

Модели данных

В2. Универсальные модели. EAV



Московский государственный технический университет
имени Н.Э. Баумана

Факультет ИБМ

сен 2024 года

Москва

Артемьев Валерий Иванович © 2024

9. Универсальные модели данных.

Модели «сущность – атрибут – значение»

- Что такое универсальная модель данных?
- Переход к метамоделям и денормализация
- Сценарии применения
- Широкие таблицы
- Узкие таблицы
- Модели «сущность – атрибут – значение» EAV
- Варианты реализации модели EAV
- Пример финансовой отчётности
- Пример магазина с широкой номенклатурой товаров
- Достоинства и недостатки модели EAV
- Альтернативы: применение XML / JSON

Вопросы для самопроверки

Актуализировать,
когда устоится

Что такое архитектура данных?

Области
управления
данными



➤ Архитектура данных

образует фундамент управления данными. Удовлетворяет насущную потребность в их представлении на разных уровнях абстракции так, чтобы они были понимаемы бизнесом и руководство могло принимать относительно этих данных соответствующие решения.

Архитектура данных включает:

- **Корпоративная модель данных**
- **Схемы потоков данных и их преобразований**

Корпоративная модель данных

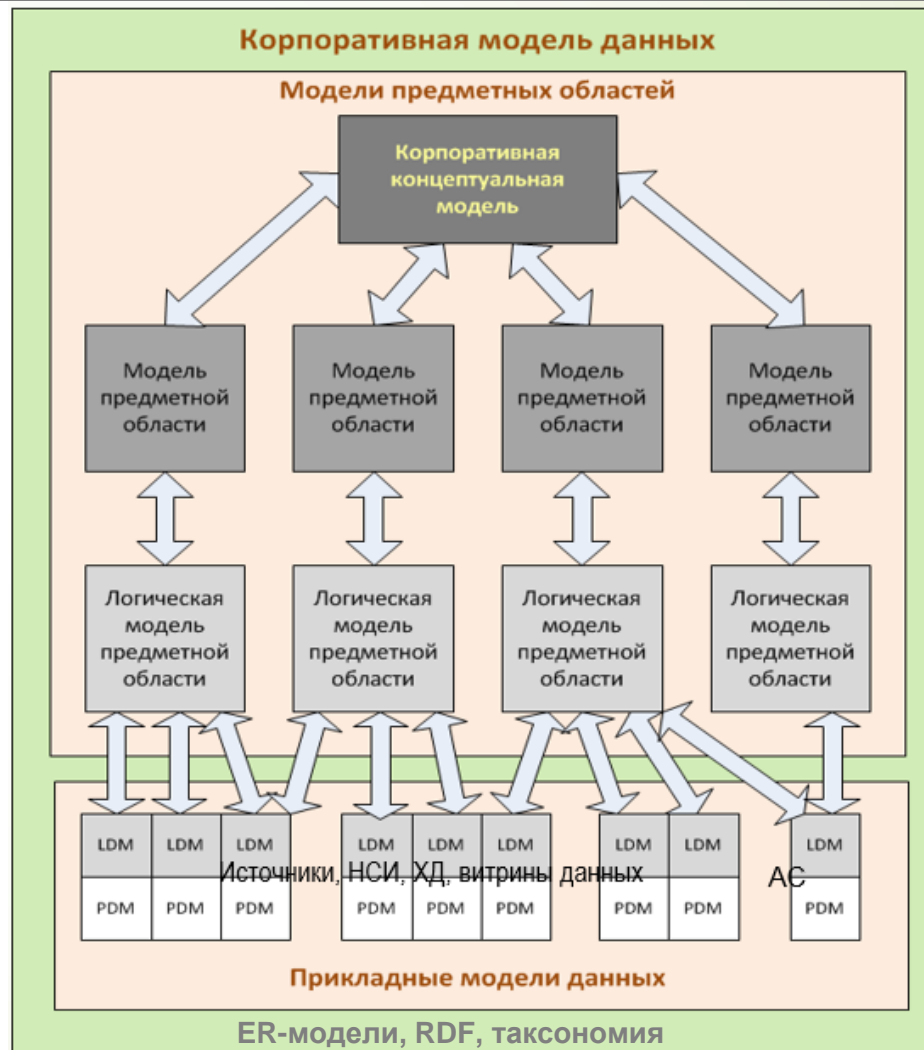
**Здесь широко используются
универсальные модели данных**

