MCP SERVER | NO CODE | CLOUD HOSTED

QuestDB (Time-Series) MCP

Querying metrics, logs, and trends via natural language.

QuestDB connects your AI agent directly to a high-performance time-series database, letting you run complex data queries using natural language. It handles everything from real-time metrics analysis and bulk data ingestion to exporting massive datasets in CSV or Parquet format.

A+ Quality Score 100/100

time-series | sql | data-ingestion | real-time-analytics | questdb

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/questdb-time-series

SCAN TO VIEW

The connectivity layer between AI and the world's software.

Vinkius sits between AI and every application. All communication passes through Vinkius Cloud via the Model Context Protocol (MCP) — with governance, observability, and security at every layer.

Your AI Connections Run Through Vinkius Cloud

The world's largest managed MCP catalog

Vinkius is the connectivity layer where AI connects to the software your business already runs. We handle the hosting, the security, the credentials, the uptime — you get agents that actually do things.

We operate the world's largest managed MCP catalog. Major SaaS platforms, CRMs, databases, and cloud providers — running, monitored, production-ready. This MCP server is hosted and maintained by the Vinkius Cloud for AI Agents.

> *The agent doesn't manage credentials, doesn't manage uptime, doesn't manage security. Vinkius does.*
>
> — Architecture principle

Four Pillars of the Vinkius Runtime

01 — Security by design

Credentials stay encrypted at rest via AES-256. The AI agent never touches raw keys — they're injected into a sandboxed V8 isolate at runtime. Actions are logged, and connections have an emergency kill switch.

02 — Built on MCP Fusion

This MCP server was built with **MCP Fusion**, the open-source framework (Apache 2.0) that powers the entire Vinkius catalog. Schema-as-firewall strips undeclared fields, compiled PII redaction runs at zero overhead, and cryptographic lockfiles produce git-diffable audit trails.

03 — Deterministic observability

Eight immutable metrics per endpoint: request volume, p95 latency, error rate, active connections, cost attribution. A live payload feed logs every tool call with mutation detection.

04 — Autonomous operations

Servers are deployed, monitored, and patched autonomously. New capabilities and security patches ship weekly. Zero-downtime deployments ensure continuous availability across all managed MCP servers.

AES-256	Ed25519	24h TTL	V8 Isolate
Encryption at rest	PKI vault signatures	Ephemeral session keys	Sandboxed execution

One Token. Instant Access.

Every MCP server on Vinkius is accessed through a **Connection Token**. Tokens are generated in the cloud dashboard and produce a unique MCP endpoint URL. Paste this URL into any MCP-compatible client — no SDK required.

A single token can serve **multiple AI clients simultaneously**, or you can issue separate tokens per client for granular access control. Each token tracks its own request count, last activity timestamp, and can be individually enabled or revoked.

MCP ENDPOINT

```
https://edge.vinkius.com/{token}/mcp
```

Claude · Cursor · VS Code · Windsurf · Grok · Gemini

Security Is the Architecture

Security in Vinkius is not a feature — it's the foundation of the runtime. The gateway enforces multiple independent protection layers between AI agents and third-party APIs.

01 — Ed25519 PKI Vault

Every workspace has an Ed25519 Master Key. Session keys are generated ephemerally (24h TTL) and signed by the Master Key. Credentials never leave the vault boundary.

02 — V8 Isolate Sandboxing

Tool code runs inside isolated-vm V8 isolates with 64 MB memory caps and per-request timeouts. No filesystem access, no network access except through the SSRF-guarded fetch bridge.

03 — SSRF Guard

All outbound HTTP requests are DNS-resolved and validated before execution. Private IP ranges (10.x, 172.16-31.x, 192.168.x, AWS metadata 169.254.x) are blocked at the network layer.

04 — DLP & PII Redaction

A ResponseGuard pipeline intercepts every tool response. Configurable redaction patterns strip sensitive fields (emails, SSNs, card numbers) before data reaches the AI agent.

