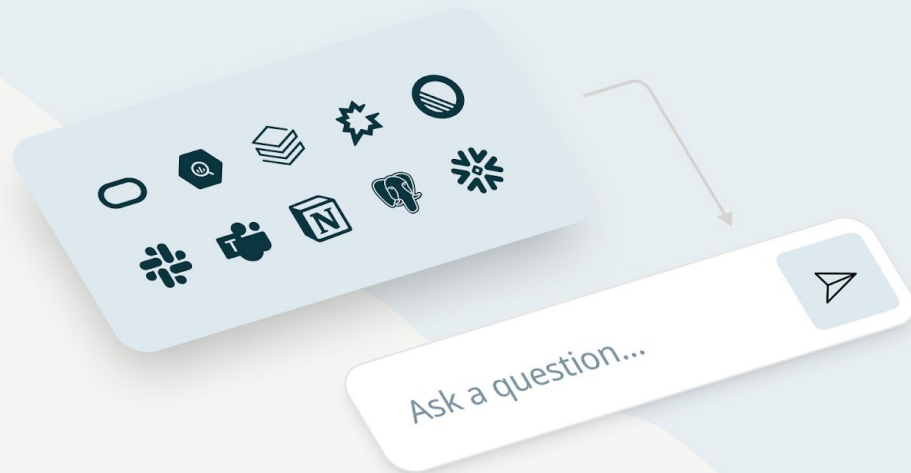


Connect. Unify. Respond.

Any data, anywhere with
human level intelligence.





Most widely-adopted AI
query engine in the world



400K Deployments

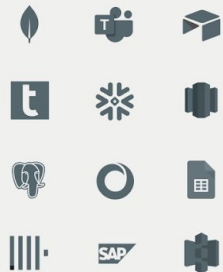


30K GitHub Stars



200+ Enterprise Datasources

Zero-ETL Support for all major
large-scale enterprise datasources



Industry-leading tech partners



Google



Microsoft

ORACLE



Backed by top investors



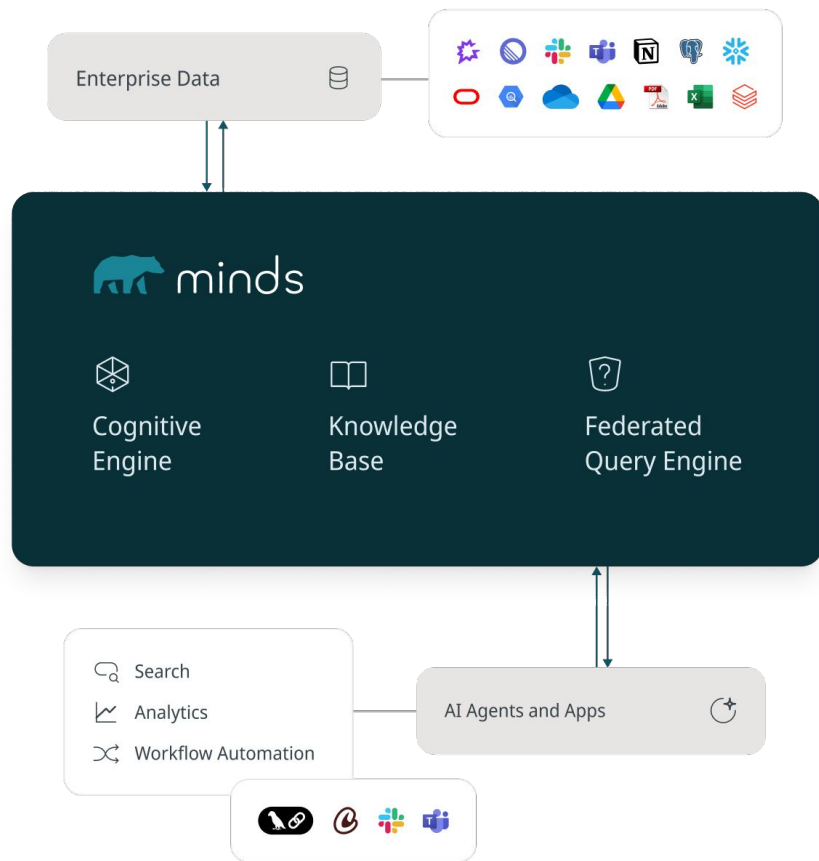
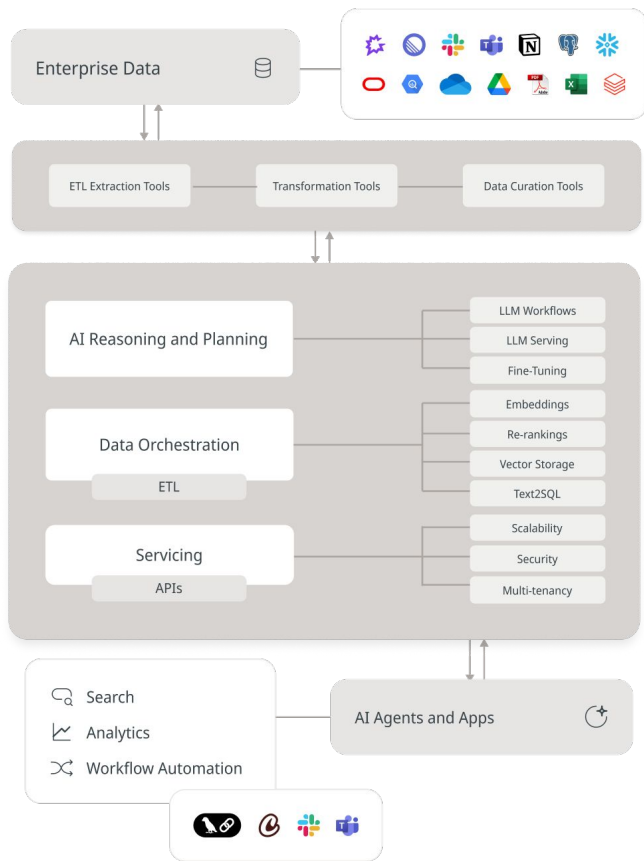
BENCHMARK

