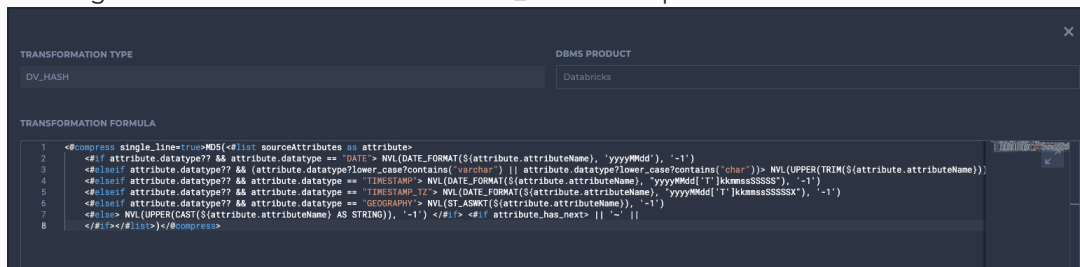


# Records without Business Key (8.1)

- Records without business key ("zero keys") are loaded to hub with unknown business key, marked with '-1'
  - Default loading logic sets unknown business key to value '-1'
  - The unknown value -logic is configurable in ADE, so it can be changed in case different logic is needed

Configured transformation formula DV\_HASH for specific DBMS:



TRANSFORMATION TYPE: DV\_HASH  
DBMS PRODUCT: Databricks

```
1 <#compress single_line=#>MD5(<#if sourceAttributes == attribute>  
2 <#if attribute.datatype? && attribute.datatype == 'DATE'> NVL( DATE_FORMAT($@attribute.attributeName, 'yyyyMMdd'), '-1')  
3 <#elseif attribute.datatype? && (attribute.datatype?lower_case?contains('varchar') || attribute.datatype?lower_case?contains('char'))> NVL(UPPER(TRIM($@attribute.attributeName)))  
4 <#elseif attribute.datatype? && attribute.datatype == 'TIMESTAMP'> NVL( DATE_FORMAT($@attribute.attributeName, 'yyyyMMdd[T]kkmmssSSSS'), '-1')  
5 <#elseif attribute.datatype? && attribute.datatype == 'TIMESTAMP_TZ'> NVL( DATE_FORMAT($@attribute.attributeName, 'yyyyMMdd[T]kkmmssSSSx'), '-1')  
6 <#elseif attribute.datatype? && attribute.datatype == 'GEOGRAPHY'> NVL(ST_ASWKT($@attribute.attributeName), '-1')  
7 <#else> NVL(UPPER(CAST($@attribute.attributeName AS STRING)), '-1') </#if> <#if attribute.has_next? || '-' ||  
8 </#if></#if></#compress>
```

Applied transformation formula in load SQL:

```
SELECT  
MD5( NVL(UPPER(CAST(bestellungid AS STRING)), '-1') ) AS dv_id
```



Just now (1s) 22

```
1 select  
2   dv_id,  
3   business_key,  
4   dv_source_system  
5 from ddvug_rdv.h_delivery_service  
6 where business_key = '-1';
```

> [See performance \(1\)](#)

Table	dv_id	business_key	dv_source_system
1	6bb61e3b7bce0931da574d19d1d82c88	-1	WEBSHOP

# Records without Business Key (8.2)

- To satellite table, records without business key should not be loaded
- In ADE, it is possible to enhance default satellite loading logic with the additional load steps, to handle "zero keys":
  - Only load to satellite valid business keys, meaning no zero keys should be inserted. Adding OPT\_WHERE condition to filter out rows from the staging-tables, where business key is NULL
  - Adding PRE or POST load step to load zero business keys to Error Vault.
- It is good practice to make ADE template of the above logic to make it reusable

**Entity Load Logic**

Import load logic from template  
Select one item from the list

ENTITY TYPE	ENTITY LOAD TYPE	TEMPLATE NAME	TEMPLATE DESCRIPTION
SAT	TRANSFORM_PERSIST	SELECT_NEW_ROWS_FROM_SOURCE_BY_INSERT_TIME	Uses OPT_WHERE and source insert time to select only the new rows from the source
SAT	TRANSFORM_PERSIST	error_vault_loading	Adding POST step to satellites to load Error Vault, in case "zero keys" appear in business keys.
SAT	TRANSFORM_PERSIST	dv_sat_link_ssat_runid_options	

[Import load logic from template](#)

