

Описание технической архитектуры BI.Qube METAVault

ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	3
ГЛОССАРИЙ.....	3
1. ЦЕЛИ И НАЗНАЧЕНИЕ METAVULT	4
2. УСТАНОВКА И ЗАПУСК	5
3. СТРУКТУРА METAVULT	5
3.1. Таблицы MetaVault	6
3.2. Хранимые процедуры и функции MetaVault	13
Пользовательские процедуры.....	14
Сервисные процедуры.....	15
3.3. Представления MetaVault	18
3.4. Источники MetaVault.....	19
3.5. Структура сущности.....	21

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 также при необходимости отслеживать ошибки допущенные в процессе настройки работы с компонентом.

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-представление	Представление, в котором собраны Hub, Satellite и Link для сущности
11.	Материализация	Процесс сохранения результата запроса бизнес-представления в таблицу для ускорения выборки.
12.	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.
13.	Суррогатный ключ	Порожденный ключ записи, порядковый номер, зачастую выбирается из последовательностей баз данных (sequence).
14.	Время загрузки	Дата, регистрирующая момент, когда ключ впервые был загружен в хранилище.
15.	Дата начала действия	Момент времени, когда запись впервые загружена или появилась после удаления.
16.	Дата конца действия	Момент удаления записи
17.	Идентификатор сущности родителя	Суррогатный ключ из таблицы Hub.
18.	Идентификатор сущности наследника	Суррогатный ключ из таблицы Hub.

1. 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 MetaStaiging (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.

2. УСТАНОВКА И ЗАПУСК

Компонент MetaVault работает под управлением СУБД: PostgreSQL (9.0 и позднее), Postgres Pro (10.22 и позднее), Arenadata Postgres (ADPG) (14.2.1).

В связи с высокой сложностью развертывания компонента в среде целевой СУБД установку компонента осуществляет вендор. Демонстрационная версия расположена по адресу 212.103.126.82:38154 подключение осуществляется в режиме удаленного рабочего стола. Имя пользователя *TestRR01*, пароль предоставляется по запросу.**СТРУКТУРА METAVALT**

Компонент MetaVault включает в себя:

- Таблицы – настроечные таблицы (tables);
- Хранимые процедуры (функции) (routines);
- Представления (views).

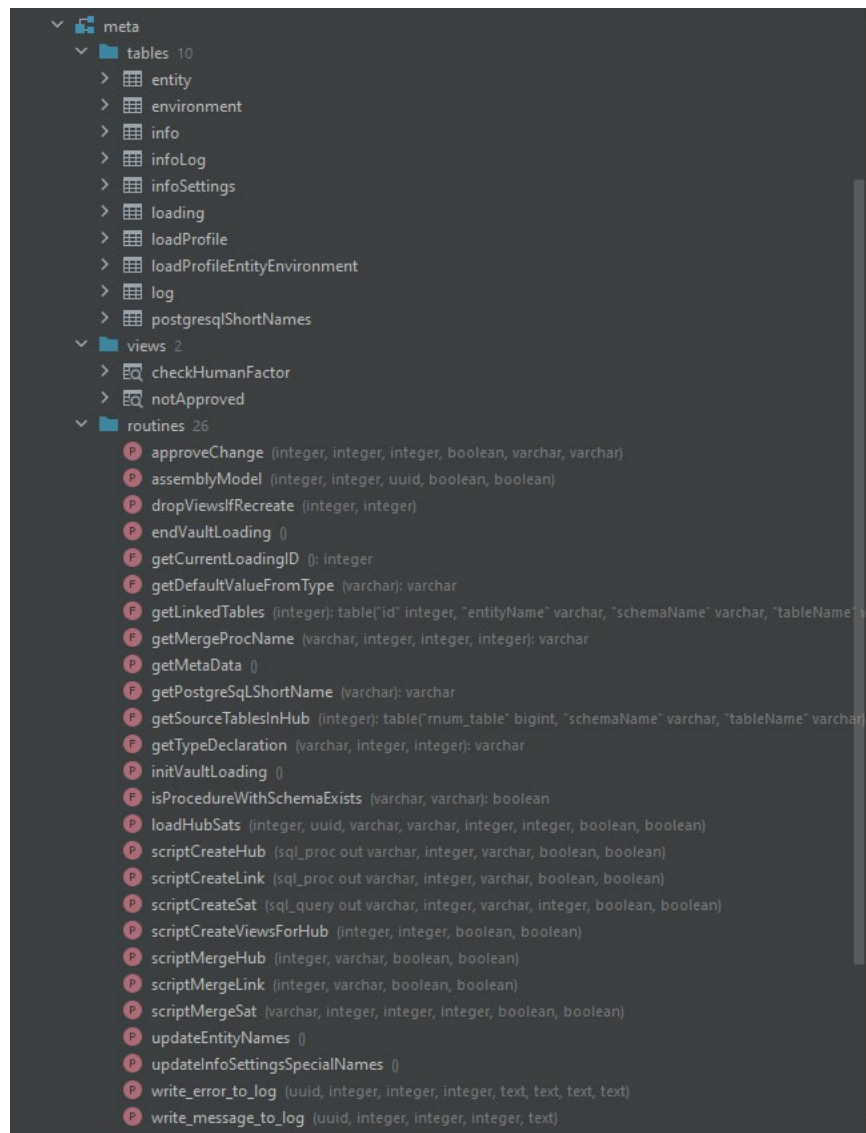


Рисунок 1 Структура MetaVault

3.1. Таблицы MetaVault

В таблицах хранится вся необходимая информация для работы компонента. Назначение каждой таблицы приведено ниже. Часть таблиц заполняется вручную, часть автоматически.

1. meta.entity (сущность) – хранит данные о таблицах (сущностях), которые будут обрабатываться в модели Data Vault.

id	name	isHub	isMaterialized	useSequenceInPK	customSequenceName
1	user	true	false	false	<null>
2	artist	true	false	false	<null>
3	album	true	false	false	<null>
4	track	true	false	false	<null>
5	track_album	false	false	false	<null>
6	album_artist	false	false	false	<null>

Рисунок 2 Таблица «meta.entity»

Поля таблицы entity:

Имя столбца	Тип данных	Источник	Назначение
id	int	Метаданные	Идентификатор сущности
name	nvarchar	Метаданные	Наименование сущности. Можно переименовывать
isHub	bit	Вручную	Поле isHub предназначено для отнесения сущности к типу хаб (True) или линк (False)
isMaterialized	bit	Вручную	Флаг, указывающий на необходимость материализации представления данной сущности
useSequenceInPK	bit	Вручную	Флаг, указывающий, что в качестве первичного ключа сущности необходимо использовать последовательность.
customSequenceName	nvarchar	Вручную	Наименование последовательности

2. «meta.environment» (среда выполнения) – позволяет вести разработку на нескольких средах. Введено для развития системы в будущем.

environmentID	name
1	основная

Рисунок 3 Таблица «meta.environment»

3. «meta.info» – таблица, в которой содержится информация о последнем состоянии метаданных – заполняется автоматически

columnName	columnID	dataType	maxLength	precision	scale	isNullable	lastLoad	deleted
Title	6	2 character varying	160	0	0	false	1	false
useSequenceInPK	7	8 boolean	0	0	0	false	1	false
customSequenceName	8	9 character varying	63	0	0	true	1	false
ArtistId	9	3 integer	32	0	0	false	1	false
ArtistId	10	1 integer	32	0	0	false	1	false
Name	11	2 character varying	120	0	0	true	1	false
CustomerId	12	1 integer	32	0	0	false	1	false
FirstName	13	2 character varying	40	0	0	false	1	false
LastName	14	3 character varying	20	0	0	false	1	false
Company	15	4 character varying	80	0	0	true	1	false
Address	16	5 character varying	70	0	0	true	1	false
City	17	6 character varying	40	0	0	true	1	false
State	18	7 character varying	40	0	0	true	1	false
Country	19	8 character varying	40	0	0	true	1	false
PostalCode	20	9 character varying	10	0	0	true	1	false
Phone	21	10 character varying	24	0	0	true	1	false
Fax	22	11 character varying	24	0	0	true	1	false
Email	23	12 character varying	60	0	0	false	1	false
SupportRepId	24	13 integer	32	0	0	true	1	false
loadProfileID	25	1 integer	32	0	0	false	1	false
name	26	2 character varying	63	0	0	false	1	false
description	27	3 character varying	0	0	0	true	1	false
environmentID	28	1 integer	32	0	0	false	1	false
name	29	2 character varying	63	0	0	false	1	false
loadProfileID	30	1 integer	32	0	0	false	1	false
entityID	31	2 integer	32	0	0	false	1	false
environmentID	32	3 integer	32	0	0	false	1	false
targetTableSchema	33	7 character varying	63	0	0	true	1	false
targetViewSchema	34	8 character varying	63	0	0	true	1	false
entRecreateTable	35	4 boolean	0	0	0	false	1	false

