**Data Vault**

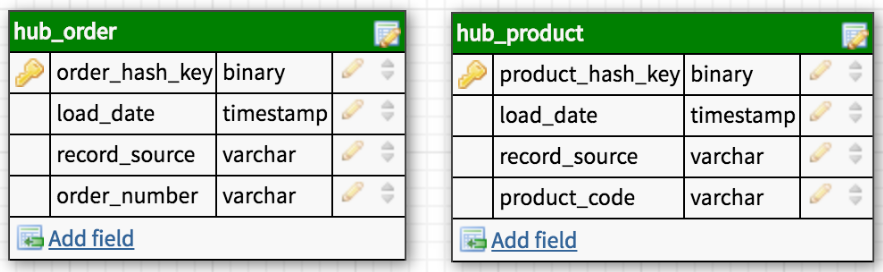
Архитектура DWH?

Data Vault — гибридный подход, объединивший достоинства знакомой многим схемы «звезды» и 3-ей нормальной формы. Впервые эта методология была анонсинована в 2000 году Дэном Линстедтом (Dan Linstedt). Подход был придуман в процессе разработки хранилища данных для Министерства Обороны США и хорошо себя зарекомендовал. Позже, в 2013 году, Дэн анонсировал версию 2.0, доработанную с учетом быстро набравших популярность технологий (NoSQL, Hadoop) и новых требований, выставляемых к DWH.

Data Vault состоит из трех основных компонентов — **Хаб (Hub)**, **Ссылка (Link)** и **Сателлит (Satellite)**.

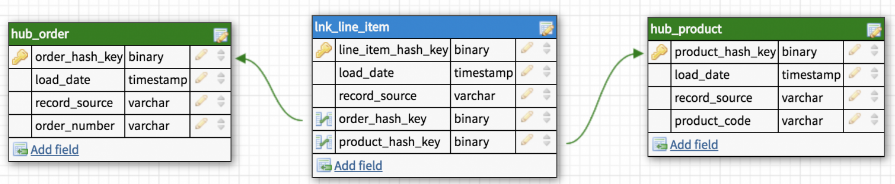
**Hub**

Хаб — основное представление сущности (Клиент, Продукт, Заказ) с позиции бизнеса. Таблица-Хаб содержит одно или несколько полей, отражающих сущность в понятиях бизнеса. В совокупности эти поля называются «бизнес ключ». Сгенерированный системой ID будет наихудшим вариантом. Бизнес ключ всегда должен быть уникальным и неизменным.  
Хаб так же содержит мета-поля *load timestamp* и *record source*, в которых хранятся время первоначальной загрузки сущности в хранилище и ее источник



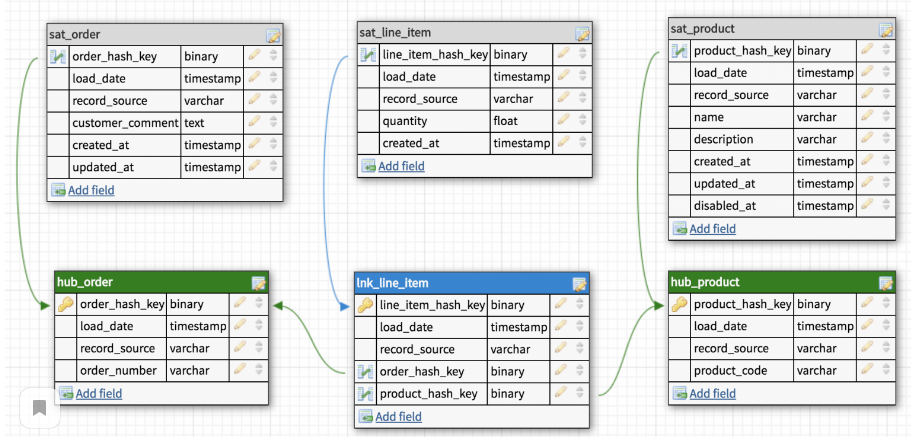
**Link**

Таблицы-Ссылки связывают несколько хабов связью многие-ко-многим. Она содержит те же метаданные, что и Хаб. Ссылка может быть связана с другой Ссылкой, но такой подход создает проблемы при загрузке, так что лучше выделить одну из Ссылок в отдельный Хаб.



**Satellite**

Все описательные атрибуты Хаба или Ссылки (контекст) помещаются в таблицы-Сателлиты. Помимо контекста Сателлит содержит стандартный набор метаданных (*load timestamp* и *record source*) и **один и только один** ключ «родителя». В Сателлитах можно без проблем хранить историю изменения контекста, каждый раз добавляя новую запись при обновлении контекста в системе-источнике.



**[+] Гибкость и расширяемость.**  
С Data Vault перестает быть проблемой как расширение структуры хранилища, так и добавление и сопоставление данных из новых источников. Максимально полное хранилище «сырых» данных и удобная структура их хранения позволяют нам сформировать витрину под любые требования бизнеса, а существующие решения на рынке СУБД хорошо справляются с огромными объемами информации и быстро выполняют даже очень сложные запросы, что дает возможность виртуализировать большинство витрин.  
**[+] Agile-подход из коробки.**  
Моделировать хранилище по методологии Data Vault довольно просто. Новые данные просто «подключаются» к существующей модели, не ломая и не модифицируя существующую структуру. При этом мы будем решать поставленную задачу максимально изолированно, загружая только необходимый минимум, и, вероятно, наша временнáя оценка для такой задачи станет точнее. Планирование спринтов будет проще, а результаты предсказуемы с первой же итерации.  
**[–] Обилие JOIN'ов**  
За счет большого количества операций join запросы могут быть медленнее, чем в традиционных хранилищах данных, где таблицы денормализованы.  
**[–] Сложность.**  
В описанной выше методологии есть множество важных деталей, разобраться в которых вряд ли получится за пару часов. К этому можно прибавить малое количество информации в интернете и почти полное отсутствие материалов на русском языке.. Большой недостаток сложности это обязательное требование к наличию витрин данных, так как сам по себе Data Vault плохо подходит для прямых запросов.  
**[–] Избыточность.**