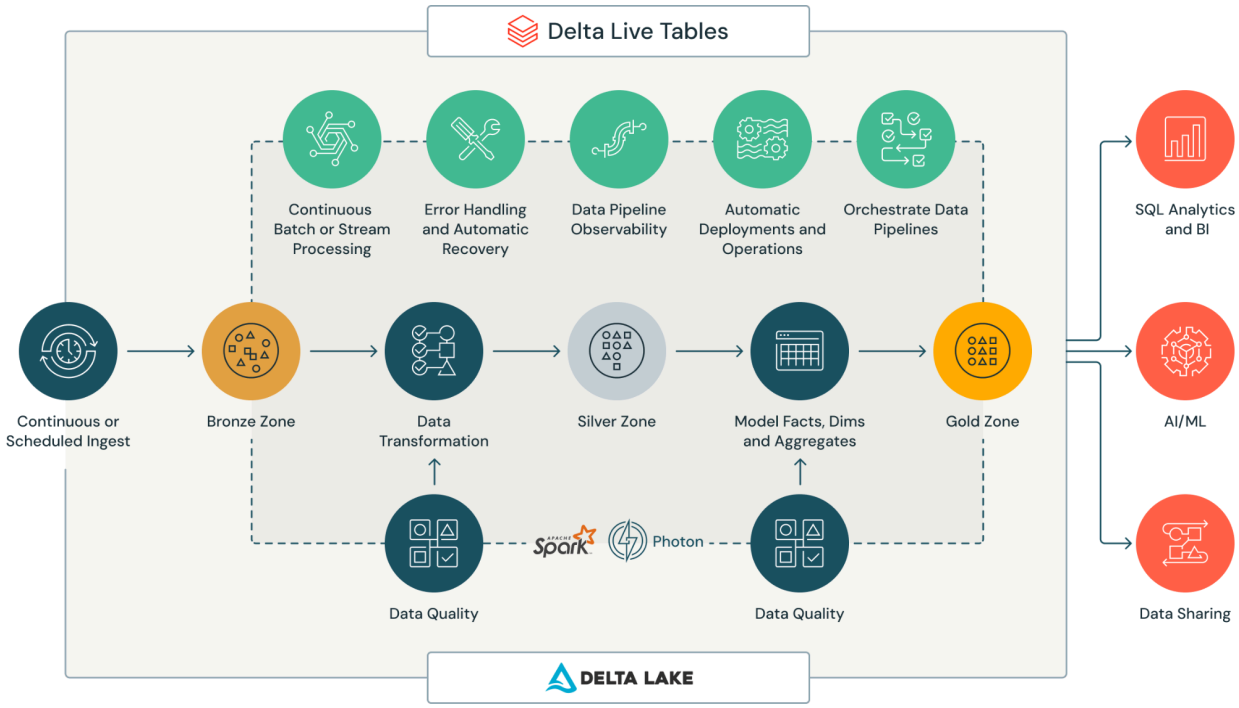
Hands-on: Databricks Delta Live Tables

Lab 01. Meu primeiro DLT

O que é Delta Live Tables?

Delta Live Tables (DLT) é uma estrutura declarativa para a criação de pipelines de dados confiáveis, sustentáveis e testáveis. Ele permite que engenheiros e analistas de dados definam transformações de dados usando SQL (ou Python) e gerencie automaticamente a infraestrutura subjacente e o fluxo de dados.

O DLT introduz um novo paradigma de engenharia de dados que se concentra em descrever as transformações que você deseja realizar em seus dados, em vez das etapas específicas para fazê-lo. Ele lida com dados de streaming (processamento contínuo de dados) e em lote(batch), é dimensionado automaticamente para corresponder à sua carga de trabalho e inclui controles de qualidade integrados e tratamento de erros.



Introdução

Neste lab hands-on, você criará e executará seu primeiro pipeline de DLT (Delta Live Tables) usando o conjunto de dados de táxi de exemplo de NYC. Exploraremos a arquitetura do medalhão, as Streaming tables e as visualizações materializadas e implementaremos verificações de qualidade de dados usando as expectativas do DLT.

Nosso exemplo de demonstração

O exemplo de demonstração neste guia ilustra a funcionalidade do Delta Live Tables usando um conjunto de dados de corrida clientes e transações bancárias.

O pipeline processa dados de streaming por meio de uma arquitetura de medalhão que consiste em camadas Bronze, Prata e Ouro.

