T54
DATABASE MANAGEMENT SYSTEM

UNIT-I

Introduction: Introduction to Database Systems: Overview – Data Models – Database System Architecture – History of Database Systems.

Entity-Relationship Model: Basic Concepts – Constraints – Keys – Design Issues – Entity Relationship Diagram – Weak Entity Sets – Extended E-R Features – Design of an E-R Database Schema.

What is Data

- Data can be facts related to any object
- Raw facts can be processed by computing machine.
- Collection of facts from which conclusions may be drawn.
- Data can be represented in the form of:
  - numbers and words which can be stored in computer’s language.

What is Information?

- Systematic and meaningful form of data.
- Knowledge acquired through study or experience.
- Information helps human beings in their decision making.

Database

- DBMS contains information about an enterprise
  - Collection of interrelated data
  - Set of programs to access the data
  - An environment that is both convenient and efficient to use

Database Applications

- Banking: all transactions
- Airlines: reservations, schedules
- Universities: registration, grades
- Sales: customers, products, purchases
- Online retailers: order tracking, customized recommendations
- Manufacturing: production, inventory, orders, supply chain
- Human resources: employee records, salaries, tax deductions

- Databases touch all aspects of our lives

What is DBMS ?

- DBMS - Database Management System.
- DBMS is a software system for creating, organizing and managing the database.
- It is an environment to the user to perform operations on the database for creation, insertion, deletion, updating and retrieval of data.

Purpose of Database Systems

- In early days, database applications were built on the top of the file systems
- Drawbacks of using file systems to store data:
  - Data redundancy and inconsistency
    - Multiple file formats, duplication of information in different files
  - Difficulty in accessing data
    - Need to write a new program to carry out each new task
  - Data isolation — multiple files and formats

