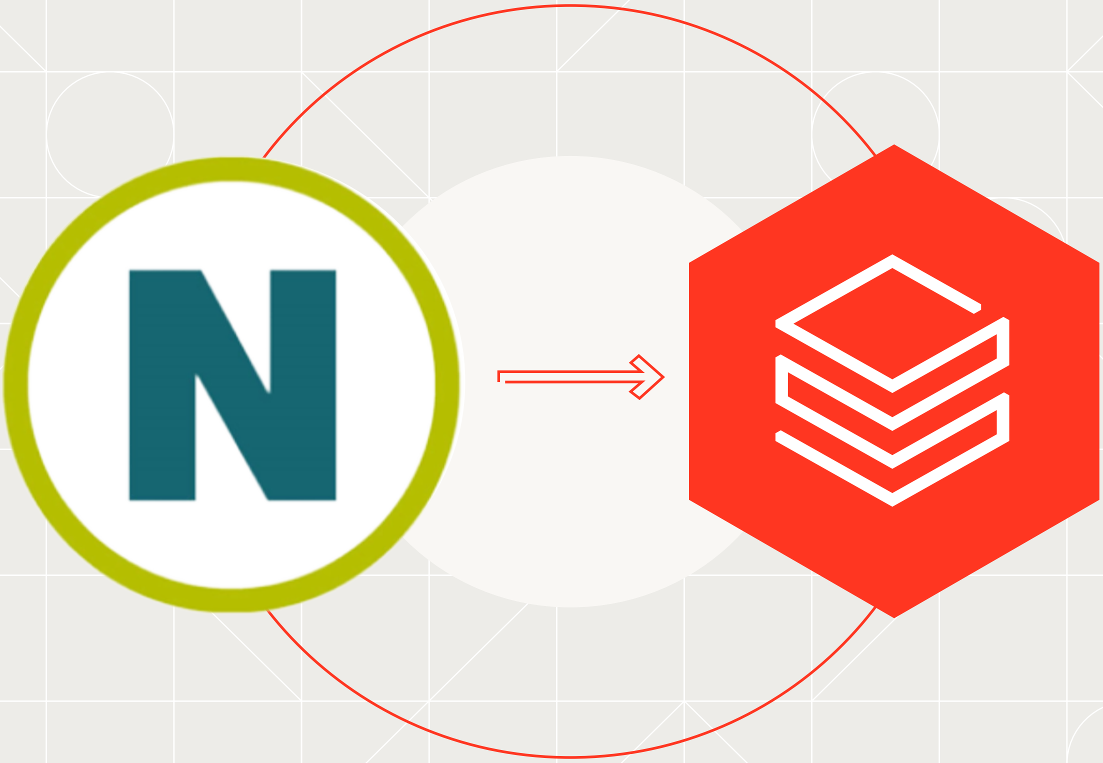
databricks

Guide

Netezza to Databricks Migration Guide



databricks

Contents

Introduction

Netezza enterprise data warehouse (EDW) appliance provides capabilities for storing and analyzing large volumes of data, but it also has some significant drawbacks. High hardware investments, ongoing maintenance costs and expensive licensing requirements often make it challenging and costly to scale. The appliance-based architecture limits flexibility, as expanding storage or processing capacity involves complex node management and can sometimes be impossible once an organization reaches hardware limitations. Migrating to Databricks Data Intelligence Platform eliminates these constraints, offering a scalable, cloud-native lakehouse solution that addresses the limitations associated with traditional hardware appliances like Netezza.

Instead of copying data between warehouses, marts, lakes and ML platforms, Databricks brings different processing engines to a single copy of data in the cloud, enabling seamless data warehousing, AI, ML and GenAI use cases on a single platform and data asset. With its lakehouse architecture, Databricks provides governance, scalability and advanced analytics while decreasing costs and operational complexity. Migrating to Databricks ensures an organization is ready for the future with a unified, efficient and intelligent data platform.

ABOUT THIS GUIDE

This guide provides a detailed roadmap for migrating data warehouse workloads from Netezza to the Databricks Data Intelligence Platform. It outlines key differences between the two systems, standard data and code migration patterns, and best practices to streamline the transition. It also demonstrates how to overhaul a current data model and extract, transform and load (ETL) procedures to fully utilize Databricks' advantages.

Additionally, it compiles proven methodologies, tool options and insights gained from successful migrations. This migration guide covers theoretical concepts and practical applications and is a comprehensive resource for organizations looking to leverage Databricks for enhanced performance, scalability and advanced analytics.

MIGRATION STRATEGY

Migrating from Netezza to Databricks requires careful planning, strategic alignment and clear objectives followed by careful planning and clear target architecture.

Following a proven structured migration methodology is critical to achieving a seamless, effective migration to Databricks, enabling an organization to realize value and position itself for rapid future innovation.

OVERVIEW OF THE MIGRATION PROCESS

Proper planning is required to migrate data and ETL processes from legacy on-premises systems to cloud technologies. This migration involves transferring data and business logic from on-premises infrastructure to the cloud.

Despite the substantial differences between Netezza and Databricks, there are surprising similarities that can facilitate the migration process:

- Despite its proprietary SQL dialect, Netezza's NZSQL adheres mostly to ANSI SQL standards, providing compatibility with Databricks SQL syntax.
- Code migration can be accomplished through code refactoring, leveraging the shared ANSI SQL compliance between the two systems.
- Fundamental data warehouse concepts exhibit similarities across Netezza and Databricks, streamlining the transition process.

The migration process typically consists of the following technical implementation phases:

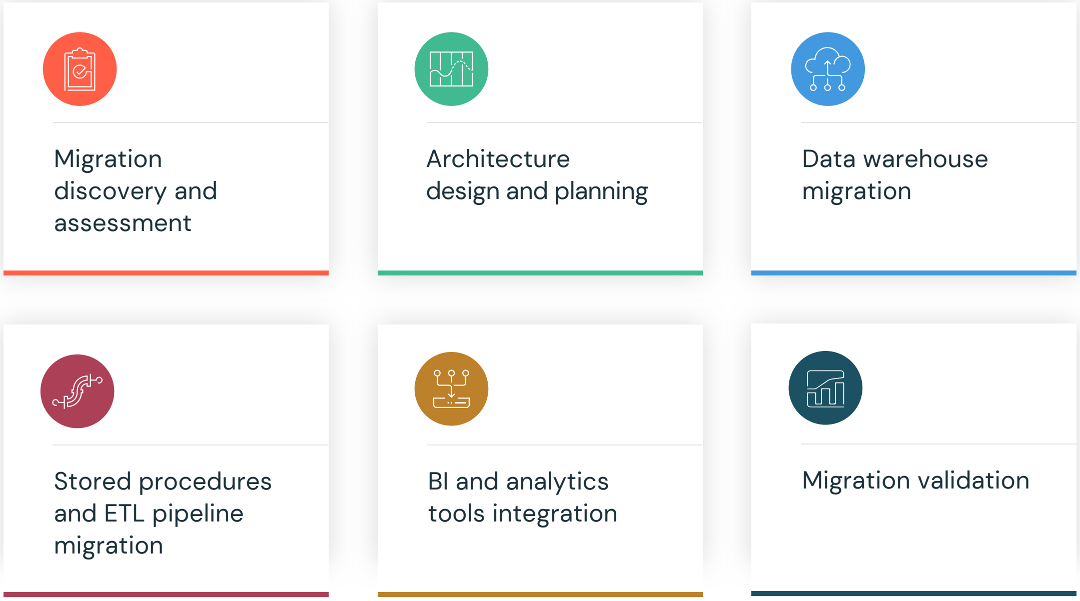

Migration discovery and assessment
Architecture design and planning
Data warehouse migration
Stored procedures and ETL pipeline migration
BI and analytics tools integration
Migration validation

