
PROYECTO 1

201902714 – KEVIN JOSUÉ CALDERÓN PERAZA

Resumen (150 palabras)

Es una base de datos cuyos datos están almacenados en la memoria principal para facilitar tiempos más rápidos de respuesta. Los datos de origen se cargan a la memoria del sistema en un formato comprimido no relacional.

Las bases de datos en memoria optimizan el trabajo relacionado con el procesamiento de consultas. La definición anterior, es lo que se realizó en este proyecto por medio de estructuras de datos, agregando así funcionalidades extras y utilizando librerías nativas del lenguaje de programación Python para obtener los datos y la representación gráfica por medio de un grafo. Tomando en cuenta distintas restricciones para dar un uso correcto a las estructuras tener persistencia sobre los datos almacenados en memoria.

Palabras clave

Estructura de datos

Matrices

Grafo

Lista

Nodo

Abstract

It is a database whose data is stored in main memory to facilitate faster response times. The source data is loaded into system memory in a non-relational compressed format.

In-memory databases optimize the work related to query processing. The previous definition is what was done in this project with data structures, this adding extra functionalities and using native libraries of the Python programming language to obtain the data and the graphic representation by means of a graph. Taking into account different restrictions to give a correct use to the structures, have persistence over the data stored in memory.

Keywords

Data Structure

Arrays

Graph

List

Node

Introducción

Cuando se almacena información en una base de datos en memoria, hay que distinguir entre almacenamiento de datos basado en columnas y en filas, aunque algunos sistemas utilizan ambas variantes de almacenamiento. Las bases de datos orientadas a filas organizan los registros de datos recopilados en una fila. El registro de datos en columnas recibe el nombre de **Column Store Format**. Al almacenar juntos los datos con valores idénticos, el sistema minimiza el volumen total de datos, por lo que el espacio de almacenamiento y los tiempos de transmisión se reducen.

En este caso el proyecto consta de esto, haciendo una simulación de base de datos en memoria, sabiendo aplicar estructuras de datos y librerías externas como lo es de **minidom** y **graphviz** para tener una mejor interpretación de los datos que nos están presentando.

Desarrollo del tema

Las bases de datos se han convertido en una herramienta poderosa en el desarrollo de la ingeniería del software, ya no podemos imaginar aplicativos sin tener base de datos. Almacenar información en base de datos, validarla y consumirla es importante, pero buscar la optimización del uso de base de datos se ha vuelto imperativo en estos últimos tiempos.

Por lo tanto, el proyecto que se realizó, se tomo en cuenta varias formas de presentar las matrices y que en este caso representan las **tuplas** y **sitios**.

En la primera parte se tomo un menú con varias opciones, en la primera opción: por medio de un **XML** se tomó las etiquetas matrices y datos, así como sus propiedades, así como sus propiedades ya que hay varias excepciones a tomar en cuenta, para tener persistencia correcta de los datos almacenados en las listas enlazadas. Esto se realizo por medio de la librería **minidom** de Python, a pesar de tener varias librerías

disponibles o realizar un analizador léxico, la interpretación de la librería es muy fácil de entenderle y amigable a cualquier programador que la utilice.

La librería **minidom**, es una implementación mínima de la interfaz **Document Object Model**, con una API similar a la de otros lenguajes. Está destinado a ser más simple que el **DOM** completo y también significativamente más pequeño. Los usuarios que aún no dominan el **DOM** deberían considerar el uso del módulo **xml.etree.ElementTree** para su procesamiento XML. Y para utilizarla simplemente se importa como:

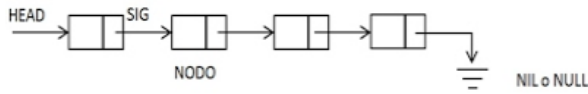
“from xml.dom import minidom”

Para almacenar la información se utilizaron dos tipos de listas en este proyecto se utilizaron la lista enlazada circular y la lista enlazada simple.

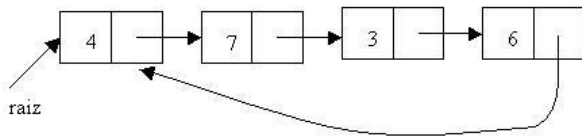
La lista enlazada simple, es la lista mas fundamental estructura de datos basada en punteros, y del concepto fundamental de esta derivan las otras estructuras de datos. Esta asigna espacio para cada elemento por separado, en su propio bloque de memoria, llamado **Nodo**. La lista conecta estos nodos usando punteros, formando una estructura parecida a una cadena.

