PhoenixAI vs. ClickHouse

Primary Champion: Data Engineer · Closest Competitor: clickhouse.com · Target Industry: Any · Company URL: https://www.phoenixdata.ai · Generated: October 25, 2026

Draft positioning — a hypothesis to test with real buyers before writing copy.

Sources: Using 6 generated reports.

Positioning Summary

Market category: Real-Time Analytical Database (Real-Time OLAP) (Existing category)

Positioning statement: For data engineers who need to serve AI agents, customer-facing analytics, and BI dashboards with sub-second queries on fresh streaming and lakehouse data at scale, PhoenixAI is a real-time analytical database that runs distributed MPP joins on normalized data, commits streaming upserts queryable in under 5 seconds without latency degradation, and isolates agent, BI, and ETL workloads on the same data with independent scaling. Unlike ClickHouse, it handles complex multi-table joins and unpredictable AI-generated SQL without denormalization workarounds, the FINAL tax, or cluster sprawl.

PhoenixAI is a real-time analytical database for data engineers and data platform teams at mid-to-large companies building AI agents, customer-facing analytics, and real-time BI dashboards. It competes against ClickHouse, Apache Pinot, Apache Druid, and the status quo (keeping ClickHouse running or doing nothing). PhoenixAI differentiates by running distributed MPP joins on normalized data without precomputation, committing streaming upserts from Kafka/Flink CDC queryable in under 5 seconds without the FINAL tax, unifying streaming and lakehouse data (Iceberg, Delta, Hudi) in one SQL query without copying data, isolating agent/BI/ETL workloads on the same data with independent scaling, and automatically handling unpredictable AI-generated SQL through its cost-based optimizer. It sits in the real-time OLAP category alongside ClickHouse and Pinot, positioned as the engine that solves their join limitations, merge-on-read latency, and cluster sprawl at production scale.

STAGE 0

Input Audit

PhoenixAI

What it is: A real-time analytical database built on the StarRocks open-source engine that runs sub-second SQL queries across streaming data (Kafka, Flink CDC) and lakehouse tables (Iceberg, Delta, Hudi) with complex joins and workload isolation.

Who it serves: Data engineers, data platform teams, senior engineering managers (data platform), and organizations building AI agents, customer-facing analytics, and real-time BI dashboards at scale.

Key features

- Distributed MPP joins on normalized data without precomputation or denormalization
- Sub-second freshness: streaming upserts/deletes from Kafka and Flink CDC queryable in <5 seconds at 100K+ events/sec
- Unified SQL across real-time (streaming) and lakehouse data (Iceberg, Delta, Hudi) in place—no data copy
- Isolated warehouses for workload isolation (agent, BI, ETL, ad-hoc) on the same data
- Cost-based optimizer for unpredictable AI-generated SQL and novel joins

- PhoenixAI Cloud (BYOC: AWS, Azure, GCP) and PhoenixAI Anywhere (self-managed, on-prem, air-gapped)
- Multi-warehouse query insights, live monitoring, audit logs, and governance
- Vector, full-text, and hybrid search; incremental materialized views (v26.2)
- Agent Fawkes AI copilot (Beta); MCP connector for Claude
- 10K+ concurrent queries at sub-second p95 latency; 90% lower infrastructure footprint claimed

Headline claims

- <1s query latency on multi-table analytical queries
- <5s ingest-to-queryable on streaming mutable data
- 10K+ sustained QPS under load
- 90% lower infrastructure footprint than other real-time analytics solutions
- 8.2× faster than Trino on Iceberg (TPC-DS 1TB benchmark)
- ClickHouse failed 12 of 22 TPC-H queries (OOM); PhoenixAI completed all with better latency (cited in materials)
- Demandbase: 60% hardware reduction, 90% storage cost reduction, 49 clusters -> 1
- Coinbase: 573B+ rows across 300+ tables, <1s latency

Tone: Technical, engineer-to-engineer, confident, direct; emphasizes production readiness, governance, and solving pain that ClickHouse and batch warehouses create for AI agents and high-concurrency workloa

ClickHouse

What it is: Open-source columnar OLAP database for real-time analytics on petabyte-scale data, available as self-hosted (Apache 2.0) or fully managed ClickHouse Cloud service across AWS, GCP, and Azure

Who it serves: Data engineers, platform engineering teams, analytics engineers, and developers at mid-to-large technology enterprises, financial services, e-commerce, gaming, cybersecurity, and AI companies requiring high-throughput analytical workloads, real-time dashboards, observability, log analysis, and time-series analytics

Key features

- Columnar storage with vectorized query execution processing billions of rows per second
- Real-time streaming ingestion from Kafka, Confluent Cloud, Amazon MSK via native Kafka engine and ClickPipes managed ingestion
- Lakehouse analytics with in-place reads/writes for 70+ formats including Iceberg, Delta Lake, Hudi with Glue and Unity Catalog integration
- Auto-scaling serverless compute (Cloud) that scales up/down and suspends during idle with consumption-based pricing
- High compression (LZ4, ZSTD, Delta, Gorilla) achieving 10-20x storage efficiency
- Materialized views for pre-computed aggregations
- Distributed queries with hash joins and replication for fault tolerance
- SQL support (ANSI SQL dialect) with MySQL wire protocol compatibility
- Bring Your Own Cloud (BYOC) deployment option for data residency and compliance
- Managed Postgres service (public beta) with native ClickHouse analytics integration
- ClickStack managed observability for logs, metrics, traces
- Langfuse acquisition adds LLM observability and agent monitoring

Headline claims

- Sub-second queries on billions of rows with blazing-fast analytical performance
- Processes hundreds of millions to over a billion rows per server per second
- 100x faster than traditional databases for analytical workloads
- 23x better cost-performance than nearest cloud data warehouse (CostBench)
- 3-5x more cost-effective than Snowflake in production with 2x faster queries and 38% better compression

- Data queryable within seconds of arrival from Kafka streams
- 10x better storage efficiency via advanced compression reducing storage costs
- Netflix ingests ~5 PB of logs per day using ClickHouse with native protocol and LZ4

Tone: Performance-obsessed and benchmark-driven with aggressive cost-performance positioning against Snowflake, Databricks, BigQuery, and Redshift; emphasizes open-source roots (Apache 2.0), developer-frien

