Databricks Lakehouse Portfolio

JDE Aerospace Data Platform on Azure Databricks

John Paul (JP) Castro

Senior Data Platform Architect

25+ Years | Aerospace, Entertainment, Telecommunications

jpcenterprises.com | github.com/johnpaul-castro

johnpaulcastro@gmail.com | (818) 943-5159

Built on Databricks 14-Day Trial | May 2026

Unity Catalog | Lakeflow Declarative Pipelines | Serverless SQL

Contents

1. Architecture Overview

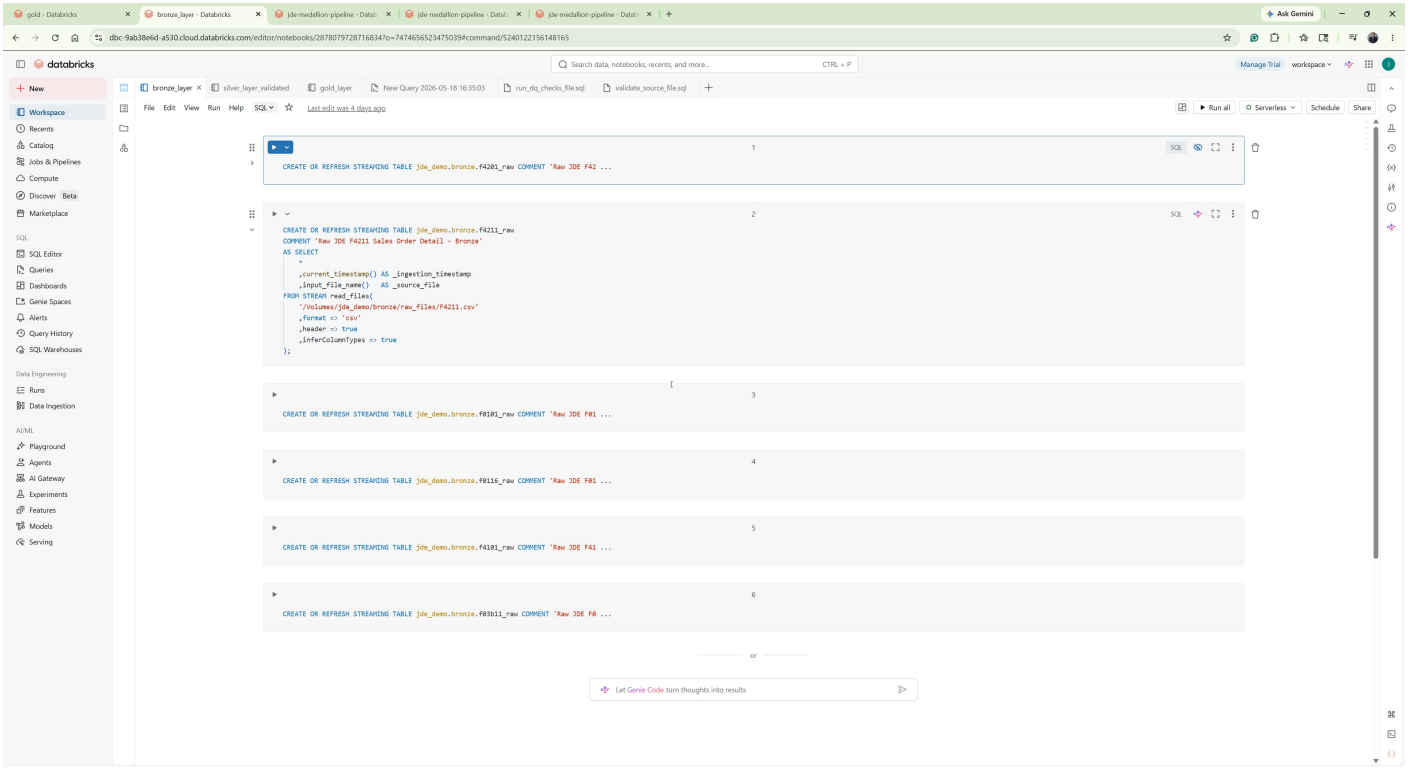
This portfolio demonstrates a complete Databricks Lakehouse implementation built on a 14-day trial workspace. The platform ingests JDE (JD Edwards) ERP data from six source tables representing a realistic aerospace distribution environment, processes it through a medallion architecture (Bronze, Silver, Gold), and delivers analytics through AI/BI dashboards and Genie Spaces.

- **6 JDE source tables:** F4201 (Sales Order Header), F4211 (Sales Order Detail), F0101 (Address Book), F0116 (Phone/Address), F4101 (Item Master), F03B11 (Accounts Receivable)
- **Medallion architecture:** Bronze (raw streaming), Silver (cleansed + validated), Gold (business models + DQ)
- **22 pipeline tables:** 6 Bronze streaming tables, 6 Silver materialized views, 10 Gold materialized views/streaming tables
- **Infrastructure:** Unity Catalog, Serverless compute, Lakeflow Declarative Pipelines, Databricks Workflows
- **Governance:** 4 role-based groups with schema-level GRANT permissions, column sensitivity tags
- **Analytics:** 4-tab AI/BI dashboard, Genie Space with natural language queries, DQ monitoring

2. Bronze Layer: Raw Ingestion

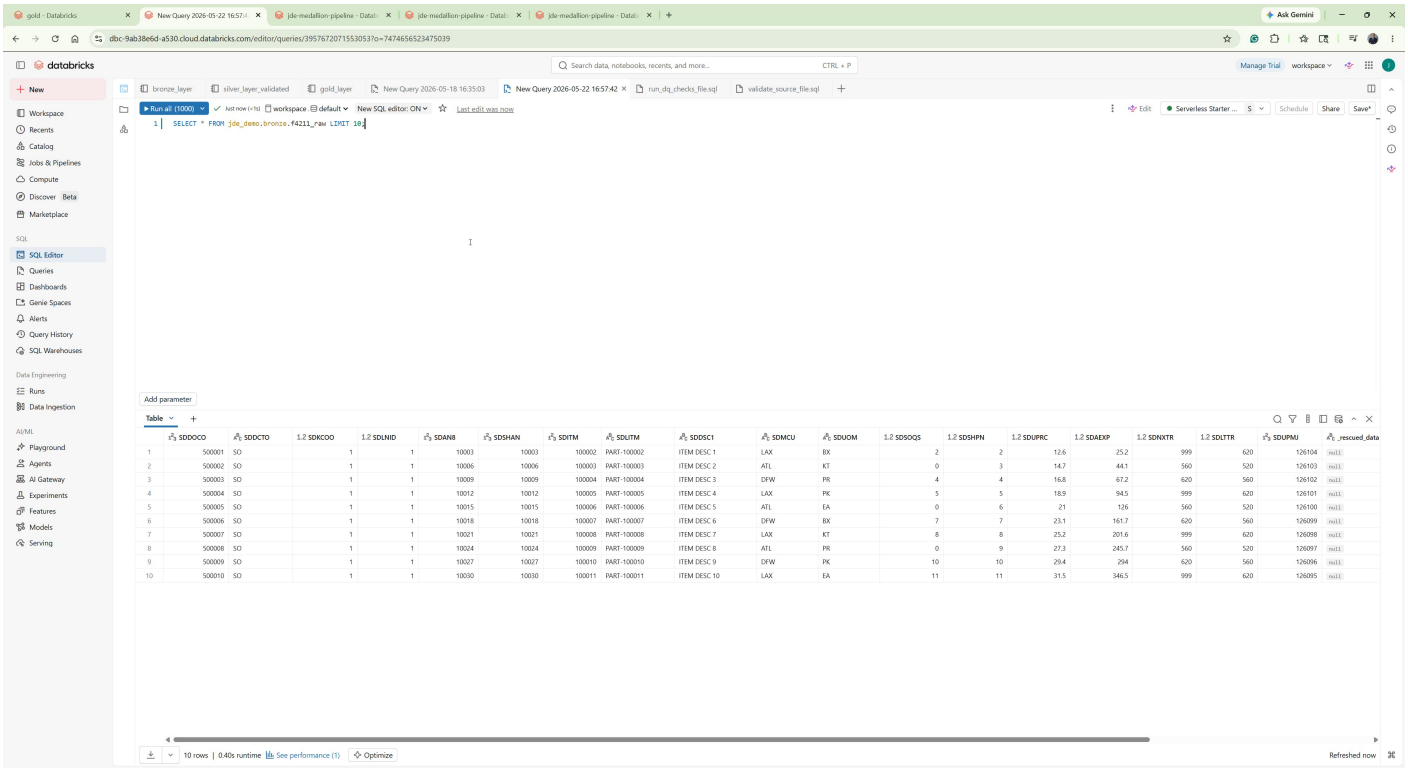
The Bronze layer uses CREATE OR REFRESH STREAMING TABLE statements to ingest CSV files from a Unity Catalog Volume. Each table reads from /Volumes/jde_demo/bronze/raw_files/ using STREAM read_files() with CSV format, header detection, and type inference. Metadata columns (_ingestion_timestamp, _source_file) are added automatically.

Bronze Layer Notebook



Bronze layer notebook: 6 streaming table definitions (F4201, F4211, F0101, F0116, F4101, F03B11) using CREATE OR REFRESH STREAMING TABLE with STREAM read_files(). F4211 expanded showing CSV format, header/inferColumnTypes options.

Bronze Raw Data: F4211 (Sales Order Detail)

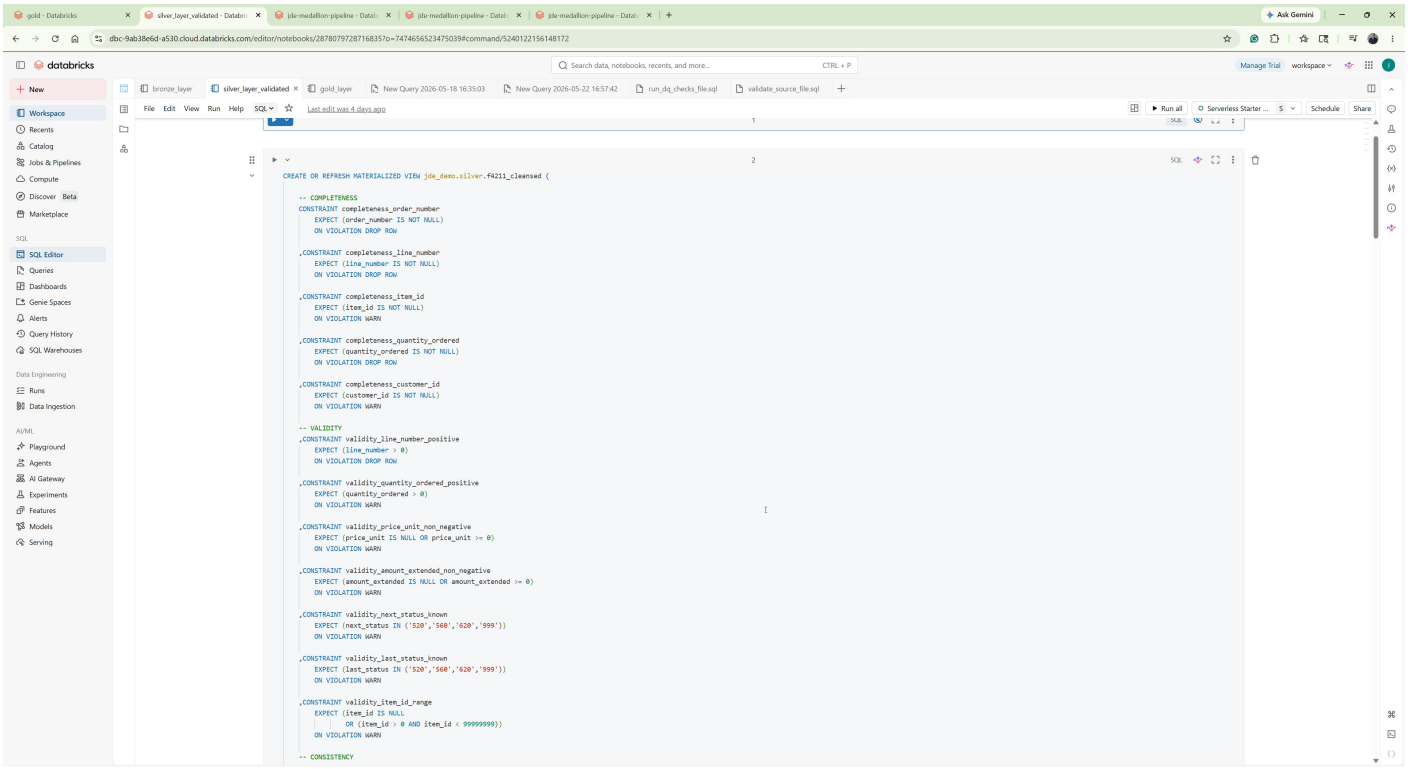


Raw JDE data in Bronze: original column names (SDDOCO, SDDCTO, SDKCOO, SDSHAN, SDITM, SDDSC1) preserved exactly as exported from JD Edwards. 10 rows, 0.40s runtime.

3. Silver Layer: Cleansed + Validated

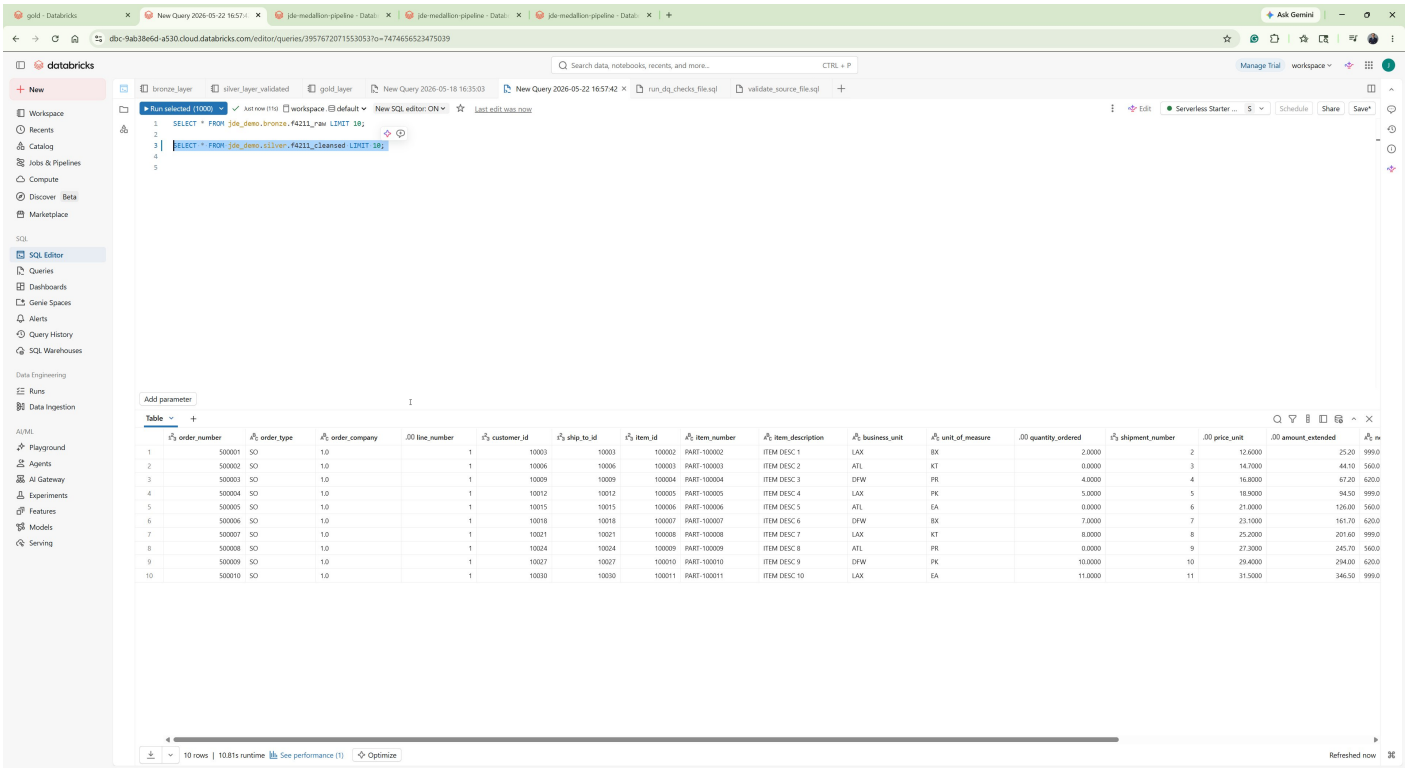
The Silver layer transforms raw JDE column names into business-friendly names, applies DLT Expectations for data quality enforcement, and adds column-level sensitivity tags for governance. Critical fields use ON VIOLATION DROP ROW while non-critical fields use ON VIOLATION WARN.

