ОПИСАНИЕ ТЕХНИЧЕСКОЙ АРХИТЕКТУРЫ BI.QUBE METAMASTERDATA

Москва, 2023

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ

MetaMasterData относится к классу MDM-систем и представляет собой решение по управлению основными данными, которое предоставляет надежный способ управления потоком данных через корпоративные ИТ-системы. Главная цель которого – обеспечить единство представления массивов данных в информационных системах. Кроме того, такой тип решений позволяет решить проблемы несоответствия, дублирования и несопоставимости данных.

Управление основными данными включает в себя действия, предпринимаемые организацией для нахождения и определения не транзакционных списков данных с целью компиляции управляемых главных списков. Основные данные отличаются от транзакционной большей стабильности, меньшими объемами и усложненной структурой.

Часто основные данные называют мастер-данные или эталонными данными, а также нормативно-справочной информацией.

В документе приведено описание компонента и принципы работы с ним.

Изучение данного документа позволит понять принцип работы компонента.

ГЛОССАРИЙ

1.	MetaMasterData	Инструмент BI.Qube, предназначенный для создания «идеального источника» данных, используя который, можно строить качественные аналитические срезы и повышать эффективность бизнес-процессов.
2.	Мастер-данные	Основные данные организации со схожим составом и атрибутами, хранящиеся в виде линейных или иерархических справочников.
3.	Master Data Management	Управление мастер-данными.
4.	НСИ	Нормативно-справочная информация – это фиксированные, исходно наполняемые и изменяемые только в редких случаях справочники (общероссийские или внутренние классификаторы, справочники стран, регионов, валют и т. д.).
5.	MDM-система	Это совокупность процессов и инструментов для непрерывного определения и управления мастер-

		данными и нормативно-справочной информацией в организации
6.	ИС	Информационные системы.
7.	ETL	Extract Transform Load – конвейер.
8.	Домен	Сущность с множеством допустимых значений.
9.	Золотая запись	— тезис в сфере управления мастер-данными. Он представляет собой формирование единственного, в наибольшей степени достоверного, истинного, непротиворечивого понятия об объектах предметной области. Это могут быть клиенты, контрагенты, чаще всего в сфере ритейл — это товары. Золотая запись часто именуют «единой версией правды», к которому могут по требованию обращаться все заинтересованные лица, если хотят убедиться, что находящиеся у них данные являются верными.
10.	Транзакционные данные	это данные, получаемые посредством CRUD операций.
11.	Структурированные данные	это данные, имеющие формально-определенную структуру.
12.	Полуструктурированные данные	это данные, не имеющие строго определенной структуры, однако, обладающие формальным описанием в виде тегов и/или маркеров.
13.	Неструктурированные	это данные не имеющие структуры и тегов/маркеров.
14.	Метаданные	это данные, используемые для описания какой-либо дополнительной информации о мастер-данных, к примеру конфигурационные данные.
15.	Историчность	это привязка модели к определенному времени, на протяжении которого данная модель не изменялась не изменялись.
16.	Версионность	модель, действующая в определенный период времени, сохраняющая модель данных или отдельных объектов данных: атрибутов, сущностей, справочников, связей.

17.	Модель данных	это абстрактное, самодостаточное, логическое определение объектов, с которой взаимодействует пользователь.
18.	Кроссплатформенность	способность программного обеспечения работать с несколькими аппаратно-программными комплексами
19.	Система управления базой данных (СУБД)	система, обеспечивающая контроль и управление данными, позволяющая выполнять различные административные операции.
20.	CRUD операции	операции по созданию, чтению, обновлению и удалению данных.
21.	Бизнес-правила	правила, используемые для валидации данных в соответствии с бизнес-логикой.
22.	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.

1. ЦЕЛИ И НАЗНАЧЕНИЕ METAMASTERDATA

Компонент MetaMasterData, входит в состав системы аналитического корпоративного хранилища BI.Qube и предназначен для быстрого построения и управления мастер-данными или нормативно-справочной информацией (НСИ) в организации.

Главная цель компонента — обеспечить единство представления массивов данных во всех информационных системах, посредством создания «золотой записи», то есть целостного и всестороннего представления о мастер-сущности и взаимосвязях, эталона мастер-данных, который используются всем предприятием, а иногда и между предприятиями для упрощения обмена информацией.

Компонент MetaMasterData предназначен для автоматического создания новых сущностей интегрируемых в существующую реляционную базу данных, с сохранением имеющейся модели данных. Компонент реализует стандартную функциональность по добавлению новых данных,

редактированию и удалению данных из базы данных, что позволит контролировать данные в базе данных, обогащать их новыми данными, связывать с имеющиеся данные с новыми.

