

Kompaktleitfaden

Data Engineering

Inhalt

Einführung in das Data Engineering auf Databricks 3

Empfehlungen und Best Practices14

 Tipps und Tricks zu Databricks Assistant für Data Engineers..... 15

Ausgearbeitete Notebooks und Datensätze 23

Fallstudien 25

 Cox Automotive26

01

Einführung in das Data Engineering auf Databricks

Einführung in das Data Engineering auf Databricks

Laut **einem kürzlich in der MIT Tech Review veröffentlichten Report** investieren bereits 88 % der befragten Organisationen bereits in generative KI (GenAI), setzen sie produktiv ein oder experimentieren damit. Darüber hinaus planen 71 %, eigene GenAI-Modelle zu entwickeln. Dieses wachsende Interesse an KI führt zu umfangreichen Investitionen, da KI im Begriff ist, in jeder Branche zu einem maßgeblichen Wettbewerbsvorteil zu werden. Weil aber immer mehr Organisationen bemüht sind, ihre proprietären Daten für diese Zwecke nutzbar zu machen, werden viele von ihnen mit derselben unangenehmen Wahrheit konfrontiert:

Ohne gute Daten funktionieren auch die besten GenAI-Modelle der Welt nicht.

Diese Tatsache unterstreicht, wie wichtig es ist, zuverlässige Datenpipelines aufzubauen, die große Datenmengen effizient erfassen oder streamen und eine hohe Datenqualität gewährleisten. Eine einheitliche Plattform und gutes Data Engineering sind wesentliche Erfolgsfaktoren für jede Daten- und KI-Initiative – vor allem im Zeitalter von GenAI.

Mit praktischen Anleitungen, nützlichen Mustern, Best Practices und praxisnahen Beispielen soll dieses E-Book ein Verständnis dafür vermitteln, wie Data Engineers mithilfe der **Databricks Data Intelligence Platform** die Herausforderungen dieser neuen Ära bewältigen.

Was ist Data Engineering?

Data Engineering bezeichnet die Praxis, Rohdaten aus unterschiedlichsten Datenquellen so aufzubereiten, dass sie für nachgelagerte Anwendungen wie Datenanalysen, Business Intelligence (BI) oder das Training von Machine-Learning-Modellen gespeichert und nutzbar sind. Kurz gesagt: Data Engineering ist der Prozess, bei dem Daten so aufbereitet werden, dass sie nutzbar und wertschöpfend eingesetzt werden können.

Ein effektiver Data-Engineering-Ansatz folgt im Kern einem dreistufigen Konzept:

1. Aufnehmen

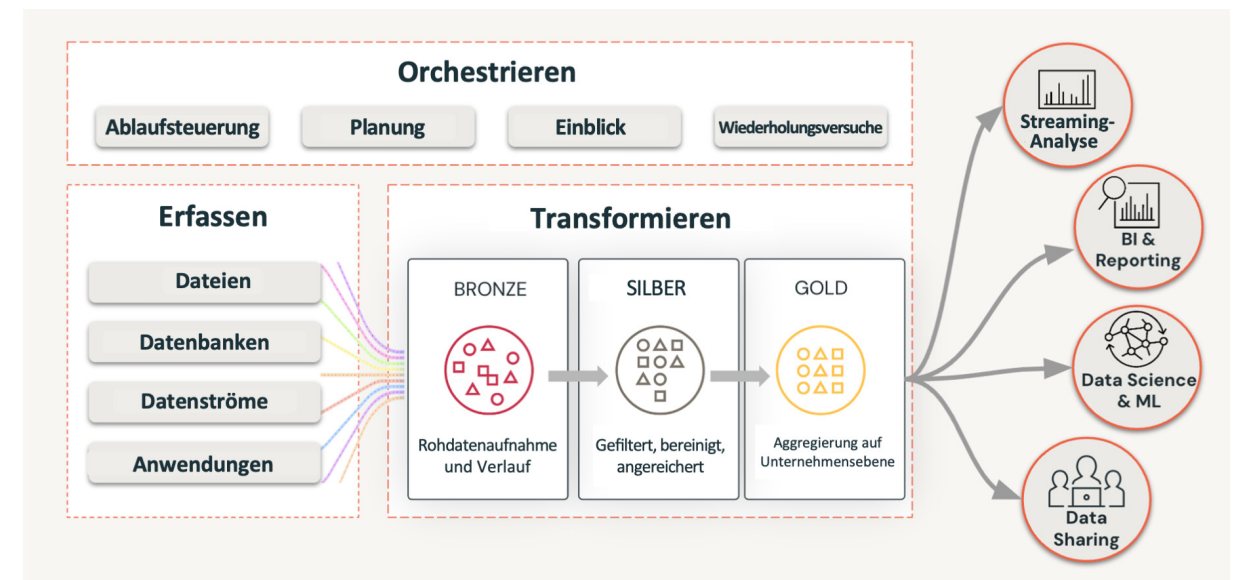
Die Datenaufnahme bezeichnet den Prozess, bei dem Daten aus einer oder mehreren Quellen auf eine Datenplattform übertragen werden. Solche Quellen können lokal oder in der Cloud gespeicherte Dateien, Datenbanken, Anwendungen oder zunehmend auch Datenströme sein, die Ergebnisse in Echtzeit liefern.

2. Transformieren

Bei der Datentransformation werden die eingelesenen Rohdaten in einer Reihe von Schritten (den so genannten „Transformationen“) gefiltert, standardisiert, bereinigt und schließlich aggregiert, damit sie in einer nutzbaren Form gespeichert werden können. Ein weit verbreitetes Muster ist die **Medaillon-Architektur**, die drei Stufen im Prozess definiert: Bronze, Silber und Gold.

3. Orchestrieren

Als „Datenorchestrierung“ wird die Art und Weise bezeichnet, wie eine Datenpipeline, die Aufnahme und Transformation durchführt, geplant und überwacht wird, sowie die Steuerung der verschiedenen Pipelineschritte und die Fehlerbehandlung (z. B. durch Ausführung eines Wiederholungslaufs).



Herausforderungen für das Data Engineering im KI-Zeitalter

Wie bereits erwähnt, ist Data Engineering der Schlüssel zur Gewinnung zuverlässiger Daten für KI-Initiativen. Data Engineers, die ETL-Pipelines und die Dateninfrastruktur aufbauen und betreuen, die den Analyse- und KI-Workloads zugrunde liegt, stehen in diesem schnelllebigen Umfeld vor besonderen Herausforderungen.

- **Uneinheitliche Datenquellen stellen die meisten Organisationen vor Probleme:** Die ISG prognostiziert, dass bis 2026 acht von zehn Unternehmen ihre Daten auf mehrere Cloud-Anbieter und lokale Rechenzentren an verschiedenen Standorten verteilt haben werden. Diese Dezentralisierung schafft eine Abhängigkeit von spezialisierten, isolierten Teams, ineffizienten Pipelines und einer kostenintensiven Entwicklung, verbunden mit einer langsamen Amortisierung. Dies schränkt die Nutzung von Daten ein und hemmt Innovationen.
- **Umgang mit Echtzeitdaten:** Ob von Mobil-Apps oder Sensoren in der Fertigung: Immer mehr Daten werden in Echtzeit gestreamt und erfordern daher auch eine latenzarme Verarbeitung, um für die Entscheidungsfindung in Echtzeit genutzt werden zu können.
- **Zuverlässige Skalierung von Datenpipelines:** Da Daten in großen Mengen und oft in Echtzeit eingehen, ist die Skalierung der Compute-Infrastruktur, auf der die Datenpipelines ausgeführt werden, insbesondere dann schwierig, wenn versucht wird, die Kosten niedrig und die Leistung hoch zu halten. Zu den wichtigsten Aufgaben von Data Engineers gehören der zuverlässige Betrieb von Datenpipelines, ihre Überwachung und die Fehlerbehebung bei Ausfällen.
- **Datenqualität:** „Garbage In, Garbage Out“: Eine hohe Datenqualität ist wesentlich, um hochwertige Modelle zu trainieren und umsetzbare Erkenntnisse aus Daten zu gewinnen. Die Sicherstellung der Datenqualität ist eine zentrale Aufgabe für Data Engineers.

- **Governance und Sicherheit:** Data Governance wird zu einer zentralen Herausforderung für alle Organisationen, deren Daten auf mehrere Systeme verteilt sind und auf die eine immer größere Anzahl interner Teams zu unterschiedlichsten Zwecken zugreifen können muss. Sicherheit und Governance von Daten sind zudem ein wichtiges regulatorisches Anliegen, mit dem viele Organisationen konfrontiert sind – vor allem in stark regulierten Branchen.

Diese Herausforderungen unterstreichen, wie wichtig es ist, die richtige Datenplattform zu wählen, um im Zeitalter der KI neue Wege zu beschreiten. Allerdings kann eine Datenplattform in diesem neuen Zeitalter auch über die bloße Bewältigung der Herausforderungen beim Aufbau von KI-Lösungen hinausgehen. Die passende Plattform kann die Erfahrung und Produktivität von Datenpraktikern – auch Data Engineers – verbessern, indem sie Informationen einfließen lässt und KI zur Unterstützung bei täglichen technischen Aufgaben nutzt.

Anders gesagt: Die neue Datenplattform ist eine Data *Intelligence* Plattform.

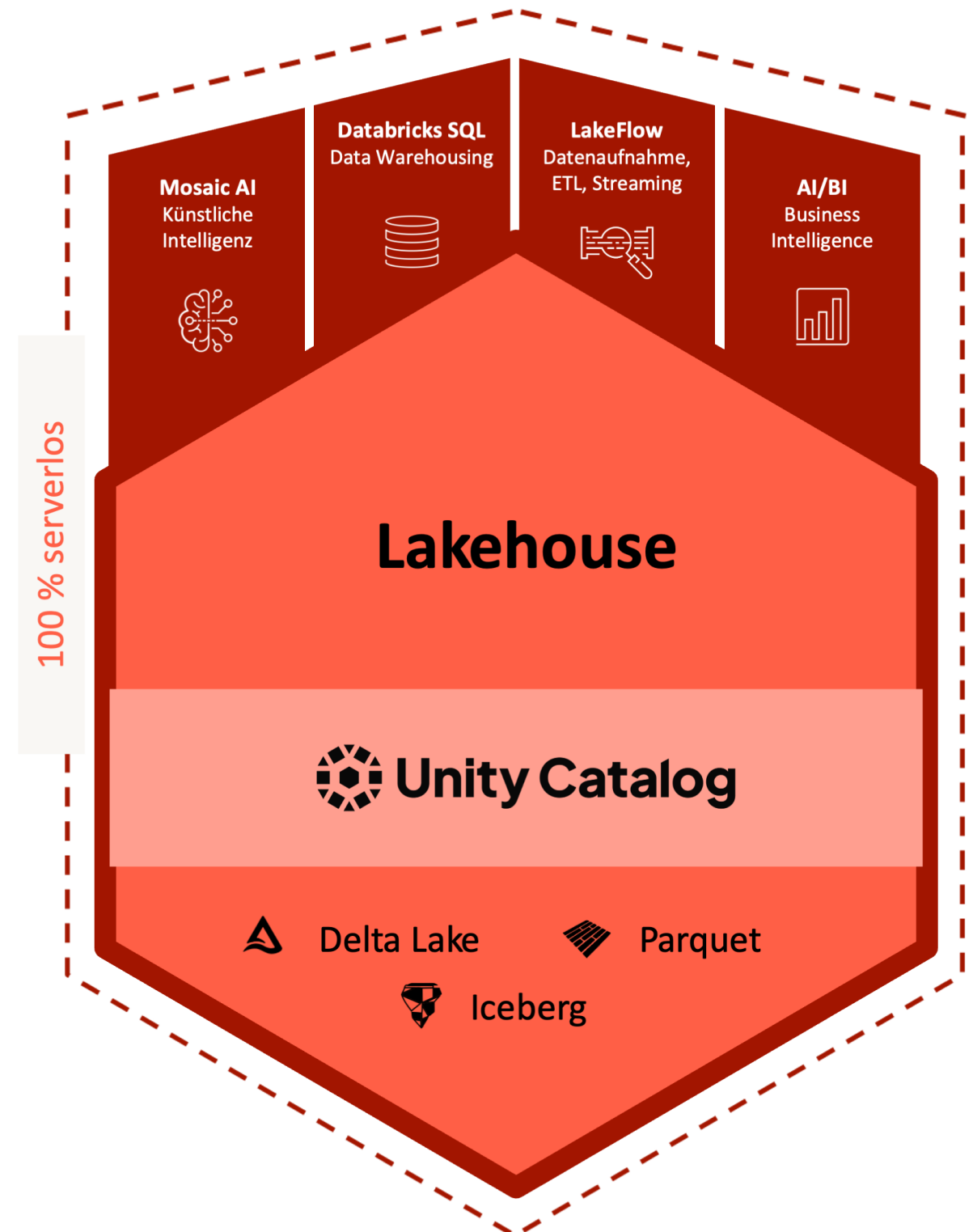
Die Databricks Data Intelligence Platform

Databricks verfolgt das Ziel, Daten und KI für alle zugänglich zu machen. So können Unternehmen ihre eigenen Daten produktiv nutzen, um individuelle Machine-Learning- und GenAI-Modelle zu entwickeln oder zu optimieren – und daraus neue Erkenntnisse für Innovation und Wachstum gewinnen.

