DBMS Architecture:

A database system is partitioned into modules that deal with each of the responsibilities of the overall system.

The **functional components of a database system** can be broadly divided into:

A. Storage manager and

The storage manager is important because databases typically require a large amount of storage space. Corporate databases range in size from hundreds of gigabytes to, for the largest databases, terabytes of data. Since the main memory of computers cannot store this much information, the information is stored on disks. Data are moved between disk storage and main memory as needed. Since the movement of data to and from disk is slow relative to the speed of the central processing unit, it is imperative that the database system structure the data so as to minimize the need to move data between disk and main memory.

B. Query processor

The query processor is important because it helps the database system to simplify and facilitate access to data. The query processor allows database users to obtain good performance while being able to work at the view level and not be burdened with understanding the physical-level details of the implementation of the system. It is the job of the database system to translate updates and queries written in a nonprocedural language, at the logical level, into an efficient sequence of operations at the physical level.

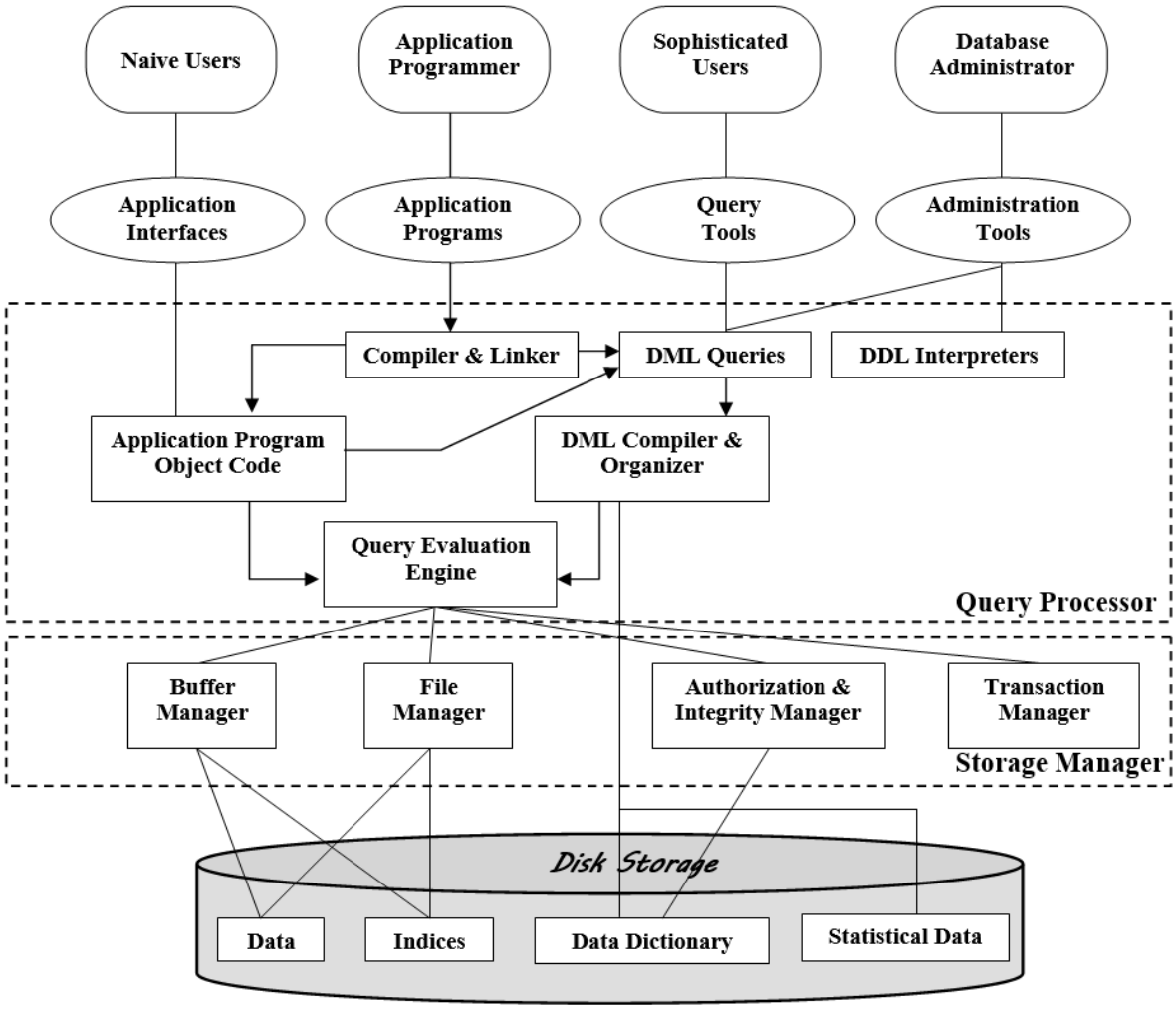

Figure: System Architecture

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 can provide fast access to data items. Like the index in this textbook, a database index provides pointers to those data items that hold a particular value. For example, we could use an index to find the instructor record with a particular ID, or all instructor records with a particular name. Hashing is an alternative to indexing that is faster in some but not all cases.

