



Guia compacto da Engenharia de dados

Conteúdo

Introdução à engenharia de dados na Databricks.....	3
Diretrizes e Melhores Práticas	14
Dicas e Truques do Databricks Assistant para Engenheiros de Dados.....	15
Notebooks e Conjuntos de Dados Prontos para Uso.....	23
Estudos de Caso	25
Cox Automotive	26

01

Introdução à Engenharia de Dados na Databricks

Introdução à engenharia de dados na Databricks

Um relatório recente da [MIT Tech Review](#) mostra que 88% das organizações pesquisadas estão investindo, adotando ou fazendo experimentos com IA generativa (GenAI), e 71% pretendem construir seus próprios modelos de GenAI. Esse aumento do interesse em IA está impulsionando grandes investimentos, à medida que a IA se torna uma vantagem competitiva diferenciada em todos os setores. Conforme mais organizações trabalham para alavancar seus dados proprietários para esse fim, muitas encontram a mesma dura verdade:

Os melhores modelos de GenAI do mundo não terão sucesso sem dados de qualidade.

Essa realidade enfatiza a importância de construir pipelines de dados confiáveis que possam ingerir ou transmitir grandes quantidades de dados de forma eficiente e garantir qualidade. Uma plataforma unificada e uma boa engenharia de dados são componentes essenciais para o sucesso em todas as iniciativas de dados e IA, especialmente na era da GenAI.

Usando orientações práticas, padrões úteis, melhores práticas e exemplos do mundo real, este livro fornecerá a você uma compreensão de como a [Databricks Data Intelligence Platform](#) auxilia os engenheiros de dados a enfrentarem os desafios desta nova era.

O que é a engenharia de dados?

A [engenharia de dados](#) consiste em extraír dados brutos de uma fonte de dados e processá-los para que sejam armazenados e organizados para um caso de uso posterior, como análise de dados, Business Intelligence (BI) ou treinamento em modelos de machine learning (ML). Em outras palavras, é o processo de preparar dados para que se possa extrair valor deles.

Uma forma prática de enxergar a engenharia de dados é usar a seguinte estrutura, que inclui três partes principais:

1. Ingestão

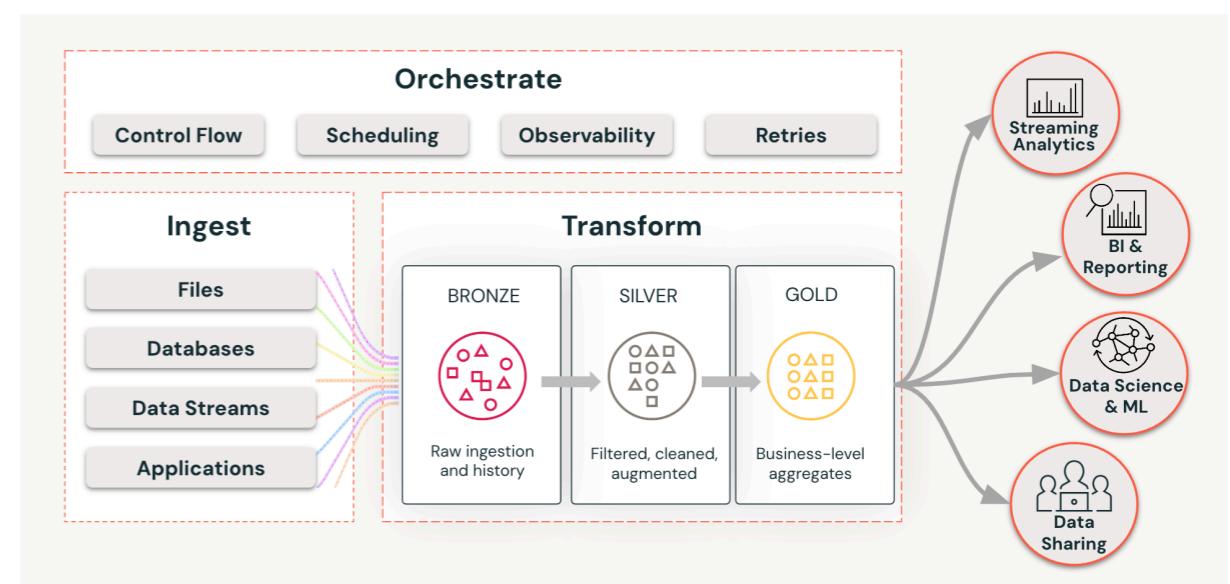
A ingestão de dados é o processo de trazer dados de uma ou mais fontes para uma plataforma de dados. Essas fontes de dados podem ser arquivos armazenados no local ou em serviços de armazenamento em nuvem, bancos de dados, aplicativos e, cada vez mais, streams de dados que produzem eventos em tempo real.

2. Transformação

A transformação de dados pega dados brutos ingeridos e usa uma série de etapas (chamadas de “transformações”) para filtrá-los, padronizá-los, limpá-los e, finalmente, agregá-los para que sejam armazenados de forma utilizável. Um padrão popular é a [arquitetura medallion](#), que define três estágios no processo — Bronze, Prata e Ouro.

3. Orquestração

A orquestração de dados refere-se à maneira como um pipeline de dados que realiza a ingestão e transformação é agendado e monitorado, bem como ao controle das várias etapas do pipeline e ao tratamento de falhas (por exemplo, realizando uma nova tentativa de execução).



Desafios da engenharia de dados na era da IA

Conforme mencionado anteriormente, a engenharia de dados é essencial para assegurar dados confiáveis para iniciativas de IA. Engenheiros de dados que constroem e mantêm pipelines de ETL e a infraestrutura de dados que suporta cargas de trabalho de análise e IA enfrentam desafios específicos neste cenário em rápida evolução.

- **Fontes de dados díspares desafiam a maioria das organizações:** a ISG prevê que, até 2026, 8 em cada 10 empresas terão seus dados distribuídos por vários provedores de nuvem e data centers locais que abrangem múltiplas localizações. Essa descentralização cria uma dependência de equipes especializadas e isoladas, pipelines ineficientes e desenvolvimento com altos custos e tempo de retorno lento, o que limita o uso de dados e bloqueia a inovação.
- **Manipulação de dados em tempo real:** de aplicativos móveis a dados de sensores no chão de fábrica, cada vez mais dados são criados e transmitidos em tempo real e exigem processamento de baixa latência para que possam ser usados na tomada de decisões em tempo real.
- **Dimensionamento confiável de pipelines de dados:** com dados chegando em grandes quantidades e frequentemente em tempo real, dimensionar a infraestrutura de computação que executa pipelines de dados é desafiador, especialmente ao tentar manter os custos baixos e o desempenho alto. Executar pipelines de dados de forma confiável, monitorar pipelines de dados e solucionar problemas quando ocorrem falhas são algumas das responsabilidades mais importantes dos engenheiros de dados.
- **Qualidade dos dados:** “garbage in, garbage out” (“entra lixo, sai lixo”). A alta qualidade dos dados é essencial para treinar modelos de alta qualidade e obter insights açãoáveis. Garantir a qualidade dos dados é um desafio fundamental para os engenheiros de dados.

- **Governança e segurança:** a governança de dados está se tornando um desafio crucial para as organizações que encontram seus dados distribuídos por vários sistemas, com um número crescente de equipes internas buscando acessá-los e utilizá-los para diferentes propósitos. Proteger e governar os dados também é uma preocupação regulatória significativa que muitas organizações enfrentam, especialmente em indústrias altamente regulamentadas.

Esses desafios destacam a importância de selecionar a plataforma de dados adequada para explorar novas oportunidades na era da IA. No entanto, uma plataforma de dados nesta nova era pode ultrapassar os desafios de desenvolver soluções de IA. A plataforma certa pode melhorar a experiência e a produtividade dos profissionais de dados, incluindo engenheiros de dados, ao incorporar inteligência e usar IA para auxiliar nas tarefas diárias de engenharia.

Em outras palavras, a nova plataforma de dados é uma plataforma de *inteligência de dados*.

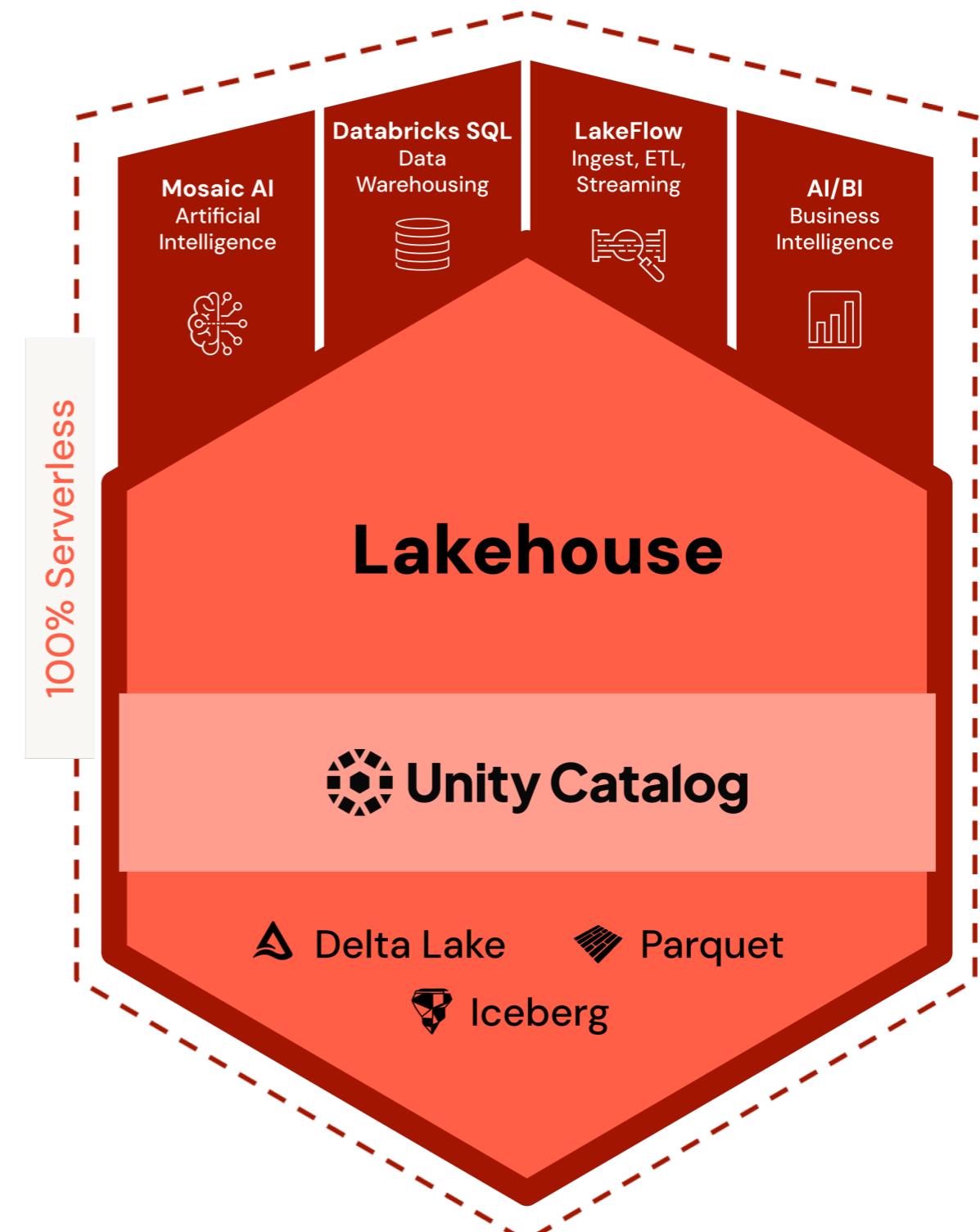
Databricks Data Intelligence Platform

A missão da Databricks é democratizar dados e IA, permitindo que as organizações utilizem seus dados únicos para desenvolver ou aperfeiçoar seus próprios modelos de machine learning e IA generativa, de modo a gerar novos insights que conduzam à inovação empresarial.

A Databricks Data Intelligence Platform é construída sobre a [arquitetura de lakehouse](#) para fornecer uma base aberta e unificada para todos os dados e governança, e é alimentada por um Mecanismo de Inteligência de Dados que entende a singularidade dos seus dados. Com essas capacidades, a Data Intelligence Platform permite que os clientes da Databricks executem uma variedade de cargas de trabalho, desde business intelligence e data warehousing até IA e ciência de dados.