Die Databricks Data Intelligence Platform basiert auf einer **Lakehouse-Architektur** und stellt so eine offene, einheitliche Basis für alle Daten und für Governance bereit. Angetrieben wird sie von einer Data-Intelligence-Engine, die die Eigenheiten Ihrer Daten berücksichtigt. Mit diesen Fähigkeiten als Grundlage ermöglicht die Data Intelligence Platform Databricks-Kunden die Ausführung einer Vielzahl von Workloads, von Business Intelligence und Data Warehousing bis hin zu KI und Data Science.

Damit Sie ein besseres Verständnis der Databricks Platform gewinnen, finden Sie nachfolgend eine Übersicht der zentralen Architekturkomponenten – mit besonderem Fokus auf den Bereich Data Engineering.

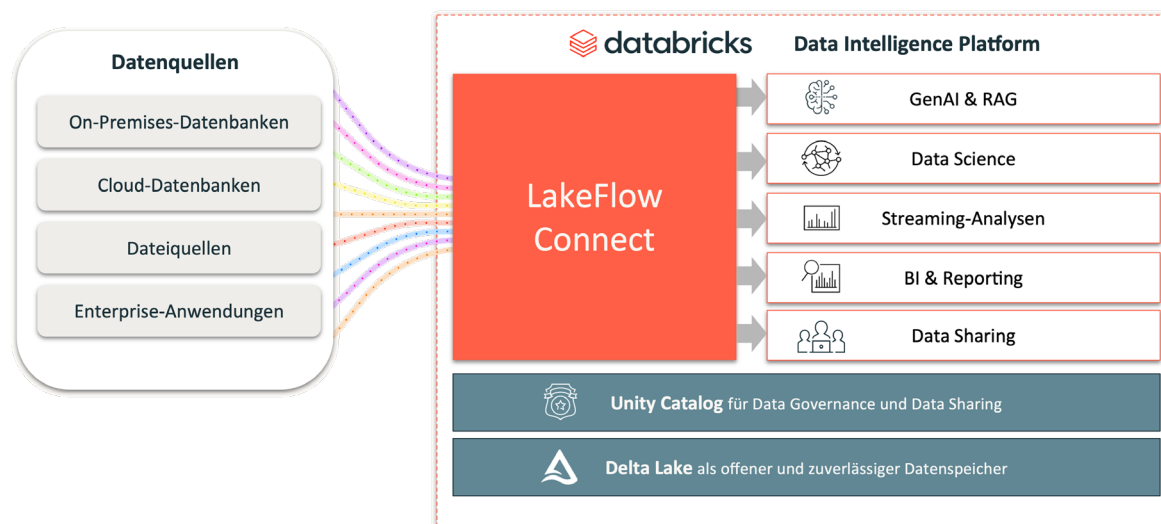
Mit der Databricks Data Intelligence Platform können Sie alle Ihre Daten- und KI-Initiativen realisieren. Als zu 100 % serverlose Plattform bietet sie Ihnen integrierte Funktionen wie Disaster Recovery, Kostenkontrolle und Enterprise-Sicherheit. Zu den wichtigsten Komponenten gehören Mosaic AI mit End-to-End-KI für generative und klassische KI, Databricks SQL, das leistungsstärkste Data Warehouse in der Cloud, Tools für eine effiziente Datenaufnahme und zuverlässige Transformation wie Workflows und Delta Live Tables (DLT), um sicherzustellen, dass Sie alle Ihre Daten für jede Workload effizient verwalten können, sowie AI/BI, das umfassend in Databricks SQL integriert ist und womit Sie Data Intelligence bequem auf Ihr gesamtes Unternehmen ausweiten können.



Datenaufnahme mit Databricks LakeFlow

Mit Databricks können Organisationen Daten aus verschiedenen Systemen effizient in eine zentrale, offene und einheitliche Lakehouse-Architektur verschieben. Databricks **LakeFlow Connect** bietet zur Datenaufnahme native Konnektoren für beliebige SaaS-Anwendungen, Datenbanken und Dateiquellen, mithilfe derer jeder Praktiker inkrementelle Datenpipelines beliebiger Größe erstellen kann. Diese integrierten Konnektoren gewährleisten eine effiziente, inkrementelle End-to-End-Aufnahme, eine unkomplizierte Einrichtung mit einer intuitiven Benutzeroberfläche oder API-Zugriff sowie Governance über Unity Catalog – alles basierend auf der Data Intelligence Platform. LakeFlow Connect ist Bestandteil von **LakeFlow**, der neuen Databricks-Lösung für einheitliches Data Engineering, die die Bereiche Aufnahme, Transformation und Orchestrierung abdeckt. Sie wird als erste dieser drei Komponenten eingeführt und ist mit bestehenden Tools kompatibel. Neben LakeFlow Connect gibt es mit Databricks Auto Loader auch einen Konnektor für die Cloud-Objektspeicherung, der mit strukturiertem Streaming und Delta Live Tables kompatibel ist.

Native Datenaufnahme für die Data Intelligence Platform



Datenzuverlässigkeit und -leistung mit Delta Lake

Um Offenheit, Zuverlässigkeit und ein durchgängiges Lifecycle-Management in Data Lakes zu ermöglichen, setzt die Lakehouse-Architektur von Databricks auf **Delta Lake** auf. Delta Lake ist ein leistungsstarkes Open-Source-Speicherformat. Es adressiert zentrale Herausforderungen wie die Verarbeitung strukturierter und unstrukturierter Daten, die Sicherstellung von Datenqualität, das datenschutzkonforme Löschen von Daten sowie die effiziente Modifikation von Daten zur Änderungserfassung (Change Data Capture). Benutzer von Delta Lake UniForm können nun Delta-Tabellen mit Hudi- und Apache Iceberg™-Clients lesen und behalten so die Kontrolle über ihre Daten. Darüber hinaus ermöglicht **Delta Sharing** die einfache und sichere gemeinsame Nutzung von Datasets innerhalb wie außerhalb der Organisation.

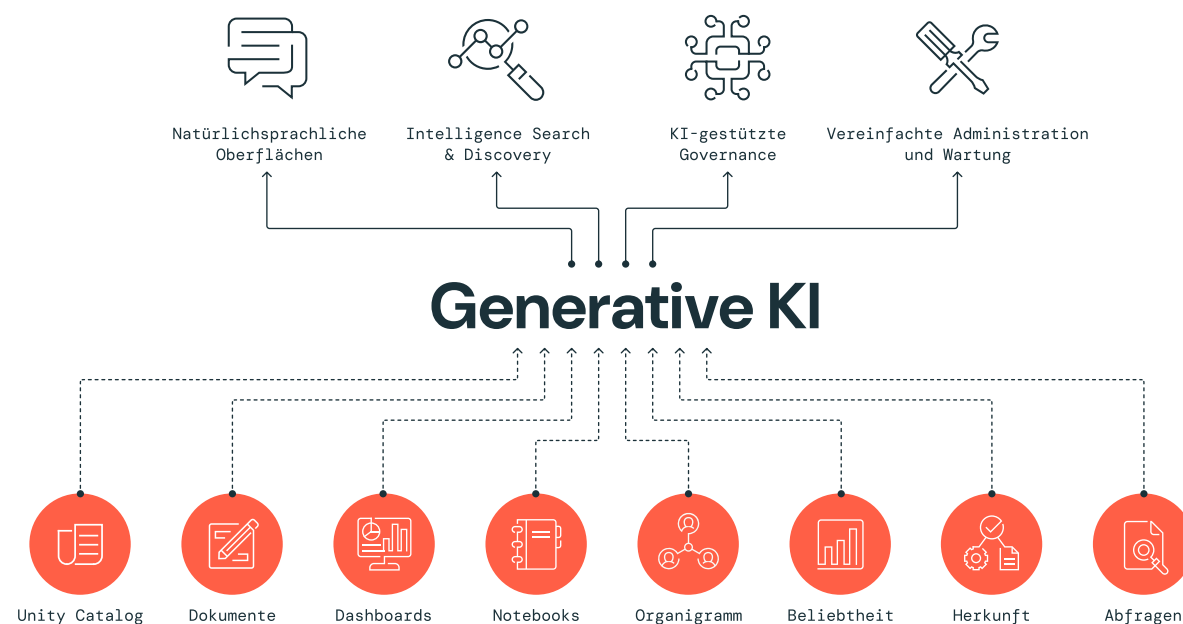
Einheitliche Governance mit Unity Catalog

Mit **Unity Catalog** profitieren Data-Engineering- und Governance-Teams von einem unternehmensweiten Datenkatalog mit einer zentralen Oberfläche, über die sie Berechtigungen verwalten, Audits zentralisieren, die Datenherkunft automatisch bis auf Spaltenebene nachverfolgen und Daten plattform-, cloud- und regionsübergreifend austauschen können.



Generative KI mit der Data-Intelligence-Engine

Databricks bringt KI zu Ihren Daten, damit Sie KI in die Welt hinaustragen können. Das Herzstück der **Databricks Data Intelligence Platform** ist dabei die Data-Intelligence-Engine. Dank Databricks können Sie KI auf Basis Ihrer eigenen Daten erfolgreich nutzen, um Erkenntnisse zu demokratisieren und Kosten zu senken. Als erstes System seiner Art nutzt die Data-Intelligence-Engine KI, um alle Teile der Databricks Data Intelligence Platform zu versorgen. Unter Verwendung von Signalen aus Ihrer gesamten Databricks-Umgebung – einschließlich Unity Catalog, Dashboards, Notebooks, Datenpipelines und Dokumentation –, erstellt die Data-Intelligence-Engine hochgradig spezialisierte und treffsichere GenAI-Modelle, die Ihre Daten, Ihre Nutzungsmuster und Ihre Geschäftsterminologie verstehen.



Zuverlässige Datenpipelines und Streaming-Verarbeitung in Echtzeit mit Delta Live Tables

Delta Live Tables ist ein deklaratives ETL-Framework, mit dem Datenteams ETL-Prozesse im Streaming- und Batch-Verfahren vereinfachen und rentabler gestalten können. Sie müssen lediglich die Transformationen definieren, die Sie an Ihren Daten vornehmen möchten; die DLT-Pipelines erledigen dann automatisch Aufgabenorchestrierung, Cluster-Management, Überwachung, Datenqualität und Fehlerverwaltung. So sind Engineers in der Lage, ihre Daten wie Code zu behandeln und durch Anwendung von Best Practices aus dem modernen Software Engineering – wie Testen, Fehlerbehandlung, Überwachung und Dokumentation – zuverlässige Pipelines umfassend einzusetzen. DLT unterstützt sowohl Python als auch SQL vollständig und ist gleichermaßen auf das Arbeiten mit Streaming- und Batch-Workloads zugeschnitten.

