

Модели данных

В4. Универсальные модели Data Vault



Московский государственный технический университет
имени Н.Э. Баумана

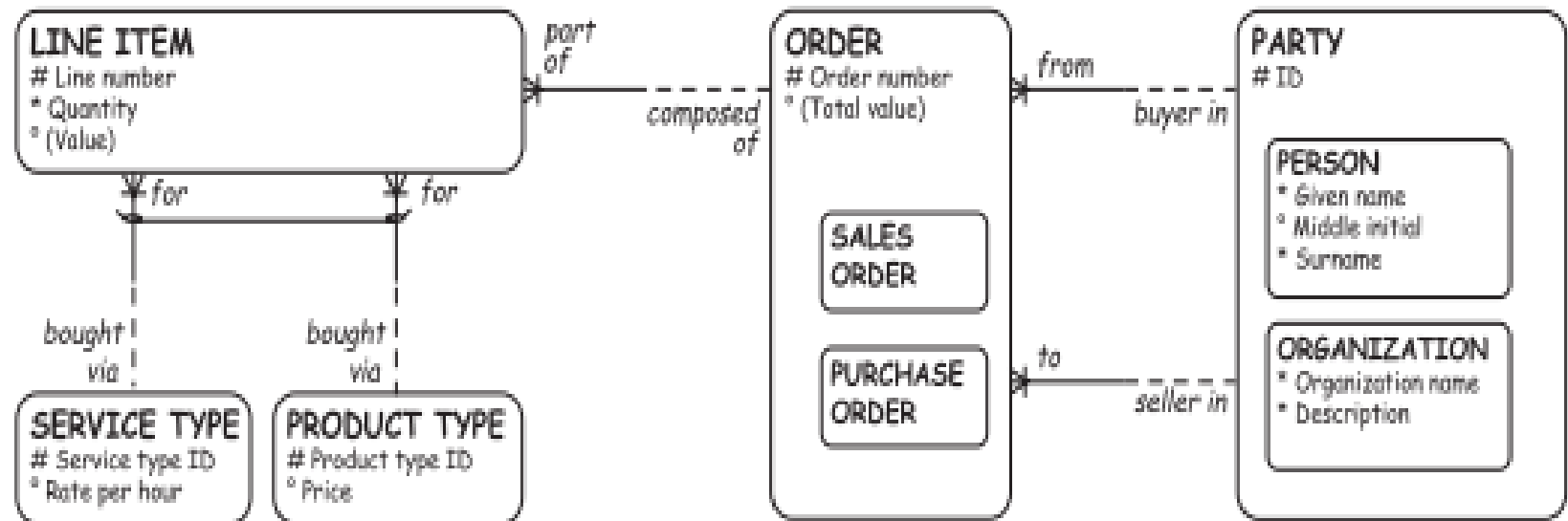
Факультет ИБМ

Июль 2024 года

Москва

Артемьев Валерий Иванович © 2024

Моделирование предметных областей



В4. Универсальные модели данных

Data Vault

- Что такое универсальная модель данных Data Vault?
- Сценарии применения (проектирование ХД, отраслевые модели ХД)
- Основные элементы модели (hub, link и satellite)
- Элемент Hub – идентификатор сущности
- Элемент Link – представление связей между сущностями
- Элемент Satellite – атрибуты сущностей и связей
- Правила создания модели

Актуализировать,
когда устоится

Что такое универсальная модель данных Data Vault?

Техника физического моделирования данных, которая помогает спроектировать гибкое и масштабируемое корпоративное хранилище данных.

Сочетает преимущества 3NF и схемы «звезда» ROLAP.

Представляет собой набор нормализованных таблиц:

- детализированный,
- исторически отслеживаемый,
- уникально связанный,
- поддерживающий функциональные области бизнеса.

Организует данные вокруг бизнес-процессов и функций (функционально ориентирован), а не отдельных предметных областей. Бизнес-ключи имеют горизонтальный характер, что позволяет интегрировать данные из различных частей организации и предоставляет целостное видение бизнеса.

