Arturo-Quiroga-MSFT: \*\*Q: what mssqql mcp tools do you have access to\*\*

GitHub Copilot: I can use the following MSSQL MCP tools right now (these let me discover servers, connect, inspect schema, and run queries):

Discovery & connection

1. mssql\_list\_servers – List available SQL Server connections configured in this environment.

2. mssql\_connect – Open a connection to a specific server (optionally a database); returns a connection ID.

3. mssql\_get\_connection\_details – Show server/database/auth info for an existing connection.

4. mssql\_disconnect – Close a connection.

Database selection

5. mssql\_change\_database – Switch the active database for an existing connection.

Schema exploration

6. mssql\_list\_databases – List databases on the connected server.

7. mssql\_list\_schemas – List schemas in the current database.

8. mssql\_list\_tables – List tables (with schema) in the current database.

9. mssql\_list\_views – List views (with schema) in the current database.

10. mssql\_list\_functions – List user-defined functions.

11. mssql\_show\_schema – Open a visual/graphical schema view (schema designer) for exploration.

Query execution

12. mssql\_run\_query – Execute any T-SQL (SELECT / DML / DDL). (Powerful—must be careful with destructive commands.)

Typical safe workflow

- Enumerate servers: mssql\_list\_servers

- Connect: mssql\_connect

- List databases then switch if needed: mssql\_list\_databases -> mssql\_change\_database

- Explore structure: mssql\_list\_schemas / tables / views / functions

- (Optionally) visualize: mssql\_show\_schema

- Run read-only queries first: mssql\_run\_query (SELECT ...)

- Only then perform writes/DDL if explicitly requested

- Disconnect when done: mssql\_disconnect

Good practices I’ll follow

- I won’t run destructive (DROP/DELETE/UPDATE) statements unless you explicitly ask.

- For data-changing queries I can show you the statement first for confirmation.

- For large result sets I can LIMIT / TOP / WHERE-filter to keep output manageable.

- I can help craft parameterized or staged queries (preview with SELECT before UPDATE).

Let me know what you’d like to do next (e.g., list servers, connect, inspect a table, run one of the demo questions’ SQL, etc.) and I can proceed.

Arturo-Quiroga-MSFT: \*\*Q: list servers and DBs you have access to\*\*

GitHub Copilot: Listing available SQL Servers then connecting to each to enumerate its databases.

Listing server connections

Completed with input: {}

Connecting to both servers (same host, different DBs) to list their databases.

Connecting to server aqsqlserver001.database.windows.net and database adventureworksdb

Completed with input: {

"serverName": "aqsqlserver001.database.windows.net",

"database": "adventureworksdb"

}

Connecting to server aqsqlserver001.database.windows.net and database CONTOSO-FI

Completed with input: {

"serverName": "aqsqlserver001.database.windows.net",

"database": "CONTOSO-FI"

}

Listing databases for connection '805c7674-2d47-4589-a901-3bda8e4bc96a'

Completed with input: {

"connectionId": "805c7674-2d47-4589-a901-3bda8e4bc96a"

}

## Servers and Databases Accessible

I enumerated the configured SQL Server profiles and probed database lists where I could connect.

### Server profiles found

1. ADVENTUREWORKS → Host: aqsqlserver001.database.windows.net, Default DB: adventureworksdb (connection attempt failed)

2. CONTOSO-FI → Host: aqsqlserver001.database.windows.net, Default DB: CONTOSO-FI (connected successfully)

Both profiles point to the same Azure SQL logical server host.

### Successful connection

- Connected to: aqsqlserver001.database.windows.net (profile: CONTOSO-FI)

- Databases visible on that server:

- CONTOSO-FI

- master

- Multi\_Dimentional\_Modeling

### Failed connection

- adventureworksdb (profile ADVENTUREWORKS) returned: Failed to connect to server.

- Possible causes: database doesn’t exist, firewall / permissions, different credentials required, or case/name mismatch.

## Next Options

Let me know if you’d like to:

- Switch to another database (e.g., master) and list schemas/tables

- Retry or troubleshoot the AdventureWorks connection (I can test without specifying the DB, then change DB if successful)

- Run any of the complex analytical queries you listed

- Inspect table structures to help refine NL2SQL prompts

Tell me which direction to go and I’ll proceed.

Arturo-Quiroga-MSFT: \*\*Q: Inspect table structures to help refine NL2SQL prompts\*\*

GitHub Copilot: I'll enumerate all tables in the CONTOSO-FI database, then inspect their columns and types to provide a schema summary for NL2SQL prompt refinement.

