snowflake®

构建人工智能管道：智能数据工程必备指南

由吉尔伯托·赫纳兹 主导开发者布道师，Snowflake

目录

需求与设计：驾驭数据工程两大转变

当今的数据工程正经历着两大主要转变——职能转变和形式转变。第一种转变显而易见：人工智能正在从根本上重新定义数据工程师在各个层面的职能。它对数据的无限需求给数据工程团队带来了过度的要求——这些要求对于成功至关重要，然而却极其难以维持。第二种转变是形式上的转变，即数据工程师必须如何满足这些新增长的需求。我们已经看到数据工程师从主要从事重复性、手动劳动转变为更具战略性的执行，借鉴软件开发的最佳实践来提升他们的工作。他们不再仅仅是数据管道工和管道构建者；他们是任何数据驱动型组织的运营架构师。而且，此时已无法回头。

当我们思考现代数据工程时，重点已不再是手动连接每一个点。这根本无法满足人工智能的需求。随着数据量呈指数级增长，并且迅速变得可用和可使用，工程师需要更高效地工作以保持同步。

正是这种更现代、更偏向断言式的管道构建方法，为数据工程师彻底改变了游戏规则。通过抽离每个步骤的细枝末节，转而专注于期望的最终状态，数据工程师得以提升其生产力，实现此前看似遥不可及的收益。

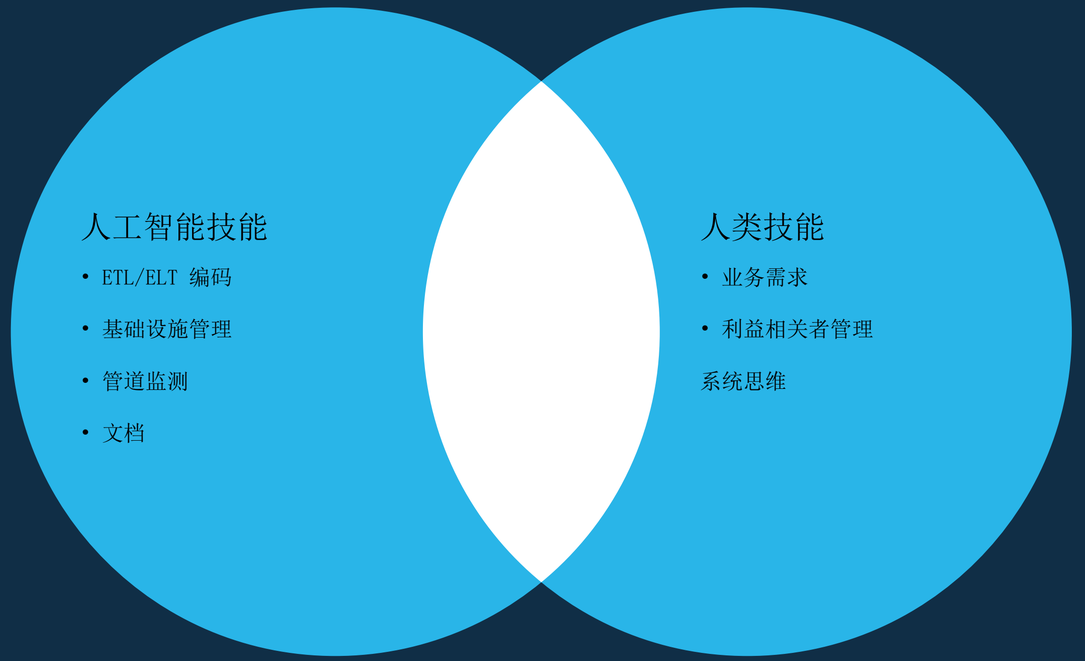
以编码代理为例。短短数月内，这些工具，包括Cursor、Claude Code和Snowflake的Cortex Code，就彻底改变了我们对软件开发乃至数据工程的认识。这是如何实现的？多年来，数据工程团队一直默默借鉴软件定义生命周期的最佳实践。他们将基础设施视为代码，并创建结构化、版本控制的环境，其中数据管道与无状态软件代码非常相似。由于这些AI编码代理在软件工程问题上接受了大量训练，因此它们能够轻松适应这种现代数据工程形式。