Figure 1: Technical migration approach

Phase 1: Migration Discovery and Assessment

Conducting a migration assessment is crucial before migrating any data or workloads. This assessment enables Databricks to:

- Gain insight into data ingress and egress, ETL patterns, data volume, orchestration tools and execution frequency.
- Understand the technologies involved in upstream and downstream integrations.
- Assess the business criticality and value of the existing systems.
- Evaluate the existing security framework and access control mechanisms.
- Gather pertinent information to provide a realistic estimation of the effort required for migration.
- Compare and calculate infrastructure costs.
- Identify any imminent deadlines, particularly regarding license renewal fees for the existing Netezza setup.
- Document any functional or cross-functional dependencies in the migration plan.

Databricks recommends automation tools, such as the recently acquired BladeBridge Code Analyzer or Partner Analyzer, to expedite the gathering of migration-related information during this phase.

Introduction

Phase 1:
Migration Discovery
and Assessment

Phase 2:
Architecture Design
and Planning

Phase 3:
Data Warehouse
Migration

Phase 4:
Stored Procedures
and ETL Pipelines
Migration

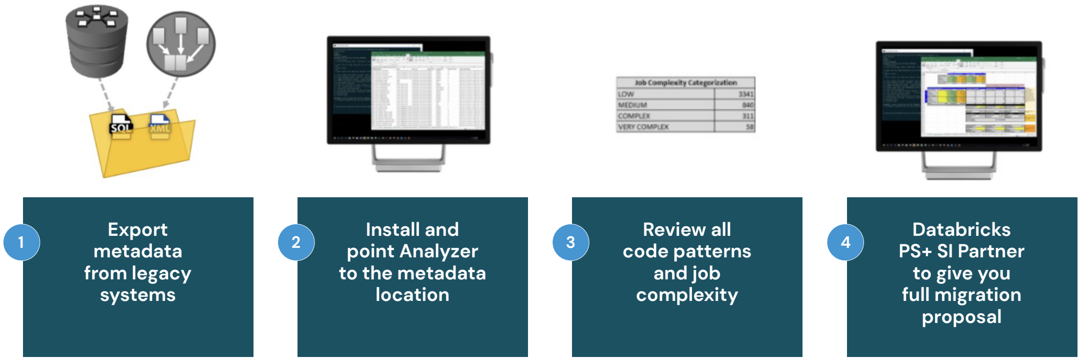
Phase 5:
BI and Analytics
Tools Integration

Phase 6:
Migration Validation

Need Help
Migrating?

Typically, we need to understand Netezza system usage via its system views and catalog tables and get its CPU consumption metrics, inventory of objects, and code complexity of the Netezza stored procedures. We capture the types of workloads, long-running ETL queries and user access patterns. This level of analysis aids in pinpointing databases and pipelines that contribute to high operational costs and complexity, thereby supporting the prioritization process.

Our BladeBridge Code Analyzer not only classifies queries based on their complexity in "T-shirt sizes" (small, medium, large etc.) — but also assesses function compatibility of Netezza scripts and stored procedures, which is vital in ensuring seamless migration.


SQL
XML
1
Export metadata from legacy systems
2
Install and point Analyzer to the metadata location
3
Review all code patterns and job complexity
4
Databricks PS+ SI Partner to give you full migration proposal

Job Complexity Categorization	
LOW	3341
MEDIUM	840
COMPLEX	311
VERY COMPLEX	58

Figure 2: Running Databricks migration analyzer

Phase 2: Architecture Design and Planning

Netezza is a highly customized EDW appliance that has storage, compute and memory tightly coupled and integrated with field-programmable gate arrays (FPGAs) for fast performance. While this architecture provides good performance, it is available in only one Infrastructure form factor which makes it complex, highly specialized, rigid and very expensive to scale. As distributed computing like Spark Data processing framework became more popular — and with the advent of cloud computing the concept of just paying for the compute you use in the cloud became the standard — the economics of EDW appliances collapsed.

The Databricks SQL Warehouse for example is completely serverless and provides superior analytics and data warehousing capabilities using distributed design using horizontal scaling, enabling data distribution and computations across multiple nodes in a cluster. This capability allows Databricks to process large datasets and handle high query volumes efficiently, surpassing the capabilities of a traditional EDW appliance.

EDW ARCHITECTURE

In legacy EDW architectures, data from various systems is typically ingested via ETL tools or ingestion frameworks. After landing in the raw layer, the data progresses to the stage or central layer, where further cleansing and processing occur. Finally, it moves to the last layer, containing the most complex business logic.

After the final layer is prepared, use it for reporting purposes through third-party BI tools such as Tableau, PowerBI, etc.

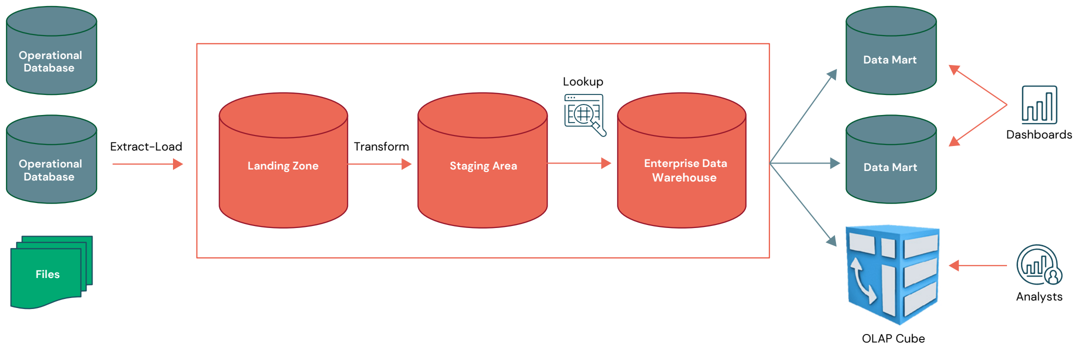


Figure 3: Netezza reference architecture of an enterprise data warehouse

It is imperative to analyze the current architecture comprehensively, which involves understanding upstream and downstream integrations and the respective tools and technologies.

Following this analysis, assess the potential for modernizing each stage of the target architecture and how well an organization can transition from legacy systems to modern alternatives at each stage. Key decisions on data ingest into cloud storage include evaluating features like Databricks Autoloader, Lakeflow Connect or Lakehouse Federation. The stage evaluates ETL modernization, partners and BI tool compatibility checks. A target architecture and tooling roadmap is created that guides the migration process.

Below is an example of a data warehousing architecture on Databricks with various ISV Partner Integration options.

