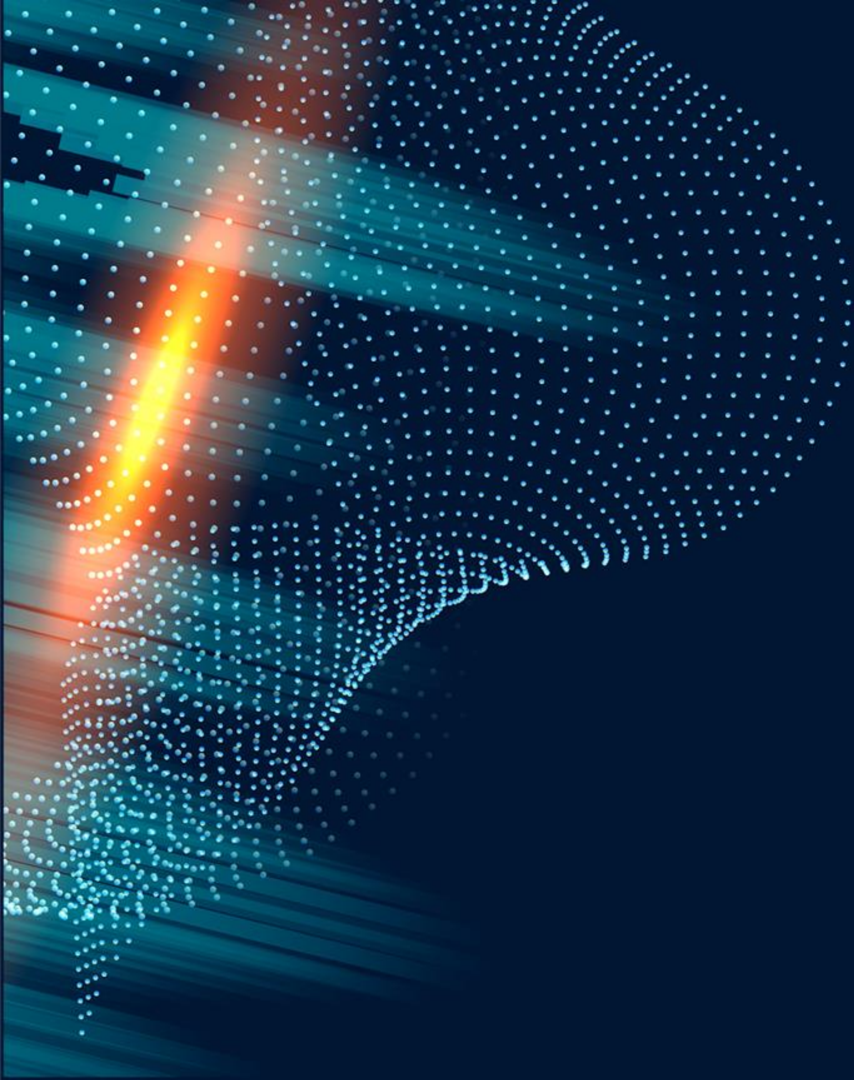
TOPIC 1

Fundamentals of Database Management System

Upon completion of this topic, students should be able to:

- 1.1 Identify database
- 1.2 Identify DBMS
- 1.3 Identify data model



1.1 Identify database

1.1 Identify database

1.1.1 Define database

Data:
Known facts that can be recorded and have an implicit meaning/ Raw facts; that is, facts that have not been yet processed to reveal their meaning to the end user.

Information:
Facts (data) that are arranged in meaningful patterns.

Database:
A collection of related data/ Shared collection of logically related data (and a description of this data), designed to meet the information needs of an organization.

Database Management System (DBMS):
A software package/ system to facilitate the creation and maintenance of a computerized database.

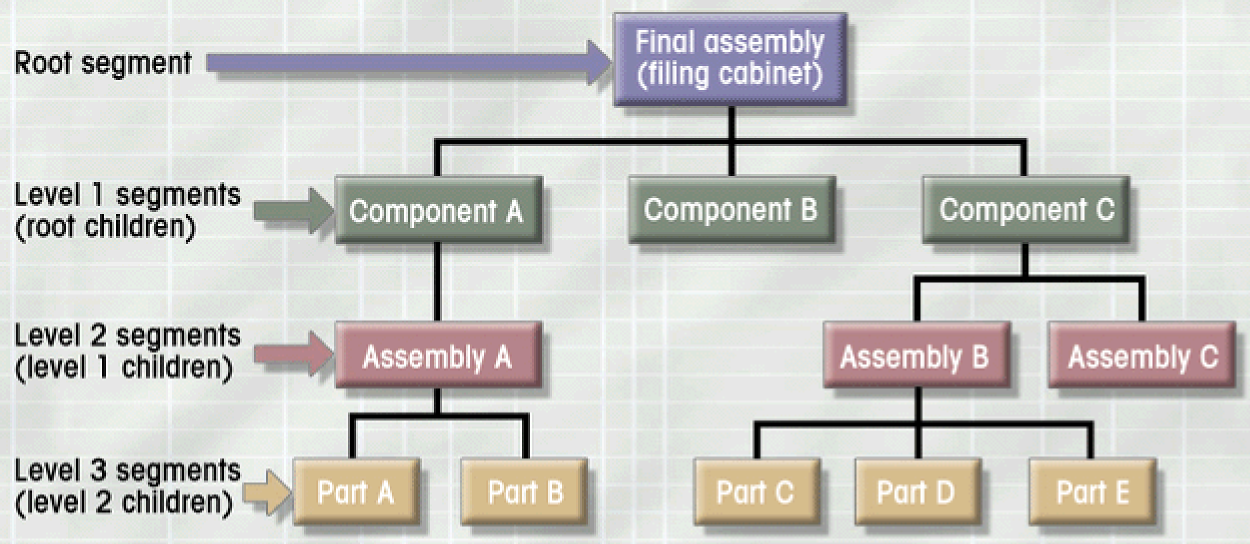
EVOLUTION OF DATABASE : TYPES OF DATA MODEL

- Object Based Logical Model
    - Entity Relationship Model

- Record Based Logical Models
    - Hierarchical Data Model
    - Network Data Model
    - Relational Data Model

Hierarchical Database Model

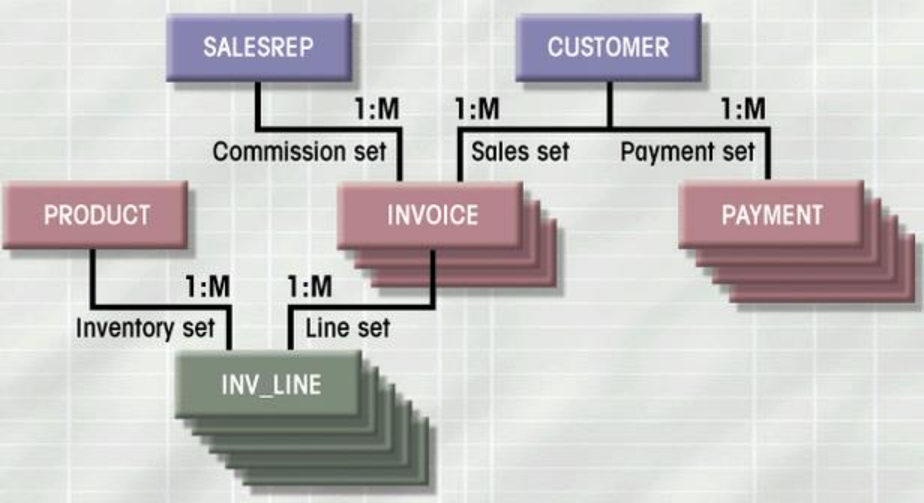
- Logically represented by an upside down tree
  - Each parent can have many children
  - Each child has only one parent


Root segment
Final assembly (filing cabinet)
Level 1 segments (root children)
Component A
Component B
Component C
Level 2 segments (level 1 children)
Assembly A
Assembly B
Assembly C
Level 3 segments (level 2 children)
Part A
Part B
Part C
Part D
Part E

- Advantages
  - Conceptual simplicity
  - Database security and integrity
  - Data independence
  - Efficiency
- Disadvantages
  - Complex implementation
  - Difficult to manage and lack of standards
  - Lacks structural independence
  - Applications programming and use complexity
  - Implementation limitations

Network Database Model

- Each record can have multiple parents
  - Composed of sets
  - Each set has owner record and member record
  - Member may have several owners



- Advantages
  - Conceptual simplicity
  - Handles more relationship types
  - Data access flexibility
  - Promotes database integrity
  - Data independence
  - Conformance to standards
- Disadvantages
  - System complexity
  - Lack of structural independence

Relational Database Model

- Perceived by user as a collection of tables for data storage
- Tables are a series of row/column intersections
- Tables related by sharing common entity characteristic(s)