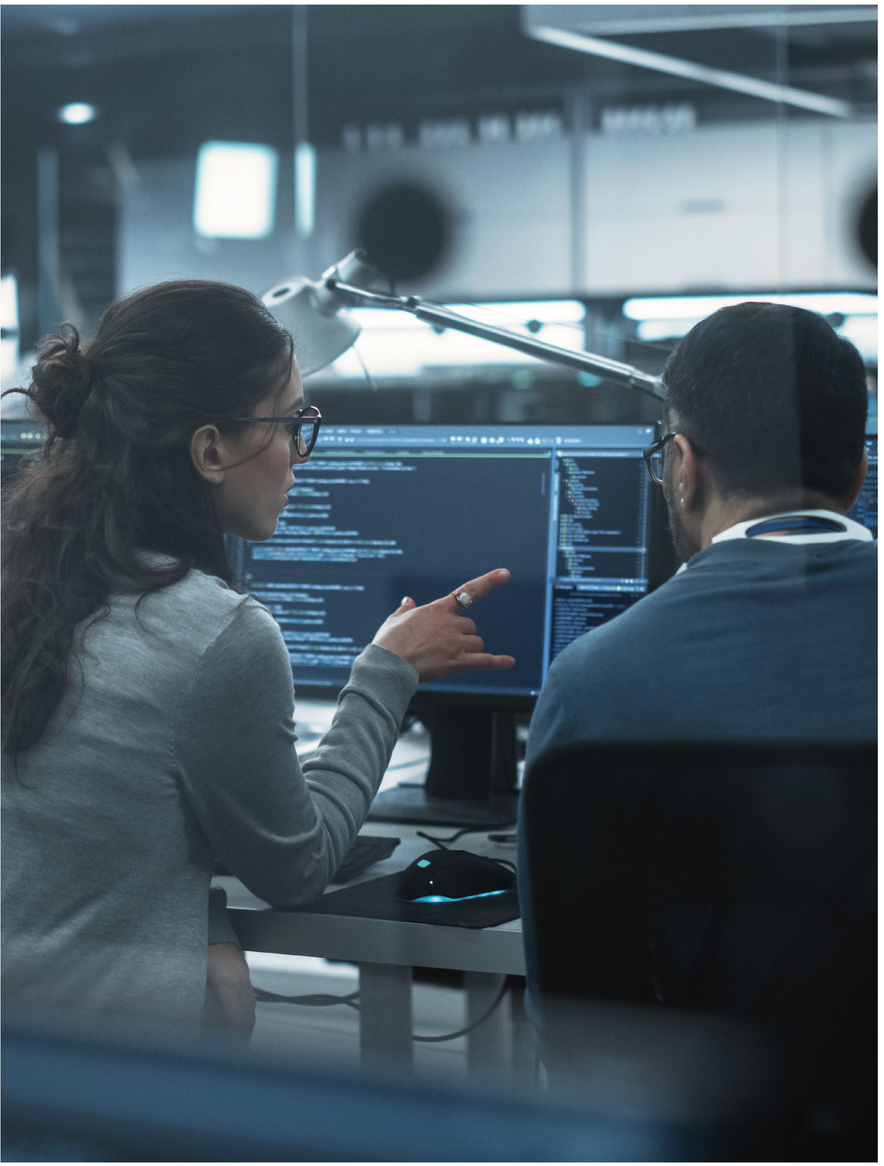
这种方法的转变——转向更现代、更明确的思维方式——为使AI工具发挥作用创造了合适的条件。但更重要的是，它提供了AI规模化运行所需的“安全网”。过去，修复管道意味着直接在生产环境中运行原始SQL命令；但如果出现故障，调查问题会极其复杂。如今，现代方法意味着变更会被提交到版本控制，只有在确认是良好状态时才会测试和部署。能够轻松测试变更并回滚，是信任AI编写或管理数据工作流之前的一个严格前提。

如今，信任人工智能并非意味着盲目迷信。关键在于建立对底层数据工程流程的信任。我们已看到组织同时运行数千个数据管道，达到人力几乎无法监控每个运行环节的阶段。很快，我们将迈向自主型人工智能时代，软件代理将承担更大规模的实际管道构建工作。数据工程师的角色将再次提升，从编写单个脚本转向高级数据建模和系统需求。他们将更贴近业务，为人工智能、分析和应用确保数据的可用性与质量。

归根结底，数据工程未来的关键并非编写更好的脚本来移动数据，而是构建能够为您连接数据的弹性系统。本书旨在帮助您实现这一目标。书中，首席开发者布兰道特·赫南德斯将带您了解数据管道的ITD（摄取-转换-交付）框架。他将在介绍每个步骤相关的传统方法——包括其优点和缺点——之后，重点介绍现代工具和方法，以帮助数据工程团队适应眼前的变化格局。

— 雪flake 数据工程产品副总裁 Chris Child





ITD框架：入门指南

数据工程领域非常多样化，提供了许多独特的挑战以及大量不同的应用场景——所有这些都可以通过多种方式解决，使用多种工具。

本学科的广泛范围增加了两个主要转变，这些转变正在实质性地影响数据工程师如何设计和部署数据管道：(1) 人工智能的需求和使用案例正在给数据管道需要解决的问题带来压力；(2) 经过充分测试、成功的软件工程最佳实践正在进入数据工程领域。这意味着数据工程师不仅要改进他们的工作流程，还要改进他们设计和部署、维护数据管道的思维模型。简而言之，对数据工程师的要求越来越高，现代数据管道正在演变为反映这些转变和需求。

摄取、转换和交付是数据管道中常被引用的“ITD框架”的核心组成部分。这些组成部分涵盖了将原始数据摄取到数据平台中，将其从原始状态转换为可用的状态以促进洞察提取，以及交付使用转换后数据的产品——例如仪表板、应用程序，以及现在与人工智能相关的产物，如语义视图。如果你是一位经验丰富的数据工程师，但对ITD框架不熟悉，你会发现它与提取、加载、转换（ELT）工作流程是相当的。

摄入

转型

配送

无论您倾向于遵循哪种管道框架，其组成部分正受到解决和扩展人工智能用例的需求以及转向使用软件开发最佳实践来设计数据管道的根本影响。

例如，数据摄取——即原始数据加载到数据平台的过程——正受到根本性影响。人工智能要求所有数据，无论其来源如何，都必须被摄取到数据平台，并尽可能快地进行处理。

数据转换将原始状态的数据转化为针对特定应用场景的精炼状态。AI解决方案要求数据转换过程自动进行，无需维护，并在可能的情况下使用人工智能来执行转换。

数据经过转换后，数据工程师会提供包含这些数据的产品，例如经过策划的数据集、仪表盘和数据应用。人工智能正在增加管道应当提供的新一代可行数据产品，例如为人工智能代理提供上下文语义层，以及任何业务用户都可以从中提取洞察的数据对话界面。

这是数据工程师的职责，要解决这些挑战。他们负责设计并部署能够解决传统数据相关挑战的数据管道。

其作用正受到人工智能的深刻影响，这不仅体现在摄取、转换和交付数据产品等各个环节，也反映在人工智能为每个环节带来的新挑战。

在本电子书中，我们将为您介绍Snowflake中所有现代化的、面向AI的数据工程工具，这些工具旨在简化人工智能数据工程师所处理的数据全生命周期。我们将向您展示如何利用AI进行数据摄取（在某些情况下，也可以连接到数据而无需移动数据）、数据转换和数据产品交付。