Para obter uma melhor compreensão da Databricks Platform, aqui está uma visão geral das diferentes partes da arquitetura no que diz respeito à engenharia de dados.

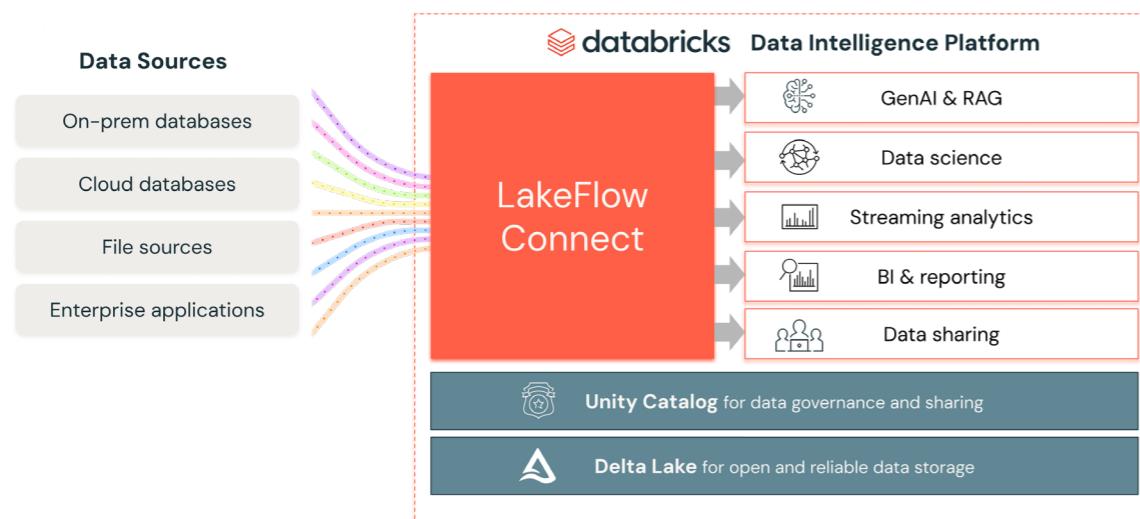
A Databricks Data Intelligence Platform permite que você execute todas as suas iniciativas de dados e IA. Como uma plataforma 100% serverless, ela oferece recursos integrados, como recuperação de desastres, controles de custos e segurança empresarial. Os componentes principais apresentam o Mosaic AI com IA de ponta a ponta para IA generativa e clássica; o Databricks SQL, o data warehouse mais eficiente na nuvem; ingestão eficiente de dados e ferramentas de transformação confiáveis, como Workflows e Delta Live Tables (DLT), para garantir que você possa gerenciar todos os seus dados para qualquer carga de trabalho; e AI/BI, que se integra de maneira avançada ao Databricks SQL para expandir facilmente a inteligência de negócios em toda a sua empresa.



Ingestão de dados com o Databricks LakeFlow

A Databricks permite que as organizações movam dados de vários sistemas com eficiência para uma arquitetura de lakehouse única, aberta e unificada. O Databricks **LakeFlow Connect** oferece conectores nativos para ingestão de dados de aplicativos SaaS populares, bancos de dados e fontes de arquivos, que qualquer profissional pode usar para construir pipelines de dados incrementais em escala. Esses conectores integrados proporcionam ingestão incremental eficiente de ponta a ponta, configuração fácil com uma interface de usuário simples ou acesso à API, e governança via Unity Catalog — tudo impulsionado pela Data Intelligence Platform. O LakeFlow Connect faz parte do **LakeFlow** — a nova solução unificada de engenharia de dados da Databricks que abrange ingestão, transformação e orquestração — e é o primeiro desses três componentes a ser lançado, compatível com as ferramentas existentes. Além do LakeFlow Connect, o Databricks Auto Loader, um conector para armazenamento de objetos na nuvem, é compatível com o Structured Streaming e o Delta Live Tables.

Native ingestion for the Data Intelligence Platform

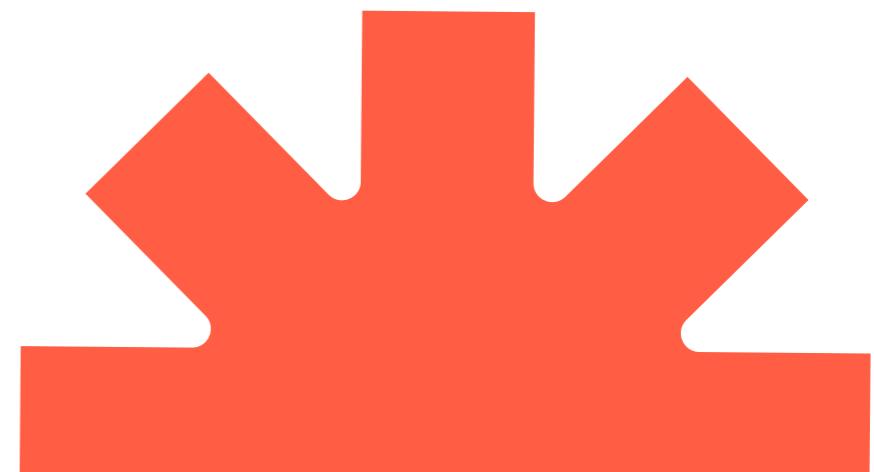


Confiabilidade e desempenho de dados com o Delta Lake

Para trazer abertura, confiabilidade e gerenciamento do ciclo de vida aos data lakes, a arquitetura de lakehouse da Databricks é construída sobre o **Delta Lake**, um formato de armazenamento de código aberto e de alto desempenho que resolve desafios relacionados à ingestão de dados não estruturados/estruturados, à aplicação da qualidade dos dados, dificuldades com a exclusão de dados para conformidade ou problemas com a modificação de dados para captura de dados. Usuários do Delta Lake UniForm agora podem ler tabelas Delta com clientes Hudi e Apache Iceberg™, mantendo o controle de seus dados. Além disso, o **Delta Sharing** permite o compartilhamento fácil e seguro de conjuntos de dados dentro e fora da organização.

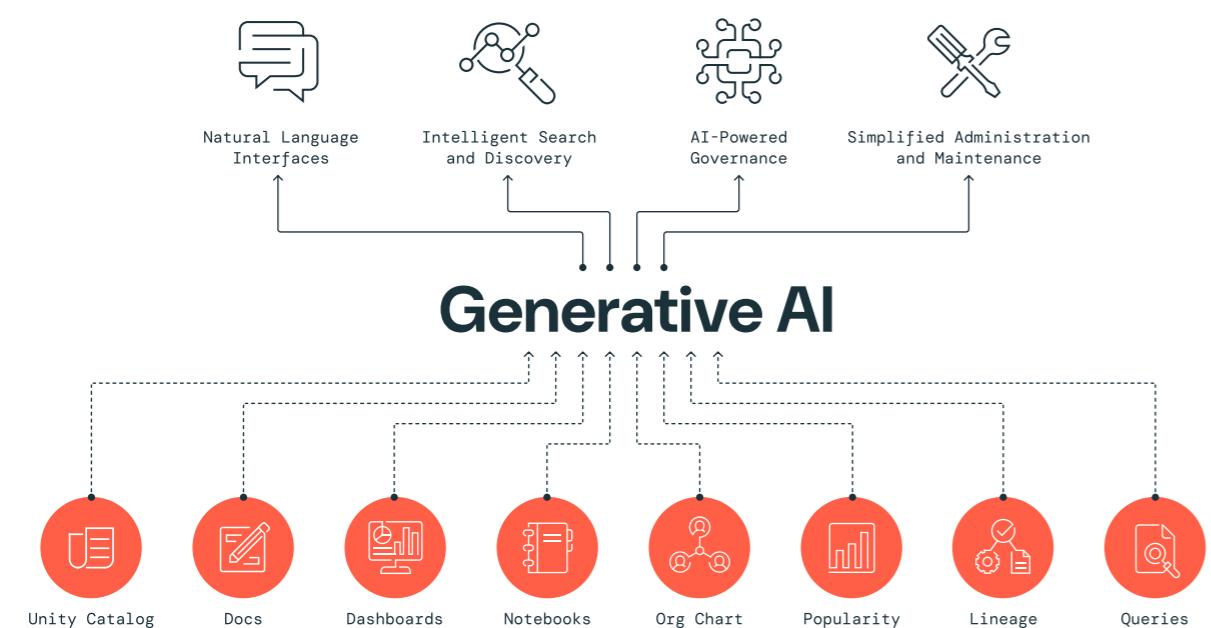
Governança unificada com o Unity Catalog

Com o **Unity Catalog**, as equipes de engenharia e governança de dados se beneficiam de um catálogo de dados para toda a empresa com uma interface única para gerenciar permissões, centralizar a auditoria, rastrear automaticamente a linhagem de dados até o nível da coluna e compartilhar dados entre plataformas, nuvens e regiões.



IA generativa com o Mecanismo de Inteligência de Dados

A Databricks traz IA para seus dados para ajudá-lo a levar a IA para o mundo, e no coração da **Databricks Data Intelligence Platform** está o Mecanismo de Inteligência de Dados. A Databricks ajuda você a ter sucesso com IA usando seus próprios dados para democratizar insights e reduzir custos. O Mecanismo de Inteligência de Dados, o primeiro do gênero, usa IA para alimentar todas as partes da Databricks Data Intelligence Platform. Utilizando sinais em todo o seu ambiente Databricks, incluindo o Unity Catalog, painéis, notebooks, pipelines de dados e documentação, o Mecanismo de Inteligência de Dados cria modelos de IA generativa altamente especializados e precisos que compreendem seus dados, seus padrões de uso e sua terminologia de negócios.



Pipelines de dados confiáveis e processamento de streaming em tempo real

com Delta Live Tables **Delta Live Tables** é uma estrutura ETL declarativa que ajuda as equipes de dados a simplificar e tornar o ETL econômico em streaming e batch. Basta definir as transformações que você deseja realizar em seus dados e permitir que os pipelines DLT lidem automaticamente com a orquestração de tarefas, gerenciamento de cluster, monitoramento, qualidade de dados e tratamento de erros. Os engenheiros podem tratar seus dados como código e aplicar as melhores práticas modernas de engenharia de software, como testes, tratamento de erros, monitoramento e documentação para implantar pipelines confiáveis em escala. O DLT oferece suporte total a Python e SQL e é adaptado para funcionar com cargas de trabalho em streaming e batch.

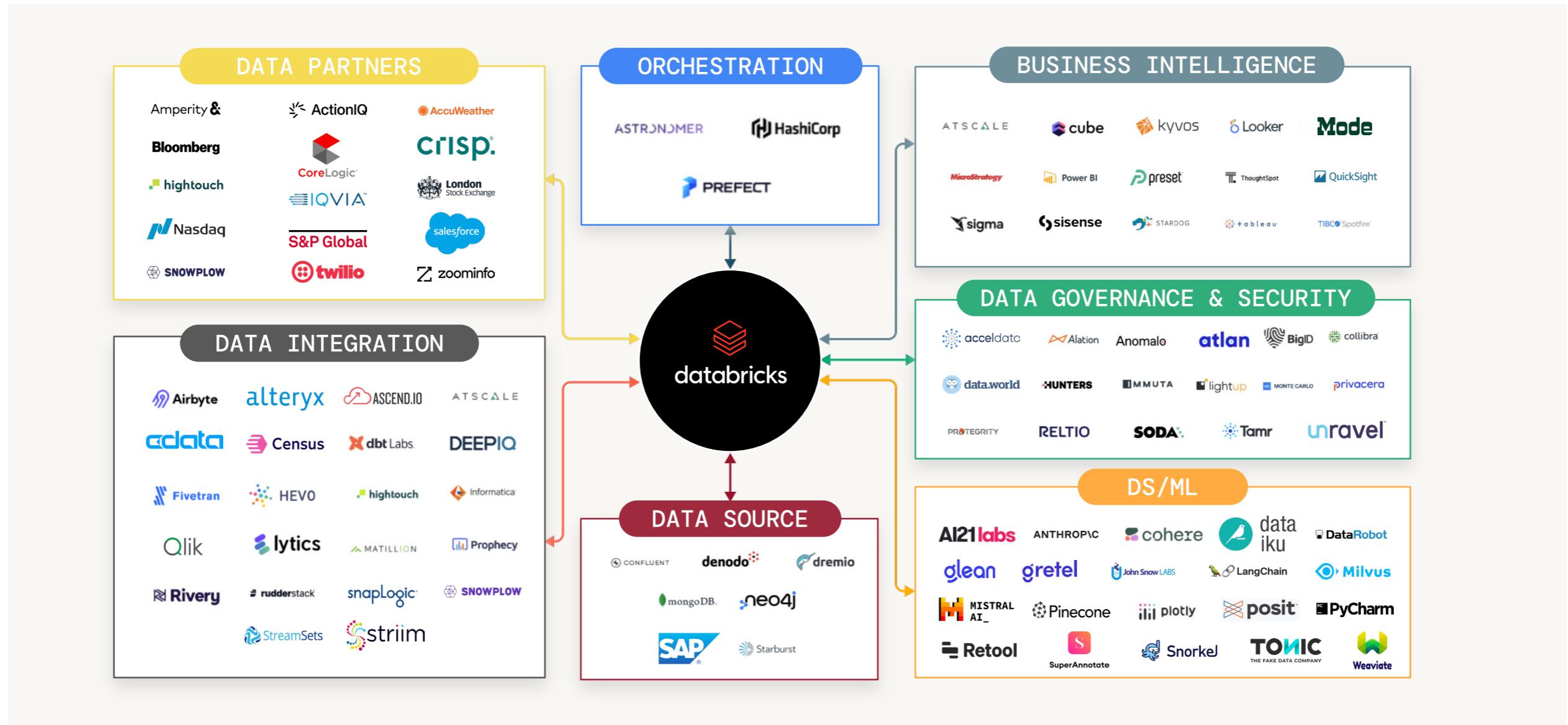
Orquestração unificada de dados com Databricks Workflows