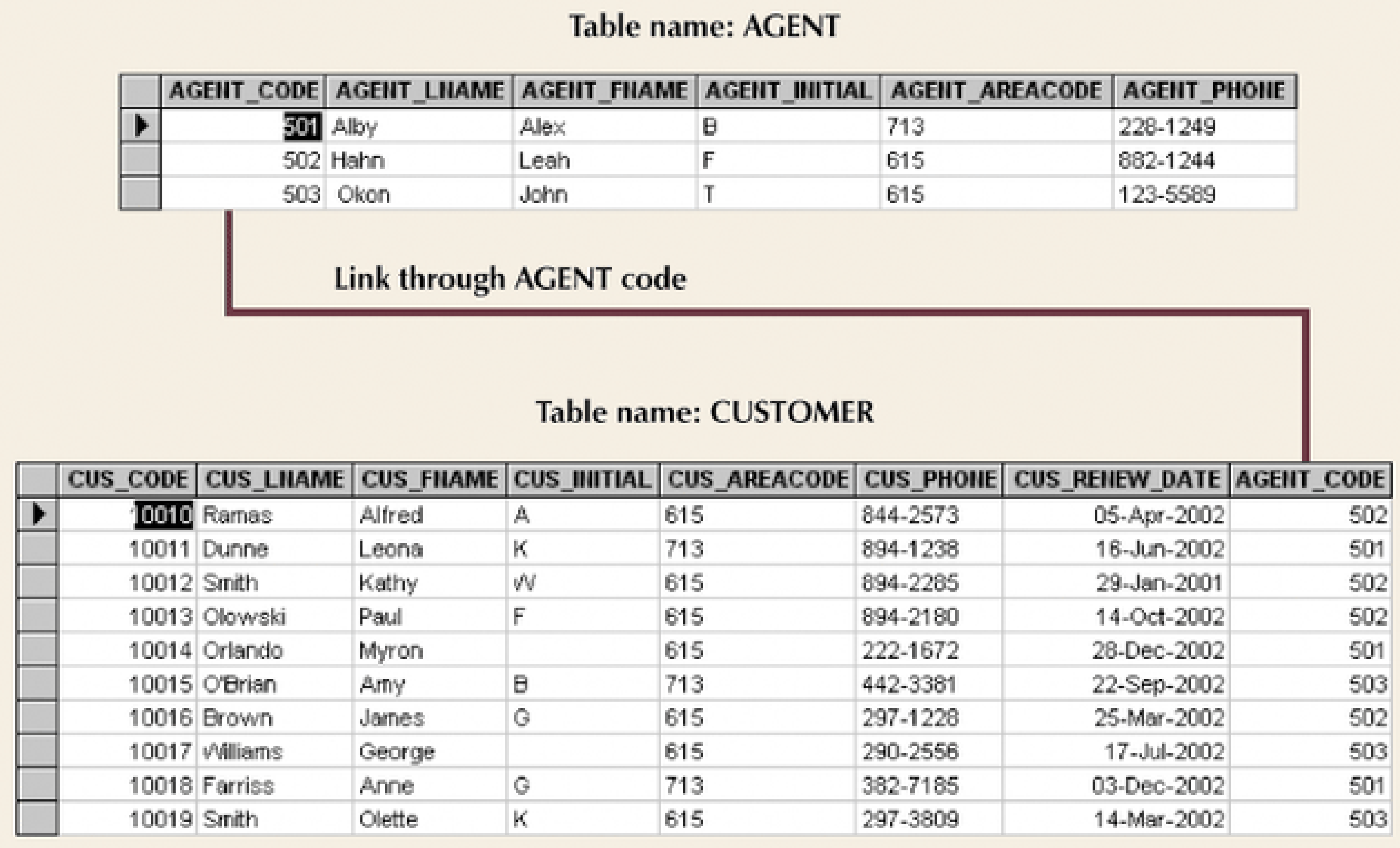


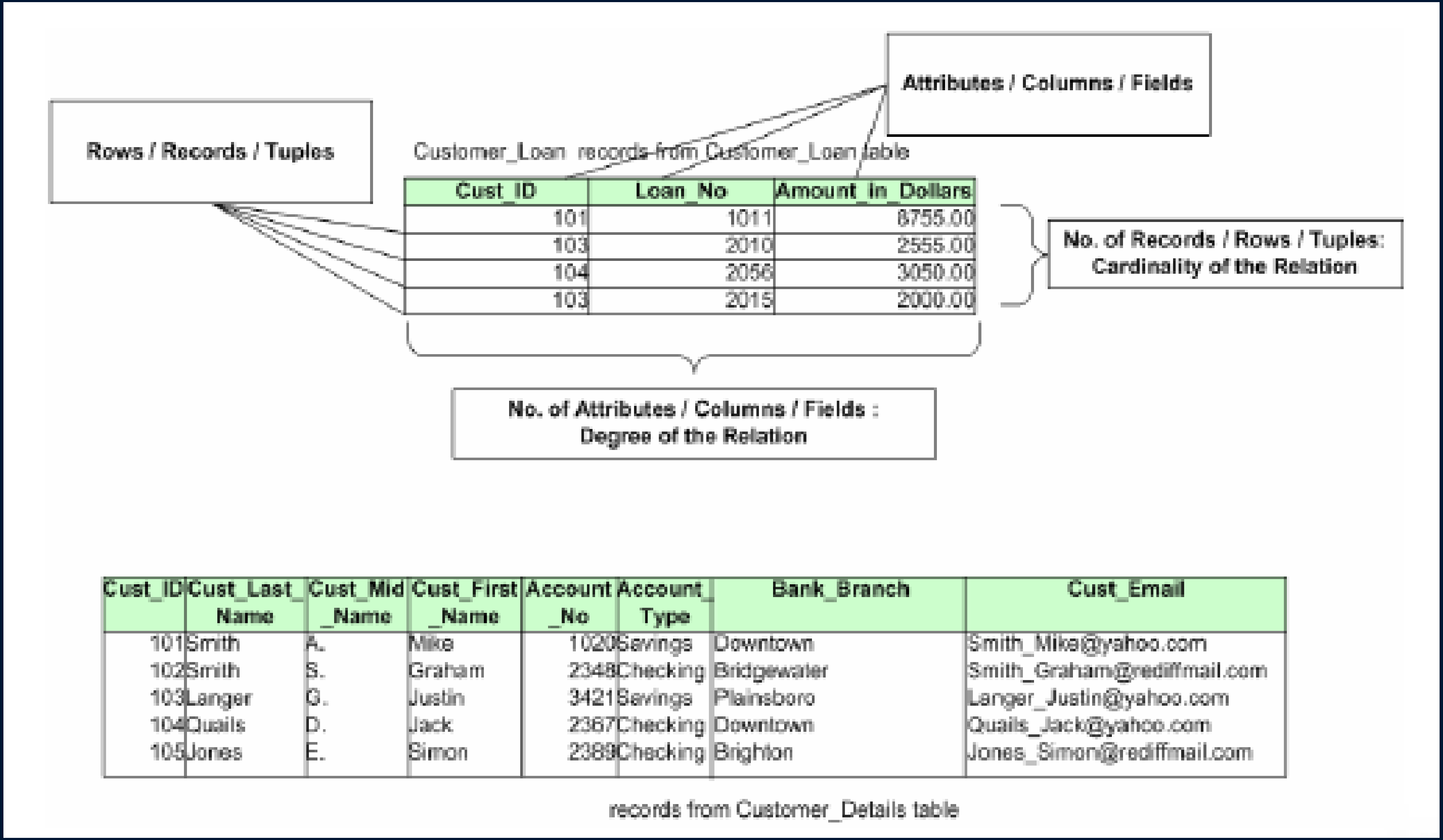
Table name: AGENT

AGENT_CODE	AGENT_LNAME	AGENT_FNAME	AGENT_INITIAL	AGENT_AREACODE	AGENT_PHONE
501	Alby	Alex	B	713	228-1249
502	Hahn	Leah	F	615	882-1244
503	Okon	John	T	615	123-5589

Link through AGENT code

Table name: CUSTOMER

CUS_CODE	CUS_LNAME	CUS_FNAME	CUS_INITIAL	CUS_AREACODE	CUS_PHONE	CUS_RENEW_DATE	AGENT_CODE
10010	Ramas	Alfred	A	615	844-2573	05-Apr-2002	502
10011	Dunne	Leona	K	713	894-1238	16-Jun-2002	501
10012	Smith	Kathy	W	615	894-2285	29-Jan-2001	502
10013	Olowski	Paul	F	615	894-2180	14-Oct-2002	502
10014	Orlando	Myron		615	222-1672	28-Dec-2002	501
10015	O'Brian	Amy	B	713	442-3381	22-Sep-2002	503
10016	Brown	James	G	615	297-1228	25-Mar-2002	502
10017	Williams	George		615	290-2556	17-Jul-2002	503
10018	Farriss	Anne	G	713	382-7185	03-Dec-2002	501
10019	Smith	Olette	K	615	297-3809	14-Mar-2002	503



Rows / Records / Tuples

Attributes / Columns / Fields

Customer_Loan records from Customer_Loan table

Cust_ID	Loan_No	Amount_in_Dollars
101	1011	8755.00
103	2010	2555.00
104	2056	3050.00
103	2015	2000.00

