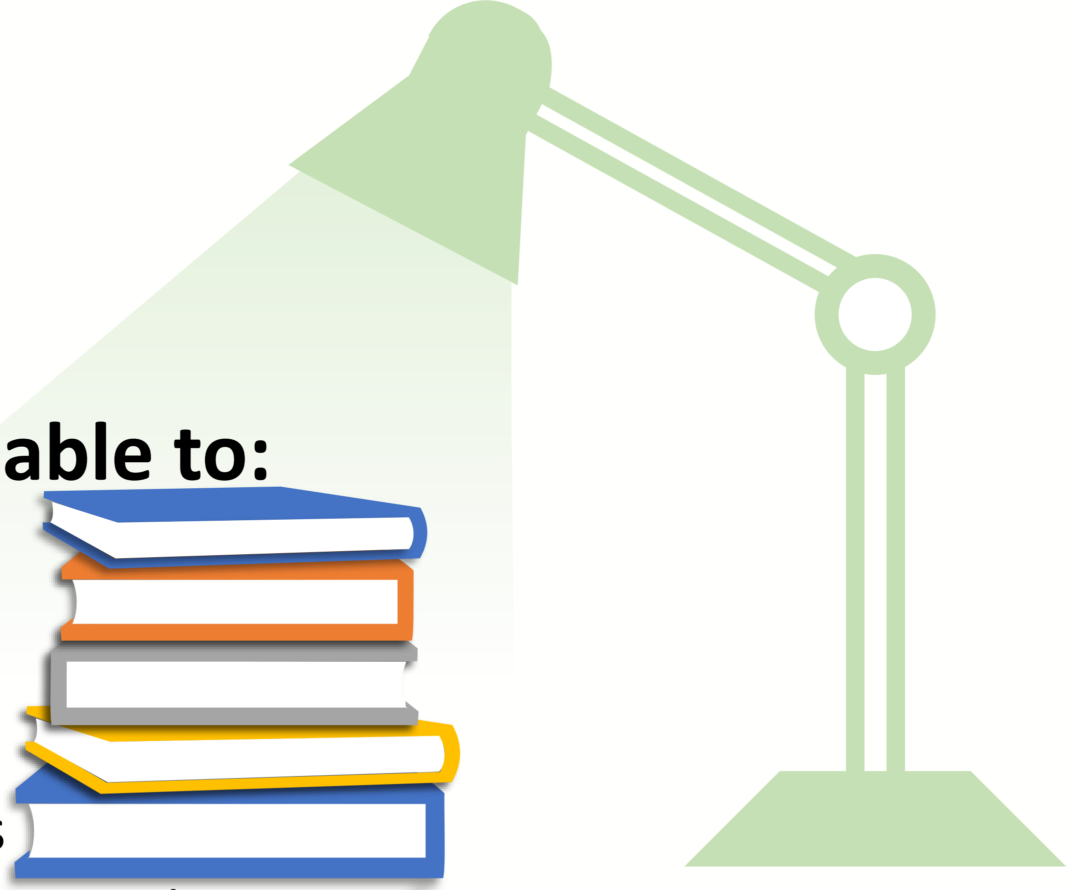
Gs. Dr. Syaifulnizam Abd Manaf
C1-29 Block C, FSKTM

Database Principles: CCS3400

Chapter 1: Database System

Learning Objectives



By the end of this chapter, you will be able to:

- Explain the role of databases in organizations
- Define core database concepts and terminology
- Compare file processing vs. database approach
- Describe database architectures and environments
- Recognize the evolution and latest trends in database systems

Introduction

Every organization generates massive amounts of data.
This data must be **stored, managed, and accessed**—often by multiple users, from anywhere, at any time.

📊 Data powers activities such as:

- **Data management** – storing and organizing records
- **Data analysis** – turning raw facts into insights

💡 **Real-world examples of database use:**

🎓 *Education*: Student records & transcripts
💰 *Finance*: Banking & accounting transactions
🛒 *E-commerce*: Inventory tracking & online orders
🏥 *Healthcare*: Patient histories & treatment records
🤝 *CRM*: Customer details & interaction history

The Data Explosion

The **volume of data** being generated, stored, and processed is **skyrocketing**.

- As the world goes increasingly **digital** and products become **smarter**, this growth will only accelerate.

But more data doesn't just mean more storage, it means more **opportunities**.

