MCP SERVER | NO CODE | CLOUD HOSTED

Google BigQuery MCP

Run Complex SQL Queries on Petabytes of Data.

Google BigQuery connects your AI agent directly to massive data warehouses. You can run complex Standard SQL queries on petabytes of structured information without leaving your chat client. Use this MCP to inspect schemas, audit job runs, and analyze huge datasets conversationally.

D Quality Score 65/100

sql | data-warehouse | big-data | cloud-computing | data-pipelines | query-optimization

MAINTAINED BY VINKIUS CLOUD FOR AI AGENTS · JUNE 2026
vinkius.com/mcp/google-bigquery

SCAN TO VIEW

The infrastructure that powers AI agents in the real world.

Vinkius connects AI to the world's software through secure, enterprise-grade infrastructure — enabling real-world execution at scale, built on the Model Context Protocol (MCP).

Your AI Connections Run Through Vinkius Cloud

The world's largest managed MCP catalog

Vinkius is the cloud infrastructure where AI agents connect 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.

Google BigQuery MCP

7 tools available
Cloud-hosted on Vinkius

This MCP lets you treat your data warehouse like a giant spreadsheet that talks back. Instead of logging into the console just to run one query or check a column name, you talk to your agent, and it handles the heavy lifting against Google BigQuery. You can ask questions about customer behavior, operational metrics, or complex financial trends, and it writes, runs, and summarizes the exact Standard SQL needed.

It's like having a dedicated data analyst sitting next to you who knows every table structure and job status in your system. Need to know if last night's background pipeline finished correctly? You can list recent jobs and check the error traces instantly. This capability makes it invaluable for anyone needing quick validation against terabytes of rows.

When you connect this MCP via Vinkius, your agent gets full visibility across all your structured data—from discovering deep table column mappings to running complex aggregations over massive datasets purely through conversational prompts.

Core Capabilities

01 — Querying Structured Data

The agent executes explicit Standard SQL commands against your BigQuery dataset, allowing you to extract precise data subsets.

02 — Inspecting Database Structure

You can get detailed metadata on any specific dataset or table, including column types and partitioning logic.

03 — Listing Dataset Contents

The agent lists all active datasets within your GCP project so you know exactly where to start looking for data.

04 — Auditing Job Performance

You can list recent query jobs and retrieve detailed reports on job runs, including processing bytes and failure reasons.

One Click on Vinkius — From Prompt to Execution

Available at **vinkius.com/mcp/google-bigquery** — connect your AI agent in three steps.

01 Subscribe to this MCP and provide your GCP Project ID along with an active OAuth or Service Account Token.

02 Your AI client authenticates the connection and establishes access across your specified data warehouse environment.

03 You prompt your agent conversationally (e.g., 'Show me the top 3 countries...'), and it handles running the necessary Standard SQL query, presenting only the result.

The bottom line is that you interact with your massive database using natural language prompts instead of complex console commands.

Built For

This MCP is for anyone who has to answer questions about data but hates the administrative overhead of a traditional BI dashboard. If you're spending time clicking through GCP consoles just to validate a column name or check a job status, this is for you.

Marketing Analyst

You use it to translate vague business questions into optimized SQL queries that pull specific customer cohort data from massive user tables.

Data Engineer

You check for failing scheduled queries or explore undocumented columns on the fly, diagnosing pipeline issues without manual console navigation.

Backend Developer

You quickly confirm if application background processes successfully inserted required rows into staging tables, verifying data integrity before deployment.

What Changes When You Connect

01 Stop jumping between tabs. Instead of leaving your chat client to validate data constraints or summarize daily logs, you can use the agent's job listing tool to audit workloads right where you are working.

02 No more guessing column names. Use `get_table` to pull precise schema details for any table, confirming types and clustering logic before writing a single line of SQL.

03 Turn conversations into data. You can ask high-level questions (like 'What were the top 3 signups?') and let the agent translate that into optimized Standard SQL using `execute_query`.

04 Audit pipelines easily. If you suspect an overnight cron job failed, use `list_jobs` and then `get_job` to read the root cause trace directly—no need to open the GCP console.

05 Understand your data structure instantly. The agent can traverse nested datasets using `list_datasets`, mapping out the entire logical topology of your project.

Real-World Applications

Checking Pipeline Health

A Data Engineer notices a scheduled report is missing data. Instead of manually checking logs, they ask their agent to use `list_jobs` and check the most recent job's status via `get_job`. The agent immediately flags that the job failed due to an unrecognized column name.

Ad-Hoc Market Analysis

A Marketing Analyst needs a quick report on customer acquisition channels. They ask their agent, which uses `execute_query`, to write and run complex SQL joining multiple large tables to calculate the top revenue sources for the month.

Schema Discovery

A Backend Developer inherits a new database schema. Rather than spending hours clicking through documentation, they ask their agent to use `get_table` on the main user table, instantly providing the column mappings and data types needed for integration.