Champion priorities (from saved report)

1. Design and maintain scalable real-time data pipelines from streaming sources (Kafka, Flink CDC) with sub-second to seconds-level data freshness
2. Build lakehouse architectures that unify streaming and historical data (Iceberg, Delta, Hudi) for analytics and ML workloads
3. Support AI agent and ML workloads with low-latency query infrastructure that handles unpredictable, complex SQL patterns and high concurrency
4. Optimize query performance and reduce infrastructure costs through right-sized compute, partitioning, workload isolation, and cost-based optimization
5. Implement data governance, quality checks, schema management, role-based access controls, and regulatory compliance (GDPR, HIPAA) across pipelines
6. Reduce pipeline complexity by eliminating redundant ETL, enabling direct queries on lakehouse tables, and consolidating clusters/tools
7. Ensure high concurrency and sub-second latency for customer-facing analytics, embedded dashboards, and production BI under heavy load

Input gaps — overall confidence: high

- No granular pricing tiers or volume discount structure beyond the AWS Marketplace listings ($0.0065/unit Cloud, $175/CPU/month Enterprise) — Weakens value-prop and objection-handling stages if pricing comparison to ClickHouse or status quo is needed [medium]
- User's answers do not specify which features are roadmap vs. live; homepage and docs describe features as present, but Agent Fawkes is labeled Beta and requires support to enable — May inflate feature claims if roadmap items are positioned as live capabilities [low]
- No direct customer language from call notes, support tickets, or community forums—only published case studies and AWS/G2 reviews — Messaging may miss real buyer objections and language patterns; limits authenticity of pain-point framing [high]
- Competitor (ClickHouse) deep dive not yet performed in this stage; user's answers reference ClickHouse weaknesses but do not cite primary sources — Competitive differentiation stage will need web research to validate claims [medium]

Company profile, feature set, customer proof, and Data Engineer priorities are well-documented in user answers, generated reports, and public materials; missing only granular pricing and unfiltered customer verbatims.

STAGE 1

Competitive Alternatives

ClickHouse (competitor)

Proven open-source OLAP engine with huge community, mature ecosystem, and documented sub-second performance on analytical workloads. Data engineers pick it for real-time analytics dashboards, logging/observability, and high-throughput batch analytics where denormalized schemas are acceptable and joi

Where it falls short

- Distributed joins lack shuffle algorithms—each node processes independently, causing OOM failures on complex multi-table queries at scale

- ReplacingMergeTree forces a choice: stale duplicate rows until background merge completes, or use FINAL which degrades query performance 5-20× on tables with many unmerged parts
- Unpredictable LLM-generated SQL with novel join patterns breaks or times out without extensive manual tuning and denormalization workarounds
- Query optimizer is static and cannot adapt to runtime conditions; no cost-based optimization for complex joins
- Cloud pricing ($0.22–$0.39/unit-hour) is expensive relative to self-hosting, and autoscaling feels overly aggressive without granular control

Apache Pinot (competitor)

Purpose-built for user-facing, low-latency analytics with strict p99 SLAs. Data engineers choose Pinot for customer-facing dashboards and high-concurrency workloads (10K+ QPS) where ultra-low latency (single-digit milliseconds) is critical and data can be denormalized.

Where it falls short

- Multi-table joins are limited and expensive; system assumes denormalized, wide-table schemas
- SQL completeness trails general-purpose analytical databases—complex ad-hoc queries with window functions or nested subqueries underperform
- Requires external tools (Presto, Spark) to prepare and transform data before ingestion; no native SQL-based transformation during ingest
- Append-only architecture makes upserts and deletes operationally complex; no native UPDATE or DELETE support

Apache Druid (competitor)

Excels at time-series rollups and pre-aggregation with sub-second queries on streaming telemetry. Data engineers pick Druid for operational dashboards, network monitoring, and exploratory time-based analytics where pre-aggregated metrics suffice.

Where it falls short

- Complex joins are not its strength and significantly impact latency; limited native join support compared to general OLAP engines
- Pre-aggregation at ingestion means queries requiring original raw rows (late-arriving dimension enrichment, individual event lookup) are impossible or require a separate raw data store
- More moving parts to set up and manage compared to single-binary OLAP engines; schema changes require altering ingestion specs
- Append-only segment-based architecture makes real-time upserts and deletes operationally complex and inefficient

Snowflake (competitor)

Dominant cloud data warehouse with organizational footprint, unified governance, and warehouse-first workflows already in place. Data engineers inherit Snowflake when it's the company standard and use it for BI dashboards, batch analytics, and operational reporting.

Where it falls short

- Query latency is seconds (3–10s typical, even with Interactive Warehouses), not sub-second, making it unsuitable for customer-facing analytics or AI agent workloads requiring <1s responses
- Snowpipe Streaming delivers seconds-level freshness (5+ seconds ingest-to-query), insufficient for real-time use cases requiring sub-second data availability
- 60-second minimum billing per warehouse start inflates costs for short, frequent queries typical of agent and embedded analytics workloads
- Interactive Warehouses require multi-cluster configuration and tuning to scale; single-cluster deployments saturate quickly under high concurrency

Databricks (competitor)

Unified lakehouse platform combining data engineering, ML, and analytics. Data engineers choose Databricks when they need Spark-based ETL, ML model training, and Delta Lake governance in one platform with deep Python/Spark ecosystem integration.

Where it falls short

- Cold start time is 3–7 minutes for cluster warmup, adding latency to ad-hoc queries and making it unsuitable for low-latency agent or customer-facing workloads
- Query latency for analytical workloads is seconds to tens of seconds (Reyden engine in beta claims sub-second but not GA), not sub-second at production scale
- Streaming mode achieves millisecond latency only for streaming ingestion and transformation, not for complex analytical SQL queries joining large tables
- Requires managing separate compute clusters, notebooks, and job orchestration—operational overhead compared to pure analytical databases

Status quo: Keep current ClickHouse deployment running, wait for it to break, or do nothing (status quo)

Migration risk, schema redesign complexity, and operational effort to rebuild pipelines outweigh performance gains when current pain isn't severe enough. Data engineers tolerate query timeouts, FINAL tax, and occasional OOM errors until agent workloads or customer-facing SLAs force a change.

