Instituto Tecnológico de Costa Rica

Ingeniería en Computadores

Algoritmos y estructuras de datos 1

Grupo 02

Segundo proyecto II semestre

Documentación
“TinySQLDb”

Repositorio github:

https://github.com/StevenTEC295/BaseDatosProyecto2.git

Profesor: Leonardo Araya Martínez

Estudiantes:

Fiorela Sofía González Rubí - 2024211034

Steven Josué Pérez Aguilar - 2024118003

Año: 2024

Tabla de contenido

Introducción

Este proyecto tiene como objetivo desarrollar TinySQLDb, un motor de bases de datos relacional básico, con el fin de familiarizarse con los conceptos fundamentales detrás de estos sistemas y aprovechar el uso de conceptos de estructuras de bases jerárquicas para optimizar su funcionamiento. El proyecto implica la implementación de un cliente que permita ejecutar consultas SQL a través de PowerShell y un servidor que procese dichas consultas, almacenando y gestionando los datos de manera eficiente. El proyecto introduce conceptos como la creación y gestión de índices con árboles, optimizando el rendimiento en la búsqueda de datos.

El desarrollo de este sistema sencillo permitirá profundizar en el funcionamiento de los motores de bases de datos comerciales, como MySQL. A través de este proceso, se espera comprender mejor los desafíos y consideraciones que deben tenerse en cuenta al diseñar sistemas de almacenamiento de datos robustos y eficientes.

Descripción del problema

El problema que se plantea en este proyecto consiste en diseñar e implementar un motor de bases de datos relacional sencillo. El propósito es familiarizarse con los principios fundamentales de los sistemas de bases de datos, así como con la manipulación de datos mediante SQL.

El sistema debe incluir un cliente y un servidor, que interactúan mediante sockets. El cliente será un módulo de PowerShell que permitirá al usuario enviar consultas SQL a través de la línea de comandos, las cuales serán procesadas por el servidor. El servidor, por su parte, estará compuesto por varias capas: una interfaz de API encargada de gestionar la comunicación entre el cliente y el servidor, una capa de procesamiento de consultas para validar y ejecutar las sentencias SQL, y un gestor de datos almacenados que se encargará de acceder a los archivos de las bases de datos.

El sistema debe ser capaz de realizar operaciones básicas como la creación de bases de datos, tablas, y la inserción, selección, actualización y eliminación de datos. Además, se requiere la implementación de un sistema de índices, mediante árboles, que permita optimizar las consultas al reducir el tiempo de búsqueda de registros en las tablas.

El principal reto de este proyecto radica en la correcta implementación de los árboles utilizados para los índices, ya que estos son fundamentales para optimizar el rendimiento del sistema al realizar búsquedas y consultas en las tablas. Los índices deben estructurarse mediante árboles que permiten mapear los valores de las columnas a posiciones específicas en los archivos de las tablas. Esto evita realizar búsquedas secuenciales, que serían ineficientes en tablas de gran tamaño, al proporcionar acceso directo a los registros correspondientes.

En resumen, el problema se enfoca en la creación de un sistema de bases de datos relacional básico, que permita a los usuarios realizar operaciones SQL desde un cliente de PowerShell, mientras que el servidor gestiona la ejecución de estas operaciones y asegura el correcto manejo de los datos y los índices asociados.

Descripción de la solución

A continuación, se detallan los aspectos más relevantes de la implementación de TinySQLDb, desarrollado para este proyecto.

Implementación del servidor y la comunicación cliente-servidor

El servidor se encarga de recibir, procesar y responder a las consultas SQL que le envía el cliente a través de sockets. El servidor fue diseñado en C#, utilizando la clase Socket de .NET, y está compuesto por varias capas, cada una con responsabilidades bien definidas.

El servidor inicia escuchando en una IP y puerto específicos, esperando conexiones entrantes. Cuando un cliente envía una solicitud, el servidor recibe el mensaje, lo deserializa y lo convierte en un objeto de petición SQL. Luego el procesador de consultas se encarga de coordinar su ejecución y el servidor envía el resultado de vuelta al cliente.

Implementación del cliente en PowerShell

El cliente, implementado como un módulo de PowerShell, permite al usuario enviar consultas SQL y recibir respuestas en formato de tabla. El módulo utiliza sockets para conectarse al servidor, enviando las consultas y esperando las respuestas correspondientes. La función Execute-MyQuery se encarga de gestionar todo el flujo, desde la lectura del archivo SQL hasta la visualización de los resultados.

Implementación del procesamiento de consultas

El código se ha escrito para interpretar las sentencias SQL y ejecutar cada una de las operaciones. Esto incluye el manejo de CREATE TABLE, INSERT, SELECT, UPDATE, y DELETE, entre otras. La capa valida que las consultas sean correctas utilizando el system catalog.

Implementación de almacenamiento de datos

El sistema almacena las bases de datos y tablas como archivos binarios en el sistema de archivos. Cada tabla se guarda en un archivo separado, y la información sobre las bases de datos y sus estructuras se almacena en el system catalog. Este catálogo permite validar las consultas SQL antes de ejecutarlas, verificando la existencia de bases de datos, tablas y columnas.

Implementación de índices para optimización de consultas

Para optimizar las consultas de búsqueda, se implementaron índices utilizando árboles. Estos índices se crean sobre columnas específicas de las tablas y permiten realizar búsquedas más rápidas al evitar la necesidad de recorrer secuencialmente todos los registros de la tabla.

Visualización de los resultados

Los resultados de las consultas se presentan en el cliente en formato de tabla utilizando las capacidades nativas de PowerShell. Esto facilita la visualización de los datos y proporciona una experiencia de usuario más intuitiva.

Diseño general