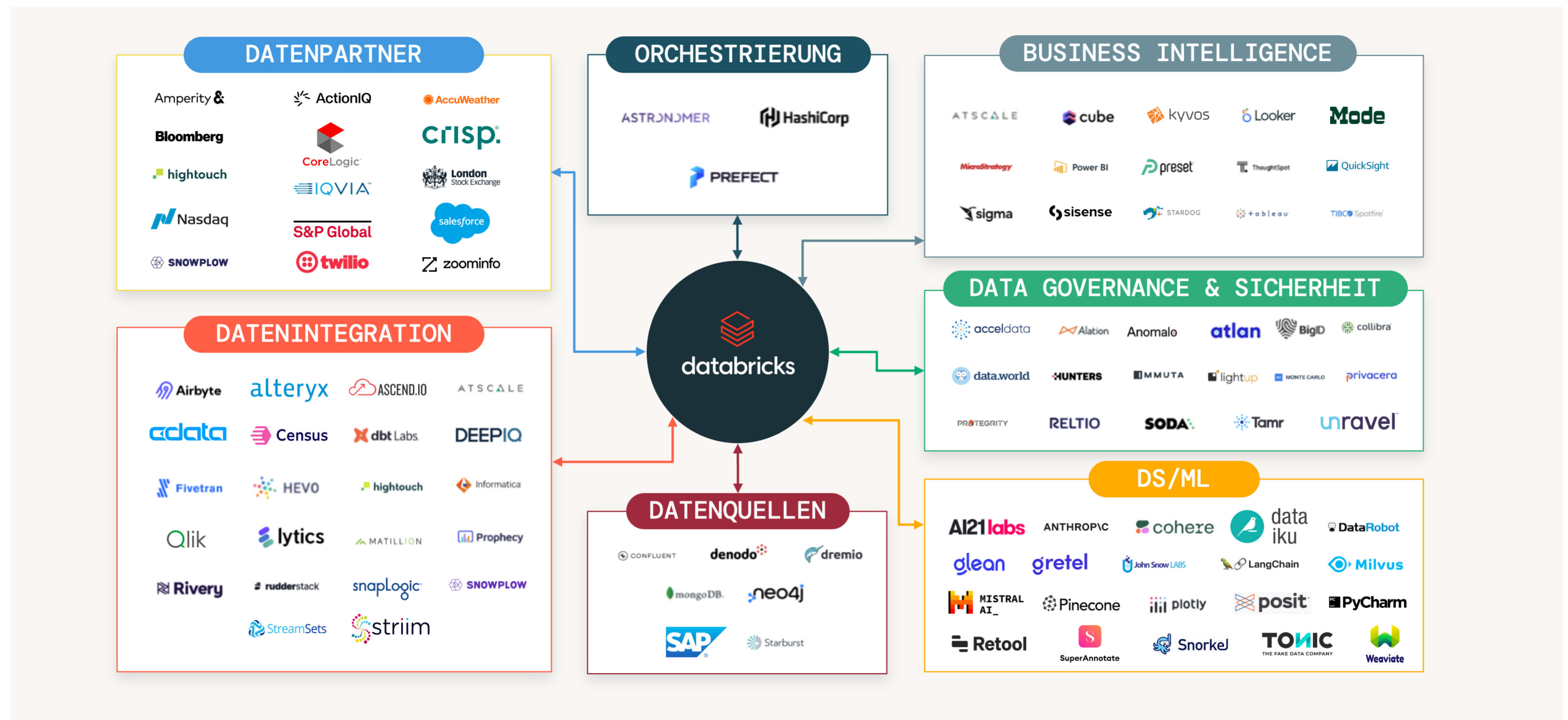
Einheitliche Datenorchestrierung mit Databricks Workflows

Databricks Workflows bietet eine einfache, aber zuverlässige Orchestrierungslösung für Daten und KI auf der Data Intelligence Platform. Mit Databricks Workflows können Sie mehrstufige Workflows definieren, um ETL-Pipelines, ML-Trainings-Workflows und mehr zu implementieren. Es bietet erweiterte Ablaufsteuerungsfunktionen und unterstützt verschiedene Task-Typen und Optionen zum Auslösen von Workflows. Als plattformnatives Orchestrierungs-Tool bietet Databricks Workflows außerdem eine fortschrittliche Beobachtungsfunktion zur Überwachung und Visualisierung der Workflow-Ausführung sowie Warnfunktionen, die bei Problemen Alarm schlagen. Databricks Workflows bietet Optionen für Serverless Compute und erlaubt so intelligentes Skalieren und eine effiziente Task-Ausführung.

Ein reichhaltiges Ökosystem an Datenlösungen

Die Data Intelligence Platform setzt auf Open-Source-Technologien auf und nutzt offene Standards. Dadurch können Sie führende Datenlösungen mit allem nutzen, was Sie auf dem Lakehouse entwickeln.

Eine große Auswahl an **Technologiepartnern** macht es einfach und bequem, Technologien zu integrieren, auf die Sie bei der Migration zu Databricks angewiesen sind. Und noch ein Vorteil: Sie sind nicht an einen abgeschotteten Datentechnologie-Stack gebunden.

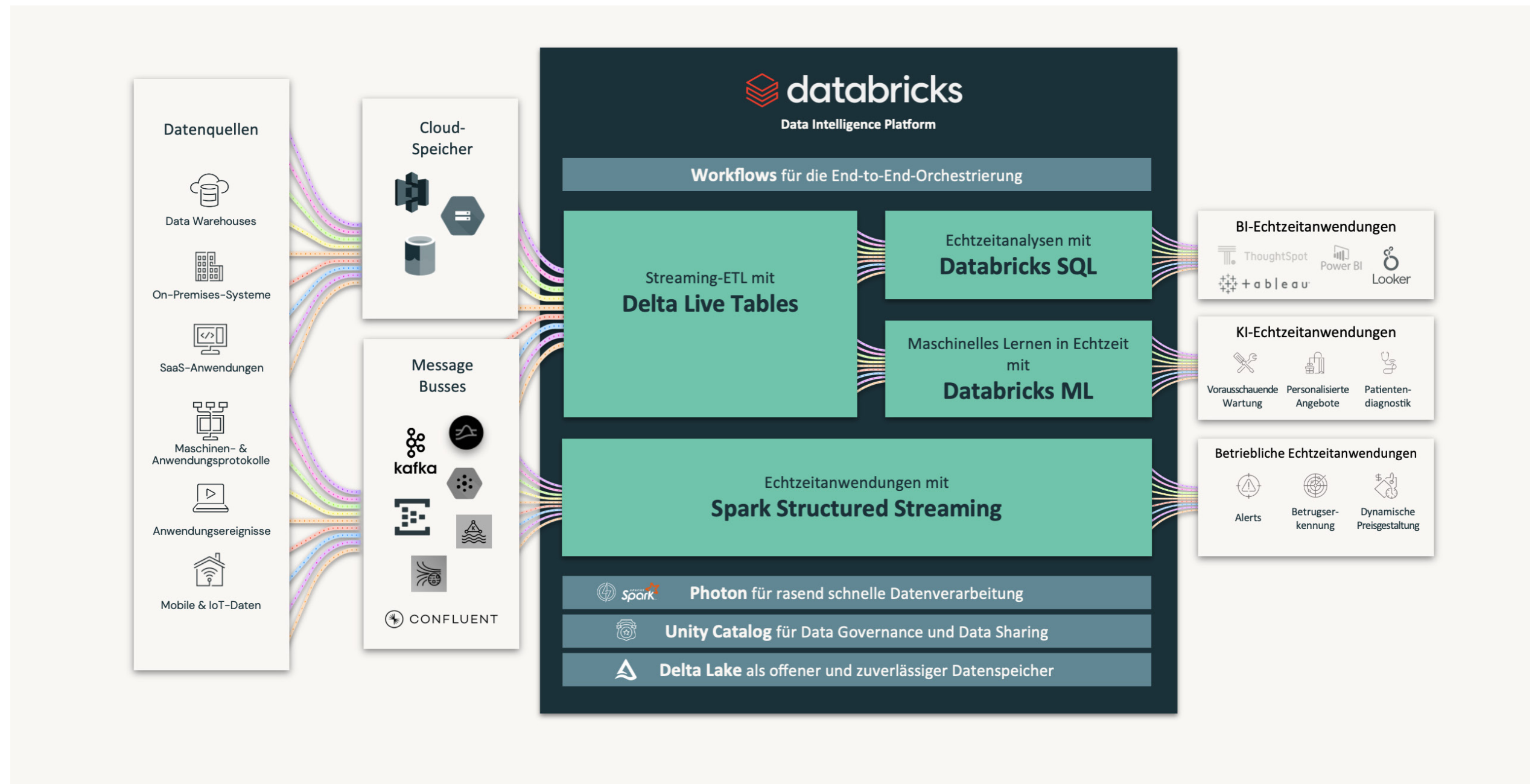


Die Data Intelligence Platform lässt sich in eine Vielzahl von Technologien integrieren.

Warum Data Engineers sich für die Data Intelligence Platform entscheiden

Inwiefern hilft die Data Intelligence Platform bei den zuvor besprochenen Herausforderungen im Data Engineering?

- **Effiziente Aufnahme, breites Spektrum an Datenkonnektoren:** Mit Databricks können Sie Daten effizient erfassen, denn es werden nur neue Daten oder Tabellenaktualisierungen importiert. Mit einer wachsenden Anzahl nativer Konnektoren für beliebte Datenquellen sowie einem breiten Netzwerk von Partnern für die Datenaufnahme können Sie Daten ganz leicht aus isolierten Systemen auf Ihre Datenplattform übertragen. Das Erfassen und Speichern Ihrer Daten in Delta Lake unter Nutzung der Zuverlässigkeits- und Skalierbarkeitseigenschaften der Data Intelligence Platform ist der erste Schritt, um aus Ihren Daten Mehrwert zu generieren und Innovation zu beschleunigen.
- **Stream-Datenverarbeitung in Echtzeit:** Die Data Intelligence Platform vereinfacht Entwicklung und Betrieb, denn sie automatisiert die Produktionsaspekte, die mit der Erstellung und Pflege von Echtzeitdaten-Workloads verbunden sind. Delta Live Tables bietet einen deklarativen Ansatz zur Definition von Streaming-ETL-Pipelines, und Spark Structured Streaming hilft bei der Erstellung von Echtzeitanwendungen für die Echtzeit-Entscheidungsfindung (siehe Diagramm auf der nächsten Seite).