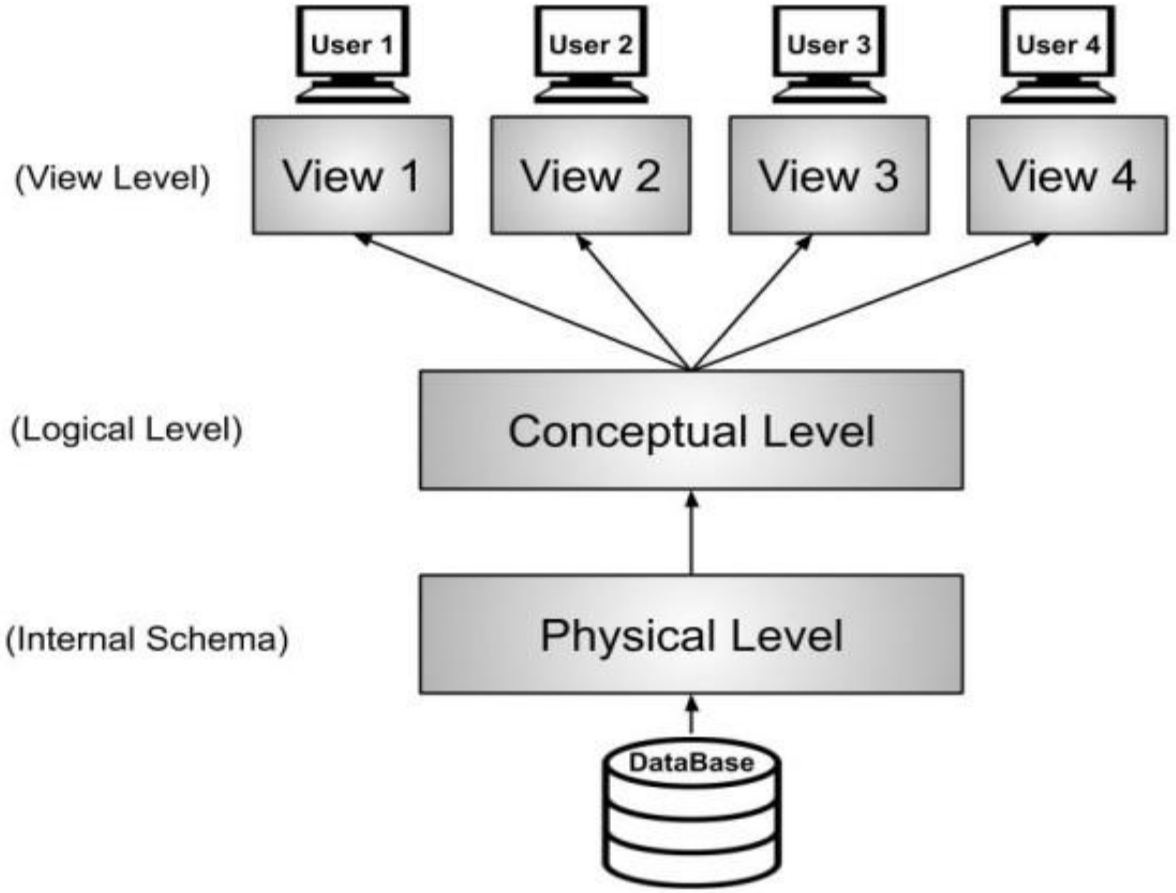
Purpose of Database Systems

- Integrity problems
  - Integrity constraints (e.g. account balance > 0) become "buried" in program code rather than being stated explicitly
  - Hard to add new constraints or change existing ones
- Atomicity of updates
  - Failures may leave database in an inconsistent state with partial updates carried out
  - Example:
    - Transfer of funds from one account to another should either complete or not happen at all

Purpose of Database Systems

- Concurrent access by multiple users
  - Concurrent accessed needed for performance
  - Uncontrolled concurrent accesses can lead to inconsistencies
    - Example: Two people reading a balance and updating it at the same time
- Security problems