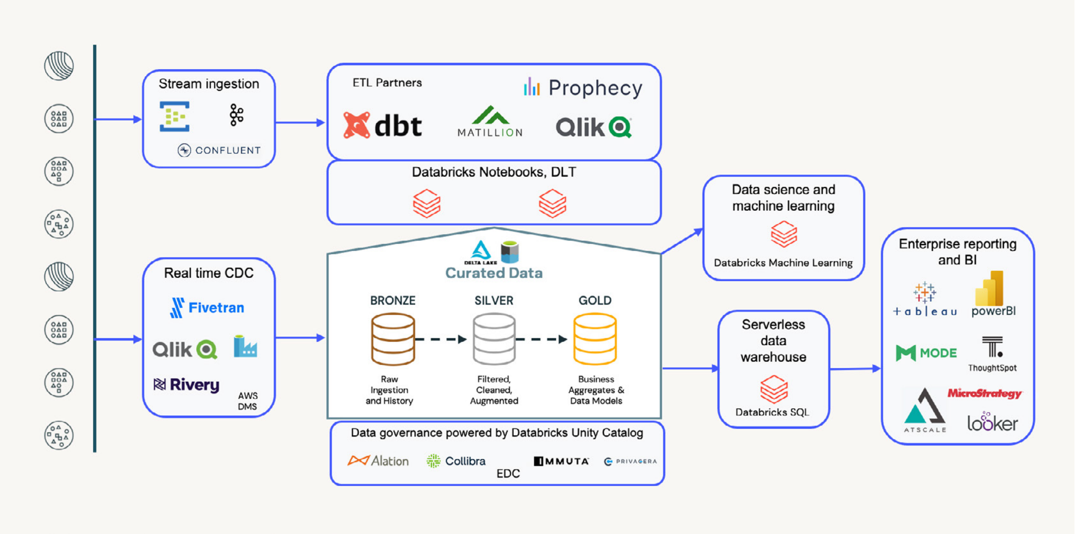

Stream ingestion
CONFLUENT
ETL Partners
Prophecy
dbt
MATILLION
Qlik
Databricks Notebooks, DLT
Real time CDC
Fivetran
Qlik
Rivery
AWS DMS
DELTA LAKE
Curated Data
BRONZE
SILVER
GOLD
Raw Ingestion and History
Filtered, Cleaned, Augmented
Business Aggregates & Data Models
Data governance powered by Databricks Unity Catalog
Alation
Collibra
EDC
IMMUTA
PRIVACERA
Data science and machine learning
Databricks Machine Learning
Serverless data warehouse
Databricks SQL
Enterprise reporting and BI
tableau
powerBI
MODE
ThoughtSpot
MicroStrategy
ATSCALE
looker

Figure 4: Modern data warehousing on Databricks

Undertaking a thorough analysis of source systems feeding Netezza, ingestion methodology, ETL jobs and their design, Data Governance tool integrations and BI and AI consumption patterns and figuring out how to replace each architectural pattern in the target Databricks architecture. This systematic process ensures a comprehensive understanding of the current legacy EDW architecture and facilitates a smoother transition to target architecture by identifying equivalent services and functionalities.

Typically, by the end of this phase, we have a good handle on the scope and complexity of the migration and can come up with a more accurate migration plan and cost estimate. Be sure to use the Databricks Well-architected Framework while designing the Databricks architecture on any cloud. (Azure Databricks well-architected framework | AWS Databricks well-architected data lakehouse | Introduction to the well-architected data lakehouse).

Phase 3: Data Warehouse Migration

CONSIDERATIONS FOR WORKSPACE CREATION

The Databricks environment can be set up by following the Setup and Administration Guides. Please see the Azure Databricks Administration Guide, AWS Databricks administration introduction, or Databricks administration introduction on GCP, depending on the cloud of choice. .

CONSIDERATIONS FOR SCHEMA AND DATA MIGRATION

Once the Databricks Unity Catalog and workspaces have been established, the initial migration phase involves migrating schema and data, including metadata such as table data definition language (DDL) scripts, views and table data.

As you navigate migrating data out of Netezza, it's crucial to consider several key decisions. These include:

- What is the target design for the migrating tables?
- Should the destination retain the same hierarchy of catalogs, databases, schemas, and tables?
- We recommend considering the potential cleanup or reorganization of the existing data footprint in Databricks. This step could significantly enhance the efficiency and effectiveness of the migration process, reducing potential issues, simplifying data management and optimizing resource use in the new environment.

There are also a few recommendations that can help to enable a smoother and less risky migration:

- Data modeling: As part of the migration, a similar data model might need to be refactored or reproduced in an automated and scalable fashion. Visual data modeling tools like Quest ERWIN or sqldbm can be found in Databricks Partner Connect. These tools can help accelerate the development and deployment of the refactored data model with just a few clicks. Such tools can reverse engineer Netezza data models (table structures and views) and implement tables in Databricks easily.
- When migrating DDLs, verifying the provenance of the data schema (e.g., source data) is essential. Consider data type compatibility between Netezza and Databricks and make changes accordingly.
- We recommend using the Databricks medallion architecture to call out landing zones in Bronze, central repository or data domains in Silver and presentation layer in Gold for logical data organization in the Lakehouse architecture. For more details on modeling approaches and design patterns, refer to Data Warehouse Modeling Techniques.

PHASE 3.1: SCHEMA MIGRATION

Before offloading tables to Databricks, it's essential to establish their schema within the Databricks environment. The extracted table DDLs can be converted to Databricks, keeping the data type compatibility and changes in mind.

DATA TYPES

NETEZZA	DATABRICKS SQL	NOTES
INTEGER/INT/ INT4	INT	Represents 4-byte signed integer numbers
SMALLINT/INT2	SMALLINT	Represents 2-byte signed integer numbers

NETEZZA	DATABRICKS SQL	NOTES
BYTEINT/INT1	TINYINT	Represents 1-byte signed integer numbers
BIGINT/INT8	BIGINT	Represents 8-byte signed integer numbers
DECIMAL	DECIMAL(p,s)	Represents numbers with maximum precision p and fixed scale s
NUMERIC	BIGINT	Represents 8-byte signed integer numbers
NUMERIC(p,s)	DECIMAL(p,s)	Represents numbers with maximum precision p and fixed scale s
FLOAT(p)	FLOAT	Represents 4-byte single-precision floating point numbers
REAL/FLOAT(6)	FLOAT	Represents 4-byte single-precision floating point numbers
DOUBLE PRECISION/ FLOAT(14)	DOUBLE	Represents 8-byte double-precision floating point numbers
CHAR/ CHARACTER	STRING	Represents character string values
VARCHAR	STRING	Represents character string values
NCHAR	STRING	Represents character string values
NVARCHAR	STRING	Represents character string values
VARBINARY	BINARY	Represents character string values
ST_GEOMETRY		
BOOLEAN/BOOL	BOOLEAN	Represents Boolean values

NETEZZA	DATABRICKS SQL	NOTES
DATE	DATE	Represents values comprising values of fields year, month, and day, without a time zone
TIME		Returns the current session local time
TIMETZ/TIME WITH TIME ZONE	current_timezone	Returns the current session's local timezone
TIMESTAMP	TIMESTAMP	Represents values comprising values of fields year, month, day, hour, minute, and second, with the session local timezone
	VOID	Represents the untyped NULL
	INTERVAL intervalQualifier	Represents intervals of time either on a scale of seconds or months
	ARRAY <elementType>	Represents values comprising a sequence of elements with the type of elementType
	MAP <keyType,valueType>	Represents values comprising a set of key-value pairs
	STRUCT < [fieldName : fieldType [NOT NULL][COMMENT str][, …]] >	Represents values with the structure described by a sequence of fields

Key pointers to keep in mind during DDL conversion:

1 Table options such as distributions and indexes are not applicable in Delta tables.

2 Several keywords like DISTRIBUTE ON for data distribution should be converted to z-order indexing by using the ZORDER BY clause. Or use the automatic liquid clustering feature, which uses Predictive Optimization to analyze how your tables are queried and intelligently select the most effective clustering keys based on your workload. Predictive Optimization dynamically adjusts the clustering scheme as query patterns change, eliminating the need for manual tuning or data layout decisions when setting up your Delta tables.

