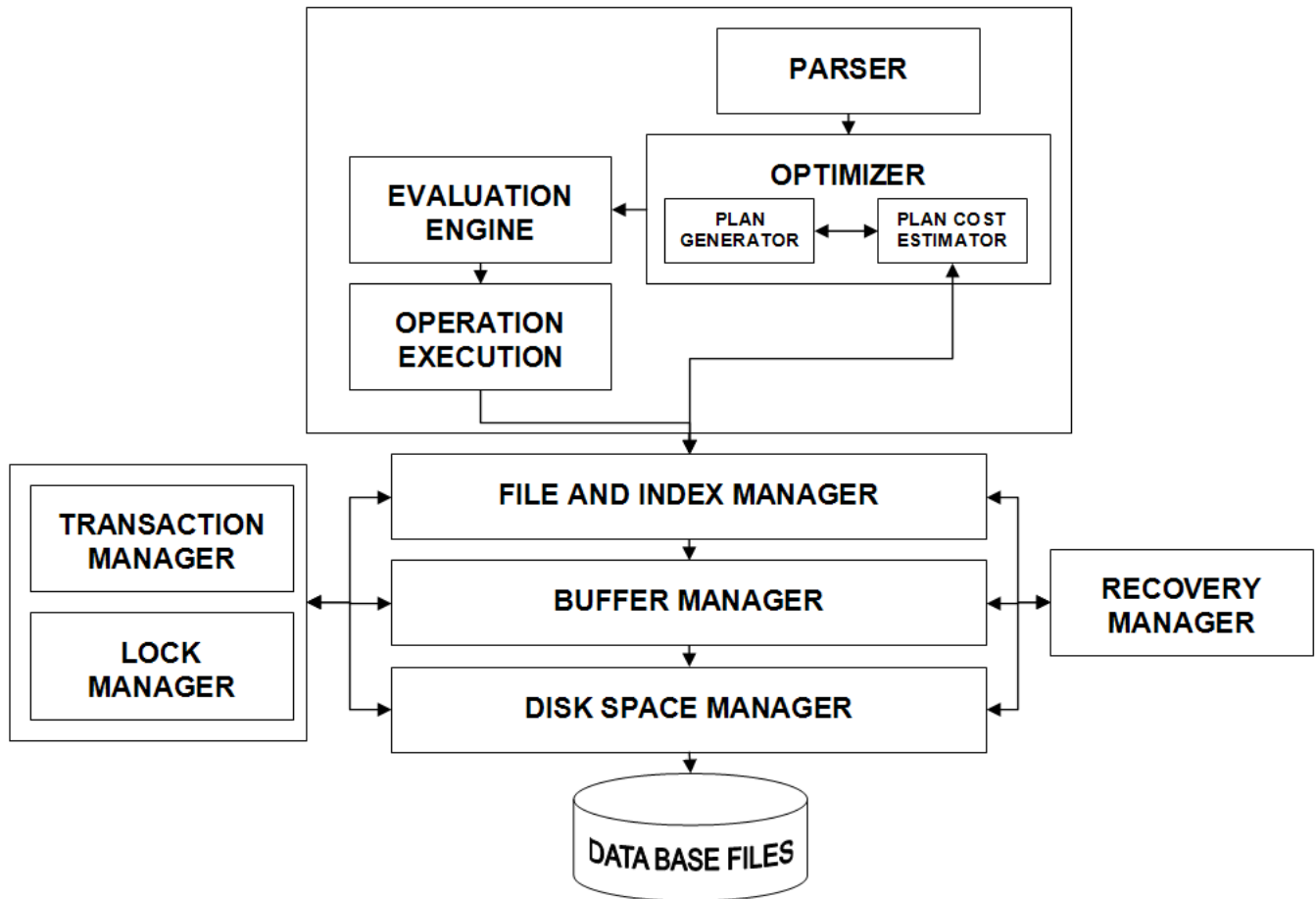


0.1. Arquitectura de MiniBase



La arquitectura del código de Minibase en C++ contiene más funcionalidad, con lo cual más componentes, que el código en Java. Como se puede ver por el diagrama, Minibase consta de un modulo de optimización, que sirve también de punto de acceso a las bases de datos utilizando SQL, y un modulo de bajo nivel que ocupa de abstraer los archivos de bases de datos y ofrecer índices al modulo de optimización.

La versión en Java ofrece una funcionalidad muy limitada, solamente están presentes **FILE AND INDEX MANAGER**, **BUFFER MANAGER**, **DISK SPACE MANAGER**, **OPERATION EXECUTION** y los archivos de la base de datos, incluyendo el catálogo (**DATA BASE FILES**). Como parte del trabajo practico se implementó una mínima parte del parser (carga batch de tablas). El resto de las componentes no están implementadas, como manejo de transacciones, locks, recuperación, parser SQL y optimizador. Estas componentes son esenciales para una bases de datos moderna.

0.1.1. Interacción entre componentes

Las componentes implementadas interactúan entre si principalmente por medio del pattern iterator. Este pattern ofrece una interfaz simple para recorrer los datos de una tabla, de un índice o del resultado de una operación como un join.

La componente de más bajo nivel es el **DISK SPACE MANAGER** que se ocupa de manejar los archivos en el disco. Toda interacción con el **DISK SPACE MANAGER** se hace indirectamente utilizando el **BUFFER MANAGER**. Esta componente interactúa con el **DISK SPACE MANAGER** a nivel de paginas, creando una cache para agilizar el acceso a estas.

El **FILE AND INDEX MANAGER** implementa la abstracción **HeapFile** para poder interactuar con el **BUFFER MANAGER** a nivel de tuplas. También permite la utilización de iteradores para recorrer una tabla. Otra estructura que implementa es el **B+Tree** para el manejo de índices. Esta componente se encarga de actualizar el catalogo cuando se introduce algún cambio a la base de datos.

Por ultimo esta la componente **OPERATION EXECUTION** que implementa las operaciones básicas de manejo de una base de datos y utiliza el pattern iterator para interactuar con los índices y con las tablas. También es la encargada de hablar con el catalogo para la manipulación de las tablas.