Este guia aborda todas as habilidades e conceitos fundamentais de engenharia de dados: O Delta Live Tables pode lidar com o processamento de dados de streaming, implementar verificações de qualidade de dados e criar exibições prontas para análise. Ele fornece uma demonstração prática de como estruturar um pipeline de dados usando a arquitetura medallion, refinando e analisando progressivamente os dados à medida que eles se movem por cada camada.

O guia demonstra a integração da DLT com o Unity Catalog, mostrando como configurar a linhagem e a governança de dados. Por fim, ele aborda como o Assistente do Databricks aprimora a engenharia de dados integrando inteligência de dados, ajudando na otimização de consultas e melhorias no pipeline.

Noções básicas sobre a arquitetura Medalhão

A arquitetura medalhão é um padrão de design de dados usado para organizar logicamente os dados em um lakehouse, com o objetivo de melhorar de forma incremental e progressiva a estrutura e a qualidade dos dados à medida que fluem por cada camada. Esse padrão normalmente inclui três camadas:

1. Bronze (dados brutos): essa camada contém dados brutos ingeridos de várias fontes. Em nosso pipeline, isso é representado pelas tabelas `bronze transactions, fraud_reports, banking customers` e `country_coordinates.` É uma prática recomendada evitar regras ou transformações complexas neste estágio, concentrando-se na ingestão de todos os dados como estão, para que nenhum dado seja perdido devido a possíveis erros em regras de processamento complexas.
2. Silver (limpeza de qualidade de dados): Esta camada fornece uma visão mais refinada de nossos dados. As tabelas têm algum nível de limpeza e conformidade aplicado a elas. Em nosso pipeline, temos duas tabelas Silver: `silver_transac`tions.

3. Gold (nível de negócios com curadoria): essa camada fornece dados selecionados prontos para consumo por usuários de negócios, análises e ML. Geralmente, inclui agregações e junções entre tabelas diferentes. Em nosso pipeline, isso é representado pela view `gold_transactions`.

Essa arquitetura é uma diretriz para estruturar melhor os pipelines de dados, mas deve ser adaptada para atender às necessidades organizacionais específicas. Alguns projetos podem se beneficiar de camadas adicionais, enquanto outros podem combinar ou omitir camadas.

Streaming tables e Materialized Views

Neste pipeline DLT, usaremos Streaming tables e Materialized Views:

- Streaming tables: essas tabelas são projetadas para ingerir ou processar novos dados à medida que eles chegam continuamente. Eles usam a `sintaxe STREAMING TABLE` e são ideais para processamento de dados em tempo real de dados de streaming de fontes de dados somente acréscimo. As Streaming tables dão suporte ao processamento incremental, manipulando novos dados com eficiência sem reprocessar todo o conjunto de dados.

- Materialized Views: são exibições pré-computadas que armazenam o resultado. Eles são atualizados automaticamente quando os dados subjacentes são alterados. As Materialized Views são ideais para agregações ou junções complexas que você deseja calcular antecipadamente para acelerar a consulta ou melhorar o desempenho do painel.

Criar um novo Notebook

A partir de agora, você poderá escolher fazer o exercício na linguagem que mais se sente confortável, SQL ou Python:

- DLT SQL
- DLT Python

DLT SQL

Vamos começar e criar um notebook para nosso pipeline de dados!

1. No workspace do Databricks, acesse a pasta **Semana 03 - Engenharia de Dados com DLT/03_lab_dlt_credit/01.1-DLT-fraud-detection-SQL**

2. Passe por todo o conteúdo do noteebook e a cada código vá executanto e preenchendo o que estiver comentado no código com o `-- TODO:`, execute a instrução que ele apresenta.

3. Dica 1: comente todos os códigos, e vá execurando o DLT, etapa por etapa.

4. Dica 2: sempre que quiser testar o resultado das suas transformações antes de executar no pipeline DLT, execute o sql fora do `CREATE` para ir acompanhando o resultado pelo notebook. E para ler a tabela já criada ao invés de rodar `bronze_df = FROM STREAM(live.bronze_transactions)`, você executaria `bronze_df = FROM STREAM(bronze_transactions).`

DLT Python

Vamos começar e criar um notebook para nosso pipeline de dados!

