DBT Test - Store Failures

When a pre-configured test is run, any failures it encounters will be stored when used ***dbt test --store-failures*** command. If you set ***store_failures*** config to either true or false in yaml configuration file, the configuration will have more precedence over the "--store-failures" flag.

POC steps for test runs

1. **Test Cases Development:** Both built-in tests and custom test cases were used. For built-in test cases, ***not_null*** constraint was used in ***location_id*** column of ***tmp_location*** model and configured to store failures. For custom test cases, ***primary key violation*** and ***foreign key violation*** tests were developed in SQL and run with ***dbt test –store-failures*** command.

Figure 1: store failure configuration set to true

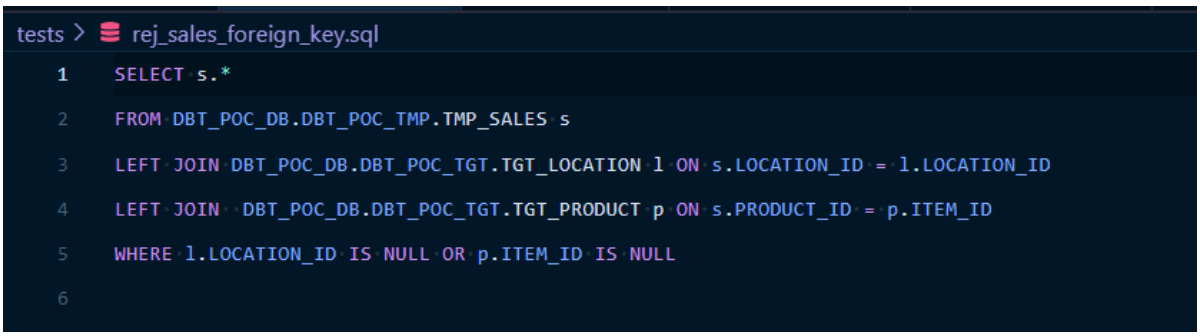


Figure 2: foreign key violation test

tests > test_primary_key.sql

You, 2 days ago | 1 author (You)

```sql
SELECT
    'TMP_PRODUCT' as table_name,
    ITEM_ID as primary_key_column,
    COUNT(*) as num_records
FROM DBT_POC_DB.DBT_POC_TMP.TMP_PRODUCT
GROUP BY ITEM_ID
HAVING COUNT(*) > 1

UNION ALL

SELECT
    'TMP_LOCATION' as table_name,
    LOCATION_ID as primary_key_column,
    COUNT(*) as num_records
FROM DBT_POC_DB.DBT_POC_TMP.TMP_LOCATION
GROUP BY LOCATION_ID
HAVING COUNT(*) > 1
```

You, 2 days ago • Check for working of test sql for staging …

```sql
UNION ALL

SELECT
    'TMP_SALES' as table_name,
    TXN_ID as primary_key_column,
    COUNT(*) as num_records
FROM DBT_POC_DB.DBT_POC_TMP.TMP_SALES
GROUP BY TXN_ID
HAVING COUNT(*) > 1
```

Figure 3: primary key violation test

2. **DBT Test Run:**
    a. First ***dbt test*** command was run without ***–store-failures*** flag. All tests ran successfully.
    b. However, a new schema DBT_DBT_TEST__AUDIT was created with table NOT_NULL_TMP_LOCATION_LOCATION_ID.
    c. This was created as the result of including ***store_failures: true*** config in ***not_null*** test for ***location_id*** column of ***tmp_location*** table.
    d. This newly created table mimicked the ***tmp_location*** table and was empty as no test failures had occurred.
    e. When tested with a null ***location_id***, the NOT_NULL_TMP_LOCATION_LOCATION_ID table stored the failed ***tmp_location*** record as failure.

```
$ dbt test
08:02:54  Running with dbt=1.6.5
08:02:55  Registered adapter: snowflake=1.6.4
08:02:55  Unable to do partial parsing because saved manifest not found. Starting full parse.
08:02:56  Found 14 models, 4 tests, 0 sources, 0 exposures, 0 metrics, 377 macros, 0 groups, 0 semantic models
08:02:56
08:03:08  Concurrency: 4 threads (target='dev')
08:03:08
08:03:08  1 of 4 START test not_null_tmp_location_location_id ..... [RUN]
08:03:08  2 of 4 START test rej_sales_foreign_key ..... [RUN]
08:03:08  3 of 4 START test test_primary_key ..... [RUN]
08:03:08  4 of 4 START test unique_tmp_location_location_id ..... [RUN]
08:03:11  4 of 4 PASS unique_tmp_location_location_id ..... [PASS in 3.10s]
08:03:12  2 of 4 PASS rej_sales_foreign_key ..... [PASS in 3.81s]
08:03:12  3 of 4 PASS test_primary_key ..... [PASS in 3.92s]
08:03:12  1 of 4 PASS not_null_tmp_location_location_id ..... [PASS in 4.22s]
08:03:12
08:03:12  Finished running 4 tests in 0 hours 0 minutes and 15.90 seconds (15.90s).
08:03:12
08:03:12  Completed successfully
08:03:12
08:03:12  Done. PASS=4 WARN=0 ERROR=0 SKIP=0 TOTAL=4
```