Where it falls short

- Technical debt accumulates: denormalization workarounds proliferate, cluster sprawl increases infrastructure costs, and manual tuning becomes unsustainable as query patterns diversify
- Cannot support new AI agent or customer-facing analytics initiatives that require sub-second latency on complex joins and fresh streaming data
- Team velocity slows as engineers spend time firefighting query failures, optimizing around ClickHouse limitations, and maintaining duplicate pipelines for real-time and historical data

---

STAGE 2

Differentiators

Distributed MPP joins on normalized data without precomputation or denormalization

vs. ClickHouse, Apache Pinot, Apache Druid · Source: Research

ClickHouse lacks data shuffling between nodes—each node processes its part independently, causing RAM spikes and query crashes on complex joins (500M events × 20M users example). Pinot and Druid assume denormalized wide-table schemas; multi-table joins are limited/expensive. PhoenixAI runs complex multi-table joins on normalized schemas with sub-second latency (Coinbase: 573B rows across 300+ tabl

Sub-second freshness with streaming upserts and deletes queryable in <5 seconds without FINAL tax

vs. ClickHouse, Snowflake, Databricks · Source: Research

ClickHouse ReplacingMergeTree forces a choice: stale duplicate rows until background merge, or SELECT FINAL which is 5-20× slower and forces single-threaded merges on large tables. Snowflake Snowpipe Streaming delivers 5+ second latency. Databricks streaming mode is for ingestion/transformation, not analytical query freshness. PhoenixAI commits upserts in-place with <5s ingest-to-queryable at 100K

Unified SQL across real-time streaming and lakehouse data (Iceberg, Delta, Hudi) in place—no data copy, no separate pipeline

vs. ClickHouse, Apache Pinot, Apache Druid, Snowflake · Source: Your answer

ClickHouse, Pinot, Druid, and Snowflake require separate ingestion pipelines or staging for lakehouse data.

PhoenixAI queries Iceberg, Delta, Hudi in place alongside streaming Kafka tables in one SQL statement. TPC-DS 1TB benchmark: 8.2× faster than Trino on Iceberg, 1.3× faster than Databricks. Coinbase queries 573B rows across live Kafka feeds and lakehouse tables with <1s latency.

Isolated warehouses for workload isolation (agent, BI, ETL, ad-hoc) on the same data with independent scaling

vs. ClickHouse, Snowflake, Databricks · Source: Saved report

ClickHouse requires separate clusters for workload isolation, leading to cluster sprawl (Demandbase: 49 clusters). Snowflake and Databricks require separate virtual warehouses/clusters, each with duplicate data or shared storage but separate compute provisioning. PhoenixAI isolated warehouses run on the same data with independent compute scaling per workload (Demandbase: 49 ClickHouse clusters -> 1

Cost-based optimizer that automatically handles unpredictable AI-generated SQL and novel joins without manual tuning

vs. ClickHouse, Apache Pinot, Apache Druid · Source: Your answer

ClickHouse query planner is static (auto-reordering for 2-table joins added in 24.12, but no runtime adaptation). Pinot and Druid optimizers assume denormalized schemas with limited join support. PhoenixAI cost-based optimizer handles LLM-generated SQL with global runtime filters and query feedback delivering 10x–100x speedups on multi-join patterns (Demandbase replaced ClickHouse architecture tha

BYOC deployment (AWS, Azure, GCP) where data and compute stay inside customer's own cloud account, plus self-managed on-prem and air-gapped support

vs. ClickHouse, Snowflake, Databricks · Source: Your answer

ClickHouse Cloud runs in ClickHouse's cloud account (though BYOC launched recently). Snowflake runs in Snowflake's account. Databricks offers serverless in Databricks account or classic compute in customer account. PhoenixAI Cloud BYOC deploys fully managed service in customer's AWS/Azure/GCP account (data never leaves); PhoenixAI Anywhere supports self-managed Kubernetes in VPC, on-prem, and air-

Incremental materialized views (IMV) that refresh automatically on base table updates without full recomputation

vs. ClickHouse, Apache Pinot, Apache Druid · Source: Your answer

ClickHouse materialized views are static aggregations updated on insert; no incremental refresh on upserts/deletes. Pinot and Druid pre-aggregate at ingestion but do not support incremental materialized views on arbitrary SQL. PhoenixAI v26.2 shipped incremental materialized views that refresh only affected partitions when base tables change.

Vector, full-text, and hybrid search inside the database (v26.2) for AI/RAG workloads

vs. ClickHouse, Apache Pinot, Apache Druid, Snowflake, Databricks · Source: Your answer

ClickHouse has experimental vector search (cosine distance UDF). Pinot, Druid lack native vector search. Snowflake and Databricks require external vector stores or beta features. PhoenixAI v26.2 ships vector, full-text, and hybrid search natively inside the analytical database, enabling RAG and semantic search on the same platform as analytical queries.

Native Kafka connector supporting Protobuf, Debezium CDC, and multi-topic loads with schema registry integration

vs. Snowflake, Databricks · Source: Your answer

Snowflake Snowpipe Streaming requires Kafka Connect or custom SDK integration. Databricks uses Spark Structured Streaming with external connectors. PhoenixAI provides starrocks-kafka-connector with native Protobuf, Debezium CDC, multi-topic batch loads, and Confluent Schema Registry integration out of the box.

Not counted (not live yet)

- Agent Fawkes AI copilot is labeled Beta and requires PhoenixAI Support to enable; treated as not generally available for competitive claims

STAGE 3

Value Ladder

Value ladder

STAGE 4

Champion & Situation

Triggers

Pains (most painful first)

Tasks it helps with

Tasks it doesn’t touch

STAGE 5

Market Category

Data engineers will compare PhoenixAI to ClickHouse, Apache Pinot, and other real-time OLAP databases at first glance because the homepage says 'real-time analytical database' and the use cases (sub-second queries, streaming ingestion, high concurrency) match that category. This comparison helps PhoenixAI because its core differentiators—distributed MPP joins without denormalization, sub-second freshness without FINAL tax, unified streaming + lakehouse SQL, and workload isolation—directly address the known pain points (join limitations, merge-on-read latency, cluster sprawl) that data engineers hit with ClickHouse and other real-time OLAP engines. The category frame makes the value obvious.