No. of Records / Rows / Tuples: Cardinality of the Relation

No. of Attributes / Columns / Fields : Degree of the Relation

Cust_ID	Cust_Last_Name	Cust_Mid_Name	Cust_First_Name	Account_No	Account_Type	Bank_Branch	Cust_Email
101	Smith	A.	Mike	1020	Savings	Downtown	Smith_Mike@yahoo.com
102	Smith	S.	Graham	2348	Checking	Bridgewater	Smith_Graham@rediffmail.com
103	Langer	G.	Justin	3421	Savings	Plainsboro	Langer_Justin@yahoo.com
104	Quails	D.	Jack	2367	Checking	Downtown	Quails_Jack@yahoo.com
105	Jones	E.	Simon	2389	Checking	Brighton	Jones_Simon@rediffmail.com

records from Customer_Details table

Relational Database Model

- Advantages
  - Structural independence
  - Improved conceptual simplicity
  - Easier database design, implementation, management, and use
  - Ad hoc query capability with SQL
  - Powerful database management system

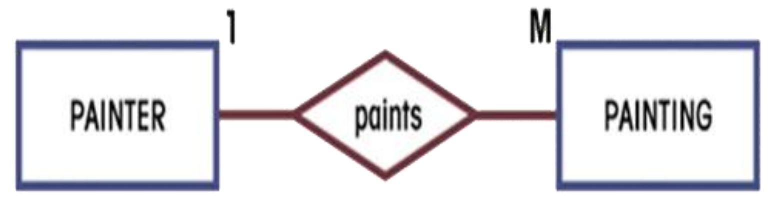
Relational Database Model

- Disadvantages
  - Substantial hardware and system software overhead
  - Poor design and implementation is made easy
  - May promote “islands of information” problems

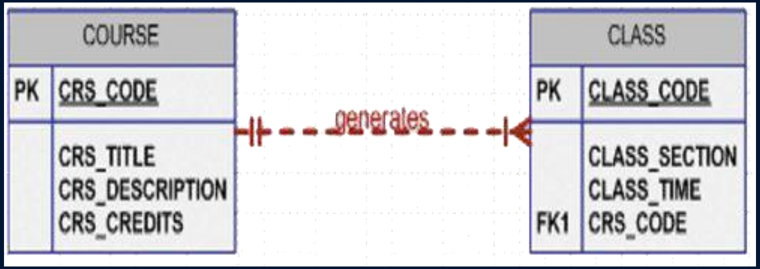
Entity Relationship Database Model

- Complements the relational data model concepts
- Represented in an entity relationship diagram (ERD)
- Based on entities, attributes, and relationships

Chen Model



Crow’s Foot Model



Entity Relationship Database Model

- Advantages
  - Exceptional conceptual simplicity
  - Visual representation
  - Effective communication tool
  - Integrated with the relational database model
- Disadvantages
  - Limited constraint representation
  - Limited relationship representation
  - No data manipulation language
  - Loss of information content

Object-Oriented Database Model

- Objects or abstractions of real-world entities are stored
  - Attributes describe properties
  - Collection of similar objects is a class
    - Methods represent real world actions of classes
    - Classes are organized in a class hierarchy
  - Inheritance is ability of object to inherit attributes and methods of classes above it

Object-Oriented Database Model

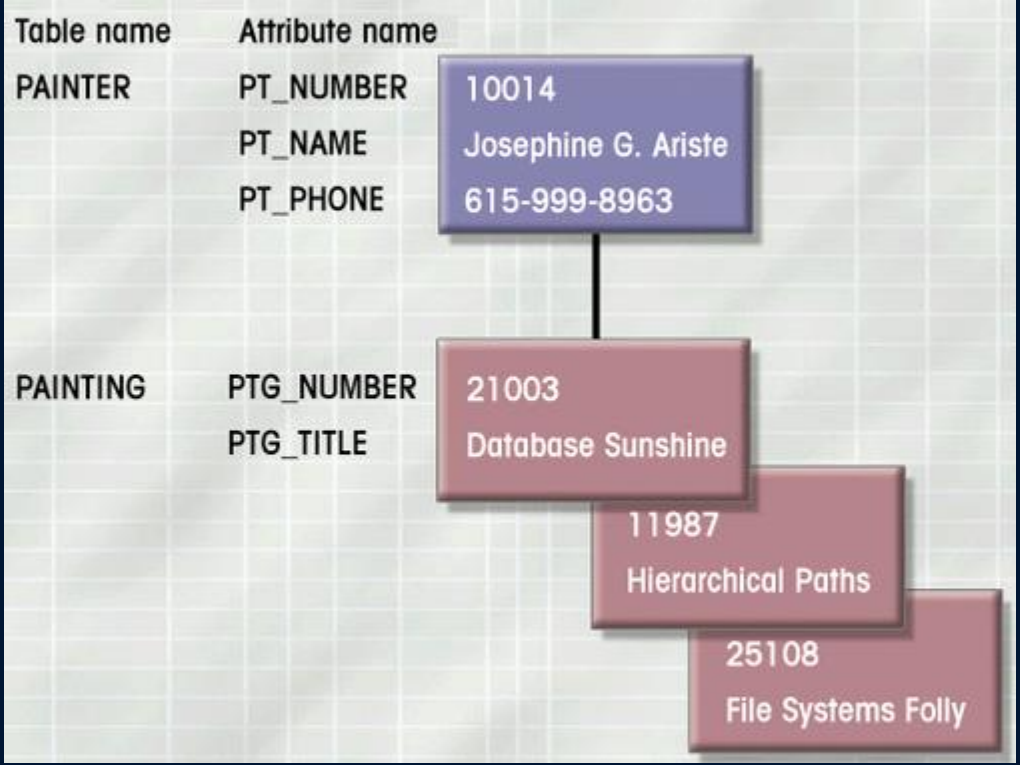


Table name

Attribute name

PAINTER

PT_NUMBER

PT_NAME

PT_PHONE

10014

Josephine G. Ariste

615-999-8963

PAINTING

PTG_NUMBER

PTG_TITLE

21003

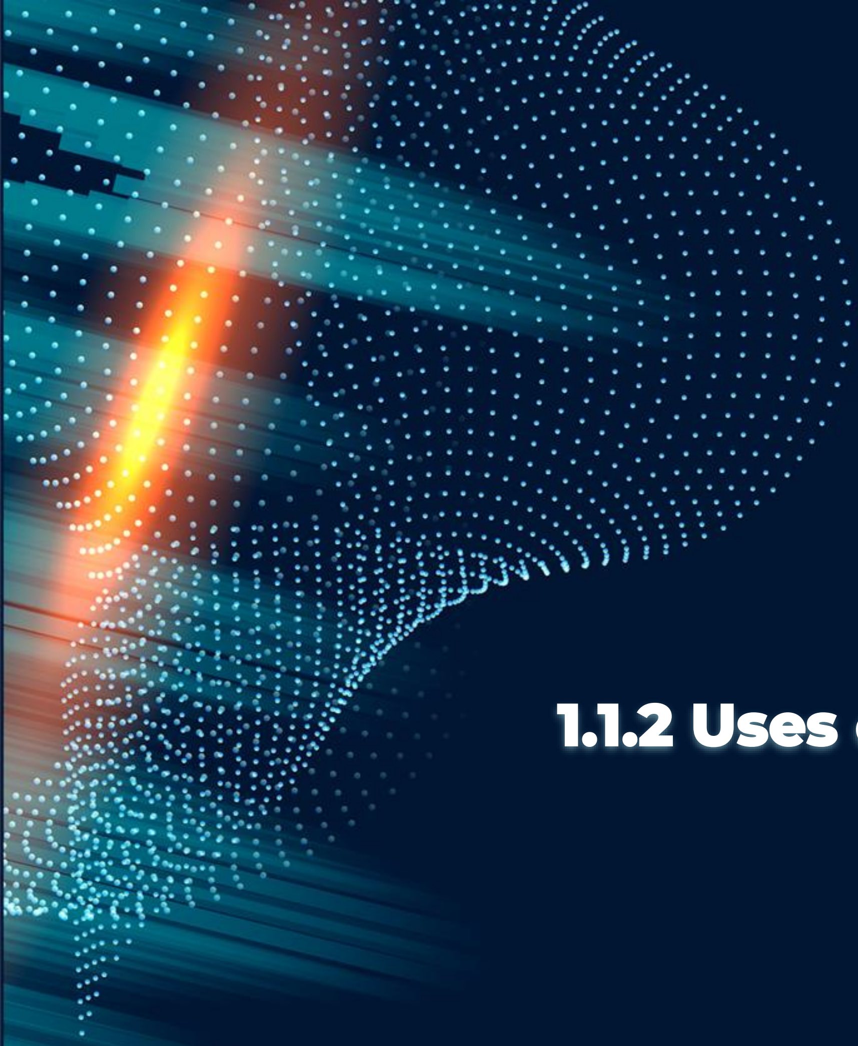
Database Sunshine

11987

Hierarchical Paths

25108

File Systems Folly



1.1.2 Uses of database in the business

Databases touch all aspects of our lives. Some of the major areas of application are as follows:

- Banking
- Airlines
- Universities
- Manufacturing and selling
- Human resources

- Give an example for each of these area?