3 Databricks supports IDENTITY columns on bigint columns.

4 Specify additional Delta table properties via the TBLPROPERTIES clause, e.g., delta.targetFileSize, delta.tuneFileSizesForRewrites, delta.columnMapping.mode, and others.

PHASE 3.2: DATA MIGRATION

Transferring legacy on-premise data to a cloud storage location for seamless consumption in Databricks can be a demanding task, but we have a few viable options:

1. **Netezza native options:** Leveraging Netezza's data transfer utilities like NZUNLOAD can export data at high performance using parallelization, and this technique is specifically optimized for transferring large volumes of data to cloud storage. Once data is in cloud storage, the Databricks autoloader can be used to consume it and load it in Databricks.

2. **Cloud Data Transfer Tools:** Native AWS tooling, such as AWS Database Migration Services (DMS) or Azure Data Factory (ADF), can be used to migrate data from Netezza to cloud storage in CSV, parquet or Delta Lake formats. From that point onward, Databricks autoloader can be used.

3. **Databricks Ingestion ISV Partners:** Databricks Ingestion ISV Partners such as Qlik can replicate data from Netezza to the Databricks Delta table for historical and CDC data.

PHASE 3.3: OTHER DATABASE OBJECTS MIGRATION

Other Database Objects such as Views, Stored procedures, Macros, Functions, and Teradata Load Utilities can also be easily migrated to Databricks via our automated code conversion processes. Please review this helpful cheat sheet packed with essential tips and tricks to help you start on Databricks using SQL programming in no time! Some key pointers while converting Netezza-specific SQL objects:

- **Views** — typically used for data access control. In the context of the medallion architecture, they can be considered part of the Gold layer. Views would also be used as an intermediate data structure while transforming data and publishing business KPIs to final users.
- **Netezza SPM Materialized Views** — Netezza sorted, projected and materialized (SPM) Views can be converted to Databricks' Materialized Views to replace the functionality.
- **Netezza Stored Procedures** — Databricks now support SQL scripting so that stored procedures for lift and shift are now easily possible.
- **Netezza SQL Functions** — Convert Netezza SQL functions to Databricks SQL functions.

Stored procedures implementation in Databricks

Note that with the newly released Databricks SQL Scripting support, you can now easily deploy or convert powerful procedural logic within Databricks. Databricks SQL scripting supports compound statement blocks (with BEGIN….END). Within the Databricks SQL scripting procedures, we can declare local variables, user-defined functions, use condition handlers for catching exceptions and use flow control statements such as FOR loops over query results, conditional logic such as IF and CASE CASE and means to break out loops such as LEAVE and ITERATE. These features make stored procedures migration to Databricks even easier.

Implement Slowly Changing Dimensions

Here are a quick refresher and code snippets on quickly implementing slowly changing dimensions (SCDs).

- How to implement SCDs when you have duplicates – Part 1
- How to implement SCDs when you have duplicates – Part 2: DLT
- APPLY CHANGES API: Simplify change data capture in DLT

PHASE 3.4: DATA GOVERNANCE MIGRATION

When discussing security migration, it's essential to consider both authentication and authorization. When planning the migration from Netezza, it's critical to understand the differences between the two platforms and accurately map Netezza's data security policies to Databricks Data Intelligence Platform security policies.

Audit Logging

In Databricks, most user-related security features are managed through the Unity Catalog. Audit logging is also available in system tables. For more information, refer to the Audit log system.

Unity Catalog

We recommend leveraging Databricks Unity Catalog, which offers a unified governance layer for data and AI within the Databricks Data Intelligence Platform. With Unity Catalog, organizations can seamlessly govern their structured and unstructured data, machine learning models, notebooks, dashboards and files on any cloud or platform.

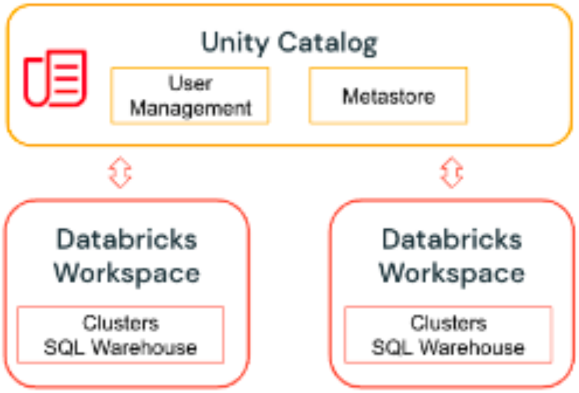


Figure 5: Unity Catalog works across clouds or platforms

Key features of Unity Catalog include:

- **Define once, secure everywhere:** Unity Catalog offers a single place to administer data access policies that apply across all workspaces.
- **Standards-compliant security model:** Unity Catalog's security model is based on standard ANSI SQL and allows administrators to grant permissions in their existing data lake using familiar syntax at the level of catalogs, databases (also called schemas), tables and views.
- **Built-in auditing and lineage:** Unity Catalog automatically captures user-level audit logs that record access to your data. Unity Catalog also captures lineage data that tracks how data assets are created and used across all languages.
- **Data discovery:** Unity Catalog lets you tag and document data assets and provides a search interface to help consumers find data.
- **System tables:** Unity Catalog lets you easily access and query your account's operational data, including audit logs, billable usage and lineage.