💡 With the right tools, organizations can:

🚀 **Drive business strategies** using data-driven insights

🤝 **Enhance customer experiences** by answering questions instantly

🏆 **Gain competitive advantage** by turning raw data into intelligence

📊 According to a **recent IBM study**, one of the **top CEO priorities** is leveraging data and AI insights to stay ahead of the competition.

🌍 Data in Action: Real-World Examples

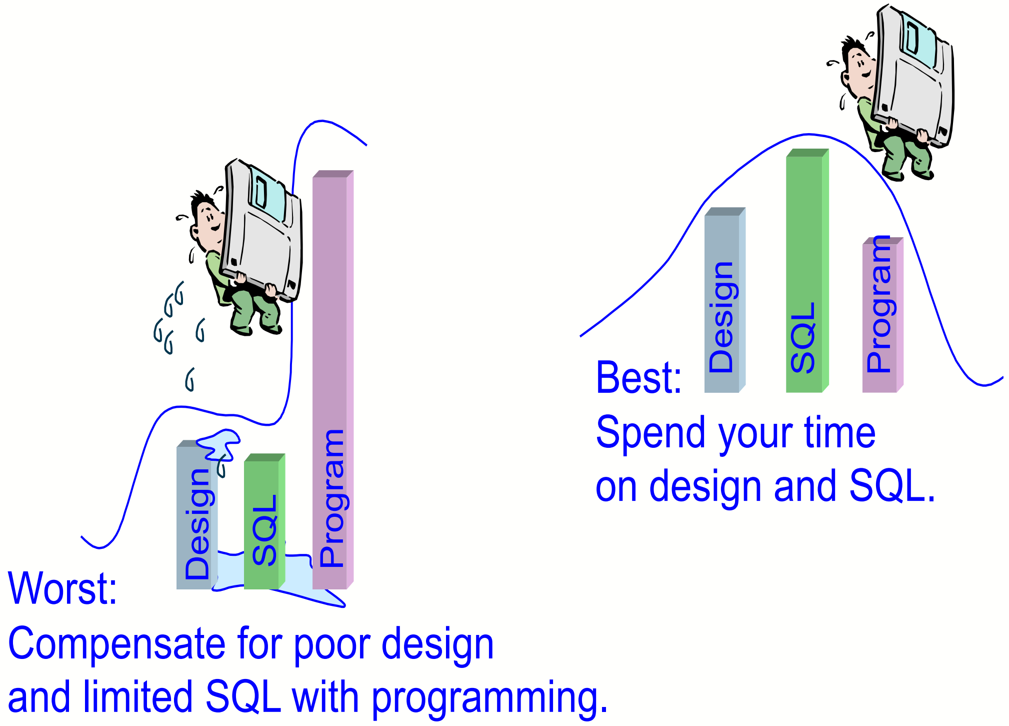
- Some examples from a variety of domains:
  - ✈️ **Airlines**
    - Track flight delays for high-value customers
    - Predict arrival times & suggest gate connections
  - 🍔 **Fast Food Chains**
    - Long lines? Show quick-serve menu items
    - Short lines? Suggest high-margin meals
  - 🛒 **Online Shopping**
    - Recommend products based on buying behavior
    - Personalize promotions to boost sales
- The essence of the above examples is the **ability to collect, organize and manage data.**

Database

- Databases are used to **store, manipulate and retrieve data** in mostly every type of organization: business, health care, education, government and libraries.
- Database technology can be used by an individual up to large number of users accessing databases.
- From personal computers to enterprise-wide distributed applications.
- It can be accessed by users and remote users through diverse technologies such as ATM, Web browsers, smartphones and office environment

Database Management System (DBMS)