Market category

Real-Time Analytical Database (Real-Time OLAP) — Existing category

Helps: Buyers already search for 'real-time OLAP' or 'real-time analytical database' and understand the use cases (streaming analytics, customer-facing dashboards, high concurrency). PhoenixAI's differentiators (distributed joins, no FINAL tax, unified lakehouse SQL, workload isolation) are immediately relevant and solve known ClickHouse/Pinot/Druid pain points. Shortest path to evaluation.

Hurts: Lumped in with ClickHouse open-source, which has huge brand recognition and a massive community. Data engineers may assume PhoenixAI is 'yet another ClickHouse alternative' and not investigate further. Also competes with Apache Pinot, Druid, and StarRocks open-source (PhoenixAI's own foundation), which are free. Risk of being seen as expensive managed version of open-source without clear different

Positioning statement: For data engineers who need to serve AI agents, customer-facing analytics, and BI dashboards with sub-second queries on fresh streaming and lakehouse data at scale, PhoenixAI is a real-time analytical database that runs distributed MPP joins on normalized data, commits streaming upserts queryable in under 5 seconds without latency degradation, and isolates agent, BI, and ETL workloads on the same data with independent scaling. Unlike ClickHouse, it handles complex multi-table joins and unpredictable AI-generated SQL without denormalization workarounds, the FINAL tax, or cluster sprawl.

Other categories considered

Agentic AI Database / AI Agent Data Platform — Subcategory

Helps: Aligns with PhoenixAI's homepage positioning ('The real-time analytical database for AI agents'), $80M Series B messaging ('defining the category of agentic AI databases'), and the Demandbase/Conductor proof points where ClickHouse couldn't handle LLM-generated SQL. Differentiates from general-purpose ClickHouse by making cost-based optimizer, workload isolation, and unpredictable query handling t

Hurts: Buyers searching for 'real-time analytics' or 'streaming database' won't find this subcategory unless PhoenixAI educates them first. Requires marketing investment to explain what an 'agentic AI database' is and why it's different from real-time OLAP. Narrows the addressable market to only teams building AI agents, excluding the larger real-time BI, customer-facing analytics, and operational analyt

Unified Real-Time Data Platform / Live Data Foundation — New category

Helps: Positions PhoenixAI as the architecture that eliminates the need for separate real-time OLAP (ClickHouse/Pinot), lakehouse query engines (Trino/Databricks), and warehouse sprawl—unifying streaming, historical, and agent workloads in one platform. The Demandbase proof point (49 clusters -> 1) and unified SQL across Kafka + Iceberg differentiators support this framing. Could command premium pricing i

Hurts: Requires multi-year, multi-million-dollar marketing investment to educate buyers on why they need a new category instead of picking best-of-breed tools. Data engineers are conservative and prefer proven categories with peer references and ecosystem integrations. PhoenixAI's $80M Series B and ~70-80 employees are insufficient to create and own a new category against entrenched competitors (ClickHou

Check with real buyers

- When data engineers evaluate PhoenixAI against ClickHouse, do they care more about eliminating the FINAL tax or eliminating denormalization workarounds? Which pain point drives the conversation?
- Do buyers recognize 'real-time analytical database' as a category, or do they search for 'ClickHouse alternative,' 'streaming analytics database,' 'OLAP for agents,' or something else? What language do inbound leads use?
- Is workload isolation (isolated warehouses for agent/BI/ETL) a top-3 buying criterion, or is it a nice-to-have that only matters after query performance and freshness are proven?
- Do data engineers need proof that PhoenixAI handles AI agent workloads specifically, or is general sub-second query performance + high concurrency proof sufficient to win the deal?
- When PhoenixAI loses to status quo (no decision), is it because migration risk and operational effort outweigh the pain, or because the current ClickHouse pain isn't severe enough yet?

Your Answers (inputs used)

In one or two sentences, what does your product do and who is it for?

PhoenixAI is a real-time analytical database that runs sub-second SQL queries across streaming data (Kafka, Flink CDC) and lakehouse tables (Iceberg, Delta, Hudi) with complex joins and high concurrency. It's built for data engineers and data platform teams who need to serve AI agents, customer-facing analytics, and BI dashboards with fresh data at scale.

Sources: phoenixdata.ai homepage; Battle Card Generator 2026-10-25; Persona Builder 2026-10-25; AWS Marketplace listing

Which features are live today, and which are still on the roadmap?

Live:

- Distributed MPP joins on normalized data without precomputation
- Sub-second freshness with streaming upserts/deletes from Kafka and Flink CDC (100K+ events/sec, queryable in <5s)
- Unified SQL across real-time and lakehouse data (Iceberg, Delta, Hudi) in place
- Isolated warehouses for workload isolation (agent, BI, ETL, ad-hoc)
- Cost-based optimizer for AI-generated SQL
- PhoenixAI Cloud (BYOC: AWS, Azure, GCP) and PhoenixAI Anywhere (self-managed, on-prem, air-gapped)
- Multi-warehouse query insights, live monitoring, and audit logs
- Agent Fawkes AI copilot (SQL generation, optimization, schema browsing) — listed as Beta, requires PhoenixAI

Support to enable
- Incremental materialized views (IMV) shipped in v26.2
- Vector, full-text, and hybrid search inside the database (v26.2)
- MCP connector for Claude
- Multi-AZ deployment, maintenance windows, Apache Ranger integration, shared VPC (AWS)

Roadmap / unclear:

Assumption: The homepage shows multi-cloud deployment architecture and describes features as available, but Agent Fawkes is explicitly labeled Beta and must be enabled by support. No other features are marked as coming soon or waitlist on the website or in Sept 2026 release notes. Materialized views and search are described as shipped in 26.2. If there are unreleased features, they are not disclosed on public pages reviewed.

Sources: phoenixdata.ai homepage; PhoenixAI Cloud release notes (docs.phoenixdata.ai/Cloud/release_notes); PhoenixAI blog posts (incremental materialized views, vector search, Agent Fawkes, MCP connector); Battle Card Generator report mentioning customer deployments

What can your product or team do that isn't on your website?

