СЛЁРМ Обучение для инженеров и технических лидеров

Дата-инженер

Традиционная архитектура хранилищ данных

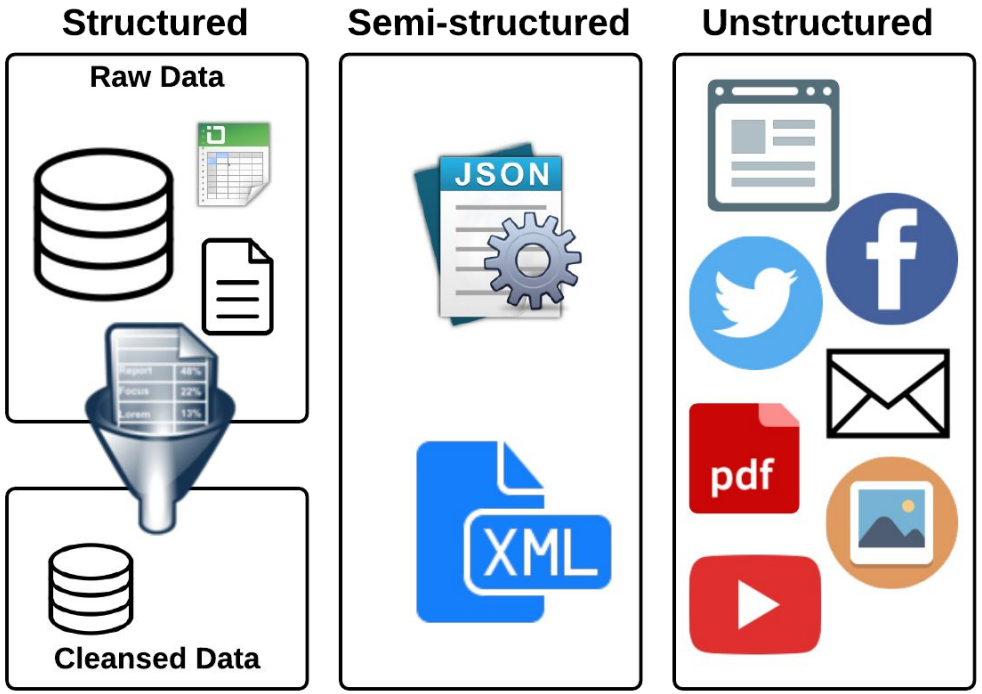
Николай Марков

Цели урока. Что вы узнаете:

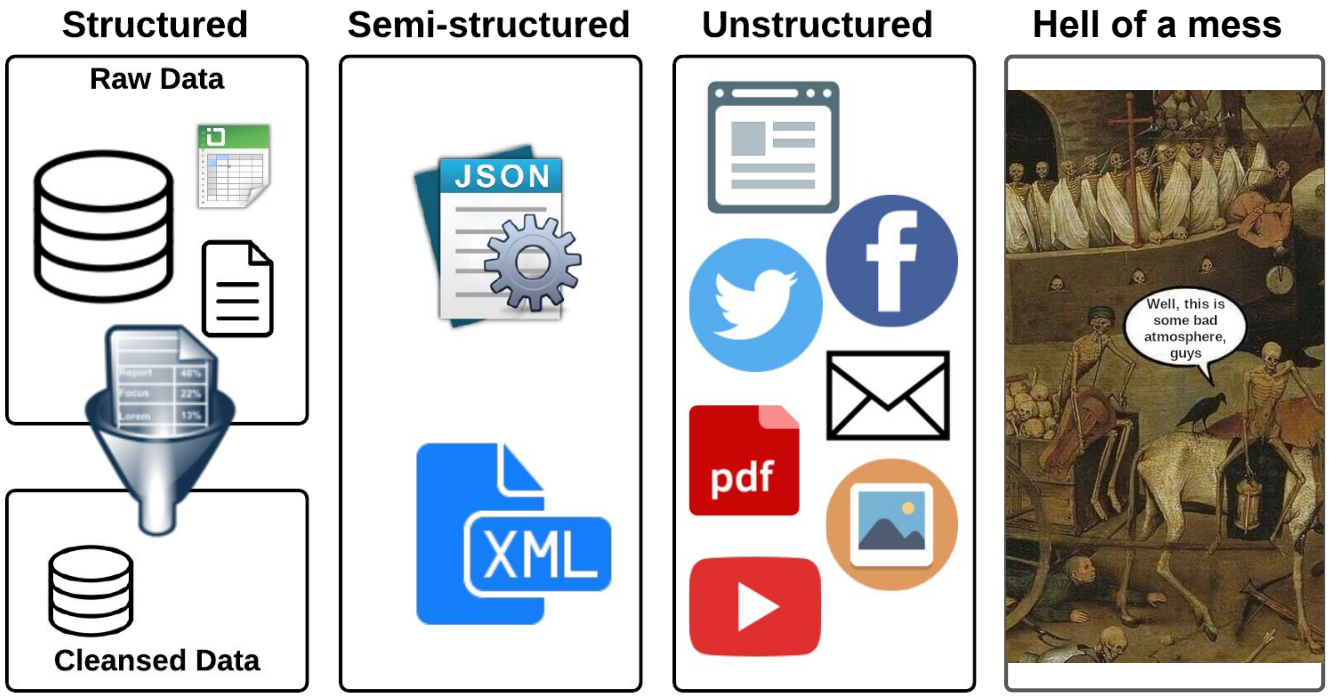
1. Как компании работают с данными
2. Как устроены Data Lake и Data Warehouse
3. Классические паттерны построения систем аналитики
4. Введение в методологии проектирования DWH