Рисунок 4 Таблица «meta.info»

4. «meta.infoLog» – информация об изменении полей источника – заполняется автоматически, необходима для отслеживания возможных изменений в источнике данных.

id	action	objectID	sourceDB	schemaName	tableName	columnName	columnID	dataType
1	355 upsert	10587	bique	meta	entity	id	1	integer
2	356 upsert	10587	bique	meta	entity	isHub	3	boolean
3	357 upsert	10587	bique	meta	entity	isMaterialized	4	boolean
4	358 upsert	10587	bique	meta	entity	useSequenceInPK	8	boolean
5	359 upsert	10609	bique	meta	loadProfile	loadProfileID	1	integer
6	360 upsert	10619	bique	meta	environment	environmentID	1	integer
7	361 upsert	10626	bique	meta	loadProfileEntityEnvironment	loadProfileID	1	integer
8	362 upsert	10626	bique	meta	loadProfileEntityEnvironment	entityID	2	integer
9	363 upsert	10626	bique	meta	loadProfileEntityEnvironment	environmentID	3	integer
10	364 upsert	10626	bique	meta	loadProfileEntityEnvironment	entRecreateTable	4	boolean
11	365 upsert	10626	bique	meta	loadProfileEntityEnvironment	entRecreateMergeProc	5	boolean
12	366 upsert	10626	bique	meta	loadProfileEntityEnvironment	entReloadData	6	boolean
13	367 upsert	10626	bique	meta	loadProfileEntityEnvironment	isEnabled	9	boolean
14	368 upsert	10667	bique	meta	infoLog	id	1	integer
15	369 upsert	10667	bique	meta	infoLog	objectID	3	integer
16	370 upsert	10667	bique	meta	infoLog	columnID	12	integer
17	371 upsert	10667	bique	meta	infoLog	maxLength	14	integer
18	372 upsert	10667	bique	meta	infoLog	precision	15	integer
19	373 upsert	10667	bique	meta	infoLog	scale	16	integer
20	374 upsert	10667	bique	meta	infoLog	isNullable	17	boolean
21	375 upsert	10667	bique	meta	infoLog	lastLoad	19	integer
22	376 upsert	10667	bique	meta	infoLog	deleted	20	boolean
23	377 upsert	10667	bique	meta	infoLog	approved	21	boolean
24	378 upsert	10667	bique	meta	infoLog	ignored	22	boolean
25	379 upsert	10676	bique	meta	loading	id	1	integer

Рисунок 5. Таблица «meta.infoLog»

«meta.infoSettings» – таблица, задающая структуру Raw Data Vault. Для каждого поля автоматически определяются параметры, полученные в результате чтения метаданных источника. Остальные параметры задаются вручную. Здесь настраиваются Satellite сущности, разделения полей по ним. Имеется возможность указать флаг удаления для источников, на которых присутствует историчность.

id	sourceDB	schemaName	tableName	columnName	isBK	BKOrderInHub	entityID	linkTo	LinkToBK	LinkID
1	18	bique	public	Track	GenreId	false	<null>	5	<null>	<null>
2	15	bique	public	Track	AlbumId	false	<null>	5	4	1
3	14	bique	public	Track	UnitPrice	false	<null>	5	<null>	<null>
4	12	bique	public	Album	ArtistId	false	<null>	4	3	1
5	7	bique	staging	user	id	true	1	<null>	<null>	<null>
6	6	bique	staging	user	first_name	false	<null>	2	<null>	<null>
7	8	bique	staging	user	last_name	false	<null>	2	<null>	<null>
8	5	bique	staging	user	birth	false	<null>	2	<null>	<null>
9	9	bique	public	Artist	ArtistId	true	1	3	<null>	<null>
10	10	bique	public	Artist	name	false	<null>	3	<null>	<null>
11	11	bique	public	Album	AlbumId	true	1	4	<null>	<null>
12	13	bique	public	Album	title	false	<null>	4	<null>	<null>
13	21	bique	public	Track	name	false	<null>	5	<null>	<null>
14	20	bique	public	Track	milliseconds	false	<null>	5	<null>	<null>
15	22	bique	public	Track	TrackId	true	1	5	<null>	<null>
16	17	bique	public	Track	composer	false	<null>	5	<null>	<null>
17	16	bique	public	Track	bytes	false	<null>	5	<null>	<null>
18	19	bique	public	Track	mediaTypeId	false	<null>	5	<null>	<null>

Рисунок 6. Таблица «meta.infoSettings»

Имя столбца	Тип данных	Источник	Назначение
id	int	Метаданные	Идентификатор записи метаданных
source_DB	sysname	Метаданные	Наименование базы данных таблицы-источника
schema name	sysname	Метаданные	Наименование схемы таблицы-источника
table name	sysname	Метаданные	Наименование таблицы-источника
column name	sysname	Метаданные	Наименование поля таблицы-источника
is_BK	bit	Вручную	Является ли поле бизнес-ключом. Все бизнес-ключи заносятся в хаб.
BK_order_in_hub	tinyint	Вручную	Внутренняя нумерация бизнес-ключей в хабе, используется для внешних ссылок на этот хаб.
entity_id	int	Вручную	Идентификатор сущности (хаба), к которой относится поле. Необходимо заполнить для всех бизнес-ключей и сателлитов. Если значение не указано, поле не используется при построении модели.
link_to	int	Вручную	Если нужно задать линк, то указать id хаба, на который ссылается данное поле. Если присоединяется таблица атрибутов (структура бизнес-ключей в таблице продублирована, в ней содержатся только сателлиты), то указать ссылку на одноименное поле из родительской таблицы (у таких таблиц разные значения inner_group, но entity_id одинаковый, link_id отсутствует, так как это объединение данных в таблицах, а не линк)
Link_to_BK	tinyint	Вручную	Ссылка на поле BK_order_in_hub. Указывается значение поля, на которое ссылается по линку данное поле. Необходимо для конкретизации ссылки в случае использования составного бизнес-ключа.
link_id	int	Вручную	Идентификатор линка, к которому относится указанная связь
new_name	nvarchar(128)	Вручную	Новое название поля. На данный момент не используется.
changes_frequency	tinyint	Вручную	Категория времени изменения для сателлитов. Используется для их группировки по таблицам.
sat_recreate_table	bit	Вручную	Пересоздание сателлита данного хаба (можно пометить одно или больше полей сателлита одной change frequency)
sat_recreate_merge	bit	Вручную	Пересоздание процедуры merge сателлита данного хаба (можно пометить одно или больше полей сателлита одной change frequency)

Имя столбца	Тип данных	Источник	Назначение
sat_reload	bit	Вручную	Обновление данных (merge) спутника данного хаба (можно пометить одно или больше полей спутника одной change frequency)
isDeletionFlag	bit	Вручную	Флаг, который переключает режим обработки удалённых записей. По умолчанию запись считается удалённой, если она пропадает из таблицы-источника. Если в таблице-источнике есть специальное поле-признак удаления записи, то можно переключить данный флажок у нужного поля, и тогда запись будет помечаться удалённой только если она помечена в таблице-источнике
column_id	int	Метаданные	
data_type	sysname	Метаданные	
max_length	smallint	Метаданные	
precision	tinyint	Метаданные	
scale	tinyint	Метаданные	
is_nullable	bit	Метаданные	
is_identity	bit	Метаданные	

Таблица «meta.infoSettings» позволяет:

- Настроить бизнес-ключи

	id	sourceDB	schemaName	tableName	columnName	isBK	BKOrderInHub	entityID
1	18	biqube	public	Track	GenreId	false	<null>	
2	15	biqube	public	Track	AlbumId	false	<null>	
3	14	biqube	public	Track	UnitPrice	false	<null>	
4	21	biqube	public	Track	Name	false	<null>	
5	20	biqube	public	Track	Milliseconds	false	<null>	
6	22	biqube	public	Track	TrackId	true	1	
7	17	biqube	public	Track	Composer	false	<null>	
8	16	biqube	public	Track	Bytes	false	<null>	
9	19	biqube	public	Track	MediaTypeId	false	<null>	

Рисунок 7. Настройка бизнес-ключей

- Определить наличие ссылок между сущностями и создать эти ссылки

	columnName	isBK	BKOrderInHub	entityID	linkTo	linkToBK	LinkID	newName	changesFrequency	columnID	dataType
1	d	false	<null>	5	<null>	<null>	<null>	genre_id	1	5	integer
2	d	false	<null>	5	4	1	6	album_id	1	3	integer
3	ice	false	<null>	5	<null>	<null>	<null>	unit_price	1	9	numeric
4		false	<null>	5	<null>	<null>	<null>	name	1	2	character
5	econds	false	<null>	5	<null>	<null>	<null>	milliseconds	1	7	integer
6	d	true	1	5	<null>	<null>	<null>	track_id	1	1	integer
7	ar	false	<null>	5	<null>	<null>	<null>	composer	1	6	character
8		false	<null>	5	<null>	<null>	<null>	bytes	1	8	integer
9	ypeId	false	<null>	5	<null>	<null>	<null>	media_type_id	1	4	integer

Рисунок 8. Тест на наличие ссылок между сущностями

- Изменить наименование полей (указать новое наименование)

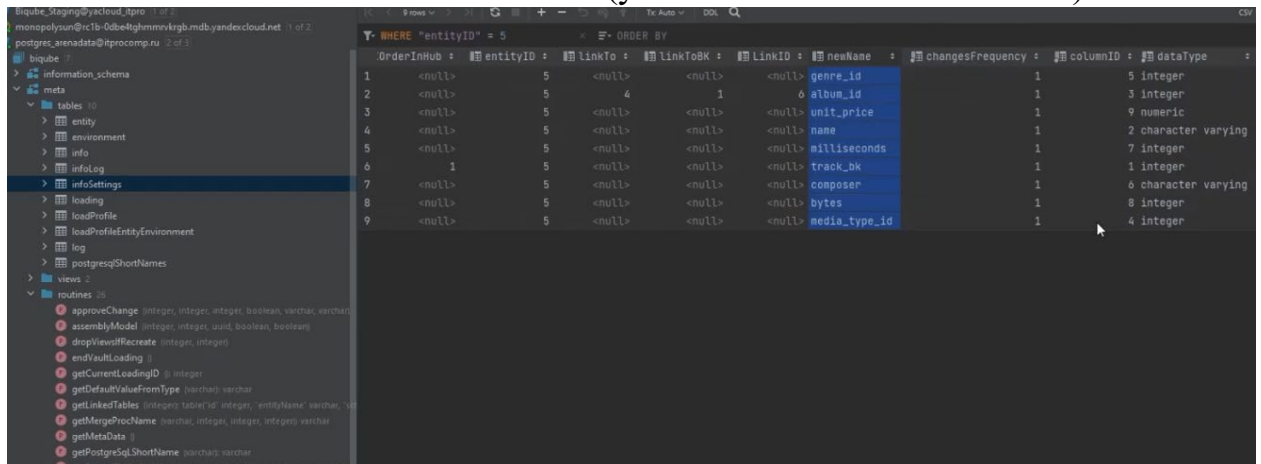


Рисунок 9. Изменение наименования полей

- Разделить сущности по Satellite

entityID	linkTo	linkToBK	LinkID	newName	changesFrequency	columnID	dataType
9	<null>	<null>	<null>	<null>	1	3	text
1	9	<null>	<null>	<null>	1	1	integer
9	<null>	<null>	<null>	<null>	2	2	character varying

Рисунок 10. Разделение сущности по Satellite

5. «meta.loading» – таблица, фиксирующая данные о начале и конце загрузки - заполняется автоматически через хранимые процедуры, «meta.initVaultLoading»(начинает) «meta.endVaultLoading» (заканчивает).

Рисунок 11. Таблица «meta. loading»

6. «meta.loadProfile» – таблица - профиль загрузки для сущностей. Содержит перечень профилей загрузки. В профиле могут содержаться сущности, таким образом происходит их логическое разделение.

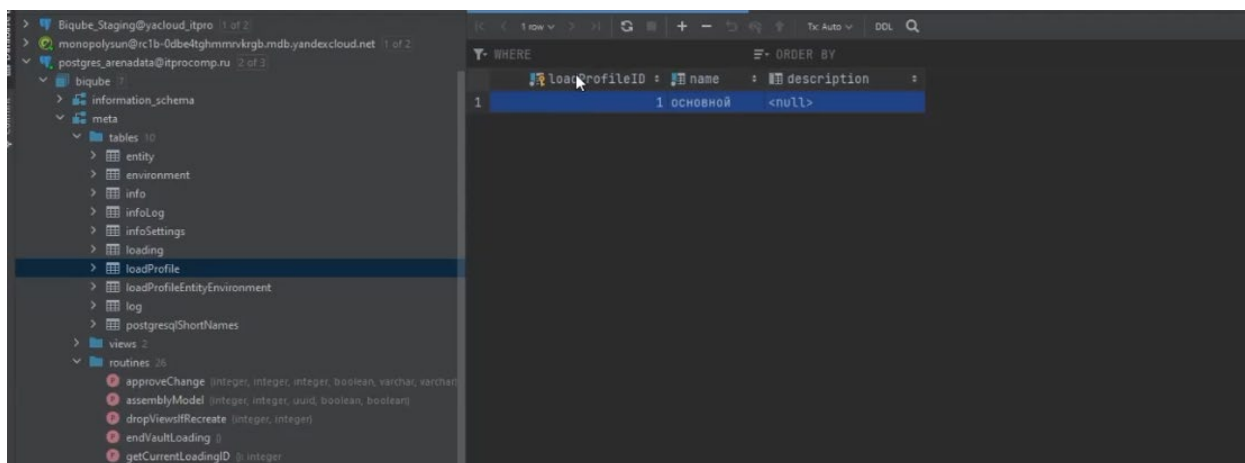


Рисунок 12. Таблица «meta.loadProfile»

Имя столбца	Тип данных	Источник	Назначение
LoadProfileID	int	Вручную	Идентификатор профиля загрузки
Name	nvarchar	Вручную	Наименование профиля загрузки
Description	nvarchar	Вручную	Описание профиля загрузки

7. «meta.loadProfileEntityEnvironment» – таблица, в которой указываются настраиваемые сущности: профиль загрузки, среда выполнения, идентификатор сущности, пересоздание сущности, пересоздание процедуры заполнения сущности процедурами, схема для хранения внутренних объектов Data Vault, схема для хранения бизнес-представлений Data Vault.

Рисунок 13. Таблица «meta.loadProfileEntityEnvironment»

Имя столбца	Тип данных	Источник	Назначение
LoadProfileID	int	Вручную	Идентификатор профиля загрузки
EntityID	int	Вручную	Идентификатор сущности
EnvironmentID	int	Вручную	Идентификатор среды
ent_recreateTable	bit	Вручную	Пересоздание данного хаба (будут потеряны все данные сущности)
ent_recreateMergeProc	bit	Вручную	Пересоздание процедуры merge данного хаба
ent_reloadData	bit	Вручную	Обновление данных (merge) хаба
isEnabled	bit	Вручную	Включение и отключение загрузки

