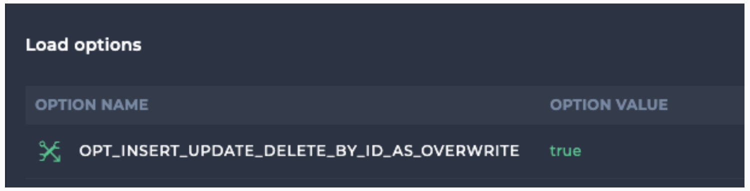
Reference table (6.1)

- Product type reference data:
  - Default entity type REF was used to save the reference data about the product type
  - Default loading logic of REF-table is configurable per environment.
    - There are various pre-defined options, such as truncate-insert, insert-overwrite or historize the reference-table if needed
  - The default loading logic in this environment was defined as INSERT OVERWRITE to reference table

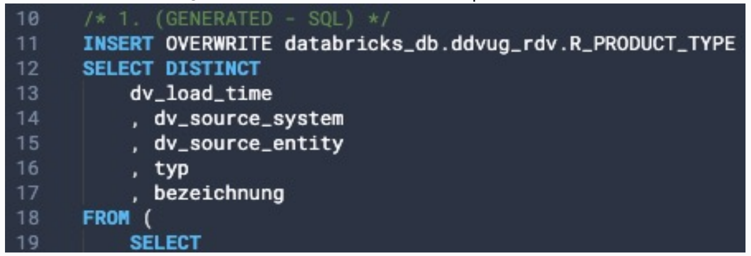
Defined option in ADE load:



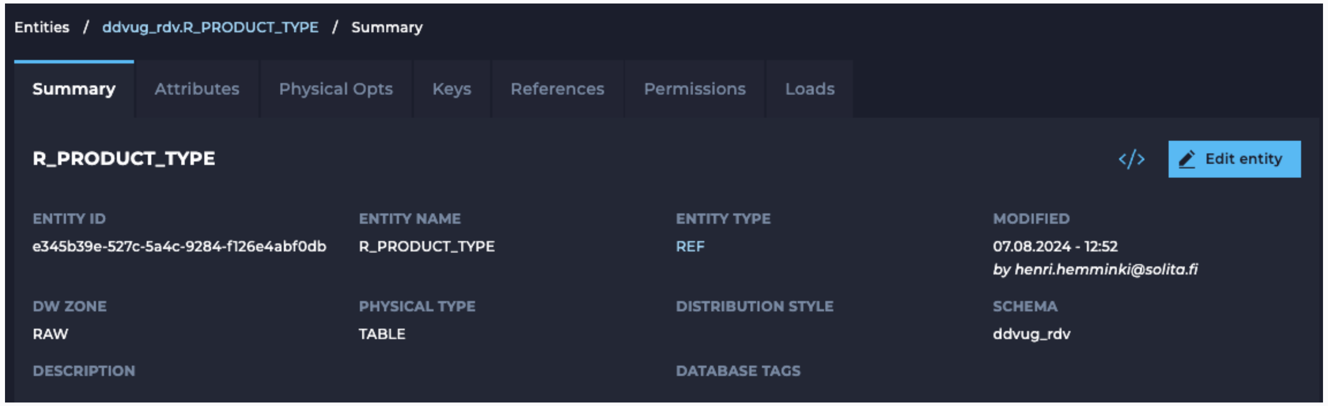
Load options

OPTION NAME	OPTION VALUE
OPT_INSERT_UPDATE_DELETE_BY_ID_AS_OVERWRITE	true

Generated SQL for the above load option:



```sql
/* 1. (GENERATED - SQL) */
INSERT OVERWRITE databricks_db.ddvug_rdv.R_PRODUCT_TYPE
SELECT DISTINCT
    dv_load_time
    , dv_source_system
    , dv_source_entity
    , typ
    , bezeichnung
FROM (
    SELECT
```