  - Hard to provide user access to some, but not all, data
- Database systems offer solutions to all the above problems

Level of Abstraction



Levels of Data Abstraction

Level of Abstraction

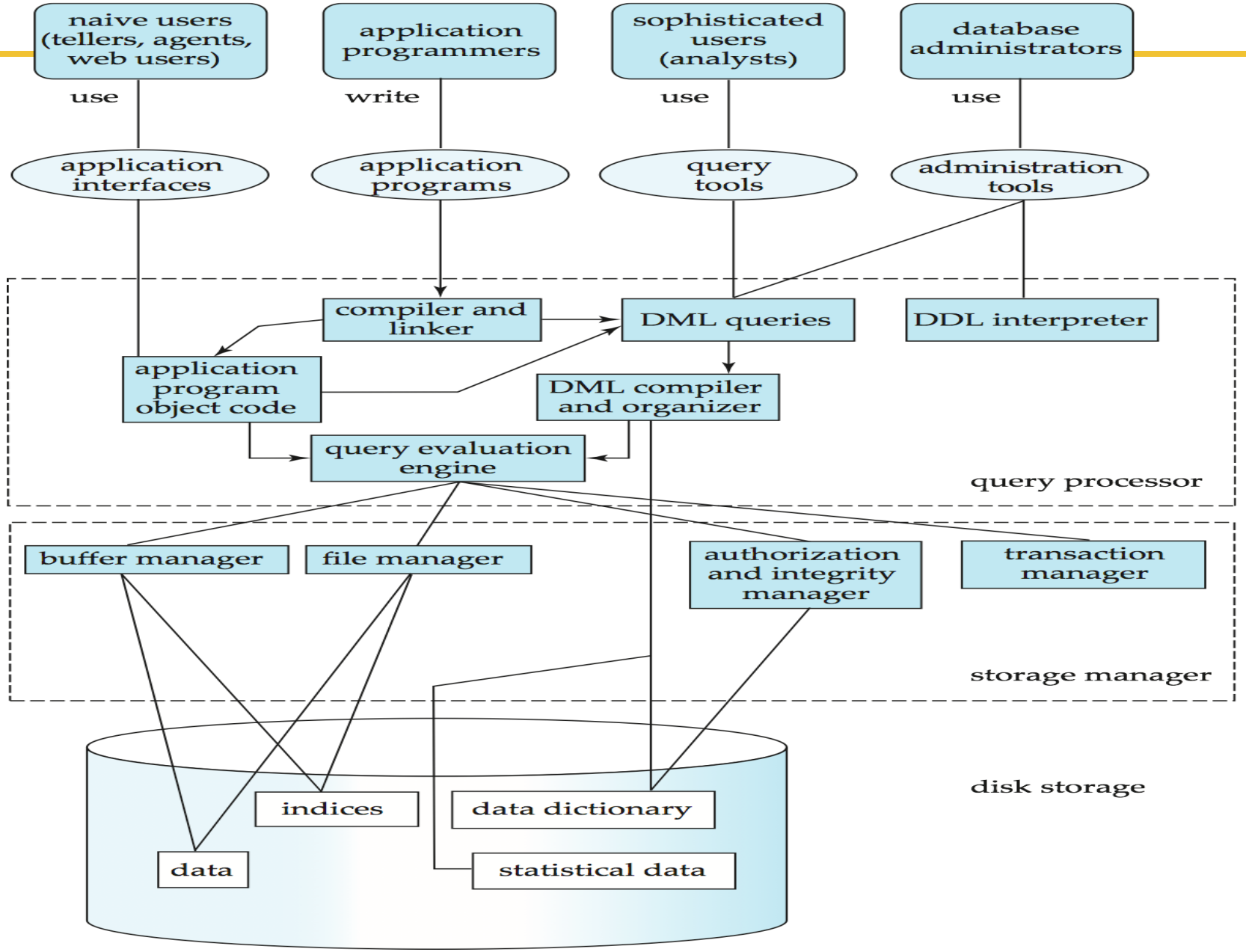
- Physical: This is the lowest level of data abstraction. It tells us how the data is actually stored in memory.
- Logical: The next level of data abstraction. It tells us what data is actually stored in the database in the form of tables. It also stores the relationship among the data entities in relatively simple structures.
- View: This is the highest level of abstraction. Only a part of the actual database is viewed by the users. This level exists to ease the accessibility of the database by an individual user.