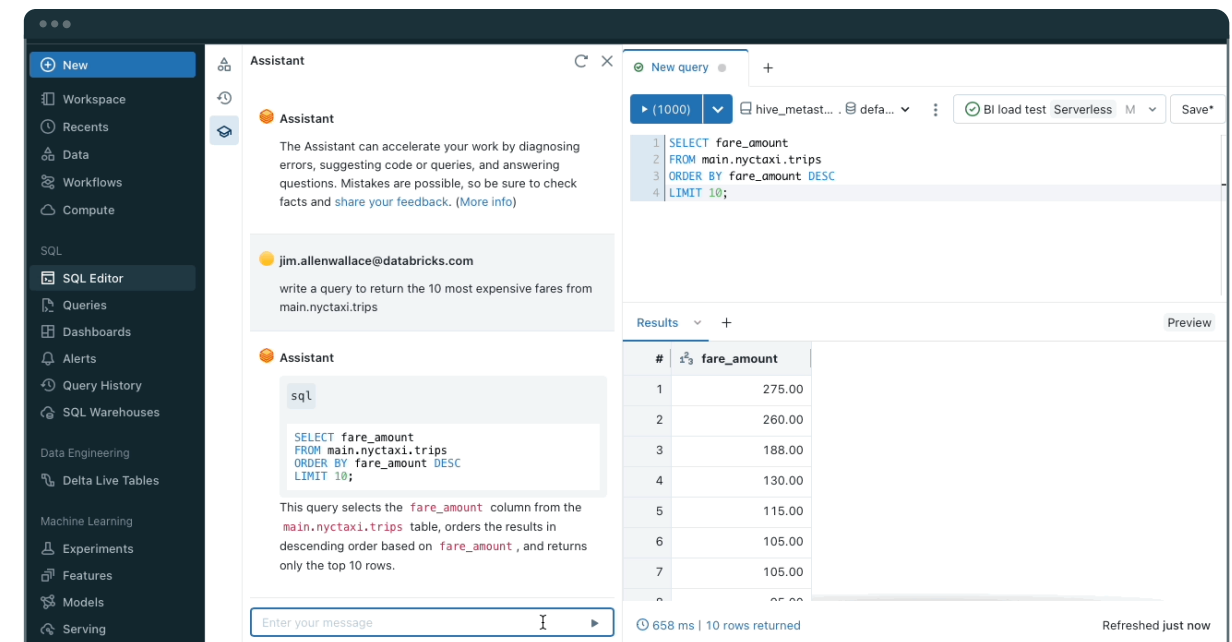


Ein einheitliches Toolset für die Echtzeit-Datenverarbeitung

- **Zuverlässige Datenpipelines im großen Stil:** Sowohl **Delta Live Tables** als auch **Databricks Workflows** nutzen intelligentes Autoscaling und ein automatisch optimiertes Ressourcenmanagement, um hochgradig skalierte Workloads zu bewältigen. Mit der Lakehouse-Architektur wird die hohe Skalierbarkeit von Data Lakes mit der hohen Zuverlässigkeit von Data Warehouses kombiniert. Möglich ist dies dank Delta Lake, dem Speicherformat, das die Grundlage des Lakehouse bildet.
- **Datenqualität:** Hohe Zuverlässigkeit – angefangen bei der Speicherebene mit **Delta Lake** bis hin zu den datenqualitätsspezifischen Funktionen von Delta Live Tables – sorgt für eine hohe Datenqualität. Hierzu gehören etwa die Festlegung von „Erwartungswerten“ für Daten, um beschädigte oder fehlende Daten abzufangen, sowie automatische Wiederholungsversuche. Darüber hinaus bieten sowohl Databricks Workflows als auch Delta Live Tables dem Data Engineer uneingeschränkte Transparenz, wodurch Probleme schneller und einfacher gelöst werden können.
- **Einheitliche Governance mit sicherem Datenaustausch:** **Unity Catalog** stellt ein einheitliches Governance-Modell für die gesamte Plattform bereit, sodass jedes Dataset und jede Pipeline einheitlich reguliert werden. Datasets sind auffindbar und können mithilfe von Delta Sharing sicher für interne wie externe Teams freigegeben werden. Außerdem stellt Unity Catalog als plattformübergreifende Governance-Lösung wertvolle Herkunftsinformationen bereit, sodass sich leicht nachvollziehen lässt, wie die einzelnen Datasets und Tabellen weiterverwendet werden und woher sie stammen.

Darüber hinaus profitieren Data Engineers, die die Data Intelligence Plattform nutzen, von den neuesten Innovationen in Form von GenAI-infundierter Intelligenz:

- **KI-gestützte Produktivität:** **Databricks Assistant** ist ein kontextsensitiver KI-Assistent, der insbesondere für Data Engineers nützlich ist. Er bietet eine dialogorientierte API, um Daten abzufragen, Code zu generieren, Codeabfragen zu erläutern und sogar Probleme zu beheben.



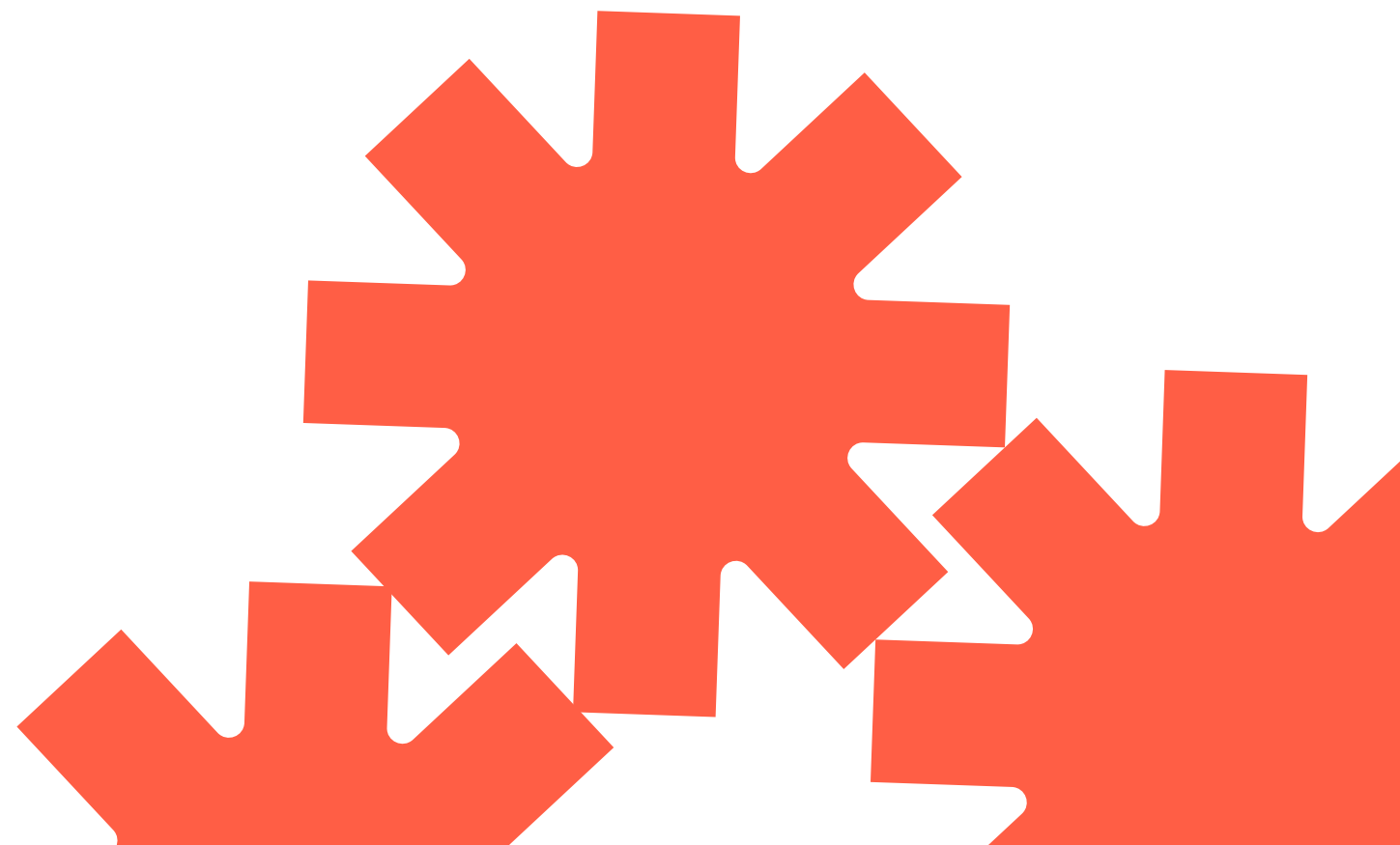
Fazit

Je stärker das Bestreben von Organisationen wird, mit KI innovativ zu werden, desto mehr rückt das Data Engineering in den Fokus, denn es ermöglicht den Aufbau zuverlässiger Echtzeit-Datenpipelines, die KI erst realisierbar machen. Mit der Databricks Platform, die auf einer Lakehouse-Architektur aufsetzt und von Data Intelligence angetrieben wird, sind Data Engineers für den erfolgreichen Umgang mit den zentralen Herausforderungen der modernen Datenlandschaft gerüstet. Dank der fortschrittlichen Funktionen der Data Intelligence Platform verbringen Data Engineers deutlich weniger Zeit mit der Verwaltung komplexer Pipelines oder der Behebung von Problemen rund um Zuverlässigkeit, Skalierbarkeit und Datenqualität. Endlich können sie sich ganz der Innovation widmen und einen Mehrwert für ihre Organisation schaffen.

BEST PRACTICES UMSETZEN

Im nächsten Kapitel werden Best Practices für Data Engineering und End-to-End-Anwendungsfälle beschrieben. Dabei stützen wir uns auf Beispiele aus der Praxis. Ob Datenaufnahme oder Echtzeitverarbeitung, Orchestrierung oder Datenverbund: Sie werden lernen, wie Sie bewährte Muster anwenden und die verschiedenen Funktionen der Data Intelligence Platform optimal nutzen.

In diesem Leitfaden finden Sie in den verschiedenen **Databricks Solution Accelerators** praktische Datasets und Codebeispiele, damit Sie direkt loslegen und erste Lösungen auf der Data Intelligence Platform entwickeln können.



02

Empfehlungen und Best Practices

Tipps und Tricks zu Databricks Assistant für Data Engineers

von Jackie Zhang, Rafi Kurlansik und Richard Tomlinson

GenAI revolutioniert die Arbeitsweise von Teams, und Databricks Assistant nutzt diese Fortschritte gezielt. Sie können mit Assistant Daten über eine dialog-basierte Oberfläche abfragen und so produktiver in Ihrem Databricks Workspace arbeiten. Betrieben wird Assistant von DatabricksIQ, der Data-Intelligence-Engine für Databricks; sie sorgt für sichere Daten und auf das Unternehmen zugeschnittene Antworten. Mit Databricks Assistant können Sie Ihre Aufgabe in natürlicher Sprache beschreiben, um komplexen Code zu generieren, zu optimieren oder zu debuggen, ohne Ihre Entwicklungsumgebung zu verlassen.

In diesem Kapitel zeigen wir, wie Sie Databricks Assistant optimal nutzen – mit dem Fokus darauf, wie Assistant Data Engineers das Leben erleichtert: nämlich durch Automatisierung von Routinearbeiten, gesteigerte Produktivität und Konzentration und schnellere Wertschöpfung. Anschließend folgen Beispiele zu verschiedenen Rollen aus der Datenpraxis. Seien Sie also gespannt auf die kommenden Beiträge, die sich vor allem (aber nicht nur) an Data Scientists und SQL-Analysten richten.

DATENAUFNAHME

Für Data Engineers ist das Laden von Daten in Delta-Lake-Tabellen häufig der erste Schritt bei der Arbeit mit Databricks. Schauen wir uns zwei Beispiele an, wie Assistant dabei unterstützt: eines mit APIs, eines mit Dateien aus dem Cloud-Speicher. Wir zeigen jeweils Prompt und Ergebnis. Wie im Blogpost „5 Tipps“ erwähnt, führt eine möglichst konkrete Formulierung des Prompts zu den besten Ergebnissen – genau daran halten wir uns bei diesen Beispielen.

Um Daten von der datausa.io-API zu erhalten und sie mit Python in eine Delta-Lake-Tabelle zu laden, haben wir den folgenden Prompt verwendet:

Hilf mir, Daten aus dieser API in eine Delta-Lake-Tabelle zu übertragen:

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

Verwende auf jeden Fall PySpark und sei präzise! Wenn die Spalten im Spark-DataFrame Leerzeichen enthalten, entferne diese aus dem Spark-DataFrame.