8. «meta.log» – информация о процессе сборки модели Data Vault – заполняется автоматически в ходе работы хранимой процедуры «meta.AssemblyModel».


```
▼ routines 26
P approveChange (integer, integer, integer, boolean, varchar, varchar)
P assemblyModel (integer, integer, uuid, boolean, boolean)
P dropViewsIfRecreate (integer, integer)
P endVaultLoading ()
F getCurrentLoadingID (): integer
F getDefaultValueFromType (varchar): varchar
F getLinkedTables (integer): table["id" integer, "entityName" varchar, "schemaName" varchar, "tableName" varchar, "isRoot" boolean]
F getMergeProcName (varchar, integer, integer, integer): varchar
P getMetaData ()
F getPostgreSQLShortName (varchar): varchar
F getSourceTablesInHub (integer): table["rnum_table" bigint, "schemaName" varchar, "tableName" varchar]
F getTypeDeclaration (varchar, integer, integer): varchar
P initVaultLoading ()
F isProcedureWithSchemaExists (varchar, varchar): boolean
P loadHubSats (integer, uuid, varchar, varchar, integer, integer, boolean, boolean)
P scriptCreateHub (sql_proc out varchar, integer, varchar, boolean, boolean)
P scriptCreateLink (sql_proc out varchar, integer, varchar, boolean, boolean)
P scriptCreateSat (sql_query out varchar, integer, varchar, integer, boolean, boolean)
P scriptCreateViewsForHub (integer, integer, boolean, boolean)
P scriptMergeHub (integer, varchar, boolean, boolean)
P scriptMergeLink (integer, varchar, boolean, boolean)
P scriptMergeSat (varchar, integer, integer, integer, boolean, boolean)
P updateEntityNames ()
P updateInfoSettingsSpecialNames ()
P write_error_to_log (uuid, integer, integer, integer, text, text, text, text)
P write_message_to_log (uuid, integer, integer, integer, text)
```

Рисунок 16. Список хранимых процедур и функций MetaVault

В процессе работы с MetaVault «вручную» могут вызываться только пять хранимых процедур: «meta.initVaultLoading», «meta.endVaultLoading», «meta.getMetaData», «meta.ApproveChange», «meta.AssembleModel» – пользовательские процедуры. Пользовательские процедуры при необходимости вызывают сервисные процедуры, вызов которых пользователем в ручном режиме не рекомендован.

Пользовательские процедуры

Ниже приведено описание пользовательских процедур:

1. «meta.initVaultLoading» – инициализация загрузки MetaVault
Параметров не имеет.
2. meta.EndVaultLoarding – завершение загрузки MetaVault
Параметров не имеет.
3. «meta.getMetaData» – хранимая процедура получает данные из источника и записывает в таблицу «meta.info»
Параметров не имеет.
4. «meta.ApproveChange» – хранимая процедура добавляет конкретное поле в таблицу infoSettings, создает сущность при необходимости.
Параметры:
 - current_id – идентификатор поля из meta.info

- load_profile_id – идентификатор профиля загрузки
- environment_id – идентификатор среды разработки
- create_entity – нужно ли создавать сущность
- target_table_schema – схема, в которой будут находиться внутренние объекты модели
- target_view_schema – схема, в которой будет находиться бизнес-представление

5. «meta.AssembleModel» – хранимая процедура выполняет сборку модели для связки профиль-среда выполнения. Пересоздает сущности, перезагружает данные в них.

Параметры:

- load_profile_id - идентификатор профиля загрузки
- environment_id - идентификатор среды разработки
- session_id – идентификатор сессии, по умолчанию нулевой

Сервисные процедуры

Ниже приведено описание сервисных процедур:

6. «meta.DropViewsIfRecreate» – удаляет сгенерированное представление

Параметры:

- load_profile_id - идентификатор профиля загрузки
- environment_id - идентификатор среды разработки

7. «meta.GetCurrentLoadingID» – получает текущий номер загрузки

Параметров не имеет.

Возвращаемое значение:

Номер текущий загрузки Vault.

8. «meta.GetDefaultValueFromType» – функция, которая получает дефолтное выражение из типа

Параметры:

- Тип данных PostgreSQL

Возвращаемое значение:

Таблица с информацией о значении по умолчанию.

9. «meta.GetLinkedTables» – получает связи между сущностями, чтобы построить модель

Параметры:

link_id – идентификатор Link

Возвращаемое значение:

Таблица с информацией о связанных таблицах

10. «meta.GetMergeProcName» – по наименованию сущности возвращает наименование хранимых процедур

Параметры:

- `schema_name` – наименование схемы
- `entity_id` – идентификатор сущности
- `change_freq` – частота изменения
- `load_profile_id` – идентификатор профиля загрузки

Возвращаемое значение:

Наименование хранимой процедуры

11. «`meta.GetSourcesTablesHub`» – выдает бизнес-ключи, которые есть в таблице-источнике

Параметры:

- `hub_id` – идентификатор Hub

Возвращаемое значение:

Таблица, содержащая информацию о бизнес-ключах

12. «`meta.GetTypeDeclaration`» - возвращает строку для создания типа в таблице

Параметры:

- `type` – наименование типа PostgreSQL
- `max_length` – максимальная длина типа
- `precision` – точность типа

Возвращаемое значение:

Строка, по которой можно создать значение в таблице.

13. «`meta.isProsedureWithShemaExists`» – проверяет существует ли хранимая процедура

Параметры:

- `procedureName`
- `schemaName`

14. «`meta.LoadHubSats`» – перегружает данные во всех `satellite` указанной сущности

Параметры:

- `hub_id` – идентификатор `hab`
- `session_id` – идентификатор сессии
- `table_trg_schema` -
- `view_trg_schema`
- `load_profile_id` - идентификатор профиля загрузки
- `environment_id` – идентификатор среды

15. «`meta.ScriptCreateHub`» – создает сущность `Hub`, таблицу и технический `Satellite`

Параметры:

- `sql_proc` – скрипт процедуры, возвращаемый параметр
- `hub_id` – идентификатор `Hub`
- `table_trg_schema` – схема для внутренних таблиц `MetaVault`

16. «meta.ScriptCreateLink» – создает линк

Параметры:

- sql_proc – скрипт процедуры, возвращаемый параметр
- link_id – идентификатор Link
- table_trg_schema – схема для внутренних таблиц MetaVault

17. «meta.scriptCreateSat» – создает сущность сателита

Параметры:

- sql_proc – скрипт процедуры, возвращаемый параметр
- change_freq – номер Satellite
- table_trg_schema – схема для внутренних таблиц MetaVault

18. «meta.scriptCreateViewForHub» – создает бизнес-представление

Параметры:

- load_profile_id идентификатор профиля загрузки
- environment_id идентификатор среды выполнения

19. «meta.ScriptMergeHub» – создает хранимую процедуру для перегрузки hub

20.

Параметры:

- hub_id – идентификатор Hub
- table_trg_schema – схема, в которой находится Hub

21. «meta.ScriptMergeLink» - создает хранимую процедуру для перегрузки link

Параметры:

- link_id – идентификатор Link
- table_trg_schema – схема, в которой находится Link

22. «meta.ScriptMergeSat» - создает хранимую процедуру для перегрузки данных сателита

Параметры:

- table_trg_schema – схема, в которой находится Satellite
- hub_id – идентификатор Hub, к которому принадлежит Satellite
- change_freq – номер Satellite в Hub
- load_profile_id – идентификатор профиля загрузки

23. «meta.updateEntityNames» – обновляет поле в таблице meta.InfoSettings, это обеспечивает уникальность наименований во внутренних таблицах MetaVault

Параметров не имеет.

24. «meta.write_error_to_log» – запись ошибки в таблицу «meta.log»

Параметры:

- session_id – идентификатор сессии
- profile_id – идентификатор профиля загрузки
- environment_id - идентификатор среды выполнения

- entity_id идентификатор сущности
- message – сообщение, которое будет записано в лог
- error_code – код ошибки
- error_context – контекст ошибки
- error_msg_detail – детальное описание ошибки

25. «meta.write_message_to_log» – записывает сообщение в лог, таблицу «meta.log»