- Improved data sharing: The DBMS helps create an environment in which end users have better access to more and better-managed data. Such access makes it possible for end users to respond quickly to changes in their environment.

- Improved data security. The more users access the data, the greater the risks of data security breaches.

- Better data integration. Wider access to well-managed data promotes an integrated view of the organization’s operations and a clearer view of the big picture. It becomes much easier to see how actions in one segment of the company affect other segments.

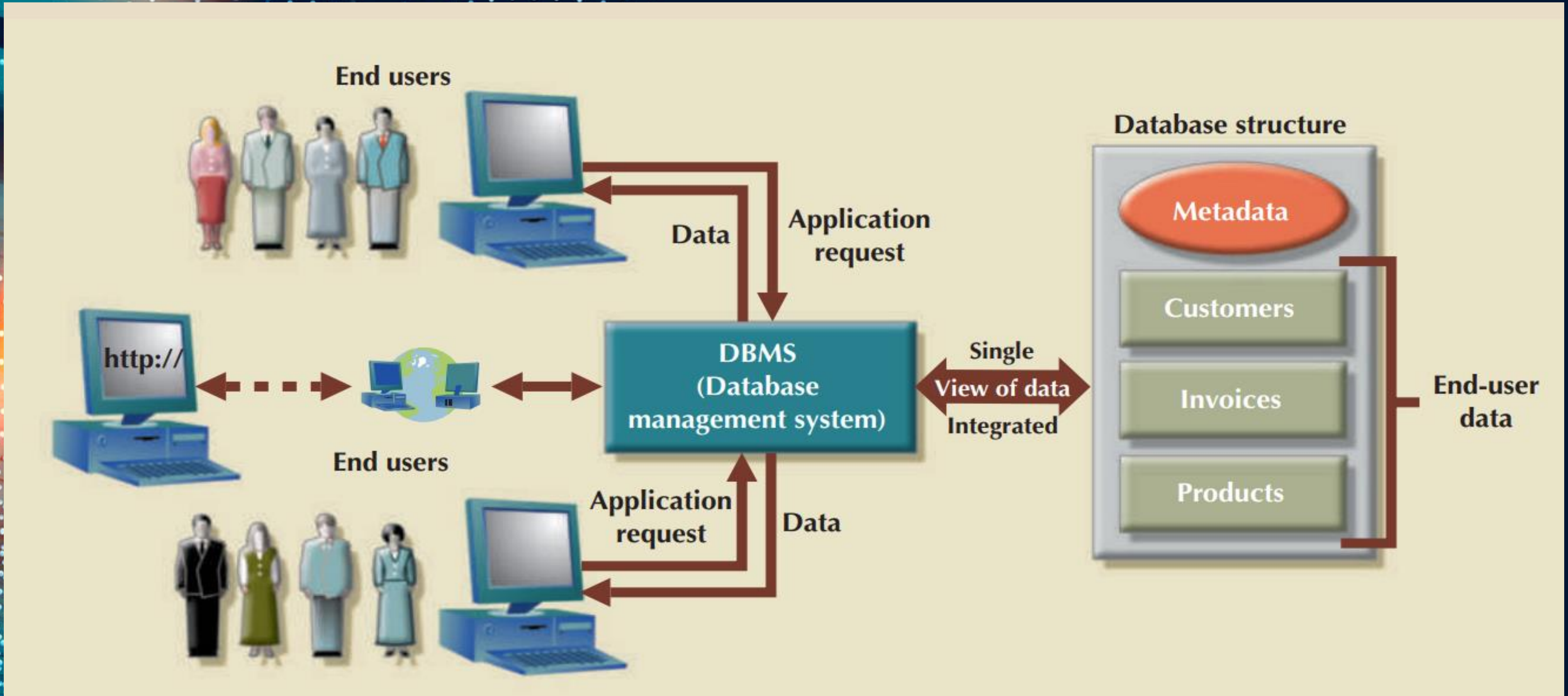
- Minimized data inconsistency. Data inconsistency exists when different versions of the same data appear in different places.

- Improved data access. The DBMS makes it possible to produce quick answers to ad hoc queries. From a database perspective, a query is a specific request issued to the DBMS for data manipulation.

- Improved decision making. Better-managed data and improved data access make it possible to generate better-quality information, on which better decisions are based.

- Increased end-user productivity. The availability of data, combined with the tools that transform data into usable information, empowers end users to make quick, informed decisions that can make the difference between success and failure in the global economy.

1.1.3 Relate the importance of databases to everyday life



DBMS enables the data in the database to be shared among multiple applications or users. Second, the DBMS integrates the many different users' views of the data into a single all-encompassing data repository

1.1.4 Three major steps in the database development process

a. **a. Requirement analysis**

b. Stage in the design cycle when you find out everything you can about the data the client needs to store in the database and the conditions under which that data needs to be accessed.

c.

b. Database design

d. The process that yields the description of the database structure and determines the database components. The second phase of the database life cycle

c. Implementation

e. Process of installing database software, performing configuration and customization, running and testing the database and then integrating it with applications.

1.1.5 Describe the sharing concept of data in database

Sharing of data and multiuser system

Allow many users to access the same database at the same time. This access is achieved through features called **concurrency control strategies**. These strategies ensure that the data accessed are always correct and that data integrity is maintained.

1.1.6 Properties of databases

PROPERTIES	DESCRIPTIONS
Completeness	Ensures that users can access the data they want. Note that this includes ad hoc queries, which would not be explicitly given as part of a statement of data requirements.
Integrity	Ensures that data is both consistent data and correct and ensures that users trust the database.
Flexibility	Ensures that a database can evolve (without requiring excessive effort) to satisfy changing user requirements.
Efficiency	Ensures that users do not have too long response times when accessing data.
Usability	Ensures that data can be accessed and manipulated in ways which match user requirements.

1.2 Identify DBMS

Database Management System (DBMS):
A software package/ system to facilitate the creation and maintenance of a computerized database.

1.2.2 The purpose of database system

Database management systems were developed to handle the following difficulties of typical file-processing systems supported by conventional operating systems:

- Data redundancy and inconsistency
- Difficulty in accessing data
- Data isolation – multiple files and formats
- Integrity problems
- Atomicity of updates
- Concurrent access by multiple users

1.2.3 Various common DBMS

Additional types of software applications have been used in the past and may be still in use on older, legacy systems at various organizations around the world. Some typical examples of DBMS include:

Oracle,
DB2
Microsoft Access
Microsoft SQL Server
PostgreSQL
MySQL
FileMaker

1.2.4 Traditional approach to information processing

Traditional File System

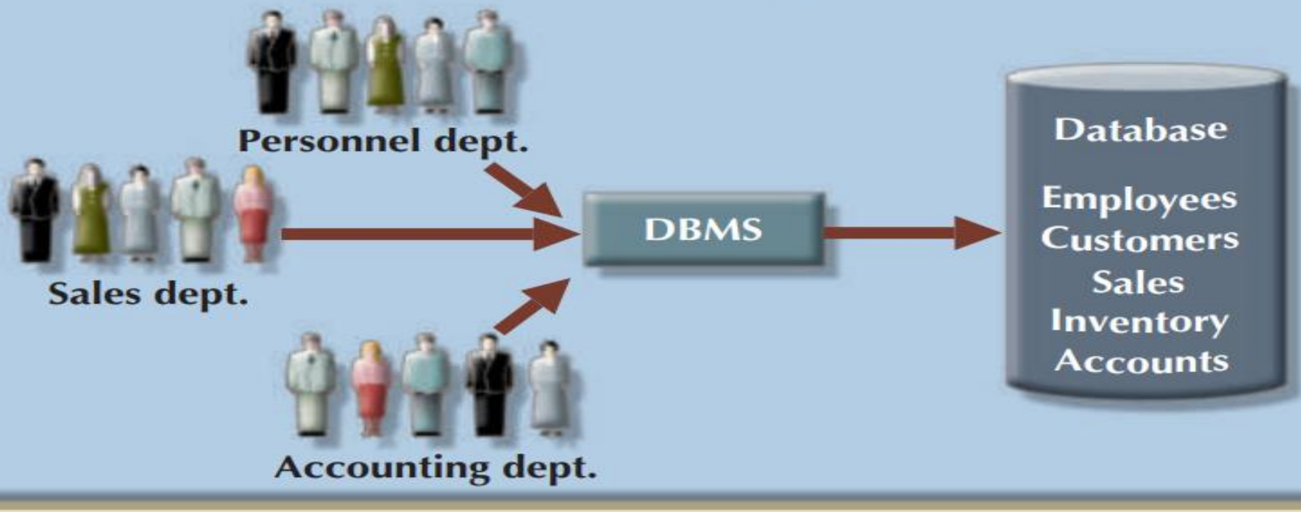
File System is collection of data. In this system, user has to write procedures for managing database. It provides details of data representation and storage of data.

