FPA INGESTION SCRUM

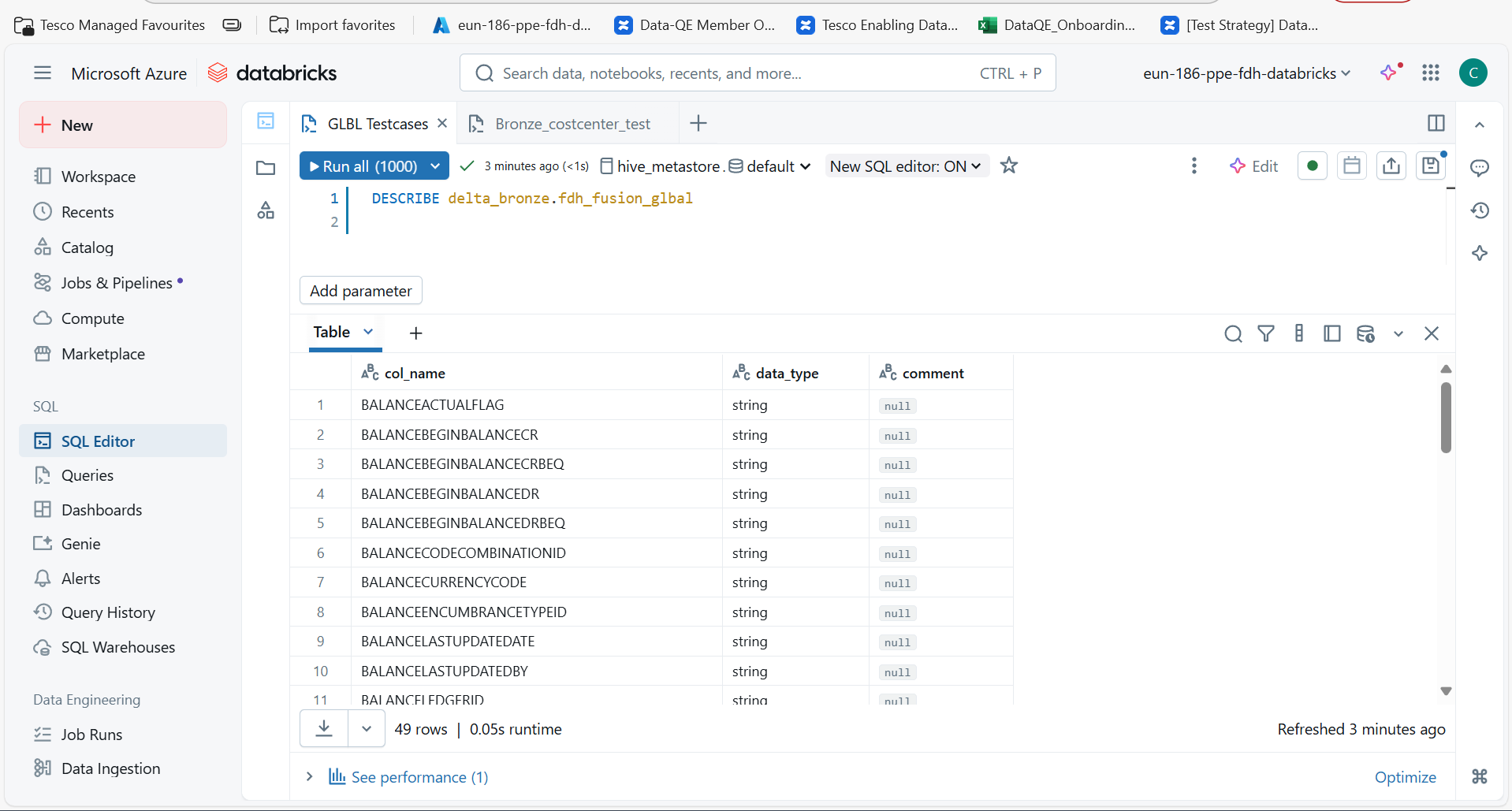
TEST CASES

Jira link : [https://tesco.atlassian.net/projects/FPA?selectedItem=com.atlassian.plugins.atlassian-connect-plugin:com.kanoah.test-manager\_\_main-project-page#!/v2/testCases?projectId=11014](https://tesco.atlassian.net/projects/FPA?selectedItem=com.atlassian.plugins.atlassian-connect-plugin:com.kanoah.test-manager__main-project-page%23!/v2/testCases?projectId=11014)

Fusion\_GL\_Balances

Test Case 1 - Validate table delta\_gold.fusion\_gl\_balances and its metadata is matching with Miro/Interface Contract

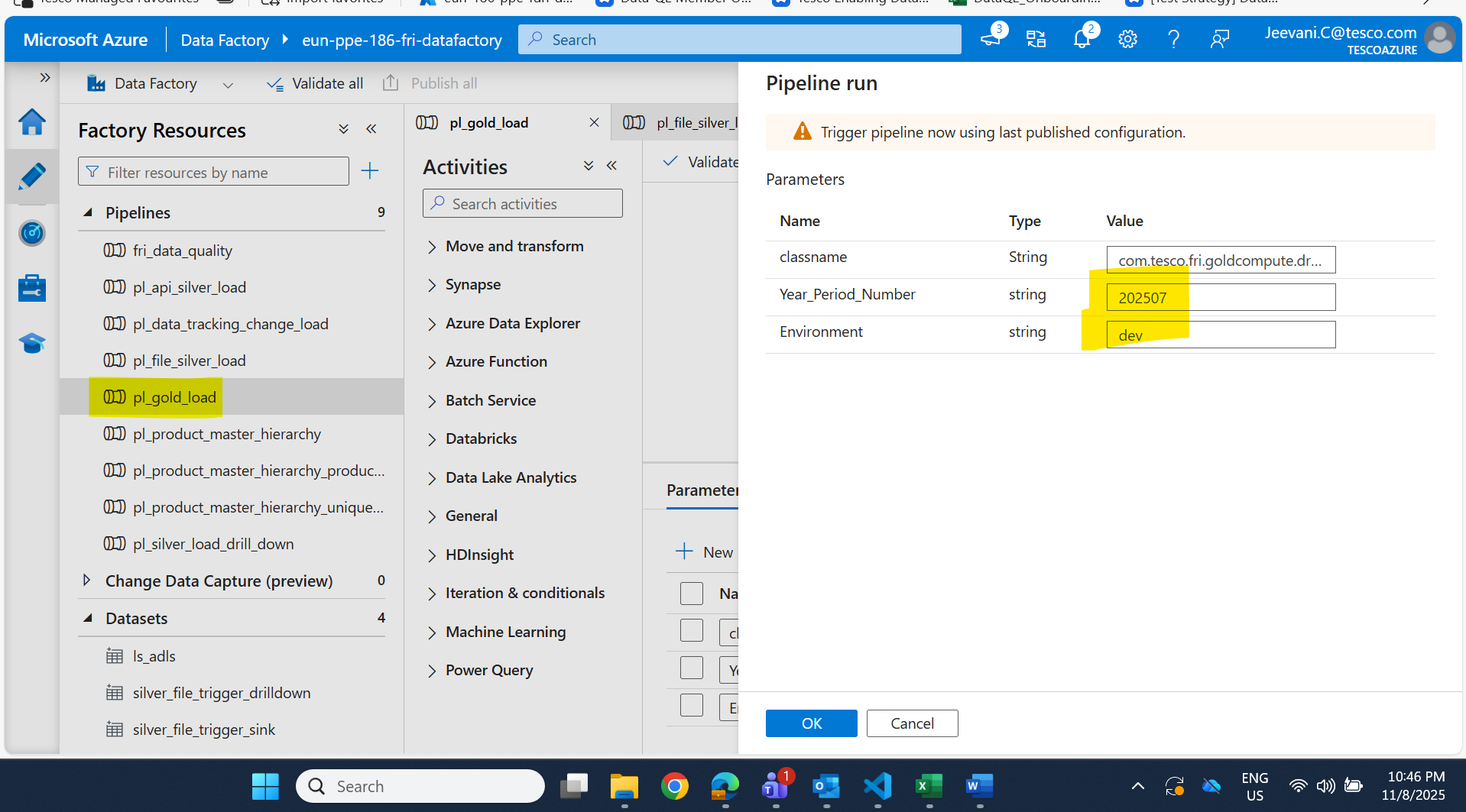
QUERY : DESCRIBE delta\_gold.fusion\_gl\_balances



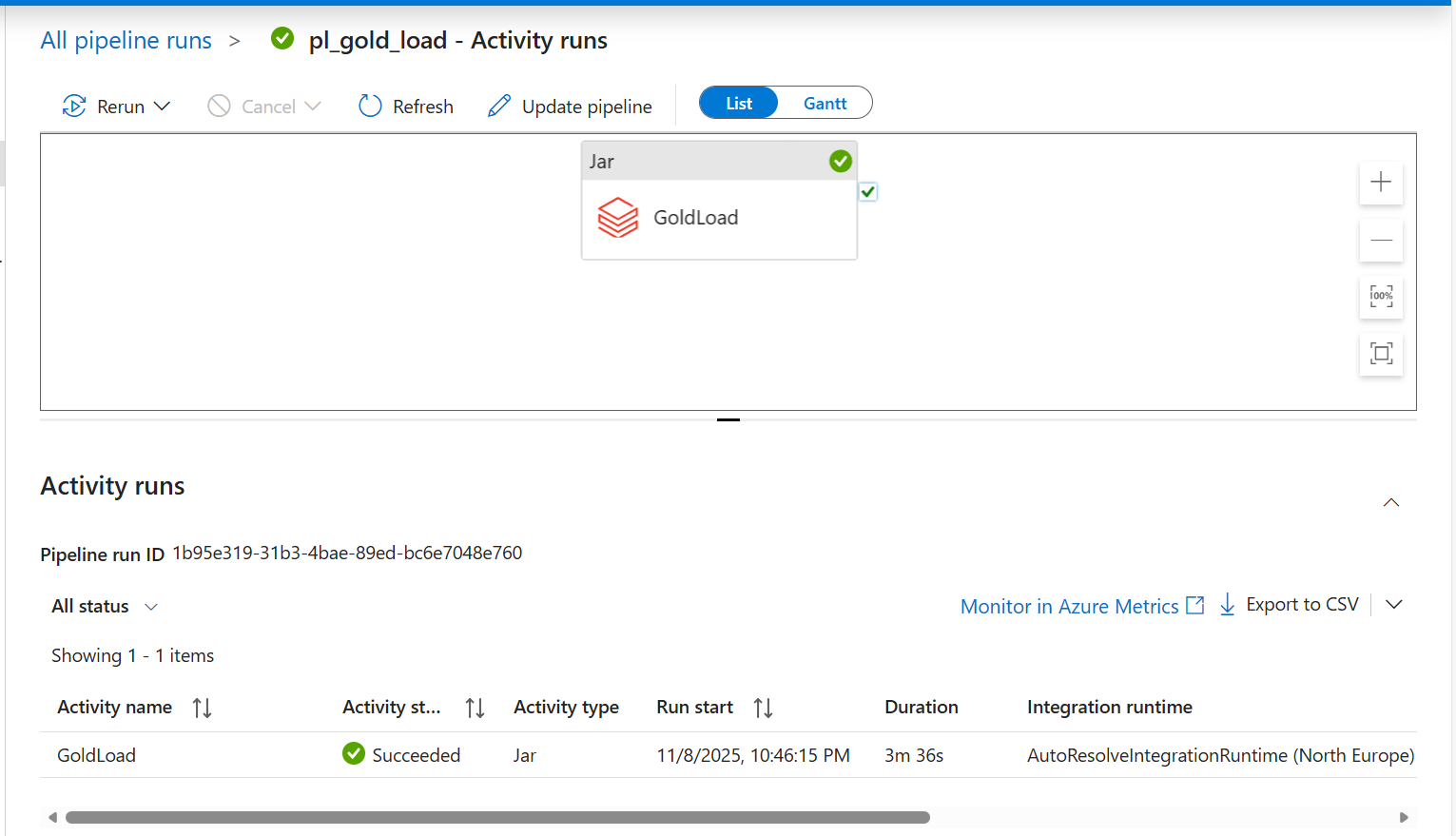
Here I have checked the Metadata ( Column names and data type ) with Miro and Contract too.

TEST CASE 3 : 3-Verify on running the job pl\_gold\_load manually, if the job is successful Then the data should be moved from silver to gold.

Triggered pl\_gold\_load pipeline for YPN 202507 and it is successfull



Output:



**Test Case 9:** 9-Verify the row count matches between silver and gold layer based on fusion\_code, fusion\_cost\_centre unique combinations

Ensure that the row count for unique combinations of fusion\_code (account) and fusion\_cost\_centre in silver matches the row count for gl\_account and gl\_cost\_centre\_code in gold.

Query :

**SELECT COUNT(\*) AS silver\_count**

**FROM (**

**SELECT fdmee\_fusion\_string AS fusion\_code,**

**source\_cost\_centre AS fusion\_cost\_centre**

**FROM delta\_silver.fact\_fdmee\_transformed\_data**

**GROUP BY fdmee\_fusion\_string, source\_cost\_centre**

**) silver**

**UNION ALL**

**SELECT COUNT(\*) AS gold\_count**

**FROM (**

**SELECT gl\_account,**

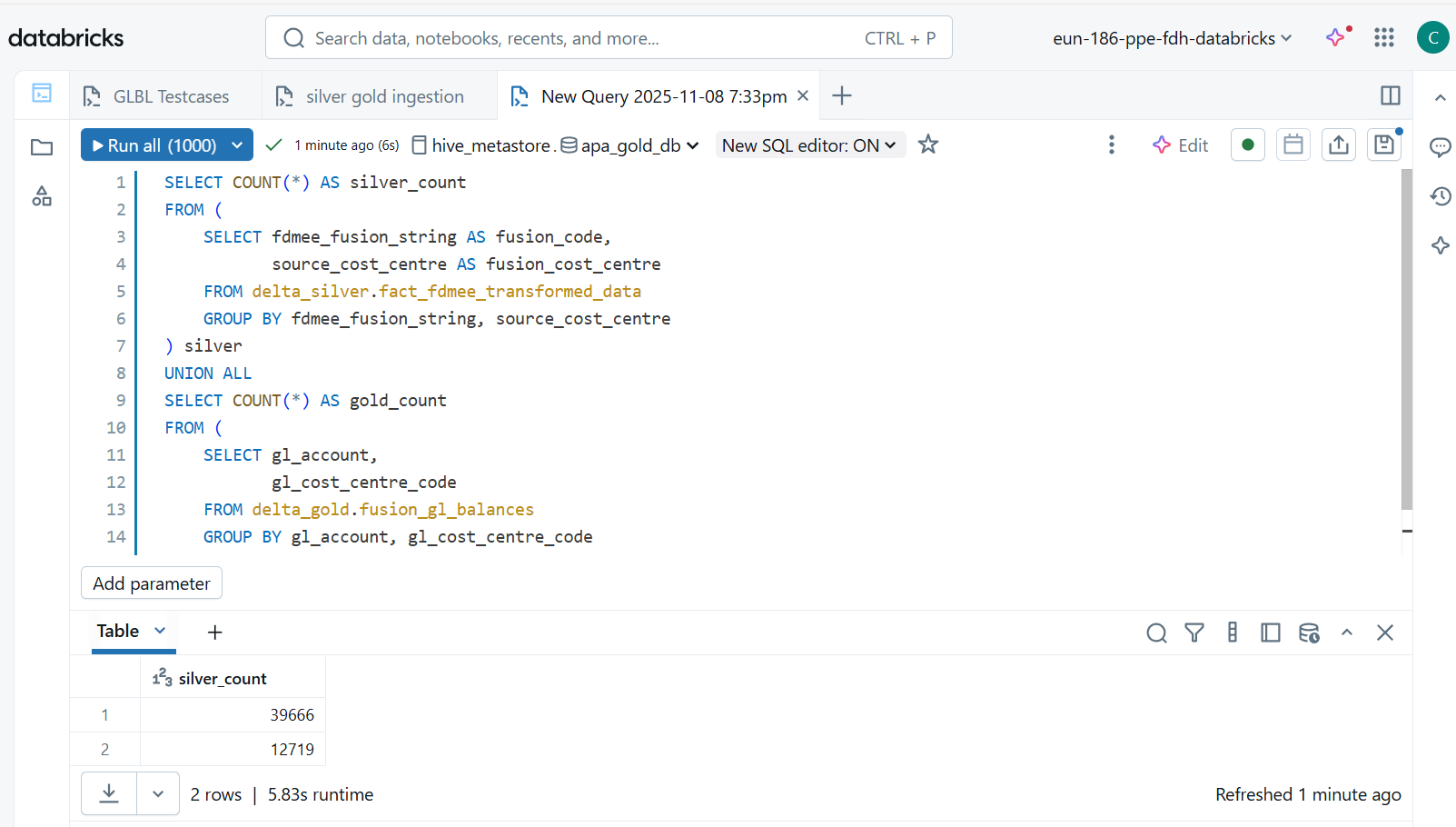
**gl\_cost\_centre\_code**

**FROM delta\_gold.fusion\_gl\_balances**

**GROUP BY gl\_account, gl\_cost\_centre\_code**

**) gold;**

Output:

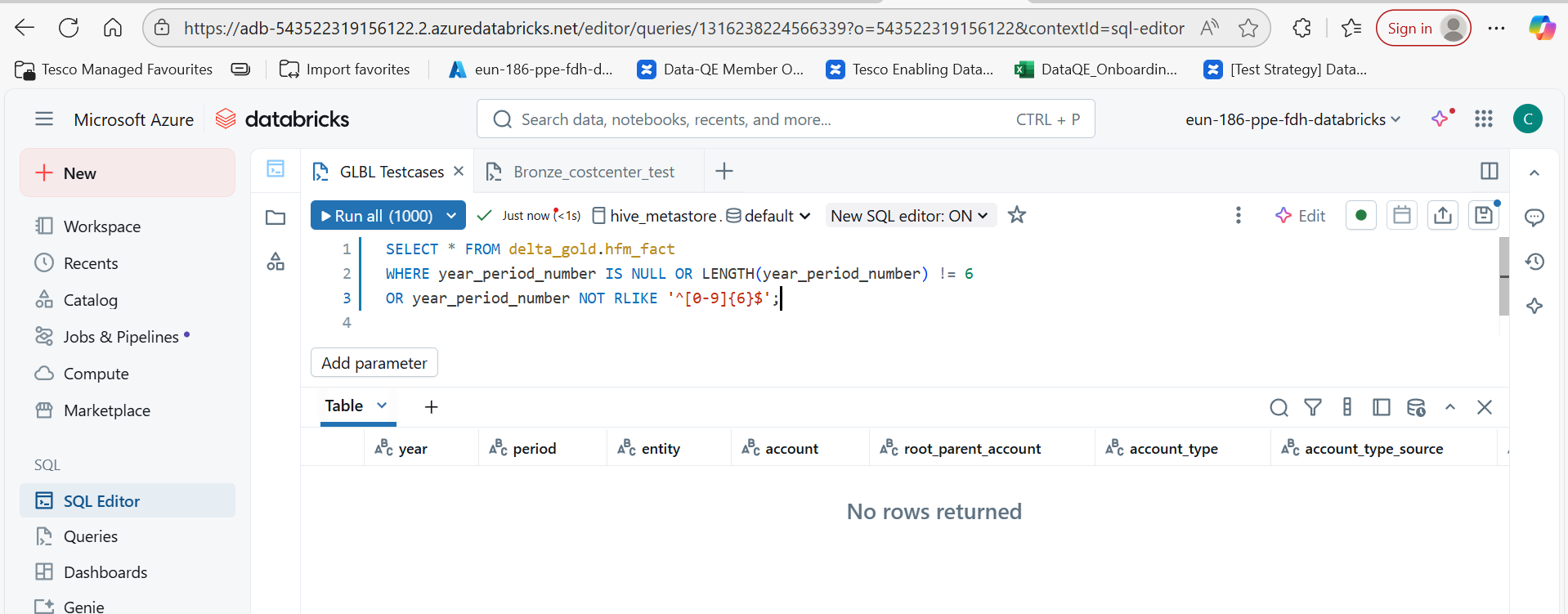


TEST CASE 10 : 10-Ensure YEAR\_PERIOD\_NUMBER is a 6-digit string (e.g., 202412).