DLT Expectations: F4211 Cleansed



Silver layer DLT Expectations for F4211: COMPLETENESS constraints (order_number, line_number NOT NULL with DROP ROW; item_id, customer_id with WARN), VALIDITY constraints (positive values, known status codes IN ('520','560','620','999')), item_id range checks.

Silver Cleansed Data: F4211

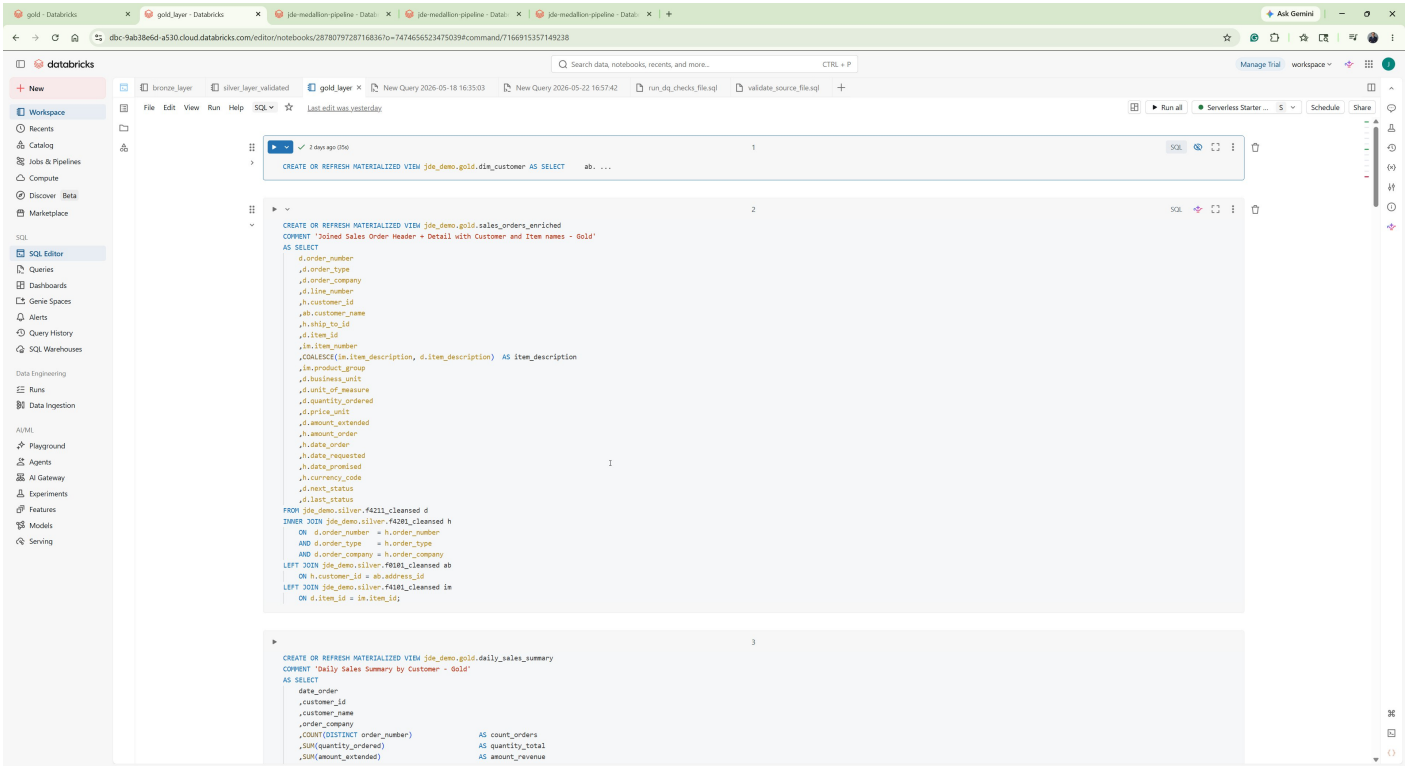


Silver cleansed data: JDE codes transformed to business names (SDDOCO to order_number, SDSHAN to customer_id, SDITM to item_id, SDDSC1 to item_description, SDMCU to business_unit). 10 rows, 10.81s runtime.

4. Gold Layer: Business Models

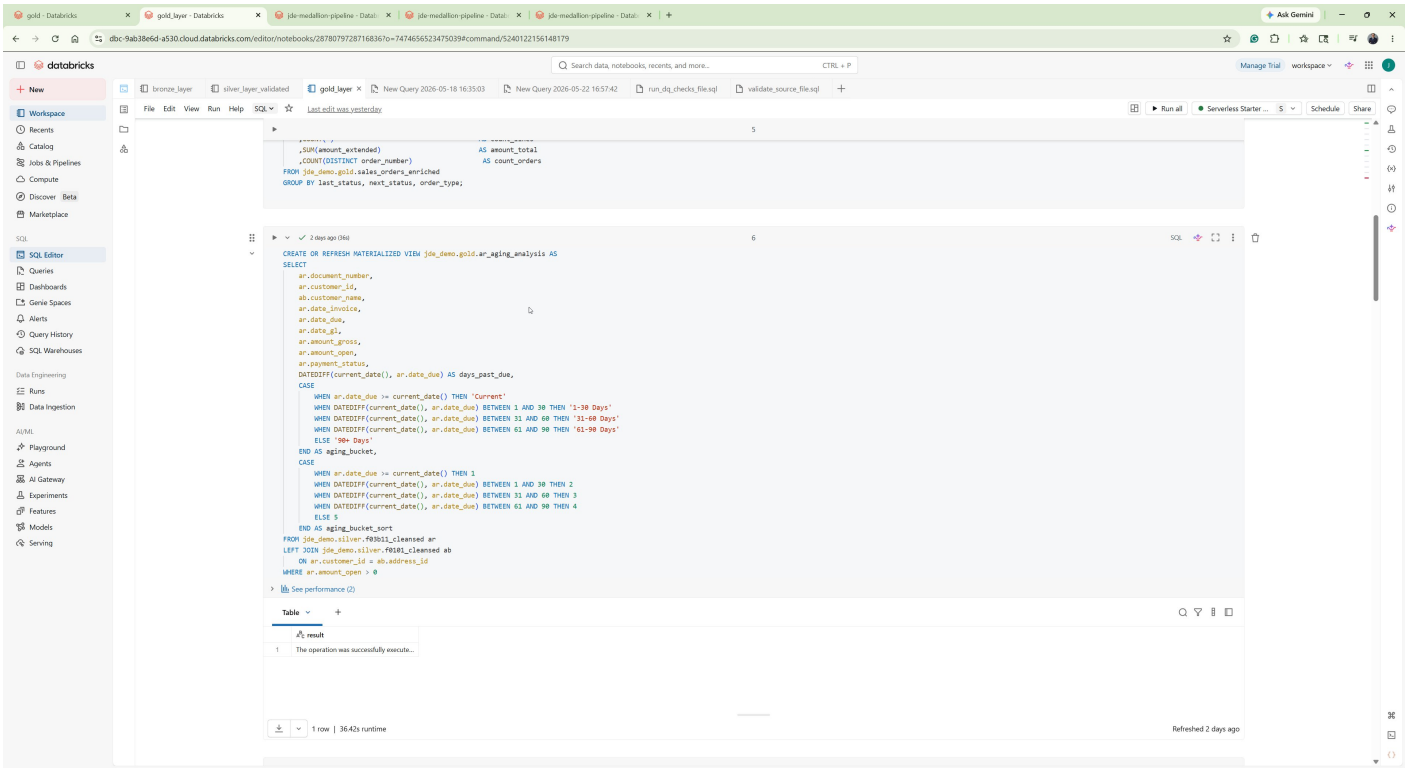
The Gold layer builds business-ready analytical models: denormalized sales orders, daily sales summaries, AR aging analysis, customer dimensions (including SCD Type 2), and data quality monitoring views.

Gold Layer Notebook: Materialized Views



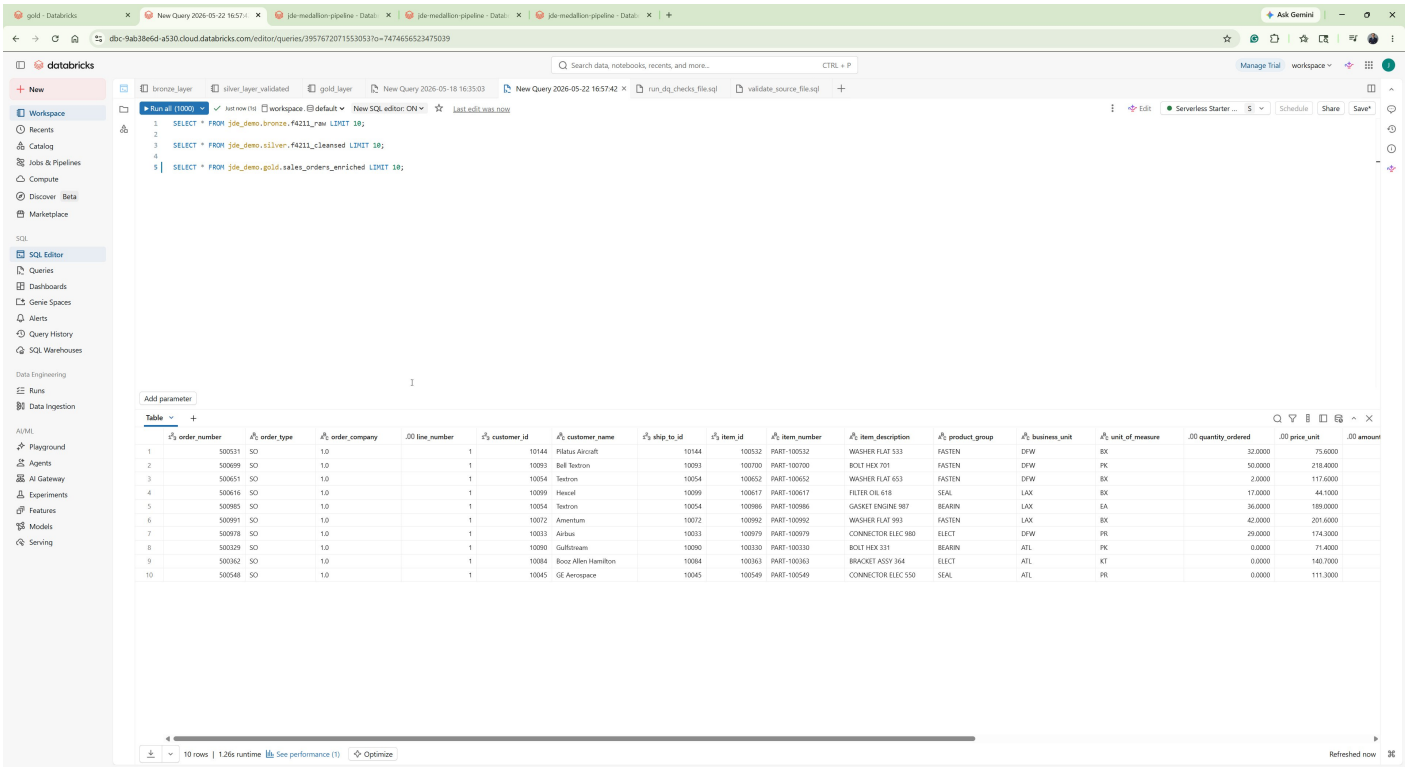
Gold layer notebook: sales_orders_enriched (INNER JOIN F4211 to F4201 on order_number/type/company, LEFT JOIN F0101 for customer names, LEFT JOIN F4101 for item details, COALESCE for item_description). daily_sales_summary with COUNT(DISTINCT), SUM aggregations by customer and date.

AR Aging Analysis (Materialized View SQL)



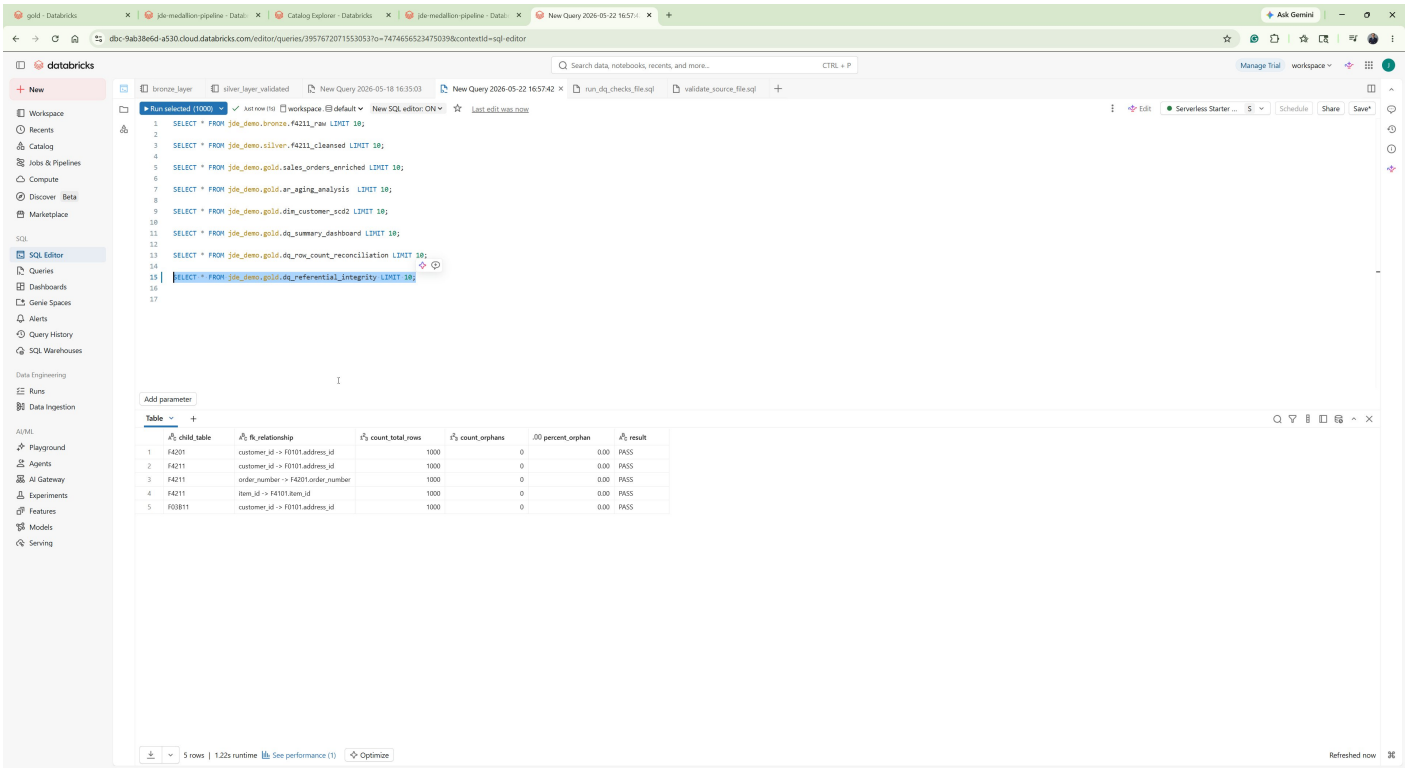
Gold layer: ar_aging_analysis materialized view with DATEDIFF aging buckets (Current, 1-30, 31-60, 61-90, 90+ Days), aging_bucket_sort for ordering, joining silver.f03b11_cleansed to silver.f0101_cleansed. WHERE amount_open > 0. Successfully executed in 36.42s.