Концептуальный обзор всех
предметных областей
корпорации

Концептуальное
представление сущностей и
связей для каждой
предметной области

Логическое представление
для каждой предметной
области

Подробные логические и
физические модели данных,
специфичные для
реализации приложений или
проектов



Традиционные модели данных

- Ориентированы на создание БД для *узкого применения*
- *Затрудняют адаптацию* структуры и состава данных
- Добавление атрибута или сущности *приводит к реорганизации* схемы
- *Нельзя использовать* модель для разных предметных областей
- *Специализированы* для предметной области и *неуниверсальны*:
 - определённой бизнес-сущности соответствует таблица
 - каждый экземпляр сущности – запись в таблице
 - каждый атрибут экземпляра сущности – колонка таблицы
 - каждое значение – в своей ячейки таблицы.

country	year	cases	population
Afghanistan	1999	7745	19987071
Afghanistan	2000	7666	2059360
Brazil	1999	37737	17206362
Brazil	2000	80488	174504898
China	1999	217258	1272015272
China	2000	216766	128012583

variables

country	year	cases	population
Afghanistan	1999	7745	19987071
Afghanistan	2000	7666	2059360
Brazil	1999	37737	17206362
Brazil	2000	80488	174504898
China	1999	217258	1272015272
China	2000	216766	128012583

observations

country	year	cases	population
Afghanistan	1999	7745	19987071
Afghanistan	2000	7666	2059360
Brazil	1999	37737	17206362
Brazil	2000	80488	174504898
China	1999	217258	1272015272
China	2000	216766	128012583

values

Что такое универсальная модель данных?

Универсальная модель данных (УМД) представляет собой стандартную реляционную модель для гибкого представления разных наборов данных и / или определённые приёмы метамоделирования с помощью стандартных компонентов.

Два основных вида УМД:

1. Создание и расширение УМД с *фиксированным набором таблиц* требует только добавления новых записей в существующие таблицы без изменения структуры данных.
2. Создание и расширение УМД путём соединения *типовых компонентов метамодели по определённым правилам.*

Возможности и особенности универсальных моделей данных

- УМД – результат стандартизации моделей данных
- Переход к метамоделям данных
- Структура УМД может не зависеть от набора данных
- УМД можно настроить для разных проблемных областей
- Повышение надежности решений по проектированию БД
- Сокращению затрат на создание и развитие БД.

Сценарии применения

Неопределённый или переменный состав атрибутов сущности предметной области

Пример 1. Химические анализы минералов

Пример 2. Состав продуктов питания

Пример 3. Состав финансовой отчётности

Пример 4. Медицинская диагностика

Объединение сущностей, различающихся атрибутами

Пример 5. Ассортимент товаров в магазине

Пример 6. Работа с разными типами клиентов компании

Неопределённый состав сущностей предметной области

Пример 7. Учёт объектов городского хозяйства

Пример 8. Прототипирование БД (?)

Модели «сущность – атрибут – значение» EAV (Entity-Attribute-Value)