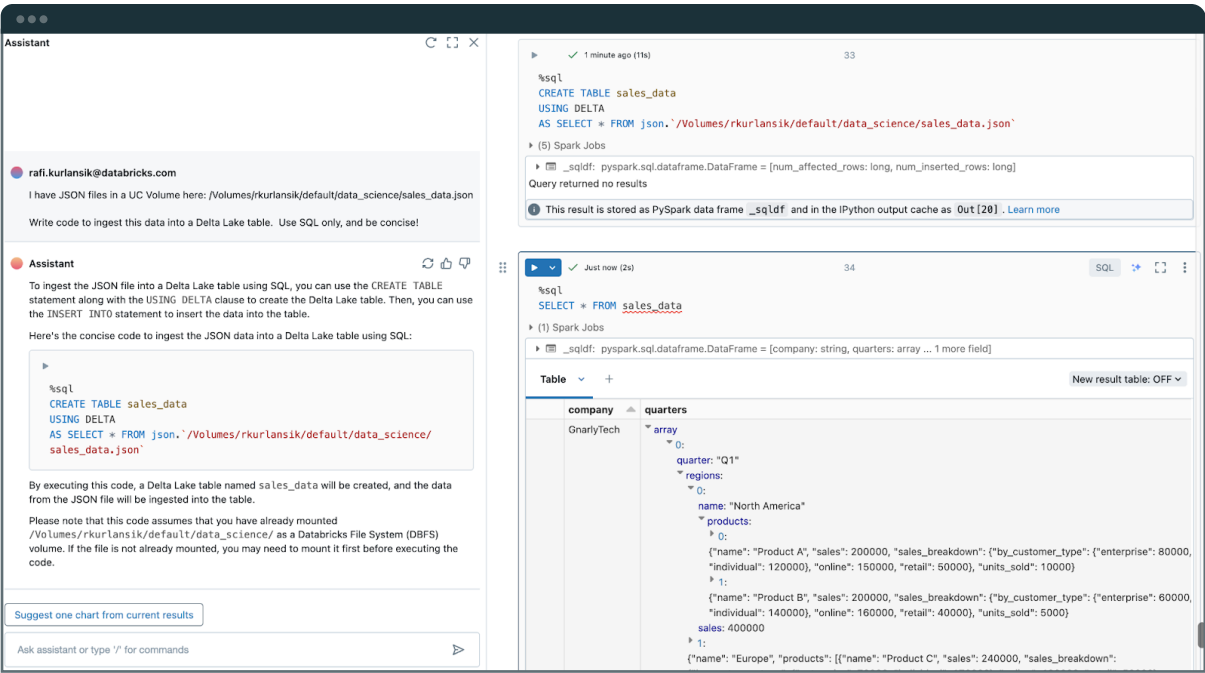
The screenshot shows the Databricks Assistant interface. On the left, a chat window displays a prompt asking for help to ingest data from the datausa.io API into a Delta Lake table. The assistant responds with a concise Python code snippet using PySpark. On the right, the code is shown in a code editor, and below it, a table preview displays the fetched data.

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

Ein ähnlicher Prompt kann verwendet werden, um JSON-Dateien aus dem Cloud-Speicher in Delta-Lake-Tabellen zu importieren, diesmal jedoch mit SQL:

Ich habe hier JSON-Dateien auf einem UC-Volume: /Volumes/rkurlansik/default/data_science/sales_data.json

Schreibe einen Code, um diese Daten in eine Delta-Lake-Tabelle aufzunehmen. Verwende nur SQL und fasse dich kurz!



TRANSFORMATION VON UNSTRUKTURIERTEN IN STRUKTURIERTE DATEN

Gemäß den Grundsätzen für bereinigte Daten sollte jede Zelle einer Tabelle eine einzelne Beobachtung mit einem korrekten Datentyp enthalten. Komplexe Strings oder geschachtelte Datenstrukturen stehen oft im Widerspruch zu diesem Prinzip. Daher besteht die Arbeit von Data Engineers oft darin, strukturierte Daten aus unstrukturierten Daten zu extrahieren. Wir wollen zwei Beispiele untersuchen, bei denen Assistant bei dieser Aufgabe besonders gut abschneidet: beim Arbeiten mit regulären Ausdrücken und bei der Auflösung geschachtelter Datenstrukturen.

Reguläre Ausdrücke

Reguläre Ausdrücke (Regex) stellen ein Mittel dar, um strukturierte Daten aus durcheinandergewürfelten Zeichenketten zu extrahieren. Allerdings ist das Erstellen eines korrekten regulären Ausdrucks zeitaufwendig und umständlich. In dieser Hinsicht ist Assistant ein Segen für alle Data Engineers, die sich mit Regex abmühen.

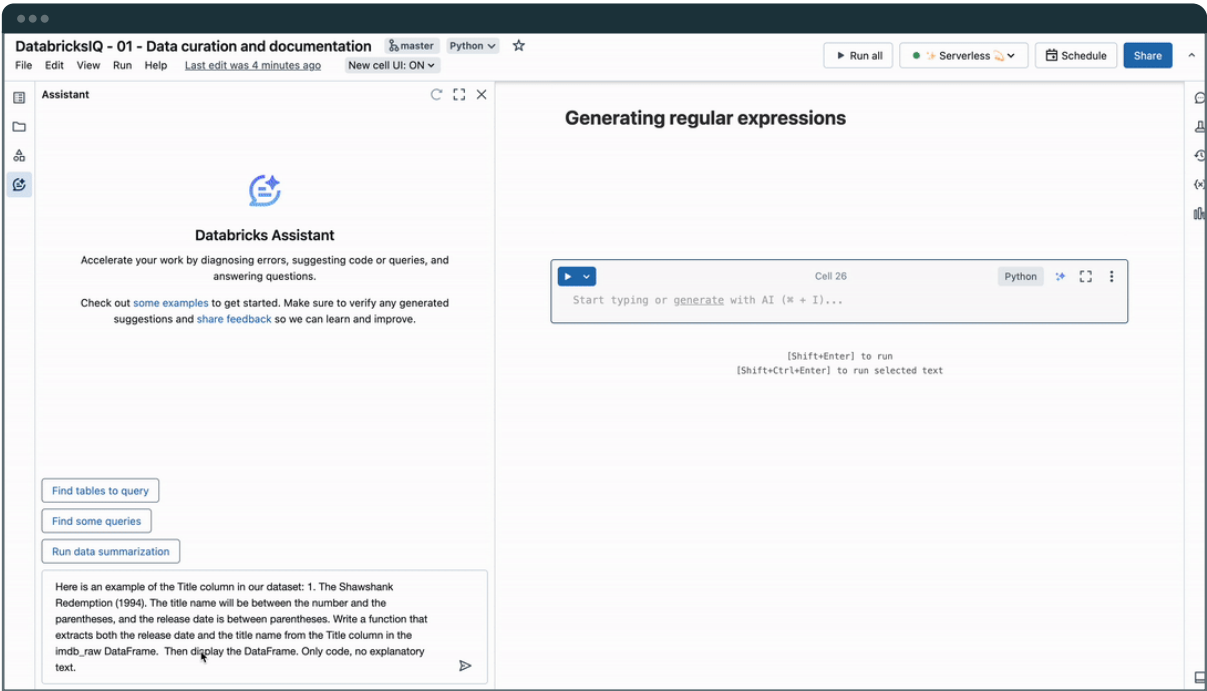


Betrachten Sie das folgende Beispiel, das die Spalte *Title* aus dem Dataset der Internet Movie Database (IMDb) verwendet:

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

Diese Spalte enthält zwei verschiedene Angaben: den Filmtitel und das Erscheinungsjahr. Mit dem folgenden Prompt ermittelt Assistant einen regulären Ausdruck, um den String in mehrere Spalten zu zerlegen.

Hier sehen Sie zunächst ein Beispiel für die Spalte „Title“ in unserem Dataset: 1. The Shawshank Redemption (1994). Der Titel steht zwischen der Zahl und den Klammern, das Veröffentlichungsjahr dagegen in den Klammern. Erstelle eine Funktion, die sowohl das Veröffentlichungsjahr als auch den Titelnamen aus der Spalte „Title“ im DataFrame „imdb_raw“ extrahiert.



Die Angabe eines Beispiel-Strings im Prompt erleichtert Assistant das Auffinden des richtigen Ergebnisses. Wenn Sie mit sensiblen Daten arbeiten, empfehlen wir Ihnen, ein fiktives Beispiel zu erstellen, das demselben Muster folgt. Jedenfalls gibt es jetzt ein Problem weniger, um das Sie sich bei Ihrer Arbeit als Data Engineer kümmern müssen.

Geschachtelte Strukturen, Arrays (JSON, XML usw.)

Bei der Datenaufnahme über API, gespeicherte JSON-Dateien oder NoSQL-Datenbanken können die resultierenden Spark-DataFrames tief verschachtelt sein. Es ist dann oft schwierig, sie korrekt zu vereinfachen. Betrachten Sie die folgenden fiktiven Umsatzdaten im JSON-Format:

```
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]

company	quarters	year
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}	2024

1 row | 51.40 seconds runtime | Refreshed 1 minute ago

Als Data Engineer sollen Sie das geschachtelte Array vereinfachen und Umsatzmetriken für das jeweilige Produkt extrahieren. Normalerweise wäre für diese Aufgabe umfangreiches Ausprobieren erforderlich – selbst dann, wenn die Daten verhältnismäßig einfach sind. Assistant berücksichtigt jedoch die DataFrame-Schemata im Arbeitsspeicher und generiert so den für die Aufgabe erforderlichen Code. Mit einem einfachen Prompt erhalten wir in Sekundenschnelle die gewünschten Ergebnisse:

Schreibe PySpark-Code, um den DataFrame zu vereinfachen und den Umsatz pro Produkt und Kunde zu extrahieren.

Nested JSON

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]

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}

1 row | 1.22 seconds runtime | Refreshed now

REFACTORING, DEBUGGING UND OPTIMIERUNG

Data Engineers müssen regelmäßig Code überarbeiten, der von anderen Teammitgliedern stammt – etwa weniger erfahrenen Entwicklern oder Mitarbeitenden, die das Unternehmen verlassen haben. In solchen Fällen kann Assistant schlecht geschriebenen Code erläutern, da er Kontext und Absicht versteht. Er schlägt effizientere Algorithmen vor, verbessert die Lesbarkeit oder ergänzt Kommentare.

Optimieren von Dokumentation und Wartungsfreundlichkeit

Der folgende Python-Code berechnet die Gesamtkosten der Artikel in einem Online-Warenkorb.

```

1  def calculate_total(cart_items):
2      total = 0
3      for i in range(len(cart_items)):
4          if cart_items[i]['type'] == 'book':
5              discount = cart_items[i]['price'] * 0.05
6              total += cart_items[i]['price'] - discount
7          else:
8              total += cart_items[i]['price']
9      return total

