OVERVIEW OF DATABASE MANAGEMENT SYSTEM

Overview of Database management system

- Database and need for DBMS
- Database applications
- Characteristics of DBMS
- Data independence
- Data abstraction
- Data models
- DBMS Architecture
- Database users
- Types Of databases

Why Databases

SOMAIYA VIDYAVIHAR UNIVERSITY
K J Somaiya Institute of Management

A Day in Neha's Life
Databases she interacts with each day

Neha goes for grocery shopping

1. Product Data
2. Customer Data
3. Inventory Data
4. Payment Data

Visits Hospital for an appt

1. Patient Data
2. Doctor & appt Data
3. Pharmacy Data

Online Shopping

1. Product Data
2. Customer Data
3. Payment Data
4. Recommendations

Plan for a travel

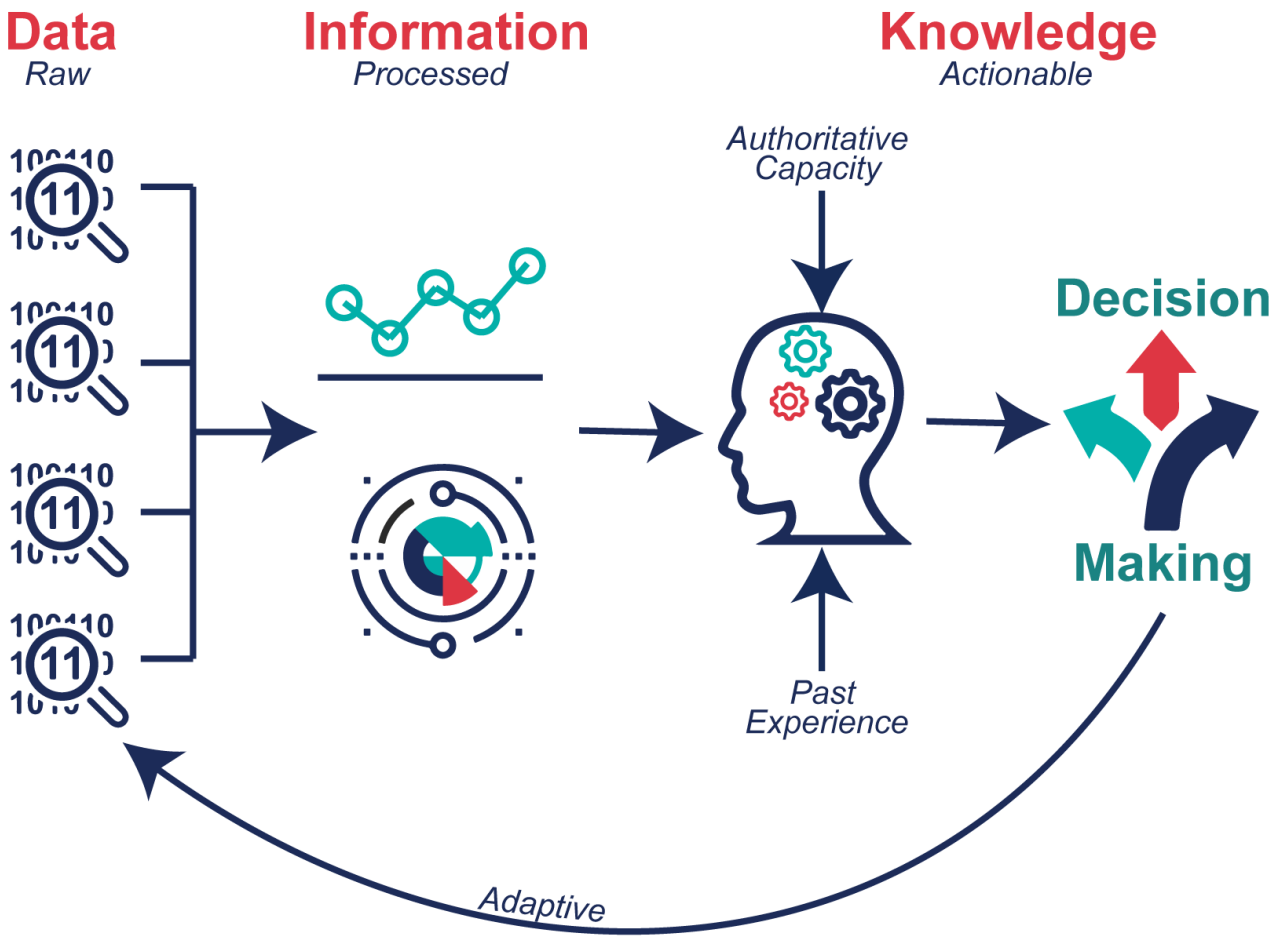
1. Flights
2. Hotels
3. Customers

Before bed, checks her social media accounts

1.Users
2. Posts
3. Friends
4. Recommendations

Database

- Data - Known facts that can be recorded and have an implicit meaning.
- Information - the result of processing raw data to reveal its meaning.
- Knowledge : familiarity , awareness and understanding of information
- Data Management – Proper data generation, storage and retrieval
- Database - A collection of related data.
  - User Data + Meta DAta



Why Databases

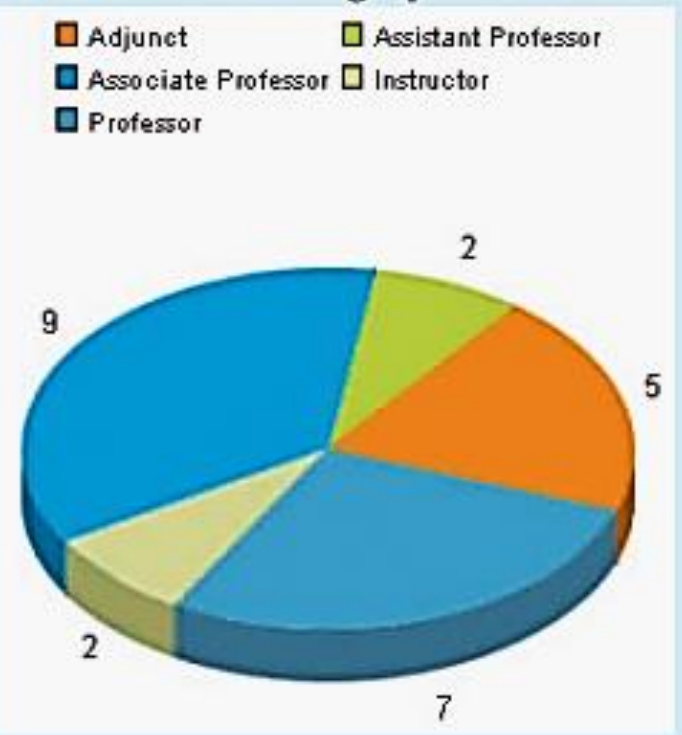
- All businesses must keep data.
- Availability of the data to decision makers.
- The ultimate purpose of all business information systems is to help businesses use information as an organizational resource.
- Decisions are based on information generated from data.
- *Information is produced by processing data and reveals the meaning of data.*
- *Accurate, relevant, and timely information is the key to good decision making.*
- *Good decision making is the key to organizational survival in a global environment.*

