

# Data Vault 2.0

В каких случаях внедрять,  
разбор основных проблем  
применения методологии при  
построении DWH на Greenplum

Денис Лукьянов  
Руководитель направления  
архитектуры данных

**ecom.tech**

# ecom.tech

ритейл реального времени в цифрах

650 тыс.

заказов каждый день Самокат  
доставляет своим пользователям

120 городов

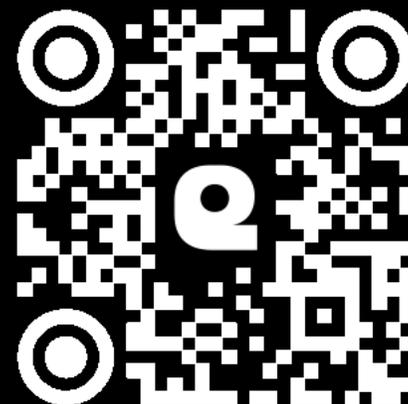
с доступом к быстрой  
доставке продуктов

20 млн

уникальных посетителей  
«Мегамаркета» каждый месяц

60 тыс.

пунктов выдачи заказов  
и постаматов по всей России



# 0 чѐм сегодня поговорим

[01]

Общая схема  
Data Vault

[02]

Генерация ключей

[03]

Выделение  
сущностей

[04]

Выделяем  
и убираем связи

[05]

Формат истории  
и PIT таблицы

[06]

Унификация  
и расчеты

[07]

Итог — схема полного  
тракта построения DWH

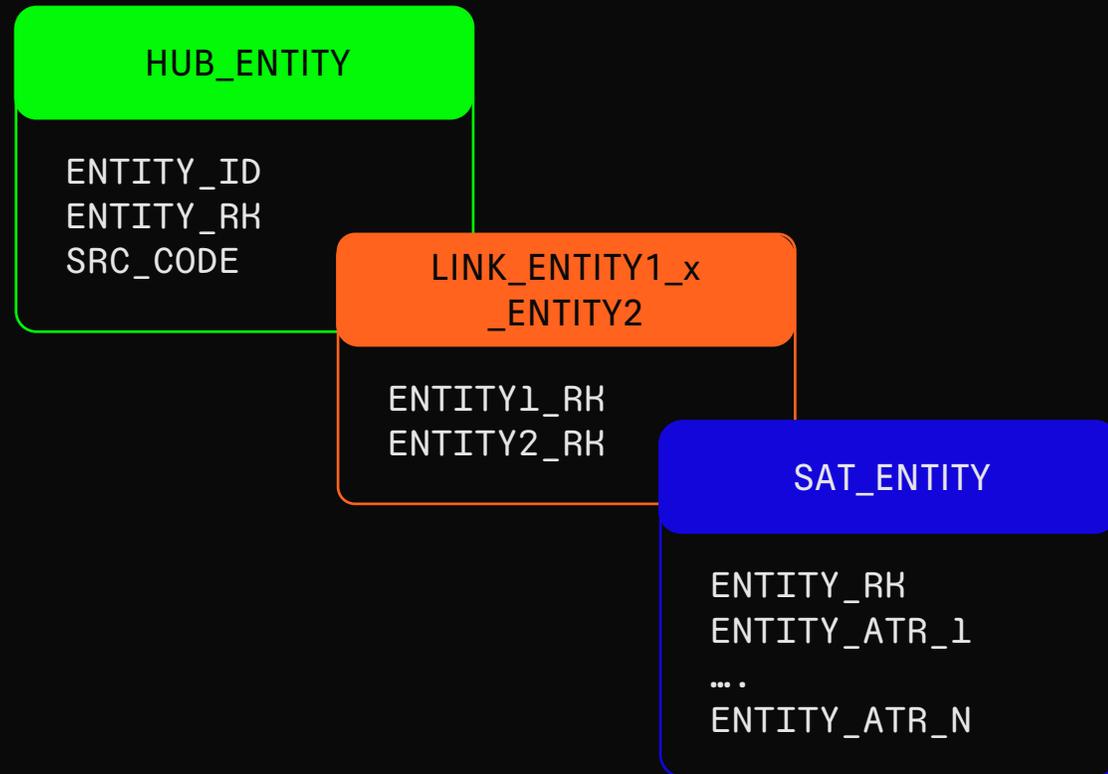


raw data  
vault

[01]

# Raw Data Vault

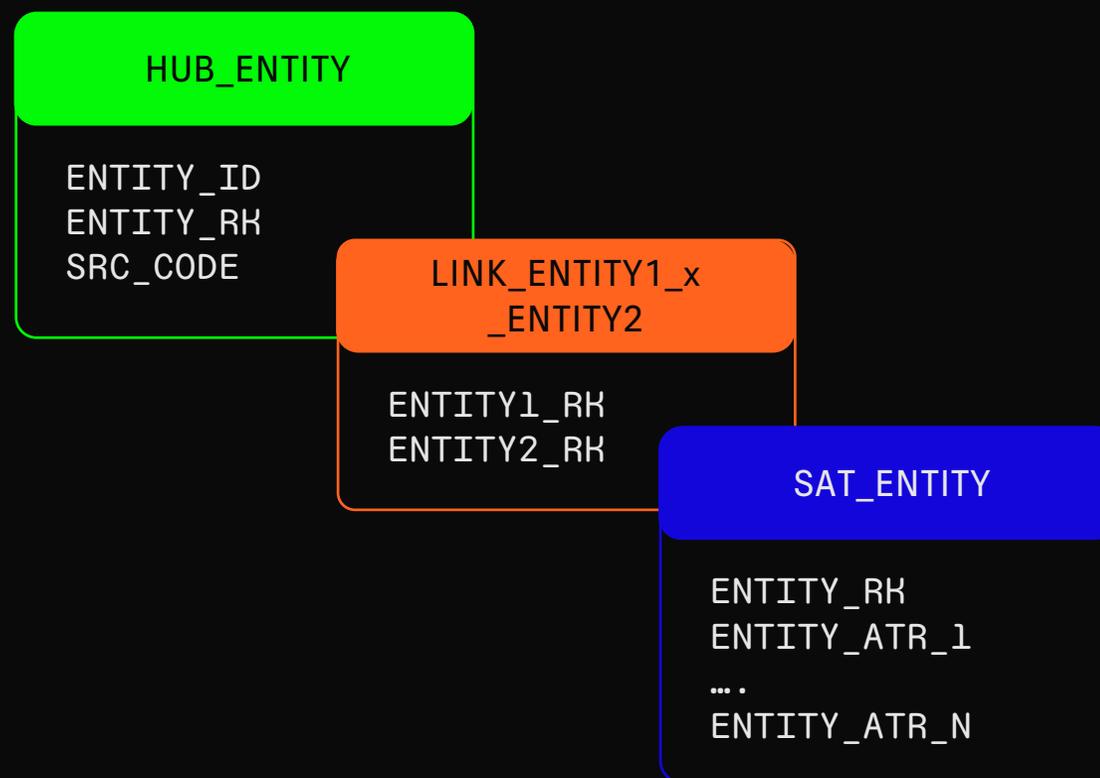
**HUB** — таблица, содержащая связку бизнес-ключа сущности и суррогатного ключа хранилища



# Raw Data Vault

**HUB** — таблица, содержащая связку бизнес-ключа сущности и суррогатного ключа хранилища

**LINK** — таблица связей сущностей по суррогатным ключам

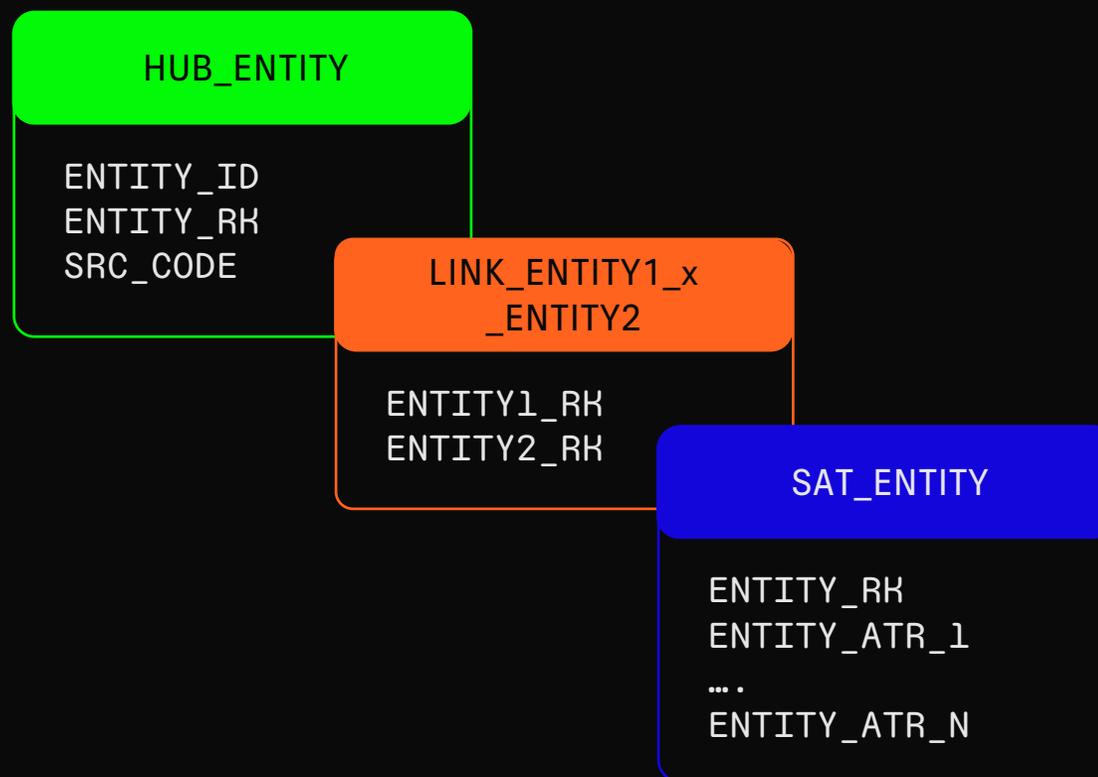


# Raw Data Vault

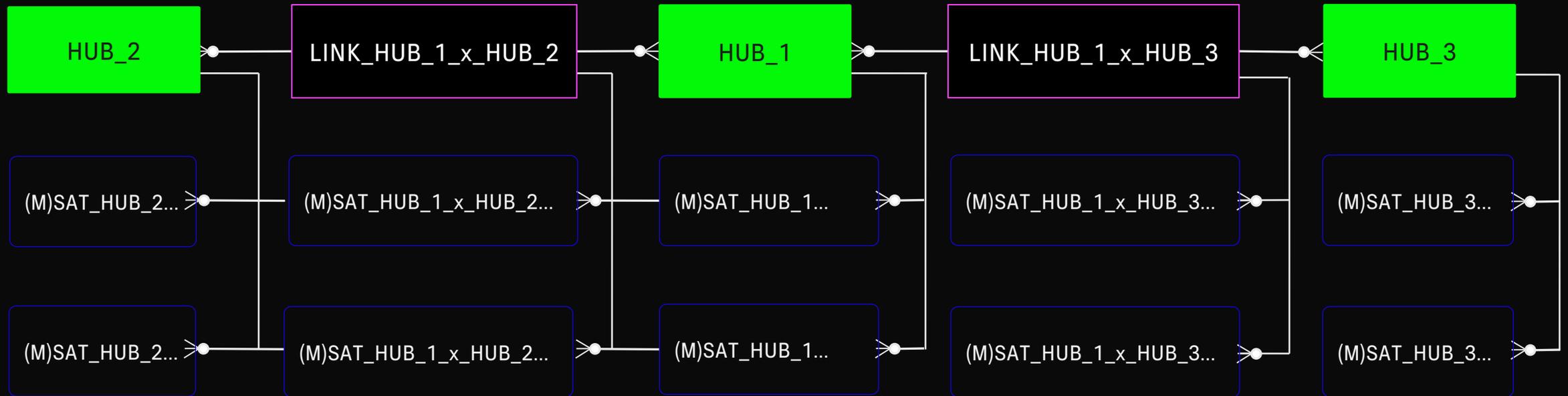
**HUB** — таблица, содержащая связку бизнес-ключа сущности и суррогатного ключа хранилища