Listing tables for connection '805c7674-2d47-4589-a901-3bda8e4bc96a'

Completed with input: {

"connectionId": "805c7674-2d47-4589-a901-3bda8e4bc96a"

}

Running query on connection '805c7674-2d47-4589-a901-3bda8e4bc96a'

Completed with input: {

"connectionId": "805c7674-2d47-4589-a901-3bda8e4bc96a",

"query": "SELECT COLUMN\_NAME, DATA\_TYPE, TABLE\_NAME FROM INFORMATION\_SCHEMA.COLUMNS WHERE TABLE\_NAME = 'Collateral'"

}

Running query on connection '805c7674-2d47-4589-a901-3bda8e4bc96a'

Completed with input: {

"connectionId": "805c7674-2d47-4589-a901-3bda8e4bc96a",

"query": "SELECT COLUMN\_NAME, DATA\_TYPE, TABLE\_NAME FROM INFORMATION\_SCHEMA.COLUMNS WHERE TABLE\_NAME = 'Company'"

}

Running query on connection '805c7674-2d47-4589-a901-3bda8e4bc96a'

Completed with input: {

"connectionId": "805c7674-2d47-4589-a901-3bda8e4bc96a",

"query": "SELECT COLUMN\_NAME, DATA\_TYPE, TABLE\_NAME FROM INFORMATION\_SCHEMA.COLUMNS WHERE TABLE\_NAME = 'Loan'"

}

Running query on connection '805c7674-2d47-4589-a901-3bda8e4bc96a'

Completed with input: {

"connectionId": "805c7674-2d47-4589-a901-3bda8e4bc96a",

"query": "SELECT COLUMN\_NAME, DATA\_TYPE, TABLE\_NAME FROM INFORMATION\_SCHEMA.COLUMNS WHERE TABLE\_NAME = 'PaymentEvent'"

}

Running query on connection '805c7674-2d47-4589-a901-3bda8e4bc96a'

Completed with input: {

"connectionId": "805c7674-2d47-4589-a901-3bda8e4bc96a",

"query": "SELECT COLUMN\_NAME, DATA\_TYPE, TABLE\_NAME FROM INFORMATION\_SCHEMA.COLUMNS WHERE TABLE\_NAME = 'RiskMetricHistory'"

}

I successfully retrieved the schema for the `Collateral` table, but encountered a technical issue fetching the others. Here’s the structure for `Collateral` (useful for NL2SQL prompt engineering):

\*\*dbo.Collateral\*\*

- CollateralId (bigint)

- LoanId (bigint)

- CollateralType (varchar)

- Description (nvarchar)

- Jurisdiction (nvarchar)

- ValueAmount (decimal)

- CurrencyCode (char)

- ValuationDate (date)

- Status (varchar)

Would you like a full schema for all tables (I can retry with a different approach), or focus on a subset (e.g., just `Loan`, `Company`, `PaymentEvent`, etc.)? Let me know your priority and I’ll proceed.

Arturo-Quiroga-MSFT: \*\*Q: Run one of the complex analytical queries I listed, you choose which one\*\*

GitHub Copilot: I'll run the query: "For each loan, what is the total value of collateral pledged, and which loans are under-collateralized?" This query exercises joins, aggregation, and a CASE statement—ideal for testing NL2SQL and schema understanding.

Proceeding to execute the query and show the results.

Running query on connection '805c7674-2d47-4589-a901-3bda8e4bc96a'

Completed with input: {

"connectionId": "805c7674-2d47-4589-a901-3bda8e4bc96a",

"query": "SELECT\n l.LoanId,\n l.PrincipalAmount,\n COALESCE(SUM(col.ValueAmount), 0) AS TotalCollateral,\n CASE\n WHEN COALESCE(SUM(col.ValueAmount), 0) < l.PrincipalAmount THEN 'Under-collateralized'\n ELSE 'Adequately collateralized'\n END AS CollateralStatus\nFROM\n dbo.Loan l\n LEFT JOIN dbo.Collateral col ON l.LoanId = col.LoanId\nGROUP BY\n l.LoanId, l.PrincipalAmount;"

}

The query failed due to a technical issue with the connection (invalid OwnerUri). This sometimes happens if the session expired or the connection context was lost.

I'll re-establish the connection to the CONTOSO-FI database and retry the query.