6 Adams	Q	John	ACCT	N418	jqadams@mtsu.edu	Associate Professor	1989 PhD
7 Jackson	C	Andrew	ECON	N303	ajackson@mtsu.edu	Associate Professor	1999 PhD
8 Van Buren	T	Martin	FIN	N306	mvanburen@mtsu.edu	Professor	1988 PhD
9 Harrison	R	William	MKTG	N118	wharrison@mtsu.edu	Professor	1994 PhD
10 Tyler	M	John	MGMT		JTyler@mtsu.edu	Assistant Professor	2000 Ed D
11 Polk		Cheryl	MKTG	N143	cpolk@mtsu.edu	Associate Professor	2002 PhD
12 Taylor	G	Zachary	ACCT	N415	ztaylor@mtsu.edu	Associate Professor	1996 PhD
13 Fillmore		Millard	JCB	N219	mfillmore@mtsu.edu	Professor	1992 PhD
14 Pierce	A	Franklin	MKTG	N359	fpierce@mtsu.edu	Instructor	2005 MBA
15 Buchanan	T	James	MGMT	N146	jbuchanan@mtsu.edu	Associate Professor	1996 DBA
17 Lincoln	W	Larry	MGMT	N150	llincoln@mtsu.edu	Associate Professor	1995 PhD
18 Johnson		Andrew	ISYS	N360	ajohnson@mtsu.edu	Professor	1987 PhD
19 Grant		Katie	MKTG	N120	kgrant@mtsu.edu	Assistant Professor	1989 DBA
20 Rutherford		Hayes	ACCT	N408	hrutherford@mtsu.edu	Professor	1992 PhD
21 Garfield	T	Denise	ACCT		dgarfield@mtsu.edu	Assistant Professor	2018 PhD
22 Arthur		Emily	ACCT	N413	earthur@mtsu.edu	Associate Professor	2003 JD
23 Cleveland	G	Robert	ACCT	N401	rcleveland@mtsu.edu	Associate Professor	1997 PhD
24 Harrison	X	Patricia	BULA	N406	pharrison@mtsu.edu	Associate Professor	2001 JD
25 McKinley	B	Priscilla	ISYS	N363	pmckinley@mtsu.edu	Adjunct	1994 MS
26 Roosevelt	F	Hillary	MGMT	N104	hroosevelt@mtsu.edu	Associate Professor	2002 PhD
27 Wilson		Laura	BCEN	N449	lwilson@mtsu.edu	Professor	1992 PhD
28 Harding		Warren	MKTG	N114	wharding@mtsu.edu	Professor	1984 Ed D
29 Coolidge		Calvin	ECON	N316	ccoolidge@mtsu.edu	Professor	1975 PhD
30 Hoover		Lisa	MGMT		lhoover@mtsu.edu	Adjunct	1978 MBA
31 Truman		Betty	ACCT	N416	btruman@mtsu.edu	Professor	1971 Ed D
32 Johnson		Robert	BCEN	N240	rjohnson@mtsu.edu	Professor	2001 PhD

c) Information in summary format

Rank	COUNT	%/INFS	TOT/COL	%/COL. TOT.	%/COL. FAC.
Adjunct	5	20.00%	23	21.74%	3.27%
Assistant Professor	2	8.00%	28	7.14%	1.31%
Associate Professor	9	36.00%	37	24.32%	5.88%
Instructor	2	8.00%	18	11.11%	1.31%
Professor	7	28.00%	47	14.89%	4.58%

d) Information in graphical format



Evolution of file system Data processing

- **Manual File System**
- **Computerized File Systems**

Limitations of data processing systems

Data Redundancy and inconsistency

Difficulty in accessing the data

Data Isolation

Concurrent access anomalies

Security problems

Integrity Problems

Database Management System (DBMS)

- a complex set of software programs that controls the organization, storage, management and retrieval of data in a database.
- categorized according to the data structure or types. It is a set of prewritten programs that are used to store, update and retrieve a database.
- accepts request for data from the application program and instructs the operating system to transfer the appropriate data.

Database System

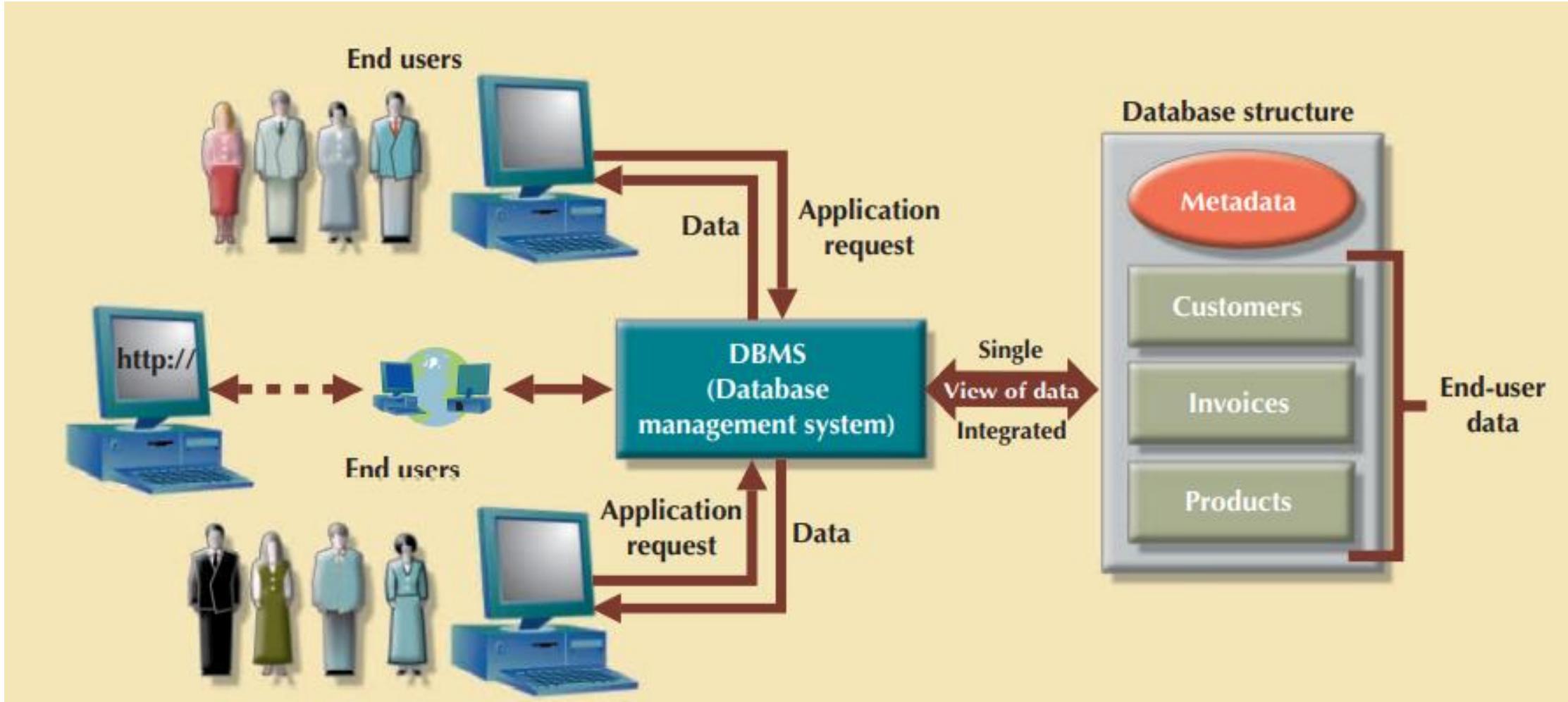
- The DBMS software together with the data.
- Sometimes, the applications are also included.