O **Databricks Workflows** oferece uma solução de orquestração simples e confiável para dados e IA na Plataforma de Inteligência de Dados. O Databricks Workflows permite que você defina fluxos de trabalho de várias etapas para implementar pipelines de ETL, fluxos de trabalho de treinamento de ML e muito mais. Ele oferece recursos aprimorados de fluxo de controle e suporta diferentes tipos de tarefas e opções de acionamento de fluxo de trabalho. Como o orquestrador nativo da plataforma, o Databricks Workflows também fornece observabilidade avançada para monitorar e visualizar a execução do fluxo de trabalho, juntamente com recursos de alerta para quando problemas surgirem. O Databricks Workflows oferece opções de compute serverless para que você possa aproveitar o dimensionamento inteligente e a execução eficiente de tarefas.

Um ecossistema robusto de soluções de dados

A Data Intelligence Platform é construída com tecnologias de código aberto e utiliza padrões abertos para que as principais soluções de dados possam ser aproveitadas com qualquer coisa que você desenvolva no lakehouse.

Uma grande coleção de **parceiros de tecnologia** facilita e simplifica a integração de tecnologias das quais você depende ao migrar para a Databricks — e você não fica preso a uma pilha fechada de tecnologia de dados.

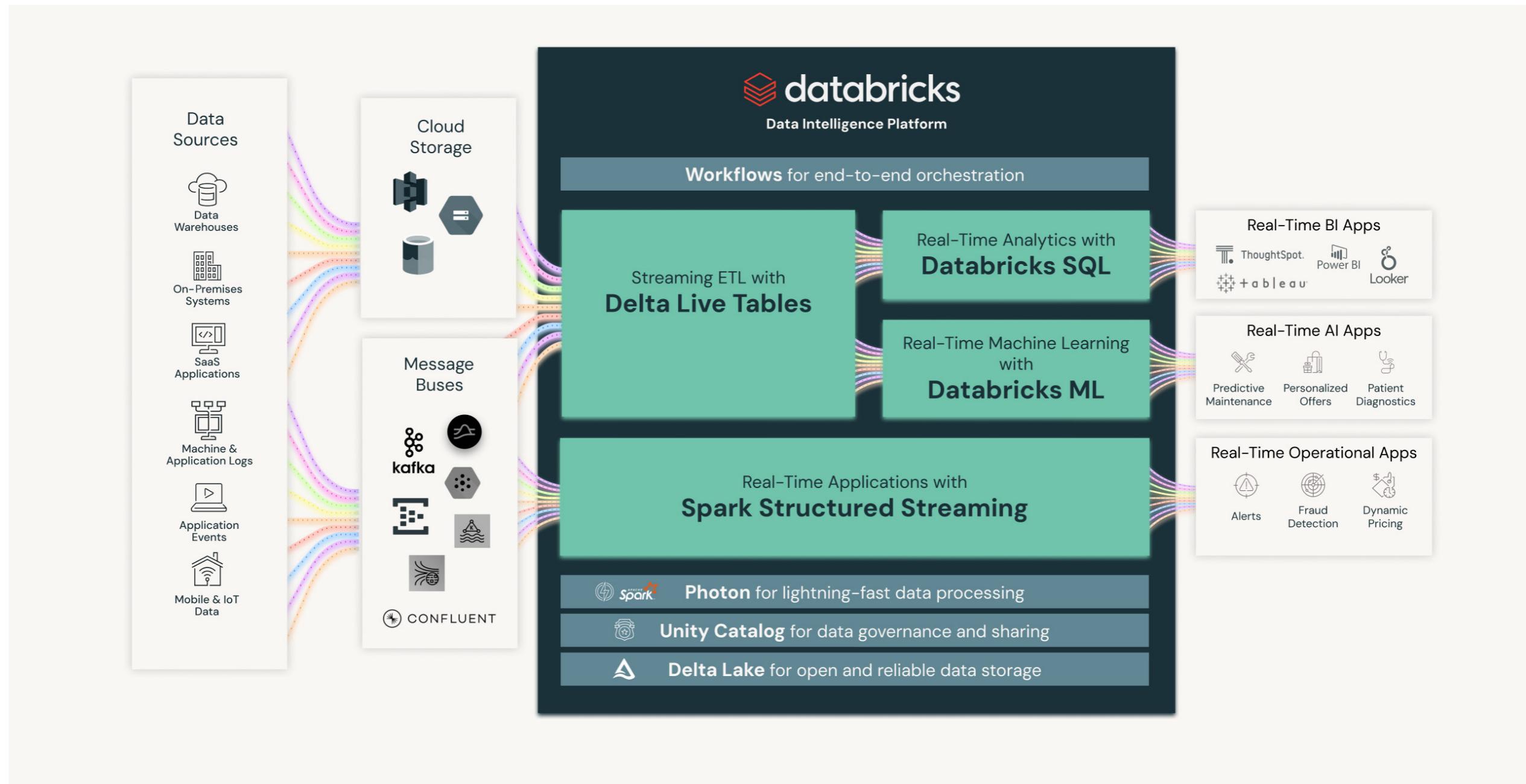


A Data Intelligence Platform se integra a uma grande coleção de tecnologias

Por que engenheiros de dados escolhem a Data Intelligence Platform

Então, como a Data Intelligence Platform ajuda em cada um dos desafios de engenharia de dados discutidos anteriormente?

- **Ingestão eficiente, ampla gama de conectores de dados:** a Databricks permite que você ingira dados de forma eficiente, trazendo apenas novos dados ou atualizações de tabelas. Com um conjunto crescente de conectores nativos para fontes de dados populares, bem como uma ampla rede de parceiros de ingestão de dados, você pode mover facilmente dados de sistemas isolados para sua plataforma de dados. Ingerir e armazenar seus dados no Delta Lake enquanto aproveita a confiabilidade e a escalabilidade da Data Intelligence Platform é o primeiro passo para extrair valor de seus dados e acelerar a inovação.
- **Processamento de fluxo de dados em tempo real:** a Data Intelligence Platform simplifica o desenvolvimento e as operações ao automatizar os aspectos de produção associados à construção e manutenção de cargas de trabalho de dados em tempo real. O Delta Live Tables oferece uma maneira declarativa de definir pipelines de ETL de streaming, e o Spark Structured Streaming auxilia na construção de aplicativos em tempo real para decisões em tempo real (consulte o diagrama na próxima página).

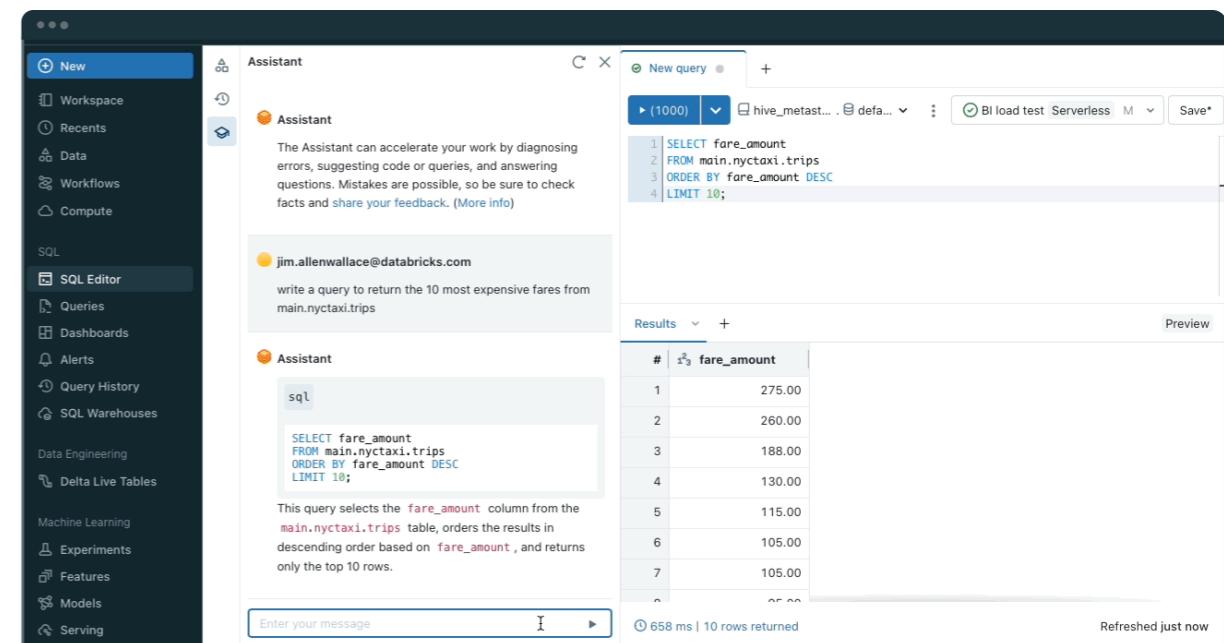


Um conjunto unificado de ferramentas para processamento de dados em tempo real

- **Pipelines de dados confiáveis em escala:** tanto o [Delta Live Tables](#) quanto o [Databricks Workflows](#) utilizam dimensionamento automático inteligente e gerenciamento de recursos auto-otimizado para lidar com cargas de trabalho de alta escala. Com a arquitetura lakehouse, a alta escalabilidade dos data lakes é combinada com a alta confiabilidade dos data warehouses, graças ao Delta Lake — o formato de armazenamento que está na base do lakehouse.
- **Qualidade dos dados:** alta confiabilidade — começando no nível de armazenamento com o [Delta Lake](#) e combinada com recursos específicos de qualidade de dados oferecidos pelo Delta Live Tables — garante alta qualidade dos dados. Esses recursos incluem a definição de “expectativas” de dados para lidar com dados corrompidos ou ausentes, bem como tentativas automáticas. Além disso, tanto o Databricks Workflows quanto o Delta Live Tables oferecem total observabilidade aos engenheiros de dados, tornando a resolução de problemas mais rápida e fácil.
- **Governança unificada com data sharing seguro:** o [Unity Catalog](#) fornece um modelo único de governança para toda a plataforma, garantindo que todos os conjuntos de dados e pipelines sejam governados de maneira consistente. Os conjuntos de dados são facilmente descobertos e podem ser compartilhados com segurança com equipes internas ou externas usando o Delta Sharing. Além disso, como o Unity Catalog é uma solução de governança multiplataforma, ele oferece informações valiosas sobre a linhagem, facilitando a compreensão completa de como cada conjunto de dados e tabela é utilizado downstream e de onde se origina o upstream.

Além disso, os engenheiros de dados que utilizam a Data Intelligence Platform se beneficiam de inovações de ponta na forma de inteligência infundida por GenAI:

- **Produtividade impulsionada por IA:** particularmente útil para engenheiros de dados, o [Databricks Assistant](#) é um assistente de IA com reconhecimento de contexto que oferece uma API de conversação para realizar query de dados, gerar código, explicar queries de código e até corrigir problemas.



