

# **SNOWPARK: ENTWICKLUNG BESSERER DATEN-PIPELINES UND -MODELLE IN DER DATA CLOUD**



CHAMPION  
GUIDE

E-BOOK

# INHALTS- VERZEICHNIS

- 3** Kurzzusammenfassung
- 5** Entwicklung besserer Pipelines und Datenmodelle mit Snowpark
- 7** Snowpark-Anwendungsfälle für Data Engineering
- 9** Snowpark-Anwendungsfälle für Data Science
- 12** Anwendungsfälle von anderen Snowflake-Partnern
- 13** Ihr Einstieg in Snowpark
- 15** Über Snowflake

# KURZZUSAMMENFASSUNG

Snowflake begann die Entwicklung der Data Cloud durch die völlige Neugestaltung der Datenwelt und das Überdenken der Frage, wie ein zuverlässiges, sicheres, hochleistungsfähiges und skalierbares Datenverarbeitungssystem für die Cloud konzipiert sein sollte.

Mit Snowpark, einem Developer-Framework in Snowflake, können alle Datenbenutzenden ihre Arbeit in der Snowflake Data Cloud zentralisieren – mitsamt nativer Unterstützung für Python, SQL, Java und Scala (siehe Abbildung 1). So können Data Engineers, Data Scientists und Data Developers auf gewohnte Weise und mit der Sprache ihrer Wahl programmieren und Pipelines, ML-Workflows (maschinelles Lernen) und Datenapplikationen schneller und sicherer ausführen – und das auf einer einzigen Plattform. Snowpark erweitert die Snowflake-Vorteile von SQL-Benutzenden auf all Ihre Datenteams. So profitieren Python-, Java- und Scala-Benutzende von der Performance, Flexibilität und Governance der Snowflake-Verarbeitungs-Engine.

Entwickelnde interagieren mit Snowflake über drei wesentliche Snowpark-Komponenten:

- **Snowpark DataFrame API.** Sie erstellen Abfragen mit vertrauten DataFrames – direkt über die Snowflake-Benutzeroberfläche (Preview) oder über Ihre bevorzugte IDE. Push-Down-Prozesse profitieren von der Performance und Skalierbarkeit der elastischen Snowflake-Verarbeitungs-Engine.
- **Snowpark-UDFs.** Sie integrieren eigene Logiken, die in nativem Python- oder Java-Code geschrieben wurden, und führen sie mit User Defined Functions (UDFs) direkt in Snowflake aus.
- **Gespeicherte Prozeduren.** Sie operationalisieren und orchestrieren Ihre Python-, Java- oder Scala-Pipelines und benutzerdefinierten Logiken direkt in Snowflake und stellen sie dann Ihren SQL-Benutzenden zur Verfügung.

Da die Leistung von Python auf seinem umfassenden Ökosystem aus Open-Source-Paketen basiert, sind in Snowpark für Python bereits die beliebtesten Pakete von Anaconda installiert – samt Integration des Conda-Paketmanagers, damit sich Benutzende keine Gedanken über manuelle Installationen oder fehlende Abhängigkeiten machen müssen.

Was können Sie mit Snowpark entwickeln? In diesem E-Book stellen wir Ihnen einige Beispiel-Anwendungsfälle vor, doch die Lösung unterstützt noch viele weitere. Lassen Sie uns anfangen.

