Duplicates in the loading data (7.1)

- In the duplicate case, there are different cases of duplicates, an actual duplicate and a product with same produktid 21, but it is a different product
- To handle these cases, there are various options in ADE
  - Option 1: insert all records and use business rules
  - Option 2: prevent loading to satellite
- Duplicates were added to delivery 3, so the incoming data looks like the following for produktid:s 20 and 21

produktid	katid	bezeichnung	preis
20	GMiCHILI	Chili, Jalapeno „Ruben"	8.00
20	GMiCHILI	Chili, Jalapeno „Ruben"	8.00
21	GMiCHILI	Chili „Vietnam - Landsorte"	4.00
21	GMiPFEFFERO...	Pfefferoni „Vietnamese"	3.50

Duplicates in the loading data (7.2)

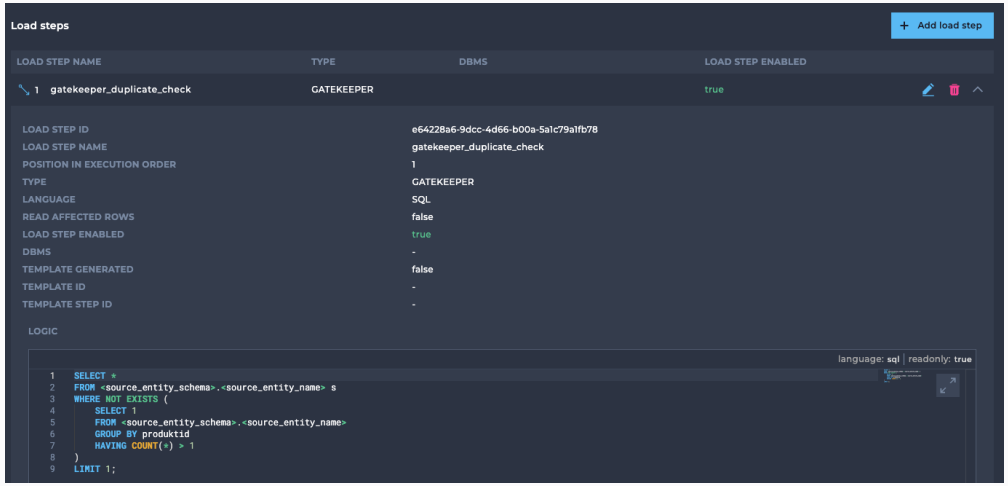
- Option 1: Load the data to satellite as is, meaning duplicates from the same batch will be inserted. On default, this is possible with ADE's satellite loading logic
  - A business rule would be created in Business Data Vault or Data Mart to identify the correct product for produktid 21
  - Source system experts should be contacted on what is the actual product and why the erroneous data occurred
  - In these cases, also a data test (smoke grey) would be added to ADE loading logic, to notify if these cases happen again

ADE satellite logic allows inserting duplicate business keys (produktid 21) in the same batch.
Since produktid 20 was delivered again with the original value and with the same datahash, the satellite logic will historize it again. This is due to it changing in delivery 2 to something else and back to original in delivery 3.

dv_id	dv_load_time	dv_run_id	dv_datahash	produktid	katid	umfang
98f13708210194c475687be6106a3b84	2024-09-13T11:06:10.772+00:00	1726225568196	f5bb76accc8e12175fac78123867caa9	20	GMiCHILI	6
3c59dc048e8850243be8079a5c74d079	2024-09-13T11:06:10.772+00:00	1726225568196	8c625db76f87a24e31ddcbf6735d57df	21	GMiCHILI	6
98f13708210194c475687be6106a3b84	2024-09-13T11:15:34.476+00:00	1726226132548	07bb9add3d428c163d4df1fb9110da57	20	MiGCHILI	6
3c59dc048e8850243be8079a5c74d079	2024-09-13T11:15:34.476+00:00	1726226132548	fc1d4add821420986d1b2183ee68e501	21	MiGCHILI	6
98f13708210194c475687be6106a3b84	2024-09-16T09:48:17.438+00:00	1726480092977	f5bb76accc8e12175fac78123867caa9	20	GMiCHILI	6
3c59dc048e8850243be8079a5c74d079	2024-09-16T09:48:17.438+00:00	1726480092977	8c625db76f87a24e31ddcbf6735d57df	21	GMiCHILI	6
3c59dc048e8850243be8079a5c74d079	2024-09-16T09:48:17.438+00:00	1726480092977	346e5590316f850b01b1fb6095f7eec7	21	GMiPFEFFERO...	7

Duplicates in the loading data (7.3)

- Option 2: Stopping duplicate business keys to be loaded to a satellite
  - ADE has pre-defined load step type called GATEKEEPER, which can be used to prevent data loading
  - In this case, a GATEKEEPER was added to check if there are duplicates by produktid and thus it will stop loading the satellite
  - Grey smoke test was added to test out and notify of duplicate business keys




```sql
/* 1. gatekeeper_duplicate_check (GATEKEEPER - SQL) */
SELECT *
FROM ddvug_staging.STG_PRODUKT_DDVUG_WEBSHOP s
WHERE NOT EXISTS (
    SELECT 1
    FROM ddvug_staging.STG_PRODUKT_DDVUG_WEBSHOP
    GROUP BY produktid
    HAVING COUNT(*) > 1
)
LIMIT 1;

/* 2. (GENERATED - SQL) */
INSERT INTO databricks_db.ddvug_rdv.S_PRODUCT_WEBSHOP (
```

```sql
/* 3. smoke_test_duplicate_bk (SMOKE_GREY - SQL) */
SELECT
    COUNT(*),
    produktid
FROM ddvug_staging.STG_PRODUKT_DDVUG_WEBSHOP
GROUP BY produktid
HAVING COUNT(*) > 1;
```

Smoke type	DAG name	DAG run ID	Process name	End time	Test name
GREY SMOKE	DDVUG_WEBSHOP_DB	manual__2024-09-18T11:16:54+00:00	load_stg_produkt_ddvug_webshop_from_produkt_01_db	2024-09-18 14:17:26	smoke_test_duplicate_bk