UNIT 10 DATABASE ARCHITECTURE

Structure

10.0 OBJECTIVES

The database approach in information retrieval has been a significant development in meeting information requirements in multiple user environments.

After reading the Unit, you will be able to :

- understand the evolutionary pattern of development of database approach;
- comprehend the need for a database management system (DBMS) and its primary objectives; and
- comprehend database architecture with its three levels.

10.1 INTRODUCTION

The need for a database approach was felt to overcome the constraints inherent in file-oriented access methods. Data independence was one of the major goals for the evolution of a database management system.

The three-level database architecture provides the necessary framework for implementing the logical and physical data independence. The architecture is also sometimes referred to as ANSI/SPARC (American National Standards Institutions/ Standards Planning and Requirements Committee) architecture.

The three schemas (levels) in the architecture are only descriptions of data. The only data that actually exists is at the physical level.

10.2 DEFINITIONS AND BASIC CONCEPTS

Some commonly used database terms have been defined in the following paragraphs to help you understand the data architecture concepts.

10.2.1 Data and Information

Data is the raw material from which information is derived as the end product. Data represents a set of characters that have no meaning on their own, i.e., it consists of just symbols. On processing, meaning is attached to data which is transformed into information.

To illustrate the difference between these two terms let us consider an example. The digits 050643 as such have no meaning. But if we are told that the first two digits represent the month, the next two digits the day of the month and the last two the year, then the set 050643 may represent the date of birth of a person. Processed in another manner, the same digits written as 643050 may represent the telephone number of an individual.

10.2.2 Database and Database Management System (DBMS)

10.2.2.1 Database

A database is a collection of logically related data arranged in a structured form, designed to meet the information requirements of multiple users. It may also be defined as a collection of non-redundant operational data sharable between different application systems.

10.2.2.2 Database Management System (DBMS)

DBMS is a collection of software that is used to store, delete, modify and retrieve data that is stored in a database. DBMS acts as an interface between the user and the data (Fig. 1.1).

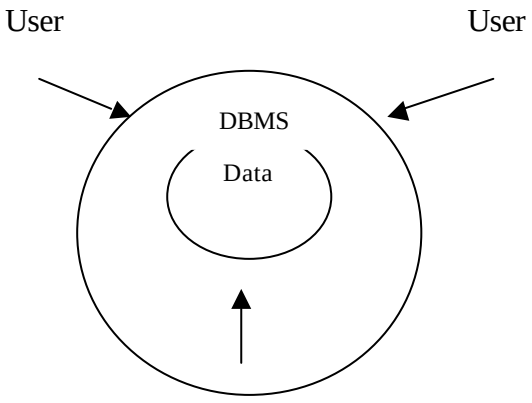

Fig. 1.1 : Data, DBMS and Users

10.2.3 Data Hierarchy

In database approach, data is organized in a hierarchy starting from the lowest unit of data represented as BIT. Hierarchy in the organization of data in the descending order of complexity has been represented in Fig. 1.2.

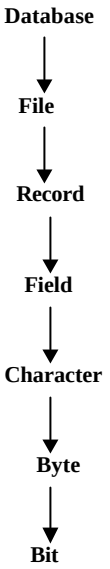

Fig. 1.2 : Data hierarchy

From this hierarchy it is clear that a database is made up of files. Files are composed of records and each record consists of fields or data items. Each field is composed of characters which are made up of bytes. And lastly, bytes decompose into bits.

10.2.4 Data Integrity

The degree of correctness, timeliness, and relevance of data stored in a database indicates data integrity. Data integrity can be ensured by certain checks and validation procedures carried out whenever an update operation is attempted and also by the elimination of data redundancy. Some database management systems have features which support data validation. For example in ORACLE (a relational database management system), triggers can be used for this purpose.

10.2.5 Data Independence

Data independence is a property by which data files are insulated from application programmes that use those files. The close link between the data and the access programme is weakened and the database made more flexible for user requirements. With data independence, a programmer can write programmes in FORTRAN, PASCAL, or any other language of his choice without bothering what language programmes originally created the files that he wants to access.