DBMS provides advantages

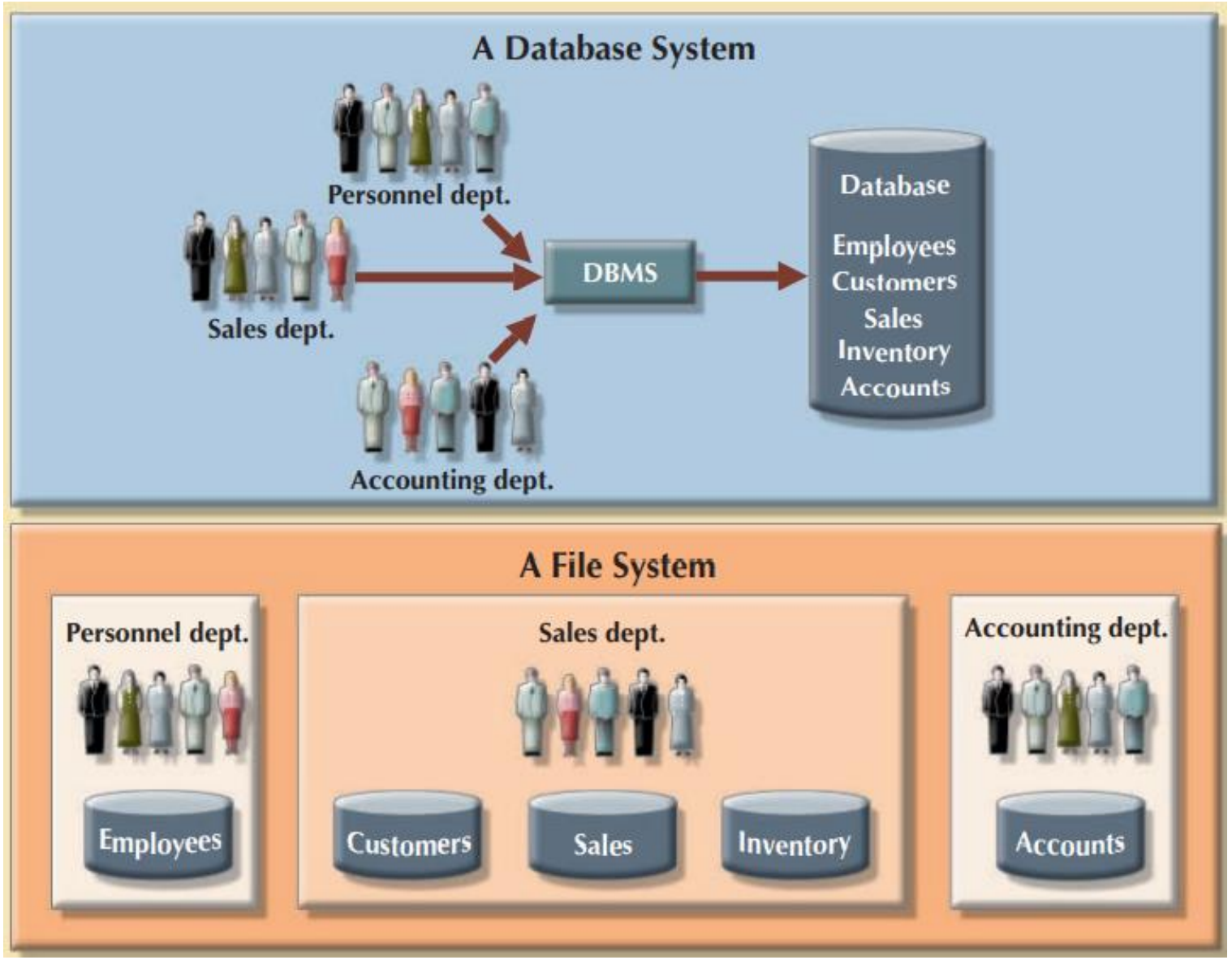
- Improved Data Sharing
- Improved Data Security
- Better Data Integration
- Minimized Data Inconsistency
- Improved Data Access
- Improved Decision Making
- Increased End-User Productivity

Role of DBMS

- The DBMS serves as the intermediary between the user and the database.
- The database structure itself is stored as a collection of files, and the only way to access the data in those files is through the DBMS.
- The DBMS presents the end user (or application program) with a single, integrated view of the data in the database.
- The DBMS receives all application requests and translates them into the complex operations required to fulfill those requests.
- The DBMS hides much of the database's internal complexity from the application programs and users.
- The application program might be written by a programmer using a programming language such as Visual Basic.NET, Java, or C#, or it might be created through a DBMS utility program


End users
Data
Application request
Database structure
Metadata
Customers
Invoices
Products
End-user data
http://
DBMS (Database management system)
Single
View of data
Integrated
End users
Application request
Data

- Logically related data stored in a single logical data repository.
- Might be physically distributed among multiple storage facilities.
- Eliminates most of the file system's issues – inconsistency, redundancy, dependance…
- Stores data structures, relationship between those structures and access paths to those structures


A Database System
Personnel dept.
Sales dept.
Accounting dept.
DBMS
Database
Employees
Customers
Sales
Inventory
Accounts
A File System
Personnel dept.
Employees
Sales dept.
Customers
Sales
Inventory
Accounting dept.
Accounts

Characteristics

- Multiuser and concurrent access
- Data integrity
- Data security
- Less redundancy
- Backup and Recovery
- ACID
- Sharing of Data
- ...

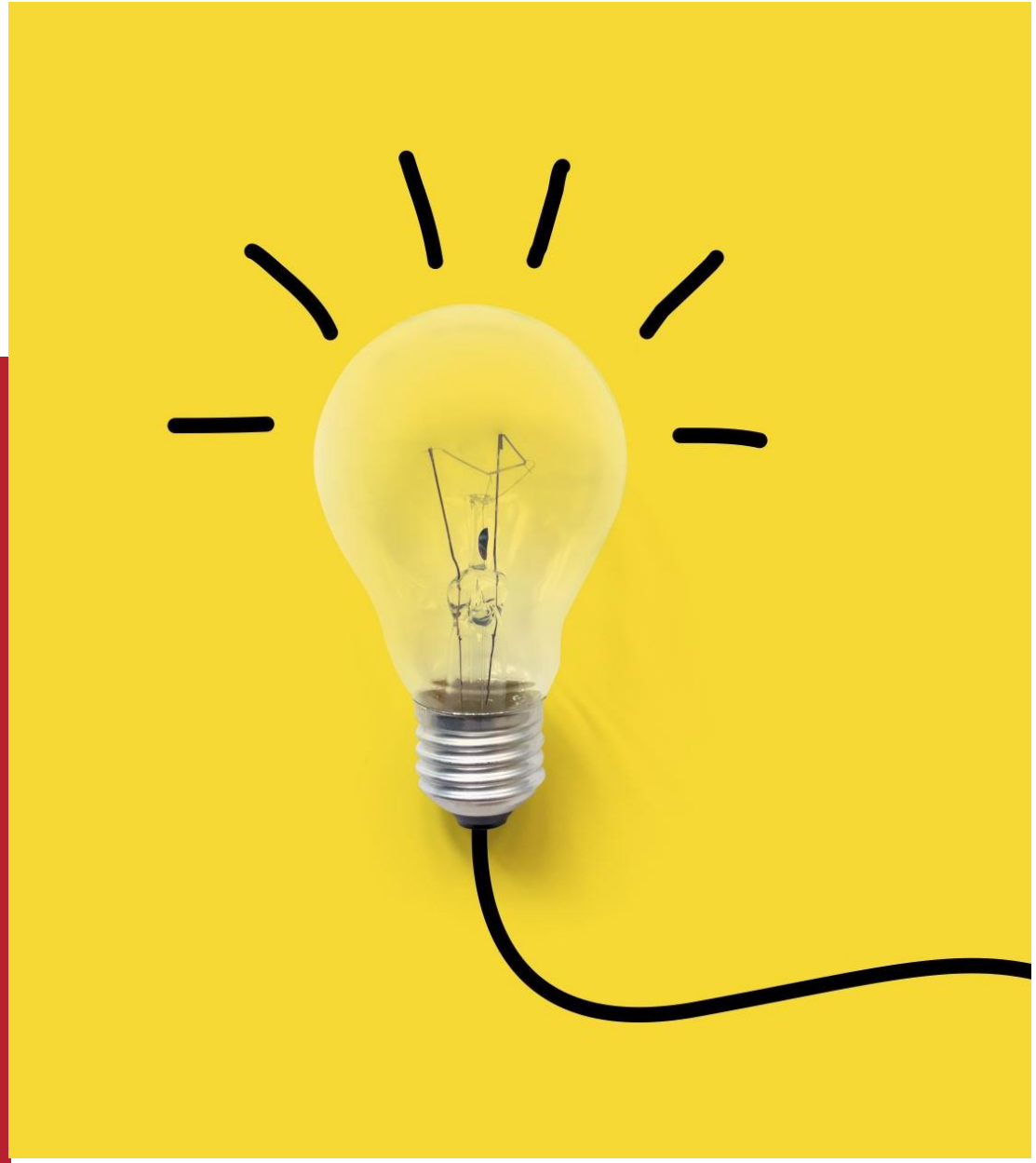
Database System Environment

- Database System – Define and regulate the collection, storage, management and use of data.
- It is composed of
  - Hardware, Software
  - People
  - Procedures
  - Data
- Database solutions must be cost effective and tactically strategically effective.

