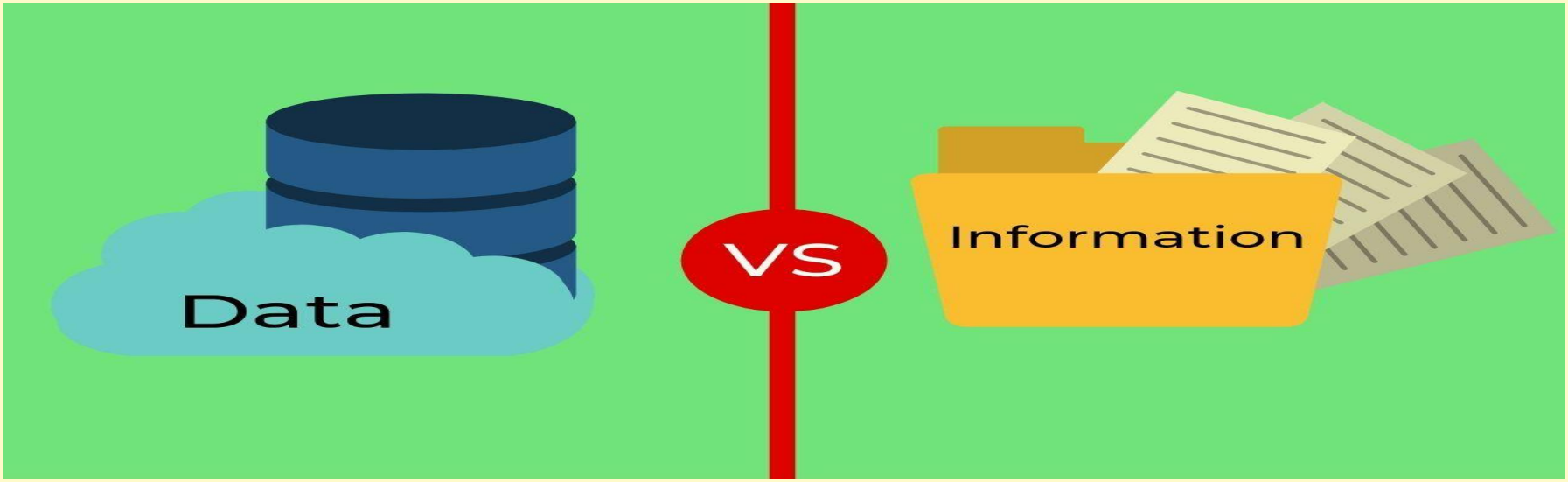
MODULE1:
DATABASE MANAGEMENT SYSTEMS

Prepared by,
Dr ARUNA M G
AI&ML

CONTENTS

- **Introduction to Databases:** Introduction, An Example, Characteristics of Database approach, Advantages of using DBMS approach, When not to use a DBMS
- **Database System Concepts and Architecture:** Data models, Schemas and instances, Three schema architecture and data independence Database languages and interfaces, The database system environment,
- **SQL:** SQL Data Definition and Data Types specifying basic constraints in SQL, Basic retrieval queries in SQL, Insert, Delete and Update statements in SQL, Additional features of SQL, More complex SQL Queries, Specifying Constraints as Assertions and Triggers, Views (Virtual Tables) in SQL, Schema Change Statement in SQL.


Data
VS
Information

DATA	INFORMATION
1) Data is collection of raw facts and figures.	1) Information is processed data.
2) Data is not arranged.	2) Information is arranged.
3) Data is unorganized.	3) Information is organized.
4) Data does not depend on information.	4) Information depends on data.
5) Data is a low-level knowledge.	5) Information is the second level of language.
6) Data or raw data is not enough to make a decision.	6) The information is sufficient to help to make a decision in respective

An example of data is a student's test score

Ex: The average score of a class is the information derived from the given data.

WHAT IS DATA?

- Data means recorded facts or Known facts that have implicit (unquestioned) meaning are stored on computer media.
- Facts, figures, statistics etc. having no particular meaning.

The Data can be:

- Structured: numbers, text, dates
- Unstructured: images, video, documents
- Example :(USN,name.age)

a) 1,Ramu, 19
b) 23,Suma,20

CONTINUED..

- ***Information***: data processed to increase knowledge in the person using the data
- ***Metadata / System catalog***: data that describes the properties or characteristics of the data, including data types, field sizes, allowable values, and data context user data.

Table 1-1 Example Metadata for Class Roster

Data Item		*Value*				
Name	**Type**	**Length**	**Min**	**Max**	**Description**	**Source**
Course	Alphanumeric	30			Course ID and name	Academic Unit
Section	Integer	1	1	9	Section number	Registrar
Semester	Alphanumeric	10			Semester and year	Registrar
Name	Alphanumeric	30			Student name	Student IS
ID	Integer	9			Student ID (SSN)	Student IS
Major	Alphanumeric	4			Student major	Student IS
GPA	Decimal	3	0.0	4.0	Student grade point average	Academic Unit

WHAT IS DATABASE?

- A database is a collection of related data or information, which stores data in form of table.
- A Database:
  - collection of interrelated data + description of data

information, which stores data in form of table. Fields

Table Name: **EMPLOYEE**

SSN	Name	Age	Address	Dno	Salary
123467	Aruna	25	#45,Bangalore	5	20000
123469	Suri	26	#35,hassan	4	12000
123468	Mohan	27	#23,Bangalore	1	15000
123567	Jagadish	28	Null	5	20000

Record

Table Name:**DEPARTMENT**

Dname	Dnumber	Dlocation
Research	5	Bangalore
Administration	4	Bangalore
Headquarters	1	Bombay

Fig :A database that stores employee and department information

DEFINITIONS OF DATABASE

- ***Def 1:*** Database is an *organized collection* of *logically related data*
- ***Def 2:*** A database is a *shared* collection of *logically related data* that is stored to meet the requirements of *different users* of an organization
- ***Def 3:*** A database is a *self-describing* collection of integrated records
- ***Def 4:*** A database models a *particular real world system* in the computer in the form of data

WHAT IS DB SYSTEM?

- ***Database System***
- A database system is a software system, which supports the definition and use of a database
- DDL: Data Definition Language
- DML: Data Manipulation Language

Salesperson Number	Salesperson Name	City	State	Office Number	Commission Percentage	Year of Hire
137	Baker	Detroit	MI	1284	10	1995

Facts about salesperson Baker.

DATABASE MANAGEMENT SYSTEM (DBMS)

- A DBMS is a collection of interrelated files and set of programs that allow users to access and modify these files.
- **or**
- A DBMS is a collection of programs that enables the users to create & maintain a database.
- DBMS is general-purpose software that facilitates the process of defining, constructing, manipulating and sharing database among various users and applications.

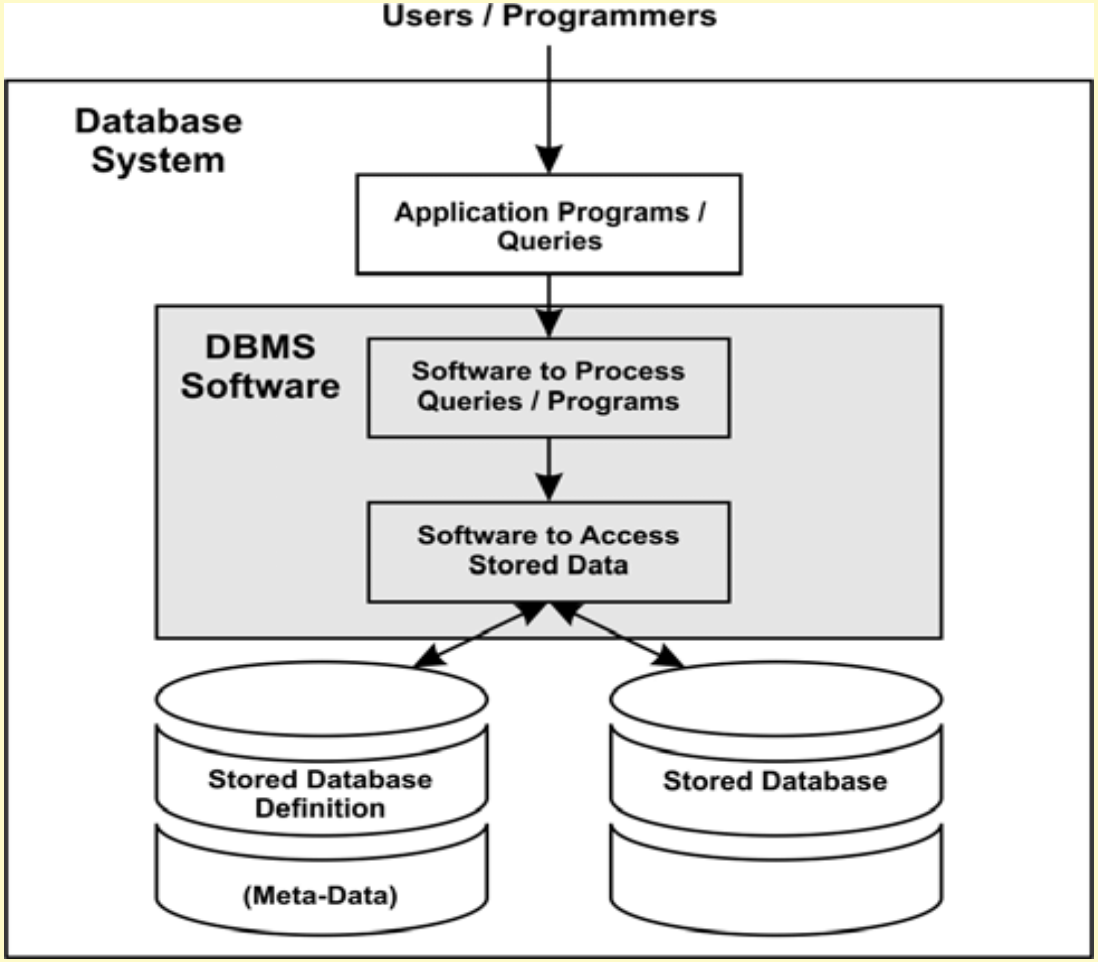
DATABASE MANAGEMENT SYSTEM (DBMS)

- A Database Management System (DBMS) is:
  - A software system designed to store, manage, and facilitate access to databases.
  - The primary goal of a DBMS is to provide a way to store and retrieve database information that is both convenient and efficient.
  - **The DBMS Functionality**
- Defining a Database: It involves the specifying types of data types, structures and constraints of the data to be stored in the database.
- Constructing a Database: The process of storing the database on a secondary storage medium.

CONTINUE..

- **Manipulating a Database:** It includes functions for querying (retrieve specific data), updating, reporting, insertions, deletions and modifications to its content.
- Concurrent processing and sharing by a set of users and programs – yet, keeping all data valid and consistent.
- **Other features:**
- **Protecting the Database**: It involves the two types of protection i.e. system protection against hardware/software (crashes) and security protection against unauthorized access.
- **Maintaining the Database:** Maintain the database over long period of time.
- Example's: Access, Oracle, MySQLServer, dBase, Sybase, Clipper, MySQL, PostgreSQL, Apache Derby, SQLite , Ingres, etc.