QUERY : SELECT \* FROM delta\_gold.fusion\_gl\_balances

WHERE year\_period\_number IS NULL OR LENGTH(year\_period\_number) != 6

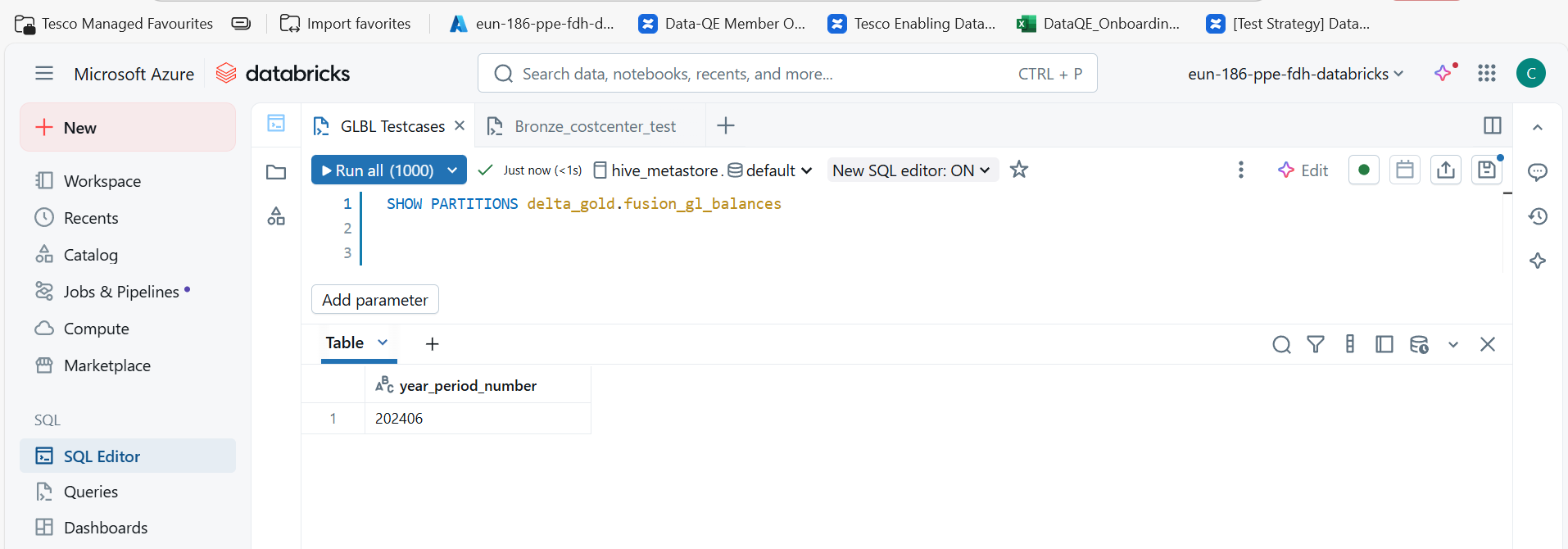
OR year\_period\_number NOT RLIKE '^[0-9]{6}$';



I went to miro and searched for the column year\_period\_number, I got that hfm\_fact gold table has that column, so I have used that table to check the data structure of year\_period\_number

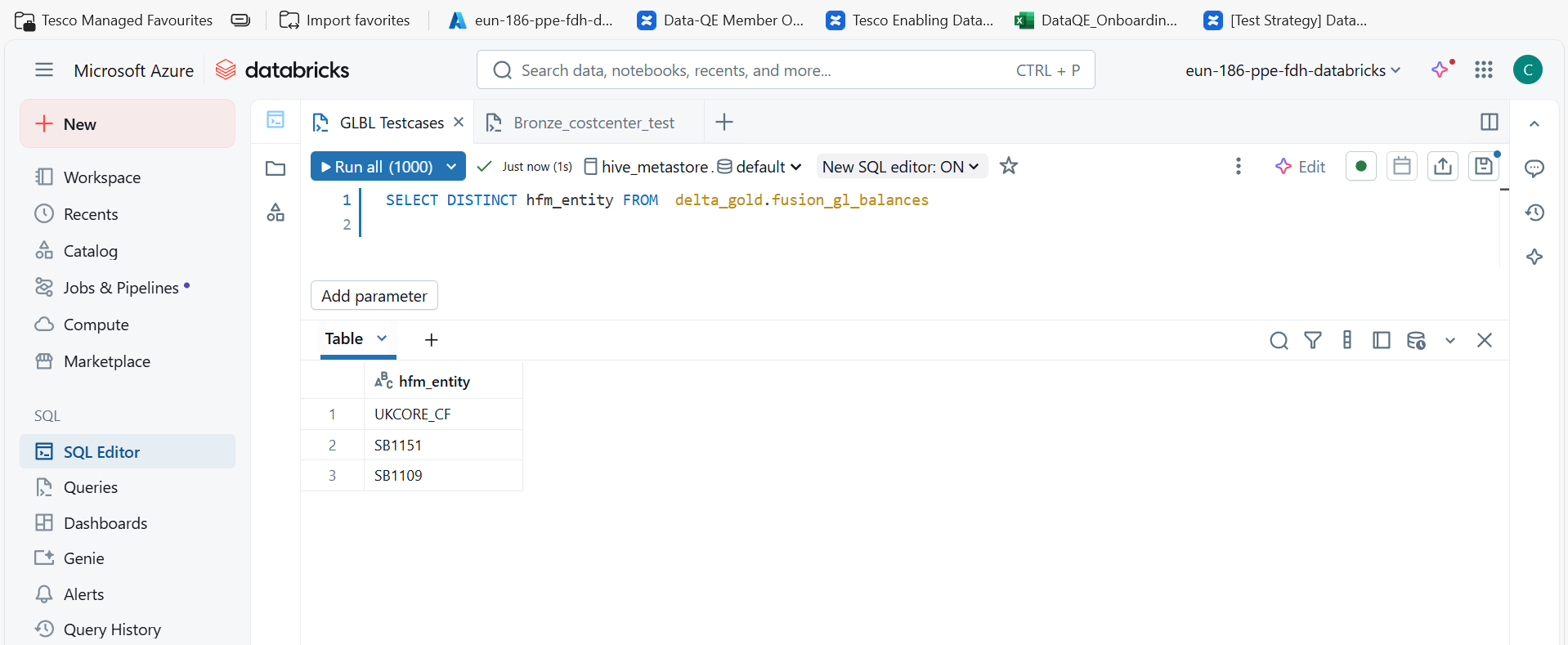
TEST CASE 13 - 13-Verify if the partition column is year\_period\_number

QUERY : SHOW PARTITIONS delta\_gold.fusion\_gl\_balances

  
  
In this table delta\_gold.fusion\_gl\_balances I have checked the partition column used , it shown year\_period\_number

TEST CASE 15 - Verify the column hfm\_entity contains only 3 values 'UKCORE\_CF','SB1109','SB1151' and not null

QUERY : SELECT DISTINCT hfm\_entity FROM  delta\_gold.fusion\_gl\_balances

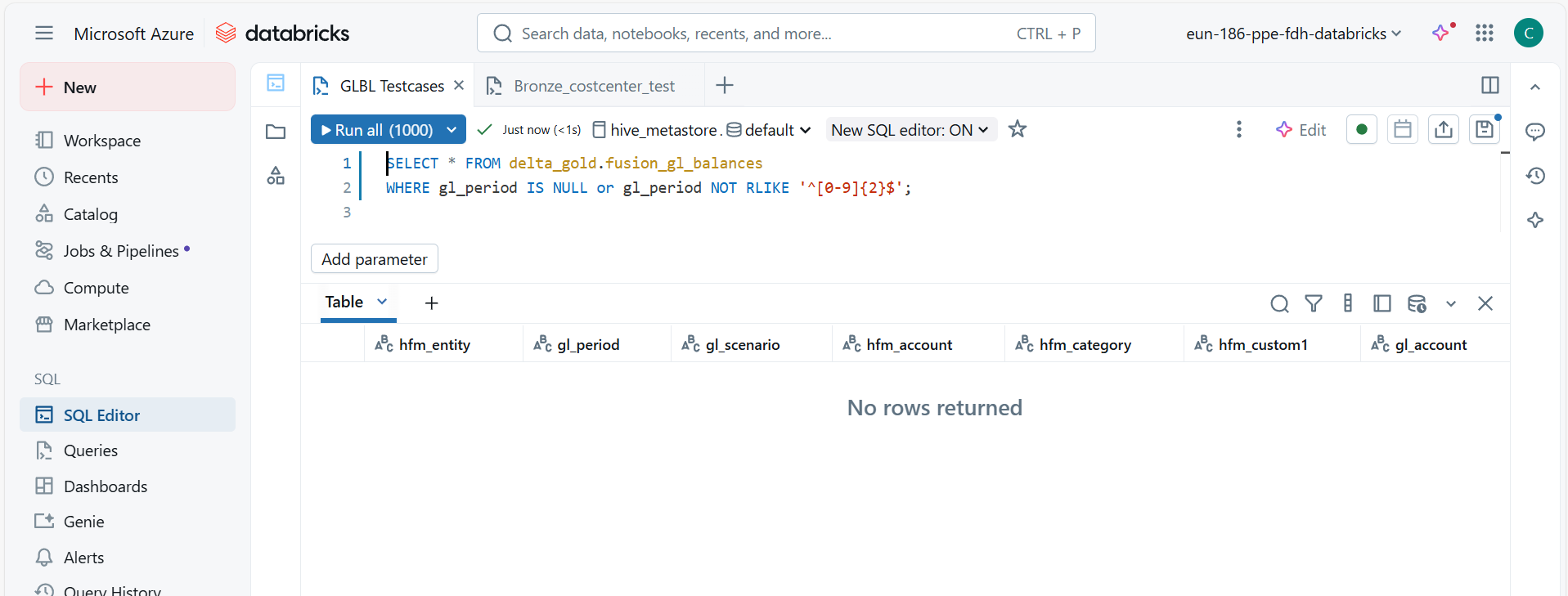


The above query returns only 3 unique set of entity and null is not there so, test case is passed.

TEST CASE 17 : 17-Verify the column gl\_period contains only numeric and only 2 digit and not null

QUERY : SELECT \* FROM delta\_gold.fusion\_gl\_balances

WHERE gl\_period IS NULL or gl\_period NOT RLIKE '^[0-9]{2}$'

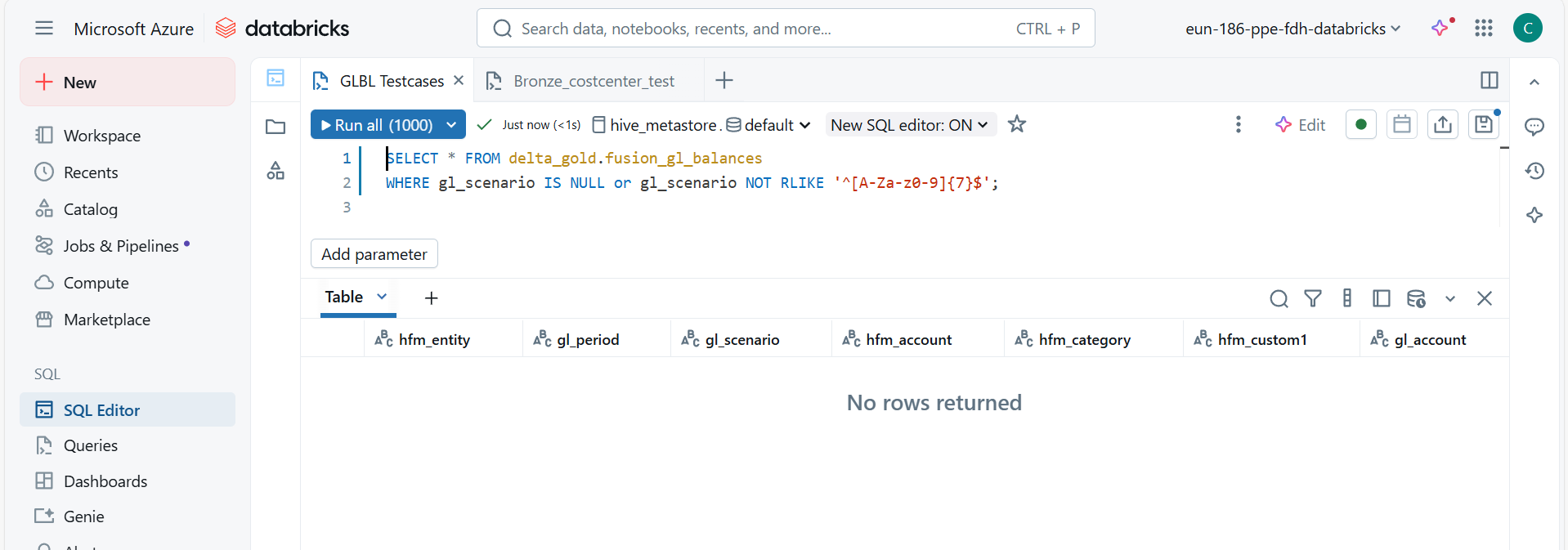


Null rows returns show that all data followed the requested data structure in test case

TEST CASE 19 : 19-Verify the column gl\_scenario contains alphanumeric and length is only 7 and not null

QUERY : SELECT \* FROM delta\_gold.fusion\_gl\_balances

WHERE gl\_scenario IS NULL or gl\_scenario NOT RLIKE '^[A-Za-z0-9]{7}$';

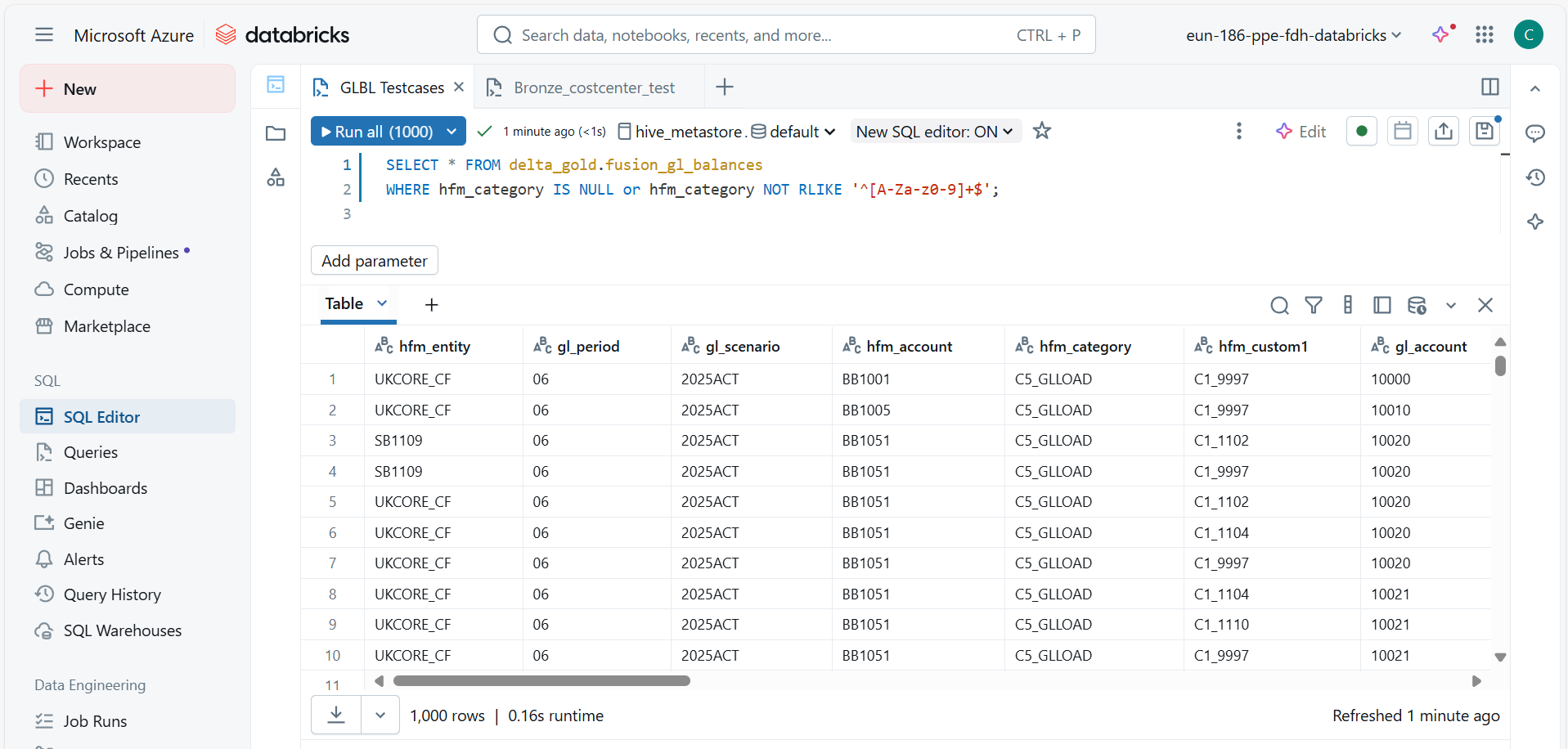


Null rows returns show that all data followed the requested data structure in test case