Основным назначением является возможность создавать правила обработки данных: изменение типов данных, создание новых данных на основе имеющихся (новые поля, сущности), создание масок ввода и отображения данных, создания золотой записи. При этом сохраняется все история произведенных преобразований, что позволят всегда вернуться к исходному состоянию редактируемой базы данных.

Компонент MetaMasterData включен в качестве расширения в компонент MetaVault и без него автономно работать не может.

2. АРХИТЕКТУРА КОМПОНЕНТА

2.1. Описание архитектуры

В основу построения системы положена трехуровневая архитектура, которая включает в себя отдельные сервисы.

Такая архитектура позволяет улучшить надежность и устойчивость системы, а также обеспечить взаимодействие с каждым сервисом по отдельности через REST-API.

Программный комплекс состоит из трех уровней: клиентская часть, серверная часть, часть хранилища данных.

Клиентская часть представляет собой веб-приложение, которое взаимодействует с серверной частью при помощи REST-API. Для разработки клиентской части использовались следующие технологии: TypeScript, библиотека React js, Ant Design, HTML и CSS.

Серверная часть программного комплекса включает в себя приложение-сервер, которое является кроссплатформенным и может быть развернуто как на локальной машине, так и в облаке. Для разработки серверной части использовались технологии .NET 6 CORE.

Использование подхода хранения данных Data Vault (реализованного в компоненте MetaVault) имеет ряд преимуществ. Например, она обладает высокой масштабируемостью и способностью поддерживать сложные бизнес-правила. Также она обеспечивает быстрый доступ к данным, что важно для систем, где происходит обработка больших объемов информации.

Еще одно преимущество заключается в том, что подход хранения данных Data Vault обеспечивает надежность хранения данных, поскольку она предусматривает использование специальных таблиц для хранения истории

изменений данных. Это позволяет восстанавливать данные в случае ошибок или сбоев в системе.

Взаимодействие с компонентом MetaVault через NuGet пакет также обеспечивает простоту управления и мониторинга системы. Все необходимые инструменты для администрирования и настройки системы уже включены в состав NuGet пакета, что упрощает работу с ним.

Кроме того, NuGet пакет обеспечивает совместимость с другими платформами и языками программирования, что расширяет возможности использования системы.

В целом, использование стороннего программного комплекса Data Vault для хранения мастер-данных и метаданных в программном комплексе интеграции и управления мастер-данных является эффективным и надежным решением, которое обеспечивает высокую отказоустойчивость и масштабируемость системы. Кроме этого, отпадает необходимость интеграции мастер данных в хранилище данных, так как интеграция реализована уже на уровне ядра хранилища.

Рассмотрим подсистемы подробнее.

- **Подсистема безопасности** – позволяет управлять пользователями, группами, а также их правами.
- **Подсистема работы с иерархическими моделями** – позволяет создавать, редактировать и удалять домены, модули и сущности, а также их атрибуты, правила валидации. Состоит из следующих модулей.
- **Подсистема версионирования –** позволяет версионировать структуру сущностей, атрибутов и данных, а также отменять уже примененные изменения.
- **Подсистема работы с данными: Модуль управления данными** – позволяет управлять данными сущностей, включая работу с атрибутами на доменную сущность.
- **Подсистема работы с метаданными: Модуль управления метаданными** – позволяет управлять метаданными профилей, доменов, сущностей, представлений, а также атрибутов.
    - **Подсистема валидации данных** – позволяет управлять правилами валидации данных посредством визуального

интерфейса и строить правила валидации по принципу ЕСЛИ-ТО-ИНАЧЕ, а также подключать существующие CUSTOM правила с хранилища; позволяет валидировать внесенные данные в сущность посредством созданных бизнес-правил.

- **Подсистема работы с «Золотой записью»** – позволяет настраивать приоритет источников данных, а также получать «золотой» элемент посредством слияния данных об одной записи с разных источников; слияния записей осуществляется при помощи специального алгоритма, в котором находятся записи об одной сущности и сливаются в «золотую» запись.

Схематично Архитектура программного комплекса управления мастер-данными изображена на рисунке ниже.