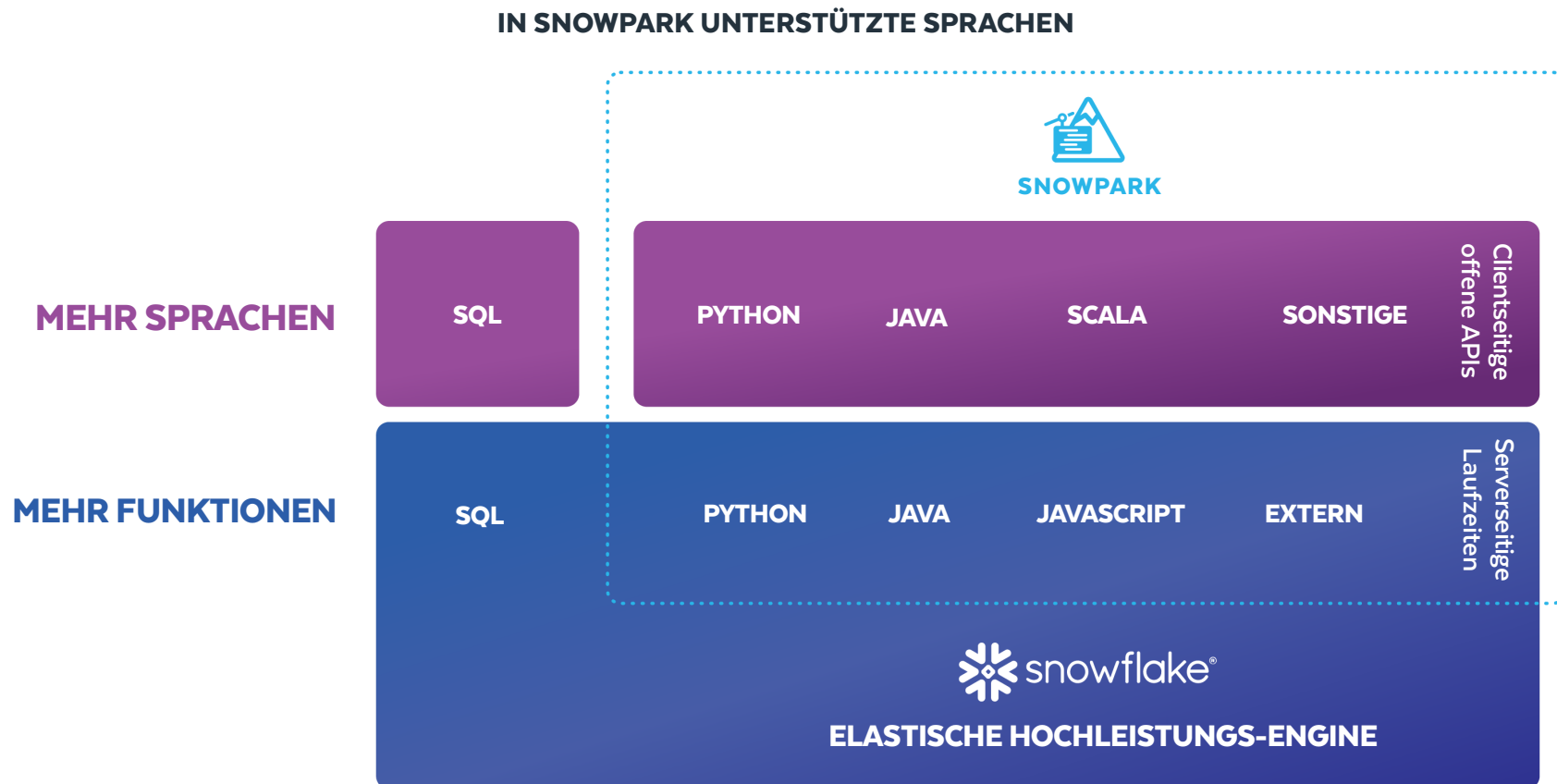


Abbildung 1: Mit Snowpark können Entwickelnde und Techniker:innen direkt mit Snowflake interagieren, ohne dabei Daten verschieben zu müssen.

# ENTWICKLUNG BESSERER PIPELINES UND DATENMODELLE MIT SNOWPARK

Um Ihnen zu vermitteln, wie Snowpark funktioniert, sehen wir uns ein allgemeines Beispiel mit personenbezogenen Daten an.

## OPTIMIERTE ABFRAGEN, NAHTLOSE UMWANDLUNGEN

Mit der Snowpark API können Entwickelnde Abfragen erstellen, indem sie ihrem Code DataFrames hinzufügen. So müssen sie keine SQL-Zeichenfolgen erstellen oder übergeben. Hierzu ein Beispiel:

```
session = Session.builder.configs(connection_params).create()

sales_df = session.table('sales')
line_items_df = session.tables('sales_details')
query = sales_df \
    .join(line_items_df, sales_df.col('id') == line_items_
df.col('sid')) \
    .group_by(line_items_df.col('product_id')) \
    .count()
```

Die Snowpark API bietet umfassende Unterstützung in der Entwicklungsumgebung, beispielsweise mit Typüberprüfung, IntelliSense und Fehlermeldungen. Hinter den Kulissen werden sämtliche DataFrame-Vorgänge auf transparente Weise in SQL-Abfragen umgewandelt, die dann an die skalierbare Snowflake-Verarbeitungs-Engine übertragen werden.

Mit Snowpark können Sie auch eine benutzerdefinierte Python-Logik schreiben – mit UDFs und gespeicherten Prozeduren (Stored Procedures), die direkt in Snowflake ausgeführt werden. Im folgenden Beispiel werden mit benutzerdefiniertem Python-Code die personenbezogenen Daten maskiert:

```
def mask_pii(pii):
    # Custom PII detection logic
```

Mit der Snowpark API können Sie diesen Code ganz einfach als **User Defined Function (UDF)** klassifizieren und diese anschließend in DataFrame-Vorgängen verwenden:

```
mask_pii_udf = udf(mask_pii)

session.table('emails') \
    .with_column('body', mask_pii_udf(col('body'))) \
    .show()
```

Die Snowpark API überträgt Ihre Logik an Snowflake, wo sie in einer sicheren Python-Sandbox gemeinsam mit Ihren Daten ausgeführt wird.