Need for DBMS

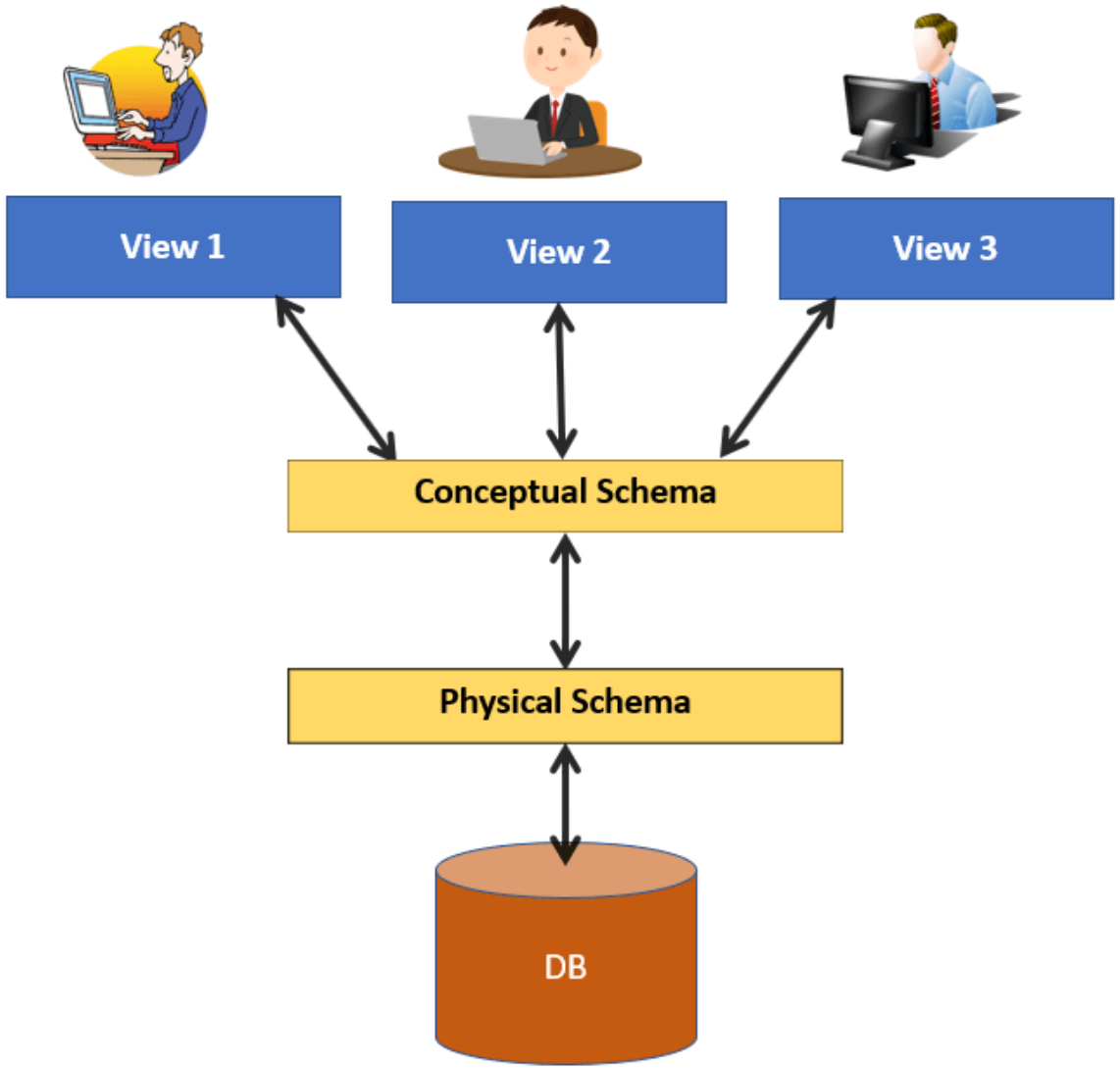
- Database contains information about a particular enterprise
  - Collection of interrelated data
  - Set of program to access the data
  - An environment which is efficient and convenient to use
- **Database Applications**
  - Banking: all transactions
  - Airlines : schedule and reservations
  - Universities: Registrations and grades
  - Sales: Customer, products, purchases
  - Online retailer: Order tracking, Customized Recommendations
  - Manufacturing: Production, inventory, orders
  - Human Resource : Employee records, tax deductions, salaries
- Database touch all aspects of life.

Difference Database Management Systems



Task 1: Design the HR Data

Dmine Pvt. Ltd. is a data management and analytical service provider with an employee strength of 100. The company wants to store and manage the employee data to apply various operations on time. Discuss on what data need to be stored for each employee and design a data management sheet to store and fetch information.



Data Abstraction

Several levels of abstractions in order to simplify user's interactions with the systems

Data Abstraction

- **Physical level:**
  - describes how a record (e.g., customer) is stored.
  - Low level complex data structures are described in detail.
- **Logical level:**
  - describes what data is stored in database, and the relationships among the data.
  - used by the database administration who must decide what information is to be kept in the database.
- **View level:**
  - Describes only part of the entire database
  - Application programs hide details of data types.
  - Views can also hide information (such as an employee's salary) for security purposes.

Data Abstraction

```
type customer = record
                    customer_id : string;
                    customer_name : string;
                    customer_street : string;
                    customer_city : string;
                end;
```

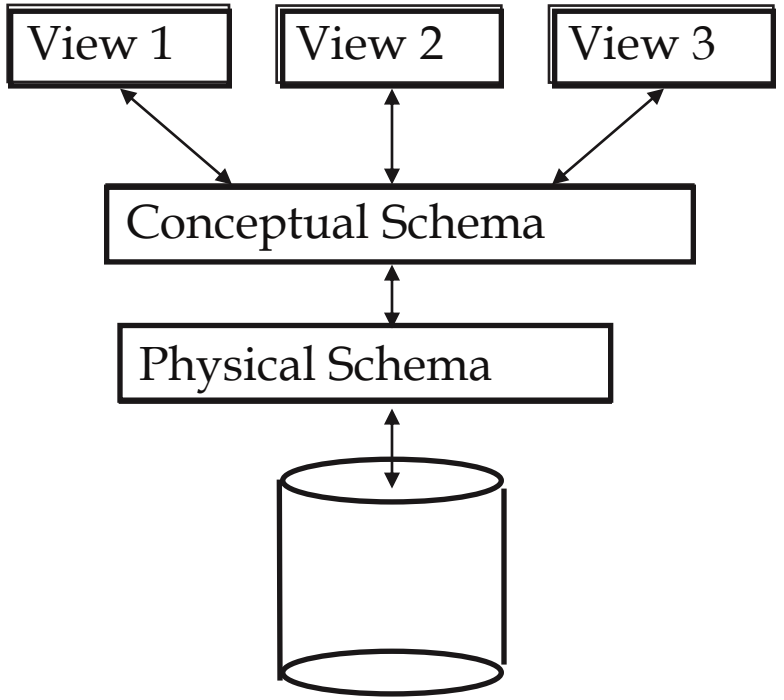
- At **physical level**, a customer record can be described as a block of consecutive storage locations (For example, words or bytes).
- The database system hides the lowest level storage details from database programmer.
- Database administration may be aware of certain details of the physical organization of the data

Data Abstraction

- At the **logical level**, each such record is described by a type definition
- The interrelationship among these record types is defined.
- Programmers & database administrators works at this level of abstraction.
- At the **view level**, users see a set of application programs that hide details of the data types.
- at the view level several views of the database are defined and the database users see these views.
- the views also provide a security mechanism to prevent users from accessing parts of the database.
- For example, customer details cannot be accessed by other cutomers.