Figure 4: dbt test command run with tests pass

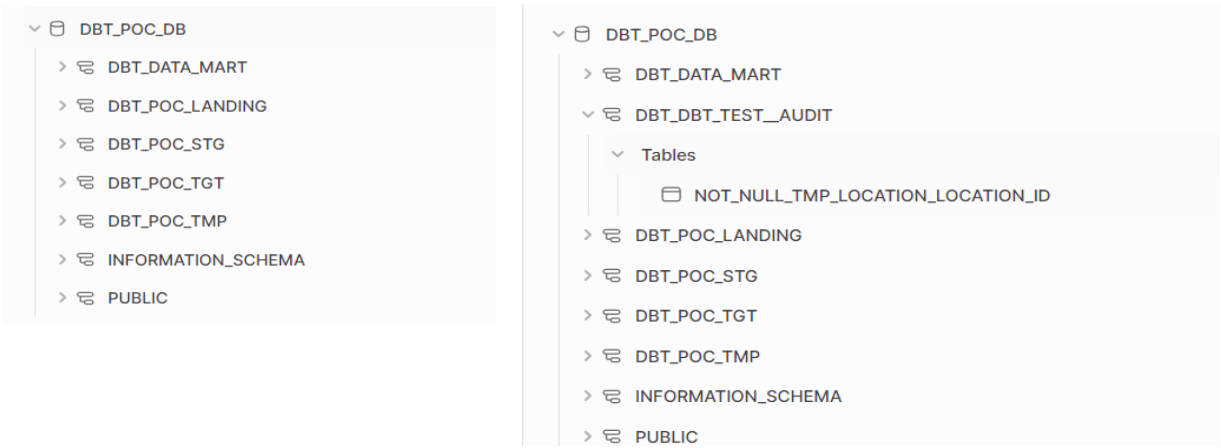


Figure 5: Snowflake database before/after

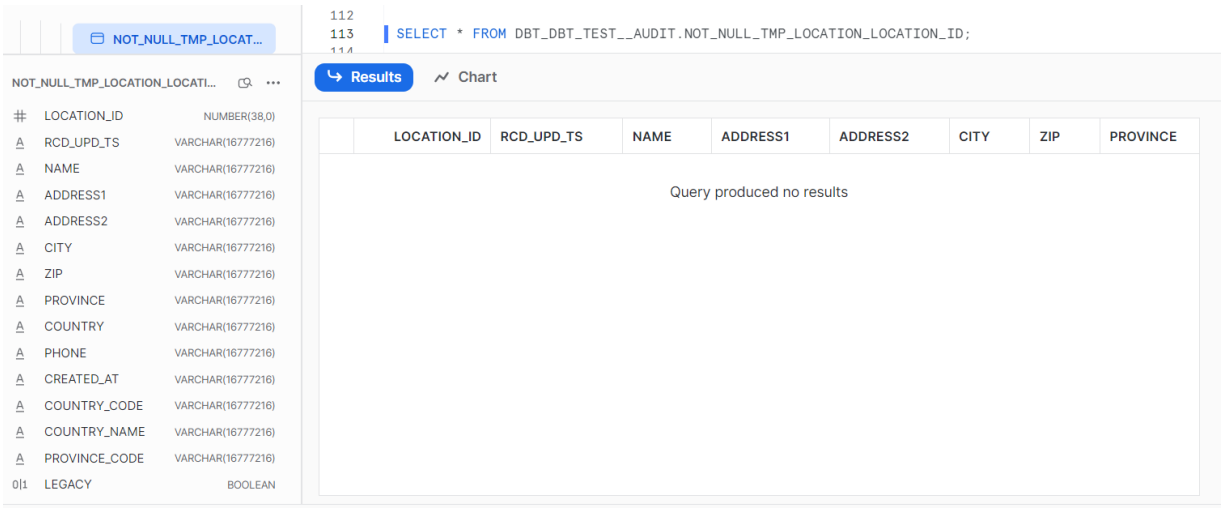


Figure 6: no failure stored for test pass

```
$ dbt test
08:32:41  Running with dbt=1.6.5
08:32:42  Registered adapter: snowflake=1.6.4
08:32:43  Found 14 models, 4 tests, 0 sources, 0 exposures, 0 metrics, 377 macros, 0 groups, 0 semantic models
08:32:43
08:32:51  Concurrency: 4 threads (target='dev')
08:32:51
08:32:51  2 of 4 START test rej_sales_foreign_key ..................................... [RUN]
08:32:51  1 of 4 START test not_null_tmp_location_location_id ......................... [RUN]
08:32:51  3 of 4 START test test_primary_key .......................................... [RUN]
08:32:51  4 of 4 START test unique_tmp_location_location_id ........................... [RUN]
08:32:54  2 of 4 PASS rej_sales_foreign_key ........................................... [PASS in 2.75s]
08:32:54  4 of 4 PASS unique_tmp_location_location_id ................................. [PASS in 2.91s]
08:32:54  3 of 4 PASS test_primary_key ................................................ [PASS in 3.39s]
08:32:55  1 of 4 FAIL 1 not_null_tmp_location_location_id ............................. [FAIL 1 in 4.12s]
08:32:55
08:32:55  Finished running 4 tests in 0 hours 0 minutes and 12.71 seconds (12.71s).
08:32:55
08:32:55  Completed with 1 error and 0 warnings:
08:32:55
08:32:55  Failure in test not_null_tmp_location_location_id (models\tmp\schema.yml)
08:32:55    Got 1 result, configured to fail if != 0
08:32:55
08:32:55    compiled Code at target\compiled\dbt_implementation_poc\models\tmp\schema.yml\not_null_tmp_location_location_id.sql
08:32:55
08:32:55    See test failures:
```

