Building a Distributed Database Using Go

Course Name: Distributed Database

Submitted To: Dr. Ibrahim ElSamman

Submitted By:

- Ahmed Mohamed Ahmed Mohamed CS1

- Abdelrahman Ali Zaher Abdo دواعي تخرج

- Moamen Bakr ElSedik Noaman CS4

1. Introduction

This project demonstrates a simplified distributed database system developed using the Go programming language. It is designed to showcase core principles of distributed systems, including replication, node communication, and failover prevention.

.

2.Objective

To build a minimal distributed database system in Go that:

- Supports real-time communication between master and slave nodes.
- Allows query execution from multiple clients.
- Demonstrates basic replication and fault tolerance mechanisms.

3.Technologies Used

Component	Technology
Language	Go (Golang), Python
DBMS	MySQL
Network	TCP Socket
GUI	Tkinter (Python)
Environment Config	.env using godotenv

4. System Architecture

- **Master Node**: Handles database creation, critical query execution, and broadcasting non-critical operations.

- **Slave Nodes**: Connect to the master and execute permitted queries. Receive and replicate data from master.

- **GUI Clients**: Tkinter-based interfaces for user interaction with master or slave nodes.

5. Functional Requirements

- Master can create/drop databases and tables.

- All nodes can insert, update, delete, and select records.

- Queries sent from GUIs or terminals.

- Queries validated against node role.

- Data replicated from master to slaves.

6. System Design and Code Explanation

a.master.go

- Starts TCP server and handles slave connections.
- Filters critical queries from unauthorized sources.
- Executes valid queries and returns results.
- Broadcasts non-critical queries to slaves.

b.slave.go

- Connects to master over TCP.
- Receives and executes commands from master.
- Executes valid local queries with constraints.

c.GUI

- `master_gui.py`
- `slave_gui.py`: Connects to master's IP and allows only limited queries.

d.env

Contains:

```
DB_USER=root
DB_PASS=rootroot
```

7.Features Implemented

✅ Master-Slave Communication over TCP
✅ Query Execution and Filtering
✅ Dynamic Table Creation
✅ Database-Level Security via `.env`
✅ GUI for User Interaction
✅ Data Replication

8.Database Design

Table	Columns
`student`	`id`, `name`
`course`	`id`, `name`
`student_course`	`student_id`, `course_id` (bridge table)

- Many-to-many relationship managed via bridge table `student_course`.