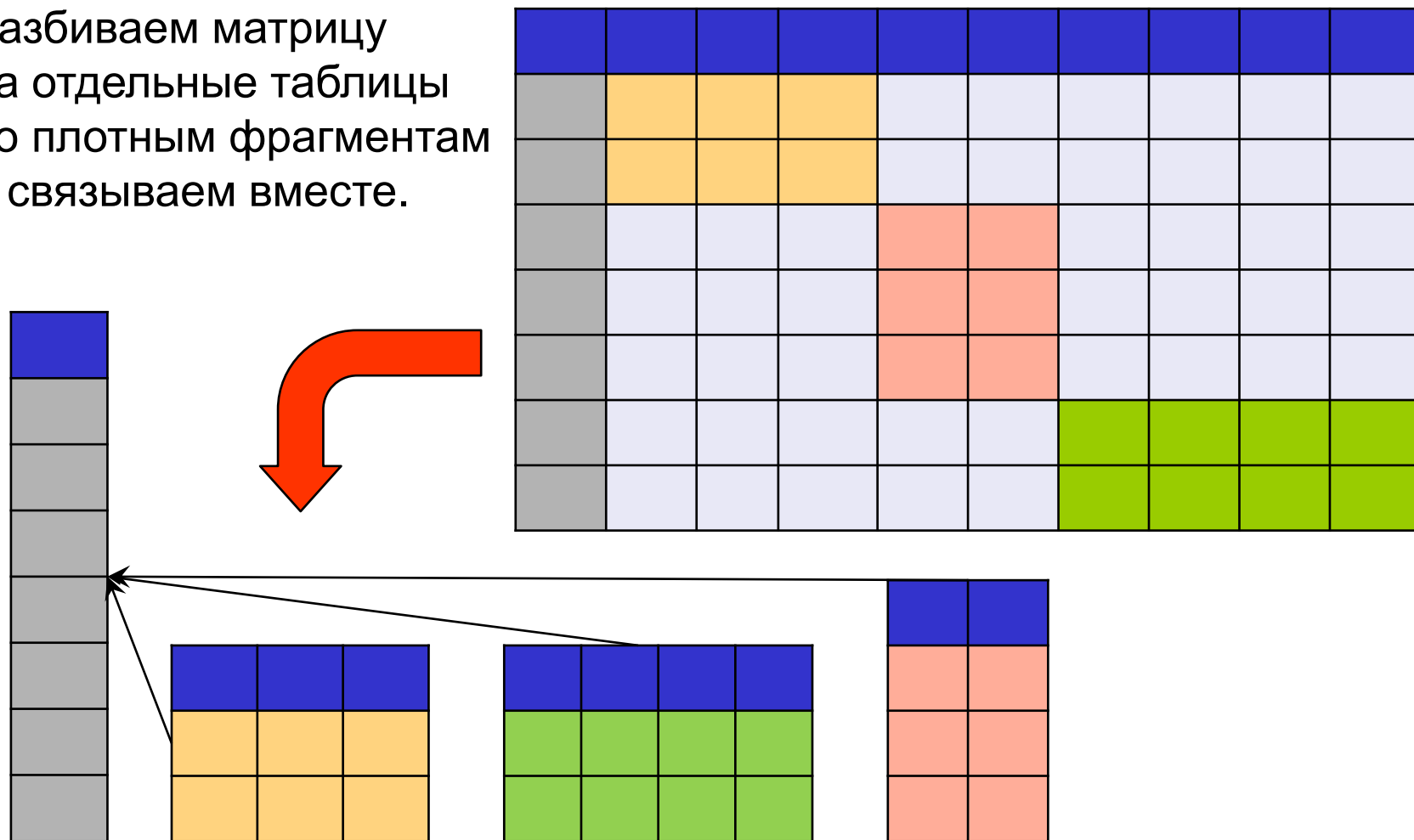
Универсальная модель данных с фиксированным набором таблиц для гибкого представления переменного (неопределённого) состава атрибутов - свойств одной сущности или для объединения сущностей, различающихся атрибутами - свойствами.

Модель оптимизирована для эффективного хранения разреженных матриц или набора неопределённых свойств или значений данных.

