COURSE CODE: UCI 204

COURSE TITLE : PRINCIPLES OF RELATIONAL DATABASE SYSTEMS

TOPIC 1. FUNDAMENTALS OF DATABASES

MASENO UNIVERSITY

eCampus

Introduction

Databases touch all aspects of our lives, be it at our grocery shops, at home, in the super-market, at the banks, in our mobile phones, at the airport, at the beach, name it. The databases ensures that we have a convenient and efficient way to access and processing our data

Topic Objective:

At the end of this topic the learner should be able to:

- ✓ Understand the traditional file based approach
- ✓ Determine the problems and limitations of the file-based approach
- ✓ Appreciate the Relational Database approach
- ✓ Understand the role of the RDBMS
- ✓ Determine the merits and demerits of RDBMS

Definitions and Terms

- ✓ **Database:** A collection of related data.
- ✓ **Data:** Known facts that can be recorded and have an implicit meaning.
- ✓ **Mini-world:** Some part of the real world about which data is stored in a database. e.g., Student grades and transcripts at a Maseno University.
- ✓ **Database Management System (DBMS):** A software package/ system to facilitate the creation and maintenance of a computerized database. MS Access, MySQL, Oracle, dBase, SQL Server etc, fall here.
- ✓ **Database System:** The DBMS software together with the data itself. Sometimes, the applications are also included, the Ministry of Finance's Integrated Financial Management Information System [IFMIS], falls here.

Traditional file based approach

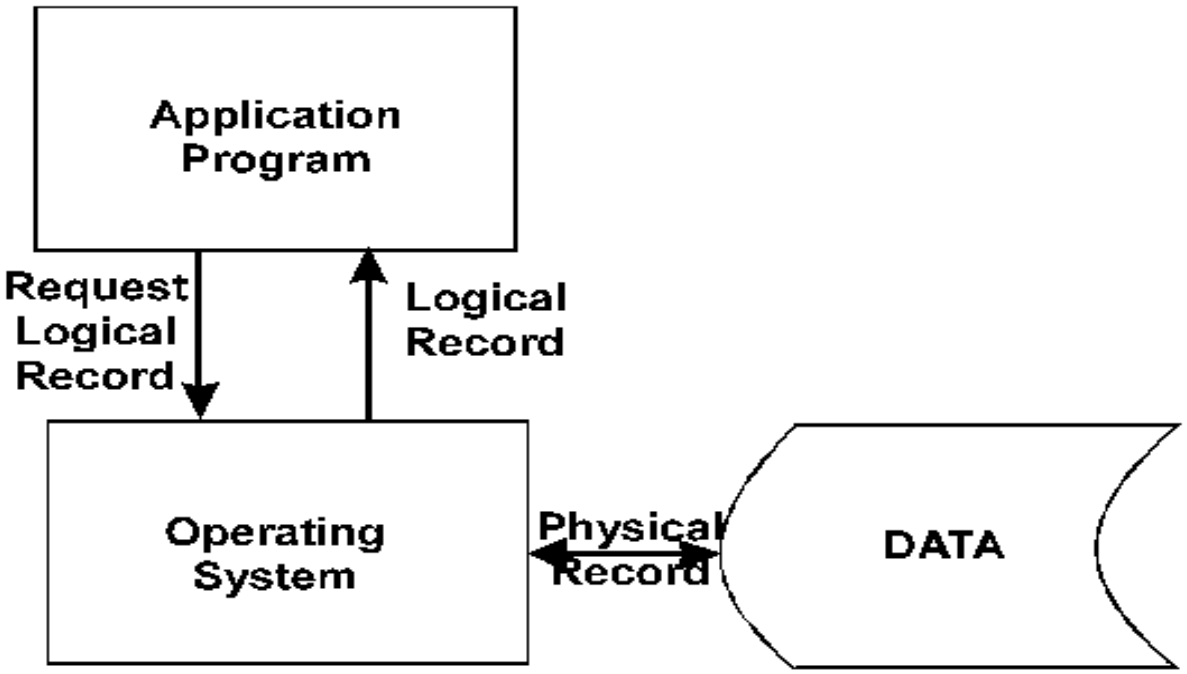
The traditional filing system (TFS) is a technique of collecting, arranging computer files and storing the information in the file (data). Basically it organizes these files into a database for the storage, organization, manipulation, and retrieval by the computer's operating system.

File-based systems were an early attempt to computerize the manual filing system. File-based system is a collection of application programs that perform services for the end-users, such as updating, insertion, deletion adding new files to database etc. Each program defines and manages its data exclusively.

