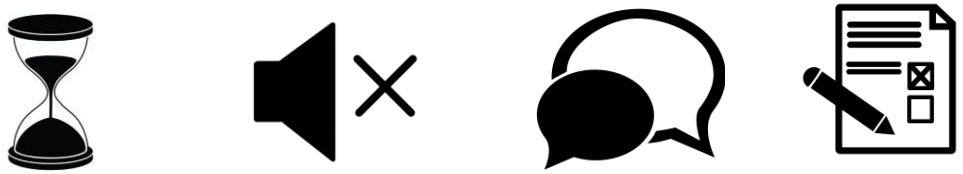
17 Mar, 2026

# Data Lineage in CDGC: A Practical Guide to Key Categories and Setup Approaches – Part 1


- Puneet Dudeja, Senior Solutions Architect, CSA
- Vivek Singh, Senior Solutions Architect, CSA
- Avanish Srivastava, Senior Manager, CSA


Where data & AI come to LIFE

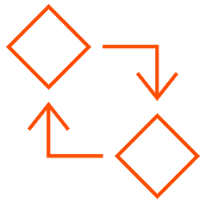
# Housekeeping Tips



- Today's Webinar is scheduled for 1 hour
- The session will include a webcast and then your questions will be answered live at the end of the presentation
- All dial-in participants will be muted to enable the speakers to present without interruption
- Questions can be submitted to "All Panelists" via the Q&A option and we will respond at the end of the presentation
- The webinar is being recorded and will be available on our Success Portal - where you can download the slide deck for the presentation. The link to the recording will be emailed as well.
- Please take time to complete the post-webinar survey and provide your feedback and suggestions for upcoming topics.

Bootstrap trial and POC Customers

Enriched Customer Onboarding experience



Product Learning Paths and Weekly Expert Sessions

Informatica Concierge

Tailored training and content recommendations

# More Information

## Success Portal

https://success.informatica.com

## Communities & Support

https://network.informatica.com

## Documentation

https://docs.informatica.com

## University

https://www.informatica.com/in/services-and-training/informatica-university.html

# Safe Harbor

Disclaimer: The information being provided herein is for informational purposes only. The development, release and timing of any Informatica product, service or functionality described herein remain at the sole discretion of Informatica and should not be relied upon in making a purchasing decision. Statements made herein are based on information currently available, which is subject to change. Such statements should not be relied upon as a representation, warranty or commitment to deliver specific products, services or functionality in the future.

# Agenda

1. Introduction
2. What is Data Lineage ?
3. Types of Data Lineage supported in CDGC
4. Automated Technical Lineage and how it is created in CDGC
5. Reference Objects and Connection Assignment
6. Download assignment report and analysis
7. Connection Aware Scan
8. Use Case for Technical Data Lineage
9. Live demo to show above use case in action
10. Q&A

# What is Data Lineage

**Data Lineage** visually maps data's journey from source to destination, providing essential context and traceability that unlocks data's trustworthiness and usability. This transparency is critical for high-monetization, agent-driven tech enterprises, enabling reliable, **context-rich data foundations** that power AI modernization and intelligent agents along with ensuring regulatory compliances and other key data insights.

## Why Data Lineage Matters

- ✅ **Impact Analysis**: Understand what breaks if you change a table, column, or process
- ✅ **Data Quality & Observability**: Trace data quality issues back to their source
- ✅ **Compliance & Audit**: Demonstrate data provenance for regulations (GDPR, CCPA, SOX)
- ✅ **Data Governance**: Enforce policies and understand data ownership
- ✅ **Migration & Modernization:** Map dependencies before cloud/platform migrations
- ✅ **AI Models & Agents** : Ensures AI agents and models have trusted, traceable data for reliable decisions.