Sales Orders Enriched: Query Results



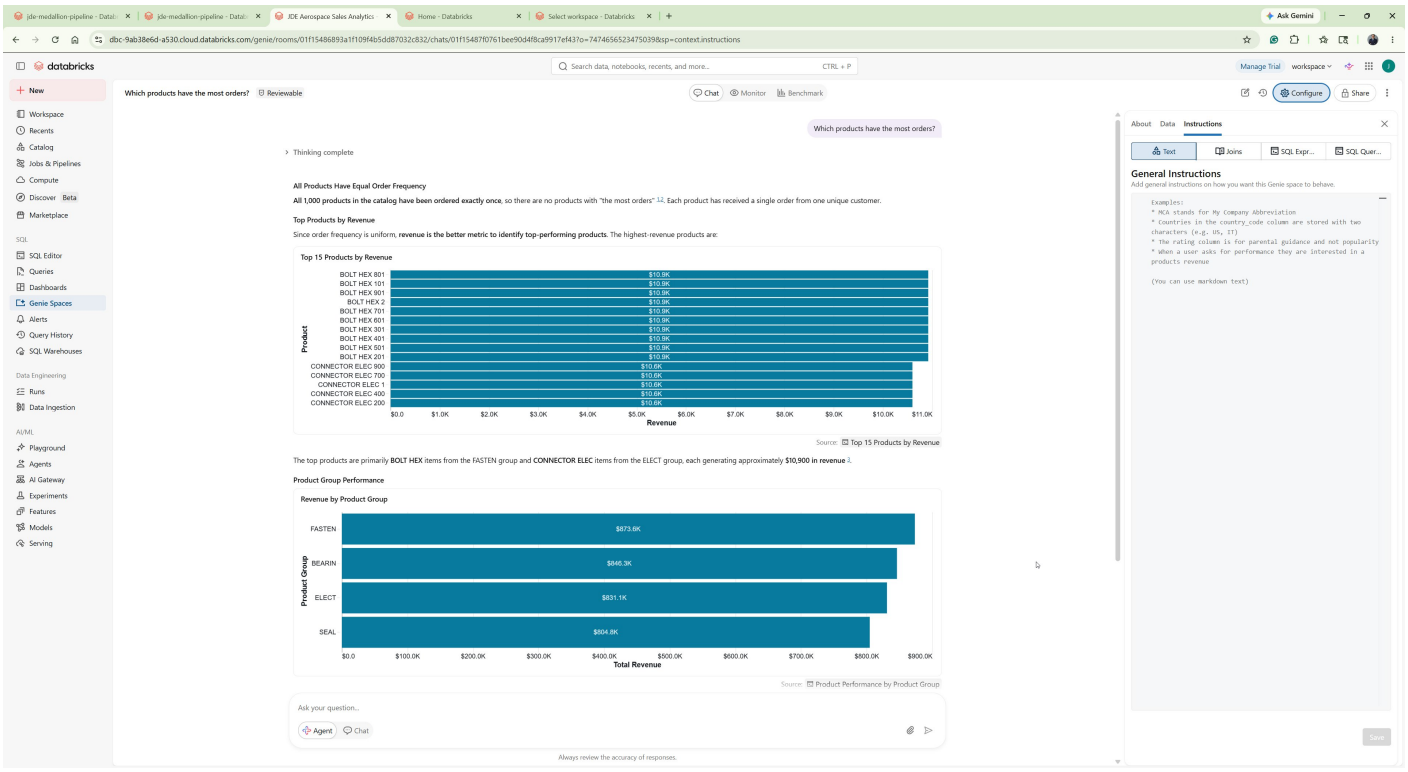
Gold: sales_orders_enriched query results showing denormalized data with customer names (Pilatus Aircraft, Bell Textron, Hexcel, Amentum, Airbus, Gulfstream, Booz Allen Hamilton, GE Aerospace), product groups (FASTEN, SEAL, BEARIN, ELECT), business units (DFW, LAX, ATL).

AR Aging Analysis: Query Results



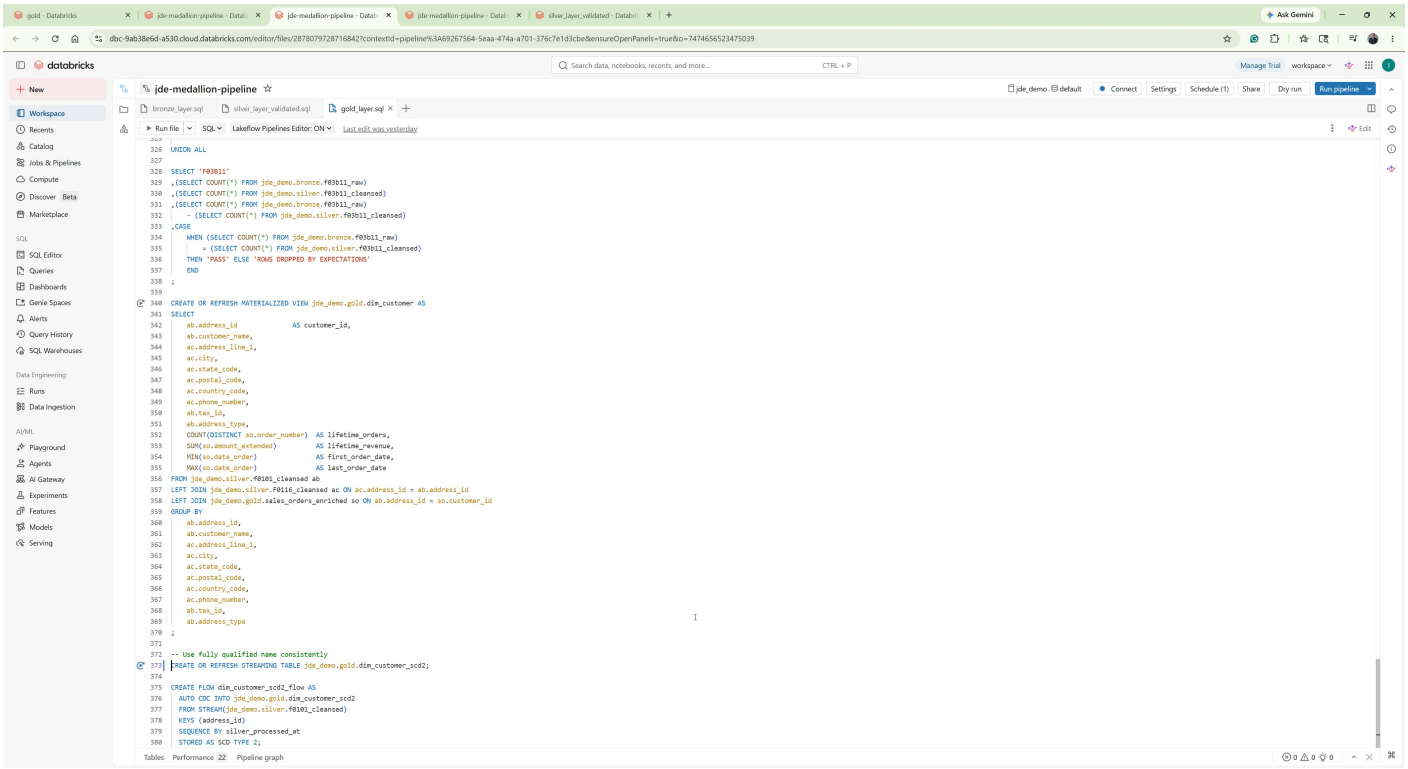
Gold: ar_aging_analysis results for Boeing (customer_id 10021) showing invoice dates, due dates, GL dates, gross/open amounts, payment_status (H=Hold, O=Open), days_past_due calculations, and aging_bucket assignments (Current, 1-30 Days).

Customer Dimension (SCD Type 2)



Gold: dim_customer_scd2 with surrogate address_key, address_type (V=Vendor, E=Employee, C=Customer), customer names (Ducommun, Lockheed Martin, Kaman Aerospace, SpaceX, Curtiss-Wright, RTX), _START_AT/_END_AT timestamps for SCD Type 2 history tracking.

SCD Type 2 Pipeline Code



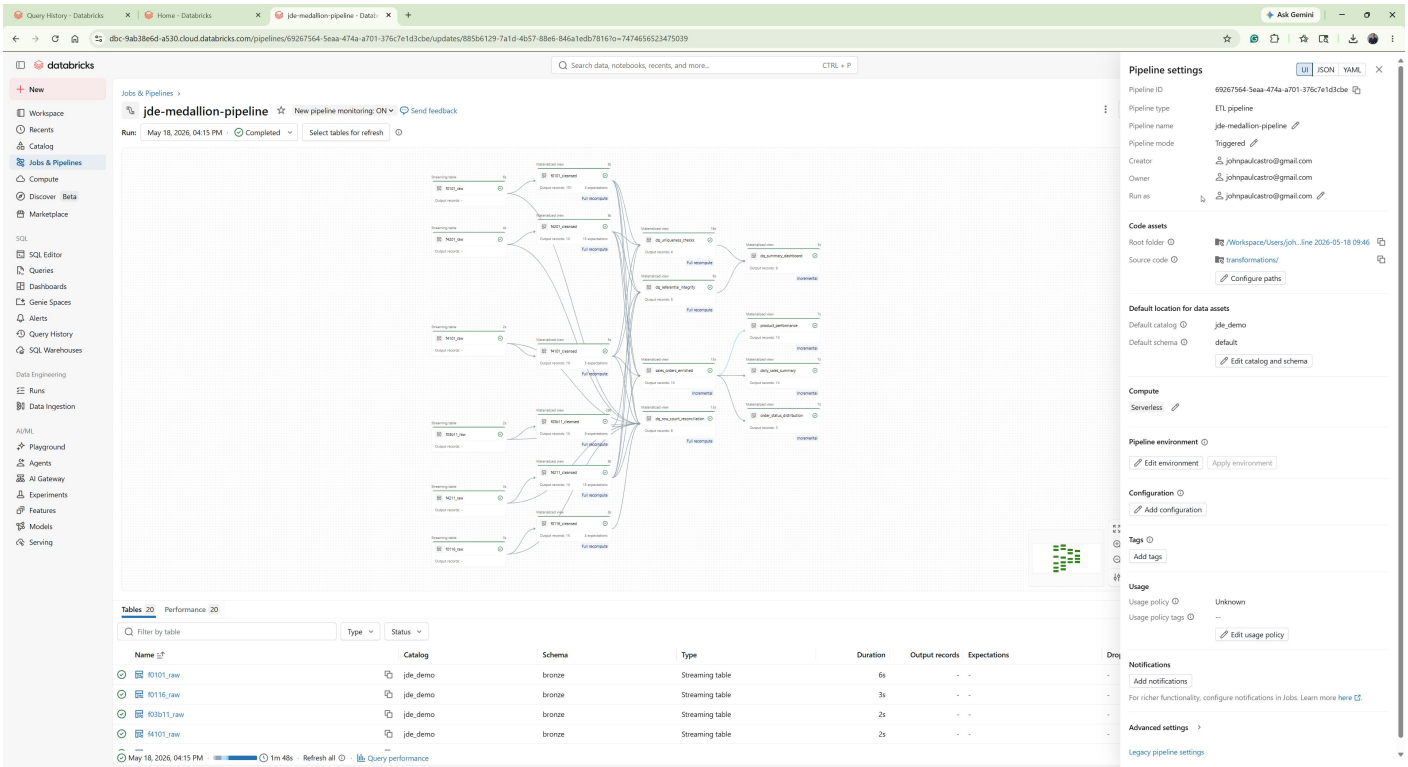
```
UNION ALL

SELECT 'F03B11'
,(SELECT COUNT(*) FROM jde_demo.bronze.f03b11_raw)
,(SELECT COUNT(*) FROM jde_demo.silver.f03b11_cleansed)
,(SELECT COUNT(*) FROM jde_demo.bronze.f03b11_raw)
  - (SELECT COUNT(*) FROM jde_demo.silver.f03b11_cleansed)
,CASE
    WHEN (SELECT COUNT(*) FROM jde_demo.bronze.f03b11_raw)
       = (SELECT COUNT(*) FROM jde_demo.silver.f03b11_cleansed)
    THEN 'PASS' ELSE 'ROWS DROPPED BY EXPECTATIONS'
    END
;

CREATE OR REFRESH MATERIALIZED VIEW jde_demo.gold.dim_customer AS
SELECT
    ab.address_id             AS customer_id,
    ab.customer_name,
    ac.address_line_1,
    ac.city,
    ac.state_code,
    ac.postal_code,
    ac.country_code,
    ac.phone_number,
    ab.tax_id,
    ab.address_type,
    COUNT(DISTINCT so.order_number)  AS lifetime_orders,
    SUM(so.amount_extended)          AS lifetime_revenue,
    MIN(so.date_order)               AS first_order_date,
    MAX(so.date_order)               AS last_order_date
FROM jde_demo.silver.f0101_cleansed ab
LEFT JOIN jde_demo.silver.f0116_cleansed ac ON ac.address_id = ab.address_id
LEFT JOIN jde_demo.gold.sales_orders_enriched so ON ab.address_id = so.customer_id
GROUP BY
    ab.address_id,
    ab.customer_name,
    ac.address_line_1,
    ac.city,
    ac.state_code,
    ac.postal_code,
    ac.country_code,
    ac.phone_number,
    ab.tax_id,
    ab.address_type
;

-- Use fully qualified name consistently
CREATE OR REFRESH STREAMING TABLE jde_demo.gold.dim_customer_scd2;

CREATE FLOW dim_customer_scd2_flow AS
  AUTO CDC INTO jde_demo.gold.dim_customer_scd2
  FROM STREAM(jde_demo.silver.f0101_cleansed)
  KEYS (address_id)
  SEQUENCE BY silver_processed_at
  STORED AS SCD TYPE 2;
```

Pipeline editor: dim_customer materialized view (lifetime_orders, lifetime_revenue, first/last_order_date), dim_customer_scd2 streaming table with CREATE FLOW using AUTO CDC INTO dim_customer_scd2, FROM STREAM(silver.f0101_cleansed), KEYS(address_id), SEQUENCE BY silver_processed_at, STORED AS SCD TYPE 2.

5. Lakeflow Declarative Pipeline (DLT)

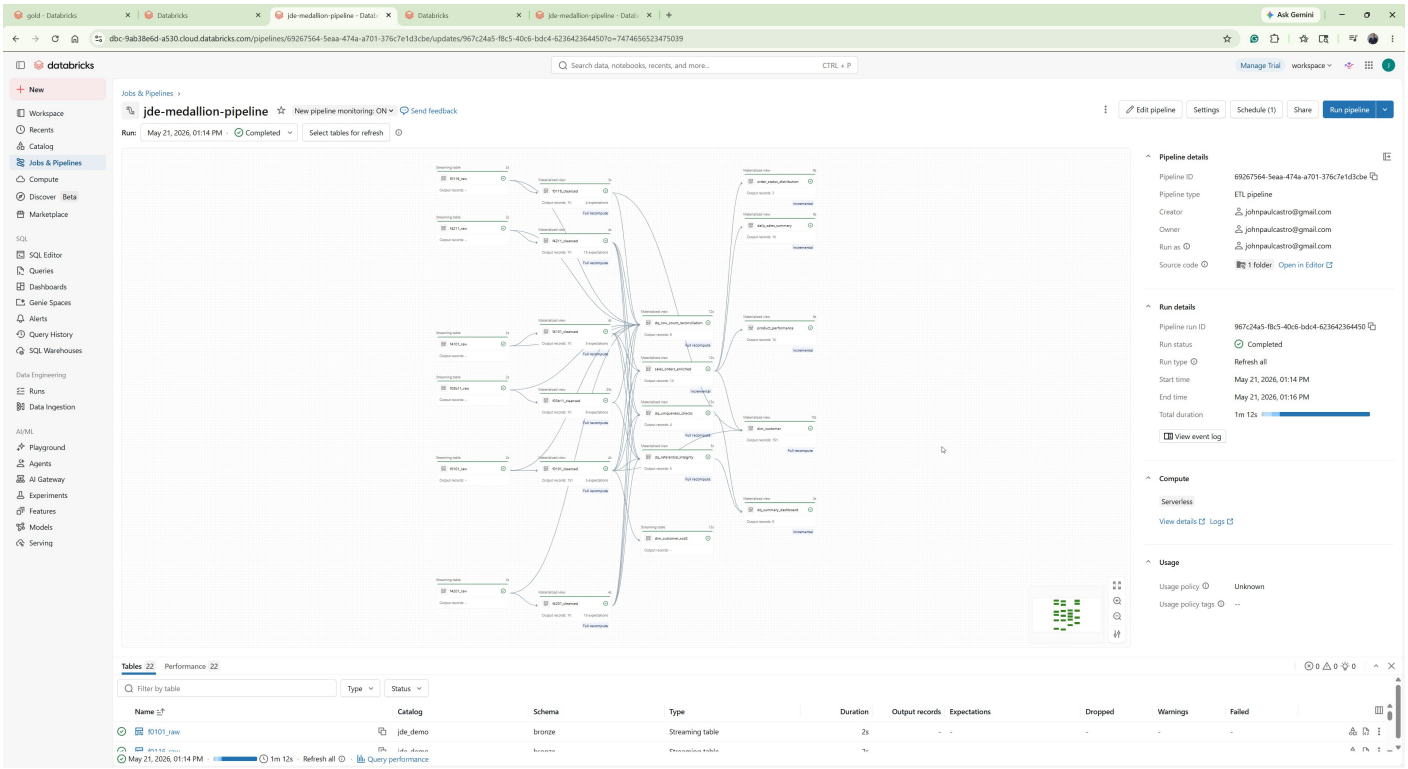
The jde-medallion-pipeline is an ETL pipeline running on Serverless compute in Triggered mode. It processes 22 tables across Bronze, Silver, and Gold layers in a single declarative pipeline with full lineage tracking. Pipeline code is organized in three SQL files: bronze_layer.sql, silver_layer_validated.sql, and gold_layer.sql.