Essentially, file-based systems require the application program to deal with stored data by its location: where within a record a field is held, where that record is located within the file, and where that file is located on the storage device.

The application programmer is protected to some extent in file-based systems from the realization that it's all happening by numbers because the compiler software and the operating system software work together to do all the translations from names to numbers, and vice-versa.

However, this means that programs must be re-compiled (translated from programmer language to machine language) whenever the data layout is changed. (This was the essence of the Y2K problem!)



When a computer user wants to store data electronically they must do so by placing data in files. Files are stored in specific locations on the hard disk (directories). The user can create new files to place data in, delete a file that contains data, rename the file, etc which is known as file management; a function provided by the Operating System (OS).

Disadvantages of traditional file based system

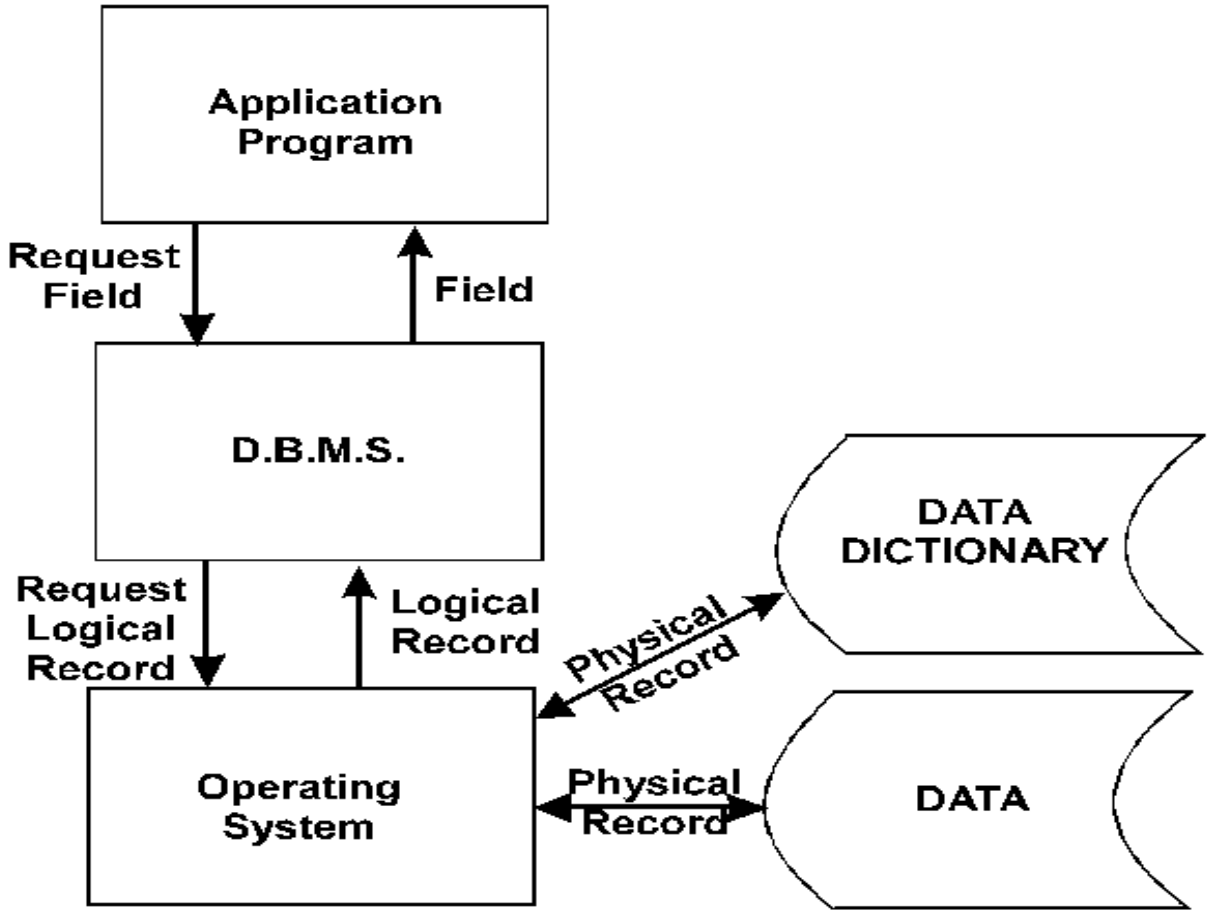
- ✓ No need of external storage
- ✓ Provides less security.
- ✓ No need of a highly technical person to handle the database.
- ✓ Redundancy is more.
- ✓ Processing speed is high as compared to DBMS
- ✓ Less integrity.
- ✓ High complexity in updating of database.

