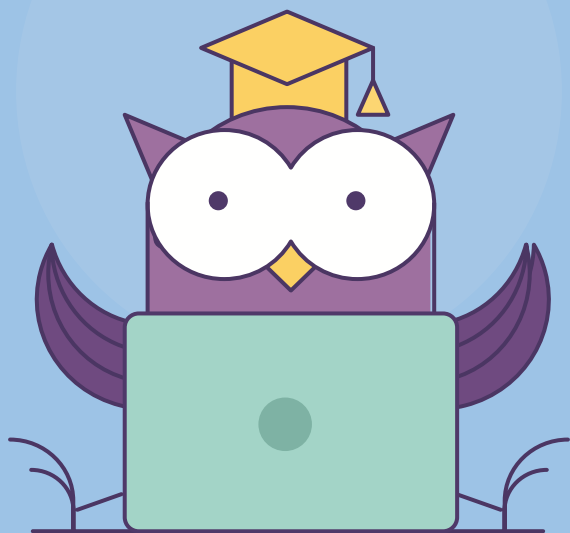


Инженер Данных. Хранилища данных - 2.



Меня хорошо слышно && видно?



Напишите в чат, если есть проблемы!

Ставьте + если все хорошо
Ставьте - если есть проблемы

План занятия

- Лучшие практики MPP DWH
- Подходы к моделированию хранилища
 - Bill Inmon
 - Ralph Kimball
 - Data Vault
 - Anchor Modeling
- Демо

Лучшие практики MPP

Лучшие практики MPP

- Параллельная обработка
- Оптимальное хранение данных (согласно сценариям использования)
- Резервирование и репликация данных
- Мониторинг производительности и своевременное устранение проблем
- Управление ресурсами и группами пользователей

Сегментация. Зачем?

- Равномерное распределение
- Локальность операций с данными на узлах



Перекосы в данных

- Пример: Самый крупный клиент банка
- Пример: Одна техническая УЗ на весь front-office



Как распределить строки?

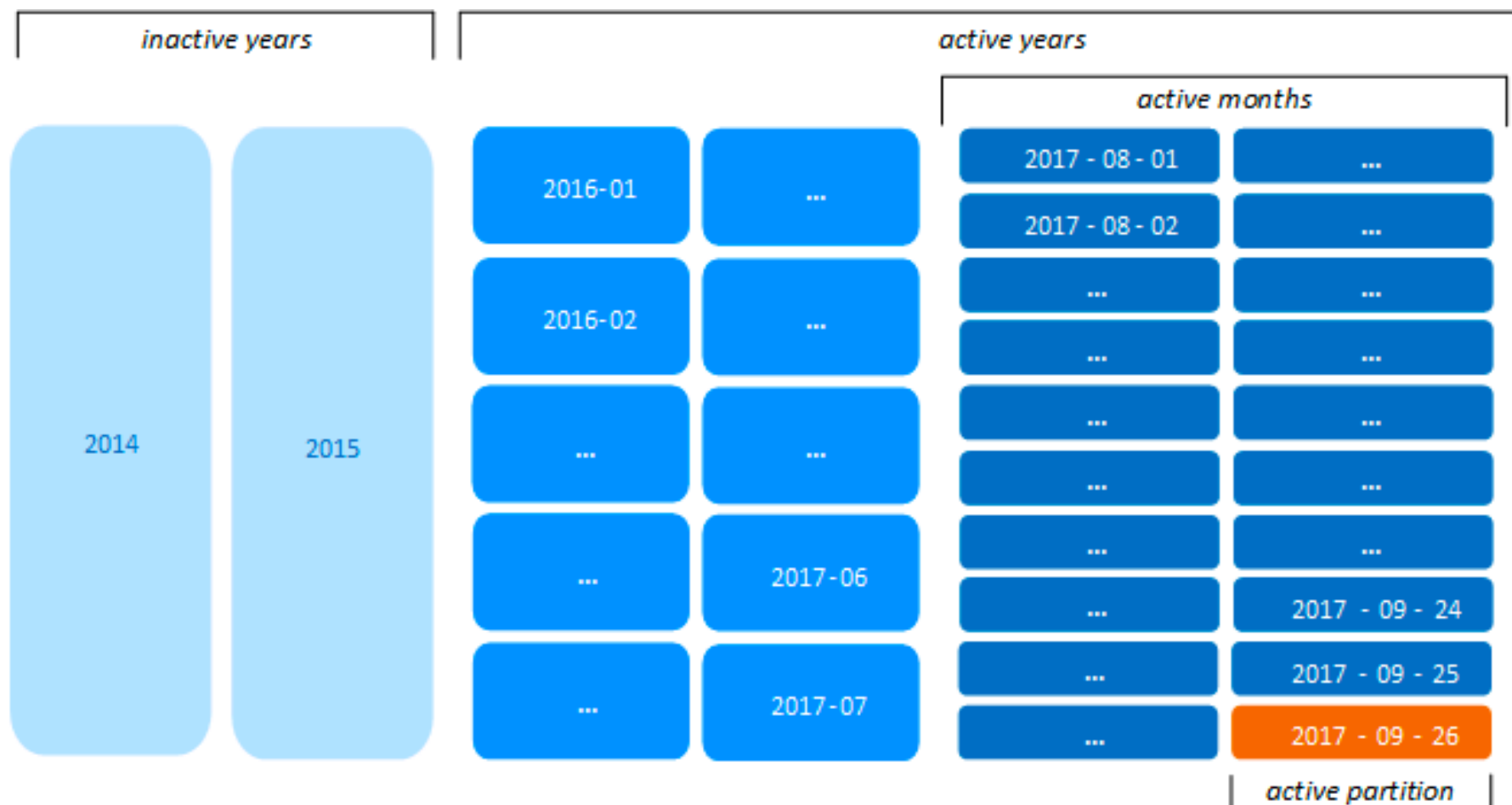
NPS:

- Результаты (ответы)
- Клиенты
- Кампании
- Волны кампаний (группы)
- Продукт

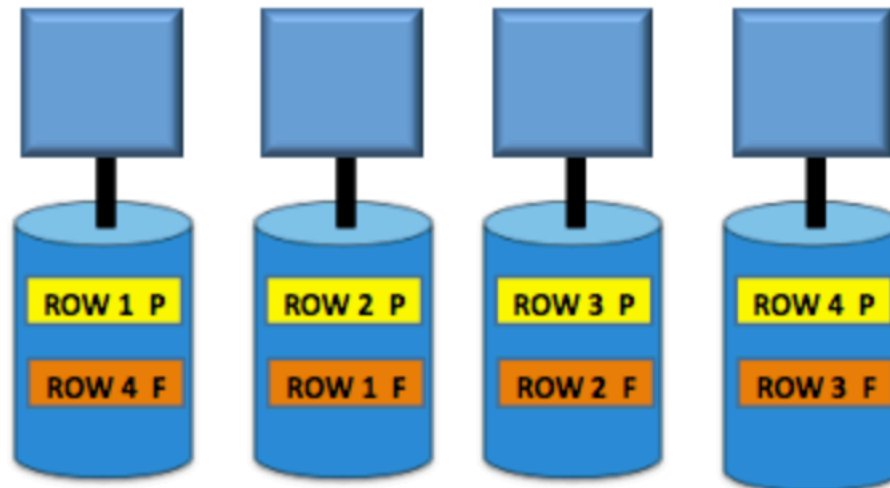
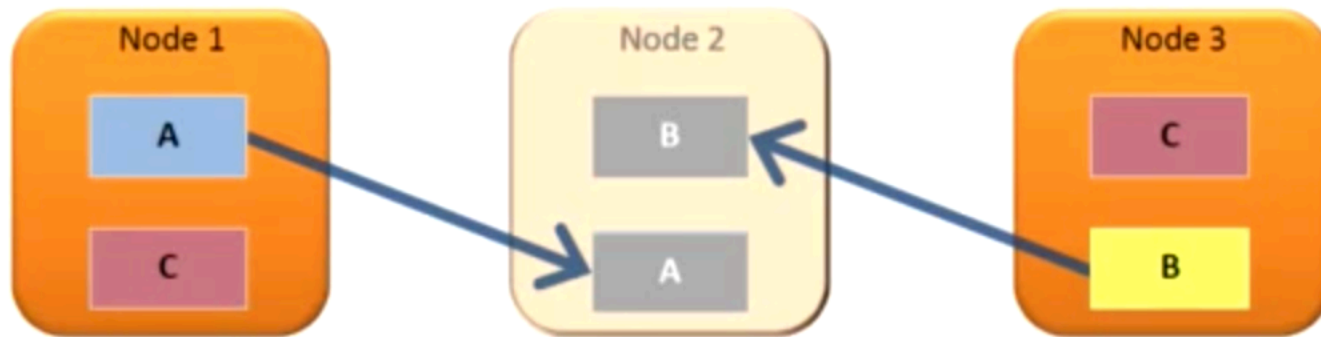
Банковские транзакции:

- Транзакции - 500КК строк
- Карты – 150К строк
- Клиенты – 80К строк
- Счета – 50К строк
- Продукты – 5К строк

Партиционирование данных



Резервирование данных



Физическое хранение данных

<pre>CREATE PROJECTION store_orders_p1 (product_number encoding rle, store_number encoding block_dict, vendor_number encoding rle, order_number encoding deltaval, date_ordered encoding rle) AS</pre>	Column list and encodings
<pre>SELECT product_number, store_number, vendor_number, order_number, date_ordered FROM store_orders</pre>	Base query
<pre>ORDER BY date_ordered, vendor_number, order_number</pre>	Sort order
<pre>SEGMENTED BY hash (store_number)</pre>	Segmentation clause
<pre>ALL NODES KSAFE;</pre>	K-safety

Исключить OLTP-нагрузку

- Минимум UPDATE / DELETE
- Delete Vectors
- Read-optimized + Write-optimized

План запроса

- **Cost-based optimizer**
- Access path: Проекции, Индексы
- Join Method: Merge, Hash, NL, Product
- Join Geography: local, redistribution, duplication
- Join Order: Последовательность таблиц

План запроса

Access Path:

```
+-GROUPBY HASH (SORT OUTPUT) (GLOBAL RESEGMENT GROUPS) (LOCAL RESEGMENT GROUPS) [Cost: 3M, Rows: 15M] (PATH ID: 2)
```

```
| Group By: s.product_key, p.product_description
| Execute on: All Nodes
| +---> JOIN HASH [Semi] [Cost: 1M, Rows: 15M] (PATH ID: 3) Inner (BROADCAST)
| |      Join Cond: (s.store_key = subQ_1.store_key)
```

Operator

```
| |      Materialize at Input: s.store_key
```

```
| |      Materialize at Output: s.product_key
```

```
| |      Execute on: All Nodes
```

```
| | +-- Outer -> JOIN HASH [Cost: 906K, Rows: 30M] (PATH ID: 4) Inner (BROADCAST)
```

```
| | |      Join Cond: (s.product_key = p.product_key) AND (s.product_version = p.product_version)
```

```
| | |      Execute on: All Nodes
```

```
| | | +-- Outer -> STORAGE ACCESS for s [Cost: 893K, Rows: 30M] (PATH ID: 5)
```

```
| | | |      Projection: store.store sales fact b0
```

```
| | | |      Materialize: s.product_key, s.product_version
```

```
| | | |      Execute on: All Nodes
```

```
| | | |      Runtime Filters: (SIP2(HashJoin): s.product_key), (SIP3(HashJoin): s.product_version),
(SIP4(HashJoin): s.product_key, s.product_version), (SIP1(HashJoin): s.store_key)
```

```
| | | +-- Inner -> STORAGE ACCESS for p [Cost: 365, Rows: 60K] (PATH ID: 6)
```

```
| | | |      Projection: public.product_dimension_b0
```

```
| | | |      Materialize: p.product key, p.product version, p.product description
```

```
| | | |      Execute on: All Nodes
```

```
| | +-- Inner -> SELECT [Cost: 68, Rows: 32] (PUSHED GROUPING) (PATH ID: 7)
```

```
| | |      Execute on: All Nodes
```

```
| | | +---> STORAGE ACCESS for store_dimension [Cost: 68, Rows: 32] (PATH ID: 8)
```

```
| | | |      Projection: store.store_dimension_b0
```

```
| | | |      Materialize: store_dimension.store_key
```

```
| | | |      Filter: (store_dimension.store_state = 'MA')
```

```
| | | |      Execute on: All Nodes
```

Cost

Projection

Column Materialization

Path ID

Статистики для плана запроса

- Количество строк
- Демография данных
 - Распределение
 - Минимум и максимум
 - Наличие партиций
 - Количество значений NULL
- Размер строки

Доступ к ресурсам

- Разграничение доступа
- Ресурсные группы
- Мониторинг потребления ресурсов

```
133 CREATE RESOURCE POOL elt
134     MEMORYSIZE '40G'
135     MAXMEMORYSIZE '160G'
136     EXECUTIONPARALLELISM AUTO
137     PRIORITY 100
138     RUNTIMEPRIORITY HIGH
139     RUNTIMEPRIORITYTHRESHOLD 5
140     QUEUETIMEOUT 600
141     RUNTIMECAP '2700 SECOND'
142     PLANNEDCONCURRENCY 40
143     MAXCONCURRENCY 40
144     CPUAFFINITYMODE ANY
145     CPUAFFINITYSET None
146     CASCADE TO DEFAULT;
```














Мониторинг

- Отслеживание метрик в режиме реального времени
- CPU
- IO
- Memory
- Network

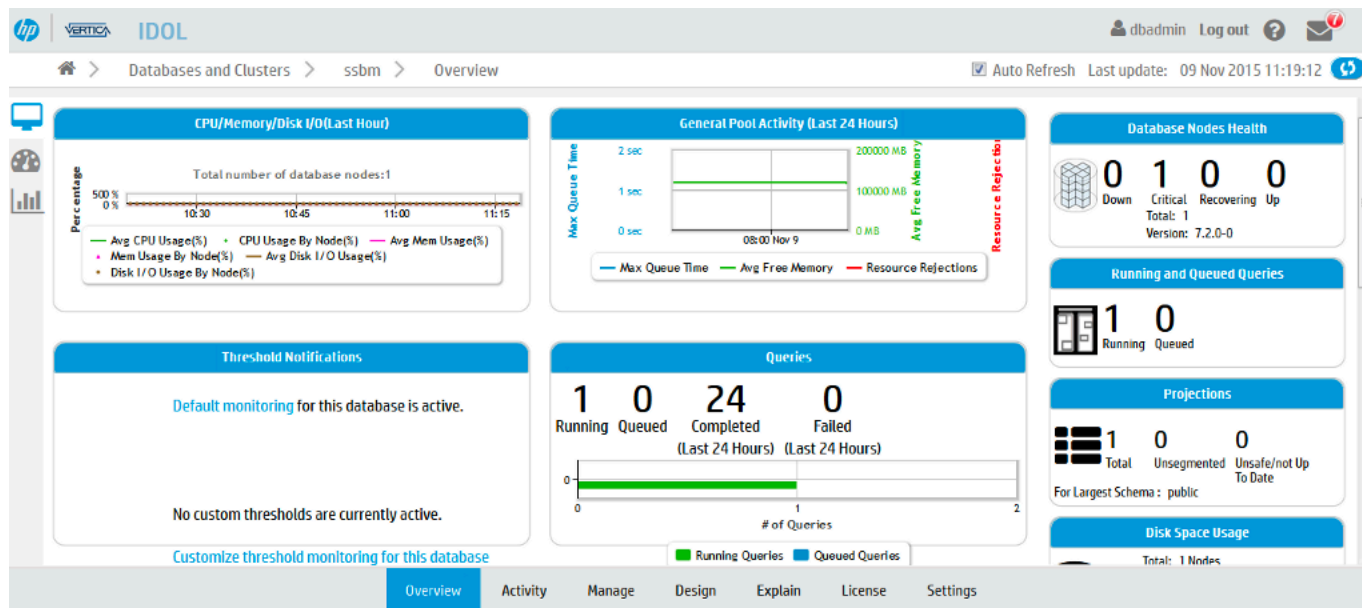
Мониторинг

TERADATA VIEWPOINT Artemiy System Session Utilities +

Add Content Select the portlets you want to add to your page

SYSTEM OVERVIEW	SESSION MANAGEMENT	UTILITIES
 Canary Response Times ? Monitor the canary query response times of a system.	 Completed Queries ? View the final status of queries that ran on a system.	 Remote Console ? Run Teradata Database console utilities remotely from Teradata Viewpoint.
<input checked="" type="checkbox"/> Productivity ? Monitor system health and query performance trends of a system.	 Lock Viewer ? View lock contention between sessions on a system.	 SQL Scratchpad ? Run SQL queries and retrieve the results.
 Space Usage ? Monitor disk space utilization.	 My Queries ? View and manage your queries from multiple systems.	 Viewpoint Calendar ? Schedule and communicate date and time-based events to users.
 System Health ? Monitor all systems' status and key metrics against defined thresholds.	 Query Groups ? View and manage groups of queries that meet predefined criteria.	
 Today's Statistics ? Monitor current resource use and compare statistics to the same period week-over-week.	 Query Monitor ? View and manage all queries on a single system.	
	 Query Spotlight ? View completed queries that exceeded user-defined thresholds.	