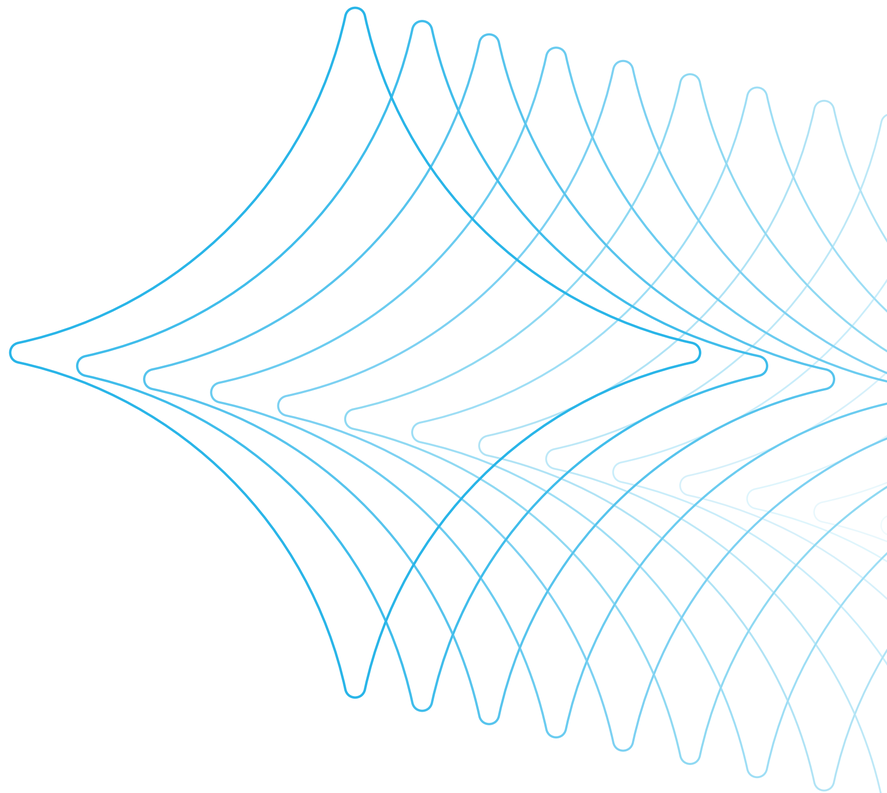
《雪花如何完美融合》

持续流：为AI速度而设计数据摄取架构

“数据是人工智能的食粮。” 这句出自世界著名数据科学家和人工智能远见者Andrew Ng的睿智之言，反映了当前对数据工程领域产生了重大影响的实质性转变。具体而言，对人工智能解决方案的持续需求正迫使数据工程师重新评估他们管理、架构和实现对大量以非结构化为主的实时数据访问的方式。这通常需要将数据从源系统迁移到可访问且可扩展的湖仓（lakehouses）。数据迁移的关键部分之一，是许多管道中的第一步：数据摄取（data ingestion）。

数据摄取是指将原始、未处理的数据从源加载到目标的过程。通常，目标是一个特定的工具、本地存储解决方案或一个集中的云数据平台。摄取过程发生的速率被称为摄取延迟，并且它可以因用例而异。

在某些情况下，数据摄取可以大大简化。例如，许多现代数据平台现在通过连接器支持零拷贝解决方案进行数据摄取，这使得数据团队可以直接将数据转换和人工智能应用于源数据。对于 Apache Iceberg™ 等开放表格式中的数据也是如此。许多现代数据平台现在支持 Iceberg 集成，这意味着它们可以即时连接到外部平台中的 Iceberg 数据，并直接处理该数据，而无需复制或移动数据。



人工智能对传统模式的影响

批处理优先摄取

按预定时间间隔批量摄入数据，数十年来已成为一种普遍且广受接受的模式。数据工程师通常按日或按小时摄入数据，很长一段时间里，这种方式满足了绝大多数用例需求。人工智能正在从根本上改变这一现状。通过向人工智能和自主系统提供新鲜数据，可以获取竞争优势。以制造业或金融服务行业为例，实时数据能够帮助人工智能解决方案提高生产效率，或帮助团队快速应对最新的市场趋势。人工智能正在降低对过时数据的容忍度，数据工程师现在必须将连续摄入工作流程架构到更多的管道中。

结构化与非结构化数据成为常态

多年来，在数据工程领域，摄取结构化数据（例如行和列）一直是常态。当半结构化数据（如JSON）大规模出现时，数据工程师们进行了适应，并为其构建了可靠的摄取模式。然而，情况并非如此（指非结构化数据）。人工智能分析并从PDF、图像、音频文件甚至视频中提取价值的能力，正给数据工程师带来压力，要求他们为这些数据类型找出可比较的、经过充分测试的摄取解决方案。

定制脚本和连接器蔓延

随着数据解决方案范围的扩大，构建自定义脚本或集成第三方工具以连接数据源是不可避免的。但人工智能极大地增加了可用数据源的数量。谷歌文档、Slack 消息、PDF 文件、社交媒体、通话录音——所有这些都已成为极具可行性和潜在价值的数据源，为人工智能提供了更丰富的上下文层。为每个数据源构建自定义解决方案或购买现成连接器既难以管理，又难以扩展。数据工程师需要更精简的方法来连接这些爆炸式增长的数据源。

人工智能领域的新需求

人工智能在处理向量嵌入、标记数据集以及其他为机器学习优化的丰富数据方面效率很高。但从零开始快速实现这些功能可能是一项挑战。这一需求催生了一个全新的数据类别：AI就绪数据集，它们是数据管道（为人工智能提供数据）的即插即用解决方案。如今，数据工程师需要访问预先构建和筛选的数据集，而不是总是从基础开始构建数据摄取管道。

解决方案

雪花（Snowflake）提供了多种功能，旨在应对人工智能（AI）工作负载所要求的容量、种类和速度。

雪管道

Snowpipe 可实现从云存储到 Snowflake 的持续、无服务器的数据摄取。与调度批量加载不同，Snowpipe 会自动检测并加载 Amazon S3、Google Cloud Storage 或 Microsoft Azure Blob Storage 中新到达的文件。数据在到达后几分钟内即可加载完成，无需您配置或管理计算资源——直接实现了从批量优先到持续摄取的转变。

雪鸟流式传输

雪pipe流式传输通过API将低延迟数据直接摄入Snowflake表，无需在云存储中进行中间暂存即可实现亚秒级延迟。与传统雪pipe不同，雪pipe流式传输通过简单的软件开发工具包（SDK）直接接收来自您应用程序、物联网设备或流式平台的数据行。其核心优势在于无需专用流式基础设施（如Kafka集群），即可作为无服务器、按量付费的服务提供实时摄入。

