Date
Page 1

# Chapter 1: Introduction to Database Management System

## * Introduction

A database-management system is a collection of interrelated data and a set of programs to access those data. The collection of data, usually referred to as the database, contains information relevant to an enterprise. The primary goal of a DBMS is to provide a way to store and retrieve database information that is both convenient and efficient.

## * Database-System Applications

Banking: For customer, information, accounts, loans and banking transaction

Universities: for student information, course registration and grades

Sales: For customer, product and purchase information

Scanned by CamScanner

* Purpose of DataBase System

Database systems arose in response to early methods of computerized of commercial data As an example of such methods, typical of the 1960s, consider a part of the a bank enterprise that, among other data, keeps information about all computers Coust customers of savings accounts. one way to keep the information on a computer is to store it in operating files. To allow users to manipulate The information, the system has a number of application programs that manipulates the files, including programs to:

- Debit or credit an account
- Add a new account
- Find balance of an account.

The typical file-processing system is supported by a conventional operating system. The system stores permanent records in various files, and it needs diff application programs to extract records from, and add records to, the appropriate files. Before database management systems (DBMs) came along, organizations usually stored information in such systems.

MRP ₹ 130/- For Educational Use And

Scanned by CamScanner

Keeping organizational information in a file-processing system has a number of major disadvantages:

- Data redundancy and inconsistency
  - Since different programers create the files & application programs over a long period of time the various files are likely to have diff structures and the programs may be written in several programming languages
  - moreover, the same information may be duplicated in several places (files).
  - For example, the address and telephone number of a particular customer may appear in a file that consists of savings-accounts records and in the file that consits checking-acc records. This redundancy leads to higher storage and access cost.
  - In addition, it may lead to data inconsistency that is the various copies of the same data may no longer agree.
  - for example, a changed customer address may be reflected in savings-account records but nowhere else in the system

Scanned by CamScanner

- Difficulty in accessing data:
- Suppose that one of the bank officers needs to find out the names of all customers who live within a particular postal-code area.
- The officer ask the data-processing dept to generate such a list
- Because the designers of the original system did not anticipate this request, there is no application program on hand to meet it.
- There is however, an application program to generate the list of all customers.
- The bank officers now has 2 options; either obtain list of all customers and extract the needed information manually or ask a system programmer to write the neccessary. Both alternatives are obviously unsatisfactory
- Suppose that such a program is written, and that, several days later, the same officer needs to trim the same list and needs only those customers who have an account balance of $ $10,000 or more. As expected, a program to generate such a list does not exist.

Scanned by CamScanner

- Again, the officer has the preceding two options, neither of which is satisfactory.

The point here is that conventional file-processing environment do not allow needed data to be retrieved in convenient and efficient manner. More responsive data-retrieval system are required for general use.

- 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.

- Integrity Problems:

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

- For example, the balance of certain types of bank accounts may never fall below a prescribed amount (say, $25). Developers enforce these constraints in the system by adding appropriate code in the various application program.

Scanned by CamScanner

- However, when new constraints are added, it is difficult to change the programs to enforce them. The problem is compounded when constraints involve several data items from different files.

- Atomicity Problems

- A computer system, like any other mechanical/electrical device is subject to failure.

- In many applications, it is crucial that, if a failure occurs, the data be restored to the consistent state that existed prior to the failure.

- 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 ~~has~~ ~~been~~ ~~debited~~ was removed from account A but was not credited to B, resulting in an inconsistent database state.

- Clearly, it is essential to database consistency that either both the credit and debit occur, or that neither occur. i.e fund transfer must be atomic - it must happen in its entirety or not at all