Характеристики данных

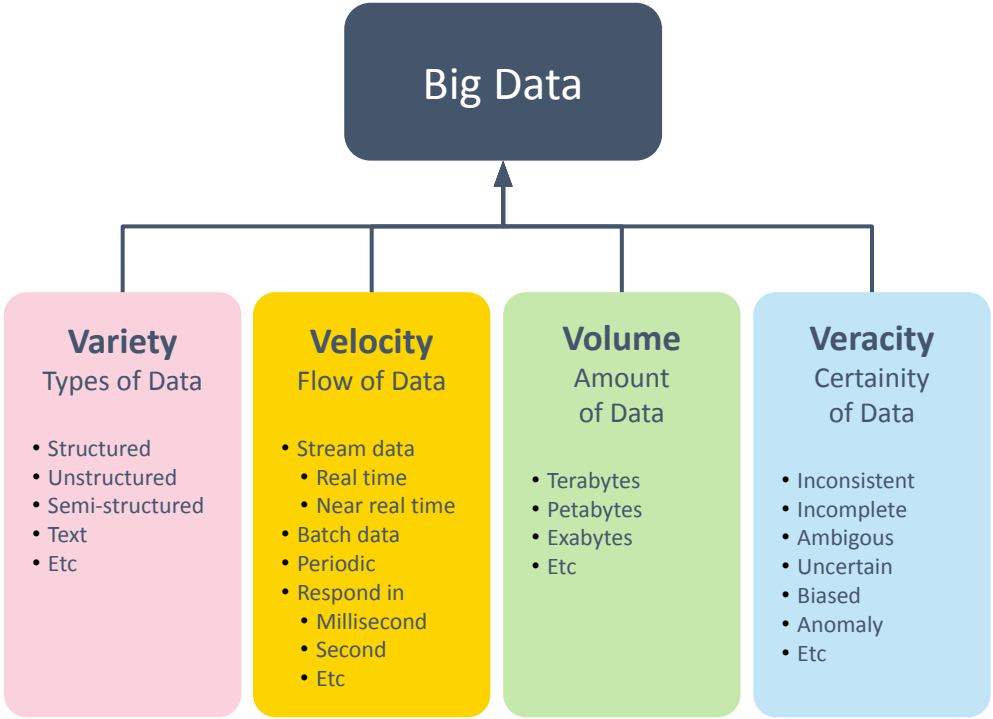
Типы данных в организации

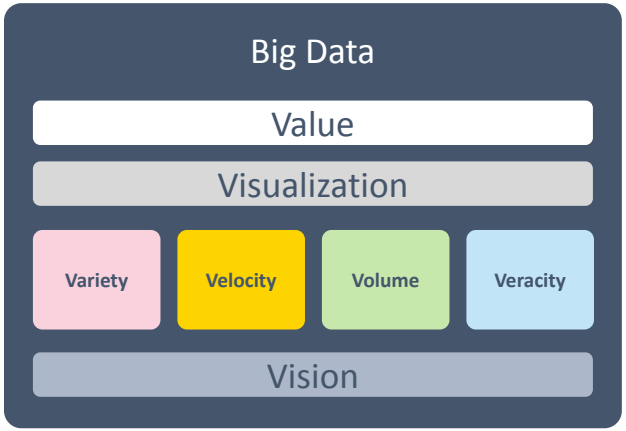


Типы данных в организации



Много букв V

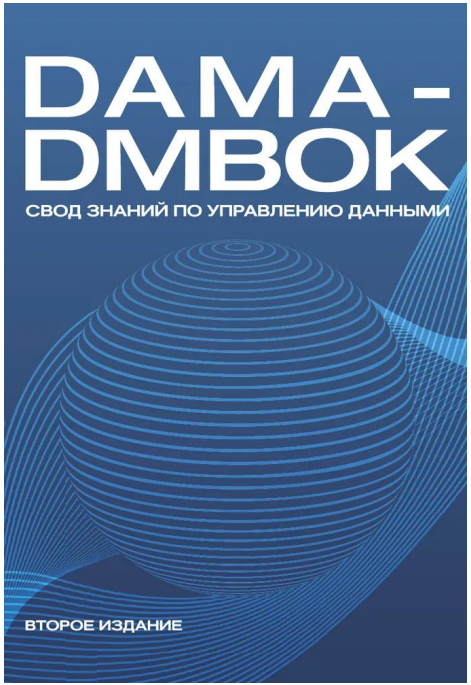


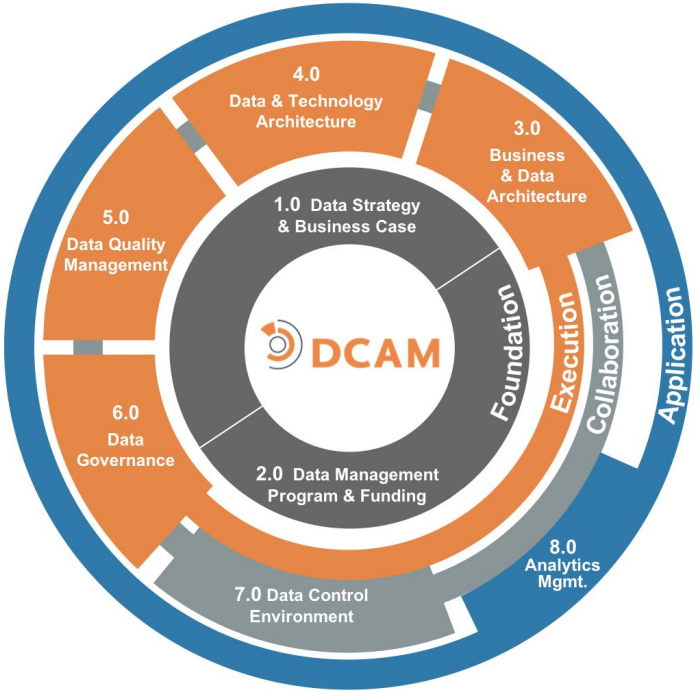


Master Data Management

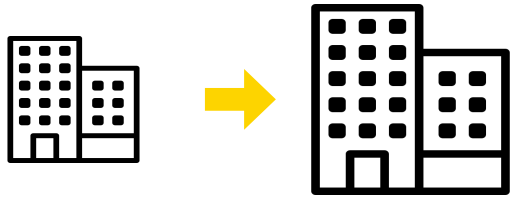
- Бизнеса имеет **доменную область**, в которой он оперирует
- В ней есть определенные **бизнес-объекты**, представление о которых необходимо иметь, чтобы **принимать решения**
- Подобными объектами могут быть **Клиент**, **Дилер**, **Локация**, **Маркетинговая Кампания**
- Готовую коллекцию объектов называют **Golden Record**, типичный пример — **360-градусный обзор клиентов**
- **MDM** — название **процессов по формированию** этих объектов **на основе собираемых данных**