Figure 7: dbt test command run with test failure

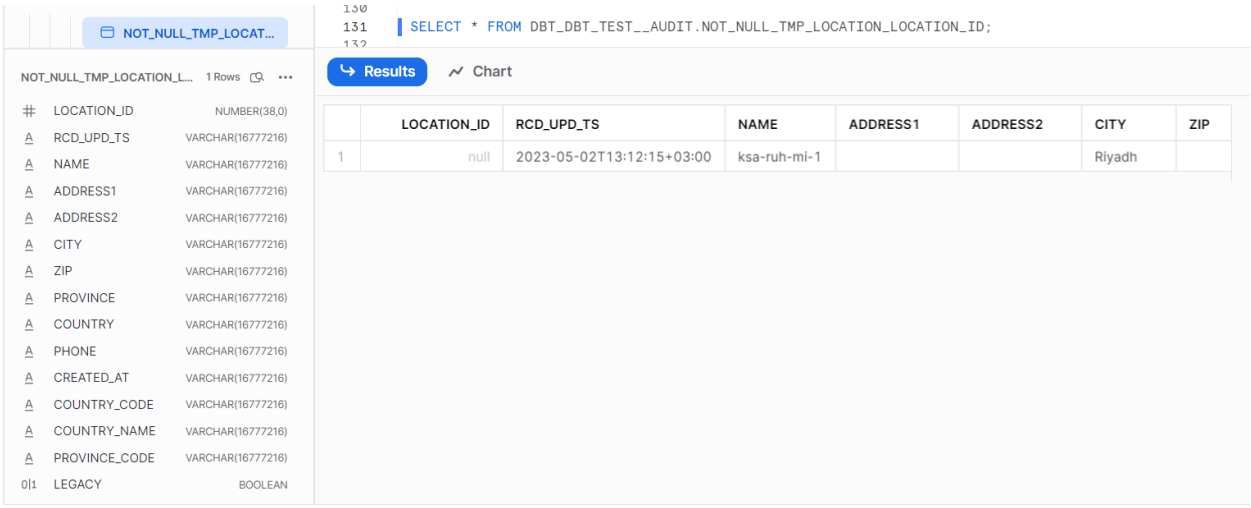


Figure 8: failure stored for test failed

3. DBT Test Run with –store-failures flag:

a. Next, ***dbt test*** command was run with ***–store-failures*** flag. For this test, ***store_failures*** config was set to false in ***not_null*** test for ***location_id*** column of ***tmp_location*** table.

b. ***location_id*** for another record was also set to null. The test failed. However, the failed record was not inserted in NOT_NULL_TMP_LOCATION_LOCATION_ID table. This verified the precedence of ***store_failures*** config over ***–store-failures*** flag.

c. This test run also failed ***test_primary_key*** custom test created for primary key violation as there were two records in ***tmp_location*** with ***location_id*** null.

Table TEST_PRIMARY_KEY had one record with same data model as defined in the ***test_primary_key.sql*** inside tests directory.

d. There were also REJ_SALES_FOREIGN_KEY, UNIQUE_TMP_LOCATION_LOCATION_ID tables created for ***rej_sales_foreign_key*** custom test and built in unique test for ***location_id*** of ***tmp_location*** table.

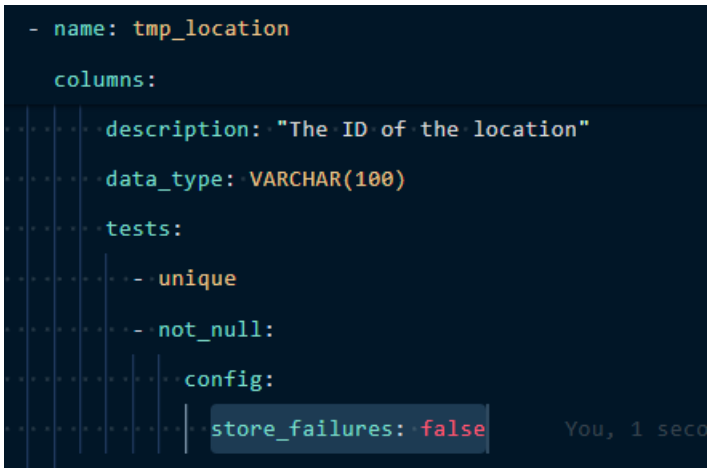


Figure 9: store failure configuration set to false

```
$ dbt test --store-failures
08:56:55  Running with dbt=1.6.5
08:56:56  Registered adapter: snowflake=1.6.4
08:56:56  Found 14 models, 4 tests, 0 sources, 0 exposures, 0 metrics, 377 macros, 0 groups, 0 semantic models
08:56:56
08:57:04  Concurrency: 4 threads (target='dev')
08:57:04
08:57:04  1 of 4 START test not_null_tmp_location_location_id ..... [RUN]
08:57:04  2 of 4 START test rej_sales_foreign_key ..... [RUN]
08:57:04  3 of 4 START test test_primary_key ..... [RUN]
08:57:04  4 of 4 START test unique_tmp_location_location_id ..... [RUN]
08:57:07  1 of 4 FAIL 2 not_null_tmp_location_location_id ..... [FAIL 2 in 2.81s]
08:57:08  4 of 4 PASS unique_tmp_location_location_id ..... [PASS in 3.87s]
08:57:08  2 of 4 PASS rej_sales_foreign_key ..... [PASS in 4.09s]
08:57:08  3 of 4 FAIL 1 test_primary_key ..... [FAIL 1 in 4.48s]
08:57:09
08:57:09  Finished running 4 tests in 0 hours 0 minutes and 12.15 seconds (12.15s).
08:57:09
08:57:09  Completed with 2 errors and 0 warnings:
08:57:09
08:57:09  Failure in test not_null_tmp_location_location_id (models\tmp\schema.yml)
08:57:09    Got 2 results, configured to fail if != 0
08:57:09
08:57:09    compiled Code at target\compiled\dbt_implementation_poc\models\tmp\schema.yml\not_null_tmp_location_location_id.sql
08:57:09
08:57:09    See test failures:
  ---------------------------------------------------------------------------
  select * from DBT_POC_DB.DBT_dbt_test__audit.not_null_tmp_location_location_id
  ---------------------------------------------------------------------------
08:57:09
08:57:09  Failure in test test_primary_key (tests\test_primary_key.sql)
08:57:09    Got 1 result, configured to fail if != 0
08:57:09
08:57:09    compiled Code at target\compiled\dbt_implementation_poc\tests\test_primary_key.sql
```