5. No workspace do Databricks, acesse a pasta **Semana 03 - Engenharia de Dados com DLT/03_lab_dlt_credit/01.1-DLT-fraud-detection-Python**

6. Passe por todo o conteúdo do noteebook e a cada código vá executanto e preenchendo o que estiver comentado no código com o `# TODO:`, execute a instrução que ele apresenta.

7. Dica 1: comente todos os códigos, e vá execurando o DLT, etapa por etapa.

8. Dica 2: sempre que quiser testar o resultado das suas transformações antes de executar no pipeline DLT, execute o python fora da função DLT para ir acompanhando o resultado pelo notebook. E para ler a tabela já criada ao invés de rodar `bronze_df = dlt.read_stream("bronze_transactions")`, você executaria `bronze_df = spark.table("bronze_transactions").`

Expectativas de qualidade de dados

As expectativas são uma maneira de definir regras de qualidade de dados diretamente em seu pipeline. As expectativas filtram os dados problemáticos antes que eles cheguem às suas tabelas. Em nosso exemplo, temos uma expectativa:

SQL

```
 CONSTRAINT correct_customer_id EXPECT (customer_id IS NOT NULL) ON VIOLATION DROP ROW
```

Python

```
@dlt.expect_or_drop("correct_customer_id","customer_id IS NULL")
```

Essa expectativa é aplicada à tabela `silver_transactions`. Ele garante que apenas os registros com uma distância de viagem positiva sejam incluídos na tabela. Todas as linhas que violam essa restrição são descartadas (ação **DROP**). Como alternativa, eles podem ser retidos com um aviso (ação **WARN**) ou o pipeline pode ser configurado para falhar se ocorrer alguma violação (ação **FAIL**), dependendo de quão estritamente você deseja impor essa regra de qualidade de dados.

A qualidade dos dados é mais do que apenas uma palavra-chave em uma definição de pipeline. A Plataforma de Inteligência de Dados da Databricks garante a qualidade dos dados em três níveis distintos:

1. Nível arquitetônico: A arquitetura de medalhão (MA) garante a qualidade dos dados refinando progressivamente os dados por meio das camadas Bronze, Prata e Ouro, cada uma com níveis crescentes de limpeza e validação.

2. Nível de código: as restrições de qualidade de dados, implementadas como expectativas de DLT, impedem que dados incorretos fluam para nossas tabelas.

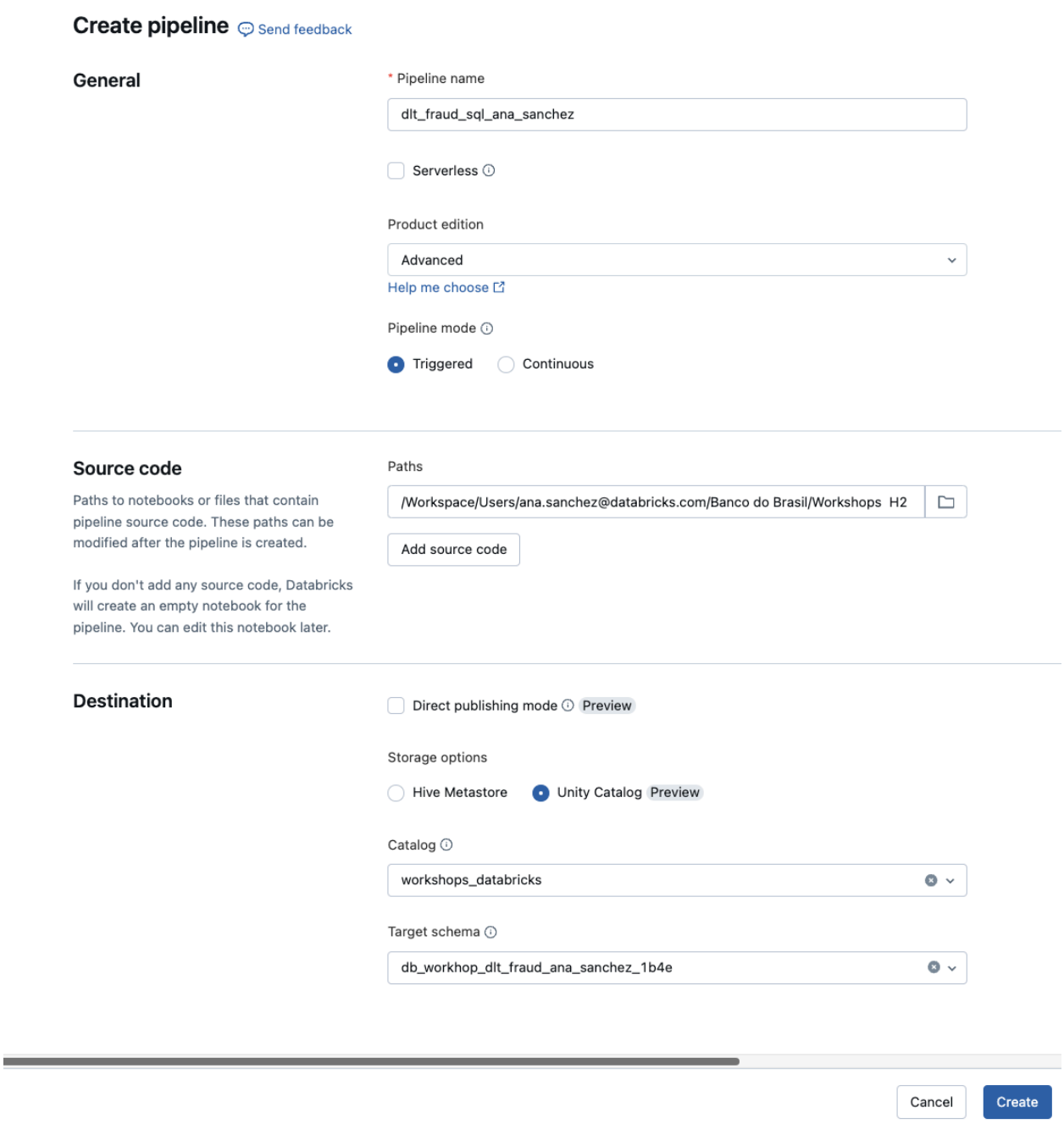
Você pode adicionar mais expectativas em cada camada para controles de qualidade de dados progressivamente mais rígidos.