Mayfield

Y Combinator



MindsDB Simplifies and Derisks AI Implementations





Why MindsDB?

Most accurate solution in the market, paired with unique capabilities



Zero ETL Zero Copy

Leave your data where it lies, no need for pipelines.



Run inside Your Trust Boundary

Managed Cloud, VPC,
OnPrem, even air-gapped.
Run where you need.



Industry Leading Accuracy

97%+ on industry
benchmarks.



Actionable Answers, ASAP

Ask questions of your live
data within 24 hours of
deploying.



The MindsDB Stack

A Seamless Opportunity for Maximum AI Value

Federated Query Engine (Connect)

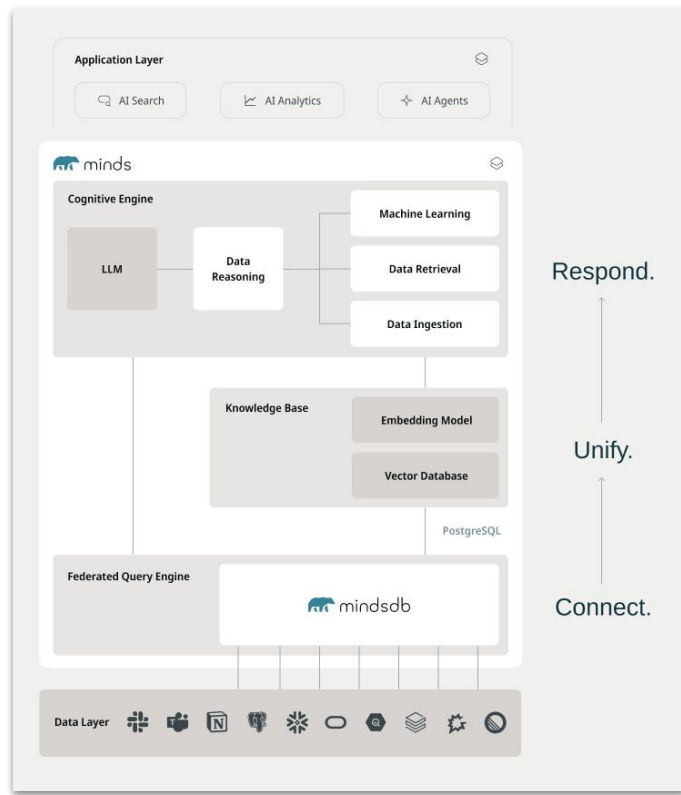
- Retrieves accurate, up-to-date knowledge
- Translates from SQL to any query language