## WEGFALL MANUELLER PROGRAMMIERUNG DANK AUTOMATISIERUNG

Um Ihnen zu zeigen, wie Snowpark allgemeine Abläufe vereinfacht, sehen wir uns an, wie Sie die Erkennung personenbezogener Daten auf alle Zeichenfolgen-Spalten in einer Tabelle anwenden können. Mit SQL müssten Sie manuell eine Abfrage für jede Tabelle programmieren oder Code schreiben, der die Abfrage generiert. Mit Snowpark können Sie stattdessen ganz einfach eine allgemeine Routine schreiben:

```
def mask_table(df):
    return df.select(map(lambda field : mask_pii_udf(col(field.name)) \
        if field.DataType == StringType() \
        else col(field.name), df.schema))
```

Mit dieser allgemeinen Routine können Sie dann mühelos sämtliche personenbezogenen Daten in allen Tabellen maskieren:

```
masked_emails = mask_table(session.table('emails'))
```

Sie brauchen nur wenige Zeilen Code, damit Snowpark dynamisch leistungsstarke schemabasierte Abfragen erstellt.

## KOMPLEXE LOGIKEN MIT PYTHON-UDFS

Mit Python-UDFs können Sie komplexe Logiken in einer einfachen Funktionsoberfläche bereitstellen:

```
class Sentiment:
    def score(text):
```

Bei der Entwicklung dieser Funktionen können Sie Ihr bestehendes Toolset – darunter Versionsverwaltung, Entwicklungsumgebungen und Debugging-Tools – sowie alle Bibliotheken verwenden, die Sie brauchen. Snowpark bietet Ihnen die Möglichkeit, nützlichen Code von GitHub oder anderen Quellen abzurufen und in Snowflake zu nutzen.

Die Verwendung von Python-Code in SQL ist ganz einfach: Erstellen Sie ein Modul, importieren Sie es in Snowflake und registrieren Sie eine Funktion.

```
create function score(text string)
returns float
language python
runtime_version = '3.8'
handler = 'Sentiment.score'
imports = ('@my_stage/sentiment.py')
```

Jetzt können alle SQL-Benutzenden die von Ihnen eingerichtete Logik wie jede andere Funktion verwenden:

```
select id, score(body)
from emails;
```

Sie können UDFs auch mit einem reinen Python-Ansatz erstellen, damit Snowpark Ihre Python-Funktion bündelt und auf der Serverseite mit dem @udf-Dekorator und der Funktion UDF.register bereitstellt.

Mit Python-UDFs können Sie bestehende Tools für komplexe Fälle verwenden. Doch für einfachere Anwendungsfälle beinhaltet diese neue Funktion auch einfache Inline-Definitionen:

```
create or replace function reverse(s string)
returns string
language python
runtime_version = '3.8'
handler = 'Reverse.reverse'
target_path = '@my_stage/reverse.py'
as $$
class Reverse:
    def reverse(s):
        return s[::-1]
$$
```



# SNOWPARK-ANWENDUNGSFÄLLE FÜR DATA ENGINEERING

Data Engineering ist vor allem aus zwei Gründen komplex. Erstens müssen Data Engineers oft mehrere Tools und Infrastrukturen für jede Programmiersprache verwenden, was die Architektur der Daten-Pipeline übermäßig kompliziert macht. Einer der Hauptgründe für die Unterstützung mehrerer Sprachen ist die teamübergreifende Zusammenarbeit beim Data Engineering. Datenanalyst:innen bevorzugen SQL und GUI-basierte Tools. Data Scientists hingegen bereiten Daten lieber über Notebooks und Python vor. Und Data Engineers und Entwickelnde brauchen zusätzliche Tools, um komplexe Code- und Programmieraufgaben zu bewältigen. Daten müssen oft zwischen diesen verschiedenen Systemen übertragen werden, was zu einer komplexen Architektur führt, die Sicherheit und Data Governance beeinträchtigen kann.

Zweitens ist das Verwalten von und das Arbeiten mit einer Infrastruktur zur Datenverarbeitung in der Regel mit erheblichem manuellen Aufwand verbunden. Dementsprechend sind Data Engineers oft überlastet und verbringen den Großteil ihrer Zeit damit, Pipelines zu warten und zu reparieren.

Mit **Snowpark**, einem modernen Developer-Framework für Snowflake, können diese Data Engineers einfache, kontrollierte und schnelle Pipelines in der Programmiersprache ihrer Wahl entwickeln. Snowpark bietet Data Engineers drei wichtige Vorteile: eine zentrale Plattform, die mehrere Sprachen unterstützt und ohne externe Rechenressourcen auskommt, herausragende Governance sowie schnellere, günstigere und resilientere Pipelines. Die intelligente Infrastruktur von Snowflake beseitigt die Komplexität, denen Data Engineers häufig gegenüberstehen – die Lösung funktioniert einfach, sodass sich Teams auf das konzentrieren können, was wirklich zählt.

