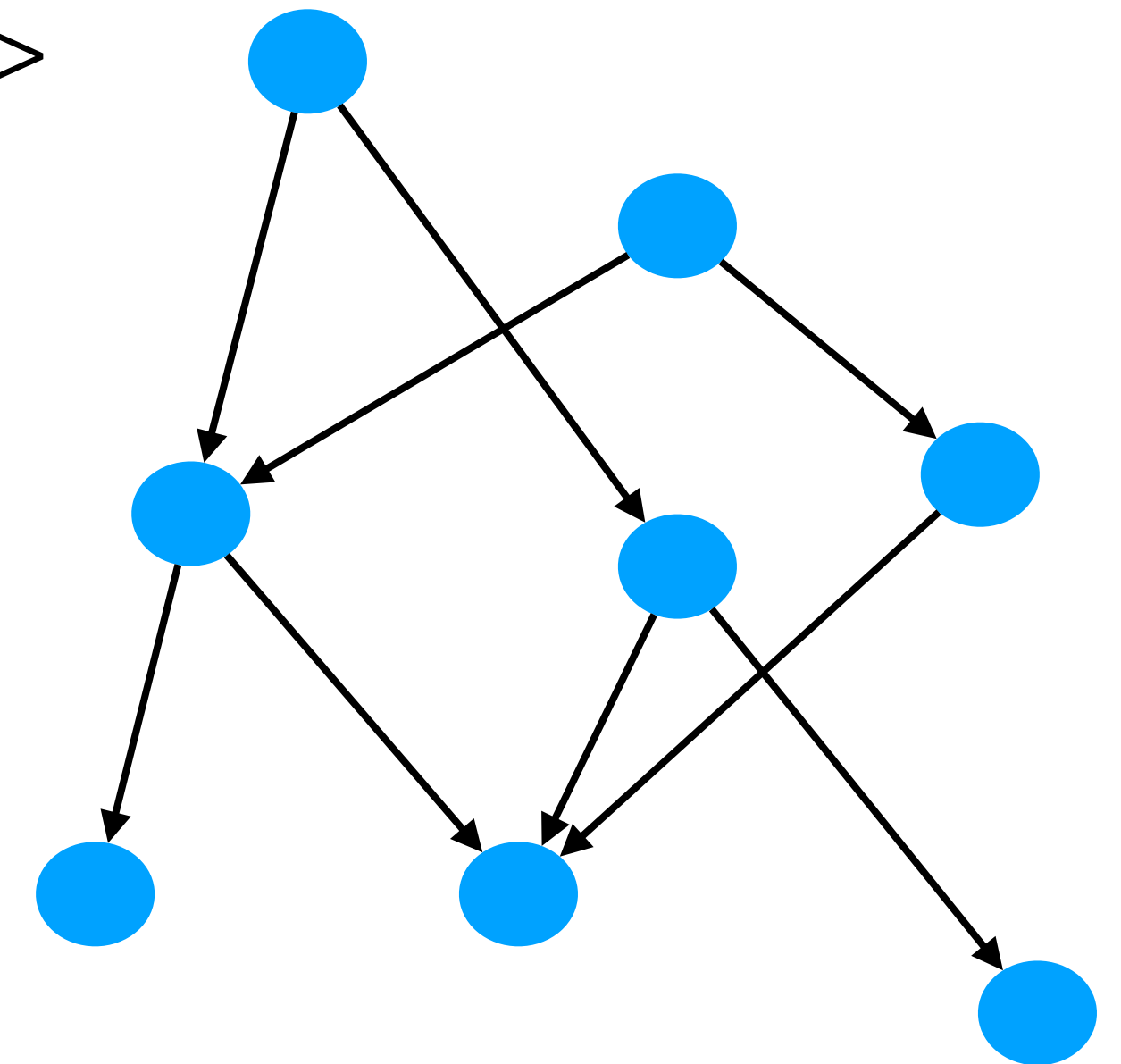
- Вспоминаем устаревшую иерархическую модель данных
- Необходима хотя бы описывающая схема
- Каждый тип представляется отдельной таблицей
- Необходимы дополнительные (суррогатные) атрибуты для идентификации и представления упорядочения элементов (объектов)
- Ограничения на значения элементов представляются ограничениями целостности

Модели для представления графов



Тернарная модель данных

- < идентификатор объекта, имя атрибута, идентификатор значения >
- Произвольные ориентированные графы
- Семантические сети (1980-е гг.)
- RDF: < субъект, предикат, объект >
- Linked data
- Обычно крайне неэффективны при прямолинейном представлении в базе данных SQL



Бинарная модель данных

- Отдельно для каждого атрибута: { идентификатор-объекта, значение }
- Применяется как логическая структура в «колоночных» СУБД
- Эффективна для задач оперативной аналитической обработки (OLAP)
- Не следует рассматривать как альтернативу модели данных SQL

А еще есть

- Плоские файлы (CSV, ...)
- Ключ-значение
- ...

Модели данных: итоги

- Разнообразие моделей данных отражает разнообразие потребностей
- Существенные свойства любой модели данных можно отобразить в другие модели
- Особенности данных, которые невозможно представить в модели, необходимо реализовывать в программном коде приложения
- Чем проще и беднее модель, тем больше потребуется программировать