Pipeline DAG: May 18 Run (Settings View)



jde-medallion-pipeline: Completed run (May 18, 1m 48s), 20 tables. Pipeline Settings showing ETL type, Triggered mode, Serverless compute, jde_demo catalog. Table list: f0101_raw (6s), f0116_raw (3s), f03b11_raw (2s), f4101_raw (2s) as streaming tables.

Pipeline DAG: May 21 Run (Details View)

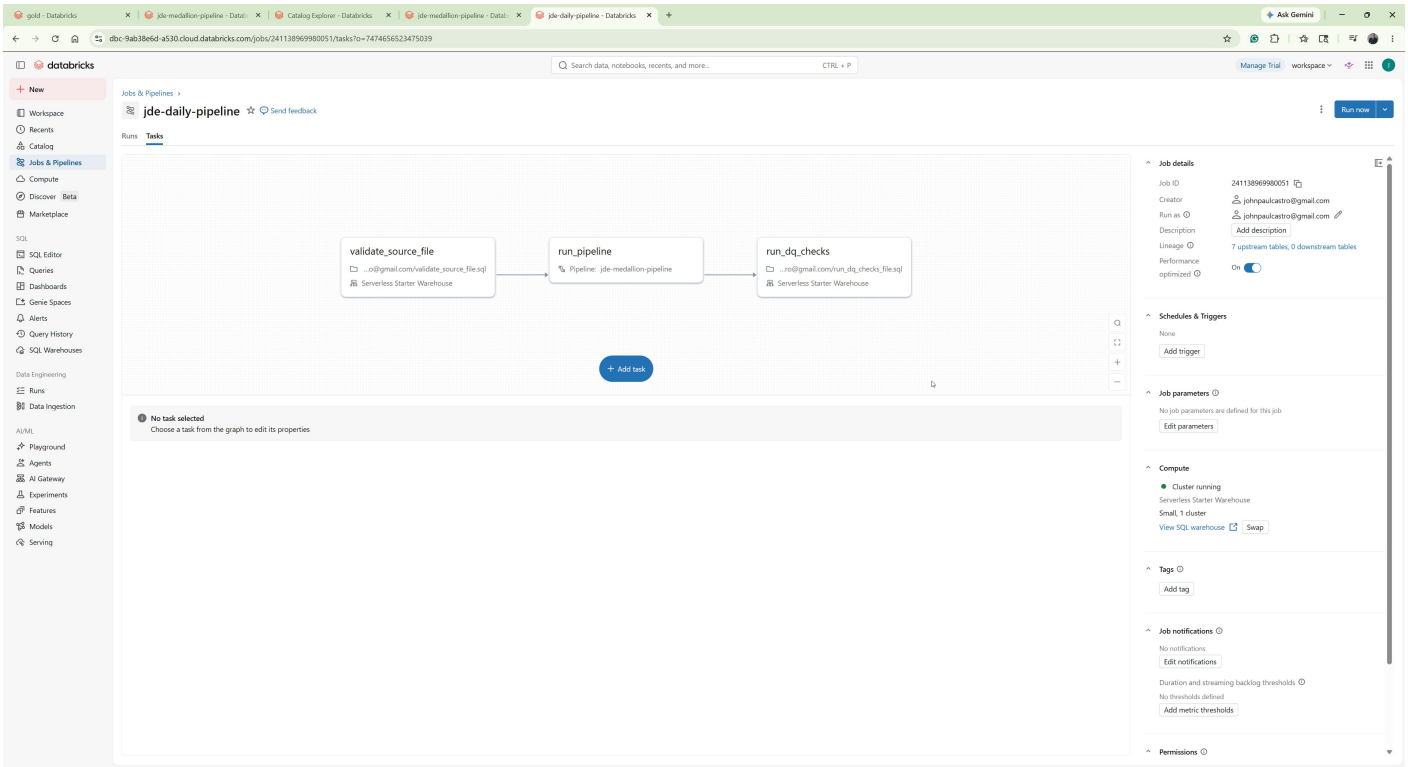


jde-medallion-pipeline: Completed Refresh All run (May 21, 1m 12s), 22 tables. Run details showing Pipeline run ID, Serverless compute, source code in transformations/ folder, full DAG visualization with all Bronze/Silver/Gold nodes.

6. Workflow Orchestration

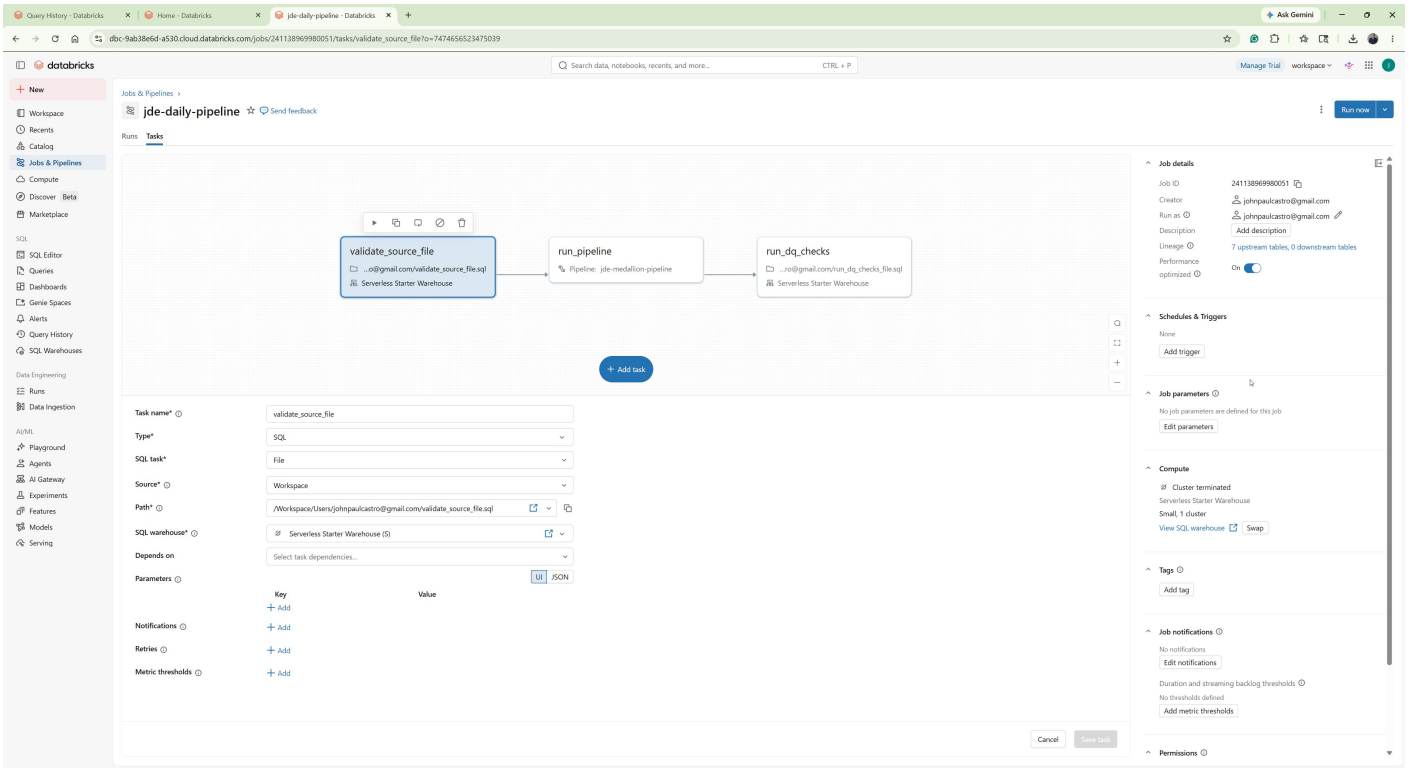
The jde-daily-pipeline workflow orchestrates the full data pipeline as a 3-task DAG: validate_source_file (SQL) validates Bronze data exists, run_pipeline (Pipeline) triggers the DLT pipeline, and run_dq_checks (SQL) runs data quality checks. Each task depends on the previous one succeeding. Performance optimization is enabled.

Workflow DAG: Overview



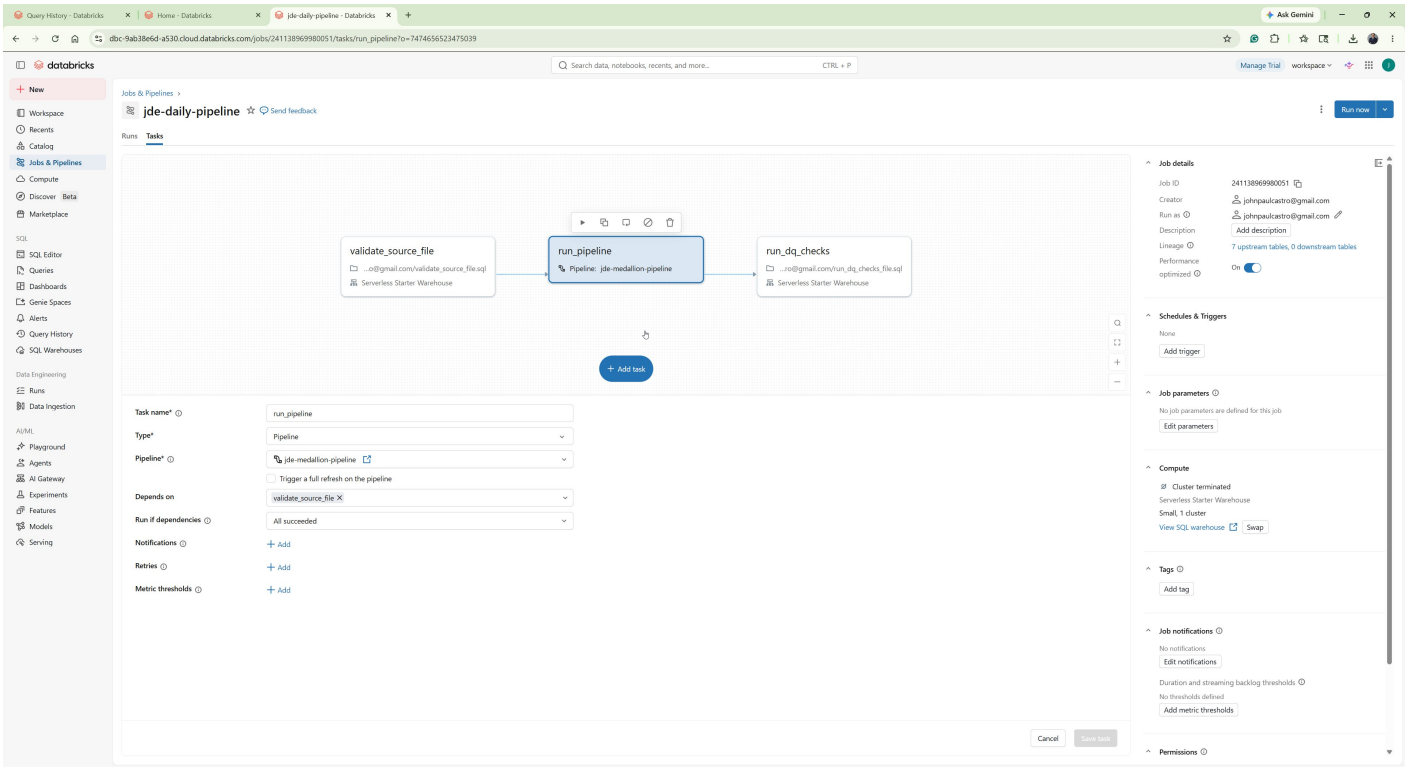
jde-daily-pipeline: 3-task DAG (validate_source_file > run_pipeline > run_dq_checks). Cluster running, 7 upstream tables, performance optimized ON. Job ID visible, Serverless Starter Warehouse.

Task 1: validate_source_file



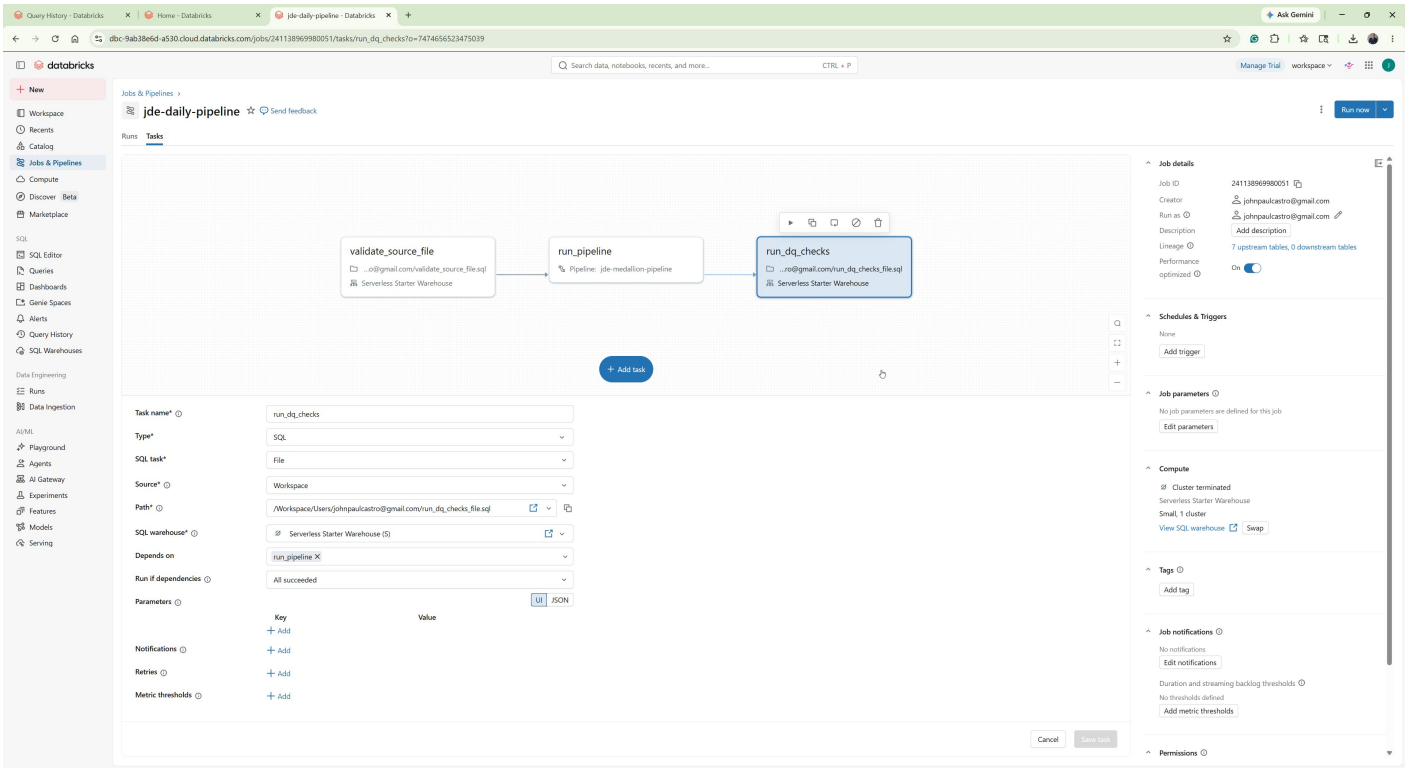
Task 1 detail: SQL type, File source from /Workspace/Users/johnpaulcastro@gmail.com/validate_source_file.sql, Serverless Starter Warehouse (S), no upstream dependencies.