Knowledge Base (Unify)

- Makes sense of structured & unstructured data
- Avoids cost and complexity of moving data

Cognitive Engine (Respond)

- Supports leading proprietary and OSS LLMs
- Orchestrates w/reasoning and planning to surface the best knowledge





Knowledge Base - Features

Unified SQL Interface

Manage and query AI systems with standard SQL commands like **CREATE**, **INSERT**, and **SELECT**, reducing the learning curve for developers.

Intelligent Semantic Search

Searches based on conceptual meaning, using embedding and reranking models to find the most relevant results.

Flexible Model & Storage Integration

Natively connects to major LLM providers (OpenAI, Azure, Bedrock) and vector databases, offering maximum flexibility .



Knowledge Base - Features

Powerful Hybrid Queries

Combine semantic search and keyword search with structured metadata filters in a single query to pinpoint the most accurate information.

Automated Data Management

Automates complex data handling, including text chunking, embedding, and updates, to streamline the entire data ingestion process.

Built-in Performance Evaluation

Use a dedicated **EVALUATE** command to measure the accuracy and relevance of your knowledge base, ensuring optimal performance.



Introducing FAISS Integration in v25.14

In-Memory Performance

Designed specifically for vector operations, FAISS keeps indexes in RAM to deliver microsecond-level query speeds. This in-memory architecture is optimized for similarity search at scale, ensuring fast performance regardless of dataset size

Scale Without Compromise

Designed to handle millions of vectors efficiently, FAISS maintains consistently low latency at scale. Whether you're querying 1,000 or 2 million documents, query performance remains constant.

Production-Ready

FAISS is battle-tested by Meta AI Research and used by industry leaders for billion-scale search. MindsDB brings this proven technology to your database with simple SQL—no complex infrastructure required

Create a Knowledge Base

Example



CREATE FAISS Handler

First step: Create a FAISS database connection that will store your vector indexes on disk and load them into memory for fast queries

```
CREATE DATABASE db_faiss
WITH
  ENGINE = 'duckdb_faiss',
  PARAMETERS =
    {"persist_directory":"/home/ubuntu/faiss"};
```



CREATE KNOWLEDGE BASE

Open AI

embedding_model1: Specifies the model used to convert your text data into vector representations (embeddings). This allows for semantic searching (finding content based on meaning, not just keywords).

```
CREATE KNOWLEDGE_BASE kb_stack_faiss
USING
    storage=db_faiss.stack,
    embedding_model = {
        "provider": "openai",
        "model_name" : "text-embedding-3-small"
    },
    content_columns=['Body'],
    metadata_columns=[
        "PostTypeId",
        "AcceptedAnswerId",
        "LastActivityDate",
        "LastEditDate",
        "LastEditorUserId",
        "OwnerUserId",
        "Tags"
    ]
```

Query a Knowledge Base

Example



SELECT * FROM my_kb

Search at Document level

Action: Performs semantic search on `<kb_name>` using `WHERE content = '<query_text>'`.

Returns: Document `id` and `relevance` score.

Optional Filtering: Use `relevance` (0-1) to set minimum relevance.

Optional Limit: Use `LIMIT <N>` to restrict the number of results.

```
SELECT *  
FROM <kb_name>  
WHERE content = '<query_text>'  
AND relevance >= 0.6 -- optional  
LIMIT <N>;
```

Create an Agent

Example



Create Agent

Model: which language model or AI engine to use. It includes parameters like **provider** (e.g., Google), **model_name** (e.g., gemini), and authentication details like **api_key**.

Data: specifies the data sources connected to your agent or knowledge base, including **tables** from databases and **knowledge_bases** for semantic search capabilities.