Мониторинг



Мониторинг



Моделирование данных

Абстрактное DWH

Stage + CORE + Marts

- **Stage** – приземляем данные as is. Нужен для разгрузки источников от трансформаций, если их нельзя делать на ходу.
- **CORE** – это то, что мы будем разбирать. Это способ моделирования данных в хранилище, который отвечает нашим целям.
- **Marts** – слой представления данных для пользователей.

Наши ... цели?

Требования к самому хранилищу

- Гетерогенность источников
- Гибкость модели данных
- Скорость обновления
- Устойчивость к объёму
- Поддержка историчности

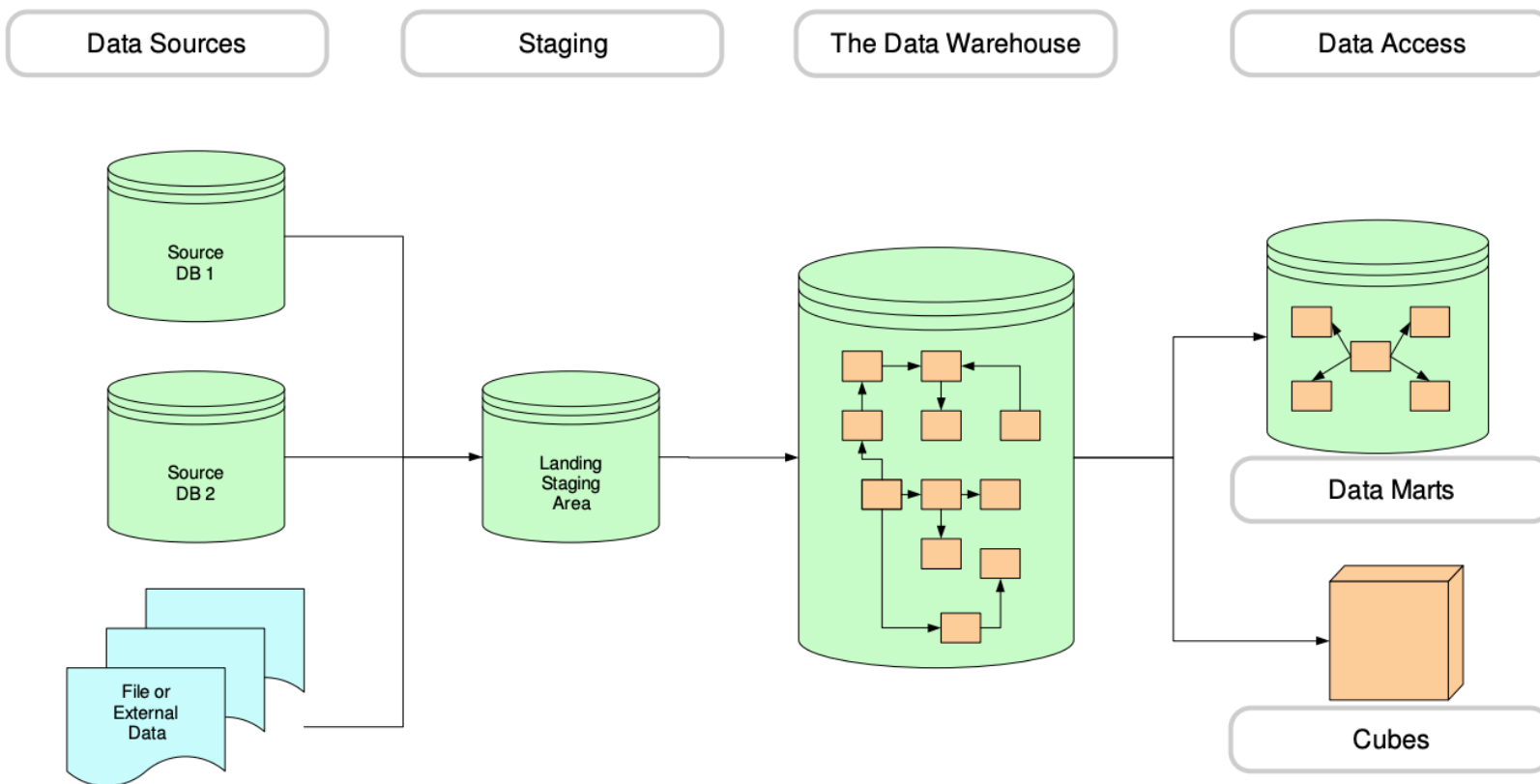
Типовые кейсы:

- Включение новых данных
- Отработка изменений на уровне CORE (DDS)
- Создание пользовательских представлений и их сопровождение
- ...

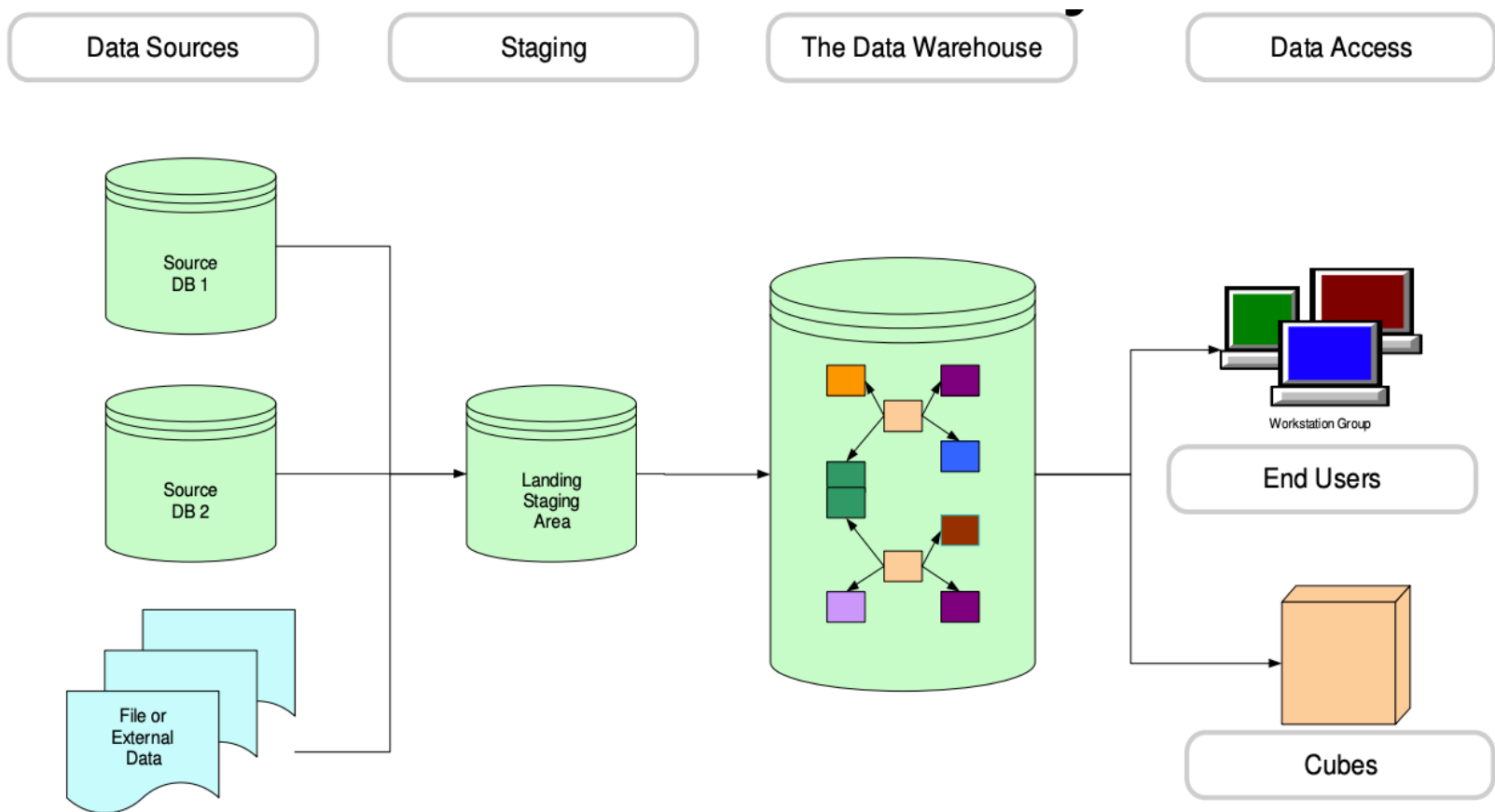
История подходов

- Inmon: top-down, normalized, integrated, enterprise
- Kimball: bottom-up, denormalized, process-based, specific
- Data Vault (Daniel Lindstedt et al)
- Anchor Modelling (Lars Rönnbäck et al)

