

Apache Doris Roadmap 2026

Scale Intelligence, Accelerate Insights

2025 Community Progress

Two landmark releases in 2025 — from Lakehouse deepening to AI-native capabilities

v3.1

2025 H1

Deepening Lakehouse & Semi-Structured Analytics

VARIANT Semi-Structured Upgrade

- Sparse columns: 10K+ sub-columns support
- Schema Template for stable indexing & cost control

Inverted Index & Full-Text Search

- Index V3: up to 20% storage reduction
- 3 new tokenizers: ICU, IK, Basic + custom tokenizer

Lakehouse(Data warehousing)

- MV fully supports data lakes; Iceberg/Paimon upgrades
- Flexible Column Updates; MOW lock optimization
- Data Traits: up to 10x perf gain in specific workloads

v4.0

2025 H2

One Engine for Analytics, Full-Text & Vector Search

AI Capabilities: Vector Search + AI Functions

- Built-in vector index — no external vector DB needed
- AI Functions: call LLMs via SQL for NLP tasks
- HSAP: vector + full-text + analytics in one SQL

Better Full-Text Search

- New SEARCH() function: ES-like Query String DSL
- Text search scoring; better tokenization

ETL/ELT & Performance

- Spill Disk for Join/Agg/Sort/CTE operators
- TopN lazy materialization: dozens of times faster
- SQL Cache default on: 100x parsing efficiency

2026 New Demands on Data Infrastructure in the AI Era

How AI is reshaping the data landscape, and where Doris fits in

Built for the AI Era

Semi-Structured Data Explosion

Variant data type enables schema-free ingestion + columnar performance for dynamic business logs & AI data

AI Observability

Unified real-time storage for LLM Traces, Logs & Metrics — built-in inverted index for instant analysis

Higher Concurrency & Real-time

High-performance vectorized engine handles Agent-driven analytical query bursts at millisecond latency

Converged Analytics Platform

One engine for Observability, Log Analytics & Real-time Dashboards — replacing fragmented data stacks

Core of the AI Data Stack

Knowledge Store for AI (RAG)

Hybrid Search (vector + full-text) builds the knowledge base for LLMs — precise context augmentation in one SQL

Multimodal Data Management

Unified metadata & sample management for text, image, audio, video training data — powering ML pipelines

AI-Native Query Interface

MCP protocol + Semantic Layer enable AI Agents to interact with data deterministically via natural language

Unified Analytics Engine

One engine for AI Observability, Log Analytics, Real-time Dashboards — no more fragmented data stacks

Apache Doris in 2026 — AI Data Infrastructure

For Multimodal Data & Agentic Interaction



01 Semi-Structured Data Analytics

Schema-free ingestion for logs & AI observability at scale with Variant data type



02 Hybrid Search for RAG

Vector + full-text search in one engine, scaling to 10B vectors per table



03 Agentic Analytics

MCP protocol & Semantic Layer let AI agents command data autonomously



04 Multimodal Data & AI SQL

SQL-native AI on unstructured data, plus multi-modal dataset management

2026 Roadmap Deep Dive

- 4 AI-Powered Scenarios
- Platform Foundation

Scenario 1: Semi-Structured Data Analytics

From Logs to AI Observability — Schema-Free at Scale

Pain Points

- ELK cost & rigid schema for business logs
- LLM latency/cost tracking, complex trace chains

How Doris Solves It

Variant data type

— schema-free + columnar performance

Unified storage for Traces/Logs/Metrics

(Langfuse / OpenTelemetry integration)

2026 Roadmap Highlights

Nested Variant support

Optimized sparse column storage

Optimized string compression

Partial update for variant sub-fields

Support 10K columns per file

Scenario 2: RAG / Knowledge Base

Semantic + Keyword — Dual Precision Hybrid Search

Pain Point

Pure vector search misses proper nouns (model names, person names); pure text search misses deep context.

Solution: Hybrid Search

Full-Text Search + Vector Search in one engine, one SQL query — keyword precision meets semantic understanding.

2026 Roadmap Highlights

Disk-based ANN → 10B vectors per table

Vector index in merge-on-write engine

Query string & Boolean query expressions

Multi-index per column

Index-only scan for vector index

Optimized compaction for vector index

Scoring + global lazy materialization

Vector & text index on Iceberg tables

Scenario 3: Multimodal Data & AI SQL

SQL is AI — Manage, Analyze & Train on Multimodal Data

Pain Points

- Analyzing unstructured data requires Python + expensive AI teams
- ML training data (audio, video, text, labels) hard to retrieve & manage

Solution

AI SQL — call LLM via standard SQL for classification, sentiment analysis, summarization

Unified metadata & sample management for multi-modal training data

2026 Roadmap Highlights

Python UDF support

File data type

Scenario 4: Agentic Analytics

Give AI the Command of Data

Pain Point

Text-to-SQL fails on complex schemas. AI agents can't "understand" database tables — they need a bridge between natural language and physical data structures.

Solution: MCP + Semantic Layer

MCP protocol as the communication standard, Semantic Layer as the "translator" between AI models and physical tables — enabling deterministic, verifiable data interaction.