- Data is stored in files.
- Each file has specific format.
- Programs that use these files depend on knowledge about that format.
- In earlier days, database applications were built on top of file systems.

This approach is mostly obsolete but;

- Understanding problems inherent in file based systems may prevent us from repeating these problems in our database system.
- Understanding how file system works is extremely useful when converting a file-based system to a database system.

A Database System



Personnel dept.

Sales dept.

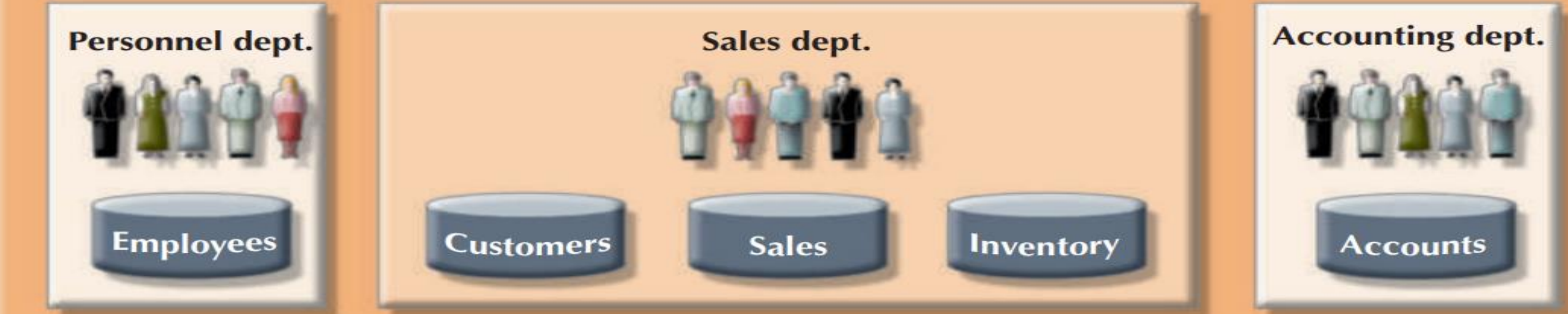
Accounting dept.

DBMS

Database

Employees
Customers
Sales
Inventory
Accounts

A File System



Personnel dept.

Employees

Sales dept.

Customers

Sales

Inventory

Accounting dept.

Accounts

1.2.5 Disadvantages Traditional approach to information processing.

No	Terms	Traditional	Database
1.	Data Security	As the decentralized, it is not enough to have tight security in one department but no security in another department	As the centralized, it is relatively easy to protect the data against all kind of threats whether accidental or intentional.
2.	Data Redundancy	Uncontrolled data redundancy – each application (or department) has its own data. If another application or department needed similar data, the data need to re-entered all over again.	Data duplication/ replication is minimized

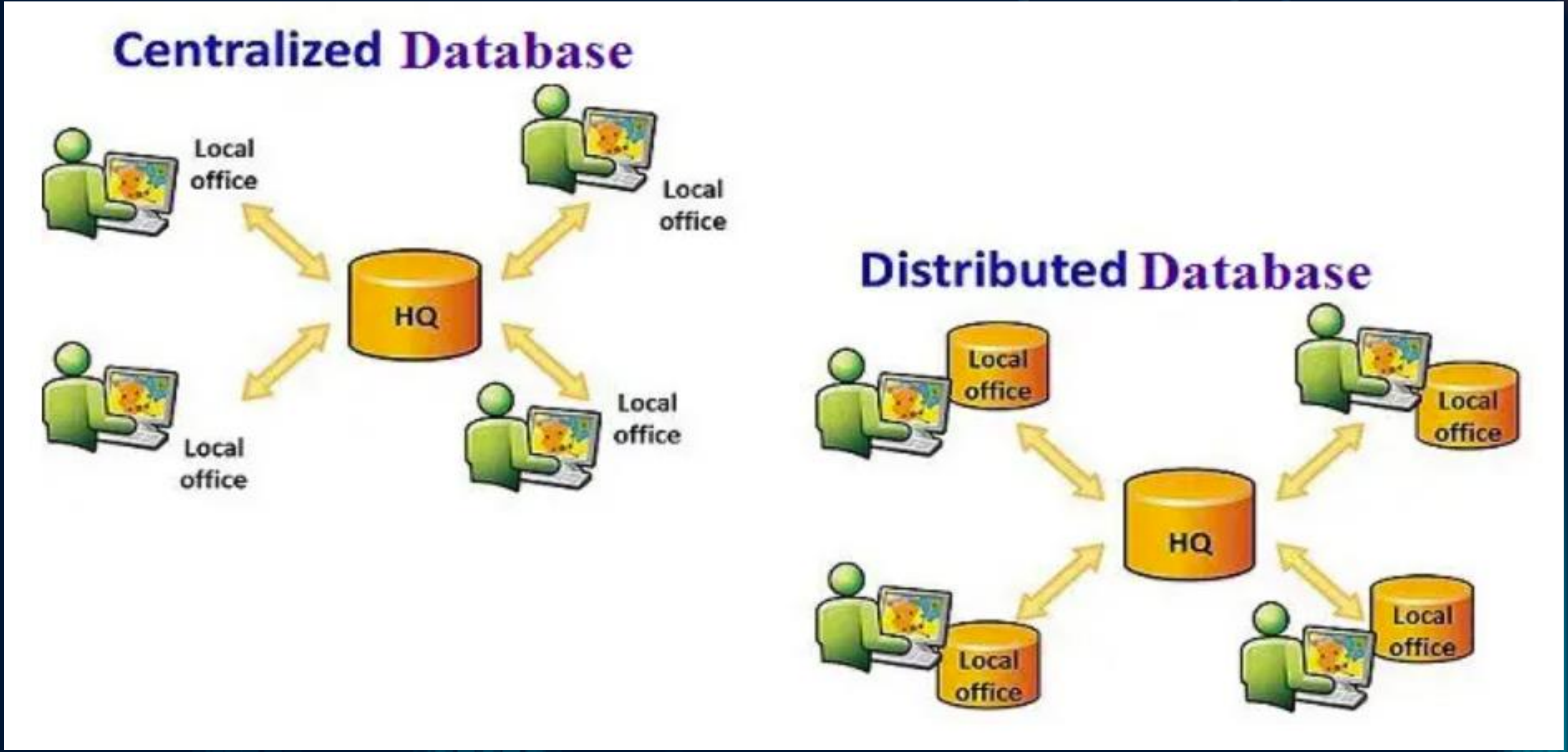
No	Terms	Traditional	Database
3.	Data Isolation	Because data are scattered in various files, and files may be in different formats, writing new application programs to retrieve the appropriate data is difficult.	Although multiple transactions execute concurrently it must appear as if the transaction are running serially (one after the other).
4.	Data Dependence	To change the file structure, need to code, test and document all over again	The code and data descriptions (or file structured) are separated. The data description can change without having to change the code that operates on the data.

No	Differentiation	Traditional	Database
5.	Lack of Flexibility	it cannot deliver ad-hoc reports or respond to unanticipated information requirements in a timely fashion.	With time the cost of data storage reduced making it possible to store data in tables more flexible. This eliminated data redundancy while at the same time provide easier access to data.
6.	Concurrency access anomalies	data may be accessed by many different application programs that have not been coordinated previously.	Overall performance of the system and faster response, many systems allow multiple users to update the data simultaneously.

1.2.6 Importance of DBMS

It organizes a database files and provides end users more access and control over their data. To accomplish this, A DBMS allows users to manipulate the data in their database files, including creating, editing, and updating it when needed.

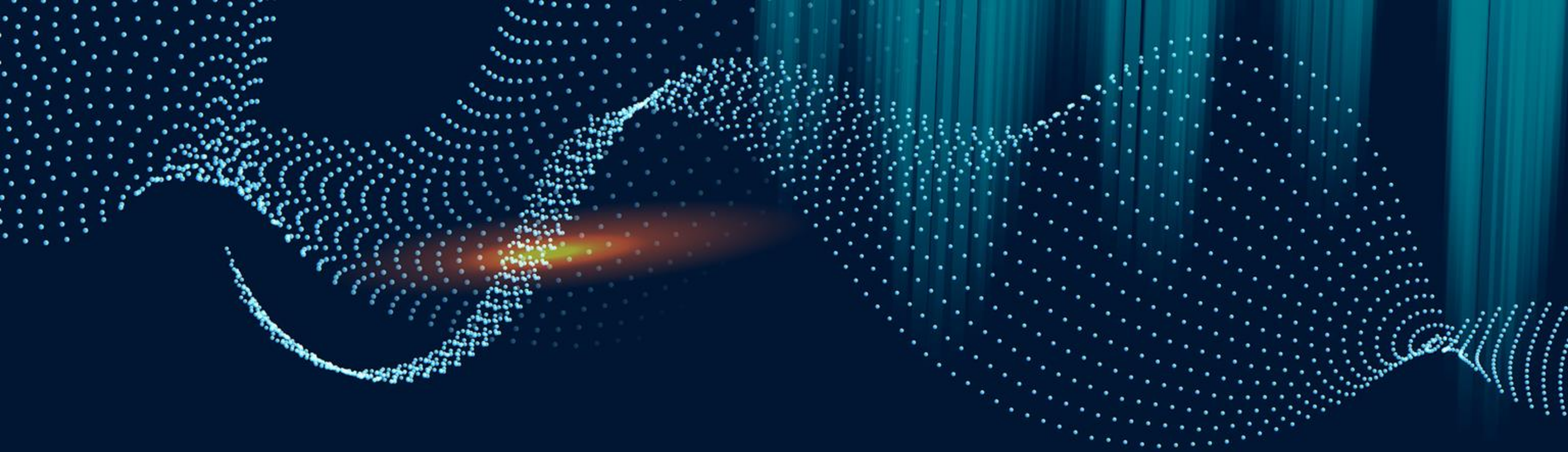
1.2.7 Two generic database architecture: Centralized & Distribute



Distributed databases vs. centralized databases

Here are 5 main differences between distributed databases and centralized database systems.