Подход Bill Inmon



Подход Ralph Kimball

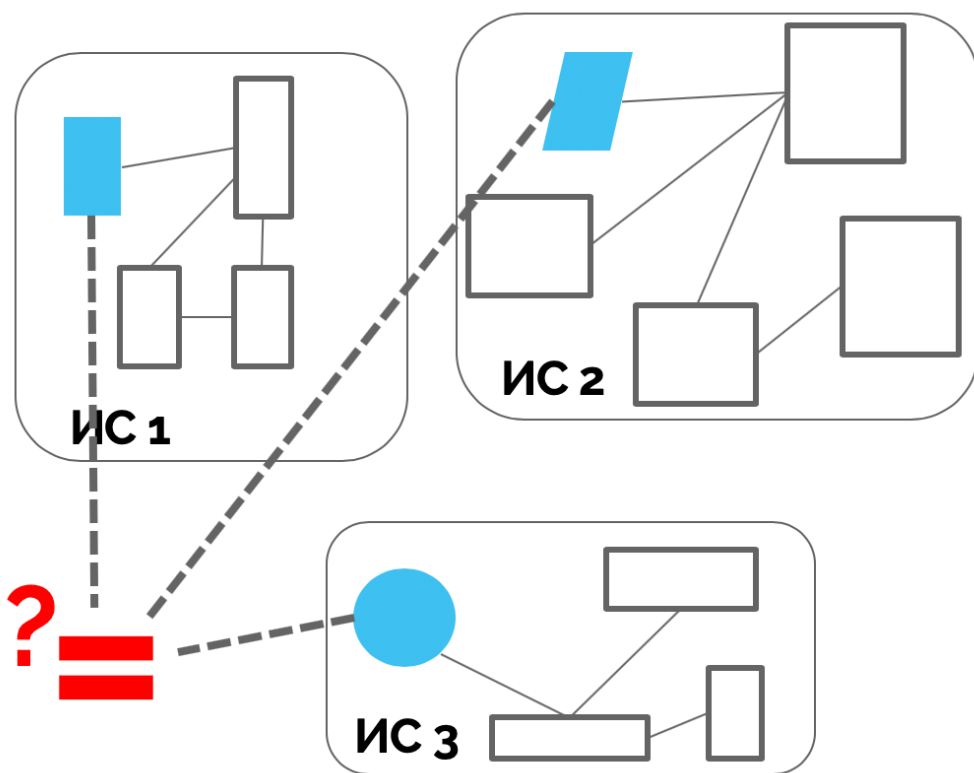


Data Vault

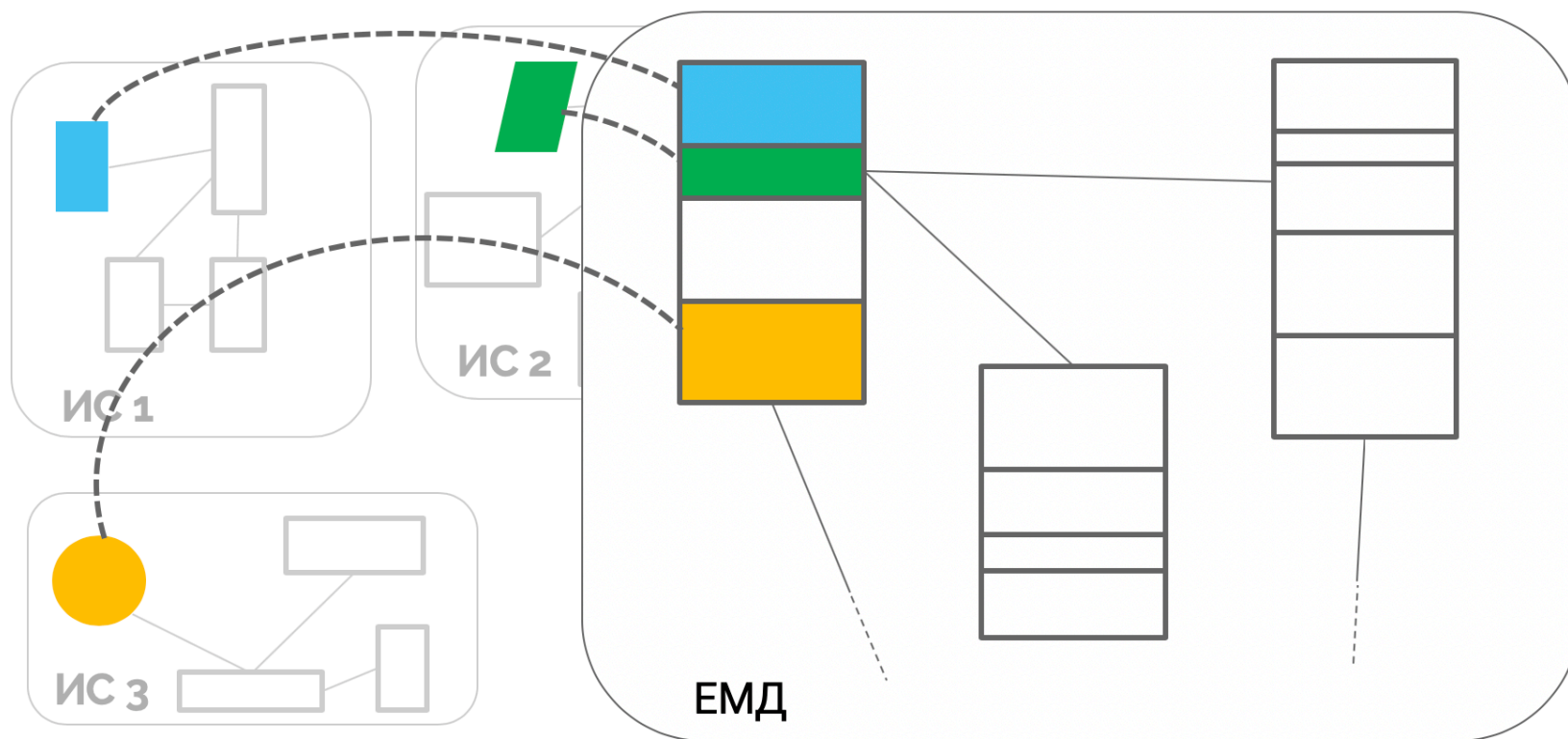
- Есть типизация физических объектов СУБД
- Есть методология их создания, сопровождения
- Есть примеры реализации

* [HP Vertica, проектирование хранилища данных, больших данных](#)

Разные модели ИС



Единая модель



Типы сущностей в DV2.0



HUB

Хранит связь бизнес-ключа с хэш-ключом.

- И метаданные.
- Всё. Никаких свойств объекта не хранится.

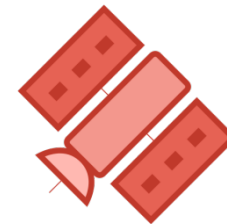


LINK

Используется для связи между хабами

И ТОЛЬКО ХАБАМИ

- Хранит ≥ 2 хэш-ключей хабов и метаданные.
- Больше ничего.
Интертемпоральности у линков тоже нет.