**LINK** — таблица связей сущностей по суррогатным ключам

**SAT** — таблица атрибутов сущности

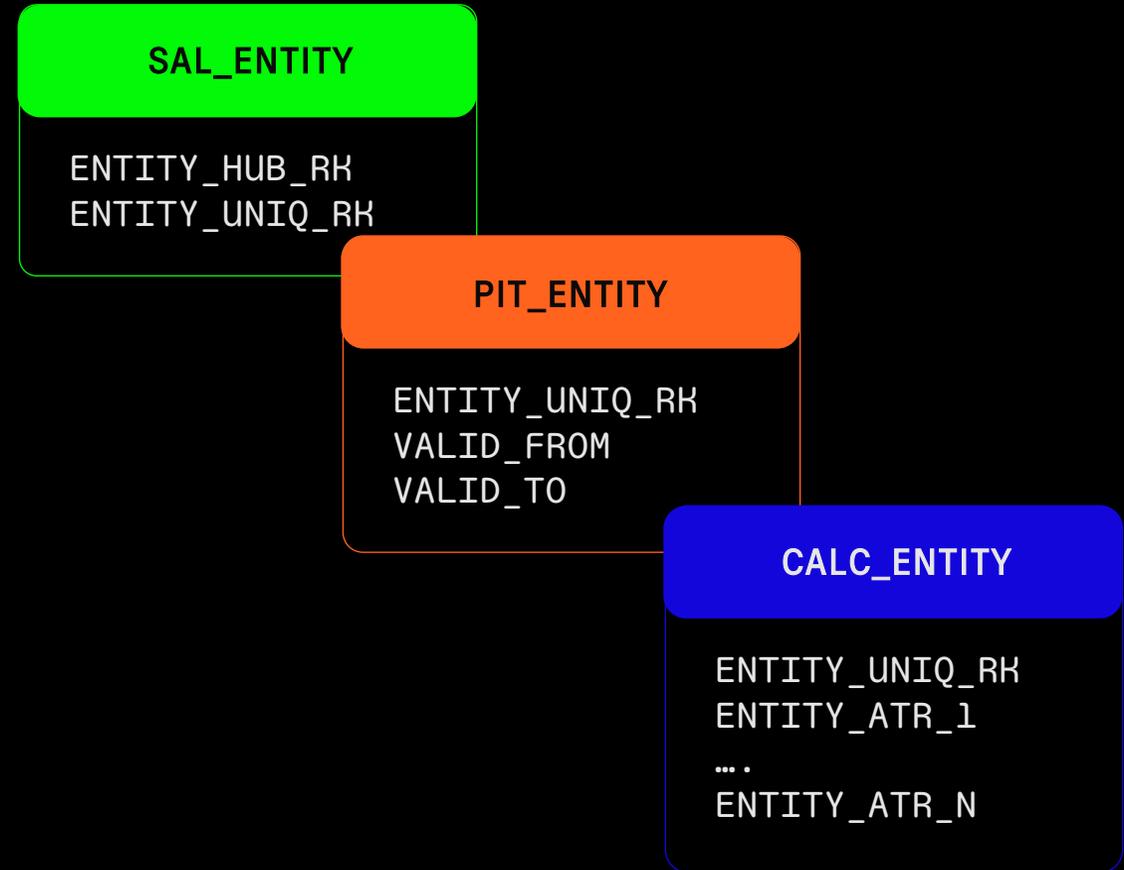


# Raw Data Vault



# Business Vault

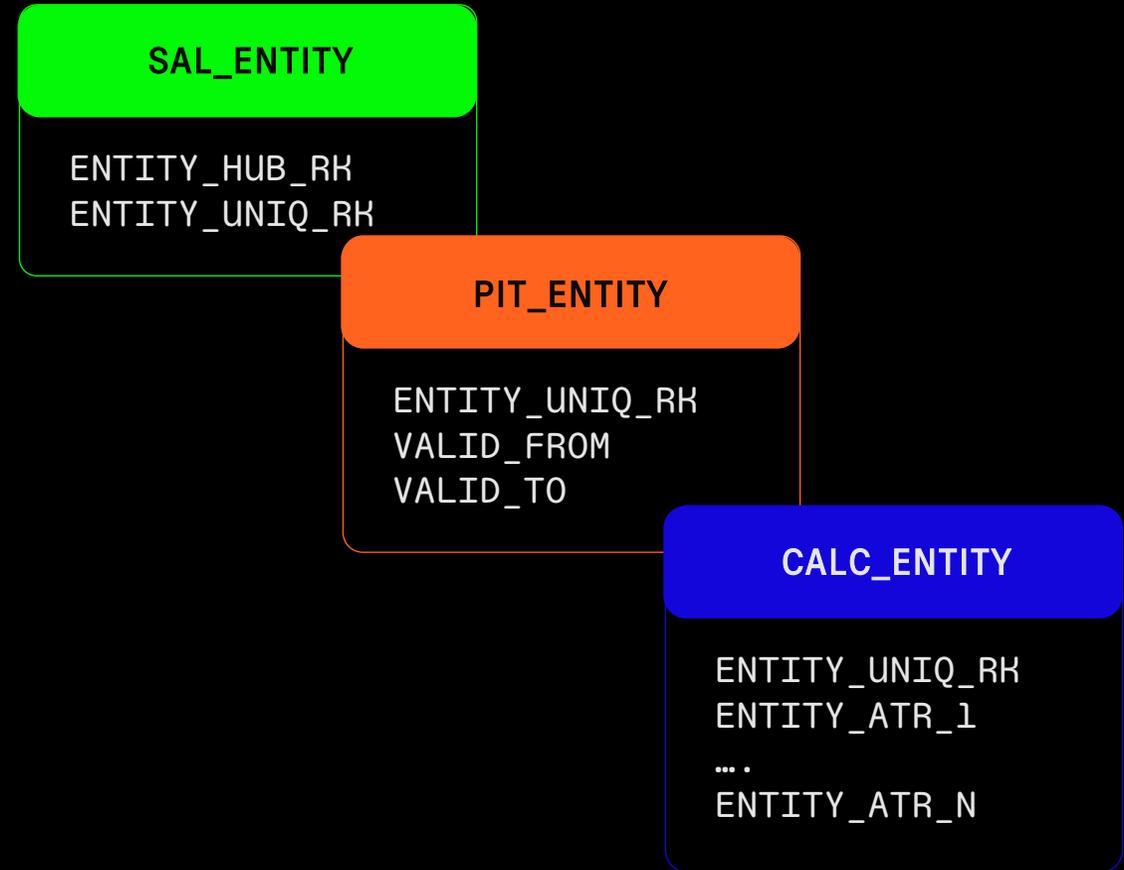
**SAL** — таблицы унификации ключей  
сущности