DMBOK и DCAM

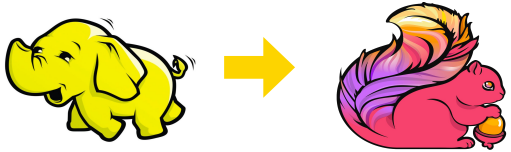




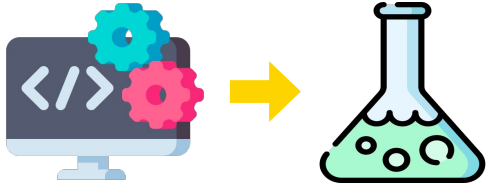
Выбор архитектуры зависит от компании



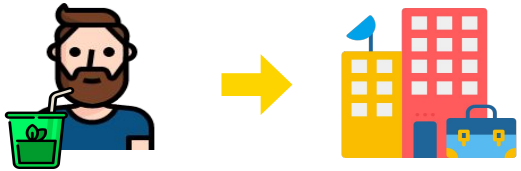
Маленькая или большая?



Legacy или свежак?



Продуктовая или R&D?



Смузи в коворкинге
или кровавый энтерпрайз?

Data Lake
и Data Warehouse

Что такое Data Lake

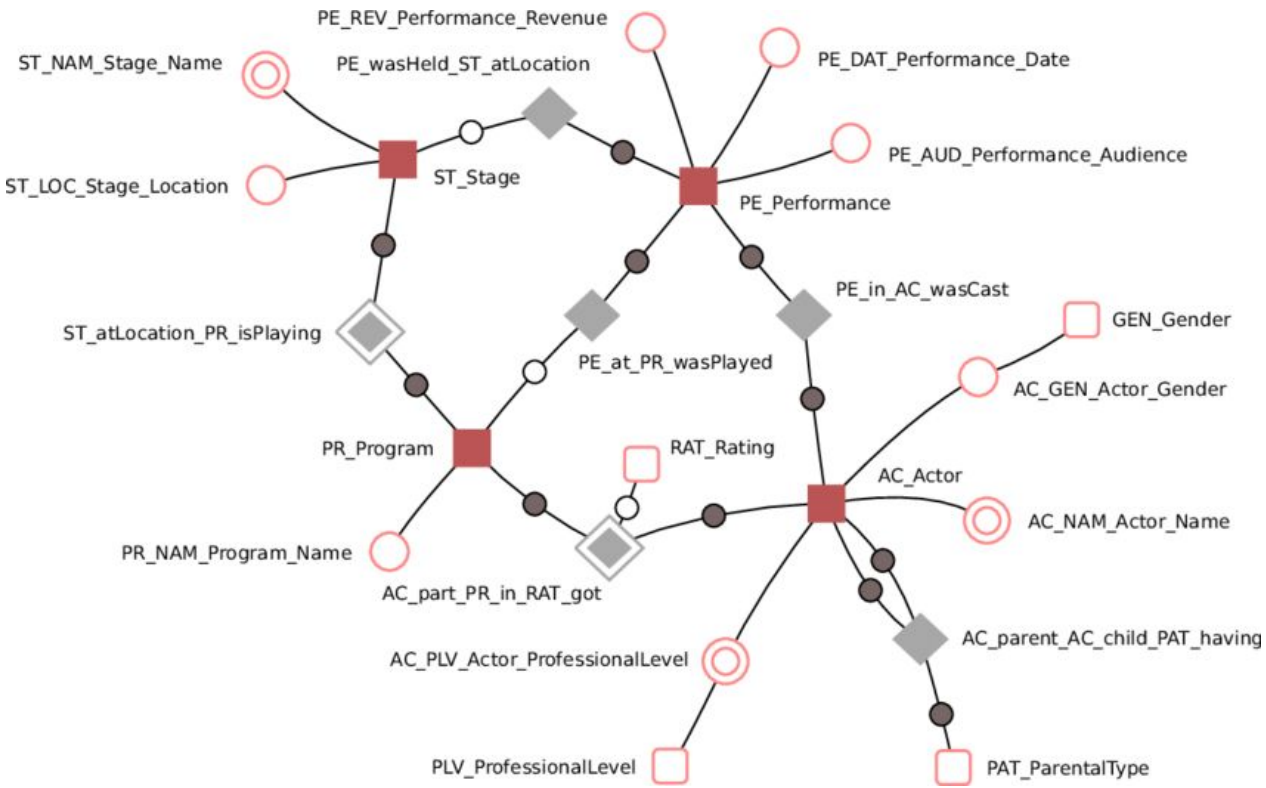
- Все данные компании **в сыром виде** агрегируются в **одно общее место**
- В некоторых компаниях в качестве Data Lake используют структурированные БД, но это скорее исключение
- Обычно под Data Lake понимается **объектное хранилище**, которое хранит данные просто в виде **бинарных блобов**
- С помощью использования специализированных форматов (**Parquet**, **ORC** etc.) можно получить **ускорение запросов** поверх данных в озере

Варианты решений

- Обычно хорошо интегрируется с существующими инструментами Big Data
- Проверенное временем решение
- Может быть не очень простым в поддержке

- Полностью совместим с API S3
- Довольно прост в поддержке и устройстве
- Относительно новый проект, иногда с мелкими проблемами

Структурируем в Data Warehouse



Структурируем в Data Warehouse

Exasol

GREENPLUM DATABASE®

SingleStore

VERTICA

Облачные решения
обсудим чуть дальше

Денормализованные витрины

ClickHouse

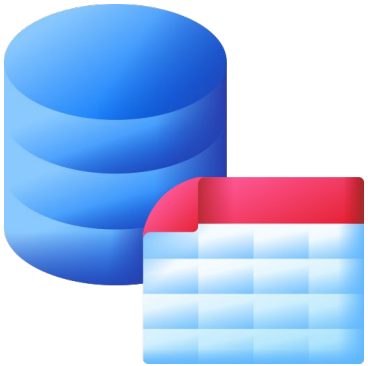
mongoDB.