# Types of Data Lineage supported in CDGC

## Technical Lineage

| Lineage Type | Source/Trigger | Level | Description | Automation Method |
|---|---|---|---|---|
| Technical Automated Lineage | ETL Scanners | Dataset/Data Element | Table-to-table and column-to-column flows from scanned metadata (CDI/IICS, PowerCenter, RDBMS) | Automatic via scan + connection assignment |
| | BI Scanners | Dataset/Data Element | Table to BI Dataset and Column to BI Field from scanned BI source metadata (Tableau, Power BI, MicroStrategy) | Automatic via scan + connection assignment |
| | DB Scripts / Stored Procedures | Dataset/Data Element | Table-to-table and column-to-column flows from scanned metadata (Stored procedures, script scanners) | Automatic via scan + connection assignment |
| | AI/ML Models Scanner | Dataset | Table-Model (Databricks, Google Vertex AI) | Automatic via scan + connection assignment |
| Inferred/Linked Lineage | CLAIRE AI or Rule-based linking | Dataset/Data Element | AI or rule-based matching across catalog sources; can auto-accept based on confidence thresholds | Automatic via Link Catalog Sources jobs |

# Types of Data Lineage supported in CDGC

| Lineage Type | Source/Trigger | Level | Description | Automation Method |
|---|---|---|---|---|
| External Lineage Ingestion | Databricks Unity Catalog, Apache Atlas, Microsoft Purview | Dataset/Data Element | Imports upstream/downstream lineage from external catalogs when "Lineage from Unity" is enabled | Automatic via scanner configuration |
| Custom Lineage | Source and target Datasets | Dataset/Data Element | Fill gaps where automated lineage can't reach | User curated |

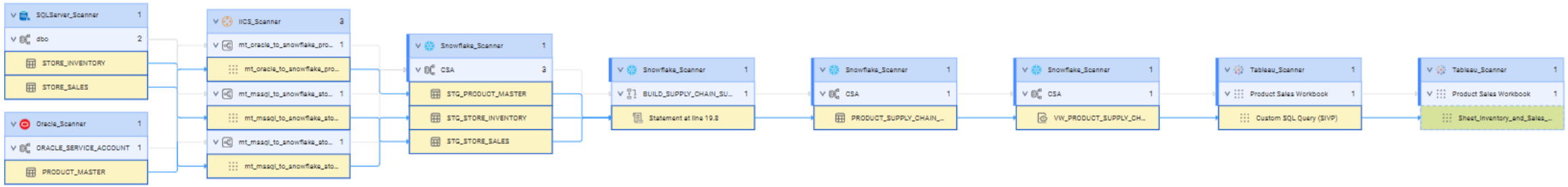
## Business Lineage

| Lineage Type | Source/Trigger | Level | Description | Automation Method |
|---|---|---|---|---|
| Business Lineage | Business Datasets | Business Dataset/Data Element | System-to-System and dataset-to-dataset providing a high-level view for business stakeholders | Semi-automatic (requires user curation) |

# What is Technical Automated Lineage in CDGC ?

**Technical Automated Lineage** in CDGC is the visual representation of data flow across systems, automatically generated from scanned metadata. It shows how data moves from sources through transformations to targets at three levels of granularity:

- ✅ **Catalog Source Level**: Database/file system connections
- ✅ **Data Set Level**: Tables, files, views, mapping tasks
- ✅ **Data Element Level**: Columns, fields, attributes



# Workflow Diagram – Technical Lineage Creation in CDGC

Administrator (Connections)

↓

Test & Save

↓

MCC (Catalog Sources) ← Configure IICS/IDMC Metadata

↓

Run Scans (Sources, Targets, ETL)

↓

Monitor Scan Status

↓

MCC (Connection Assignment)

↓

Assign ETL Connections → Actual Connections

↓

Export Assignment Report (Validate Matches)

↓

CDGC (Lineage Tab)

↓

View Technical Lineage (Catalog/Dataset/Element)