Data independence can be logical, physical or geographical. By logical or physical data independence we mean the ability to change the logical or physical structures of data without requiring any changes in the application programmes which manipulate that data. Geographical data independence, a characteristic of distributed database management systems, makes location of data transparent to the users.

Data independence is an important concept which will be considered in more detail while discussing the architecture of database management systems.

Self Check Exercise

Note: i) Write your answer in the space given below.

ii) Check your answer with the answers given at the end of this unit.

.....

.....

.....

.....

10.3 OBJECTIVES OF DATABASE MANAGEMENT SYSTEMS (DBMS)

10.3.1 The Need for DBMS

Let us consider the scenario of data processing before the advent of database management systems. In the file-oriented system, which was used earlier, a master file (the file which contains all the up-to-data data on a subject) is created using a programming language. Access techniques based on the requisite queries on the data are embedded in the file at the time of its creation. Any change in the master file, i.e., addition of a new field or change in the structure of an existing field, has to be implemented by creating the file all over again along with the modified access techniques.

To illustrate the limitations of data management using master files let us consider the example of a database of an educational institution running professional courses. Let us assume that the database consists of three master files of student, faculty and course records. The student master file has been created using FORTRAN and has such fields as student identification number, name, address, gender, course, high-school grade and examinations cleared.

The faculty master file uses COBOL and consists of fields like faculty identification number, name, gender, department, salary, qualifications and teaching hours. The course master file is based on PASCAL and covers such data as course identification number, course title, class number, section number and students attending the course.

Now suppose there is a query to provide the names of all the female students being taught by female faculty members. This query cannot be answered by the available master file despite the fact that the data needed for the query exists in the database. The difficulty lies in the fact that the needed data is available in two master files created in different programming languages and having their own access techniques. To answer the query a new master file with data items derived from the student and faculty master files will have to be created and new programmes for accessing the data written. This makes data retrieval cumbersome and time-consuming.

Take another situation when the accounts department of the institution in the example also wants to use the database and needs the student master file with additional fields like stipend paid, fees due, penalties charged etc. To meet the requirement, another copy of the student master file with new fields is created. Similarly, there may be copies of faculty and course master files created to meet specific requirements. This results in duplication of data, i.e., data redundancy. In such circumstances it

becomes difficult to keep the master files identically updated, i.e., propagate the updates in all the copies.

These limitation and drawbacks were at the core of development of database management systems.

10.3.2 The Goals of DBMS

The database management systems have the following goals :

i) To provide retrieval flexibility. It should be relatively easy to link data from different files.

ii) To facilitate reduction of data duplication and elimination of multiple copies of a master file. Data redundancy control helps in overcoming updating problems and promotes data integrity.

iii) To ensure a high level of data independence. The data is hidden from the programming language, operating system and processing environment. It should be up to DBMS to convert the stored data into a form that could be used in whatever language the programmer desires to use.

10.4 EVOLUTION OF DBMS

The database management concepts have undergone tremendous changes over the years. The chronological evolution of database management systems has been described below :

1.4.1 Chronology

i) **Primitive (first generation) Systems**

This stage is illustrated in Fig. 1.3a. Here programming overheads are high as the user has the responsibility for all the I/O (Input/Output) programming as well as file management. Application programmes (software that supports end user activity) have to be written to access the data stored in secondary storage devices-DASD (Direct Access Storage Devices). This phase represents the period of the mid 60s.