Data Engineers nutzen Snowpark in der Regel für folgende Bereiche:

- **ETL/ELT:** Datenteams können mit Snowpark Rohdatenmodelle in modellierte Formate umwandeln, und zwar unabhängig vom Typ – darunter JSON, Parquet und XML. Alle Datentransformationen können dann als Snowpark Stored Procedures gespeichert werden, um Jobs über Snowflake Tasks oder andere Orchestrierungstools auszuführen und zu planen.
- **Benutzerdefinierte Logik:** Benutzende können mit Snowparks User Defined Functions (UDFs) die Architektur optimieren – ohne komplexe Datenverarbeitung und benutzerdefinierte Python- oder Java-Logiken und auf derselben Plattform, wo auch SQL-Abfragen und Datentransformationen ausgeführt werden. Sie müssen keine separaten Cluster managen, skalieren oder betreiben.
- **Data-Science- und ML-Pipelines:** Datenteams können das integrierte Anaconda-Repository und den Paketmanager verwenden, um gemeinsam ML-Daten-Pipelines in die Produktion zu übertragen. Die trainierten ML-Modelle können auch als UDF gespeichert werden, um die Modellinferenz nahe bei Ihren Daten auszuführen. So gelangen Modelle schneller von der Entwicklung in die Produktion.

## ANWENDUNGSFALL: DATENTRANSFORMATION MIT DBT

Bei der Datentransformation werden Quelldaten bereinigt, kombiniert und modelliert, damit sie später genutzt werden können. Bisher war SQL die beliebteste Methode zur Datentransformation, und Datenexpert:innen haben Datentransformations-Pipelines mit SQL entwickelt, oft mithilfe von ETL/ELT-Tools. Aber in letzter Zeit wird jedoch immer häufiger die DataFrame-API in Sprachen wie Python für diese Aufgabe eingesetzt. In den meisten Fällen können Datenfachleute mit beiden Ansätzen die gleichen Datentransformationen durchführen. Für welchen sie sich entscheiden, ist meist eine Frage der Präferenz und der jeweiligen Anwendungsfälle. Es gibt jedoch auch Anwendungsfälle, in denen eine bestimmte Datentransformation nicht in SQL ausgedrückt werden kann und deshalb ein anderer Ansatz erforderlich ist. Der beliebteste Ansatz für diese Anwendungsfälle ist die Kombination aus Python und einer DataFrame-API.

Eine der beliebtesten Lösungen für die Datentransformation ist derzeit dbt. Die Lösung unterstützt einen SQL-first Transformations-Workflow. 2022 hat dbt die Unterstützung für Python eingeführt. Da dbt nun sowohl SQL als auch Python unterstützt, können Benutzende Transformationen in der Sprache schreiben, mit der sie am besten vertraut sind und die sich optimal für den jeweiligen Anwendungsfall eignet. Und mit dbt in Snowpark können Benutzende Analysen über Tools durchführen, die im Open-Source-Ökosystem von Python verfügbar sind, darunter aktuelle Pakete für Data Engineering und Data Science – und das alles innerhalb des dbt-Frameworks, mit dem viele SQL-Benutzende vertraut sind.

## ANWENDUNGSFALL: BEREITSTELLUNG ZUVERLÄSSIGER DATEN MIT TALEND

Ihre Analysen sind immer nur so gut wie Ihre Daten. Aber wie messen Sie die Qualität dieser Daten? Sie können jetzt Zustandsprüfungen Ihrer Daten in Snowflake durchführen, indem Sie den Data Trust Score™ von Talend nutzen – eine Kernfunktion des Produkts „Talend Data Inventory“. Der Trust Score™ hilft Ihnen dabei, wichtige Probleme zu erkennen und zu diagnostizieren, und ermöglicht es allen wichtigen Stakeholdern, fundiertere Entscheidungen zu treffen.

Um Profile aus großen Dataset-Mengen einer externen Applikation zu erstellen, wird in der Regel eine Stichprobe Ihrer Daten extrahiert. So kann die Analyse auf einem anderen System durchgeführt werden als auf dem, wo die Daten gespeichert sind. Wenn Sie Daten außerhalb von Snowflake übertragen, entstehen hierdurch ganz eigene Herausforderungen: Risiken rund um Datensicherheit und -schutz, Gebühren für den Dateneingang und -ausgang, fehlgeschlagene Verarbeitung aufgrund nicht skalierbarer Ressourcen oder auch ungenaue Ergebnisse, da sie nur auf einer Stichprobe der relevanten Daten beruhen.

Durch die Berechnung des Trust Score™ für Snowflake-Datasets fallen viele dieser Herausforderungen weg, da die Berechnung des Scores nativ innerhalb von Snowflake mithilfe von Java-UDFs erfolgt. Er ist äußerst genau, da die Ergebnisse auf dem gesamten Dataset basieren, nicht nur auf einer Stichprobe. Java-UDFs können entweder mit **Snowpark** oder **SQL** aufgerufen werden. Talend nutzt dabei beide Ansätze: einen während der Prototyping-Phase und den anderen für die Operationalisierung der Funktion in unserem Produkt „Talend Data Inventory“.