# Business Vault

**SAL** — таблицы унификации ключей  
сущности

**PIT** — таблица расчета полной истории  
сущности (опционально)

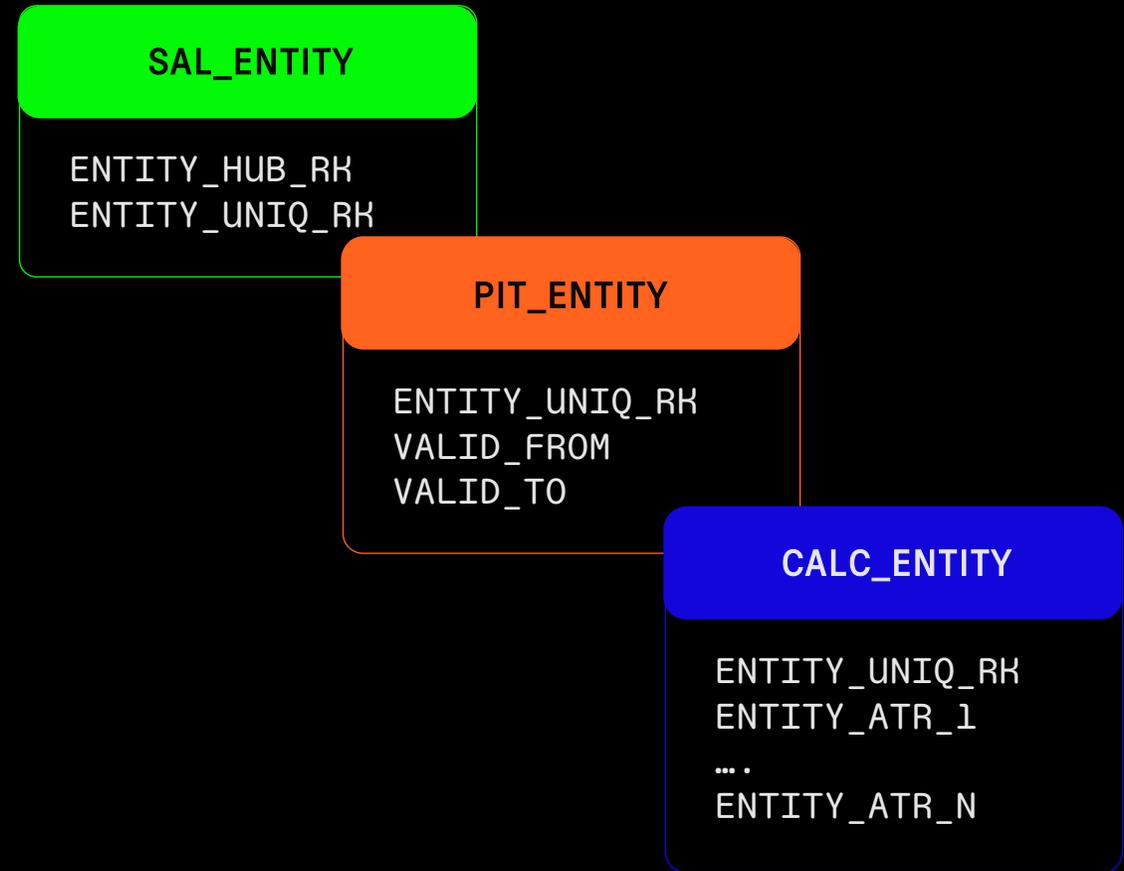


# Business Vault

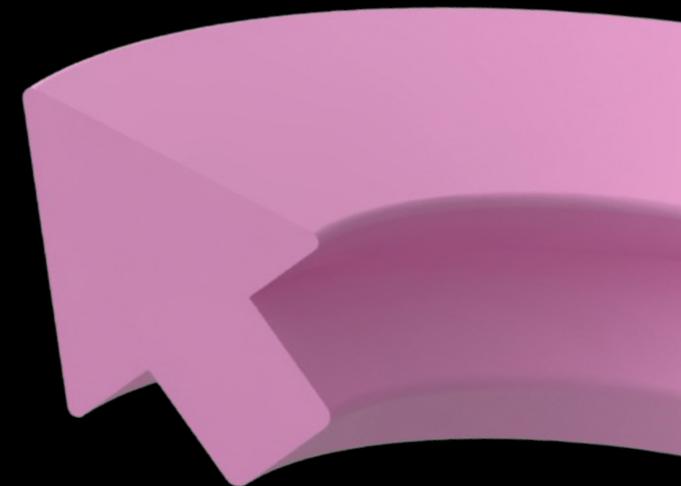
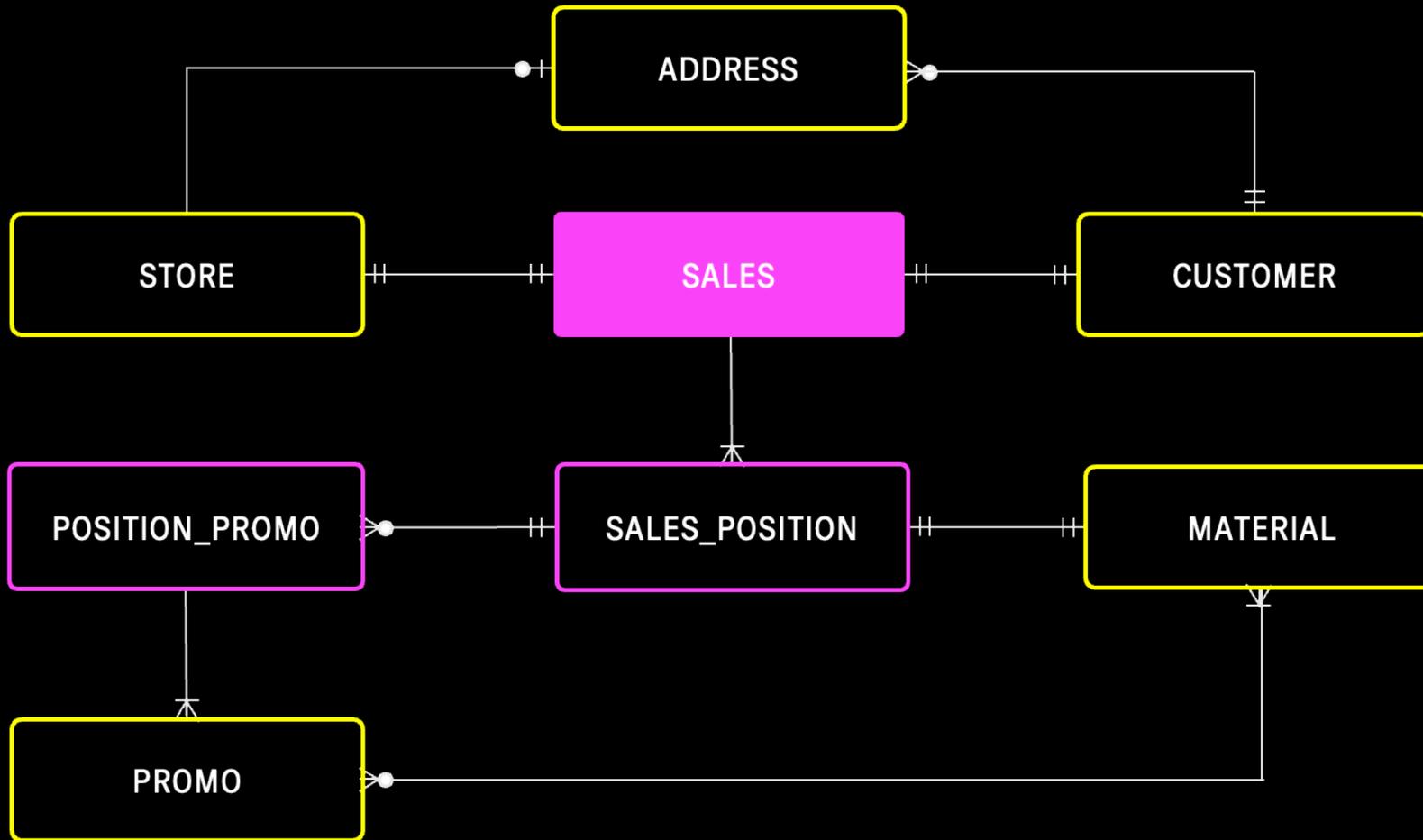
**SAL** — таблицы унификации ключей сущности

**PIT** — таблица расчета полной истории сущности (опционально)

**Bridge/Calculation** tables —  
таблицы с любыми видами расчетов



# Корпоративная модель данных: продажи

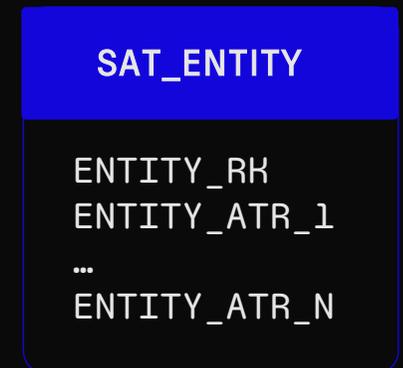
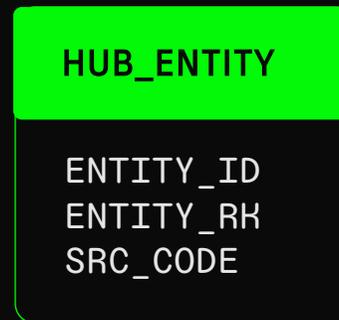
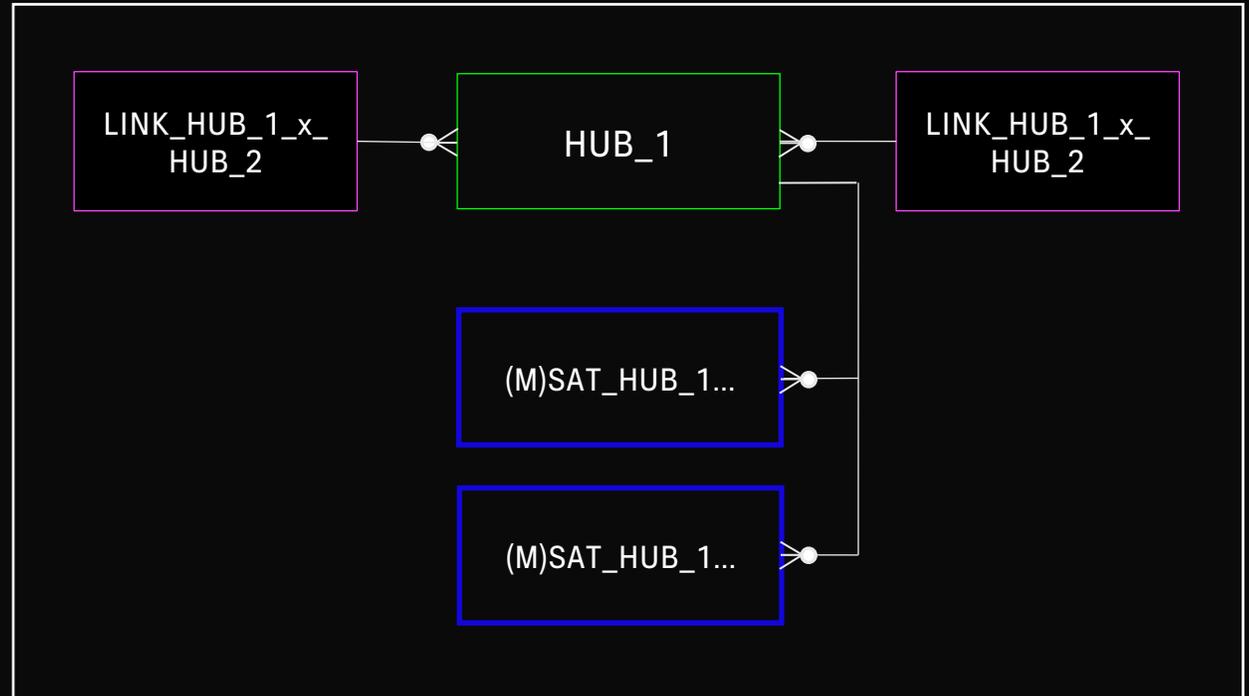


# ГЕНЕРАЦИЯ КЛЮЧЕЙ

[02]

