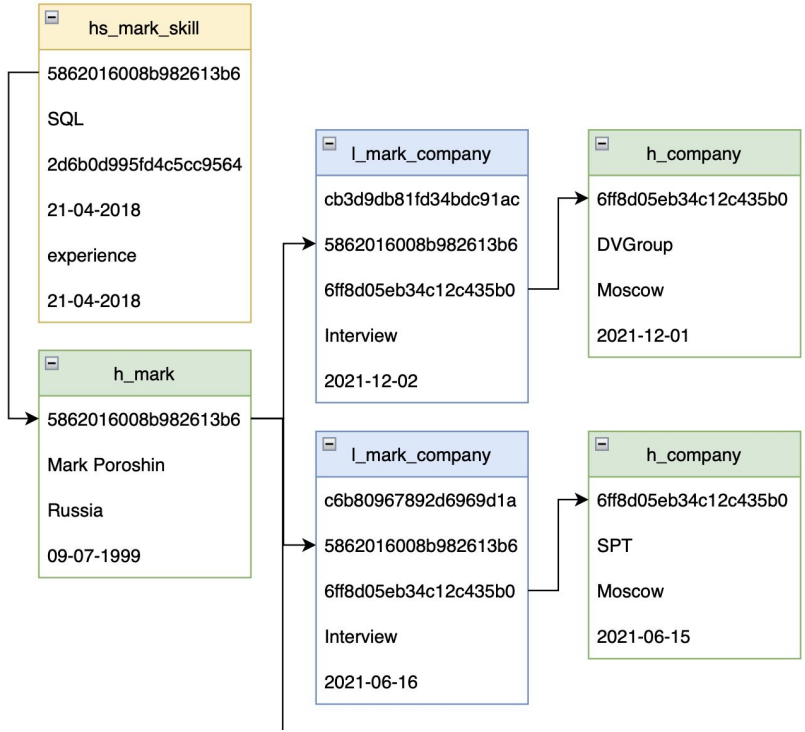
Построение DataVault с помощью DBT

Порошин Марк

Про себя

1. Занимаюсь DWH в компании spt
2. До этого строил Data Vault на таком же стеке в компании DVGroup
3. Создал dbt-ападтер для greenplum
4. Пилю форк проекта dbtvault с поддержкой greenplum



План доклада

1. Зачем вообще нужен этот доклад
2. Какую архитектуру используют в компании
3. Причины выбора конкретных технологий
4. Посмотрим какие есть методологии построения хранилища
5. Что такое DataVault 2.0 и чем он лучше DataVault
6. DBT, зачем нужно и как работает
7. Как с помощью DBT строить DataVault 2.0
8. Демо построения DataVault 2.0

