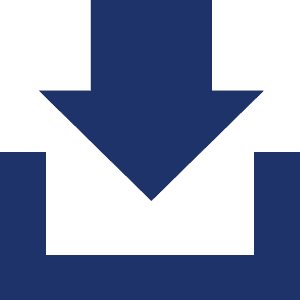
## Database Systems Concepts



Get This Book

**Download Book:**

* [PDF](http://cnx.org/exports/cdab6ade-b651-4ee9-afda-72e5a183a24c%401.pdf/database-systems-concepts-1.pdf)

Page by: Nguyen Kim Anh

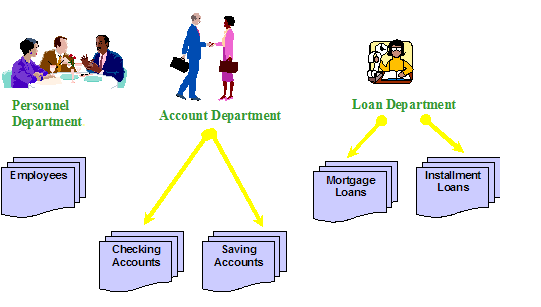
Summary

This module introduces the fundamental concepts of database systems

Before the advent of Database systems

One way to keep the information on a computer is to store it in the permanent files. The system has a number of application programs, each of them is defined to manipulate the data files. These application programs have been written on request of the users in the organization. New application will be added to the system as the need arises. The system just described is called the file-based system.

Consider a traditional banking system which using the file-based system in managing the organization’s data in the picture below. As we can see, there are different departments in the Bank, each of them have their own applications which manage and manipulate differents data files. For Banking system, the programs can be the one to debit or credit an account, , find the balance of an account, add a new mortgage loan or generate monthly statements etc.

File-based approach for banking system

Keeping organizational information in this approach has a number of disadvantages, including :

* Data Redundancy: Since files and applications are created by different programmer of various departments over long period of time, it might lead to several problems:
  + Inconsistency in data format
  + The same information may be kept in several different place (files).
  + Data inconsistency which means various copies of the same data are conflicting ; waste storage space and duplication of effort
* Data Isolation
  + It is difficult for new application to retrieve the appropriate data which might be stored in various files.
* Integrity problems
  + Data values must satisfy certain consistency contraints which are specified in the application programs.
  + It is difficult to add change the programs to enforce new constraint
* Security problems
  + There are constraint regarding accessing privileges
  + Application is added to the system in the ad-hoc manner so it is difficult to enforce those constraints
* Concurrent – access anomalies
  + Data may be accessed by many applications that have not been coordinated previously so it is not easy to provide a stategy to support multiple users to update data simutaneously

These difficulties have prompted the development of a new approach in managing large amount of organizational information – database approach. In the following section, we shall see the concepts that have been introduced to get over the problems mentioned.

Database Approach

Database and database technology play an important role in most of social areas where computer are used, including business, education, medicine etc. To understand the fundamental of database system, we start from introducing the basic concepts in this area.

Fundamental Concepts

Database is a shared collection of related data which will be used to support the activities of particular organization. Database can be viewed as a repository of data that is defined once and then is accessed by various users. A database has the following properties:

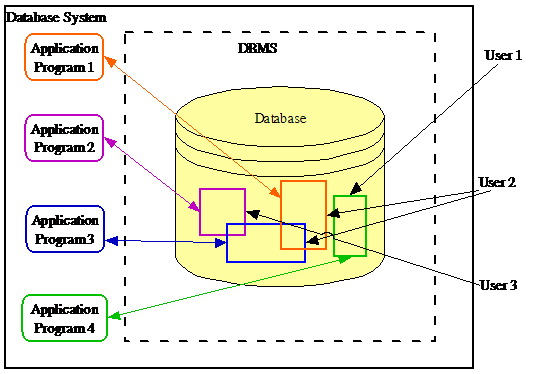
It is a representation of some aspect of the real world; or perhaps, a collection of data elements (facts) representing real­world information.

Database is logical coherent and internally consistent.