雪花Openflow

雪花 Openflow 帮助数据工程师同时解决从众多数据源、以不同数据格式、以期望的摄取速度摄取数据所面临的挑战。它提供了一个可视化的、基于流的用户界面，用于连接源与目的地，并快速设置摄取延迟。它支持超过 20 个针对常用数据源的预构建连接器——例如 PostgreSQL 等流行数据库、Google Drive 等 SaaS 应用程序等——以及针对特殊用例的自定义流程。Openflow 支持结构化、半结构化和非结构化数据的摄取，直接应对了摄取传统行列之外的数据类型的压力。

连接器（零ETL）

雪花云的原生连接器——通常被称为“零ETL”连接器——为雪花云与Salesforce、ServiceNow、Google Analytics和Microsoft Dynamics等流行数据源之间提供直接集成。这些连接器由雪花云构建和维护，无需第三方摄取工具或自定义管道代码。数据直接从源流向雪花云，无需中间暂存或单独的管理基础设施，这直接解决了连接器蔓延的问题。

Apache Iceberg™ 表格

Apache Iceberg TM 是一种为大规模分析工作负载设计的开放表格式。Snowflake 为 Iceberg 表提供原生支持，允许数据工程师以 Iceberg 格式导入数据，并在 Snowflake 内直接查询。这对那些拥有现有 Iceberg 数据的团队（这些数据可能由 Apache Spark™、Trino 或其他引擎编写）而言意义重大，他们希望将这些数据直接导入 Snowflake 而无需转换。对于需要在多个工具之间实现数据互操作的团队来说，Iceberg 提供了他们所需的开放性。

雪花市场（AI就绪型数据产品）

雪花市场为用户提供了对数千种由数据提供者共享的数据产品的访问权限——其中包括专门为机器学习和人工智能工作负载精心策划的AI就绪数据集。从数据摄取的角度来看，市场代表了一种根本不同的方法：您无需构建管道从源中提取数据，而是访问已存在于雪花中的数据。当您在市场上订阅一个数据集时，该数据会作为共享数据库出现在您的雪花账户中——实时、可查询，并由提供者自动更新。这直接满足了AI就绪数据的新需求，而无需承担从头构建管道的开销。

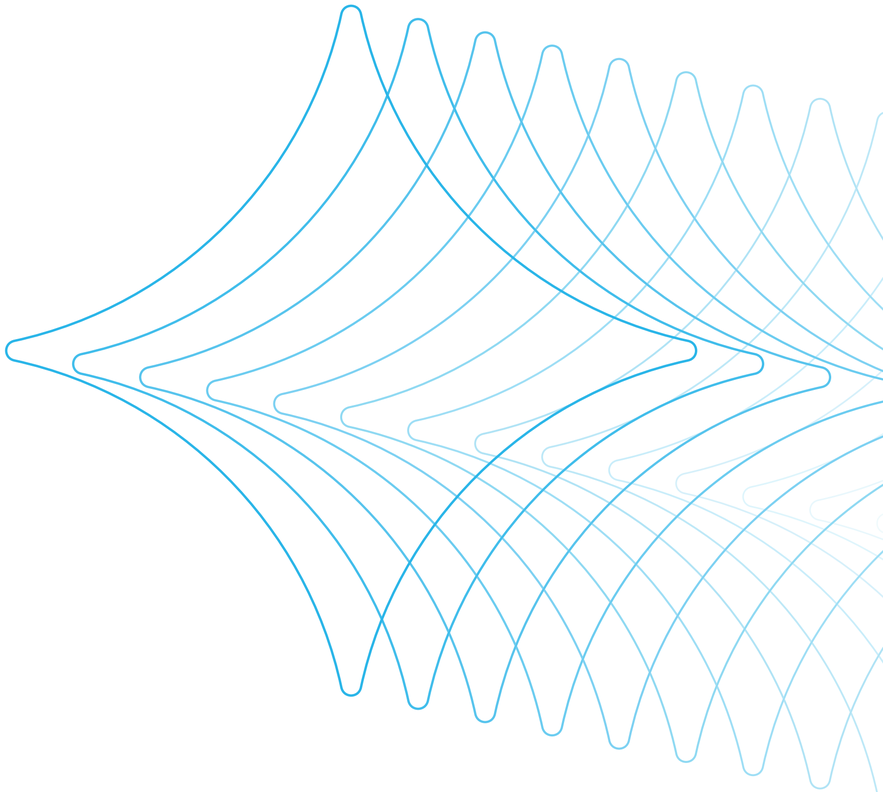
收集和拥有数据当然是一个关键的第一步，但之后呢？当数据被清理、塑形和丰富，使其能为依赖它的个人和系统所用时，数据才变得有用。这便引出了下一阶段：转型。

数据转型，而非代码：转向声明式工作流

原始数据本身价值有限。数据工程师必须利用代码、逻辑和精心编排的步骤，整合并创建高质量、经过筛选的数据。数据转换——即重塑、丰富并准备已摄入数据以供下游使用的过程——是数据工程师花费大部分时间的地方。

数据转换产生的结果形式多样：包含聚合和计算的精选数据集、为下游应用准备的数据记录、机器学习的训练数据等。数据工程师处理数据集的方式很大程度上取决于数据的性质，在某些情况下，也取决于数据转换的环境。

随着人工智能（AI）影响批处理式数据摄取方法并将其转变为持续工作流，数据工程师现在必须实现能够快速执行的数据转换。此外，AI能够处理的大量非结构化数据意味着数据工程师需要新的工具来大规模处理此类数据。在许多情况下，这意味着利用AI本身来转换数据。最后，以SQL优先的数据工程师正越来越频繁地深入到Python（以及其他编程语言）的世界。由于AI涵盖众多工作负载、用例和行业，这意味着用于输入AI的数据可以在各种开发环境中进行处理，包括那些并非以SQL优先的环境。