Task 2: run_pipeline



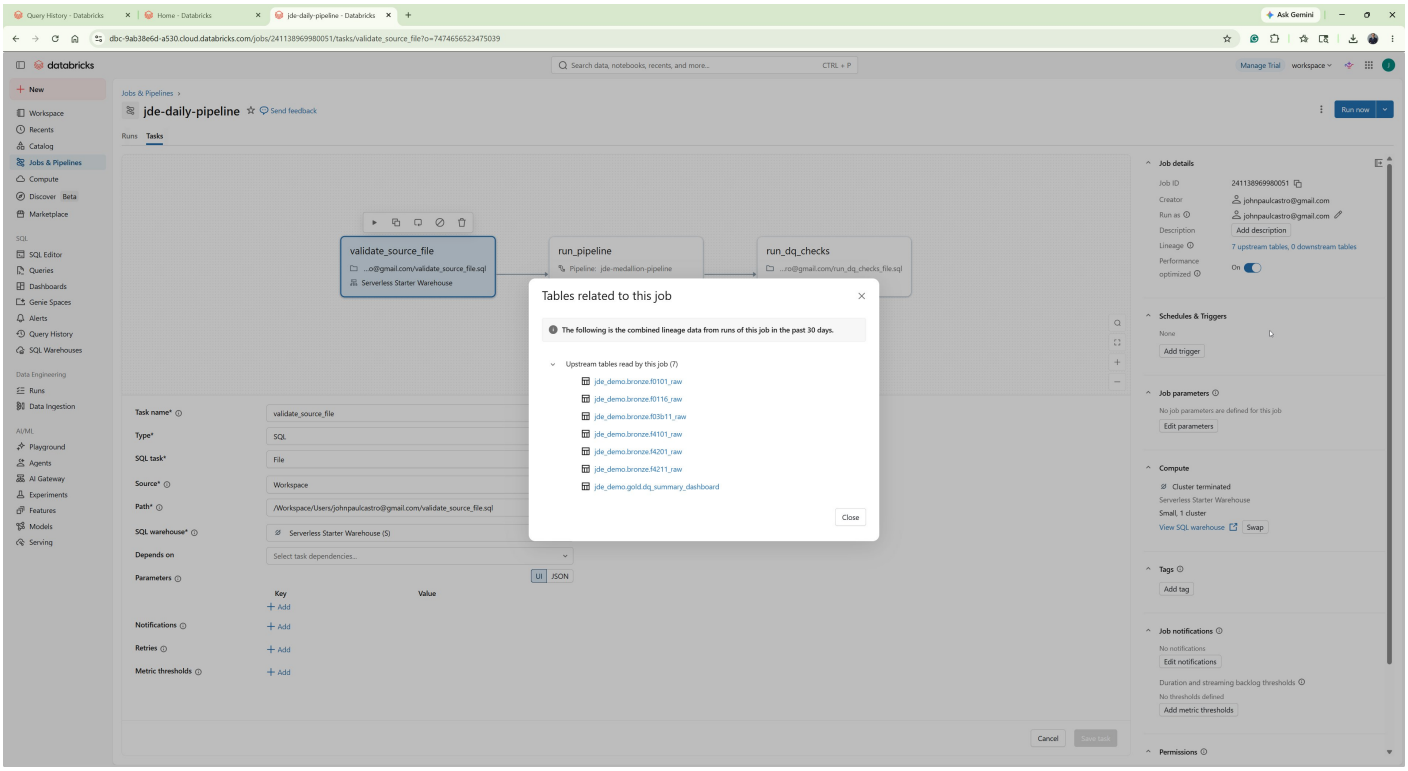
Task 2 detail: Pipeline type, triggers jde-medallion-pipeline, depends on validate_source_file (All succeeded). Option to trigger full refresh on pipeline.

Task 3: run_dq_checks



Task 3 detail: SQL type, File source from run_dq_checks_file.sql, depends on run_pipeline (All succeeded). Final step validates data quality after pipeline completion.

Workflow Lineage: Upstream Tables

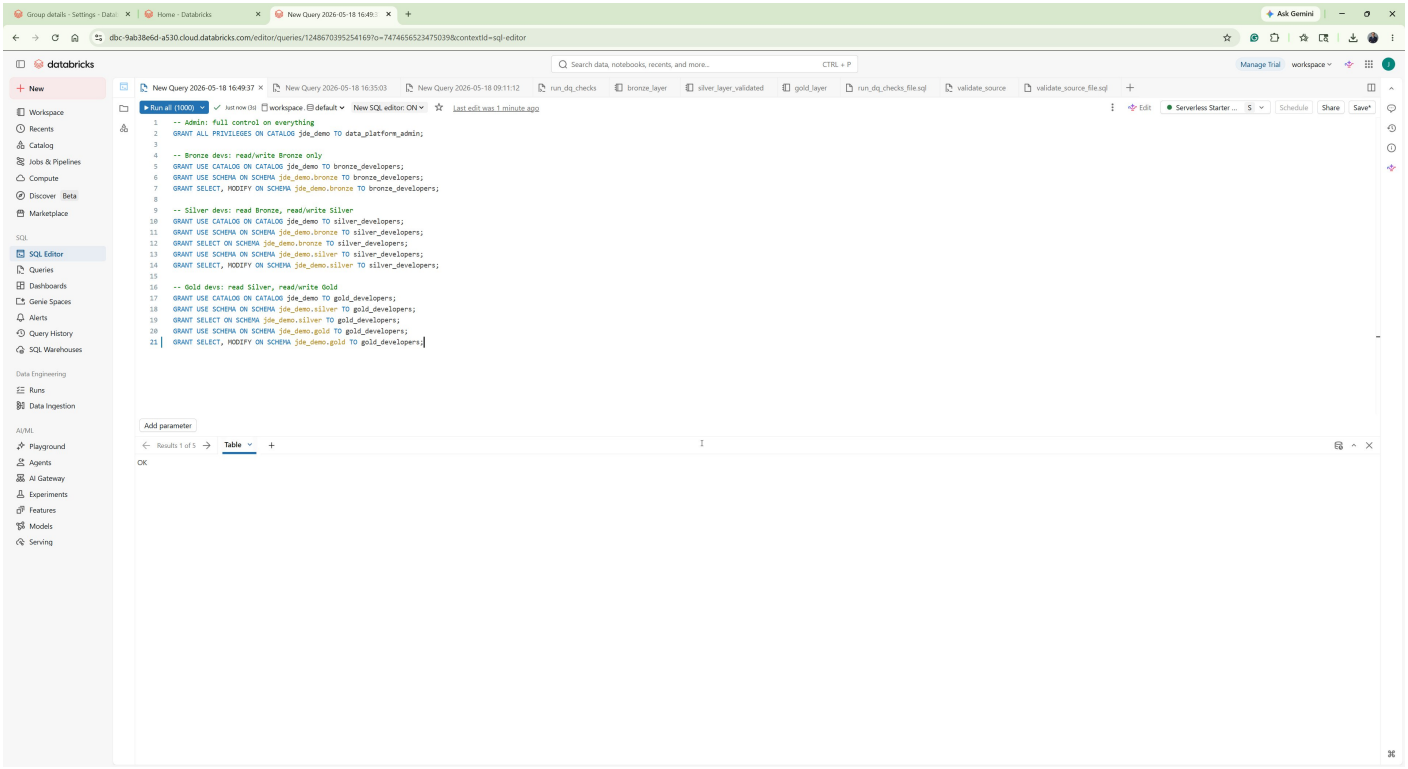


Tables related to this job: 7 upstream tables from combined lineage over 30 days. jde_demo.bronze (f0101_raw, f0116_raw, f03b11_raw, f4101_raw, f4201_raw, f4211_raw) plus jde_demo.gold.dq_summary_dashboard.

7. Unity Catalog RBAC

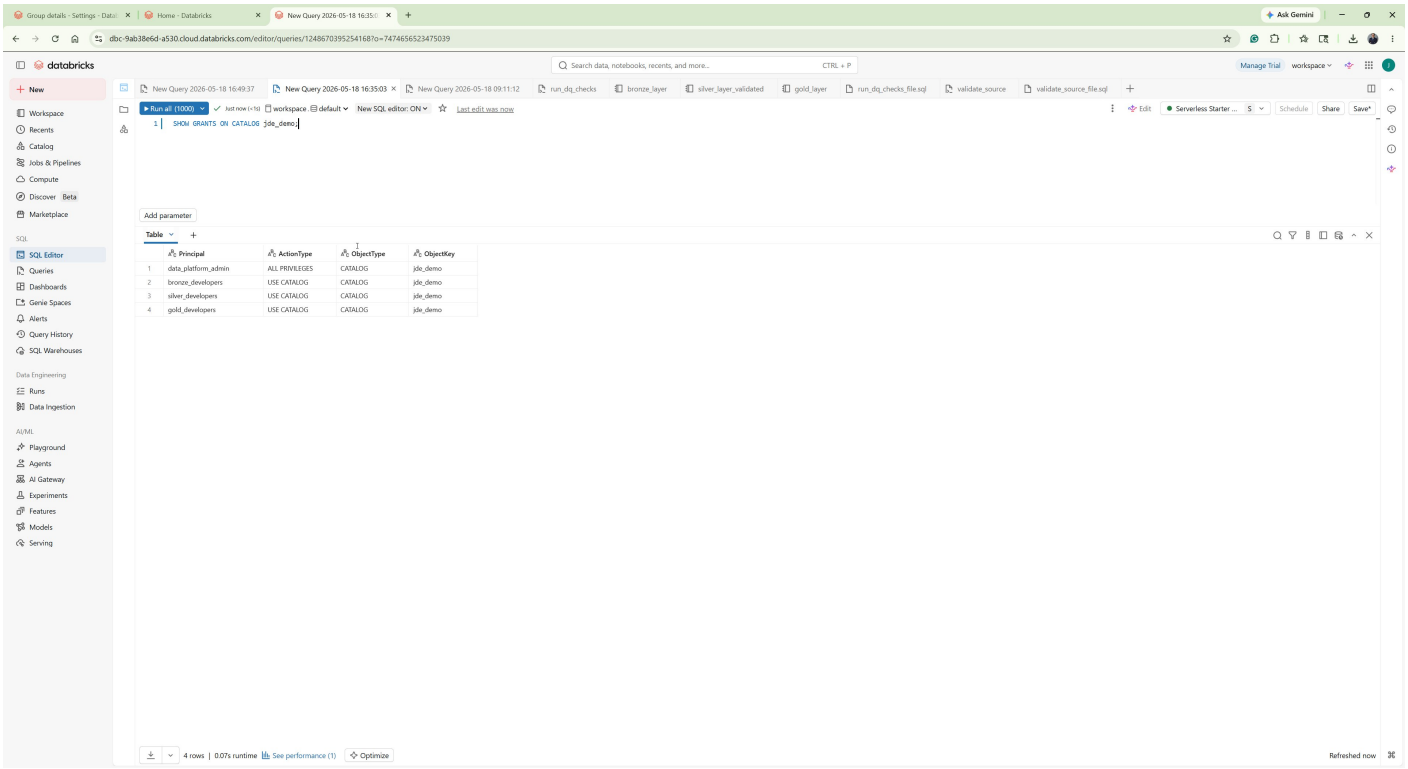
Role-based access control is implemented through four Unity Catalog groups with schema-level permissions following a progressive access model: Bronze developers can only modify Bronze, Silver developers can read Bronze and modify Silver, Gold developers can read Silver and modify Gold, and data_platform_admin has full control.

GRANT Statements: Full RBAC Script



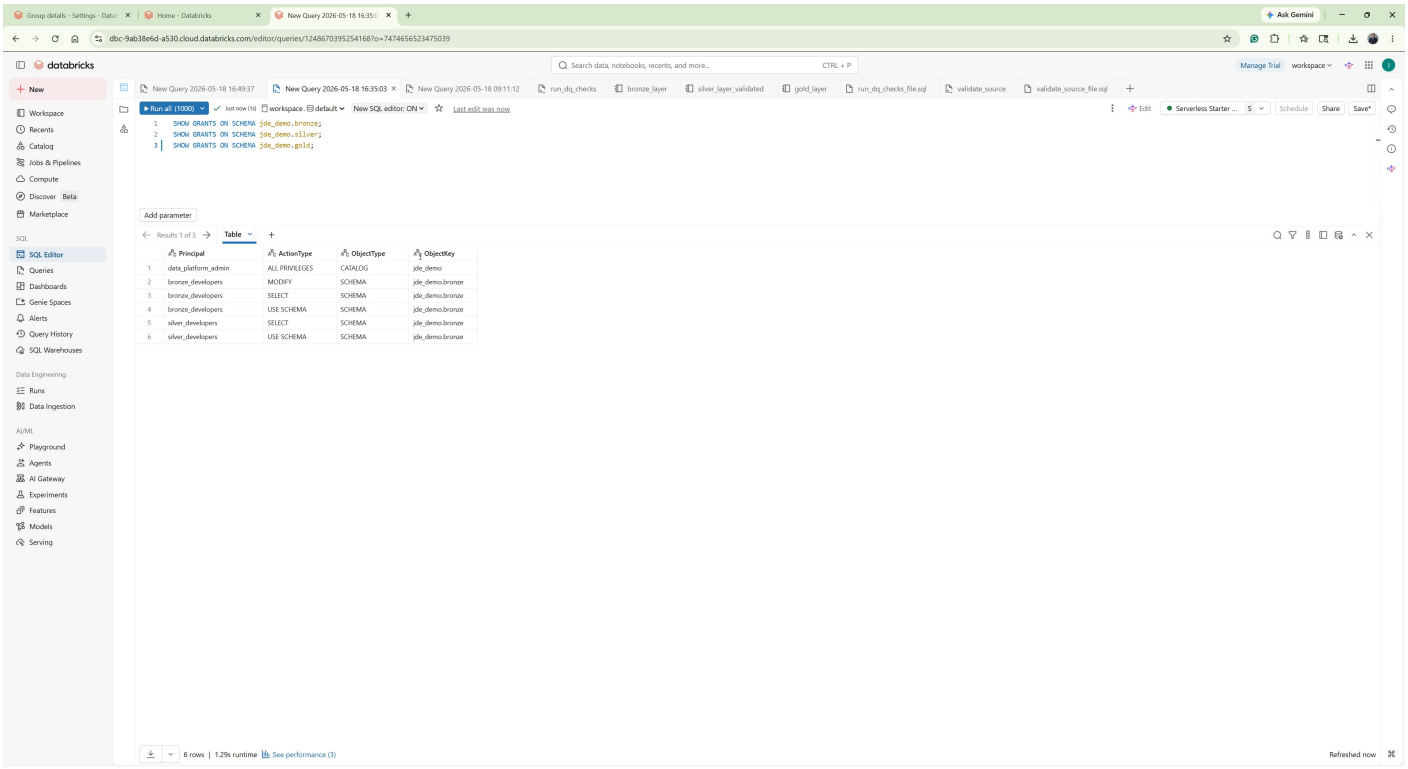
RBAC GRANT script: data_platform_admin gets ALL PRIVILEGES ON CATALOG. Bronze devs get read/write Bronze only. Silver devs get read Bronze + read/write Silver. Gold devs get read Silver + read/write Gold. Result: OK.

SHOW GRANTS ON CATALOG: jde_demo



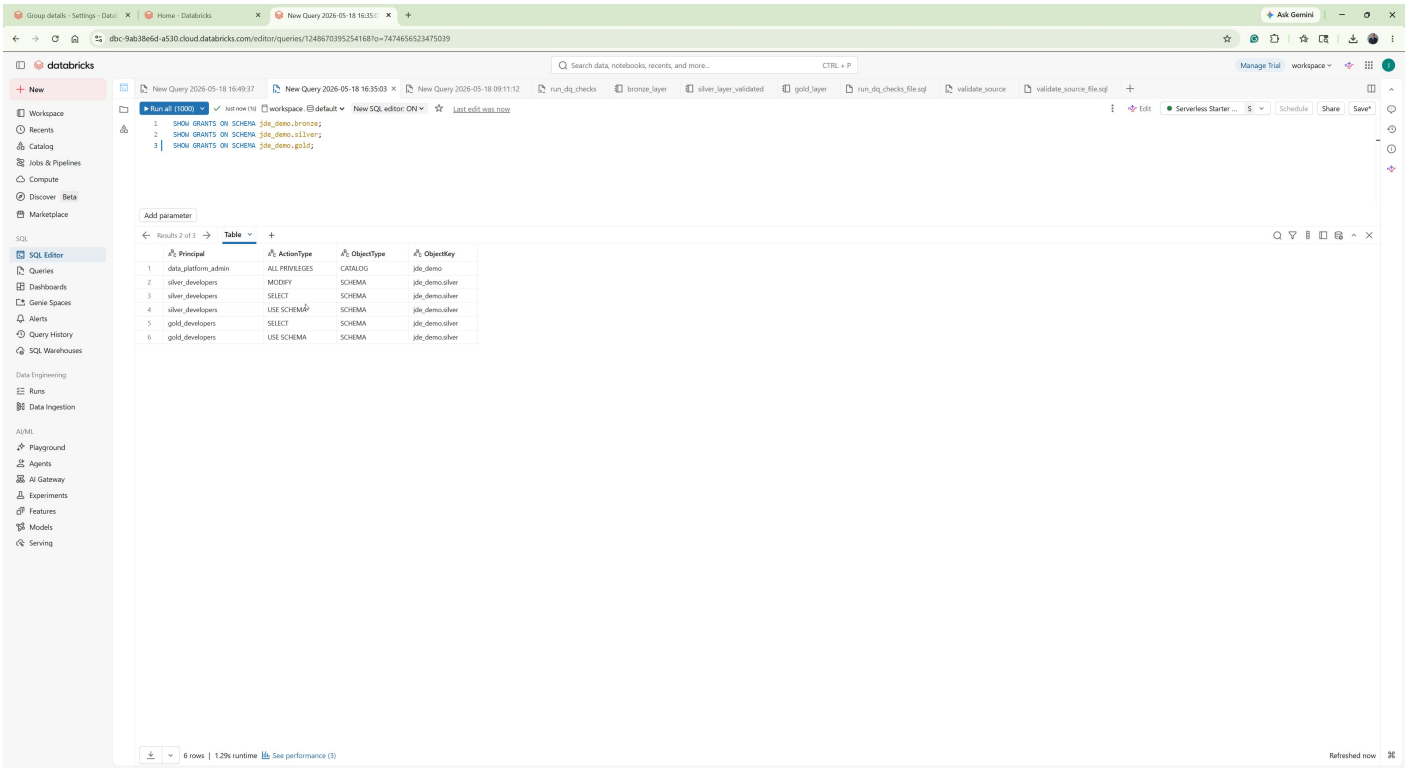
Catalog-level grants: data_platform_admin has ALL PRIVILEGES, bronze_developers, silver_developers, and gold_developers each have USE CATALOG on jde_demo.