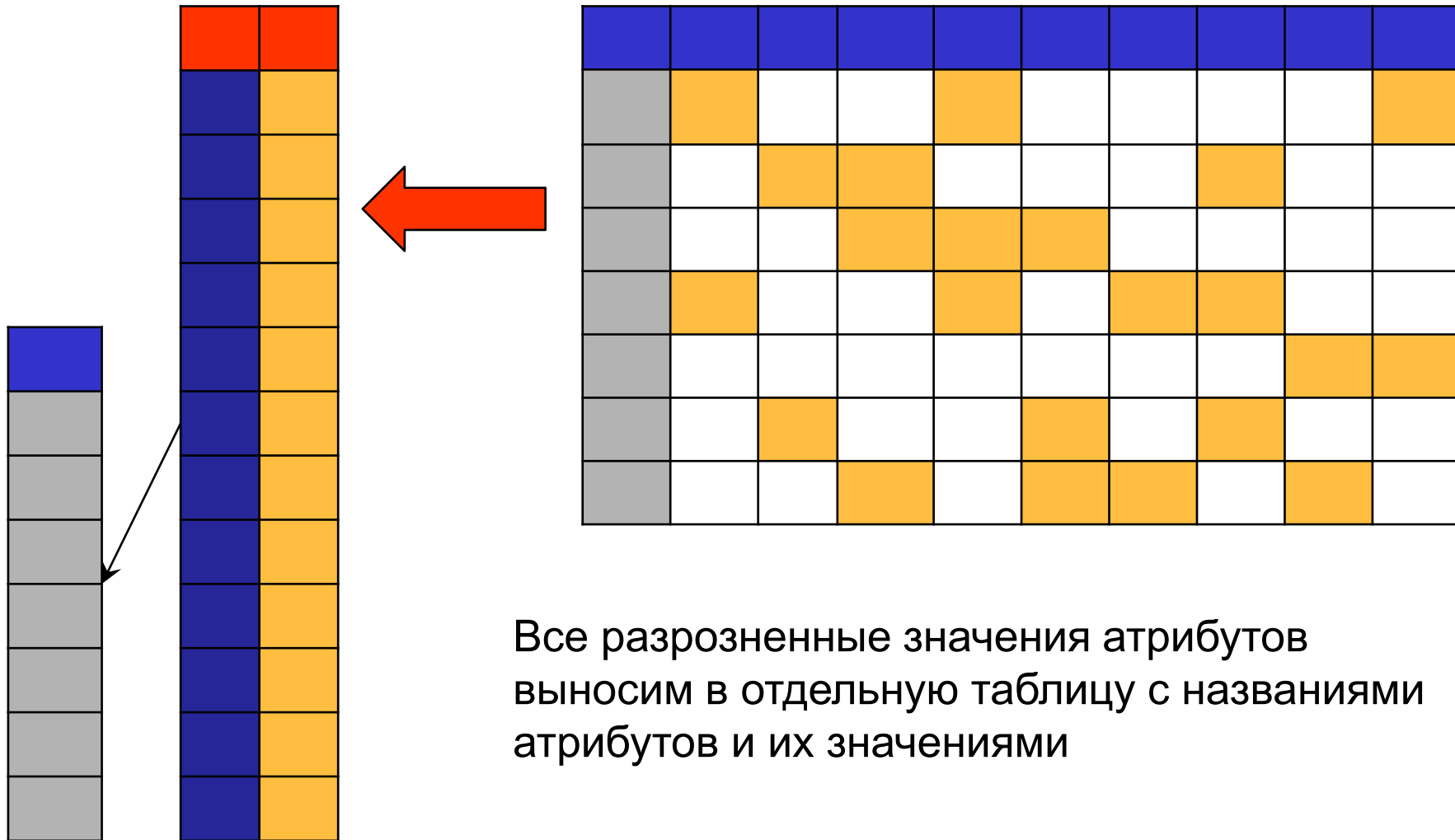
Другие названия *вертикальная модель данных* или *открытая схема*.

Разреженная таблица с плотными фрагментами

Разбиваем матрицу на отдельные таблицы по плотным фрагментам и связываем вместе.



Разреженная таблица с неплотными фрагментами



Идея метамоделирования таблицы

Широкая (горизонтальная) таблица «Простыня»

№	Имя	Фамилия	Пол	Дата рожд	Отрасль
1	Marinna	Kauschke	Female	1973-03-15	Financial Services
2	Olia	O' Mullan	Female	1973-03-24	Health
3	Brigitte	Whellams	Female	1973-05-09	n/a
4	Ivy	Farr	Female	1973-07-03	IT
5	Beverlee	Ungerechts	Female	1973-10-03	Manufacturing