TEST CASE 22 : 22-Verify the hfm\_category is alphanumeric and not null

QUERY : SELECT \* FROM delta\_gold.fusion\_gl\_balances

WHERE hfm\_category IS NULL or hfm\_category NOT RLIKE '^[A-Za-z0-9]+$';

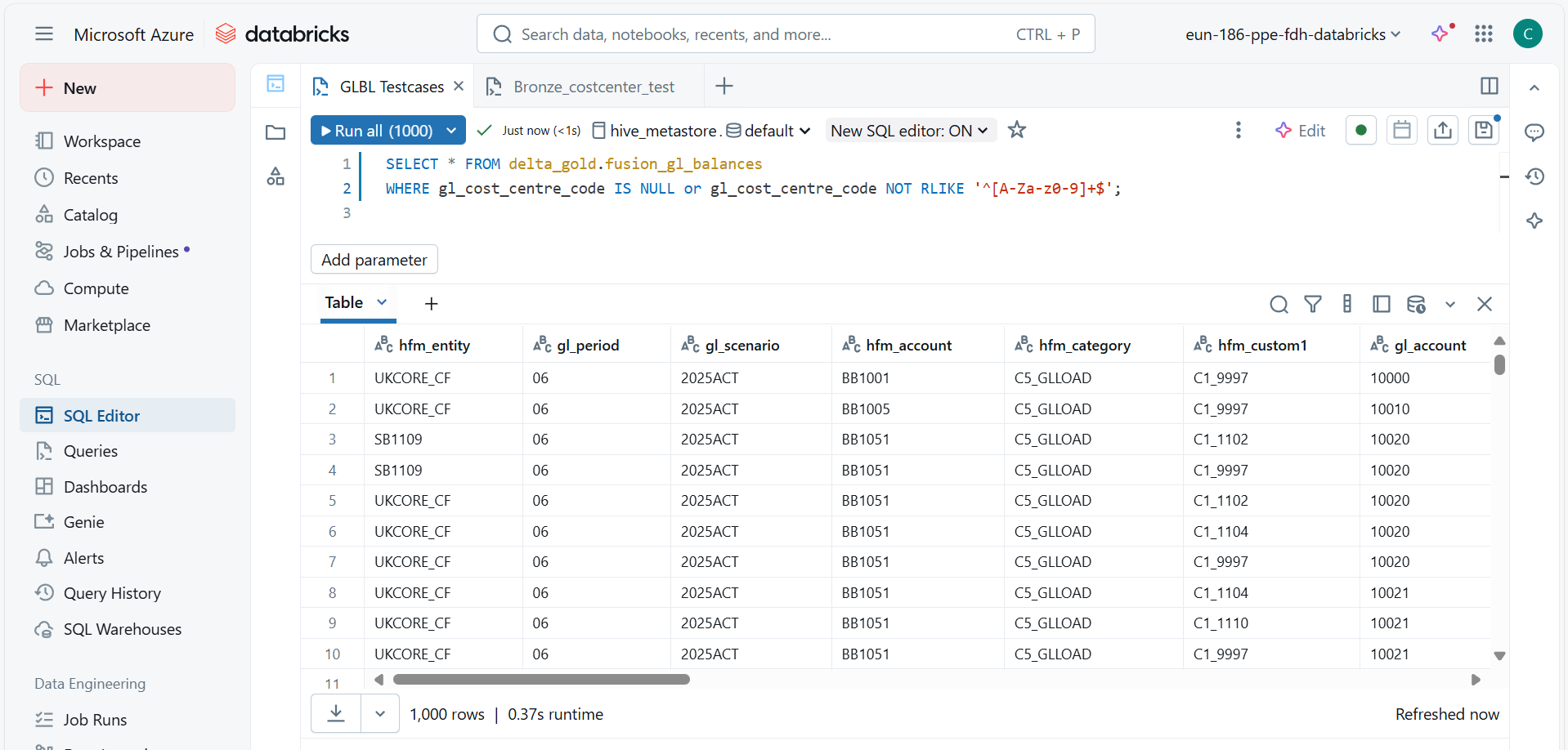


The hfm\_catgeory contains ‘\_’ (underscore), that why rows are returned, but according to test case it should not be there, underscore, space, @, hyphen (-) etc. should not be consider as alpha numeric

TEST CASE 24 and 26 : 26-Verify the column gl\_cost\_centre\_code is alphanumeric and contains null

QUERY : SELECT \* FROM delta\_gold.fusion\_gl\_balances

WHERE gl\_cost\_centre\_code IS NULL or gl\_cost\_centre\_code NOT RLIKE '^[A-Za-z0-9]+$';

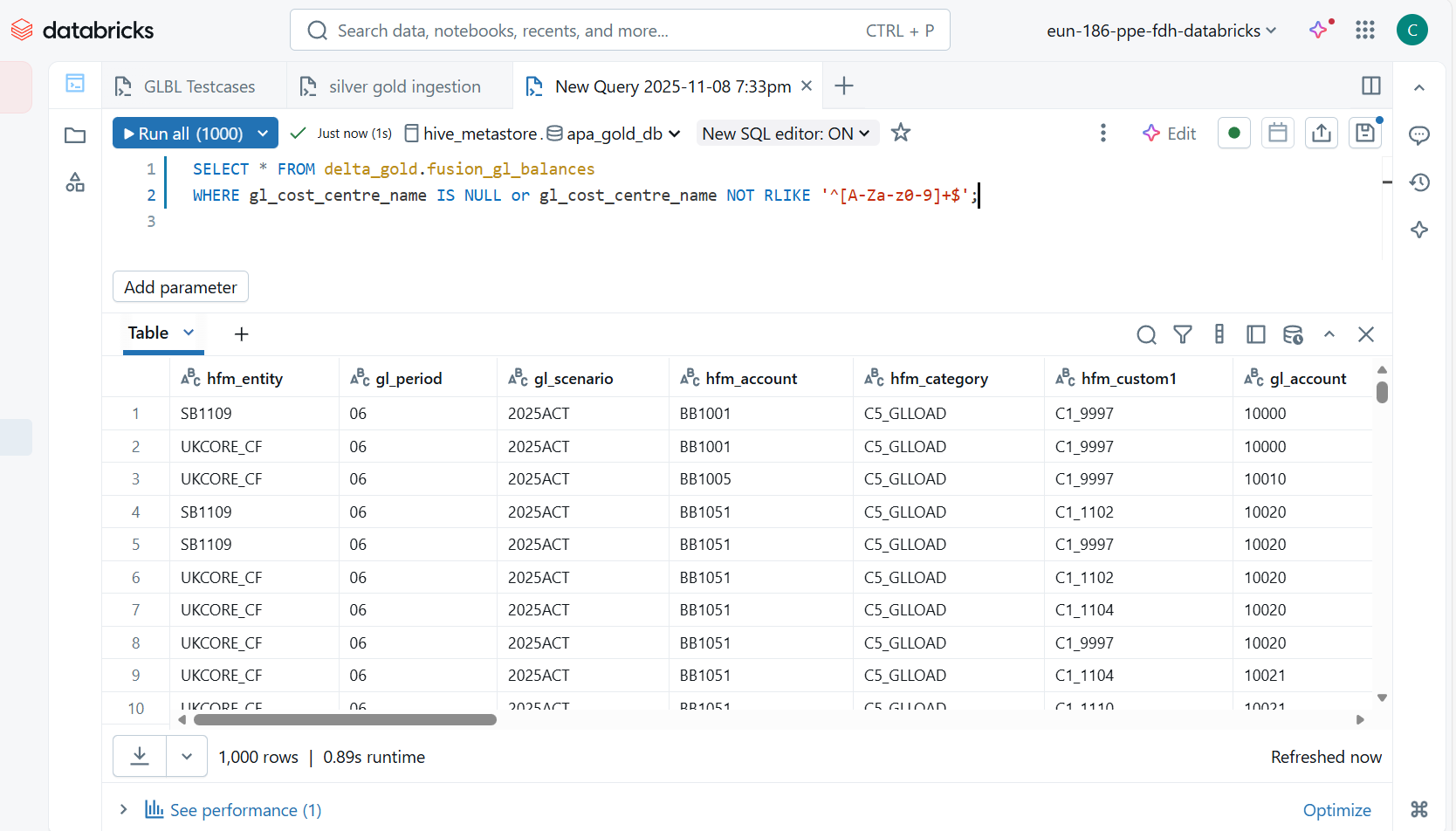


The gl\_cost\_centre\_code contains ‘\_’ (underscore), that why rows are returned, but according to test case it should not be there, underscore, space, @, hyphen (-) etc. should not be consider as alpha numeric.

TEST CASE 28 : 28-Verify the column gl\_cost\_centre\_name is alphanumeric and contains null

QUERY : SELECT \* FROM delta\_gold.fusion\_gl\_balances

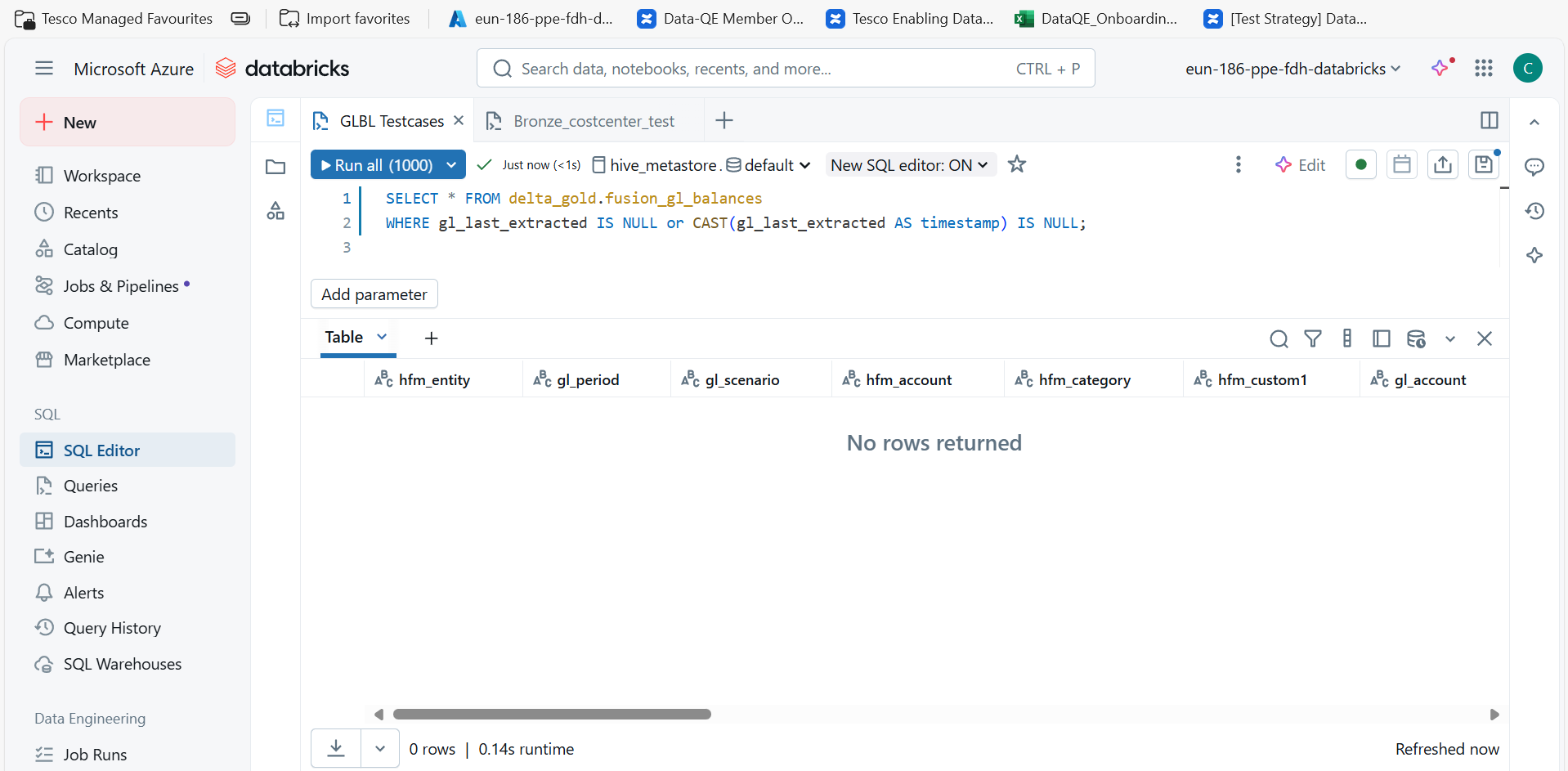
WHERE gl\_cost\_centre\_name IS NULL or gl\_cost\_centre\_name NOT RLIKE '^[A-Za-z0-9]+$';



TEST CASE 30 : 30-Verify the column gl\_last\_extracted is timestamp and not null

QUERY : SELECT \* FROM delta\_gold.fusion\_gl\_balances

WHERE gl\_last\_extracted IS NULL or CAST(gl\_last\_extracted AS timestamp) IS NULL;



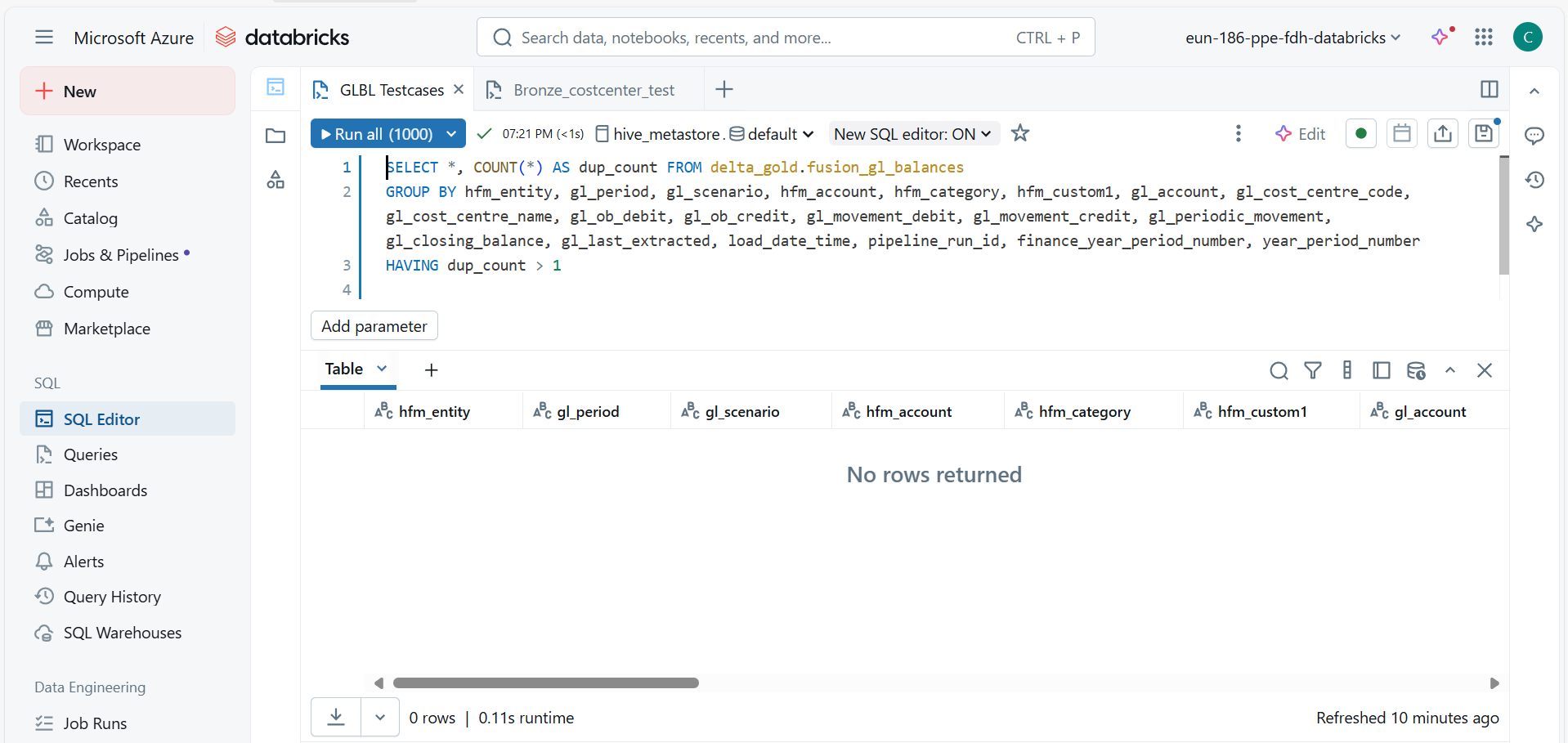
The above query CAST(gl\_last\_extracted AS timestamp) IS NULL checks if the value cant be converted to a valid timestamp format ( for exa, ‘2024-13-99’ or ‘invalid’). The query returns zero rows, all timestamp values are non null and valid.