SHOW GRANTS ON SCHEMA: Bronze



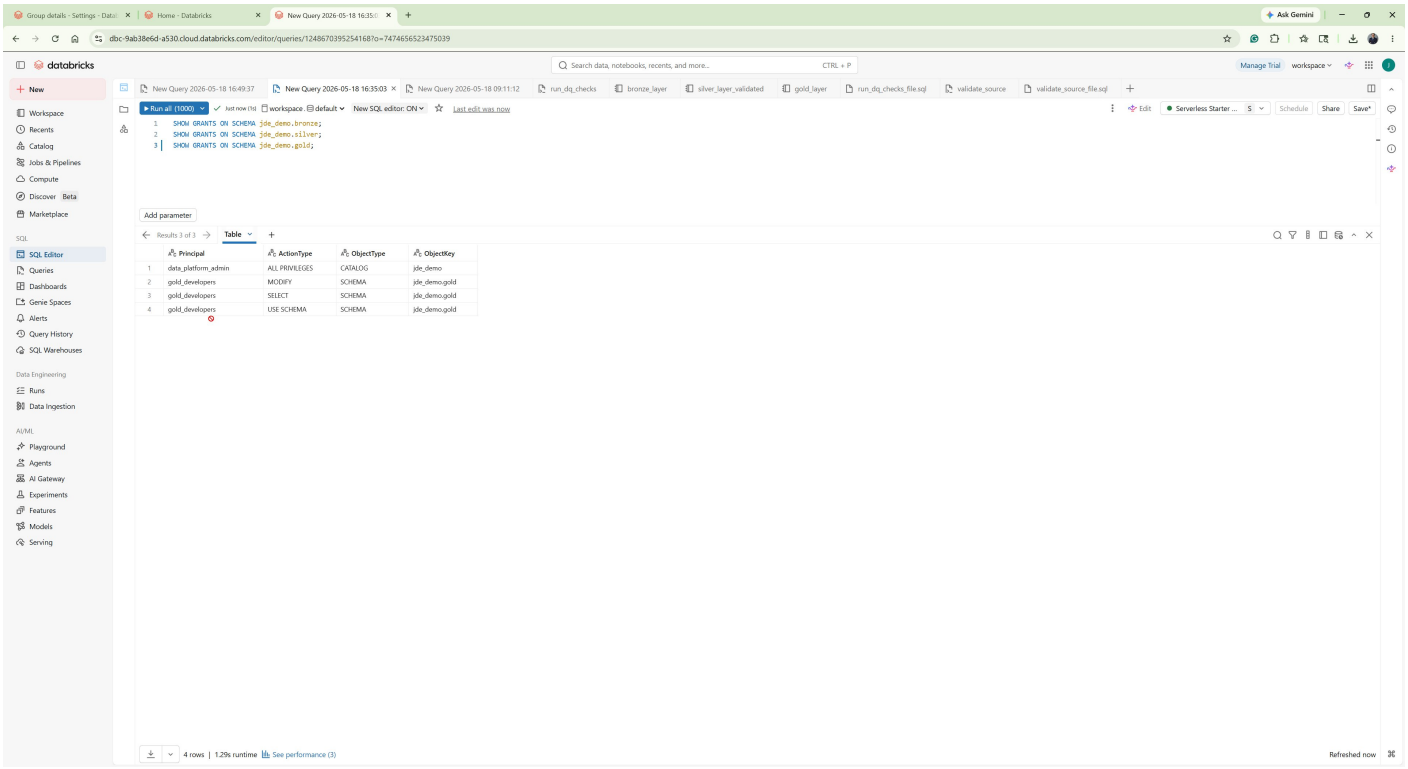
Bronze schema grants: bronze_developers have MODIFY/SELECT/USE SCHEMA on jde_demo.bronze. silver_developers have SELECT/USE SCHEMA (read-only downstream access).

SHOW GRANTS ON SCHEMA: Silver



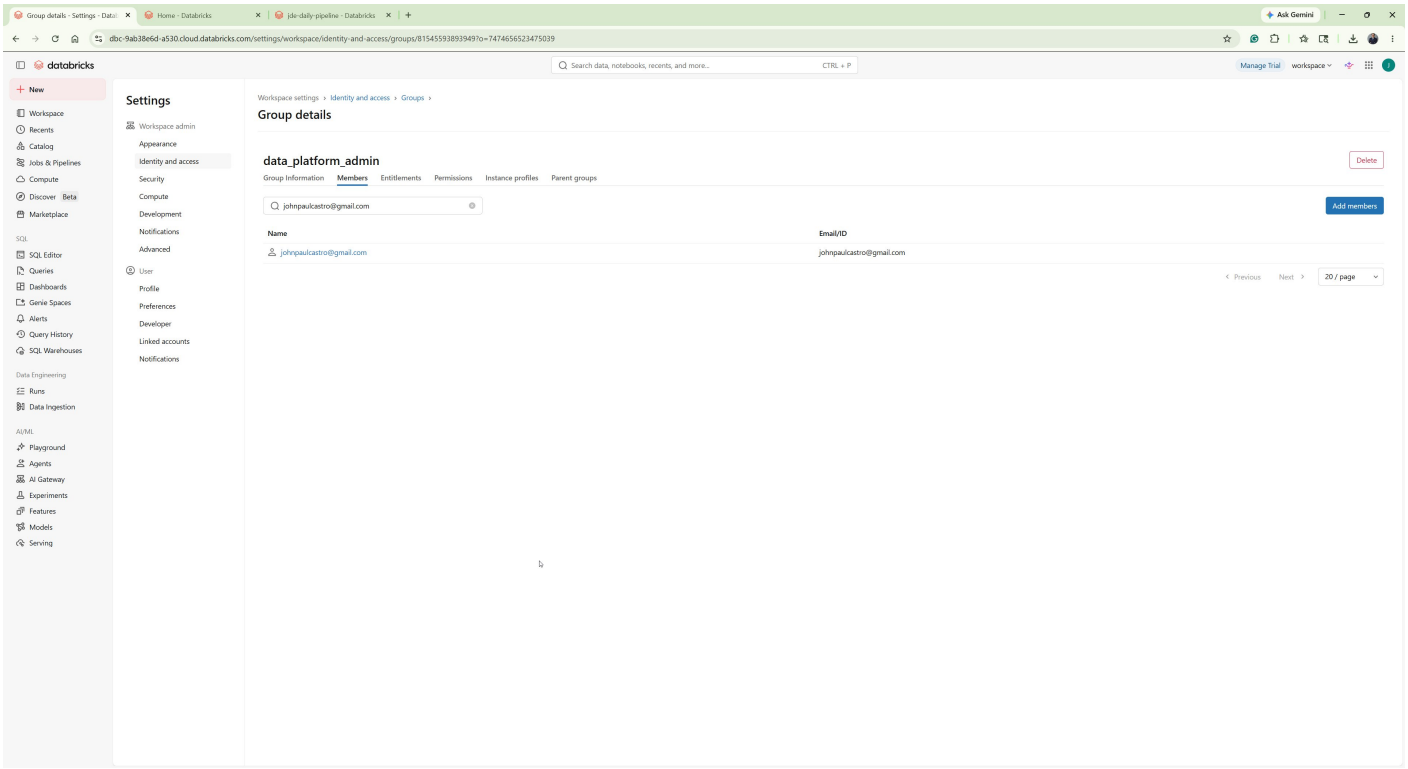
Silver schema grants: silver_developers have MODIFY/SELECT/USE SCHEMA on jde_demo.silver. gold_developers have SELECT/USE SCHEMA on jde_demo.silver (read-only downstream).

SHOW GRANTS ON SCHEMA: Gold



Gold schema grants: gold_developers have MODIFY/SELECT/USE SCHEMA on jde_demo.gold. data_platform_admin retains ALL PRIVILEGES at the catalog level.

Group Details: data_platform_admin

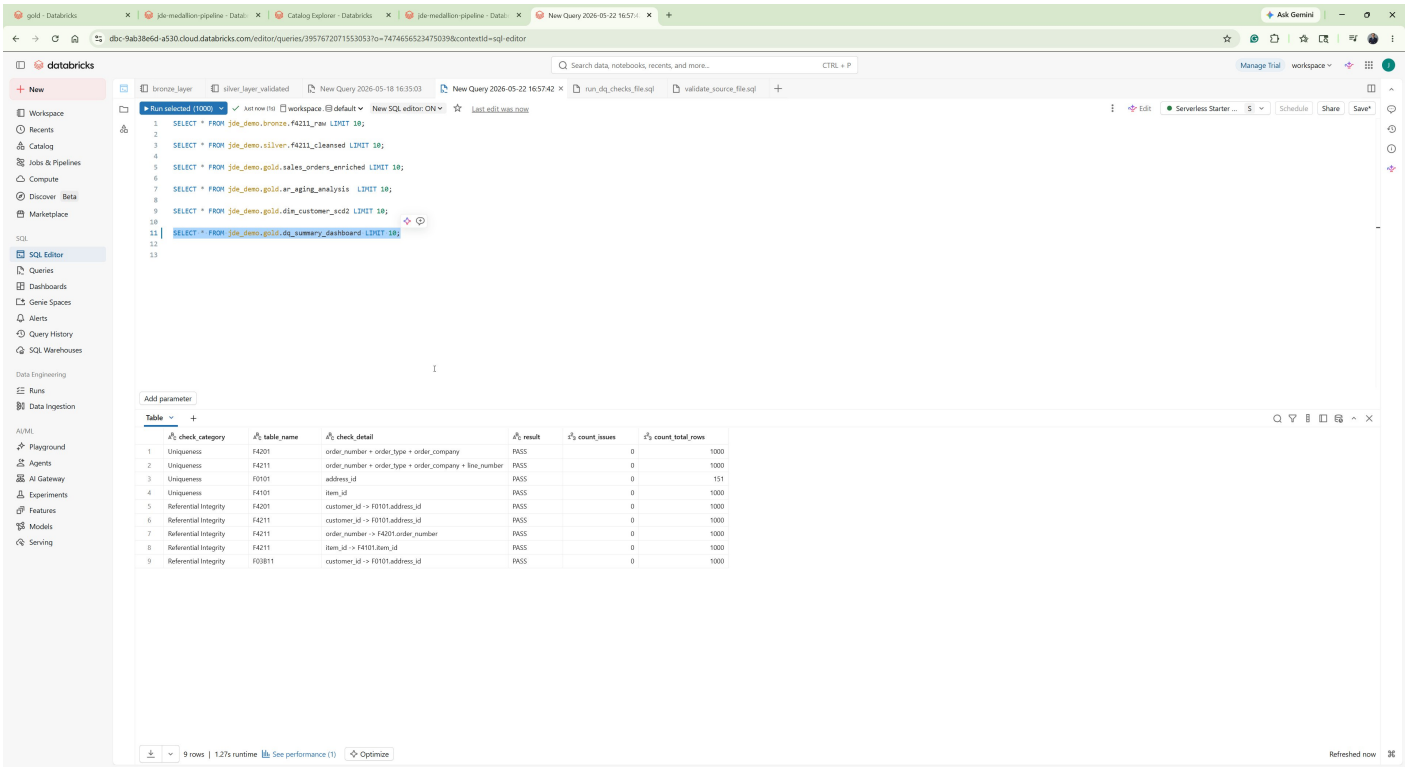


Identity and Access settings: data_platform_admin group membership showing johnpaulcastro@gmail.com. Workspace admin > Identity and access > Groups path.

8. Data Quality Framework

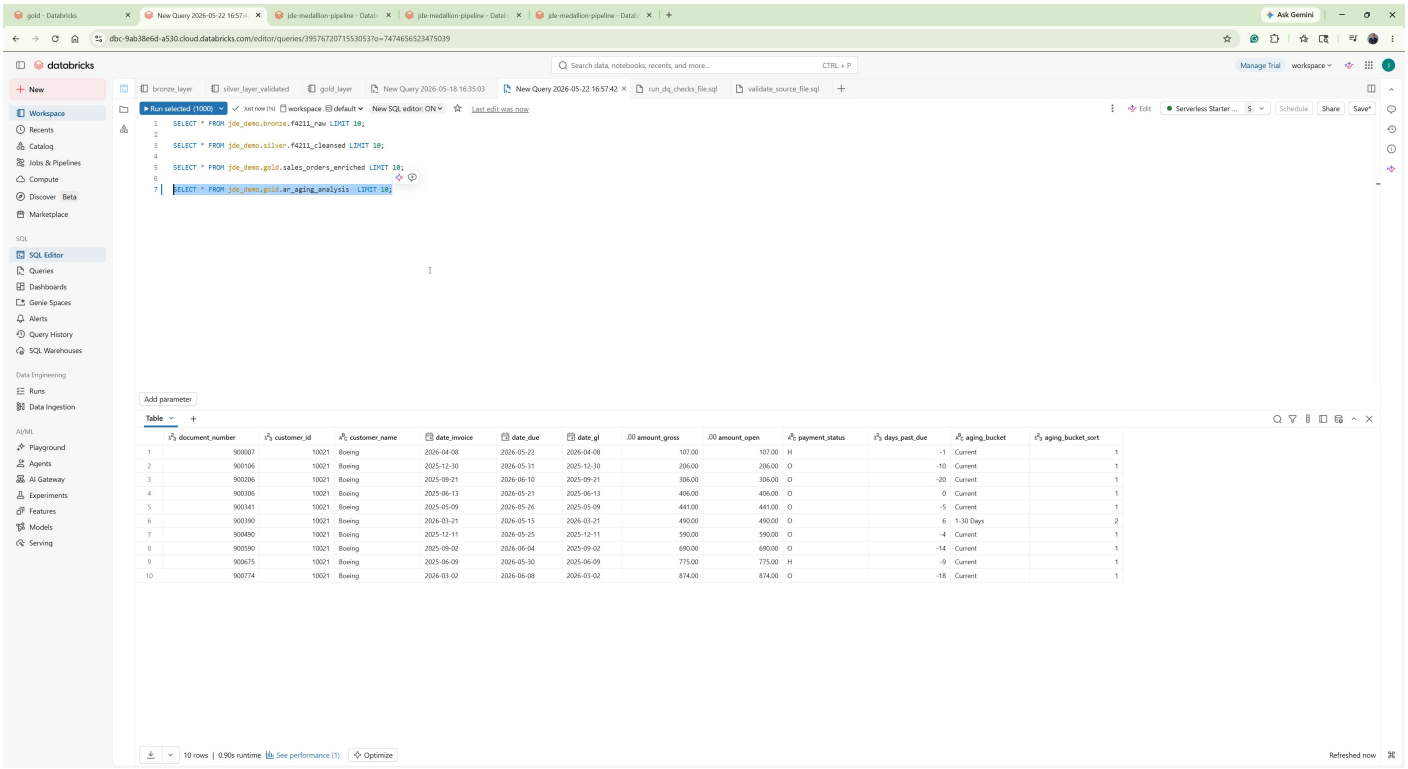
The data quality framework runs as the final task in the orchestration workflow. It includes three check categories: Uniqueness (composite key validation), Referential Integrity (foreign key validation across tables), and Row Count Reconciliation (Bronze vs Silver count matching to detect dropped rows from DLT Expectations).