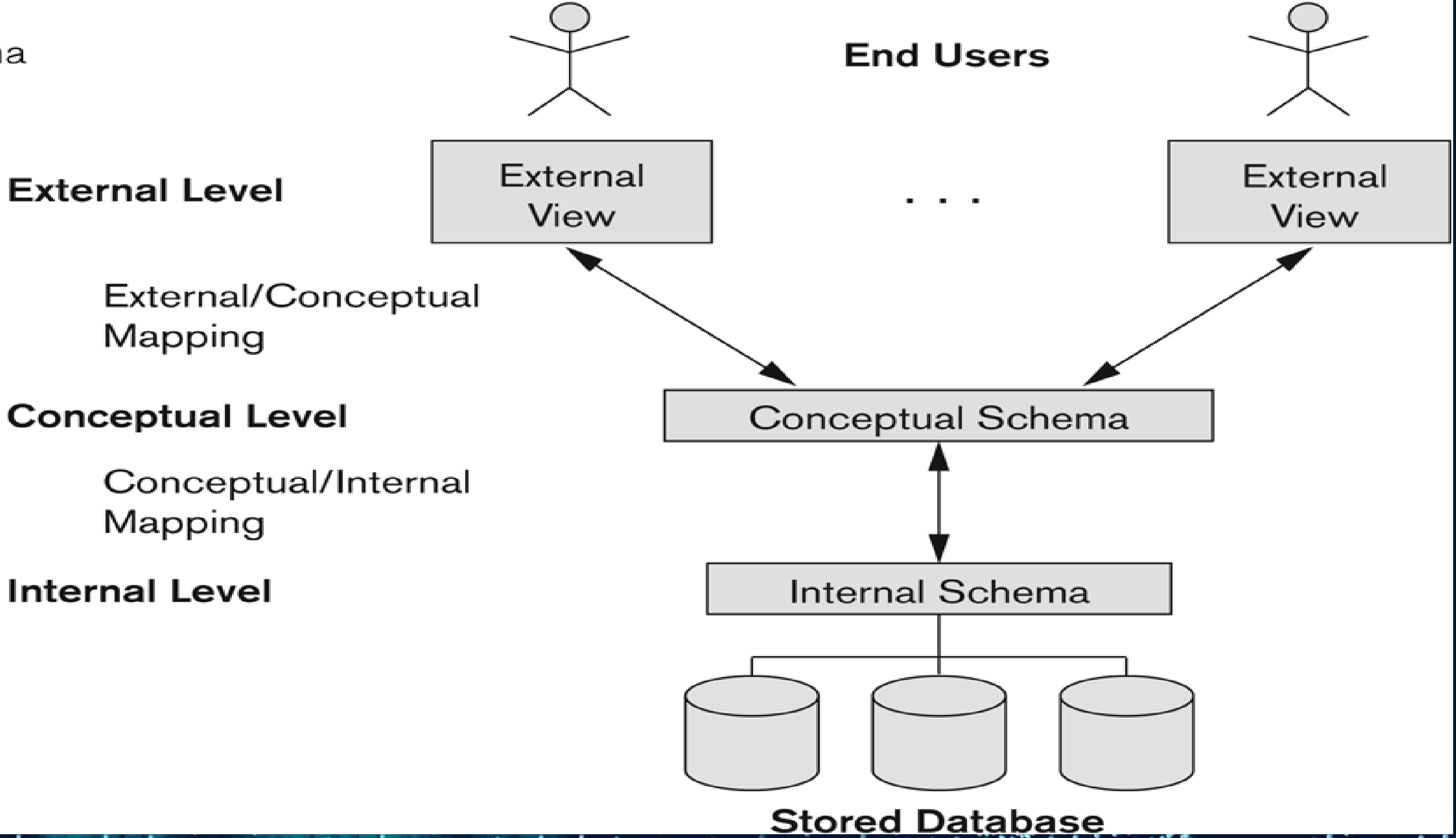
Distributed database	Centralized database
Consists of multiple database files located at different sites	Consists of a single, central database file
Allows multiple users to access and manipulate data	Bottlenecks when multiple users access same file simultaneously
Files delivered quickly from location nearest the user	Files may take longer to deliver to users
If one site fails, data is retrievable	Single site means downtime in cases of system failures
Multiple files from dispersed databases must be synchronized	Simpler to update and manage data in single, central system