Unity Catalog Object Hierarchy

Unity Catalog consists of a hierarchy of securable objects. The following illustrates a top flow of primary objects:

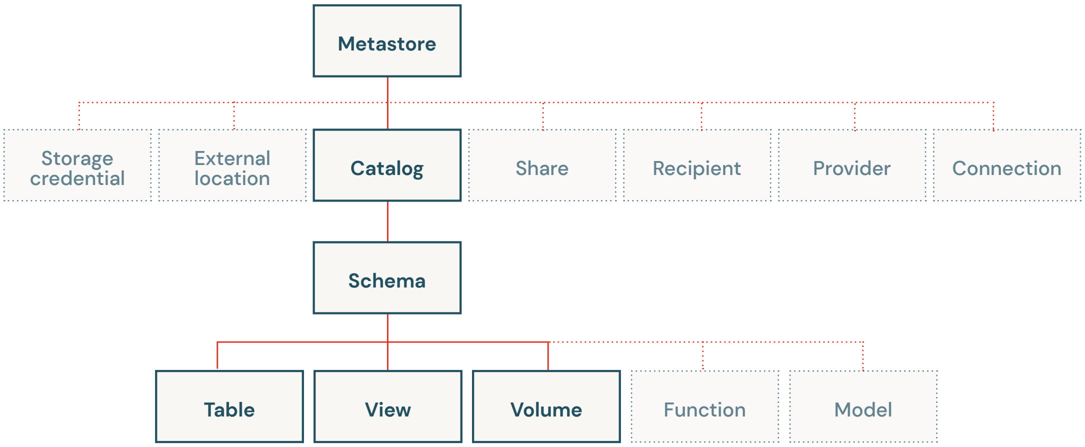


Figure 6: Unity Catalog object hierarchy

Phase 4:
Stored Procedures and ETL Pipelines Migration

Data orchestration migration, stored procedure migration, and ETL migration are the key elements of the migration process, and Databricks Automated Code converters can help with these.

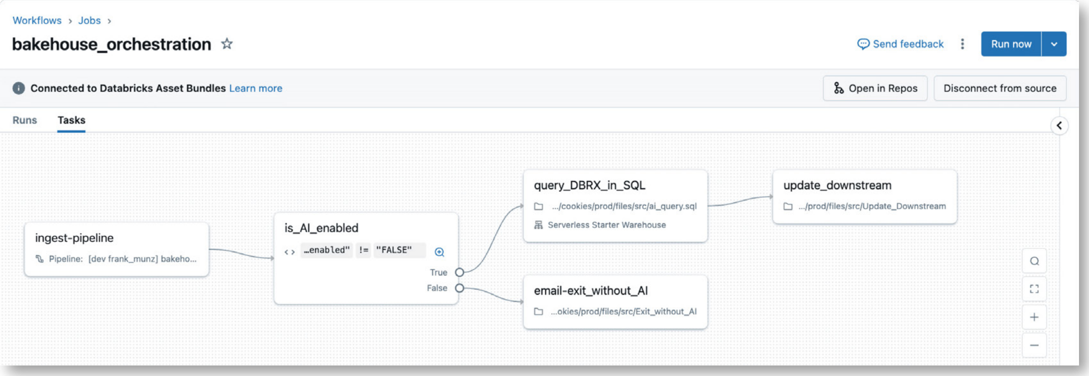
ORCHESTRATION MIGRATION

ETL orchestration involves coordinating and scheduling end-to-end pipelines, including data ingestion, integration and result generation. Netezza typically manages this orchestration using third-party tools like Control-M or Autosys. When migrating these workflows, there are usually several options available to replicate this orchestration functionality:

1. Use Databricks Workflows to orchestrate the migrated pipelines. Databricks Workflows support various tasks, such as Python scripts, Notebooks, dbt transformations and SQL tasks. The customer needs to provide job sequences and schedules as a prerequisite for converting them into Databricks workflows.

2. DLT Pipelines provides a standard framework for building batch and streaming use cases. It also includes critical data engineering features such as automatic data testing, deep pipeline monitoring, and recovery. It also has out-of-the-box functionality for Slow Change Dimension (SCD) Type 1 and Type 2 tables.

3 | External tools like Apache Airflow are also possible. Considering the tightly coupled Databricks Workflows with the Databricks Intelligence Platform, we recommend using Databricks Workflows for better integration, simplicity and lineage.



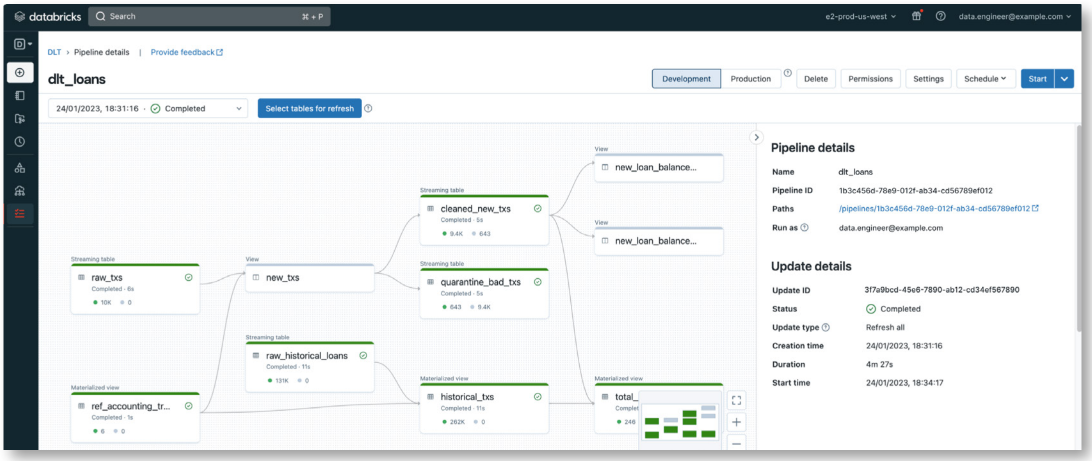
Workflows › Jobs ›

bakehouse_orchestration ☆

Connected to Databricks Asset Bundles Learn more

Send feedback
Run now
Open in Repos
Disconnect from source
Runs
Tasks
ingest-pipeline
Pipeline: [dev frank_munz] bakeho...
is_AI_enabled
...enabled" != "FALSE"
True
False
query_DBRX_in_SQL
.../cookies/prod/files/src/ai_query.sql
Serverless Starter Warehouse
email-exit_without_AI
...okies/prod/files/src/Exit_without_AI
update_downstream
.../prod/files/src/Update_Downstream

Figure 7: Databricks Workflows



DLT › Pipeline details | Provide feedback

dlt_loans

24/01/2023, 18:31:16 · Completed

Pipeline details

Name dlt_loans
Pipeline ID 1b3c456d-78e9-012f-ab34-cd56789ef012
Paths /pipelines/1b3c456d-78e9-012f-ab34-cd56789ef012
Run as data.engineer@example.com

Update details

Update ID 3f7a9bcd-45e6-7890-ab12-cd34ef567890
Status Completed
Update type Refresh all
Creation time 24/01/2023, 18:31:16
Duration 4m 27s
Start time 24/01/2023, 18:34:17

Figure 8: DLT pipelines

SOURCE/SINK MIGRATION

Like orchestration, GUI-based ETL mappings or like MS SQL Server through SSIS are usually in a typical legacy EDW-ETL architecture to extract data from source systems, transform it and load it into the final tables.

1 | Source data connections

- Duplicate and configure ingestion tools to point to Databricks Delta Lake instead of Netezza staging. Delta is an open source format widely supported as the target data format for popular data ingestion tools.

2 | Sink data connections

- Ingestion tools and frameworks will now generate data in Delta tables instead of Netezza tables.

Note: If you use ETL tools like IBM DataStage to write to Netezza, their latest version can now write to Delta Lake format in cloud storage. Databricks can then efficiently process these files via Databricks autoloader. Alternatively, you must convert these DataStage workflows to Databricks Notebooks or DLT.