10 cart_items = [{'name': 'Python Programming', 'type': 'book', 'price': 50},
11               {'name': 'Laptop', 'type': 'electronics', 'price': 800}]
12 total_price = calculate_total(cart_items)

```

Die Verwendung von Bedingungsblöcken in diesem Code erschwert die Lesbarkeit und ist bei steigendem Projektumfang zunehmend ineffizient. Außerdem fehlen Kommentare, die den Code erläutern. Zum Einstieg bietet es sich an, Assistant um eine schrittweise Erläuterung des Codes zu bitten. Wenn der Data Engineer den Code erst einmal verstanden hat, kann Assistant ihn transformieren und mit dem folgenden Prompt effizienter und lesbarer machen:

Schreibe diesen Code so um, dass er effizienter, ordentlich kommentiert und gemäß den Standards für die Dokumentation von Python-Funktionen dokumentiert ist.

Im folgenden Beispiel wird der Code klar dokumentiert; statt bedingter Blöcke kommen Generatorausdrücke zum Einsatz, was die Speicherauslastung bei größeren Datasets verbessert.

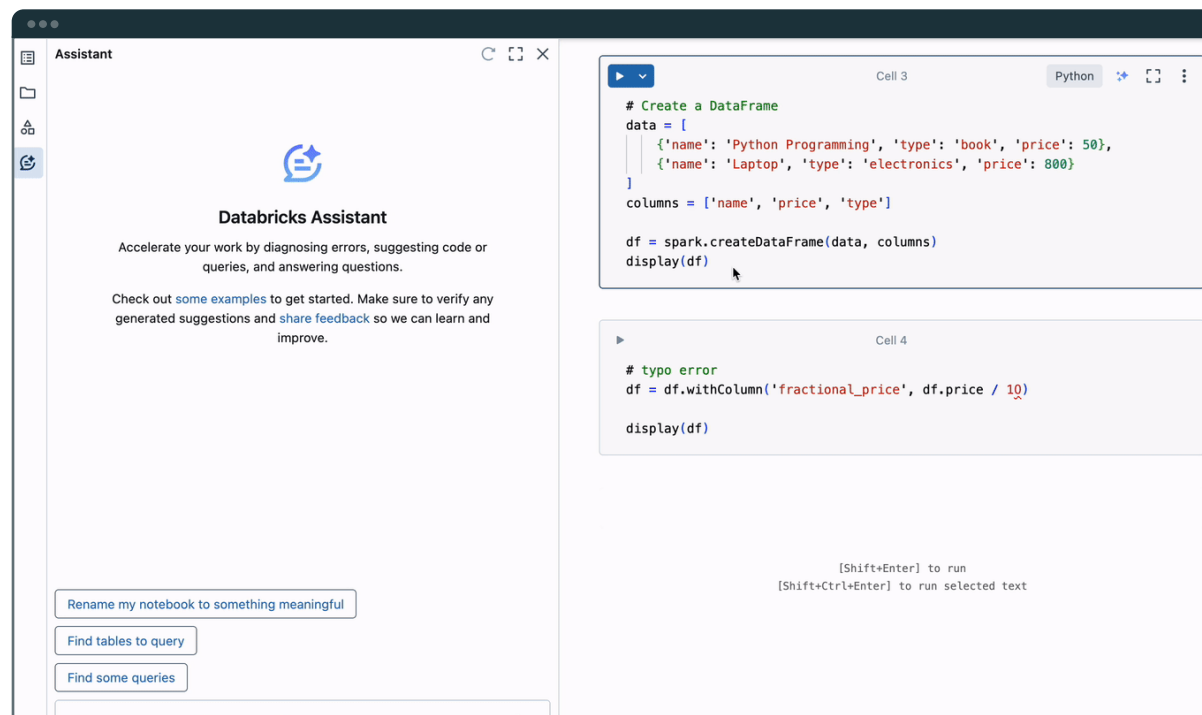
```

1  def calculate_total(cart_items):
2      """
3      Calculate the total price of items in the shopping cart.
4
5      Apply a 5% discount to book items. The total is calculated by summing
6      the prices, taking into account the discount for books.
7
8      Parameters:
9      cart_items (list of dict): A list of dictionaries, where each dictionary
10                                 represents an item in the cart with keys 'name',
11                                 'type', and 'price'.
12
13      Returns:
14      float: The total price of all items in the cart after discounts.
15      """
16      # Use a generator expression to iterate over cart items, applying a 5% discount
17      # to books.
18      return sum(item['price'] * 0.95 if item['type'] == 'book' else item['price'] for
19                item in cart_items)
20
21 # Example usage
22 cart_items = [{'name': 'Python Programming', 'type': 'book', 'price': 50},
23               {'name': 'Laptop', 'type': 'electronics', 'price': 800}]
24 total_price = calculate_total(cart_items)

```

Fehler diagnostizieren

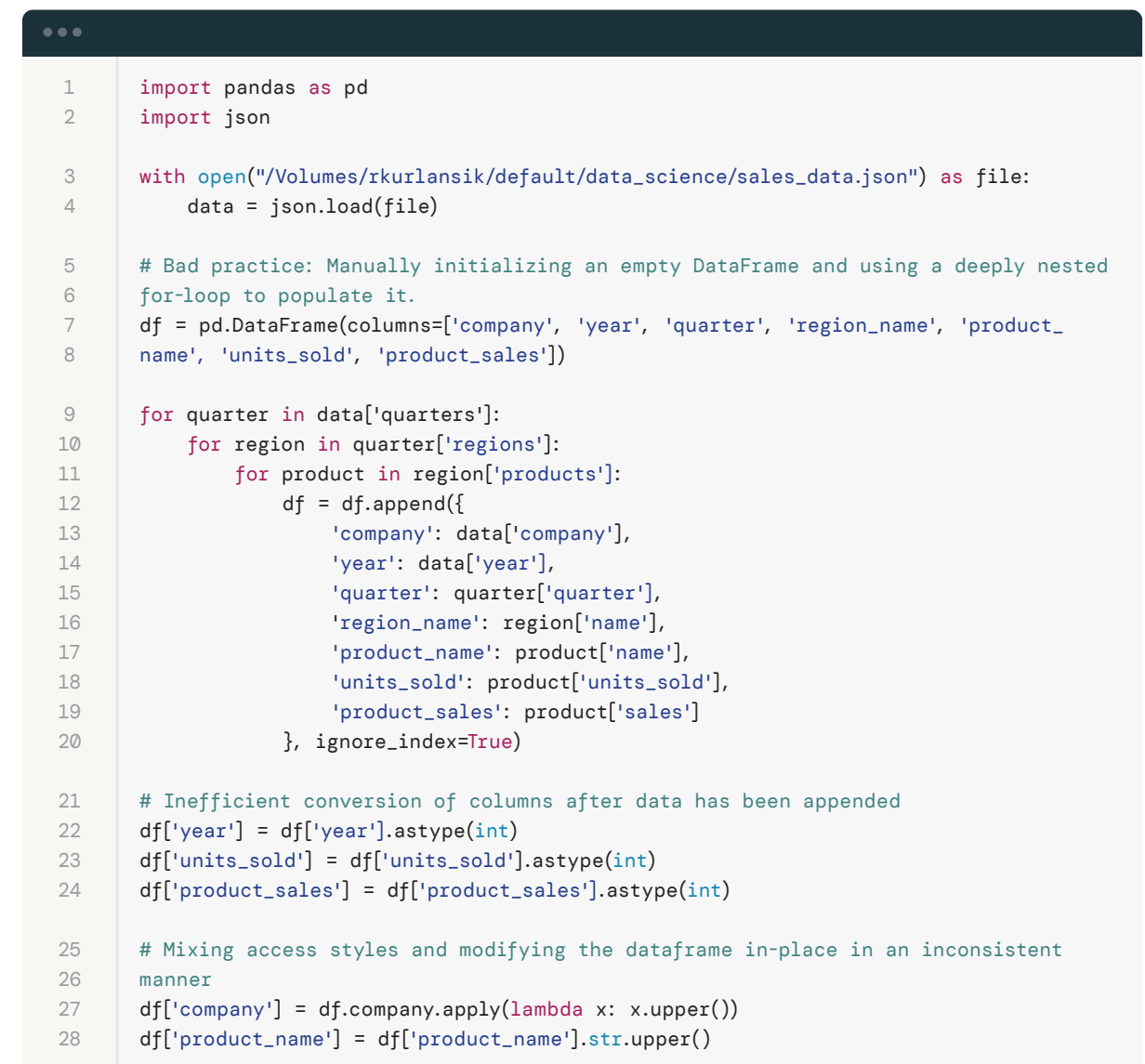
Früher oder später kommen Data Engineers nicht mehr um das Debugging herum. Mit Assistant ist es nicht mehr nötig, mehrere Browser-Tabs zu öffnen oder zwischen Kontexten zu wechseln, um die Ursache von Fehlern im Code zu ermitteln. So kann man sich ganz auf die eigentliche Arbeit konzentrieren, was die Produktivität enorm steigert. Um zu verstehen, wie das bei Assistant funktioniert, erstellen wir einen einfachen PySpark-DataFrame und lösen einen Fehler aus.



Im obigen Beispiel wird beim Hinzufügen einer neuen Spalte zum DataFrame ein Tippfehler eingefügt. Die Null in „10“ ist eigentlich der Buchstabe „O“, was zu einem Syntaxfehler *invalid decimal literal* führt. Assistant bietet sofort an, den Fehler zu diagnostizieren. Er erkennt den Tippfehler korrekt und schlägt einen berichtigten Code vor, der im Editor in die aktuelle Zelle eingefügt werden kann. Durch die Diagnose und Korrektur von Fehlern auf diese Weise können Sie sich stundenlanges Debugging ersparen.

Transpilieren von pandas in PySpark

pandas ist eine der beliebtesten Python-Bibliotheken für die Datenbearbeitung und wird von Data Scientists auf der ganzen Welt verwendet. Bleiben wir bei unseren JSON-Umsatzdaten und stellen wir uns eine Situation vor, in der ein unerfahrener Data Scientist sein Bestes gegeben hat, um die Daten mit pandas zu vereinfachen. Das sieht nicht schön aus und entspricht auch nicht den Best Practices, aber das Ergebnis stimmt:



Normalerweise kann pandas nur auf einem einzelnen Computer ausgeführt werden. Bevor dieser Code in die Produktion überführt und zur Bearbeitung von Milliarden von Datenzeilen eingesetzt wird, sollte der Data Engineer ihn in PySpark konvertieren. Hierzu muss der Data Engineer den Code verstehen und ihn so umformulieren, dass er wartungsfreundlich, prüfbar und leistungsfähig ist. Auch hierfür hat Assistant in Sekundenschnelle wieder eine bessere Lösung parat.

Assistant

rafi.kurlansik@databricks.com

Rewrite this inefficient pandas code in PySpark

Assistant

Certainly! Here's the rewrite of the inefficient pandas code using PySpark:

```
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()
```

Beachten Sie, dass der generierte Code die Erstellung einer *SparkSession* beinhaltet, was in Databricks nicht erforderlich ist. Wie jedes LLM kann auch Assistant sich mitunter irren oder halluzinieren. Als Data Engineer sind Sie der eigentliche Autor Ihres Codes, und es ist wichtig, dass Sie jeden generierten Code überprüfen und nachvollziehen, bevor Sie mit der nächsten Aufgabe fortfahren. Wenn Sie ein solches Verhalten bemerken, passen Sie Ihren Prompt entsprechend an.

TESTS VERFASSEN

Ein zentraler Schritt im Data Engineering ist das Schreiben von Tests, die die korrekte Transformationslogik Ihres DataFrames sicherstellen und fehlerhafte Daten in der Pipeline frühzeitig erkennen. Bleiben wir bei unserem Beispiel mit den JSON-Umsatzdaten: Mit Assistant lässt sich ganz einfach testen, ob eine der Umsatzspalten negativ ist. Solange die Werte in den Umsatzspalten nicht kleiner als null sind, können wir davon ausgehen, dass unsere Daten und Transformationen in diesem Fall korrekt sind.

Nested JSON Python

File Edit View Run Help Last edit was 17 minutes ago New cell UI: ON

Assistant

```
"explode(quarters) as 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")