2026 Roadmap Highlights

Open Metadata API

Unified permission for Iceberg REST Catalog

Platform Foundation

Query Engine | Storage | Data Lake | Vibe Coding

Query Engine — From Fast to Complete

New SQL Powers | Full-Pipeline Acceleration | Large Query Stability

Run Any Query

Complete critical SQL gaps for more analytics scenarios

- **ASOF Join** — time-series nearest match
- **Recursive CTE** — hierarchical data traversal
- **UNNEST** — array/JSON explode in SQL
- **MERGE INTO** — one SQL for CDC/upsert sync

10x Faster

Scan → Filter → Join → Expression, every stage accelerated

- **Condition Cache** — skip repeated filters
- **Zonemap Expression** — skip data blocks
- **Column Pruning** — for complex types
- Partition pruning, broadcast join, CASE/LIKE, short-circuit evaluation

Run Big, Run Stable

16 GB memory handles TPC-DS 10 TB — no OOM

- **Spill-to-Disk** — graceful overflow to disk
- **Global Buffer Mgmt** — fair sharing across queries
- **Progress Bar** — real-time query progress

Storage — Scale, Cache, Elasticity

Wider Tables | Smart Caching | Cloud-Native Elasticity

Scale Without Limits

Handle wider tables, bigger data, and auto-managed lifecycle

- **10K Columns** — wide tables for logs & Variant
- **100GB+ Tablets** — less sharding overhead
- **MOW Ingest Opt.** — faster real-time updates
- **Auto Lifecycle** — TTL, sparse column, compression

Smart Caching

Near-local speed on remote storage via intelligent caching

- **Cross-group Preheat** — warm data on new nodes
- **Distributed Cache** — cross-group data sharing
- **Cache Policy** — block/allow list control
- Diskless optimization, granular cache stats SQL

True Elastic Cloud

Scale on demand, isolate workloads, pay only for what you use

- **Elastic Scheduling** — seconds to scale up/down
- **Read-Write Sep.** — isolated read/write paths
- **Persistent Meta Cache** — instant restart recovery

Data Lake — Read, Write, Govern

Native Lake Perf | Full Iceberg/Paimon Support | Unified Governance

Query the Lake Fast

Query Iceberg/Paimon at near-internal-table speed

- **Parquet Page Cache** — cut Parquet I/O by 50%+
- **Data Cache by Default** — zero-config for lake queries
- **Condition Cache** — reuse filters on lake tables
- **Distributed Planning** — parallel scan across nodes

Open Ecosystem

Read, write, and manage Iceberg/Paimon as native tables

- **Iceberg/Paimon Table** — full DML support
- **Arrow Flight Catalog** — high-speed data exchange
- **Fluss Integration** — streaming lakehouse

Unified Governance

One permission model across internal + external tables

- **REST Catalog Perms** — unified catalog access
- **Third-party Auth** — Kerberos, Ranger, etc.
- **Open Metadata API** — share metadata externally

Codebase for AI Agent

Making Doris Agent-Friendly for the New Development Era

Goal: Reshape the codebase so AI agents can efficiently contribute to Doris development

CI/CD Improvements

- Faster build cycles for agent feedback
- Hermetic build for reproducibility
- Refactor third-party builds to CMake

Code Modularization

- Cleaner module boundaries
- Standardized interfaces
- Reduce cross-module side effects

Developer Experience

- Better docs & code annotations for AI
- Structured test scaffolding
- IAM role-based auth (multi-cloud)

2026 Roadmap — At a Glance

4 AI Scenarios | Platform Foundation | Vibe Coding

AI Capability Scenarios

Semi-Structured Data Analytics

From Logs to AI Observability

- Nested Variant type
- 10K columns per file
- Partial update for Variant
- Sparse column storage

Hybrid Search & RAG

Semantic + Keyword Precision

- Disk ANN, 10B vectors
- Index-only scan
- Scoring + lazy materialization
- Vector index on Iceberg

Multimodal Data & AI SQL

SQL is AI — Analyze & Train

- Python UDF
- File data type
- LLM calls via SQL
- Training data management

Agentic Analytics

Give AI Command of Data

- MCP protocol standard
- Semantic Layer bridge
- Open Metadata API
- REST Catalog permissions

Platform Foundation

Query Engine

From Fast to Complete

- TPC-DS 10TB in 16GB RAM
- ASOF Join / Recursive CTE
- MERGE INTO / UNNEST
- Spill-to-Disk / Progress Bar

Storage

Scale, Cache, Elasticity

- 100GB+ tablets, 10K columns
- Elastic scheduling (seconds)
- Distributed cache sharing
- Auto lifecycle (TTL)

Data Lake

Read, Write, Govern

- Parquet Page Cache 50%+
- Iceberg/Paimon full DML
- Arrow Flight Catalog
- Incremental MV refresh

Vibe Coding Infrastructure

CI/CD

- Faster build cycles
- Hermetic build
- CMake third-party

Modularization

- Cleaner boundaries
- Standard interfaces
- Reduce side effects

Developer Exp.

- AI code annotations
- Test scaffolding
- IAM multi-cloud auth

Community Partners

Guest presentations from our partner ecosystem

Incremental Computation & Materialized Views

Building a Streaming Warehouse — Together with the Community

Foundation

Binlog for Internal Tables

Generate row-level change logs (Insert / Update / Delete) for Doris internal tables — the data source for incremental computation

Binlog for Lake Tables

Read change logs from Iceberg / Paimon tables — bridging the incremental data path across Lakehouse

Incremental Framework

Table Stream

Capture DML changes on tables as consumable change data streams — enabling CDC, dual-stream join & incremental aggregation

Dynamic Table

Declaratively define incremental computation logic — the system automatically tracks upstream changes and refreshes results incrementally

Binlog (Internal + Lake) → Table Stream (Change Data) → Dynamic Table (Incremental Compute)
MV Rewrite (Auto Acceleration)

Dynamic Table × MV Transparent Rewrite = Real-time Data Freshness

Combine Dynamic Table with MV transparent rewrite — queries automatically route to the latest pre-computed results, delivering real-time analytics without user-side complexity.

THANK YOU

Apache Doris Roadmap 2026

GitHub: github.com/apache/doris

Roadmap Issue: #60036

Join the community and contribute!