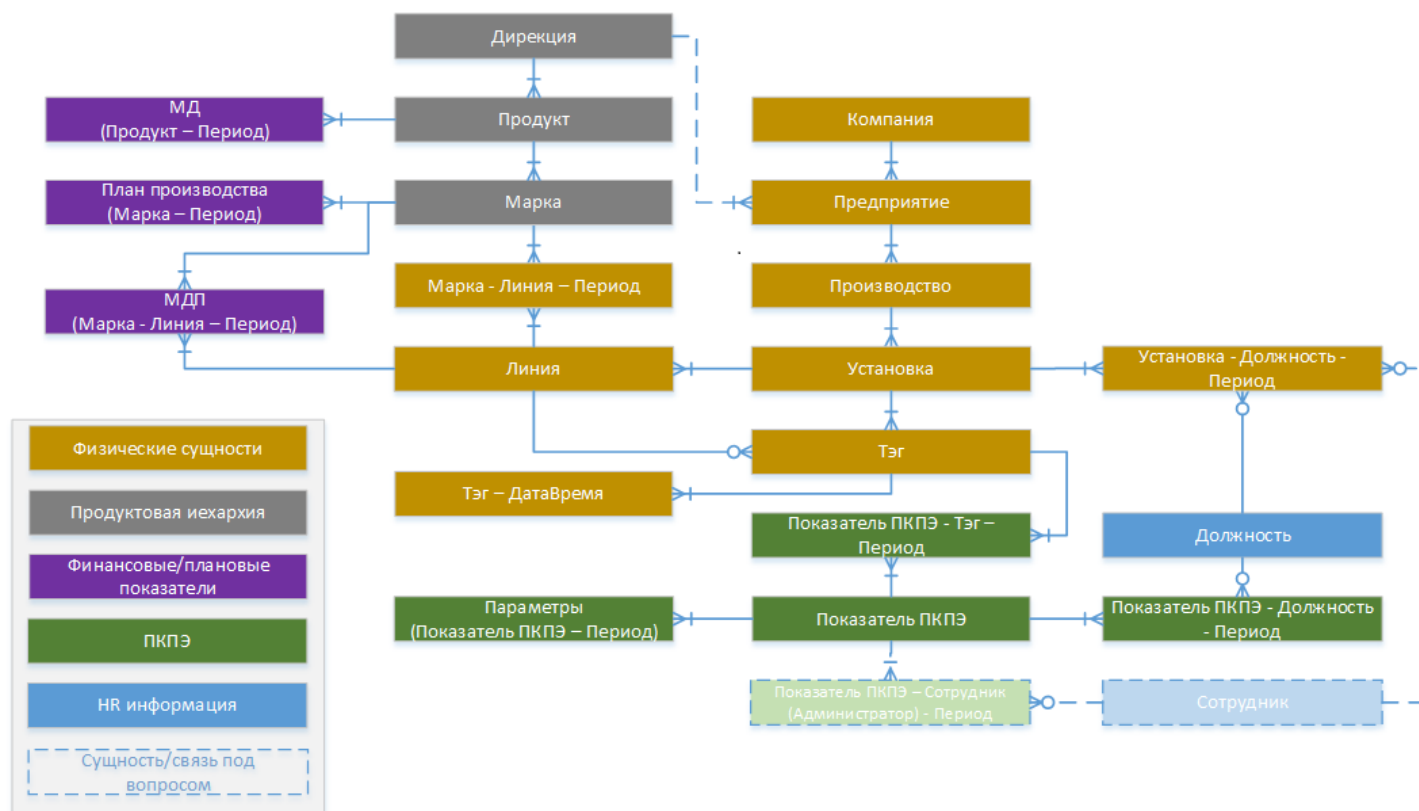
SATELLITE

Это весь контекст.

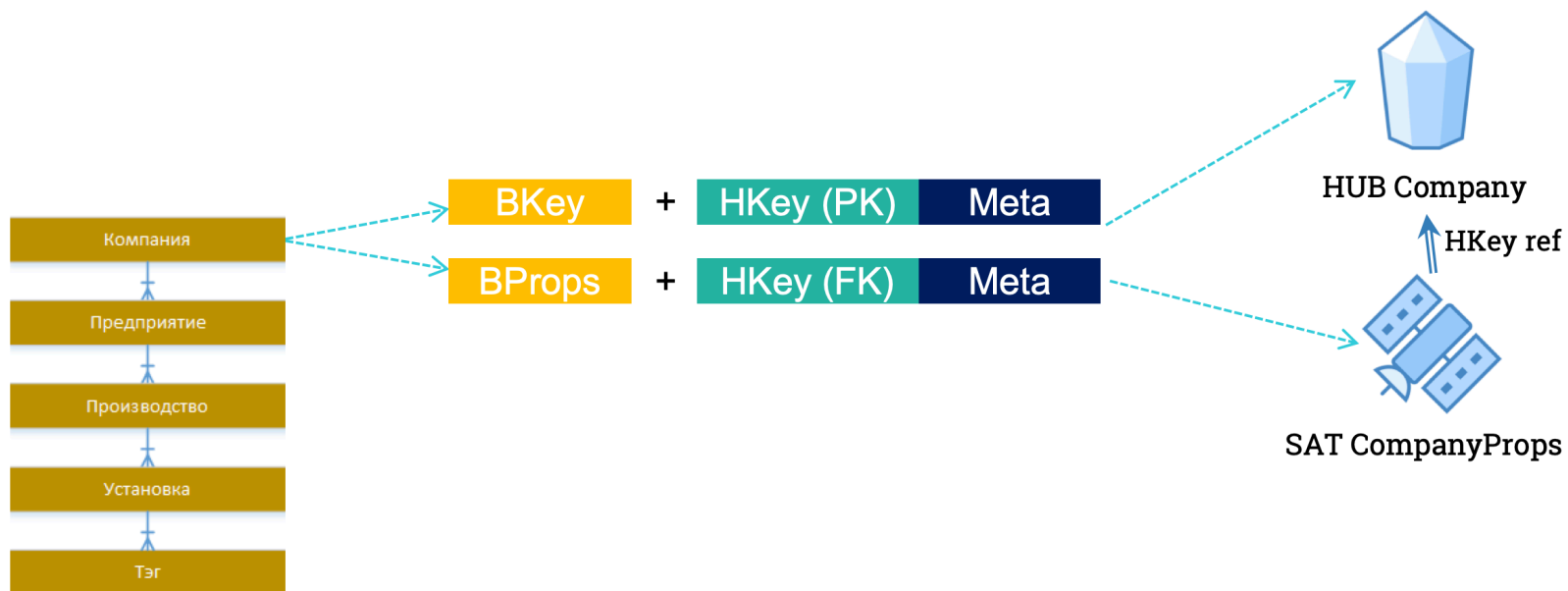
- Хранит все свойства, отношения и прочее.
- Отражает изменения во времени.
- Всегда ссылается только на 1 ХАБ или Линк. Т.е. содержит ссылку на РК этих сущностей

Пример ЛМД

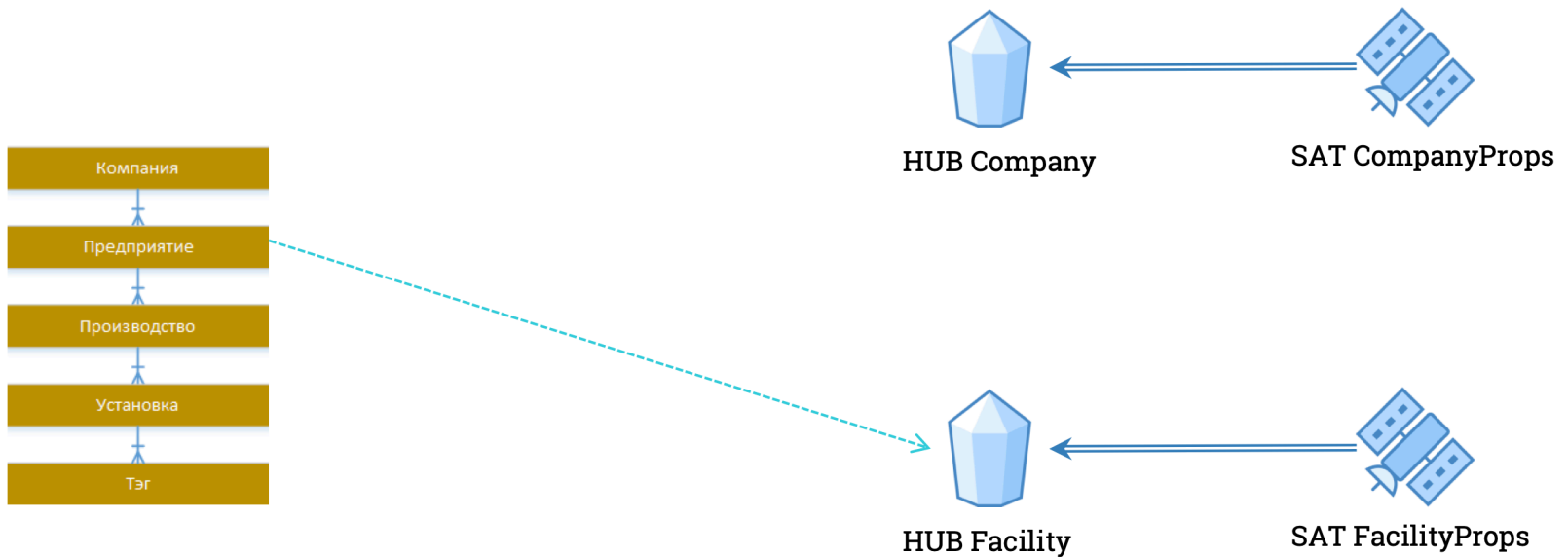
- Это обычная логическая модель
- Сущность определяется логическим разрезом (иногда, не учитывая интертемпоральность)
- Именно так обычно выглядит модели в ИС
- Что с ними делать в ХД?



