

# Database Management System 4

## Database Architecture

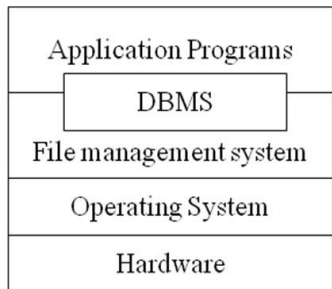
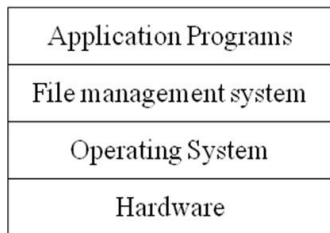
Hrudaya Kumar Tripathy  
School of Computer Engineering,  
KIIT University

## Data Storage and Querying

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

- The **storage manager** is important because databases typically require a large amount of storage space
- The **query processor** is important because it helps the database system simplify and facilitate access to data

The overall computer system consists of four modules as: Hardware, Operating system, File management system and Application program



## Storage Manager

### Storage Manager

A 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. The storage manager is responsible for the interaction with the file manager. Storage manager is responsible for storing, retrieving and updating data in the database

The storage manager components include:

- **Authorization and Integrity Manager**: This module tests for the satisfaction of integrity constraints and checks the authority of users to access data
- **Transaction Manager**: Transaction manager ensures that the database remains in a consistent (correct) state despite system failures and concurrent transaction executions proceed without conflicting
- **File Manager**: This module manages the allocation of space on disk storage and data structures used to represent information stored on the disk

## Storage Manager...

- **Buffer Manager**: Buffer manager is responsible for fetching data from the disk storage into main memory. The buffer manager is a critical part of the database system

The storage manager implements several data structures as part of the physical system implementations:

- **Data files**: These are files in the physical memory used to store the database itself
- **Data Dictionary**: Data dictionary stores the metadata (data about data) that provides the information about the definitions of the data items and their relationships, authorizations, and usage statistics. In addition, any changes made to the physical structure of the database are automatically recorded in the data dictionary
- **Indices**: Indices are used to provide faster access to data items stored in the physical storage

## Query Processor

### Query Processor

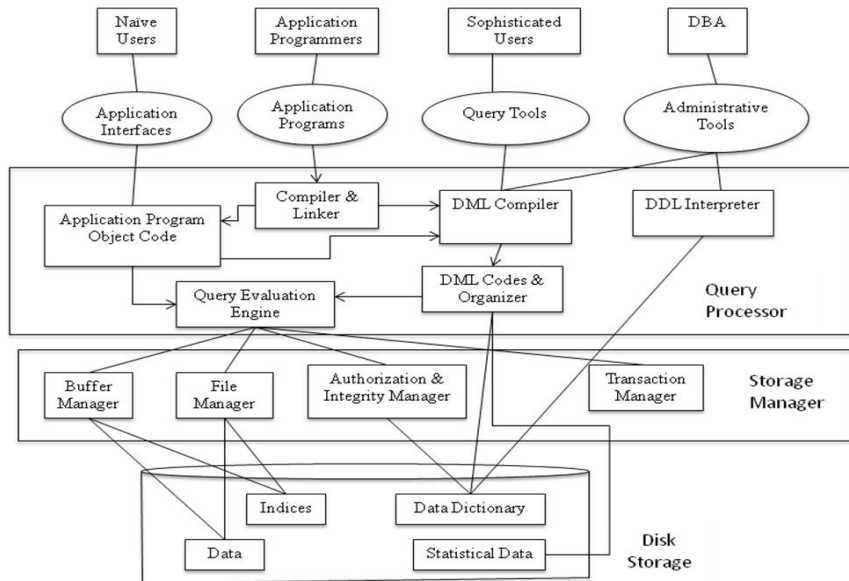
The work of query processor is to execute the query successfully

The major components of query processor include:

- **DDL Interpreter**: This is the interpreter used to interpret DDL statements and records the definitions in the data dictionary
- **DML Compiler**: DML compiler translates the DML statements in a query language into an evaluation plan consisting of low-level instructions that the query evaluation engine understands. When a user wants to perform a DML operation, the data dictionary has to be checked for the validation purpose
- **Query Evaluation Engine**: This module executes the low-level instructions generated by the DML compiler

## Database Architecture

The overall database architecture is:



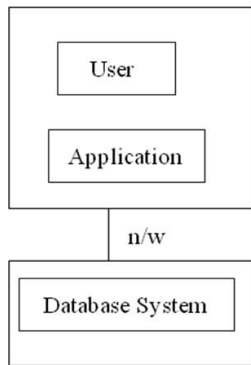
## Application Architecture

### Application Architecture

Client machines are those on which the remote database users work.  
Server machines are those on which the database system runs

- **2-Tier Architecture:** Here, the application is partitioned into a component that resides at the client machine, which invokes database system functionality at the **server machine through query language**. The two tiers are: **Data server** and **Client application**
- **3-Tier Architecture:** Here, the **client machine acts as a front end** and doesn't contain any direct database calls. The client end communicates with an **application server**, usually via a form **interfaces**. The application server in turn **communicates with a database system** to access data. The 3-tier applications are more appropriate for large applications, and the applications that run on the web

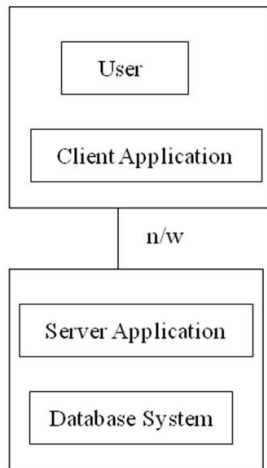
## Application Architecture...



2- Tier Architecture

Client

Server



3- Tier Architecture



## Disadvantages of Database Processing

The major disadvantages are:

- **Larger file size:** In order to support all the complex functions that it provides to users, a DBMS must be a large program that occupies a great amount of disk space as well as a substantial amount of internal memory
- **Increased Complexities:** The complexity and breadth of the functions provided by a DBMS make it a complex product. Users of the database system must learn a great deal to understand the features of the system in order to take full advantages of it
- **Greater Impact of Failure:** If several users are sharing the same database, a failure on the part of any one user that damages the database in some way might affect all the other users connected.
- **More difficult recovery:** The database must first be restored to the condition it was in when it was last known to be correct, any updates made by users since that time must be redone. The greater the number of users involved in updating the database, the more complicated this task becomes