**Load steps**

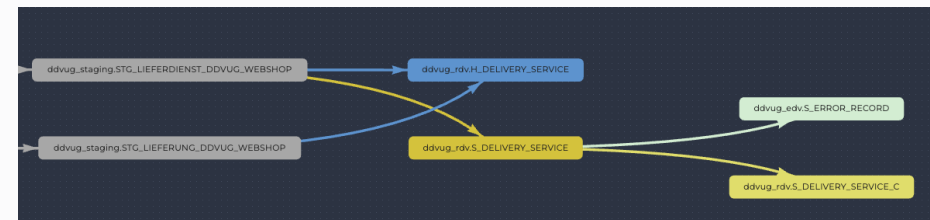
LOAD STEP NAME	TYPE	DBMS	LOAD STEP ENABLED
1 load_error_vault	POST		true
2 dq_business_key_null	SMOKE_GREY		true

**Load options**

OPTION NAME	OPTION VALUE	CUSTOM TYPE
OPT_WHERE	lieferdienstid is not null	No

# Records without Business Key (8.3)

- With the error vault template, a satellite entity was created for the error records
- The previously mentioned template is possible to add to satellite loads to prevent inserting error records
- All error records are stored in the same error satellite:



```
1 select
2   dv_id,
3   dv_load_time,
4   dv_datahash,
5   records
6 from ddvug_edv.s_error_record;
```

dv_id	dv_load_time	dv_datahash	records
1943352b0232c75e0b7b126692c3bf16	2024-09-16T10:07:12.623+00:00	4e567b1bd15dbb9ae4356a3d67c4737d	{\"dv_id\": \"6bb61e3b7bce0931da574d19d1d82c88\", \"dv_load_time\": \"2024-09-16T10:07:12.623Z\", \"dv_run_id\": \"1726481230652\", \"dv_source_system\": \"WEBSHOP\", \"dv_source_entity\": \"PRODUKT\", \"dv_package_version\": \"1002698\", \"dv_load_name\": \"load_s_delivery_service_from_stg_lieferdienst_ddvug_webshop_01_db\", \"dv_datahash\": \"4e567b1bd15dbb9ae4356a3d67c4737d\", \"name\": \"Femen AG\", \"telefon\": \"06257/27937692\", \"email\": \"l_platzek@hoster.none\", \"strasse\": \"Westkirchener Stra\u00dfe\", \"hausnummer\": \"198\", \"plz\": \"9125\", \"ort\": \"Chemnitz\"}
51f3195c7d831b76b3a1397ea51fb578	2024-09-16T10:07:12.623+00:00	740fb434bbc18b1d1e2dea359c21e1a3	{\"dv_id\": \"6bb61e3b7bce0931da574d19d1d82c88\", \"dv_load_time\": \"2024-09-16T10:07:12.623Z\", \"dv_run_id\": \"17264812...

2 rows | 1.44 seconds runtime

# Records without Business Key (8.4)

- To detect data issues, ADE has built-in data tests, called data smoke tests
  - In this case, a smoke grey test could be added to satellite load to detect "zero keys" and notify when those occur
  - A smoke grey is logged to data quality dashboards and if notification channel is configured, the information about failed test is sent there

Smoke grey test defined to satellite load

The screenshot shows the configuration for a smoke test named 'dq\_business\_key\_null'. The 'LOAD STEP NAME' is 'dq\_business\_key\_null' and the 'TYPE' is 'SMOKE\_GREY'. The 'LANGUAGE' is 'SQL' and the 'POSITION IN EXECUTION ORDER' is '2'. The 'DBMS PRODUCT SPECIFIC (OPTIONAL)' is '(not selected)'. The 'READ AFFECTED ROWS' checkbox is unchecked, and the 'LOAD STEP ENABLED' checkbox is checked. The 'LOGIC' section contains the following SQL query:

```
1 select
2   count(1)
3 from <source_entity_schema>.<source_entity_name>
4 where lieferdienstid is null
5 having count(1) > 1
```

ADE will log failed smoke tests and display those on Runtime dashboards

Smoke type	DAG name	DAG run ID	Process name	End time	Test name	Test query	Failed rows
GREY SMOKE	DDVUG_WEBSHOP_DB	manual_2024-08-29T10:48:45+00:00	load_s_delivery_service_from_stg_lieferdienst_ddvug_webshop_01_db	2024-08-29 13:49:51	dq_business_key_null	select count(1) from <source_entity_schema>.<source_entity_name> where lieferdienstid is null having count(1) > 1;	1