# Генерация ключей

Sequence — надежно и синхронно

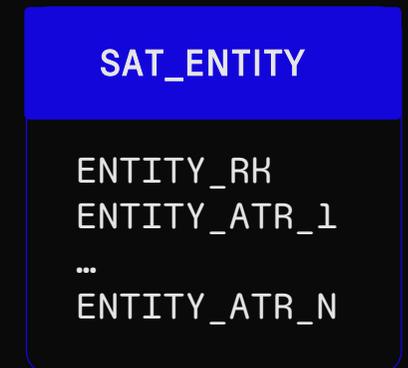
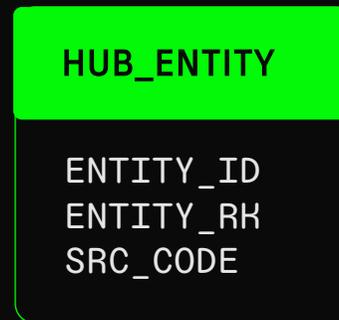
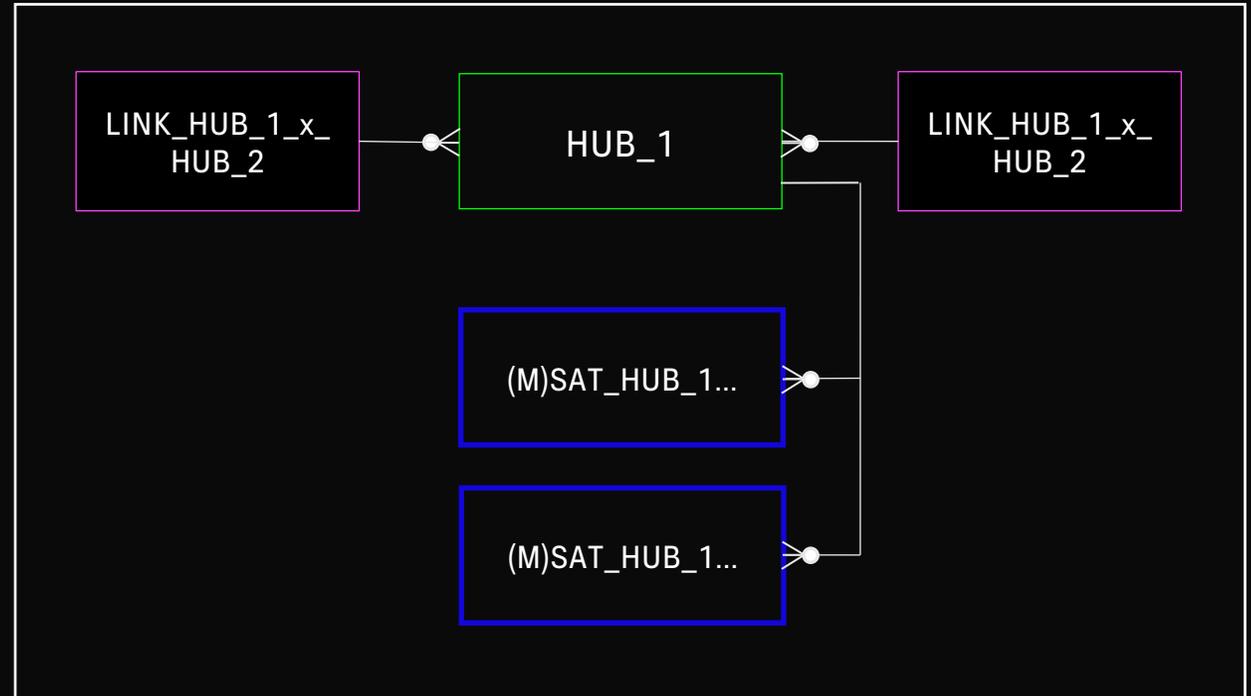


# Генерация ключей

**Sequence** — надежно и синхронно

**Hash** — коллизии и ассинхрон  
(шанс коллизии у md5 50% на  $2^{64}$  записей)

Используем hash,  
если у вас нет явных  
требований на 100%  
точность в DWH.  
А если они есть,  
переубеждаем  
тех, кто их поставил



# Генерация ключей

```
material_rk_3 = f(material_bk)
```

`material_rk_3` — минимум затрат, но есть риски не найти одинаковый `bk` во всех связанных системах и таблицах.

# Генерация ключей

```
material_rk_3 = f(material_bk)
```

```
material_rk_2 = f(material_id # material_src_code)
```

`material_rk_3` — минимум затрат, но есть риски не найти одинаковый bk во всех связанных системах и таблицах.

`material_rk_2` — генерация от мастер системы. Ломается при миграции мастера.

# Генерация ключей

```
material_rk_3 = f(material_bk)
```

```
material_rk_2 = f(material_id # material_src_code)
```

```
material_rk_1 = f(material_id # sales_src_code)
```

Учитывая тенденцию  
к микросервисной архитектуре  
наших источников, рекомендуем  
(и лично используем) вариант 1.  
Но выбор за вами



**material\_rk\_3** — минимум затрат, но есть риски не найти одинаковый bk во всех связанных системах и таблицах.

**material\_rk\_2** — генерация от мастер системы. Ломается при миграции мастера.

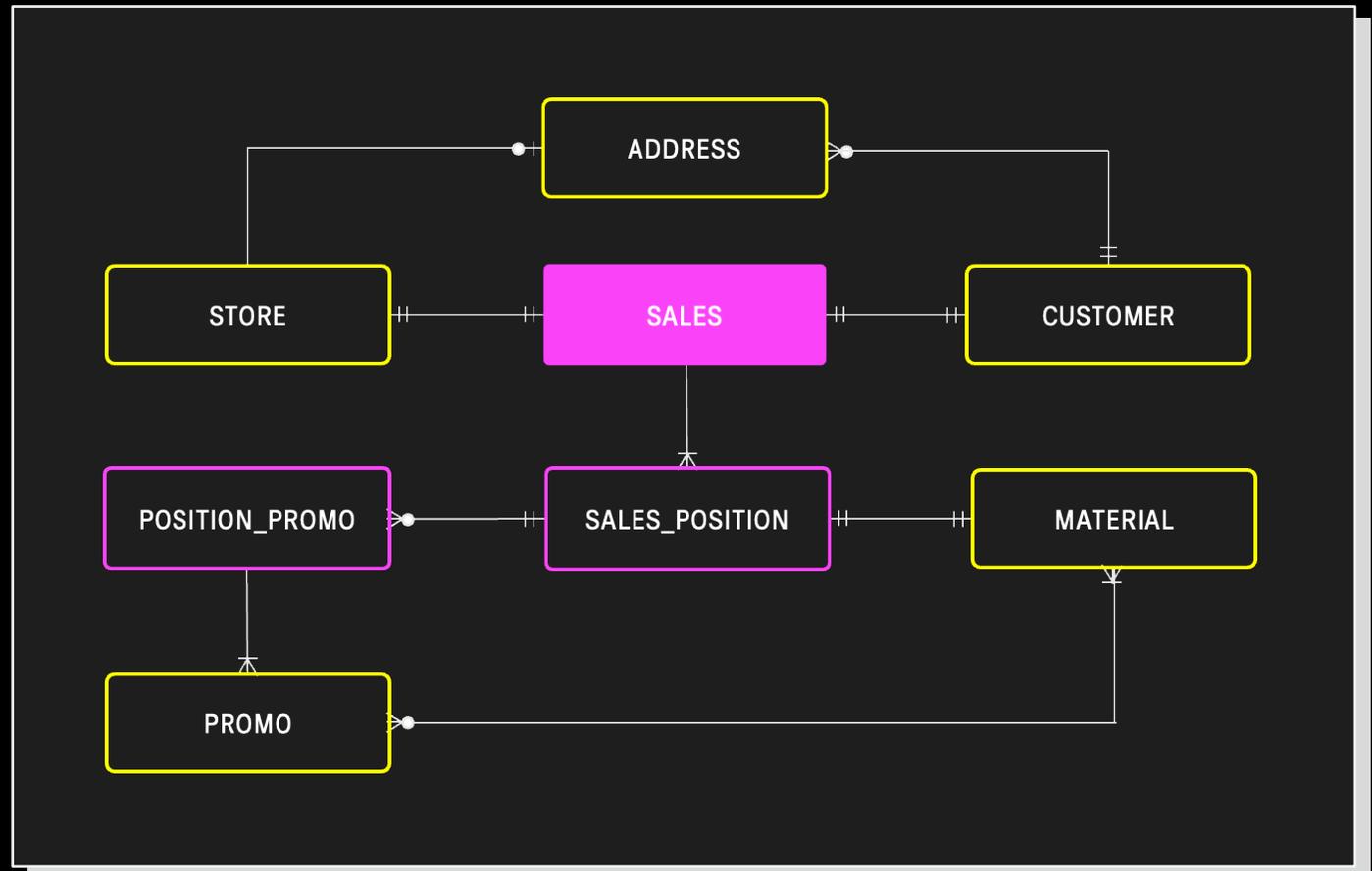
**material\_rk\_1** — генерация от системы из которой пришла запись. Больше надежности, больше избыточности.

# ВЫДЕЛЕНИЕ СУЩНОСТЕЙ

# Выделяем hub

1	hub_sales*	✓
2	hub_address	✓
3	hub_store	✓
4	hub_customer	✓
5	hub_material	✓
6	hub_promo	✓

1	hub_sales_position	???
2	hub_position_promo	???



# Моделирование фактов – теория из коробки

Hub + полная обяза — вариант  
без условий

# Моделирование фактов – теория из коробки

**Hub + полная обязательна** — вариант без предусловий

**Link** — у сущности отсутствует самостоятельный ключ, является связью нескольких сущностей и бизнес даты(классика - остатки)

# Моделирование фактов – теория из коробки

**Hub + полная обвязка** — вариант без предусловий

**Link** — у сущности отсутствует самостоятельный ключ, является связью нескольких сущностей и бизнес даты(классика - остатки)

**Transactional link** — комбинация hub+sat+link в одном объекте. Может использоваться только в случаях:

- загрузка строго из одного источника
- отсутствие изменений по ключу

К сожалению, вариант «около вырожденный» в микросервисной архитектуре

# Немного софистики и опции:

Как можно смоделировать детализацию событий

1	hub_sales_position
2	hub_position_promo

# Немного софистики и опции:

Как можно смоделировать детализацию событий

1	hub_sales_position
2	hub_position_promo

## [01] Pros

- честное всегда рабочее с точки зрения модели решение
- не требует дополнительного анализа

# Немного софистики и опции:

Как можно смоделировать детализацию событий

1	hub_sales_position
2	hub_position_promo

## [01] Pros

- честное всегда рабочее с точки зрения модели решение
- не требует дополнительного анализа

## [02] Cons

- неоптимально с точки зрения MPP систем (соединения по ключам с различным распределением)
- модель по умолчанию разрастается по количеству объектов (хабы, линки, в дальнейшем SAL/PIT)

# Немного софистики и опции:

Как можно смоделировать детализацию событий

1	msat_HUB_SALES_sales_position
2	msat_HUB_SALES_position_promo

## Технические ограничения:

1. все ссылки на детализацию содержат ключ основной сущности
2. один экземпляр порождается строго одной системой

# Немного софистики и опции:

Как можно смоделировать детализацию событий

1	msat_HUB_SALES_sales_position
2	msat_HUB_SALES_position_promo

## Технические ограничения:

1. все ссылки на детализацию содержат ключ основной сущности
2. один экземпляр порождается строго одной системой

### [01] Pros

- оптимальное последующее соединение таблиц
- меньше объектов модели

# Немного софистики и опции:

Как можно смоделировать детализацию событий

1	msat_HUB_SALES_sales_position
2	msat_HUB_SALES_position_promo

## Технические ограничения:

1. все ссылки на детализацию содержат ключ основной сущности
2. один экземпляр порождается строго одной системой