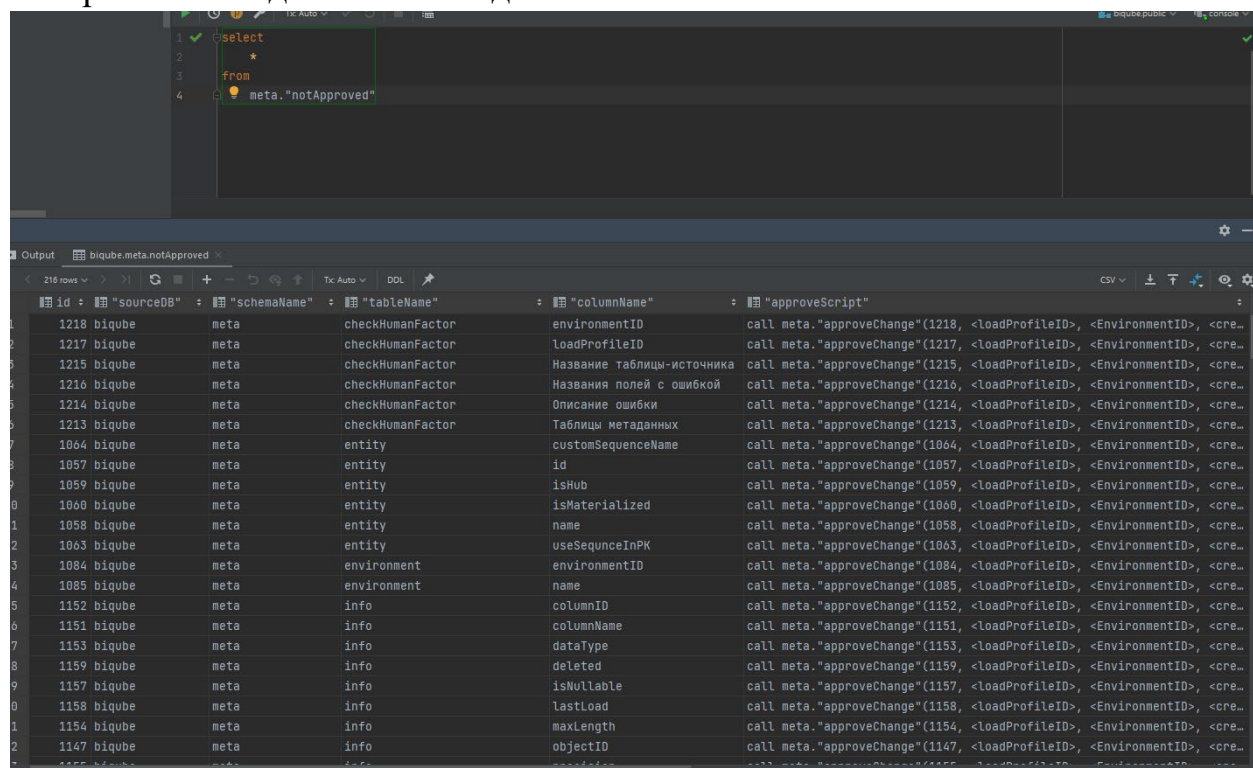
Параметры:

- session_id – идентификатор сессии
- profile_id – идентификатор профиля загрузки
- environment_id - идентификатор среды выполнения
- entity_id идентификатор сущности
- message – сообщение, которое будет записано в лог

3.3. Представления MetaVault

В процессе работы с MetaVault вручную может редактироваться только одно представление - meta.notApproved.

Представление meta.notApproved позволяет просматривать поля, которые можно добавить в модель Data Vault.



id	sourceDB	schemaName	tableName	columnName	approveScript
1218	biqube	meta	checkHumanFactor	environmentID	call meta."approveChange"(1218, <loadProfileID>, <EnvironmentID>, <cre...
1217	biqube	meta	checkHumanFactor	loadProfileID	call meta."approveChange"(1217, <loadProfileID>, <EnvironmentID>, <cre...
1215	biqube	meta	checkHumanFactor	Название таблицы-источника	call meta."approveChange"(1215, <loadProfileID>, <EnvironmentID>, <cre...
1216	biqube	meta	checkHumanFactor	Названия полей с ошибкой	call meta."approveChange"(1216, <loadProfileID>, <EnvironmentID>, <cre...
1214	biqube	meta	checkHumanFactor	Описание ошибки	call meta."approveChange"(1214, <loadProfileID>, <EnvironmentID>, <cre...
1213	biqube	meta	checkHumanFactor	Таблицы метаданных	call meta."approveChange"(1213, <loadProfileID>, <EnvironmentID>, <cre...
1064	biqube	meta	entity	customSequenceName	call meta."approveChange"(1064, <loadProfileID>, <EnvironmentID>, <cre...
1057	biqube	meta	entity	id	call meta."approveChange"(1057, <loadProfileID>, <EnvironmentID>, <cre...
1059	biqube	meta	entity	isHub	call meta."approveChange"(1059, <loadProfileID>, <EnvironmentID>, <cre...
1060	biqube	meta	entity	isMaterialized	call meta."approveChange"(1060, <loadProfileID>, <EnvironmentID>, <cre...
1058	biqube	meta	entity	name	call meta."approveChange"(1058, <loadProfileID>, <EnvironmentID>, <cre...
1063	biqube	meta	entity	useSequenceInPK	call meta."approveChange"(1063, <loadProfileID>, <EnvironmentID>, <cre...
1084	biqube	meta	environment	environmentID	call meta."approveChange"(1084, <loadProfileID>, <EnvironmentID>, <cre...
1085	biqube	meta	environment	name	call meta."approveChange"(1085, <loadProfileID>, <EnvironmentID>, <cre...
1152	biqube	meta	info	columnID	call meta."approveChange"(1152, <loadProfileID>, <EnvironmentID>, <cre...
1151	biqube	meta	info	columnName	call meta."approveChange"(1151, <loadProfileID>, <EnvironmentID>, <cre...
1153	biqube	meta	info	dataType	call meta."approveChange"(1153, <loadProfileID>, <EnvironmentID>, <cre...
1159	biqube	meta	info	deleted	call meta."approveChange"(1159, <loadProfileID>, <EnvironmentID>, <cre...
1157	biqube	meta	info	isNullable	call meta."approveChange"(1157, <loadProfileID>, <EnvironmentID>, <cre...
1158	biqube	meta	info	lastLoad	call meta."approveChange"(1158, <loadProfileID>, <EnvironmentID>, <cre...
1154	biqube	meta	info	maxLength	call meta."approveChange"(1154, <loadProfileID>, <EnvironmentID>, <cre...
1147	biqube	meta	info	objectID	call meta."approveChange"(1147, <loadProfileID>, <EnvironmentID>, <cre...

Рисунок 17. Представление «meta.notApproved»

Представление «meta.checkHumanFactor» проверяет метаданные на наличие ошибок, вызванных человеческим фактором. Если представление

выдает данные, то существуют ошибки, подсказка к исправлению находится в выводе представления.

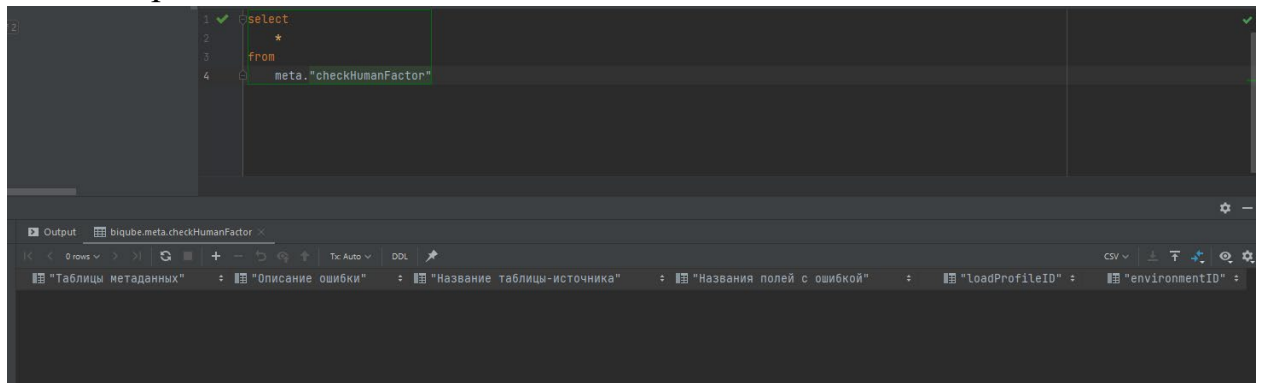


Рисунок 18. Представление «meta.checkHumanFactor»

3.4. Источники MetaVault

Источником могут выступать любые таблицы и представления в той же базе данных, что и MetaVault. Ниже приведен пример размещения объектов базы данных источника данных для MetaVault. Так по умолчанию предложена схема Public, содержащая демо-данные, представленные в реляционной форме для перекладки их в модель Data Vault.