MRP ₹ 130/- For Educational Use And

Scanned by CamScanner

Date
Page 7

## View Of Data

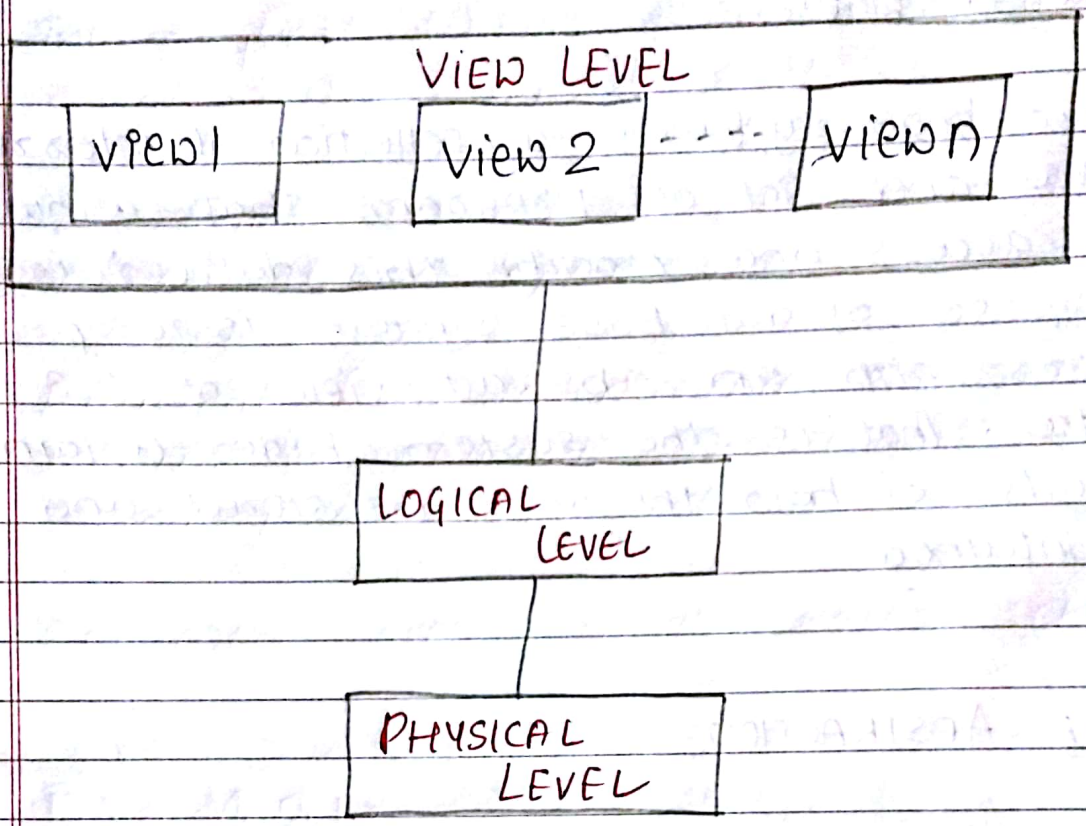
A database system is a collection of interrelated data and set of programs that allow user to access and modify this data. A major purpase of database system is to provide users with an abstract view of the data. That is, the system hides certain details of how the data are stored and maintained.

## DATA ABSTRACTION

For the system to be usable, it must retrieve data efficiently. The need for efficiency has led users to designers to use complex data structures to represent data in the database.

Since many database-system users are not computer trained, developers hide the complexity from users through several levels of abstraction, to simplify users interactions with system.

Scanned by CamScanner




- Physical level :- The lowest level of abstraction describes how the data are actually stored. The physical level describes complex low-level data structures in detail.

- Logical Level : The next-higher level of abstraction describes what data are stored in the database and what relationships exist among those data.

```
type Instructor = ~~record~~ record
        ID : String ;
        Salary : Integer ;
end;
```

Scanned by CamScanner

- View Level: The highest level of abstraction describes only a part of entire database.
  - Even though the logical level uses simpler structures, complexity remains because of the variety of information stored in large database.
  - many of the users of the database system do not need all this information; instead, they need to access only a part of the database
  - The view of the level of abstraction exists to simplify their interaction with the system.
  - The system may provide many views for the same database.

## INSTANCES AND SCHEMAS

Databases change over time as information is inserted and deleted

The collection of information stored in the database at a particular moment is called an instance of the database.

The overall design of the database is called the database schema. Schemas are changed infrequently, if at all

Database systems have several schemas, partitioned according to the levels of abstraction.

Scanned by CamScanner

The physical schema describes the database design at the physical level

The logical schema describes the database design at the logical level

Physical Data Independence - the ability to modify the physical schema without changing the logical schema

- Applications depend on the logical schema

## * DATA MODELS

Underlying the structure of a database is called data model; a collection of conceptual tools for describing data, data relationships, data semantics, and consistency constraints. A data model provides a way to describe the design of a database at the physical, logical and view level.

Scanned by CamScanner

- **Relational Model:-** The relational model uses a collection of tables to represent both data & the relationships among those data.

  Each table has multiple columns, and each column has a unique name.

  The relation model is an example of a record-based model. Record-based model are so named because the database is structured in fixed-format of records of several types.

- **The Entity-Relationship Model :-** The entity-relationship (E-R) data model is based on a perception of a real world that consits of a collection of basic objects, called entities, and a relationships among those objects.

  An entity is a "thing" or "object" in a real world that is distinguishable from the other objects. It is widely used in database design.

Scanned by CamScanner

Object-Based Data Model:- The object-oriented data model is another data model that has seen increasing attention. It can be seen as extending the E-R model with notions of encapsulation, methods (functions), and object identity.