3. Nível transacional: O formato Delta Lake subjacente garante a qualidade dos dados por meio de transações ACID (atomicidade, consistência, isolamento, durabilidade), garantindo a integridade dos dados mesmo em caso de falhas ou operações simultâneas.

Essa abordagem multicamadas para a qualidade dos dados — arquitetônica, em nível de código e transacional — fornece uma estrutura robusta para manter a integridade dos dados em todo o processo de pipeline de dados.

Definir a configuração do pipeline

Depois de criar seu notebook com o código do pipeline , você precisa definir as configurações principais do pipeline, pois ambas são necessárias para executar o pipeline. O código do pipeline em seu notebook define o que o pipeline faz — as transformações de dados, as verificações de qualidade e as tabelas criadas. As configurações de pipeline determinam como o pipeline é executado — em quais notebooks ele consiste, seu ambiente, a configuração de computação sem servidor e os parâmetros de execução. Juntos, esses componentes formam um pipeline executável do Delta Live Tables.



1. No workspace do Databricks, em "Data Engineering", clique em "Delta Live Tables" ou "Pipelines" (o que estiver aparecendo em seu Databricks), e clique em "Create Pipeline".

2. Em Configurações do pipeline, insira um nome descritivo para o pipeline: "dlt_fraud_sql_SEUNOME" ou "dlt_fraud_python_SEUNOME".

