

## Introduction to Database Systems

A database is a collection of related data. It is a collection of information that exists over a long period of time, often many years.

The common use of the term database usually refers to a collection of data that is managed by a database management system or DBMS.

A database has the following implicit properties:

-A database represents some aspect of the real world, sometimes called a miniworld or the universe of discourse (UoD). Changes to the miniworld are reflected in the database.

A database is a logically coherent collection of data with some inherent meaning.

A database is designed, built, and populated with data for a specific purpose. It has an intended group of users and some applications in which the user is interested.

A DBMS is a powerful tool for creating and managing large amount of data efficiently and allowing it to persist over long period of time.

A DBMS is a collection of programs that enable users to create and maintain a database. The DBMS is a general purpose software system that facilitates the processes of defining, constructing, manipulating and sharing databases among various users and applications.

The DBMS is expected to:

Allow users to create new databases and specify their logical structure of data or schemas, using a specialized data definition language.

Give users the ability to query the data, (a query is a question about the data) and modify the data, using an appropriate language called the query language or data manipulation language.

Support the storage of very large amount of data over a long period of time, allowing efficient access to the data for queries and data modifications.

Enable durability, the recovery of data in the face of failures, data inconsistency or intentional misuse.

Control access to data from many users at once, without allowing unexpected interaction between users (isolation) or incomplete operations on data (atomicity).

**Database Catalog or Dictionary or meta-data:** Defining a database involves specifying the data types, structures, and constraints of the data to be stored in the database. The database definition is stored by the DBMS in the form of a database catalog or dictionary; it is called meta-data.

**Constructing** the database is the process of storing the data based on the catalog using some storage device controlled by the DBMS.

**Manipulating** a database includes functions such as querying the database to retrieve specific data, updating the database to reflect changes in the miniworld.

Sharing a database allows multiple users or programs to access the database simultaneously.

An application program accesses a database by sending queries or request for information to the DBMS. A **query** typically causes some data to be retrieved.

A **transaction** may cause some data to be read and some data to be written into the database.

### **Data Stored in Other ways:**

Besides storing data in databases, there are other ways of storing data. For example, word documents and excel spreadsheets, etc...

#### **Files:**

Using files such as text files or Word documents to store data has many advantages especially when the amount of data is small. Besides, there is no need to get a database system or learn how to use it. It is easy to write applications using standard programming languages.

Disadvantages of using files:

Limited user interface

Are not efficient in handling large amount of data

Do not provide support to specifying relationships between data items

Do not support data integrity.

Do not provide a query language

#### **Spreadsheets:**

Advantages:

Provide structured storage model (rows and columns)

Support data types, Allow fields to be computed (based on other fields) and others.

However they also suffer from most disadvantages listed for files.

### **Early Database Management Systems:**

The first database management systems evolved from file systems and appeared in the late 1960's.

File systems store data over a long period of time and they allow for storage of large amount of data. However, they do not generally guarantee that data will not be lost and they do not support efficient access to data when their location is not known.

The first applications to use DBMS were ones where data was composed of many small items and many queries or modifications were required (banking, airline reservations, corporate record keeping, etc...)

Nowadays, databases have evolved to be of any size and complexity: a simple address book of hundreds of records to millions of entries (IRS, Amazon)