6	Skipp	Swales	Male	1973-11-14	Entertainment
7	Leighton	Firbanks	Male	1973-12-22	Retail
8	Claudetta	Ricciardiello	Female	1974-04-30	Financial Services
9	Harland	Messenger	Male	1974-05-28	Retail

Узкая (вертикальная) таблица «Портянка»

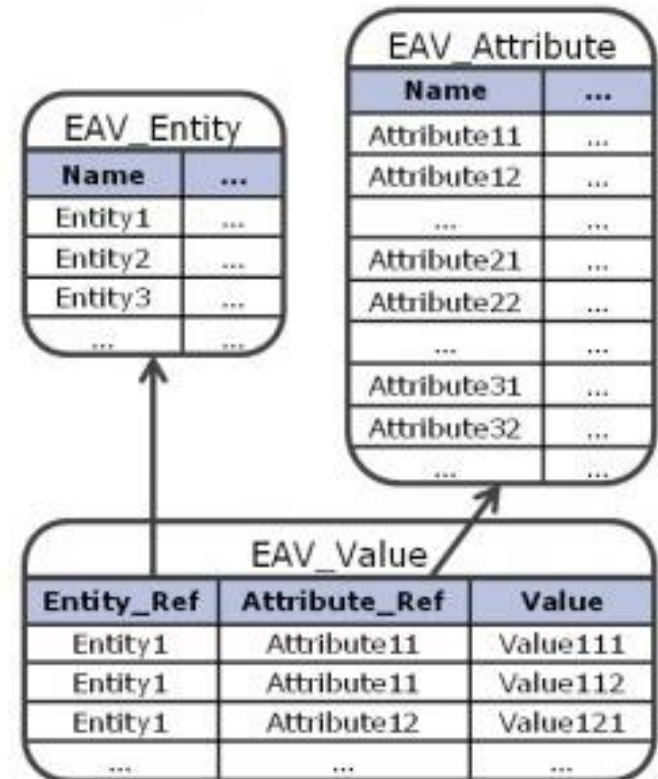
№	Имя атрибута	Значение
1	Имя	Marinna
1	Фамилия	Kauschke
1	Пол	Female
1	Дата рожд	1973-03-15
1	Отрасль	Financial Services
2	Имя	Olia
2	Фамилия	O' Mullan
2	Пол	Female
2	Дата рожд	1973-03-24
2	Отрасль	Health
3	Имя	Brigitte
3	Фамилия	Whellams
3	Пол	Female
3	Дата рожд	1973-05-09
4	Имя	Ivy
4	Фамилия	Farr
4	Пол	Female
4	Дата рожд	1973-07-03
4	Отрасль	IT
...