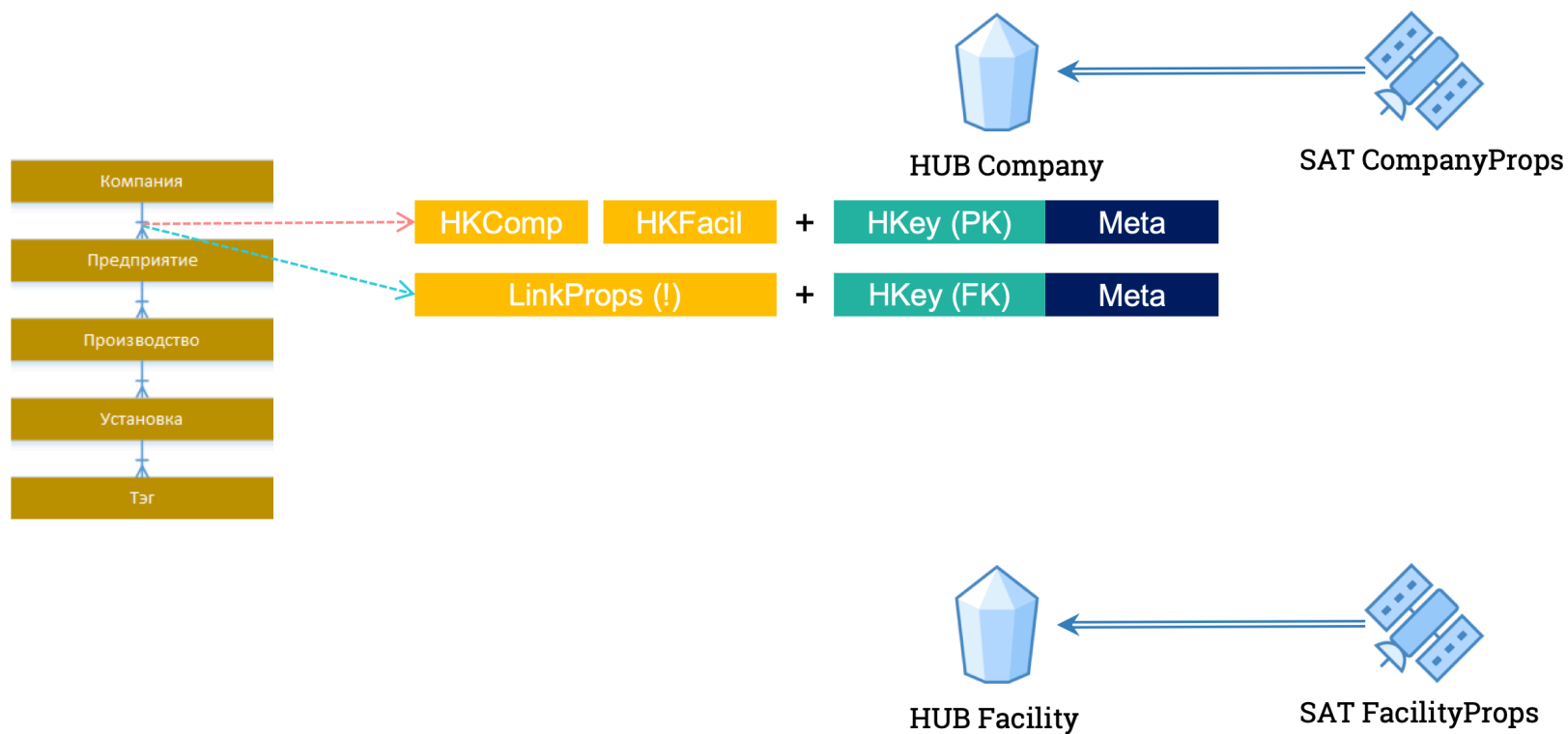
Преобразуем сущности



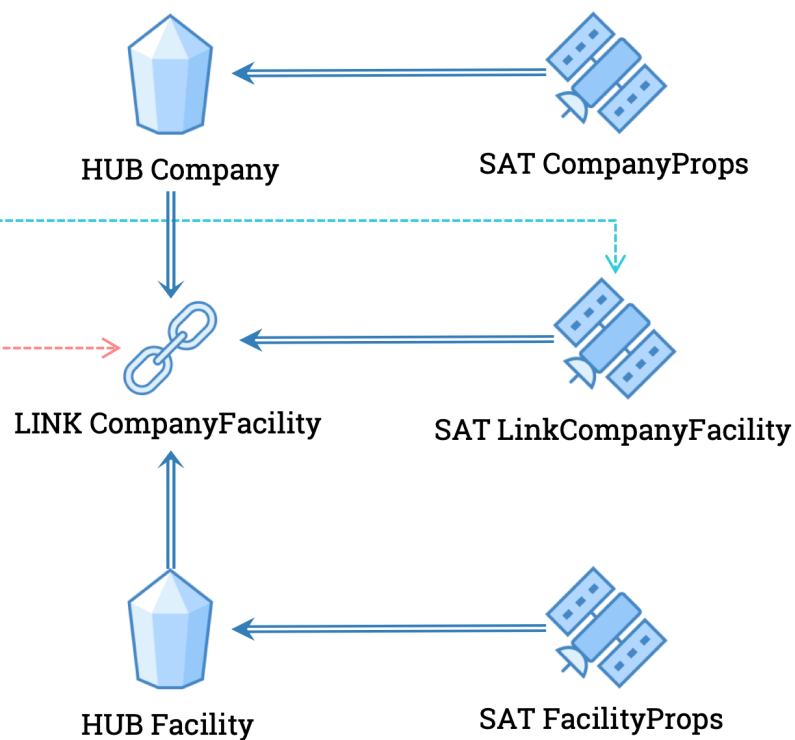
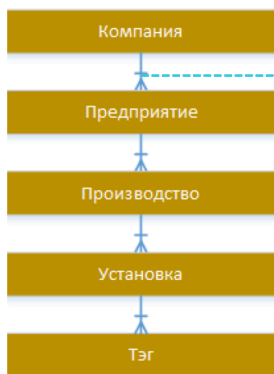
Преобразуем сущности



Преобразуем линки



Преобразуем линки



Что внутри

GENERATED FIELDS
EXAMPLE FIELDS



HUB Company

HK_Company (PK)

LoadDate (PK)
RecordSource

CompanyID



SAT CompanyProps

HK_SatCompany (PK)
HK_Company (FK)

LoadDate (PK)
RecordSource

LoadEndDate
RecordHash
DirectorName
PhoneNumber
KPP



LINK CompanyFacility

HK_LINK (PK)

LoadDate (PK)
RecordSource

HK_Company (FK)
HK_Facility (FK)

И теперь еще раз



HUB

Хранит связь бизнес-ключа с хэш-ключом.

- И метаданные.
- Всё. Никаких свойств объекта не хранится.