The screenshot shows the Databricks Data Intelligence Platform interface. On the left is a sidebar with navigation links: New, Workspace, Recents, Data, Workflows, Compute, SQL (SQL Editor, Queries, Dashboards, Alerts, Query History, SQL Warehouses), Data Engineering, Delta Live Tables, Machine Learning, Experiments, Features, Models, and Serving. The main area has tabs for Assistant and SQL. The SQL tab shows a query editor with the following code:

```

1 SELECT fare_amount
2 FROM main.nyctaxi.trips
3 ORDER BY fare_amount DESC
4 LIMIT 10;
    
```

Below the code, the Assistant pane displays a message from 'jim.allenwallace@databricks.com' asking for the 10 most expensive fares from 'main.nyctaxi.trips'. The results pane shows a table with 10 rows of fare amounts:

#	fare_amount
1	275.00
2	260.00
3	188.00
4	130.00
5	115.00
6	105.00
7	105.00
8	105.00
9	105.00
10	105.00

A note below the table explains: "This query selects the fare_amount column from the main.nyctaxi.trips table, orders the results in descending order based on fare_amount, and returns only the top 10 rows." At the bottom of the interface, there is a message input field and a status bar indicating "658 ms | 10 rows returned" and "Refreshed just now".

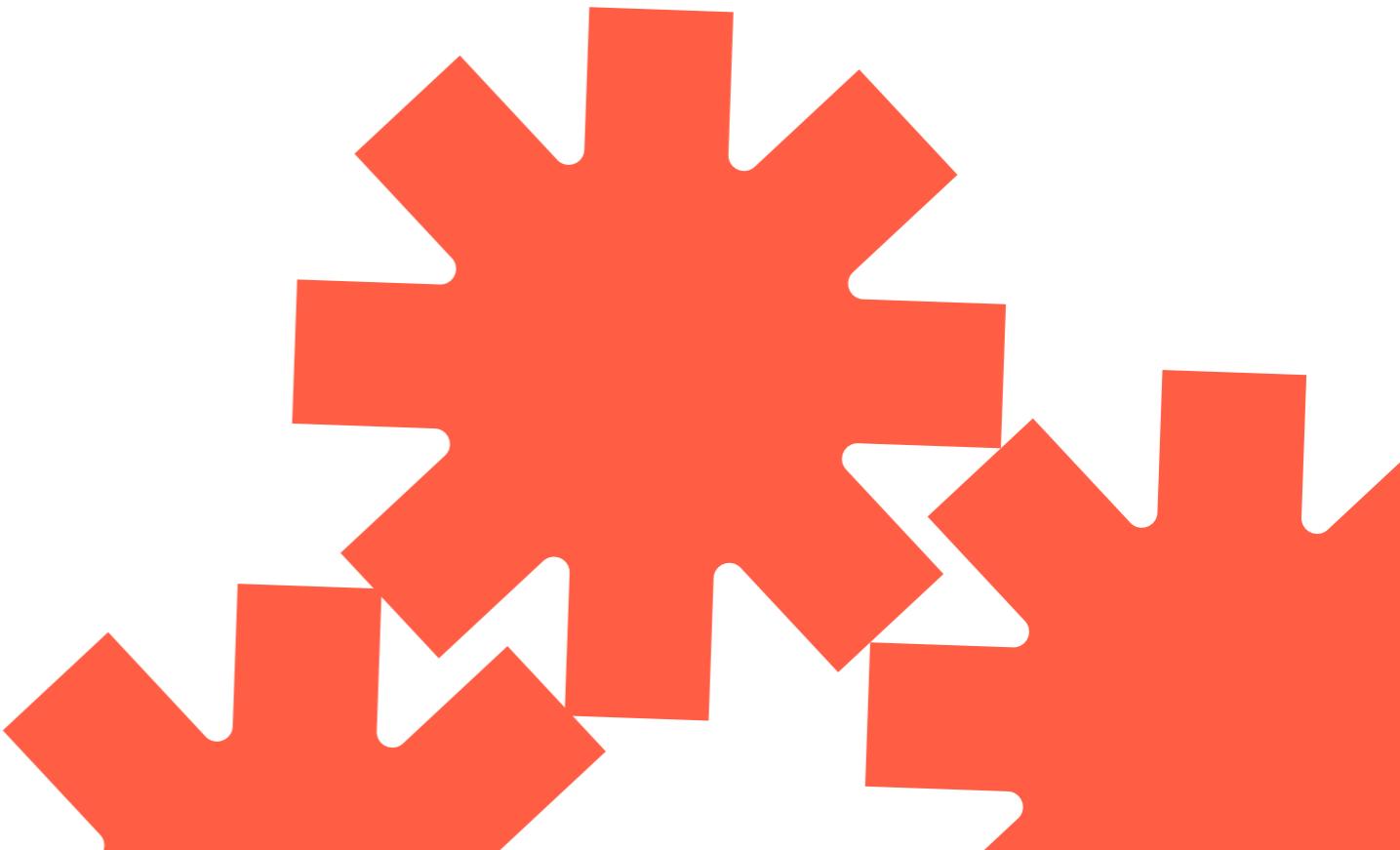
Conclusão

À medida que as organizações buscam inovar com IA, a engenharia de dados se torna crucial para o sucesso ao fornecer pipelines de dados confiáveis e em tempo real que viabilizam a IA. Com a Databricks Platform, construída em uma arquitetura de lakehouse e impulsionada por Inteligência de Dados, os engenheiros de dados estão preparados para enfrentar os desafios críticos no cenário moderno de dados. Ao utilizar as capacidades avançadas da Data Intelligence Platform, os engenheiros de dados não precisam dedicar tanto tempo gerenciando pipelines complexos ou enfrentando problemas de confiabilidade, escalabilidade e qualidade de dados. Em vez disso, eles podem se concentrar na inovação e em trazer mais valor para a organização.

SIGA AS PRÁTICAS RECOMENDADAS COMPROVADAS

Na próxima seção, descrevemos as melhores práticas para engenharia de dados e casos de uso de ponta a ponta extraídos de exemplos do mundo real. Desde a ingestão de dados e o processamento em tempo real até a orquestração e a federação de dados, você aprenderá a aplicar padrões comprovados e a aproveitar ao máximo as diferentes capacidades da Data Intelligence Platform.

À medida que você explora o restante deste guia, pode encontrar conjuntos de dados e exemplos de código nos vários **aceleradores de soluções Databricks**, para que possa colocar a mão na massa e começar a construir na Data Intelligence Platform.



02

Orientações e práticas recomendadas

Dicas e truques do Databricks Assistant para engenheiros de dados

por Jackie Zhang, Rafi Kurlansik e Richard Tomlinson

A revolução da IA generativa está transformando a maneira como as equipes trabalham, e o Databricks Assistant utiliza o melhor desses avanços. Ele permite que você consulte dados por meio de uma interface de conversação, tornando-o mais produtivo em seu Databricks Workspace. O Assistente é alimentado pelo DatabricksIQ, o mecanismo de inteligência de dados para Databricks, ajudando a garantir que seus dados estejam seguros, e que as respostas sejam precisas e adaptadas às especificidades de sua empresa. O Databricks Assistant permite que você descreva sua tarefa em linguagem natural para gerar, otimizar ou depurar código complexo sem interromper sua experiência como desenvolvedor.

Neste capítulo, discutiremos como aproveitar ao máximo o seu Databricks Assistant e nos concentraremos em como o Assistente pode melhorar a vida dos engenheiros de dados ao eliminar o tédio, aumentar a produtividade e a imersão, e acelerar o tempo para obtenção de valor. Faremos uma série de postagens direcionadas a diferentes perfis de profissionais de dados, então fique atento às próximas publicações voltadas para cientistas de dados, analistas de SQL e outros.

INGESTÃO

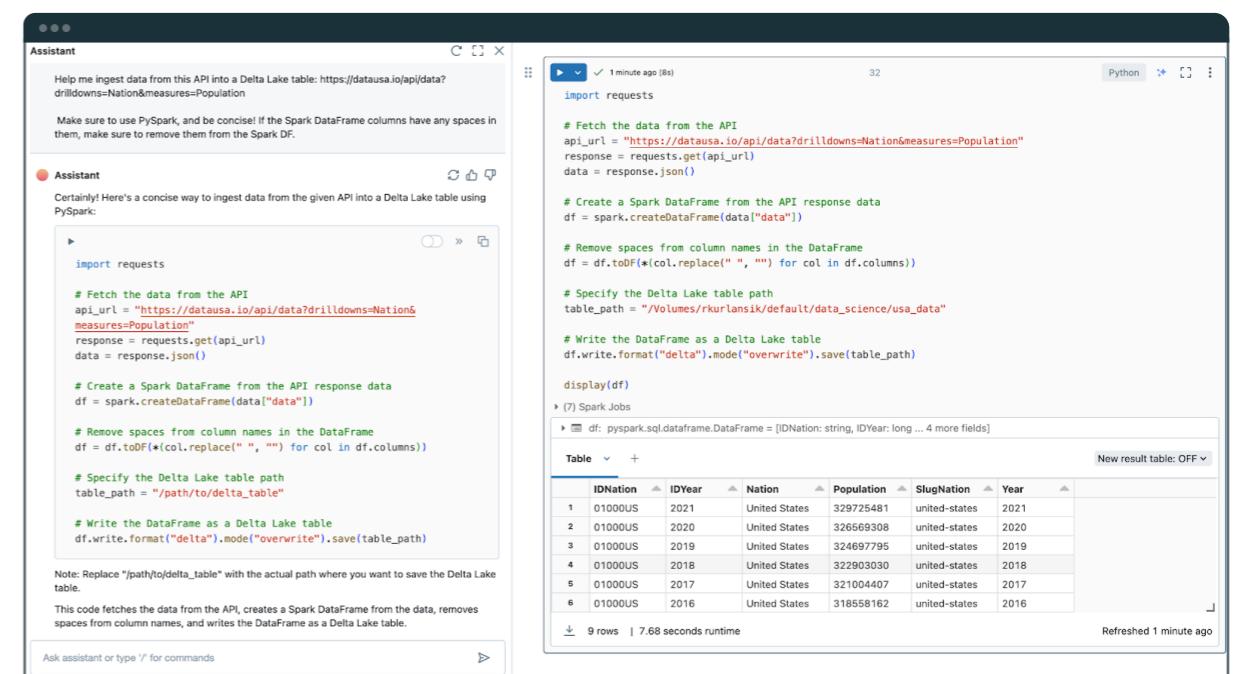
Ao trabalhar com a Databricks como engenheiro de dados, a ingestão de dados em tabelas do Delta Lake é frequentemente a primeira etapa. Vamos examinar dois exemplos de como o Assistente ajuda a carregar dados: um de APIs e outro de arquivos em armazenamento na nuvem. Para cada um, compartilharemos o prompt e os resultados. Conforme mencionado no [blog das 5 dicas](#), ser específico em um prompt proporciona os melhores resultados, uma técnica consistentemente utilizada neste artigo.

Para obter dados da API do datausa.io e carregá-los em uma tabela Delta Lake com Python, utilizamos o seguinte prompt:

Ajude-me a ingerir dados desta API em uma tabela Delta Lake:

<https://datausa.io/api/data?drilldowns=Nation&measures=Population>

Certifique-se de usar o PySpark e seja conciso. Se as colunas do DataFrame do Spark tiverem espaços, certifique-se de removê-los do DF do Spark.