## ANWENDUNGSFALL: VERWALTUNG DES ENTWICKLUNGSLEBENSZYKLUS MIT DATAOPS.LIVE

Immer mehr Datenprodukte werden mithilfe der fortschrittlichen Funktionen in Snowpark entwickelt. Durch die erweiterte Partnerschaft zwischen DataOps.live und Snowflake können Snowpark-Workloads gemeinsam mit allen anderen Objekten in Snowflake verwaltet werden – über den gesamten Lebenszyklus von Entwicklung, Test und Produktion hinweg. Das reicht von leistungsstarkem Data Engineering über das regelmäßige automatische Neutrainieren von ML-Modellen bis hin zum Ausführen von Inferenzen, wenn neue Daten in Pipelines geladen werden. Zusätzlich zur normalen Source of Truth, die im Git-Repository von DataOps.live gespeichert ist, erfüllt das Repository auch alle gängigen Anforderungen der Software-Entwicklung. Mit DataOps.live können Sie Software verzweigen, versionieren, kompilieren, testen sowie Artefakte erstellen – ganz so, wie Sie es von jedem anderen Softwareprojekt gewohnt sind.

In vielen Fällen führt eine Snowpark-Applikation erweiterte Datenmanipulationen durch, die über die Möglichkeiten in SQL hinausgehen. Die Ergebnisse werden jedoch in Snowflake gespeichert. In diesen Fällen können die Ergebnisse der Snowpark-Applikation mithilfe der automatisierten Datentests in der Modellierungs- und Transformations-Engine von DataOps.live validiert werden.

# SNOWPARK-ANWENDUNGSFÄLLE FÜR DATA SCIENCE

Die Fortschritte bei der Datenprogrammierbarkeit in Snowpark sorgen für mehr Flexibilität und Erweiterbarkeit. Data Scientists können in Snowflake ihre bevorzugten Programmiersprachen nutzen, wie z. B. Python, um auf Daten zuzugreifen, sie zu visualisieren und sie im Rahmen ihrer ML-Workflows zu verarbeiten. So können Unternehmen den Wert ihrer Daten maximieren, einschließlich unstrukturierter Daten oder auch Daten von Drittanbietern.

Unabhängig von der verwendeten Sprache übernimmt die intelligente Infrastruktur von Snowflake die Skalierung und Performance-Optimierung, damit Data Scientists mehr Zeit mit der Entwicklung von Modellen verbringen können. Das umfasst auch das groß angelegte Single-Node-Training mit Snowpark-Optimized Warehouses. Diese verfügen im Vergleich zu Standard-Warehouses über den 16-fachen Speicher und den 10-fachen lokalen Cache und eignen sich hervorragend für speicherintensive Vorgänge wie Training und Inferenz. Externe Funktionen können Daten in extern gehostete Modelle einspeisen, in denen Data Scientists möglicherweise bereits Onlineprognosen oder Prognosen mit niedriger Latenz durchführen.

Sämtliche Schritte, die Rohdaten in ML-Einblicke umwandeln, können mit Snowflake Streams und Tasks automatisiert werden. Doch sie lassen sich natürlich auch in andere Transformations- und Orchestrierungs-Frameworks integrieren, darunter dbt und Apache Airflow.

Und Snowflake geht sogar noch einen Schritt weiter: Hier wird das Modell mit der Option bereitgestellt, Apps ausschließlich über Python mit Streamlit zu erstellen – einer Integration, die sich derzeit in der Private Preview befindet.

Die Erweiterbarkeit von Snowflake, damit Benutzende das Tool ihrer Wahl verwenden können, geht über IDEs hinaus und deckt auch führende Plattformen ab, die Snowflake zum Speichern und Verarbeiten von Daten verwenden. So erhalten Sie auch über andere Plattformen wie Hex Notebooks, Dataiku MLOps-Plattformen oder Tecton Feature Stores ein großartiges Benutzererlebnis. Sie können sogar Low-Code-Plattformen einsetzen, um die Vorteile der Snowflake Data Cloud auf viele Benutzende innerhalb Ihres Unternehmens zu erweitern.

## ANWENDUNGSFALL: EINHEITLICHE TOOLS UND INFRASTRUKTUREN FÜR SQL UND PYTHON MIT HEX

SQL und Python gehören zu den beliebtesten Programmiersprachen im modernen Data Stack für Transformationen, Analysen und ML. Während SQL schon seit Langem die bevorzugte Datenbanksprache für Datenabfragen und -transformationen ist, hat sich Python als Programmiersprache für Data Science und ML durchgesetzt. Data Scientists und Data Engineers müssen oft mehrere Tools kombinieren, um eine einzige Analyse abzuschließen, wenn sie mit mehreren Sprachen arbeiten. Das ist nicht nur frustrierend, sondern nimmt auch viel Zeit in Anspruch.