• **Founded and maintain the StarRocks open-source project (Apache 2.0, Linux Foundation)** — PhoenixAI created StarRocks and still leads development; the commercial product extends the core engine with enterprise governance, workload isolation, multi-AZ HA, and 24/7 SLA support. (Source: phoenixdata.ai/starrocks, About page, YouTube, job listings)

• **Founder expertise in distributed systems and real-time data at scale** — CEO James Li, CPO Andy Ye, and CTO Alvin Zhao spent years building and operating real-time systems in demanding production environments; President Rick Underwood previously CEO of Clumio (acquired by Commvault) and VP at Snowflake through 2020 IPO. (Source: phoenixdata.ai/about, BigDATAwire press release, Tracxn)

• **30-day free trial with no credit card, full production features** — Not a limited sandbox; connect real data sources, run complex queries, evaluate against actual workload on both Cloud BYOC and Anywhere deployment models. (Source: phoenixdata.ai/pricing)

• **Native integrations: Confluent Cloud, AWS (S3, MSK, Glue), Databricks, dbt, Apache Ranger, Kafka Connect, Flink, schema registries** — Kafka connector (starrocks-kafka-connector) supports Protobuf, Debezium CDC, multi-topic loads; dbt-starrocks for transformation; AWS ISV Accelerate partner. (Source: Partners page, docs on Kafka/Confluent loading, Integrations page)

• **Subscription-based and usage-based pricing models** — PhoenixAI Cloud BYOC at $0.0065/unit (1% of standard CCU), cancel anytime, no end date; PhoenixAI Enterprise at $175/month per CPU core; pricing depends on query volume, data scale, region, support tier. (Source: AWS Marketplace listings, phoenixdata.ai/pricing)

• **SQL and connector compatibility with open-source StarRocks** — Schemas, SQL, and connectors carry over; ANSI SQL + MySQL wire protocol, JDBC/ODBC drivers work with existing BI tools. (Source: StarRocks comparison page, Product page)

Add from your own knowledge: any proprietary customer onboarding playbooks, embedded services/SRE support tiers, private Slack channels, reference architecture templates, or founder advisory engagements you offer that aren't documented publicly.

Sources: phoenixdata.ai/about; phoenixdata.ai/starrocks; phoenixdata.ai/pricing; phoenixdata.ai/partners;

docs.phoenixdata.ai Kafka/Confluent loading guides; AWS Marketplace PhoenixAI Cloud and Enterprise listings; BigDATAwire Rick Underwood press release; Tracxn company profile

Who are your best current customers or design partners?

Publicly named customers (from case studies, homepage logos, and press):

• **Coinbase** — 3,000+ employees, Finance/Crypto, fraud detection and blockchain analytics. Champion: Xinyu Liu, Senior Staff Software Engineer (phoenixdata.ai case study).
• **Demandbase** — ~900–1,000 employees, B2B SaaS/ABM. Champion: Ryan Nowacoski, Senior Engineering Manager, Data Platform (phoenixdata.ai case study, PR).
• **Conductor** — 300–500 employees, SaaS/Marketing Tech (AEO), agentic analytics. Assumption: Chief Product Officer or Data Platform Engineer led the purchase (phoenixdata.ai case study).
• **Eightfold.ai** — ~800–1,000 employees, HR Tech/Talent Intelligence SaaS, migrated from Redshift. Assumption: VP of Engineering or Data Platform lead (phoenixdata.ai case study).
• **Pinterest** — ~5,000 employees, Social Media, migrated from Druid for advertiser analytics. Assumption: Data Platform Engineer or Analytics Engineering lead (phoenixdata.ai case study).
• **Coinbase**, **Expedia**, **Celonis**, **Fanatics**, **AppLovin**, **Intuit**, **Tencent**, **Mihoyo** shown as logos or named in press/case studies.

Pattern: Mid-market to enterprise (300–18,000+ employees), SaaS/fintech/e-commerce/social. Champions are **Senior Data Platform Engineers, Engineering Managers (Data Platform), Senior Staff Software Engineers**, or **VPEs/CTOs** when the migration is strategic (e.g., replacing ClickHouse or Redshift at scale).

Assumption: Data Engineering Manager or Data Platform Engineering Manager role typically champions the purchase unless a larger architectural migration or AI agent workload is involved, in which case VP of Engineering or Head of Data Platform leads.

Sources: phoenixdata.ai/case-studies (Coinbase, Demandbase, Conductor, Eightfold, Pinterest, Celonis); phoenixdata.ai homepage (logos: Coinbase, Pinterest, Expedia, Demandbase, Eightfold, Celonis, Tencent, Mihoyo); Battle Card Generator 2026-10-25 (customer references, job titles); Find My Customer 2026-10-25 (customer list, job titles, pain points); PR Newswire 2026-06-12 (AppLovin, Coinbase, Conductor, Demandbase named); phoenixdata.ai/product/phoenixai-anywhere (Coinbase, Fanatics quotes with job titles)

How do you describe what you are today, and what words do prospects use for it?

How PhoenixAI describes itself today:

Homepage title: "The real-time analytical database for AI agents."

Meta description: "Connect agents to streaming and lakehouse data with sub-second SQL, complex joins, and governed access inside your cloud."

LinkedIn tagline: (not provided in materials)

Category words and search terms prospects use:

Assumption: Real-time OLAP database, real-time analytical database, streaming analytics database, lakehouse analytics platform, AI agent database, OLAP for AI workloads, analytical database for agents.

Assumption: Data engineers searching for alternatives to ClickHouse, Apache Druid, Apache Pinot, or Databricks for agent workloads likely use terms like "real-time analytics," "streaming SQL," "lakehouse query engine," "OLAP for high concurrency," "sub-second queries," "Kafka analytics," "Iceberg query engine."

Assumption: AWS Marketplace reviews label it "real-time analytical database for AI agents and customer-facing analytics" (external G2 reviews present but content not accessible). The Battle Card and Find My Customer reports confirm the "real-time analytical database" positioning is used in sales materials and that competitors are also framed as "real-time OLAP" or "analytical database" products.

Sources: phoenixdata.ai homepage; AWS Marketplace PhoenixAI listing; Battle Card Generator 2026-10-25; Find My Customer 2026-10-25; Web research: real-time OLAP database category definition (textql.com, velodb.io)

How much appetite do you have for creating a new category?