PostgreSQL

druid

Итоговый выбор варианта

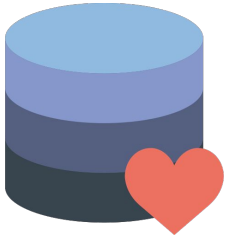
- Много разных источников
- Кросс-доменная аналитика
- Сложные структурные запросы
- Долгое хранение



- Достаточно простые данные на входе (например, clickstream/event sourcing)
- Статистический анализ с небольшой задержкой
- Нет нужды в сложных JOIN’ах
- Быстро устаревающие данные

Lambda и Kappa

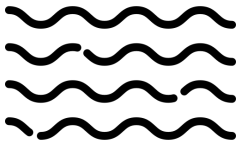
Баланс сложности и простоты



result = query(all data)

Batching

Девиз: «Делаем сложные вещи медленно»

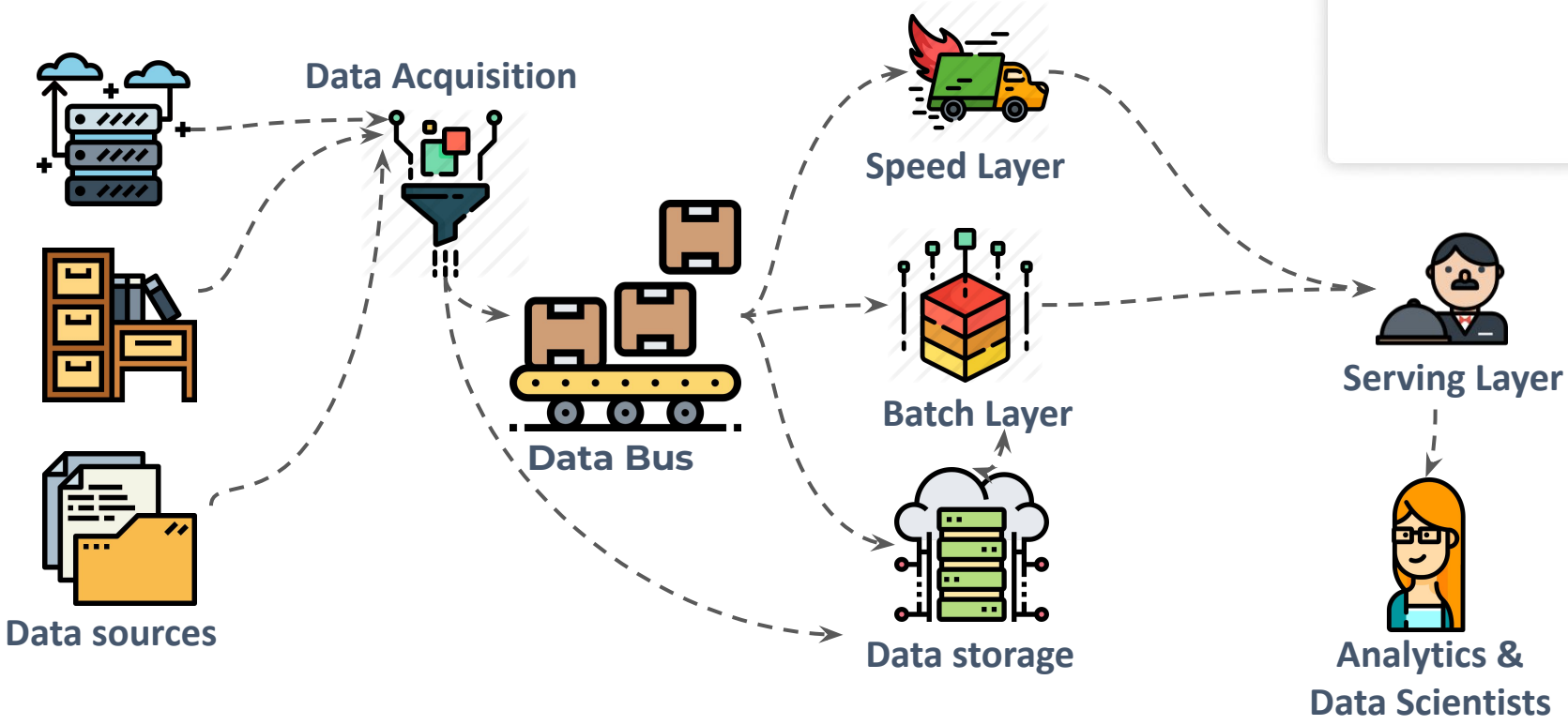


result = aggregation(small chunk of data)

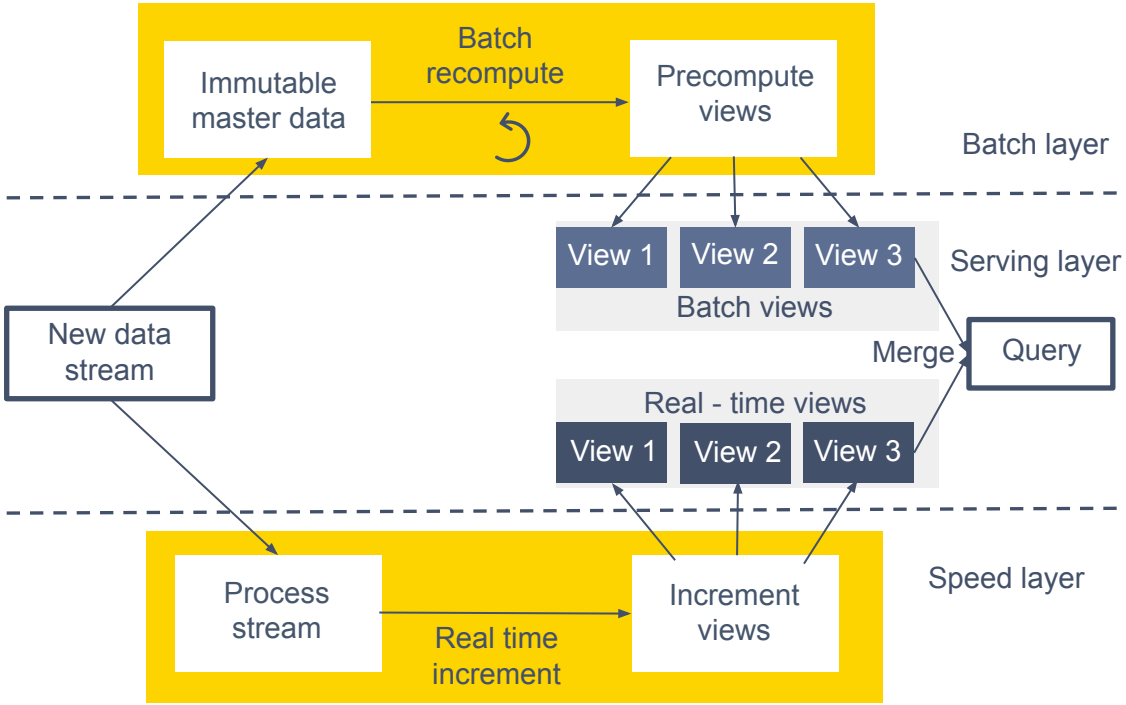
Streaming

Девиз: «Делаем простые вещи быстро»

