

Instituto Tecnológico de Costa Rica
Escuela de Ingeniería en Computadores
Algoritmos y Estructuras de Datos I (CE 1103)

TinySQLDb

Estudiantes

Eduardo José Canessa Quesada
Luis Felipe Chaves Mena

Asignatura

Proyecto II

Profesor

Leonardo Andres Araya Martinez

Periodo: II Semestre

Fecha: 10 de Octubre, 2024

Índice general

1	Introducción	1
1.1	Descripción del Problema	1
2	Descripción de la Solución	2
2.1	Arquitectura del Sistema	2
2.2	Funcionalidades Principales	2
2.3	Tecnologías Utilizadas	3
3	Diseño General	5

1 Introducción

El propósito de este proyecto consiste en el diseño e implementación de un motor de bases de datos relacional básico denominado TinySQLDb. Se tiene como objetivo la comprensión profunda del funcionamiento fundamental de los motores de bases de datos. Para ello, se utiliza el lenguaje de programación C# y se aplica el paradigma de orientación a objetos, lo que permite una estructura más clara y modular del código.

El sistema está compuesto por dos componentes principales: un cliente desarrollado en PowerShell y un servidor encargado de gestionar las consultas SQL. Este servidor integra diversas funcionalidades, incluyendo un procesador de consultas y un gestor de datos almacenados. A través de este diseño, se busca no solo facilitar la interacción entre el cliente y el servidor, sino también optimizar el procesamiento y la gestión de las operaciones de la base de datos.

Además, el proyecto permite experimentar con la ejecución de consultas SQL, la manipulación de datos y el análisis de resultados, así mismo para la implementación de varias funcionalidades se recurrió al uso de árboles binarios, con el fin de optimizar el manejo de datos en temas referentes con la búsqueda, el guardado y la eliminación de los datos conservados en el motor de bases de datos.

1.1. Descripción del Problema

En este proyecto, los estudiantes deben implementar un sistema gestor de bases de datos capaz de procesar y ejecutar un subconjunto de sentencias SQL. El sistema debe tener la capacidad de gestionar bases de datos modeladas como carpetas del sistema de archivos, con tablas representadas por archivos binarios. Se debe proporcionar un cliente en PowerShell que permita a los usuarios enviar consultas SQL al servidor, el cual manejará la lógica de procesamiento de consultas, validación y acceso a datos.

2 Descripción de la Solución

2.1. Arquitectura del Sistema

API Interface

Responsable de la comunicación entre el servidor y los clientes. Aquí se encuentra el archivo *Server.cs* que recibe solicitudes SQL y las dirige al procesador de consultas.

Query Processor

Se encarga de analizar, validar y ejecutar las consultas SQL. Aquí se encuentran archivos como *SQLQueryProcessor.cs* y *SQLSentenceProcessor.cs*, que transforman las consultas en operaciones sobre los datos almacenados

Stored Data Manager

Gestiona el almacenamiento de datos en archivos binarios, simulando tablas y columnas. Los datos se organizan y almacenan dentro de directorios que representan bases de datos, con archivos individuales que representan las tablas.

Binary Table Manager

2.2. Funcionalidades Principales

- CREATE DATABASE : En el directorio de *TinySQL* se creará otro directorio que será la base de datos con el nombre indicado según el comando.
- SET DATABASE : Se selecciona en el directorio de *TinySQL* la base de datos donde se desea trabajar según el nombre indicado por el comando.
- CREATE TABLE : Una vez se realice *SET DATABASE*, en la base de datos

elegida se creará una tabla con el nombre indicado y la cantidad de columnas con sus respectivos nombres. Esto se guardará en un archivo binario *.Table* con un formato específico.

- DROP TABLE : Una vez se realice *SET DATABASE*, se tomará el nombre de tabla por eliminar y se realiza la eliminación. Si no se encuentra se retona un *OperationStatus.Error* que marca que no se encontró el nombre del archivo.
- SELECT : Una vez se este en una base de datos, recibe nombre de un tabla y una serie de columnas, esta puede ser un ***, que represente todas las columnas, o los nombres separados solamente por comas. Luego se recibe 2 parámetros opcionales que son *whereClause* y *orderClause*, si *whereClause* es nulo entonces la operación se realiza sobre todas las columnas, de otra forma toma las filas que cumplan con la condición. *OrderClause* nos dice si se deben enseñar datos de manera ascendente o descendente en la tabla, ya que el objetivo de este comando es devolver valores al *client* en forma de tabla.
- UPDATE : Una vez se este en una base de datos, recibe nombre de un tabla, un *set statement* que nos indicará el nuevo valor de variable de una columna, y una *whereClause* que nos dirá cuales filas cambiaran su valor indicado por *set statement*, si *whereClause* es nulo, se modifican todas las filas.
- DELETE : Una vez se este en una base de datos, recibe nombre de un tabla, y una *whereClause*, y las filas que cumplan este *whereClause* seran eliminadas.
- INSERT INTO : Sobre una base de datos existente, se realiza una inmersión de datos que estarás asociados a una columna, esta inmersión será en el archivo binario mediante una clase especializada en la edición de las tablas. Con el fin de cumplir con este objetivo esta clase recorre el archivo, mientras a su vez extrae información pertinente a los datos para insertar con el formato adecuado los datos.

2.3. Tecnologías Utilizadas

1. Lenguaje de Programación - C# es el lenguaje principal utilizado para implementar la lógica del motor de bases de datos, aprovechando su capacidad para

manejar la manipulación de archivos binarios y su robusto sistema de tipos.

2. PowerShell 7 - Un cliente externo en PowerShell 7 permite ejecutar comandos SQL, facilitando la interacción del usuario con el sistema.
3. Sockets - La comunicación entre el cliente en PowerShell y el servidor TinySQLDb se realiza mediante sockets, permitiendo la ejecución de sentencias SQL remotas.
4. Archivos Binarios - El almacenamiento de datos se maneja mediante archivos binarios.

3 Diseño General