05 — Cryptographic Audit Trail

Every request is signed into a SHA-256 hash chain with Ed25519 signatures. Events form a tamper-proof, SIEM-exportable forensic record.

06 — Honeytoken Trap System

Phantom credentials are injected into isolated environments. If a honeytoken is used outside Vinkius infrastructure, the server is quarantined instantly.

Emergency Kill Switch

EU AI Act Art. 14(1) Compliant

The kill switch is an **emergency halt** mechanism — not a simple toggle. When triggered, it executes three actions atomically:

01 — Server deactivated

The MCP server is immediately taken offline across the entire cluster.

02 — All tokens revoked

Every connection token is invalidated. Total lockout — reconnection blocked until new tokens are issued.

03 — WebSocket connections killed

Active connections terminated via Redis pubsub broadcast. Propagates to every runtime node in the cluster.

Full Visibility. Zero Guesswork.

The Vinkius cloud dashboard includes a full MCP Governance suite — real-time analytics and security controls for production AI operations.

Control Plane

KPI dashboard with request volume, latency, success rate, token consumption, and AI-generated operational briefings.

Agent Activity

Which AI clients are connecting, how often, and what they're doing — real-time session tracking.

FinOps

Cost tracking per tool, payload compression savings, budget optimization signals, and consumption trends.

Tool Health

Slowest and most error-prone tools, with actionable root-cause insights and performance baselines.

Firewall & DLP

PII redaction activity, sensitive data protection counters, and security event timeline.

Incident Log

Error trends, failure rates, status-code breakdowns, and forensic audit trail access.

> Get started at **cloud.vinkius.com** — connect your AI agent in under 60 seconds.

QuestDB (Time-Series) MCP

4 tools available
Cloud-hosted on Vinkius

You can treat your database like an extension of your conversation. Instead of writing boilerplate SQL, simply ask your agent for the average temperature over the last hour or check what happened to a metric two weeks ago. This MCP lets you run complex queries and manage time-series data—whether it's sensor readings, stock prices, or server logs—all through natural language commands. You can also import large amounts of raw data, which automatically builds the necessary tables and schema for you. If your agent needs to extract results for a report, it exports everything cleanly as CSV or Parquet files. Vinkius hosts this MCP, making high-speed time-series analysis available from any compatible client.

Core Capabilities

01 — Run complex database queries

The agent executes standard SELECT, INSERT, and DDL statements to query or modify data in the QuestDB instance.

02 — Bulk import structured data

You can feed tabular files like CSV or TSV directly into tables; the MCP automatically figures out which columns are needed.

03 — Extract large result sets

It pulls query results and exports them immediately as ready-to-use CSV or Parquet files for external analysis.

04 — Check database health

The agent runs a quick check to confirm the server is online and reports its current version number.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/questdb-time-series** — connect your AI agent in three steps.

01 First, subscribe to this MCP on Vinkius and provide your specific QuestDB connection URL along with any necessary authentication credentials.

02 Next, direct your agent to perform a task—for instance, asking it to find the average CPU utilization for last month or import a new batch of sensor readings.

03 The system executes the required operation against the database and returns the actionable result, whether that's a data table, an exported file link, or a simple status message.

The bottom line is you get high-speed access to complex time-series functions without ever leaving your natural language interface.

Built For

Anyone who deals with metrics, logs, or data that changes over time needs this. This MCP is essential for the DevOps engineer tired of clicking through dashboard panels at 2 a.m., and the analyst who can't wait for a DBA to write a custom SQL query.

Data Analyst

Performing ad-hoc analysis on historical metrics, such as calculating year-over-year growth or finding the average value of sensor readings over specific time windows.

DevOps Engineer

Monitoring database health and running maintenance tasks. They use this to check server status or verify that data ingestion pipelines are running correctly.

Data Engineer

Managing the lifecycle of time-series datasets, including creating new tables, modifying schemas, and importing large batches of raw telemetry data.

What Changes When You Connect

01 Stop writing boilerplate SQL. Instead of crafting a complex query for every metric you need, just ask your agent to run the SELECT statement using `execute_sql`. You get the data instantly.