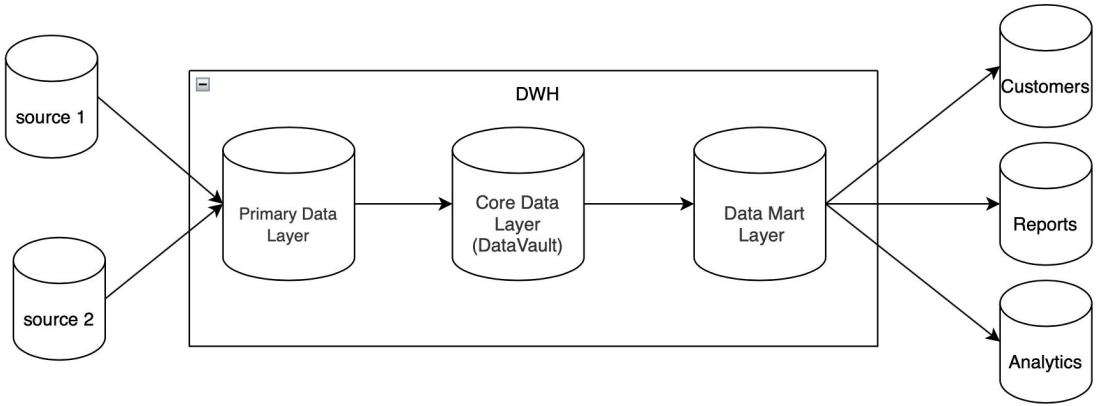
Два месяца, полет нормальный

Что сделано

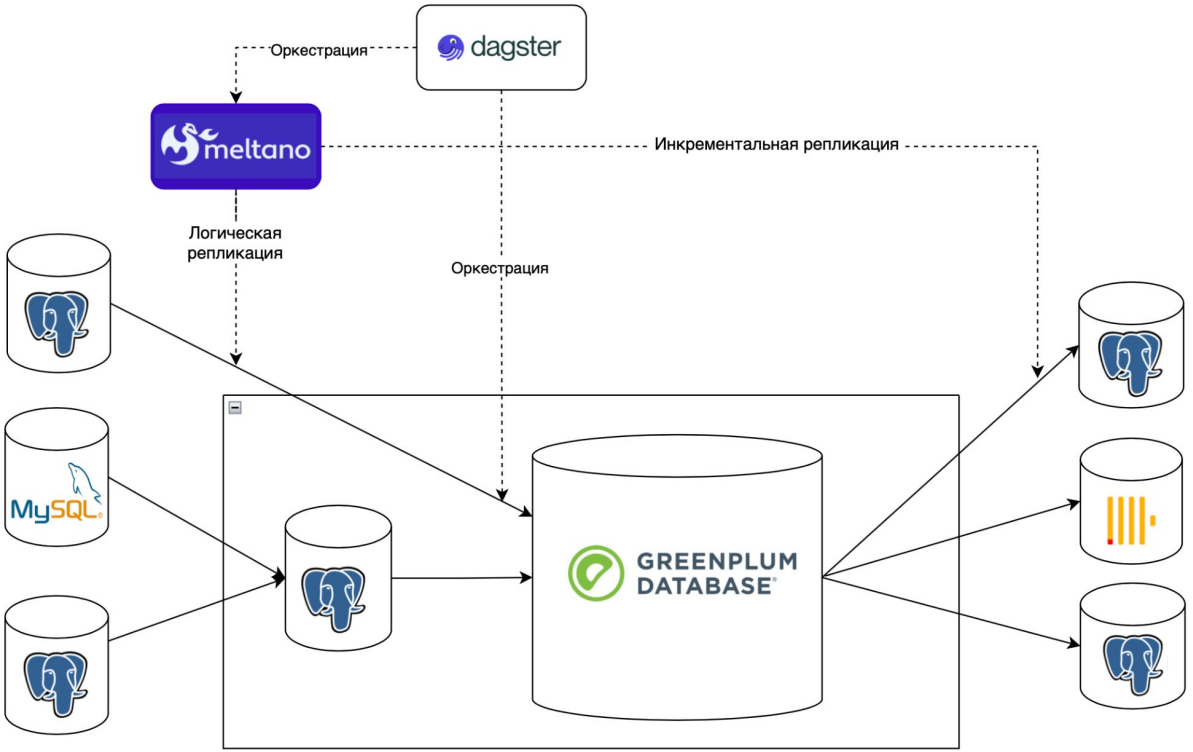
1. Команда из двух человек
2. Два месяца работы proof of concept
3. Два месяца разгребания backlog’a
4. По итогу реализован полный пайплайн (Репликация данных, Построение детального слоя, Оркестрация) + Витрины дошли до пользователей

Что мы хотим получить?