Connecting to server aqsqlserver001.database.windows.net and database CONTOSO-FI

Completed with input: {

"serverName": "aqsqlserver001.database.windows.net",

"database": "CONTOSO-FI"

}

Running query on connection '48b5a8c3-3575-4ec1-8603-ca792da5a3fc'

Completed with input: {

"connectionId": "48b5a8c3-3575-4ec1-8603-ca792da5a3fc",

"query": "SELECT\n l.LoanId,\n l.PrincipalAmount,\n COALESCE(SUM(col.ValueAmount), 0) AS TotalCollateral,\n CASE\n WHEN COALESCE(SUM(col.ValueAmount), 0) < l.PrincipalAmount THEN 'Under-collateralized'\n ELSE 'Adequately collateralized'\n END AS CollateralStatus\nFROM\n dbo.Loan l\n LEFT JOIN dbo.Collateral col ON l.LoanId = col.LoanId\nGROUP BY\n l.LoanId, l.PrincipalAmount;"

}

The query ran successfully. Here are sample results for "For each loan, what is the total value of collateral pledged, and which loans are under-collateralized?":

| LoanId | PrincipalAmount | TotalCollateral | CollateralStatus |

|--------|----------------|----------------|----------------------------|

| 1 | 10,000,000.00 | 13,000,000.00 | Adequately collateralized |

| 2 | 25,000,000.00 | 32,500,000.00 | Adequately collateralized |

| 3 | 25,000,000.00 | 45,000,000.00 | Adequately collateralized |

| ... | ... | ... | ... |

All loans in this sample are "Adequately collateralized" (TotalCollateral ≥ PrincipalAmount).

If you want to see more rows, filter for under-collateralized loans, or run another analytical query, let me know!

Arturo-Quiroga-MSFT: \*\*Q: show first 10 rows and also show me the SLQ query\*\*

GitHub Copilot: I'll re-run the query to fetch only the first 10 rows and display both the SQL and the results for you.

Here is the SQL query used:

```sql

SELECT

l.LoanId,

l.PrincipalAmount,

COALESCE(SUM(col.ValueAmount), 0) AS TotalCollateral,

CASE

WHEN COALESCE(SUM(col.ValueAmount), 0) < l.PrincipalAmount THEN 'Under-collateralized'

ELSE 'Adequately collateralized'

END AS CollateralStatus

FROM

dbo.Loan l

LEFT JOIN dbo.Collateral col ON l.LoanId = col.LoanId

GROUP BY

l.LoanId, l.PrincipalAmount

ORDER BY

l.LoanId

OFFSET 0 ROWS FETCH NEXT 10 ROWS ONLY;

```

Now running this query to show the first 10 rows.

Running query on connection '48b5a8c3-3575-4ec1-8603-ca792da5a3fc'

Completed with input: {

"connectionId": "48b5a8c3-3575-4ec1-8603-ca792da5a3fc",

"query": "SELECT\n l.LoanId,\n l.PrincipalAmount,\n COALESCE(SUM(col.ValueAmount), 0) AS TotalCollateral,\n CASE\n WHEN COALESCE(SUM(col.ValueAmount), 0) < l.PrincipalAmount THEN 'Under-collateralized'\n ELSE 'Adequately collateralized'\n END AS CollateralStatus\nFROM\n dbo.Loan l\n LEFT JOIN dbo.Collateral col ON l.LoanId = col.LoanId\nGROUP BY\n l.LoanId, l.PrincipalAmount\nORDER BY\n l.LoanId\nOFFSET 0 ROWS FETCH NEXT 10 ROWS ONLY;"

}

Here is the SQL query used to fetch the first 10 rows:

```sql

SELECT

l.LoanId,

l.PrincipalAmount,

COALESCE(SUM(col.ValueAmount), 0) AS TotalCollateral,

CASE

WHEN COALESCE(SUM(col.ValueAmount), 0) < l.PrincipalAmount THEN 'Under-collateralized'

ELSE 'Adequately collateralized'

END AS CollateralStatus

FROM

dbo.Loan l

LEFT JOIN dbo.Collateral col ON l.LoanId = col.LoanId

GROUP BY

l.LoanId, l.PrincipalAmount

ORDER BY

l.LoanId

OFFSET 0 ROWS FETCH NEXT 10 ROWS ONLY;