Сценарии применения Data Vault

Хранилища данных (Data Warehouse)

- реляционные базы данных SQL
- массивно-параллельные базы данных NewSQL

Аналитические хранилища данных

- озёра данных (Data Lake) на основе NoSQL в среде Big Data
- динамические хранилища данных (Dynamic DW)
- разведывательный анализ и Data Mining

Управление взаимоотношениями с клиентами (CRM)

Финансовый аудит

The diagram illustrates the Data Vault 2.0 architecture, organized into three main layers:

- Уровень корпоративного хранилища данных (слой DDS), модель Data Vault 2.0:** This central layer contains the core data storage components.
 - Enterprise Data Warehouse:** The primary storage layer, containing:
 - Business Vault:** The central hub for data, divided into **Operational Vault** and **Metrics Vault**.
 - Staging:** A **Staging Area (relational)** that receives data from sources via **(batch)** or **(real-time)** paths.
 - Enterprise BI Solution:** The layer for data delivery and analysis, containing:
 - Information Delivery:** Includes **Star Schemas Cubes**, **Report Collection**, and **Complex Business Rules**.
 - Data Marts:** **Meta Mart**, **Metrics Mart**, and **Error Mart**.
- Область подготовки данных (слой Staging/ODS):** This layer, which collects unprocessed data from sources, is represented by the **Staging Area (relational)** and the **NoSQL** database.
- Уровень представления (слой Data Marts & Information Marts):** This layer, which presents data through analytical storefronts like star schemas, is represented by the **Information Delivery** and **Data Marts** components.

Data flow is shown from external sources (Sales, Finance, Contracts) through the **SOA/ESB** and **Staging** area into the **Business Vault**, and finally to the **Information Delivery** layer for reporting and analysis.

Основные элементы модели Data Vault

Простота конструкции и гибкость схемы данных за счёт минимума основных элементов:

1. **Хаб (Hub)** – *идентификатор* бизнес-сущности
2. **Связь (Link)** – поддержка *многосторонних и многосвязных отношений* бизнес-сущностей
3. **Спутник (Satellite)** – *контекстная информация* бизнес-сущностей и связей, представленная в *ретроспективе* (факты, метрики и описания) .

Хабы и Связи образуют стабильный «скелет» модели данных, Спутники добавляют необходимую «плоть».

Элемент Хаб (Hub) – идентификатор бизнес-сущности

Таблица, хранящая основное представление бизнес-сущности предметной области с функциональной позиции (Клиент, Продукт, Заказ и пр.).

Содержит следующие поля:

- *Уникальный и неизменный бизнес-ключ* – одно или несколько полей, идентифицирующих сущность (ИНН организации или VIN автомобиля).
- *Служебные поля:*
 - *первичный ключ*, рекомендуется хэш бизнес-ключа, сгенерированный с помощью MD5 или SHA-1
 - *время первоначальной загрузки сущности* в хранилище (load timestamp)
 - *источник данных* (record source) – название системы, базы или файла, откуда были взяты данные.

Определение Хабов и бизнес-ключей


Критерии определения хабов и их бизнес-ключей в моделировании Data Vault 2.0:

- **Уникальность и стабильность:** Бизнес-ключи должны быть уникальными и устойчивыми к изменениям, чтобы обеспечить точную идентификацию и отслеживание объектов.
- **Значимость для бизнеса:** Выбранные хабы и бизнес-ключи имеют важное значение для бизнес-операций и предоставляют критически важные данные для аналитики и отчётности.
- **Удобство интеграции:** Хабы и бизнес-ключи должны легко интегрироваться с другими системами и процессами, поддерживая консистентность данных и лёгкость доступа к информации.


Не рекомендуется использовать типы данных (плавающие числа или даты и времена), которые могут измениться при миграции или конверсии данных.

Примеры Хабов для представления бизнес-сущностей Продукт и Заказ

Продукт

hub_product		
	product_hash_key	binary
	load_date	timestamp
	record_source	varchar
	product_code	varchar

Заказ

hub_order		
	order_hash_key	binary
	load_date	timestamp
	record_source	varchar
	order_number	varchar

Элемент Связь (Link) – представление связей между бизнес-сущностями

Таблица, объединяющая несколько бизнес-сущностей связью «многие-ко-многим».




Содержит следующие поля:

- первичный ключ – хеш или составной ключ связываемых бизнес-сущностей
- внешние ключи связываемых сущностей
- дата и время загрузки данных (load timestamp)
- источник данных для записи (record source).


Адаптирует отношение «много ко многим» из 3NF и решает проблемы с масштабируемостью и гибкостью.
Не допустимы ссылки между элементами Link.