人工智能对传统模式的影响

转型变革的强制式编排

传统上，数据工程师依赖命令式方法来管理他们的转换。他们明确定义作业应如何运行，包括顺序和时间。工程师们使用 Airflow、cron 作业和任务调度等工具来帮助他们编排转换是很常见的。这是一种常见且经过实战检验的方法，但它将调度和依赖管理的全部负担都压在了数据工程师身上。人工智能降低了手动编排的容忍度。由于数据工程师需要构建更多管道，他们还需要确保数据转换尽可能快地处理。

为处理数据设置独立的计算环境

当数据工程师需要使用 Python 或 Spark 进行转换时，他们无法在运行 SQL 转换的同一环境中运行它们。相反，他们必须启动独立的基础设施——Spark 集群、独立的 Python 环境——并在系统之间移动数据，才能执行转换。这种运营开销意味着数据工程师将时间花在管理基础设施上，而不是从数据中创造价值。借助人工智能，如今管道进入 Python 世界的规范做法是利用所需的库和自定义算法。数据工程师需要一致且灵活的编程环境来满足这些需求。

增量处理与开销

增量处理——仅转换新数据或已更改数据，而非整个表——在大规模应用中至关重要。但实施起来十分复杂。数据工程师必须手动管理变更跟踪、处理迟到数据、考虑模式漂移，并构建逻辑以处理故障恢复场景。任务、流和存储过程等工具有所帮助，但它们增加了运维复杂度。其结果是形成脆弱的管道，需要数据工程师持续关注。为了让AI产生价值，它需要通过健壮的管道来获取数据，这些管道应具备自主、持续和高效的特点。

人工智能领域的新需求

除了AI对传统方法的影响外，AI也给数据转换阶段带来了全新的需求。例如，鉴于目前为AI解决方案提供数据的大部分是非结构化的，数据工程师需要弄清楚如何将情感检测、答案提取和文本摘要等AI能力直接嵌入到他们的转换流程中。AI要求这些能力必须通过SQL函数实现，而不能依赖外部机器学习服务、自定义存储过程或用户定义函数（UDFs）。

解决方案

这是Snowflake如何帮助数据工程师应对传统数据转换模式以及AI引入的新模式所带来的压力。

核心代码

Cortex Code是一款专为Snowflake开发而设计的AI编程助手。与通用的代码补全工具不同，Cortex Code能够理解Snowflake的上下文，例如您的表、您的模式以及您的转换逻辑。它通过帮助数据工程师更快地编写转换、调试问题并生成模板代码，从而加速数据工程工作流，所有这些都在一个单一、灵活的开发环境中完成。

动态表

动态表功能可在Snowflake中实现声明式、持续刷新的数据转换。用户无需编写用于调度和编排作业的命令式逻辑，而是将转换定义为SQL查询，并指定目标延迟时间——Snowflake将负责何时以及如何刷新数据。在底层，动态表利用变更跟踪功能自动执行增量刷新，无需手动处理迟到数据、模式变更或故障恢复。这意味着数据工程师可以专注于数据管道本身，而无需处理繁琐的编排开销。

Snowflake中的dbt项目

对于使用流行开源框架的团队，Snowflake 提供原生支持。通过在 Snowflake 上创建和管理 dbt 项目，您可以获得自助式数据的所有优势，而无需自行托管和维护开源软件的管理负担。数据工程师喜欢 dbt，因为该方法通过 refs 管理依赖关系，让他们可以专注于转换逻辑，而不是手动编排作业。

雪公园

Snowpark 支持在 Snowflake 的计算引擎上原生运行 Python、Java 和 Scala 的转换。对于那些超出 SQL 优势范围（如复杂的算法逻辑、自定义机器学习预处理或与 Python 库的集成）的转换，Snowpark 允许数据工程师编写类似 DataFrame 的代码，这对于 Spark 和 pandas 用户来说感觉很熟悉。其核心优势在于无需单独配置或管理集群，也无需在不同平台或环境之间移动数据。

Snowpark 连接器

Snowpark Connect允许数据工程师在Snowflake上运行现有的Spark工作负载，而无需重写代码。对于在Spark管道方面有重大投资的公司，这提供了一个快速迁移到现代化基础设施的途径。数据团队可以将现有的Spark作业指向Snowflake，并随着时间的推移逐步淘汰遗留集群，而不是面临昂贵的全面重写。再次，数据工程师可以为他们自己的管道利用一个单一、灵活的开发环境。

Cortex AI的功能

雪花脑核心AI功能将AI原生操作直接引入SQL。COMPLETE、CLASSIFY_TEXT、SENTIMENT、EXTRACT_ANSWER和SUMMARIZE等函数可直接在标准SQL查询中调用——无需外部机器学习服务、无需自定义UDF、无需单独的基础设施。这意味着数据工程师可以通过单个函数调用，将情感分析或实体提取添加到转换管道中，使他们能够专注于转换而非外部逻辑。

高效的数据转换能将原始数据转化为有意义的信息，但仅有良好建模、干净的数据本身并不能产生洞见。其价值在于当这些数据到达需要它们的消费者和系统时才得以实现。这便引出了下一阶段：洞见传递。

超越仪表盘：为人类与AI消费者提供语义层

所有数据管道都会输出某种有价值的产品，供下游消费者或系统使用。这些数据产品可以有多种形式，具体取决于使用场景。

例如，几十年来，标准的交付成果通常以数据集市、表格、黄金层和BI仪表盘的形式出现。这些通常有助于满足业务用户（如分析师）查询干净、转换后、有用的数据的需求，或者满足将数据输入到另一个系统进行进一步处理的需求。

人工智能正在改变“交付”的含义。出现了新的消费者，例如代理和应用程序，以及新的交付成果格式，比如语义层。此外，人工智能也改变了谁可以查询下游数据。过去只属于SQL优先数据分析师的权限，现在可以向具有不同背景的人开放，包括那些没有编程语言经验的人。这是因为数据对话界面已成为常态，数据工程师也必须思考如何让数据管道满足这些新的范式。简而言之，数据工程师现在既向人类也向人工智能系统进行交付。