SIMPLIFIED DATABASE SYSTEM ENVIRONMENT



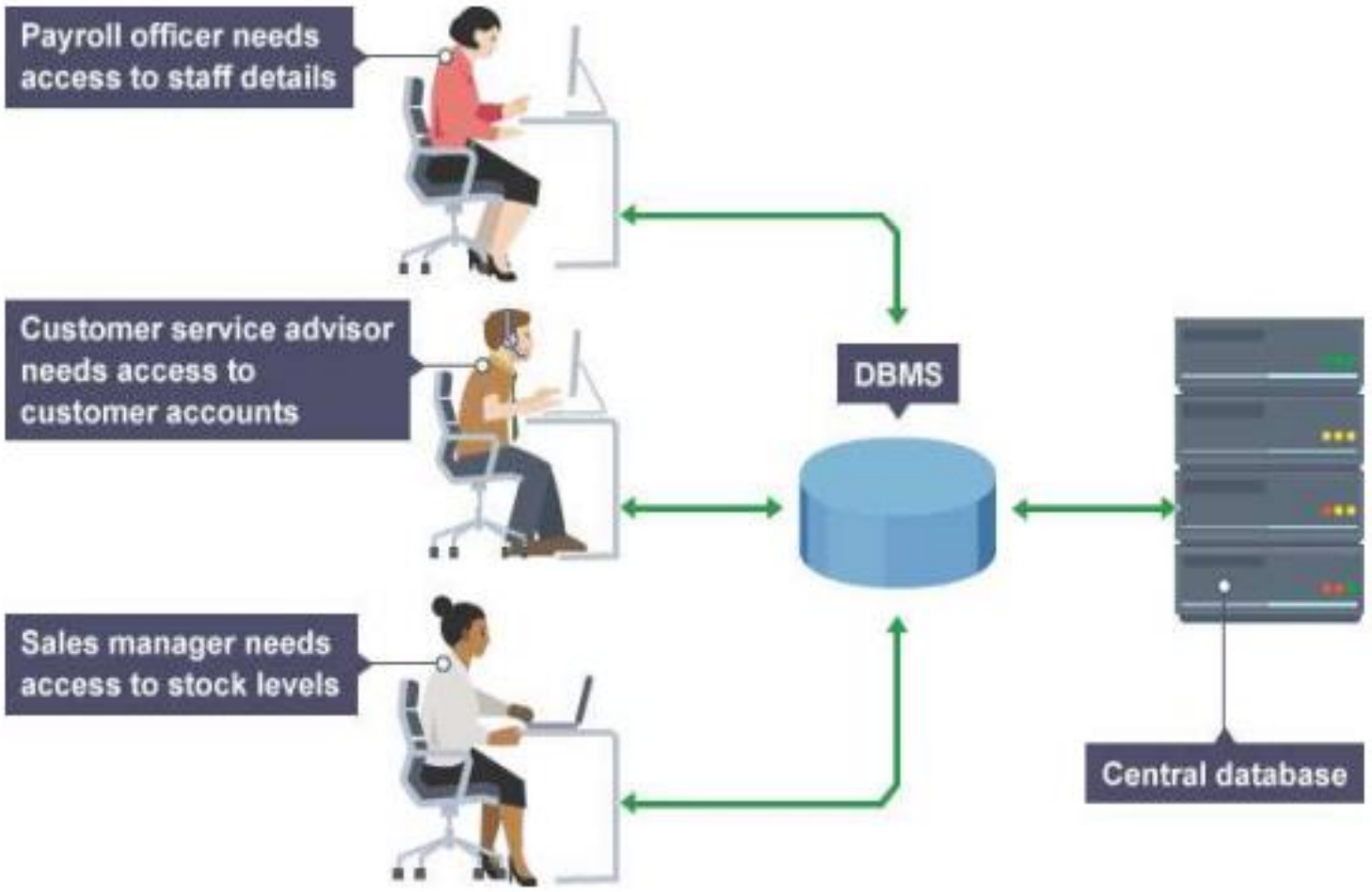


Figure 1.1: Empolyees are accessing Data through DBMS

DIFFERENT TYPES OF DBMS

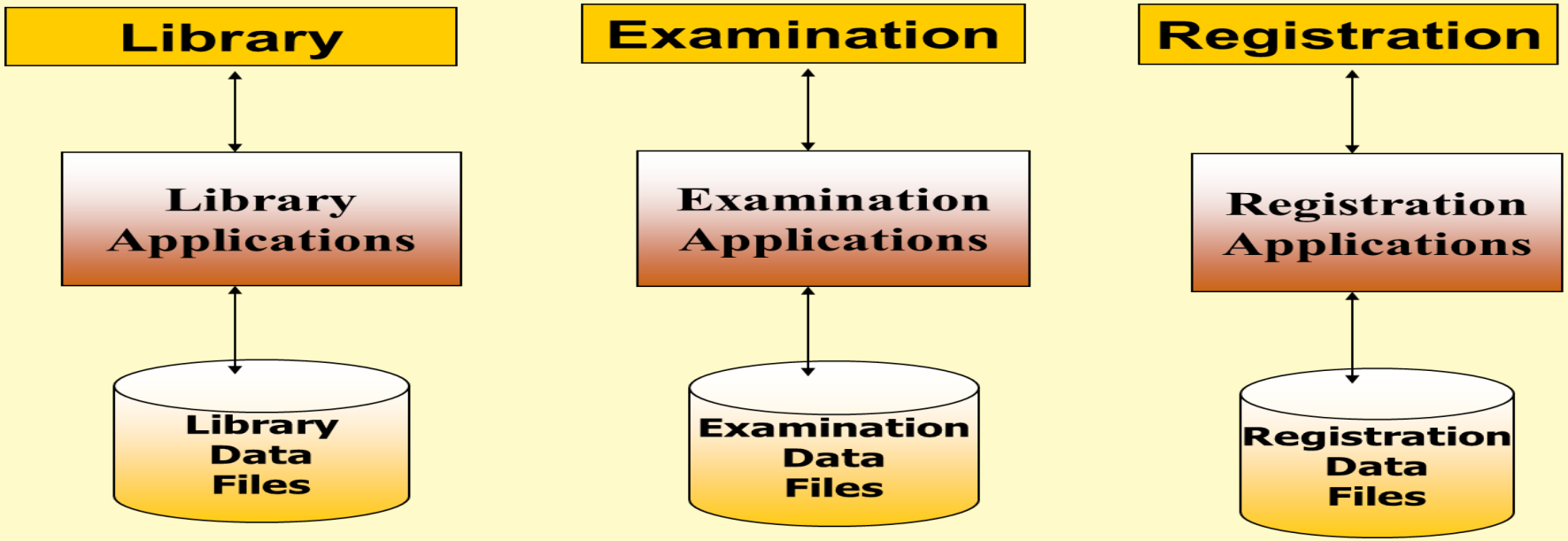
- **Traditional Database applications**: The information stored and accessed in computer either numeric or textual pattern..
- **Multimedia Databases**: Store picture, video clips and sound message.
- **Geographic Information Systems (GIS)**: It can store and analysis maps, weather data and satellite images.
- **Data Warehouses and online analytical processing (OLAP)**: This is used for decision making
- **Real-time and Active Databases**: It can be used in controlling industrial and manufacturing processes.
- **Database Search Technique:** Used in World Wide Web (www) to improve the search for information that is needed for browsing in the internet.
- **Mobile databases**

DATABASE APPLICATIONS

- Data can be stored in RAM
  - this is what every programming language offers!
  - RAM is fast, and random access
  - Isn't this heaven?
- Every OS includes a File System
  - manages *files* on a magnetic disk
  - allows *open, read, seek, close* on a file
  - allows protections to be set on a file
  - drawbacks relative to RAM?

FILE PROCESSING SYSTEM VERSUS A DBMS

- File processing system was the earliest storage type database system.
- In file processing system the permanent records are stored in various files, and different application programs are written to extract records from, and to add records to appropriate files.
- Each program defines and manages its own data.



- Program and Data Interdependence

File Processing Systems

Library
Reg_Number
Name
Father Name
Books Issued
Fine

Examination
Reg_Number
Name
Address
Class
Semester
Grade

Registration
Reg_Number
Name
Father Name
Phone
Address
Class

THE FILE SYSTEM HAS SOME DISADVANTAGE

- **Uncontrolled redundancy(**The same data stored in multiple places**)**
- **Inconsistent data(**Incorrectness in data entry)
- **Inflexibility**
- **Difficulty in accessing data**
- **Data isolation**
- **Integrity problems**
- **Atomicity problems**
- **Concurrent access anomalies**
- **Security problem**
- **Limited data sharing**
- **Poor enforcement of standards**
- **Low programmer productivity**
- **Excessive program maintenance**
- **Excessive data maintenance**

DUPLICATION OF DATA (DATA REDUNDANCY)

- Same data is held by different programs.
- Different systems/programs have separate copies of the same data
- The different values and/or different formats for the same item.

- Waste of space to have duplicate data
- Causes more maintenance headaches
- The biggest problem:

  When data changes in one file, could cause inconsistencies (Vulnerable to Inconsistency)

  Compromises data integrity (data reliability)

INCONSISTENCY DATA

- Incorrectness in data entry.
- Example: A changes made in customer address in savings-account records will be reflected but not in other files of banking system this makes the data inconsistency.
-

CUSTOMER_SAVING

Account_number	Name	Age	Balance	Address
123467	Aruna	25	20000	#45,Bangalore
123469	Suri	26	12000	#35,hassan
123468	Mohan	27	15000	#23,Bangalore
123567	Jagadish	28	20000	#323,Bangalore
123467	Arun	25	20000	#45,Bangalore

Spelling Mistake in name field

CHECKING_ACCOUNT

Account_number	Name	Address
123467	Aruna	#45,Bangalore
123469	Suri	#35,hassan
123468	Mohan	#23,Bangalore
123567	Jagadish	Null

Changes not reflected

CONTINUED..

- **Inflexibility**: If any necessary changes like adding a file or extend the data elements in an existing file the entire file has to be changed. This makes the affecting the stored data and the existing applications programs.
- Example:
  - In student table need to add email id to existing DB.
-

STUDENT

USN	Name	Age	Address
123467	Aruna	25	#45,Bangalore
123469	Suri	26	#35,hassan

CONTINUED..

- **Difficulty in accessing data**: If requested query is not designed for original system. Then check the particular file to get the requested information manually or ask the system programmer to write the needed application program.
- Example: If bank officer's needs to find out the names of all customers who live within a ***particular postal-code*** area in a city's.
- There is **no application program** on hand to meet it in the original system.
- The file-processing systems do not allow needed data to be retrieved in a convenient and efficient manner.

CONTINUED..

- **Data isolation**: The data are scattered in various files, and files may be different formats, it is difficult to write new application programs to retrieve the appropriate data.

- **Integrity problems**: The data values stored in the database must satisfy certain types of consistency constraints.
- Example: The balance of a bank account may never fall below Rs.500. Developers enforce these constraints in the system by adding code in the various application programs. If new constraints wanted to add, it is difficult to change the programs.
-

CONTINUED..

- **Atomicity problems**: If any mechanical or electrical types of failure occur to the system, the data should be restored to the consistent state. Atomic means it must happen entirely or not at all.
- Example: If J wants to transfer Rs2000 from his account to S. If a system failure occur during execution of the program, that Rs2000 is removed from J but not credited to account S, resulting in an inconsistent state. To stay in consistent state either both credit and debit occur, or that neither occur.
-

CUSTOMER_SAVING

Account_number	Name	Age	Balance	Address
123469	Suri	26	12000	#35,hassan
123468	Mohan	27	15000	#23,Bangalore
123567	Jagadish	28	18000	#323,Bangalore

Figure1.4 Atomicity problems

inconsistent state

CONTINUED..

- **Concurrent access anomalies:** System allows multiple users to update, delete and retrieve the data simultaneously.
- If two programs run concurrently, they may both read the value, and write back the value. Depending on which one writes the value last, that value maybe correct or incorrect value.
- **Example:** If two customers withdraw amount from J (say Rs200 and Rs300 respectively) at the same time, the result of concurrent executions may leave the account in an incorrect state.

- **Security problem:** Not every user of the database system should be able to access all the data.
- Example: In a student database system, updating marks is allowed only for lecturer, only viewing the data for student and restricted for updating the data.

PROGRAM-DATA DEPENDENCE

- Each application programmer must maintain their own data/program code
- Each application program needs to include code for the metadata of each file
- Each application program must have its own processing routines for reading, inserting, updating and deleting data
- Lack of coordination and central control
- Non-standard file formats

CONTINUED..

- **Limited Data Sharing**
  - No centralized control of data
  - Programs are written in different languages, and so cannot easily access each other's files.
- **Lengthy Development Times**
  - Programmers must design their own file formats
- **Excessive Program Maintenance**
  - 80% of of information systems budget
- **Vulnerable to Inconsistency**
  - Change in one table need changes in corresponding tables as well otherwise data will be inconsistent

MAIN CHARACTERISTICS OF THE DATABASE APPROACH VERSUS THE FILE PROCESSING

1. Self-describing nature of a database system
2. Insulation between programs and data, and data abstraction
3. Support of multiple views of the data
4. Sharing of data and multiuser transaction processing

CONTINUE..

- **Self-describing nature of a database system:**
  - A DBMS **catalog** stores the description of a particular database (e.g. data structures, types, and constraints)
  - The description is called **meta-data**.
  - This allows the DBMS software to work with different database applications.
- **Insulation between programs and data:**
  - Called **program-data independence**.
  - Allows changing data structures and storage organization without having to change the DBMS access programs.

RELATIONS

Relation_name	No_of_columns
EMPLOYEE	5
DEPARTMENT	3
PROJECT	3
DEPENDENT	4

(a)

An example of a database catalog for the database in figure (a)

COLUMNS

Column_name	Data_type	Belongs_to_relaiton
SSN	Character(10)	EMPLOYEE
Name	Character(10)	EMPLOYEE
Age	Integer	EMPLOYEE
Address	Character(30)	EMPLOYEE
Salary	Decimal(10,2)	EMPLOYEE
Dname	Character(10)	DEPARTMENT
Dnumber	Character(10)	DEPARTMENT
Dlocation	Character(30)	DEPARTMENT
Pname	Character(10)	PROJECT
…..	……..	…..
……	……..	…….

(b)

Example: To add a new field Birthdate to Employee records, we only need to change the description of Employee records in the catalog; no change in the programs.

Data Item Name	Starting Position in Record	Length in characters (bytes)
SSN	1	10
Name	11	10
Age	21	1
Address	22	30
Salary	52	10,2

internal storage for EMPLOYEE record, based on the database catalog in Figure (a)

PROGRAM-OPERATION INDEPENDENCE

- In object-oriented and object-relational systems, users can define operations on data as part of the database definitions.
-
- An **operation (also called a function or method)** is specified in two parts.
  - The **interface (or signature)** of an operation includes the operation name and the data types of its arguments (or parameters).
  - The **implementation (or method)** of the operation is specified separately and can be changed without affecting the interface.
- User application programs can operate on the data by invoking these operations through their names and arguments, regardless of how the operations are implemented. This may be termed **program-operation independence**

DATA ABSTRACTION

- The characteristic that allows program-data independence and program-operation independence is called data abstraction.
- A data model is used to hide storage and implementation details and also present the users with a conceptual view of the database.