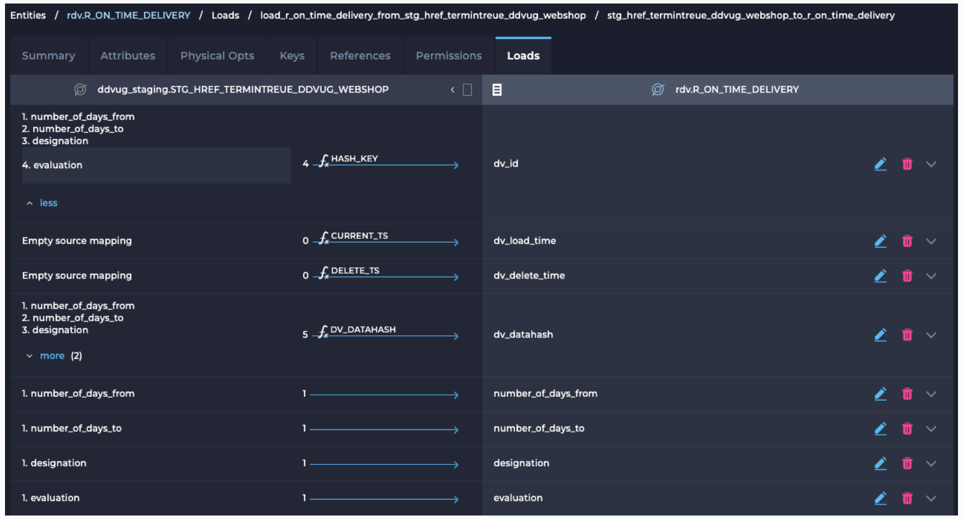
R_PRODUCT_TYPE

ENTITY ID	ENTITY NAME	ENTITY TYPE	MODIFIED
e345b39e-527c-5a4c-9284-f126e4abf0db	R_PRODUCT_TYPE	REF	07.08.2024 - 12:52 *by henri.hemminki@solita.fi*
DW ZONE	**PHYSICAL TYPE**	**DISTRIBUTION STYLE**	**SCHEMA**
RAW	TABLE		ddvug_rdv
DESCRIPTION		**DATABASE TAGS**	

Agile Data ENGINE

Reference table (6.2)

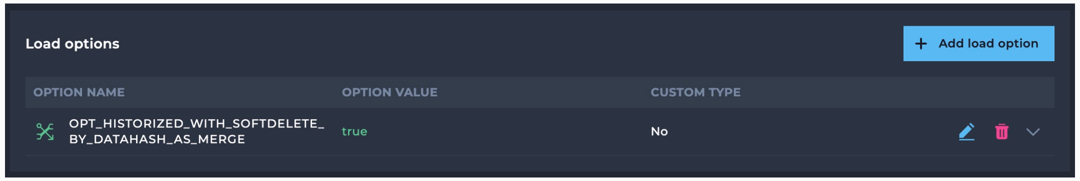
- Historized reference table:
  - R_ON_TIME_DELIVERY historized reference table was created with the following mappings:



Entities / rdv.R_ON_TIME_DELIVERY / Loads / load_r_on_time_delivery_from_stg_href_termintreue_ddvug_webshop / stg_href_termintreue_ddvug_webshop_to_r_on_time_delivery

ddvug_staging.STG_HREF_TERMINTREUE_DDVUG_WEBSHOP		rdv.R_ON_TIME_DELIVERY
1. number_of_days_from 2. number_of_days_to 3. designation 4. evaluation	4 HASH_KEY	dv_id
Empty source mapping	0 CURRENT_TS	dv_load_time
Empty source mapping	0 DELETE_TS	dv_delete_time
1. number_of_days_from 2. number_of_days_to 3. designation more (2)	5 DV_DATAHASH	dv_datahash
1. number_of_days_from	1	number_of_days_from
1. number_of_days_to	1	number_of_days_to
1. designation	1	designation
1. evaluation	1	evaluation

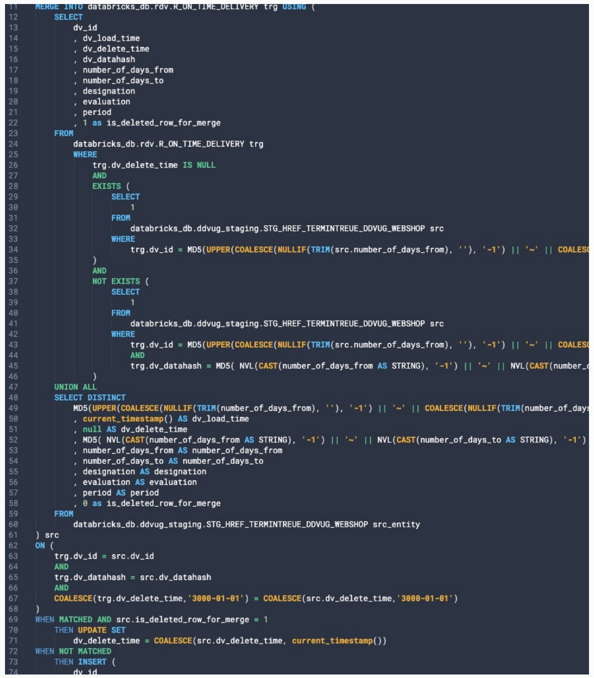
  - A merge pattern with full historization using soft deletes was selected to keep track of full history:



Load options

OPTION NAME	OPTION VALUE	CUSTOM TYPE
OPT_HISTORIZED_WITH_SOFTDELETE_BY_DATAHASH_AS_MERGE	true	No

Generated SQL for the merge:

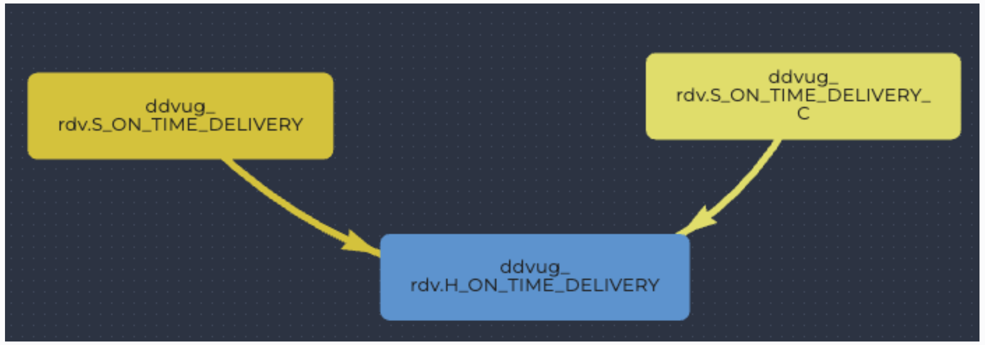


```sql
MERGE INTO databricks_db.rdv.R_ON_TIME_DELIVERY trg USING (
    SELECT
        dv_id
        , dv_load_time
        , dv_delete_time
        , dv_datahash
        , number_of_days_from
        , number_of_days_to
        , designation
        , evaluation
        , period
        , 1 as is_deleted_row_for_merge
    FROM
        databricks_db.rdv.R_ON_TIME_DELIVERY trg
        WHERE
            trg.dv_delete_time IS NULL
            AND
            EXISTS (
                SELECT
                    1
                FROM
                    databricks_db.ddvug_staging.STG_HREF_TERMINTREUE_DDVUG_WEBSHOP src
                WHERE
                    trg.dv_id = MD5(UPPER(COALESCE(NULLIF(TRIM(src.number_of_days_from), ''), '-1') || '~' || COALESC
            )
            AND
            NOT EXISTS (
                SELECT
                    1
                FROM
                    databricks_db.ddvug_staging.STG_HREF_TERMINTREUE_DDVUG_WEBSHOP src
                WHERE
                    trg.dv_id = MD5(UPPER(COALESCE(NULLIF(TRIM(src.number_of_days_from), ''), '-1') || '~' || COALESC
                    AND
                    trg.dv_datahash = MD5( NVL(CAST(number_of_days_from AS STRING), '-1') || '~' || NVL(CAST(number_
            )
    UNION ALL
    SELECT DISTINCT
        MD5(UPPER(COALESCE(NULLIF(TRIM(number_of_days_from), ''), '-1') || '~' || COALESCE(NULLIF(TRIM(number_of_days
        , current_timestamp() AS dv_load_time
        , null AS dv_delete_time
        , MD5( NVL(CAST(number_of_days_from AS STRING), '-1') || '~' || NVL(CAST(number_of_days_to AS STRING), '-1')
        , number_of_days_from AS number_of_days_from
        , number_of_days_to AS number_of_days_to
        , designation AS designation
        , evaluation AS evaluation
        , period AS period
        , 0 as is_deleted_row_for_merge
    FROM
        databricks_db.ddvug_staging.STG_HREF_TERMINTREUE_DDVUG_WEBSHOP src_entity
) src
ON (
    trg.dv_id = src.dv_id
    AND
    trg.dv_datahash = src.dv_datahash
    AND
    COALESCE(trg.dv_delete_time,'3000-01-01') = COALESCE(src.dv_delete_time,'3000-01-01')
)
WHEN MATCHED AND src.is_deleted_row_for_merge = 1
    THEN UPDATE SET
        dv_delete_time = COALESCE(src.dv_delete_time, current_timestamp())
WHEN NOT MATCHED
    THEN INSERT (
        dv_id
```

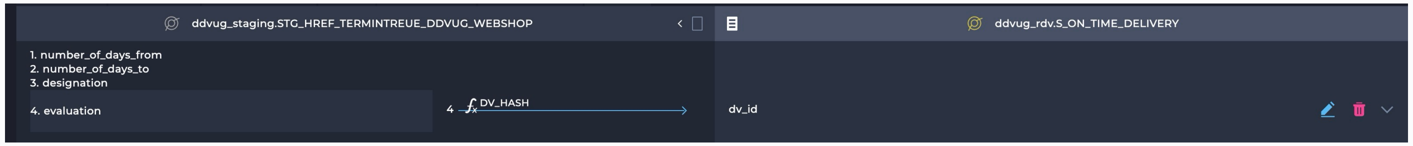
Agile Data ENGINE

Reference table (6.3)

- Historized reference data, alternative approach:
  - One way to historize the reference data is to model it as a standard hub and a satellite:


ddvug_ rdv.S_ON_TIME_DELIVERY
ddvug_ rdv.S_ON_TIME_DELIVERY_ C
ddvug_ rdv.H_ON_TIME_DELIVERY

  - Primary key dv_id (for both hub and satellite) is populated from all fields to handle missing business key:


ddvug_staging.STG_HREF_TERMINTREUE_DDVUG_WEBSHOP

1. number_of_days_from
2. number_of_days_to
3. designation
4. evaluation

4 DV_HASH → dv_id

ddvug_rdv.S_ON_TIME_DELIVERY