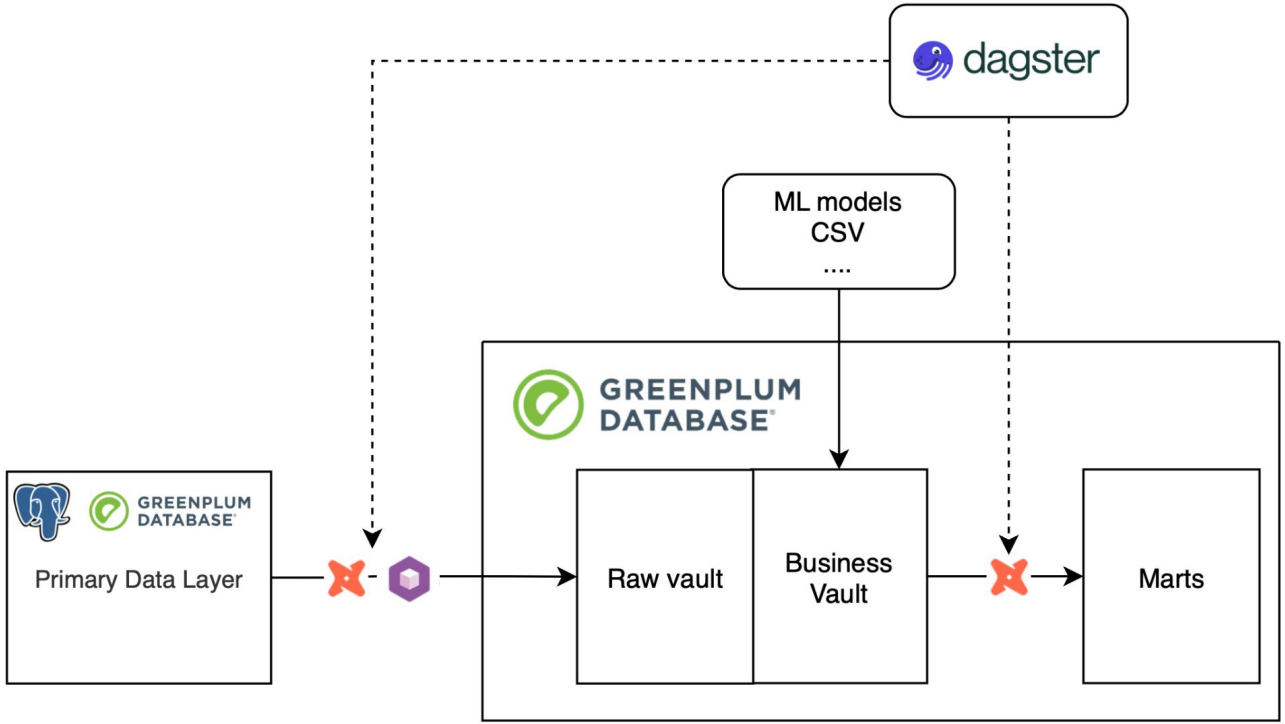
- Есть данные из разных источников
- Источники могут появляться в real-time
- Хотим единое хранилище с детальным слоем
- Хотим ELT
- Transform в базе данных



Конкретный кейс

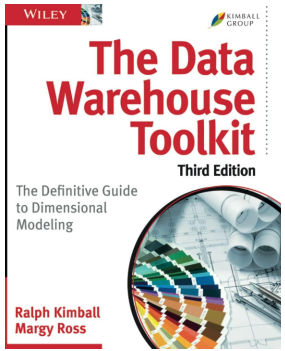


Внутри Greenplum



Какие существуют методологии для построения OLAP

1. Звезда (Эффективные запросы, страдает гибкость)
2. Снежинка (Эффективные хуже, чем у звезды, меньше дискового пространства, гибкость все еще страдает)
3. Data Vault (Больше join’ов, высокая гибкость)
4. Anchor Modeling (Максимальная нормализация, очень много join’ов)



Data Vault

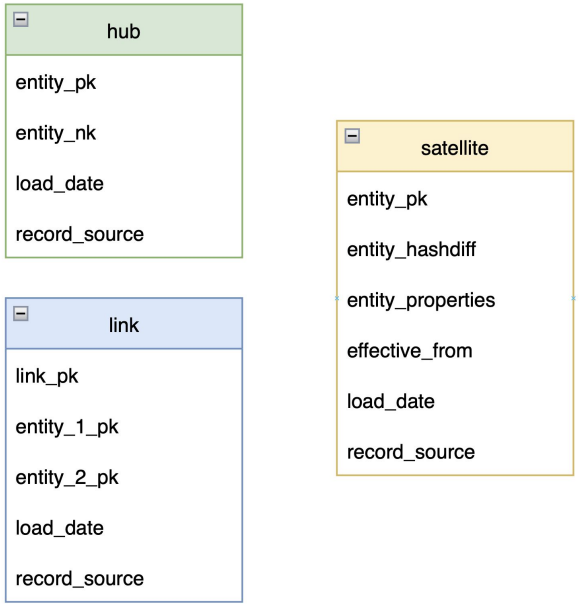
1. Гибкий
2. Простой
3. Поддерживает версионирование данных (scd2)
4. Модный
5. Уменьшаем дублирование данных за счет высокой нормализации данных
6. Меньше джоинов чем в 6й нормальной форме