02 Handling massive datasets is easy. Use `import_data` to drop CSV or TSV files into the database. The MCP handles schema creation and partitioning automatically, so you don't have to pre-process anything.

03 Reporting shouldn't require manual exports. After running a query with `execute_sql`, use `export_data` to get your results as clean CSV or Parquet files ready for sharing.

04 Know if the database is healthy before you start. Running the `ping` tool gives you instant server status and version confirmation, letting you verify operational status in seconds.

05 The whole process feels seamless. You keep everything within your chat interface—from querying data to exporting it—without ever opening a separate SQL client.

Real-World Applications

Finding the average server load for last quarter

A DevOps engineer needs to know the quarterly trend of CPU usage. They ask their agent, which uses `execute_sql` to run a time-series query and returns the specific average metric, eliminating the need to dive into multiple dashboard tabs.

Onboarding new sensor data streams

An analyst receives a large batch of raw temperature readings in a CSV. Instead of manually writing schema creation scripts, they use `import_data` and simply upload the file; the MCP builds the table structure automatically.

Preparing data for an external report

A product manager asks their agent to pull the top 10 most active users from a specific period. The agent uses `execute_sql` and then calls `export_data`, giving the PM a clean CSV file they can immediately attach to a presentation.

Checking database connectivity during an incident

When the application goes down, a support technician first runs the `ping` tool. The agent confirms the server status and version instantly, isolating whether the problem is connectivity or a logic bug.

Patterns to Avoid

Treating it like a document search

✗ AVOID

Trying to ask QuestDB for 'all documents mentioning latency spikes'. The database only handles structured time-series metrics, not unstructured text.

✓ INSTEAD

If you have time-stamped metrics (like 'latency_ms'), use `execute_sql` with a specific column name and time range. Use the MCP to query data that has a clear structure.

Writing massive, multi-step scripts

✗ AVOID

The user writes a 50-line script: connect → select → process → export.

✓ INSTEAD

Use the agent's natural language interface. Ask it to 'find X and export as CSV'. The MCP handles calling `execute_sql` followed by `export_data` in two simple steps.

Ignoring data source validation

✗ AVOID

Trying to manually write the schema for a new dataset, risking missing columns or incorrect types.

✓ INSTEAD

Use `import_data`. The tool automatically inspects your uploaded CSV/TSV file and creates the table structure for you, saving hours of manual DDL work.

The Right Fit

You should use this MCP if your data is fundamentally time-based: logs, sensor readings, financial metrics, or performance counters. If you need to know 'what happened at X time,' this is the tool. Don't use it if your primary goal is unstructured text search (like searching a knowledge base) or general relational joins across disparate systems that aren't time-indexed. For pure document storage and retrieval, look for vector database MCPs instead.

The Pain of Manual Data Extraction

Today, getting a clear picture of system performance means jumping between dashboards, writing complex SQL queries that take hours to debug, and then manually running exports just so you can attach the data to a slide deck. It's copy-pasting numbers from one tab into another, hoping you didn't miss a time zone or a crucial metric.

With this MCP connected via Vinkius, your agent handles all that complexity. You simply tell it what metrics you need and what date range you care about. The agent executes the query, pulls the specific data points, and delivers a clean, ready-to-use export file—no manual clicking required.

QuestDB: Instant Data Access

The ability to run ad-hoc queries is immediate. You don't need a dedicated SQL window; you just ask for the average temperature, and the agent uses `execute_sql` to get it. Need to check if the service is up? One prompt runs the `ping` tool.

What changes is that your workflow moves from 'write-debug-run-export' to simply 'ask.' You get real-time insights directly in your conversational interface.

QuestDB (Time-Series) - 4 Tools

These tools allow your agent to run SQL queries, import raw files, check the server health, and export results from the QuestDB time-series database.

#	TOOL	DESCRIPTION
01	`execute_sql`	Use this to run any standard SQL operation, like querying specific metrics or making schema changes (DDL/DML).
02	`export_data`	Extracts the results from a query and packages them for easy download as CSV or Parquet files.
03	`import_data`	Feeds in new data from CSV or TSV, automatically setting up the necessary tables and columns if they don't exist yet.
04	`ping`	Confirms the database server is operational and returns its current version number.