Instance and Schema

- Schema – the overall logical structure of the database
  - Example: The database consists of information about a set of customers and accounts and the relationship between them)
- Instance – The data stored in database at a particular moment of time is called **instance** of database.
- Physical Data Independence – the ability to modify the physical schema without changing the logical schema

Database Architecture

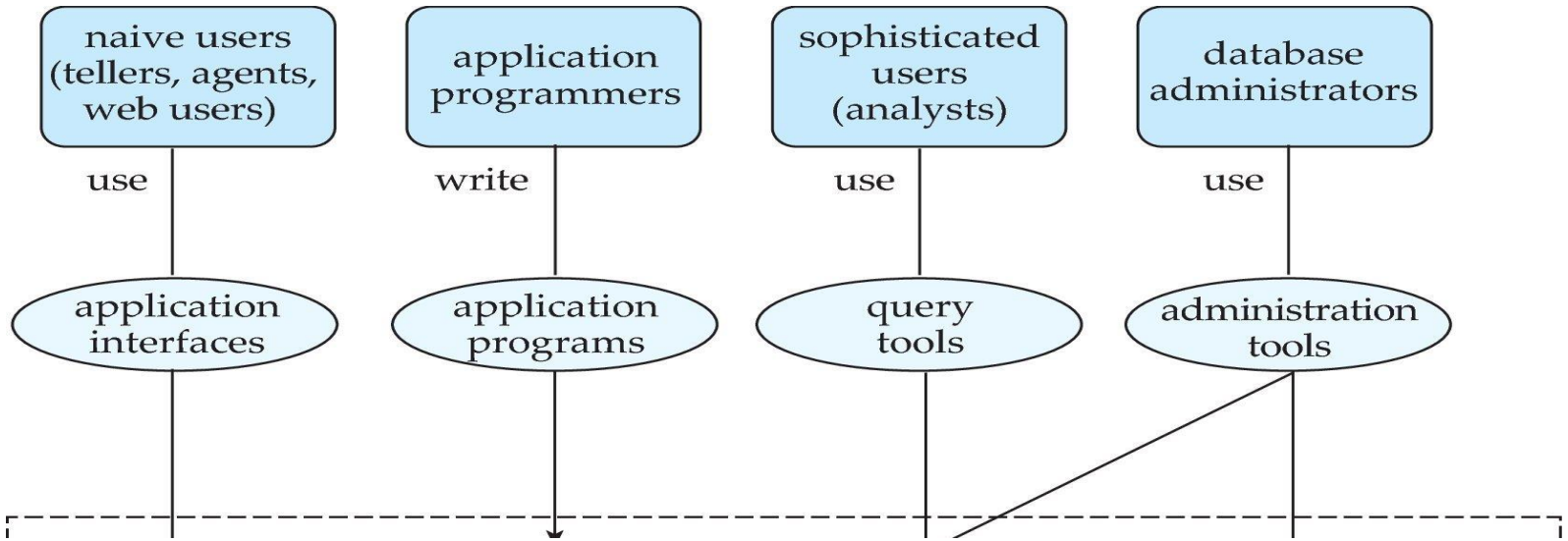
Database Architecture



Database Architecture

- Has 4 Parts
  - Database Users
  - Query Processor
  - Storage Manager
  - Disk Storage

Database-Users



Naïve User

Bank Tellers, Reservation Clerks
Data Entry Operators
Interacts the DB through Application
Not aware of the presence of DB System

Sophisticated User

Scientist, Business Analyst, Engineers
Aware of the presence of DB System
Interact the system without writing programs
Use SQL to write Queries to Perform operations

Application User

Develops an application programs or UI
Using high level language
Application Program access the database

Specialized User

Develops a specific application programs
These application don't fit into traditional data processing
Like – CAD, Multimedia DB Programs

Database Users

Database Administrator

•Single Person or Team

•Responsible for managing the whole database system.

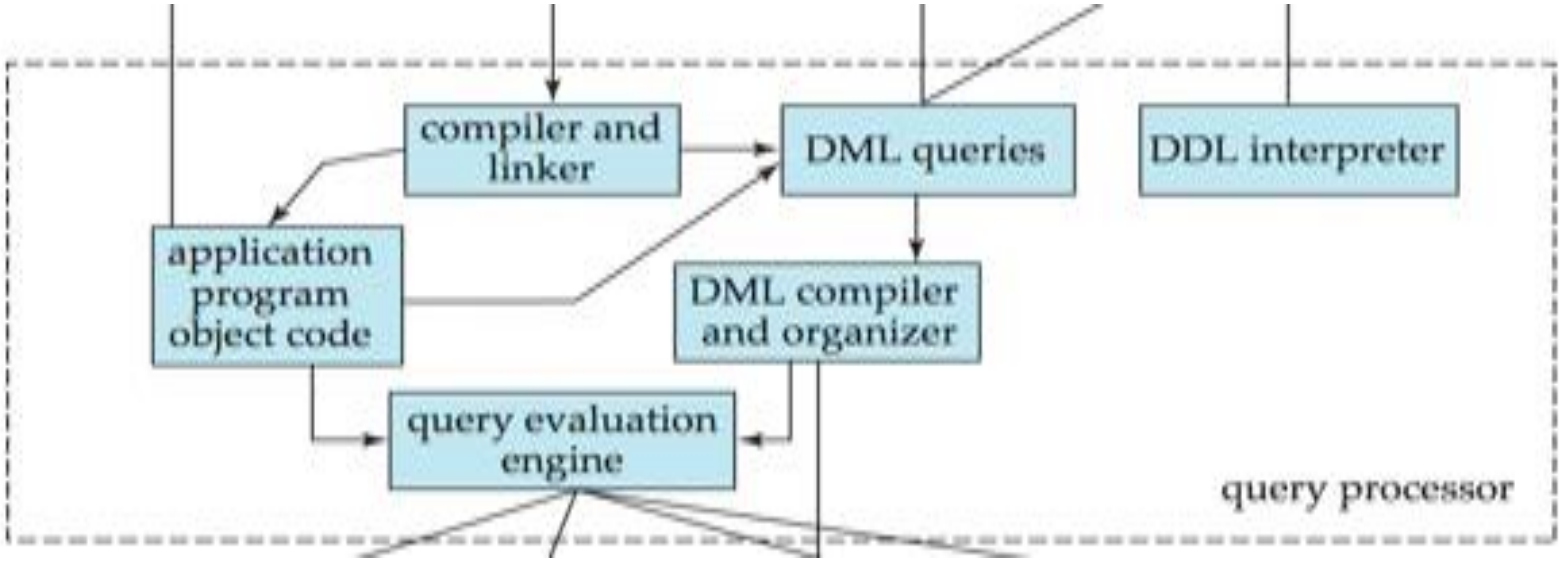
•Designs, creates and maintains the database.