### HEX UND SNOWFLAKE IN AKTION

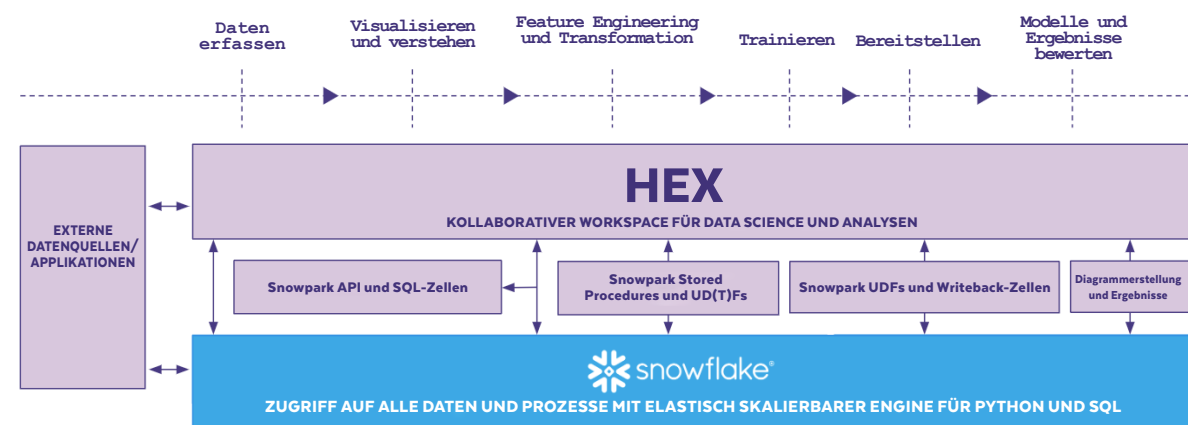


Abbildung 2: Hex-Snowflake-Integration für Data Science und Analytik

Hex ist eine moderne Plattform für Analytik und Data Science, mit der Sie ganz leicht Verbindungen zu Daten herstellen, diese Daten in kollaborativen SQL- und Python-basierten Notebooks analysieren und die Arbeit dann in Form interaktiver Datenvisualisierungen und -Dashboards teilen können. Um Benutzenden schnelle, nahezu unbegrenzte Verarbeitungsskalierbarkeit zu bieten, überträgt Hex die Datenberechnung auf die Snowflake-Plattform (siehe Abbildung 2), anstatt Daten in ein Notebook zu laden, was die Geschwindigkeit beeinträchtigen und Latenzen verursachen kann. Über den Analyse-Workspace von Hex können Sie mithilfe von Python und SQL in nur wenigen Minuten Einblicke aus den Daten in Snowflake gewinnen.

### ANWENDUNGSFALL: SKALIERUNG VON DATENAUFBEREITUNGSFUNKTIONEN UND PRÄDIKTIVE MODELLBEWERTUNG MIT DATAIKU

Dataiku bietet eine umfassende visuelle Oberfläche für Geschäfts- und Datenteams, in der sie Daten aufbereiten und Prognosemodelle Schritt für Schritt in einer visuellen Daten-Pipeline erstellen können. No-Code-, Low-Code- und Full-Code-Benutzende wie Datenanalytist:innen, Fachleute und Data Scientists können Dataiku mit leistungsstarkem AutoML verwenden, um das beste Modell zu finden – mit vollständiger Unterstützung für Code-Notebooks, IDEs, Git und CI/CD-Tools. Mit Dataiku können Benutzende ohne zusätzliche Konfiguration direkt in Snowflake auf Daten zugreifen, indem sie entweder No-Code-Rezepte verwenden oder in der Sprache ihrer Wahl programmieren, einschließlich Python mit Snowpark. Mit der Snowpark-Integration von Dataiku können Benutzende über die Dataiku-Benutzeroberfläche Daten aufbereiten, Modelle bewerten und diese Modelle dann bereitstellen, ohne Daten aus Snowflake zu verschieben (siehe Abbildung 3). Alle Berechnungen von Datasets – egal, ob groß oder klein – werden in Snowflake übertragen, um eine sichere, kontrollierte Verarbeitung zu gewährleisten. Mit der Snowflake-Integration von Dataiku können Daten-, Fach- und IT-Teams in einer einzigen Umgebung zusammenarbeiten, um groß angelegte Data-Science-Projekte aufzubauen und bereitzustellen.