# How Technical Lineage is created in CDGC ?

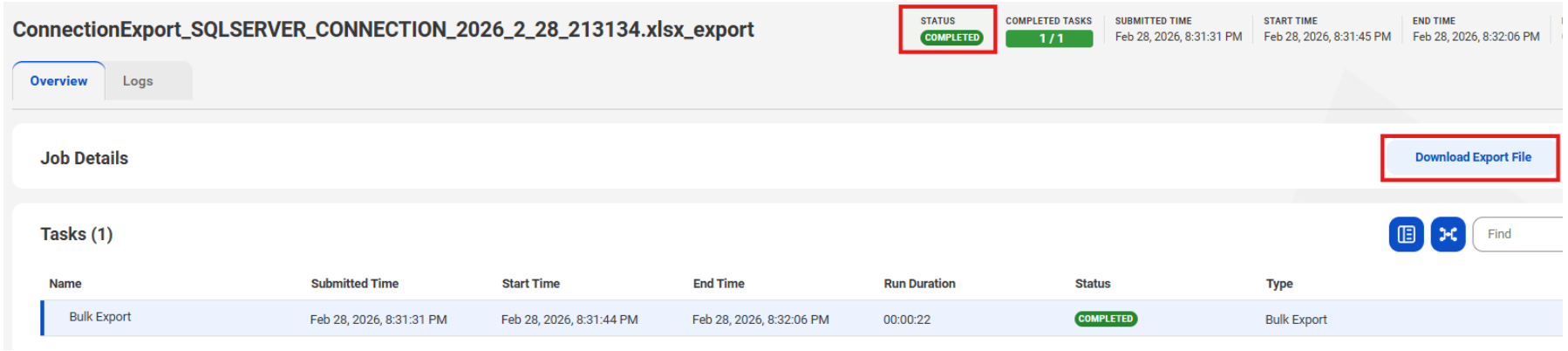
## Pre-requisites

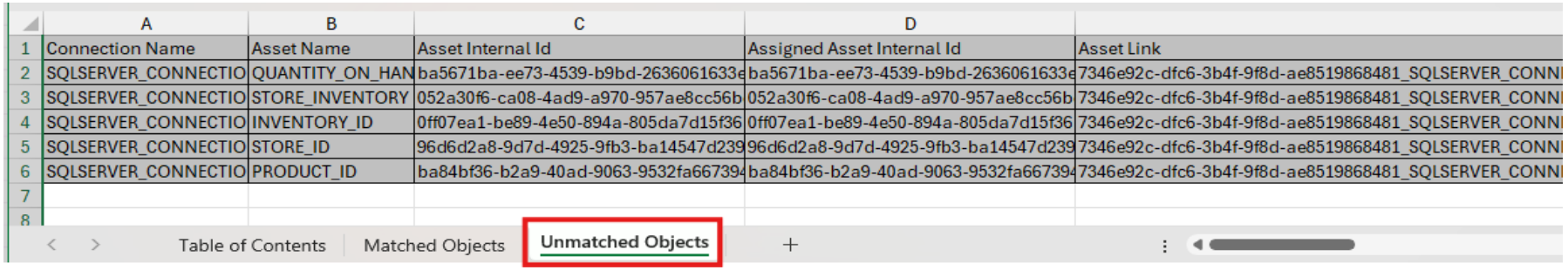
✅ **Secure Agent** with required connectors enabled (Administrator > Runtime Environments > Enable/Disable Services, Connectors)
✅ **Network access** to source/target systems (verify ports, TLS versions, firewall rules)
✅ **Appropriate privileges** on source/target systems for metadata extraction
✅ **CDGC and MCC** services enabled in your IDMC org

## Step-by-Step Process

✅ Create **Connections** in Administrator (Administrator > Connections > New)
✅ Setup **scanner configuration** in Metadata Command Center (MCC > Catalog Sources > Create) to execute metadata scan
✅ Make sure to have at least **1 data source, target and ETL system** scanned to generate end to end technical lineage.
✅ Perform **Connection Assignment** (MCC > Configure > Lineage > Assign Connections)

- [x] Monitor Assignment Status (MCC > Monitor > Connection Assignment)
- [x] **Export Assignment Report** to review matched/unmatched objects