Пример Связи Строка заказа между бизнес-сущностями Продукт и Заказ


Строка заказа

lnk_line_item		
	line_item_hash_key	binary
	load_date	timestamp
	record_source	varchar
	order_hash_key	binary
	product_hash_key	binary

Продукт

hub_product		
	product_hash_key	binary
	load_date	timestamp
	record_source	varchar
	product_code	varchar

Заказ

hub_order		
	order_hash_key	binary
	load_date	timestamp
	record_source	varchar
	order_number	varchar

Элемент Спутник (Satellite) – атрибуты бизнес-сущностей и связей

Таблица, хранящая контекстные данные бизнес-сущности или связи и их версии.

Содержит следующие поля:

- внешний ключ родительской Хаба или Связи
- набор атрибутов (фактов, метрик и описаний) Хаба или Связи
- дата начала действия (Start date)
- дата окончания действия (End date)
- метка активности (active_flag)
- метка удаления (deleted_flag)
- временная метка загрузки (load timestamp)
- источник данных (record source)
- хэш-слепок (HashDiff) всех бизнес-атрибутов, полученный с помощью MD5 или SHA-1, для упрощения их обновления

Выделение Спутников

Один источник данных – один Спутник

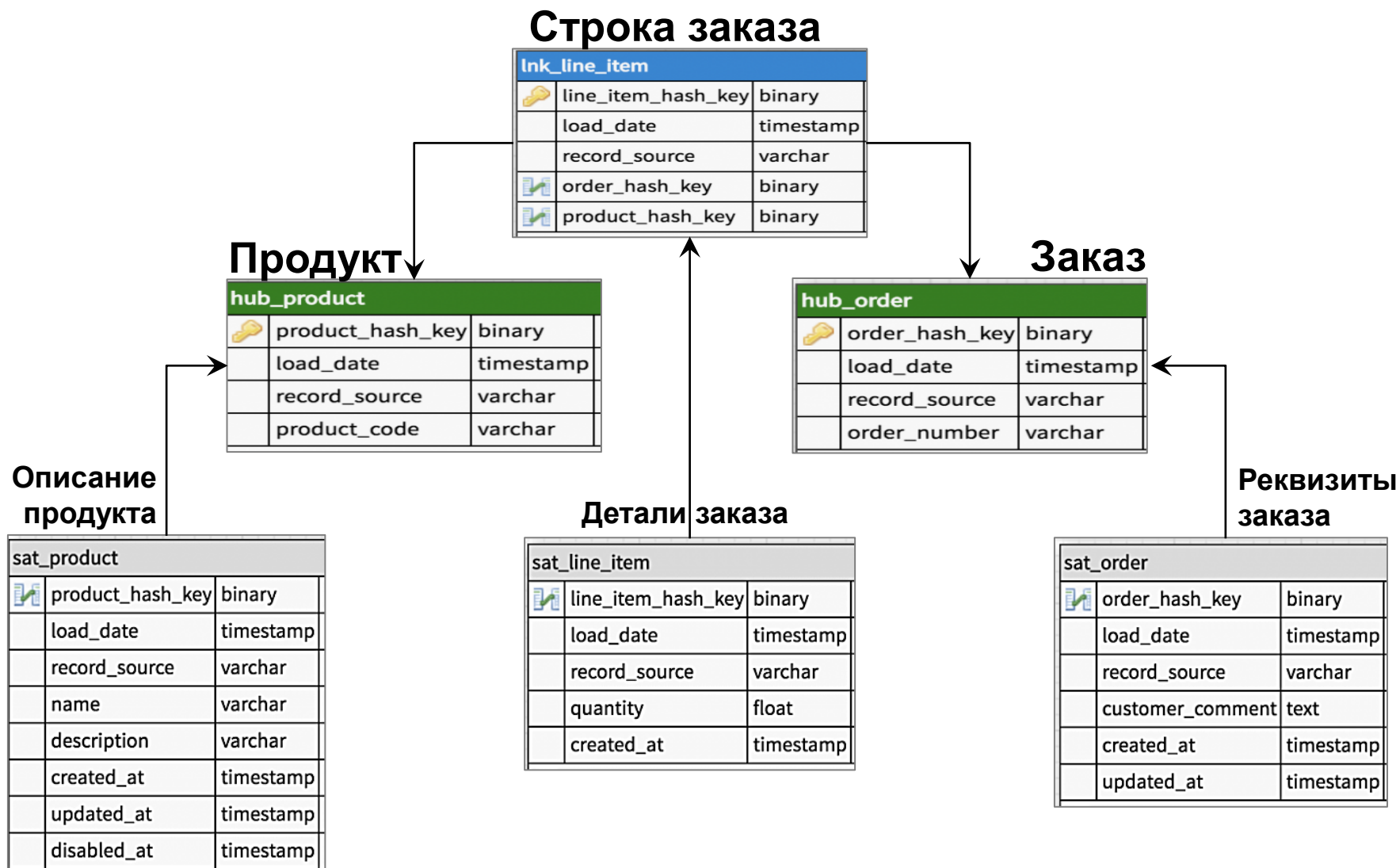
- Позволяет добавлять новые источники данных без изменения существующих Спутников
- Не надо изменять входные данные под существующую структуру.
- Сохраняет историю исходной системы
- Максимум параллелизма загрузки без конкуренции за ресурсы
- Обеспечивает интеграцию данных в реальном времени
- Не нужна одновременно готовность потоковых и пакетных данных