# Show the resulting DataFrame
display(exploded_df)
```

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

Find tables to query

Find some queries

Ask assistant or type '/' for commands

3	GnarlyTech	2024	Q1	Europe	Product C	190000	50000
4	GnarlyTech	2024	Q1	Europe	Product D	40000	20000
5	GnarlyTech	2024	Q1	Asia	Product E	220000	80000
6	GnarlyTech	2024	Q2	North America	Product A	180000	60000

10 rows | 0.82 seconds runtime Refreshed 17 minutes ago

Cell 25 Python

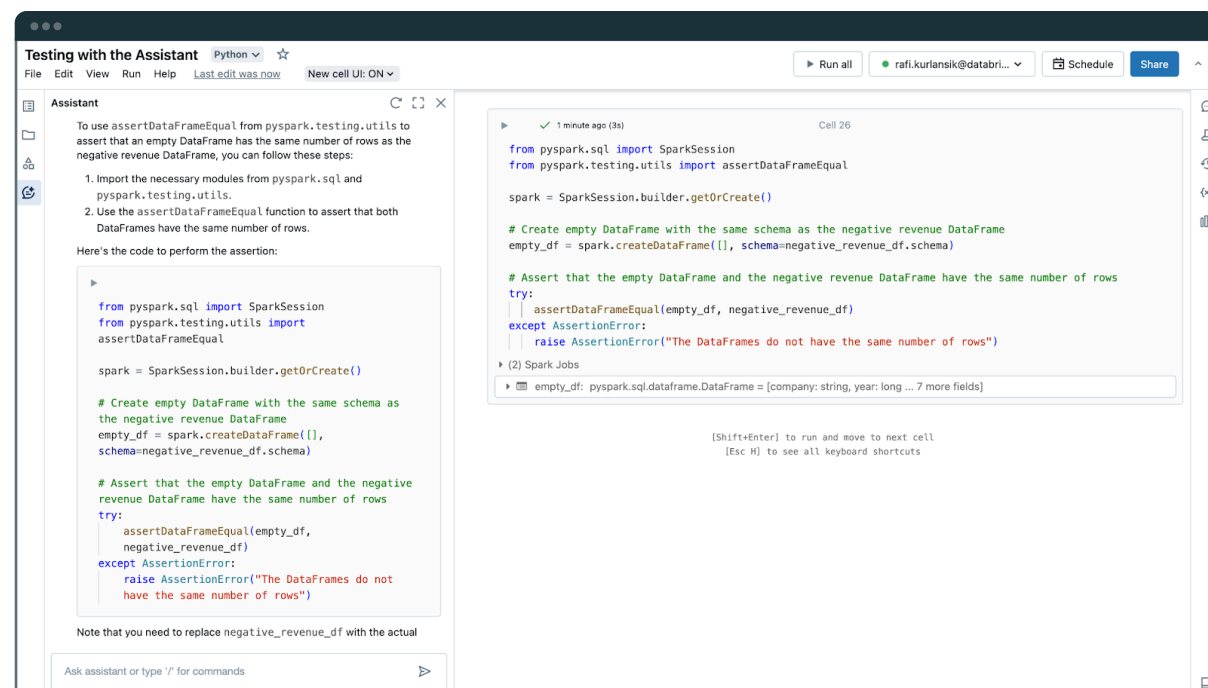
Start typing or generate with AI (* + I)...

[Shift+Enter] to run and move to next cell
[Esc H] to see all keyboard shortcuts

Entsprechend dieser Logik können wir Assistant mit dem folgenden Prompt auffordern, den Test in die native PySpark-Testfunktionalität zu integrieren:

Entwirf einen Test mit `assertDataFrameEqual` aus `pyspark.testing.utils`, um zu überprüfen, ob ein leerer DataFrame die gleiche Anzahl an Zeilen hat wie unser negativer Umsatz-DataFrame.

Der Assistant liefert daraufhin funktionsfähigen Code, mit dem Sie schnell in die Testentwicklung einsteigen können.

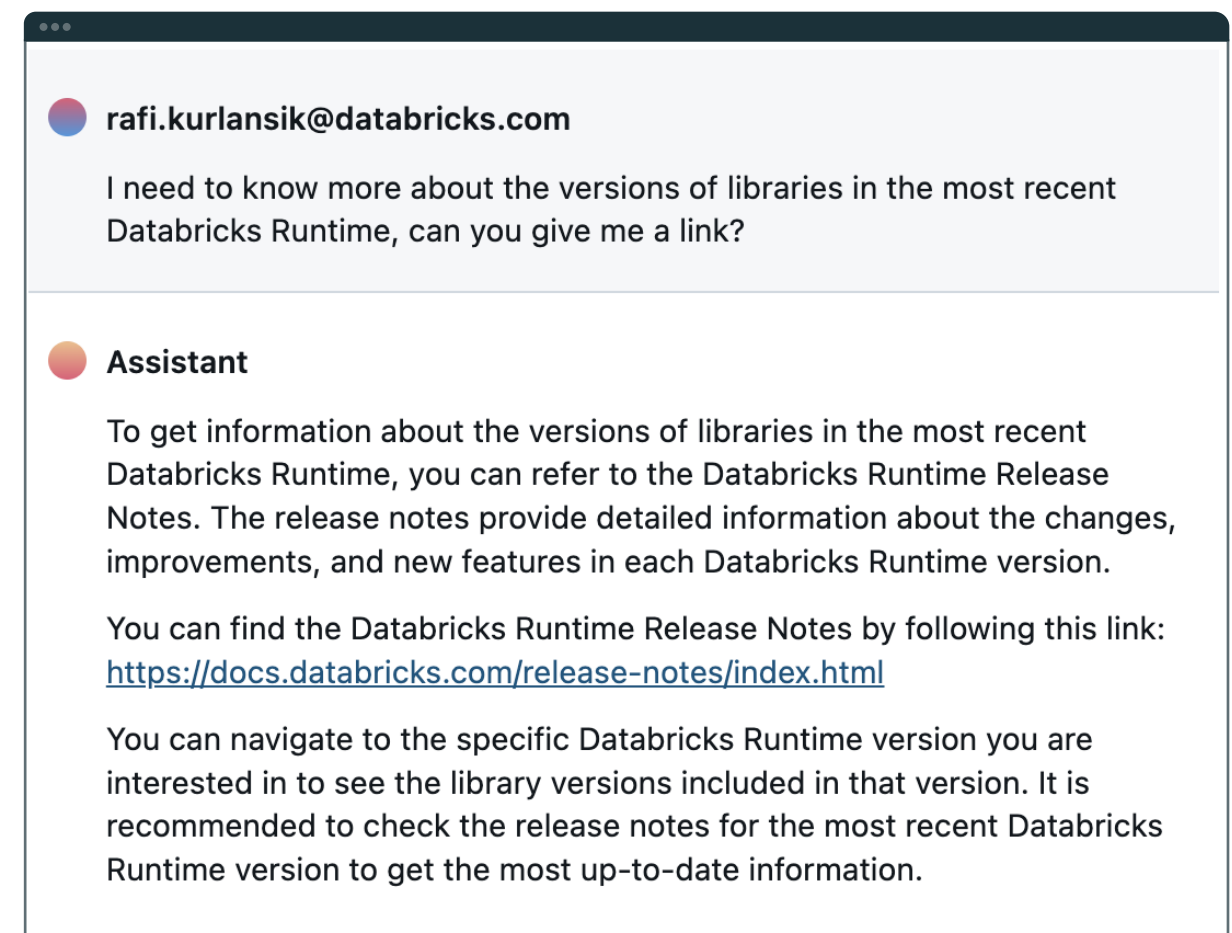


Dieses Beispiel verdeutlicht, dass ein konkret formulierter und detaillierter Prompt zu besseren Ergebnissen führt. Wenn wir den Assistant ohne weitere Vorgaben einfach darum bitten, Tests zu schreiben, ohne weitere Details zu liefern, wird die Qualität der Ergebnisse deutlich stärker schwanken. Wenn wir hingegen konkret und eindeutig angeben, was wir brauchen – einen Test, der PySpark-Module verwendet, die auf der von ihm verfassten Logik aufbauen –, erzielen wir in der Regel bessere Ergebnisse, als wenn wir erwarten, dass Assistant unsere Absichten korrekt errät.

HILFE ERHALTEN

Assistant ist aber nicht nur grundsätzlich in der Lage, Code zu verbessern und verstehen, sondern verfügt über Kenntnis der gesamten Databricks-Dokumentation und -Wissensdatenbank. Diese Informationen werden regelmäßig indiziert und Assistant über eine RAG-Architektur als zusätzlicher Kontext zur Verfügung gestellt. So können Benutzer nach Produktfunktionen und -konfigurationen suchen, ohne die Databricks Platform zu verlassen.

Wenn Sie beispielsweise Einzelheiten zur Systemumgebung der von Ihnen verwendeten Version von Databricks Runtime benötigen, kann Assistant Sie zur entsprechenden Seite in der Databricks-Dokumentation weiterleiten.



Assistant kann einfache, beschreibende und umgangssprachlich formulierte Fragen beantworten und verbessert so das Nutzererlebnis beim Arbeiten mit Databricks-Funktionen und beim Lösen von Problemen. Er kann Benutzern sogar beim Einreichen von Supporttickets helfen! Weitere Einzelheiten finden Sie im Ankündigungsartikel.

FAZIT

Die Eintrittsbarriere für hochwertiges Data Engineering wurde dank der Leistungsfähigkeit der generativen KI von Databricks Assistant gesenkt. Egal, ob Sie Einsteiger sind und Hilfe beim Umgang mit komplexen Datenstrukturen benötigen oder ein erfahrener Fachmann, der sich reguläre Ausdrücke erstellen lassen möchte: Assistant wird Ihr Leben einfacher machen. Seine Kernkompetenz, Code zu verstehen, generieren und dokumentieren, steigert die Produktivität von Data Engineers unabhängig vom Qualifikationsniveau. Wenn Sie mehr erfahren möchten, lesen Sie die [Databricks-Dokumentation](#), um direkt mit Databricks Assistant durchzustarten.

03

Ausgearbeitete Notebooks und Datensätze

Dieser Abschnitt enthält mehrere Solution Accelerators: kostenlose und gebrauchsfertige Beispiele für Datenlösungen aus verschiedenen Branchen vom Handel über die Fertigung bis hin zum Gesundheitswesen. Alle nachstehenden Szenarien enthalten Notebooks mit Code und Schritt-für-Schritt-Anleitungen, die Ihnen den Einstieg erleichtern. Sammeln Sie praktische Erfahrung mit der Databricks Data Intelligence Platform, indem Sie Folgendes einfach selbst ausprobieren:



Gesamteffektivität des Geräteparks

Sensordaten von Geräten erfassen – für aussagekräftige Kennzahlen und fundierte, datengestützte Entscheidungen

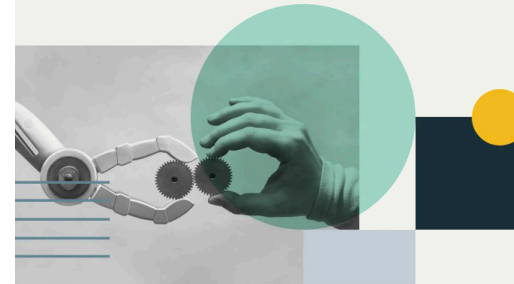
 [Lösung erkunden](#) 



Point-of-Sale-Analytics in Echtzeit

Aktuelle Lagerbestände für verschiedene Produkte in vielen Filialen mit Delta Live Tables berechnen

 [Lösung erkunden](#) 



Digitale Zwillinge

Mit digitalen Zwillingen (virtuellen Darstellungen von Geräten und Objekten) Abläufe optimieren und Erkenntnisse gewinnen

 [Lösung erkunden](#) 



Empfehlungs-Engines für die Personalisierung

Mit personalisierten Empfehlungen Kundenerlebnisse verbessern und die Konversionsrate steigern

 [Lösung erkunden](#) 



Preistransparenzdaten richtig verstehen

Gesundheitsdaten effizient erfassen, Kosten besser zu verstehen und Transparenz zu schaffen

 [Lösung erkunden](#) 

Weitere Solution Accelerators mit gebrauchsfertigen Notebooks finden Sie hier:

[Databricks Solution Accelerators](#) 

04 | Fallstudien

COX AUTOMOTIVE



BRANCHE

Automobilsektor

LÖSUNG

ESG auf Datenbasis, Auflösung von Kundenentitäten, Bedarfsplanung, Produktabgleich

PLATFORM

Workflows, Unity Catalog, Delta Sharing, ETL

CLOUD

Azure

Wie Cox Automotive das Kaufen, Verkaufen und Nutzen von Autos revolutioniert

„Wir nutzen Databricks Workflows als Standardtool für die Orchestrierung, um unsere ETL-Prozesse zu optimieren und rund 300 Jobs zu automatisieren, von denen ca. 120 regelmäßig ausgeführt werden sollen.“

– Robert Hamlet, Lead Data Engineer, Enterprise Data Services, Cox Automotive

Cox Automotive Europe gehört zu Cox Automotive, dem weltweit größten Dienstleistungsunternehmen der Automobilbranche. Das Unternehmen verfolgt das Ziel, das Kaufen, Verkaufen, Besitzen und Nutzen von Fahrzeugen neu zu denken. In enger Zusammenarbeit mit Automobilherstellern, Flottenbetreibern und Händlern unterstützt Cox Automotive Europe dabei, Leistung und Rentabilität über den gesamten Lebenszyklus hinweg zu steigern. Das Geschäft ist auf die Kernbedürfnisse der Kunden in den Bereichen Fahrzeuglösungen, Remarketing, Finanzierung, Handel und Mobilität zugeschnitten. Zu den Marken des Unternehmens in Europa gehören Manheim, Dealer Auction, NextGear Capital, Modix und Codeweavers.

Das Enterprise Data Services Team von Cox hat kürzlich eine Plattform eingeführt, die unternehmensweite Daten konsolidiert und es den Data Scientists ermöglicht, neue datengesteuerte Produkte und Services schneller und effizienter zu entwickeln. Um dem kleinen Engineering-Team eine einheitliche Plattform für Daten und Analysen mit Orchestrierung und Governance zu bieten, setzt das Enterprise Data Services Team auf die Databricks Data Intelligence Platform, Workflows, Unity Catalog und Delta Sharing.