| | A | B | C | D | |
|---|---|---|---|---|---|
| 1 | Connection Name | Asset Name | Asset Internal Id | Assigned Asset Internal Id | Asset Link |
| 2 | SQLSERVER_CONNECTIO | QUANTITY_ON_HAN | ba5671ba-ee73-4539-b9bd-2636061633e | ba5671ba-ee73-4539-b9bd-2636061633e | 7346e92c-dfc6-3b4f-9f8d-ae8519868481_SQLSERVER_CONNI |
| 3 | SQLSERVER_CONNECTIO | STORE_INVENTORY | 052a30f6-ca08-4ad9-a970-957ae8cc56b | 052a30f6-ca08-4ad9-a970-957ae8cc56b | 7346e92c-dfc6-3b4f-9f8d-ae8519868481_SQLSERVER_CONNI |
| 4 | SQLSERVER_CONNECTIO | INVENTORY_ID | 0ff07ea1-be89-4e50-894a-805da7d15f36 | 0ff07ea1-be89-4e50-894a-805da7d15f36 | 7346e92c-dfc6-3b4f-9f8d-ae8519868481_SQLSERVER_CONNI |
| 5 | SQLSERVER_CONNECTIO | STORE_ID | 96d6d2a8-9d7d-4925-9fb3-ba14547d239 | 96d6d2a8-9d7d-4925-9fb3-ba14547d239 | 7346e92c-dfc6-3b4f-9f8d-ae8519868481_SQLSERVER_CONNI |
| 6 | SQLSERVER_CONNECTIO | PRODUCT_ID | ba84bf36-b2a9-40ad-9063-9532fa66739 | ba84bf36-b2a9-40ad-9063-9532fa66739 | 7346e92c-dfc6-3b4f-9f8d-ae8519868481_SQLSERVER_CONNI |
| 7 | | | | | |
| 8 | | | | | |



- [x] Run scan in connection-aware mode (if needed)

# What are IDMC Auto Catalog and IICS Scanner

## IDMC Auto Catalog (IDMC Metadata)

✅ Near-real-time synchronization of **Cloud Data Integration (CDI)** and **Application Integration (CAI)** design-time assets and run-time lineage into CDGC
✅ No scanner required—continuous sync via **Bulk Sync** (bootstrap) and **Realtime Sync** jobs
✅ Automatically resolves connections and lineage as sources/targets are scanned

## IICS Scanner (Catalog Source)

✅ Traditional agent-based, scheduled scanner that extracts IICS assets by running a configured catalog source in MCC
✅ Batch-oriented with on-demand or scheduled scans
✅ Provides detailed scanner logs and explicit lifecycle controls
✅ Supports MDM SaaS business objects, Ref 360 and advanced mappings

# Key Differences b/w IDMC Auto Catalog vs IICS Scanner

| Aspect | IDMC Auto Catalog | IICS Scanner |
|---|---|---|
| Sync Mode | Near-real-time, continuous | Batch, scheduled/on-demand |
| Topology | Same IDMC org only (CDGC + CDI) | Cross-organization supported |
| Delete Mode | RETAIN only; no delete mode | DELETE mode supported |
| Lineage | Appears after task runs post-enablement | Extracts latest mapping instance XML |
| Filters | Include-only (no exclude) | Include/exclude with wildcards |
| Setup | Enable in MCC + configure DIS properties | Create catalog source + schedule scans |
| IPU Consumption | More due to near real-time | On-demand basis |
| Logs | Limited visibility | Detailed scanner logs for troubleshooting |
| Assets Extracted | CDI, CAI - Design Time Only, CDQ – For CDQ bulk import templates | CDI, MDM Ref360, MDM SaaS, Advanced Mappings |

# Reference Objects in CDGC

**Reference objects** (also called **reference assets**) in CDGC are **placeholder technical assets** automatically created during catalog scans to represent upstream or downstream objects from other catalog sources. They enable CDGC to maintain **end-to-end technical lineage** even when the actual endpoint assets haven't been scanned yet or connections haven't been assigned.

## Key Purposes

✅ **Cross-source lineage continuity** — When lineage spans multiple catalog sources (e.g., a Snowflake view in one source references a base table in another), reference objects act as placeholders until connection assignment resolves them to actual scanned assets.

✅ **ETL-to-database lineage** — For IICS, PowerCenter, or other ETL scans, reference datasets and elements are created for database tables/columns before you assign connections to the actual database endpoints.

✅ **Unresolved dependencies** — When scans encounter parameterized jobs, cross-schema references, or objects outside current scan filters, reference objects preserve the lineage structure.