### [01] Pros

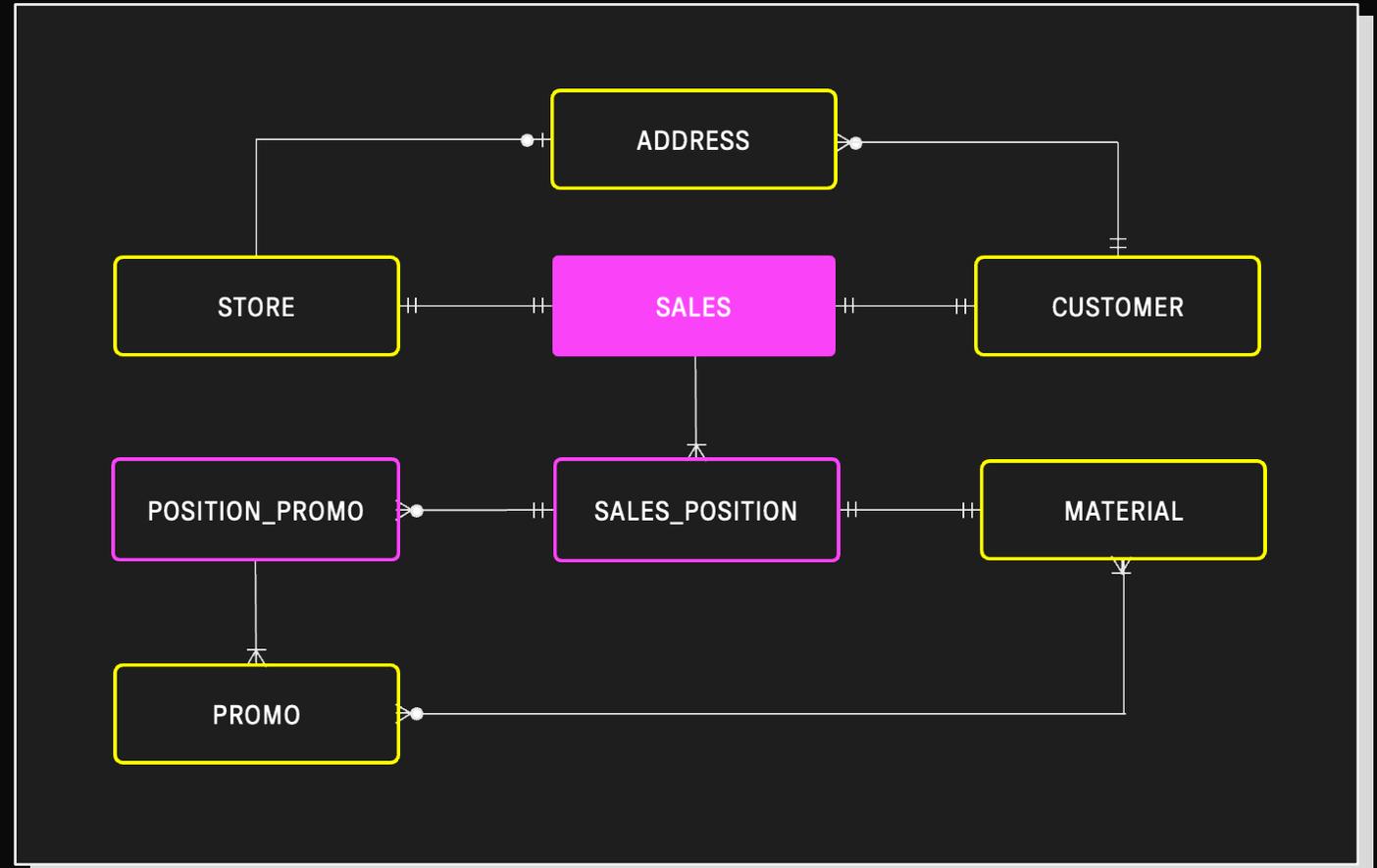
- оптимальное последующее соединение таблиц
- меньше объектов модели

### [02] Cons

- есть предусловия для такого решения
- требуется дополнительный анализ, если риск изменения предусловий с течением времени

# Выделяем hub

1	hub_sales	✓
2	hub_address	✓
3	hub_store	✓
4	hub_customer	✓
5	hub_material	✓
6	hub_promo	✓



1	hub_sales_position/msat_sales_....
2	hub_position_promo/msat_sales_....

Однозначно правильного ответа нет

# ВЫДЕЛЕНИЕ СВЯЗЕЙ

# Связи (LINK): немного теории

01

Изменение кардинальности связи не порождает изменения модели ядра хранилища.

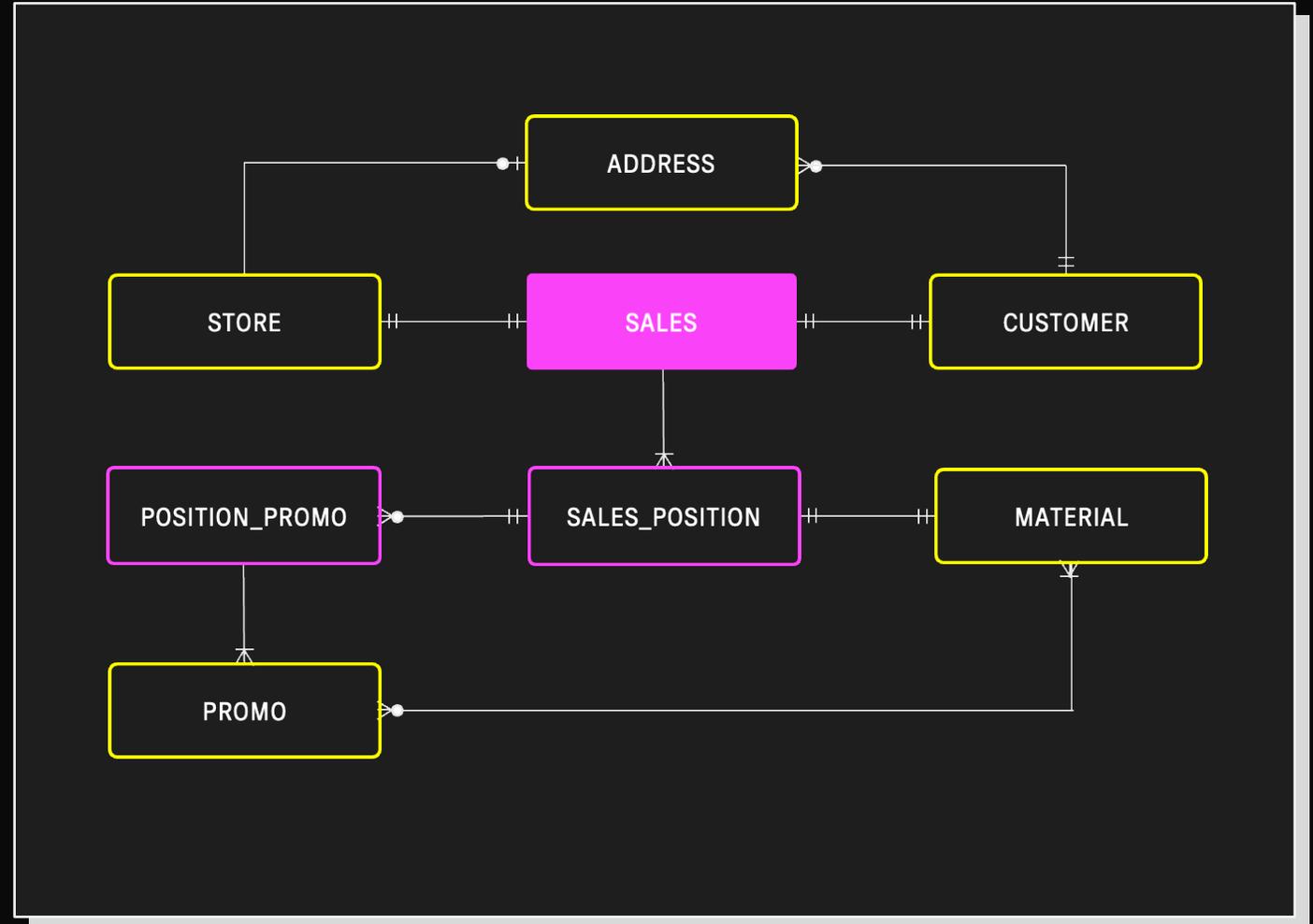
02

Накопление истории связи и ее переиспользование — в отрыве от остальных атрибутов (нормализация).



# Выделяем link

1	link_address_x_store	✓
2	link_address_x_customer	✓
3	link_material_x_promo	✓

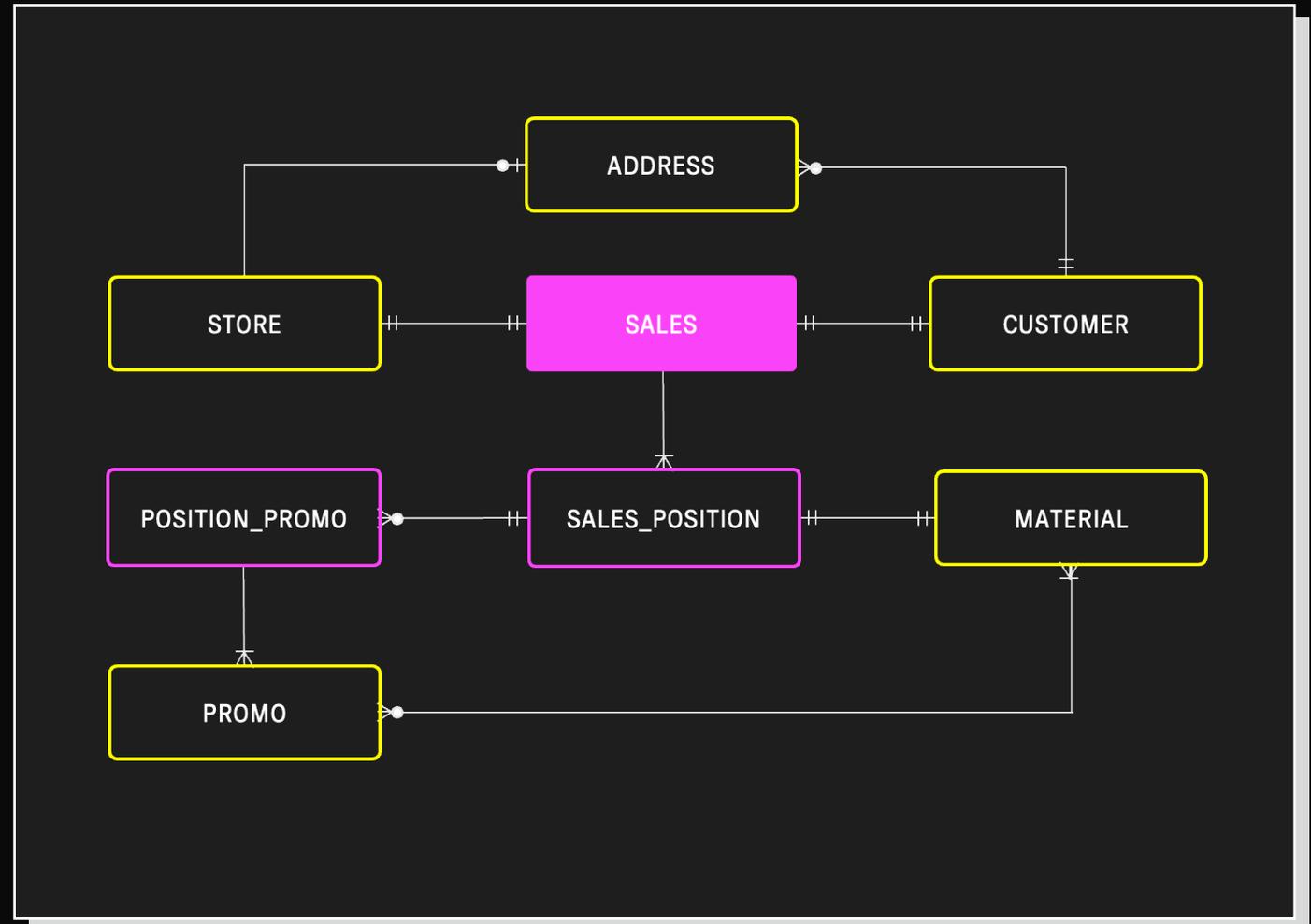


1	link_sales_x_...	???
2	link_position_x_...	???
3	link_position_promo_x_...	???