SUPPORT OF MULTIPLE VIEWS OF THE DATA

- Each user may see a different view of the database, which describes only the data of interest to that user.

Example: One user is interested in accessing and printing the specified employee field details. A second user, who is interested in checking salary details of employee who work in department 5.

(a) EMPLOYEE

SSN	Name
123467	Aruna
123469	Sita
123468	Sai
123567	Babu

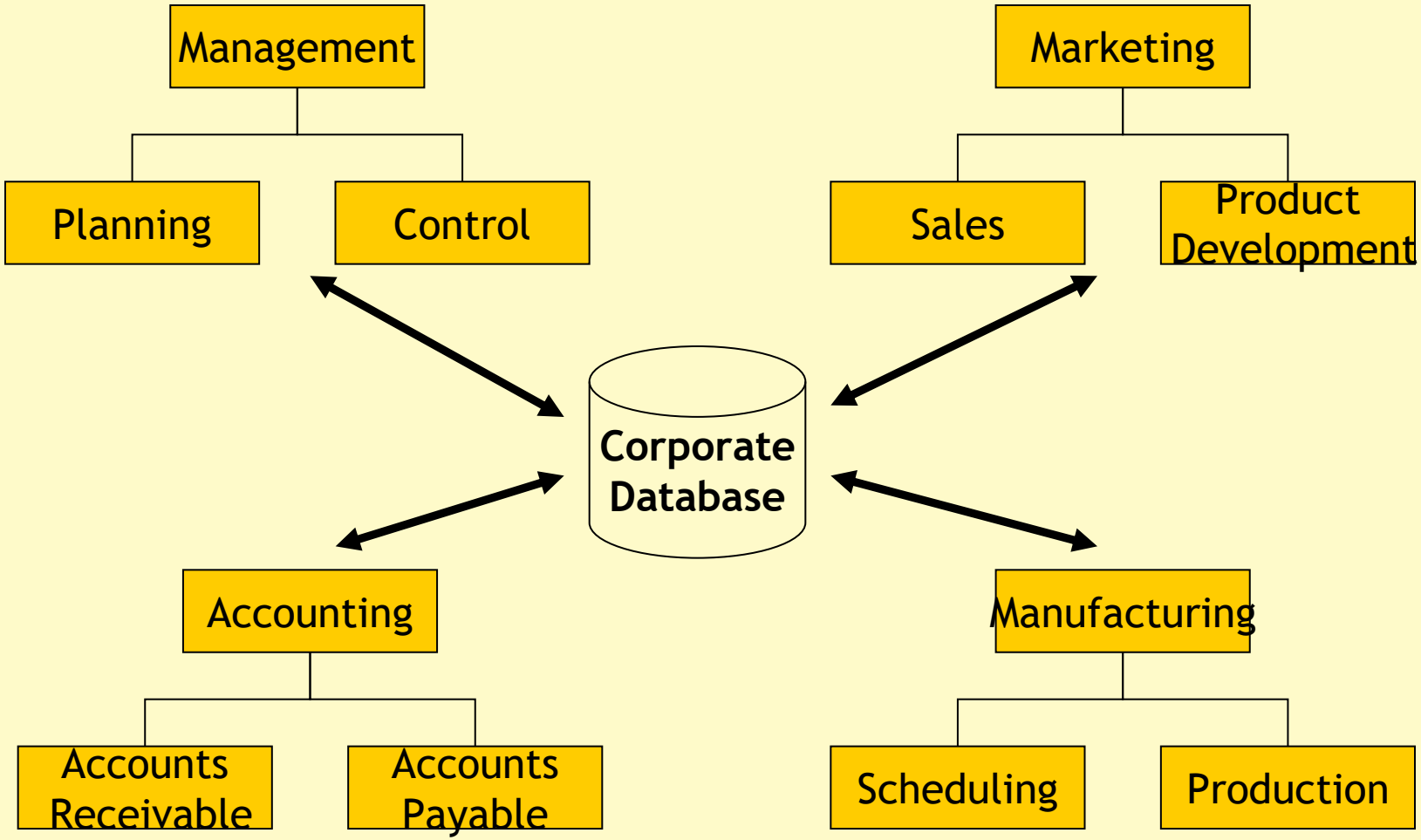
(b)EMPLOYEE

SSN	Name	Dno
123467	Aruna	5
123567	Babu	5

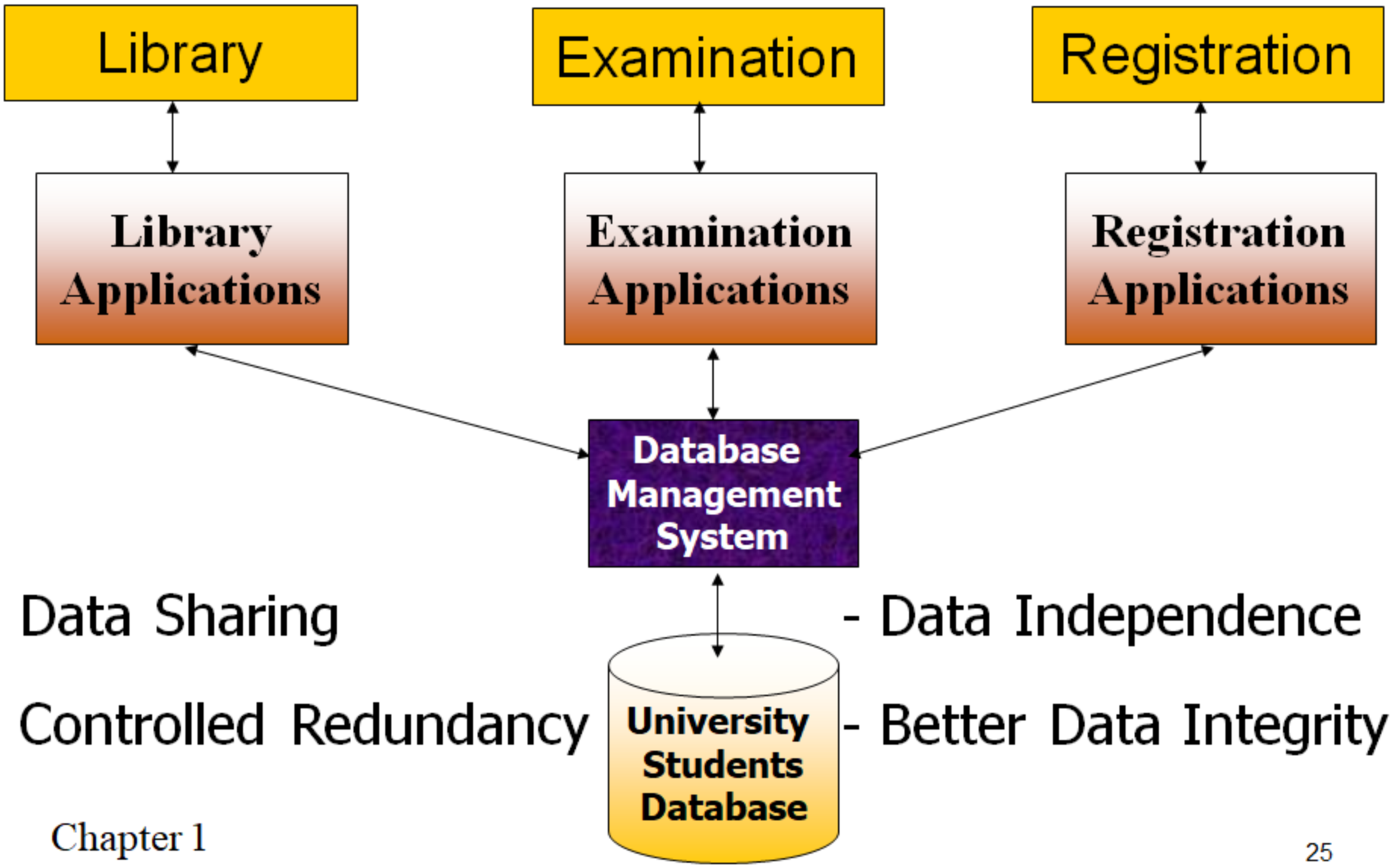
SHARING OF DATA AND MULTIUSER TRANSACTION PROCESSING:

- It allows multiple users to retrieve and to update the database at the same time.
- Concurrency control within the DBMS guarantees that each transaction is **correctly executed or completely aborted**.
- OLTP (Online Transaction Processing) is a major part of database applications This allows hundreds of concurrent transactions to execute per second.
- **Example: Any online booking DB system**
- when multiuser accesses the information about the seat details of the flight and reserve a seat on an airline reservation database system ensure that each seat can be accessed by only one user at a time. The DBMS software ensures that concurrent transactions operate correctly and efficiently.

THE CONCEPT OF A SHARED ORGANIZATIONAL DATABASE



Advantages of Database Approach



- Data Sharing
- Controlled Redundancy
- Data Independence
- Better Data Integrity

ADVANTAGES OF USING THE DATABASE APPROACH

- Controlling redundancy in data storage and in development and maintenance efforts.
- Sharing of data among multiple users.
- Restricting unauthorized access to data.
- Providing persistent storage for program Objects

In Object-oriented DBMSs

Example: ”Programming languages like Pascal, java or C++ the values of program variables are discarded/lost once a program terminates, till programmer explicitly stores them in permanent files.

- Providing Storage Structures (e.g. indexes) for efficient Query Processing

CONTINUED..

- Providing backup and recovery services.
- Providing multiple interfaces to different classes of users.
- Representing complex relationships among data.
- Enforcing integrity constraints on the database.
- Drawing inferences and actions from the stored data using deductive and active rules

ADDITIONAL IMPLICATIONS OF USING THE DATABASE APPROACH

- Potential for enforcing standards:
  - This is very crucial for the success of database applications in large organizations.
  - The database permits the DBA to define and enforce standards among database users.
  - **Standards** refer to data item names, display formats, screens, report structures, meta-data (description of data), Web page layouts, etc.
- Reduced application development time:
  - Once a database is up and running, substantially less time is generally required to create new applications using DBMS facilities.
  - Incremental time to add each new application is reduced.

CONTINUED..

- Flexibility to change data structures:
  - Database structure can be changed as per the requirements change without affecting the stored data and the existing application programs.
- Availability of current/ Up-to-Date Information:
  - Extremely important for on-line transaction systems such as airline, hotel, car reservations.
  - As soon as one user's update is applied to the database, all other users can immediately see this update.
- Economies of scale:
  - Makes the organization profitable by combining many application & data.
  - Wasteful overlap of resources and personnel can be avoided by consolidating data and applications across departments

WHEN NOT TO USE A DBMS

- Main inhibitors (costs) of using a DBMS:
  - High initial investment and possible need for additional hardware.
  - Overhead for providing generality, security, concurrency control, recovery, and integrity functions.
- When a DBMS may be unnecessary:
  - If the database and applications are simple, well defined, and not expected to change.
  - If there are stringent real-time requirements that may not be met because of DBMS overhead.
  - If access to data by multiple users is not required.

CONTINUED..

- When no DBMS may suffice:
  - If the database system is not able to handle the complexity of data because of modeling limitations
  - If the database users need special operations not supported by the DBMS.

LIMITATIONS OF DBMS

- Complexity
- Size
- Cost
  - Software
  - Hardware
  - Conversion
- Performance
- Vulnerability

DEFINE MODEL, DATA MODEL AND SCHEME

Model

- A model is described by the schema which is held in the data dictionary.
- A structure that demonstrates all the required features of the parts of the real world which is of interest to the users of the information in the model.
- Representation and reflection of the real world (Universe of Discourse)

Data Model

- A *data model* is a collection of concepts for describing data.
- It describe structure of a database: the data types, relationships, constraints, semantics and operational behavior.
  - It is a tool for data abstraction

Database Schema:

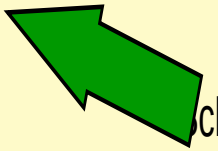
The description of a database is called database schema, which is specified during database design and is not expected to change frequently.

UNIVERSITY DB

A *schema* is a description of a particular collection of data, using a given data model.