The object-relation model combines features of the object-oriented data model and relational data model.

Historically, the network data model & hierarchical do hierarchical data model preceded the relation data model.
These models are tied closely to the underlying implementation, and complicated the task of modeling data.
As a result they are used little now, except in old database code that is still in service in some places.

Scanned by CamScanner

# * DATABASE LANGUAGES

A database system provides a data-definition language to specify the database schema and a data-manipulation language to express database queries and updates.
In practice, the data-definition & data-manipulation are not two seperate languages, instead they simply form parts of a single database language, such as the widely used SQL Language

## 1] Data-Manipulation Language :-

A data-manipulation language (DML) is a language that enables user to access @ or manipulate data as organized by the appropriate data model. Types of access are:-

- Retrieval of information stored in database
- Insertion of new information in database
- Deletion of information from the database
- Modification of information stored in database

Scanned by CamScanner

eg:-
Here is an example of an SQL query that finds the name of all customers who reside in Harrison:

```
select customer.customer_name
from customer
where customer.customer_city = 'Harrison'
```

There are basically two types:-

Procedural DMLs :- require a user to specify what data are needed and how to get those data.

Declarative DMLs (also referred as non procedural DMLs) require a user to specify what data are needed without specifying how to get those data.

The portion of a DML that involves information retrieval is called a query language.

Scanned by CamScanner

## DATA-DEFINITION LANGUAGE

We specify a database schema by a set of definitions expressed by a special language called a data-definition language (DDL). The DDL is also used to specify additional properties of the data.

We specify the storage structure and access method used by the database system by a set of statements in a special type of DDL called data storage and definition language.

The data values stored in the database must satisfy certain consistency constraints.
eg:- Balance of an account should not fall below $100.

The DDL, just like any other programming language, gets as input some instructions (statements) and generates some output. The output of the DDL is placed in the data dictionary which contains metadata - that is, data about data.
The data dictionary is considered to be a special type of table, which can only be accessed and updated by the database system itself (not a regular user).

A database system consults the data dictionary before reading & modifying actual data

Scanned by CamScanner

# TRANSACTION MANAGEMENT

Often, several operations on the database form a single logical unit of work. An example is a fund transfer, in which one account (say A) is debited and ~~say~~ another account (say B) is credited.

Clearly, it is essential that either both that credit and debit occur, or that neither occur. That is fund transfer must happen in its entirety or not at all. This all or none requirement is called **Atomacity**

In addition, it is essential that the execution of the fund transfer preserve the consistency of the database. That is, the value of sum A + B must be poeserved. This correctnem requirement is called **consistency**.

Finally, after the succesful execution of fund transfer, the new values of accounts A & B must persist, despite the possibility of system failure. This persistence requirement is called **durability**

Scanned by CamScanner

A transaction is a collection of operations that performs a single logical function in a database application

Each transaction is a unit of both atomacity and consistency.