## DATAIKU UND SNOWFLAKE IN AKTION

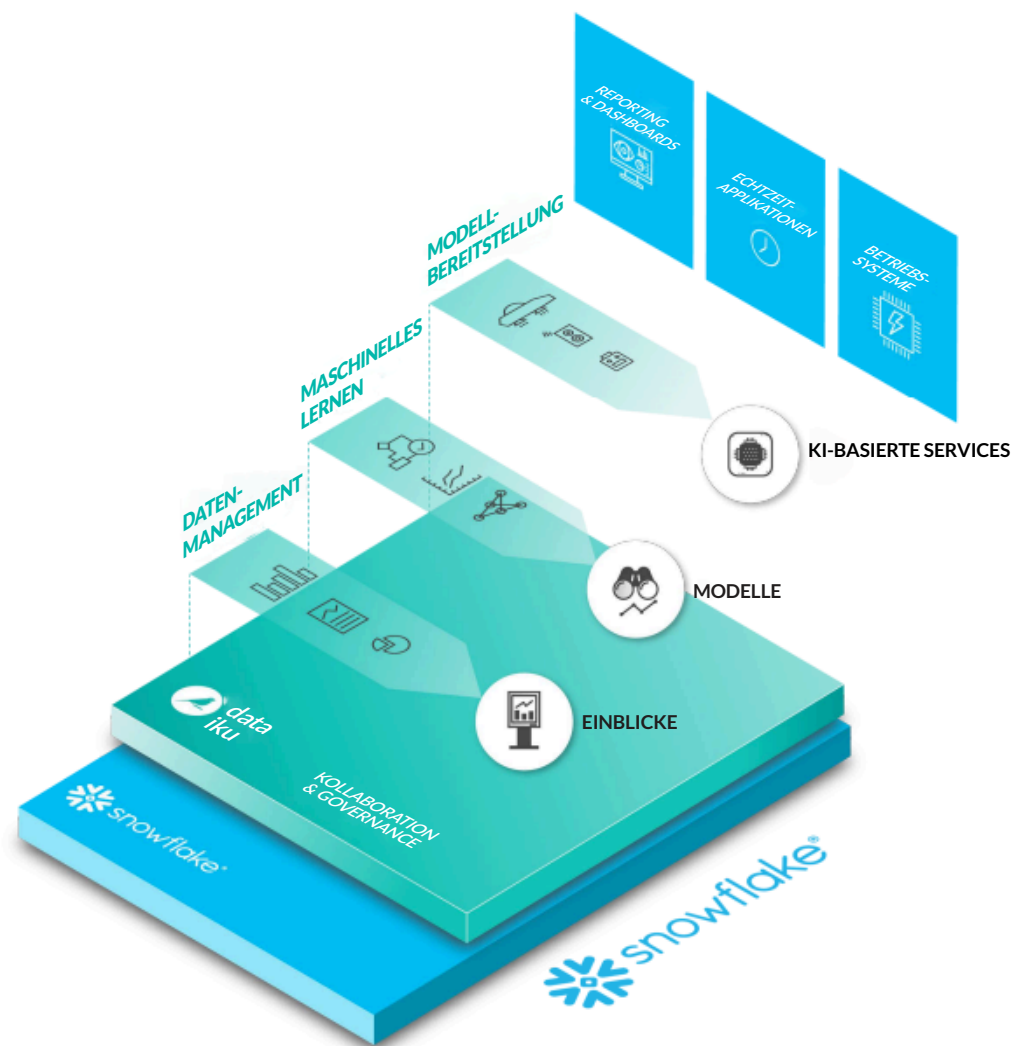


Abbildung 3: Mit der Dataiku-Integration können Benutzende Daten aufbereiten, Modelle bewerten und Modelle bereitstellen, die Daten in Snowflake verwenden.

## ANWENDUNGSFALL: VERWALTEN UND BEREITSTELLEN UMFANGREICHER ML-FEATURES MIT FEATURE STORES

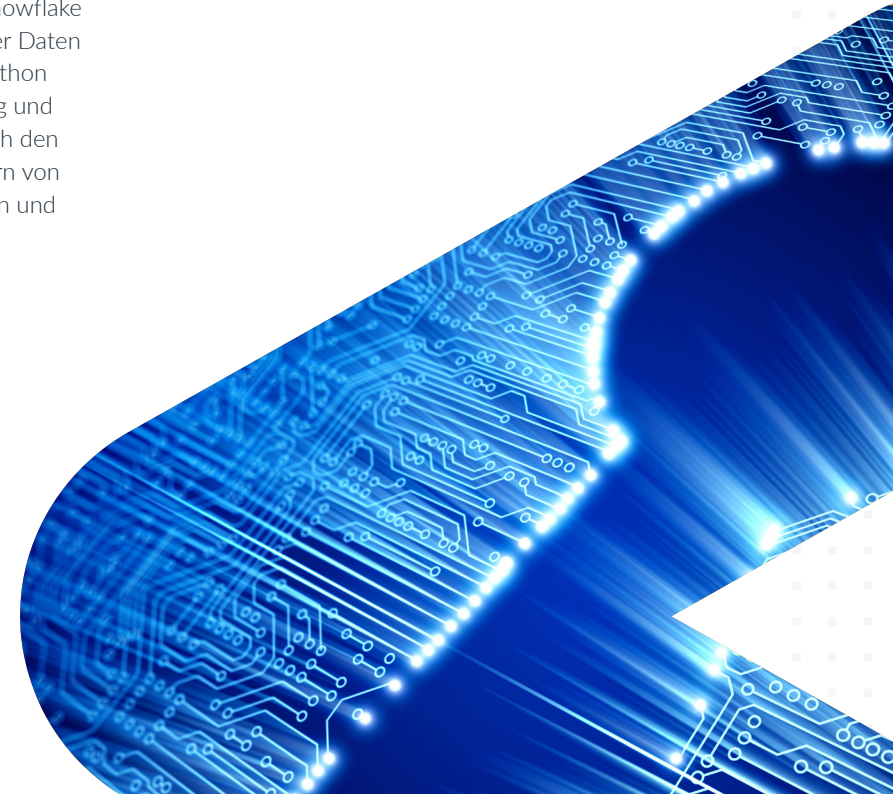
Data Scientists stehen bei der Entwicklung neuer ML-Modelle vor der mühsamen Aufgabe, Daten aufzubereiten und Features zu erstellen. Features werden durch Beschaffung und Aufbereitung von Datenspalten in einem bestimmten Format erstellt, die in Modelle für maschinelles Lernen eingespeist werden können. Sobald die Features für ein Modell erstellt wurden, stehen Data Scientists vor der nächsten Herausforderung: entweder dieselben Features neu schreiben oder Zeit damit verbringen, nach vorhandenen Features zu suchen, um sie für das nächste Modell zu verwenden.