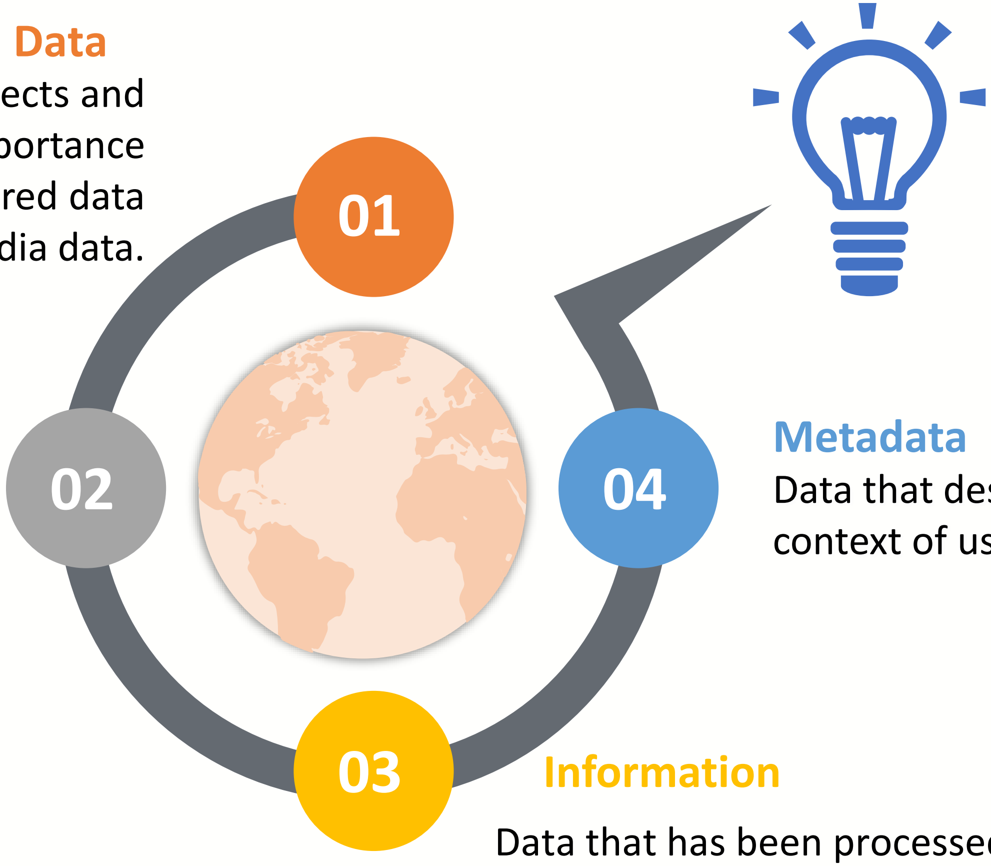
- A **DBMS** is a software system that allows users to **create, store, organize, update, and retrieve data** efficiently and securely.
- It provides a structured environment for building database applications — minimizing redundancy, enforcing integrity, and improving data sharing.
- Understanding the concepts of database design, queries, and application building will reduce the time for developing complex applications.



Basic Concept and Definition

Data

Stored representation of objects and events that have meaning and importance in the user's environment. Structured data or unstructured/multimedia data.



Database

A collection of data stored in a standardized format; designed to be shared by multiple users. It may be of any size and complexity.

Metadata

Data that describes the properties and context of user data.

Information

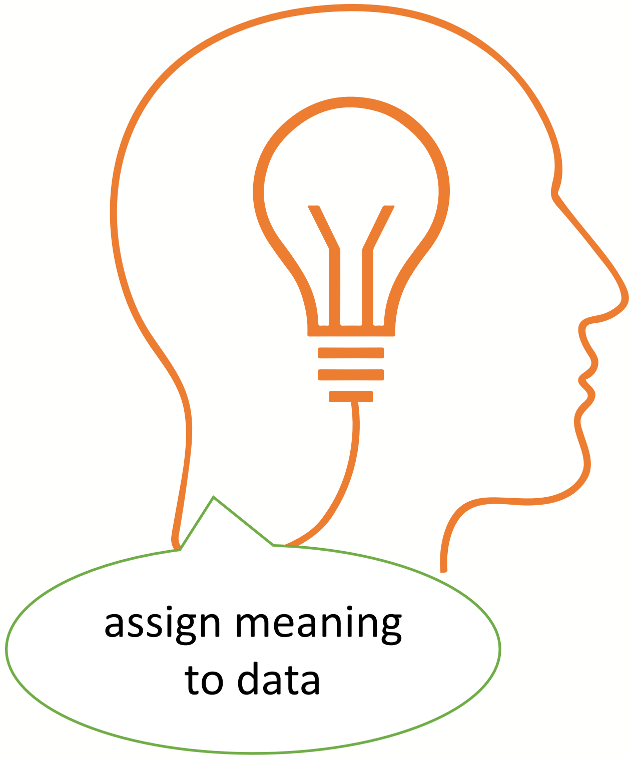
Data that has been processed in such a way as to increase the knowledge of the person who uses the data. Databases today may contain either data or information.