Type 2 Slowly Changing Dimension

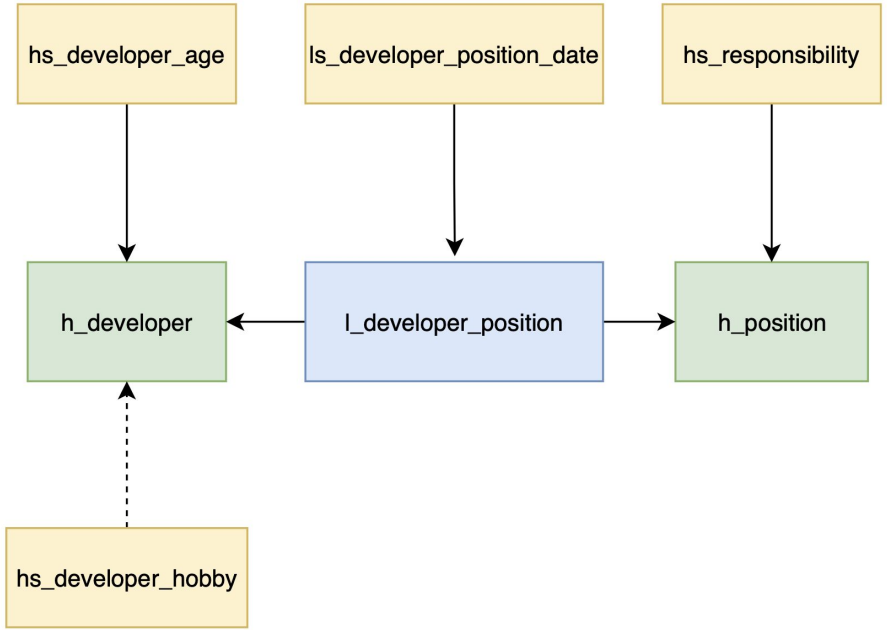
Product Dim (Source)		
Product Name	Product ID	Product Descr
12 inch box	012	12 inch glued box
10 inch box	010	10 inch ~~glued~~ box
		10 inch **pasted** box

Product Dim (Target)					
SID	Source Product ID	Product Name	Product Descr	EFF_ START_DT	EFF_ END_DT
0001	012	12 inch box	12 inch glued box	Jan-01-1753	Dec-31-9999
0002	010	10 inch box	10 inch glued box	Jan-01-1753	May-12-06
0003	**010**	**10 inch box**	**10 inch pasted box**	**May-12-06**	**Dec-31-9999**

Data Vault в картинках



Основные сущности



Пример

Data Vault vs Data Vault 2.0

Data Vault предполагает использование хешей в качестве суррогатного ключа вместо sequence'a. Это позволяет загружать hub, link и satellite параллельно. И суррогатный ключ может составляться из набора полей.

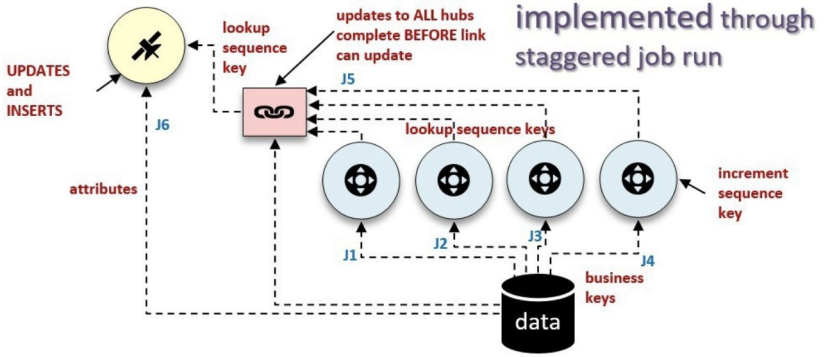


Схема загрузки Data Vault

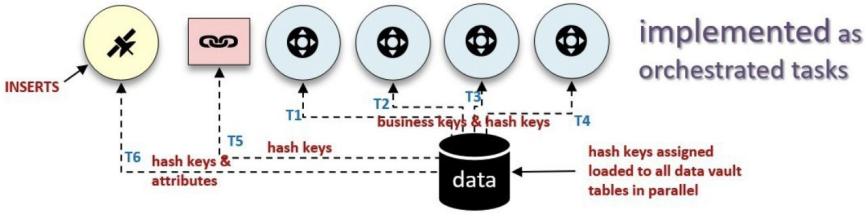


Схема загрузки Data Vault 2.0

Почему Greenplum

GREENPLUM DATABASE

Аргументы “за”