Database Management System

The improvement of the File-Based System (FBS) was the Database Management System (DBMS) which came up in the 60's.

The Database Management System removed the trouble of manually locating data, and having to go through it. The user could create a suitable structure for the data beforehand, to place the information in the database that the DBMS is managing. Hence, the physical organizing of files is done away with and provides the user with a logical view of the data input.

Remember that a database is a collection of interrelated information stored in a database server; these data will be stored in the form of tables. The primary aim of database is to provide a way to store and retrieve database information fast and in an efficient manner.



Types of Database Management Systems

Historically, there have been three types of DBMS:

- Hierarchical
- Network
- Relational

Hierarchical Databases-

These DBMS required that all entities within a database be defined in parent-child relationships. For example, to refer to a household, you would first have to specify not only its municipality, but also its

county, its province/state, its country. In hierarchical databases, each parent record carried a pointer to its first child, each child to the next child of the same parent, and the last child back to the parent. There was no way to establish relationships amongst cousins, but simply those amongst siblings, parents, grand-parents, etc. From another viewpoint, all relationships were one-to-many.

Network Databases-

Here, the emphasis shifted from strictly parent-child relationships to other kinds of relationships, but required that these relationships be pre-defined, typically on the basis of Entity-Relationship charts, which showed how each element of data was related to other elements of data. This permitted many-to-many types of relationships, but required that these relationships be defined before the data could be stored, let alone retrieved.

Relational Databases [our main concern]

Finally, these DBMS are completely different! Here, the data itself defines the relationships, so that queries can investigate values and relationships which were not defined when the database was designed. More about relational databases in later lectures!

Merits and Roles of Database Management Systems

- Control of data redundancy: although the database approach does not remove redundancy completely, it controls the amount of redundancy in the database.
- Data consistency: by removing or controlling redundancy, the database approach reduces the risk of inconsistencies occurring. It ensures all copies of the idea are kept consistent.
- More information from the same amount of data: with the amalgamation of the operated data, it is possible to derive additional information for the same data.
- Sharing of data: database belongs to the entire organization and can be shared by all authorized users.
- Improved data integrity: database integrity provides the validity and consistency of stored data. Integrity is usually expressed in terms of constraints, which are consistency rules that the database is not permitted to violate.
- Improved security: provides protection of data from unauthorized users. It will require user names and passwords to identify user type and their access right in the operation including retrieval, insertion, updating and deletion.
- Enforcement of standards: the integration of the database enforces the necessary standards including data formats, naming conventions, documentation standards, update procedures and access rules.
- Economy of scale: cost savings can be obtained by combining all organization's operational data into one database with applications to work on one source of data.
- Balance of conflicting requirements: by having a structural design in the database, the conflicts between users or departments can be resolved. Decisions will be based on the

base use of resources for the organization as a whole rather than for an individual person.

- Improved data accessibility and responsiveness: by having integration in the database approach, data accessing can cross departmental boundaries. This feature provides more functionality and better services to the users.

- Increased productivity: the database approach provides all the low-level file-handling routines. The provision of these functions allow the programmer to concentrate more on

the specific functionality required by the users. The fourth-generation environment provided by the database can simplify the database application development.

- Improved maintenance: provides data independence. As a change of data structure in the database will affect the application program, it simplifies database application

Maintenance.

- Increased concurrency: database can manage concurrent data access effectively. It ensures no interference between users that would not result any loss of information or loss of integrity.

- Improved backing and recovery services: modern database management system provides facilities to minimize the amount of processing that can be lost following a failure by using the transaction approach.

Demerits

- Complexity: A difficult software. All users must be familiar with its functionality and take full advantage of it. Therefore, training for the administrators, designers and users is required.

- Size: Uses a substantial amount of main memory as well as large amount of disk space in order to make it run efficiently.

- Cost of DBMS: A multi-user database management system may be very expensive. Even after the installation, there is a high regular annual maintenance cost on the software.

- Cost of conversion: When moving from a file-base system to a database system, the company is required to have additional expenses on hardware acquisition and training cost.

- Performance: As the database approach is to cater for many applications rather than exclusively for a particular one, some applications may not run as fast as before.

- Higher impact of a failure: Increases the vulnerability of the system due to the centralization. As all users and applications reply on the database availability, the failure of any component can bring operations to a halt and affect the services to the customer seriously.