

MCP SERVER

NO CODE

CLOUD HOSTED

# Snowflake MCP

## Query Data & Map Schemas Inside Your IDE

Snowflake MCP connects your AI client directly to your Snowflake data cloud. Chat with your IDE to run complex SQL queries, map nested schemas across databases and tables, or check compute costs without ever leaving your local codebase.

**A+** Quality Score 100/100

sql-warehouse

data-cloud

query-execution

schema-mapping

data-warehousing

big-data



# 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.

### 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.

### 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.

### 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**

Encryption at rest

**Ed25519**

PKI vault signatures

**24h TTL**

Ephemeral session keys

**V8 Isolate**

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.

### 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.

### 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.

### 06 — Honeypot Trap System

Phantom credentials are injected into isolated environments. If a honeypot 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.

**FinOps**

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

**Firewall & DLP**

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

**Agent Activity**

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

**Tool Health**

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

**Incident Log**

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

Get started at [cloud.vinkius.com](https://cloud.vinkius.com) — connect your AI agent in under 60 seconds.

# Snowflake MCP

7 tools available

Cloud-hosted on Vinkius

Stop jumping between your code editor and the browser just to look at a table definition. This MCP lets you chat with your agent about your data architecture and get live results. You can ask it to list all available databases, then drill down into schemas or tables—all within the flow of your work. Need to validate a complex join? Tell your bot to execute the SQL query right there, keeping everything native to your AI client. Vinkius hosts this connection, giving your agent deep access to Snowflake's entire data landscape so you can quickly build models and diagnose issues without writing boilerplate setup code.

---

## Core Capabilities

### 01 — Audit Data Structures

List all databases, schemas, and tables in the account to map out complex data relationships.

### 03 — Track Compute Resources

List and monitor active virtual warehouses to understand current compute costs and usage patterns.

### 02 — Run Live Queries

Execute SQL queries directly against your Snowflake instance, allowing for immediate read-only results.

### 04 — Diagnose Data Workloads

Check the status of long-running or asynchronous data engineering queries.

# One Click on Vinkius — From Prompt to Execution

Available at [vinkius.com/mcp/snowflake](https://vinkius.com/mcp/snowflake) — connect your AI agent in three steps.

- 01** Subscribe to this MCP and provide your explicit Snowflake Account identifier (e.g., abc123.us-east-1).
- 02** Inject your authentication token or JWT key pair string into the connection.
- 03** Ask your AI client a question, like 'Show me all tables in the Sales schema,' and it runs the command for you.

The bottom line is that your agent handles the complex connectivity details; you just talk to it about your data.

---

## Built For

Anyone who spends time in an IDE writing SQL against massive cloud datasets. This MCP targets data engineers frustrated by context switching, and analytics teams that need real-time schema validation before modeling.

### Data Engineer

Validates raw data landing spots or checks for internal environment changes using the `'list_stages'` tool directly from their coding window.

### Analytics Engineer (dbt)

Generates highly accurate SQL modeling by having the agent live check definitions and list all available tables with `'list_tables'`.

### Software Architect

Writes scripts to pull diagnostic query metrics or map out data flows without downloading heavy SDK kits locally.

---

## What Changes When You Connect

- 01** Eliminate context switching. You keep your AI agent and Snowflake data visible in one place, letting you validate commands against the live engine.

- 
- 02 Audit compute costs instantly. Use `list_warehouses` to see exactly which clusters are running, helping control expensive operational overhead.

---

  - 03 Map complex structures easily. Chain tools like `list_databases` , followed by `list_schemas` , and then `list_tables` until you have a full picture of your data lineage.

---

  - 04 Stay current on long jobs. If an ETL pipeline is running for hours, use `get_query_status` to check its progress without guessing.