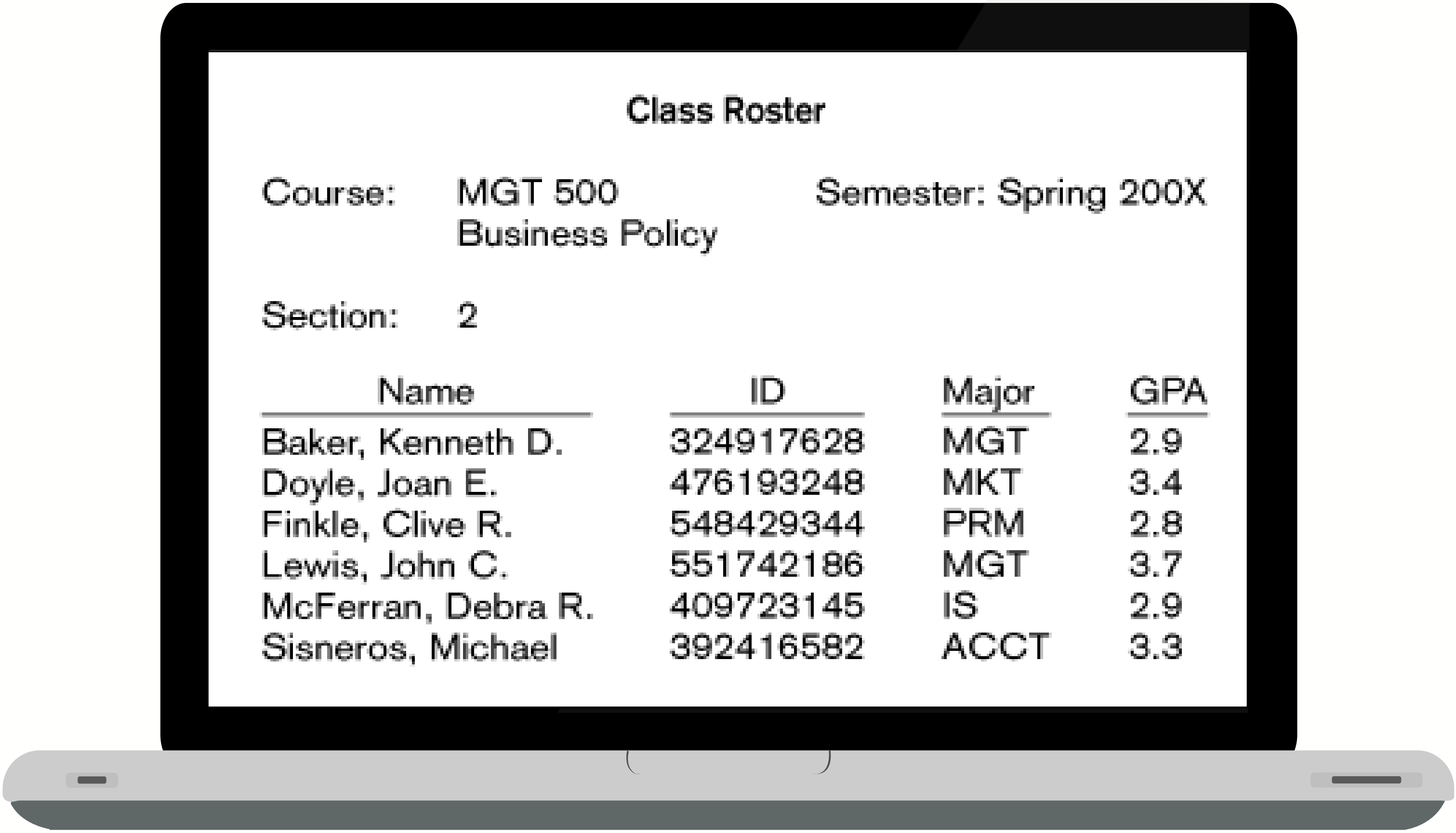
Example of Data

Baker, Kenneth D.	324917628	MGT	2.9
Doyle, Joan E.	476193248	MKT	3.4
Finkle, Clive R.	548429344	PRM	2.8
Lewis, John C.	551742186	MGT	3.7
McFerran, Debra R.	409723145	IS	2.9
Sisneros, Michael	392416582	ACCT	3.3