Project Mapping

A consultant is onboarding to a new client's data environment. They use `list_datasets` to get an overview of all available logical groupings in the project, quickly mapping out where different domains (finance, marketing, operations) store their records.

Patterns to Avoid

Assuming Data Location

✗ AVOID

A user assumes a table called 'user_data' exists but doesn't know which dataset it belongs to. They try running a query and get an error, wasting time searching the console.

✓ INSTEAD

First, ask the agent to `list_datasets`. Once you find the correct container, use `list_tables` on that specific dataset. This confirms the exact location before attempting any queries.

Ignoring Job Failures

✗ AVOID

A job fails overnight, but the user only gets a generic notification and doesn't know if it was a syntax error or a data issue.

✓ INSTEAD

Use `list_jobs` to find the recent run ID, then use `get_job`. This provides the full root cause trace, telling you exactly why the workflow halted.

The Right Fit

Use this MCP if your primary problem is accessing and querying massive amounts of structured data stored in Google BigQuery. You need to execute complex Standard SQL queries against petabytes of records, audit job performance, or validate schemas conversationally without opening a web console. Don't use it if you are dealing with unstructured text (use an NLP-focused agent) or if your data is spread across multiple, unconnected databases (you might need a more generalized ETL tool). If all you need to do is run one simple SELECT statement on a single file, a basic database connector might suffice. But for enterprise-grade, multi-table analysis and job auditing, this MCP is the right fit.

The Data Console Maze

Today, validating data often means logging into the GCP console. You click dataset names, then table names, then run a test query just to check if that column is partitioned correctly or if your last background job actually finished. It's slow, it’s manual, and you spend more time navigating tabs than analyzing insights.

With this MCP, those steps vanish. Tell your agent what data you need—whether you want to find the full schema details using `get_table` or check if a complex calculation ran successfully using `list_jobs`. You get the answer directly in your chat window.

Google BigQuery MCP: Conversational Data Access

You stop copying error messages and pasting them into a ticket. Instead, you ask your agent to use `get_job` when something goes wrong. It reads the full failure trace for you.

The difference is that data analysis stops being a sequence of clicks and starts being a conversation.

Google BigQuery: 7 Available Tools

These tools let you manage the structure and content of your data warehouse. Use them to list datasets, check table schemas, run custom SQL queries, or audit job histories.

#	TOOL	DESCRIPTION
01	`list_datasets`	Lists every dataset available in your active GCP project.
02	`get_dataset`	Retrieves detailed information about a single, specified BigQuery dataset.
03	`list_tables`	Lists all the tables contained within one specific dataset.
04	`get_table`	Gets the full metadata and schema details for any given BigQuery table.
05	`execute_query`	Runs an explicit Standard SQL command that you specify to pull data.
06	`list_jobs`	Lists recent execution jobs and run history within BigQuery for auditing purposes.
07	`get_job`	Retrieves comprehensive details about a specific, completed or failed job run.

See It in Action

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

U Get the table schema for `users_prod` in the `analytics` dataset.

Schema fetched. `users_prod` contains 12 columns, notably `user_id` (STRING, required), `signup_timestamp` (TIMESTAMP, partitioned), and `plan_tier` (STRING). Would you like to check some sample rows?

U Find out the top 3 countries with the most signups this month in the `users` table.

Executing SQL... The query completed successfully processing 12MB. The top 3 countries are: 1) United States (12,400), 2) Brazil (8,900), 3) Japan (4,150).

U Did the overnight cron job compute successfully or did it fail?

Listing recent jobs. Job `cron_transform_01` (ID: 11a0ab3-xxx) failed globally at 03:00AM. Root cause trace indicates native `Unrecognized name: user_account_id` syntax failure halting workflow explicitly mapped.

Frequently Asked Questions

01 **Can I query my data from different datasets using Google BigQuery MCP?**

Yes, the agent allows you to reference multiple datasets in a single prompt. As long as you have appropriate permissions and know the names, it can write cross-dataset queries for you.

02 **What if my SQL query is too complex? Will Google BigQuery MCP handle it?**

The agent handles writing and running Standard SQL. You just need to describe the desired outcome in plain language, and it writes the optimized code for execution.

03 **How do I check if a specific column exists using Google BigQuery MCP?**

First, use `get_table` on the relevant dataset. This tool provides the full schema map, letting you confirm every column name and its data type.

04 **Does this MCP help me troubleshoot failed data pipelines?**

Absolutely. Use `list_jobs` to see recent activity, then use `get_job` on the problematic ID to read the error trace directly, pinpointing the syntax or data issue.

05 **Is this MCP only for reading data?**

No. While it focuses heavily on querying and auditing, its structured nature allows you to confirm data integrity before building out write processes in your application's code.

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 `"google-bigquery": { "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

Google BigQuery 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 Google BigQuery. 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	Google BigQuery MCP
Server ID	`019d755c-2518-71c9-915e-6319a95da2ce`
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/google-bigquery.

© 2026 Vinkius · vinkius.com · support@vinkius.com