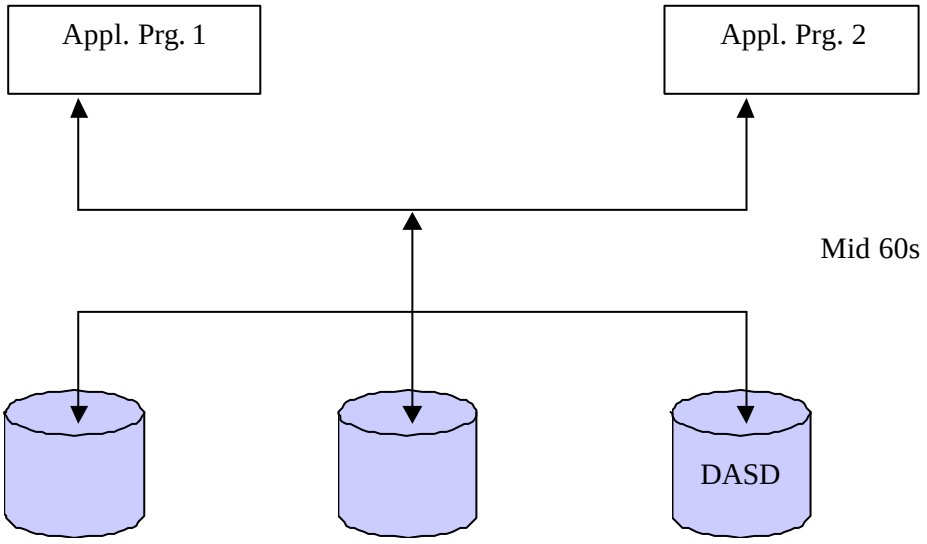

Fig. 1.3a : First generation systems

ii) Second Generation Systems

Appearing in the late 60s this generation has been represented in Fig. 1.3b. In this case, the operating system takes care of file management and other routines, thereby relieving the programmer from this effort. Though physical data independence is achieved, logical data independence remains to be built into the system.

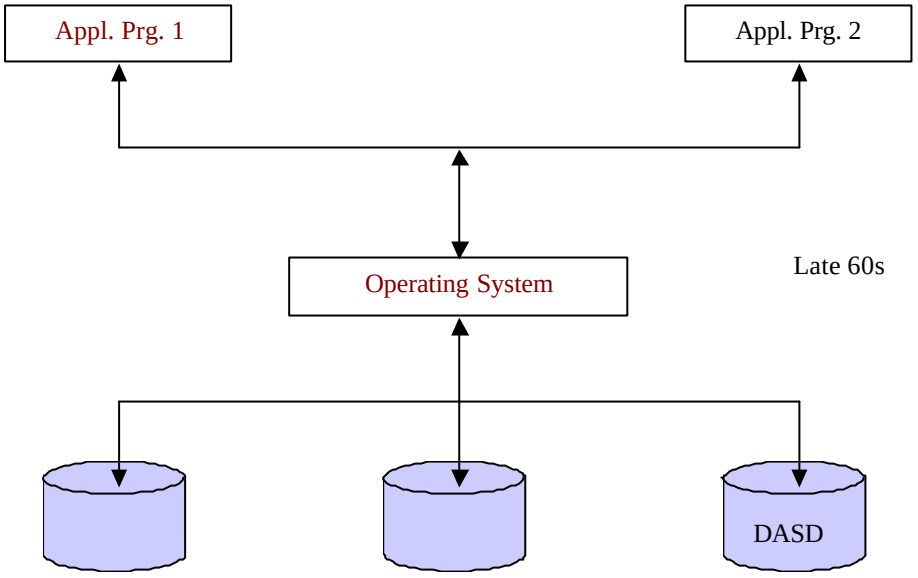

Fig. 1.3b : Second generation systems

iii) Third Generation Systems

Third generation systems (Fig. 1.3c) evolved in the early 70s. Here a layer of DBMS has been added over the operating system. The systems provide physical as well as logical data independence. The advantage it gives is that query processing can be attempted by offering higher levels of operation on the logical view of the data available from DBMS to the application programs.