The screenshot shows the Databricks Assistant interface with a Python code editor. The code fetches data from the DataUSA API, creates a Spark DataFrame, removes spaces from column names, specifies the Delta Lake table path, and writes the DataFrame as a Delta Lake table. The resulting DataFrame is displayed in a table view with 9 rows and a runtime of 7.68 seconds.

```

Assistant
Help me ingest data from this API into a Delta Lake table: https://datausa.io/api/data?drilldowns=Nation&measures=Population
Make sure to use PySpark, and be concise! If the Spark DataFrame columns have any spaces in them, make sure to remove them from the Spark DF.

Assistant
Certainly! Here's a concise way to ingest data from the given API into a Delta Lake table using PySpark:

import requests
# Fetch the data from the API
api_url = "https://datausa.io/api/data?drilldowns=Nation&measures=Population"
response = requests.get(api_url)
data = response.json()

# Create a Spark DataFrame from the API response data
df = spark.createDataFrame(data["data"])

# Remove spaces from column names in the DataFrame
df = df.toDF(*[col.replace(" ", "") for col in df.columns])

# Specify the Delta Lake table path
table_path = "/Volumes/rkurlansik/default/data_science/usa_data"

# Write the DataFrame as a Delta Lake table
df.write.format("delta").mode("overwrite").save(table_path)

display(df)

```

IDNation	IDYear	Nation	Population	SlugNation	Year	
1	01000US	2021	United States	329725481	united-states	2021
2	01000US	2020	United States	326569308	united-states	2020
3	01000US	2019	United States	324697795	united-states	2019
4	01000US	2018	United States	322903030	united-states	2018
5	01000US	2017	United States	321004407	united-states	2017
6	01000US	2016	United States	318558162	united-states	2016

Um prompt semelhante pode ser usado para importar arquivos JSON do armazenamento em nuvem para tabelas do Delta Lake, desta vez utilizando SQL:

Eu tenho arquivos JSON em um Volume UC aqui: /Volumes/rkurlansik/default/data_science/sales_data.json

Escreva um código para carregar esses dados em uma tabela Delta Lake. Utilize apenas SQL e seja conciso.

```
%sql
CREATE TABLE sales_data
USING DELTA
AS SELECT * FROM json './Volumes/rkurlansik/default/data_science/sales_data.json'

%sql
SELECT * FROM sales_data
```

Table company quarters

GnarlyTech [array] 0: quarter: "Q1" regions 0: name: "North America" products 0: {"name": "Product A", "sales": 200000, "sales_breakdown": {"by_customer_type": {"enterprise": 80000, "individual": 120000}, "online": 150000, "retail": 50000}, "units_sold": 10000} 1: {"name": "Product B", "sales": 200000, "sales_breakdown": {"by_customer_type": {"enterprise": 60000, "individual": 140000}, "online": 160000, "retail": 40000}, "units_sold": 5000} sales: 400000 1: {"name": "Europe", "products": [{"name": "Product C", "sales": 240000, "sales_breakdown":

TRANSFORMANDO DADOS NÃO ESTRUTURADOS EM ESTRUTURADOS

Seguindo os princípios de dados organizados, cada célula de uma tabela deve conter uma única observação com um tipo de dado apropriado. Strings complexas ou estruturas de dados aninhadas frequentemente conflitam com este princípio e, como consequência, o trabalho de engenharia de dados envolve a extração de dados estruturados a partir de dados não estruturados. Vamos explorar dois exemplos em que o Assistente se destaca nesta tarefa: utilizando expressões regulares e desmembrando estruturas de dados aninhadas.

Expressões regulares

Expressões regulares são uma forma de extrair dados estruturados de strings desordenadas, mas determinar a regex correta demanda tempo e é trabalhoso. Nesse sentido, o Assistente é uma vantagem para todos os engenheiros de dados que enfrentam dificuldades com expressões regulares.

Considere este exemplo usando a coluna *Título* do conjunto de dados IMDb:

^A _C Title	^A _C Certificate	^A _C Duration	^A _C Genre
1. The Shawshank Redemption (1994)	R	142 min	Drama

Esta coluna contém duas observações distintas: título do filme e ano de lançamento. Com o prompt a seguir, o Assistente identifica uma expressão regular apropriada para dividir a string em várias colunas.

Aqui está um exemplo da coluna Título em nosso conjunto de dados: 1. Um Sonho de Liberdade (1994). O nome do título ficará entre o número e os parênteses, e a data de lançamento ficará entre parênteses. Escreva uma função que extraia tanto a data de lançamento quanto o nome do título da coluna Título no DataFrame `imdb_raw`.

Generating regular expressions

Cell 26

Start typing or generate with AI (x + I)...

[Shift+Enter] to run
[Shift+Ctrl+Enter] to run selected text

Databricks Assistant

Accelerate your work by diagnosing errors, suggesting code or queries, and answering questions.

Check out [some examples](#) to get started. Make sure to verify any generated suggestions and share feedback so we can learn and improve.

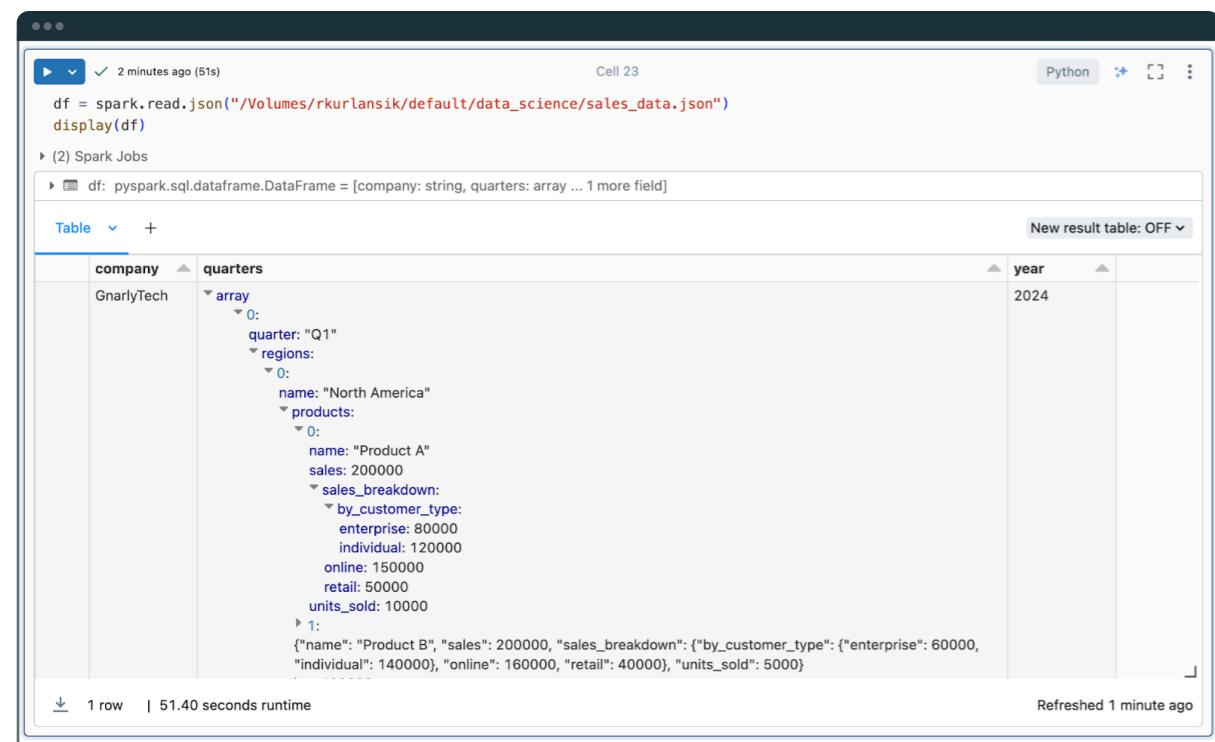
Find tables to query
Find some queries
Run data summarization

Here is an example of the Title column in our dataset: 1. The Shawshank Redemption (1994). The title name will be between the number and the parentheses, and the release date is between parentheses. Write a function that extracts both the release date and the title name from the Title column in the `imdb_raw` DataFrame. Then display the DataFrame. Only code, no explanatory text.

Fornecer um exemplo da string em nosso prompt ajuda o Assistente a encontrar o resultado correto. Caso você esteja lidando com dados sensíveis, sugerimos criar um exemplo fictício que siga o mesmo padrão. De qualquer forma, agora você tem um problema a menos para se preocupar no seu trabalho de engenharia de dados.

Estruturas aninhadas, matrizes (JSON, XML etc.)

Ao ingerir dados via API, arquivos JSON em armazenamento ou bancos de dados NoSQL, os DataFrames do Spark resultantes podem estar profundamente aninhados e difíceis de transformar corretamente. Confira esses dados de vendas simulados no formato JSON:



```
df = spark.read.json("/Volumes/rkurlansik/default/data_science/sales_data.json")
display(df)

(2) Spark Jobs
df: pyspark.sql.dataframe.DataFrame = [company: string, quarters: array ... 1 more field]

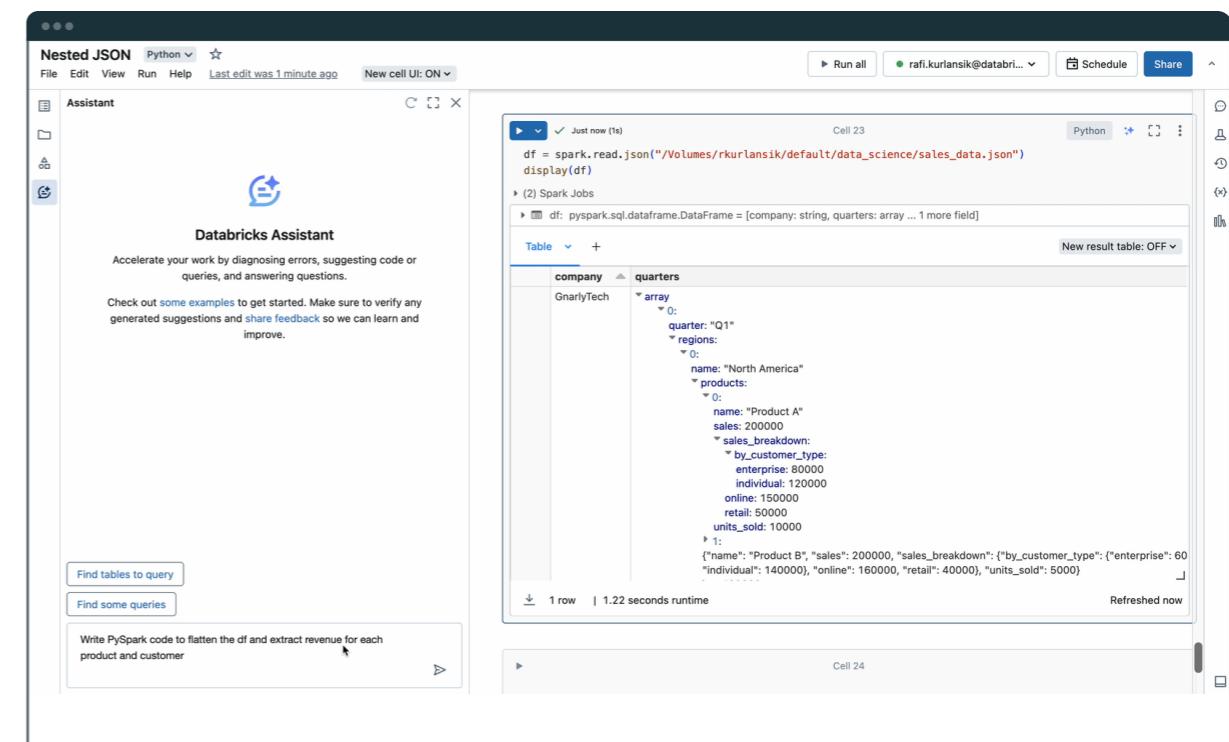
Table +
```

company	quarters	year
GnarlyTech	array <ul style="list-style-type: none"> 0: quarter: "Q1" regions: 0: name: "North America" products: 0: name: "Product A" sales: 200000 sales_breakdown: by_customer_type: enterprise: 80000 individual: 120000 online: 150000 retail: 50000 units_sold: 10000 1: {"name": "Product B", "sales": 200000, "sales_breakdown": {"by_customer_type": {"enterprise": 60000, "individual": 140000}, "online": 160000, "retail": 40000}, "units_sold": 5000} 	2024

1 row | 51.40 seconds runtime

Os engenheiros de dados podem ser solicitados a desestruturar a matriz aninhada e extrair métricas de receita para cada produto. Normalmente, essa tarefa exigiria um processo significativo de tentativa e erro — mesmo em um caso em que os dados sejam relativamente simples. O Assistente, no entanto, estando ciente dos esquemas de DataFrames que você tem na memória, gera código para concluir o trabalho. Utilizando um prompt simples, obtemos os resultados desejados em segundos.

Escreva um código PySpark para desestruturar o DataFrame e extrair a receita de cada produto e cliente



Nested JSON Python File Edit View Run Help Last edit was 1 minute ago New cell UI: ON

Assistant

Databricks Assistant

Accelerate your work by diagnosing errors, suggesting code or queries, and answering questions.

Check out some examples to get started. Make sure to verify any generated suggestions and share feedback so we can learn and improve.