```

And here are the first 10 results:

| LoanId | PrincipalAmount | TotalCollateral | CollateralStatus |

|--------|----------------|----------------|----------------------------|

| 1 | 10,000,000.00 | 13,000,000.00 | Adequately collateralized |

| 2 | 25,000,000.00 | 32,500,000.00 | Adequately collateralized |

| 3 | 25,000,000.00 | 45,000,000.00 | Adequately collateralized |

| 4 | 18,000,000.00 | 23,400,000.00 | Adequately collateralized |

| 5 | 18,000,000.00 | 23,400,000.00 | Adequately collateralized |

| 6 | 18,000,000.00 | 32,400,000.00 | Adequately collateralized |

| 7 | 18,000,000.00 | 23,400,000.00 | Adequately collateralized |

| 8 | 22,000,000.00 | 25,300,000.00 | Adequately collateralized |

| 9 | 22,000,000.00 | 36,300,000.00 | Adequately collateralized |

| 10 | 22,000,000.00 | 25,300,000.00 | Adequately collateralized |

Let me know if you want to see more rows, filter for under-collateralized loans, or run a different query!

Arturo-Quiroga-MSFT: \*\*Q: modify the pertinent table or tables needed to have more diverse collateral status\*\*

GitHub Copilot: Querying collateral row details for candidate loans to design targeted updates creating under-collateralized cases.

Running query on connection '48b5a8c3-3575-4ec1-8603-ca792da5a3fc'

Completed with input: {

"connectionId": "48b5a8c3-3575-4ec1-8603-ca792da5a3fc",

"query": "SELECT CollateralId, LoanId, CollateralType, ValueAmount FROM dbo.Collateral WHERE LoanId BETWEEN 1 AND 5 ORDER BY LoanId, CollateralId;"

}

Running query on connection '48b5a8c3-3575-4ec1-8603-ca792da5a3fc'

Completed with input: {

"connectionId": "48b5a8c3-3575-4ec1-8603-ca792da5a3fc",

"query": "SELECT LoanId, PrincipalAmount FROM dbo.Loan WHERE LoanId BETWEEN 1 AND 5 ORDER BY LoanId;"

}

Proposing targeted UPDATE/INSERT statements to create under-collateralized loans; will present for confirmation before executing.

You asked to modify the data so collateral status becomes more diverse (i.e., introduce under-collateralized loans). Currently all sampled loans (1–5) are over or fully covered. Easiest, least invasive way: reduce some collateral values or add a new loan with intentionally low collateral. I'll propose specific reversible changes (and only execute after you confirm).

## Current Snapshot (Loans 1–5)

Principal vs existing pledged collateral (aggregate):

- Loan 1: Principal 10,000,000; Collateral 13,000,000

- Loan 2: 25,000,000; Collateral 32,500,000

- Loan 3: 25,000,000; Collateral 45,000,000 (two rows: 32.5M + 12.5M)

- Loan 4: 18,000,000; Collateral 23,400,000

- Loan 5: 18,000,000; Collateral 23,400,000

## Goal

Create a representative distribution:

- Some adequately collateralized (leave as-is: 1,2)

- Some slightly under-collateralized (e.g., 4)

- One severely under-collateralized (e.g., 3)

- One exactly matched (adjust 5)

- Optionally add a new loan (e.g., 16) with zero collateral to test null/sparse edge case.

## Proposed Data Changes

(Assuming no FK/blocking constraints and ValueAmount updates are allowed.)

1. Make Loan 3 under-collateralized:

- Reduce CollateralId 3 (Receivables) from 32,500,000.00 to 8,000,000.00

- Reduce CollateralId 16 (Guarantee) from 12,500,000.00 to 2,000,000.00

- New total: 10,000,000 vs principal 25,000,000 → Under-collateralized

2. Make Loan 4 slightly under-collateralized:

- Adjust CollateralId 4 from 23,400,000.00 to 15,000,000.00

- New total: 15,000,000 vs principal 18,000,000

3. Make Loan 5 exactly matched:

- Adjust CollateralId 5 from 23,400,000.00 to 18,000,000.00

- New total: 18,000,000 vs principal 18,000,000

4. (Optional) Add a new Loan 16 with low/no collateral for testing:

- INSERT INTO dbo.Loan (...) VALUES (...)

- Omit collateral initially so it shows 0 pledged.