# Выделяем link

1. Изменение кардинальности связи не порождает изменения модели ядра хранилища
2. Накопление истории связи и ее переиспользование в отрыве от остальных атрибутов (нормализация)

1	link_sales_x_...	???
2	link_position_x_...	???
3	link_position_promo_x_...	???



# Fact Satellite

Тип объекта компонента – Raw Vault, предназначенный для хранения набора атрибутов и связей сущности, находящейся на грани между измерением и фактом

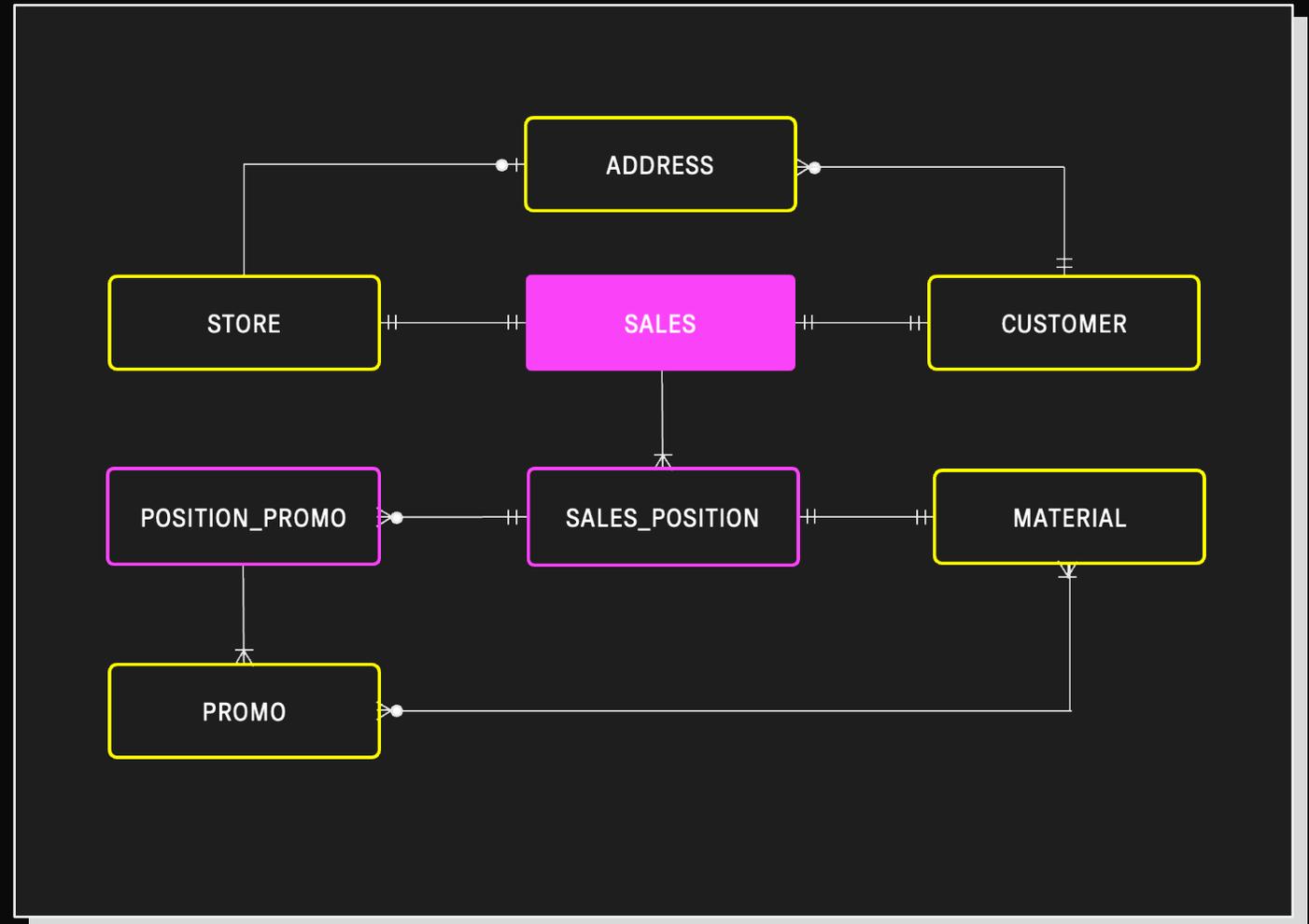
## Выполнение всех условий:

- 01 по бизнес-смыслу сущность является событием (заказ, сделка, чек)
- 02 приблизительный объем данных > 1 млн записей в день
- 03 связи-сущности, включенные в объект, не несут бизнес-смысла без атрибутов
- 04 соотношение PK сущности к FK имеет кардинальность 1-1 или n-1 и не может измениться без изменения бизнес-процесса и порождения другой сущности



# Выделяем link

1	link_address_x_store	✓
2	link_address_x_customer	✓
3	link_material_x_promo	✓



формат истории  
и РИТ публицы

# Satellite-формат истории

Полуоткрытый интервал

`valid from`

Закрытый интервал

`valid from + valid to`



# Satellite-формат истории

Полуоткрытый интервал

`valid from`

Закрытый интервал

`valid from + valid to`

## Pros

- загрузка only insert
- нет лишних операций в тракте

# Satellite-формат истории

Полуоткрытый интервал

`valid from`

Закрытый интервал

`valid from + valid to`

## Pros

- загрузка only insert
- нет лишних операций в тракте

## Cons

- неюзабельно без специального вида таблиц или оконных функций в каждом запросе

# Satellite-формат истории

Полуоткрытый интервал

`valid from`

## Pros

- загрузка only insert
- нет лишних операций в тракте

Закрытый интервал

`valid from + valid to`

## Pros

- прозрачно для использования без дополнительных операций (в теории)

## Cons

- неюзабельно без специального вида таблиц или оконных функций в каждом запросе

# Satellite-формат истории

Полуоткрытый интервал

`valid from`

## Pros

- загрузка only insert
- нет лишних операций в тракте

## Cons

- неюзабельно без специального вида таблиц или оконных функций в каждом запросе

Закрытый интервал

`valid from + valid to`

## Pros

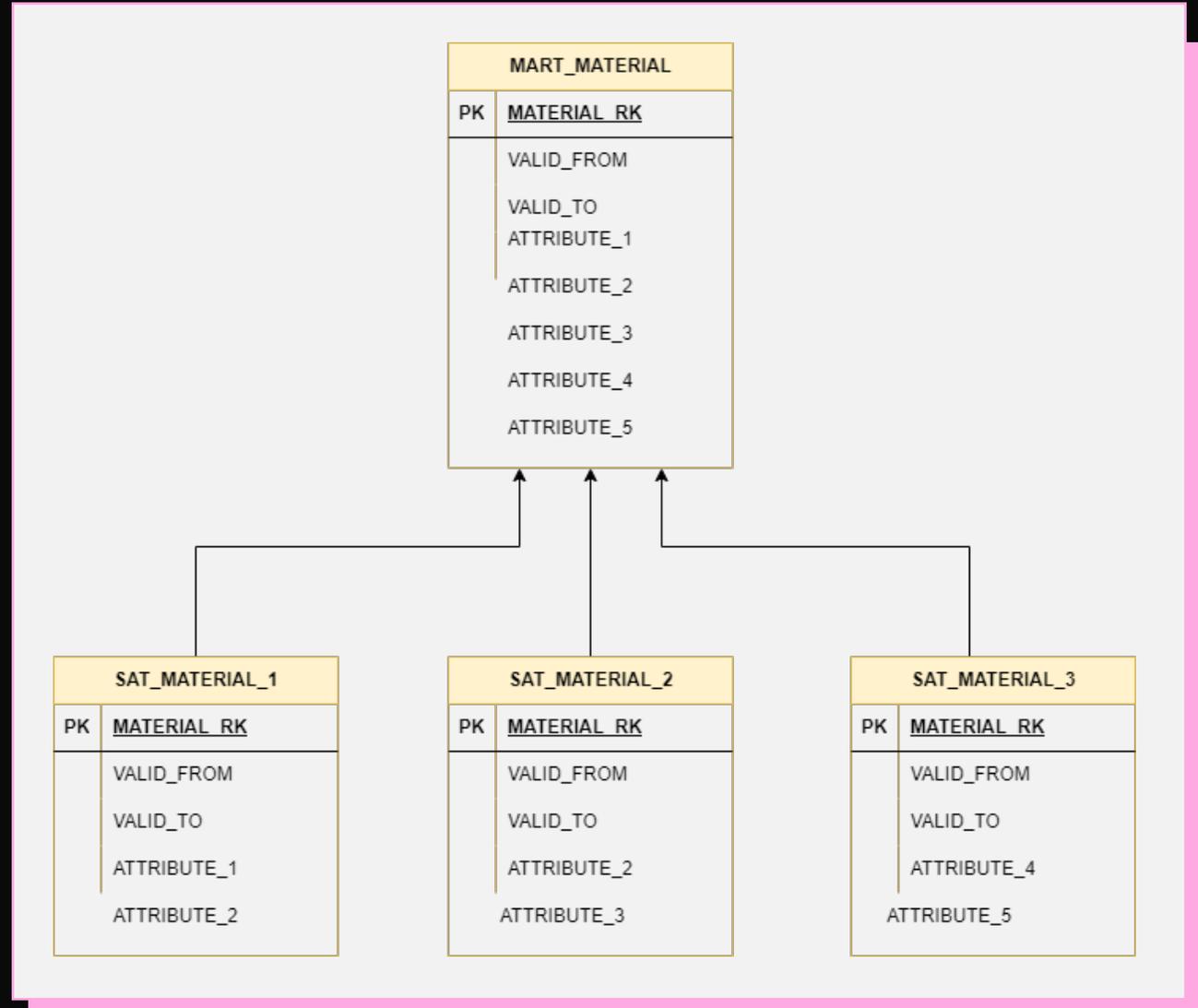
- прозрачно для использования без дополнительных операций (в теории)