Find tables to query

Find some queries

Write PySpark code to flatten the df and extract revenue for each product and customer

```
df = spark.read.json("/Volumes/rkurlansik/default/data_science/sales_data.json")
display(df)

(2) Spark Jobs
df: pyspark.sql.dataframe.DataFrame = [company: string, quarters: array ... 1 more field]

Table +
```

company	quarters
GnarlyTech	array <ul style="list-style-type: none"> 0: quarter: "Q1" regions: 0: name: "North America" products: 0: name: "Product A" sales: 200000 sales_breakdown: by_customer_type: enterprise: 80000 individual: 120000 online: 150000 retail: 50000 units_sold: 10000 1: {"name": "Product B", "sales": 200000, "sales_breakdown": {"by_customer_type": {"enterprise": 60000, "individual": 140000}, "online": 160000, "retail": 40000}, "units_sold": 5000}

1 row | 1.22 seconds runtime

REFATORAÇÃO, DEPURAÇÃO E OTIMIZAÇÃO

Outro cenário que os engenheiros de dados enfrentam é reescrever o código escrito por outros membros da equipe, sejam eles menos experientes ou que tenham deixado a empresa. Nesses casos, o Assistente pode analisar e explicar código mal escrito, compreendendo seu contexto e intenção. Ele pode sugerir algoritmos mais eficientes, refatorar o código para melhorar a legibilidade e adicionar comentários.

Aprimorando a documentação e a manutenibilidade

Este código em Python calcula o custo total dos itens em um carrinho de compras online.

•••

```

1  def calcular_total(itens_do_carrinho):
2      total = 0
3      for i in range(len(itens_do_carrinho)):
4          if itens_do_carrinho[i]['type'] == 'reservar':
5              discount = itens_do_carrinho[i]['preço'] * 0.05
6              total += itens_do_carrinho[i]['preço'] - discount
7          else:
8              total += itens_do_carrinho[i]['preço']
9      return total
10
11 itens_do_carrinho = [{"name": "Programação Python", "tipo": "reservar", "preço": 50},
12                         {"name": "Laptop", "tipo": "eletrônicos", "preço": 800}]
13 preço_total = calcular_total(itens_do_carrinho)

```

O uso de blocos condicionais neste código torna-o difícil de ler e inefficiente em larga escala. Além disso, não há comentários para explicar o que está acontecendo. Um bom ponto de partida é solicitar ao Assistente que explique o código detalhadamente, passo a passo. Assim que o engenheiro de dados compreender o código, o Assistente pode transformá-lo, tornando-o mais eficiente e legível com o seguinte comando:

Reescreva este código de forma mais eficiente, com comentários adequados e documentado conforme os padrões de documentação de funções do Python

O exemplo gerado abaixo documenta adequadamente o código e utiliza expressões geradoras em vez de blocos condicionais para otimizar a utilização da memória em grandes conjuntos de dados.

•••

```

1  def calcular_total(itens_do_carrinho):
2      """
3          Calcule o preço total dos itens no carrinho de compras.
4
5          Aplique um desconto de 5% para reservar itens. O total é calculado somando
6          os preços, considerando o desconto para reservas.
7
8          Parâmetros:
9          itens_do_carrinho (lista de dicionários): uma lista de dicionários, onde cada
10             dicionário representa um item no
11             carrinho com as chaves 'nome', 'tipo'
12             e 'preço'.
13
14          Retornos:
15             float: o preço total de todos os itens no carrinho após a aplicação de
16             descontos.
17             """
18
19             # Utilize uma expressão geradora para percorrer os itens do carrinho,
20             # aplicando um desconto de 5% nas reservas.
21             return sum(item['preço'] * 0.95 if item['type'] == 'reservar' else item['preço']
22             for item in itens_do_carrinho)
23
24
25             # Exemplo de uso
26             itens_do_carrinho = [{"name": "Programação Python", "tipo": "reservar", "preço": 50},
27                               {"name": "Laptop", "tipo": "eletrônicos", "preço": 800}]
28             preço_total = calcular_total(itens_do_carrinho)

```

Diagnóstico de erros

Inevitavelmente, engenheiros de dados precisarão fazer depuração. O Assistente elimina a necessidade de abrir várias tabs do navegador ou alternar contextos para identificar a causa dos erros no código, e manter o foco é um grande impulso de produtividade. Para compreender como isso funciona com o Assistente, vamos criar um DataFrame simples em PySpark e provocar um erro.

```

# Create a DataFrame
data = [
    {"name": "Python Programming", "type": "book", "price": 50},
    {"name": "Laptop", "type": "electronics", "price": 800}
]
columns = ["name", "price", "type"]

df = spark.createDataFrame(data, columns)
display(df)

```

```

# typo error
df = df.withColumn('fractional_price', df.price / 10)

display(df)

```

No exemplo acima, um erro de digitação é introduzido ao adicionar uma nova coluna ao DataFrame. O zero em “10” é na verdade a letra “O”, resultando em um erro de sintaxe de *literal decimal inválido*. O Assistente imediatamente oferece diagnosticar o erro. Ele identifica corretamente o erro de digitação e sugere o código corrigido que pode ser inserido no editor na célula atual. Diagnosticar e corrigir erros dessa forma pode economizar horas de tempo gasto na depuração.

Transpilando pandas para PySpark

Pandas é uma das bibliotecas de manipulação de dados mais bem-sucedidas em Python e é amplamente utilizado por cientistas de dados. Mantendo nossos dados de vendas em JSON, vamos imaginar uma situação em que um cientista de dados iniciante fez o melhor para achar os dados usando pandas. Não é o ideal, não segue as melhores práticas, mas gera o resultado correto:

```

import pandas as pd
import json

with open("/Volumes/rkurlansik/default/data_science/sales_data.json") as file:
    data = json.load(file)

# Prática inadequada: inicializar manualmente um DataFrame vazio e utilizar um
# loop for profundamente aninhado para preenchê-lo.
df = pd.DataFrame(columns=['empresa', 'ano', 'trimestre', 'nome_da_região', 'nome_do_'
                           'produto', 'unidades_vendidas', 'vendas_do_produto'])

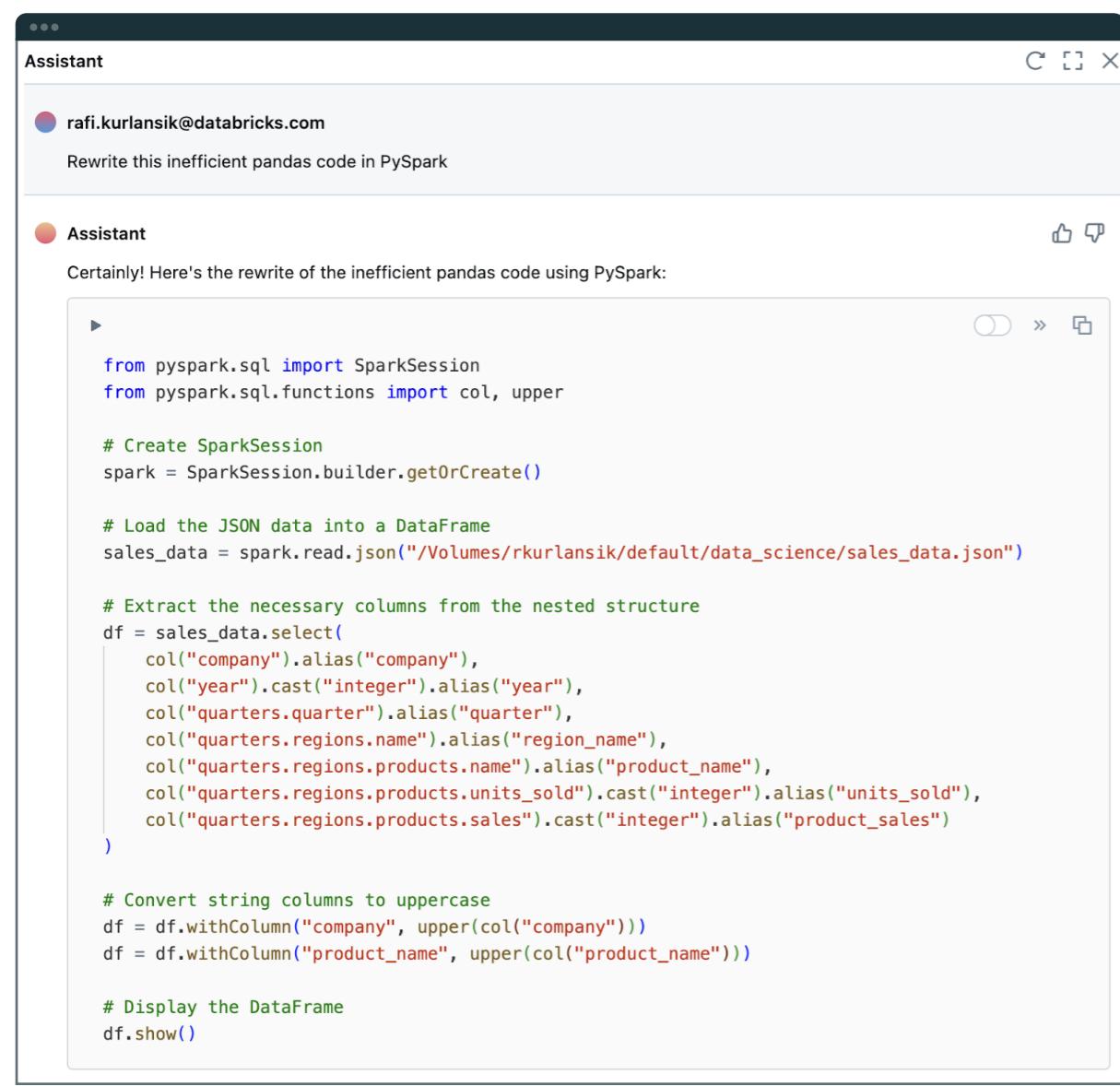
for o_trimestre in dados['trimestres']:
    for a_região in trimestre['regiões']:
        for produto in região['produtos']:
            df = df.append({
                'empresa': data['empresa'],
                'ano': data['ano'],
                'trimestre': trimestre['trimestre'],
                'nome_da_região': região['nome'],
                'nome_do_produto': produto['nome'],
                'unidades_vendidas': produto['unidades_vendidas'],
                'vendas_do_produto': produto['vendas']
            }, ignore_index=True)

# Conversão ineficiente de colunas após a anexação dos dados
df['ano'] = df['ano'].astype(int)
df['unidades_vendidas'] = df['unidades_vendidas'].astype(int)
df['vendas_do_produto'] = df['vendas_do_produto'].astype(int)

# Mistura de estilos de acesso e modificação do DataFrame no local de forma
# inconsistente
df['empresa'] = df.empresa.aplicar(lambda x: x.upper())
df['nome_do_produto'] = df['nome_do_produto'].str.upper()

```

Por default, o Pandas é limitado a ser executado em uma única máquina. O engenheiro de dados não deve colocar este código em produção e executá-lo em bilhões de linhas de dados até que seja convertido para PySpark. Este processo de conversão inclui garantir que o engenheiro de dados comprehenda o código e o reescreva de uma forma que seja sustentável, testável e com bom desempenho. O Assistente mais uma vez encontra uma solução melhor em segundos.