TEST CASE 41 : 41-Ensure duplicate records are not loaded into Gold Layer delta\_gold.fusion\_gl\_balances

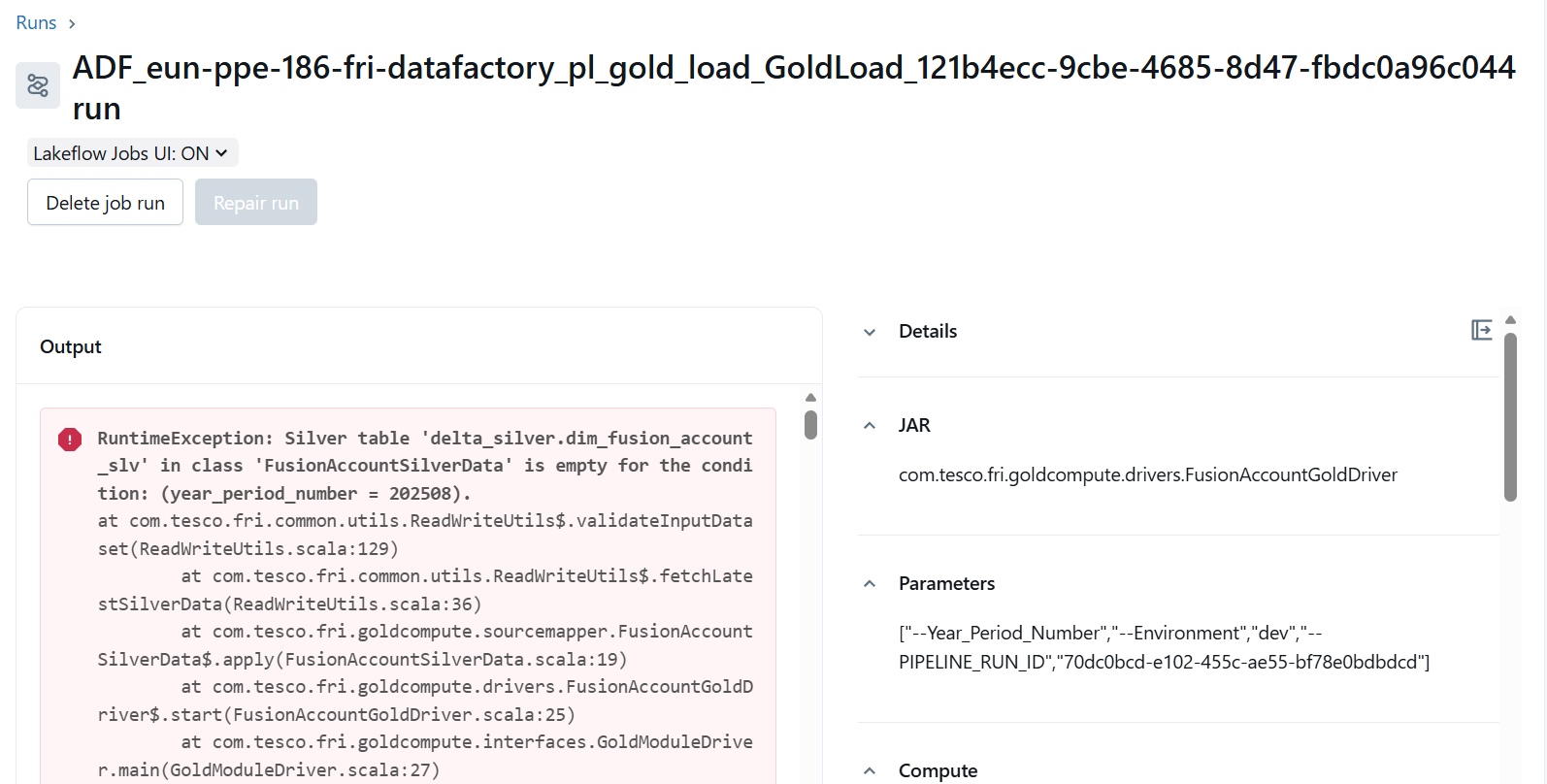
QUERY : SELECT \*, COUNT(\*) AS dup\_count FROM delta\_gold.fusion\_gl\_balances

GROUP BY hfm\_entity, gl\_period, gl\_scenario, hfm\_account, hfm\_category, hfm\_custom1, gl\_account, gl\_cost\_centre\_code, gl\_cost\_centre\_name, gl\_ob\_debit, gl\_ob\_credit, gl\_movement\_debit, gl\_movement\_credit, gl\_periodic\_movement, gl\_closing\_balance, gl\_last\_extracted, load\_date\_time, pipeline\_run\_id, finance\_year\_period\_number, year\_period\_number

HAVING dup\_count > 1



O rows returned, shows that duplicate data was not there.

TEST CASE 2 : 2-Verify on running the job pl\_gold\_load manually, if the job fails Then data shouldn't move from silver to gold. Email notification should also be triggered for the same.  
  


I ran it in dev environment which is pl\_gold\_load table, but due to lack of data in **delta\_silver.dim\_fusion\_account\_slv**  table, it have not ran and it failed. The data doesn’t moved to gold table, checked in ADF Monitor too.

TEST CASE 4 - 4-Verify all the dependency tables/files are available as a prerequisite for loading the table.

Ensure that all required source tables exist, contain data for the specified period, and that the aggregated source data matches the data in the gold table (fusion\_gl\_balances). This guarantees data integrity before proceeding with the load.

All dependency tables should be accessible:

* dim\_fdmee\_mapping\_slv
* fact\_fusion\_glbalances\_slv
* fact\_fdmee\_transformed\_data
* dim\_fusion\_account\_hierarchy\_slv
* dim\_fusion\_costcentre\_slv
* dim\_hfm\_entity\_hierarchy\_slv

TARGET TABLE : fusion\_gl\_balances (gold)

QUERY :

SELECT HFM\_ENTITY,

gl\_period,

gl\_scenario,

hfm\_account,

hfm\_category,

gl\_account,

gl\_cost\_centre\_code,

gl\_cost\_centre\_name,

finance\_year\_period\_number,

ROUND(gl\_closing\_balance, 2)

FROM (

SELECT a.target\_entity AS HFM\_ENTITY,

b.period AS gl\_period,

b.scenario AS gl\_scenario,

a.target\_account AS hfm\_account,

a.target\_ledger AS hfm\_category,

a.source\_account AS gl\_account,

a.source\_cost\_centre AS gl\_cost\_centre\_code,

e.description AS gl\_cost\_centre\_name,

CONCAT(SUBSTR(b.scenario, 1, 4), b.period) AS finance\_year\_period\_number,

SUM(c.closing\_balance) AS gl\_closing\_balance

FROM delta\_silver.dim\_fdmee\_mapping\_slv a

JOIN delta\_silver.fact\_fusion\_glbalances\_slv b

ON b.fusion\_coa\_account = a.source\_account

JOIN delta\_silver.fact\_fdmee\_transformed\_data c

ON c.fdmee\_fusion\_string = a.fusion\_string

LEFT JOIN delta\_silver.dim\_fusion\_costcentre\_slv e

ON a.source\_cost\_centre = e.fusion\_code

WHERE a.year\_period\_number = 202310

AND b.year\_period\_number = 202310

AND c.year\_period\_number = 202310

GROUP BY a.target\_entity, b.period, b.scenario,

a.target\_account, a.target\_ledger,

a.source\_account, a.source\_cost\_centre, e.description

)

MINUS

SELECT hfm\_entity,

gl\_period,

gl\_scenario,

hfm\_account,

hfm\_category,

gl\_account,

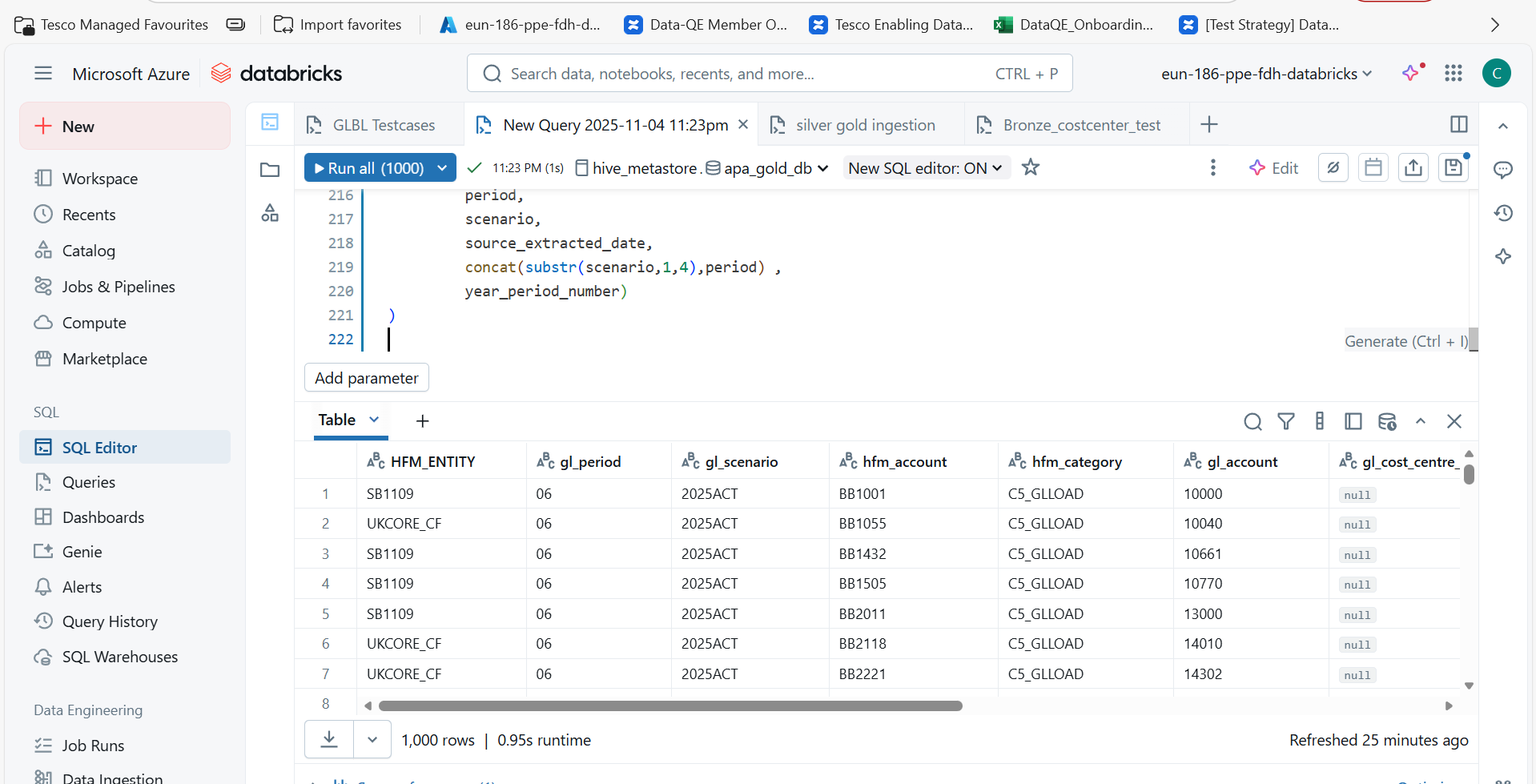
gl\_cost\_centre\_code,

gl\_cost\_centre\_name,

finance\_year\_period\_number,

ROUND(gl\_closing\_balance, 2)

FROM delta\_gold.fusion\_gl\_balances;



* Result : returned rows that are missing in delta\_gold.fusion\_gl\_balances.

Query Explanation :

* The query compares two datasets to find records that exist in the transformed silver-layer data but not in the gold-layer fusion\_gl\_balances. This is typically used for reconciliation of GL balances for a specific period.
* It aggregates data from multiple tables in the delta\_silver schema. The subquery maps source GL accounts and cost centers to HFM entities and accounts, calculates the closing balance, and prepares key fields like period, scenario, and finance year-period number.
* The subquery joins mapping (dim\_fdmee\_mapping\_slv), GL balances (fact\_fusion\_glbalances\_slv), transformed data (fact\_fdmee\_transformed\_data), and cost center descriptions (dim\_fusion\_costcentre\_slv). It filters for year\_period\_number = 202310 (October 2023) and groups by entity, account, and cost center.
* The outer query selects the same columns, rounds the closing balance, and uses MINUS to return rows that are missing in delta\_gold.fusion\_gl\_balances.

TEST CASE 6 - 6-Verify the data is loading for all the default columns.

The test case validates that when the pipeline runs successfully and default columns are populated, all default columns contain valid data. The same SQL query is used to check data consistency between the silver-layer transformation and the gold-layer final balances.

Used same query

And I got same output of rows

TEST CASE 7 : 7-Check for primary key column

This table doesn’t have Primary column , so we need to check any duplicates present in table or not .

SQL to detect duplicates:

SELECT hfm\_entity,

       gl\_period,

       gl\_scenario,

       hfm\_account,

       hfm\_category,

       gl\_account,

       gl\_cost\_centre\_code,

       finance\_year\_period\_number,

       year\_period\_number,

       COUNT(\*) AS duplicate\_count

FROM delta\_gold.fusion\_gl\_balances

GROUP BY hfm\_entity,

         gl\_period,

         gl\_scenario,

         hfm\_account,

         hfm\_category,

         gl\_account,

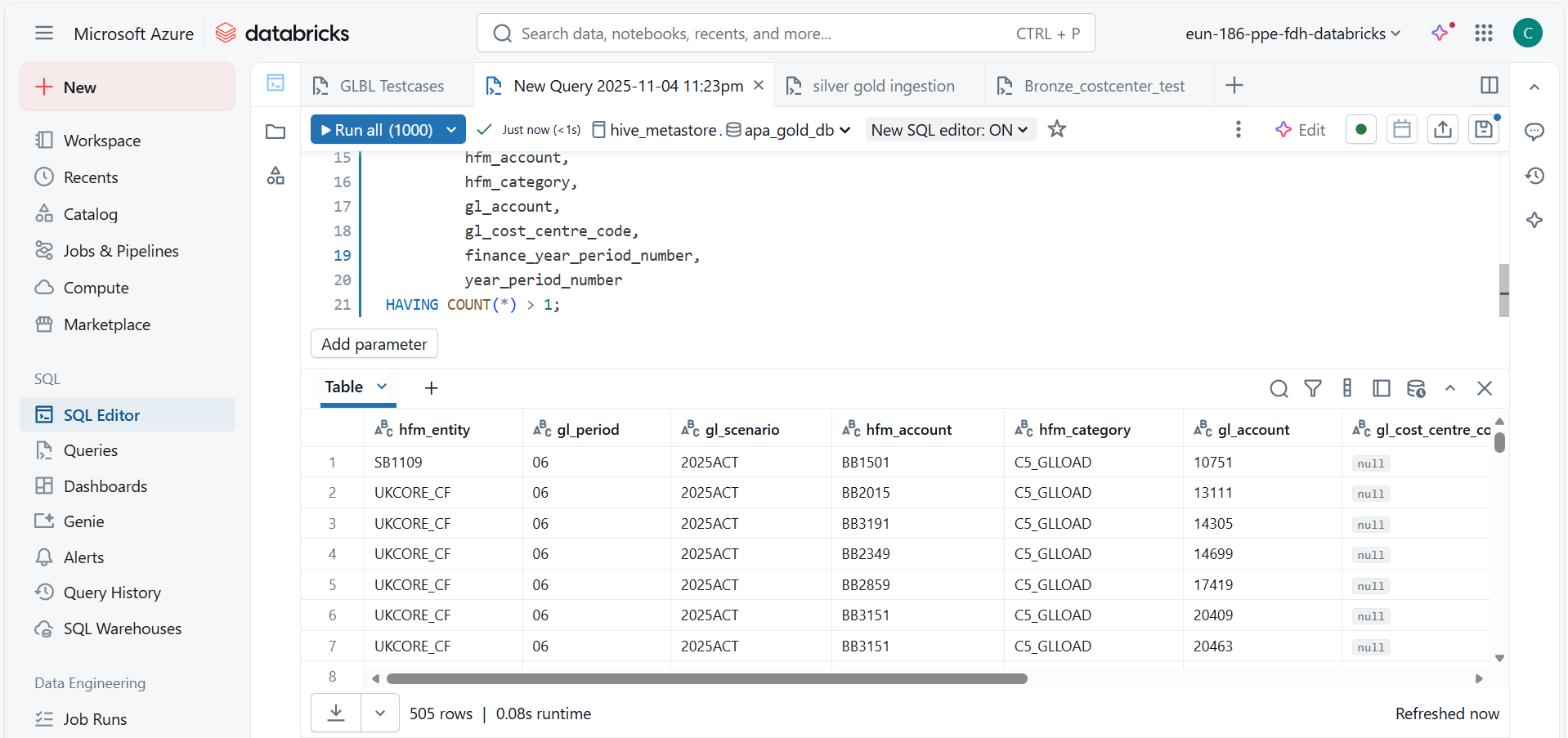
         gl\_cost\_centre\_code,

         finance\_year\_period\_number,

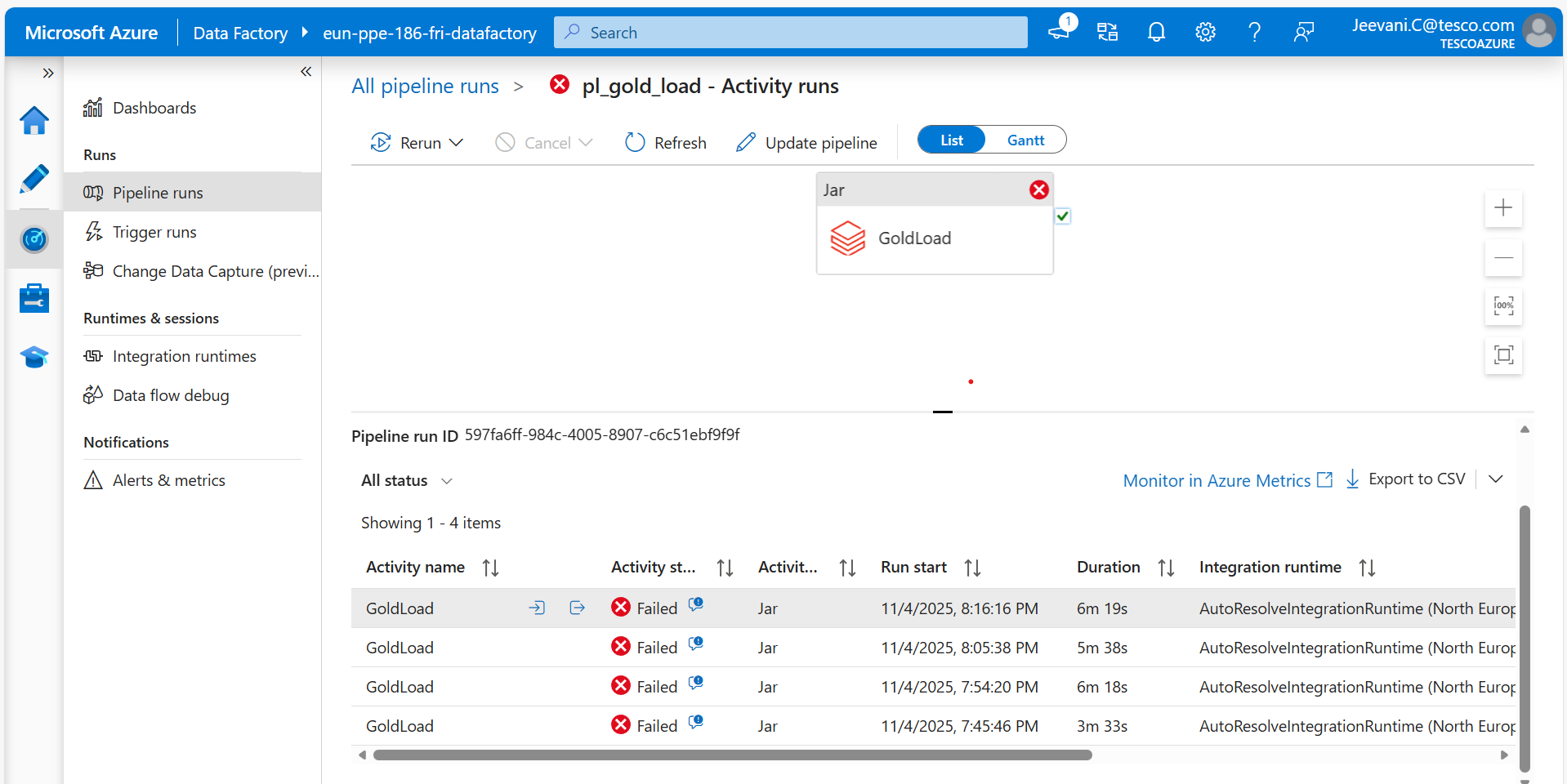
         year\_period\_number

HAVING COUNT(\*) > 1;

