Sairam INSTITUTIONS

Sri SAI RAM ENGINEERING COLLEGE
INSTITUTE OF TECHNOLOGY
West Tambaram, Chennai - 44

YEAR	SEM
II	IV

CS8492

DATABASE MANAGEMENT SYSTEMS

UNIT NO. I

INTRODUCTION

1.1 PURPOSE OF DATABASE SYSTEM
VIEWS OF DATA

INTRODUCTION

DATABASE

Database is collection of data which is related by some aspect. Data is collection of facts and figures which can be processed to produce information. Mostly data represents recordable facts. Data aids in producing information which is based on facts. A **database management system** stores data, in such a way which is easier to retrieve, manipulate and helps to produce information.

So a database is a collection of related data that we can use for

- Defining - specifying types of data
- Constructing - storing & populating
- Manipulating - querying, updating, reporting

DISADVANTAGES OF FILE SYSTEM OVER DATABASE

In the early days, File-Processing system is used to store records. It uses various files for storing the records. Drawbacks of using file systems to store data:

- **Data redundancy and inconsistency**
  - Multiple file formats, duplication of information in different files
- **Difficulty in accessing data**
  - Need to write a new program to carry out each new task
- **Data isolation** — multiple files and formats
- **Integrity problems**
  - Hard to add new constraints or change existing ones
- **Atomicity problem**

  - Failures may leave database in an inconsistent state with partial updates carried out. E.g. transfer of funds from one account to another should either complete or not happen at all
- **Concurrent access anomalies.**

  - Concurrent accessed needed for performance

- **Security problems.**

Database systems offer solutions to all the above problems.

1.1 PURPOSE OF DATABASE SYSTEM

The typical file processing system is supported by a conventional operating system. The system stores permanent records in various files, and it needs different application programs to extract records from, and add records to, the appropriate files. A file processing system has a number of major disadvantages.

- 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

1. Data redundancy and inconsistency:

In file processing, every user group maintains its own files for handling its data processing applications.

Example:

Consider the **UNIVERSITY** database. Here, two groups of users might be the course registration personnel and the accounting office. The accounting office also keeps data on registration and related billing information, whereas the registration office keeps track of student courses and grades. Storing the same data multiple times is called data redundancy. This redundancy leads to several problems.

•Need to perform a single logical update multiple times.

•Storage space is wasted.

•Files that represent the same data may become inconsistent.

Data inconsistency is the various copies of the same data may no larger Agree.
Example:

One user group may enter a student's birth date erroneously as JAN-19-1984, whereas the other user groups may enter the correct value of JAN-29-1984.

2. Difficulty in accessing data

File processing environments do not allow needed data to be retrieved in a convenient and efficient manner.

3. Data isolation

Because data are scattered in various files, and files may be in different formats, writing new application programs to retrieve the appropriate data is difficult.

4. Integrity problems

The data values stored in the database must satisfy certain types of consistency constraints. Example:

The balance of certain types of bank accounts may never fall below a prescribed amount. Developers enforce these constraints in the system by addition appropriate code in the various application programs

5. Atomicity problems

Atomic means the transaction must happen in its entirety or not at all. It is difficult to ensure atomicity in a conventional file processing system.

Example:

Consider a program to transfer $50 from account A to account B. If a system failure occurs during the execution of the program, it is possible that the $50 was removed from account A but was not credited to account B, resulting in an inconsistent database state.

6. Concurrent access anomalies

For the sake of overall performance of the system and faster response, many systems allow multiple users to update the data simultaneously. In such an environment, interaction of concurrent updates is possible and may result in inconsistent data. To guard against this possibility, the system must maintain some form of supervision.

But supervision is difficult to provide because data may be accessed by many different application programs that have not been coordinated previously.

Example: When several reservation clerks try to assign a seat on an airline flight, the system should ensure that each seat can be accessed by only one clerk at a time for assignment to a passenger.