1. Нет альтернатив

Чего не хватает

1. Кластерный индекс
2. join elimination
3. Клиенты пг не поддерживают параллельную запись реплика+мастер

Почему Greenplum

GREENPLUM DATABASE®

Аргументы “за”

1. Нет альтернатив
2. Совместимость с клиентами Postgres
3. Масштабируется
4. Справляется с join’ами
5. Есть в yandex cloud
6. Колоночные таблицы
7. Сжатие
8. Партиционирование
9. Кейсы успешного внедрения

Чего не хватает

1. Кластерный индекс
2. join elimination
3. Клиенты пг не поддерживают параллельную запись реплика+мастер

Чем можно строить Data Vault

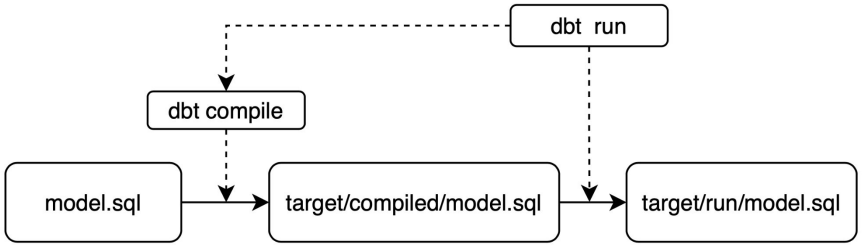
1. Написать на чистом sql
2. Написать свой фреймворк
3. DBT

DBT

1. Transform в ELT процессе
2. Jinja шаблонизация (параметризованные SQL запросы)
3. Решает за разработчика вопрос backup’а таблиц во время изменения таблицы
4. Уменьшаем дублирование кода, избавляемся от boilerplate
5. Пользовательские пакеты, например dbtvault, dbtexpectations, dbtutils
6. Просто интегрировать с Dagster (Airflow)

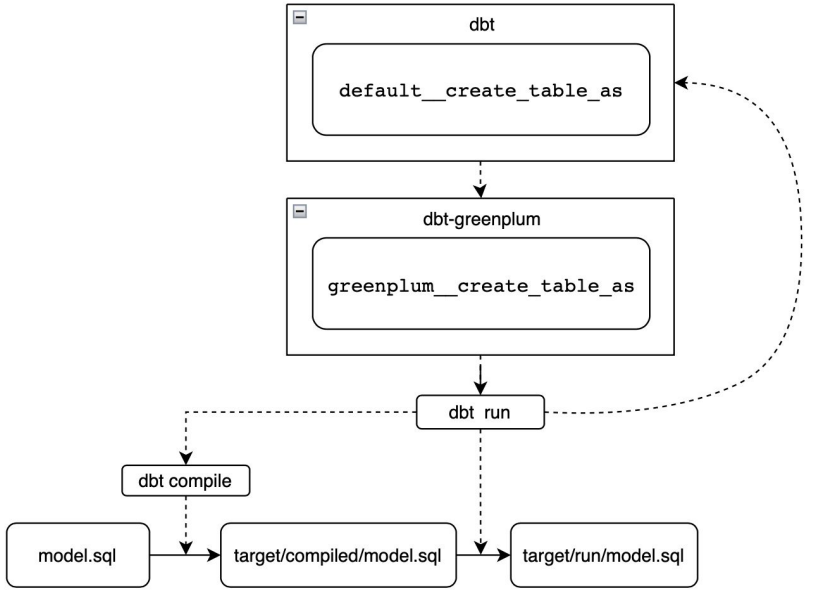
DBT выполнение модели

1. На этапе компиляции модели получаем executable select запрос
2. На этапе выполнения добавляем материализацию модели, так же получаем доступ к соединению к базе и можем вытащить что либо из базы, например, словари



DBT адаптеры

1. В конфигурации dbt указываете адаптер базы, с которой работаем
2. dbt берет специфичный sql код из адаптера



DBT материализация

1. view - создание вью из select запроса
2. table - создание таблицы из select запросы
3. incremental - наполнение таблицы с помощью insert запроса
4. пользовательские материализации
5. материализации специфичные для конкретных баз данных

DBT адаптер для greenplum

Был коннектор для postgresql, который не поддерживал специфичные для greenplum фичи, но в целом был рабочим. Решение - запилить свой коннектор.

