**Database Management System (DBMS)**

A **database management system** (DBMS) is a software tool that makes it possible to organize data in a database.

* Collection of interrelated data
* Set of programs to access the data
* DBMS contains information about a particular enterprise
* DBMS provides an environment that it both convenient an efficient to use

The standard acronym for database management system is **DBMS**, so you will often see this instead of the full name. The ultimate purpose of a database management system is to store and transform data into information to support making decisions.

A DBMS consists of the following three elements:

1. The **physical database**: the collection of files that contain the data
2. The **database engine**: the software that makes it possible to access and modify the contents of the database
3. The **database scheme**: the specification of the logical structure of the data stored in the database

While it sounds logical to have a DBMS in place, it is worth thinking for a moment about the alternative. What would the data in an organization look like without a DBMS? Consider yourself as the organization for a moment, and the data are all the files on your computer. How is your data organized? If you are like most typical computer users, you have a large number of files, organized in folders.

**Functions of a DBMS**

So what does a DBMS really do? It organizes your files to give you more control over your data.

A DBMS makes it possible for users to create, edit and update data in database files. Once created, the DBMS makes it possible to store and retrieve data from those database files.

More specifically, a DBMS provides the following functions:

* Concurrency: concurrent access (meaning 'at the same time') to the same database by multiple users
* Security: security rules to determine access rights of users
* Backup and recovery: processes to back-up the data regularly and recover data if a problem occurs
* Integrity: database structure and rules improve the integrity of the data
* Data descriptions: a data dictionary provides a description of the data

Within an organization, the development of the database is typically controlled by **database administrators (DBAs)** and other specialists. This ensures the database structure is efficient and reliable.

# Purpose of Database Systems

* To see why database management systems are necessary, let's look at a typical ``file-processing system'' supported by a conventional operating system.

The application is a savings bank:

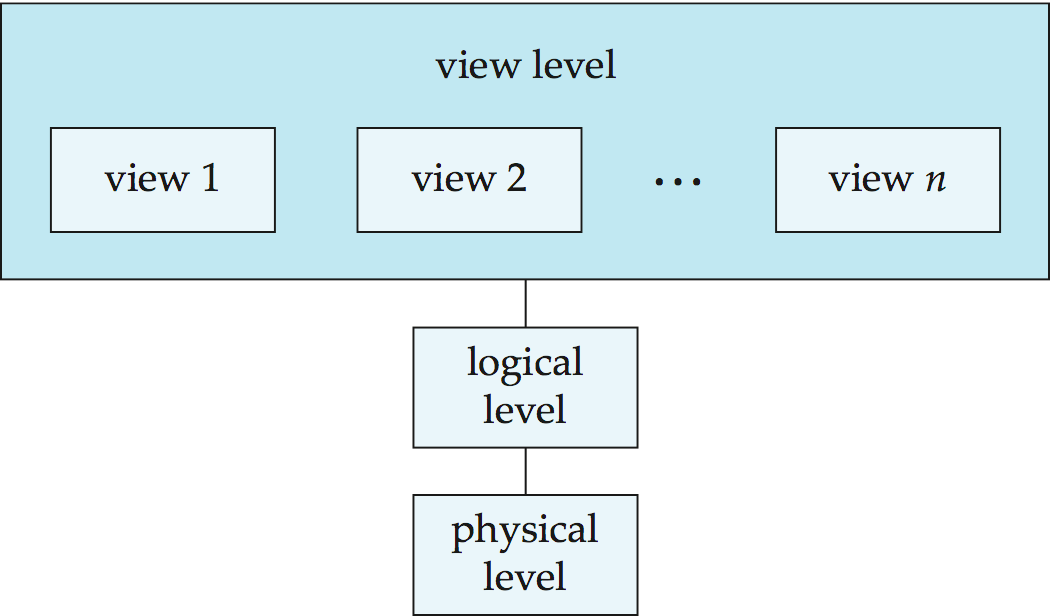
* + Savings account and customer records are kept in permanent system files.
  + Application programs are written to manipulate files to perform the following tasks:
    - Debit or credit an account.
    - Add a new account.
    - Find an account balance.
    - Generate monthly statements.
* Development of the system proceeds as follows:
  + New application programs must be written as the need arises.
  + New permanent files are created as required.
  + **but** over a long period of time files may be in different formats, and
  + Application programs may be in different languages.
* So we can see there are problems with the straight file-processing approach:
  + Data redundancy and inconsistency
    - Same information may be duplicated in several places.
    - All copies may not be updated properly.
  + Difficulty in accessing data
    - May have to write a new application program to satisfy an unusual request.
    - E.g. find all customers with the same postal code.
    - Could generate this data manually, but a long job...
  + Data isolation
    - Data in different files.
    - Data in different formats.
    - Difficult to write new application programs.
  + Multiple users
    - Want concurrency for faster response time.
    - Need protection for concurrent updates.
    - E.g. two customers withdrawing funds from the same account at the same time - account has $500 in it, and they withdraw $100 and $50. The result could be $350, $400 or $450 if no protection.
  + Security problems
    - Every user of the system should be able to access only the data they are permitted to see.
    - E.g. payroll people only handle employee records, and cannot see customer accounts; tellers only access account data and cannot see payroll data.
    - Difficult to enforce this with application programs.
  + Integrity problems
    - Data may be required to satisfy constraints.
    - E.g. no account balance below $25.00.
    - Again, difficult to enforce or to change constraints with the file-processing approach.
    - Data redundancy and inconsistency
* Difficulty in accessing data
* Data isolation
* Multiple files and formats
* Integrity problems
* Atomicity of updates
* Concurrent access by multiple users
* Security problems

**Disadvantages of DBMS**

1. Complexity:
   * The provision of the functionality that is expected of a good DBMS makes the DBMS an extremely complex piece of software. Failure to understand the system can lead to bad design decisions, which can have serious consequences for an organization.
2. Size:
   * The complexity and breadth of functionality makes the DBMS an extremely large piece of software, occupying many megabytes of disk space and requiring substantial amounts of memory to run efficiently.
3. Performance:
4. The DBMS file based system is written to be more general, to cater for many applications rather than just one. The effect is that some applications may not run as fast as they used to.