Converting Data into **Information**

Context helps users understand data





Class Roster

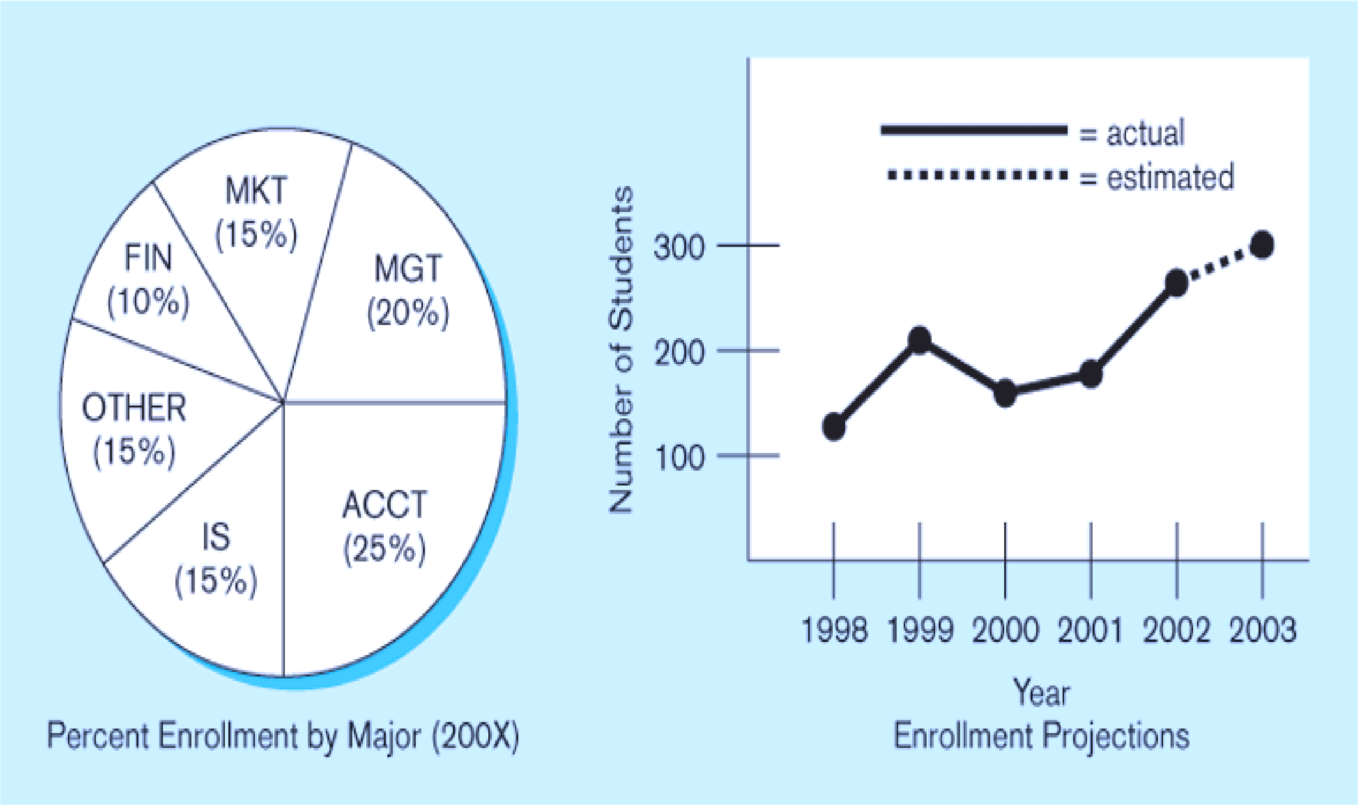
Course: MGT 500 Business Policy

Semester: Spring 200X

Section: 2

Name	ID	Major	GPA
Baker, Kenneth D.	324917628	MGT	2.9
Doyle, Joan E.	476193248	MKT	3.4
Finkle, Clive R.	548429344	PRM	2.8
Lewis, John C.	551742186	MGT	3.7
McFerran, Debra R.	409723145	IS	2.9
Sisneros, Michael	392416582	ACCT	3.3