Data Abstraction

- Many *views*, single *conceptual (logical) schema* and *physical schema*.
  - Views describe how users see the data.
  - Conceptual schema defines logical structure
  - Physical schema describes the files and indexes used.

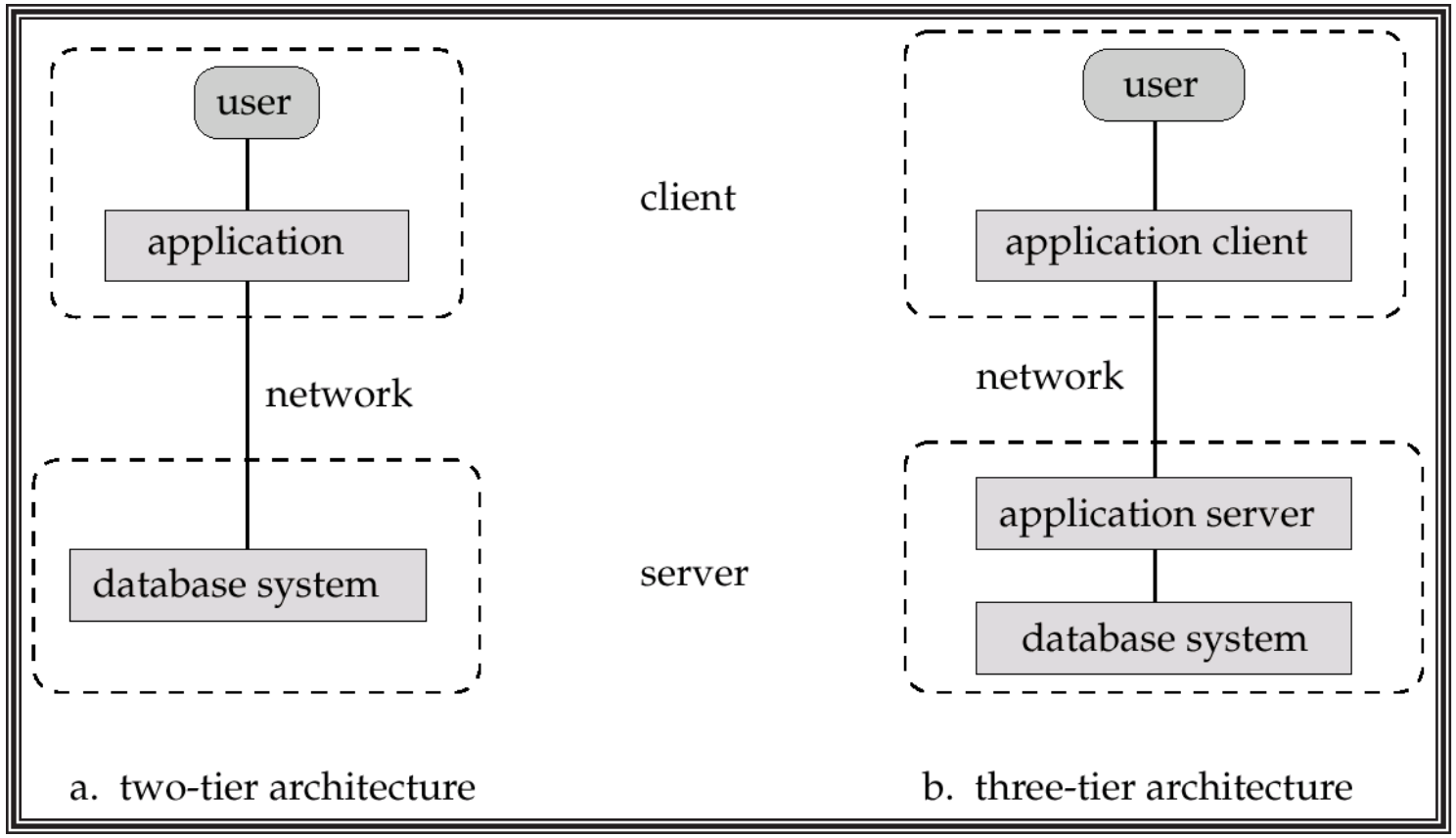


Data Independence

The ability to modify the Schema Definition in one level without affecting a schema definition in the next higher levels

- **Physical Data Independence** – the ability to modify the physical schema without changing the logical schema
  - Applications depend on the logical schema
  - the interfaces between the various levels and components should be well defined so that changes in some parts do not seriously influence others.
- **Logical Data Independence** – the ability to modify the conceptual schema without changing the view schema(causing the application programs to be rewritten).
- Logical Data independence is difficult to achieve than physical data independence since application programs are heavily dependent on the logical structure of the data they access.

Application Architectures



Instances and Schemas

- **Schema** – the logical structure of the database
  - Example: The database consists of information about a set of customers and accounts and the relationship between them)
  - **Physical schema**: database design at the physical level
  - **Logical schema**: database design at the logical level
- **Instance** – the actual content of the database at a particular point in time

STUDENT

Name	Student_number	Class	Major

COURSE

Course_name	Course_number	Credit_hours	Department

PREREQUISITE

Course_number	Prerequisite_number

SECTION

Section_identifier	Course_number	Semester	Year	Instructor

GRADE_REPORT

Student_number	Section_identifier	Grade

Instances and Schemas

:University Database

Example of a database state

COURSE

Course_name	Course_number	Credit_hours	Department
Intro to Computer Science	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Database	CS3380	3	CS

SECTION

Section_identifier	Course_number	Semester	Year	Instructor
85	MATH2410	Fall	04	King
92	CS1310	Fall	04	Anderson
102	CS3320	Spring	05	Knuth
112	MATH2410	Fall	05	Chang
119	CS1310	Fall	05	Anderson
135	CS3380	Fall	05	Stone

GRADE_REPORT

Student_number	Section_identifier	Grade
17	112	B
17	119	C
8	85	A
8	92	A
8	102	B
8	135	A

PREREQUISITE

Course_number	Prerequisite_number
CS3380	CS3320
CS3380	MATH2410
CS3320	CS1310

Figure 1.2
A database that stores student and course information.

Exercise

Create a database state for Trader data.

Employees

Customers

Order

Product

Payment

Design a database schema for Hospital Management System.

Data Models

- Record Based Models
  - Relational Model
  - Network Model
  - Hierarchical Model
- Object Based Model
  - Entity Relationship model
  - Object Oriented Model

Relational Model

- A relation appears as a two-dimensional table.
- The RDBMS organizes the data so that its external view is a set of relations or tables.
- The data is not stored as tables: the physical storage of the data is independent of the way in which the data is logically organized.