See It in Action

Real prompts you can use once this MCP is connected to your AI agent through Vinkius Cloud.

U Check if the QuestDB server is online and show me the version.

I've pinged the server. It is currently online and running QuestDB version 8.1.0.

U Execute a query to find the average temperature from the 'sensors' table for the last hour.

Executing SQL... The average temperature for the last hour is 22.5°C based on 1,200 data points in the 'sensors' table.

U Export the last 1000 rows of the 'trades' table as a CSV file.

I've initiated the data export. You can now download the CSV containing the last 1000 trade records.

Frequently Asked Questions

01 **How do I connect QuestDB (Time-Series) MCP using the `ping` tool?**

You just ask your agent to check the status. The agent automatically runs the `ping` function, which confirms if the database is online and reports its current version number for you.

02 **Can QuestDB (Time-Series) MCP handle data I don't have a schema for?**

Yes. Use the `import_data` tool. You upload your CSV or TSV file, and the MCP automatically detects and creates the necessary tables and columns before ingestion.

03 **What is the best way to get data out of QuestDB (Time-Series) MCP?**

For reporting, use `export_data`. It takes your query results and packages them into professional CSV or Parquet files that are ready for any external analysis tool.

04 **Does the `execute_sql` tool support complex joins?**

Yes. Since it executes standard SQL, you can run full DML/DDL operations and perform complex joins across different tables within your time-series data.

05 **Is QuestDB (Time-Series) MCP only for monitoring logs?**

No. While great for logs, it handles any metric that changes over time—think stock prices, sensor readings, or server usage counts—as long as the data is structured by time.

Go Live in 60 Seconds

Get your connection token from **cloud.vinkius.com**, then paste the endpoint URL into any MCP-compatible client.

YOUR MCP ENDPOINT

```
https://edge.vinkius.com/[TOKEN]/mcp
```

CLIENT	WHERE TO CONFIGURE
Claude AI	Profile → Customize → Connectors → "+" → Add custom connector → Paste endpoint
Cursor	Settings → Features → MCP Servers → "+ Add New MCP Server" → Type: SSE → Paste endpoint
VS Code	Ctrl/Cmd+Shift+P → "MCP: Add Server" → add `"questdb-time-series": { "url": "…" }`
Windsurf	MCP Settings → `mcp_settings.json` → Add endpoint URL
ChatGPT	Settings → Tools & plugins → Add MCP server → Paste endpoint
Gemini	Extensions → Add MCP Server → Paste endpoint URL

ASK AN AI ABOUT THIS

Let your preferred AI explain this MCP server

- Ask ChatGPT
- Ask Claude
- Ask Perplexity
- Ask Gemini
- Ask Grok

READY TO CONNECT

QuestDB (Time-Series) is live on Vinkius Cloud.

Get your connection token, paste it into your AI agent, and start building. No SDK. No deployment. Just results.

Start at cloud.vinkius.com →

vinkius.com · support@vinkius.com

INDEPENDENT PLATFORM DISCLAIMER

Vinkius is an independent platform and is not affiliated with, endorsed by, sponsored by, verified by, or otherwise authorized by QuestDB (Time-Series). All third-party trademarks, logos, and brand names are the property of their respective owners. Their use in this document is strictly for informational purposes to identify service compatibility and interoperability.

DOCUMENT INFORMATION

Generated	June 2026
MCP Server	QuestDB (Time-Series) MCP
Server ID	`019e38de-cb28-732b-bc30-ed8e4637e2b6`
Platform	Vinkius Cloud for AI Agents
Endpoint	`https://edge.vinkius.com/{token}/mcp`

LICENSE & USAGE

This document is generated automatically by the Vinkius PDF Engine. Content reflects the MCP server configuration at the time of generation and may change as updates are deployed. For the most current information, visit vinkius.com/mcp/questdb-time-series.