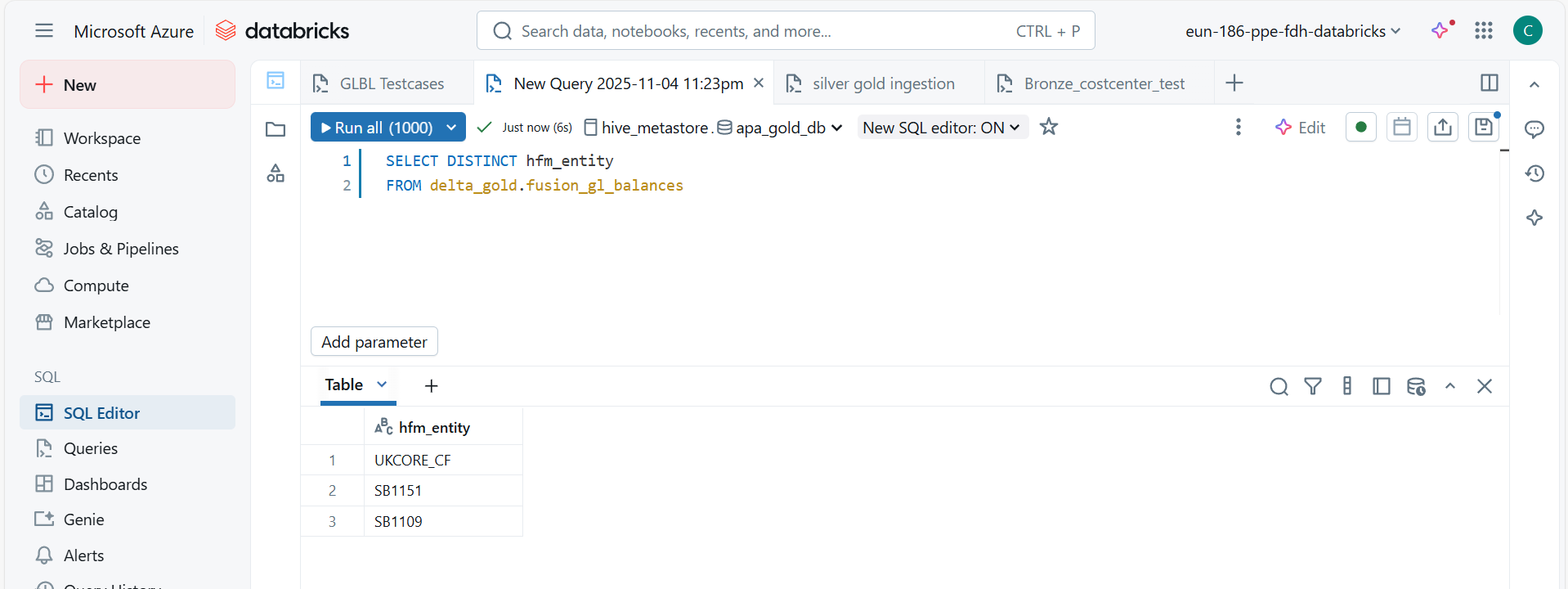
* GROUP BY on the columns that should uniquely identify a record.
* COUNT(\*) > 1 in HAVING finds groups with more than one row means duplicates.



It returned 505 rows, shows duplicates in gold table

TEST CASE 11. 11-Verify the email notification for successful/failure job run.  
  
I ran pl\_gold\_load, but I didn’t get any email notification  


TEST CASE 14 : 14-Verify the column hfm\_entity is picked from delta\_silver.dim\_fdmee\_mapping\_slv(target\_entity) only for 'UKCORE\_TOT' entities except 'SB4001','SB4002','SB4003','SB4006' and named as 'UKCORE\_CF' except for 'SB1109','SB1151'



The query shows that data is having unique hfm entities which was shared in same in test case.

TEST CASE 20 : 20-Verify the column hfm\_account is mapped from target\_account in delta\_silver.dim\_fdmee\_mapping\_slv against the source\_account

To Verify that hfm\_account in delta\_gold.fusion\_gl\_balances is correctly mapped from target\_account in delta\_silver.dim\_fdmee\_mapping\_slv, based on the corresponding source\_account.  
  
Query :

SELECT

    f.year\_period\_number,

    f.hfm\_account,

    m.target\_account,

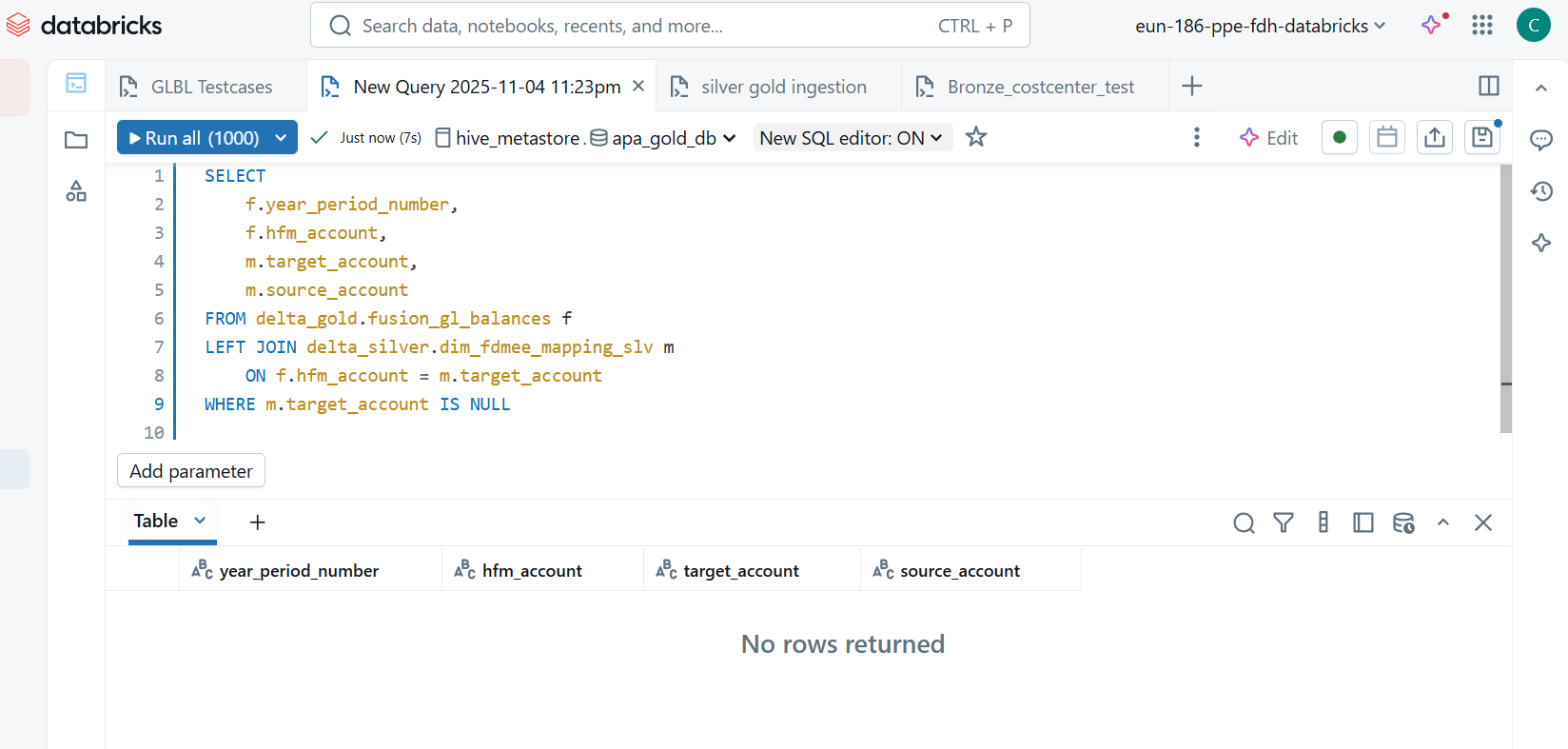
    m.source\_account

FROM delta\_gold.fusion\_gl\_balances f

LEFT JOIN delta\_silver.dim\_fdmee\_mapping\_slv m

    ON f.hfm\_account = m.target\_account

WHERE m.target\_account IS NULL



It returned Zero rows, shows mapping are perfect in gold table.

TEST CASE 21 : 21-Verify the column hfm\_category is mapped from target\_ledger in delta\_silver.dim\_fdmee\_mapping\_slv against the source\_account

To Verify that hfm\_category in delta\_gold.fusion\_gl\_balances is mapped from target\_ledger in delta\_silver.dim\_fdmee\_mapping\_slv against the corresponding source\_account.

Query :

SELECT

    f.year\_period\_number,

    f.hfm\_category,

    m.target\_ledger,

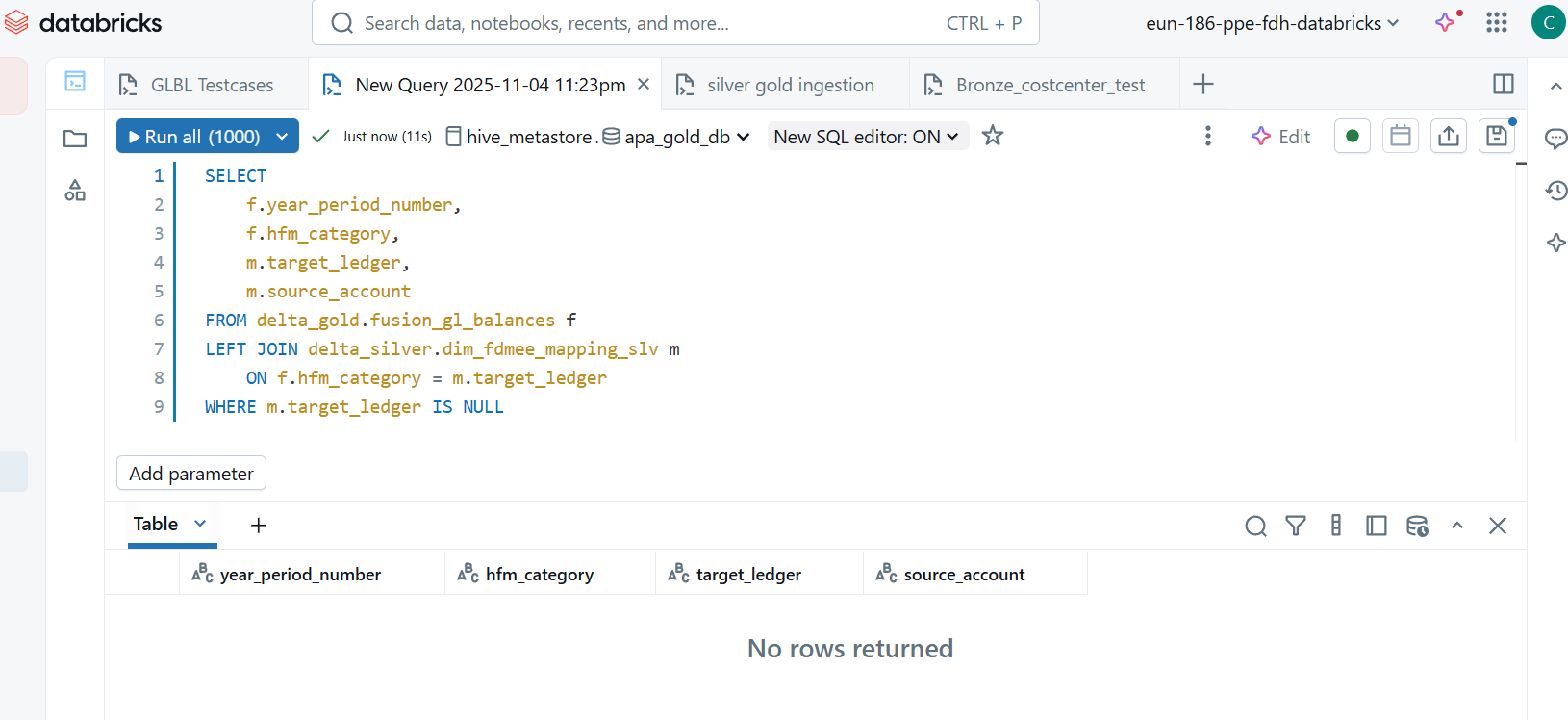
    m.source\_account

FROM delta\_gold.fusion\_gl\_balances f

LEFT JOIN delta\_silver.dim\_fdmee\_mapping\_slv m

    ON f.hfm\_category = m.target\_ledger

WHERE m.target\_ledger IS NULL



Empty results shows mapping is correct

TEST CASE 23 : 23-Verify the column gl\_account is mapped from node in delta\_silver.dim\_fusion\_account\_hierarchy only for Balance Sheet and P&L accounts

To Verify that gl\_account in delta\_gold.fusion\_gl\_balances is mapped from fusion\_code (node) in delta\_silver.dim\_fusion\_account\_hierarchy\_slv, **only for Balance Sheet and P&L accounts**.

Query :   
  
SELECT

    f.year\_period\_number,

    f.gl\_account,

    h.fusion\_code,

    h.account\_nature

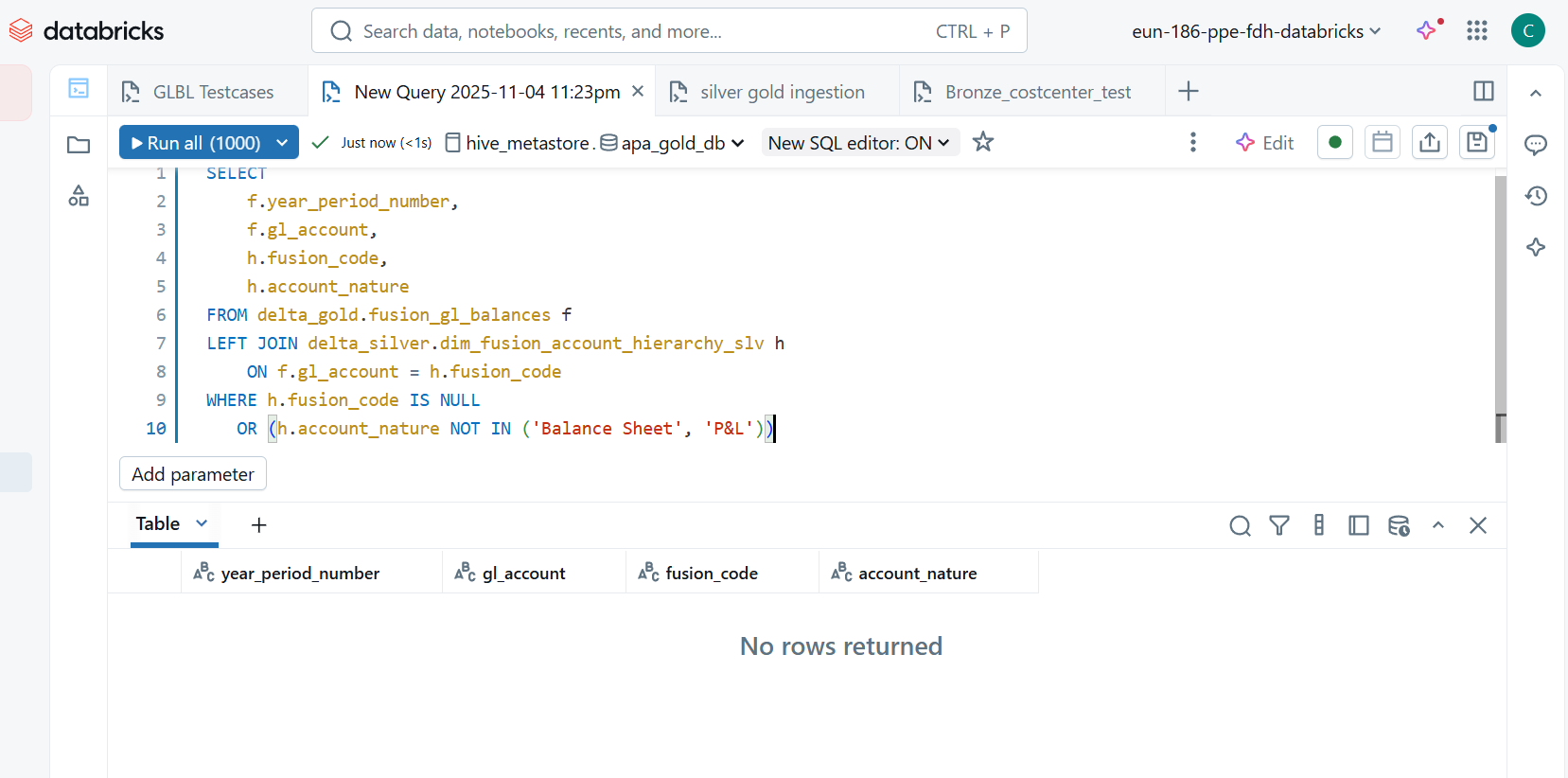
FROM delta\_gold.fusion\_gl\_balances f

LEFT JOIN delta\_silver.dim\_fusion\_account\_hierarchy\_slv h

    ON f.gl\_account = h.fusion\_code

WHERE h.fusion\_code IS NULL

   OR (h.account\_nature NOT IN ('Balance Sheet', 'P&L'))

****

No rows returned shows that mapping is perfect.

TEST CASE 25 : 25-Verify the column gl\_cost\_centre\_code is mapped from source\_cost\_centre in delta\_silver.dim\_fdmee\_mapping\_slv against the source\_account, leave blank for Balance Sheet accounts  
To verify gl\_cost\_centre\_code in **delta\_gold.fusion\_gl\_balances** is mapped from source\_cost\_centre in **delta\_silver.dim\_fdmee\_mapping\_slv** against the corresponding source\_account.

For **Balance Sheet accounts**, gl\_cost\_centre\_code should be **blank** (NULL or empty).