Один уровень детальности – один Спутник

Рекомендуется дополнительно разделять данные по уровню детальности или темпу изменения атрибутов.

Один спутник — для неизменных атрибутов, другой – для суточных данных, третий – для ежемесячных данных и т. д.

Пример Спутников для атрибутов Продукта и Заказа, Строки заказа



Последовательность создания модели Data Vault

1. Определение Хабов на основе бизнес-сущностей и их использования в предметной области.
2. Выявление Связей через возможные отношения между бизнес-ключами и понимание их контекста.
3. Определение Спутников – моделирование контекста каждой бизнес-сущности и транзакции (Связи), соединяющей Хабы.
4. Моделирование point-in-time таблиц, производных от Спутников.

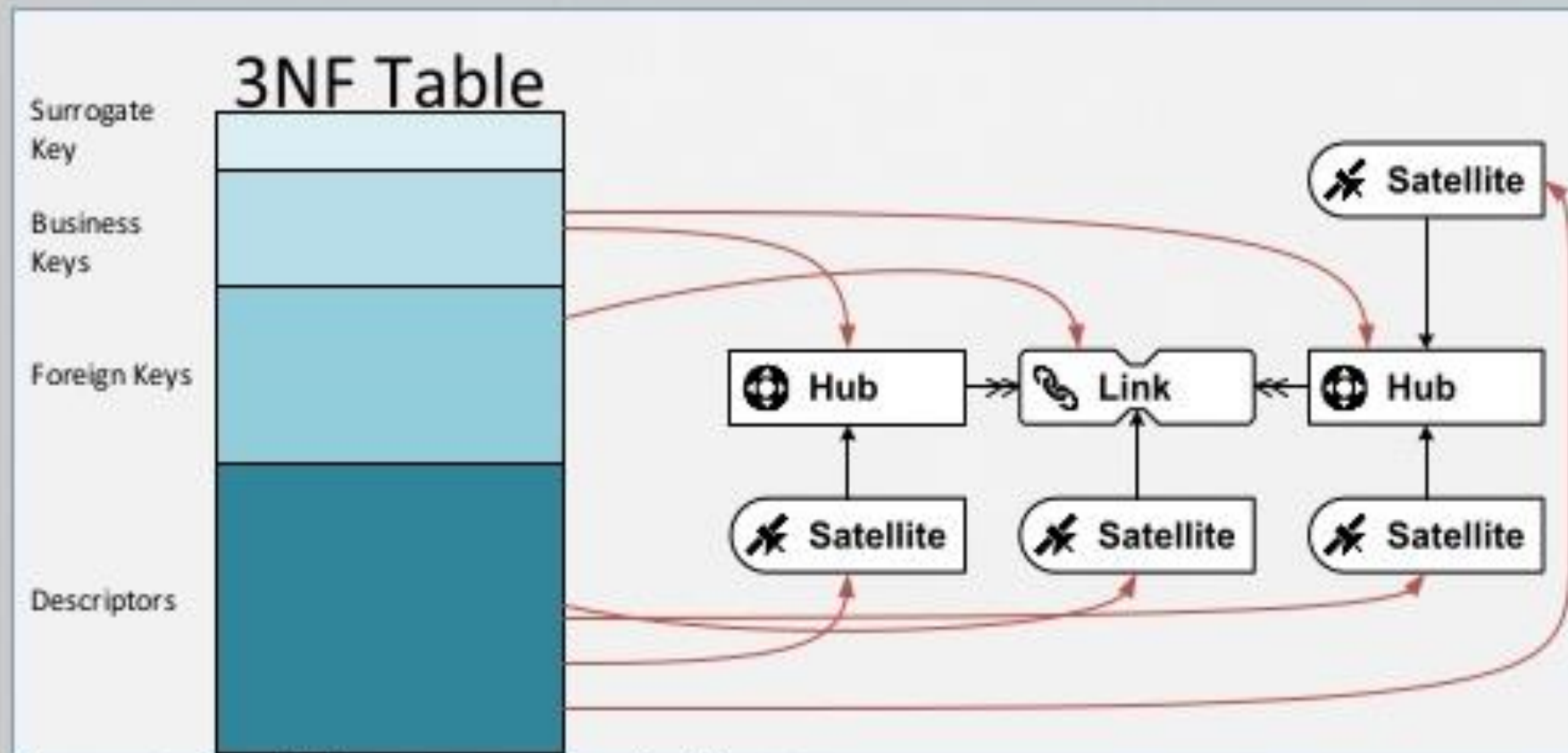
Правила создания модели Data Vault (1)