If a GUI-based ETL tool is preferred, additional integration with tools like Cloud, Prophecy or Matillion will be necessary.

These additional integrations should be accounted for when estimating the overall effort, and you should make necessary provisions to avoid surprises during the implementation phase.

STORED PROCS AND ETL CODE CONVERSION

Migrating from Netezza SQL to Databricks SQL requires identifying and replacing any incompatible/proprietary Netezza SQL functions or syntax. Databricks has mature code converters and migration tooling to make this process smoother and highly automated.

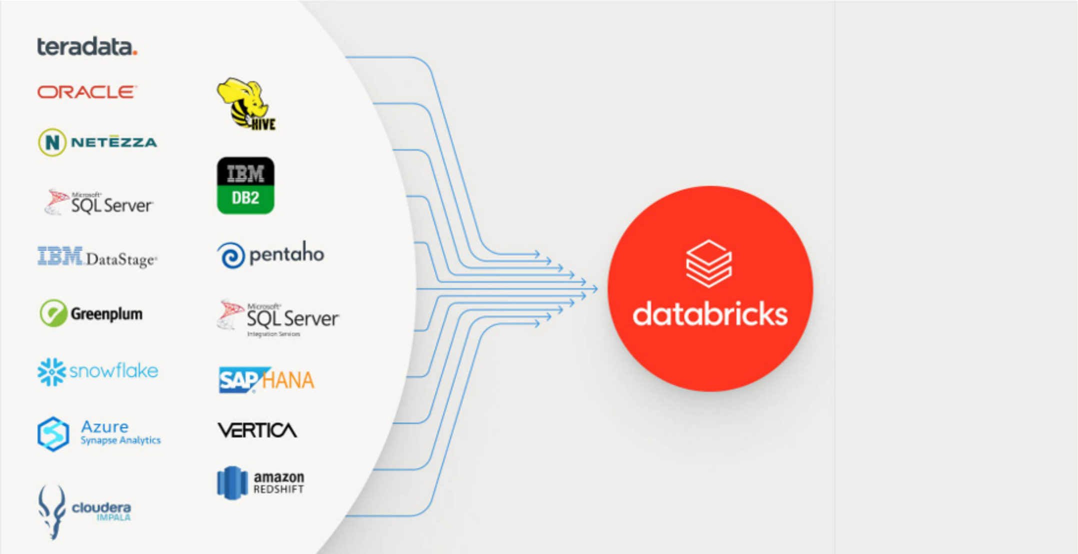


Figure 9: Databricks migration tooling simplifies legacy EDW migrations

Databricks Code Converter (BladeBridge)

Databricks Code Converter (acquired from BladeBridge) offers automated tooling to modernize and convert Teradata code to Databricks.

- **Automated conversion:** Databricks Converter can automatically convert SQL workloads, significantly speeding up and de-risking migration projects.
- **Broad support:** It supports a wide range of legacy EDW and ETL platform syntax and can convert legacy code to Databricks.
- **Broad adoption by services firms:** Most System Integrator partners have deep expertise and access to our converters.
- **Cost and time-effective:** Our Converter reduces the cost and time required for a migration project by automating the process.
- **Decreases complexity:** The tool reduces the complexity of the migration process by providing a systematic approach to conversion.

Databricks Code Converter supports schema conversion (tables and views), SQL queries (select statements, expressions, functions, user-defined functions, etc.) and stored procedures. The conversion configuration is externalized, meaning users can extend conversion rules during migration projects to handle new code pattern sets to achieve a more significant percentage of automation. You can create a migration proposal with automated converter tooling via Databricks Professional Services or our certified Migration Brickbuilder SI Partners. Databricks Code Converter tooling requires Databricks professional services or a Databricks SI Partner agreement.

CODE OPTIMIZATION

Many queries will likely need to be refactored and optimized during the migration process. Easy techniques like automated liquid clustering and predictive optimization make performance tuning almost an automated process in Databricks. Predictive optimization uses techniques like:

1 | Compaction – which optimizes file sizes.

2 | Liquid clustering – that incrementally clusters incoming data, enabling optimal data layout and efficient data skipping.

3 | Running vacuum – which reduces costs by deleting unneeded files from storage.

4 | Automatic updating of statistics – running the ANALYZE STATISTIC command on the required columns for best performance.

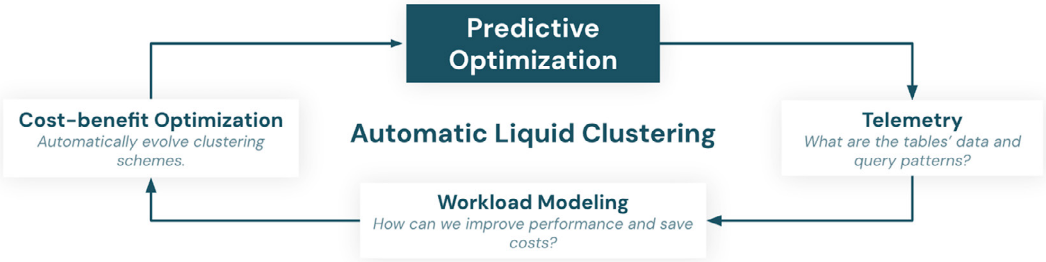
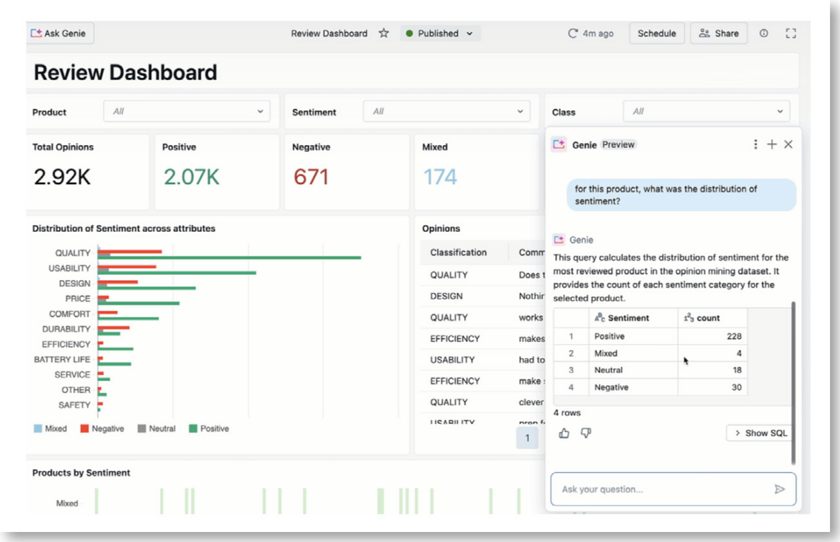

Predictive Optimization
Telemetry
What are the tables' data and query patterns?
Workload Modeling
How can we improve performance and save costs?
Cost-benefit Optimization
Automatically evolve clustering schemes.
Automatic Liquid Clustering

Figure 10: Automatic liquid clustering

Phase 5: BI and Analytics Tools Integration

UNLOCK ADVANCED ANALYTICS WITH DATABRICKS SQL

With the data and ETL migration done and security and governance setup via Unity Catalog – unleash your AI and BI use cases by spinning up Databricks SQL!



for this product, what was the distribution of sentiment?