Users/Programmers

Database System

Application Programs/Queries

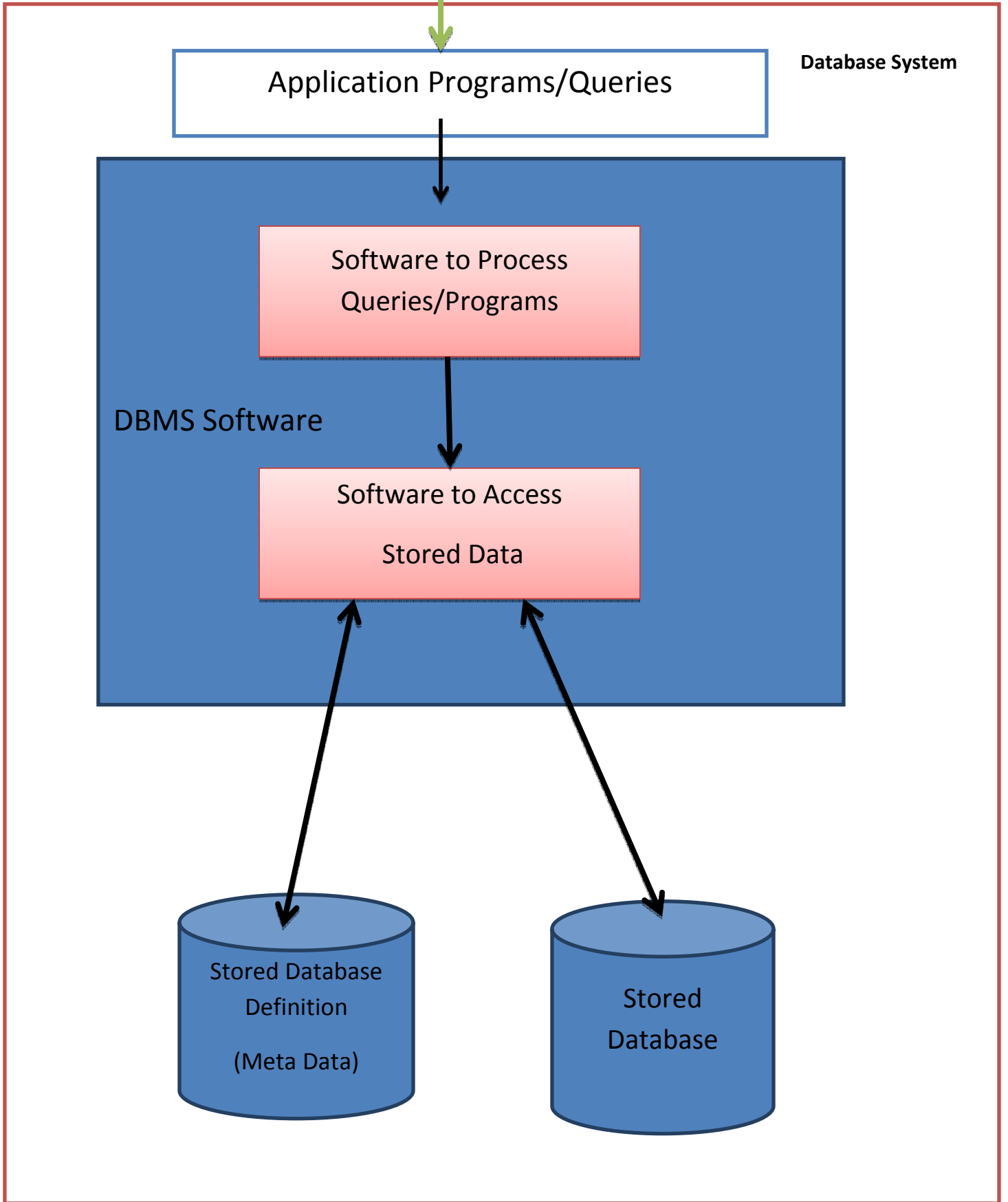
DBMS Software

Software to Process  
Queries/Programs

Software to Access  
Stored Data

Stored Database  
Definition  
(Meta Data)

Stored  
Database



## **Insulation between Programs and Data and Data Abstraction:**

### **Data Abstraction:**

In traditional file processing, the structure of data files is embedded in the application programs, so any changes to the file may require changing the programs that access that file. DBMS access programs do not require such changes in most cases. The structure of data files is stored in the DBMS catalog separately from access programs. This is called **program-data independence (encapsulation)**.

In some types of databases, such as OO (object-oriented) and OR (object-relational) systems, users can define functions or methods and only provide their prototype or interface (name, number and datatype of its arguments) to application programs that use them without knowing the details of their implementation. This is termed as **program-operation independence (Abstraction)**.

The DBMS provides users with a conceptual representation of the data that does not include how the data is stored or how the operations are implemented.

### **Database Views:**

A database typically has many users, each of whom requires a different perspective or view of the database. A view may be a subset of the database or it may contain virtual data that is derived from the database files but is not explicitly stored. A multiuser DBMS whose users have a variety of distinct applications must provide the ability for defining multiple views.

## **Sharing of Data and Multiuser Transaction Processing:**

A multiuser DBMS must allow multiple users to access the database at the same time. It is essential if data for multiple applications is to be integrated and maintained in the same database, that the DBMS includes concurrency control. Concurrency control ensures that several users trying to update the same data at the same time do so in a controlled manner, so that the result of the updates is correct.