---

  - 05 Build accurate models. By letting the agent examine table definitions via `list_tables` , you ensure your generated SQL has correct column names and data types.
- 

---

## Real-World Applications

### Diagnosing a Broken ETL Pipeline

An engineer notices a job failed overnight. Instead of checking dashboards, they ask their agent to use `list_warehouses` first. This confirms the compute cluster is active, and then they use `get_query_status` to see exactly why the last run timed out.

### Quickly validating a JOIN

A developer suspects two tables are linking incorrectly. They ask the agent to execute an `execute_sql` query that joins the fields, and the resulting output immediately shows if the join key is missing or wrong.

### Building a New Data Model

An analytics team needs to join customer data with geo-data. They prompt their agent to first list all schemas and then use `list_tables` to confirm the exact naming conventions before writing any SQL.

### Understanding Data Ingestion Paths

A data architect needs to know where raw files are landing. They prompt the agent to `list_stages` to see all internal and external paths, immediately confirming the correct source location for their next script.

---

# Patterns to Avoid

---

## Juggling browser tabs

### ✗ AVOID

Having to copy a schema name from the Snowflake website into your local IDE, then switching back to run the query.

### ✓ INSTEAD

Just talk to your agent. Tell it what you need—like 'list all schemas in master\_db'—and let it handle the API calls using `list_schemas` and running the result directly.

---

## Guessing table names

### ✗ AVOID

Writing a query that fails because you misremembered whether a column was named 'user\_id' or 'customer\_identifier'.

### ✓ INSTEAD

First, run `list_tables` inside the schema to get an accurate list of available tables. Then use that information to build your query with `execute_sql`.

---

## Over-complicating JOINS

### ✗ AVOID

Writing a massive multi-part SQL statement when all you really need is a small piece of diagnostic data.

### ✓ INSTEAD

Start simple. Use the agent to execute specific, targeted queries using `execute_sql` first. If that works, then try building out the full join.

---

## The Right Fit

Use this MCP if your primary bottleneck is schema discovery or running highly complex SQL against a live data cloud without leaving your current coding environment. You need to constantly validate column names, check resource usage ( `list_warehouses` ), and map out deep directory structures (using `list_databases` and `list_schemas` ). Don't use this if you only need simple read access or basic connectivity; those are handled by generic database connectors. If your goal is just file transfer, you might prefer a dedicated staging tool over one focused on query execution.

---

## Navigating massive data warehouses used to be an archaeological dig.

Today, if you need to check the schema for a table or run a complex report, your process involves opening the Snowflake web UI. You jump between database trees, manually click through schemas, and finally copy-paste the full query into a dedicated execution window. This constant switching kills flow.

With this MCP, you keep everything in your AI client. You just chat with your agent—'Show me all tables under the sales schema.' The results come back instantly, letting you build or debug queries without touching another browser tab.

---

## Snowflake Schema Mapping and Querying

You no longer have to manually run multiple commands just to map a data source. You can ask your agent to systematically list all databases, then traverse down through schemas using `list_schemas`, and confirm the table structure with `list_tables`—all in one conversation.

The result is immediate architectural clarity. Your agent gives you the full picture of what's available, allowing you to run accurate queries via `execute_sql` right away.

---

# Snowflake: 7 Tools for Data Cloud Management

These seven tools allow your agent to systematically discover, validate, execute against, and monitor every aspect of your Snowflake account's data structure.

#	TOOL	DESCRIPTION
01	<code>list_databases</code>	Retrieves a list of every database available within the Snowflake account.
02	<code>list_schemas</code>	Shows all schemas contained inside one specific database.
03	<code>list_tables</code>	Lists all tables that exist in a given schema.
04	<code>execute_sql</code>	Runs a specified SQL query against the Snowflake data cloud, prioritizing read-only statements.
05	<code>list_warehouses</code>	Shows all virtual computing warehouses associated with the account.
06	<code>list_stages</code>	Lists both internal and external data stages used for data loading.
07	<code>get_query_status</code>	Checks the status of a background or asynchronous query that is still running.