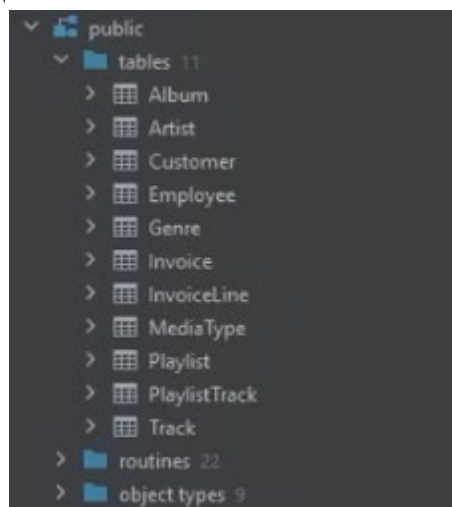


Рисунок 19. База «Public» - демо-база

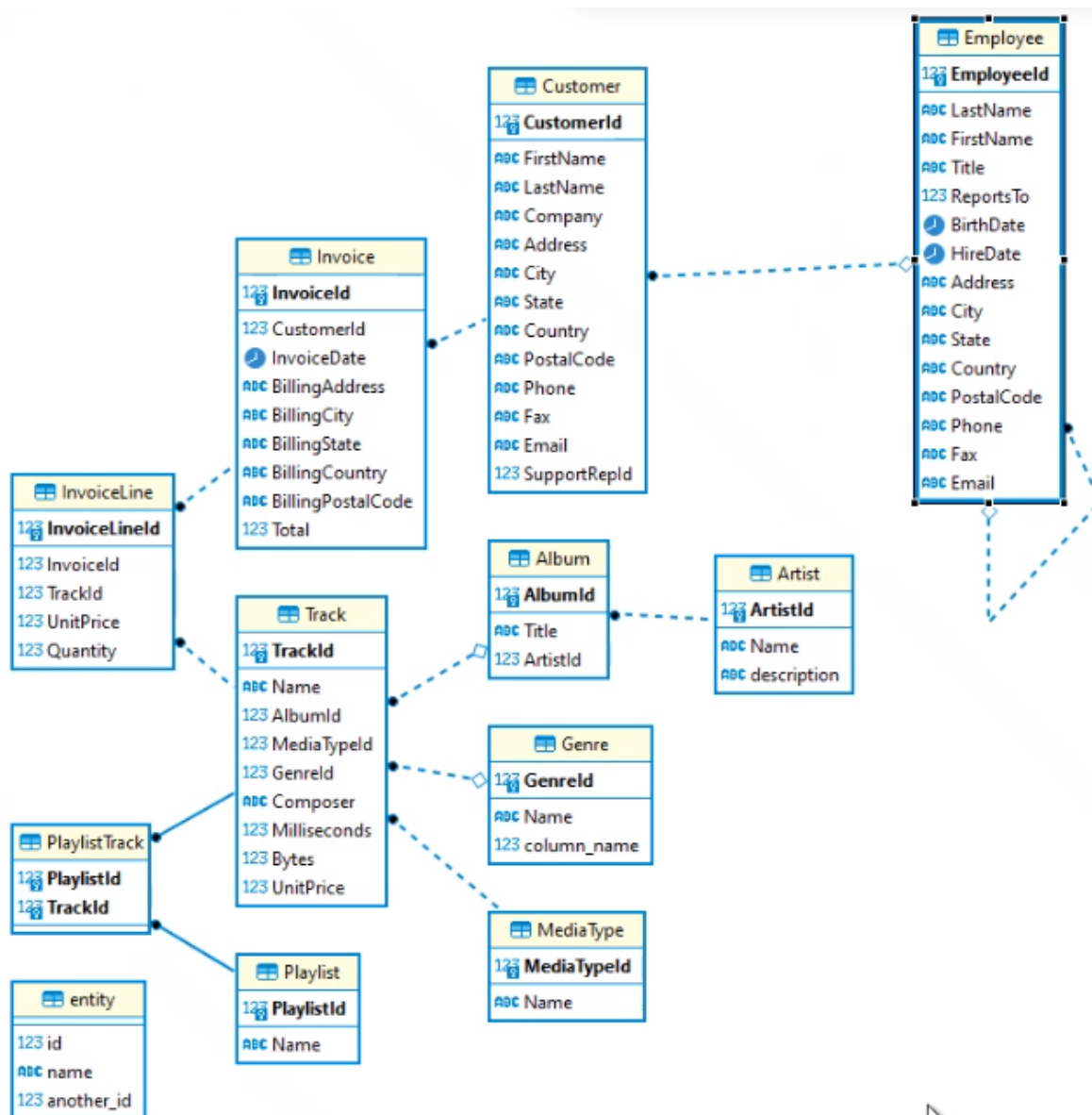


Рисунок 20. Ег-модель демо-базы

Далее в примерах будут использоваться две таблицы из этой базы: Album и Artist. На рисунке ниже представлена структура таблицы источника «Album».

	AlbumId	Title
1	3	Restless and Wild
2	4	Let There Be Rock
3	5	Big Ones
4	6	Jagged Little Pill
5	7	Facelift
6	8	Warner 25 Anos
7	9	Plays Metallica By Four Cellos
8	10	Audioslave
9	11	Out Of Exile
10	12	BackBeat Soundtrack
11	13	The Best Of Billy Cobham
12	14	Alcohol Fueled Brewtality Live! [Disc 1]
13	15	Alcohol Fueled Brewtality Live! [Disc 2]
14	16	Black Sabbath
15	17	Black Sabbath Vol. 4 (Remaster)
16	18	Body Count
17	19	Chemical Wedding
18	20	The Best Of Buddy Guy - The Millenium Collection
19	21	Prenda Minha
20	22	Sozinho Remix Ao Vivo
21	23	Minha Historia
22	24	Afrociberdelia
23	25	Da Lama Ao Caos
24	26	Acústico MTV [Live]
25	27	Cidade Negra - Hits
26	28	Na Pista

Рисунок 21. Таблица – источник «Album»

3.5. Структура сущности

Каждая сущность MetaVault имеет четкую структуру. В данном разделе представлена структура сущности «Album» из демо-таблицы «public.Album». На рисунке ниже выделены все таблицы относящиеся к «Album»:

Таблицы:

- hub_Album,
- hub_Album_Sat_Tecnical
- link_Album_Artist
- link_Album_Artist_Sat_Tecnical
- sat__Album_1

Хранимые процедуры:

- CreateHab_hub_Album,
- Createlink_Album_Artist
- CreateSat__Album_1
- MargeHab_Album
- Margelink_Album_Artist
- MargeSat__Album_cFreq_1_IProf