STORAGE MANAGER:

Description

i) The storage manager is the component of a database system that provides the **interface between the low-level data stored in the database and the application programs** and queries submitted to the system.
ii) The storage manager is responsible **for the interaction with the file manager**.
iii) The raw data are stored on the disk using the file system provided by the operating system.
iv) The storage manager translates the **various DML statements into low-level file-system commands.**
v) Thus, the storage manager is responsible for storing, retrieving, and updating data in the database.

The storage manager components include:

• **Authorization and integrity manager** - which tests for the satisfaction of integrity constraints and checks the authority of users to access data.

• **Transaction manager** - which ensures that the database remains in a consistent (correct) state despite system failures, and that concurrent transaction executions 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. The buffer manager is a critical part of the database system, since it enables the database to handle data sizes that are much larger than the size of main memory.

THE QUERY PROCESSOR:

The query processor components include:

• **DDL interpreter** - which interprets DDL statements and records the definitions in the data dictionary.

• **DML compiler** - which translates DML statements in a query language into an evaluation plan consisting of low-level instructions that the query evaluation engine understands. A query can usually be translated into any of a number of alternative evaluation plans that all give the same result. The DML compiler also 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.

USERS:

A primary goal of a database system is to retrieve information from and store new information into the database. People who work with a database can be categorized as database users or database administrators.

Database Users and User Interfaces

There are four different types of database-system users, differentiated by the way they expect to interact with the system.

1. **Na¨ıve users** are unsophisticated users who interact with the system by invoking one of the application programs that have been written previously.

For example, a clerk in the university who needs to add a new instructor to department A invokes a program called new hire. This program asks the clerk for the name of the new instructor, her new ID, the name of the department (that is, A), and the salary.

The typical user interface for na¨ıve users is a forms interface, where the user can fill in appropriate fields of the form. Na¨ıve users may also simply read reports generated from the database.

As another example, consider a student, who during class registration period, wishes to register for a class by using a Web interface. Such a user connects to a Web application program that runs at a Web server. The application first verifies the identity of the user, and allows her to access a form where she enters the desired information. The form information is sent back to the Web application at the server, which then determines if there is room in the class (by retrieving information from the database) and if so adds the student information to the class roster in the database.

2. **Application programmers** are computer professionals who write application programs. Application programmers can choose from many tools to develop user interfaces. Rapid application development (RAD) tools are tools that enable an application programmer to construct forms and reports with minimal programming effort.

3. **Sophisticated users** interact with the system without writing programs. Instead, they form their requests either using a database query language or by using tools such as data analysis software. Analysts who submit queries to explore data in the database fall in this category.

4. **Specialized users** are sophisticated users who write specialized database applications that do not fit into the traditional data-processing framework. Among these applications are computer-aided design systems, knowledgebase and expert systems, systems that store data with complex data types (for example, graphics data and audio data), and environment-modeling systems.

DATABASE ADMINISTRATOR ROLE:

One of the main reasons for using DBMSs is to have central control of both the data and the programs that access those data. A person who has such central control over the system is called a database administrator (DBA). The functions of a DBA include:

1. **Schema definition** - The DBA creates the original database schema by executing a set of data definition statements in the DDL.

2. **Storage structure and access-method definition**.

3. **Schema and physical-organization modification**. The DBA carries out changes to the schema and physical organization to reflect the changing needs of the organization, or to alter the physical organization to improve performance.

4. **Granting of authorization** for data access. By granting different types of authorization, the database administrator can regulate which parts of the database various users can access. The authorization information is kept in a special system structure that the database system consults whenever someone attempts to access the data in the system.

5. **Routine maintenance** - Examples of the database administrator's routine maintenance activities are:

    a. *Periodically backing up the database*, either onto tapes or onto remote servers, to prevent loss of data in case of disasters such as flooding.
    b. Ensuring that enough free disk space is available for normal operations, and *upgrading disk space as required*.
    c. *Monitoring jobs running on the database* and ensuring that performance is not degraded by very expensive tasks submitted by some users.

* * *