## Types of Reference Objects

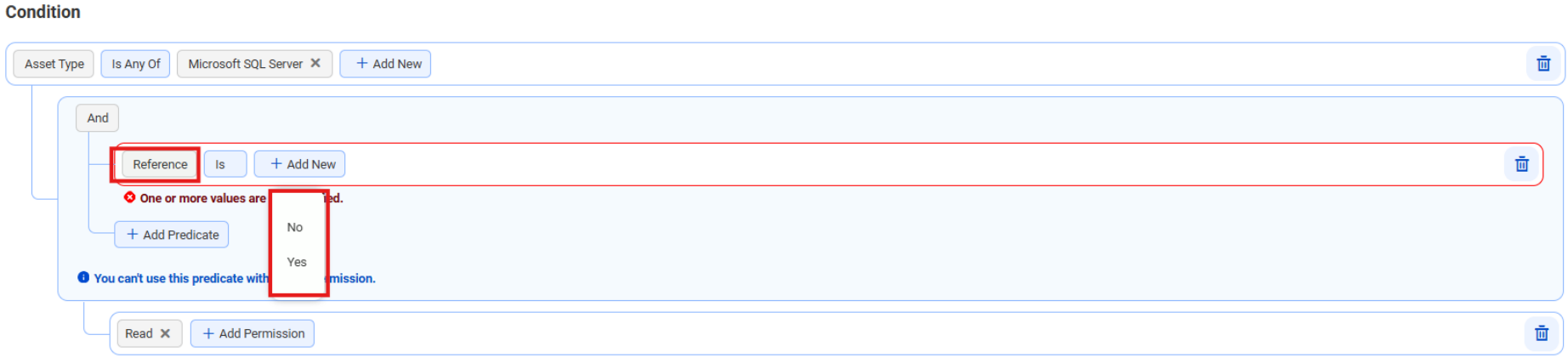
✅ **Reference Data Sets** (tables, views, files)
✅ **Reference Data Elements** (columns, fields)
✅ **Reference Data Sources** (databases, schemas)

## How to Resolve Reference Objects

✅ **Scan both endpoints** — Ensure both the source (e.g., ETL job) and target (e.g., database tables) are scanned with appropriate filters
✅ **Connection Assignment** — In MCC, navigate to **Configure > Lineage** or **Monitor > Connection Assignment**
✅ **Assign connections** — Map the reference objects to the correct catalog source containing the actual assets
✅ **Run and verify** — Aim for 100% match rate; if references persist, verify filters and consider re-running in Delete mode to purge stale references

# Important Notes

✅ **Consumption impact** — Reference assets count toward IPU consumption like regular technical assets
✅ **Visibility** — Use PBAC rules to control which users can see reference objects (filter on Reference = Yes attribute)




✅ **Resolution is required** — Reference objects remain until you perform connection assignment/Lineage Discovery to link them to actual scanned assets

# Connection Assignment

**Connection assignment** is the process of linking **reference objects** (placeholder assets) extracted from ETL/reporting tools to **actual scanned objects** in endpoint catalog sources, enabling CDGC to render **end-to-end technical lineage** with real tables/files instead of placeholders.

## Why Is It Needed?

When CDGC scans ETL/reporting tools (IICS, PowerCenter, Power BI, etc.), it extracts metadata about **connections** and **referenced objects** (tables, files, schemas) that the ETL jobs use. However, these are initially created as **reference datasets**—placeholders that don't yet link to the actual scanned assets in your database or file system catalog sources.

## Key Properties in Assignment

- ✅ **Connection Name** — Derived from the ETL tool (e.g., IICS connection name, file path with underscores)
- ✅ **Parent Catalog Source** — The ETL/reporting source that generated the reference connection
- ✅ **Assigned Catalog Source** — The endpoint catalog source containing the actual objects
- ✅ **Endpoint** — The specific schema/database/file path to bind to
- ✅ **Objects (Matched/Unmatched %)** — Shows how many reference objects successfully resolved

# Connection Assignment Approaches

| Aspect | Manual Connection Assignment | Lineage Discovery |
|---|---|---|
| Approach | Explicit, one-by-one assignment | AI-assisted recommendations with curation |
| Automation | Fully manual | CLAIRE AI suggests matches; you accept/reject |
| Scope | Connection-level (granular) | Catalog-source level with optional connection drill-down |
| Repeatability | One-time, auto-apply in future runs | Accepted pairs auto-apply in future runs |
| Conflict Handling | Manual detection and resolution | UI warns about duplicates/splits; guided resolution |
| IPU Cost Control | N/A (no discovery job) | Filters (type/name/asset group) reduce search space |
| UI Location | Configure → Lineage → Lineage Discovery → **Assign Connections** | Configure → Lineage → **Lineage Discovery** |
| When to Use | Precise control, overrides, edge cases, no discovery support | Scale, ongoing curation |