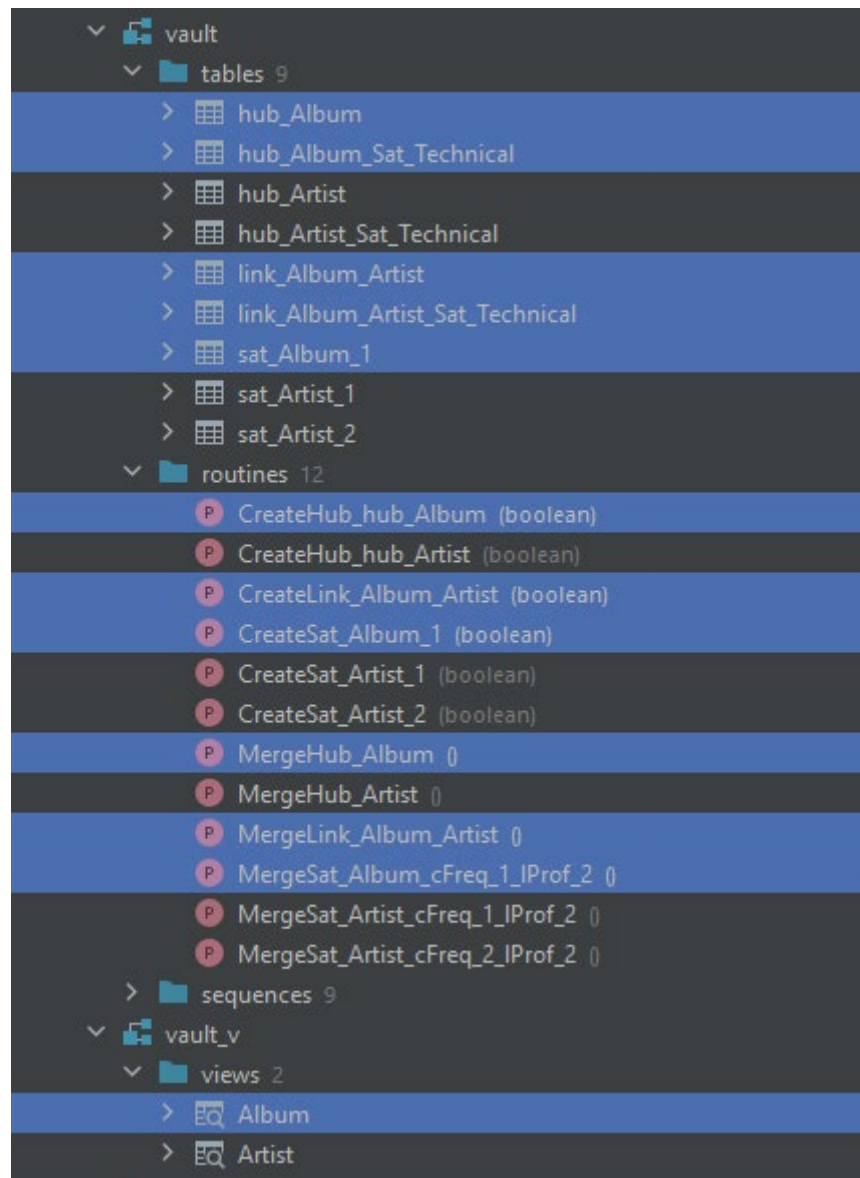


Рисунок 22. Объекты для работы с «Album»

По результатам работы MetaVault генерируется структура взаимосвязанных таблиц, представляющих разложение сущности в методологии Data Vault. Для удобства работы с моделью отдельно автоматически создается бизнес-представление, в котором находится только актуальная информация.

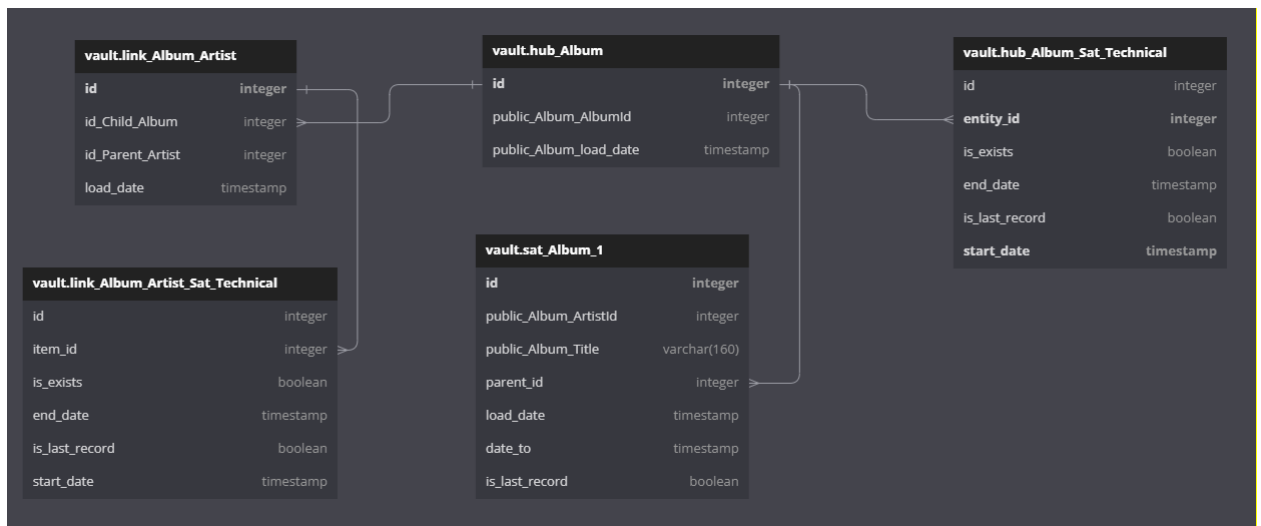


Рисунок 23 Структура взаимосвязанных таблиц MetaVault для сущности «Album»

В таблице «hub_Album» хранятся следующие атрибуты (Рисунок 24):

- Все бизнес-ключи, определяющие запись.
- Суррогатный ключ – порожденный ключ записи, порядковый номер, зачастую выбирается из последовательностей баз данных (sequence).
- Время загрузки – дата, регистрирующая момент, когда ключ впервые был загружен в хранилище

	id	public_Album_AlbumId	public_Album_LoadDate
1	1	2	2022-11-15 09:06:48.425403
2	2	3	2022-11-15 09:06:48.425403
3	3	4	2022-11-15 09:06:48.425403
4	4	5	2022-11-15 09:06:48.425403
5	5	6	2022-11-15 09:06:48.425403
6	6	7	2022-11-15 09:06:48.425403
7	7	8	2022-11-15 09:06:48.425403
8	8	9	2022-11-15 09:06:48.425403
9	9	10	2022-11-15 09:06:48.425403
10	10	11	2022-11-15 09:06:48.425403
11	11	12	2022-11-15 09:06:48.425403
12	12	13	2022-11-15 09:06:48.425403
13	13	14	2022-11-15 09:06:48.425403
14	14	15	2022-11-15 09:06:48.425403
15	15	16	2022-11-15 09:06:48.425403
16	16	17	2022-11-15 09:06:48.425403
17	17	18	2022-11-15 09:06:48.425403
18	18	19	2022-11-15 09:06:48.425403
19	19	20	2022-11-15 09:06:48.425403
20	20	21	2022-11-15 09:06:48.425403
21	21	22	2022-11-15 09:06:48.425403

Рисунок 24 Hub сущности «Album»

В таблице «hub_Album_Sat_Tecnical» хранятся следующие атрибуты (Рисунок 24):

- Идентификатор записи – суррогатный ключ из таблицы Hub.
- Метка существования записи.
- Метка признака последней записи.
- Дата начала действия – момент времени, когда запись впервые загружена или появилась после удаления.
- Дата конца действия – момент удаления записи.

id	entity_id	is_exists	end_date	is_last_record	start_date
1	1	true	9999-01-01 00:00:00.000000	true	2022-11-15 09:06:48.425403
2	2	true	9999-01-01 00:00:00.000000	true	2022-11-15 09:06:48.425403
3	3	true	9999-01-01 00:00:00.000000	true	2022-11-15 09:06:48.425403
4	4	true	9999-01-01 00:00:00.000000	true	2022-11-15 09:06:48.425403
5	5	true	9999-01-01 00:00:00.000000	true	2022-11-15 09:06:48.425403
6	6	true	9999-01-01 00:00:00.000000	true	2022-11-15 09:06:48.425403
7	7	true	9999-01-01 00:00:00.000000	true	2022-11-15 09:06:48.425403
8	8	true	9999-01-01 00:00:00.000000	true	2022-11-15 09:06:48.425403
9	9	true	9999-01-01 00:00:00.000000	true	2022-11-15 09:06:48.425403
10	10	true	9999-01-01 00:00:00.000000	true	2022-11-15 09:06:48.425403
11	11	true	9999-01-01 00:00:00.000000	true	2022-11-15 09:06:48.425403
12	12	true	9999-01-01 00:00:00.000000	true	2022-11-15 09:06:48.425403
13	13	true	9999-01-01 00:00:00.000000	true	2022-11-15 09:06:48.425403
14	14	true	9999-01-01 00:00:00.000000	true	2022-11-15 09:06:48.425403
15	15	true	9999-01-01 00:00:00.000000	true	2022-11-15 09:06:48.425403
16	16	true	9999-01-01 00:00:00.000000	true	2022-11-15 09:06:48.425403
17	17	true	9999-01-01 00:00:00.000000	true	2022-11-15 09:06:48.425403
18	18	true	9999-01-01 00:00:00.000000	true	2022-11-15 09:06:48.425403
19	19	true	9999-01-01 00:00:00.000000	true	2022-11-15 09:06:48.425403
20	20	true	9999-01-01 00:00:00.000000	true	2022-11-15 09:06:48.425403
21	21	true	9999-01-01 00:00:00.000000	true	2022-11-15 09:06:48.425403

Рисунок 25 Hub Technical Satellite сущности «Album»

Сущность «link_Album_Artist» содержит следующие атрибуты (Рисунок 26):