Attributes

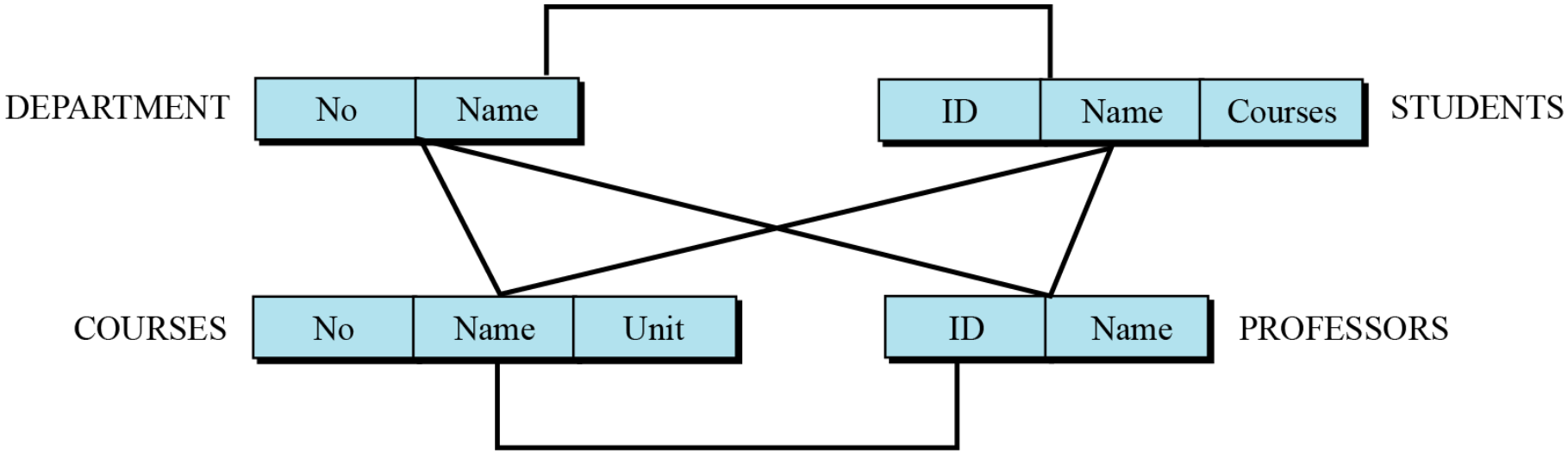
No	Course-Name	Unit
CIS15	Intro to C	5
CIS17	Intro to Java	5
CIS19	UNIX	4
CIS51	Networking	5

Tuples

COURSES

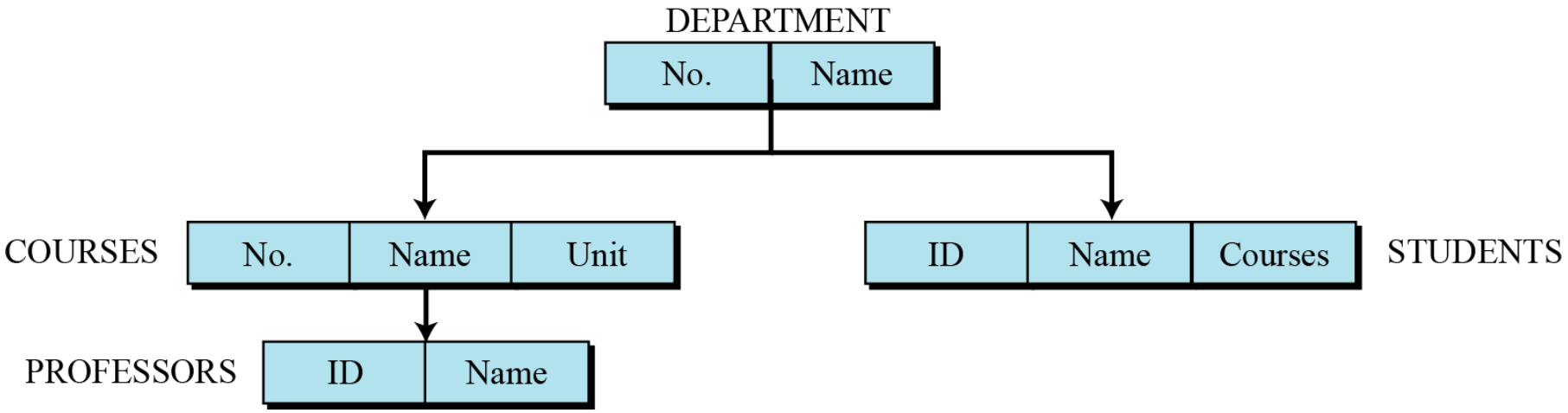
Network database model

In the network model, the entities are organized in a graph, in which some entities can be accessed through several paths

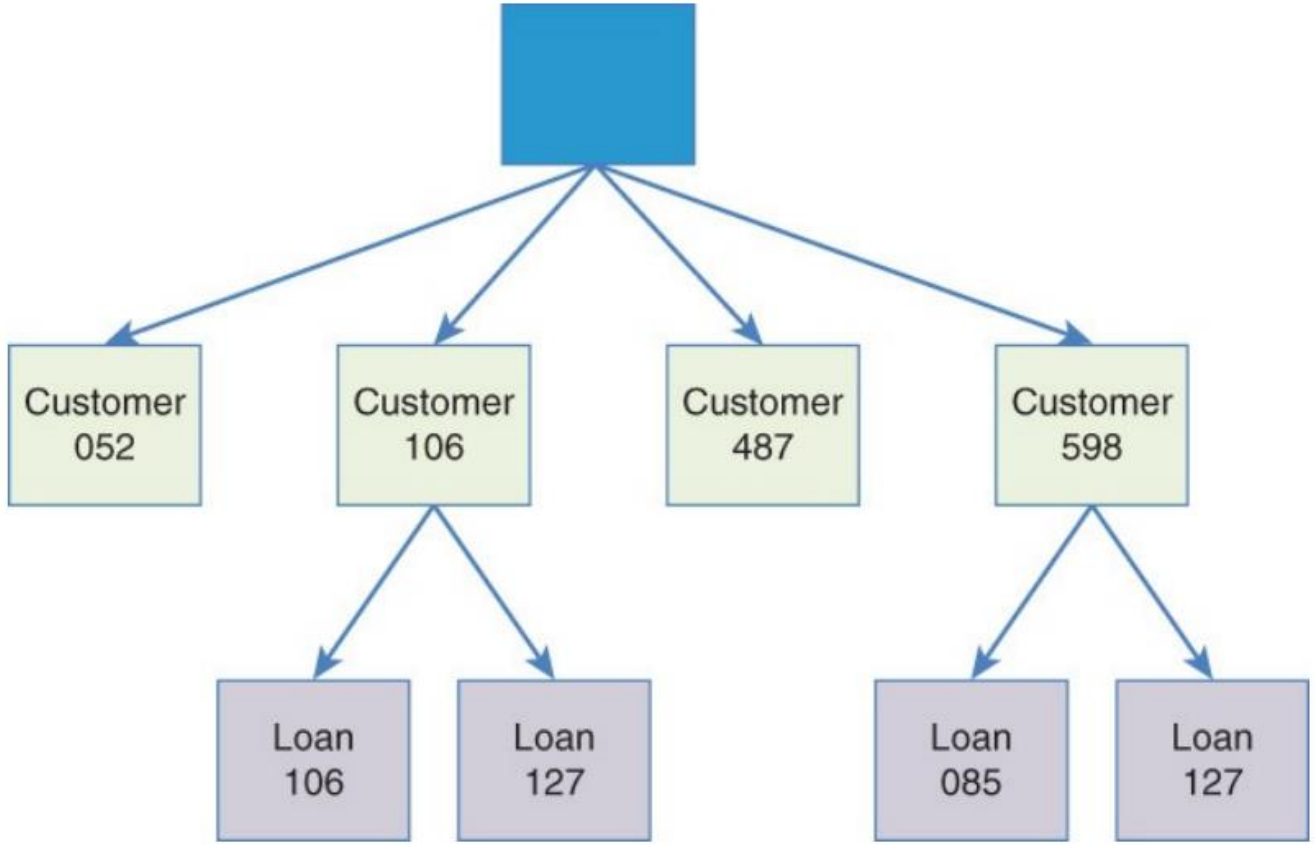


Hierarchical database model

- In the hierarchical model, data is organized as an inverted tree.
- Each entity has only one parent but can have several children.
- At the top of the hierarchy, there is one entity, which is called the root.



- Consider the kind of data the loan department of a bank may track. It has customers and each customer has one or more loans. For each customer, the loan department would want to track the customer's name, address, and phone number. For each loan, the loan department should track the amount of the loan, the interest rate, the date the loan was made, and the date the loan is due.