<cite index="1-2,6-2">PhoenixAI raised $80 million in Series B funding in June 2026, led by Sky9 Capital with participation from Atypical Ventures (reported as Lowkey Ventures in one source), Olive Technology Ventures, and existing investors</cite>. <cite index="9-9">Total known funding is around $100 million</cite>. <cite index="1-5">The company is headquartered in Menlo Park, California</cite>, and <cite index="12-9">has 72 employees</cite> (reported as 76–78 by other sources in August 2026). <cite index="6-8">Sky9's Managing Partner said the company is "defining the category"</cite> of agentic AI databases. <cite index="11-8">PhoenixAI rebranded from CelerData in May 2026 and is the original creator of StarRocks, the open-source analytical database</cite>.

Assumption: Public signals suggest an early-stage Series B company with ~70–80 employees and strong investor backing; creating a new category is feasible but would require sustained marketing investment and thought leadership beyond the current "agentic AI database" positioning, which competes in the established real-time OLAP market.

Sources: pulse2.com — PhoenixAI Series B announcement June 2026; phoenixdata.ai/newsroom — PhoenixAI Series B led by Sky9 Capital; PitchBook — PhoenixAI profile, 72 employees; Dealroom — PhoenixAI funding and employee range 51-200; Tracxn — PhoenixAI $80M total funding, 78 employees August 2026

What do prospects use today instead of you, and who do you lose deals to (including “no decision”)?

ClickHouse (clickhouse.com): Data engineers pick it for proven OLAP performance, huge open-source community, and mature ecosystem. Strong on batch analytics and logging, but struggles with distributed joins at scale, upserts require the FINAL tax that degrades query speed, and unpredictable LLM-generated SQL breaks without manual tuning. Demandbase replaced a 49-cluster ClickHouse architecture that couldn't handle agent workloads.

Apache Pinot, Apache Druid: Open-source real-time OLAP alternatives. Pinot excels at high concurrency (10K+ QPS) but limited SQL completeness and requires denormalized schemas. Druid strong on time-series rollups but minimal native join support. Pinterest migrated from Druid.

StarRocks, Apache Doris, Firebolt: Direct real-time OLAP competitors with similar messaging (sub-second queries, streaming + lakehouse, agent workloads). Assumption: PhoenixAI is built on StarRocks open-source, so loses to StarRocks open-source when teams prefer self-managed over commercial.

Databricks, Snowflake, BigQuery, Redshift: Cloud data warehouses. Data engineers choose them for existing organizational footprint, unified governance, and warehouse-first workflows. Latency is seconds-to-minutes vs. sub-second, but inertia and "good enough" analytics win when fresh data isn't critical. Eightfold migrated from Redshift.

Status quo — keep ClickHouse running, wait, or do nothing: Assumption: Most common "competitor." Migration risk, schema redesign complexity, and operational effort outweigh performance gains when current pain isn't severe enough. ClickHouse teams hit query timeouts, join limitations, or agent-workload failures before they search for alternatives.

Assumption: PhoenixAI cannot see lost deals or "no decision" data; these alternatives are inferred from competitor analysis, case study migrations, and persona pain points.

Sources: phoenixdata.ai homepage; Battle Card Generator 2026-10-25 (ClickHouse comparison, Demandbase case study); Find My Customer 2026-10-25 (competitor list: StarRocks, Apache Doris, Apache Pinot, Firebolt, ClickHouse, Apache Druid, SingleStore, Tinybird); phoenixdata.ai case studies (Eightfold Redshift migration, Pinterest Druid migration); Web research: real-time OLAP comparison 2026 (estuary.dev, guideflow.com, pingcap.com, velodb.io); Web research: ClickHouse vs Pinot vs Druid (risingwave.com, jusdb.com, ksolves.com); Web research: ClickHouse migration pain points (altinity.com, severalnines.com, sivaro.in)

Where are clickhouse.com’s claims wrong or overstated in practice?

"Fast open-source OLAP database for real-time analytics" — Complex joins remain bottlenecked

<cite index="21-6,21-7,21-8,21-9">ClickHouse choked on joins between a 500M-record events table and 20M users table with no data shuffling between nodes, spiking RAM and crashing queries.</cite> <cite index="22-5,22-8,22-9">ClickHouse lacks dedicated distributed JOIN algorithms and data shuffling, so each node processes its part independently—this works for simple queries but becomes slow or fails for complex JOINs on large tables.</cite> <cite index="25-4">ClickHouse does not perform shuffle joins (redistributing both sides by join key across nodes); this is a known limitation as of 2025.</cite> <cite index="23-1,23-4">Distributed JOINs default to broadcasting the right side, which is efficient only for small dimension tables but falls apart at scale.</cite> Source: Medium, Glassflow.ai, neverblink.ai, oneuptime.com.

"Real-time streaming ingestion from Kafka" queryable in seconds — FINAL tax or stale data trade-off

<cite index="2-5">On a table with many unmerged parts, FINAL can be 5-20x slower than a regular scan.</cite> <cite index="5-6,5-7">Avoid SELECT ... FINAL—it forces single-threaded merges and is extremely slow on large tables.</cite> <cite index="8-8,8-9,8-10">Asynchronous merging brings significant real-time issues; data merging operations often experience delays of several minutes or longer, so query results may contain duplicate data within the time window from insert until merging completes.</cite> Source: oneuptime.com, e6data.com, oreateai.com.

"Auto-scaling serverless compute"—aggressive, not granular

<cite index="15-2,15-3">The autoscaling feature of ClickHouse Cloud doesn't currently offer granular control; ClickHouse determines when to scale, and it feels overly aggressive.</cite> Source: Gartner Peer Insights.

"ClickHouse Cloud" pricing—expensive relative to self-hosting

<cite index="10-7">ClickHouse's cloud pricing can feel expensive relative to self-hosting.</cite> <cite index="12-5,12-6">Reviewers note the cloud option can become expensive and that usage needs monitoring to keep spend predictable; users report that data modeling and complex query optimization require deeper expertise, with a learning curve for newcomers.</cite> <cite index="28-1,28-2">The challenge with the cloud offering for ClickHouse is that it's too expensive; the real pricing fight is not raw dollars per terabyte, it is whether managed ClickHouse removes enough operational pain to justify giving back the open source cost advantage.</cite> <cite index="31-14,31-15">Basic assumes 6 hours of activity per day, so the $66.52 figure bakes in significant idle time; running Basic continuously costs $186.27/month—nearly 3x the headline.</cite> Source: G2, Instaclustr, Sacra, DEV Community.