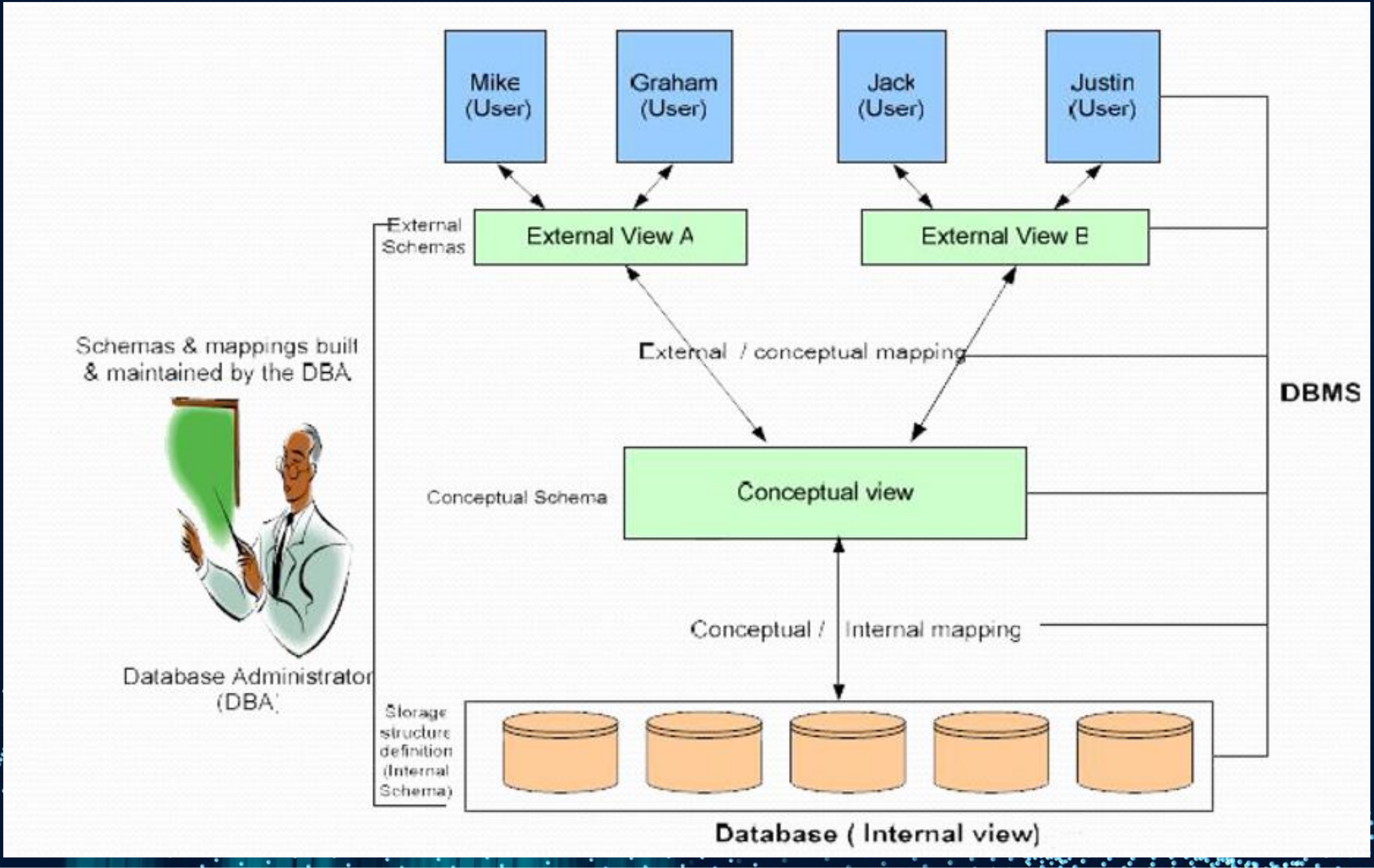


1.3 Identify data Model

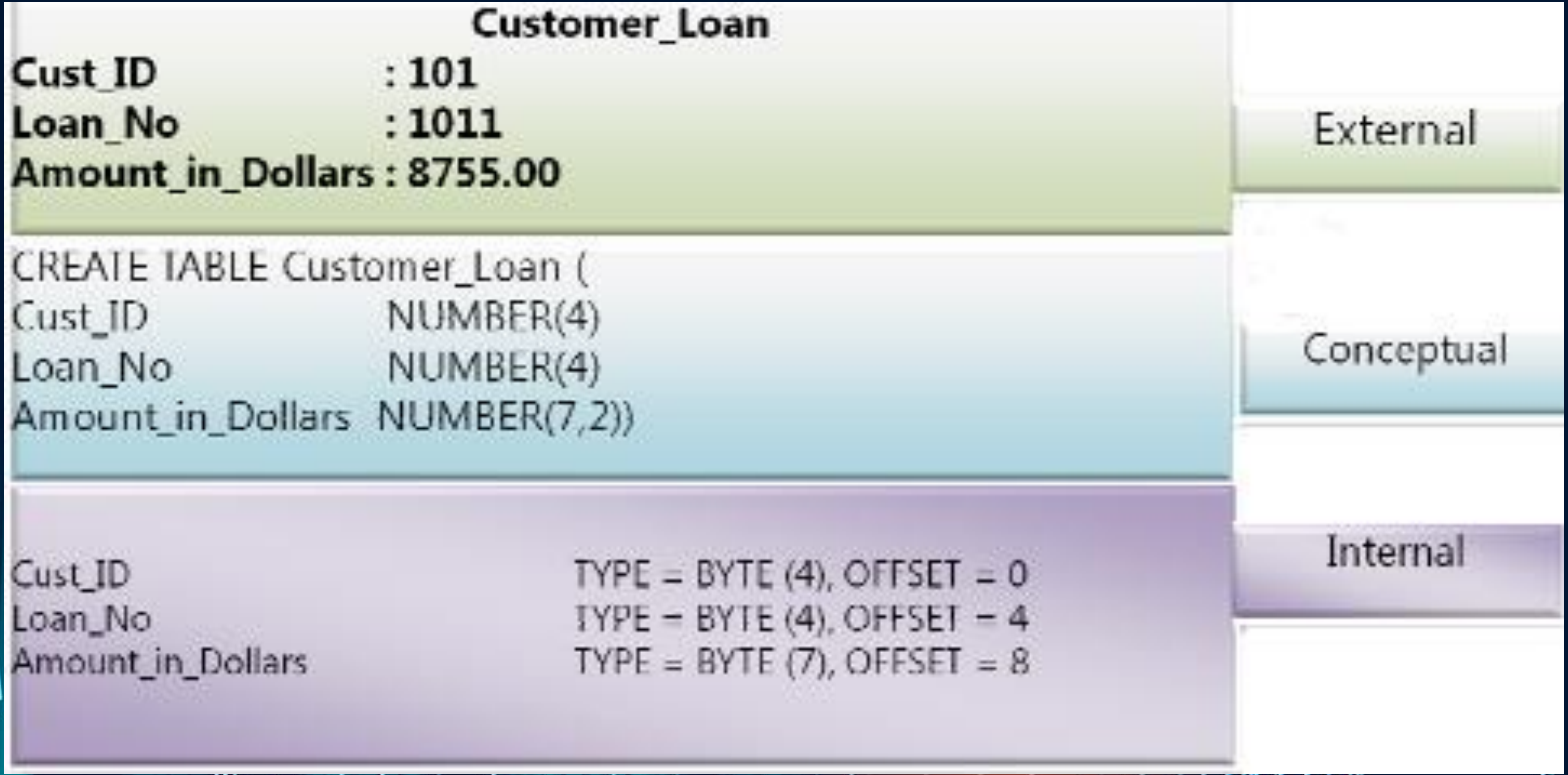
Three-Schema Architecture

Figure 2.2
The three-schema architecture.




Mike (User)
Graham (User)
Jack (User)
Justin (User)
External Schemas
External View A
External View B
Schemas & mappings built & maintained by the DBA
External / conceptual mapping
DBMS
Conceptual Schema
Conceptual view
Conceptual / Internal mapping
Database Administrator (DBA)
Storage structure definition (Internal Schema)
Database (Internal view)

Example of Three-Schema Architecture



Customer_Loan

```
Cust_ID           : 101
Loan_No           : 1011
Amount_in_Dollars : 8755.00
```

```
CREATE TABLE Customer_Loan (
Cust_ID            NUMBER(4)
Loan_No            NUMBER(4)
Amount_in_Dollars  NUMBER(7,2))
```

```
Cust_ID                 TYPE = BYTE (4), OFFSET = 0
Loan_No                 TYPE = BYTE (4), OFFSET = 4
Amount_in_Dollars       TYPE = BYTE (7), OFFSET = 8
```

Three-Schema Architecture

- Mappings among schema levels are needed to transform requests and data.

- Programs refer to an external schema and are mapped by the DBMS to the internal schema for execution.

- Data extracted from the internal DBMS level is reformatted to match the user’s external view (e.g. formatting the results of an SQL query for display in a Web page)

Client-Server Architectures

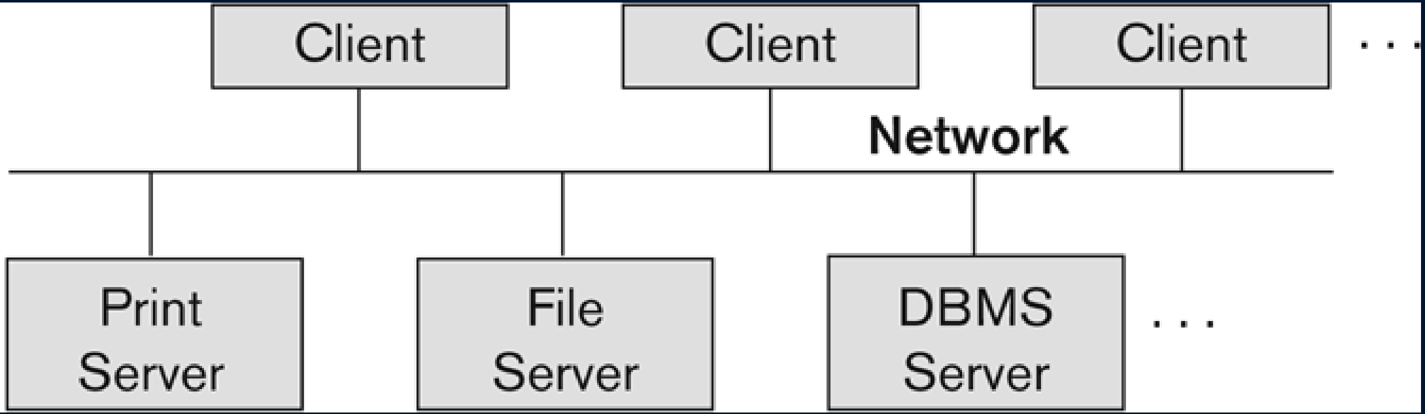
- Client–server architectures provide a flexible way for DBMSs to interact with computer networks.
- The distribution of work among clients and servers and the possible choices to locate data and software are much more complex.
- Two types of Client-Server Architecture
  - Two-tier Client-Server
  - Three-tier Client-Server

Basic 2-tier Client-Server Architectures

- Specialized Servers with Specialized functions
  - Print server
  - File server
  - DBMS server
  - Web server
  - Email server
- Clients can access the specialized servers as needed

Logical two-tier Client Server Architecture

Figure 2.5
Logical two-tier client/server architecture.



Clients

- Provide appropriate interfaces through a client software module to access and utilize the various server resources.
- Clients may be diskless machines or PCs or Workstations with disks with only the client software installed.
- Connected to the servers via some form of a network.

  (LAN: local area network, wireless network, etc.)