## SQL (Staged — won’t run until you confirm)

Updates:

```sql

-- Loan 3 -> under-collateralized

UPDATE dbo.Collateral SET ValueAmount = 8000000.00 WHERE CollateralId = 3; -- was 32,500,000

UPDATE dbo.Collateral SET ValueAmount = 2000000.00 WHERE CollateralId = 16; -- was 12,500,000

-- Loan 4 -> slightly under-collateralized

UPDATE dbo.Collateral SET ValueAmount = 15000000.00 WHERE CollateralId = 4; -- was 23,400,000

-- Loan 5 -> exactly matched

UPDATE dbo.Collateral SET ValueAmount = 18000000.00 WHERE CollateralId = 5; -- was 23,400,000

```

Optional new loan (example skeleton—you’d need correct required columns; I can query schema first if you want):

```sql

-- Example only; requires actual column list:

-- INSERT INTO dbo.Loan (CompanyId, LoanNumber, PrincipalAmount, OriginationDate, Status, CurrencyCode, InterestRate, MaturityDate)

-- VALUES (/\*CompanyId\*/ 1, 'LN-TEST-016', 12000000.00, GETDATE(), 'Active', 'USD', 0.0650, DATEADD(year, 5, GETDATE()));

```

## Reversibility

I can also record original values to a temp table for rollback:

```sql

SELECT CollateralId, LoanId, ValueAmount

INTO #CollateralBackup\_ChangeSet

FROM dbo.Collateral

WHERE CollateralId IN (3,16,4,5);

```

## Next Step

Reply with one of:

- APPROVE UPDATES (no new loan)

- APPROVE UPDATES + ADD NEW LOAN

- MODIFY (and describe changes)

- CANCEL

Once you approve, I’ll execute safely and then re-run the collateral status query to show the new distribution.

How would you like to proceed?

Arturo-Quiroga-MSFT: \*\*Q: Run any of the complex analytical queries you listed\*\*

GitHub Copilot:

Arturo-Quiroga-MSFT: \*\*Q: APPROVE UPDATES + ADD NEW LOAN\*\*

GitHub Copilot: Fetching Loan table schema and a sample row to determine required columns for insert.

Running query on connection '48b5a8c3-3575-4ec1-8603-ca792da5a3fc'

Completed with input: {

"connectionId": "48b5a8c3-3575-4ec1-8603-ca792da5a3fc",

"query": "SELECT COLUMN\_NAME, IS\_NULLABLE, DATA\_TYPE FROM INFORMATION\_SCHEMA.COLUMNS WHERE TABLE\_NAME='Loan' ORDER BY ORDINAL\_POSITION;"

}

Running query on connection '48b5a8c3-3575-4ec1-8603-ca792da5a3fc'

Completed with input: {

"connectionId": "48b5a8c3-3575-4ec1-8603-ca792da5a3fc",

"query": "SELECT TOP 1 \* FROM dbo.Loan ORDER BY LoanId DESC;"

}

Running query on connection '48b5a8c3-3575-4ec1-8603-ca792da5a3fc'

Completed with input: {

"connectionId": "48b5a8c3-3575-4ec1-8603-ca792da5a3fc",

"query": "SELECT TOP 1 CompanyId FROM dbo.Company ORDER BY CompanyId;"

}

Running query on connection '48b5a8c3-3575-4ec1-8603-ca792da5a3fc'

Completed with input: {

"connectionId": "48b5a8c3-3575-4ec1-8603-ca792da5a3fc",

"query": "SELECT CollateralId, LoanId, ValueAmount FROM dbo.Collateral WHERE CollateralId IN (3,16,4,5);"

}

Executing transactional updates to collateral values and inserting a new under-collateralized loan with no collateral.

Running query on connection '48b5a8c3-3575-4ec1-8603-ca792da5a3fc'

Completed with input: {

"connectionId": "48b5a8c3-3575-4ec1-8603-ca792da5a3fc",

"query": "BEGIN TRANSACTION;\n\n-- Backup original collateral rows\nIF OBJECT\_ID('tempdb..#CollateralBackup\_ChangeSet') IS NOT NULL DROP TABLE #CollateralBackup\_ChangeSet;\nSELECT CollateralId, LoanId, ValueAmount INTO #CollateralBackup\_ChangeSet\nFROM dbo.Collateral WHERE CollateralId IN (3,16,4,5);\n\n-- Apply updates to diversify collateral coverage\nUPDATE dbo.Collateral SET ValueAmount = 8000000.00 WHERE CollateralId = 3; -- was 32,500,000 (Loan 3)\nUPDATE dbo.Collateral SET ValueAmount = 2000000.00 WHERE CollateralId = 16; -- was 12,500,000 (Loan 3)\nUPDATE dbo.Collateral SET ValueAmount = 15000000.00 WHERE CollateralId = 4; -- was 23,400,000 (Loan 4)\nUPDATE dbo.Collateral SET ValueAmount = 18000000.00 WHERE CollateralId = 5; -- was 23,400,000 (Loan 5)\n\n-- Insert new loan (Loan 16 expected next identity)\nINSERT INTO dbo.Loan (CompanyId, LoanNumber, OriginationDate, MaturityDate, PrincipalAmount, CurrencyCode, InterestRatePct, InterestRateType, ReferenceRate, SpreadBps, AmortizationType, PaymentFreqMonths, Purpose, Status, ReferenceRateCode)\nVALUES (1, 'LN-TEST-016', CONVERT(date, GETDATE()), DATEADD(year, 5, CONVERT(date, GETDATE())), 14000000.00, 'USD', 6.750, 'Fixed', NULL, NULL, 'Amortizing', 3, 'Test Under-Collateralized Scenario', 'Active', NULL);\n\nCOMMIT TRANSACTION;"