Рисунок 1. Организационная схема компонента MetaMasterData

2.2. Модель данных

Для хранения информации о сформированной модели данных, построенной по методологии Data Vault, спроектирована и реализована база данных, представленная на рисунке (спроектирована ранее для компонента MetaVault)

Все таблицы в настроечной базе данных имеют технические поля: deleted, modified_at и modified_by. В таблицах строки не удаляются, а выставляется флаг удаления. При любом изменении записывается время изменения и пользователь, который инициировал изменение.

Таблицы находятся схеме meta_vault, так как планируется, что в настроечной базе данных будут находиться таблицы других BI-программ.

1. Attribute – информация о полях таблиц на источнике:

- attribute_id – идентификатор атрибута;

- source_database – наименование базы данных источника справочника;
  -
- source_schema – наименование схемы источника справочника;
- source_table – наименование таблицы справочника;
- source_column – наименование атрибута в источнике;
- overriden_name – переопределенное наименование;
- data_type – тип данных атрибута;
- precision – точность или длина строки, -1 если ограничения нет;
- scale - масштаб, используется для типов decimal(precision, scale);
- nullabled – может ли атрибут содержать NULL;
- description – описание атрибута;
- deleted – удалено ли поле;
- modified_at – время и дата последнего изменения атрибута;
- modified_by – пользователь, инициировавший последнее изменение.

2. Profile – список профилей загрузки, контейнер для сущностей:
   - profile_id – идентификатор профиля;
   - name – наименование профиля;
   - description – описание профиля;
   - deleted – удалено ли поле;
   - modified_at – время и дата последнего изменения профиля;
   - modified_by – пользователь, инициировавший последнее изменение.
3. Environment – среда выполнения:
   - environment_id – идентификатор среды выполнения;
   - name – наименование среды выполнения;
   - description – описание среды выполнения;
   - deleted – удалено ли поле;

- modified_at – время и дата последнего изменения;
- modified_by – пользователь, инициировавший последнее изменение.

4. Entity – информация о справочниках, настраивает Hub, относится к профилю загрузки:
   - entity_id – идентификатор сущности;
   - profile_id – идентификатор профиля загрузки;
   - name – название сущности;
   - description – описание сущности;
   - table_target_schema – схема в которой, будет создан Hub;
   - deleted – удалено ли поле;
   - modified_at – время и дата последнего изменения;
   - modified_by – пользователь, инициировавший последнее изменение.
5. Hub_attribute – настройка бизнес-ключа для сущности Hub:
   - hub_attribute_id – идентификатор поля Hub;
   - entity_id – идентификатор сущности;
   - attribute_id – идентификатор атрибута;
   - order – порядок поля в ключе;
   - deleted – удалено ли поле;
   - modified_at – время и дата последнего изменения;
   - modified_by – пользователь, инициировавший последнее изменение.
6. Link – настройка сущности Link между справочниками:
   - link_id – идентификатор Link;
   - name – наименование сущности Link;
   - description – описание сущности Link;
   - parent_entity_id – идентификатор родительской сущности;

- parent_attribute_id – идентификатор родительского атрибута;
- child_entity_id – идентификатор дочерней сущности;
- child_attribute_id – идентификатор дочернего атрибута;
- deleted – удалено ли поле;
- modified_at – время и дата последнего изменения;
- modified_by – пользователь, инициировавший последнее изменение.

7. Satellite – настройка Satellite для справочника:
   - satellite_id – идентификатор Satellite;
   - entity_id – идентификатор родительской сущности;
   - name – наименование сущности Satellite;
   - description – описание сущности Satellite;
   - deleted – удалено ли поле;
   - modified_at – время и дата последнего изменения;
   - modified_by – пользователь, инициировавший последнее изменение.
8. Satellite_attribute – настройка полей Satellite:
   - satellite_attribute_id – идентификатор связи Satellite и Attribute;
   - satellite_id – идентификатор связанного Satellite;
   - attribute_id – идентификатор поля Attribute;
   - deletion_flag – является ли поле признаком удаления записи;
   - deleted – удалено ли поле;
   - modified_at – время и дата последнего изменения;
   - modified_by – пользователь, инициировавший последнее изменение.
9. Business_view – список бизнес-представлений справочника:
   - business_view_id – идентификатор бизнес-представления;
   - name – наименование бизнес-представления;

- description – описание бизнес-представления;
- entity_id – идентификатор родительской сущности;
- historized – историчное ли представление;
- active_rows_used – если не историчное представление, отображать ли актуальные записи;
- deleted_rows_used – отображать ли удаленные записи;
- materialized – материализовано ли представление;
- target_view_schema – схема в которой находится бизнес-представление;
- deleted – удалено ли поле;
- modified_at – время и дата последнего изменения;
- modified_by – пользователь, инициировавший последнее изменение.