Query :

SELECT

    f.year\_period\_number,

    f.gl\_account,

    f.gl\_cost\_centre\_code,

    m.source\_cost\_centre,

    m.source\_account,

    m.target\_ledger AS hfm\_category

FROM delta\_gold.fusion\_gl\_balances f

LEFT JOIN delta\_silver.dim\_fdmee\_mapping\_slv m

    ON f.hfm\_account = m.target\_account

WHERE ((m.target\_ledger = 'P&L' AND f.gl\_cost\_centre\_code <> m.source\_cost\_centre)

        OR

        (m.target\_ledger = 'Balance Sheet' AND (f.gl\_cost\_centre\_code IS NOT NULL AND f.gl\_cost\_centre\_code <> '')))



No rows returned shows that mapping is perfect.

Test Case 31 : Verify the columns gl\_ob\_debit, gl\_ob\_credit, gl\_movement\_debit, gl\_movement\_credit, gl\_periodic\_movement, gl\_closing\_balance are mapped from delta\_silver.fact\_fdmee\_transformed\_data, summed and aggregated at fusion\_account, fusion\_cost\_centre\_code

Descritpion :

This test validates that the columns in the **gold table** delat\_gold.fusion\_gl\_balances (gl\_ob\_debit, gl\_ob\_credit, gl\_movement\_debit, gl\_movement\_credit, gl\_periodic\_movement, gl\_closing\_balance) are correctly mapped and aggregated from the **silver table** delta\_silver.fact\_fdmee\_transformed\_data.  
The aggregation should be done by summing the respective columns in the silver table and grouping by fdmee\_fusion\_string (mapped to gl\_account) and source\_cost\_centre (mapped to gl\_cost\_centre\_code).

Query :

SELECT

    g.gl\_account AS fusion\_account,

    g.gl\_cost\_centre\_code AS fusion\_cost\_centre\_code,

    g.gl\_ob\_debit,

    s.silver\_ob\_debit,

    ABS(g.gl\_ob\_debit - s.silver\_ob\_debit) AS diff\_ob\_debit,

    g.gl\_ob\_credit,

    s.silver\_ob\_credit,

    ABS(g.gl\_ob\_credit - s.silver\_ob\_credit) AS diff\_ob\_credit,

    g.gl\_movement\_debit,

    s.silver\_movement\_debit,

    ABS(g.gl\_movement\_debit - s.silver\_movement\_debit) AS diff\_movement\_debit,

    g.gl\_movement\_credit,

    s.silver\_movement\_credit,

    ABS(g.gl\_movement\_credit - s.silver\_movement\_credit) AS diff\_movement\_credit,

    g.gl\_periodic\_movement,

    s.silver\_periodic\_movement,

    ABS(g.gl\_periodic\_movement - s.silver\_periodic\_movement) AS diff\_periodic\_movement,

    g.gl\_closing\_balance,

    s.silver\_closing\_balance,

    ABS(g.gl\_closing\_balance - s.silver\_closing\_balance) AS diff\_closing\_balance

FROM delta\_gold.fusion\_gl\_balances g

LEFT JOIN (

    SELECT

        fdmee\_fusion\_string AS fusion\_account,

        source\_cost\_centre AS fusion\_cost\_centre\_code,

        SUM(ob\_debit) AS silver\_ob\_debit,

        SUM(ob\_credit) AS silver\_ob\_credit,

        SUM(movement\_debit) AS silver\_movement\_debit,

        SUM(movement\_credit) AS silver\_movement\_credit,

        SUM(periodic\_movement) AS silver\_periodic\_movement,

        SUM(closing\_balance) AS silver\_closing\_balance

    FROM delta\_silver.fact\_fdmee\_transformed\_data

    GROUP BY fdmee\_fusion\_string, source\_cost\_centre

) s

ON g.gl\_account = s.fusion\_account

AND g.gl\_cost\_centre\_code = s.fusion\_cost\_centre\_code

WHERE

    ABS(g.gl\_ob\_debit - s.silver\_ob\_debit) > 0.01 OR

    ABS(g.gl\_ob\_credit - s.silver\_ob\_credit) > 0.01 OR

    ABS(g.gl\_movement\_debit - s.silver\_movement\_debit) > 0.01 OR

    ABS(g.gl\_movement\_credit - s.silver\_movement\_credit) > 0.01 OR

    ABS(g.gl\_periodic\_movement - s.silver\_periodic\_movement) > 0.01 OR

    ABS(g.gl\_closing\_balance - s.silver\_closing\_balance) > 0.01;



The query was designed to return rows **only when there are mismatches** between the gold table values and the aggregated silver table values.

Since **zero rows were returned**, it means:

* All numeric columns (gl\_ob\_debit, gl\_ob\_credit, gl\_movement\_debit, gl\_movement\_credit, gl\_periodic\_movement, gl\_closing\_balance) in delta\_gold.fusion\_gl\_balances **match perfectly** with the aggregated values from delta\_silver.fact\_fdmee\_transformed\_data for the corresponding gl\_account and gl\_cost\_centre\_code.

TEST CASE 32 : 32-Verify the columns gl\_ob\_debit, gl\_ob\_credit, gl\_movement\_debit, gl\_movement\_credit, gl\_periodic\_movement, gl\_closing\_balance are numeric

Ensure that the columns gl\_ob\_debit, gl\_ob\_credit, gl\_movement\_debit, gl\_movement\_credit, gl\_periodic\_movement, and gl\_closing\_balance in delta\_gold.fusion\_gl\_balances are numeric (Double).

Query :

SELECT \*

FROM delta\_gold.fusion\_gl\_balances

WHERE

    CAST(gl\_ob\_debit AS DOUBLE) IS NULL AND gl\_ob\_debit IS NOT NULL

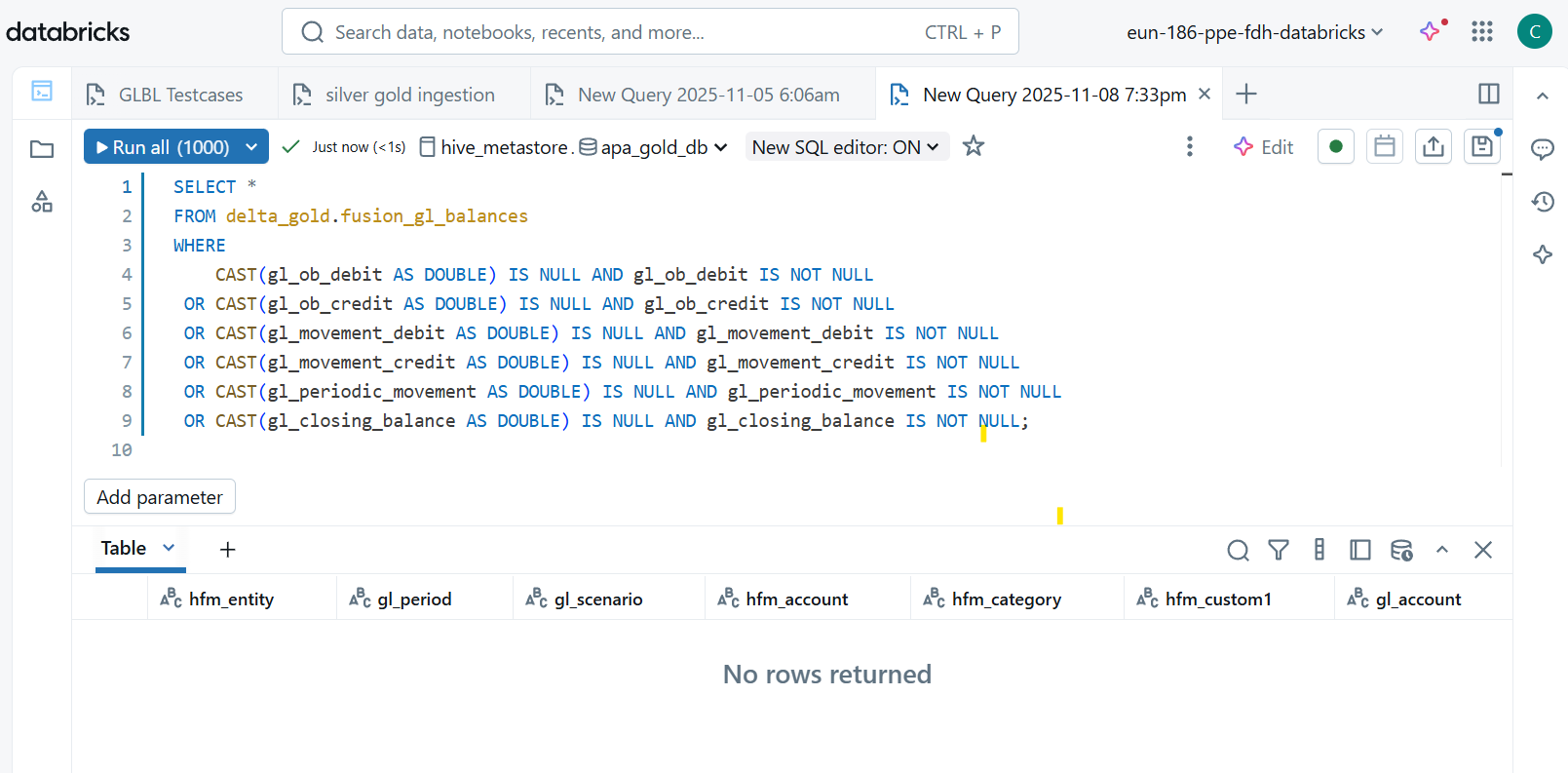
 OR CAST(gl\_ob\_credit AS DOUBLE) IS NULL AND gl\_ob\_credit IS NOT NULL

 OR CAST(gl\_movement\_debit AS DOUBLE) IS NULL AND gl\_movement\_debit IS NOT NULL

 OR CAST(gl\_movement\_credit AS DOUBLE) IS NULL AND gl\_movement\_credit IS NOT NULL

 OR CAST(gl\_periodic\_movement AS DOUBLE) IS NULL AND gl\_periodic\_movement IS NOT NULL

 OR CAST(gl\_closing\_balance AS DOUBLE) IS NULL AND gl\_closing\_balance IS NOT NULL;



query returns **zero rows**, all values in these columns are numeric (or NULL).

TEST CASE 27 : Ensure that gl\_cost\_centre\_name in delta\_gold.fusion\_gl\_balances is mapped from description in delta\_silver.dim\_fusion\_costcentre\_slv based on gl\_cost\_centre\_code = fusion\_code. For Balance Sheet accounts (identified by account\_type = 'Balance Sheet' in the silver table), gl\_cost\_centre\_name should be blank.

Ensure that gl\_cost\_centre\_name in delta\_gold.fusion\_gl\_balances is mapped from description in delta\_silver.dim\_fusion\_costcentre\_slv based on gl\_cost\_centre\_code = fusion\_code. For Balance Sheet accounts (identified by account\_type = 'Balance Sheet' in the silver table), gl\_cost\_centre\_name should be blank.

Query :

SELECT

    g.gl\_account,

    g.gl\_cost\_centre\_code,

    g.gl\_cost\_centre\_name,

    d.description AS expected\_description,

    d.account\_type

FROM delta\_gold.fusion\_gl\_balances g

LEFT JOIN delta\_silver.dim\_fusion\_costcentre\_slv d

    ON g.gl\_cost\_centre\_code = d.fusion\_code

WHERE

    (d.account\_type <> 'Balance Sheet' AND g.gl\_cost\_centre\_name <> d.description)

    OR (d.account\_type = 'Balance Sheet' AND g.gl\_cost\_centre\_name IS NOT NULL AND g.gl\_cost\_centre\_name <> '');



TEST CASE 29 : 29-Verify the column gl\_last\_extracted is mapped from source\_extracted\_date in delta\_silver.fact\_fusion\_glbalances\_slv only for Tesco GB Primary ledger

Validate that gl\_last\_extracted in delta\_gold.fusion\_gl\_balances is mapped from source\_extracted\_date in delta\_silver.fact\_fusion\_glbalances\_slv for records where the ledger corresponds to **Tesco GB Primary** (identified via source\_ledger in dim\_fdmee\_mapping\_slv).

Query :

SELECT

    g.gl\_account,

    g.gl\_cost\_centre\_code,

    g.gl\_last\_extracted,

    s.source\_extracted\_date,

    m.source\_ledger

FROM delta\_gold.fusion\_gl\_balances g

LEFT JOIN (

    SELECT fusion\_coa\_account, source\_extracted\_date, year\_period\_number

    FROM delta\_silver.fact\_fusion\_glbalances\_slv

) s ON g.gl\_account = s.fusion\_coa\_account

LEFT JOIN (

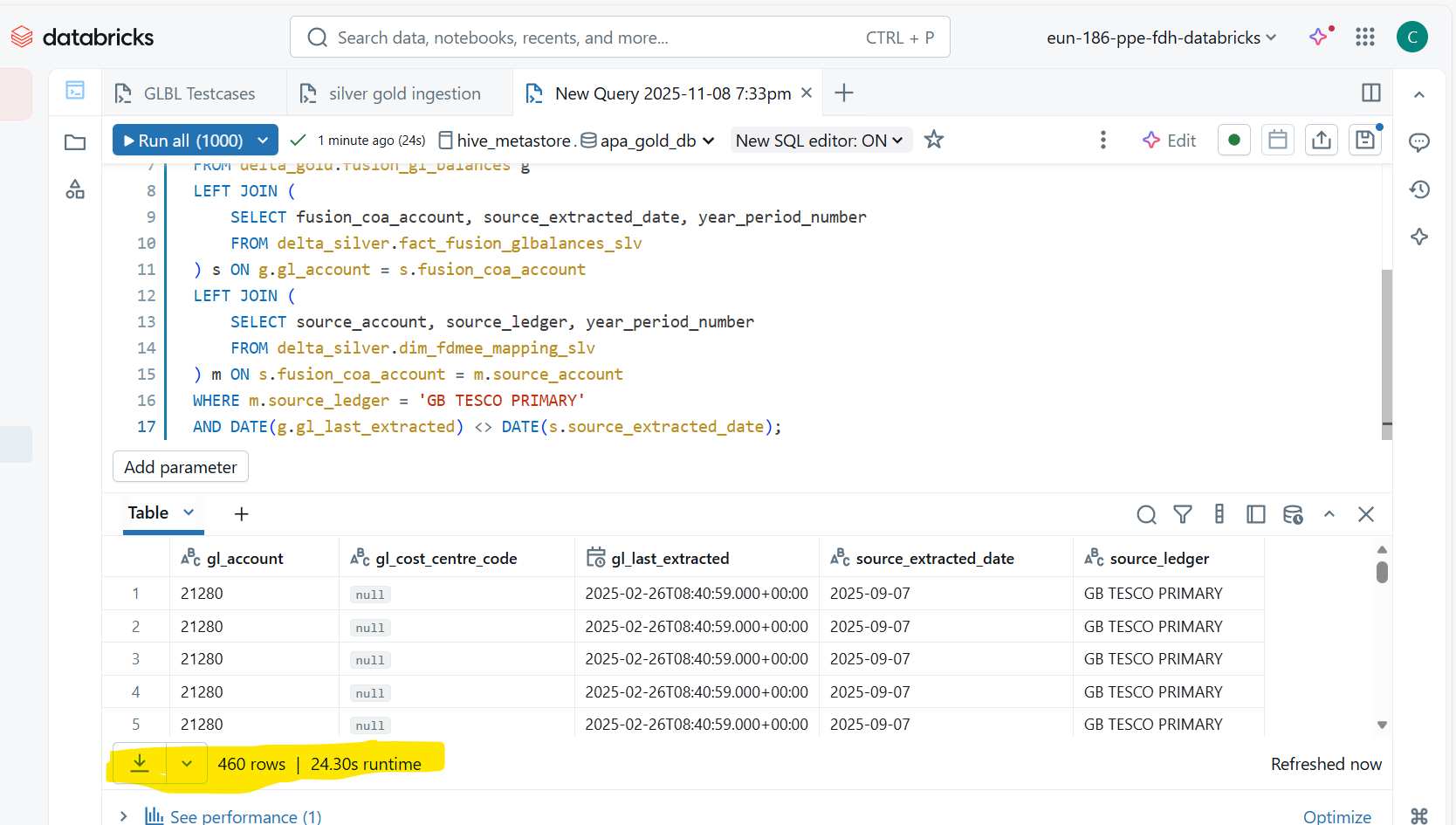
    SELECT source\_account, source\_ledger, year\_period\_number

    FROM delta\_silver.dim\_fdmee\_mapping\_slv

) m ON s.fusion\_coa\_account = m.source\_account

WHERE m.source\_ledger = 'GB TESCO PRIMARY'

AND DATE(g.gl\_last\_extracted) <> DATE(s.source\_extracted\_date);



there are **480 mismatches** where gl\_last\_extracted in the gold table does not match source\_extracted\_date from the silver table for Tesco GB Primary ledger.

TEST CASE 18 : 18-Verify the column gl\_scenario is picked from delta\_silver.fact\_fusion\_glbalances\_slv only for Tesco GB Primary ledger and transformed as concat('ACT', balance\_period\_year)

Ensure that the column gl\_scenario in the gold table delta\_gold.fusion\_gl\_balances is derived from the silver table delta\_silver.fact\_fusion\_glbalances\_slv for Tesco GB Primary ledger and transformed as concat('ACT', balance\_period\_year).