Prompt Template: contains instructions that guide the AI model's behavior, including tone, response format, and how to interpret the connected data sources.

```
CREATE AGENT my_agent
USING
    model = {
        "provider": "google",
        "model_name" : "gemini-2.5 flash",
        "api_key": "....."
    },
    data = {
        "knowledge_bases": ["mindsdb.sales_kb"],
        "tables": ["postgres_conn.customers"]
    },
    prompt_template='
        mindsdb.sales_kb stores sales analytics data';
```

Chat with your Data

Example



Query an Agent

Search By keyword

To query an agent in MindsDB, use the **SELECT FROM AGENT** syntax:

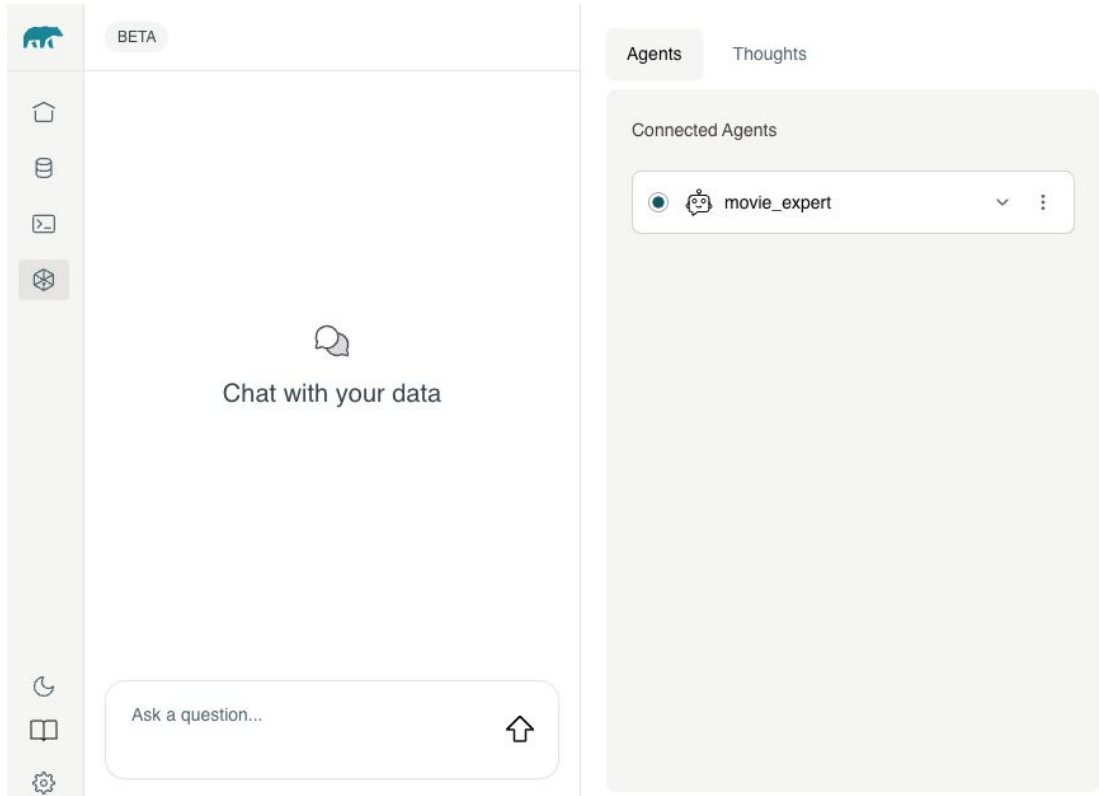
```
SELECT answer
FROM my_agent
WHERE question = 'What is the average number of orders per
customers?';
```



Chat with Your Data

Respond

Chat with your data directly on the
'Respond tab' of the OS.



Demo



Demo Setup

Data Source

Stack Overflow Dataset

- 2 Million Q&A posts
- Real-world unstructured data
- Rich metadata (tags, scores, dates)
- ~5GB total size

MindsDB Instance

AWS EC2 Configuration

- CPU: 8 cores
- RAM: 32 GB
- Storage: 100 GB SSD
- Version: MindsDB v25.14 (released today!)

Embedded in Single Instance:

- PostgreSQL (for pgvector)
- MindsDB engine

Two Knowledge Bases

Knowledge Base #1: pgvector

- Storage: PostgreSQL + pgvector extension
- Architecture: Disk-based

Knowledge Base #2: FAISS

- Storage: DuckDB + FAISS indexes
- Architecture: In-memory indexes