ROOKDB

Storage Manager for relational DBMS

Data Systems Project

Introduction

Database Management System (DBMS):

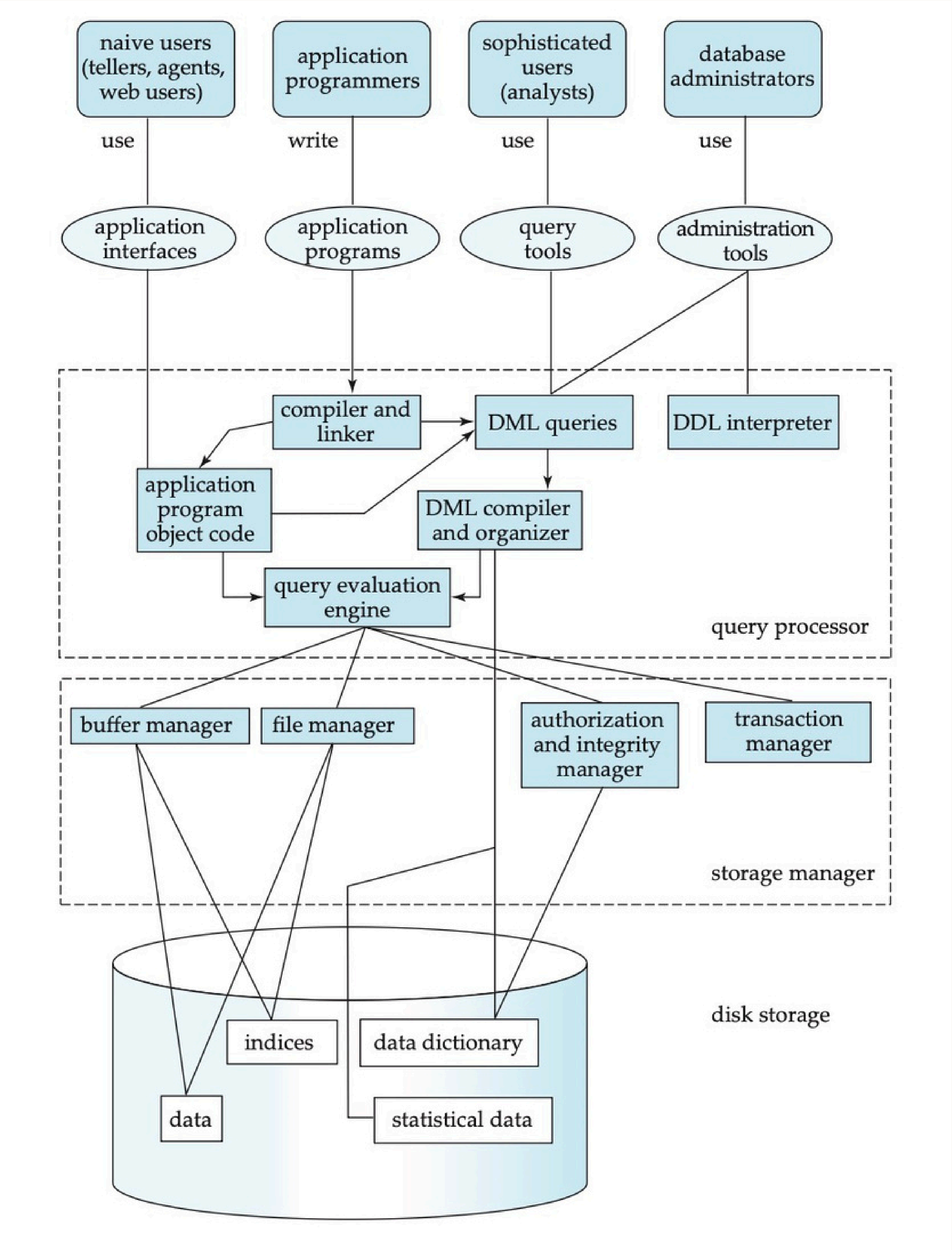
- A software system that enables users to define, create, maintain, and control access to a database.
- **Examples**: MySQL, PostgreSQL, MongoDB

Major Types of DBMS (Based on **Data Model**):

- Relational DBMS (RDBMS):
  - Data organized in tables (rows and columns)
  - Ex: MySQL
- Non-Relational (NoSQL) DBMS:
  - Data stored using flexible models (document, key-value).
  - Ex: MongoDB