---

## See It in Action

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

**U** List all running virtual warehouses I can access in my Snowflake account.



I've called `list_warehouses` for your targeted Snowflake footprint.

**Running Clusters Detected:**

1. `COMPUTE_WH` – Size: X-Small | State: SUSPENDED
2. `ANALYTICS_WH` – Size: Large | State: STARTED
3. `ETL_LOAD_WH` – Size: Medium | State: STARTED

Would you like me to run an analytical snippet under `ANALYTICS_WH` now?

**U** Write a query to grab the top 5 most engaged users from our schema and execute it.



No problem! I constructed this query to target exactly that and executed it against the system:

```
SELECT user_id, email, activity_score FROM prod_schema.users ORDER BY activity_score DESC  
LIMIT 5;
```

**Snowflake Output Data:**

1. alice@example.com (Score: 980)
  2. john84@test.io (Score: 964)
  3. m\_reyes@corp.com (Score: 941)
- [...]

Do you want me to format this as a markdown table or save the dataset?

- U** Retrieve the schema mapping for the MASTER\_DB. I need to know all nested tables before doing table joins.



Using `list_schemas`, `MASTER_DB` contains main partitions:

1. `PUBLIC`
2. `ANALYTICS_GOLD`

Digging into `ANALYTICS_GOLD` using `list_tables` :

- `dim_customers` (Type: BASE TABLE)
- `dim_geography` (Type: BASE TABLE)
- `fct_sales_records` (Type: BASE TABLE)
- `vw_live_revenue` (Type: VIEW)

I am ready to help build your exact JOINS. For example, `fct_sales_records.customer_id` directly joins on `dim_customers.id`.

---

## Frequently Asked Questions

---

### 01 How does Snowflake MCP help me with data lineage?

The MCP lets your agent use a series of listing tools (``list_databases``, ``list_schemas``, etc.) to map the deep, hierarchical structure of all available data objects within your account.

### 02 Can I check if my long-running query is still active using Snowflake MCP?

Yes. You use the ``get_query_status`` tool to retrieve real-time updates on asynchronous queries, letting you know when they finish or fail.

### 03 What should I do if my compute warehouse is running too high?

You can run ``list_warehouses`` through the MCP to see all active clusters and their current status. This helps you manage costs by identifying idle or excessive resources.

### 04 How do I get a full list of tables in Snowflake using this MCP?

You first use ``list_databases`` to narrow down the scope, then use ``list_schemas``, and finally call ``list_tables`` within that specific schema to get every table name.

**05 Does Snowflake MCP require me to know complex SQL syntax?**

No. You tell your agent what you want in plain English, and it constructs the necessary query using the ``execute_sql`` tool for you.







---

# Go Live in 60 Seconds

Get your connection token from [cloud.vinkius.com](https://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
 <b>Claude AI</b>	Profile → Customize → Connectors → "+" → Add custom connector → Paste endpoint
 <b>Cursor</b>	Settings → Features → MCP Servers → "+ Add New MCP Server" → Type: SSE → Paste endpoint
 <b>VS Code</b>	Ctrl/Cmd+Shift+P → "MCP: Add Server" → add <code>"snowflake": { "url": "..." }</code>
 <b>Windsurf</b>	MCP Settings → <code>mcp_settings.json</code> → Add endpoint URL
 <b>ChatGPT</b>	Settings → Tools & plugins → Add MCP server → Paste endpoint
 <b>Gemini</b>	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

# Snowflake 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](https://cloud.vinkius.com) →

[vinkius.com](https://vinkius.com) · [support@vinkius.com](mailto: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 Snowflake. 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	Snowflake MCP
Server ID	019d760a-a4c4-72da-b8b9-a40866890fe6
Platform	Vinkius Cloud for AI Agents
Endpoint	<a href="https://edge.vinkius.com/{token}/mcp">https://edge.vinkius.com/{token}/mcp</a>

### 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/snowflake](https://vinkius.com/mcp/snowflake).