OPTIMIERTE WERTSCHÖPFUNG DURCH EINFACHE ORCHESTRIERUNG UND MEHR TRANSPARENZ

Das Enterprise Data Services Team von Cox Automotive betreibt eine Datenplattform, die in erster Linie interne Kunden aus verschiedenen Geschäftsbereichen bedient, aber auch Dritten einige Datenfeeds bereitstellt. Dabei erfasst das Enterprise Data Services Team Daten aus verschiedenen internen Quellen und Geschäftsbereichen. „Wir nutzen Databricks Workflows als Standardtool für die Orchestrierung, um unsere ETL-Prozesse zu optimieren und rund 300 Jobs zu automatisieren – etwa 120 davon laufen regelmäßig“, sagt Lead Data Engineer Robert Hamlet vom Bereich Enterprise Data Services bei Cox Automotive.

Jobs können wöchentlich, täglich oder stündlich ausgeführt werden. Der Umfang der Daten, die derzeit in Produktionspipelines verarbeitet werden, beträgt etwa 720 GB pro Tag. Geplante Jobs rufen Daten aus verschiedenen Bereichen sowohl innerhalb als auch außerhalb des Unternehmens ab. Hamlet verwendet Databricks Workflows, um Daten an das Data-Science-Team, über Tableau an das interne Data-Reporting-Team oder direkt an Power BI zu übermitteln. „Databricks Workflows hat eine tolle Oberfläche, mit der man Workflows aller Art schnell planen kann, egal ob Notebook oder JAR“, erläutert Hamlet. „Besonders nützlich ist die Parametrisierung. Sie gibt uns Hinweise darauf, wie wir Jobs umgebungsübergreifend verlagern können. Workflows hat alle Funktionen, die man von einem Orchestrator erwarten würde.“

Hamlet schätzt auch die Fähigkeit von Workflows, Einblick in jeden Workflow-Lauf und Ausfallbenachrichtigungen zu liefern. So lassen sich Probleme schnell erkennen und beheben, bevor das Data-Science-Team betroffen ist. „Wir verwenden die Funktion für Jobbenachrichtigungen, um Ausfallmeldungen an einen Webhook zu senden, der mit unserem Microsoft Teams-Konto verknüpft ist“, sagt er. „Wenn wir dort einen Alert erhalten, überprüfen wir in Databricks, was passiert ist. Sehr nützlich ist es, die Ausführungsprotokolle einsehen und feststellen zu können, welche Fehler aufgetreten sind. Und die Funktion zum Reparieren von Läufen ist ausgesprochen praktisch, um kleine Schönheitsfehler aus einem ansonsten makellosen Verlauf zu entfernen.“

BESSERER TEAMÜBERGREIFENDER DATENZUGRIFF MIT UNITY CATALOG UND DELTA SHARING

Das Team von Hamlet setzt seit Kurzem Unity Catalog zur Verwaltung des Datenzugriffs ein und hat damit seine bisherige Methode verbessert, die nicht detailliert genug und kompliziert zu verwalten war. „Mit unserem neuen Workspace versuchen wir, mehr DevOps-Prinzipien, Infrastructure-as-Code und Gruppen zu nutzen, wo immer es möglich ist“, fährt er fort. „Ich möchte den Zugriff auf eine Vielzahl von Daten für verschiedene Gruppen und Einheiten bequem und so einfach wie möglich für mein Team verwalten können. Unity Catalog ist die ideale Lösung dafür.“

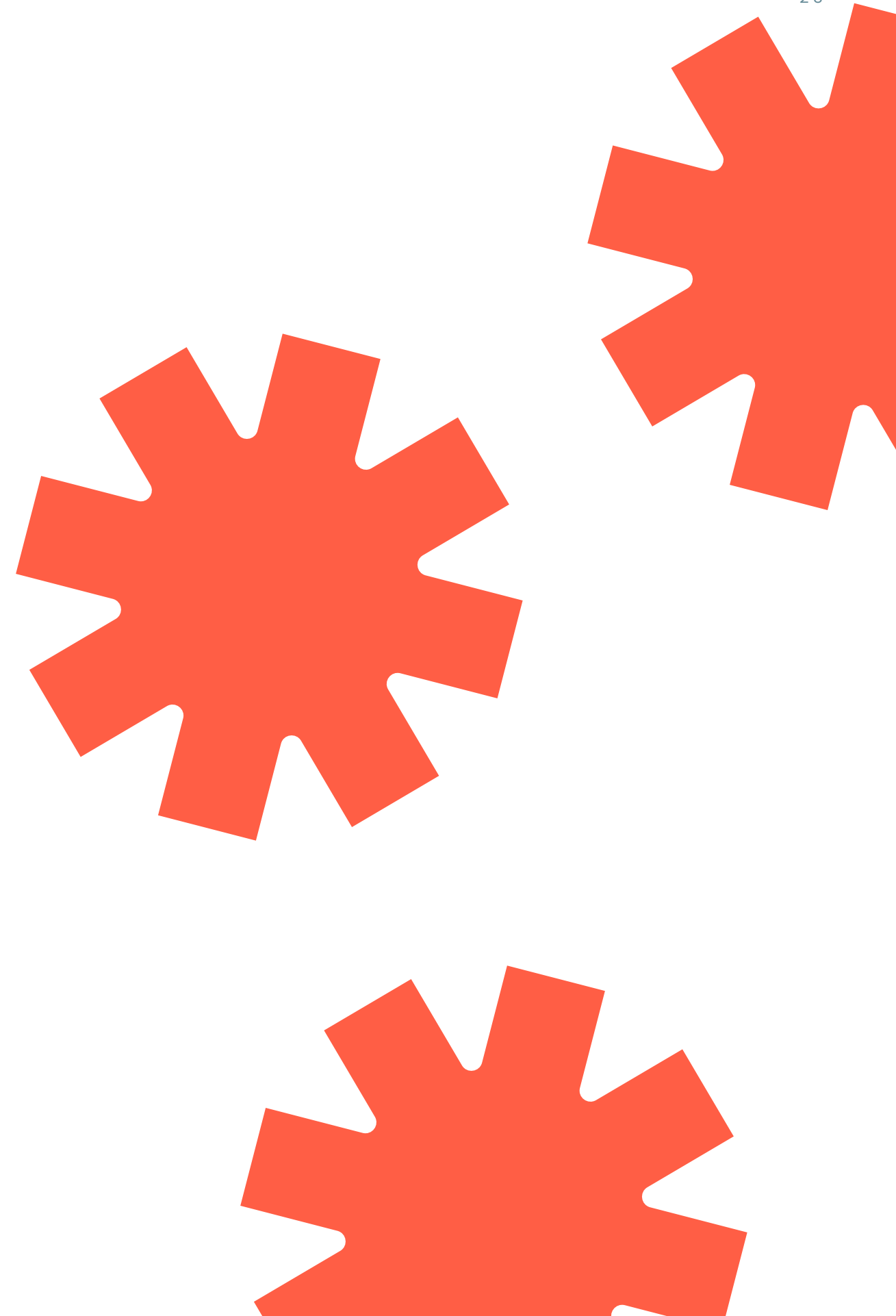
Das Enterprise Data Services Team verwendet außerdem Delta Sharing, das sich nativ in Unity Catalog integriert und es Cox erlaubt, freigegebene Daten außerhalb des Teams zentral zu verwalten und auditieren, während gleichzeitig ein hohes Maß an Sicherheit und Governance gewährleistet wird. Mit Delta Sharing können wir Daten einfach und sicher mit Unternehmensbereichen und Tochtergesellschaften teilen – ganz ohne Kopieren oder Replizieren“, sagt Hamlet. „Dadurch können wir Daten freigeben, ohne dass der Empfänger eine Identität in unserem Workspace haben muss.“

BLICK NACH VORN: EINBINDEN WEITERER FUNKTIONEN DER DATA INTELLIGENCE PLATFORM

Für die Zukunft plant Hamlet die Implementierung von Delta Live Tables (DLT), um das Erstellen und Verwalten von Batch- und Streaming-Datenpipelines zu vereinfachen, die Daten auf der Databricks Data Intelligence Plattform bereitstellen. DLT soll Data-Engineering-Teams dabei helfen, die ETL-Entwicklung und -Verwaltung zu vereinfachen. Schlussendlich kann sich Hamlet auch vorstellen, Delta Sharing zu nutzen, um Daten bequem und sicher für externe Lieferanten und Partner freizugeben und gleichzeitig alle Anforderungen an Sicherheit und Compliance zu erfüllen. „DLT verschafft uns die Möglichkeit, die Arbeit unseres Teams einfacher zu gestalten. Die Planung von Delta Live Tables wiederum wird ein weiterer Bereich sein, in dem wir Workflows einsetzen werden“, sagt er.

Hamlet freut sich auch darauf, die Funktionen zur Datenherkunft in Unity Catalog zu nutzen, um seinem Team einen klaren Überblick über die Datenflüsse im Lakehouse zu geben. So lassen sich sowohl Compliance-Anforderungen erfüllen als auch Auswirkungen von Datenänderungen analysieren. „Diese Funktion finde ich extrem spannend“, schwärmt Hamlet. „Ich hoffe, dass wir irgendwann an einem Punkt angelangt sind, an dem sich unsere Daten samt und sonders im Lakehouse befinden und wir die enge Integration mit Elementen wie Datenherkunft und erweitertem Berechtigungsmanagement besser nutzen können.“

Weitere Informationen zu Best Practices sowie Kundenberichte finden Sie in unserem [Big Book of Data Engineering](#) (nur in englischer Sprache).



Millionen Produktions-Workloads werden täglich auf Databricks ausgeführt

Nehmen Sie Batch- und Streaming-Daten ganz einfach in die **Databricks Data Intelligence Platform** auf und transformieren Sie sie. Orchestrieren Sie zuverlässige Produktionsabläufe, während Databricks Ihre Infrastruktur automatisch in großem Maßstab verwaltet. Steigern Sie die Produktivität Ihrer Teams mit integrierten Datenqualitätstests und Support für Best Practices für die Softwareentwicklung.

**Databricks
kostenlos testen**

**Einsteigen mit einer
kostenlosen Testversion**

Infos zu Databricks

Databricks ist das Unternehmen für Daten und KI. Mehr als 10.000 Unternehmen weltweit – darunter Block, Comcast, Condé Nast, Rivian, Shell und mehr als 60 Prozent der Fortune 500 – setzen auf die Databricks Data Intelligence Platform, um ihre Daten zu steuern und sie mithilfe von KI zu verwerten. Databricks wurde von den Erfindern von Lakehouse, Apache Spark™, Delta Lake und MLflow gegründet und hat seinen Hauptsitz in San Francisco mit Niederlassungen auf der ganzen Welt. Wenn Sie mehr erfahren möchten, folgen Sie Databricks auf [LinkedIn](#), [X](#) und [Facebook](#).