1. Бизнес-ключи и первичные ключи Хаба *никогда не меняются*.
2. Ключи Хабов не могут мигрировать в другие Хабы.
3. Ключи Хаба всегда мигрируют в Связи и дочерние Спутники.
4. Хабы связываются только с помощью Связей.
5. Связь должна связывать не менее двух Хабов.
6. Связь не может связываться с другими Связями.
7. Связи фиксируют отношения между элементами данных на наименьшем уровне детальности.

Правила создания модели Data Vault (2)

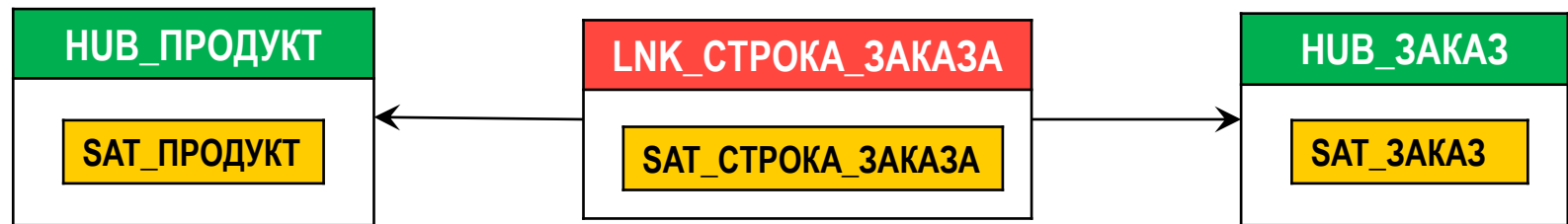
8. Спутник не может использовать суррогатный ключ.
9. Спутник связан только с Хабом или со Связью.
10. Хаб или Связь может иметь сколько угодно Спутников с различными наборами полей.
11. Спутник всегда содержит временную метку загрузки или ссылку на таблицу временных меток (календарь).
12. Спутники фиксируют только изменения без дублирования строк.
13. Данные распределяются по Спутникам на основе источника, уровня детальности и темпах изменения.
14. Для нескольких Спутников Хаба можно создать point-in-time таблицу, упрощающую запросы.