Figure 10: dbt test --store-failures command run with test failure

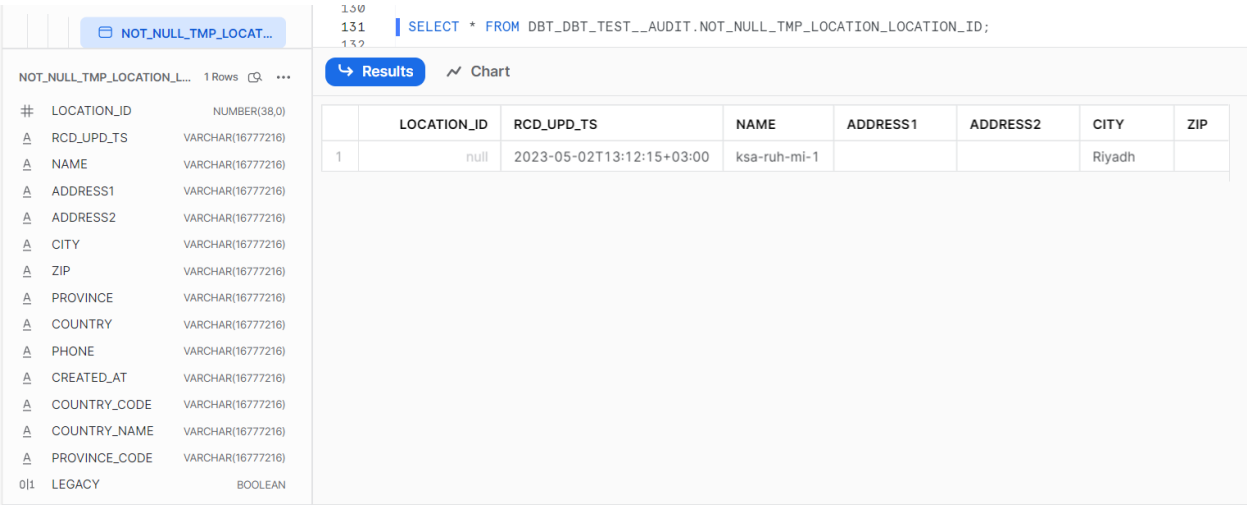


Figure 11: No failure storage after store_failures config set to false

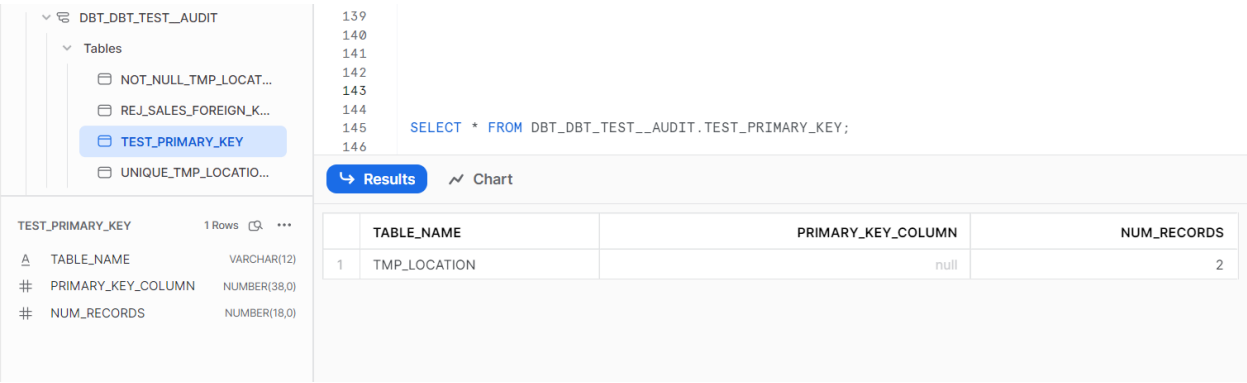
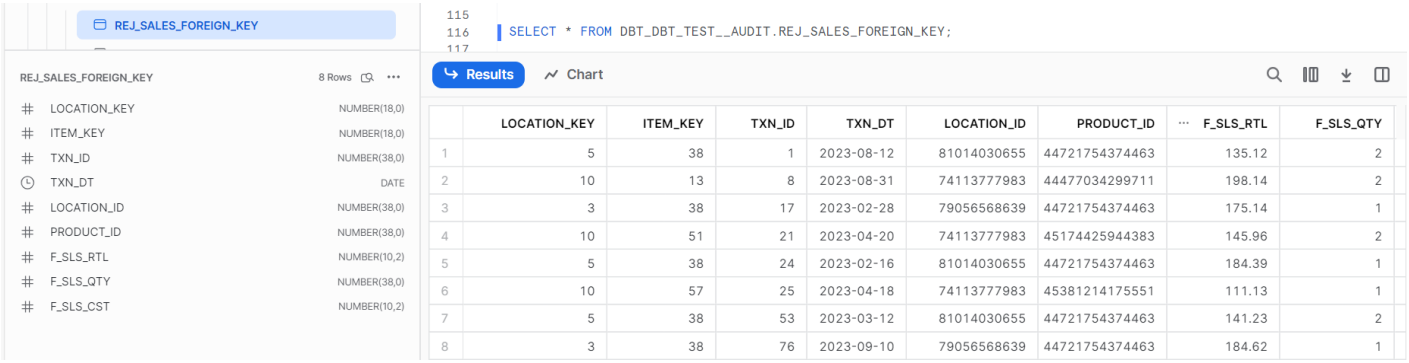