**View of Data**

An architecture for a database system.



**Data Models**

A collection of tools for describing:

* Data
* data relationships
* data semantics
* data constraints
* Object-based logical models
* entity-relationship model
* object-oriented model
* semantic model
* functional model
* Record-based logical models
* relational model (e.g., SQL/DS, DB2)
* network model
* hierarchical model (e.g., IMS)

**Data Definition Language (DDL)**

* Specification notation for defining the database schema
* DDL compiler generates a set of tables stored in a data dictionary
* Data dictionary contains metadata (i.e., data about data)
* Data storage and definition language
* special type of DDL inwhich the storage structure and access methods used by the database system are specified

**Data Manipulation Language (DML)**

* Language for accessing and manipulating the data organized by the appropriate data model
* Two classes of languages
* Procedural – user specifies what data is required and how to get those data
* Nonprocedural – user specifies what data is required without specifying how to get those data.

**Object-Relational Data Models**

Relational model: flat, “atomic” values

Object Relational Data Models

* Extend the relational data model by including object orientation and constructs to deal with added data types.
* Allow attributes of tuples to have complex types, including non-atomic values such as nested relations.
* Preserve relational foundations, in particular the declarative access to data, while extending modeling power.
* Provide upward compatibility with existing relational languages

**Transaction Management**

A transaction is a collection of operations that performs a

* single logical function in a database application
* Transaction-management component ensures that thedatabase remains in a consistent (correct) state despitesystem failures (e.g. power failures and operating system crashes) and transaction failures.
* Concurrency-control manager controls the interaction among the concurrent transactions, to ensure the consistency of the database.

**Storage Management**

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 following tasks:
* interaction with the file manager
* efficient storing, retrieving, and updating of data

**Database Administrator**

Coordinates all the activities of the database system; the

database administrator has a good understanding of the

enterprise’s information resources and needs.

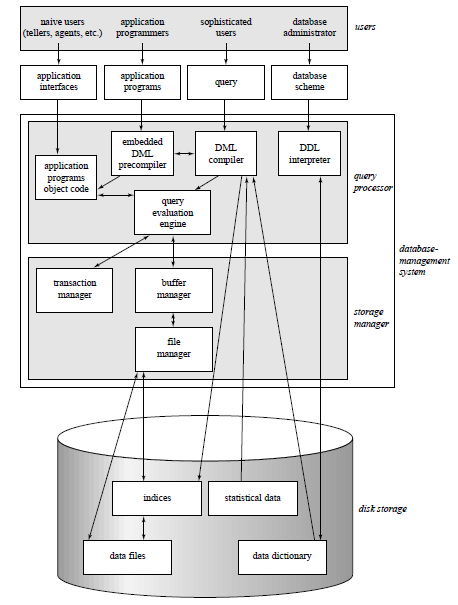
* Database administrator’s duties include:
* Schema definition
* Storage structure and access method definition
* Schema and physical organization modification
* Granting user authority to access the database
* Specifying integrity constraints
* Acting as liaison with users
* Monitoring performance and responding to changes in Requirements.

**Database Architecture**

The architecture of a database systems is greatly influenced by the underlying computer system on which the database is running.

* Centralized
* Client-server
* Parallel (multi-processor)
* Distributed

**Overall System Structure**

****

**History Of Database Management**

**1980’s**

* Birth of IBM PC.
* RDBMS market begins to boom.
* SQL becomes standardized through ANSI (American National Standards Institute) and ISO (International Organization for Standardization)
* By Mid 80’s it had become apparent that there were some fields(medicine, multimedia, physics) where relational databases were not practical, due to the types of data involved.
* More flexibility was needed in how their data was represented and accessed.
* This led to research in Object Oriented Databases in which users could define their own methods of access to data and how to represent and manipulate it. This coincided with the introduction of Object Oriented Programming languages such as C++ which started to appear

1990’s

* First OODBMS’ start to appear from companies like Objectivity. Object Relational DBMS’ hybrids also begin to appear.
* Industry shakeout begins with fewer surviving companies offering increasingly complex products at higher prices. Much of the development centers on client tools for application development such as: PowerBuilder(Sybase), Oracle Developer, Visual Basic, etc
* Development of personal/small business productivity tools such as Excel and Access from Microsoft.
* New application areas: Data warehousing and OLAP(Online Analytical Processing, a category of software tools that provides analysis of data stored in a database), internet, multimedia, etc

**Late 90’s-2000’s**

* Large investment in internet companies fuels tools-market boom for Web/Internet/DB connectors:
  + Active Server Pages, Front page, Java Servlets, JDBC, Java Beans, ColdFusion, Dream Weaver, Oracle Developer 2000, etc
* Open source projects come online with widespread use of gcc,cgi, Apache, MySQL
* Three main companies dominate in the large DB market: IBM, Microsoft, and Oracle

**REFERENCES:**

[1][*http://www.cs.sjsu.edu/faculty/lee/cs157/*](http://www.cs.sjsu.edu/faculty/lee/cs157/)

[2] [*https://en.wikipedia.org/wiki/Database*](https://en.wikipedia.org/wiki/Database)

[3][*http://www.techopedia.com/definition/24361/database-management-systems-dbms*](http://www.techopedia.com/definition/24361/database-management-systems-dbms)