* STORAGE MANAGEMENT

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 the system

* The storage manager is responsible to the foll tasks:-

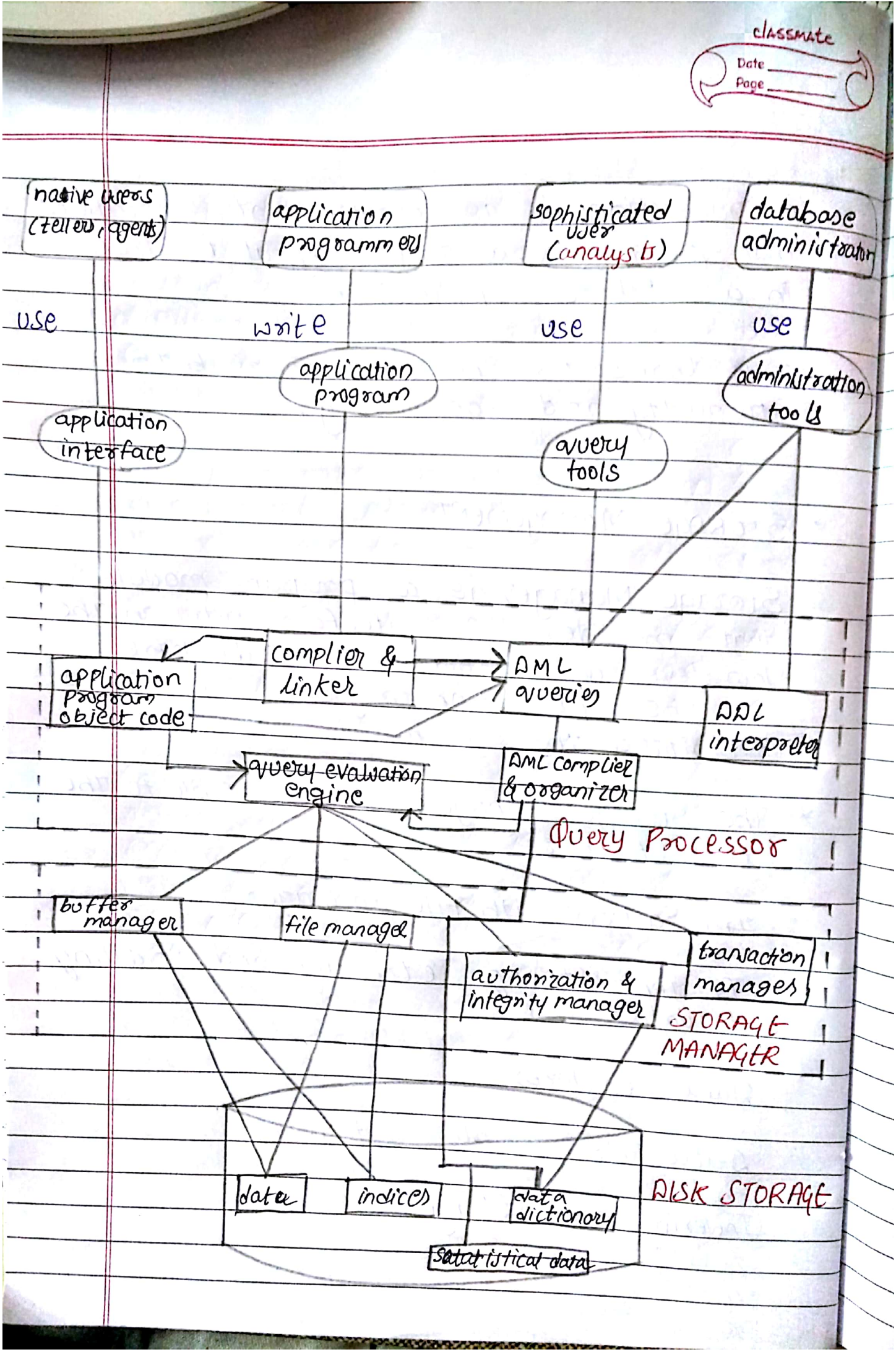
- Interaction with file manager
- efficient storing, retrieving and updating of data

Data Structures

- Data Files
- Data Dictionary

Indexing and hashing.

Scanned by CamScanner




Scanned by CamScanner

# DATABASE USERS AND USER INTERFACES

- **NAIVE USERS** are unsophisiticated users who interact with the systems by invoking one of the application programms that ~~has~~ have been written previously

For example, a bank teller who needs to transfer $50 from account A to account B invokes a program called transfer

This program asks the teller for the amount of money to be transferred, the account from which the money is to be transferred, and the account to which the money is to be transferred. Naive users may also simply read reports generated from the database

- **Application Programmers**: are computer professionals who write applications programs. Application programmers can choose from many tools to develop user interfaces. **Rapid Application Development (RAD)** tools are tools that enable an application programmer to construct forms and reports with minimal programming effort

Scanned by CamScanner

- Sophisticated Users: interact with the systems without writing programs. Instead, they form their requests in a database query language. They submit each such query to a query processor, whose function is to break down DML statements into instructions that the storage manager understands. Analysts who submit queries to explore data in the database fall in this category.

- Specialized Users are sophisticated users who write specialized database applications that do not fit into the traditional data-processing framework.

Among these applications are computer-aided design systems, knowledge base and expert systems, systems that store data for complex data types (for example, graphics data and audio data) and environment-modeling systems.

Scanned by CamScanner

## QUERY PROCESSOR

The query processor will accept query from the user and solve it by accessing the database

1. **DDL Interpreter:-** This will interpret DDL statements and fetch the definitions in the data dictionary.

2. **DML Interpreter:-** This will translate DML statements in query language into low level instructions that the query evaluation engine understands. A query can usually be translated into any number of alternative evaluation plans for same query result DML complier will select best plan for query optimization

3. **Query Evaluation Engine:-** This engine will execute low level instructions generated by the DML complier on DBMS

Scanned by CamScanner

# STORAGE MANAGER COMPONENTS

1. Authorization and Integrity manager : Checks for Integrity Constraints and authority of users to access data

2. Transaction manager : which ensures that the database remains in a consistent state although there is system failure

3. File Manager : manages allocation of space on disk storage and the data structures used to represent information stored on disk

4. Buffer Manager Responsible for retrieving data from disk storage into main memory. It is a imp part of the database system, as it enables the database to handle data sizes that are much larger then size of main memory.

Scanned by CamScanner