Миграция ключей и атрибутов в модель Data Vault

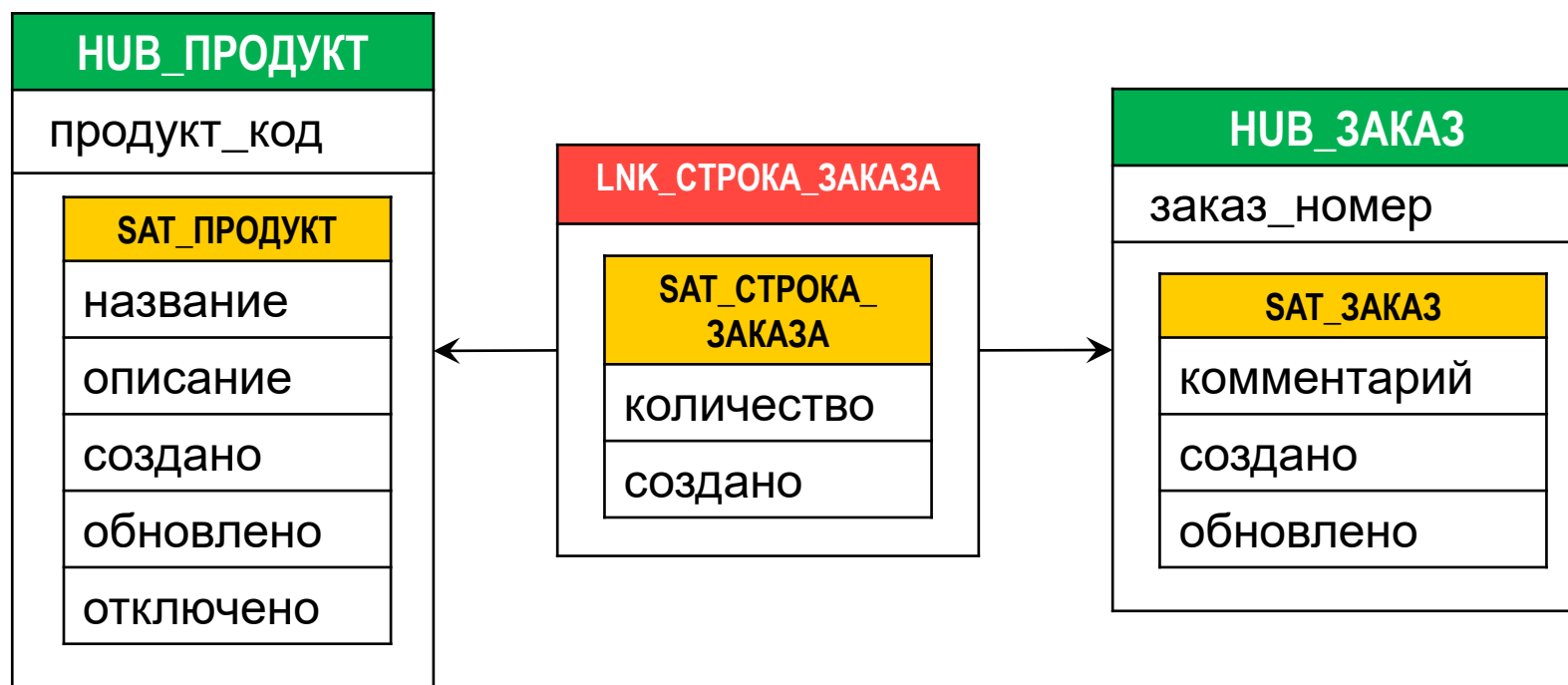


In accordance with its own representation Linstedt, 2014

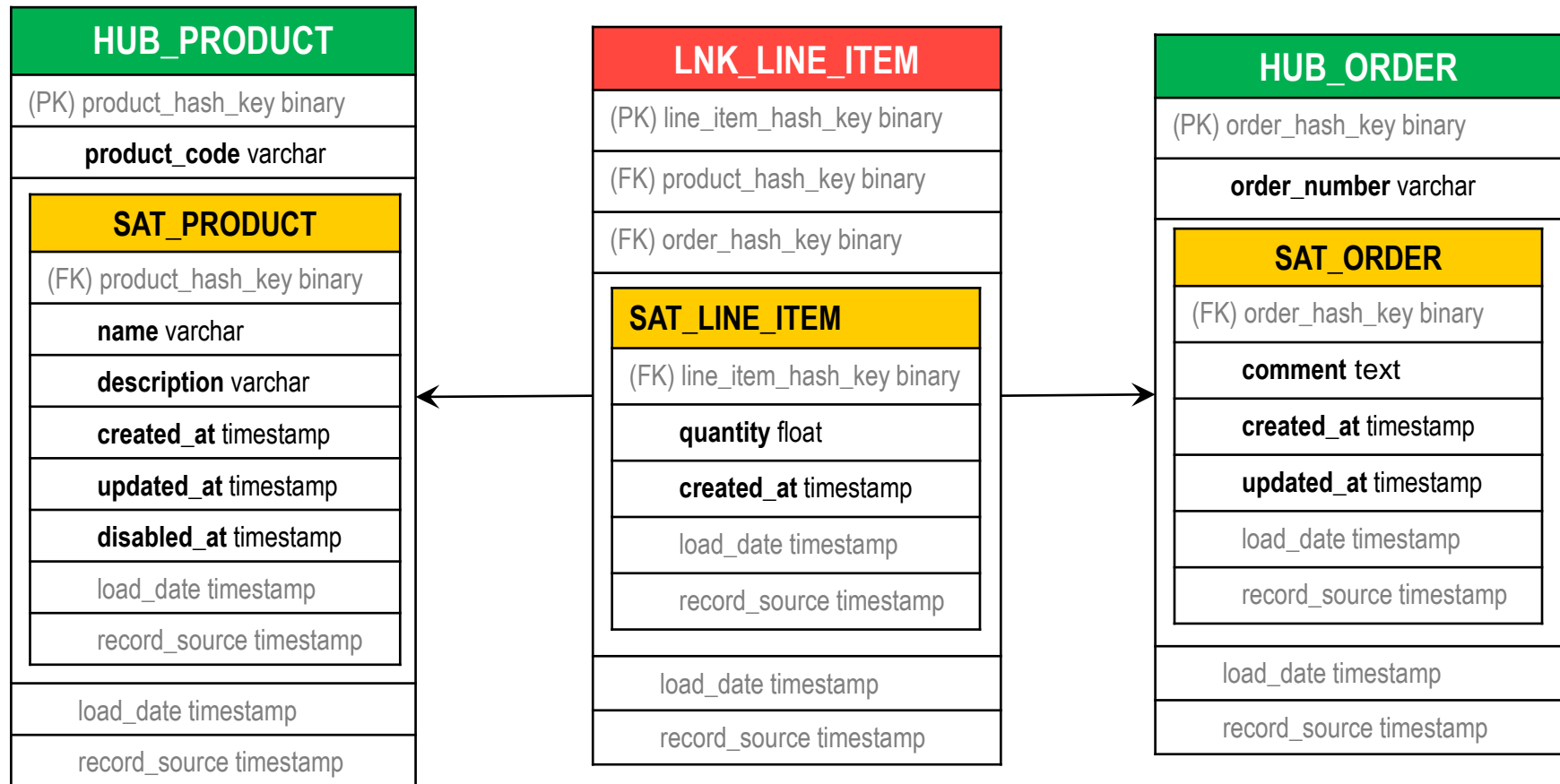
Пример концептуальной модели Data Vault с супертипами и подтипами



Пример логической модели Data Vault с супертипами и подтипами



Пример физической модели Data Vault с супертипами и подтипами



Достоинства модели Data Vault

Простота моделирования – для описания предметной области используется минимум элементов (хабы, ссылки и спутники),

Строгая система правил для описания взаимоотношений между элементами. Таблиц может быть в разы больше, чем в 3NF, но все достаточно просты и ETL становятся проще за счет однообразия.

Отсутствие избыточности данных – особенно важно в области Big Data. Спутники используются только для хранения изменений. Проще медленно меняющихся измерений SCD2, которые размножает данные.

Расширяемость модели – по мере необходимости можно изменить структуру работающего КХД, добавить и сопоставить данные из новых источников. Удобная структура хранилища сырых данных позволяют сформировать витрину данных под любые требования бизнеса.

