Databricks Assistant

Context-aware AI assistant

What is Databricks Assistant?

- Databricks Assistant is an AI-powered, context-aware pair-programmer and support agent built into the Databricks platform.

- It helps data professionals by generating code (Python/SQL), explaining complex queries, fixing errors, and automating tasks in notebooks, SQL editors, and dashboards, using Unity Catalog metadata for personalized, accurate assistance to boost productivity.

- It understands your data, code, and workflow, acting as an intelligent pair programmer and support agent that reduces development time from hours to minutes.

Key Capabilities

- **Code Generation**
  - Creates SQL, Python, and PySpark code from natural language prompts, even converting between languages (e.g., Pandas to PySpark).
- **Code Explanation & Fixing**
  - Explains complex code and identifies/fixes errors with suggested code (Quick Fix).
- **Context-Awareness**
  - Leverages Unity Catalog metadata (tables, schemas, comments, lineage) and notebook context for relevant suggestions.
- **Productivity Modes**
  - Offers Chat, Edit (inline suggestions), and Agent modes (automating multi-step tasks).
- **Data Discovery**
  - Helps find and understand data assets within your workspace.
- **Integrated Experience**
  - Available across notebooks, SQL editor, file editor with options to dock or move the pane.

How Databricks Assistant Works?

- You interact via a conversational interface, asking questions or using slash commands (e.g., /settings).
- It uses the Databricks Intelligence Engine, pulling signals from your environment for personalized help.
- It can run code on serverless compute and respects governance and permissions.

How to Get Started?

- **Enablement**
  - Workspace admins can enable the Assistant in the Account Console under Settings > Feature Enablement.
- **Access**
  - Click the Assistant icon on the top-right of any Databricks screen.
- **Customization**
  - You can add Custom Instructions (User or Workspace level) to define preferred coding conventions or role-specific guidance

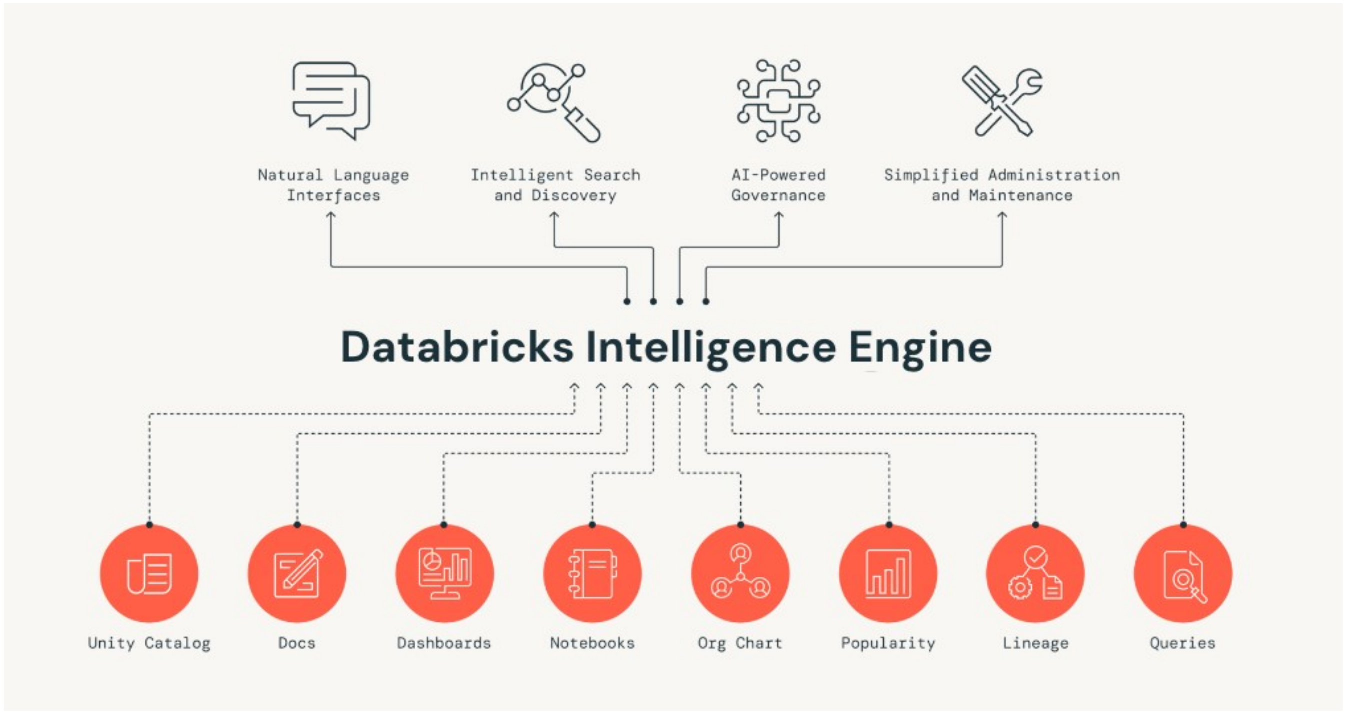
Core Architectural Components

- **Databricks Intelligence Engine**
  - This is the "brain" that provides the Assistant with knowledge of popular tables, schemas, and usage patterns across the entire company.
- **Unity Catalog Integration**
  - The Assistant uses Unity Catalog metadata—including column descriptions, tags, and comments—to ground its responses in the specific data assets of your organization.
- **Contextual Signals**
  - It continuously monitors real-time signals from the user's workspace, such as current notebook code cells, libraries, variables, and previous conversation history.