# Connection Assignment Approaches

## Lineage Discovery (CLAIRE Recommendations)

Steps:
1.) Enable **Lineage Discovery** in the catalog source configuration
2.) Define filters (catalog source type/name patterns) to narrow scope and reduce IPU cost
3.) Trigger the Discovery job (manual or scheduled)
4.) **Curate recommendations** — Accept/reject at catalog-source or connection level
5.) Accepted pairs are auto-applied for future runs
6.) Resolve conflicts (duplicates, split metadata) via curation UI

## Manual Connection Assignment (MCC UI)

Steps:
1.) Navigate to **MCC → Configure → Lineage → Lineage Discovery → Assign Connections**
2.) Select the connection row → click **Assign**
3.) Choose the **Assigned Catalog Source** and correct **Endpoint** (schema/database/file path)
4.) Monitor **Job Status** and **Objects** matched/unmatched %
5.) Export matched/unmatched lists for remediation if needed
6.) Verify lineage in CDGC shows actual datasets

# Connection Assignment – Selecting Approach

Do you have many connections across multiple catalog sources?
- YES → Use Lineage Discovery (scale + repeatability)
- NO → Do you need precise control or overrides?
  - YES → Use Manual Assignment
  - NO → Does the catalog source support Lineage Discovery?
    - YES → Use Lineage Discovery (AI-assisted)
    - NO → Use Manual Assignment (only option)

# Connection Assignment Report in CDGC

The **Connection Assignment Report** is an exportable Excel file that provides detailed visibility into which assets were **matched** and **unmatched** during a connection assignment operation. It's a critical troubleshooting and validation tool for ensuring end-to-end lineage resolution.

## What It Contains

| | |
|---|---|
| ✅ **Connection Name** | Name of the reference connection from the ETL/reporting tool |
| ✅ **Asset Name/ID** | Name and unique ID of the reference or actual asset |
| ✅ **Reference Link** | Hyperlink back to the asset in CDGC for quick navigation |
| ✅ **Location/Hierarchy** | Original location of the reference object |
| ✅ **Assigned Location/Hierarchy** | Target endpoint location (schema/database/file path) |
| ✅ **Asset Match Key** | Key used to match reference objects to actual assets |
| ✅ **Catalog Source Name/Type** | Parent catalog source that generated the reference connection |
| ✅ **Assigned Catalog Source Name/Type** | Endpoint catalog source containing actual assets |
| ✅ **Assigned Status** | **Matched** or **Unmatched** |

# How to Generate Assignment Report

## Pre-requisites

✅ **Connection assignment completed** — The report is available only after you've assigned a connection
✅ **Endpoint sources scanned** — Actual assets must exist in endpoint catalog sources (databases, file systems)
✅ **Permissions** — **View/Manage Lineage Settings** privilege required to download from Monitor

## Steps

1. **Navigate**: MCC → **Configure** → **Lineage** → **Lineage Discovery** → **Assign Connections**
2. **Select connection**: Choose the connection you've already assigned
3. **Export**: Click **Export** or **Download Assignment Report** (may be labeled "Export matched/unmatched objects")
4. **Monitor job**: Go to **Monitor** to track the export job
5. **Download**: Once the job completes, download the generated .xlsx file from Monitor
6. **Review**: Open the Excel file and analyze the **Matched** and **Unmatched** tabs

# How to Use Assignment Report

## Matched Assets Tab

- ✅ **Validate** that reference objects resolved to the correct actual tables/files
- ✅ **Verify** endpoint locations match expected schemas/databases/file paths
- ✅ **Confirm** lineage is complete by checking asset links in CDGC

## Unmatched Assets Tab

- ✅ **Identify gaps**: Objects referenced by ETL/BI tools but missing from endpoint scans
- ✅ **Naming mismatches**: Reference object names don't match actual asset names (case sensitivity, prefixes, etc.)
- ✅ **Incorrect endpoint**: Assigned to wrong schema/database/file path
- ✅ **Scope issues**: Endpoint scan filters excluded the required objects