Figure 12: failure storage for primary key violation test

4. **Additional Test Runs:**
    a. For test regarding persistence of failure in database, DBT_DBT_TEST__AUDIT schema was dropped and ***dbt run*** command was run for fresh setup.
    b. One product with null ***ITEM_ID*** and one location with null ***LOCATION_ID*** were inserted into target table to violate foreign key constraint for ***rej_sales_foreign_key*** test.
    c. Test failed and failure got stored in REJ_SALES_FOREIGN_KEY table with same data model as defined in test SQL.
    d. For testing whether failures get appended, dropped and recreated, or truncated and loaded to rejection table, previous failure was fixed and new set of records were modified to have null IDs.
    e. Test failed and failures got stored in REJ_SALES_FOREIGN_KEY table. When compared with previous snapshot of same table, the old records had been replaced by new records.
    f. Again, ***dbt test –store-failures*** command was executed after fresh setup. All test cases passed as the data was valid. No failed records got stored in REJ_SALES_FOREIGN_KEY table. Old records were deleted but the table was sill existing. This proved that the rejection table is truncated and loaded each time ***dbt test –store-failures*** command is run.

```
$ dbt test --store-failures
12:17:50  Running with dbt=1.6.5
12:17:51  Registered adapter: snowflake=1.6.4
12:17:51  Found 17 models, 4 tests, 0 sources, 0 exposures, 0 metrics, 379 macros, 0 groups, 0 semantic models
12:17:51
12:18:02  Concurrency: 4 threads (target='dev')
12:18:02
12:18:02  1 of 4 START test not_null_tmp_location_location_id ..... [RUN]
12:18:02  2 of 4 START test rej_sales_foreign_key ..... [RUN]
12:18:02  3 of 4 START test test_primary_key ..... [RUN]
12:18:02  4 of 4 START test unique_tmp_location_location_id ..... [RUN]
12:18:05  1 of 4 PASS not_null_tmp_location_location_id ..... [PASS in 3.01s]
12:18:07  4 of 4 PASS unique_tmp_location_location_id ..... [PASS in 4.73s]
12:18:07  2 of 4 FAIL 8 rej_sales_foreign_key ..... [FAIL 8 in 4.78s]
12:18:07  3 of 4 PASS test_primary_key ..... [PASS in 4.79s]
12:18:07
12:18:07  Finished running 4 tests in 0 hours 0 minutes and 15.57 seconds (15.57s).
12:18:07
12:18:07  Completed with 1 error and 0 warnings:
12:18:07
12:18:07  Failure in test rej_sales_foreign_key (tests\rej_sales_foreign_key.sql)
12:18:07    Got 8 results, configured to fail if != 0
12:18:07
12:18:07    compiled Code at target\compiled\dbt_implementation_poc\tests\rej_sales_foreign_key.sql
12:18:07
12:18:07    See test failures:
  ---------------------------------------------------------------------
  select * from DBT_POC_DB.DBT_dbt_test__audit.rej_sales_foreign_key
  ---------------------------------------------------------------------
12:18:07
12:18:07  Done. PASS=3 WARN=0 ERROR=1 SKIP=0 TOTAL=4
```

Figure 13: dbt test --store-failures command run with test failure

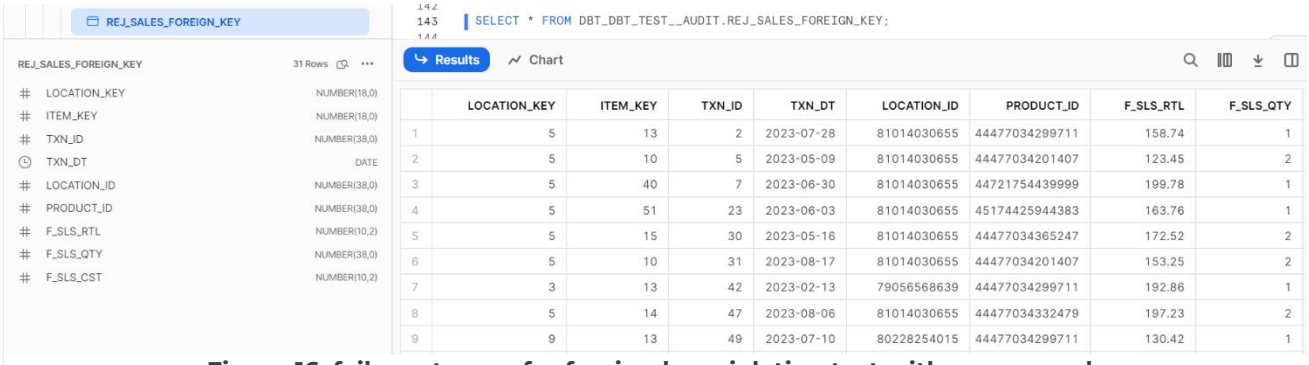


	LOCATION_KEY	ITEM_KEY	TXN_ID	TXN_DT	LOCATION_ID	PRODUCT_ID	F_SLS_RTL	F_SLS_QTY
1	5	38	1	2023-08-12	81014030655	44721754374463	135.12	2
2	10	13	8	2023-08-31	74113777983	44477034299711	198.14	2
3	3	38	17	2023-02-28	79056568639	44721754374463	175.14	1
4	10	51	21	2023-04-20	74113777983	45174425944383	145.96	2
5	5	38	24	2023-02-16	81014030655	44721754374463	184.39	1
6	10	57	25	2023-04-18	74113777983	45381214175551	111.13	1
7	5	38	53	2023-03-12	81014030655	44721754374463	141.23	2
8	3	38	76	2023-09-10	79056568639	44721754374463	184.62	1

Figure 14: failure storage for foreign key violation test

```
$ dbt test --store-failures
12:57:43  Running with dbt=1.6.5
12:57:44  Registered adapter: snowflake=1.6.4
12:57:44  Found 17 models, 4 tests, 0 sources, 0 exposures, 0 metrics, 379 macros, 0 groups, 0 semantic models
12:57:44
12:57:52  Concurrency: 4 threads (target='dev')
12:57:52
12:57:52  1 of 4 START test not_null_tmp_location_location_id ..... [RUN]
12:57:52  2 of 4 START test rej_sales_foreign_key ..... [RUN]
12:57:52  3 of 4 START test test_primary_key ..... [RUN]
12:57:52  4 of 4 START test unique_tmp_location_location_id ..... [RUN]
12:57:56  1 of 4 PASS not_null_tmp_location_location_id ..... [PASS in 3.74s]
12:57:56  4 of 4 PASS unique_tmp_location_location_id ..... [PASS in 4.38s]
12:57:56  3 of 4 PASS test_primary_key ..... [PASS in 4.53s]
12:57:56  2 of 4 FAIL 31 rej_sales_foreign_key ..... [FAIL 31 in 4.55s]
12:57:56
12:57:56  Finished running 4 tests in 0 hours 0 minutes and 12.43 seconds (12.43s).
12:57:56
12:57:56  Completed with 1 error and 0 warnings:
12:57:56
12:57:56  Failure in test rej_sales_foreign_key (tests\rej_sales_foreign_key.sql)
12:57:56    Got 31 results, configured to fail if != 0
12:57:56
12:57:56    compiled Code at target\compiled\dbt_implementation_poc\tests\rej_sales_foreign_key.sql
12:57:56
12:57:56    See test failures:
  ---------------------------------------------------------------------
  select * from DBT_POC_DB.DBT_dbt_test__audit.rej_sales_foreign_key
  ---------------------------------------------------------------------
```