人工智能对传统模式的影响

表格和集市作为主要交付成果

数据管道的一个著名且常见的最终产出长期以来都是经过良好建模的表格、表格集、数据集市或“黄金层”。数据工程师将原始数据转化为维度模型、聚合数据以及其他形式的精选数据集。下游消费者直接或通过专门的BI工具查询这些数据。但人工智能需要的不只是表格。人工智能持续需要更丰富的上下文，通过元数据、数据关系和业务定义来获得。仅有的原始表格无法提供人工智能执行准确操作所需的语义理解。

自然语言作为消费界面

长期以来，下游消费者一直使用大量BI工具查询和从干净的数据中提取洞察。数据工程师长期以来一直为Tableau、Looker和Power BI等工具构建。数据接口传统上是以UI为先的，但人工智能改变了这一点。如今，下游消费者

人们开始期待将易于使用的对话式界面作为他们查询数据的数据层。更重要的是，人工智能已将下游消费者画像扩展至包括非技术人员和数据好奇者。这也导致了对对话式界面的需求更加旺盛，用户可以在其中用自然语言提问，而不是导航仪表板或编写SQL。

仅限人类消费者

数据产品一直以来都是为人类设计的：分析团队、业务用户、高管。这些长期以来一直是默认的用户画像，他们通过可视化或报告来解读数据。但现在，得益于人工智能，代理已加入这个群体，成为了一流消费者。与人类消费者相比，一个查询数据以回答问题的AI代理将会有不同的需求。如今，数据产品必须服务于这两种用户画像，而数据工程师在考虑如何提供丰富上下文时，必须将这一点纳入考量。

人工智能领域的新需求

人工智能也催生了一种新的数据产品：语义层。这些层包含丰富的上下文信息，如业务信息、关系、元数据和定义，帮助人工智能准确、大规模地运行。数据工程师现在必须深入理解语义建模，并懂得如何设计和输出人工智能能够有效应用于结构化、半结构化和非结构化数据的准确语义层。这是在人类是主要下游消费者时并非总是必要的全新要求。

解决方案

雪花引擎提供了一套专为向人类和人工智能消费者交付数据产品而设计的原生功能。

语义观点

语义视图为您的物理表提供了一个对业务友好的层，用于定义指标、维度和关系。数据工程师创建经过策划的定义，使人工智能系统和人类都能使用这些定义来理解数据上下文。这一基础使准确的人工智能驱动查询成为可能——提供了一种语义理解，而原始表则缺乏这种理解。

皮质分析师

Cortex Analyst 可让您以自然语言查询结构化数据。用户使用日常语言提问，Cortex Analyst 则针对您定义的语义模型生成并执行 SQL 查询。这为无法编写 SQL 的非技术人员打开了数据访问途径，直接应对了从基于仪表盘到对话式消费的转变。它也加速了创建上述语义视图的过程。Cortex Analyst 足够智能，能够根据您指定的数据生成一个基准语义视图。数据工程师可以手动继续优化该语义视图，或通过持续使用 AI 来完善。

皮层搜索

Cortex Search 支持对非结构化数据进行自然语言查询。数据工程师可以在包含非结构化数据（如支持记录、产品文档、知识库文章等半结构化或非结构化数据）的表格上创建搜索服务。用户和 AI 系统可以使用自然语言查询这些数据，Cortex Search 则利用混合向量搜索和关键词搜索技术，检索最相关的结果，并将语义上下文扩展到语义视图中的结构化指标和定义之外。

皮质代理

Cortex Agents 是能够利用您的数据进行分析推理、制定计划并采取行动的人工智能代理。它们使用 Cortex Analysts 创建的语义视图等工具，以及 Cortex 搜索服务，来访问数据源并自主执行多步骤工作流。对于数据工程师而言，这意味着构建服务于机器消费者（即代理与人类分析师一同查询数据）的数据产品，以应对这一新现实。

雪花智能

雪花智能是一款面向企业的AI助手，允许业务用户使用自然语言就其数据提问。它结合了Cortex Analyst的文本转SQL能力与企业级更广泛的环境，能够实现无需SQL知识即可自助分析——为非技术人员提供所需的对话式界面。

将洞察力以最佳方式传递给目标消费者是每个数据管道的最终目标。ITD的每个阶段都建立在上一阶段的基础之上，而人工智能现在正在每一步提高标准。从一开始就掌握将人工智能集成到其管道架构中的数据工程师，将是那些从其数据平台中获得真正价值的人。

安全扩展：为高风险AI工作负载实施DataOps

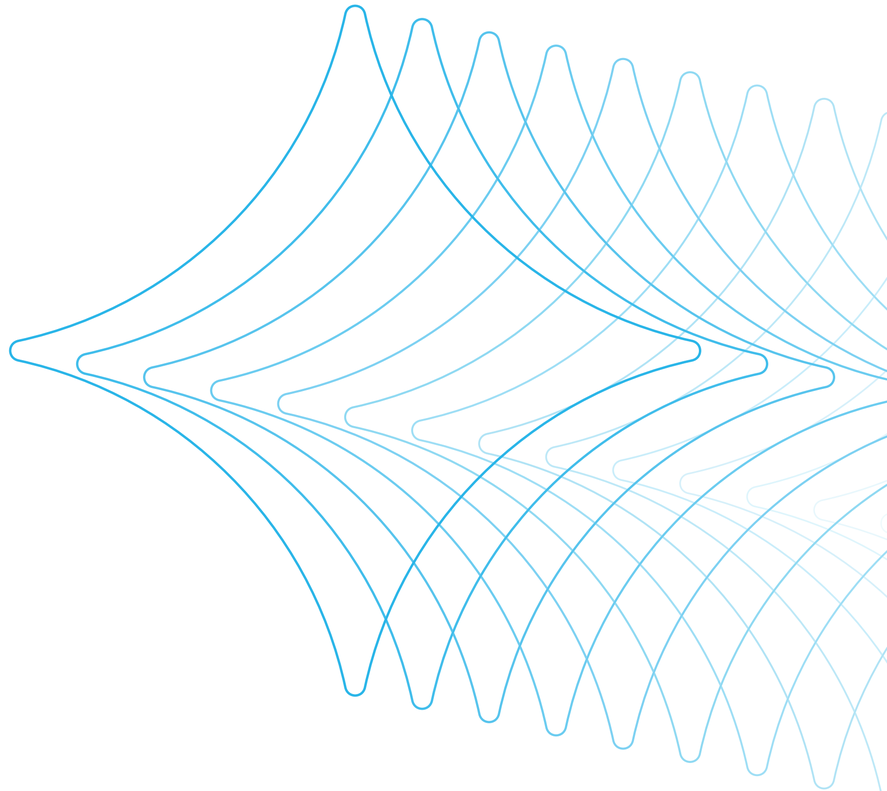
如果你问10位工程师“DevOps是什么”，你可能会得到10个不同的答案。花点时间来定义这个术语，有助于我们更好地理解它如何渗透到数据工程领域。