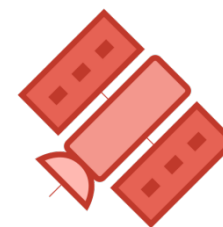


LINK

Используется для связи между хабами

И ТОЛЬКО ХАБАМИ

- Хранит ≥ 2 хэш-ключей хабов и метаданные.
- Больше ничего.
Интертемпоральности у линков тоже нет.

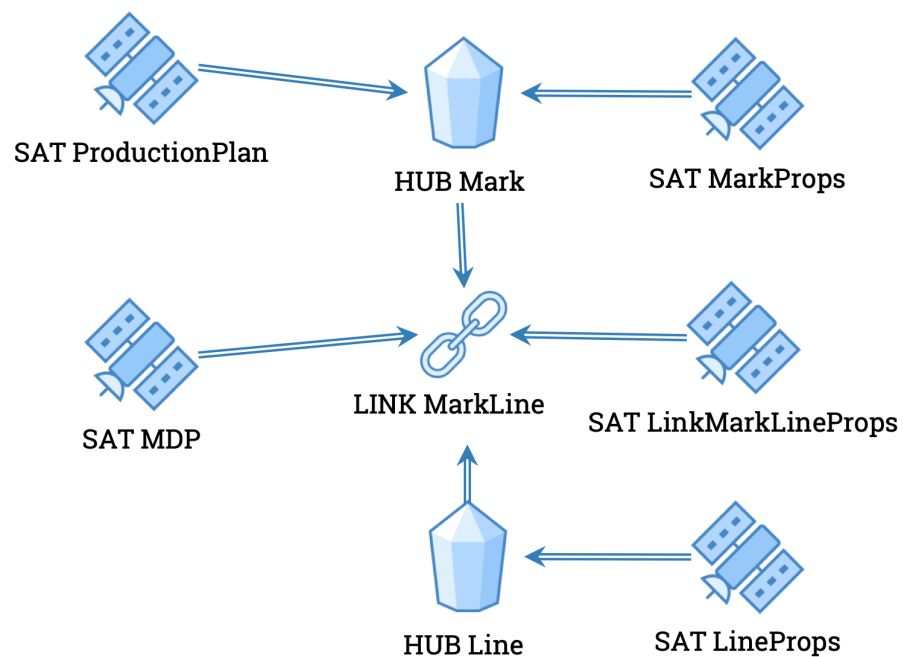
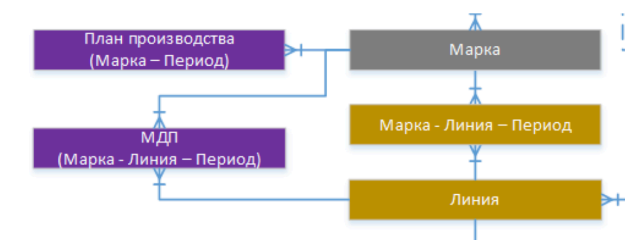


SATELLITE

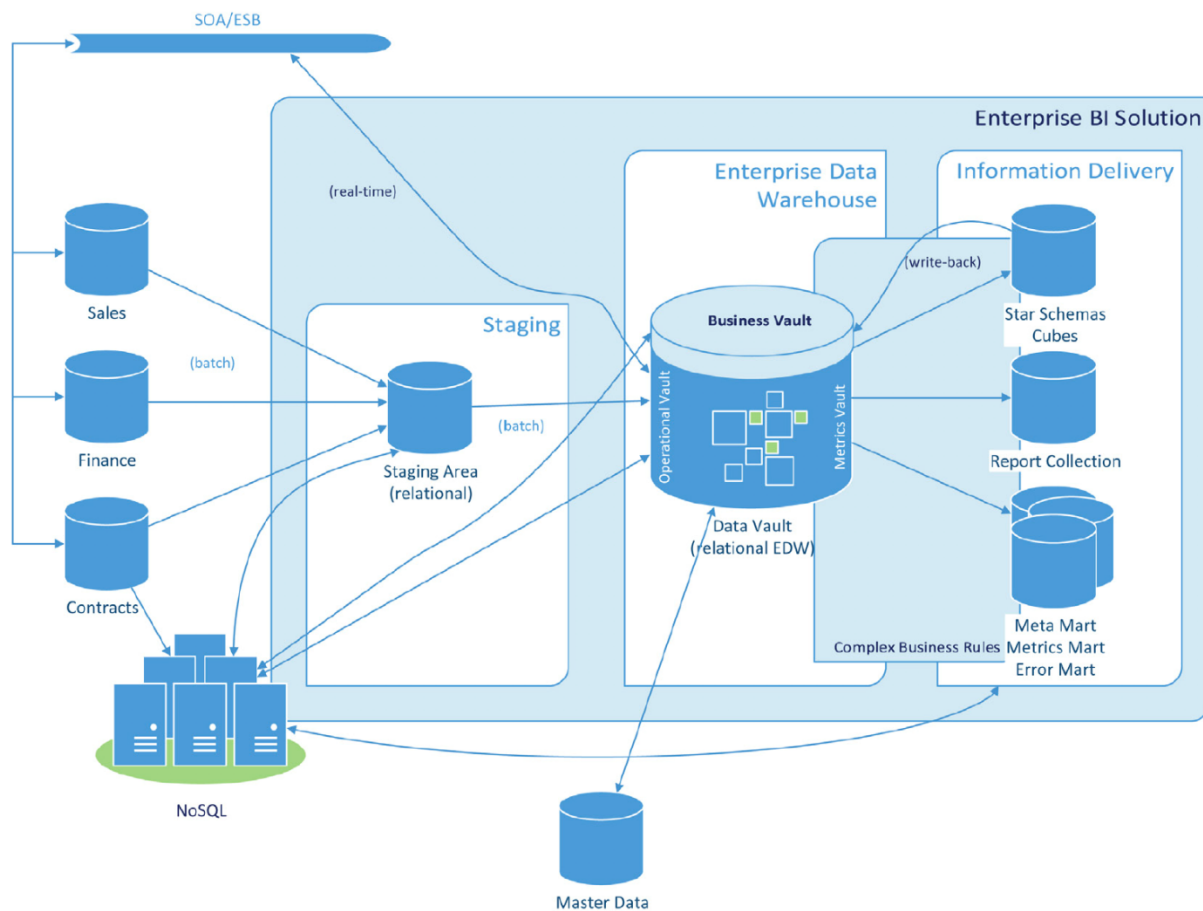
Это весь контекст.

- Хранит все свойства, отношения и прочее.
- Отражает изменения во времени.
- Всегда ссылается только на 1 ХАБ или Линк. Т.е. содержит ссылку на РК этих сущностей

Ещё примеры



А как выглядит DV 2.0?



Пример организации слоев



Но ведь не так всё просто

- Business vault -> Computed Sat, Computed Link (Exploration, Aggregate, ...)
- Business key change -> SAL Links
- Optimization -> Point-in-Time Links, Bridge Links
- ...
- Более 20 архитектурных шаблонов

Ultimate DB Model?

- Возможно. Но вот что ещё надо обсудить
- «А как лучше?». Есть архитектурные шаблоны.
- Как залить?
- Как достать?
- Что там с апдейтами?
- Что со скоростью?
- Что с разработкой?



