

**ФУНКЦИОНАЛЬНЫЕ
ХАРАКТЕРИСТИКИ
ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ
VI.QUBE
METAVULT**

Москва, 2022

Оглавление

Введение.....	3
1. Глоссарий	4
2. Цели и назначение Metavault	6
3. Функциональные возможности Metavault	6
4. Требования к программному и аппаратному обеспечению.....	8
4.1. Требования к ПО	8
4.2. Требования к аппаратному обеспечению.....	8
4.3. Требования к компетенциям пользователя	8

ВВЕДЕНИЕ

Компонент MetaVault предназначен для преобразования модели данных в представление Data Vault. Данный компонент является частью большой автоматизированной системы BI.Qube и может работать как в составе системы, так и независимо.

Компонент предназначен для быстрого построения аналитических корпоративных хранилищ данных в методологии Data Vault.

В документе приведено описание компоненты и принципы работы с ним. Рассмотрены примеры преобразования сущностей, представленных в классическом реляционном виде к виду Data Vault. Приведены примеры отслеживания ошибок.

Изучение данного документа позволит понять принцип работы компонента, принцип хранения данных в модели Data Vault, а так же при необходимости отслеживать ошибки допущенные в процессе настройки работы с компонентом.

1. ГЛОССАРИЙ

1.	MetaVault BI.Qube	Инструмент формирования модели данных на хранилище.
2.	MetaStaging BI.Qube	Инструмент, предназначенный для транспортировки данных.
3.	MetaOrchestrator BI.Qube	Инструмент управления процессами.
4.	Data Vault	Набор связанных между собой нормализованных таблиц, ориентированных на хранение детализированной информации с возможностью отслеживания происхождения данных и поддерживающих одну или несколько областей бизнеса.
5.	Hub	Обеспечивает представление функциональных областей предметной области.
6.	Link	Обеспечивает транзакционную связь между Hub-таблицами.
7.	Satellite	Предоставляет детализацию первичного ключа Hub-таблицы.
8.	Хранимая процедура	Объект базы данных, представляющий собой набор SQL-инструкций, который компилируется один раз и хранится на сервере
9.	Представление	Виртуальная таблица, содержимое которой определяется запросом
10.	Бизнес-ключ	Уникальный идентификатор, который бизнес использует в своих повседневных операциях. Примерами бизнес-ключей являются, например, номер счета-фактуры, номер сотрудника, номер клиента, номер компонента и т.д. Если бизнес утратит ключ, то будет потеряна вся информация об объекте.
11.	Суррогатный ключ	Порожденный ключ записи, порядковый номер, зачастую выбирается из последовательностей баз данных (sequence).
12.	Время загрузки	Дата, регистрирующая момент, когда ключ впервые был загружен в хранилище.

13.	Дата начала действия	Момент времени, когда запись впервые загружена или появилась после удаления.
14.	Дата конца действия	Момент удаления записи
15.	Идентификатор сущности родителя	Суррогатный ключ из таблицы Hub.
16.	Идентификатор сущности наследника	Суррогатный ключ из таблицы Hub.

2. ЦЕЛИ И НАЗНАЧЕНИЕ METAVULT

Компонент MetaVault, входит в состав системы аналитического корпоративного хранилища BI.Qube и предназначен для преобразования данных, представленных в реляционной форме в модель Data Vault.

Компонент может использоваться как отдельное независимое программное обеспечение, управляемое на уровне БД и скриптов для выполнения определенных задач, так и в связке с компонентом MetaStaging (входит в состав системы BI.Qube). Управление компонентом возможно как в ручном режиме, так и с использованием специализированного оркестратора MetaOrchestrator (входит в состав системы BI.Qube).

Кроме этого, компонент может управляться с использованием специализированного визуального интерфейса, однако в таком режиме компонент работает с ограниченной функциональностью и не представляет реального интереса.

3. ФУНКЦИОНАЛЬНЫЕ ВОЗМОЖНОСТИ METAVULT

Компонент MetaVault выполняет следующую основную функцию – преобразование данных представленных в реляционном формате к модели DataVault. Кратко принцип преобразования описан ниже. Архитектура хранилища данных с применением методологии Data Vault представлена на рис. 1.

