MCP Toolbox for Databases

GitHub

Introduction

An introduction to MCP Toolbox for Databases.

MCP Toolbox for Databases is an open source MCP server for databases. It enables you to develop tools easier, faster, and more securely by handling the complexities such as connection pooling, authentication, and more.

> Note
>
> This solution was originally named “Gen AI Toolbox for Databases” as its initial development predated MCP, but was renamed to align with recently added MCP compatibility.

Why Toolbox?

Toolbox helps you build Gen AI tools that let your agents access data in your database. Toolbox provides:

- **Simplified development**: Integrate tools to your agent in less than 10 lines of code, reuse tools between multiple agents or frameworks, and deploy new versions of tools more easily.
- **Better performance**: Best practices such as connection pooling, authentication, and more.
- **Enhanced security**: Integrated auth for more secure access to your data
- **End-to-end observability**: Out of the box metrics and tracing with built-in support for OpenTelemetry.

⚡ Supercharge Your Workflow with an AI Database Assistant ⚡

Stop context-switching and let your AI assistant become a true co-developer. By connecting your IDE to your databases with MCP Toolbox, you can delegate complex and time-consuming database tasks, allowing you to build faster and focus on what matters. This isn’t just about code completion; it’s about giving your AI the context it needs to handle the entire development lifecycle.

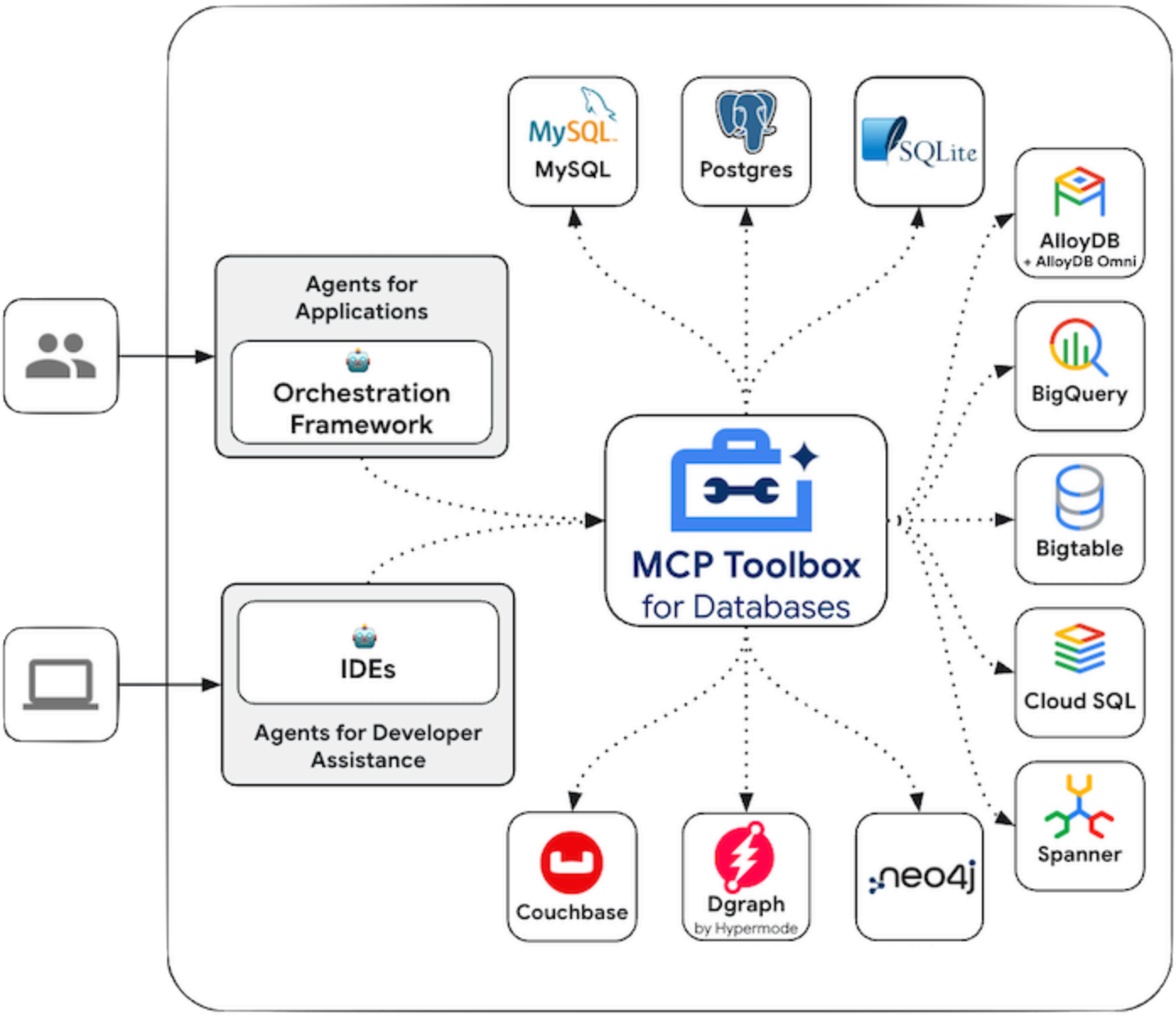
Here's how it will save you time:

- **Query in Plain English**: Interact with your data using natural language right from your IDE. Ask complex questions like, *"How many orders were delivered in 2024, and what items were in them?"* without writing any SQL.
- **Automate Database Management**: Simply describe your data needs, and let the AI assistant manage your database for you. It can handle generating queries, creating tables, adding indexes, and more.
- **Generate Context-Aware Code**: Empower your AI assistant to generate application code and tests with a deep understanding of your real-time database schema. This accelerates the development cycle by ensuring the generated code is directly usable.
- **Slash Development Overhead**: Radically reduce the time spent on manual setup and boilerplate. MCP Toolbox helps streamline lengthy database configurations, repetitive code, and error-prone schema migrations.

Learn how to connect your AI tools (IDEs) to Toolbox using MCP.

General Architecture

Toolbox sits between your application's orchestration framework and your database, providing a control plane that is used to modify, distribute, or invoke tools. It simplifies the management of your tools by providing you with a centralized location to store and update tools, allowing you to share tools between agents and applications and update those tools without necessarily redeploying your application.



Getting Started

Installing the server

For the latest version, check the releases page and use the following instructions for your OS and CPU architecture.

Binary | Container image | Compile from source

To install Toolbox as a binary:

```
# see releases page for other versions
export VERSION=0.6.0
```

```
curl -O https://storage.googleapis.com/genai-toolbox/v$VERSION/linux/amd64/to
chmod +x toolbox
```

Running the server

Configure a `tools.yaml` to define your tools, and then execute `toolbox` to start the server:

```
./toolbox --tools-file "tools.yaml"
```

You can use `toolbox help` for a full list of flags! To stop the server, send a terminate signal (`ctrl+c` on most platforms).

For more detailed documentation on deploying to different environments, check out the resources in the How-to section

Integrating your application

Once your server is up and running, you can load the tools into your application. See below the list of Client SDKs for using various frameworks:

Core | LangChain | Llamaindex

Once you've installed the Toolbox Core SDK, you can load tools:

```python
from toolbox_core import ToolboxClient

# update the url to point to your server
async with ToolboxClient("http://127.0.0.1:5000") as client:

    # these tools can be passed to your application!
    tools = await client.load_toolset("toolset_name")
```

For more detailed instructions on using the Toolbox Core SDK, see the project's README.

Last modified June 4, 2025: docs: add llms.txt file (#651) (46d7cdf4)