Объединение нескольких сущностей в модели EAV

Entity1		
Attribute11	Attribute12	...
Value111	Value121	...
Value112	Value122	...
...

Entity2		
Attribute21	Attribute22	...
Value211	Value221	...
Value212	Value222	...
...

Entity3		
Attribute31	Attribute32	...
Value311	Value321	...
Value312	Value322	...
...

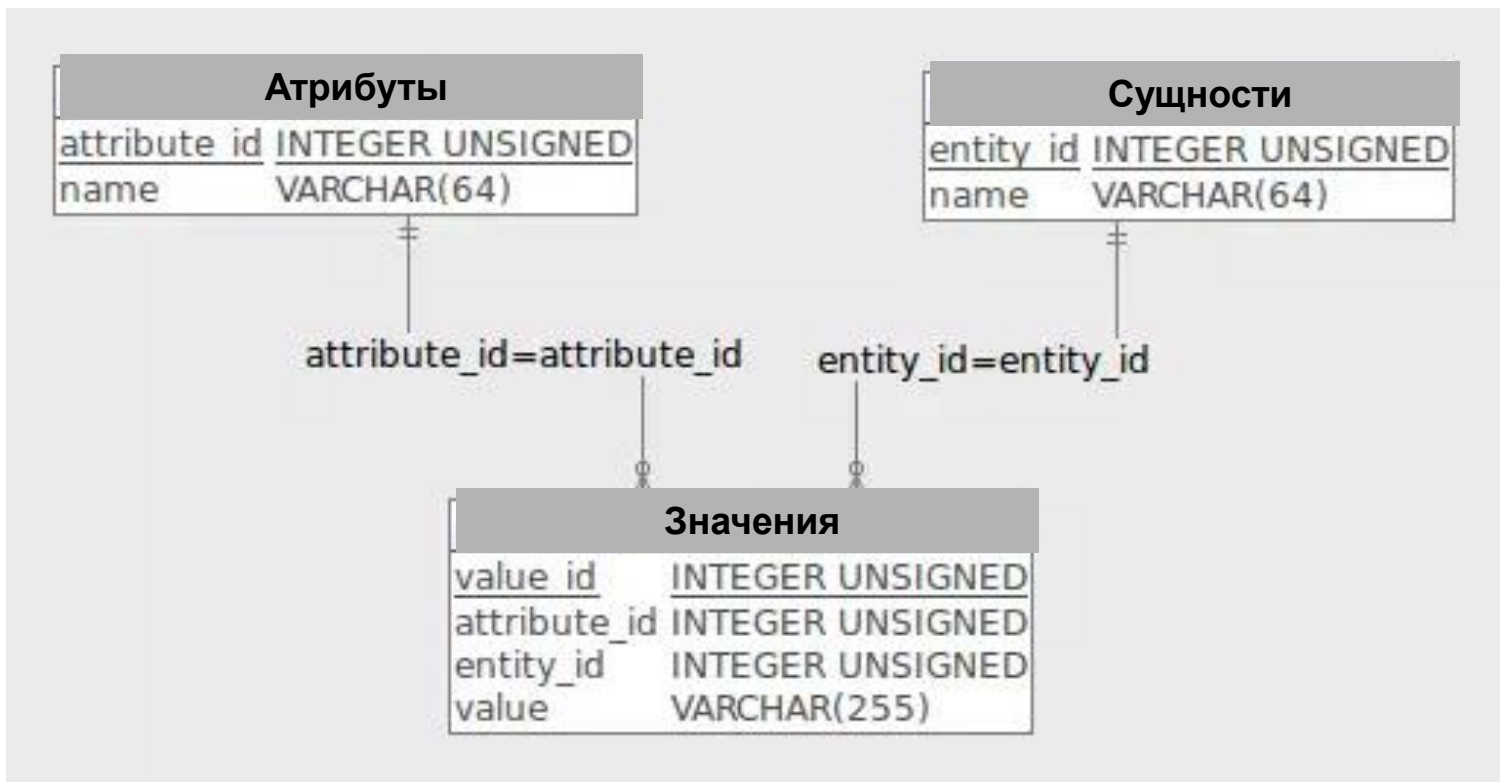


Варианты реализации модели EAV

1. Типовая реализация с 3-мя таблицами
2. Простая реализация с 2-мя таблицами
3. Реализация с раздельными типами значений
4. Реализация с отражением экземпляров сущностей
5. Гибридная реализация с выделением фиксированных и переменных атрибутов

Типовая реализация модели EAV

Денормализованная модель данных с 3-мя таблицами. В таблице **Значения** заведены ссылки на справочники **Атрибуты** и **Сущности**, а также значение атрибута. Т.е. имеем атрибут с наложенными значениями строкового типа.