"Columnar storage and vectorized query execution"—limited community and learning curve

<cite index="14-2,14-3">Delays are experienced when trying to delete databases with corrupt data, taking too much time and causing major outages; the community surrounding ClickHouse also seems limited, providing a reliance on documentation, and there is a scarcity of developers working with ClickHouse.</cite> <cite index="11-4,11-5,11-6">Users find beginner unfriendliness in ClickHouse due to complex query optimizations that challenge new users; complex query optimizations demand expertise, making it challenging for newcomers to navigate effectively.</cite> Source: PeerSpot, G2.

Sources: Medium: JOIN Limitations in ClickHouse (mohhddhassan, glassflow.ai); oneuptime.com: ClickHouse FINAL Keyword Guide; e6data.com: ClickHouse Query Performance Optimization; oreateai.com: ClickHouse FINAL Performance Practices; neverblink.ai: ClickHouse JOIN Performance; G2: ClickHouse Pricing and Pros/Cons reviews; Instaclustr: Managed ClickHouse Solutions Compared; Gartner Peer Insights: ClickHouse Reviews

What usually happens right before a Data Engineer reaches out or starts looking?

1. Legacy ClickHouse or data warehouse architecture can't handle AI agent query bursts or unpredictable LLM-generated SQL — Demandbase replaced a 49-cluster ClickHouse architecture that couldn't handle novel joins from agentic workloads; Eightfold migrated from Redshift when agent use cases required sub-second latency on complex queries. (Source: Battle Card Generator 2026-10-25, Find My Customer 2026-10-25, phoenixdata.ai case studies)

2. Customer-facing analytics or product dashboard workload outgrows current OLAP engine — Conductor needed second-level SLAs for customer-facing dashboards at terabyte scale under high concurrency; Pinterest replaced Druid for advertiser analytics. Trigger: product growth drives concurrent query volume beyond what the legacy stack can sustain. (Source: Find My Customer 2026-10-25, phoenixdata.ai case studies)

3. Real-time data freshness requirement tightens to under 5 seconds, and current warehouse can't deliver without latency hits — Intuit needed 2-second end-to-end latency for real-time ML predictions; Coinbase required fraud/compliance queries on streaming data queryable within seconds. Trigger: business or compliance SLA change that batch updates or minute-level freshness can no longer meet. (Source: Battle Card Generator 2026-10-25)

4. Infrastructure cost or cluster sprawl becomes unsustainable — Demandbase consolidated 49 ClickHouse clusters to 1, cut hardware 60% and storage 90%. Trigger: cloud bill review, cost optimization initiative, or executive mandate to reduce infrastructure footprint. (Source: Battle Card Generator 2026-10-25, phoenixdata.ai case study)

Assumption: **5. New funding round, data platform team hire, or AI/agent roadmap launch** — when a company hires 2+ data engineers or a Senior Engineering Manager (Data Platform), adopts new streaming/lakehouse tooling, or announces an AI product initiative requiring real-time data access for agents.

Sources: Battle Card Generator 2026-10-25; Find My Customer 2026-10-25; phoenixdata.ai case studies (Demandbase, Coinbase, Conductor, Eightfold, Pinterest); Persona Builder 2026-10-25; tacnode.io ClickHouse alternatives guide; chistadata.com Snowflake to ClickHouse migration framework

Which of a Data Engineer’s tasks does your product not handle?

• **Pipeline orchestration and scheduling** — PhoenixAI is the query engine, not the scheduler. Data engineers still need Airflow, Dagster, Prefect, or similar tools to schedule, monitor, and retry ETL/ELT jobs.

• **Data extraction and ingestion connectors** — PhoenixAI ingests from Kafka, Flink CDC, and S3-compatible storage, but data engineers must build or configure connectors for SaaS APIs, databases, and third-party platforms outside those sources.

• **Data transformation and dbt orchestration** — PhoenixAI supports dbt transformations via SQL, but does not replace dbt, Spark, or Python-based transformation logic. Data engineers still write and maintain transformation code.

• **Data quality frameworks and validation** — Assumption: PhoenixAI does not provide built-in data quality checks, anomaly detection, or validation rules (e.g., Great Expectations, Soda, Monte Carlo). Data engineers implement these separately.

• **Data cataloging, lineage tracking, and discovery** — Assumption: PhoenixAI does not provide a data catalog, automated lineage visualization, or metadata search. Data engineers need Apache Atlas, Atlan, Collibra, or similar tools.

• **CI/CD automation for data pipelines** — Assumption: PhoenixAI does not manage version control, testing, or deployment automation for pipeline code. Data engineers still need Git, GitHub Actions, or similar CI/CD tooling.

• **Data masking, tokenization, and PII management** — Assumption: PhoenixAI supports role-based access control and per-query audit, but does not automate data masking, encryption, or tokenization workflows. Data engineers implement these in pipelines or via external tools.

• **Cost monitoring and FinOps dashboards** — PhoenixAI Cloud provides query insights and warehouse monitoring, but data engineers managing multi-cloud or multi-tool stacks need separate FinOps platforms to track total infrastructure cost.

• **ML feature stores and model serving** — PhoenixAI provides fast SQL for feature engineering and supports ML workloads, but does not replace dedicated feature stores (e.g., Feast, Tecton) or model serving platforms.

• **Self-service data tools and documentation platforms** — Assumption: PhoenixAI does not generate or host end-user documentation, data dictionaries, or self-service portals. Data engineers build these separately.

Sources: Persona Builder 2026-10-25 (Data Engineer work priorities); phoenixdata.ai homepage and product pages; phoenixdata.ai/partners (Kafka, Confluent, dbt integrations); Battle Card Generator 2026-10-25 (PhoenixAI features vs ClickHouse); Web research: Data Engineer responsibilities 2026 (guvi.in, dataexpert.io, kaashivinfotech.com)

What proof do you have?