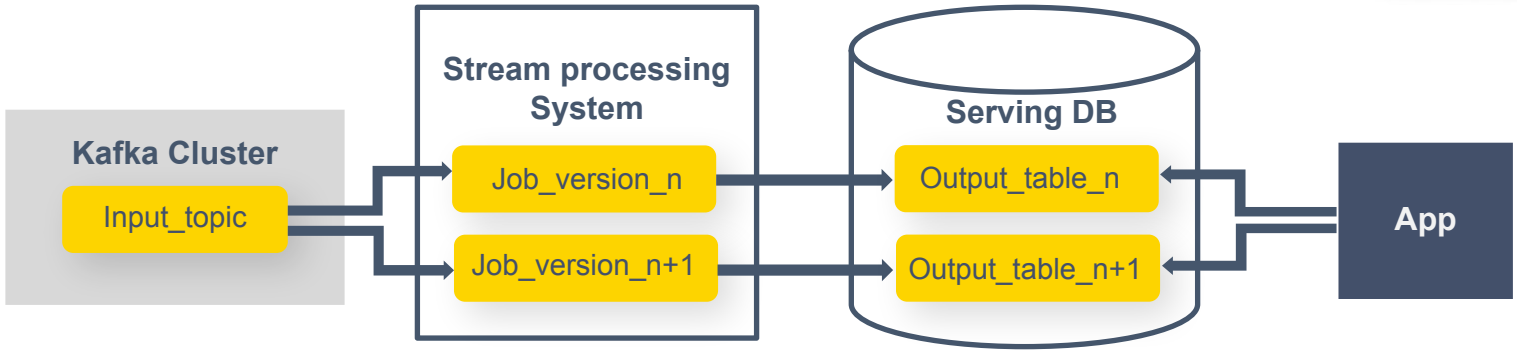
Общая схема



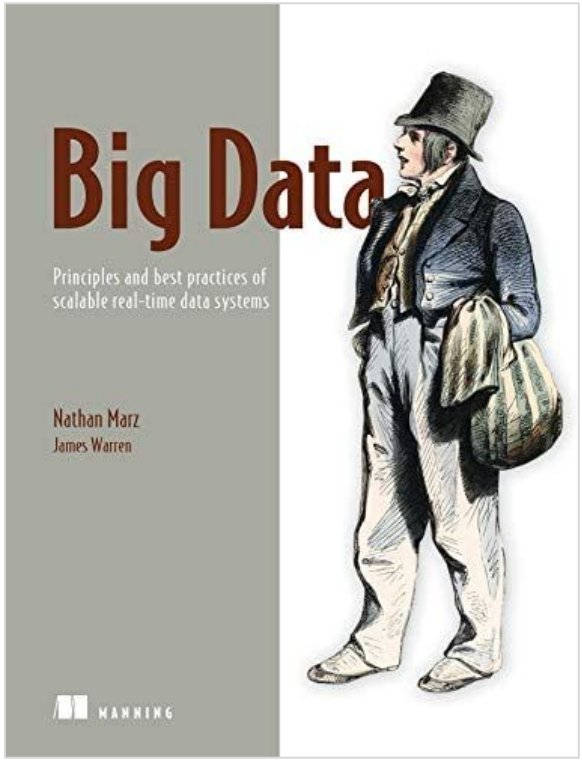
Lambda-архитектура

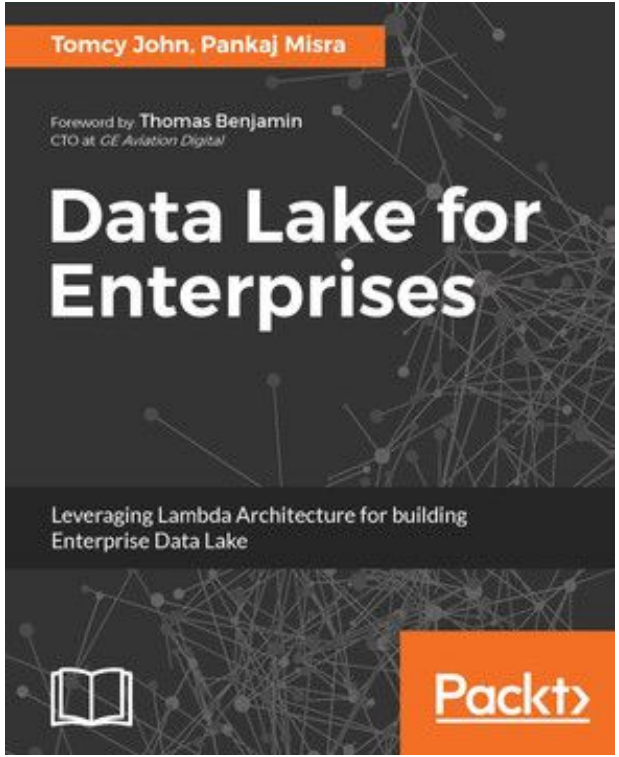


Kappa-архитектура

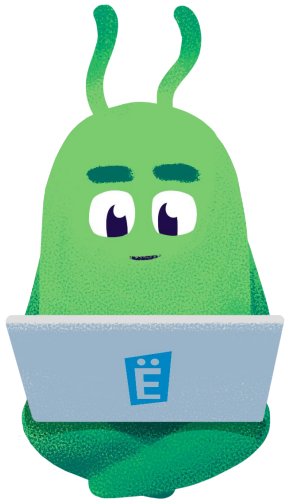


Почитать



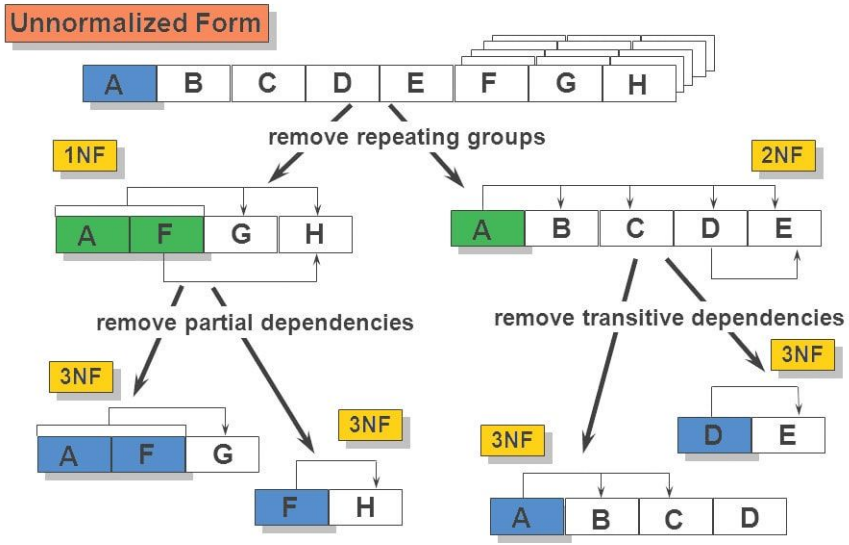


Методологии хранения данных