Genie

This query calculates the distribution of sentiment for the most reviewed product in the opinion mining dataset. It provides the count of each sentiment category for the selected product.

	Sentiment	count
1	Positive	228
2	Mixed	4
3	Neutral	18
4	Negative	30

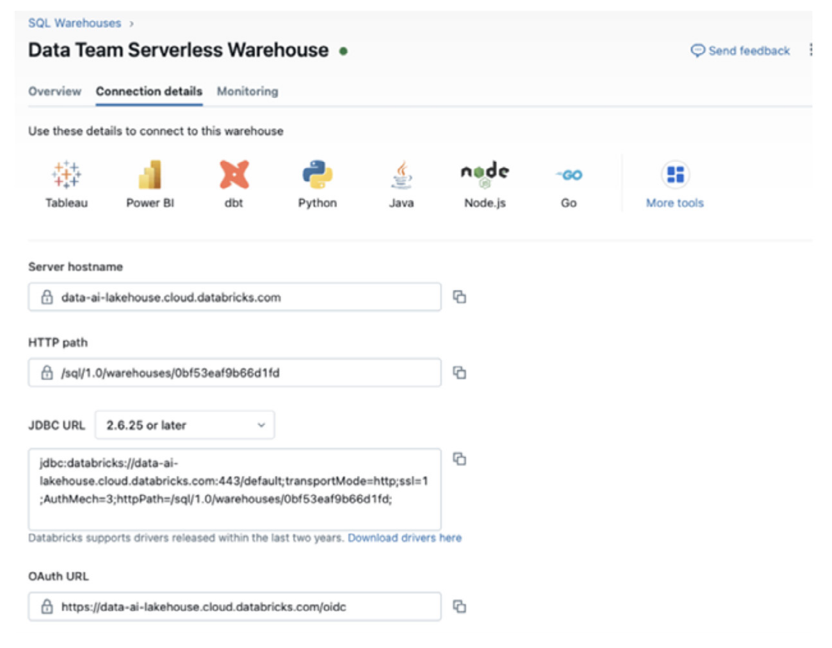
4 rows

Figure 11: Databricks SQL

Databricks SQL Warehouse is a serverless data warehousing solution that integrates seamlessly with the Databricks Data Intelligence Platform, offering a unified environment for data, analytics and AI workloads. Key features include:

- **Serverless Architecture:** Provides instant and elastic compute resources, eliminating the need for manual infrastructure management and ensuring rapid scalability to handle varying workloads efficiently.
- **AI-Driven Performance:** Utilizes AI-powered optimizations, such as the Photon query engine, Predictive IO, and Intelligent Workload Management, to enhance query execution speed and resource efficiency.

- **Unified Governance with Unity Catalog:** This solution offers centralized governance and security features, including data discovery, auditing, and fine-grained access controls, ensuring compliance and data integrity across the organization.
- **Comprehensive SQL Functionality:** Supports standard ANSI SQL, materialized views, primary and foreign key constraints, and advanced data types like 'Variant' for semi-structured data, enabling efficient and flexible data modeling and querying capabilities.
- **Seamless Integration with BI tools:** Integrate with various business intelligence tools like Tableau, Power BI, Thoughtspot, Mode and more. It also supports Lakehouse Federation to legacy EDWs like Teradata and Oracle. This integration allows users to query and govern siloed data systems as an extension of their lakehouse, enhancing data accessibility and collaboration.



Data Team Serverless Warehouse ●

Use these details to connect to this warehouse

Server hostname

data-ai-lakehouse.cloud.databricks.com

HTTP path

/sql/1.0/warehouses/0bf53eaf9b66d1fd

JDBC URL: 2.6.25 or later

```
jdbc:databricks://data-ai-
lakehouse.cloud.databricks.com:443/default;transportMode=http;ssl=1
;AuthMech=3;httpPath=/sql/1.0/warehouses/0bf53eaf9b66d1fd;
```

Databricks supports drivers released within the last two years. Download drivers here

OAuth URL

https://data-ai-lakehouse.cloud.databricks.com/oidc

Figure 12:
Databricks SQL BI integrations

These features empower organizations to perform high-performance analytics, streamline data workflows, and derive actionable insights from their data with reduced operational complexity.

DATA WAREHOUSING AND BI REPORT MODERNIZATION

Once you migrate ingestion and transformation pipelines to the Databricks Data Intelligence Platform, ensuring the business continuity of downstream applications and data consumers is critical. Databricks Data Intelligence Platform has validated large-scale BI integrations with many popular BI tools like Tableau, Power BI, Qlik, ThoughtSpot, Sigma, Looker and more.

As described in the blog (see section 3.5 Repointing BI workloads), one common way to repoint BI workloads after data migration is to test sample reports, renaming existing tables' data source/table names and pointing to the new ones.

Typically, if the schema of the tables and views post-migration hasn't changed, repointing is a straightforward exercise in handling switching databases on the BI dashboard tool. If the schema of the tables has changed, you will need to modify the tables/views in the lakehouse to match the expected schema of the report/dashboard and publish them as a new data source for the reports.

Many customers take this opportunity to optimize their BI models and semantic layers to align with business needs.

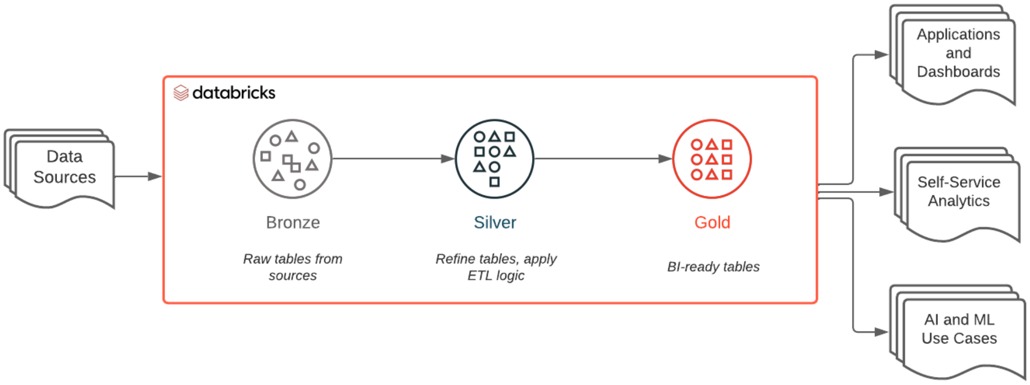
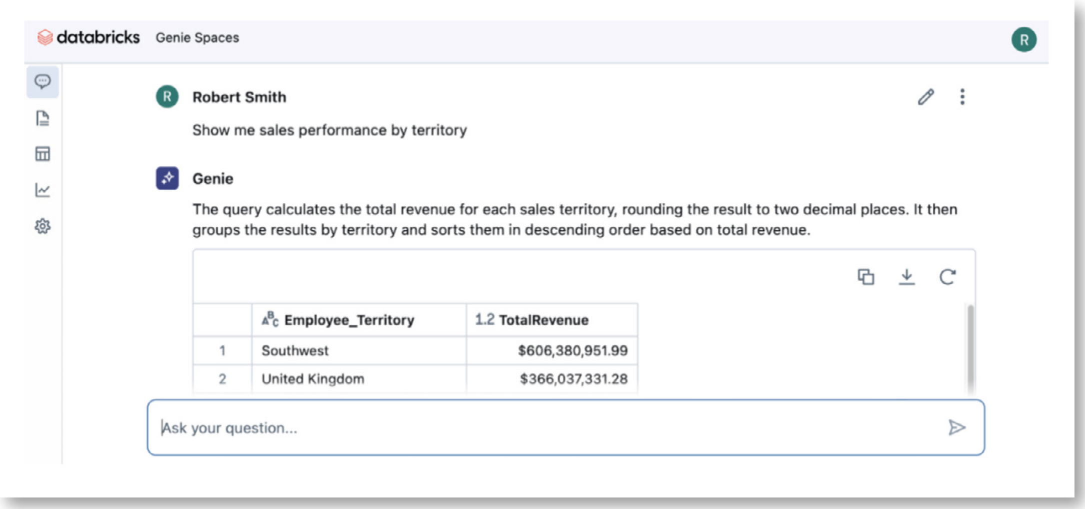