•Manages the users who can access this database, and controls integrity issues.

•And monitors the performance of the system

•Makes changes in the system as and when required.

Query Processor



Query Processor

•It Includes

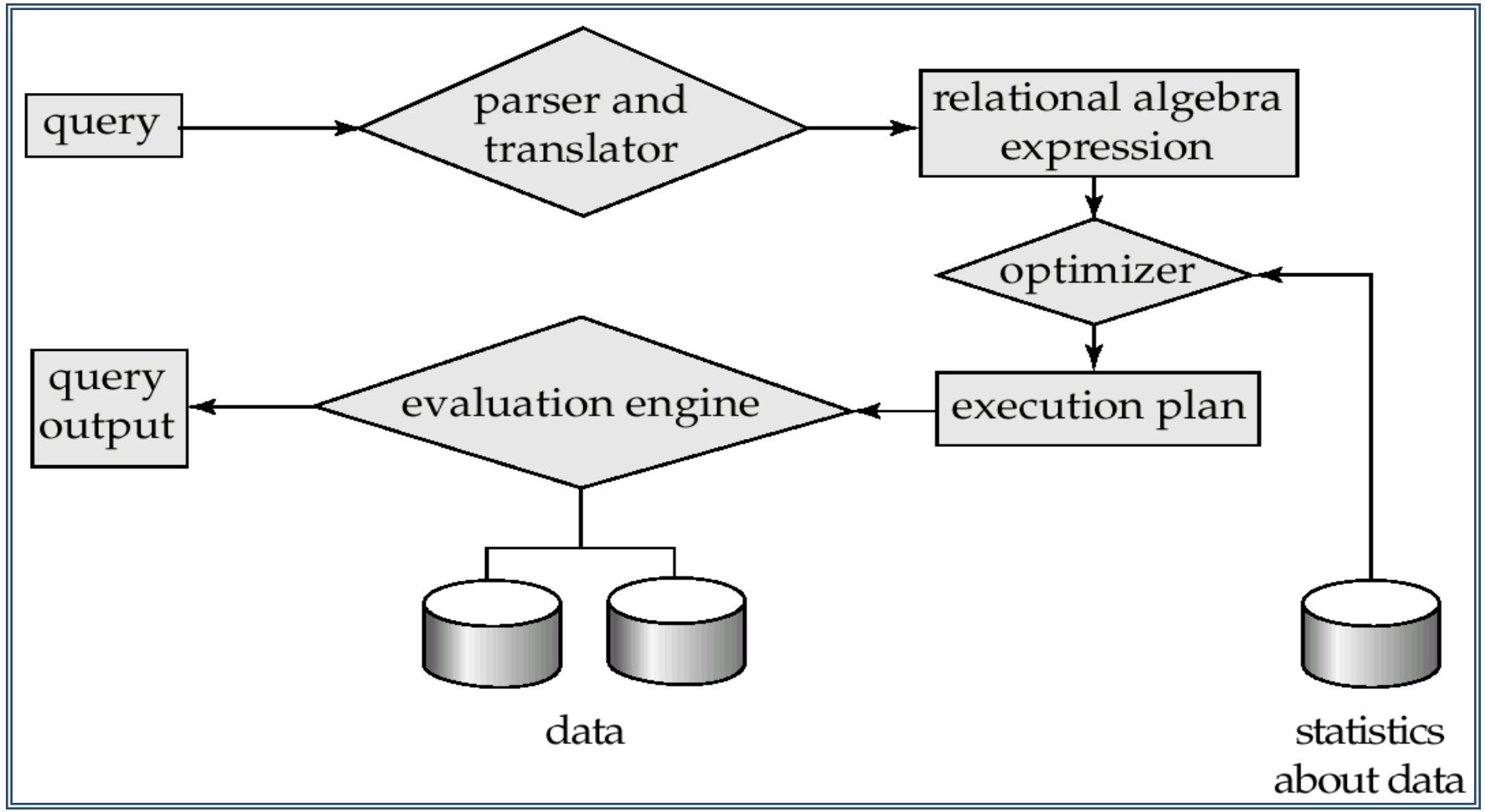
- **DDL interpreter -** interprets DDL statements and records the definitions in the data dictionary.
- **DML complier -** translates DML statements in a query language.