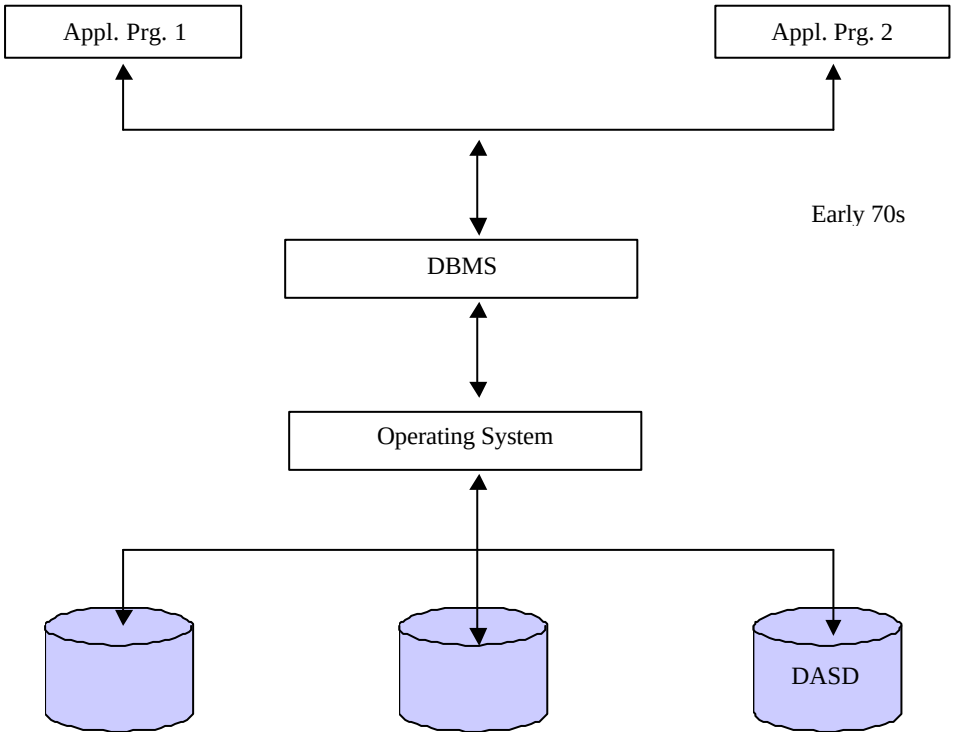

Fig. 1.3c : Third generation systems

10.4.2 Functions and Components of a DBMS

Basically there are only two operations that can be performed on data viz., retrieval and maintenance. Retrieval refers to reading data from files to serve the information requirements of a user and forms the most important function of a database management system. Maintenance concerns changing of data in stored files.

Data maintenance involves three operations : addition, deletion and modification which correspond to adding new records, deleting existing records and modify/ updating values in the existing records.

A database management system has two essential components : data definition part and data manipulation part. The data definition part provides definition or description of database objects and is written using data definition language (DDL). This part creates logical structures of entities when a database is set up, for example the fields and sub fields to be created, the length, type, etc.

The data manipulation part refers to methods of manipulating data and is implemented using data manipulation languages (DML). There are four basic methods of data manipulations : programming language interface, query languages, report writers and system utilities.

Programming language interface (PLI) or host language interface provides access to the database through some type of programming language (PASCAL, C, COBOL, etc.). Query languages allow fast retrieval of data and some of them are considered fourth generation languages (4GL), 4GLs are non-procedural languages which implies that a user has to specify only what data is required and not how it should be retrieved.

The query languages can be grouped into two categories – command-driven query languages and screen-oriented query languages. In the first case the commands are specified in English-like text, while in the second case the user enters commands through a fill-in-the blank mechanism. SQL (structured query language), a 4GL and a standard language for interfacing with relational DBMS belongs to the first category while querying data through SQL forms is an example of the second category.

Report writers represent programmes which are used to derive information from a database and generate a report that presents the information in the desired fashion. And, lastly, system utilities are programmes that allow a system manager to take back up of databases, load data into a database, restore data in case of database crash and other jobs related to database administration.

10.5 ARCHITECTURE OF A DBMS

Database management systems have a three-level architecture (see Fig. 1.4). Schema, level and view are used interchangeably to describe the architecture of a DBMS. The uppermost level in the architecture is the external level which refers to the way the users view the data. The external level is also sometimes called subschema. A user will generally be interested only in some portion of the total database which will form his external view. There may be several external views of the same database depending upon the user requirement. External schema can be used for implementing data security by restricting access to the database.