- Идентификатор сущности родителя - суррогатный ключ из таблицы Hub.
- Идентификатор сущности наследника - суррогатный ключ из таблицы Hub.
- Время загрузки – дата, регистрирующая момент, когда появилась связь.

	id	id_Child_Album	id_Parent_Artist	load_date
1	1	1	2	2022-11-15 09:06:49.920094
2	2	2	2	2022-11-15 09:06:49.920094
3	3	3	1	2022-11-15 09:06:49.920094
4	4	4	3	2022-11-15 09:06:49.920094
5	5	5	4	2022-11-15 09:06:49.920094
6	6	6	5	2022-11-15 09:06:49.920094
7	7	7	6	2022-11-15 09:06:49.920094
8	8	8	7	2022-11-15 09:06:49.920094
9	9	9	8	2022-11-15 09:06:49.920094
10	10	10	8	2022-11-15 09:06:49.920094
11	11	11	9	2022-11-15 09:06:49.920094
12	12	12	10	2022-11-15 09:06:49.920094
13	13	13	11	2022-11-15 09:06:49.920094
14	14	14	11	2022-11-15 09:06:49.920094
15	15	15	12	2022-11-15 09:06:49.920094
16	16	16	12	2022-11-15 09:06:49.920094
17	17	17	13	2022-11-15 09:06:49.920094
18	18	18	14	2022-11-15 09:06:49.920094
19	19	19	15	2022-11-15 09:06:49.920094
20	20	20	16	2022-11-15 09:06:49.920094
21	21	21	16	2022-11-15 09:06:49.920094

```

[2022-11-15 12:21:16] Connected to biqube
biqube.public> SELECT t.*
FROM vault."link_Album_Artist" t
LIMIT 501
  
```

Рисунок 26 Link сущности «Album»

В таблице «link_Album_Artist_Sat_Tecnical» хранятся следующие атрибуты (Рисунок 27):

- Идентификатор записи – суррогатный ключ из таблицы Link.
- Метка существования записи.
- Метка признака последней записи.
- Дата начала действия – момент времени, когда запись впервые загружена или появилась после удаления.
- Дата конца действия – момент удаления записи

	id	item_id	is_exists	end_date	is_last_record	start_date
1	1	1	* true	9999-01-01 00:00:00.000000	* true	2022-11-15 09:06:49.920094
2	2	2	* true	9999-01-01 00:00:00.000000	* true	2022-11-15 09:06:49.920094
3	3	3	* true	9999-01-01 00:00:00.000000	* true	2022-11-15 09:06:49.920094
4	4	4	* true	9999-01-01 00:00:00.000000	* true	2022-11-15 09:06:49.920094
5	5	5	* true	9999-01-01 00:00:00.000000	* true	2022-11-15 09:06:49.920094
6	6	6	* true	9999-01-01 00:00:00.000000	* true	2022-11-15 09:06:49.920094
7	7	7	* true	9999-01-01 00:00:00.000000	* true	2022-11-15 09:06:49.920094
8	8	8	* true	9999-01-01 00:00:00.000000	* true	2022-11-15 09:06:49.920094
9	9	9	* true	9999-01-01 00:00:00.000000	* true	2022-11-15 09:06:49.920094
10	10	10	* true	9999-01-01 00:00:00.000000	* true	2022-11-15 09:06:49.920094
11	11	11	* true	9999-01-01 00:00:00.000000	* true	2022-11-15 09:06:49.920094
12	12	12	* true	9999-01-01 00:00:00.000000	* true	2022-11-15 09:06:49.920094
13	13	13	* true	9999-01-01 00:00:00.000000	* true	2022-11-15 09:06:49.920094
14	14	14	* true	9999-01-01 00:00:00.000000	* true	2022-11-15 09:06:49.920094
15	15	15	* true	9999-01-01 00:00:00.000000	* true	2022-11-15 09:06:49.920094
16	16	16	* true	9999-01-01 00:00:00.000000	* true	2022-11-15 09:06:49.920094
17	17	17	* true	9999-01-01 00:00:00.000000	* true	2022-11-15 09:06:49.920094
18	18	18	* true	9999-01-01 00:00:00.000000	* true	2022-11-15 09:06:49.920094
19	19	19	* true	9999-01-01 00:00:00.000000	* true	2022-11-15 09:06:49.920094
20	20	20	* true	9999-01-01 00:00:00.000000	* true	2022-11-15 09:06:49.920094
21	21	21	* true	9999-01-01 00:00:00.000000	* true	2022-11-15 09:06:49.920094

```

[2022-11-15 12:21:37] Connected to biqube
biqube.public> SELECT t.*, CTID
FROM vault."link_Album_Artist_Sat_Tecnical" t
LIMIT 501
  
```

Рисунок 27 Technical Link сущности «Album»

Сущности Satellite содержат описательную информацию ключа сущности Hub. Данная информация подвергается изменениям с течением времени, и поэтому структура Satellite должна быть приспособлена для решения хранения как новой или измененной, так и исторической информации.

Таблица сущности «sat__Album_1» содержит атрибуты (Рисунок 28):

- Одно или несколько полей, описывающих ключ.
- Идентификатор записи - суррогатный ключ из таблицы Hub.
- Метка признака последней записи.
- Дата начала действия – момент времени, когда запись впервые загружена или изменились одно, или несколько полей, описывающих ключ.
- Дата конца действия – момент времени перед тем, как изменились одно или несколько полей, описывающих ключ.

id	public_Album_ArtistId	public_Album_Title	parent_id	load_date	date_to	is_last_record
1	157	Un-Led-Ed	251	2022-11-15 09:06:48.672872	9999-01-01 00:00:00.000000	true
2	200	Every Kind of Light	264	2022-11-15 09:06:48.672872	9999-01-01 00:00:00.000000	true
3	137	Live [Disc 2]	209	2022-11-15 09:06:48.672872	9999-01-01 00:00:00.000000	true
4	211	Bach: Goldberg Variations	276	2022-11-15 09:06:48.672872	9999-01-01 00:00:00.000000	true
5	50	Master Of Puppets	151	2022-11-15 09:06:48.672872	9999-01-01 00:00:00.000000	true
6	159	Aquaman	253	2022-11-15 09:06:48.672872	9999-01-01 00:00:00.000000	true
7	94	Are You Experienced?	119	2022-11-15 09:06:48.672872	9999-01-01 00:00:00.000000	true
8	8	Revelations	270	2022-11-15 09:06:48.672872	9999-01-01 00:00:00.000000	true
9	90	Live After Death	101	2022-11-15 09:06:48.672872	9999-01-01 00:00:00.000000	true
10	269	Liszt - 12 Études D'Execution...	339	2022-11-15 09:06:48.672872	9999-01-01 00:00:00.000000	true
11	85	My Way: The Best Of Frank Si...	82	2022-11-15 09:06:48.672872	9999-01-01 00:00:00.000000	true
12	19	Acústico MTV [Live]	25	2022-11-15 09:06:48.672872	9999-01-01 00:00:00.000000	true
13	140	The Doors	213	2022-11-15 09:06:48.672872	9999-01-01 00:00:00.000000	true
14	19	Cidade Negra - Hits	26	2022-11-15 09:06:48.672872	9999-01-01 00:00:00.000000	true
15	22	The Song Remains The Same (D...	136	2022-11-15 09:06:48.672872	9999-01-01 00:00:00.000000	true
16	143	Tangents	218	2022-11-15 09:06:48.672872	9999-01-01 00:00:00.000000	true
17	59	Santana - As Years Go By	196	2022-11-15 09:06:48.672872	9999-01-01 00:00:00.000000	true
18	149	Lost, Season 2	230	2022-11-15 09:06:48.672872	9999-01-01 00:00:00.000000	true
19	50	Kill 'Em All	149	2022-11-15 09:06:48.672872	9999-01-01 00:00:00.000000	true
20	90	A Matter of Life and Death	93	2022-11-15 09:06:48.672872	9999-01-01 00:00:00.000000	true
21	243	Tchaikovsky: 1812 Festival O...	307	2022-11-15 09:06:48.672872	9999-01-01 00:00:00.000000	true

```

[2022-11-15 12:19:23] Connected to biqube
biqube> public> SELECT t.*
FROM vault.\"sat_Album_1\" t
LIMIT 501

```

Рисунок 28 Satellite сущности «Album»