7. Security problems

Enforcing security constraints to the file processing system is difficult.

APPLICATIONS OF DATABASE

Database Applications

- **Banking:** all transactions
- **Airlines:** reservations, schedules
- **Universities:** registration, grades
- **Sales:** customers, products, purchases
- **Manufacturing:** production, inventory, orders, supply chain
- **Human resources:** employee records, salaries, tax deductions
- **Telecommunication:** Call History, Billing
- **Credit card transactions:** Purchase details, Statements

VIEWS OF DATA

It refers that how **database is actually stored in database, what data and structure of data used by database for data.** So describe all this database provides user with views and these are

- Data abstraction
- Instances and schemas

Data abstraction

As a data in database are stored with very complex data structure so when user come and want to access any data, he will not be able to access data if he has go through this data structure. So to simplify the interaction of user and database, DBMS hides some information which is not of user interest; a this is called **data abstraction**: - **So developer hides complexity from user and store abstract view of data.**

Data abstraction has three levels of abstractions

- Physical level (internal level)
- Logical level (conceptual level)
- View level (external level)

Physical level:-

The physical or the internal level schema describes **how the data is stored in the hardware.** It also describes how the data can be accessed. The physical level shows the data abstraction at the lowest level and it has complex data structures. Only the database administrator operates at this level.

Logical level:-

The next level of abstraction describe what data are stored in the database and what are the relationship existed among those of data.

Here, the data is stored in the form of the entity set, entities, their data types, the relationship among the entity sets, user operations performed to retrieve or modify the data and certain constraints on the data. Well adding constraints to the view of data adds the security. As users are restricted to access some particular parts of the database.

It is the developer and database administrator who operates at the logical or the conceptual level.

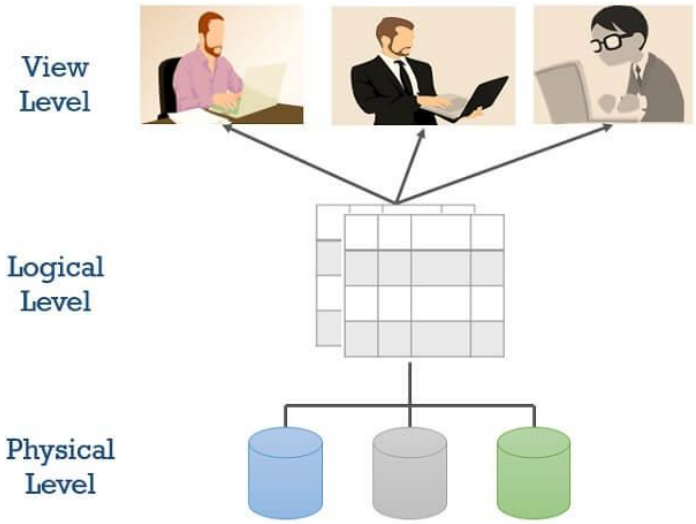
View level:-

In this level user only interact with database and the complexity remain unview. user see data and there may be many views of one data like chart and graph.

It is the highest level of data abstraction and exhibits only a part of the whole database. It exhibits the data in which the user is interested.

The view level can describe many views of the same data. Here, the user retrieves the information using different application from the database.

- The figure below describes the three-schema architecture of the database:



Three-Schema Architecture

In the figure above you can clearly distinguish between the three levels of abstraction. To understand it more clearly let us take an example: **College Database**

Now, what entity sets would be involved? Student, Lecturer, Department, Course and so on…

Now, the entity sets Student, Lecturer, Department, Course will be stored in the storage as the consecutive blocks of the memory location. This is the **physical or internal level** and is hidden from the programmers but the database administrator is it aware of it.

At the logical level, the programmers define the entity sets and relationship among these entity sets using a programming language like SQL. So, the programmers work at the logical level and even the database administrator also operates at this level.

At the view level, the users have the set of applications which they use to retrieve the data they are interested in.