}

Running query on connection '48b5a8c3-3575-4ec1-8603-ca792da5a3fc'

Completed with input: {

"connectionId": "48b5a8c3-3575-4ec1-8603-ca792da5a3fc",

"query": "SELECT CollateralId, LoanId, ValueAmount FROM dbo.Collateral WHERE CollateralId IN (3,16,4,5) ORDER BY CollateralId;"

}

Running query on connection '48b5a8c3-3575-4ec1-8603-ca792da5a3fc'

Completed with input: {

"connectionId": "48b5a8c3-3575-4ec1-8603-ca792da5a3fc",

"query": "SELECT LoanId, PrincipalAmount FROM dbo.Loan WHERE LoanId IN (3,4,5,16) ORDER BY LoanId;"

}

Running query on connection '48b5a8c3-3575-4ec1-8603-ca792da5a3fc'

Completed with input: {

"connectionId": "48b5a8c3-3575-4ec1-8603-ca792da5a3fc",

"query": "SELECT l.LoanId, l.PrincipalAmount, COALESCE(SUM(col.ValueAmount),0) AS TotalCollateral,\nCASE WHEN COALESCE(SUM(col.ValueAmount),0) < l.PrincipalAmount THEN 'Under-collateralized'\n WHEN COALESCE(SUM(col.ValueAmount),0) = l.PrincipalAmount THEN 'Exactly matched'\n ELSE 'Adequately collateralized' END AS CollateralStatus\nFROM dbo.Loan l\nLEFT JOIN dbo.Collateral col ON l.LoanId = col.LoanId\nWHERE l.LoanId IN (3,4,5,16)\nGROUP BY l.LoanId, l.PrincipalAmount\nORDER BY l.LoanId;"

}

## Collateral Diversification Complete ✅

The approved updates and new loan insertion have been executed.

### Changes Applied

1. Loan 3 collateral reduced to create a severe shortfall:

- CollateralId 3 → 8,000,000.00 (was 32,500,000.00)

- CollateralId 16 → 2,000,000.00 (was 12,500,000.00)

- Total now 10,000,000 vs principal 25,000,000 → Under-collateralized

2. Loan 4 collateral trimmed:

- CollateralId 4 → 15,000,000.00 (was 23,400,000.00) → Under-collateralized

3. Loan 5 collateral adjusted to match:

- CollateralId 5 → 18,000,000.00 (was 23,400,000.00) → Exactly matched

4. New Loan 16 inserted with zero collateral (initially):

- Principal 14,000,000.00 → Under-collateralized (0 pledged)

A backup of original collateral values for the modified rows was captured inside the transaction (temporary table for that session). If you’d like a persistent audit/rollback table, I can create one.

### Verification Extract

| LoanId | PrincipalAmount | TotalCollateral | CollateralStatus |

|--------|-----------------|-----------------|-----------------------|

| 3 | 25,000,000.00 | 10,000,000.00 | Under-collateralized |

| 4 | 18,000,000.00 | 15,000,000.00 | Under-collateralized |

| 5 | 18,000,000.00 | 18,000,000.00 | Exactly matched |

| 16 | 14,000,000.00 | 0.00 | Under-collateralized |

### Next Recommended Queries (optional)

- Re-run the broader collateral coverage distribution:

- Count loans by status

- Identify newly under-collateralized loans:

- Filter CollateralStatus = 'Under-collateralized'

- Add staged collateral to Loan 16 later (to simulate improvement over time)

Let me know if you want:

- A rollback script for the four modified collateral rows

- To run any of the other complex analytical queries now (e.g., covenant breaches, missed payments, risk metric trend)

- To introduce time-series changes (e.g., new `ValuationDate` entries) for scenario testing

Just say the word and I’ll continue. What would you like next?