•A query can be translated into any of a number of alternatives evaluation plans that all give the same result.

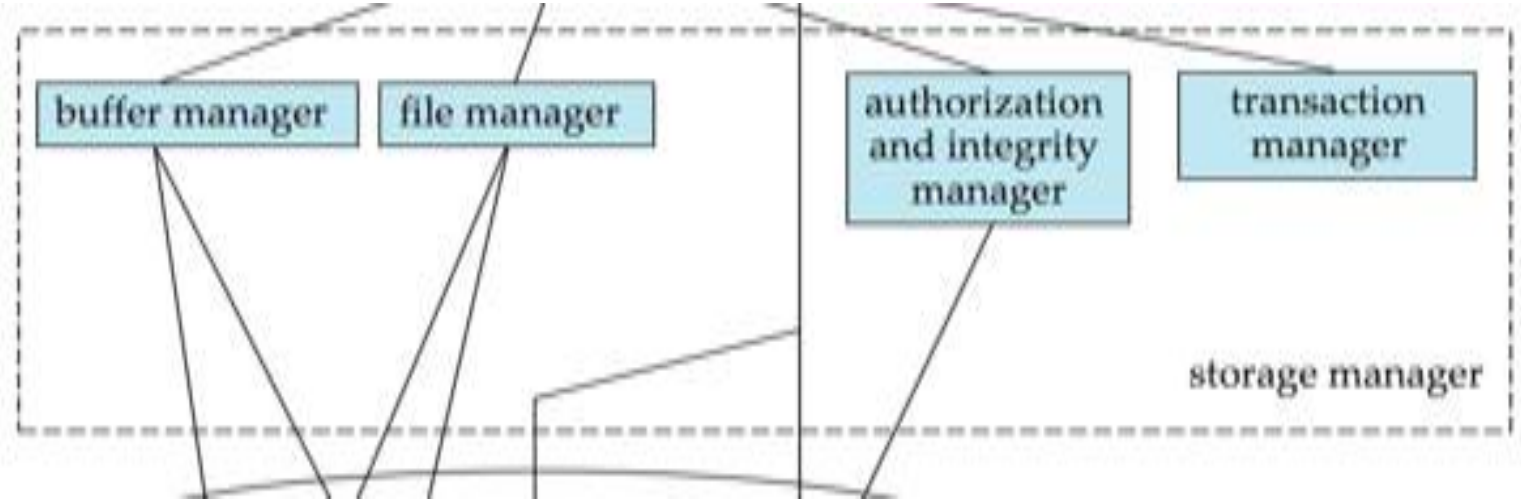
•DML complier performs query optimization, that is, it picks the lowest cost evaluation plan from among the alternatives.

•**Query evaluation engine**, which executes low-level instructions generated by the DML compiler.