## Cons

- метод загрузки U/I
- наличие двойных преобразований в тракте загрузки

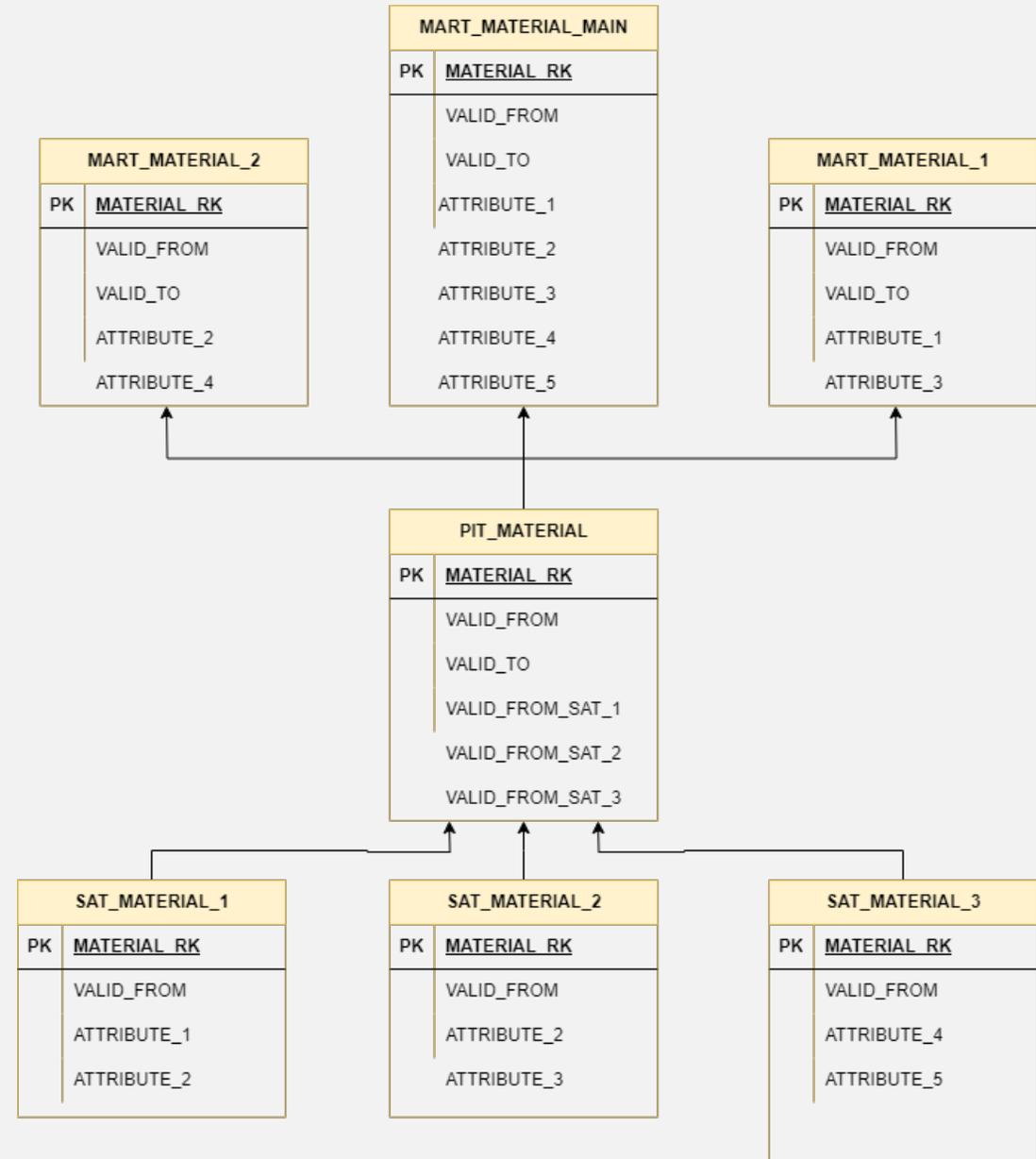
# Лишние преобразования

```
Select
  material_rk
  ,valid_from
  ,lead(valid_from) ... as valid_to
from
  (Select
    material_rk
    ,valid_from
  from sat_1
  union all
  select
    material_rk
    ,valid_from
  from sat_2
  ...) sat_union
```



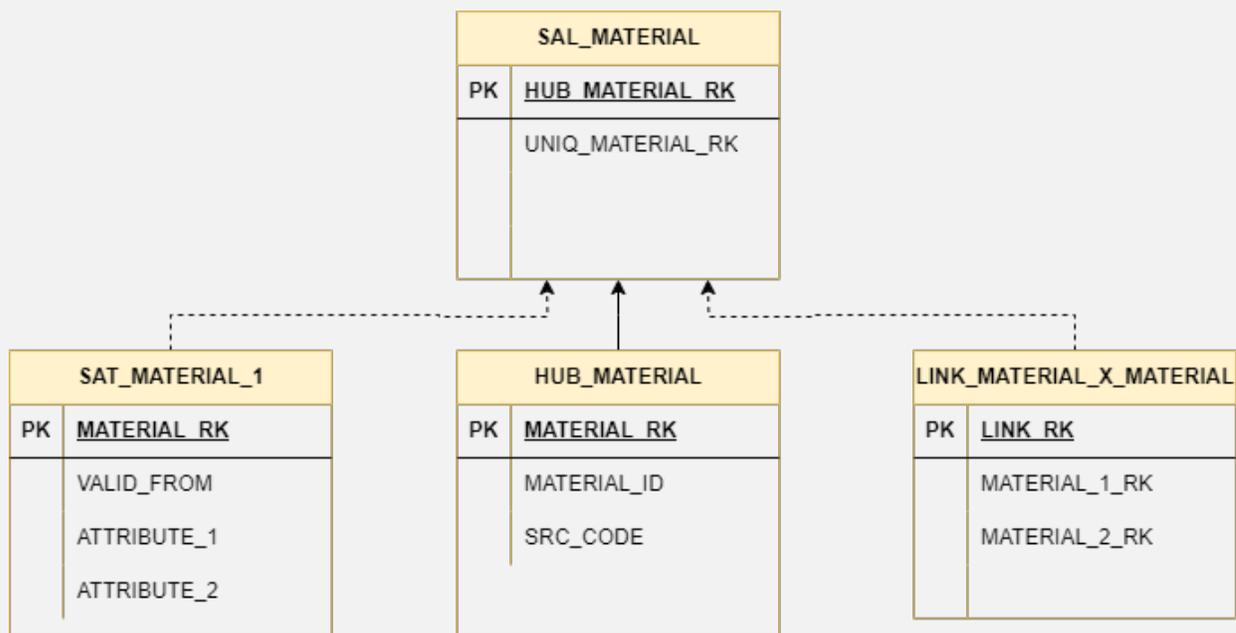
# PIT table

```
Select
  material_rk
, valid_from
, lead(valid_from) ... as valid_to
from
  (Select
    material_rk
    , valid_from
  from sat_1
 union all
  select
    material_rk
    , valid_from
  from sat_2
 ...) sat_union
```



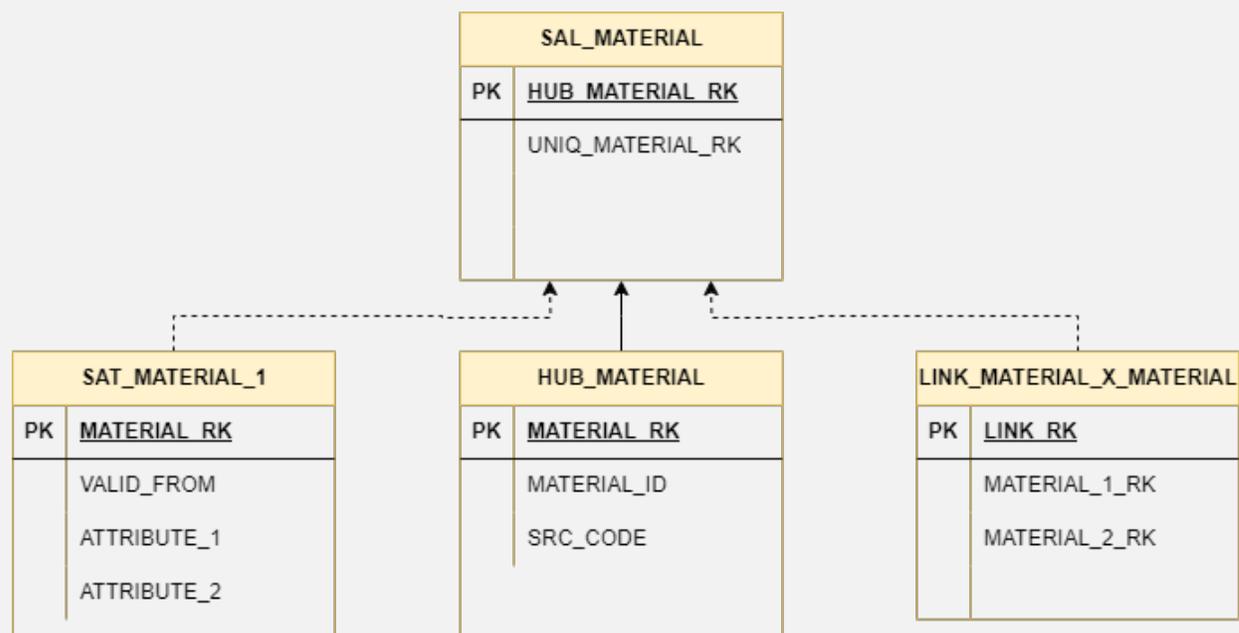
унификация  
и расчёты

# SAL унификация ключей



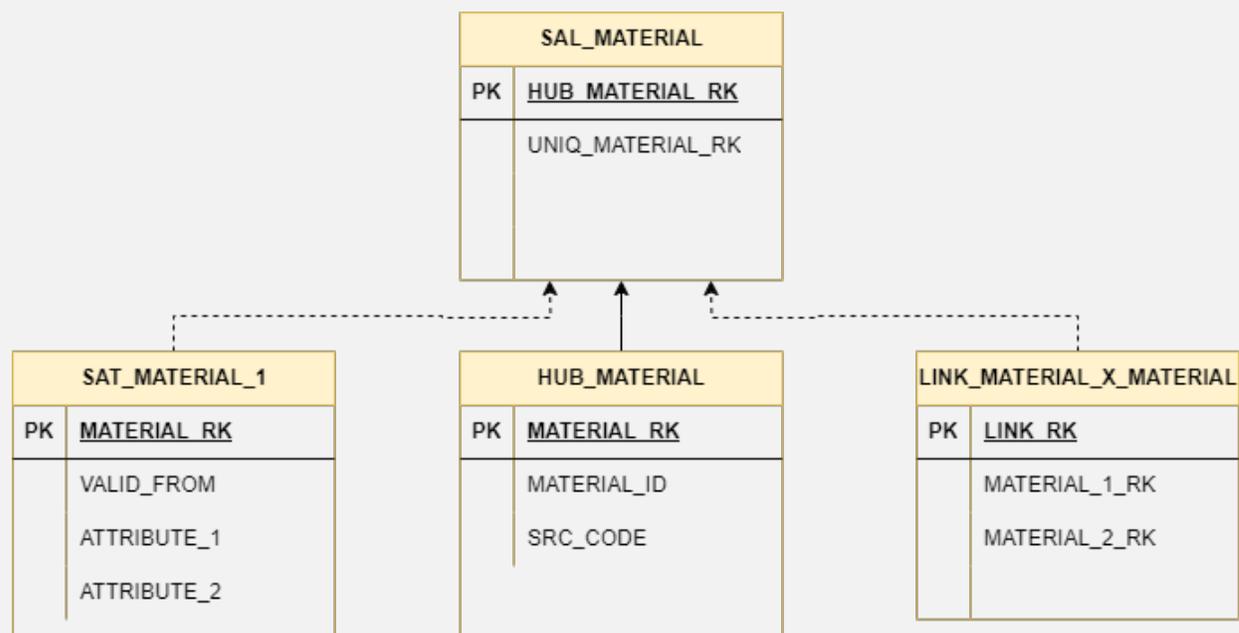
- механическая унификация  
 $rk = f(id \# src\_code)$
- `first_value(rk) over`  
(partition by id order by  
\*дата, время загрузки\*)