Converting Data into Graphical Display

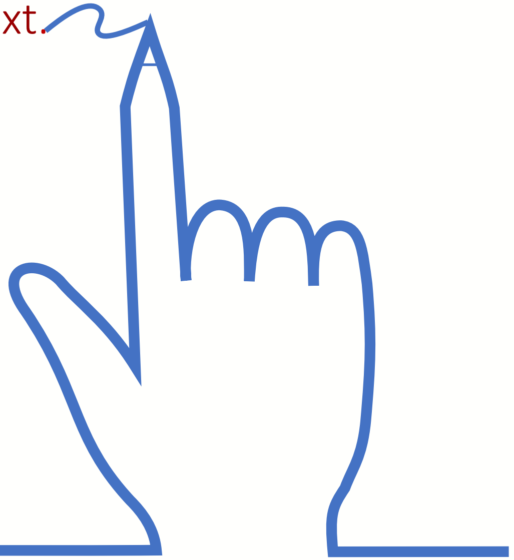


Graphical displays turn data into useful information that managers can use for decision-making and interpretation.

Example of Metadata

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

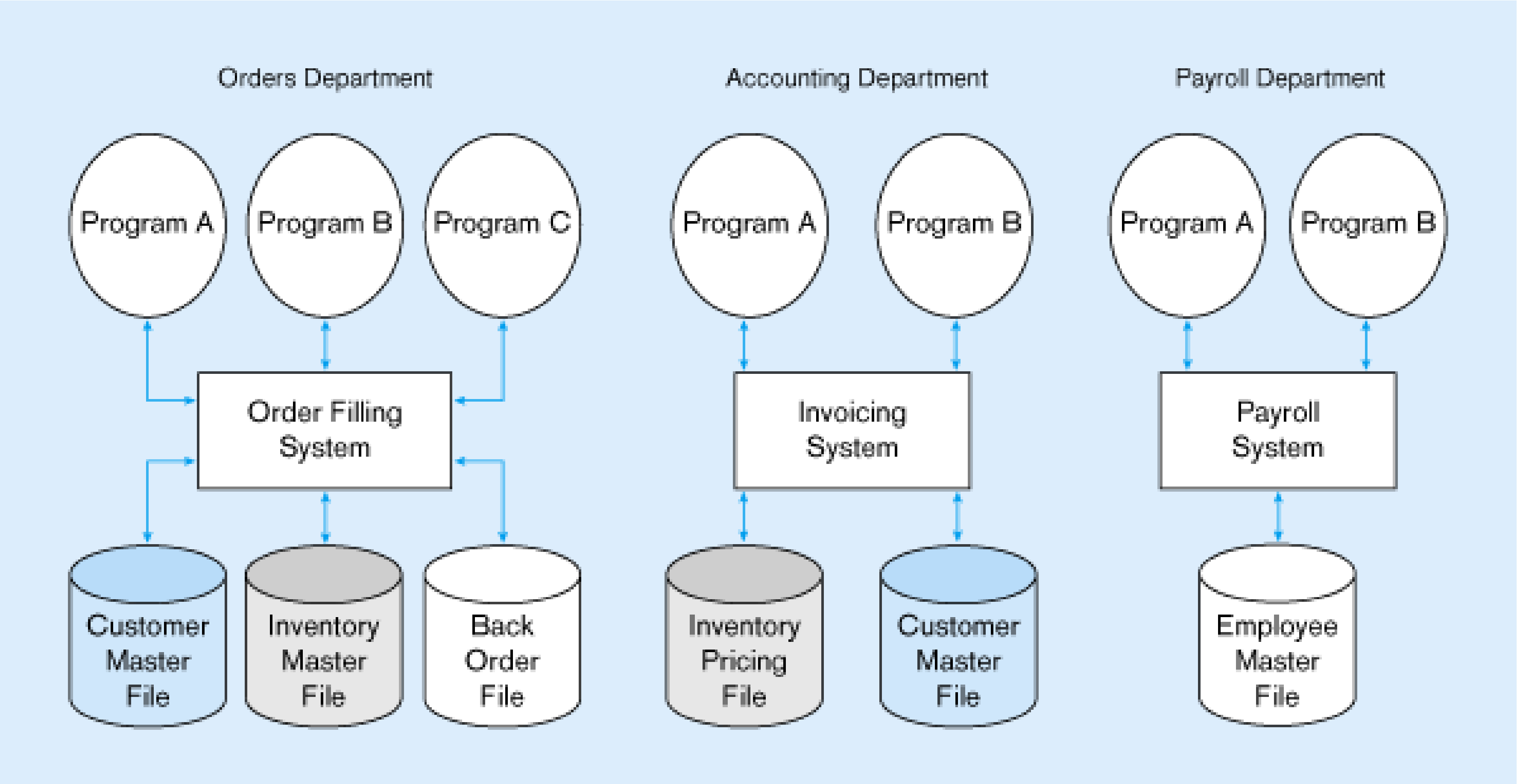
Descriptions of the properties or characteristics of the data, including data types, field sizes, allowable values, and data context.