DQ Summary Dashboard



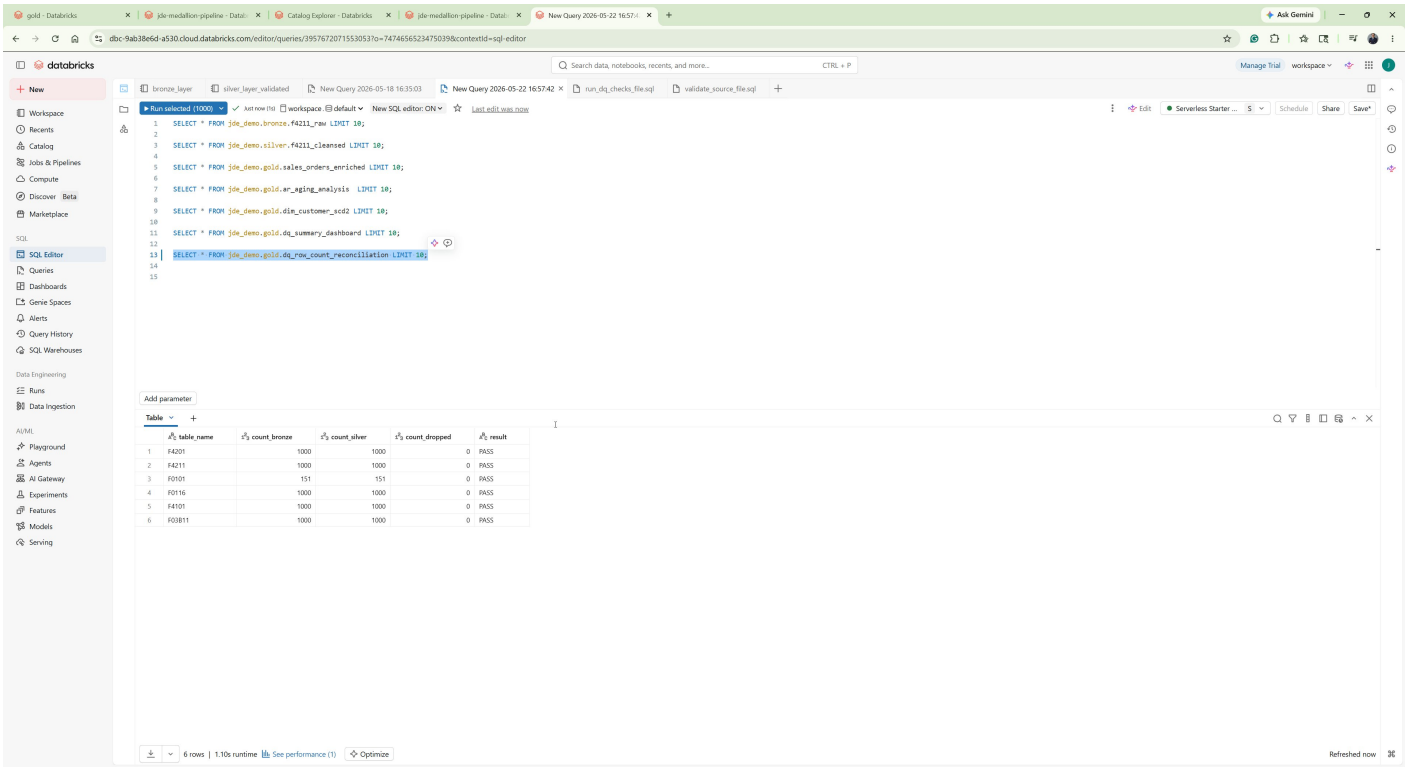
dq_summary_dashboard: 9 checks total (4 Uniqueness + 5 Referential Integrity), all PASS, 0 count_issues across all tables (F4201, F4211, F0101, F4101, F03B11). Check details show composite key patterns and FK relationships.

Referential Integrity Checks



dq_referential_integrity: 5 FK checks all PASS with 0 orphans. F4201 customer_id to F0101.address_id, F4211 customer_id to F0101.address_id, F4211 order_number to F4201.order_number, F4211 item_id to F4101.item_id, F03B11 customer_id to F0101.address_id.

Row Count Reconciliation

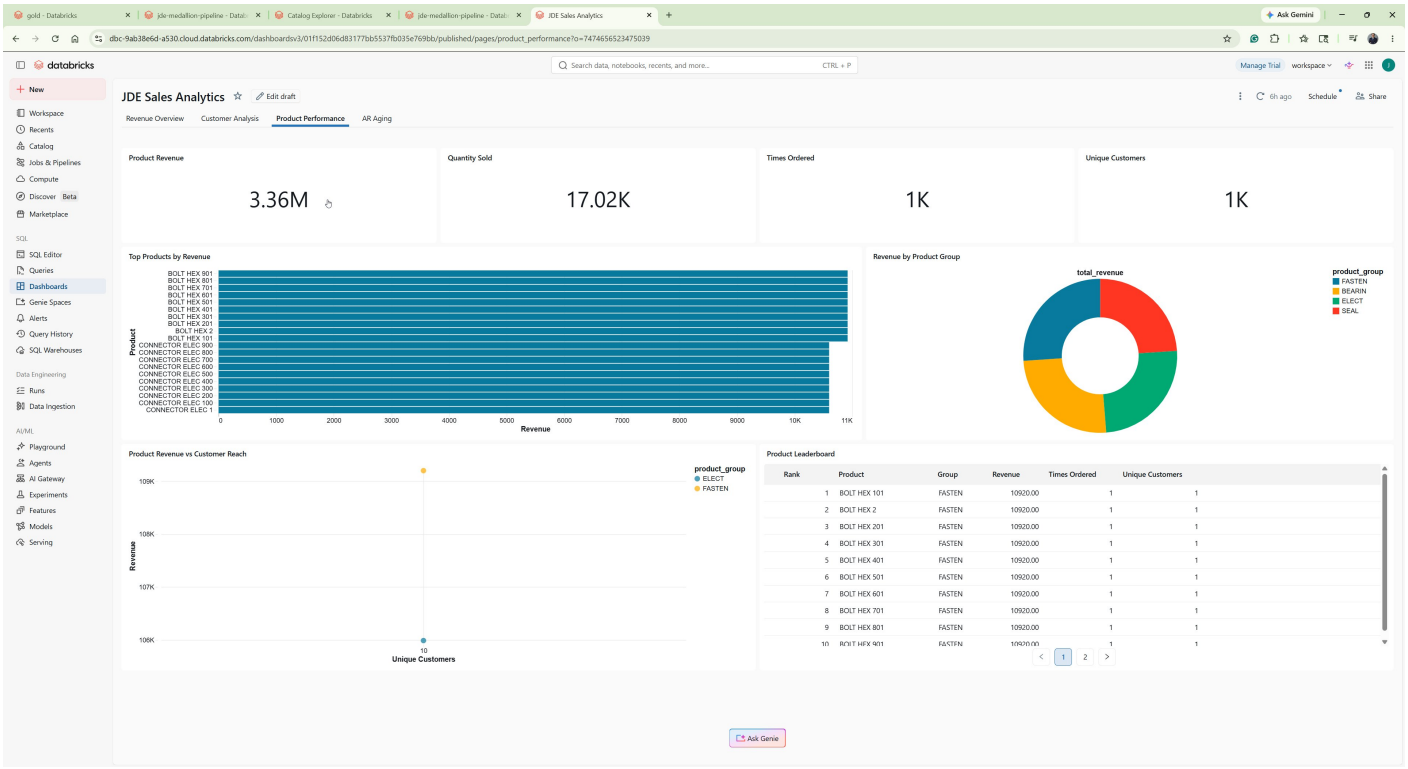


dq_row_count_reconciliation: All 6 tables PASS. Bronze vs Silver counts match perfectly (F4201: 1000/1000, F4211: 1000/1000, F0101: 151/151, F0116: 1000/1000, F4101: 1000/1000, F03B11: 1000/1000). Zero rows dropped by DLT Expectations.

9. AI/BI Dashboards

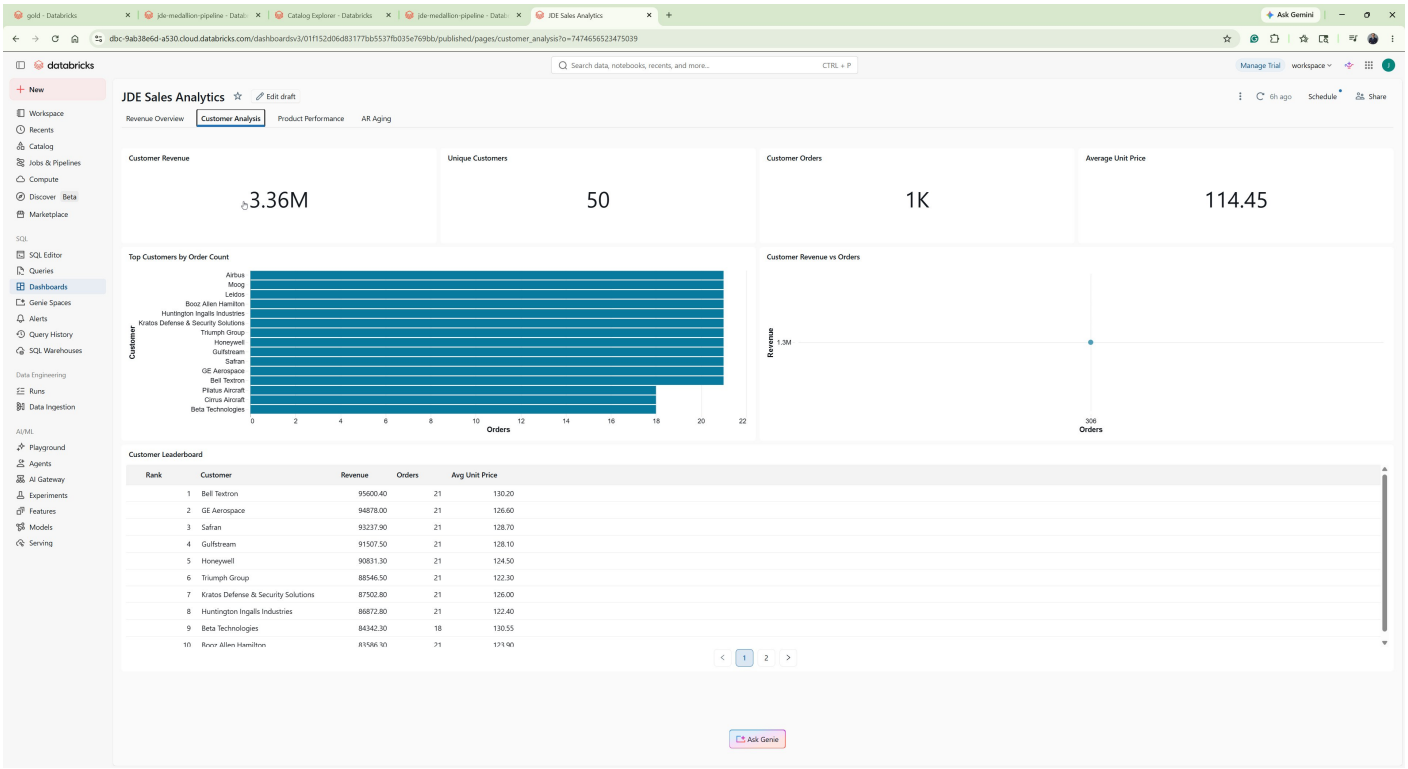
The JDE Sales Analytics dashboard is a 4-tab AI/BI dashboard built with Genie Code. It provides Revenue Overview, Customer Analysis, Product Performance, and AR Aging views with KPI cards, bar charts, donut charts, scatter plots, and detailed data tables.

Dashboard: Product Performance



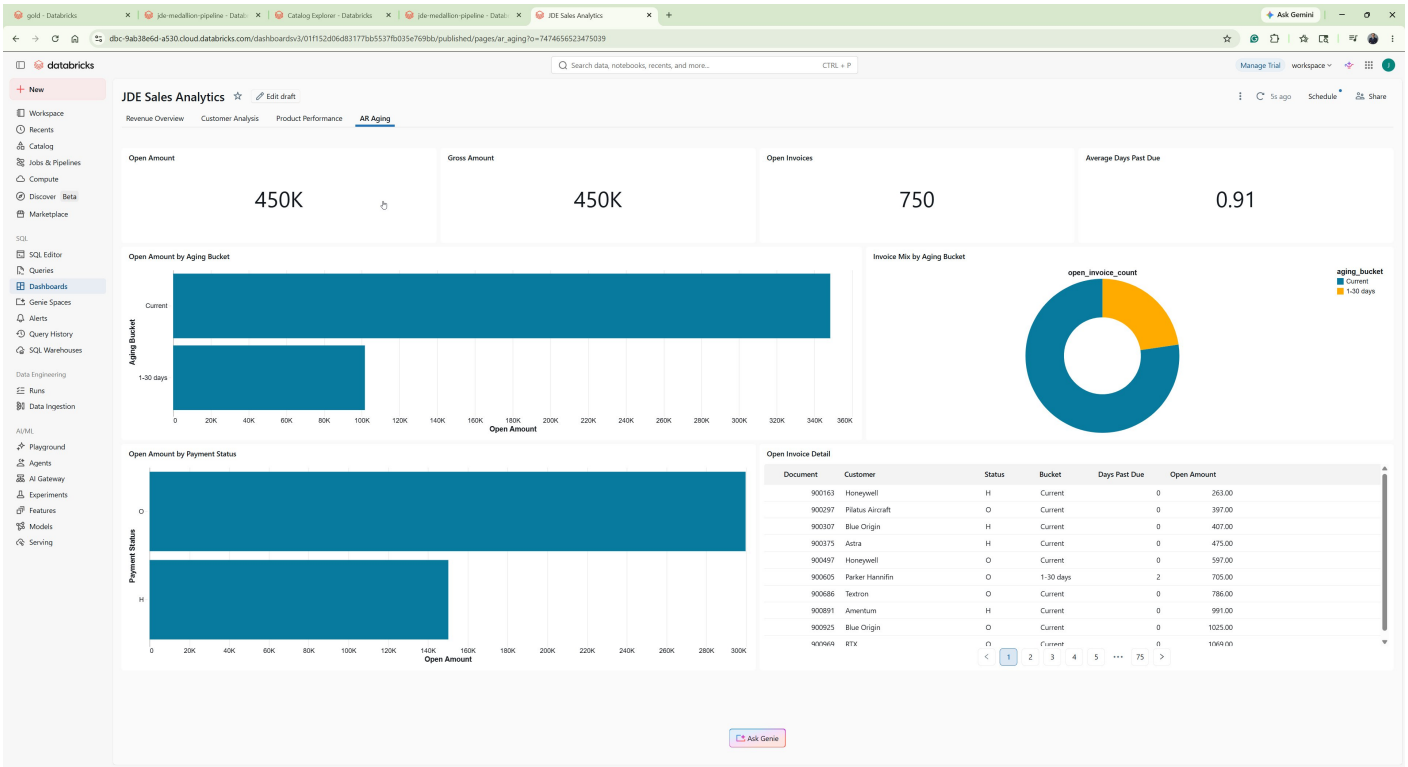
Product Performance tab: $3.36M product revenue, 17.02K quantity sold, 1K times ordered, 1K unique customers. Top Products by Revenue bar chart, Revenue by Product Group donut (FASTEN, BEARIN, ELECT, SEAL), Product Leaderboard table.

Dashboard: Customer Analysis



Customer Analysis tab: $3.36M revenue, 50 unique customers, 1K orders, $114.45 avg unit price. Top Customers by Order Count (Airbus, Moog, Leidos leading), Customer Revenue vs Orders scatter, Customer Leaderboard (Bell Textron #1 $95,600, GE Aerospace #2 $94,878, Safran #3 $93,237).