DevOps并非产品或功能。从宏观角度看，它通常是一套核心的理念和实践方法，旨在让团队能够快速、大规模地交付和维护软件。这些实践包括诸如使用源代码管理来维护代码、针对新代码运行测试、自动化新代码的部署等。这些方法帮助团队在多个开发环境中安全地测试新代码，并且早已成为软件开发领域的基本原则。

日益增多，数据工程师们正借鉴DevOps的最佳实践来构建、部署和维护数据管道。该领域甚至创造了一个类似的术语：DataOps。但究竟是什么原因导致了这种转变？

如同任何软件一样，数据管道是具有一系列关键需求的动态工程系统。例如，管道可能需要快速响应数据库对象和架构的变化。它们应具备可靠性，并尽可能减少停机时间。此外，它们也许会需要足够现代化，以便轻松与新方法集成，例如能够捕获流数据或将第三方工具集成到现有数据环境中。通过采用数据运维（DataOps）最佳实践，数据工程师可以轻松维护其管道，并在需要时快速、安全地对其进行调整。

但事情并未就此结束。人工智能正在加速DataOps的采用。AI工作负载要求更快的迭代和更高的可靠性。从一开始就实施DataOps意味着数据工程师可以更快地推进工作。此外，数据平台与AI工具的结合，使得这些实践更容易实施。现代数据平台，包括Snowflake，原生支持声明式数据管道项目，而像Cortex Code这样的AI编码助手可以帮助编写部署脚本、进行测试，并在管道进入生产环境之前通常能捕捉到错误。总而言之，这正在使DataOps成为人工智能时代现代数据管道的必要条件。



人工智能对传统模式的影响

手册式临时开发

对于数据不成熟的数据组织来说，数据管道通常由一系列脚本和查询构建而成，且没有任何正式的源代码控制或部署流程，这种情况并不少见。事实上，在某些环境中，现有的工具甚至可能不兼容git。管道的代码可能存储在本地文件、共享文件夹中，甚至直接存放在生产环境中。人工智能只会增加管道开发的数量和速度，而临时性的方法无法扩展。对于数据工程师而言，这意味着从一开始就要实施版本控制（以及与版本控制兼容的工具）。

管道数据环境

多年来，数据工程师直接在生产环境中测试新的管道代码，或者根本没有（或极少）访问单独数据环境的权限。这使得向管道代码引入新变更变得极其冒险。稍有不慎就可能导致整个系统崩溃。作为变通方法，他们可能会启动“备份”或“预发布”或“测试”数据库，但这些数据库可能无法保持高度的数据保真度。行业正向我们展示，AI工作负载风险极高，且可能迅速变得代价高昂。在生产环境中测试代码过于危险，不可尝试。因此，数据工程师正大力投入创建分离的数据环境（想想：开发、预发布、生产），并使用真实数据在将这些变更推向生产环境之前进行验证。

手动部署管道

对于能够访问独立环境的幸运数据工程师来说，将变更部署到生产环境意味着手动运行脚本或将文件在服务器之间移动。传统上，自动化程度很低，回滚能力有限（如果有的话），并且通常没有审计追踪。AI工作负载要求大规模的持续交付，但手动部署既缓慢又容易出错。如今，数据工程师必须为其管道构建持续集成/持续交付（CI/CD）自动化，以便管道能够自动进行测试、验证并部署到生产环境，同时具备完整的审计追踪和回滚能力。

人工智能领域的新需求

数据工程师不仅为人工智能构建数据管道，他们也能从编码代理带来的开发效率中获益。这些代理可以生成模板代码、建议查询优化、在测试前捕获错误，并通常加速端到端管道开发工作流程。这本质上是一项新能力，数据工程师已不再论其管道架构或数据平台如何，都期待这种协助。

解决方案

雪花地址通过原生支持版本控制、CI/CD、环境管理和AI辅助开发来解决这些转变。

核心代码

Cortex Code是一款专为Snowflake开发而设计的AI编程助手。与通用的代码补全工具不同，Cortex Code能够理解Snowflake的上下文——你的表、模式和转换逻辑。它可以帮助编写部署脚本、生成测试、调试问题并在它们进入生产环境之前捕获错误。这直接应对了AI辅助开发的新需求，帮助团队在不受质量影响的情况下更快地推进工作。

雪花DCM项目

DCM（数据库变更管理）项目支持以声明式方法将Snowflake对象定义为代码，使数据团队能够轻松实现DataOps的一项关键原则。数据工程师可以定义其数据库、模式、表、动态表以及其他许多数据环境对象所需的状态，而Snowflake将确定并应用对这些对象所做的任何变更。此外，DCM项目使团队能够首先规划其数据环境的变更，然后将这些变更部署到目标环境，例如测试环境或生产环境。这取代了过时、手动且易出错的部署流程，这些流程无法扩展以支持AI工作负载。

Git 集成

雪花云的原生 Git 集成支持版本控制的开发工作流。雪花云工作区提供了一个基于 Git 的开发环境，为数据工程师团队提供了一个共享的、可追溯的源代码工作区。代码不再存储在本地文件或共享文件夹中，而是在协作环境中集中管理，所有变更均可被追踪、审查和审计。变更可以在分支上创建并推送到 GitHub，结合雪花云 CLI 触发 CI/CD 工作流。这直接解决了从临时、未追踪的开发转向可自动部署的版本控制流程的转变。