Figure 15: dbt test --store-failures command run with test failure



	LOCATION_KEY	ITEM_KEY	TXN_ID	TXN_DT	LOCATION_ID	PRODUCT_ID	F_SLS_RTL	F_SLS_QTY
1	5	13	2	2023-07-28	81014030655	44477034299711	158.74	1
2	5	10	5	2023-05-09	81014030655	44477034201407	123.45	2
3	5	40	7	2023-06-30	81014030655	44721754439999	199.78	1
4	5	51	23	2023-06-03	81014030655	45174425944383	163.76	1
5	5	15	30	2023-05-16	81014030655	44477034365247	172.52	2
6	5	10	31	2023-08-17	81014030655	44477034201407	153.25	2
7	3	13	42	2023-02-13	79056568639	44477034299711	192.86	1
8	5	14	47	2023-08-06	81014030655	44477034332479	197.23	2
9	9	13	49	2023-07-10	80228254015	44477034299711	130.42	1

Figure 16: failure storage for foreign key violation test with new records

```
$ dbt test --store-failures
13:15:53  Running with dbt=1.6.5
13:15:54  Registered adapter: snowflake=1.6.4
13:15:54  Found 17 models, 4 tests, 0 sources, 0 exposures, 0 metrics, 379 macros, 0 groups, 0 semantic models
13:15:54
13:16:01  Concurrency: 4 threads (target='dev')
13:16:01
13:16:01  1 of 4 START test not_null_tmp_location_location_id ..... [RUN]
13:16:01  2 of 4 START test rej_sales_foreign_key ..... [RUN]
13:16:01  3 of 4 START test test_primary_key ..... [RUN]
13:16:01  4 of 4 START test unique_tmp_location_location_id ..... [RUN]
13:16:05  1 of 4 PASS not_null_tmp_location_location_id ..... [PASS in 3.21s]
13:16:05  4 of 4 PASS unique_tmp_location_location_id ..... [PASS in 3.68s]
13:16:05  3 of 4 PASS test_primary_key ..... [PASS in 3.82s]
13:16:06  2 of 4 PASS rej_sales_foreign_key ..... [PASS in 4.11s]
13:16:06
13:16:06  Finished running 4 tests in 0 hours 0 minutes and 11.86 seconds (11.86s).
13:16:06
13:16:06  Completed successfully
13:16:06
13:16:06  Done. PASS=4 WARN=0 ERROR=0 SKIP=0 TOTAL=4
```

Figure 17: dbt test --store-failures command run with tests pass

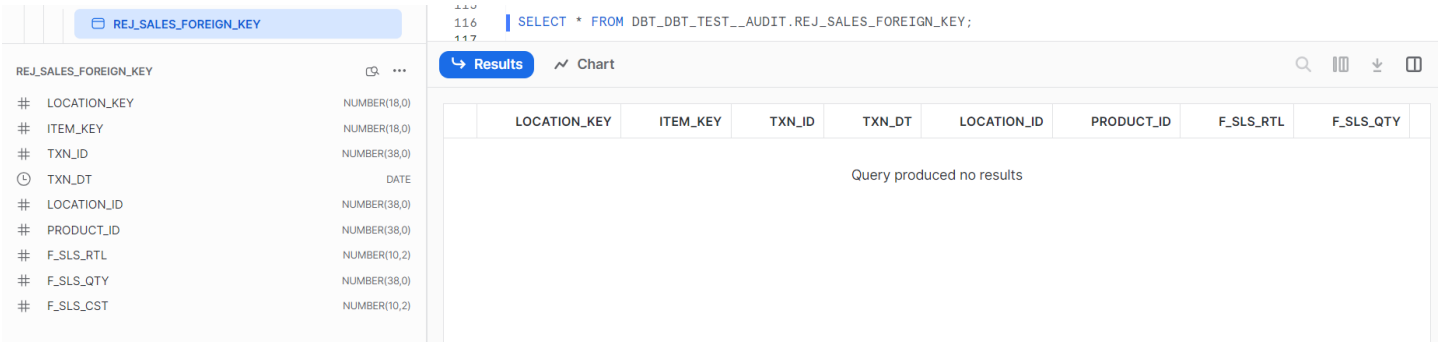


Figure 18: no failure storage for all tests pass

Rejection history maintenance

The most suitable way to maintain history of rejections stored by using DBT's ***–store-failures*** is to use ***pre_hook*** config before the next step after ***dbt test –store-failures*** command is run.