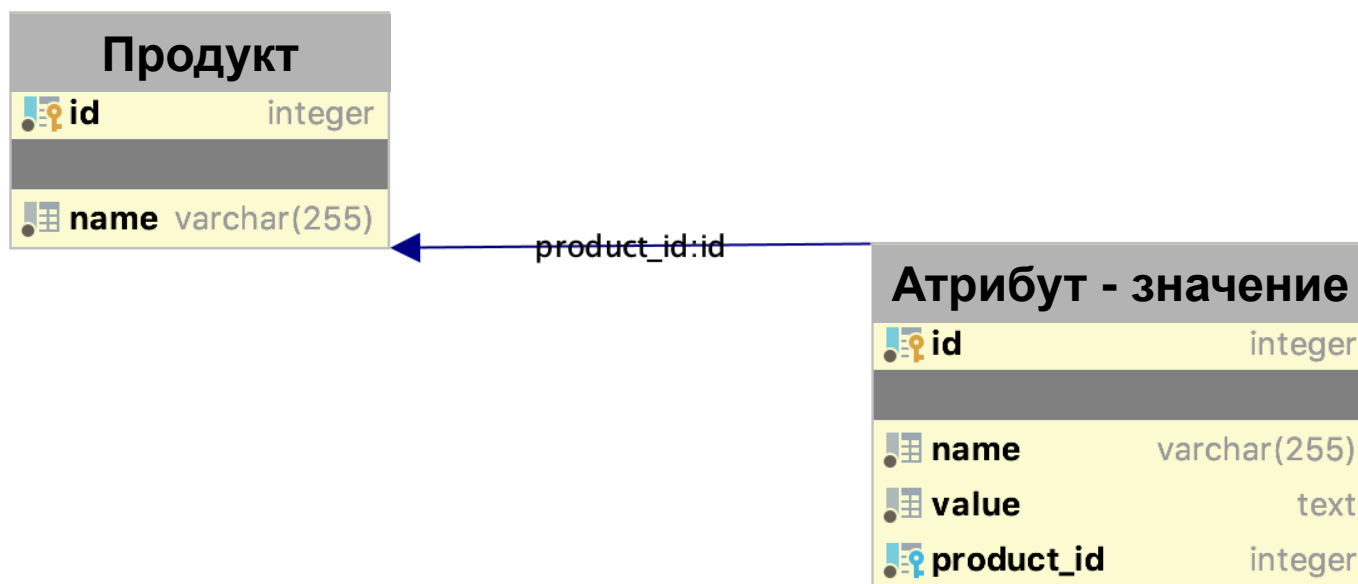


Простая реализация модели EAV

Пример каталога товаров

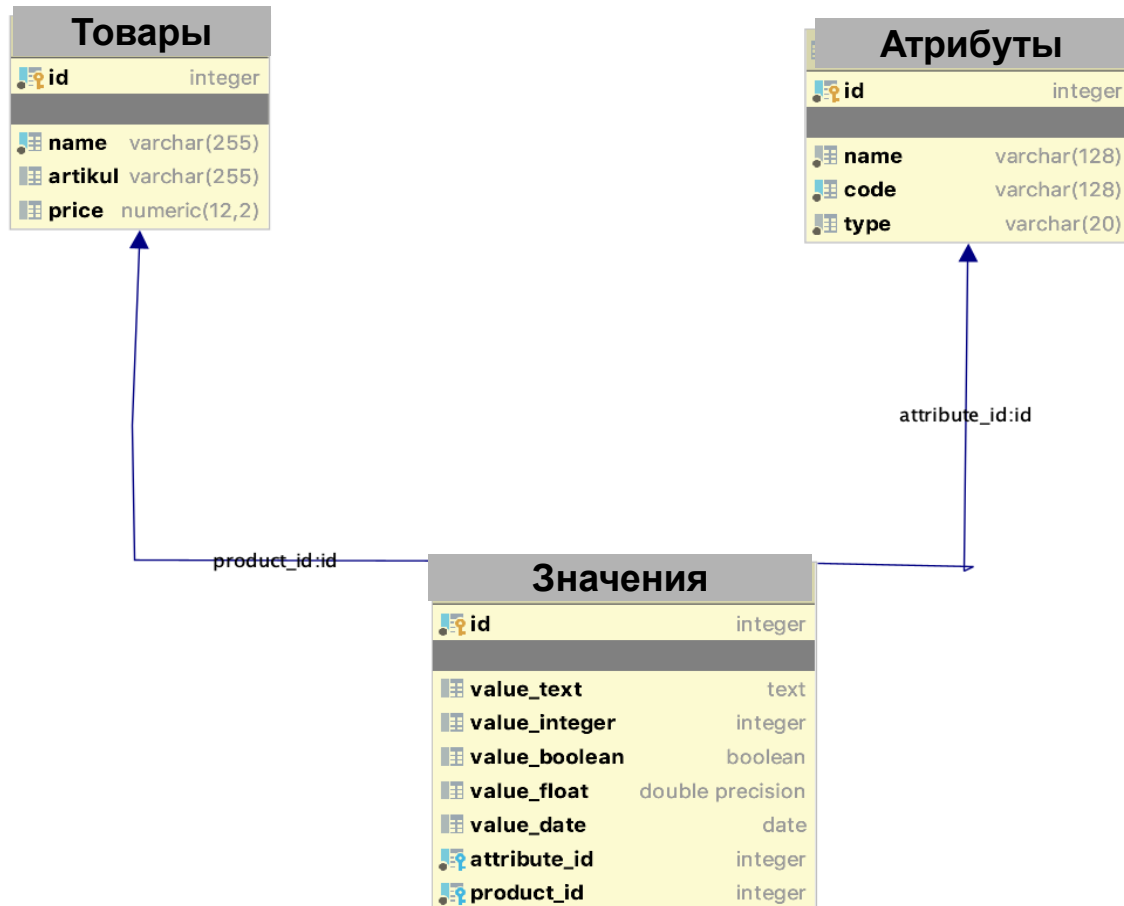
Денормализованная модель с 2-мя таблицами. В таблице **Атрибут-значение** приведены имена атрибутов, их значения и ссылка на справочник **Продуктов**.

Атрибуты всех типов данных представлены в виде строки текста.