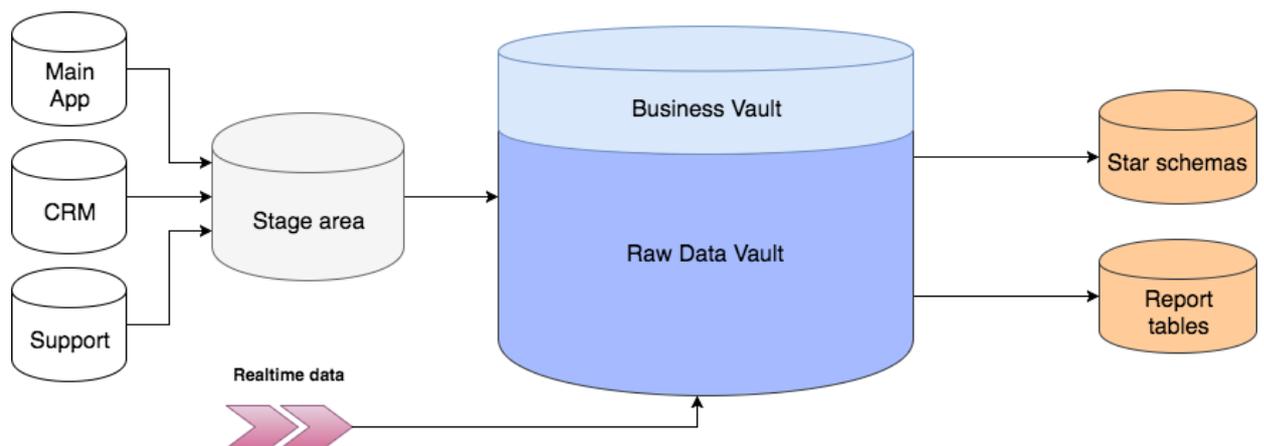


Рисунок 1 Хранилище данных в методологии Data Vault

Сначала данные из учетных систем поступают в промежуточный слой (staging area). Он используется как промежуточное звено в процессе загрузки

данных. Одна из основных функций этой зоны – это уменьшение нагрузки на операционные базы при выполнении запросов. Таблицы здесь полностью повторяют исходную структуру, но любые ограничения на вставку данных должны быть выключены с целью оставить возможность вставить даже поврежденные или неполные данные (особенно это актуально для excel-таблиц и прочих файлов).

После этого справочники разбиваются на сущности Data Vault и загружаются в Raw Data Vault. В процессе загрузки они никак не агрегируются и не пересчитываются.

Business Vault – опциональная вспомогательная надстройка над Raw Data Vault. Строится по тем же принципам, но содержит переработанные данные: агрегированные результаты, сконвертированные валюты и прочее. Разделение чисто логическое, физически Business Vault находится в одной базе с Raw Data Vault и предназначен в основном для упрощения формирования витрин.

Когда нужные таблицы созданы и заполнены, наступает очередь витрин данных (Data Marts). Каждая витрина – это отдельная база данных или схема, предназначенная для решения задач различных пользователей или отделов. В ней может быть специально собранная «звезда» или коллекция денормализованных таблиц. Если возможно, таблицы внутри витрин лучше делать виртуальными, то есть вычисляемыми «на лету». Для этого обычно используются SQL представления (SQL views).

Основные функции MetaVault

- Построение модели Data Vault по метаданным с использованием 3 типов сущностей: Hub, Link, Satellite.
- Создание бизнес-представления для справочника, соединяющего все его элементы.
- Проведение материализации бизнес-представления с сохранением всех индексов для существенного уменьшения времени выборки данных.
- Легкое расширение модели и добавление новых атрибутов в уже существующие справочники.
- Сохранение истории изменения записей в методологии SCD 2.

- Разделение атрибутов одного справочника по разным Satellite для уменьшения избыточности хранения данных (разделение по группам изменения)
- Повсеместное использование суррогатных автоинкрементных ключей, которое позволяет абстрагироваться от бизнес-ключей, ускорить операции соединения таблиц.
- Возможность назначить объект последовательности для автоинкрементных ключей разных справочников, что упрощает работу в аналитических системах при объединении одинаковых справочников из разных систем-источников
- Автоматизированность процесса сборки модели: таблицы, процедуры заполнения их данными, а также создания представления генерируется автоматически, многократно упрощая работу с методологией.
- Логическое разделение сущностей по профилям загрузки
- Независимость справочников друг от друга, возможность запускать процесс сборки различных профилей параллельно

4. ТРЕБОВАНИЯ К ПРОГРАММНОМУ И АППАРАТНОМУ ОБЕСПЕЧЕНИЮ

4.1. Требования к ПО

- БД: PostgreSQL (9.0 и позднее), Postgres Pro(10.22 и позднее), Arenadata Postgres (ADPG) (14.2.1)

4.2. Требования к аппаратному обеспечению

Минимальные аппаратные требования для установки серверной части (процессинг и БД):

- Процессор с тактовой частотой более 2.0 ГГц
- Оперативная память- 2GB
- Свободное место на жестком диске 350 Мб для исходного кода в процессе компиляции и 60 Мб для каталога инсталляции

4.3. Требования к компетенциям пользователя

- Навык работы с кодом PostgreSQL

- Написание процедур и запросов на PostgreSQL
- Создание таблиц PostgreSQL
- Вставка данных в таблицы PostgreSQL
- Выборка данных из SQL таблицы PostgreSQL
- Навык работы с моделью данных