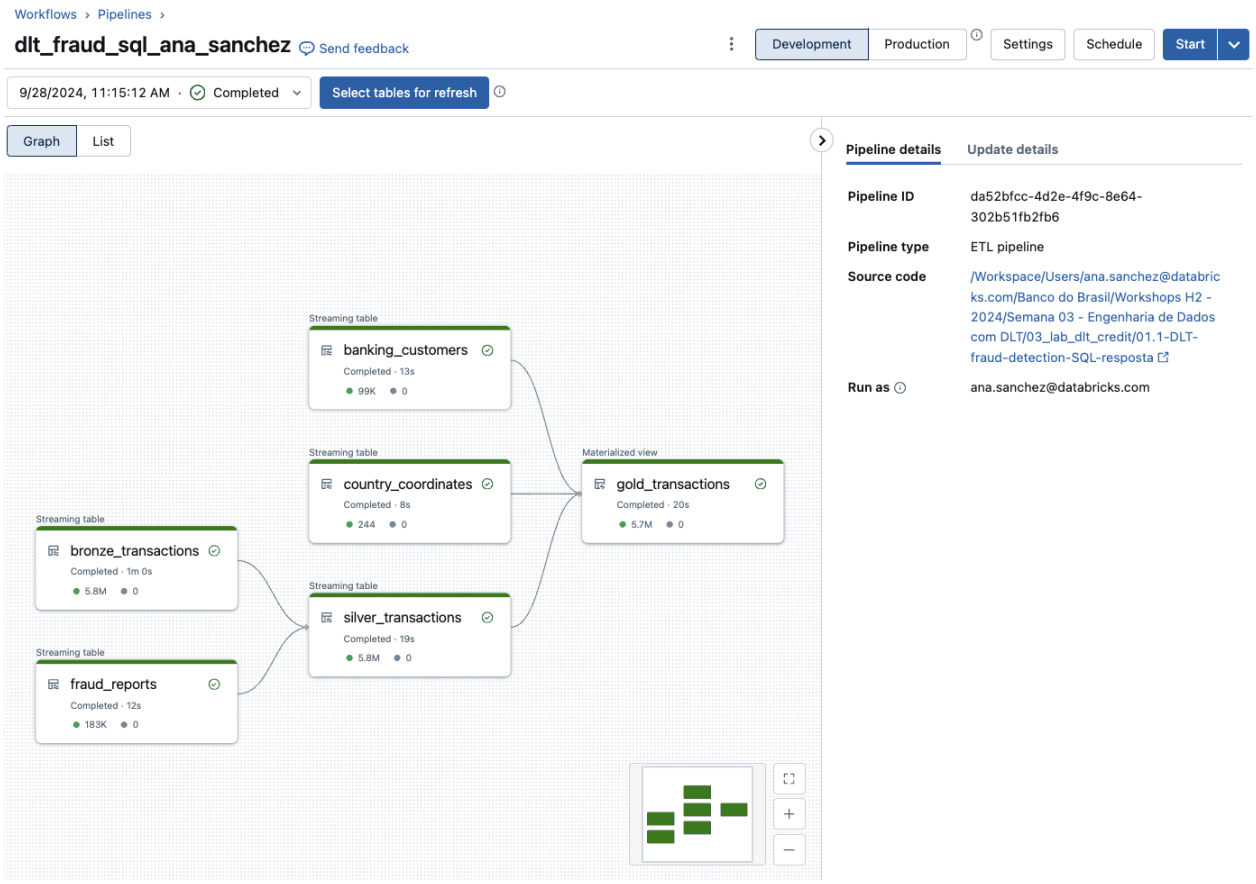
3. Os pipelines do Delta Live Tables podem ser executados no modo “Triggered” ou “Continuous”. Usaremos o modo Triggered aqui, que executa o pipeline sob demanda ou em um agendamento controlado por Databricks Workflows. Essa configuração oferece mais controle sobre a execução do pipeline do que o modo Continuous, esse outro modo é utilizado quando executamos em uma arquitetura streaming. Portanto, deixe a configuração "Pipeline Mode" como "Triggered".

4. Em "Source Code", defina o caminho para o notebook que você criou na etapa anterior.

5. O Catálogo do Databricks Unity usa a estrutura de namespace de três níveis catalog.schema.object para organizar e gerenciar ativos de dados em vários workspaces e ambientes de nuvem. Os objetos no terceiro nível podem incluir Streaming tables, Materialized Views, funções e outros ativos de dados, como modelos de IA, permitindo organização e governança de dados abrangentes. Essa estrutura hierárquica aprimora a descoberta de dados, simplifica o controle de acesso e melhora a governança geral de dados, permitindo que as organizações gerenciem seus dados com mais eficiência em escala. Em "Target", selecione um nome de catálogo do Unity existente (por exemplo, "workshops_databricks") e um schema de destino, como "db_dlt_workshop_SEUNOME". Detalhes sobre o uso de DLT com UC podem ser encontrados na documentação oficial.

