Distributed Master-Slave Database System with Web Interface

Overview:-
This project implements a distributed database system using a master-slave architecture in Golang.
It allows execution of SQL operations (SELECT, INSERT, UPDATE, DELETE) by slave nodes, while the master node supports full database control including CREATE/DROP databases/tables.
All slave queries are logged for audit.

Components:-
1. Master Node (Go):

Hosts REST APIs to manage databases, tables, and queries.

Logs all incoming queries from slaves.

Interfaces with a MySQL backend.

2. Slave Node (Go):

Receives query requests via HTTP and forwards them to the master.

Relays the response back to the UI.

3. Web Interfaces (Streamlit):

Master UI: Create/drop databases and tables, and view logs.

Slave UI: Enter and execute SQL queries interactively.

Features:-
Master-Slave architecture with centralized control.

Logs and tracks every slave query (with IP, table, database).

Base64-safe result decoding in UI.

Simple and intuitive web interface for both master and slave sides.

Technologies:-
Go (Golang)

MySQL

Streamlit (Python)

RESTful API communication

Use Cases:-
Distributed systems education

Lightweight SQL lab environment

Master-slave replication demo

Execution Steps
Start MySQL server on localhost:3306 with username: root, password: 12345678.

Run the master server:

go
Copy
Edit
go run master.go
Run the slave server:

go
Copy
Edit
go run slave.go
Launch GUIs:

arduino
Copy
Edit
streamlit run master.py
streamlit run slave_gui.py