A transaction is an executing program or process that includes one or more database accesses. Each transaction is supposed to execute a logically correct database access if executed in its entirety without interference from other transactions.

The isolation property ensures that each transaction appears to execute in isolation from other transactions, even though many of them could be executing concurrently.

The atomicity property ensures that either all the database operations are executed or none are.

### ***Actors:***

### **Database Administrators (DBA):**

In a database environment, the primary resource is the database, and the secondary resource is the DBMS and related software. The DBA is responsible for authorizing access to the database, coordinating and monitoring its use and acquiring software and hardware resources as needed. The DBA is accountable for problems such as security breach, poor response time, etc...

### **Database Designers:**

Database designers are responsible for identifying the data to be stored in the database and for choosing the appropriate structures to represent and store the data. These tasks are undertaken before the database is implemented and populated with data.

It is the responsibility of database designers to communicate with the prospective users in order to understand their needs and create a design that meets these requirements.

Database designers typically interact with each potential group of users and develop views of the database that meet the data and processing requirement of these groups. The final product must support the requirements of all user groups.

### **End users:**

Casual users: Occasional access, but different needs each time. Middle or high level managers.

### **Naïve or parametric users:**

Make up a sizable portion of database end users. Main function is to query and update the database using standard types of queries and updates or canned transactions.

### **System Analysts: (Software Engineers)**

Determine the requirements of end users, especially naïve end users and develop specification for the canned transactions that meet these requirements. Application engineers implement these specifications as programs they test, debug and document and then deploy them.

Other workers:

- **DBMS system designers and implementers:** A DBMS is a very complex software system that consists of many components or module, including implementation of the catalog, definition and implementation of query languages, interface processors, data access, concurrency control, etc... Systems designers and implementers design and implement these component, they usually work for a software company that sells the system with some customization to other non-software companies.
- **Tool developers:** include persons who design and implement tools- that facilitate database system design and use.

- Operators and maintenance personnel: are system administrations personnel who are responsible for the actual running and maintenance for the hardware and software environment for the database system.

## **Advantages of Using a DBMS:**

### **-Controlling Redundancy**

Redundancy is storing the same data multiple times. In traditional software development using file processing, each application usually maintains its own files for handling its data-processing applications. This can lead to several problems:

- Duplication of efforts: A single logical update requires entering the same data several times.
- Storage Waste: when the same data is stored multiple times, there is a waste a space.
- Data may become inconsistent: since updates can miss one or more repository or data may be entered differently from one file to the other.

In the database approach, the views of different groups are integrated during database design.

For consistency, we should have a database design that stores each logical data item, such as a student's name or birth date, - in only one place in the database. This does save space and does allow inconsistency.

However, controlled redundancy maybe useful for improving the performance of queries, in that we do not have to search multiple files to retrieve information. In such cases, the DBMS should have the capability to control this redundancy so as to prevent inconsistencies among files. This can be done by automatically checking that a field value in a table should have the same value or match that of another field in a different table. Such checks are specified to the DBMS during the database definition and design and automatically enforced by the DBMS whenever there is an update.

### **Restricting Unauthorized Access:**

When multiple users share a database, it is likely that they will have different access rights and authorization. The DBMS should provide a security and authorization subsystem, which the DBA uses to create accounts and to specify account restrictions.

Providing Persistent Storage for Program Objects and Data Structure:

An object is said to be persistent if it survives the termination of program execution and can later be directly retrieved by another program. The persistent storage of program and data is an important functionality of database systems.

#### **Providing multiple user interfaces:**

Because many users have varying levels of technical knowledge use of the database, a DBMS should provide a variety of user interfaces. These include query languages for casual users; programming language interfaces for applications programmers, forms and command code for parametric users and menu driven interfaces for natural language interfaces for stand-alone users.

#### **Representing Complex Relationships among Data:**

A database may include numerous varieties of data. A DBMS must have the capability to represent a variety of complex relationship among the data as well as retrieve and update data easily and efficiently.

#### **Enforcing Integrity Constraints:**

Integrity constraints involve specifying conditions for data correctness. The simplest one involves defining a data type for each data item. Others involve the length of a field or a range of values for a specific field.

More complex kind of constraints involve the inter relations between fields in different tables or uniqueness of data item values.

#### **Providing Backup and Recovery:**

A DBMS must provide facilities for recovering from hardware and software failures. The backup and recovery subsystem of the DBMS is responsible for recovery.