6. Deixe todas as outras configurações como padrão por enquanto.

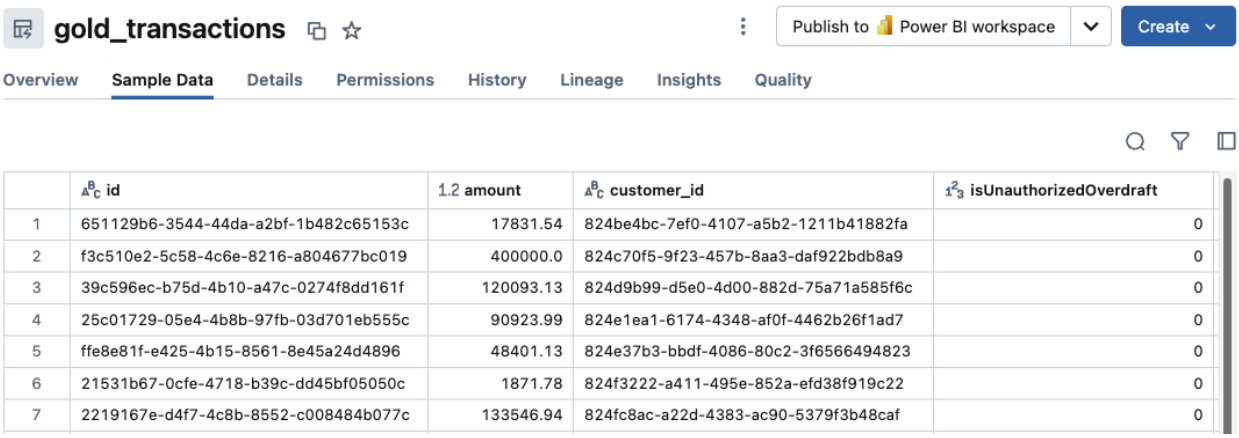
7. Execute o pipeline clicando em **Start.** Quando terminar ele ficará da seguinte forma:



Dados de exemplo e linhagem no Catálogo do Unity

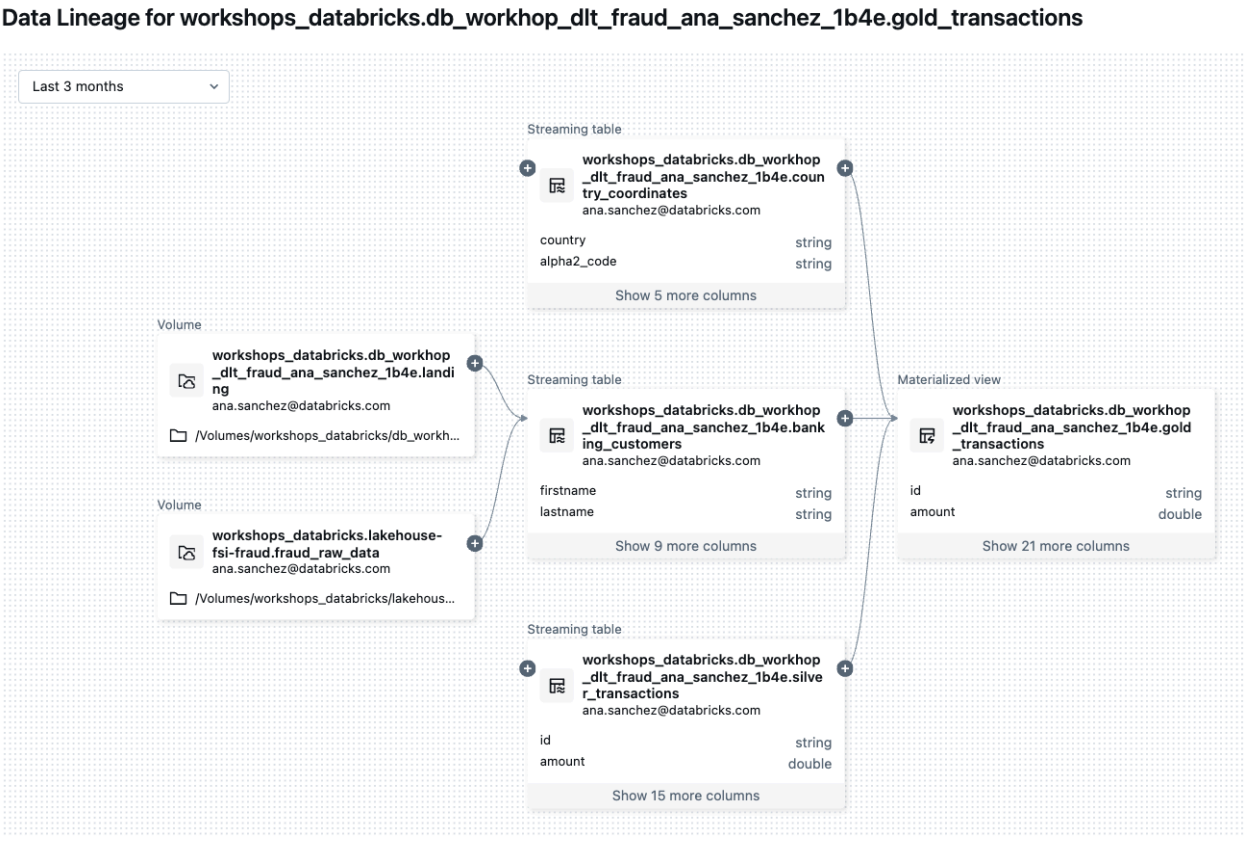
Siga as etapas abaixo para examinar os dados produzidos pelo pipeline e visualizar sua linhagem. Isso permite que você inspecione dados de amostra, visualize detalhes da tabela e explore a linhagem de dados de ponta a ponta de seus objetos de pipeline.