The screenshot shows the Databricks Assistant interface. The user has asked to "Rewrite this inefficient pandas code in PySpark". The AI has responded with the following Python code:

```

from pyspark.sql import SparkSession
from pyspark.sql.functions import col, upper

# Create SparkSession
spark = SparkSession.builder.getOrCreate()

# Load the JSON data into a DataFrame
sales_data = spark.read.json("/Volumes/rkurlansik/default/data_science/sales_data.json")

# Extract the necessary columns from the nested structure
df = sales_data.select(
    col("company").alias("company"),
    col("year").cast("integer").alias("year"),
    col("quarters.quarter").alias("quarter"),
    col("quarters.regions.name").alias("region_name"),
    col("quarters.regions.products.name").alias("product_name"),
    col("quarters.regions.products.units_sold").cast("integer").alias("units_sold"),
    col("quarters.regions.products.sales").cast("integer").alias("product_sales")
)

# Convert string columns to uppercase
df = df.withColumn("company", upper(col("company")))
df = df.withColumn("product_name", upper(col("product_name")))

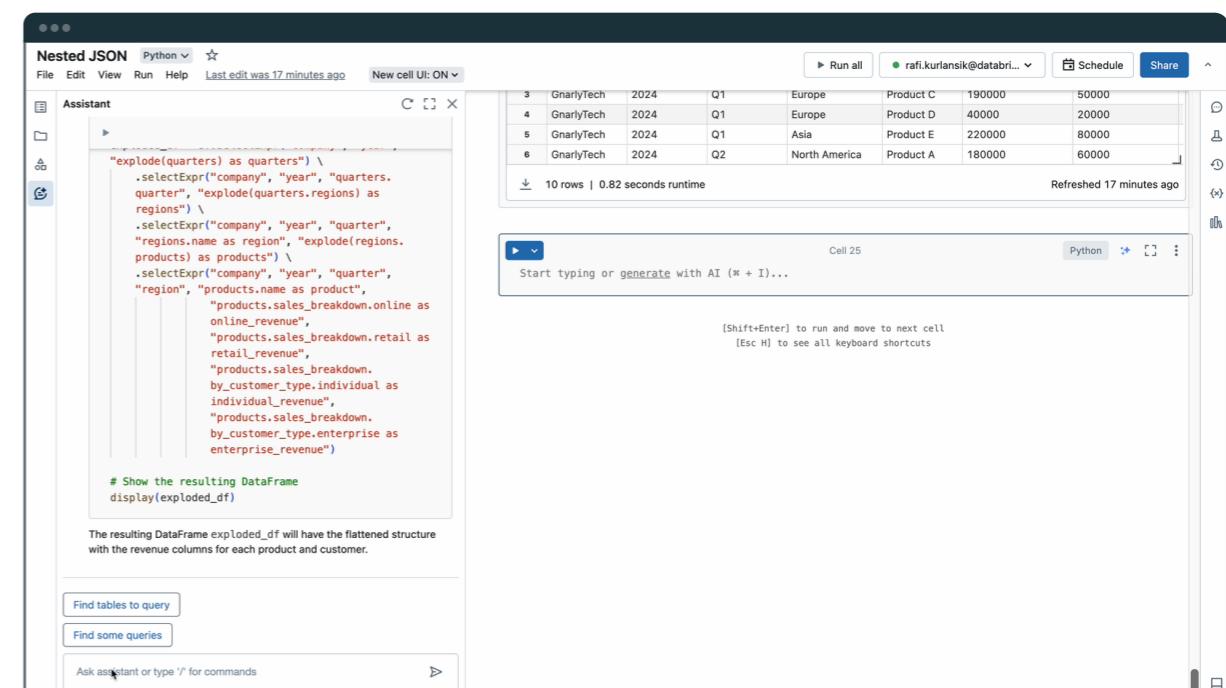
# Display the DataFrame
df.show()

```

Observe que o código gerado inclui a criação de uma `SparkSession`, que não é necessária na Databricks. Às vezes, o Assistente, como qualquer LLM, pode estar errado ou ter alucinações. Você, o engenheiro de dados, é o autor final do seu código e é importante revisar e entender qualquer código gerado antes de prosseguir para a próxima tarefa. Se você notar esse tipo de comportamento, ajuste seu prompt de acordo.

ESCREVENDO TESTES

Uma das etapas mais importantes na engenharia de dados é escrever testes para garantir que a lógica de transformação do DataFrame esteja correta e para detectar qualquer dado corrompido que passe pelo seu pipeline. Continuando com nosso exemplo dos dados de vendas JSON, o Assistente torna fácil testar se alguma das colunas de receita é negativa – desde que os valores nas colunas de receita não sejam menores que zero, podemos ter certeza de que nossos dados e transformações neste caso estão corretos.



The screenshot shows a Databricks notebook cell titled "Nested JSON". The code in the cell flattens a nested JSON structure and displays the resulting DataFrame:

```

.explode("quarters") \
.selectExpr("company", "year", "quarters.quarter", "explode(quarters.regions) as regions") \
.selectExpr("company", "year", "quarter", "regions.name as region", "explode(regions.products) as products") \
.selectExpr("company", "year", "quarter", "region", "products.name as product",
           "products.sales_breakdown.online as online_revenue",
           "products.sales_breakdown.retail as retail_revenue",
           "products.sales_breakdown.by_customer_type.individual as individual_revenue",
           "products.sales_breakdown.by_customer_type.enterprise as enterprise_revenue")

```

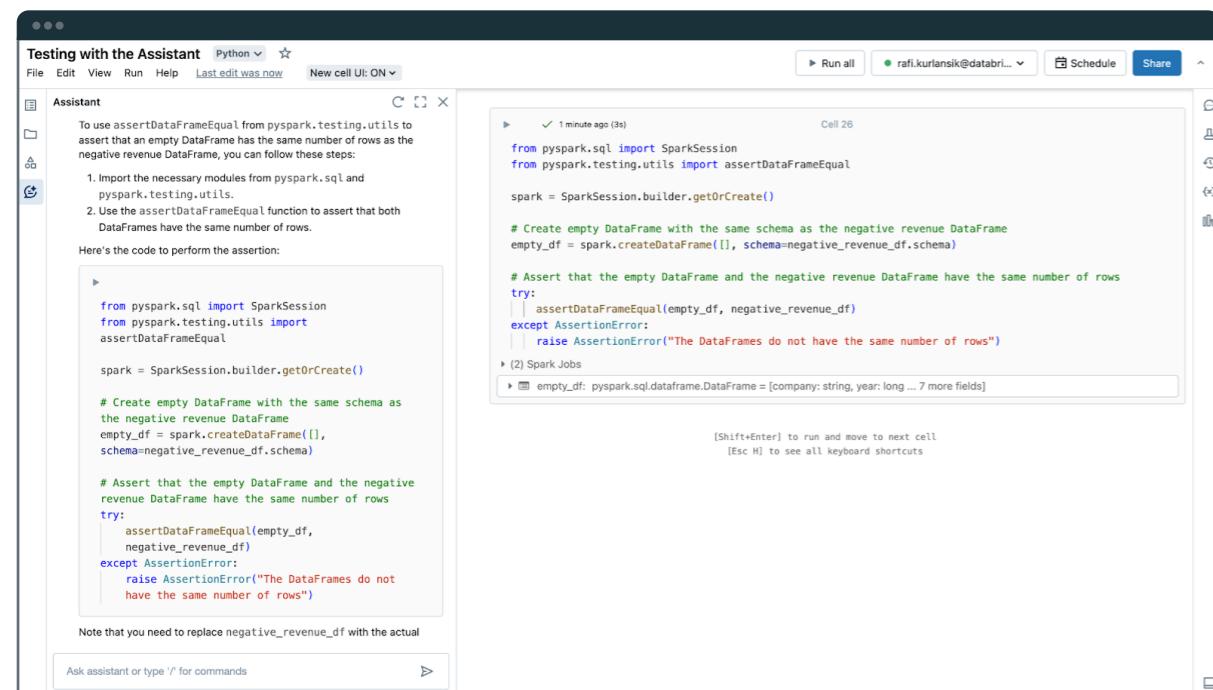
The resulting DataFrame `exploded_df` will have the flattened structure with the revenue columns for each product and customer.

	Company	Year	Quarter	Region	Product	Revenue Type	Value
3	GnarlyTech	2024	Q1	Europe	Product C	Online	190000
4	GnarlyTech	2024	Q1	Europe	Product D	Retail	40000
5	GnarlyTech	2024	Q1	Asia	Product E	Individual	220000
6	GnarlyTech	2024	Q2	North America	Product A	Enterprise	180000

Podemos expandir essa lógica pedindo ao Assistente que integre o teste à funcionalidade de teste nativa do PySpark, usando o seguinte prompt:

Escreva um teste usando assertDataFrameEqual do pyspark.testing.utils para verificar se um DataFrame vazio possui o mesmo número de linhas que o nosso DataFrame de receita negativa.

O Assistente atende, fornecendo código funcional para impulsionar nossos esforços de teste.



```

Testing with the Assistant Python ▾ ☆
File Edit View Run Help Last edit was now New cell UI: ON ▾
Run all rafi.kurlansik@databricks.com Schedule Share ▾
Assistant
To use assertDataFrameEqual from pyspark.testing.utils to assert that an empty DataFrame has the same number of rows as the negative revenue DataFrame, you can follow these steps:
1. Import the necessary modules from pyspark.sql and pyspark.testing.utils.
2. Use the assertDataFrameEqual function to assert that both DataFrames have the same number of rows.
Here's the code to perform the assertion:
from pyspark.sql import SparkSession
from pyspark.testing.utils import assertDataFrameEqual
spark = SparkSession.builder.getOrCreate()
# Create empty DataFrame with the same schema as the negative revenue DataFrame
empty_df = spark.createDataFrame([], schema=negative_revenue_df.schema)
# Assert that the empty DataFrame and the negative revenue DataFrame have the same number of rows
try:
    assertDataFrameEqual(empty_df, negative_revenue_df)
except AssertionError:
    raise AssertionError("The DataFrames do not have the same number of rows")
Note that you need to replace negative_revenue_df with the actual
Ask assistant or type '/' for commands ▾

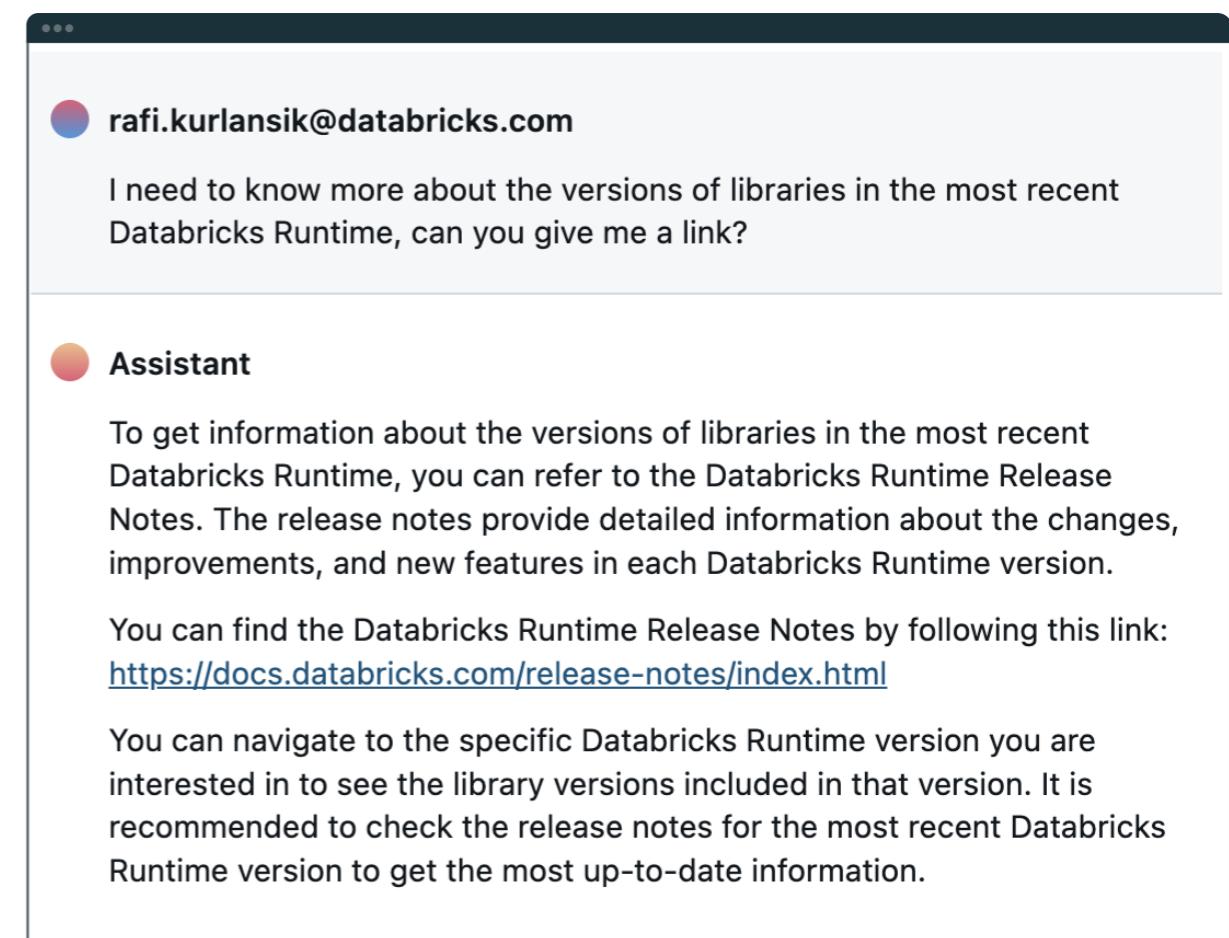
```

Este exemplo destaca que ser específico e adicionar detalhes ao seu prompt resulta em melhores resultados. Se apenas solicitarmos ao Assistente que escreva testes para nós sem fornecer detalhes, nossos resultados mostrariam maior variabilidade na qualidade. Ser específico e claro sobre o que estamos buscando — um teste que utiliza módulos PySpark e se baseia na lógica que ele desenvolveu — geralmente terá um desempenho melhor do que presumir que o Assistente pode adivinhar corretamente nossas intenções.