Реализована поддержка следующей функциональности:

1. Сжатие
2. Колоночные таблицы
3. Дистрибьюция
4. Партиционирование

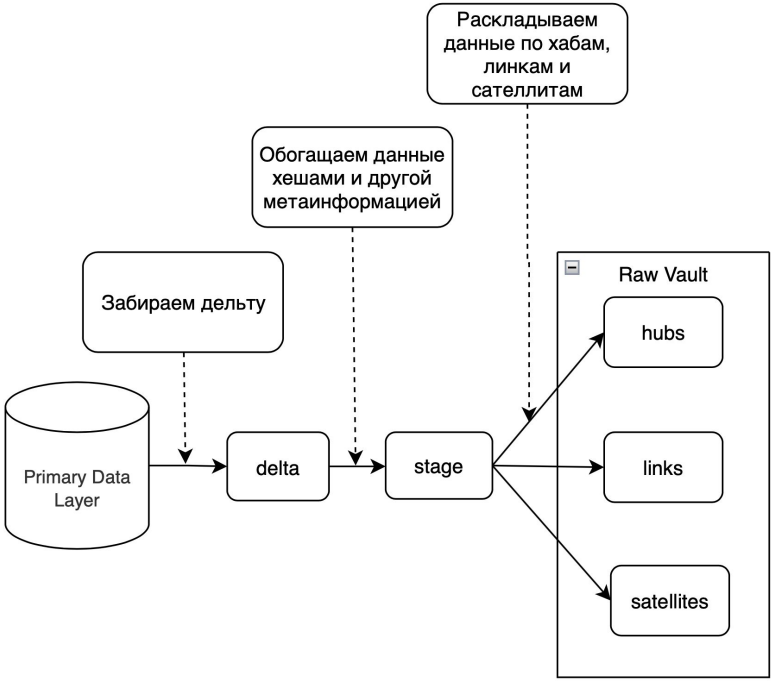
dbtvault для greenplum

Изначально пакет dbtvault реализовывался, для построение Data Vault для Snowflake, позже добавили поддержку BigQuery и SQLServer. Решение - сделать форк проекта и реализовать необходимые методы.

Как с помощью dbt строить DataVault

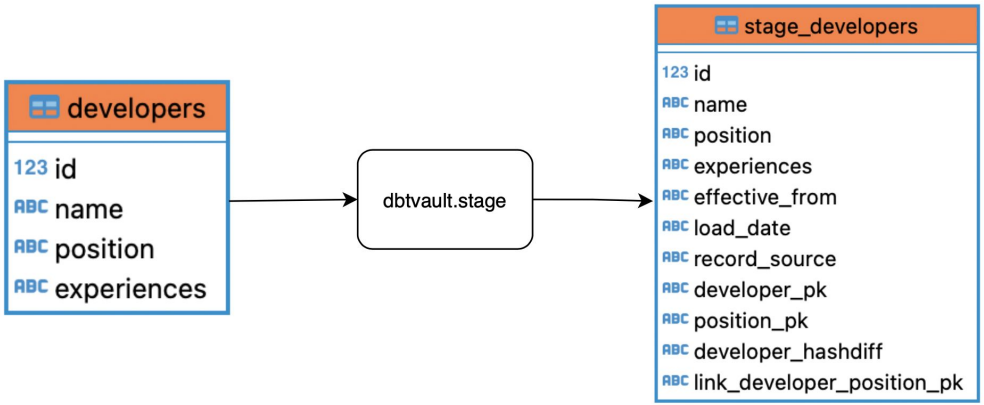
dbtvault

1. Забираем дельты с помощью sql запроса, с помощью логики, например, на основе даты загрузки
2. Обогащаем метаинформацией с помощью макроса stage из пакета dbtvault
3. Раскладываем по хабам, линкам и сателлитам с помощью макросов hub, link, sat