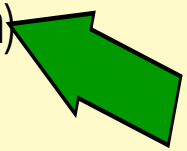
- **Schema:**
  - **Students(sid text,**
    name text,
    login text,
    age integer,
    gpa float)
  - **Courses(cid text,**
    cname text,
    credits integer)
  - **Enrolled(sid text,**
    cid text,
    grade text)

Student(studno,name,address)
Course(courseno,lecturer)



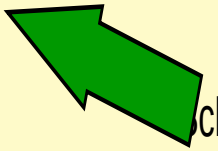
chema

Student(123,Bloggs,Woolton)
Course(321,Jones,Owens)



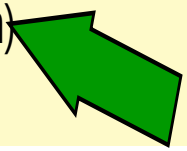
Instance

Student(studno,name,address)
Course(courseno,lecturer)



chema

Student(123,Bloggs,Woolton)
Course(321,Jones,Owens)



Instance

DEFINE THE TERM

- **Database state**: The data in the database at a particular moment in time is called a **database.**
- **state or snapshot**. It is also called the **current set of occurrences or instances** in the database.
- **Empty state**: The database with no data is called empty state.
- **Initial state**: when the database is first populated or loaded with the initial data.
- **Current state**: At any point in time, the database has a current state.
- **Valid state**: A state that satisfies the structure and constraints specified in the schema.
- **Invalid state**: If a state that do not satisfies the structure and constraints specified in the schema.
- The schema is sometimes called the **intension**, and a database state is called an **extension** of the schema.

SCHEME DIAGRAM

Schema Diagram: A displayed schema is called a Schema diagram

Example:

FLIGHT-SCHEDULE

FLIGHT#	AIRLINE	WEEKDAY	PRICE

DEPT-AIRPORT

FLIGHT#	AIRPORT-CODE

Figure1.16: Schema diagram

FLIGHT -SCHEDULE

FLIGHT#	AIRLINE	WEEKDAY	PRICE
101	delta	mo	156
545	american	we	110
912	scandinavian	fr	450
101	delta	thru	200

DEPT -AIRPORT

FLIGHT#	AIRPORT -CODE
101	atl
912	cph
545	lax

EXAMPLE

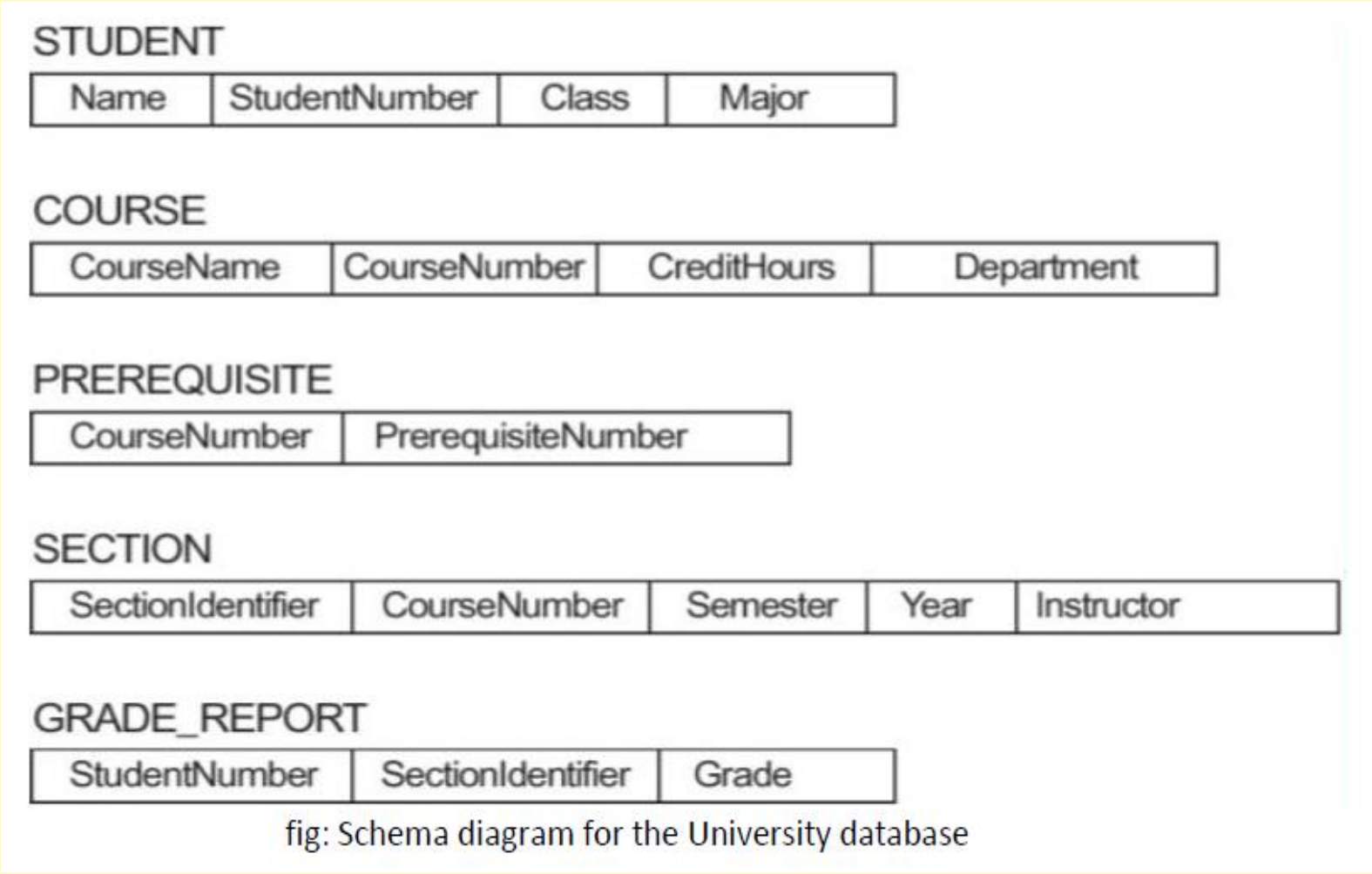


fig: Schema diagram for the University database

PROGRAM 1: INSURANCE DATABASE

Consider the *Insurance database* given below.

The primary keys are underlined and the data types are specified.

PERSON (driver-id #: String, name: String, address: String)

CAR (Regno: String, model: String, year: int)

ACCIDENT (report-number: int, date: date, location: String)

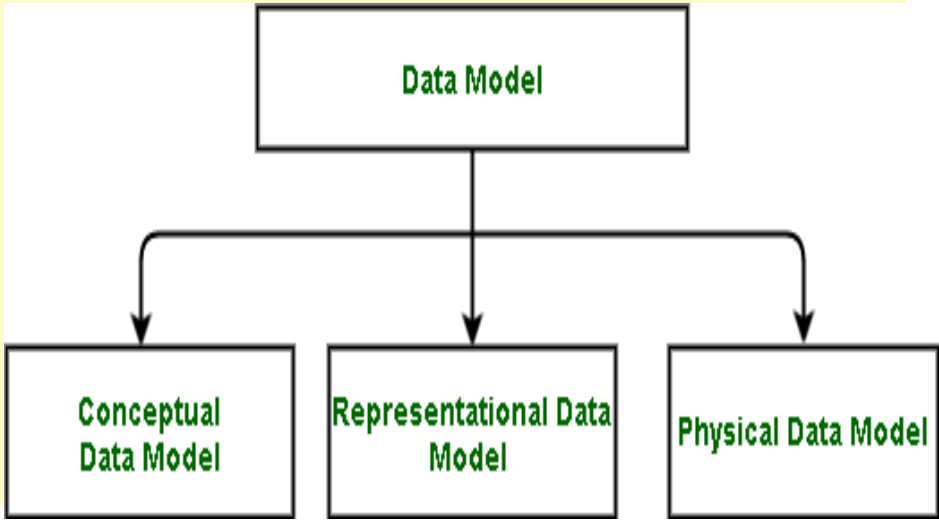
OWNS (driver-id #: String, Regno: String)

PARTICIPATED (driver-id: String, Regno: String, report-number: int, damage-amount: int)

DATA MODELS IN DBMS

- A Data Model in Database Management System (DBMS) is the concept of tools that are developed to summarize the description of the database. Data Models provide us with a transparent picture of data which helps us in creating an actual database. It shows us from the design of the data to its proper implementation of data.
- **Types of Relational Models**

1. Conceptual Data Model
2. Representational Data Model
3. Physical Data Model



CATEGORIES OF DATA MODELS

- **High-level or Conceptual data models**:
- It used to describes the structure of the database (based on entities, attributes and relationships).
- **Entity**: It represents a real world object such as an employee or a project.
- **Attributes**: It represents some property of interest that describes an entity, such as the employee's name or salary.
- **Relationship**: A relationship among the two or more entities represents an association among two or more entities.
- Example: a work-on relationship between an employee and a project

CONTINUED..

- **A low-level or physical data model**: It used to describes the details of how data is stored in the computer. These concepts generally meant for computer specialists, not for typical end users.

- **Representational or implementation data models:** It is between the **high level or low level data** models, which provides concepts that is understood by end user and also the way data is organized by within the computer.
- It hides some details of data storage but can be implemented on a computer system directly (record-based, object oriented).

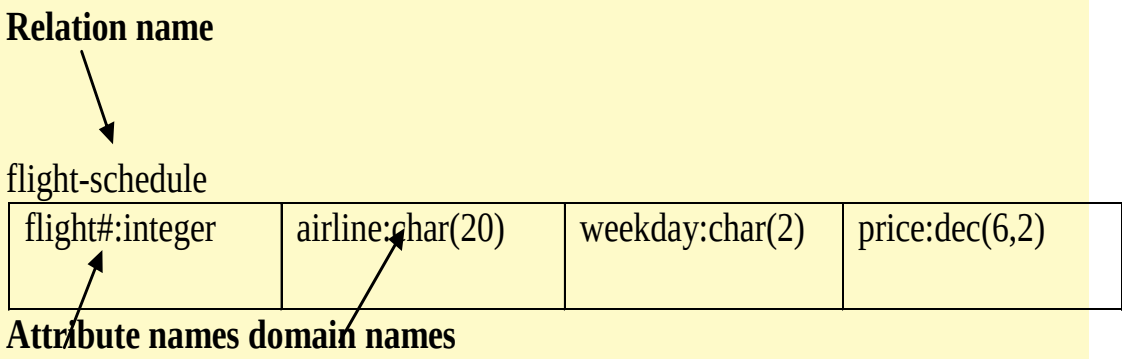
SOME OTHER DATA MODELS

- Relational Model
- Network Model
- Hierarchical Model
- Object Oriented Model
- Semantic Data Model

RELATIONAL MODEL

- Relational model was originally introduced in 1970, was heavily researched and experimented within IBM Research and several universities.
- Relational DBMS Products emerged in the early 1980s.

- It uses a collection of tables to represent both data and the relationship among those data. Each table has multiple columns and each column has unique name.
- Commercial systems include: ORACLE, DB2, SYBASE, INFORMIX, INGRES, SQL Server
- Relational Model - Data Structures
- domains
- attributes
- relations



CONTINUED..

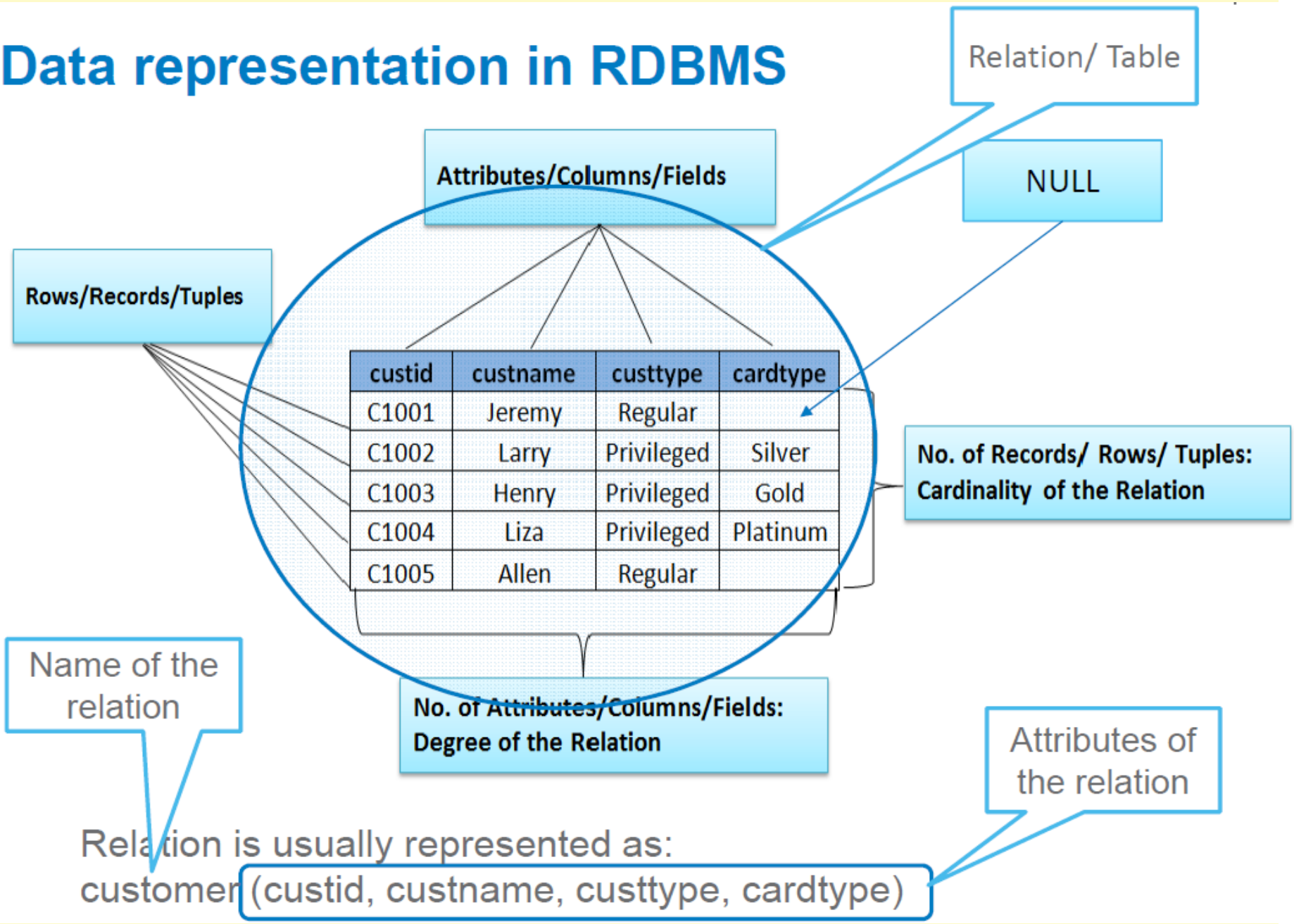
ADVANTAGES:

- Data represent in a Simplified format.
- Manipulating record is simple with the use of key attributes.
- Representation of different types of relationship is possible with this model.