Query :

SELECT

g.gl\_account,

g.gl\_scenario,

CONCAT('ACT', s.balance\_period\_year) AS expected\_gl\_scenario,

m.source\_ledger

FROM delta\_gold.fusion\_gl\_balances g

LEFT JOIN (

SELECT fusion\_coa\_account, balance\_period\_year, year\_period\_number

FROM delta\_silver.fact\_fusion\_glbalances\_slv

) s ON g.gl\_account = s.fusion\_coa\_account

LEFT JOIN (

SELECT source\_account, source\_ledger, year\_period\_number

FROM delta\_silver.dim\_fdmee\_mapping\_slv

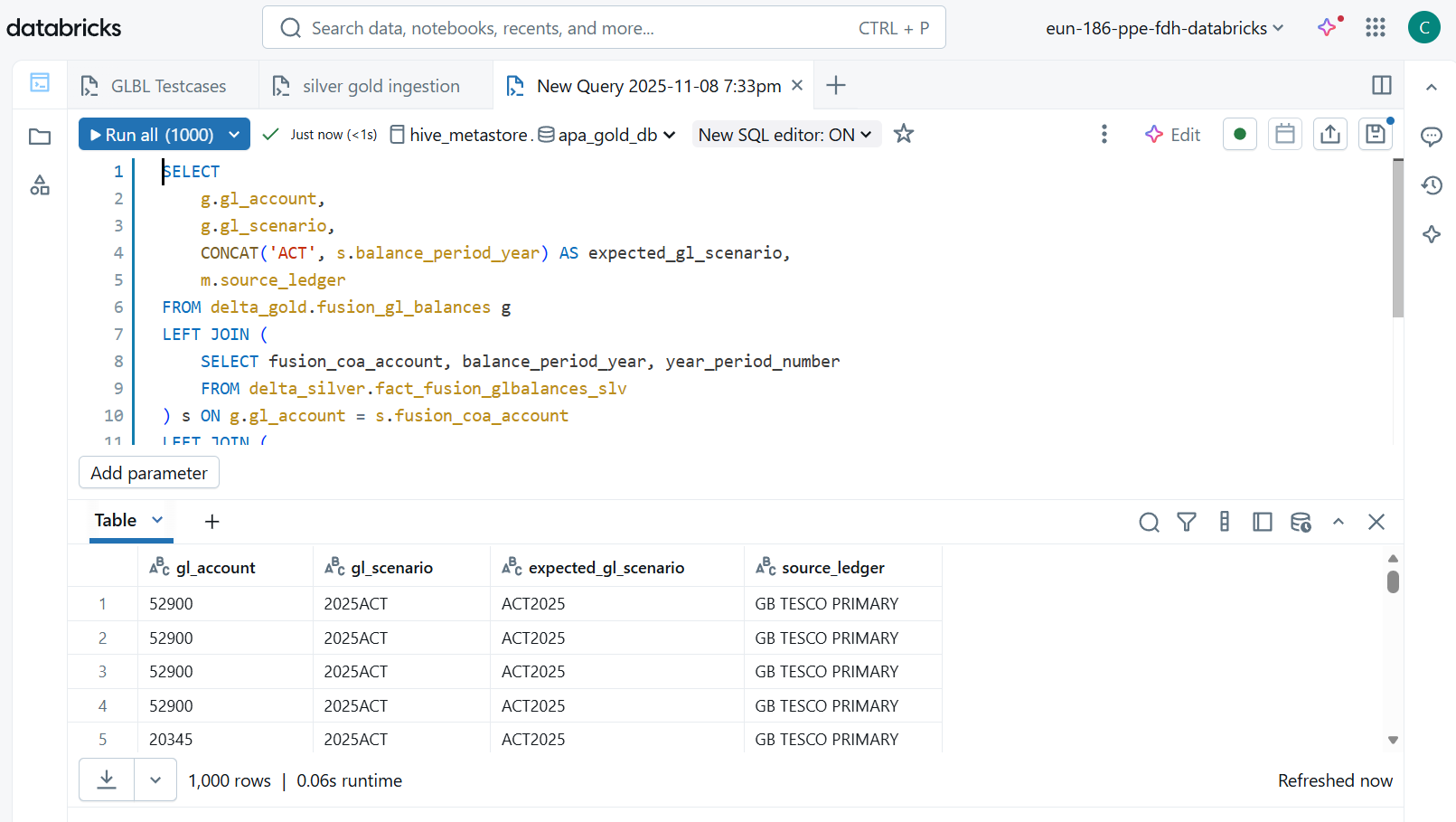
) m ON s.fusion\_coa\_account = m.source\_account

AND g.year\_period\_number = s.year\_period\_number

AND g.year\_period\_number = m.year\_period\_number

WHERE m.source\_ledger = 'GB TESCO PRIMARY'

AND g.gl\_scenario <> CONCAT('ACT', s.balance\_period\_year);



Mismatched

TEST CASE 16 : 16-Verify the column gl\_period is picked from delta\_silver.fact\_fusion\_glbalances\_slv only for Tesco GB Primary ledger and transformed as concat('0', balance\_period\_num) if length is 1

Ensure that the column gl\_period in the gold table delta\_gold.fusion\_gl\_balances is derived from balance\_period\_num in the silver table delta\_silver.fact\_fusion\_glbalances\_slv for Tesco GB Primary ledger, and transformed as:

If length(balance\_period\_num) = 1 → concat('0', balance\_period\_num)

Else → balance\_period\_num

QUERY :

SELECT

g.gl\_account,

g.gl\_period,

CASE

WHEN LENGTH(s.balance\_period\_num) = 1 THEN CONCAT('0', s.balance\_period\_num)

ELSE s.balance\_period\_num

END AS expected\_gl\_period,

m.source\_ledger

FROM delta\_gold.fusion\_gl\_balances g

LEFT JOIN (

SELECT fusion\_coa\_account, balance\_period\_num, year\_period\_number

FROM delta\_silver.fact\_fusion\_glbalances\_slv

) s ON g.gl\_account = s.fusion\_coa\_account

LEFT JOIN (

SELECT source\_account, source\_ledger, year\_period\_number

FROM delta\_silver.dim\_fdmee\_mapping\_slv

) m ON s.fusion\_coa\_account = m.source\_account

AND g.year\_period\_number = s.year\_period\_number

AND g.year\_period\_number = m.year\_period\_number

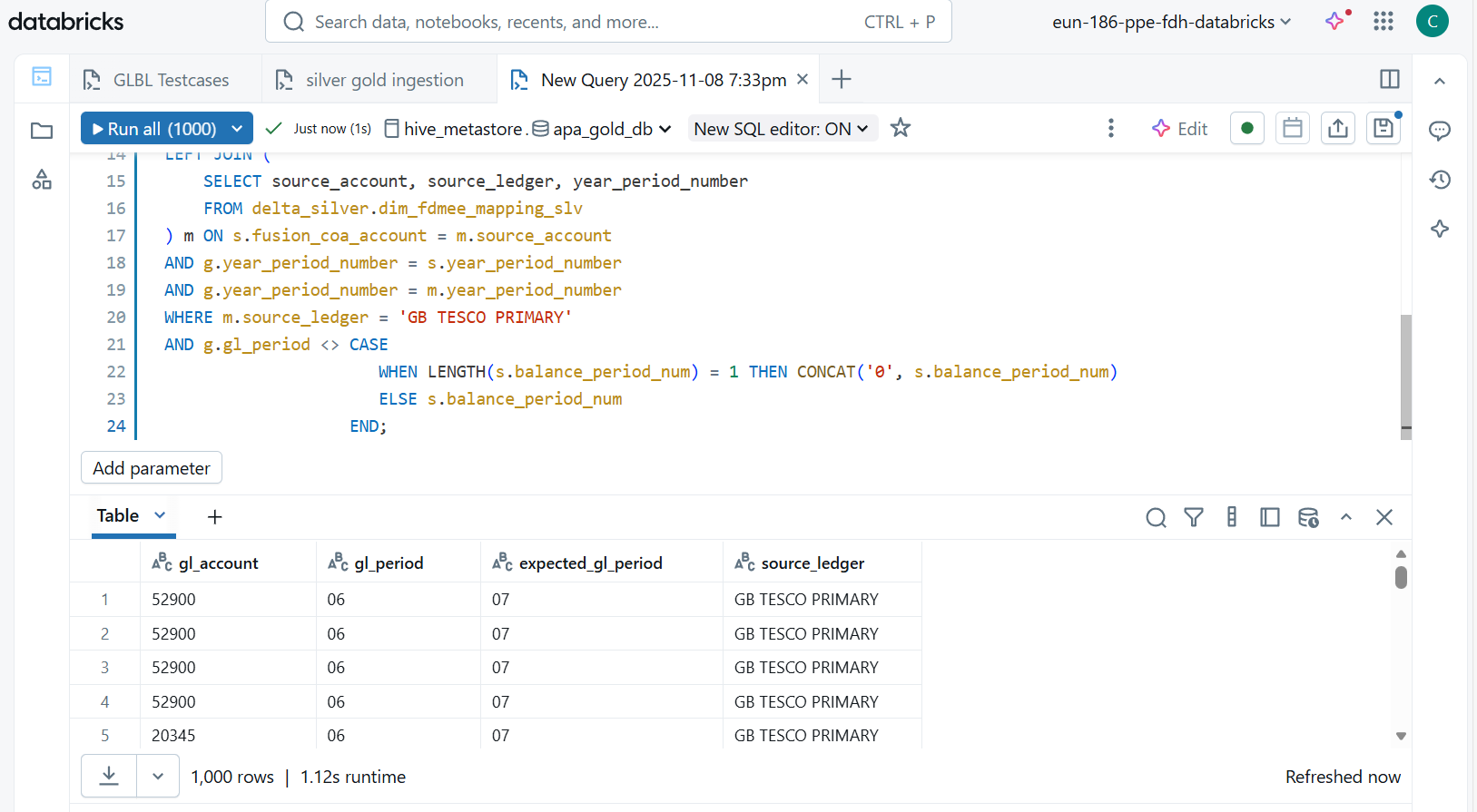
WHERE m.source\_ledger = 'GB TESCO PRIMARY'

AND g.gl\_period <> CASE

WHEN LENGTH(s.balance\_period\_num) = 1 THEN CONCAT('0', s.balance\_period\_num)

ELSE s.balance\_period\_num

END;

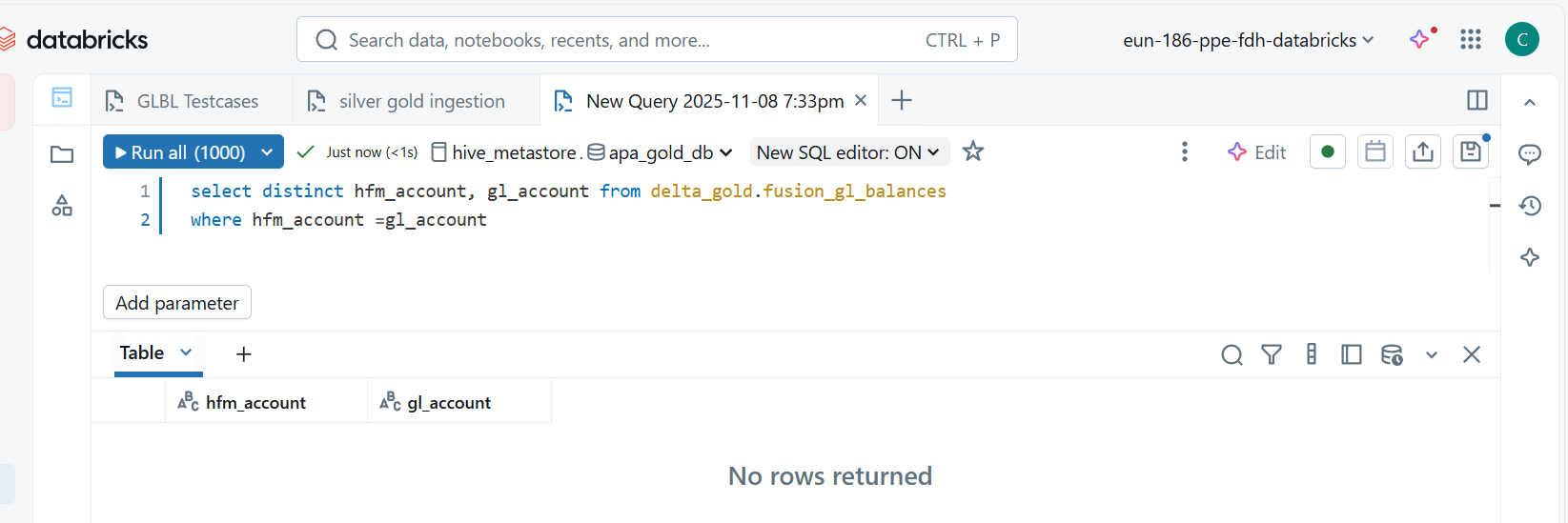


We have rows returned

TEST CASE 48-Verify the HFM code and Fusion Code are not same

QUERY : select distinct hfm\_account, gl\_account from delta\_gold.fusion\_gl\_balances

where hfm\_account =gl\_account



Test case passed.

TEST CASE 42 : 42-Validate any primary key (gl\_account, gl\_cost\_centre\_code) duplicate is there in Gold Layer delta\_gold.fusion\_gl\_balances  
  
Ensure that the combination of gl\_account and gl\_cost\_centre\_code in delta\_gold.fusion\_gl\_balances is unique (no duplicates).

QUERY :

SELECT

gl\_account,

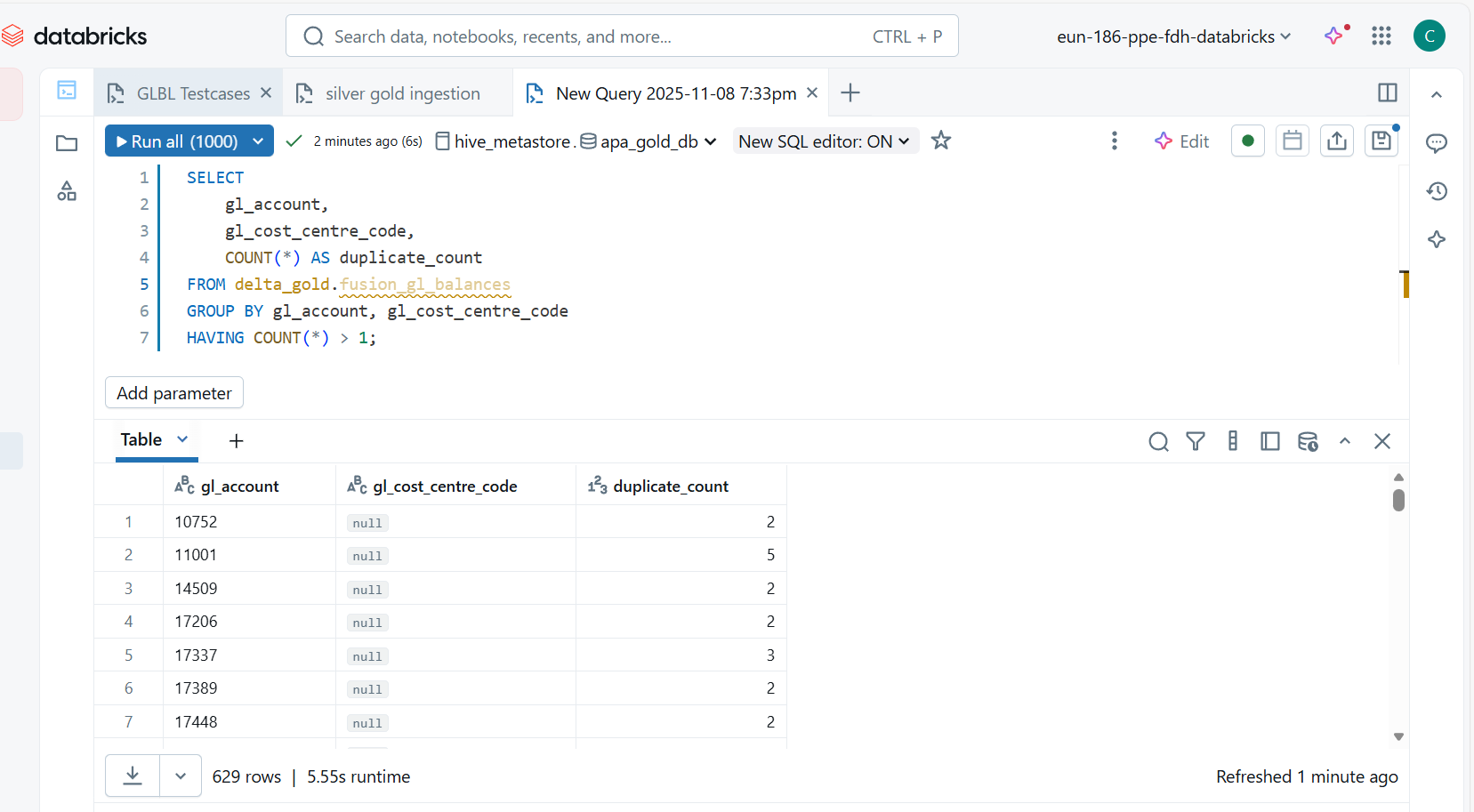
gl\_cost\_centre\_code,

COUNT(\*) AS duplicate\_count

FROM delta\_gold.fusion\_gl\_balances

GROUP BY gl\_account, gl\_cost\_centre\_code

HAVING COUNT(\*) > 1;



Duplicates are there

TEST CASE 39 : 39-Validate finance\_year\_period\_number is null or defaulted as per logic

Ensure that the column finance\_year\_period\_number in delta\_gold.fusion\_gl\_balances is either:

* **Null** when no valid transformation applies, OR
* **Defaulted correctly** based on the business logic (usually concat(substr(gl\_scenario,1,4), gl\_period)).

**✅ Business Logic:**

* If gl\_scenario and gl\_period exist, then:
* finance\_year\_period\_number = concat(substr(gl\_scenario,1,4), gl\_period)
* Else, it should be **NULL** or a defined default.