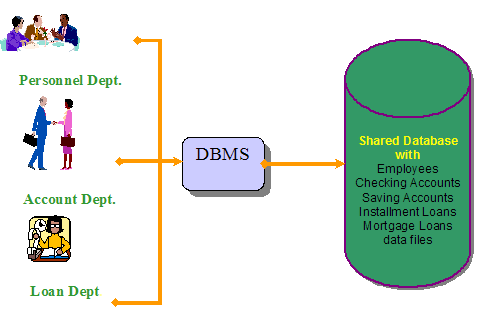
Database is designed, built, and populated with data for a specific purpose.

Database Management System (DBMS) is a collection of programs that enable users to create, maintain database and control all the access to the database. The primary goal of the DBMS is to provide an environment that is both convenient and efficient for user to retrive and store information.

Application program accesses the data stored in the database by sending requests to the DBMS.

The components of a database system

With the database approach, we can have the traditional banking system as shown in the following picture.

Database approach for banking system

Characteristics of Database approach

There are a number of characteristics that distinguish the database approach with the file-based approach. In this section, we describe in detail some of those important characteristics.

1. Self-Describing Nature of a Database System : Database System contains not only the database itself but also the descriptions of data structure and constraints (meta-data). These information is used by the DBMS software or database users if needed. This separation makes database system totally different from traditional file-based system in which data definition is a part of application programs
2. Insulation between Program and Data : In the filed base system, the structure of the data files is defined in the application programs so if user want to change the structure of a file, all the programs access to that files might need to be changed. On the other hand, in database approach, data structure is stored in the system catalog not in the programs so such changes might not occurs.
3. Support multiple views of data: A view is a subset of the database which is defined and dedicated for particular users of the system. Multiple users in the system might have different views of the system. Each view might contains only the interested data of an user or a group of user.
4. Sharing of data and Multiuser system: A multiuser database system must allow multiple users access the database at the same time. As the result, the multiuser DBMS must have concurrency control strategies to ensure that several user try to access the same data item at a time do so in the manner so that the data always be correct.

Benefits of Database Approach

1. To control Data Redundancy
   * In the Database approach, ideally each data item is stored in only one place in the database
   * However, in some case redundancy is still exists to improving system performance, but such redundancy is controlled and kept to minimum
2. Data Sharing
   * The integration of the whole data in an organization leads to the ability to produce more information from a given amount of data
3. Enforcing Integrity Constraints
   * DBMSs should provide capabilities to define and enforce certain constraints such as data type, data uniqueness.
4. Restricting Unauthorised Access
   * Not all users of the system have the same accessing privileges.
   * DBMSs should provide a security subsystem to create and control the user accounts.
5. Data Independence
   * The system data descriptions are separated from the application programs.
   * Changes to the data structure is handled by the DBMS and not embedded in the program.
6. Transaction Processing
   * The DBMS must include concurrency control subsystem to ensure that several users trying to update the same data do so in a controlled manner so that the result of the updates is correct.
7. Providing multiple views of data
   * A view may be a subset of the database. Various users may have different views of the database itself.
   * Users may not need to be aware of how and where the data they refer to is stored
8. Providing backup and recovery facilities
   * If the computer system fails in the middle of a complex update program, the recovery subsystem is responsible for making sure that the database is restored to the stage it was in before the program started executing.

Data Models, Schemas and Instances

Data Model is a collection of concepts that can be used to describe the structure of database. Structure of database means data types, relationships and constraints. In addition, most data model include a set of basic operations for specifying retrievals and modifications on the database. Data Model provides a means to achieve Data Abstraction. Data Abstraction is refers to the hiding of certain details of how the data are stored and maintained. With several levels of abstraction, the user’s view of the database is simplified and this leads to the improved understanding of data.