DBMS Server

- Provides database query and transaction services to the clients
- Relational DBMS servers are often called SQL servers, query servers, or transaction servers
- Applications running on clients utilize an Application Program Interface (**API**) to access server databases via standard interface such as:
  - ODBC: Open Database Connectivity standard
  - JDBC: for Java programming access
- Client and server must install appropriate client module and server module software for ODBC or JDBC

Two Tier Client-Server Architecture

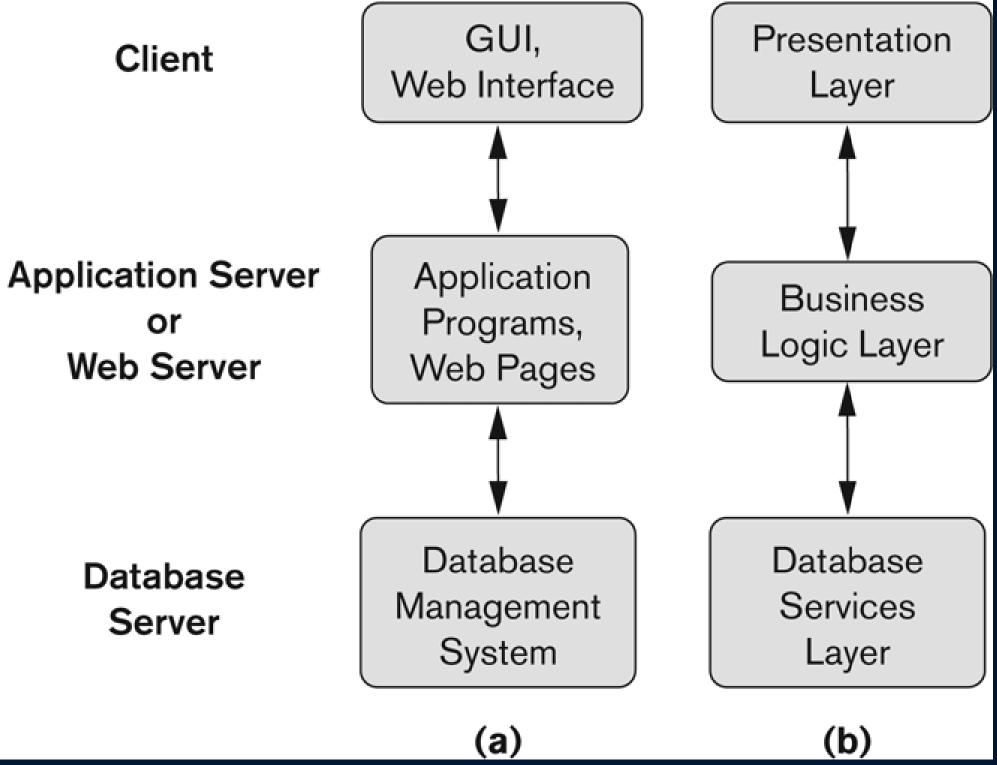
- A client program may connect to several DBMSs, sometimes called the data sources.
- In general, data sources can be files or other non-DBMS software that manages data.
- Other variations of clients are possible: e.g., in some object DBMSs, more functionality is transferred to clients including data dictionary functions, optimization and recovery across multiple servers, etc.

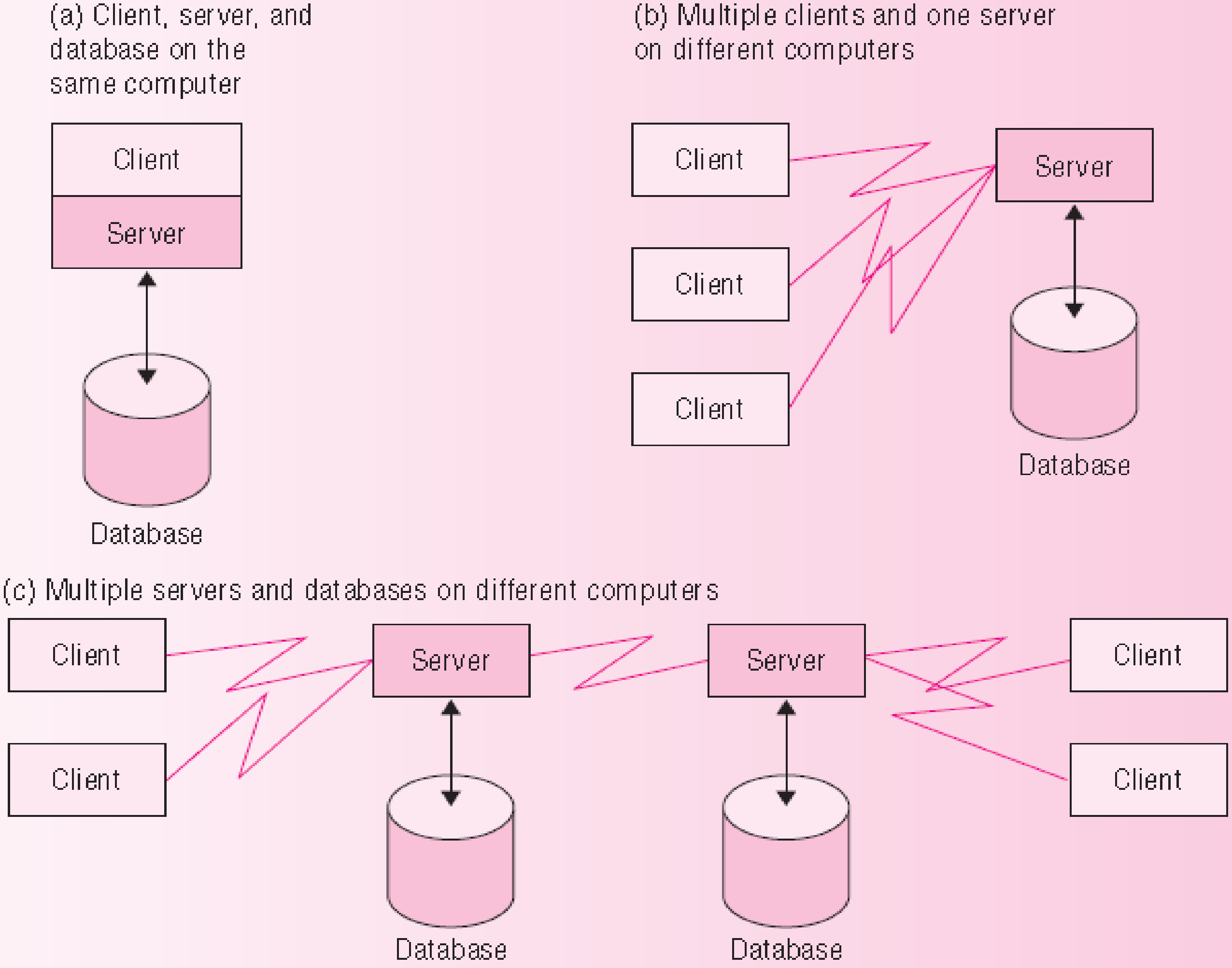
Three Tier Client-Server Architecture

- Common for Web applications
- Intermediate Layer called Application Server or Web Server:
  - Stores the web connectivity software and the business logic part of the application used to access the corresponding data from the database server
  - Acts like a conduit for sending partially processed data between the database server and the client.
- Three-tier Architecture Can Enhance Security:
  - Database server only accessible via middle tier

Three-tier Client-Server Architecture

Figure 2.7
Logical three-tier client/server architecture, with a couple of commonly used nomenclatures.




(a) Client, server, and database on the same computer
Client
Server
Database
(b) Multiple clients and one server on different computers
Client
Client
Client
Server
Database
(c) Multiple servers and databases on different computers
Client
Client
Server
Database
Server
Database
Client
Client

Categories of DBMS

- Desktop Database
- Server Database

Desktop Database

- A Desktop databases offer an inexpensive, simple solution to many less complex data storage and manipulation requirements.
- They earn their name by virtue of the fact that they are designed to run on “desktop” (or personal) computers.
- Example: Microsoft Access, FoxPro, FileMaker Pro, Paradox and Lotus Approach are the major players.

Benefits of Desktop Database

1) Desktop databases are inexpensive.

2) Desktop databases are user-friendly
   A thorough understanding of SQL is not required when using these systems . Desktop DBMSs usually offer an easy-to-navigate graphical user interface.

3) Desktop databases offer web solutions.
   Many modern desktop databases provide web functionality enabling the user to publish their data on the web in a static or dynamic fashion.

Server Database

- Server databases offer organizations the ability to manage large amounts of data efficiently and in a manner that enables many users to access and update the data simultaneously
- A server-based database can provide their user with a comprehensive data management solution.
- Example: Microsoft SQL Server, Oracle and IBM DB2

Benefits of Server Database

1) Flexibility

- Server-based databases can handle just about any data management problem user can throw at them.
- Programmer-friendly : Have *Application Programmer Interfaces* (or APIs) that provide for the rapid development of database oriented custom applications.
- The Oracle platform is even available for multiple operating systems, providing Linux junkies with a level playing field when paired off against the Microsoft folks.

Benefits of Server Database

2) Powerful performance.

- Server-based databases are as powerful as you want them to be.
- The major players are able to efficiently utilize just about any reasonable hardware platform that you’re able to construct for them.
- Modern databases can manage multiple high-speed processors, clustered servers, high bandwidth connectivity and fault tolerant storage technology.

Benefits of Server Database

3) Scalability

- This attribute goes hand-in-hand with the previous one.
- If user are willing to provide the necessary hardware resources, server databases are able to gracefully handle a rapidly expanding amount of users and/or data.

Examples of Database Applications

- Purchases from the supermarket
- Purchases using your credit card
- Booking a holiday at the travel agents
- Using the local library
- Taking out insurance
- Renting a video
- Using the Internet
- Studying at university