Operations

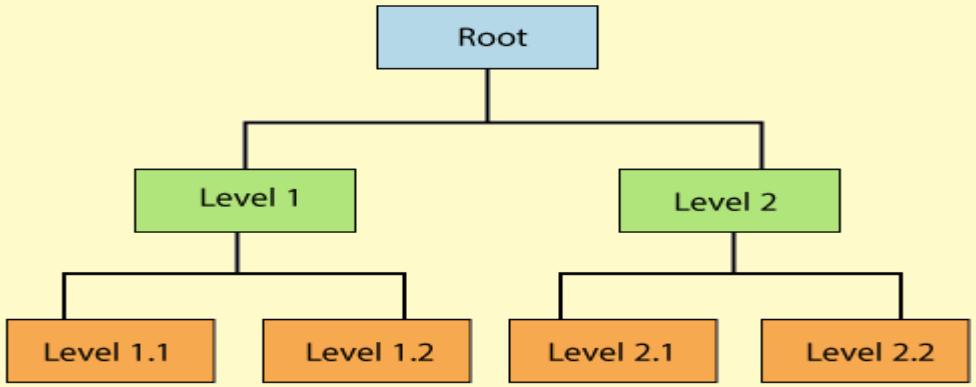
- Powerful set-oriented query languages
- Relational Algebra: procedural; describes how to compute a query; operators like JOIN, SELECT, PROJECT
- Relational Calculus: declarative; describes the desired result, e.g. SQL, QBE
- Insert, delete, and update capabilities

Data representation in RDBMS

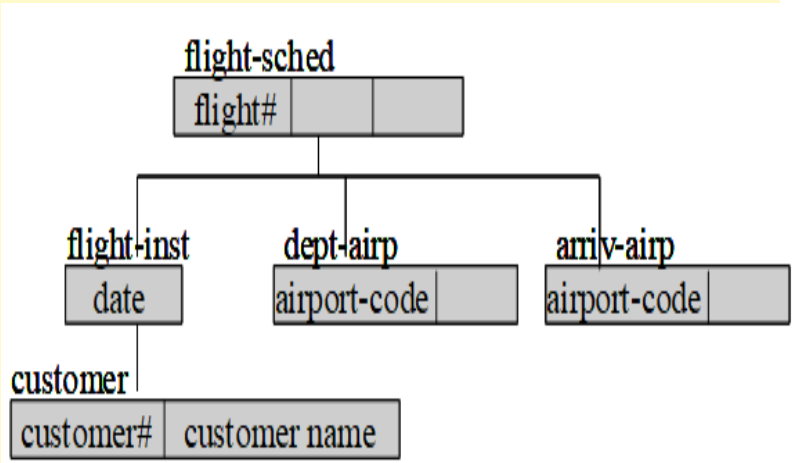


HIERARCHICAL MODEL

- It is the type of database that stores data in the form of parent-children relationship nodes. Here, it organizes data in a tree-like structure. It used to create a relationship b/n two units(1:N).
- Commercial systems include IBM's IMS, MRI's System-2000 (now sold by SAS), and CDC's MARS IV.

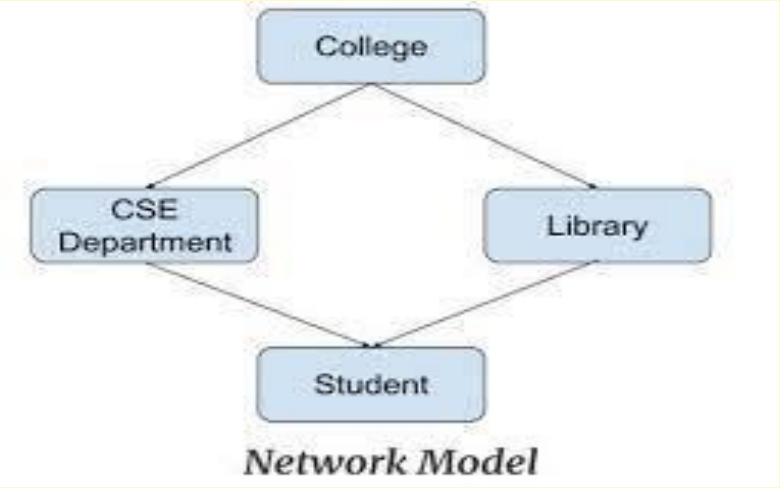


Hierarchical Database

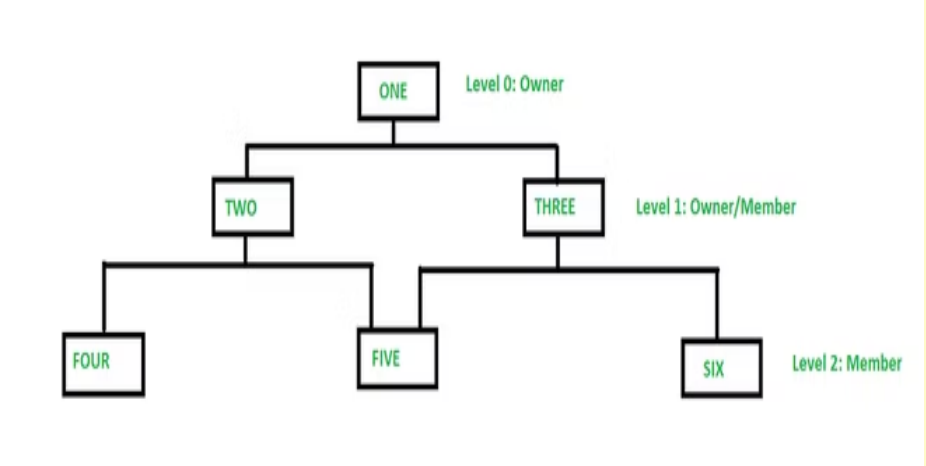


NETWORK MODEL

- The collections of records, and relationships, among data are represented by links. Which can viewed as pointers.
- Or
- The network model allows any unit to be related to other unit through fixed relationship(M:N).
- Based on the CODASYL-DBTG 1971 report. Commercial systems include, CA-IDMS and DMS-1100



Network Model

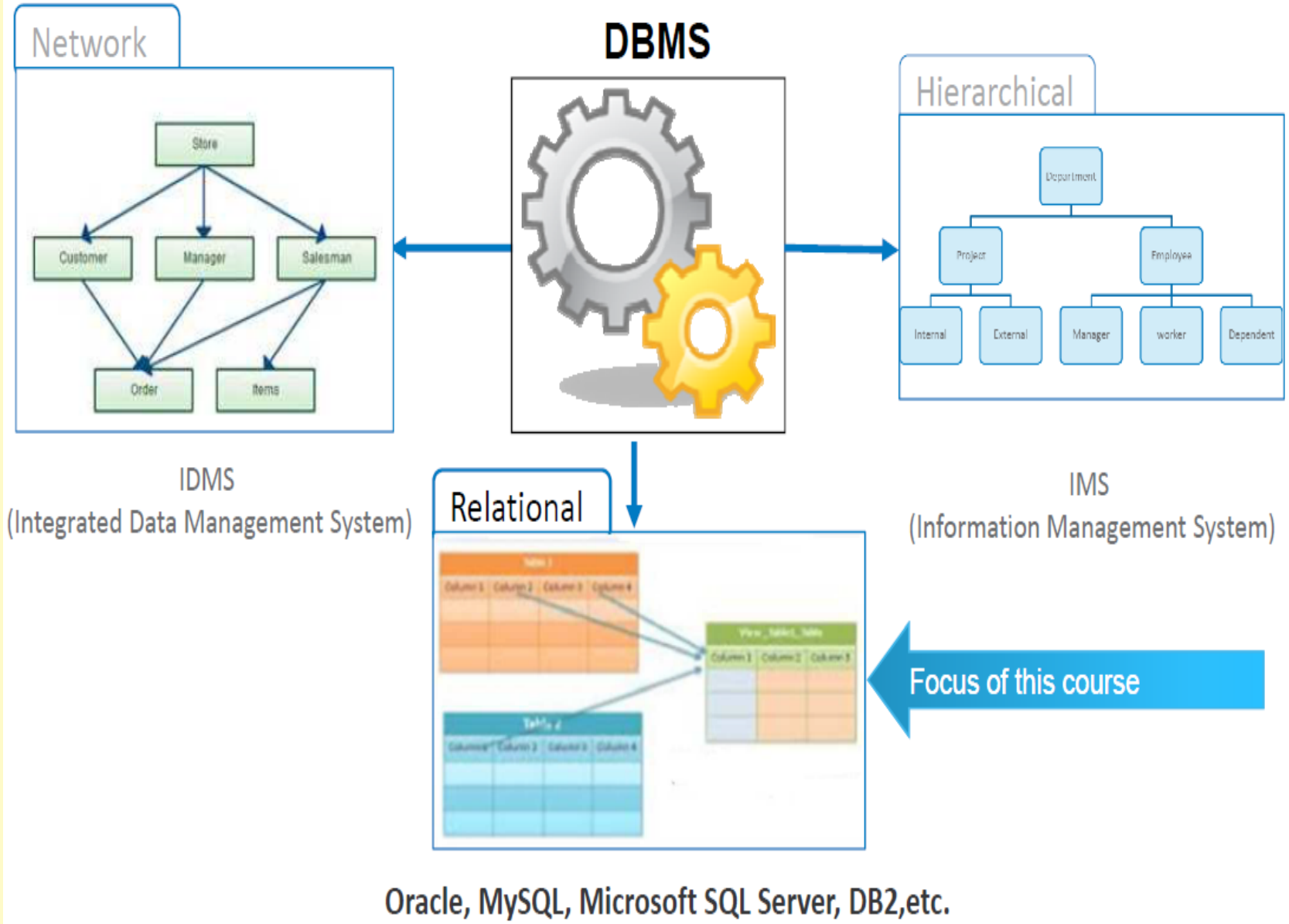


OBJECT-ORIENTED DATA MODEL

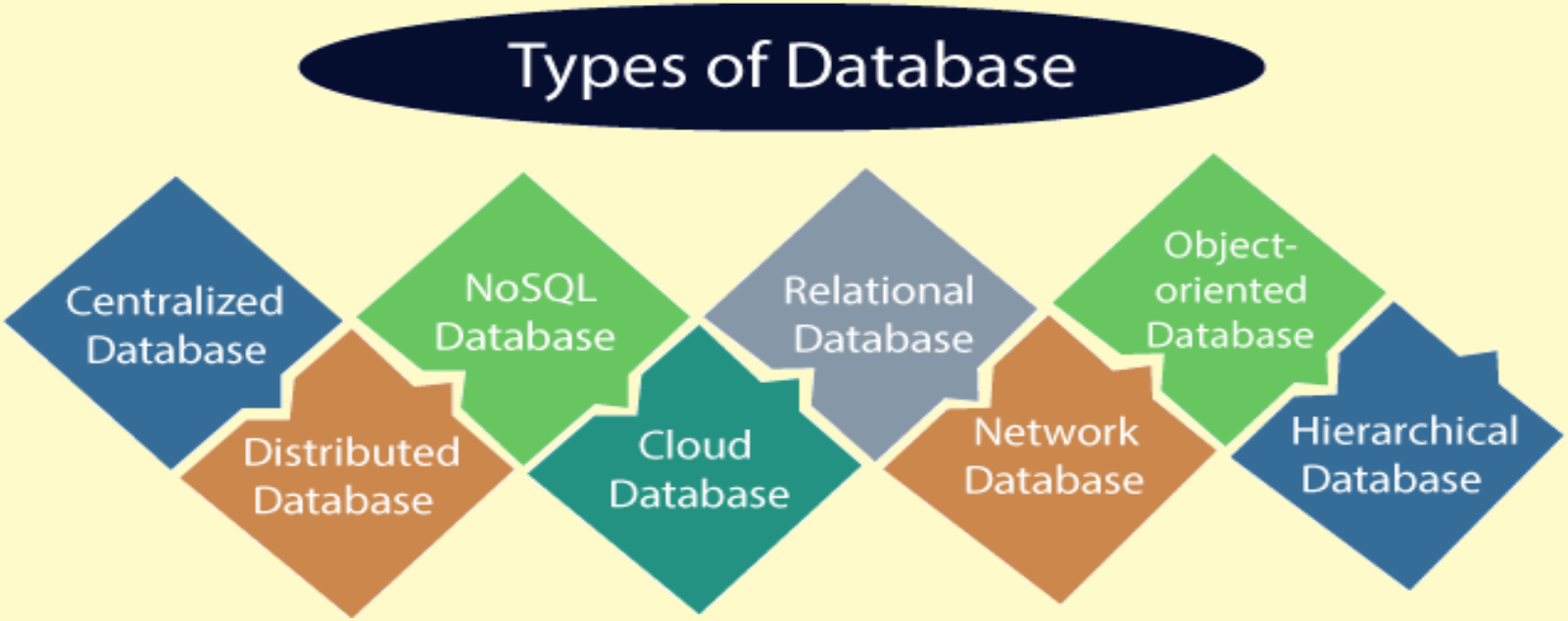
- The data and their relationships are contained in a single structure.
- *In this, real-wor*ld problems are represented as objects with different attributes.
- All objects have multiple relationships between them. It is a combination of Object Oriented programming and a Relational Database Model.
- e.g., Simula, Smalltalk, C++, Java