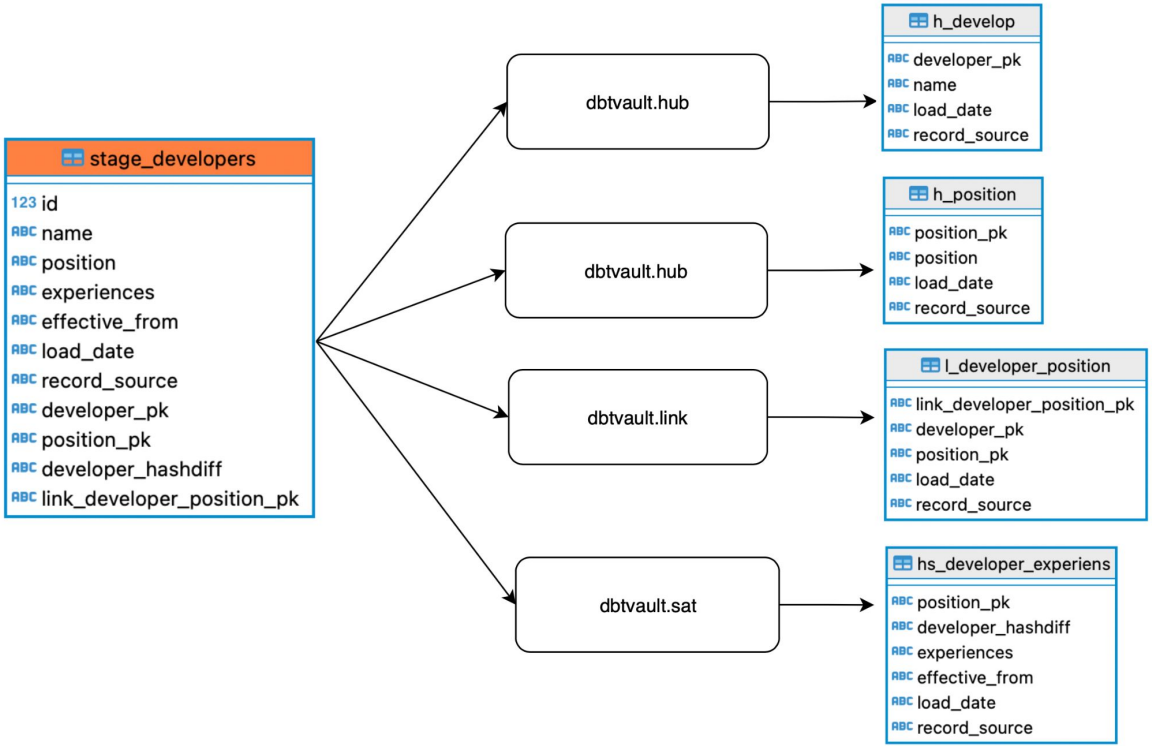
DataVault с помощью dbt (Stage)

dbtvault



DataVault с помощью dbt (RawVault)

dbtvault



Что у нас получилось

1. Порядка 50 таблиц в Raw Vault
2. Порядка 30 таблиц в Business Vault
3. Автоматизировано наполнение первичного, детального и витринного слоя
4. Задержка от получения данных до попадания в витрины ~10 минут
5. DWH решает свою основную функцию, datascience имеет доступ к детальному слою, клиенты получили обогащенные витрины

Основные проблемы с которыми мы столкнулись

1. Гринплам нежная база и некоторые запросы могут ее сломать, а так же надо следить за количеством соединений и теми запросами, которые на ней выполняются
2. Сложности в расхождении бизнес ключей в разных источниках и в целом с качеством приходящих данных
3. Репликация иногда ломается

Рекомендации

1. dbt+dbtvault - это практично, просто и работает
2. DataVault гибкий, но изменять существующие хабы может быть очень дорого, поэтому надо грамотно выделять сущности и выбирать бизнес ключи
3. Мы стараемся не использовать линки между более чем 2мя сущностями, потому что это впоследствии сложно использовать
4. Не стоит бояться переписывать загрузку DataVault на чистые dbt модели для оптимизации запросов

Ссылки

1. dbt: https://docs.getdbt.com
2. dbtvault: https://dbtvault.readthedocs.io/en/latest/
3. dbt-greenplum: https://github.com/markporoshin/dbt-greenplum
4. патч dbtvault для greenplum: https://github.com/markporoshin/dbtvault
5. https://t.me/Maaaaark

dbt_datavault_project

Что почитать

1. “Building a Scalable Data Warehouse with Data Vault 2.0“ Dan Linstedt, Michael Olschimke
2. https://habr.com/ru/company/otus/blog/588582/
3. https://habr.com/ru/post/670062/
4. https://medium.com/snowflake/advantage-data-vault-2-0-c1513f933bd8
5. https://habr.com/ru/company/glowbyte/blog/515940/