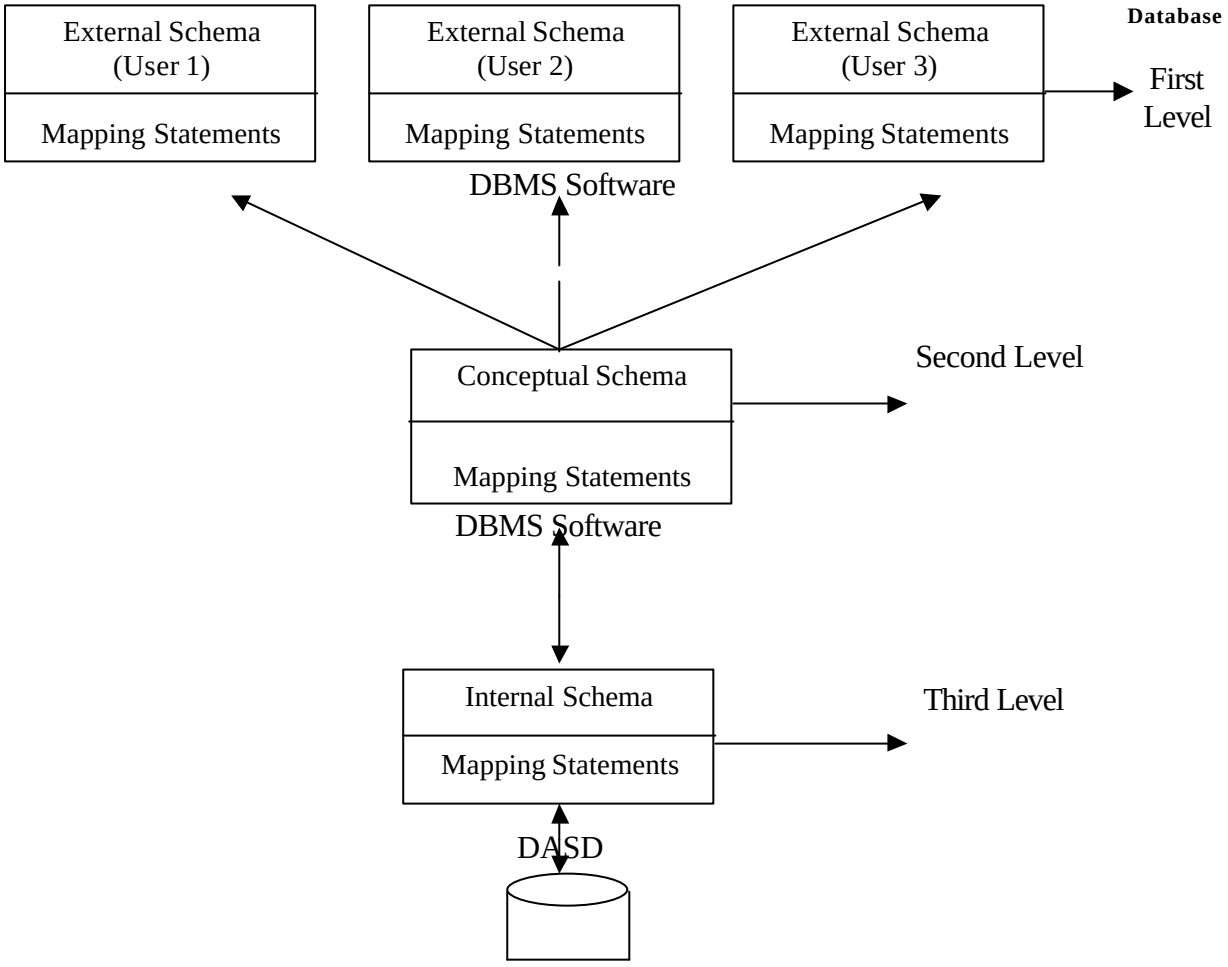

Fig. 1.4 : Architecture of a DBMS

The second level is the conceptual schema which represents the entire information content of the database. It gives a global or integrated view of the database – the view of the database administrator. The conceptual schema is written using the conceptual DDL (Data Definition Language).

Mapping statements (software programmes) at the first level which establish correspondence between the external and conceptual schemas provide data independence which implies that one can define additional database objects or modify the structures of the existing ones without requiring any existing user to change his application programmes. With logical data independence, a conceptual schema can grow and evolve with time without affecting the external schemas.