Query Processor



Storage Manager



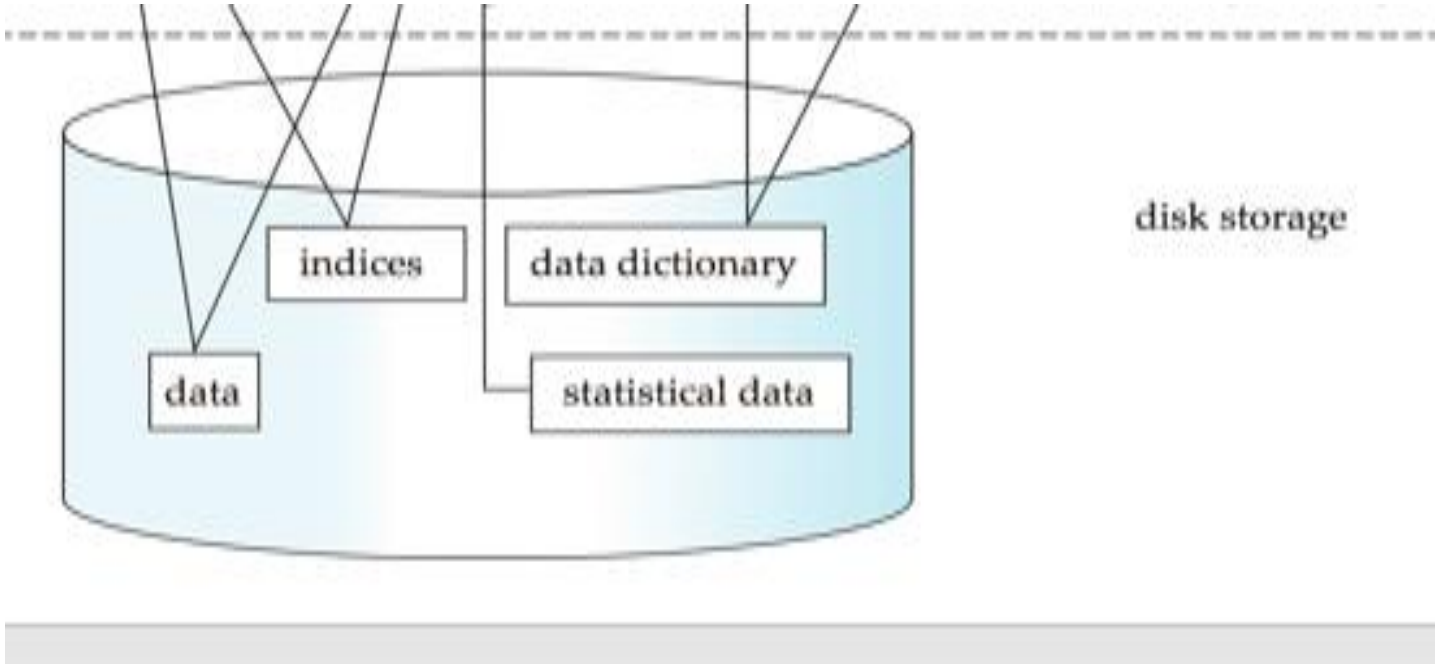
Storage Manager

Storage manager is a program module that provides the interface between the low-level data stored in the database and the application programs and queries submitted to the system.

Includes

- **Authorization and integrity manager**, which tests for the satisfaction of integrity constraints and checks authority to access the data.
- **Transaction manager**, which ensures that the database remains in a consistent state despite system failures and that concurrent transaction execution proceed without conflicting.
- **File Manager**, which manages the allocation of space on disk storage and the data structures used to represent information stored on disk.
- **Buffer manager**, which is responsible for fetching data from disk storage into main memory, and deciding what data to cache in main memory. It is used to handle data sizes that are much larger than the size of main memory.

Disk Storage



Disk Storage

The storage manager implements several data structures as part of the physical system implementation.

Data files, which store the database itself.

Data Dictionary, which stores metadata about the structure of the database, in particular the schema of the database.

Indices, which provide fast access to data items that hold particular values.

Queries?

Thank You