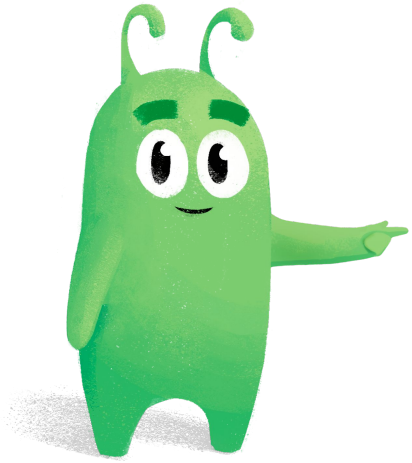


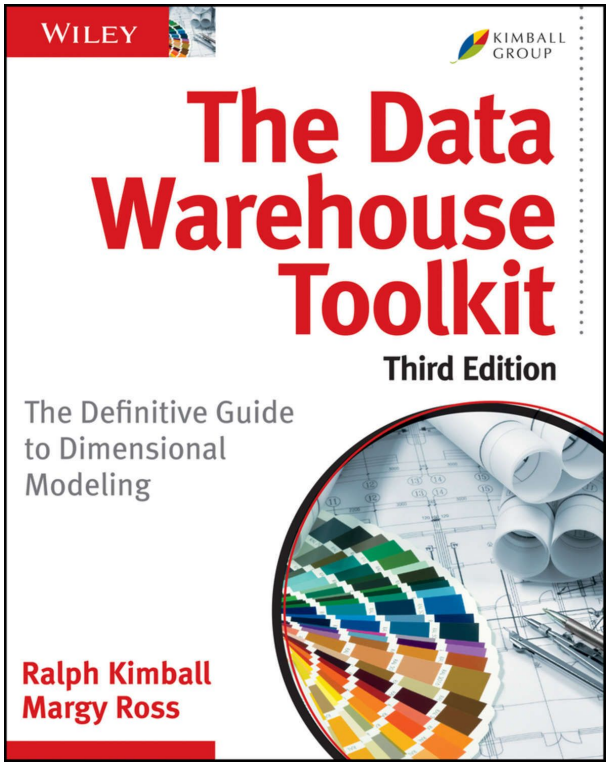
Базовые проблемы хранилищ

- До какой степени нарезать на кусочки (нормализовать)?
- Как поддерживать историчность данных?
- Как мигрировать и версионировать схему данных?
- В каком формате хранить данные на диске?
- Нужно ли поддерживать SQL?