Figure 13: Future-state architecture

One additional advantage of using Databricks for BI is that you get a conversational user interface to chat with your data using our Genie AI/BI Interface, which opens up your data for analytics and Q&A even for non-SQL users.



Robert Smith

Show me sales performance by territory

Genie

The query calculates the total revenue for each sales territory, rounding the result to two decimal places. It then groups the results by territory and sorts them in descending order based on total revenue.

	Employee_Territory	TotalRevenue
1	Southwest	$606,380,951.99
2	United Kingdom	$366,037,331.28

Ask your question...

Figure 14: Databricks Genie

Phase 6: Migration Validation

The primary validation method for a data pipeline is the resulting dataset itself. We recommend establishing an automated testing framework to apply to any pipeline. Typically, this involves using a testing framework with a script capable of automatically comparing values in both platforms.

Databricks recommends you perform the following checks at a minimum:

- Check to see if a table exists
- Check the counts of rows and columns across the tables
- Calculate various aggregates over columns and compare, for example:
  - SUM, MIN, MAX, AVG of numeric columns
  - MIN, MAX for string and date/time columns
  - COUNT(*), COUNT(NULL), COUNT(DISTINCT) for all columns

Run the pipelines in parallel for a specific period (we find one week to be an acceptable baseline, but you may wish to extend this to ensure stability) and review the comparison results to ensure the data is ingested and transformed into the proper context.

It is advisable to initiate validation with your most critical tables, which often drive the results or calculations of tables in the gold layer. This validation includes control tables, lookup tables and other essential datasets.

A robust data validation requires the following components:

- **Snapshot(s):** Data to work with, including a *pre*- and *post-version* for each script (ideal) and job being migrated.
- **Table comparison code:** A standardized way to compare the result table to determine whether the test is successful. You can compare the tables based on:
  - schema checks
  - row count checks
  - row-by-row checks
- Identifying the primary key combination from the customer is essential to check counts and row-by-row comparisons.

We have tooling such as Remorph Reconcile to streamline the reconciliation process between source data and target data residing on Databricks and other source platforms.

For more advanced table data and schema comparison, use tools like Datacompy.

Need Help Migrating?

Regardless of size and complexity, the Databricks Professional Services team and an ecosystem of certified migration services partners and ISV partners offer different levels of support (advisory/assurance, staff augmentation, scoped implementation) to accelerate your migration and ensure successful implementation.

When engaging with our experts, you can expect:

- **Discovery and Profiling:** Our team starts by clearly understanding migration drivers and identifying challenges within the existing Netezza deployment. We conduct collaborative discussions with key stakeholders, leveraging automated profiling tools to analyze legacy workloads. This is used to determine drivers of business value and total cost of ownership (TCO) savings achievable with Databricks.
- **Assessment:** Using automated tooling, we perform an analysis of existing code complexity and architecture. This assessment helps estimate migration effort and costs, refine migration scope and determine which parts of the legacy environment require modernization or can be retired.
- **Migration Strategy and Design:** Our architects will work with your team to finalize the target Databricks architecture, detailed migration plan and technical approaches for the migration phases outlined in this guide. We will help select appropriate migration patterns, tools and delivery partners and collaborate with our certified SI partners to develop a comprehensive Statement of Work (SOW).

Introduction

Phase 1:
Migration Discovery
and Assessment

Phase 2:
Architecture Design
and Planning

Phase 3:
Data Warehouse
Migration

Phase 4:
Stored Procedures
and ETL Pipelines
Migration

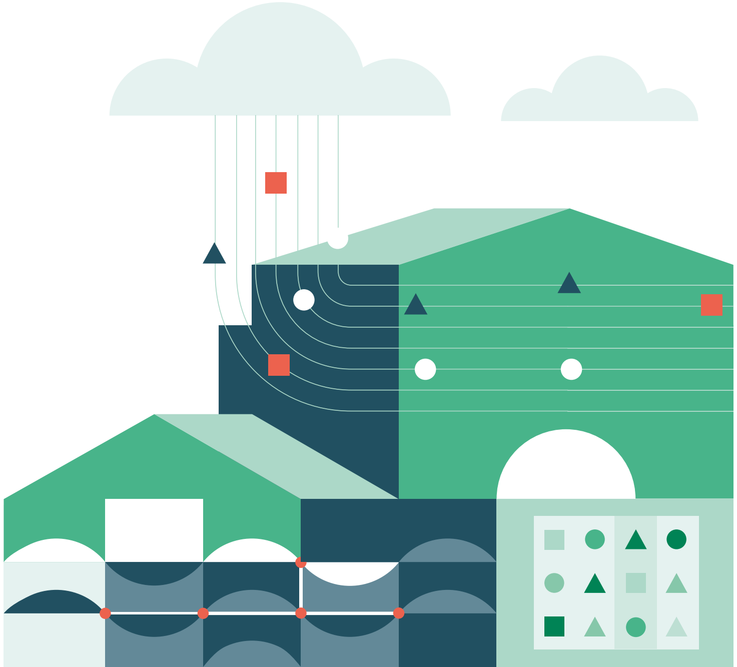
Phase 5:
BI and Analytics
Tools Integration

Phase 6:
Migration Validation

Need Help
Migrating?

- **Execute and Scale:** We and our certified partners deliver on our comprehensive migration plan and then work with your team to facilitate knowledge sharing and collaboration and scale successful practices across the organization. Our experts can help you set up a Databricks Center of Excellence (CoE) to capture and disseminate lessons learned and drive standardization and best practices as you expand to new use cases.

Contact your Databricks representative or use this form for more information. Our specialists can help you every step of the way!



About Databricks

Databricks is the data and AI company. More than 10,000 organizations worldwide — including Block, Comcast, Condé Nast, Rivian, Shell and over 60% of the Fortune 500 — rely on the Databricks Data Intelligence Platform to take control of their data and put it to work with AI. Databricks is headquartered in San Francisco, with offices around the globe, and was founded by the original creators of Lakehouse, Apache Spark™, Delta Lake and MLflow.

To learn more, follow Databricks on LinkedIn, X and Facebook.

Start your free trial