# Assignment Remediation & Common Scenarios

## Remediation Workflow

- ✅ **Export unmatched assets** from the report
- ✅ **Analyze patterns**: Look for common naming conventions, missing schemas, or path mismatches
- ✅ **Refine endpoint scans**: Adjust filters, add missing schemas/databases, or rescan with broader scope
- ✅ **Reassign or run connection-aware scan**: For dynamic SQL or unqualified schema names
- ✅ **Re-export and validate**: Confirm match rate improves after remediation

## Common Scenarios

### Scenario 1: 0% Matched After Assignment

**Symptoms**: All assets show as unmatched in the report
**Causes**:
•Endpoint catalog source not scanned or scan failed
•Incorrect endpoint level (e.g., assigned to database when scanner requires schema)
•Case sensitivity mismatch
•Objects don't exist in the assigned endpoint

# Assignment Remediation & Common Scenarios

### Scenario 2: Partial Matches (e.g., 60% Matched)

**Symptoms**: Some assets matched, others unmatched
**Causes**:
- Endpoint scan filters excluded some objects
- Naming conventions differ between ETL tool and database
- Objects span multiple schemas/databases not all assigned
- Dynamic SQL or stored procedures not resolved

# Connection-Aware Scan in CDGC

A **connection-aware scan** is an advanced scanning mode that retrieves **exact database/endpoint metadata** (schemas, tables, columns, stored procedures) from external systems and uses it to compute **complete, accurate column-level lineage**—including scenarios with stored procedures, dynamic SQL, and "SELECT *" queries*.

## Why Is It Needed ?

- ✅ **Complete column-level lineage** is required, especially for assets involving **stored procedures** or **SELECT *** queries
- ✅ Lineage still shows **reference datasets** or is **incomplete after connection assignment**
- ✅ ETL/BI tools use **dynamic SQL**, **unqualified schema names**, or **parameterized queries**
- ✅ **Pre-SQL/Post-SQL** in PowerCenter calls database procedures
- ✅ You need to resolve **cross-schema references** or **runtime-resolved objects**

# Connection-Aware Scan in CDGC

## How it works ?

### Connectionless Scan (Default)

1.) Scanner extracts metadata from ETL/BI tool (IICS, PowerCenter, Power BI, etc.)
2.) Creates **reference connections** and **reference datasets** (placeholders)
3.) Lineage is inferred from static patterns—may be incomplete

### Connection-Aware Scan (After Assignment)

1.) **Initial scan** extracts metadata (can be connectionless)
2.) **Assign connections** in MCC → map reference connections to actual endpoint catalog sources
3.) **Rescan** → CDGC now passes actual connection names and database info to the scanner
4.) Scanner queries the **live database/endpoint** to retrieve exact schemas, tables, columns, procedures
5.) **Complete column-level lineage** is computed, resolving reference objects to actual scanned assets

# Best Practices for Connection Assignment

✅ **Endpoint class type must match scanner requirements** (schema vs DB vs file system)
✅ **One endpoint per connection** reduces ambiguity and improves accuracy
✅ Always review and accept connection recommendations post-scan to ensure accuracy.
✅ Reference objects are **placeholder technical assets** automatically created during catalog scans to represent upstream or downstream objects – allow them to resolve via connection assignment.
✅ Limit **lineage discovery** to known systems to avoid unnecessary scanning and IPU usage.
✅ Periodically re-scan sources and validate lineage updates for accuracy.
✅ **Connection-aware scans** are required for SQL overrides, stored procedures, and implicit schemas
✅ Utilize CLAIRE **Inferred lineage** for data replication and data migration scenarios with consistent naming conventions.