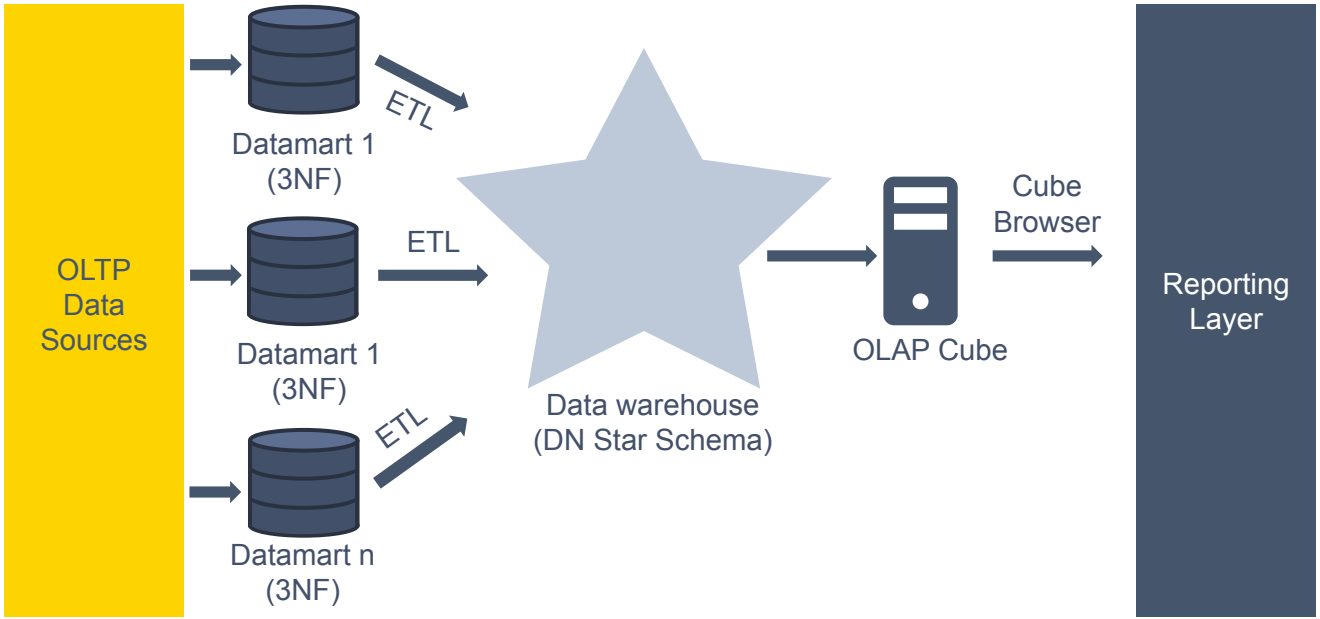


Бессмертная классика

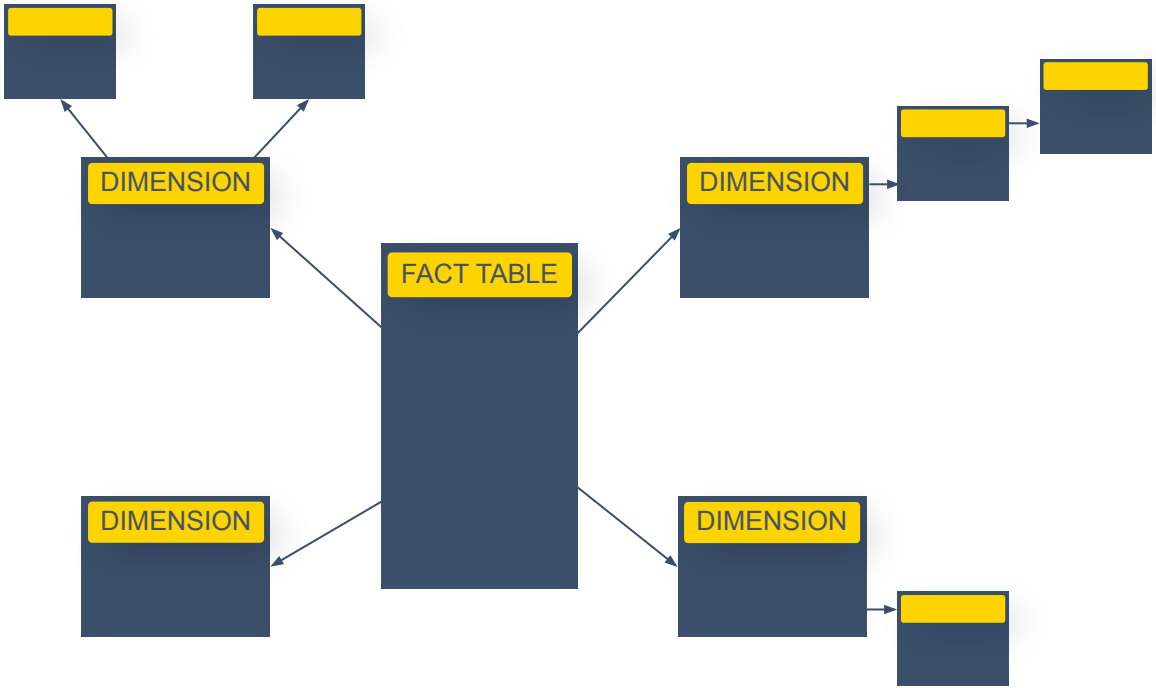




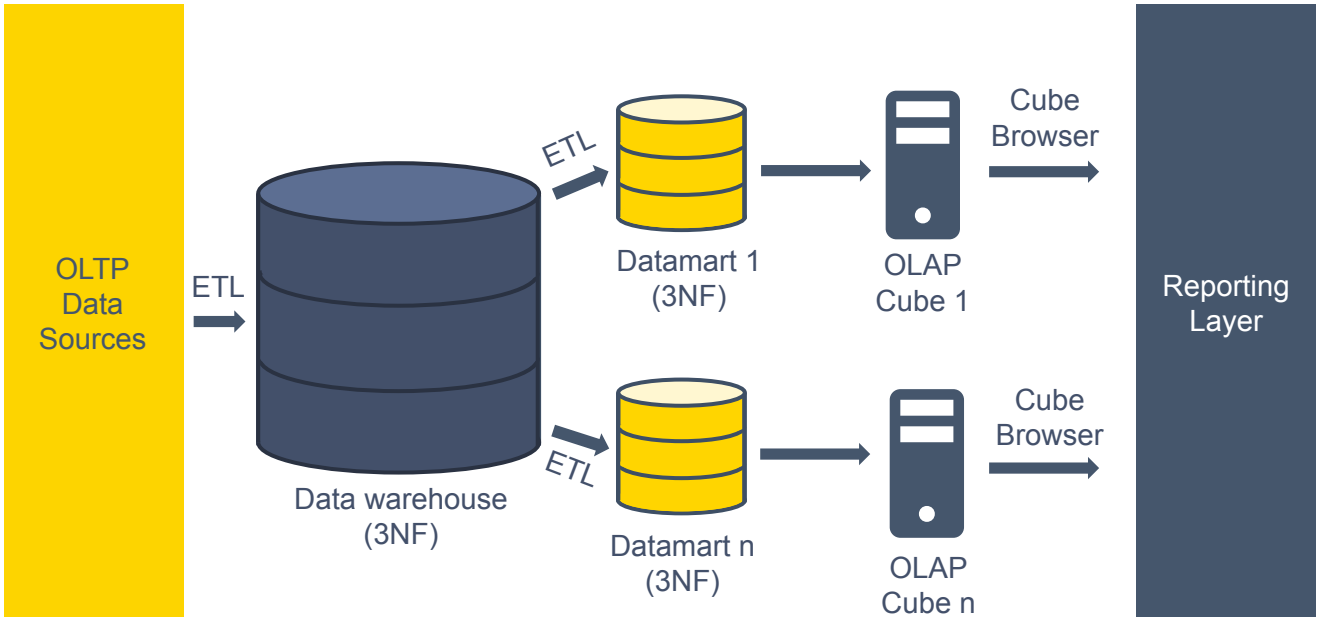
Подход Кимбалла



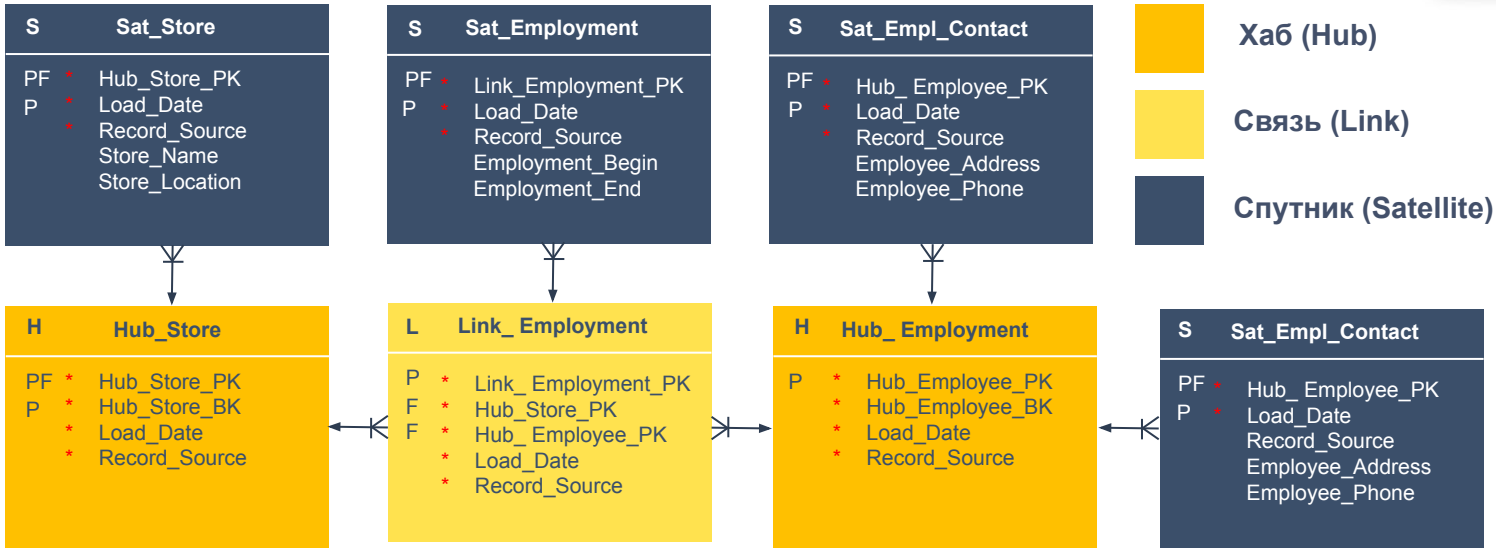
Звезда и снежинка



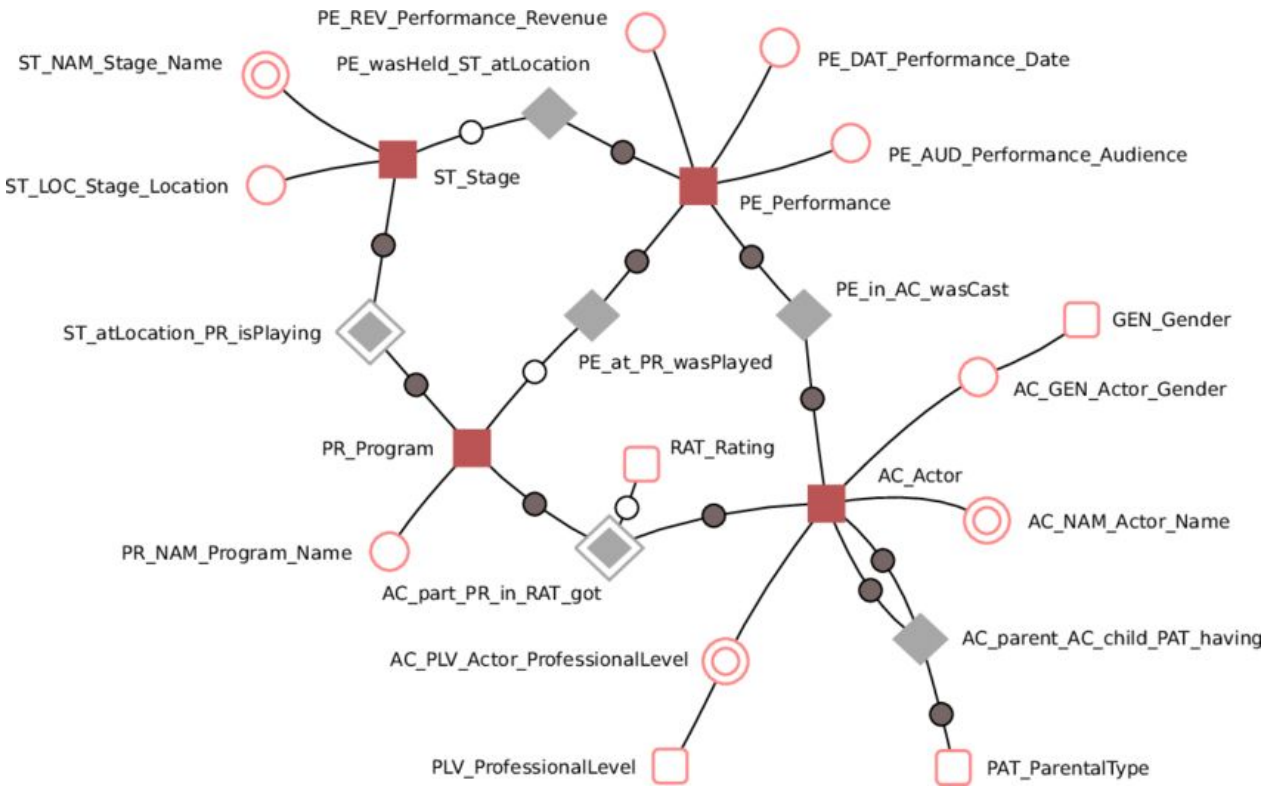
Подход Инмона



Data Vault 1.0/2.0



Anchor Modeling



Итоги. О чем поговорили:

1. Характеристики корпоративных данных влияют на то, как их хранить и как с ними работать
2. Data Lake хранит слабоструктурированные данные, DWH — структурированные
3. Lambda и Kappa — классические подходы
4. Система должна поддерживать эволюцию, в идеале без миграций

Спасибо
за внимание!