Data Intelligence Engine

Data Intelligence Engine (a.k.a **DatabricksIQ**) powers the entire Databricks Data Intelligence Platform.

Its the core AI-driven engine that learns from your organization's unique data, usage patterns, and business context to automate and simplify tasks


Natural Language Interfaces
Intelligent Search and Discovery
AI-Powered Governance
Simplified Administration and Maintenance
Databricks Intelligence Engine
Unity Catalog
Docs
Dashboards
Notebooks
Org Chart
Popularity
Lineage
Queries

Security and Governance

- Databricks Data Intelligence Engine is fully integrated with Unity Catalog, ensuring that AI-driven insights follow the organization's existing security and governance rules.

- Users only receive answers and see data that they are explicitly authorized to access

Slash Commands

Databricks Assistant uses slash commands to quickly perform common development tasks. You can trigger these by typing / in the Assistant chat pane or using the inline assistant (Cmd+I / Ctrl+I).

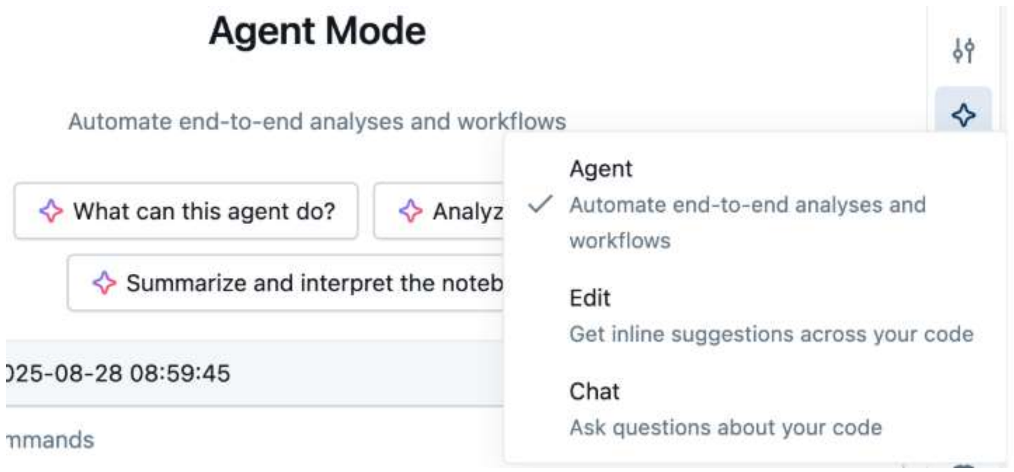
- **/**: Displays the menu of all available common commands.
- **/fix**: Analyzes the code in the current cell for errors and proposes a corrected version.
- **/explain**: Provides a natural language explanation of the logic within a code cell.
- **/optimize**: Evaluates and refactors SQL or Python code to improve performance.
- **/doc**: Automatically adds comments and documentation (docstrings) to the code.
- **/prettify**: Formats the selected code for better readability and style compliance.
- **/findTables**: Searches for relevant tables using Unity Catalog metadata.

https://docs.databricks.com/aws/en/notebooks/code-assistant#use-slash-commands-for-prompts

Interaction modes

Databricks Assistant offers four primary interaction modes designed for different stages of the data development lifecycle. These are accessible via a mode selector at the bottom of the Assistant pane or through specific keyboard shortcuts.

- **Chat Mode (Conversational)**
- **Edit Mode (Notebook-wide Refactoring)**
- **Agent Mode (Autonomous Execution)**
- **Inline Assistant (Cell-level Edits)**


Agent Mode

Automate end-to-end analyses and workflows

✓ Agent
Automate end-to-end analyses and workflows

Edit
Get inline suggestions across your code

Chat
Ask questions about your code

Chat mode

The default interface for general assistance and rapid prototyping.

- **Best for**: Asking conceptual questions, getting code snippets, or searching documentation.

- **Key Feature**: Provides responses with citations from Databricks documentation and generates code (Python/SQL) that you can copy or insert directly into your notebook.

- **Context**: Uses the current notebook cell or SQL editor tab as context to personalize answers.

Edit mode

A specialized mode for applying changes across multiple cells simultaneously with a single prompt.

- **Best for**: Large-scale tasks like renaming variables throughout a notebook, migrating Pandas code to PySpark, or standardizing code styles.

- **Key Feature**: Surfaces suggestions as inline diffs in the affected cells. You can review, accept, or reject each edit individually, or use "Accept All" to apply every change at once.

Agent mode

A (beta) feature known as the Data Science Agent that transforms the Assistant into an active collaborator.

- **Best for**: Automating multi-step workflows like full exploratory data analysis (EDA), training ML models, or building entire data pipelines.
- **Key Feature**: Unlike other modes, Agent Mode can plan steps, run code on serverless compute, analyze cell outputs, and automatically fix errors it encounters during execution. It asks for your confirmation before running code for security.

Inline Assistant

Triggered directly within a notebook code cell using Cmd+I (Mac) or Ctrl+I (Windows).

- **Best for**: Quick, focused edits within a specific block of code without opening the side panel.
- **Key Feature**: Opens a small text box inside the cell where you can type instructions like "add a bar chart for this data" or "fix the syntax error here".

Thank You