Example:

```
class flight-schedule {
type tuple (flight#: integer,
  weekdays: set ( weekday: enumeration {mo,tu,we,th,fr,sa,su})
   dept-airport: airport, arriv-airport: airport)
      method reschedule(new-dept: airport, new-arriv: airport)}
```


Network
Store
Customer
Manager
Salesman
Order
Items
DBMS
Hierarchical
Department
Project
Employee
Internal
External
Manager
worker
Dependent
IDMS
(Integrated Data Management System)
Relational
IMS
(Information Management System)
Focus of this course
Oracle, MySQL, Microsoft SQL Server, DB2,etc.

- Data on the Web and E-commerce Applications
  - Scientific Applications
  - XML (eXtensible Markup Language)
  - Image Storage and Management
  - Audio and Video Data Management
  - Data Warehousing and Data Mining
  - Spatial Data Management
  - Time Series and Historical Data Management



https://www.javatpoint.com/types-of-databases

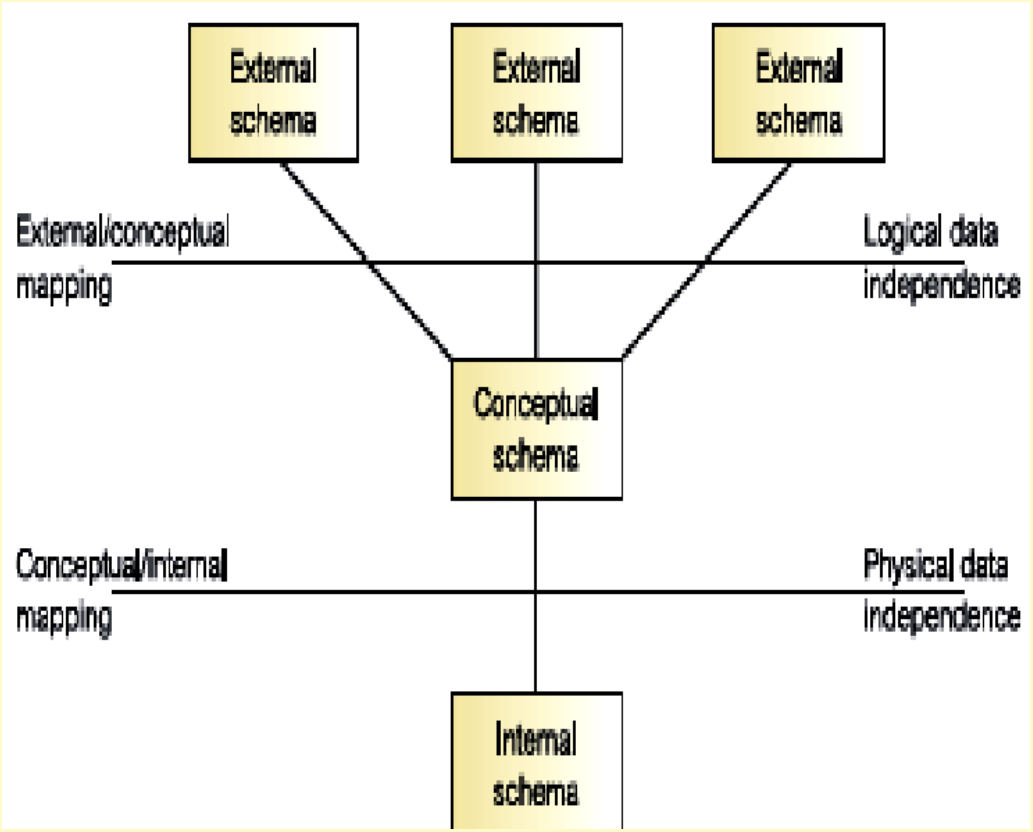
Three schema Architecture:

•It is also called **ANSI/SPARC** architecture or three-level architecture.

•The framework is used to describe the structure of a specific database system.

•It used to separate the user applications and physical database.

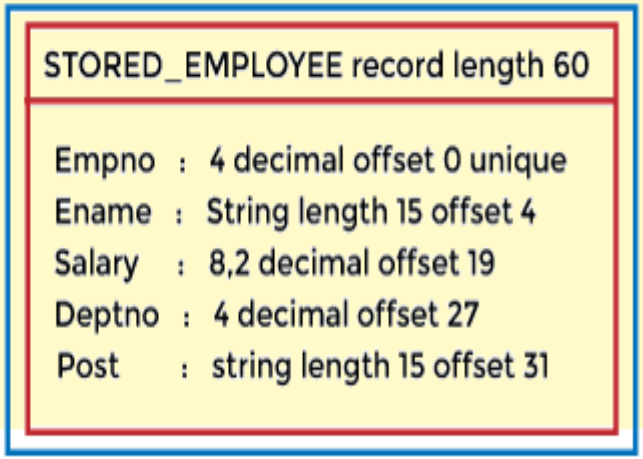
•The three schema architecture contains three-levels. It breaks the database down into three different categories.



INTERNAL LEVEL HAS AND INTERNAL SCHEMA

- It describes the physical storage structure of the database.
- Uses a physical data model and describes the complete details of data storage and access path for the database.
- The internal schema is also known as a physical schema.
- Storage space allocations.
  For Example: B-Trees, Hashing etc.
- Access paths.
  For Example: Specification of primary and secondary keys, indexes, pointers and sequencing.
- Data compression and encryption techniques.
- Optimization of internal structures.
- Representation of stored fields.

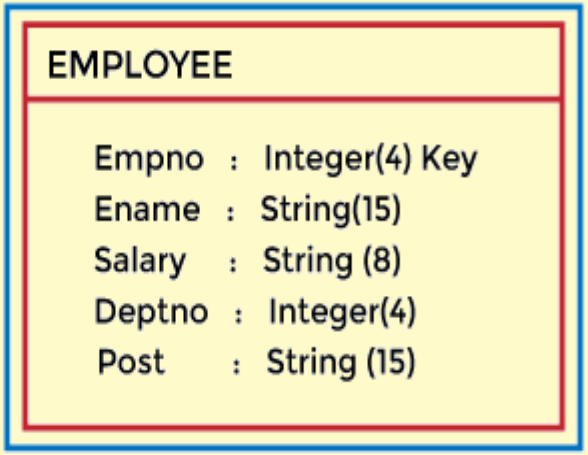
Internal view



CONCEPTUAL LEVEL HAS A CONCEPTUAL SCHEMA

- It describes the structure of the whole database for a community of users.
- Conceptual level is also known as logical level.
- Hides the details of physical storage structures and concentrates on describing entities, data types, relationships, user operations, and constraints.
- Programmers and database administrators work at this level.

Global view



EXTERNAL OR VIEW LEVEL

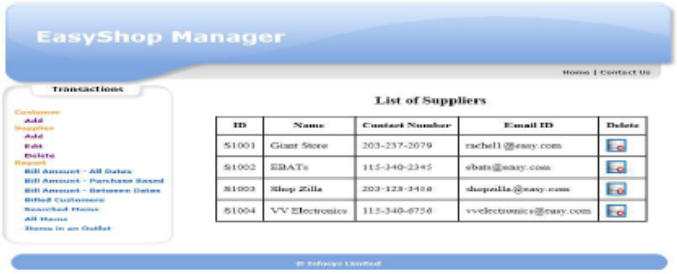
- An external schema/level is also known as view schema.
- It includes a number of external schemas or user views.
- At the external level, a database contains several schemas that sometimes called as subschema.
- The subschema is used to describe the different view of the database.
- It describes the part of the database that a particular user group is interested in and hides the rest of the database from that user group.
- .

MAPPING BETWEEN VIEWS

- The three levels of DBMS architecture don't exist independently of each other.
- Mapping is used to transform the request and response between various database levels of architecture.
- DBMS is responsible for correspondence between the three types of schema. This correspondence is called **Mapping**.

EXAMPLE:

External level
(End-user perception of data)



Conceptual level
(Database designer perception of data)

supplierid	character
suppliername	character
emailid	character

Internal level
(DBMS/OS perception of data)

supplierid	TYPE = BYTE (6)	OFFSET = 0
suppliername	TYPE = BYTE (30)	OFFSET = 6
emailid	TYPE = BYTE (30)	OFFSET = 36

Demo: EasyShop application **external view**: supplier report; **conceptual view:** describe supplier table

MAPPING BETWEEN VIEWS

There are basically two types of mapping in the database architecture:

- Conceptual/ Internal Mapping
- External / Conceptual Mapping

- In External / Conceptual mapping, it is necessary to transform the request from external level to conceptual schema.
- In Conceptual / Internal mapping, DBMS transform the request from the conceptual to internal level.

MAPPING BETWEEN VIEWS

- The three levels of DBMS architecture don't exist independently of each other.
- Mapping is used to transform the request and response between various database levels of architecture.
- DBMS is responsible for correspondence between the three types of schema. This correspondence is called Mapping.
- In External / Conceptual mapping, it is necessary to transform the request from external level to conceptual schema.
- In Conceptual / Internal mapping, DBMS transform the request from the conceptual to internal level.

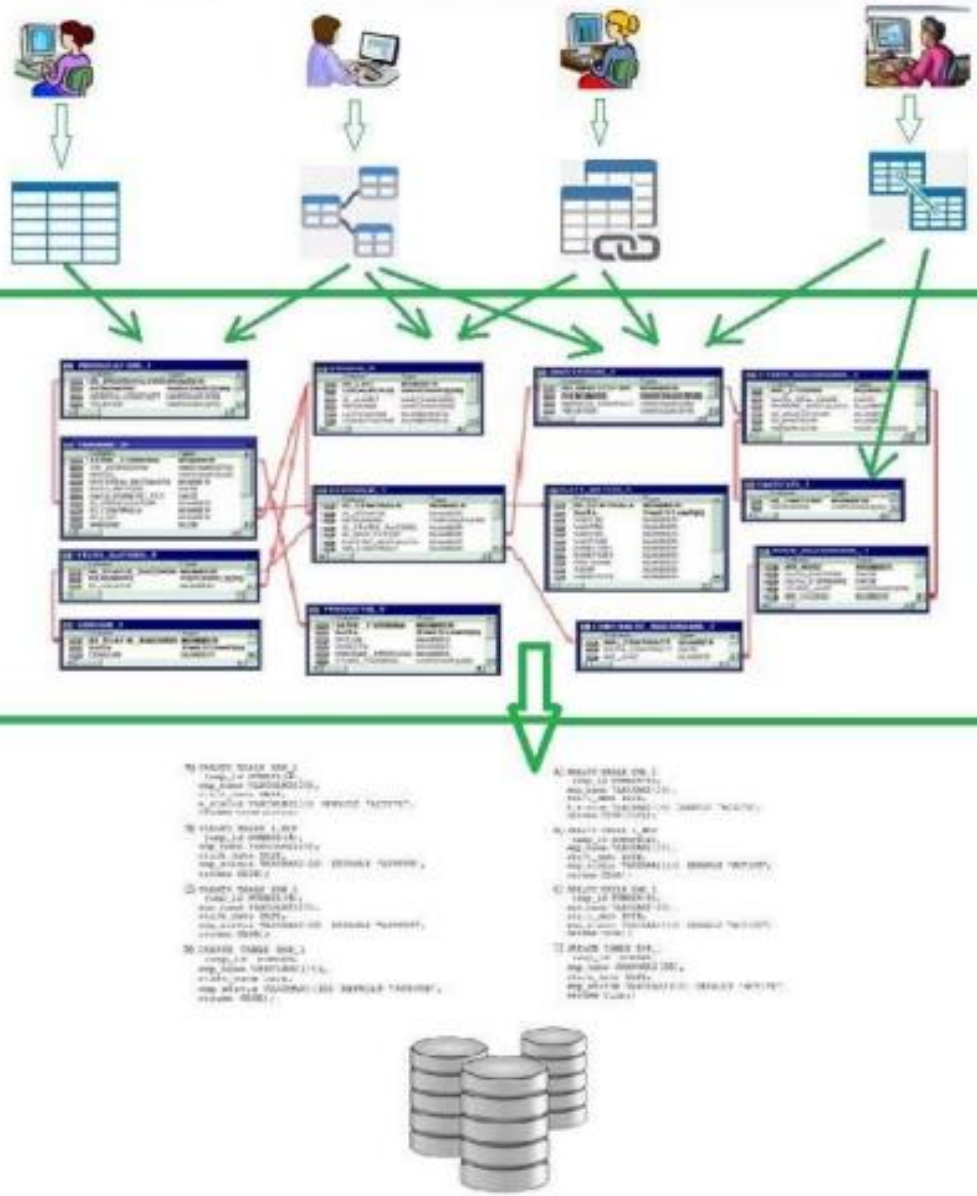
DATA INDEPENDENCE

- The capacity to change the schema at one level of a DB system without change the schema at the next higher level is called **data independence**.
- Two types of data independence:

1. **Logical data independence :** Change the conceptual schema without having to change the external schemas

2. **Physical data independence**: Change the internal schema without having to change the conceptual schema

ANSI-SPARC ARCHITECTURE - A DIFFERENT VIEW



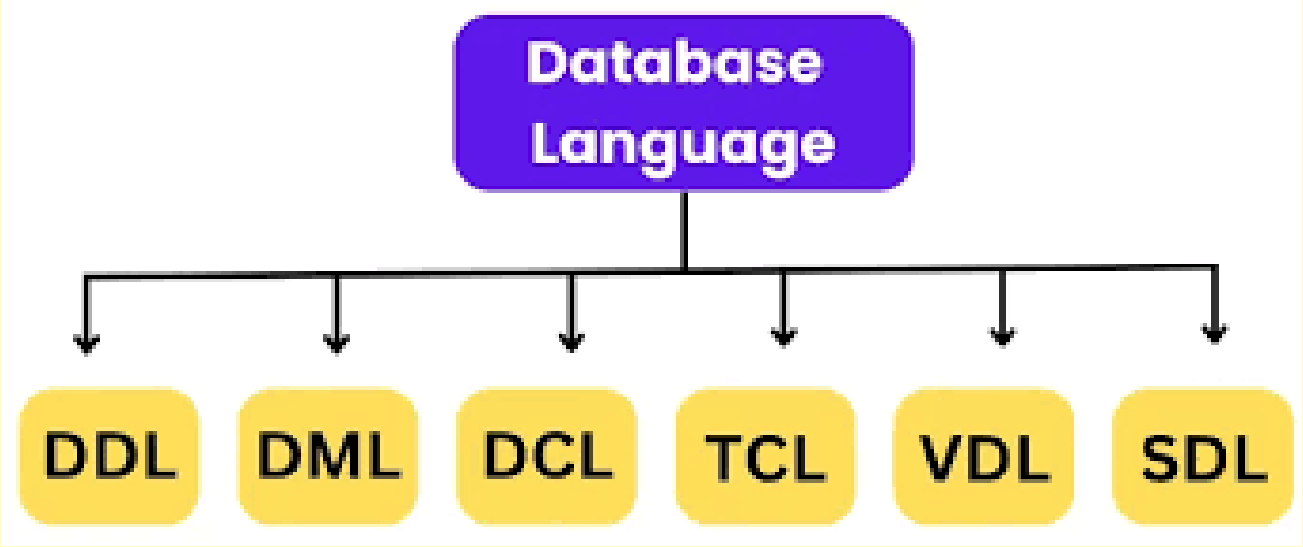
EXTERNAL MODEL :-
User Specific Data

CONCEPTUAL MODEL:-
Complete Data & Relationships

INTERNAL MODEL : -
Implementation of Tables & Relationships in DBMS Specific language.

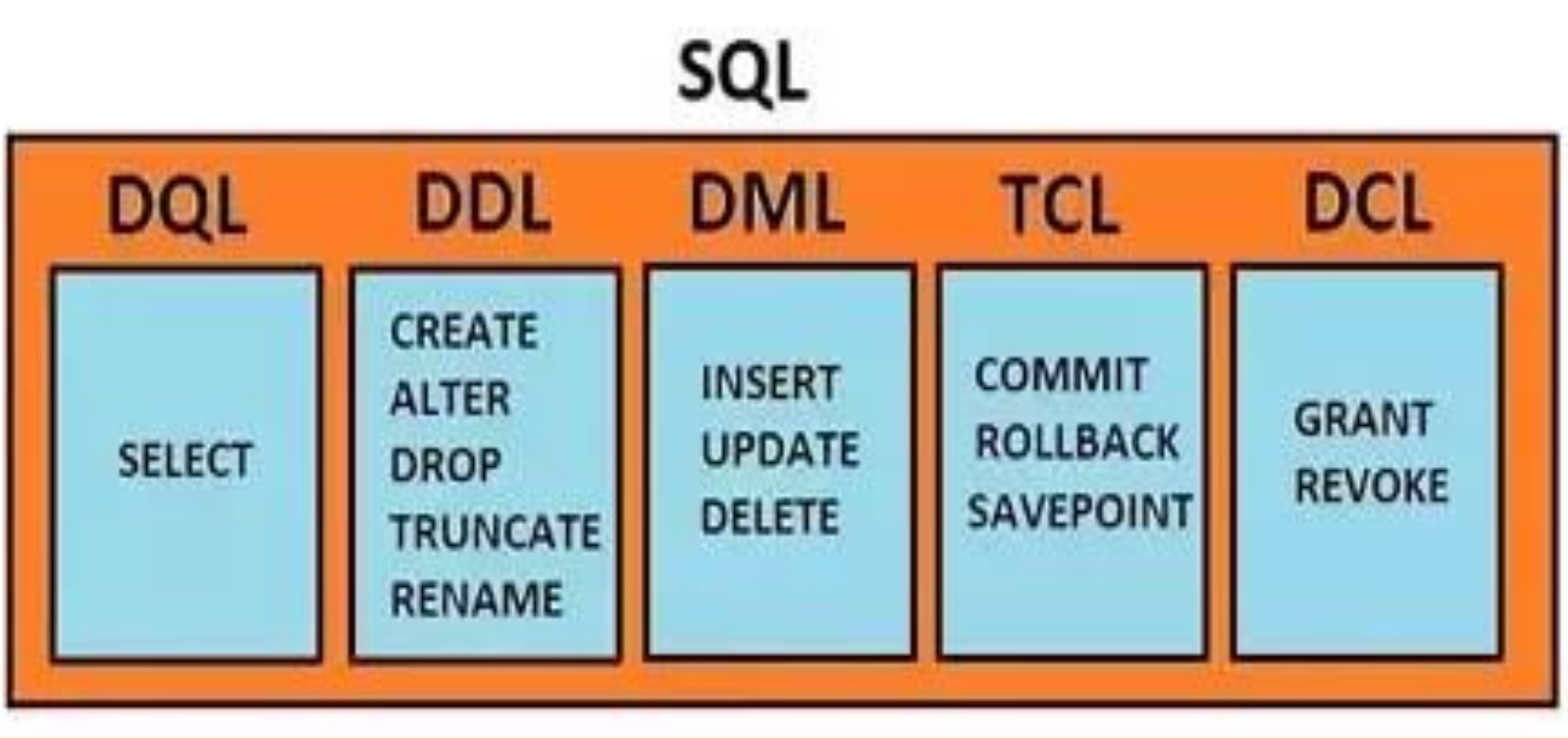
DATABASE LANGUAGES AND INTERFACES

- A DBMS has appropriate languages and interfaces to express database queries and updates.
- Database languages can be used to read, store and update the data in the database.



DBMS LANGUAGES

- **Structured Query Language(SQL)**
- DBMS packages provide an integrated feature of above languages into a single language called **Structured Query Language.**
- **Data definition language (DDL),** is used by the DBA and by database designers to define both schemas.
- **Storage definition language (SDL),** is used to specify the internal schema.
- **View definition language (VDL),** to specify user views and their mappings to the conceptual schema.
- **Data manipulation language(DML)** provides set of operations like retrieval, insertion, deletion, and modification of the data.


SQL
DQL
DDL
DML
TCL
DCL
SELECT
CREATE
ALTER
DROP
TRUNCATE
RENAME
INSERT
UPDATE
DELETE
COMMIT
ROLLBACK
SAVEPOINT
GRANT
REVOKE


Database Languages
DDL
DML
Procedural
Non-procedural

DML

- There are two main types of DMLs. They are
- **High-level or nonprocedural DML**
- **Low-level or procedural DML**

- **High-level DML:** Many DBMSs allow high-level DML statements either to be entered interactively from a display monitor or terminal or to be embedded in a general-purpose programming language.
- Retrieve many records in a single DML statement; therefore, they are called set-at-a-time or set-oriented DML's.
- A query in a high-level DML often specifies which data to retrieve rather than how to retrieve it; therefore, such languages are also called declarative.

DML

- **Low-level or procedural DML:** They must be embedded in a general-purpose programming language.
- The low-level are also called **record-at-a-time** because it retrieve only one record or objects at a time from the DB and processes each separately.
- It use programming language such as lopping, to retrieve and process each record from a set of records.
- **Ex:** Hierarchical model is DL/1 uses command like GET UNIQUE, GET NEXT, GET NEXT WITHIN PARENT etc
- Whenever DML commands, whether high level or low level, are *embedded in general-purpose programming language*, that language is called the **host language** and the DML is called **data sublanguage**. High-level DML used in a standalone interactive manner is called a **query language.**

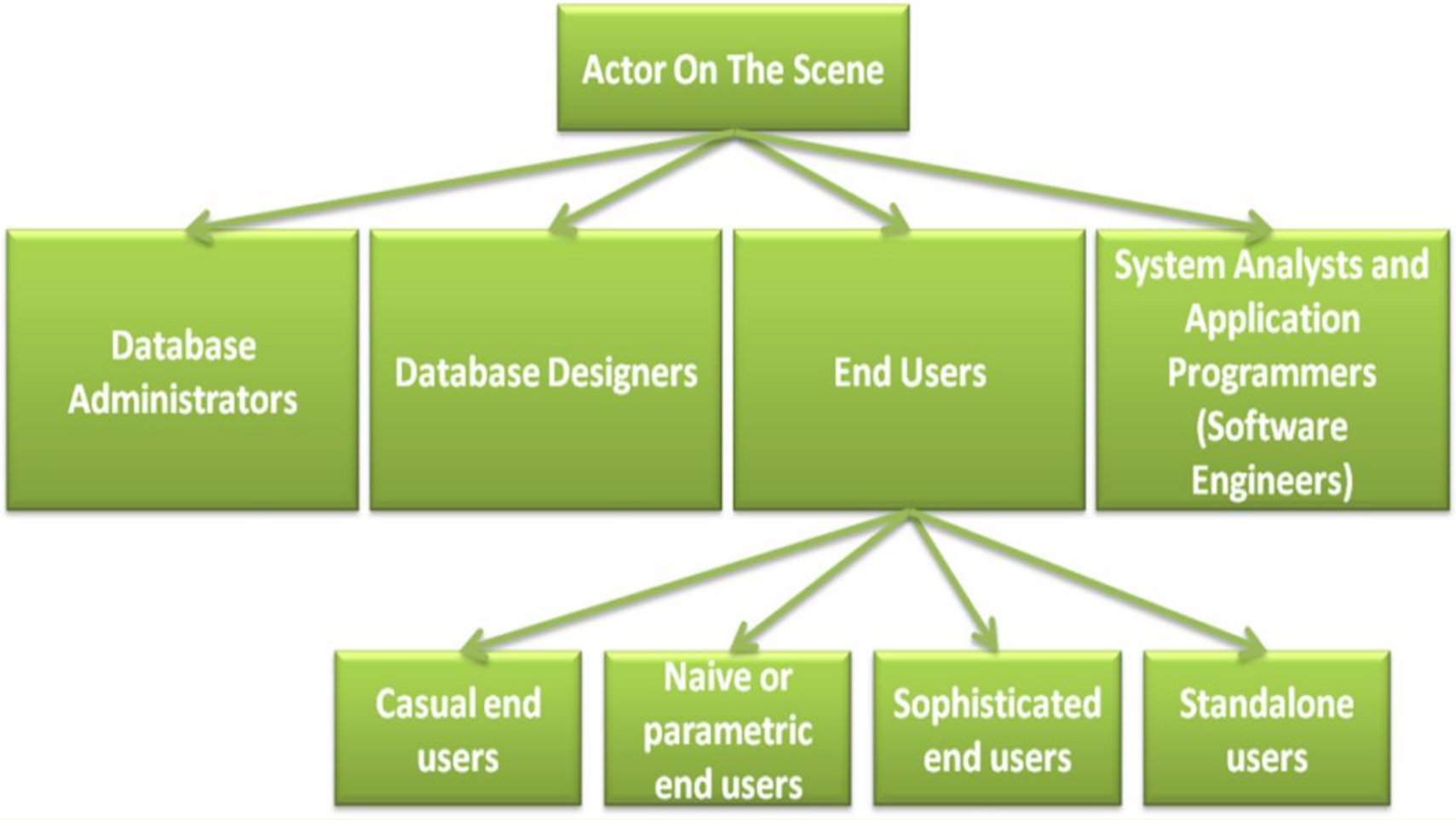
INTERFACES IN DBMS

1. **Menu-Based Interfaces** for Web Clients or Browsing.