In our case, we have checked foreign key constraint violation for ***tmp_sales*** table. Hence, it made sense to maintain its rejection history in ***pre-hook*** configuration in ***tgt_sales*** model after test command is run.

macros > handle_rejection_sales.sql

You, 1 hour ago | 1 author (You)

```
{% macro handle_rejection_sales() %}
    INSERT INTO DBT_POC_DB.DBT_POC_REJ.REJ_SALES
    SELECT *
    FROM DBT_POC_DB.DBT_DBT_TEST__AUDIT.REJ_SALES_FOREIGN_KEY;
{% endmacro %}
```

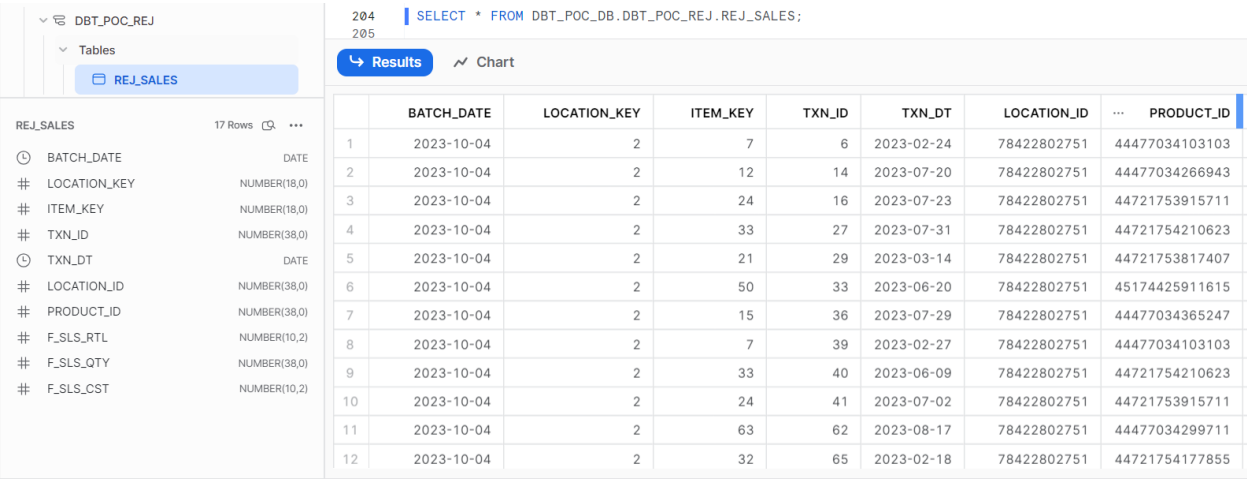
Figure 19: Macro for maintaining store failures history

models > tgt > tgt_sales.sql

You, 2 hours ago | 1 author (You)

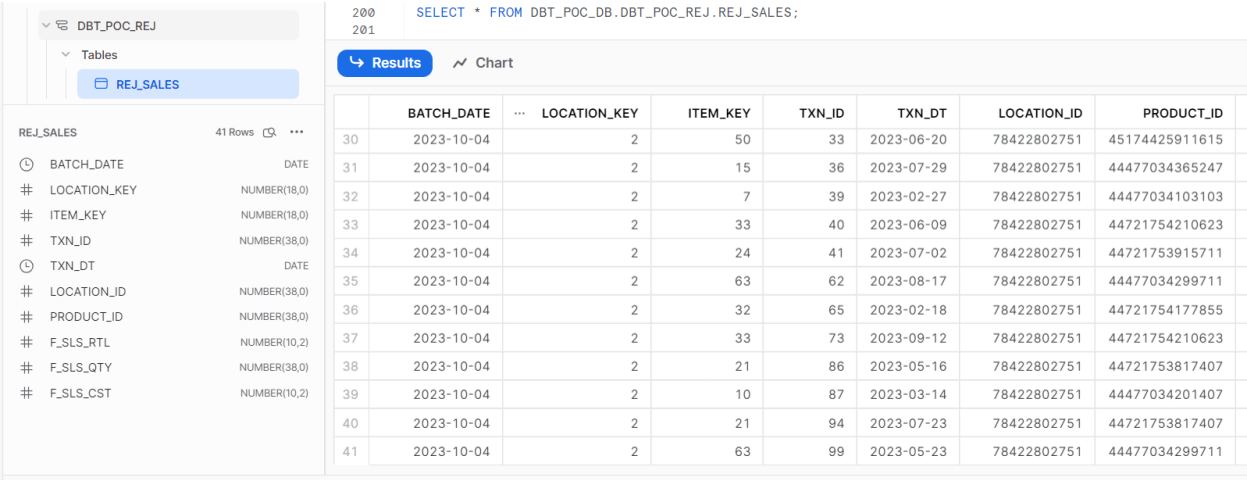
```
{{
    config(
        materialized='incremental',
        unique_key=['TXN_ID'],
        pre_hook=handle_rejection_sales()
    )
}}
```

Figure 20: Use of handle_rejection_sales macro as pre hook in tgt_sales model



	BATCH_DATE	LOCATION_KEY	ITEM_KEY	TXN_ID	TXN_DT	LOCATION_ID	PRODUCT_ID
1	2023-10-04	2	7	6	2023-02-24	78422802751	44477034103103
2	2023-10-04	2	12	14	2023-07-20	78422802751	44477034266943
3	2023-10-04	2	24	16	2023-07-23	78422802751	44721753915711
4	2023-10-04	2	33	27	2023-07-31	78422802751	44721754210623
5	2023-10-04	2	21	29	2023-03-14	78422802751	44721753817407
6	2023-10-04	2	50	33	2023-06-20	78422802751	45174425911615
7	2023-10-04	2	15	36	2023-07-29	78422802751	44477034365247
8	2023-10-04	2	7	39	2023-02-27	78422802751	44477034103103
9	2023-10-04	2	33	40	2023-06-09	78422802751	44721754210623
10	2023-10-04	2	24	41	2023-07-02	78422802751	44721753915711
11	2023-10-04	2	63	62	2023-08-17	78422802751	44477034299711
12	2023-10-04	2	32	65	2023-02-18	78422802751	44721754177855