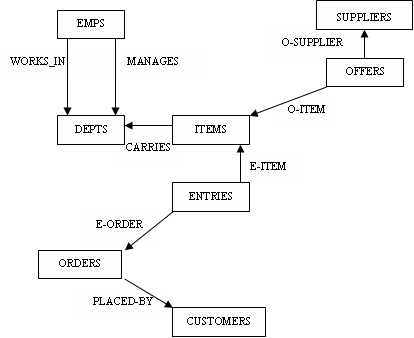
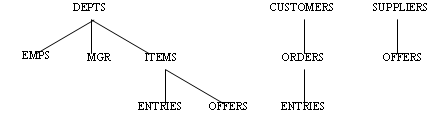
There are three levels of abstractions:

1. View level: The highest level of abstraction describes only part of the entire database. Many users will not be concerned with the large database. Instead, they need to access only a part of it so that view level abstraction is defined. There are many views for the same database.
2. Logical level: This level describes what data are stored in the whole database.
3. Physical level: The lowest level of abstraction describes how the data are actually stored.

Categories of Data Model

* High-level Conceptual Data models: Provide concepts that are close to the way people perceive data to present the data. Typical example of this type is entity – relationship model which use main concepts like entities, attributes, relationships. An entity represents real-world object such as an employee, a project. An entity has some attributes which represents properties of entity such as employee’s name, address, birthdate. A relationship represents association among entities for example a works on relationships between employee and project.
* Record-based Logical Data models: Provide concepts that can be understood by the user but not too far from the way data is stored in the computer. Three well-known data models of this type are relational data model, network data model and hierarchical data model.
  + The Relational model represents data as relations. Here is an example of relational schema for the SUPERMARKET database

|  |  |
| --- | --- |
| EMPS (ENAME, SALARY) | MANAGERS (ENAME) |
| DEPTS (DNAME, DEPT#) | SUPPLIERS (SNAME, SADDR) |
| ITEMS (INAME, ITEM#) | ORDERS (O#, DATE) |
| WORKS\_IN (ENAME, DNAME) | MANAGES (ENAME, DNAME) |
| CARRIES (INAME, DNAME) | PLACED\_BY (O#, CNAME) |
| CUSTOMERS(CNAME,CADDR,BALANCE) | INCLUDES (O#, INAME, QUANTITY) |
| SUPPLIES (SNAME, INAME, PRICES) |  |

* + The Network model represents data as record types and also represents a limited type of one to many relationship, called set type. The figure below shows a schema in network model notation Sample schema in network model
  + The Hierarchical model represent data as hierarchical tree structures. Each hierarchy represents a number of related records. Here is the schema in hierarchical model notation. Sample schema in hierarchical model notation
  + Physical Data models: Provide concepts that describe how data is actually stored in the computer.

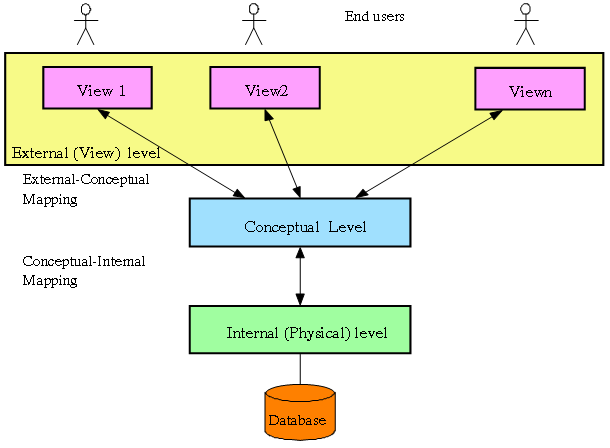
Database Instances and Schemas

The description of the database which is designed in the early stage and is not expected to change frequently is called the database schema. Database system have severals schemas.

Since information can be inserted to or deleted from database at anytime, database changes over time. At a particular moment, the collection of information stored in the database is called an instance of the database.

ANSI/SPARC Architecture (Three –level Architecture)

Three – level architecture for database system is proposed to archive the characteristics of the database approach. The goal of this architecture is separate the applications and the physical database so the actual details of how data is organized are hided from the users.

Three- level Architecture

As we can see from above picture, there are three levels of schemas in the database architecture

External level:

* In this highest level, there exists a number of views which of is defined a part of the actual database.
* Each view is provided for a user or a group of users so that it helps in simplified the interaction between the user and system.

Conceptual level: Conceptual Schema in this level describes the logical structure of the whole database.