OBTENDO AJUDA

Além de uma capacidade geral de melhorar e compreender o código, o Assistente possui conhecimento de toda a documentação e da Base de Conhecimento da Databricks. Essas informações são indexadas regularmente e disponibilizadas como contexto adicional para o Assistente através de uma arquitetura RAG. Isso permite que os usuários busquem por funcionalidades e configurações do produto sem sair da Databricks Platform.

Por exemplo, se você deseja obter detalhes sobre o ambiente do sistema para a versão do Databricks Runtime que está utilizando, o Assistente pode direcioná-lo para a página correta na documentação da Databricks.



rafi.kurlansik@databricks.com

I need to know more about the versions of libraries in the most recent Databricks Runtime, can you give me a link?

Assistant

To get information about the versions of libraries in the most recent Databricks Runtime, you can refer to the Databricks Runtime Release Notes. The release notes provide detailed information about the changes, improvements, and new features in each Databricks Runtime version.

You can find the Databricks Runtime Release Notes by following this link: <https://docs.databricks.com/release-notes/index.html>

You can navigate to the specific Databricks Runtime version you are interested in to see the library versions included in that version. It is recommended to check the release notes for the most recent Databricks Runtime version to get the most up-to-date information.

O Assistente é capaz de lidar com perguntas simples, descriptivas e conversacionais, melhorando a experiência do usuário na navegação pelos recursos da Databricks e na resolução de problemas. Ele pode até ajudar a orientar os usuários no envio de tickets de suporte. Para mais detalhes, leia o artigo do anúncio.

CONCLUSÃO

A dificuldade de acesso à engenharia de dados de qualidade foi reduzida graças ao poder da IA generativa com o Databricks Assistant. Seja você um novato em busca de ajuda sobre como trabalhar com estruturas de dados complexas ou um veterano experiente que deseja que expressões regulares sejam escritas, o Assistente melhorará sua qualidade de vida. Sua competência principal de compreender, gerar e documentar códigos aumenta a produtividade de engenheiros de dados de todos os níveis de habilidade. Para saber mais, consulte a [documentação da Databricks](#) sobre como começar a trabalhar com o Databricks Assistant hoje mesmo.

03

Notebooks e conjuntos de dados
prontos para uso

Esta seção inclui vários aceleradores de soluções – exemplos gratuitos e prontos para uso de soluções de dados de diferentes setores, desde varejo até manufatura e saúde. Cada um dos seguintes cenários inclui notebooks com código e instruções passo a passo para ajudar você a começar. Obtenha experiência prática com a Databricks Data Intelligence Platform testando o seguinte:



Eficácia geral do equipamento

Ingira dados de sensores de equipamentos para geração de métricas e tomada de decisão baseada em dados

[Explore a solução](#)

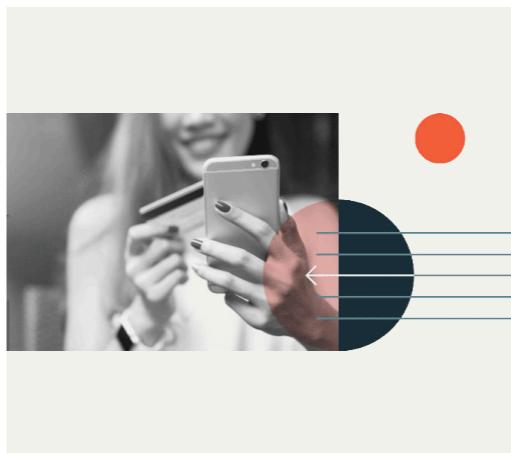
Análises de ponto de venda em tempo real

Calcule os estoques atuais de vários produtos em diversas lojas com o Delta Live Tables

[Explore a solução](#)

Gêmeos digitais

Aproveite os gêmeos digitais – representações virtuais de dispositivos e objetos – para otimizar as operações e obter insights

[Explore a solução](#)

Mecanismos de recomendação para personalização

Melhore a experiência do usuário e a conversão dos clientes com recomendações personalizadas

[Explore a solução](#)

Compreender os dados de transparência de preços

Ingira com eficiência grandes conjuntos de dados de saúde para criar transparência de preços e entender melhor os custos de assistência médica

[Explore a solução](#)

Aceleradores de soluções adicionais com notebooks prontos para uso podem ser encontrados aqui:

[Aceleradores de soluções da Databricks](#)

04

Estudos de caso



Cox AUTOMOTIVE



INDÚSTRIA
Automotiva

SOLUÇÃO

ESG orientado por dados,
Resolução de entidades de
cliente, Previsão de demanda,
Correspondência de produtos

PLATAFORMA

Workflows, Unity Catalog, Delta
Sharing, ETL

NUVEM
Azure

Cox Automotive – transformando a maneira como o mundo compra, vende e utiliza veículos

“Usamos o Databricks Workflows como nossa ferramenta de orquestração default para realizar ETL e permitir a automação de cerca de 300 jobs, dos quais aproximadamente 120 estão programados para serem executados regularmente.”

— Robert Hamlet, engenheiro-chefe de dados, serviços de dados corporativos, Cox Automotive

A Cox Automotive Europe é parte da Cox Automotive, a maior organização de serviços automotivos do mundo, e está em uma missão para transformar a forma como o mundo compra, vende, possui e utiliza veículos. Eles colaboram com fabricantes de automóveis, frotas e varejistas para otimizar o desempenho e a rentabilidade ao longo de todo o ciclo de vida do veículo. Seus negócios são estruturados em torno das necessidades centrais dos clientes em soluções de veículos, remarketing, financiamento, varejo e mobilidade. Suas marcas na Europa incluem Manheim, Dealer Auction, NextGear Capital, Modix e Codeweavers.

A equipe de serviços de dados empresariais da Cox recentemente desenvolveu uma plataforma para consolidar os dados da empresa e permitir que seus cientistas de dados criem novos produtos e serviços orientados por dados de forma mais rápida e fácil. Para capacitar sua pequena equipe de engenharia a unificar dados e análises em uma única plataforma, enquanto permite a orquestração e governança, a equipe de serviços de dados empresariais recorreu à Databricks Data Intelligence Platform, Workflows, Unity Catalog e Delta Sharing.

ORQUESTRAÇÃO E OBSERVABILIDADE FÁCEIS MELHORAM A CAPACIDADE DE ENTREGAR VALOR

A equipe de serviços de dados empresariais da Cox Automotive mantém uma plataforma de dados que atende principalmente clientes internos em várias unidades de negócios, embora também mantenha alguns feeds de dados para terceiros. A equipe de serviços de dados empresariais coleta dados de várias fontes internas e unidades de negócios. "Usamos o Databricks Workflows como nossa ferramenta de orquestração default para realizar ETL e permitir a automação de cerca de 300 jobs, dos quais aproximadamente 120 estão programados para serem executados regularmente", diz Robert Hamlet, engenheiro-chefe de dados, serviços de dados corporativos, na Cox Automotive.

Os jobs podem ser realizados semanalmente, diariamente ou por hora. Atualmente, a quantidade de dados processados nos pipelines de produção é de aproximadamente 720 GB por dia. Os jobs agendados são extraídos de diferentes áreas, tanto dentro quanto fora da empresa. Hamlet utiliza o Databricks Workflows para entregar dados à equipe de ciência de dados, à equipe interna de relatórios de dados através do Tableau, ou diretamente no Power BI. "O Databricks Workflows tem uma excelente interface de usuário que permite agendar rapidamente qualquer tipo de fluxo de trabalho, seja um notebook ou JAR", afirma Hamlet. "A parametrização tem sido especialmente útil. Ela nos fornece pistas sobre como podemos transferir jobs entre ambientes. O Workflows possui todos os recursos que você deseja em um orquestrador."

Hamlet também gosta da observabilidade em cada execução de fluxo de trabalho e das notificações de falhas do Workflows, para que possam se antecipar aos problemas rapidamente e solucioná-los antes que a equipe de ciência de dados seja impactada. "Utilizamos o recurso de notificações de job para enviar notificações de falha para um webhook, que está conectado à nossa conta do Microsoft Teams", afirma ele. "Se recebemos um alerta, acessamos a Databricks para verificar o que está acontecendo. É muito útil poder examinar os logs de execução e ver quais erros ocorreram. E o recurso Reparar execução é útil para remover imperfeições do seu histórico impecável."

O UNITY CATALOG E O DELTA SHARING APRIMORAM O ACESSO AOS DADOS ENTRE AS EQUIPES.

A equipe de Hamlet recentemente começou a utilizar o Unity Catalog para gerenciar o acesso aos dados, aprimorando o método existente, que carecia de granularidade e era difícil de administrar. "Com nosso novo workspace, estamos tentando aplicar mais princípios de DevOps, infraestrutura como código e grupos sempre que possível", ele diz. "Quero gerenciar facilmente o acesso a uma ampla gama de dados para diversos grupos e entidades, e desejo que seja o mais simples possível para minha equipe realizar essa tarefa. O Unity Catalog é a solução para isso."

A equipe de serviços de dados empresariais também utiliza o Delta Sharing, que se integra nativamente ao Unity Catalog e permite que a Cox gerencie e audite centralmente os dados compartilhados fora da equipe de serviços de dados empresariais, garantindo segurança e governança robustas. "O Delta Sharing facilita o compartilhamento seguro de dados com unidades de negócios e subsidiárias sem copiá-los ou replicá-los", diz Hamlet. "Isso nos permite compartilhar dados sem que o destinatário precise ter uma identidade em nosso workspace."

OLHANDO PARA O FUTURO: INCORPORAÇÃO DE CAPACIDADES ADICIONAIS DA PLATAFORMA DE INTELIGÊNCIA DE DADOS

No futuro, Hamlet planeja usar o Delta Live Tables (DLT) para facilitar a criação e o gerenciamento de pipelines de dados em batch e streaming que fornecem dados na Databricks Data Intelligence Platform. O DLT ajudará as equipes de engenharia de dados a simplificar o desenvolvimento e a gestão de ETL. Por fim, Hamlet poderá usar o Delta Sharing para compartilhar dados de forma fácil e segura com fornecedores e parceiros externos, atendendo às exigências de segurança e conformidade. “O DLT nos oferece a oportunidade de simplificar para nossa equipe. O agendamento de Delta Live Tables será outra área em que utilizaremos Workflows”, ele diz.

Hamlet também está ansioso para usar os recursos de linhagem de dados no Unity Catalog para fornecer à sua equipe uma visão de ponta a ponta de como os dados fluem no lakehouse para atender aos requisitos de conformidade de dados e realizar a análise de impacto das alterações de dados. “Esse é um recurso que me entusiasma”, diz Hamlet. “Por fim, espero que alcancemos um ponto em que todos os nossos dados estejam no lakehouse, permitindo-nos aproveitar melhor as integrações estreitas com aspectos como linhagem de dados e gerenciamento avançado de permissões.”

Explore mais práticas recomendadas e histórias de clientes em [O Livro Completo da Engenharia de Dados](#) (somente em inglês)

Dezenas de milhões de cargas de trabalho produtivas são executadas diariamente na Databricks

Insira e transforme facilmente dados em batch e streaming na [Databricks Data Intelligence Platform](#).

Orquestre fluxos de trabalho de produção confiáveis enquanto a Databricks gerencia automaticamente sua infraestrutura em escala. Aumente a produtividade de suas equipes com testes de qualidade de dados integrados e suporte às melhores práticas de desenvolvimento de software.

[Experimente a Databricks gratuitamente](#)

[Comece com uma demonstração gratuita](#)

Sobre a Databricks

A Databricks é a empresa de dados e IA. Mais de 10.000 organizações em todo o mundo — incluindo Block, Comcast, Condé Nast, Rivian, Shell e mais de 60% da Fortune 500 — contam com a Databricks Data Intelligence Platform para controlar seus dados e utilizá-los com IA. A Databricks tem sede em São Francisco, com escritórios em todo o mundo, e foi fundada pelos criadores originais do Lakehouse, Apache Spark™, Delta Lake e MLflow. Para saber mais, siga a Databricks no [LinkedIn](#), [X](#) e [Facebook](#).