Customer results (published case studies):

• Demandbase: 60% hardware reduction, 90% storage cost reduction, 49 clusters consolidated to 1, <1s query latency, <2 months idea to production (phoenixdata.ai/case-studies/demandbase)

• Coinbase: 573B+ rows in production across 300+ tables and 10 blockchains, 30K messages/sec Kafka ingestion, <1s latency on TPC-H 1TB benchmark (phoenixdata.ai/case-studies/coinbase)

• Conductor: queries scanning 6–7 GB / 180–240M rows completing in 389ms–2.8s, 100% decimal accuracy for AI agents, <2 months idea to production (phoenixdata.ai/case-studies/conductor)

• Eightfold.ai: 2× faster query response vs. Redshift, 2× infrastructure cost reduction at scale, <1s p95 dashboard latency multi-tenant (phoenixdata.ai/case-studies/eightfold)

• Pinterest: 50% lower p90 latency vs. Druid, 3× cost-performance efficiency using 32% of prior instances, 10-second data freshness (phoenixdata.ai/case-studies/pinterest)

Customer quotes:

• Ryan Nowacoski, Senior Engineering Manager, Data Platform, Demandbase: "PhoenixAI gives us a fast, isolated warehouse for agent workloads directly on our Apache Iceberg tables, with the optimizer handling novel joins automatically." (phoenixdata.ai homepage)

• Xinyu Liu, Senior Staff Software Engineer, Coinbase (phoenixdata.ai homepage, case study)

Customer logos: Coinbase, Demandbase, Conductor, Eightfold.ai, Pinterest, Celonis, Expedia, Fanatics, AppLovin, Intuit, Tencent, Mihoyo (phoenixdata.ai homepage)

Founder background:

• James Li, Co-Founder & CEO: 10+ years in big data, co-founded StarRocks open-source project 2020, previously Co-Founder & CEO at AbleCloud (2014–2020), Head of Open Cloud at Xiaomi (2014), Senior Software Engineer at Baidu (2007–2012), Software Engineer at Microsoft (2012–2014) (phoenixdata.ai/about, rocketreach.co, simplify.jobs)

• Andy Ye, Co-Founder & CPO: Director at Alibaba Group, CPO at Shanghai Xforceplus, co-founded StarRocks 2020 (phoenixdata.ai/about, simplify.jobs)

• Alvin Zhao, Co-Founder & CTO: 10+ years in databases and distributed systems, architect of PhoenixAI's core engine, expert in columnar storage and vectorized execution (phoenixdata.ai/about, LinkedIn)

• Rick Underwood, President: Previously CEO of Clumio (acquired by Commvault), VP at Snowflake through 2020 IPO, sales leadership at MongoDB and Zuora (phoenixdata.ai/about, LinkedIn)

Awards: PhoenixAI named AI-Powered Enterprise Agent Solution of the Year in the 2026 Tech Ascension Awards (prnewswire.com, Oct 1 2026)

Funding & press: $80M Series B led by Sky9 Capital, June 2026; total ~$100M raised; rebranded from CelerData May 2026; created and maintain StarRocks open-source project (Apache 2.0, Linux Foundation) (phoenixdata.ai/newsroom, preqin.com)

Sources: phoenixdata.ai/case-studies (Demandbase, Coinbase, Conductor, Eightfold, Pinterest); phoenixdata.ai homepage (customer quotes, logos); phoenixdata.ai/about (founder bios); rocketreach.co James Li profile; simplify.jobs CelerData company news; prnewswire.com Oct 1 2026 (Tech Ascension Award); preqin.com PhoenixAI funding profile

Paste any customer language.

PhoenixAI / CelerData customer quotes (from case studies, press releases, and AWS Marketplace):

• "PhoenixAI changed the equation: streaming updates from Kafka become queryable within seconds, analysts get sub-second responses on live normalized data, and our AI agents operate on the same real-time dataset." — Xinyu Liu, Senior Staff Software Engineer, Coinbase (Source: phoenixdata.ai homepage, PR Newswire June 2026)

• "Demandbase AI introduced unpredictable LLM-generated SQL that our previous ClickHouse-based architecture wasn't built to handle. PhoenixAI gives us a fast, isolated warehouse for agent workloads directly on our Apache Iceberg tables, with the optimizer handling novel joins automatically." — Ryan Nowacoski, Senior Engineering Manager, Data Platform, Demandbase (Source: phoenixdata.ai homepage)

• "What I appreciate most about Celerdata is its incredibly fast query performance, powered by StarRocks, and its world-class support team." — AWS Marketplace review, G2 verified (Source: AWS Marketplace PhoenixAI Cloud

listing)

• "I appreciate CelerData for the agility we have to ingest data. CelerData has positively impacted my organization as it reduced the time to market." — AWS Marketplace review, verified AWS customer, Jan 14 2026 (Source: AWS Marketplace PhoenixAI Enterprise listing)

• "I have identified issues where the transportation and documentation do not reflect the reality of the models, and I found data points issues related to bugs in the code, so for the open-source version, CelerData needs to be more consistent and reliable compared to the cloud version." — AWS Marketplace review, Jan 14 2026 (Source: AWS Marketplace PhoenixAI Enterprise listing)

StarRocks open-source user quotes (from G2 and Medium):

• "The best thing about StarRocks is that it has been desgined in such a way that it makes real time analytics easy for enterprises." — Kuldeep S., Staff Consultant, G2 verified review May 22 2024 (Source: G2 StarRocks reviews)

• "Sometimes cluster fails after loading data one database to another." — Mohammad Shabbir S., Software Engineer, G2 verified review Jan 9 2024 (Source: G2 StarRocks reviews)

• "Starrocks performed well. In our case, we had around a peak of 150 Million records stored inside, and we were still seeing millisecond level performance on point queries and simple aggregations." — Dave, Medium user, Nov 2023 (Source: medium.com/@doppledave)

Input gaps: No call notes, support tickets, community forum discussions, or Slack channel feedback available. Suggest pulling real customer language from:
• Internal customer success call notes and Slack transcripts
• Support ticket verbatims (Zendesk, Intercom, etc.)
• Closed-won deal notes from sales team
• Customer advisory board feedback sessions
• User interviews or surveys conducted by product/research team

Sources: phoenixdata.ai homepage; phoenixdata.ai/product/phoenixai-anywhere; phoenixdata.ai newsroom — Series B announcement June 2026; AWS Marketplace PhoenixAI Cloud listing — G2 reviews; AWS Marketplace PhoenixAI Enterprise listing — verified customer Jan 2026; G2 StarRocks reviews — Kuldeep S. May 2024, Mohammad Shabbir S. Jan 2024; medium.com/@doppledave — StarRocks first month review