Dashboard: AR Aging

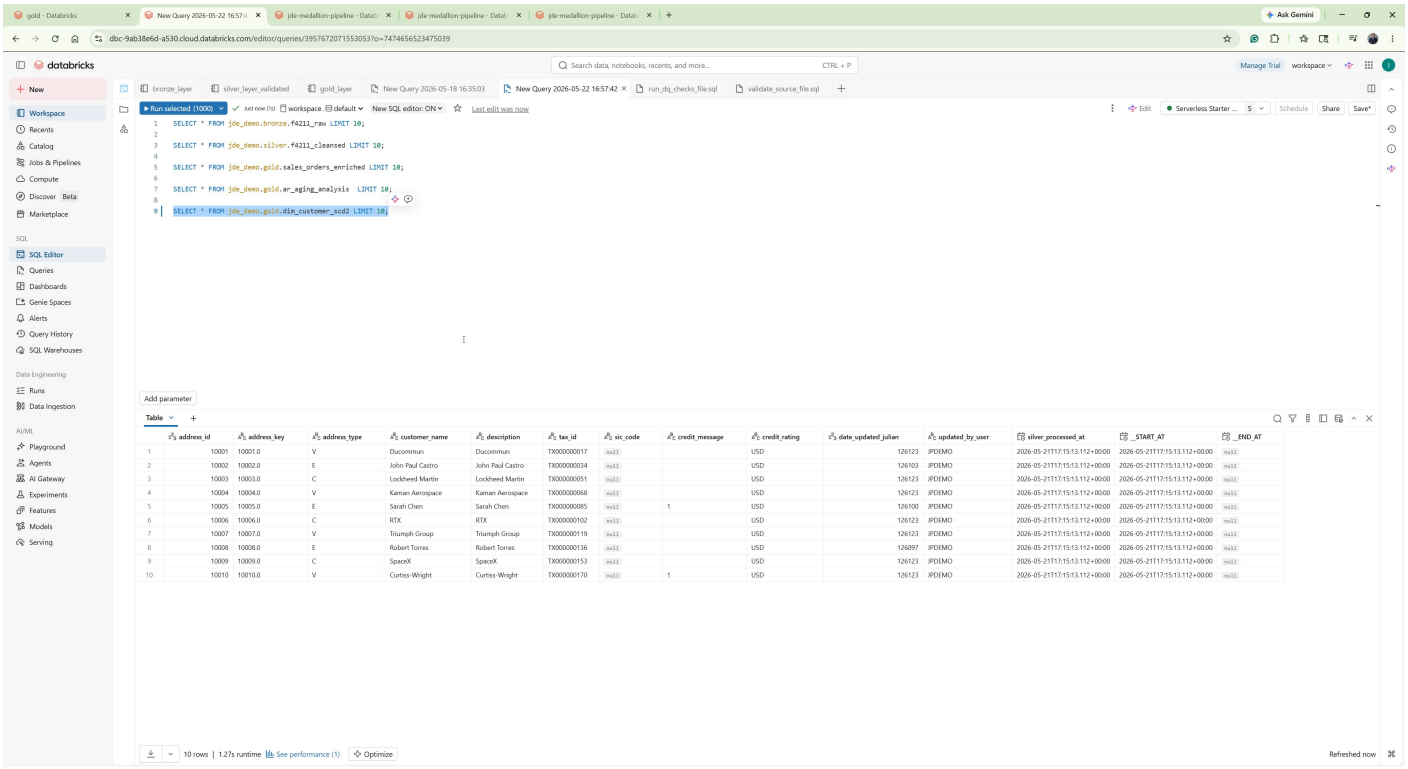


AR Aging tab: $450K open amount, $450K gross amount, 750 open invoices, 0.91 avg days past due. Open Amount by Aging Bucket (Current dominant), Invoice Mix by Aging Bucket donut, Open Amount by Payment Status, Open Invoice Detail (Honeywell, Blue Origin, Astra, RTX).

10. Genie Spaces

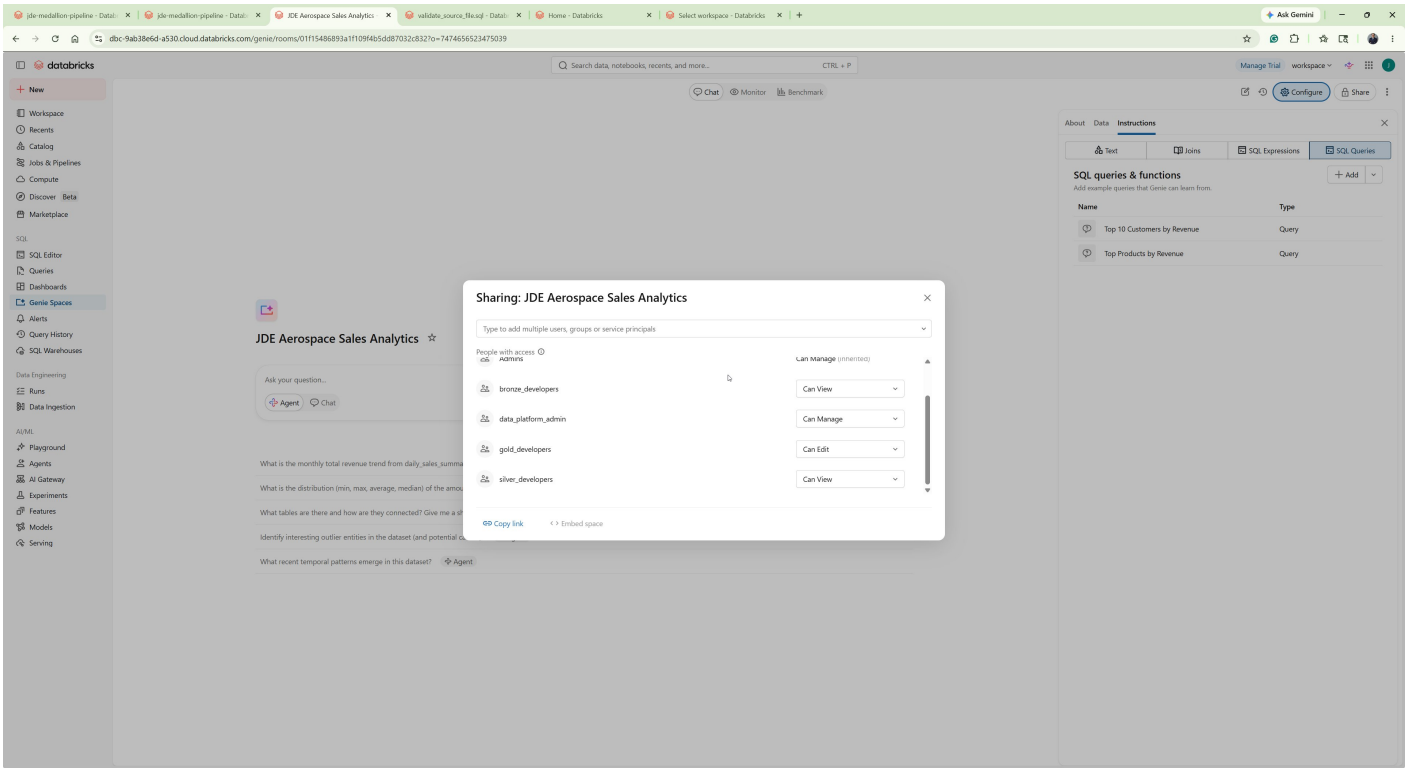
The JDE Aerospace Sales Analytics Genie Space enables natural language queries against the Gold layer. It includes pre-built SQL queries (Top 10 Customers by Revenue, Top Products by Revenue) and role-based sharing aligned with the Unity Catalog RBAC model.

Genie Space: Natural Language Query



Genie Space answering 'Which products have the most orders?' with natural language response, Top 15 Products by Revenue bar chart (BOLT HEX and CONNECTOR ELEC items at ~$10.9K each), Revenue by Product Group chart. Configure panel showing General Instructions.

Genie Space: Role-Based Sharing

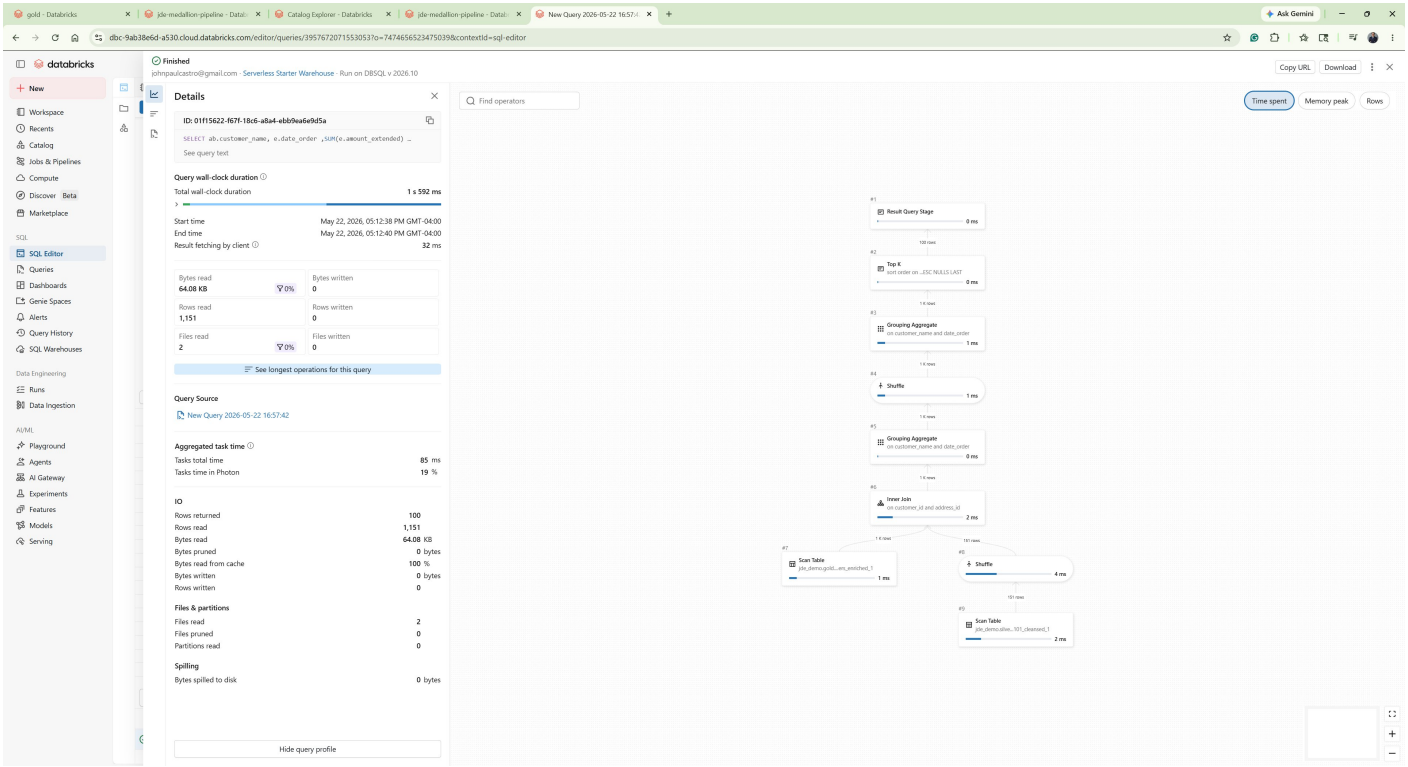


Genie Space sharing: Admins (Can Manage, inherited), bronze_developers (Can View), data_platform_admin (Can Manage), gold_developers (Can Edit), silver_developers (Can View). SQL Queries tab showing pre-built 'Top 10 Customers by Revenue' and 'Top Products by Revenue'.

11. Query Profiling

Query profiling demonstrates Databricks Serverless SQL performance with Photon engine acceleration. The execution plan shows scan, join, aggregate, and sort stages with timing breakdowns and I/O statistics.

Query Profile: Gold Layer Join

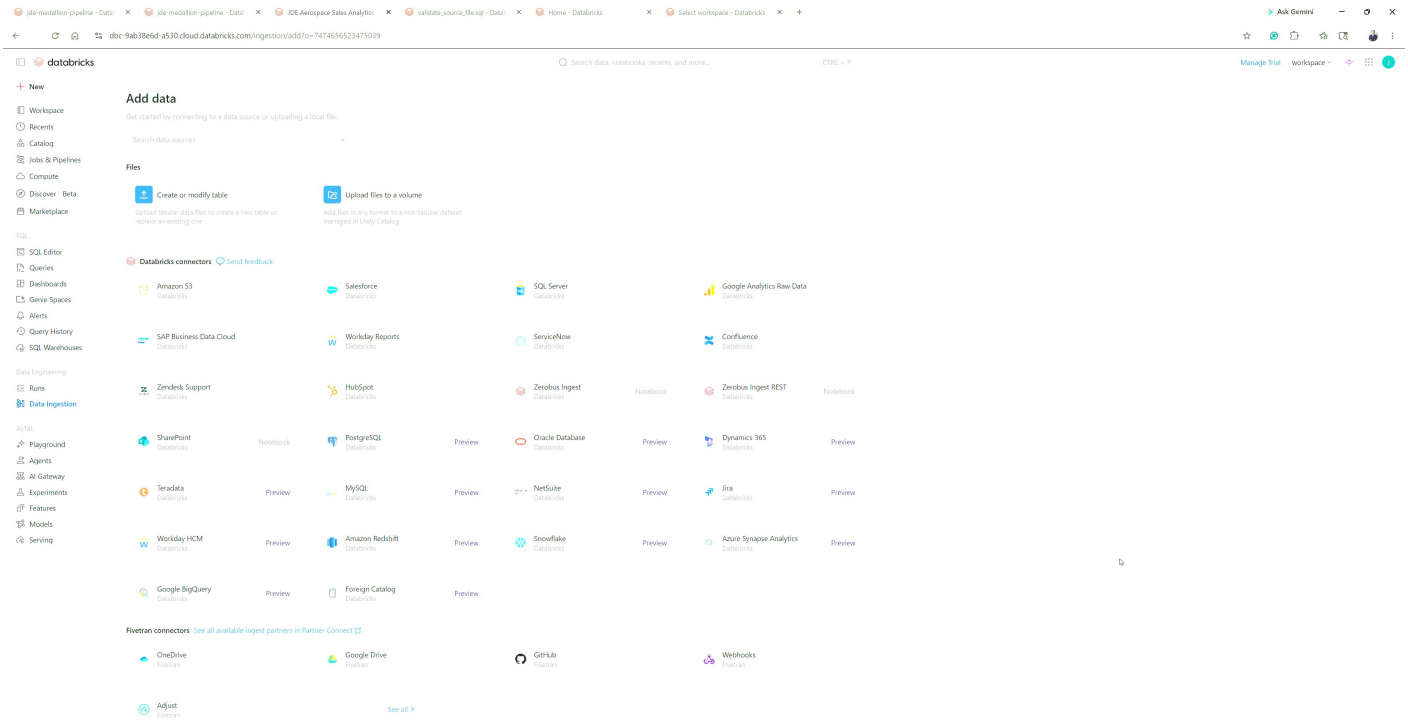


Query profile: 1.592s wall-clock duration, 19% Photon execution. Execution plan showing Scan Table (gold.sales_orders_enriched 1ms) > Inner Join on customer_id/address_id (2ms) > Grouping Aggregate (1ms) > Shuffle (4ms) > Top K sort. 1,151 rows read, 64.08 KB, 100% bytes from cache, 0 bytes spilled to disk.

12. Data Ingestion Connectors

Databricks supports native connectors to enterprise data sources including SQL Server, Salesforce, Oracle, PostgreSQL, SAP, Workday, and more. Fivetran connectors extend coverage to GitHub, Google Drive, OneDrive, and webhook-based sources. This portfolio used CSV file upload via Unity Catalog Volumes, but a production deployment would use Lakeflow Connect with the SQL Server connector for CDC-based real-time ingestion.

Add Data: Available Connectors



Data Ingestion page: Databricks connectors (Amazon S3, Salesforce, SQL Server, Google Analytics, SAP, Workday, ServiceNow, Confluence, HubSpot, PostgreSQL, Oracle, Dynamics 365, Teradata, MySQL, NetSuite, Jira, Snowflake, Azure Synapse, BigQuery, Foreign Catalog). Fivetran connectors (OneDrive, Google Drive, GitHub, Webhooks).

This portfolio was built on a Databricks 14-day trial workspace (May 2026) using real JDE ERP data patterns from 25+ years of aerospace distribution experience. Every screenshot represents live, executed work on a running Databricks environment. The platform demonstrates hands-on proficiency with Unity Catalog, Lakeflow Declarative Pipelines, Serverless SQL, Databricks Workflows, AI/BI Dashboards, Genie Spaces, and enterprise RBAC governance.

John Paul (JP) Castro | jpcenterprises.com | johnpaulcastro@gmail.com