Максимальное распараллеливание загрузки данных в хранилище за счёт максимальной независимости элементов.

Поддержка Agile-принципов – новые данные подключаются к существующей модели без модификации её структуры.

Недостатки модели Data Vault

Снижение производительности из-за большого числа операций соединения данных из разных таблиц. Запросы могут выполняться медленнее, чем в денормализованных схемах «Звезда» и в Apache Hive, замедляя MapReduce.

Обязательное наличие витрин данных, т.к. схема Data Vault не слишком хорошо подходит для прямых запросов в сырые и бизнес-ориентированные данные (Raw Vault и Business Vault).

Недостаток обучающих материалов по концепции Data Vault ощущается на практике.

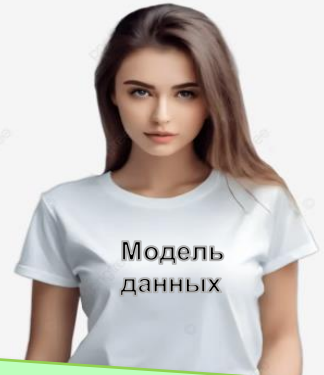
Потребность в высококвалифицированных архитекторах (разработчиках) Data Vault до сих пор актуальна.

Модель Data Vault ориентирована на физический уровень.

Модель не подходит для регуляторов, т.к. не предполагает контроль и очистку данных на входе в хранилище – обеспечивает *единственный источник факта*.

Спасибо за внимание!

Терпения и удачи всем, кто связан с моделями данных



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