* The entire database is described using simple logical concepts such as objects, their properties or relationships. Thus the complexity of the implementation detail of the data with be hided from the users.
* Internal level: Internal Schema in this level describes how the data are actually stored, how to access the data.

Data Independence

Data Independence is the ability to modify the schema in one level without affecting the schema in the higher level.

There are two levels of data independence:

* Logical data independence is the ability to make change in the conceptual schema without causing a modify in the user views or application program.
* Physical data independence is the ability to make change in the internal schema without causing a modify in the conceptual schema or application program.

Physical data independence seem to be easier to achieve since the way the data is organized in the memory affect only the performance of the system. Meanwhile, the application program depends much on the logical structure of the data that they are access.

Database Language

Data Definition Language (DDL): This is used to define the conceptual and internal schemas for a database system.

* It is not procedural language, rather a language for describing the types of entities and relationships among them in terms of a particular data model.
* Data Manipulation Language (DML): This is used to manipulate the database, which typically include retrieval, insertion, deletion, and modification of the data.

Database Users

End users

People whose jobs require access to database for querying, updating and generating report.

An end users might by one of the following

* Naïve users who use the existing appication programs to perform their daily tasks
* Sophiticated users are who use their own way to access to the database. This mean they donot use the application program provided in the system. In stead, they might define their own application or describe their need directly in a query languages.
* Specialized users maintain the personal database by using ready –make program packages that provide easy-to-use menu.

Application Programmer

People implement specific application program to access to the stored data. This kind of user need to familiar with the DBMSs to accomplish their task.

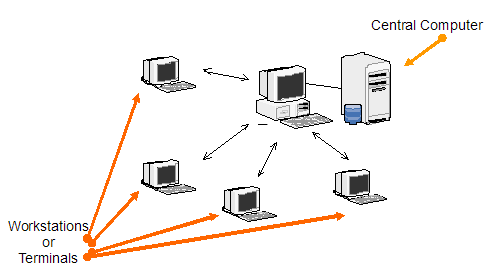
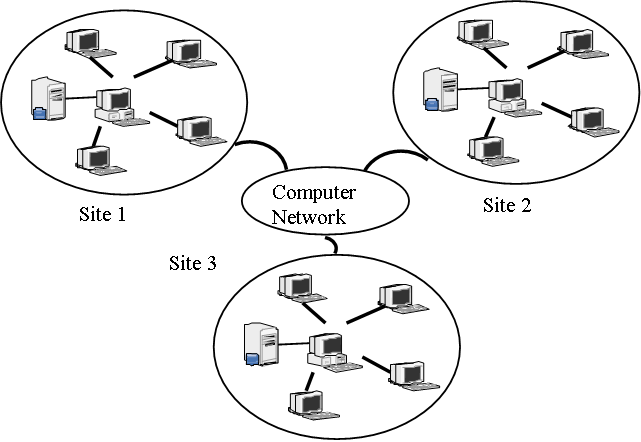
Database Administrators

A person or a group of people in the organization who is responsible for authorizing the access to the database, monitoring its use and managing all the resource to support the use of the whole database system

Classification of Database Systems

The database management systems can be classified based on several criteria.

* Based on data model: The most popular data model in today commercial DBMSs is relational data model. Almost wellknown DBMSs like Oracle, MS SQL Server, DB2, MySQL are support this model. Other traditional models can be named hierarchical data model , network data model. In the recent year, we are getting familiar with object-oriented data model but this model has not had widespread use. Some examples of Object-oriented DBMSs are O2, ObjectStore or Jasmine.
* Based on number users we can have single user database system which support one user at a time or multiuser systems which support multiple users concurrently
* Based on the ways database is distributed we have centralized or distributed database system
  + Centralized database system : Data in this kind of system is stored at a single site.
  + Distributed database system: Actual database and DBMS software are distributed in various sites connected by a computer network.
    - Homogeneous distributed Database Systems
      * Use the same DBMS software at multiple sites
      * Data exchange between various sites can be handle easily
    - Heterogeneous distributed Database Systems
      * Different sites might use different DBMS software
      * There is a software to support data exchange between sites

Centralized Database System Distributed Database System