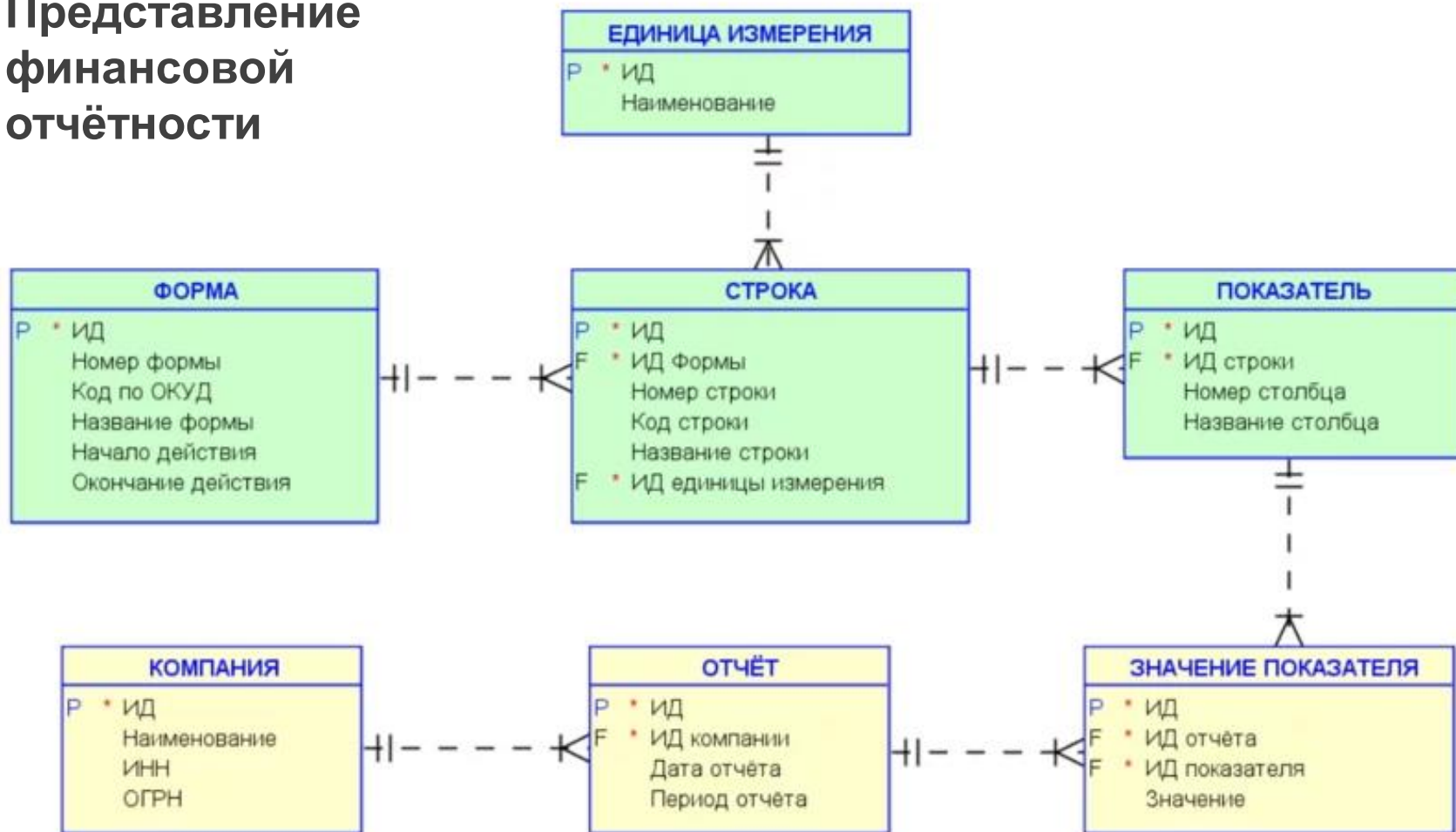
Реализация модели EAV с отдельными типами значений

Пример магазина с широкой номенклатурой товаров



Гибридная реализация модели EAV

Представление финансовой отчётности



Достоинства и недостатки модели EAV

Абстрактное представление сущности с неограниченным набором атрибутов для задания простых или составных значений различных типов данных.

Гибкость добавления новых понятий: не требуется изменение состава и структуры таблиц БД.

Автоматизированное создание и изменение атрибутов: возможно на любом этапе жизненного цикла системы.

Генерация форм ввода данных по описанию атрибутов.

Нельзя использовать ограничения целостности СУБД, надо вводить данные контролировать в клиентской программе.

Сложность обработки данных из-за трудности индексации и невозможности использовать имеющиеся справочники.

Падение производительности на порядок из-за усложнения запросов, чтения сотен тысяч записей, отсутствия оптимизации для такой модели данных и возможной конвертации данных.

Альтернативы EAV

Встроенные в БД сложные типы данных:

- XML
- JSON / JSONB

NOSQL базы данных типа key-value:

- Redis
- Memcached
- Riak
- Apache Cassandra

NOSQL базы данных типа column-family based:

- Apache Cassandra
- Apache HBase

Пример атрибута типа XML

Описания участников создания книги:

Books

(PK) book_ID SMALLINT

...

creators XML

...

Пример значения атрибута:

```
'<creators editor="Петров А.В."  
authors="Шкатов П.Н. Черненко В.М."  
artist="Хомич И.С."  
translator="Шварц Э.Г." />'
```

Пример атрибута типа JSON

Описания участников создания книги:

Books

(PK) book_ID SMALLINT

...

creators JSON

...

Пример значения атрибута:

```
{ "редактор": "Петров А.В.",  
  "авторы": ["Иванов С.П.", "Семёнов В.К."],  
  "художник": "Филин Р.Б.",  
  "переводчик": "Шварц Э.Г." }
```

**Терпения и удачи всем, кто связан
с моделированием данных**

Спасибо за внимание!

Валерий Иванович Артемьев

МГТУ имени Н.Э. Баумана, кафедра ИУ-5

Банк России

Департамент данных, проектов и процессов

Тел.: +7(495) 753-96-25

e-mail: viart@bmstu.ru