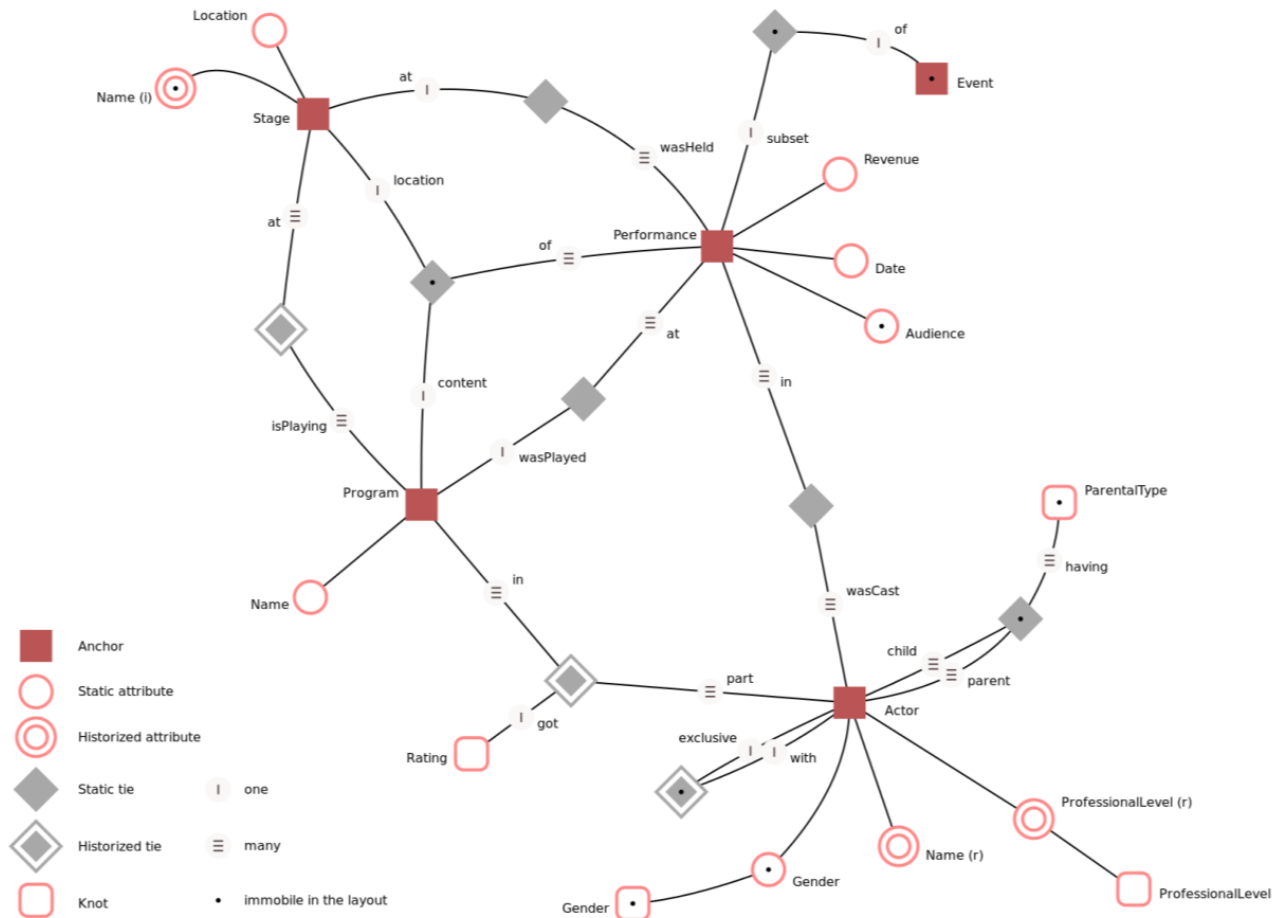
Anchor Modeling

Для моделирования используется 6-я нормальная форма

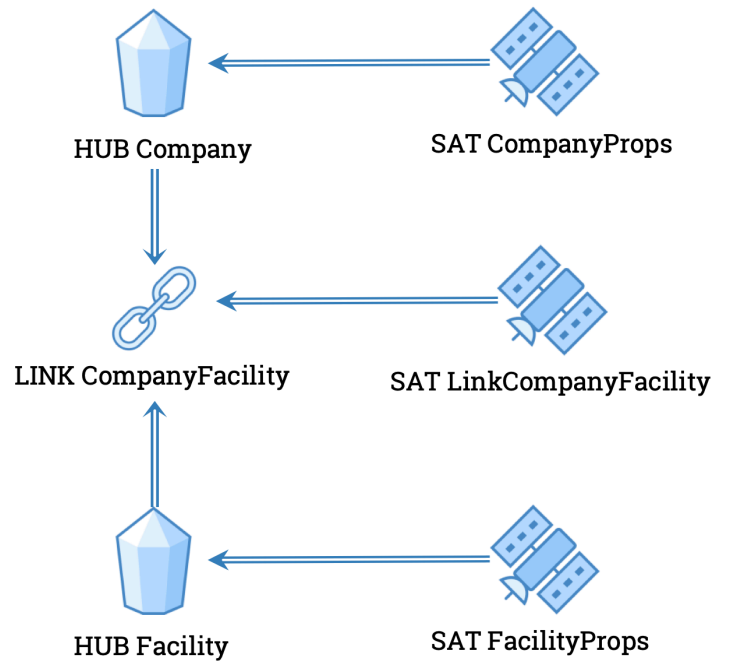
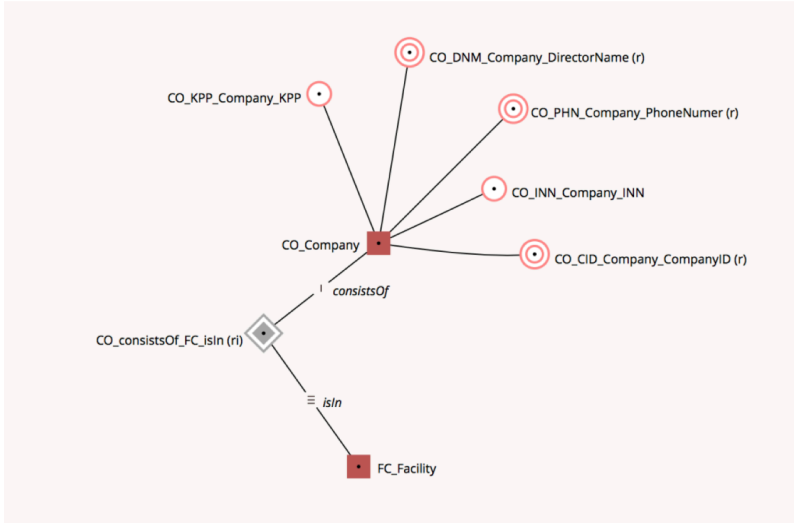
- Anchor \sim HUB (\neq)
- Tie \sim Link (\neq)
- Attribute, Knot \sim Sat (\neq)

* [Anchor Modeling Basics](#)

Anchor Modeling



Сравним?



Anchor vs DataVault

DV

Разрешена группировка атрибутов

Линки не имеют историчности

На линки можно навешивать контекст

Есть best practices по сегментированию, советы по наименованию

Сущность не всегда immutable (может быть новый бизнес ключ)

Деструктивная эволюция схемы

По дефолту использует record time в мета (load date)

Anchor

Атрибуты всегда по отдельности

Линки могут быть историчны

Можно, только если это существенно определяет связь (БНФ) (примеры – темпоральность)

Чёткая методология, вплоть до наименования сущностей и физической модели

Сущность всегда immutable

Недеструктивная эволюция схемы

По дефолту используется change time в мета

Anchor & DataVault

Обе модели

- позволяют относительно легко интегрировать новые данные
- сложны для прямого Querying со стороны пользователя
- могут быстро работать в MPP
- могут превратиться в огромный склад таблиц
- отвечают требованиям (которые были в самом начале)

Демо

Практика

- Загрузим источник
- Построим DV 2.0
- Построим витрину
- Загрузим инкремент
- Расширим атрибутный состав

```
### 1.1. Create STAGE table

CREATE TABLE IF NOT EXISTS SBX.STG_OPER (
  BUSINESS_DT DATE NOT NULL,
  DT DATE NOT NULL,
  PLANT_PRODUCT_ID VARCHAR(64) NOT NULL,
  PLAN NUMERIC(18,5),
  FACT NUMERIC(18,5),
  PRIMARY KEY (DT, PLANT_PRODUCT_ID)
)
ORDER BY
  DT,
  PLANT_PRODUCT_ID
SEGMENTED BY HASH(DT, PLANT_PRODUCT_ID) ALL NODES
;
```

Результаты занятия

- Лучшие практики MPP DWH
- Подходы к моделированию хранилища
 - Bill Inmon
 - Ralph Kimball
 - Data Vault
 - Anchor Modeling
- Демо

Ссылки на материалы

- [Vertica+Anchor Modeling = запусти рост своей грибницы](#)
- [Нюансы реализации колоночного хранения данных - Николай Голов \(Avito\)](#)
- [Data Vault basics by Dan Linstedt](#)
- [Anchor Modeling + Online Modeler](#)
- [Google BigQuery Performance best practices](#)

Домашнее задание

- Спроектировать схему данных + Построить витрину
- Использовать Vertica (Docker) или BigQuery
- Датасет: [Захват данных из divolte](#) (* или [GCP Public Datasets](#))
- Definition of Done:
 - DDL объектов
 - DML шагов преобразований
 - * Опционально: Тестирование на ошибки

Рефлексия

- Что вам запомнилось больше всего
- Пройти опрос

Ваши вопросы?