10. Entity_environment – таблица с соответствием справочника и среды выполнения:
    - entity_environment_id – идентификатор связи среды выполнения с сущностью;
    - entity_id – идентификатор связанной сущности;
    - environment_id – идентификатор связанной среды выполнения;
    - enabled – включена ли сборка справочника;
    - deleted – удалено ли поле;
    - modified_at – время и дата последнего изменения;
    - modified_by – пользователь, инициировавший последнее изменение.

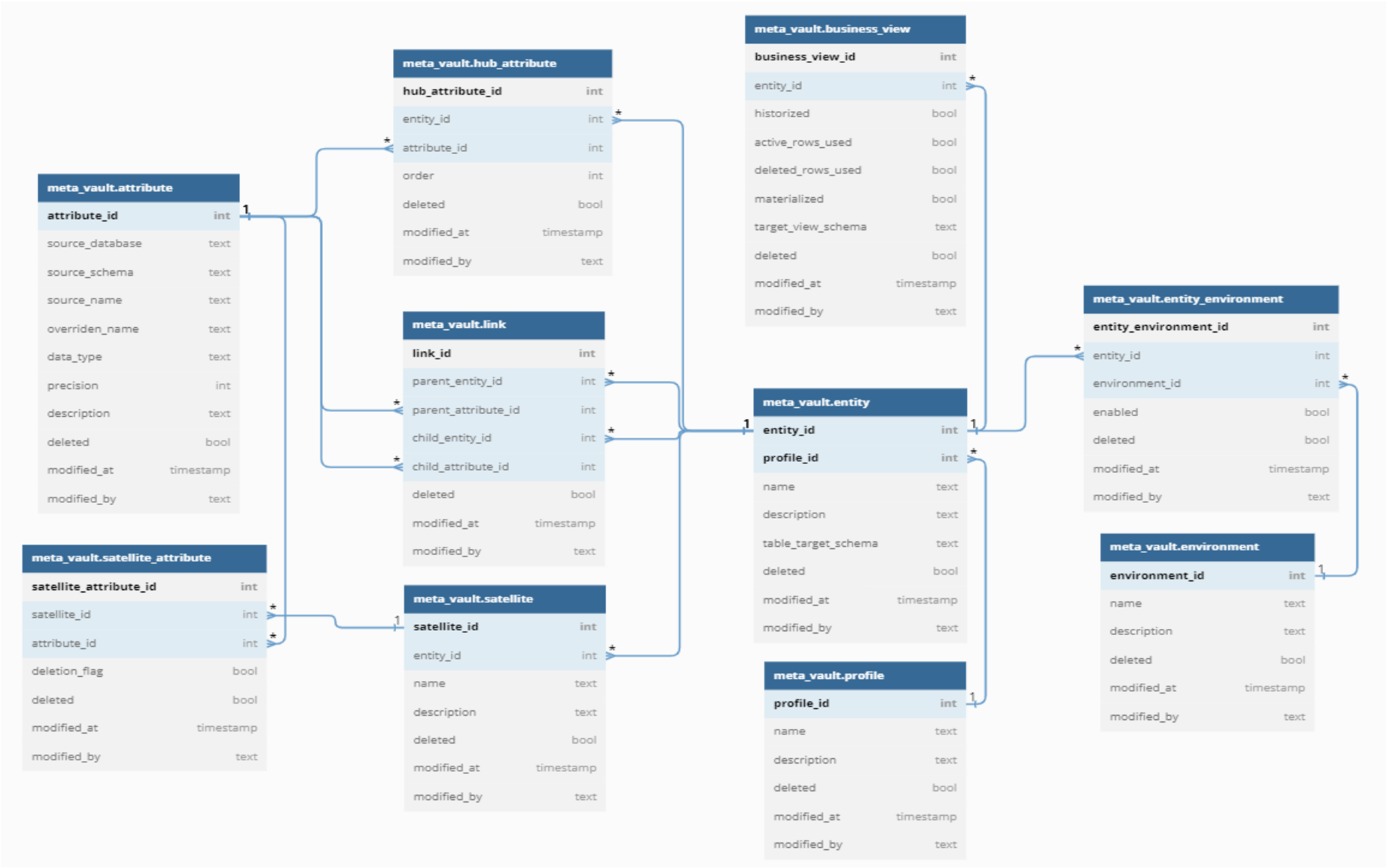

Рис.. ERD-диаграмма настроечной базы данных