RookDB Design Document

Storage Manager for RDBMS

Introduction to RookDB

RookDB is a disk-oriented database management system (DBMS) aimed at exploring the internal architecture of modern database engines, with a particular focus on the design and implementation of the **Storage Manager** of DBMS. The system follows a **Relational Database model**, similar to widely used relational DBMS such as PostgreSQL and MySQL.

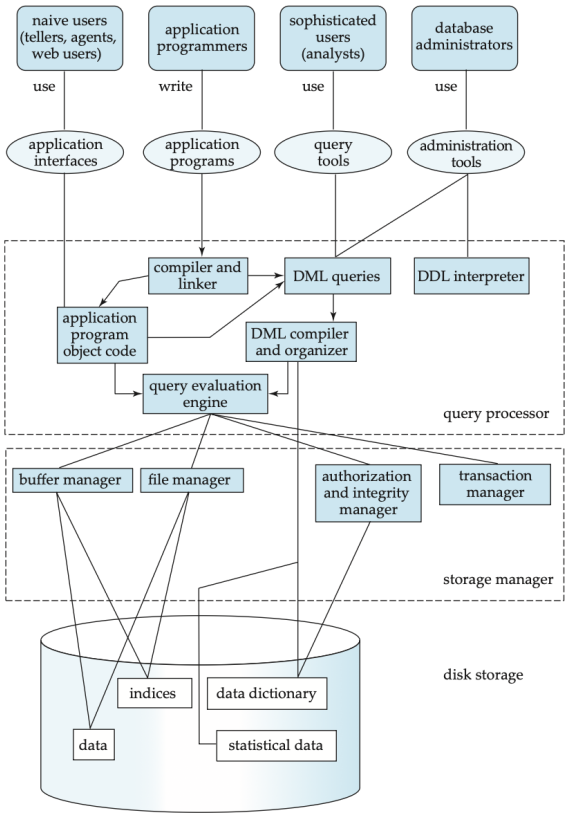


Figure 1: Relation Model DBMS Architecture

Based on the relational database architecture shown in the above figure, the primary objective of RookDB is to implement the key components of the storage manager that operate between the query processor and the underlying disk storage.

RookDB Architecture

The architecture of RookDB follows a layered design that separates logical metadata management from the physical representation of tables and the low-level organization of records within pages.

The storage manager in RookDB is broadly divided into the following four layers:

- Catalog Layer
- Table Layer
- Page Layer
- Buffer Manager Layer

Catalog Layer

The Catalog Layer is responsible for managing logical metadata required by the database system.
It maintains information about databases, tables, and column schemas. In RookDB, catalog metadata is stored in a JSON format.

For a detailed description of the catalog structure, refer to the Database Design Document available in the docs section of the GitHub repository.

Table Layer

In RookDB, each table is stored as a dedicated file within the directory corresponding to its parent database. This hierarchical organization mirrors the structure of the catalog and provides a deterministic mapping from logical table identifiers to their physical storage locations on disk.

Each table file is logically divided into two distinct regions. The first region is a fixed-size table header occupying the initial 8 KB of the file. This header stores metadata required for table management, currently stores only the total number of allocated pages. The second region consists of a sequence of fixed-size data pages, each 8 KB in size, which store tuple data along with associated slot metadata.

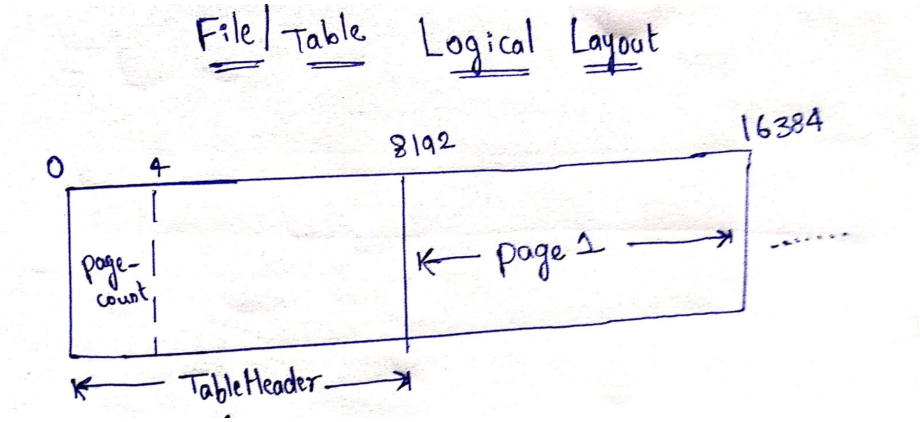
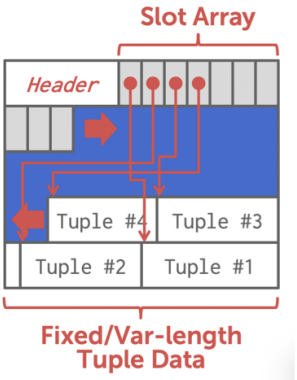
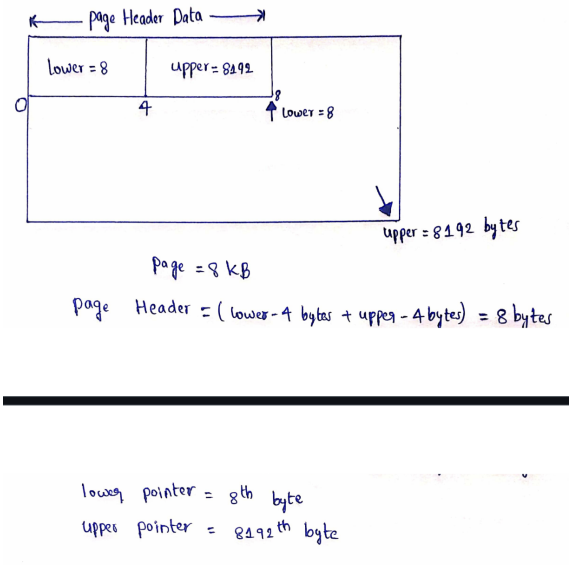


Figure 2: Logical Layout of a Table File

Page Layer

The Page Layer defines the internal layout and organization of records within a fixed-size page.
RookDB adopts a slotted-page structure inspired by PostgreSQL, consisting of a page header, an item identifier array, and tuple data.

Page Layout	Page Header Structure
	

Buffer Manager Layer

The Buffer Manager Layer maintains an in-memory cache of pages to minimize disk I/O and efficiently support data loading and manipulation.