Un nodo es un objeto como cualquier otro, y sus atributos serán los encargados de hacer un trabajo de almacenar y apuntar a otro nodo. Cada nodo tiene dos atributos: **dato**, en este caso es un objeto tipo Matriz, pero puede ser cualquier tipo de dato (int, string, arreglos, objetos, etc...) y el otro atributo siguiente o **next**, usando para hacer referencia al siguiente nodo de la lista.

Y esta puede ser opcional, pero para este proyecto se utilizó es usar un tercer atributo llamado size, que es para determinar el tamaño de la lista. Utilizado comúnmente para hacer ciclos y ponerle un fin.



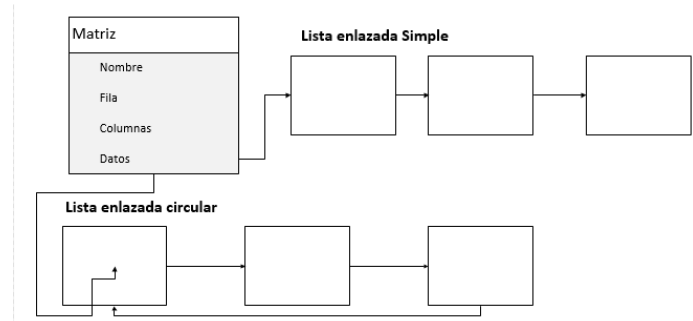
Esto básicamente se basa las demás listas, en el caso de la lista circular en es que cambia es que el ultimo nodo de la lista o elemento agregado apunta al primer elemento de la lista, simulando un círculo o no teniendo un fin.



Por otro lado, en otra de las funcionalidades se tomaron al calcular la matriz reducida fue ir buscando mediante una matriz equivalente de unos y ceros, ir cada uno de los elementos en el caso que fuese igual, si no es igual se almaceno en una lista temporal que luego va a hacer una lista de matriz de frecuencia.

Para el apartado de la grafica de la matriz de frecuencias se utilizó una función para la linealización de matrices de n dimension, en este caso se utilizó la linealización por columnas para poder asociar los nodos a los siguientes de manera que se pueda visualizar como una matriz de dos dimensiones. Es muy utilizada para la búsqueda de datos dentro de matrices, ya que el orden de complejidad baja de orden n a un orden lineal

A continuación, se le presenta un grafico como esta la estructura del proyecto, a partir de una clase Matriz, modelando los datos que incluye el archivo XML. Y el atributo de la clase Matriz se guarda en una lista enlazada simple y cada objeto de la clase Matriz se guarda en una lista Enlazada Circular.



También se tomo en cuenta el versionamiento en este proyecto. Se utilizó git para el control de versiones del proyecto y GitHub para tener almacenar en la nube el proyecto. Las releases de los proyectos se hicieron por medio de tags para tener un control de versión específica dependiendo el tipo de cambio que se hizo.

Por ejemplo: v.X.Y.Z

La X marcará la versión mayor. Es una versión que incluye cambios importantes en el funcionamiento de la aplicación.

La Y marcará una versión media. Es una versión menor que añade funcionalidades de mantenimiento o actualización de funcionalidades que ya existían.

La Z marcará una versión menor. Es una versión que esta incluye corrección menor que no afectan el funcionamiento actual del sistema, pero tampoco es una actualización que mejoré el proyecto.

Conclusiones

1. El uso de archivos XML, simplifica y separa datos de manera que al compartir información e intercambiar datos, hace que los datos estén disponibles más fácilmente.
2. El transporte de información por medio de un archivo XML hace que el proyecto sea escalable.
3. El uso de listas enlazadas en el proyecto hace que el uso de memoria se adapte dinámicamente al número de datos almacenados en la lista en cada momento.
4. Las listas enlazadas, sin importar el tipo, son dinámicas. Por lo que conforme sea necesario puede aumentar o disminuir la longitud de la lista.
5. La visualización de las matrices por medio de graphviz hace que se interprete de mejor manera los objetos que se almacena en la lista, así como los atributos del objeto.

- *Wikipedia*. (23 de 10 de 2020). Recuperado el 07 de 03 de 2021, de https://es.wikipedia.org/wiki/Base_de_datos_en_memoria

Referencias bibliográficas

- *Digital Guide IONOS*. (07 de 03 de 2019). Recuperado el 07 de 03 de 2021, de <https://www.ionos.es/digitalguide/hosting/cuestiones-tecnicas/base-de-datos-en-memoria/>
- *Grimpi IT Blog*. (22 de 11 de 2008). Recuperado el 07 de 03 de 2021, de <https://grimpidev.wordpress.com/2008/11/22/patrones-de-acceso-a-datos-active-record/>