1. Na visualização do pipeline, selecione uma das tabelas, por exemplo, "gold_transactions".
2. No lado direito, clique no link "Table name".
3. Explore as várias guias para "Details" etc. e selecione a guia "Sample Data".
4. Em "Sample Data", você pode algumas linhas da tabela.



	id	amount	customer_id	isUnauthorizedOverdraft
1	651129b6-3544-44da-a2bf-1b482c65153c	17831.54	824be4bc-7ef0-4107-a5b2-1211b41882fa	0
2	f3c510e2-5c58-4c6e-8216-a804677bc019	400000.0	824c70f5-9f23-457b-8aa3-daf922bdb8a9	0
3	39c596ec-b75d-4b10-a47c-0274f8dd161f	120093.13	824d9b99-d5e0-4d00-882d-75a71a585f6c	0
4	25c01729-05e4-4b8b-97fb-03d701eb555c	90923.99	824e1ea1-6174-4348-af0f-4462b26f1ad7	0
5	ffe8e81f-e425-4b15-8561-8e45a24d4896	48401.13	824e37b3-bbdf-4086-80c2-3f6566494823	0
6	21531b67-0cfe-4718-b39c-dd45bf05050c	1871.78	824f3222-a411-495e-852a-efd38f919c22	0
7	2219167e-d4f7-4c8b-8552-c008484b077c	133546.94	824fc8ac-a22d-4383-ac90-5379f3b48caf	0

6. Em seguida, clique na guia de linhagem e abra a conexão clicando no botão "See lineage graph". Você verá uma exibição de linhagem completa de ponta a ponta. Embora nosso exemplo se concentre em pipelines de dados, essa exibição também incluiria volumes e modelos de IA para cenários mais complexos do mundo real.



Parabéns!

Agora você criou, executou e analisou um pipeline do Delta Live Tables demonstrando o processamento de dados de streaming, verificações de qualidade de dados e a criação de exibições prontas para análise. Você não apenas obteve informações sobre o DLT e os dados da viagem de táxi, mas também provavelmente aprendeu o quão caro pode ser um passeio na Big Apple se as coisas derem errado - talvez seja hora de considerar o metrô!

À medida que você se torna mais confortável com o DLT, pode expandir esse pipeline com transformações mais complexas, fontes de dados adicionais e verificações de qualidade de dados mais abrangentes. Considere explorar recursos avançados, como captura de dados de alteração para processamento eficiente de atualizações de sistemas de terceiros ou ingestão de dados de várias fontes de streaming , como Apache Kafka, Amazon Kinesis, Google Pub/Sub ou Apache Pulsar para aprimorar ainda mais seus pipelines de dados.

Lakeflow

Durante o Data + AI Summit, apresentamos o LakeFlow, a solução unificada de engenharia de dados que consiste em LakeFlow Connect (ingestão), LakeFlow Pipelines (transformação) e LakeFlow Jobs (orquestração). À medida que evoluímos o Delta Live Tables, todos os recursos que discutimos acima se tornarão parte da nova solução LakeFlow.

Para conhecer mais sobre o LakeFlow, acesse esse vídeo: Dados como Produtos: Jornada de modernização e entrega de valor - Databricks (youtube.com)

Recursos adicionais

- Tabelas dinâmicas delta no Databricks
- Documentação do Delta Live Tables
- Centro de Demonstração do Databricks com demonstrações em vídeo DLT, tours de produtos e tutoriais
- Eventos da Databricks com workshops práticos trimestrais de DLT
- O exemplo de SQL DLT do NY Taxi no GitHub

- Demonstrações em vídeo gravadas no centro de demonstração do Databricks