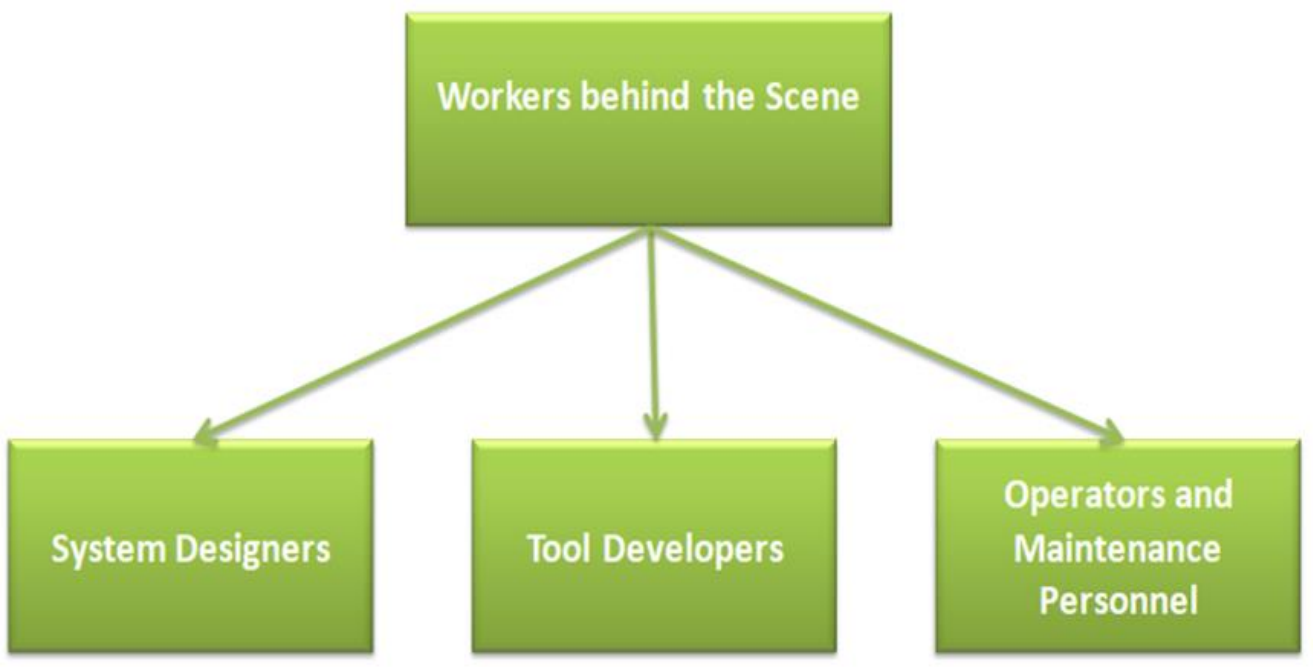
2. **Forms-Based Interfaces** displays a form to each user

3. Graphical User Interfaces displays a schema to the user in diagrammatic form
4. **Natural Language Interfaces** accepts requests written in English or some other language and attempt to understand them.
5. Speech Input and Output Interfaces
6. **Interfaces for Parametric Users** such as bank tellers, often have a small set of operations that they must perform repeatedly.
7. **Interfaces for the Database Administrator (DBA)** use privileged commands

DATABASE USERS



DB USERS


Workers behind the Scene
System Designers
Tool Developers
Operators and Maintenance Personnel

DB USERS

Users are differentiated by the way they expect to interact with the system

- **System Analysts** – Determine the requirements of end users
- **Database Designers** – Identify and Choose appropriate structures to represent and store data
- **Application Programmers** – interact with system through DML calls

END USERS

- **Casual Users**
  - uses a sophisticated database query language to specify their requests and are typically middle or high level managers
- **Sophisticated Users**
  - form requests in a database query language. Scientists/Engineers
- **Specialized Users**
  - write specialized database applications that do not fit into the traditional data processing framework
- **Naïve or Parametric Users**
  - invoke one of the permanent application programs that have been written previously. Example: people accessing database over the web, bank tellers, clerical staff
- **Standalone Users**
  - who maintain personal databases by using ready made program packages that provide easy to use menu or GUI based interfaces

DATABASE ADMINISTRATOR (DBA)

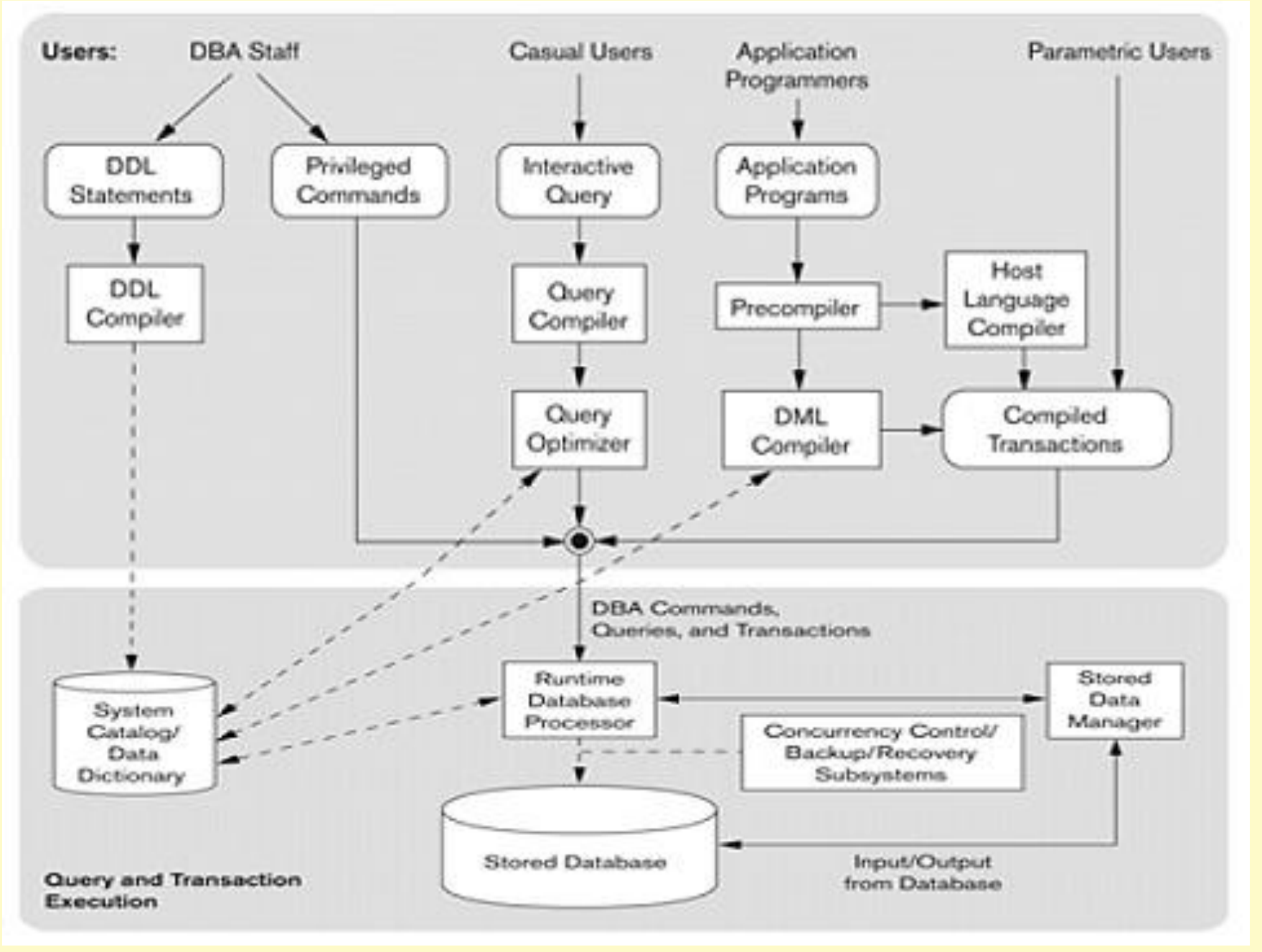
- Coordinates all the activities of the database system
  - has good understanding of the enterprise's information resources and needs.
- Responsible for Managing the resources
  - Database (primary)
  - DBMS (secondary)
- DBA's duties include:
  - Storage structure and access method definition
  - Schema and physical organization modification
  - Granting users authority to access the database
  - Backing up data
  - Monitoring performance and responding to changes in requirements
    - Database tuning

THE DATABASE SYSTEM ENVIRONMENT: DBMS COMPONENT MODULES

The figure is divided into **two halves**.

- The top half of the figure refers to the various users of the database environment and their interfaces.
- The lower half shows the internals of the DBMS responsible for storage of data and processing of transaction.

1. Database System Environment
   a. DBMS Component Modules
   b. Database System Utilities
      - Loading
      - Backup
      - File Reorganization
      - Performance Monitoring
   c. Other Tools
      - CASE tools
      - Data Dictionary System
      - Application Development Environment
      - Communications Software


Users:
DBA Staff
Casual Users
Application Programmers
Parametric Users
DDL Statements
Privileged Commands
Interactive Query
Application Programs
DDL Compiler
Query Compiler
Precompiler
Host Language Compiler
Query Optimizer
DML Compiler
Compiled Transactions
DBA Commands, Queries, and Transactions
System Catalog/ Data Dictionary
Runtime Database Processor
Stored Data Manager
Concurrency Control/ Backup/Recovery Subsystems
Stored Database
Input/Output from Database
Query and Transaction Execution

CONTINUED..

- The **DBA staff works** on defining the database and tuning it by making changes to its definition using the DDL and other privileged commands.
- **DDL compiler** processes schema definitions, specified in the DDL, and stores descriptions of the schemas (meta-data) in the DBMS catalog.
- **Catalog** includes information such as the names and sizes of files, names and data types of data items, storage details of each file, mapping information among schemas, and constraints, in addition to many other types of information that are needed by the DBMS modules

- **Casual users** and persons with occasional need for information from the database interact using some form of interface, **interactive query interface**. These queries are parsed, analyzed for correctness of the operations for the model, the name of data elements, and so on by a **query compiler** that compiles them into an internal form.

CONTINUED..

- **Query optimizer** is concerned with rearrangement and possible reordering of operations, elimination of redundancies, and use of correct algorithms and indexes during execution. It consults the system catalog for statistical and other physical information about the stored data and generates executable code that performs the necessary operations for the query and makes calls on the runtime processor.

- **Precompiler** extracts DML commands from an application program written in a host programming language. These commands are sent to the DML compiler for compilation into object code for database access. The rest of the programming is sent to the **host language compiler.**

CONTINUED..

- In lower half of the figure the runtime database processor is shown to execute

1. The privileged commands,
2. The executable query plans, and
3. The canned transactions with runtime parameters.

It works with the system dictionary and may update it with statistics.

It works with the stored data manager, for carrying out low-level input/output operations between disk and memory, management of buffers in main memory.

Concurrency control and backup and recovery systems separately as module for transaction management.

DATABASE SYSTEM UTILITIES

- **Common utilities have the following types of functions:**

1. **Loading** – is used to load existing data files into the database, transferring data from one DBMS to another, loading programs are called **conversion tools**.
2. **Backup** – dumping the entire database onto tape, this backup can be used to restore the database.
3. **Database storage reorganization** – reorganize a set of database files into a different file organization to improve performance.
4. **Performance monitoring** – monitors database usage and provides information to the DBA
5. Other utilities for sorting files, handling data compression, monitoring access by users, interfacing with the network, and performing other functions

TOOLS, APPLICATION ENVIRONMENTS, AND COMMUNICATIONS FACILITIES

- **CASE tools** used in design phase of database systems.
- **Data dictionary / data repository** is used to store catalog.
- (Stored information like schema descriptions and other information such as design decisions, application program descriptions, user information, usage standards, etc. Such system called **information repository**.)
- ***Active* data dictionary** is accessed by DBMS software and users/DBA,
- ***Passive* data dictionary** is accessed by users/DBA only.
- **Application development environment** systems provide an environment for developing database applications.

.Ex: Jbuilder, PowerBuilder system

- (It used for database design, GUI development, querying and updating, and application program development)

CONITNUED..

- **Communications software**, allow users to remote access database.
- These are connected to the database site through data communications hardware such as Internet routers, phone lines, long-haul networks, local networks, or satellite communication devices
- ▫The integrated DBMS and data communications system is called a **DB/DC system**