The third level of the architecture is the internal schema. The internal view describes how the data is actually stored and managed on the computer's secondary storage. It specifies what indexes exist, how stored fields are represented, the physical sequence of stored records, etc. The internal schema is written using the internal DDL. Conceptual/internal mapping statements ensure physical data independence – that is, the way the data is actually stored can be changed without requiring any change in the conceptual schema.

The mapping component between the internal schema and secondary storage devices (direct access storage devices) is called access method. The access method consists of a set of routines whose function is to conceal all the device-dependent details from the DBMS and present the DBMS with the stored record view i.e., the internal view. In many cases the underlying operating system performs this function.

Self Check Exercise

2) What is a schema and a sub-schema in database system? Briefly explain the purpose views in database architecture.

Note: i) Write your answer in the space given below.

ii) Check your answer with the answers given at the end of this unit.

.....

.....

.....

.....

10.6 SUMMARY

The three-schema architecture is a convenient tool for the user to visualise the schema levels in a database system. Most DBMSs do not separate the three levels completely, but support the three-schema architecture to some extent. Some DBMSs may include physical-level details in the conceptual scheme. In most DBMSs that support user views, external schemas are specified in the same data model that describes the conceptual-level information. Some DBMSs allow different data models to be used at the conceptual and external levels.

A DBMS must transform a request specified on an external schema into a request against the conceptual schema, and then into a request on the internal schema for processing over the stored database. The process of transforming requests and results between levels are called mappings.

The three-schema architecture makes it easier to achieve true data independence, both physical and logical. However, the two levels of mappings create an overhead during compilation or execution of a query or programme leading to inefficiencies in the DBMS. Because of this, few DBMSs have implemented the full three-schema architectures.

10.7 ANSWERS TO SELF CHECK EXERCISES

1) In database systems, data independence is an important concept. It refers to separation of data from the programme that use it. Data independence enables the data definition to be changed without altering the programmes.

Data independence can be either physical, logical or geographical (distributive).

Physical independence means that one can change the way the data is actually stored or accessed in the system without requiring changes in the application programmes. Logical independence means that the database can be reorganised i.e. changes can be made in the conceptual level but still the same application programmes can be used. Geographical independence implies that the application programmes are not affected by the location of data, i.e., whether the data is located on a local disk or on a remote file server. Data independence imparts great flexibility in handling data in database systems.

2) A schema is an overall conceptual or logical view of data in a database collection of all the tables in a database. The schema contains a list of all the fields, the field types, the maximum and minimum acceptable values for each field, along with information about the structure of every row in every table of the database.

   A subset or transformation of the logical view of the database schema that is required by a particular user application programmes is called a subschema. A subschema is an individual view of the database. Each individual user may have a separate view of the database depending upon the user requirements. Access to the entire database, i.e., conceptual schema, is generally not given to all the users. A view helps to provide access to only that part of the database which a user actually needs. The purpose of the views is not only to rationalise database access but also to implement security aspects.

10.8 KEYWORDS

Access Method : The method used to store, find and retrieve the data from a database.

Data Independence : Separates the data from the programme, which often enables data definition to be changed without altering the programme.

Data Integrity : Keeping accurate data which means few errors and the data reflect the true state of a database.

Schema : An overall conceptual or logical view of the relationships between the data in a database.

Subschema : A subset or transformation of the logical view of the database schema that is required by a particular user application programme.

10.9 REFERENCES AND FURTHER READING

James, F. Courtney, David, B. Paradice. (1988). Database Systems for Management. Toronto: Times Mirror/Mosby College Publishing.

Date, C.J. (1989). Introduction to Database Systems. New Delhi : Narosa Publishing House.

Ramaz Elmasri, Shaukan B. Navathe (2000). Fundamentals of Database Systems: Pearson Education Asia.

Gerald V. Post (2000). Database Management Systems : Tata McGraw-Hill Publishing Company Limited, New Delhi.

James Martin (1988). Principles of Database Management. New Delhi : Prentice-Hall of India Private Limited.

Jeffrey, D. Ullman, (1991). Principles of Database Systems. New Delhi : Galgotia Publications (P) Ltd.