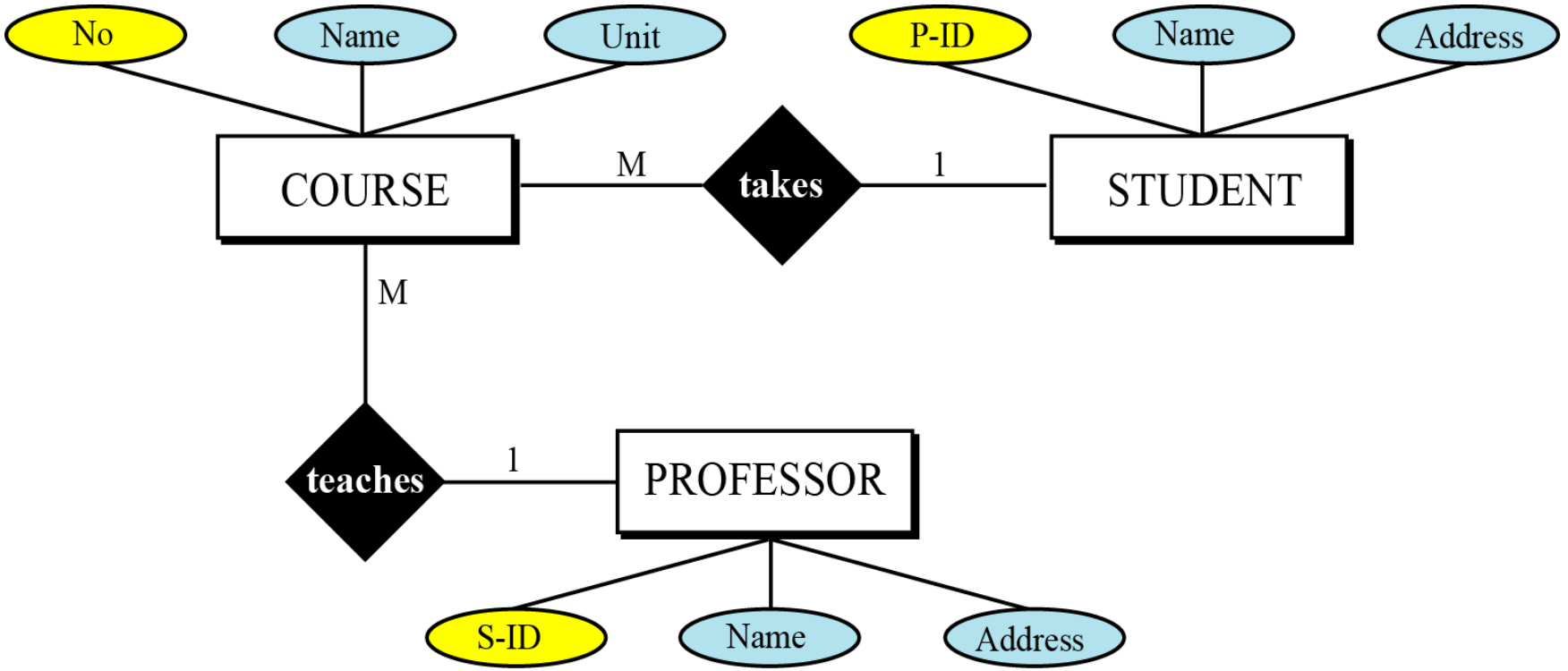


Entity-relationship models (ERM)

The database designer creates an entity-relationship (E-R) diagram to show the entities for which information needs to be stored and the relationship between those entities.

- Rectangles represent entity sets
- Ellipses represent attributes
- Diamonds represent relationship sets
- Lines link attributes to entity sets and link entity sets to relationships sets

a very simple E-R diagram with three entity sets, their attributes and the relationship between the entity sets.



Object-Oriented Data Model

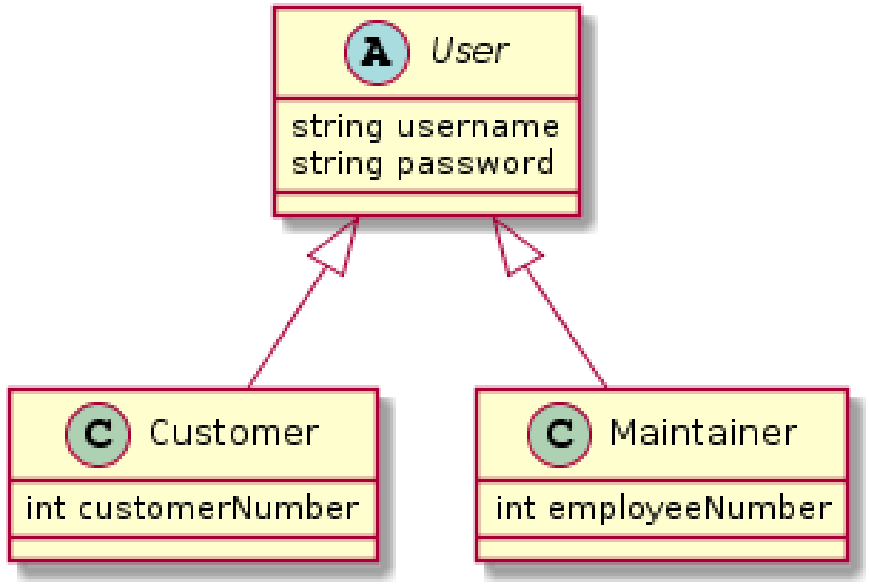
- Objects
- Attributes
- Operations
  - Encapsulation
    - Data and functions are combined in one object

Object-Oriented Data Model

- Inheritance

  - Data and functions are organized in a hierarchy

  - Objects inherit characteristics and functions of their ancestor objects



People who with the data base

- Database Administrator
- Database Users

Database Administrator

- A person who has central control of both the data and the programs that access those data.
- Database administrator's functions include:
  - Schema definition :using DDL
  - Storage structure and access method definition
  - Schema and physical organization modification
  - Granting authorization for data access
  - Routine maintenance
    - Back up the database
    - Ensuring enough disk space is available
    - Monitoring the jobs running on the database

Database Users

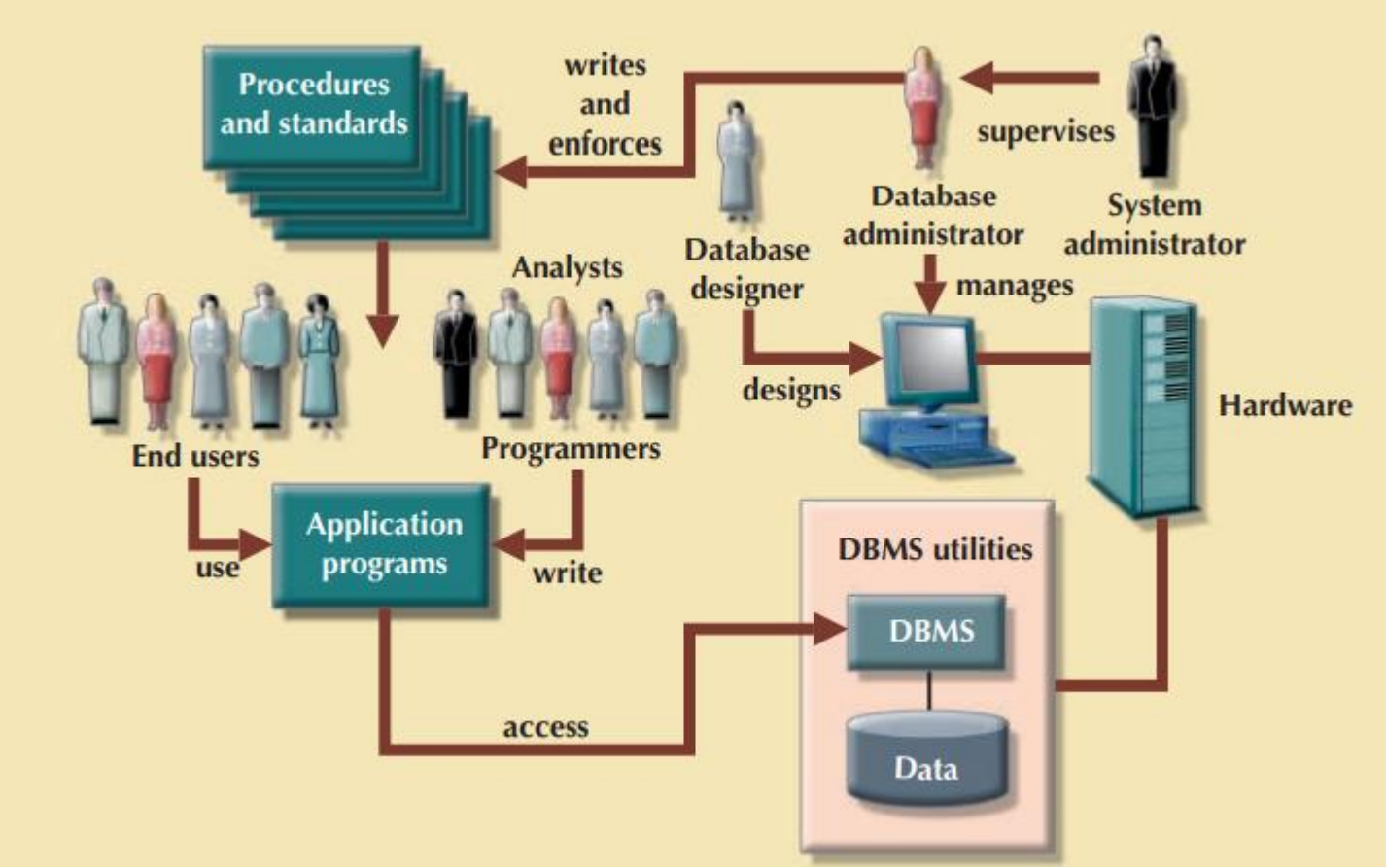
- Application programmers
- Sophisticated users
- Specialized users
- Naïve users