QUERY :

SELECT

    gl\_account,

    gl\_period,

    gl\_scenario,

    finance\_year\_period\_number,

    CASE

        WHEN gl\_scenario IS NOT NULL AND gl\_period IS NOT NULL

        THEN CONCAT(SUBSTR(gl\_scenario,1,4), gl\_period)

        ELSE NULL

    END AS expected\_finance\_year\_period\_number

FROM delta\_gold.fusion\_gl\_balances

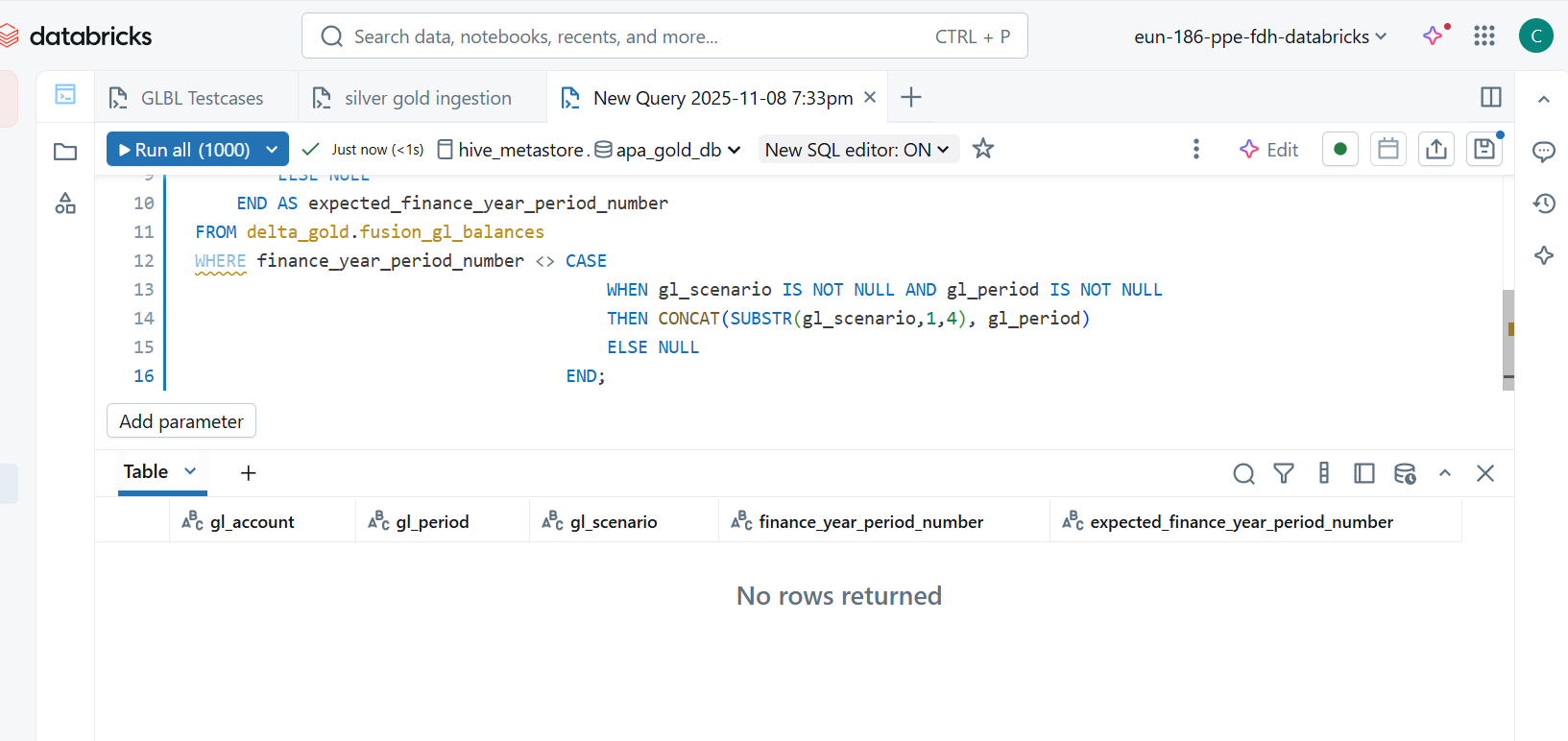
WHERE finance\_year\_period\_number <> CASE

                                        WHEN gl\_scenario IS NOT NULL AND gl\_period IS NOT NULL

                                        THEN CONCAT(SUBSTR(gl\_scenario,1,4), gl\_period)

                                        ELSE NULL

                                    END;



TEST CASE 38-Validate pipeline\_run\_id is correctly generated and unique per run

Ensure that the column pipeline\_run\_id in delta\_gold.fusion\_gl\_balances:

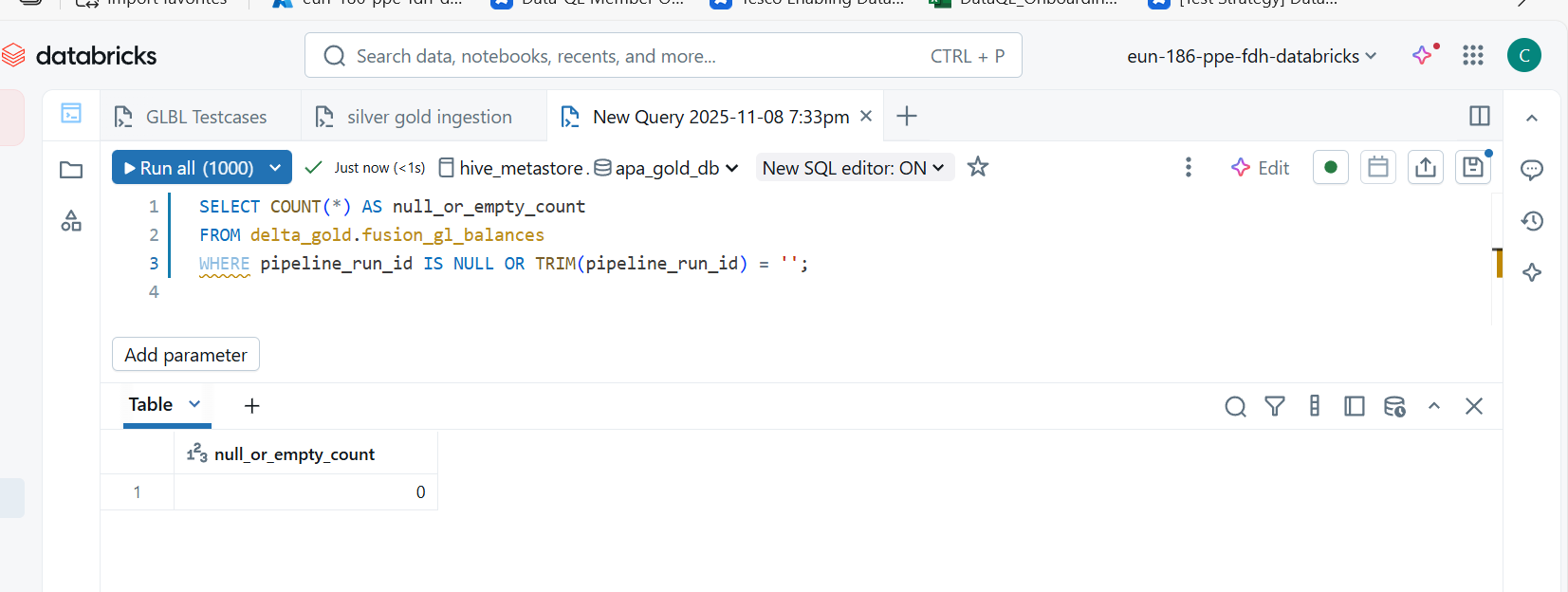
* Is **not null**.
* Is correctly generated (matches expected format or pattern).
* Is **unique for each pipeline run**.

QUERY :

SELECT COUNT(\*) AS null\_or\_empty\_count

FROM delta\_gold.fusion\_gl\_balances

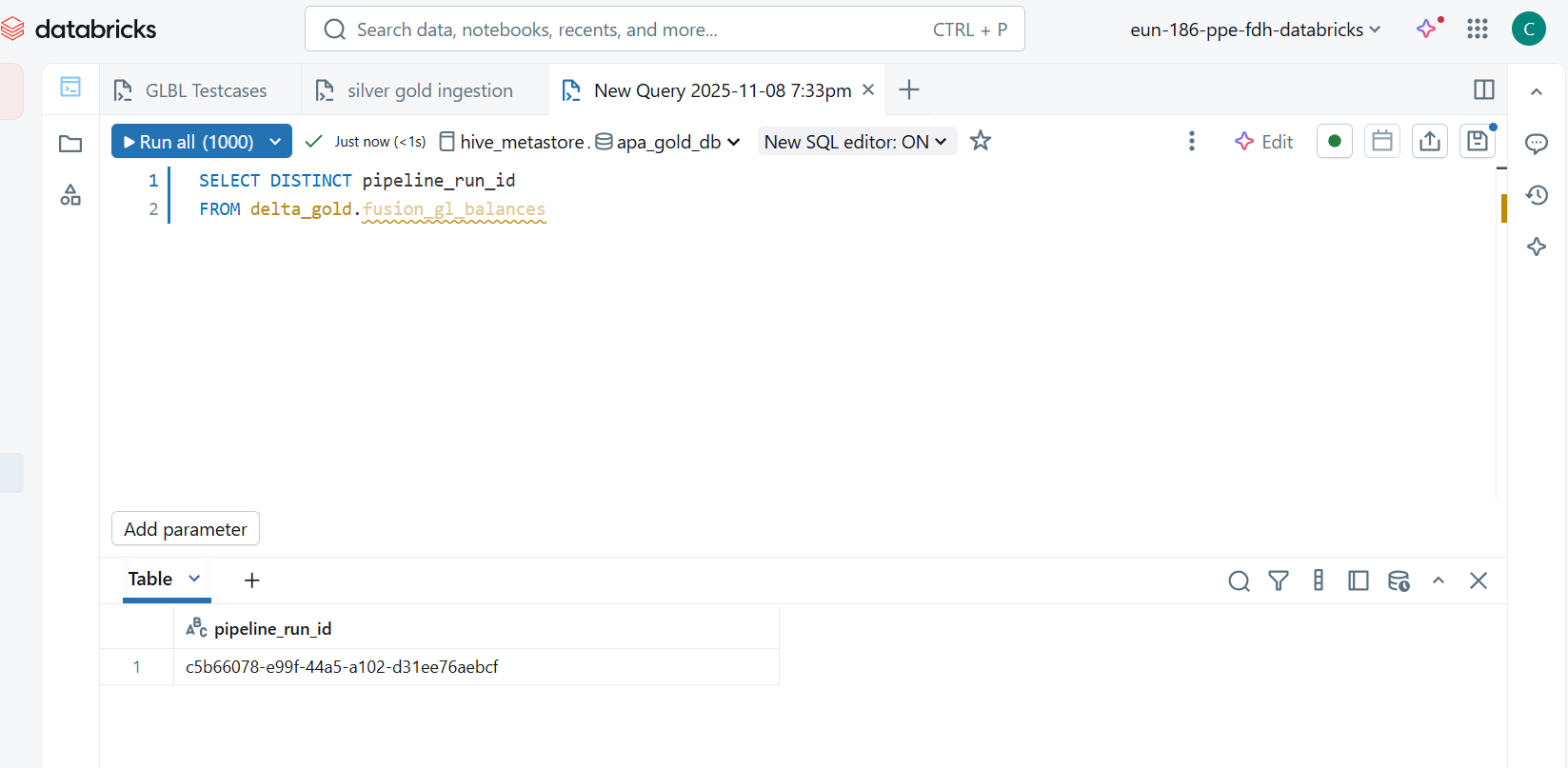
WHERE pipeline\_run\_id IS NULL OR TRIM(pipeline\_run\_id) = '';



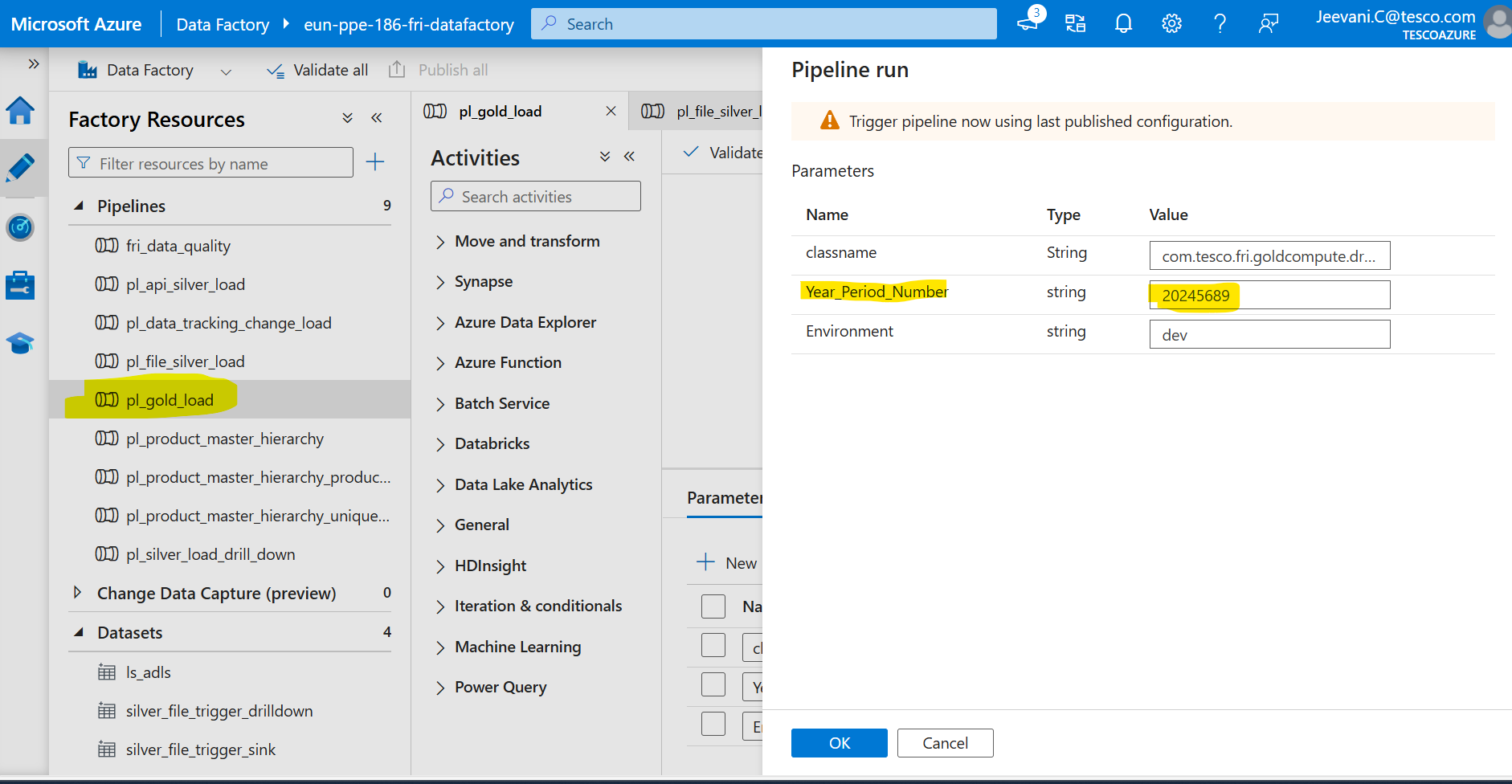
QUERY :

SELECT DISTINCT pipeline\_run\_id

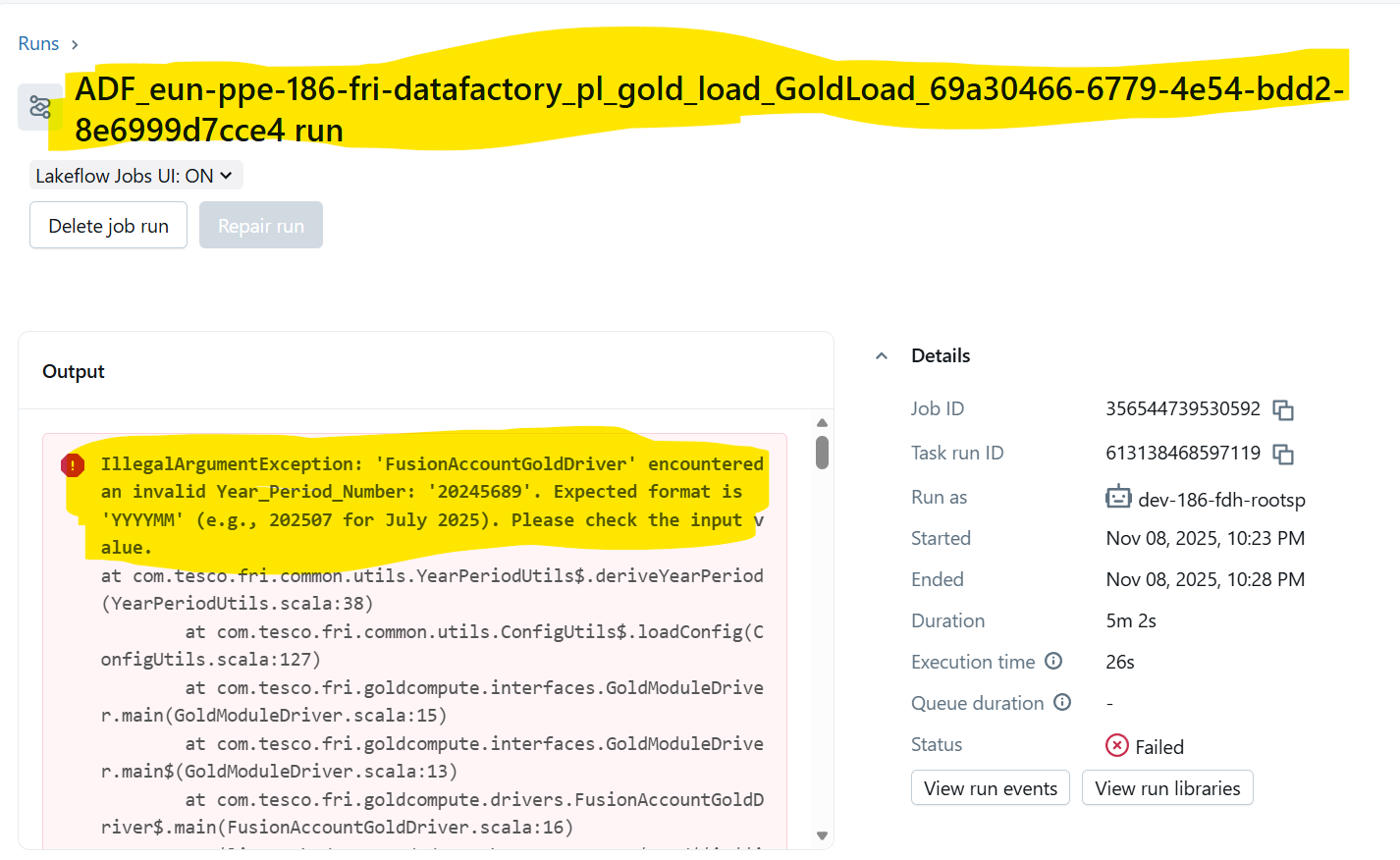
FROM delta\_gold.fusion\_gl\_balances



TEST CASE 49 : 49-Verify passing invalid YEAR\_PERIOD\_NUMBER in pipeline parameter and see how it



Output :



TEST CASE 53 : 53-Ensure all the tables are joined based on same year\_period\_number

Validate that all tables in the gold layer ETL process are joined using the same year\_period\_number to maintain period consistency.

QUERY :

SELECT g.year\_period\_number AS gold\_period,

s.year\_period\_number AS silver\_glbalances\_period,

m.year\_period\_number AS mapping\_period

FROM delta\_gold.fusion\_gl\_balances g

LEFT JOIN delta\_silver.fact\_fusion\_glbalances\_slv s

ON g.gl\_account = s.fusion\_coa\_account

LEFT JOIN delta\_silver.dim\_fdmee\_mapping\_slv m

ON g.gl\_account = m.source\_account

WHERE (s.year\_period\_number IS NOT NULL AND s.year\_period\_number <> g.year\_period\_number)

OR (m.year\_period\_number IS NOT NULL AND m.year\_period\_number <> g.year\_period\_number);

Output :