# SAL унификация ключей



- механическая унификация  $rk = f(id \# src\_code)$
- `first_value(rk) over (partition by id order by *дата, время загрузки*)`
- унификация через link разных RK из разных типов ID

# SAL унификация ключей



- механическая унификация  
 $rk = f(id \# src\_code)$
- `first_value(rk) over`  
(partition by id order by  
\*дата, время загрузки\*)
- унификация через link  
разных RK из разных типов ID
- логическая унификация  
(корнер-кейс для DWH)

# Расчёты

[01] Унификация алгоритма расчёта  
в одном объекте

[02] Переиспользование логики

# Расчёты

[01] Унификация алгоритма расчёта  
в одном объекте

[02] Переиспользование логики

## Структура:

- PK = Uniq\_entity\_rk
- valid\_from or valid\_from + valid\_to or no history?

# Расчёты

[01] Унификация алгоритма расчёта  
в одном объекте

[02] Переиспользование логики

## Структура:

- PK = Uniq\_entity\_rk
- valid\_from or valid\_from + valid\_to or no history?

## Резюме:

- ✓ у нас есть PIT = можно оставить только valid\_from

# Расчёты

[01] Унификация алгоритма расчёта  
в одном объекте

[02] Переиспользование логики

## Структура:

- PK = Uniq\_entity\_rk
- valid\_from or valid\_from + valid\_to or no history?

## Резюме:

- ✓ у нас есть PIT = можно оставить только valid\_from
- ✓ чтобы соединить N таблиц для расчетов, мы уже строили полную историю, может оставить valid\_to?

# Расчёты

[01] Унификация алгоритма расчёта  
в одном объекте

[02] Переиспользование логики

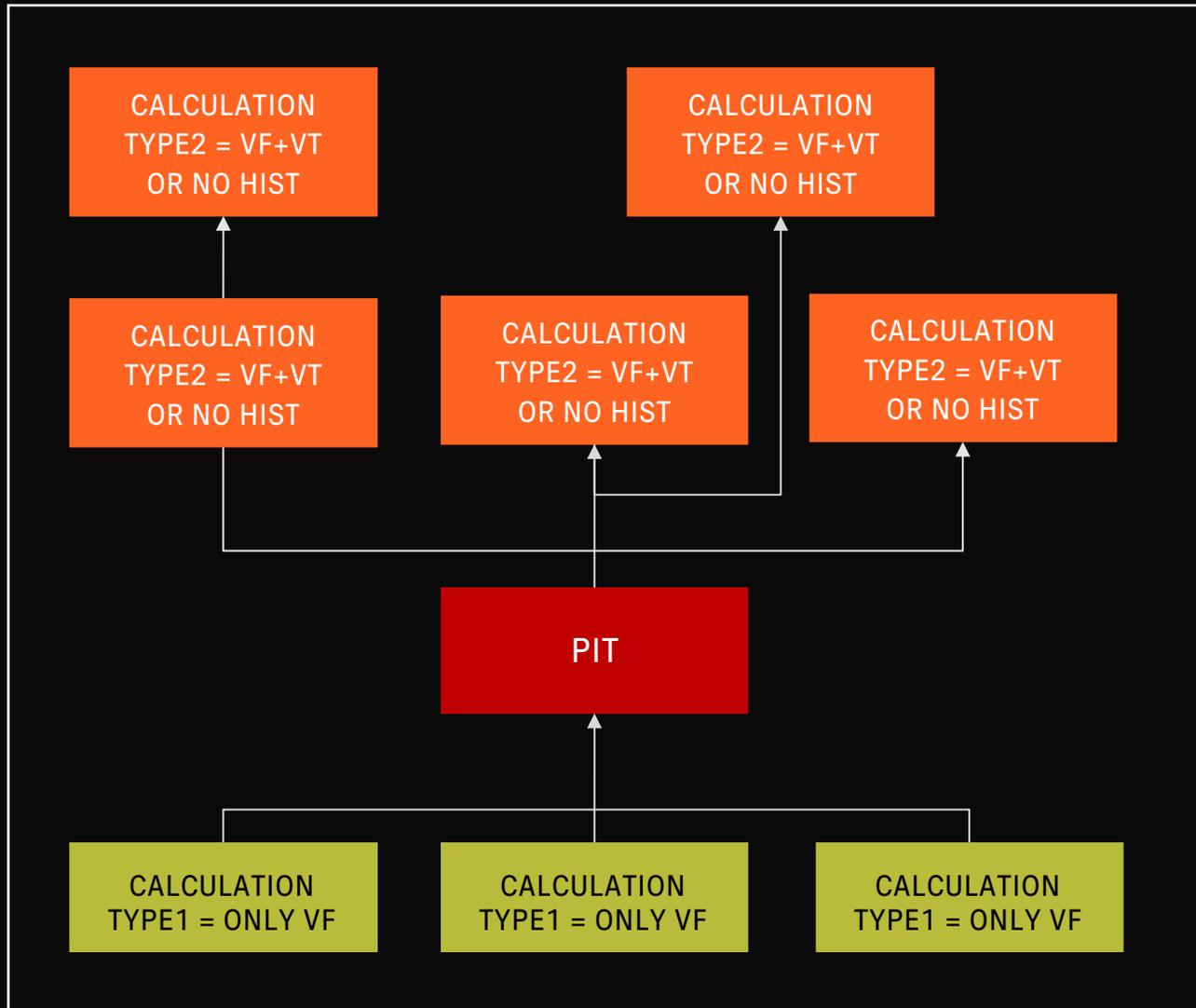
## Структура:

- PK = Uniq\_entity\_rk
- valid\_from or valid\_from + valid\_to or no history?

## Резюме:

- ✓ у нас есть PIT = можно оставить только valid\_from
- ✓ чтобы соединить N таблиц для расчетов, мы уже строили полную историю, может оставить valid\_to?
- ✓ расчет может быть на транзакционных данных / содержать только определенный статус / etc = потребности в истории вообще может не быть

# Принцип разбиения видов расчётов



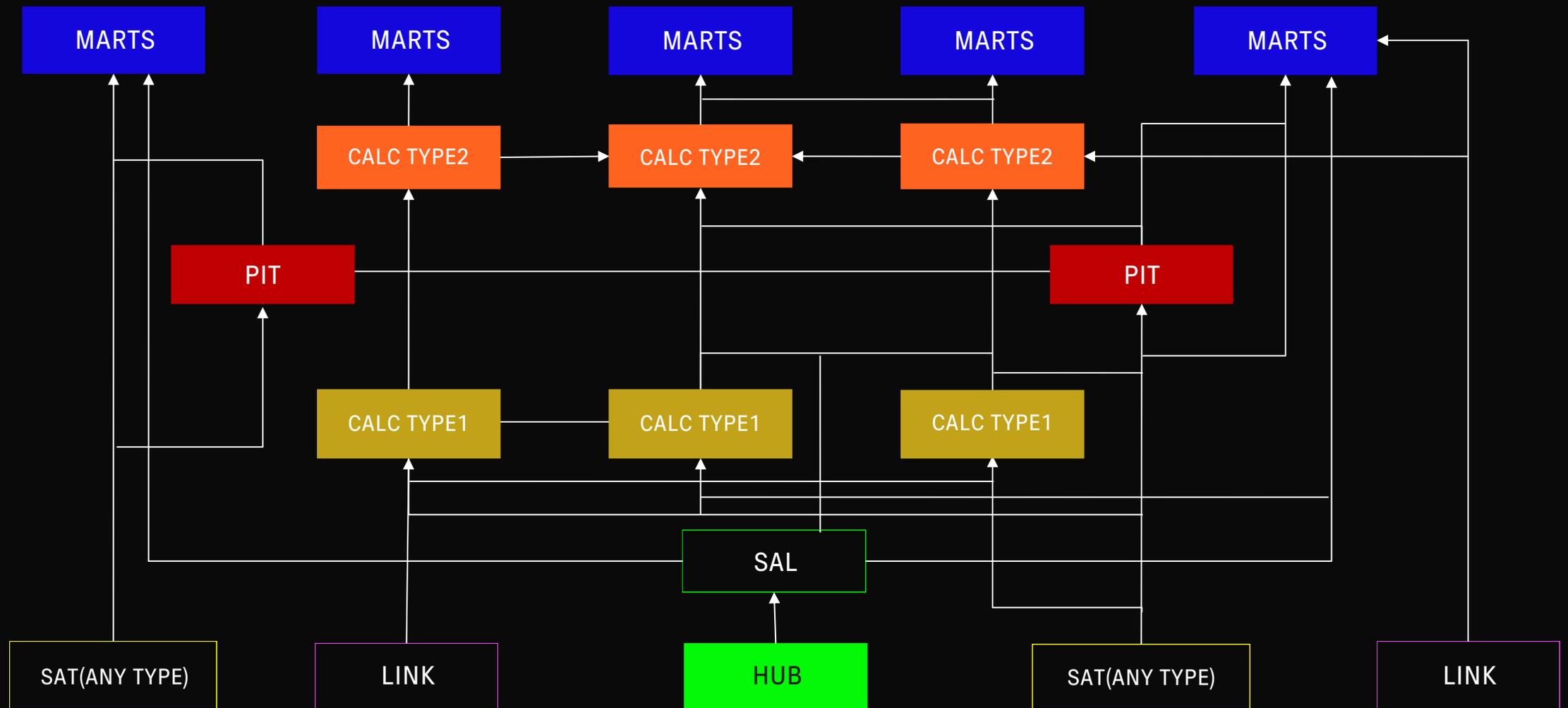
## Calc type 1

- строго только `valid_from`
- идет на вход в PIT
- не может строиться на объектах типа `calc`

## Calc type 2

- `valid_from + valid_to` или не историчен
- идет на выход из PIT или напрямую из RV, если возможно
- может строиться на любом объекте типа `calc`

# Схема полного тракта построения DWH



# Вопросы?

Денис Лукьянов

Руководитель направления  
архитектуры данных в esom.tech

DELUKYANOV@SAMOKAT.RU