Figure 21: REJ_SALES table for maintaining sales rejection history



	BATCH_DATE	LOCATION_KEY	ITEM_KEY	TXN_ID	TXN_DT	LOCATION_ID	PRODUCT_ID
30	2023-10-04	2	50	33	2023-06-20	78422802751	45174425911615
31	2023-10-04	2	15	36	2023-07-29	78422802751	44477034365247
32	2023-10-04	2	7	39	2023-02-27	78422802751	44477034103103
33	2023-10-04	2	33	40	2023-06-09	78422802751	44721754210623
34	2023-10-04	2	24	41	2023-07-02	78422802751	44721753915711
35	2023-10-04	2	63	62	2023-08-17	78422802751	44477034299711
36	2023-10-04	2	32	65	2023-02-18	78422802751	44721754177855
37	2023-10-04	2	33	73	2023-09-12	78422802751	44721754210623
38	2023-10-04	2	21	86	2023-05-16	78422802751	44721753817407
39	2023-10-04	2	10	87	2023-03-14	78422802751	44477034201407
40	2023-10-04	2	21	94	2023-07-23	78422802751	44721753817407
41	2023-10-04	2	63	99	2023-05-23	78422802751	44477034299711

Figure 22: Failed sales records appended to REJ_SALES table maintaining history

Alternate ways of testing foreign key violation

Alternate ways of testing foreign key violation include:

1. Adding ***relationships*** test in ***yml*** configuration file.
2. Using packages that enforce foreign key constraint
    a. ***cardinality_equality*** of ***dbt_utils*** package
    b. ***foreign_key*** of ***dbt_constraints*** package

```yaml
- name: tmp_sales
  description: "Temp Table for Sales"
  columns:
    - name: location_id
      description: "Foreign key to tgt_location table"
      data_type: VARCHAR(100)
      tests:
        - relationships:
            to: ref('tgt_location')
            field: location_id
            config:
              store_failures: true
```

Figure 23: test for location_id foreign key violation using relationships

Figure 24: test for product_id foreign key violation using dbt_utils package

Figure 25: test for product_id foreign key violation using dbt_constraints package

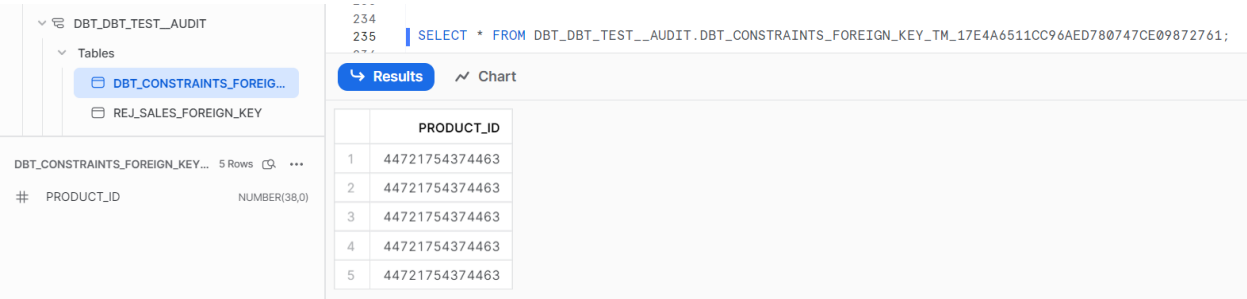


Figure 26: Storage of failure using dbt_constraints package

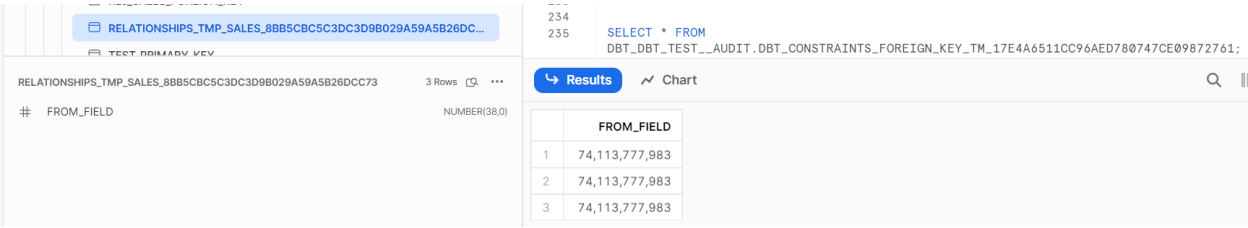


Figure 27: storage of failures using relationships

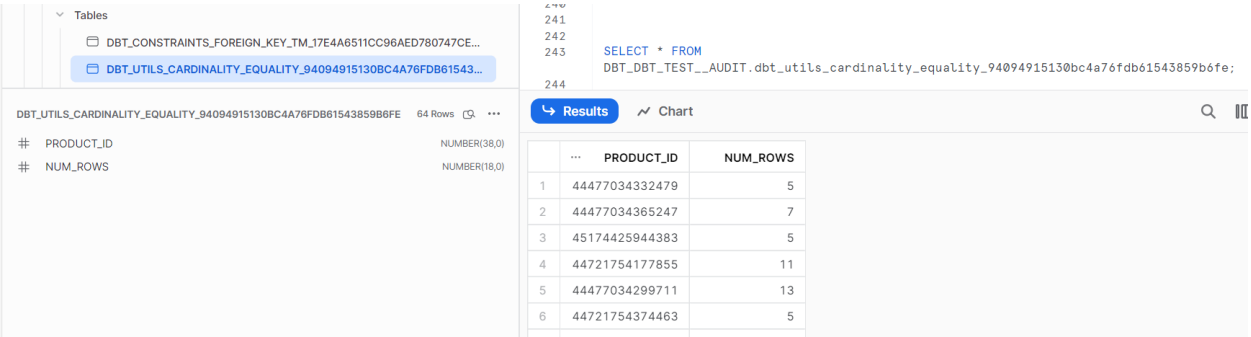


Figure 28: storage of failures using dbt_utils package

Testing foreign key violation using ***yml*** file configuration, whether through package or via relationships, store failures containing only the ***foreign_key*** column. This cannot be used as rejection table by default and need further lookups to create one. Hence using custom tests for storing failures is the suitable way to handle rejections down the line.