Glücklicherweise gab es in letzter Zeit einen starken Anstieg bei der Einführung von Feature Stores – einem zentralen Repository, das die Durchsuchbarkeit, Zusammenarbeit und Skalierbarkeit von ML-Features verbessert. Data Scientists können schnell Features finden, die transformiert und einsatzbereit sind, was sowohl zu schnelleren Experimenten als auch zu kürzeren Produktionszeiten führt. Zu den Vorteilen eines Feature Stores gehört die Möglichkeit, die Zusammenarbeit zwischen Teams durch die Wiederverwendung der Arbeit anderer Data Scientists zu verbessern. Feature Stores reduzieren außerdem den Zeit- und Arbeitsaufwand für die Bereitstellung eines trainierten Modells in einer Produktionsumgebung, da Data Scientists die oft vorhandene Daten-Pipeline nicht mehr neu definieren müssen.

Snowflake bietet zwei Ansätze für den Aufbau von Feature Stores. Bei beiden wird vermieden, dass neue Systeme oder Datensilos zwischen Data Scientists und Datenanalytist:innen geschaffen werden.

Der erste Ansatz besteht darin, eine Open-Source-Lösung wie Feast als Schnittstelle für den Feature Store zu nutzen, während Snowflake als Store und Engine für die Features dient. Die Features verbleiben auf der zentralen Datenplattform und werden von allen vorhandenen Erfassungs-, ELT- und Katalogisierungstools unterstützt. Bei dieser Lösung verfügen Unternehmen sowohl über die Verwaltung der Daten-Pipelines als auch über die Schnittstelle, über die Data Scientists Daten und Features an einem zentralen, skalierbaren Ort auffinden und abrufen können.

Der zweite Ansatz besteht darin, zusätzlich zu Snowflake eine „Managed Feature Store“-Lösung zu nutzen. Die Daten der Kund:innen verbleiben in ihrer rohen und modellierten Form in der Snowflake Data Cloud. Während die Transformation der Daten in Snowflake mit SQL oder Snowpark für Python ausgeführt wird, erfolgen die Orchestrierung und die Verwaltung der Pipeline abstrahiert durch den Anbieter des Feature Stores. Zu den Partnern von Snowflake in diesem Bereich gehören Tecton und Iguazio.



# ANWENDUNGSFÄLLE VON ANDEREN SNOWFLAKE-PARTNERN

Snowpark verfügt über ein umfassendes Partnernetzwerk, mit dem Data Engineers, Data Scientists und Entwickelnde, die andere Sprachen bevorzugen – einschließlich Python, Java und Scala – die leistungsstarken Plattformfunktionen von Snowflake und die vielen Vorteile der Snowflake Data Cloud nutzen können. Diese tief integrierten Funktionen beschleunigen das Entwickeln und Bereitstellen von Pipelines, Modellen und Applikationen. Weitere Informationen zu Partnerintegrationen finden Sie auf der Seite von [Snowpark Accelerated](#).

# IHR EINSTIEG IN SNOWPARK

Um mit der Nutzung von Snowpark zu beginnen, lesen Sie bitte unsere [Dokumentation](#) und den [Schritt-für-Schritt-Quickstart-Guide](#). Wir sind gespannt darauf, was Sie alles entwickeln werden.





# ÜBER SNOWFLAKE

Die Snowflake Data Cloud bietet jedem Unternehmen die Möglichkeit, seine Daten zu mobilisieren. Mithilfe der Data Cloud können Kunden Datensilos zusammenführen, Daten entdecken und sicher freigeben, Datenapplikationen unterstützen sowie verschiedene KI/ML- und analytische Workloads ausführen. Wo auch immer sich Daten oder Benutzer befinden, Snowflake bietet eine einheitliche Datenlösung, die sich über mehrere Clouds und geografische Regionen erstreckt. Tausende von Kunden in zahlreichen Branchen nutzen die Snowflake Data Cloud und bringen so ihre Unternehmen voran. Darunter fallen auch 639 Unternehmen der Forbes Global 2000 (G2K) aus dem Jahr 2023 (Stand: 31. Juli 2023).

Erfahren Sie mehr unter [snowflake.com](https://www.snowflake.com)



© 2023 Snowflake Inc. Alle Rechte vorbehalten. Snowflake, das Logo von Snowflake und alle sonstigen hier erwähnten Namen von Produkten, Funktionen und Services von Snowflake sind eingetragene Marken oder Marken von Snowflake Inc. in den USA und anderen Ländern. Alle anderen erwähnten oder verwendeten Markennamen oder Logos dienen ausschließlich der Identifikation und können die Marken ihrer jeweiligen Eigentümer sein. Snowflake darf nicht mit diesen Eigentümern in Verbindung gebracht oder von diesen unterstützt oder gefördert werden.