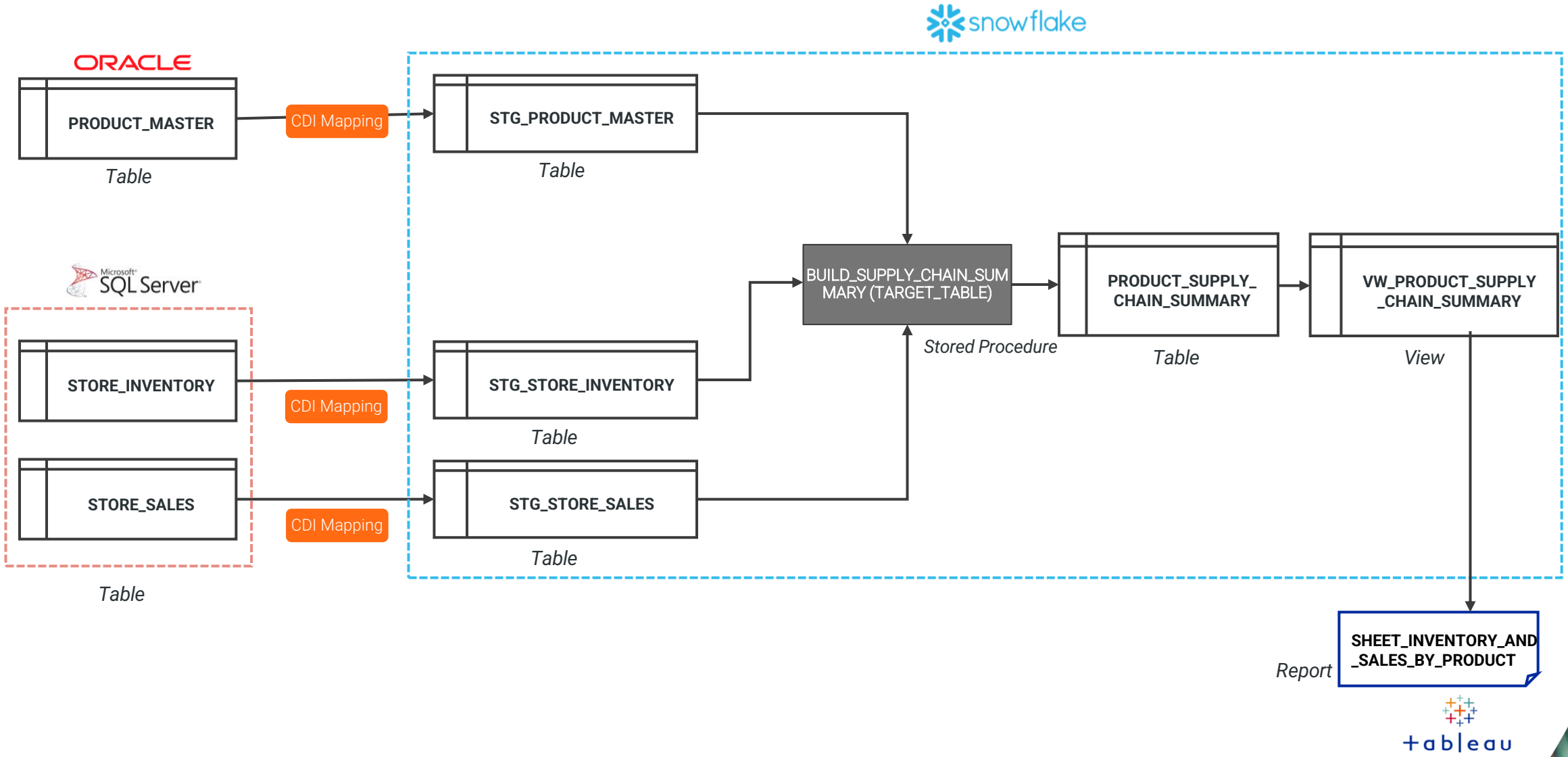
Informatica

# Demo

## Automated Data Lineage through ETL Pipelines, Stored Procedures, Views, and Reporting Tools

Where data & AI come to LIFE

# Use Case for Technical Automated Lineage



# Reference Document Links

Connection Assignment Overview

Success Accelerator - Understanding connection assignment in CDGC

Lineage Discovery - Auto Connection Assignment

How does connection assignment work for IICS resource in CDGC ?

Why is lineage incomplete after connection assignment in CDGC ?

Should connection assignment be done using database or schema ?

Duplicate assets appear in lineage when you perform connection aware scans

Connection-aware scan

# Thank You