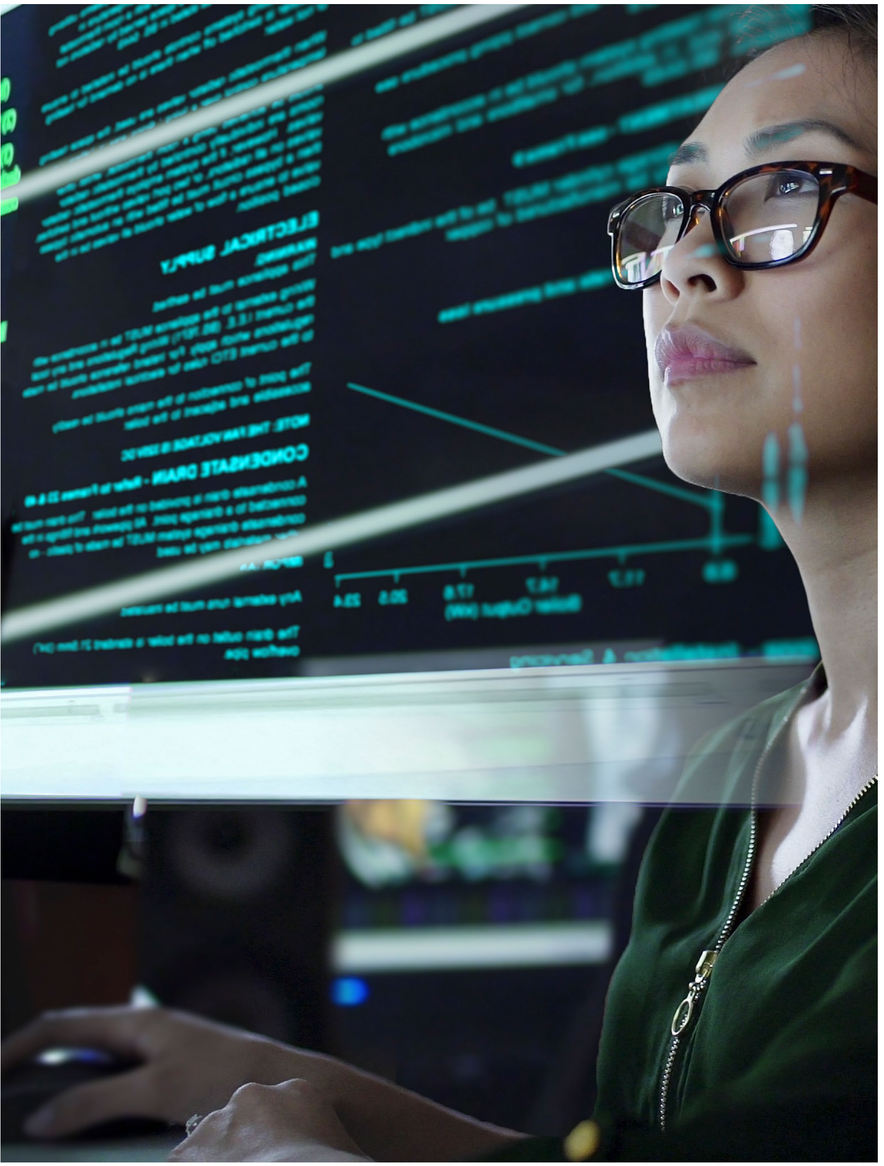
雪花CLI

雪花CLI是一个用于自动化雪花操作命令行界面。您可以通过脚本进行部署、运行测试以及以编程方式管理对象。它与GitHub Actions、GitLab CI和Jenkins等CI/CD工具集成，能够实现完全自动化的部署流程，用可重复、可审计的工作流替代手动、易出错的流程。

使用 dbt 进行测试

对于使用 dbt 的团队，Snowflake 支持原生的 dbt 执行，并内置了测试功能。您可以定义用于唯一性、引用完整性、接受值和自定义逻辑的测试。测试会在构建过程中自动运行，从而在部署前实现自动化验证，并在数据问题到达生产环境之前就捕获它们。

管道的每个阶段——数据摄取、转换和交付——都受益于DataOps所带来的纪律性和结构性。随着AI增加管道的容量、复杂度以及迭代速度，可扩展性和维护能力成为团队的关键差异点。从一开始就投资于DataOps最佳实践的数据工程师，为他们的管道能否在AI时代保持竞争力奠定了基础。



数据工程师的未来：AI时代中的运维架构师

所有数据管道都在被人工智能重塑，无论数据来自何方、如何被转换，或是它们所交付的数据产品类型。人工智能并非使这些常见阶段失效——它只是要求更高，这意味着数据工程师必须立即改进他们的工作流程。

数据工程师不再仅仅是管道构建者。他们必须考虑如何为人工智能提供数据，服务可能作为代理或属于代理系统的消费者，提供语义层，并以前所未有的速度实现运营化。

这些挑战始终存在，是当前人工智能时代许多数据工程师的现实。像Snowflake这样的现代数据平台正帮助他们缩小差距。那些能够适应将人工智能整合到其管道架构中，并融入可扩展性和可维护性的最佳实践的数据工程师，将引领潮流。

要了解更多并亲自尝试这些工具，请探索用于数据工程的 Snowflake，并学习如何使用本书中提到的工具构建更好的管道。此外，请查看 DeepLearning.AI 上的免费在线课程“构建多模态数据管道”，以了解如何构建结合图像、音频和视频数据信息的 AI 应用。

准备好更深入探索时，请查阅 Snowflake 最新数据工程开发者指南，并立即开始免费试用。

雪花（Snowflake）是人工智能时代的平台，助力企业加速创新并从数据中获取更多价值。全球超过13,300家企业，包括数百家世界顶级公司，都在使用雪花AI数据云（Snowflake AI Data Cloud）来构建、使用和共享数据、应用程序及人工智能。有了雪花，数据与人工智能将赋能每一个人。

在 snowflake.com 了解更多

（雪球：SNOW）