**Lesson 10**

**Objectives**

* Components of DBMS
  + Query Processor
  + File Manager
  + DML Preprocessor
  + DDL Compiler
  + Catalog Manager
  + Database Manager
    - Authorization Control
    - Command Processor
    - Integrity Checker
    - Query Optimizer
    - Transaction Manager
    - Scheduler
    - Recovery Manager
    - Buffer Manager

**Components of DBMS**

DBMSs are highly complex and sophisticated pieces of software that aim to provide the services discussed in the previous lectures. It is not possible to generalize the component structure of a DBMS as it varies greatly from system to system.

However, it is useful when trying to understand database systems to try to view the components and the relationships between them.

A DBMS is partitioned into several software components (or *modules*), each of which is assigned a specific operation. Some of the functions of the DBMS are supported by the underlying operating system. However, the operating system provides only basic services and the DBMS must be built on top of it. Thus, the design of a DBMS must take into account the interface between the DBMS and the operating system.

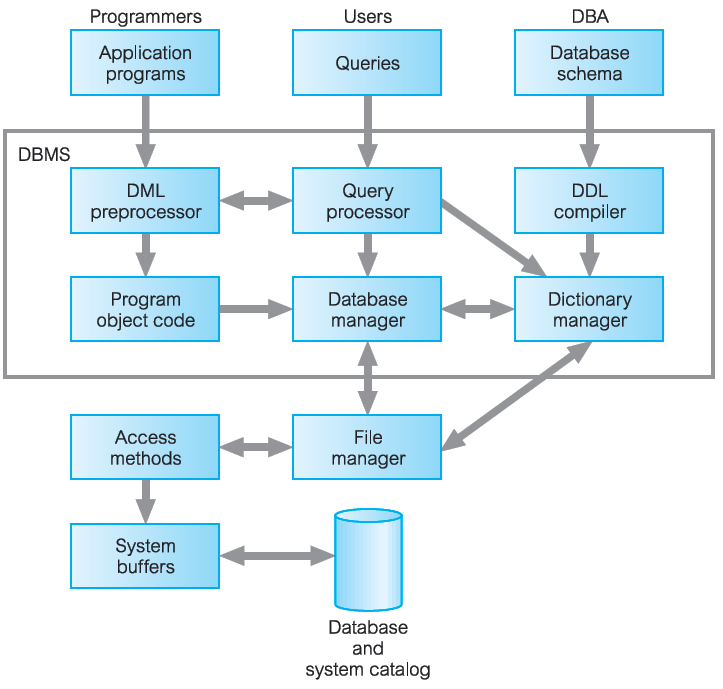
The major software components in a DBMS environment are depicted in Figure at next page. This figure shows how the DBMS interfaces with other software components, such as user queries and access methods (file management techniques for storing and retrieving data records).

**Query processor**

This is a major DBMS component that transforms queries into a series of low-level instructions directed to the database manager.

**File manager**

The file manager manipulates the underlying storage files and manages the allocation of storage space on disk. It establishes and maintains the list of structures and indexes defined in the internal schema. If hashed files are used it calls on the hashing functions to generate record addresses. However, the file manager does not directly manage the physical input and output of data. Rather it passes the requests on to the appropriate access methods, which either read data from or write data into the system buffer (or *cache*).



**DML preprocessor**

This module converts DML statements embedded in an application program into standard function calls in the host language. The DML preprocessor must interact with the query processor to generate the appropriate code.

***DDL compiler***

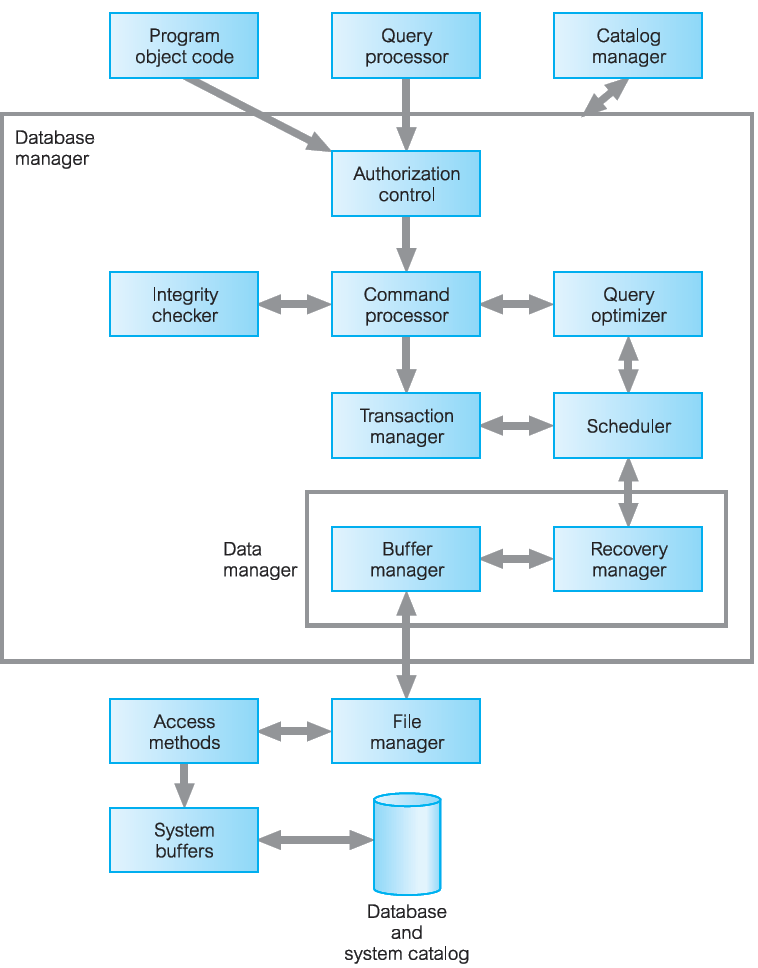
The DDL compiler converts DDL statements into a set of tables containing metadata. These tables are then stored in the system catalog while control information is stored in data file headers.

**Catalog manager**

The catalog manager manages access to and maintains the system catalog. The system catalog is accessed by most DBMS components.

**Database manager (DM)**

The DM interfaces with user-submitted application programs and queries. The DM accepts queries and examines the external and conceptual schemas to determine what conceptual records are required to satisfy the request. The DM then places a call to the file manager to perform the request. The components of the DM are shown in Figure below:



The major software components for the *database manager* are discussed one be one below:

**Authorization control**

This module checks that the user has the necessary authorization to carry out the required operation.

**Command processor**

Once the system has checked that the user has authority to carry out the operation, control is passed to the command processor.

**Integrity checker**

For an operation that changes the database, the integrity checker checks that the requested operation satisfies all necessary integrity constraints (such as key constraints).

**Query optimizer**

This module determines an optimal strategy for the query execution.

**Transaction manager**

This module performs the required processing of operations it receives from transactions.

**Scheduler**

This module is responsible for ensuring that concurrent operations on the database proceed without conflicting with one another. It controls the relative order in which transaction operations are executed.

**Recovery manager**

This module ensures that the database remains in a consistent state in the presence of failures. It is responsible for transaction commit and abort.

**Buffer manager**

This module is responsible for the transfer of data between main memory and secondary storage, such as disk and tape. The recovery manager and the buffer manager are sometimes referred to collectively as the *data manager*. The buffer manager is sometimes known as the *cache manager*.