RDBMS

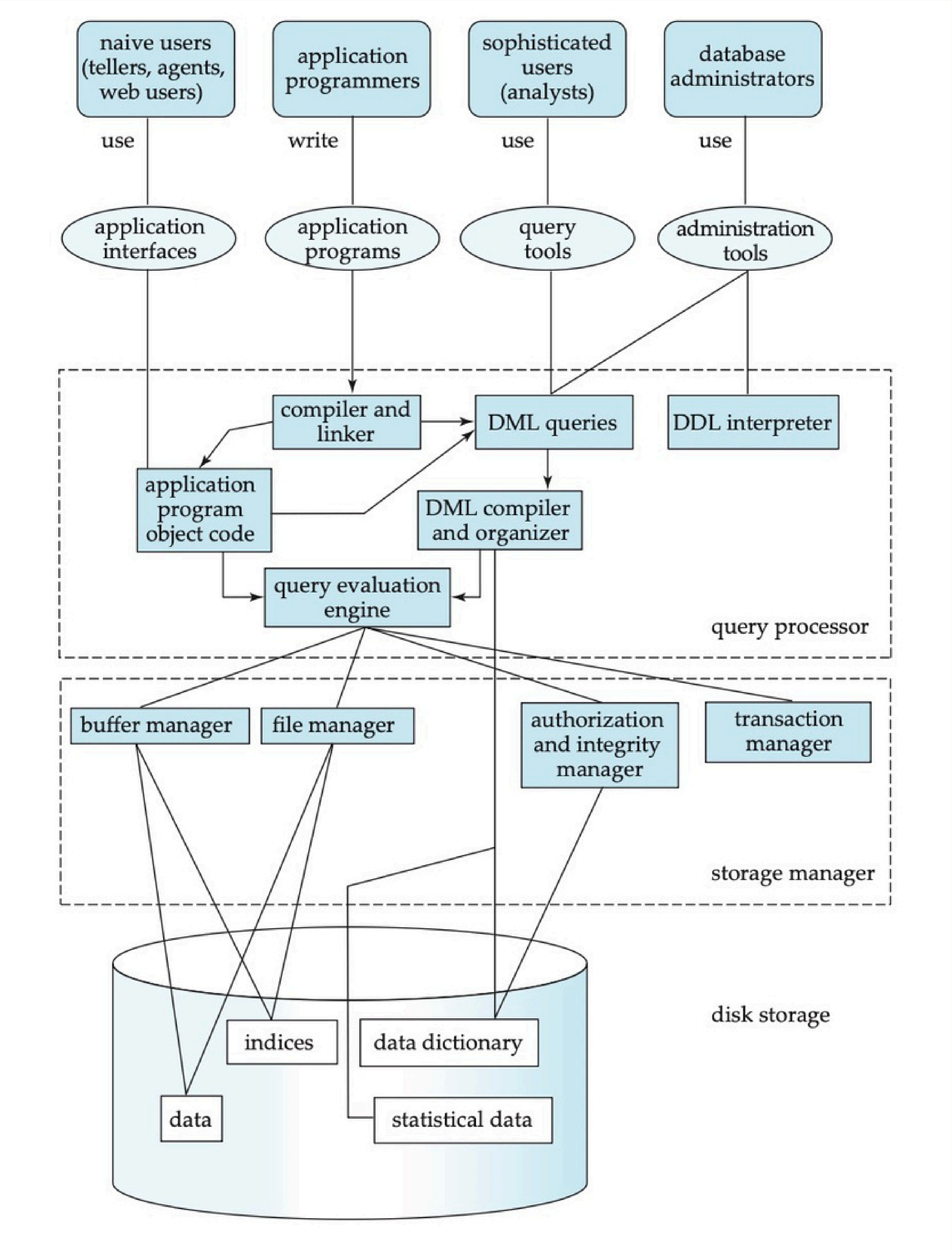
- Flow of SQL Queries
    a. Users or applications issue SQL queries
    b. Queries are processed by the **query processor**
    c. Execution requests are sent to the **storage manager**
    d. Data is read from or written to **disk storage**



ARCHITECTURE OF RDBMS

ABOUT ROOKDB

- **Storage Manager**
  - Responsible for efficiently organizing and managing the persistent storage of data.
  - Acts as an intermediate layer between the query processor and disk
- RookDB is implementing Storage Manager for RDBMS. The design of the RookDB Architecture is primarily inspired by PostgreSQL.



ARCHITECTURE OF RDBMS

CURRENT PROJECT STATUS

- Defined the logical layout of the system, including:
  - Catalog structure for managing metadata
  - Table definitions and row organization
  - Support for organizing and managing multiple databases.

- Supported operations:
  - Create Database
  - Create Table
  - Insert Rows into tables
  - Show Tuples
  - Show Databases
  - Select Database
  - Show Tables
  - Statistics of the table

CURRENT LIMITATIONS

- Supports only INT and CHAR(n) data types for table columns.
    - Not handled Primary Attributes, Null Values
    - Not implemented VARCHAR, Dates, Floating etc.,

- Buffer Manager is partially implemented.

- The catalog structure is simplified and does not fully align with those used in modern RDBMSs.

- The codebase is not fully implemented; however, it offers a well-structured starting point for extending different system components or features.

PROJECT EVALUATION

- This is an individual project.

- Students must propose the components they intend to implement as part of the project, which will be finalized jointly by the student and the TA.

- Evaluation will be based on design choices, correctness, feature completeness, code structure.

- Students are required to submit biweekly progress reports.

- Viva will be conducted as part of evaluation.

Code and Design Docs are available in:

https://github.com/RookDB/RookDB

THANK YOU