Database Users

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

- Application programmers – interact with system through DML calls
- Sophisticated users – form requests in a database query language
- Specialized users – write specialized database applications that do not fit into the traditional data processing framework
- Naïve users – invoke one of the permanent application programs that have been written previously
  - E.g. people accessing database over the web, bank tellers, clerical staff

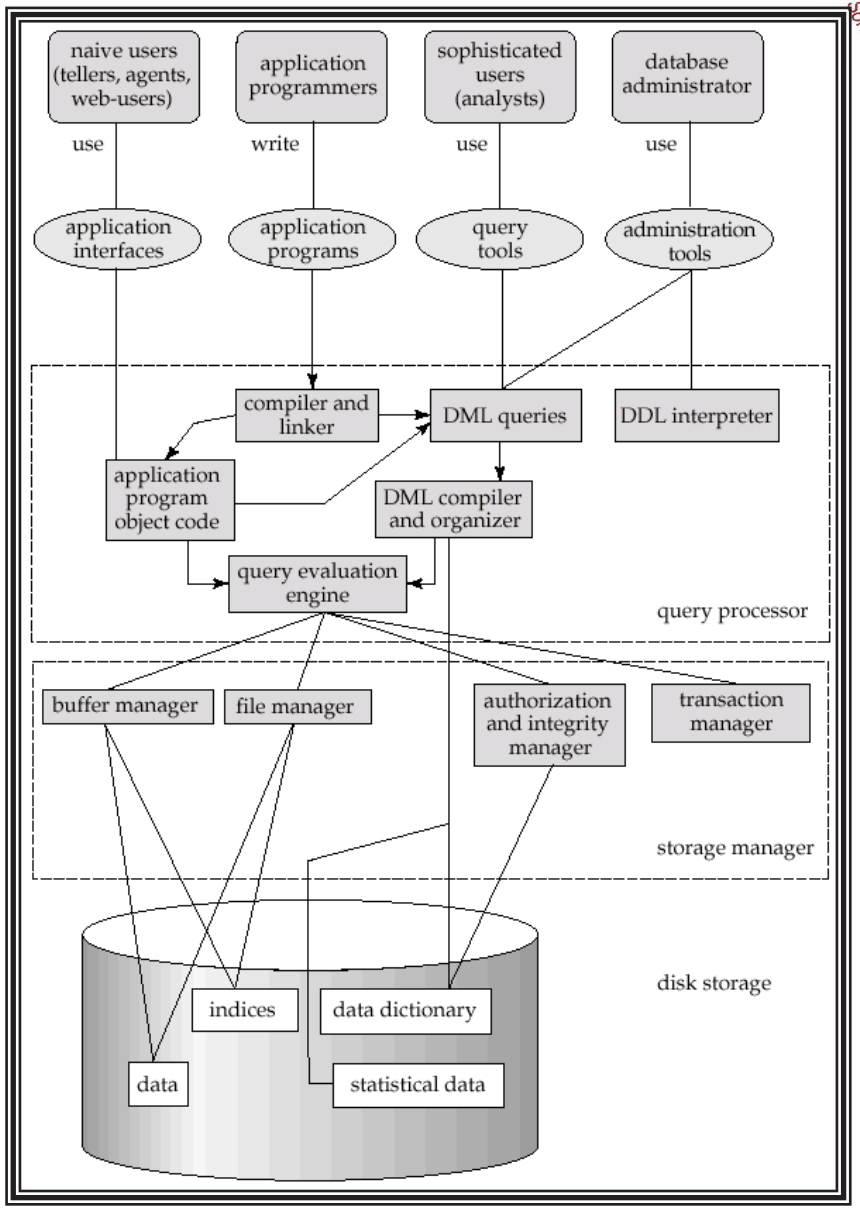
Database System Environment



Types of databases

- Several criteria are used to classify DBMSs.
  - Data model on which the DBMS is based.
    - Relational databases
    - Object oriented Databases
    - Hierarchical databases
    - Network Based databases
    - Non-relational databases are also called NoSQL databases
  - Number of users supported by the system.
    - Single user Databases: support only one user at a time
    - Multiuser databases: support multiple users concurrently
  - Number of sites over which the database is distributed
    - Centralized database :support multiple users, but the DBMS and the database themselves reside totally at a single computer site.
    - Distributed database: DBMS software distributed over many sites, connected by a computer network

Overall System Structure



ER model

Create an ER model for hospital management data.

Case : Hospital Management System

Aim: XYZ hospital is a multi specialty hospital that includes a number of departments, rooms, doctors, nurses, compounders, and other staff working in the hospital. Patients having different kinds of ailments come to the hospital and get checkup done from the concerned doctors. If required, they are admitted in the hospital and discharged after treatment. The aim of this case study is to design and develop a database for the hospital to maintain the records of various departments, rooms, and doctors in the hospital. It also maintains records of the regular patients, patients admitted in the hospital, the check up of patients done by the doctors, the patients that have been operated, and patients discharged from the hospital.

Description: In hospital, there are many departments like Orthopedic, Pathology, Emergency, Dental, Gynecology, Anesthetics, I.C.U., Blood Bank, Operation Theater, Laboratory, M.R.I., Neurology, Cardiology, Cancer Department, Corpse, etc. There is an OPD where patients come and get a card (that is, entry card of the patient) for check up from the concerned doctor. After making entry in the card, they go to the concerned doctor's room and the doctor checks up their ailments. According to the ailments, the doctor either prescribes medicine or admits the patient in the concerned department. The patient may choose either private or general room according to his/her need. But before getting admission in the hospital, the patient has to fulfill certain formalities of the hospital like room charges, etc. After the treatment is completed, the doctor discharges the patient. Before discharging from the hospital, the patient again has to complete certain formalities of the hospital like balance charges, test charges, operation charges (if any), blood charges, doctors' charges, etc. Next we talk about the doctors of the hospital. There are two types of the doctors in the hospital, namely, regular doctors and call on doctors. Regular doctors are those doctors who come to the hospital daily. Calls on doctors are those doctors who are called by the hospital if the concerned doctor is not available.