Traditional File Processing System

- In early computer applications (early 1980s), the traditional file processing approach was commonly used.
- This approach design the data processing to meet individual department rather than the whole organization.
- A file is a collection of related records.
- In file processing system, major data files are associated with each application, and it is typical there is **duplication** of some files used by applications.

Illustration of Traditional File Processing System



Drawbacks of Traditional File Processing System

1. Program-data dependence
2. Duplication of data
3. Inconsistent data
4. Limited data sharing
5. Lengthy development time
6. Excessive program maintenance

Illustration of Program-data Dependence

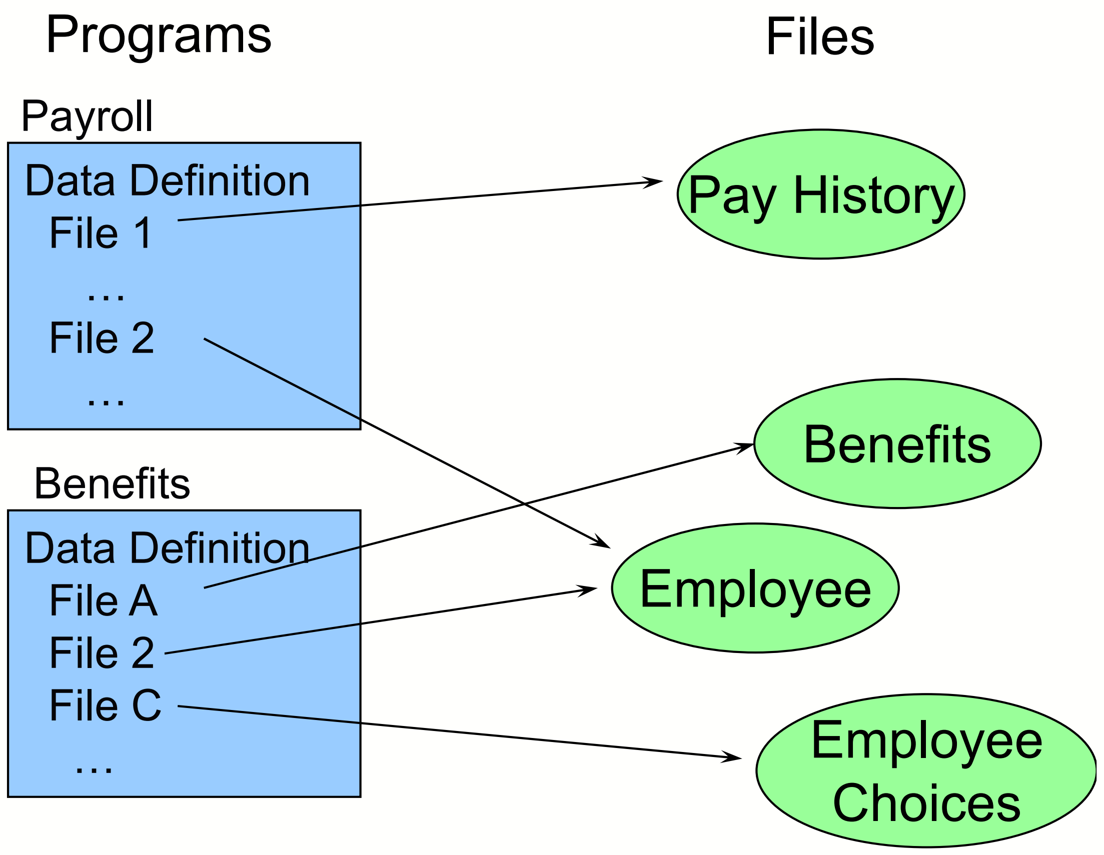


Illustration of Program-data Dependence

COBOL

File Division
01 Employees
  02 ID
  02 Name
  02 Address
  02 Cell Phone
01 Department
  02 ID
  02 . . .

More programs

File Division
01 Employees
...

Employee File

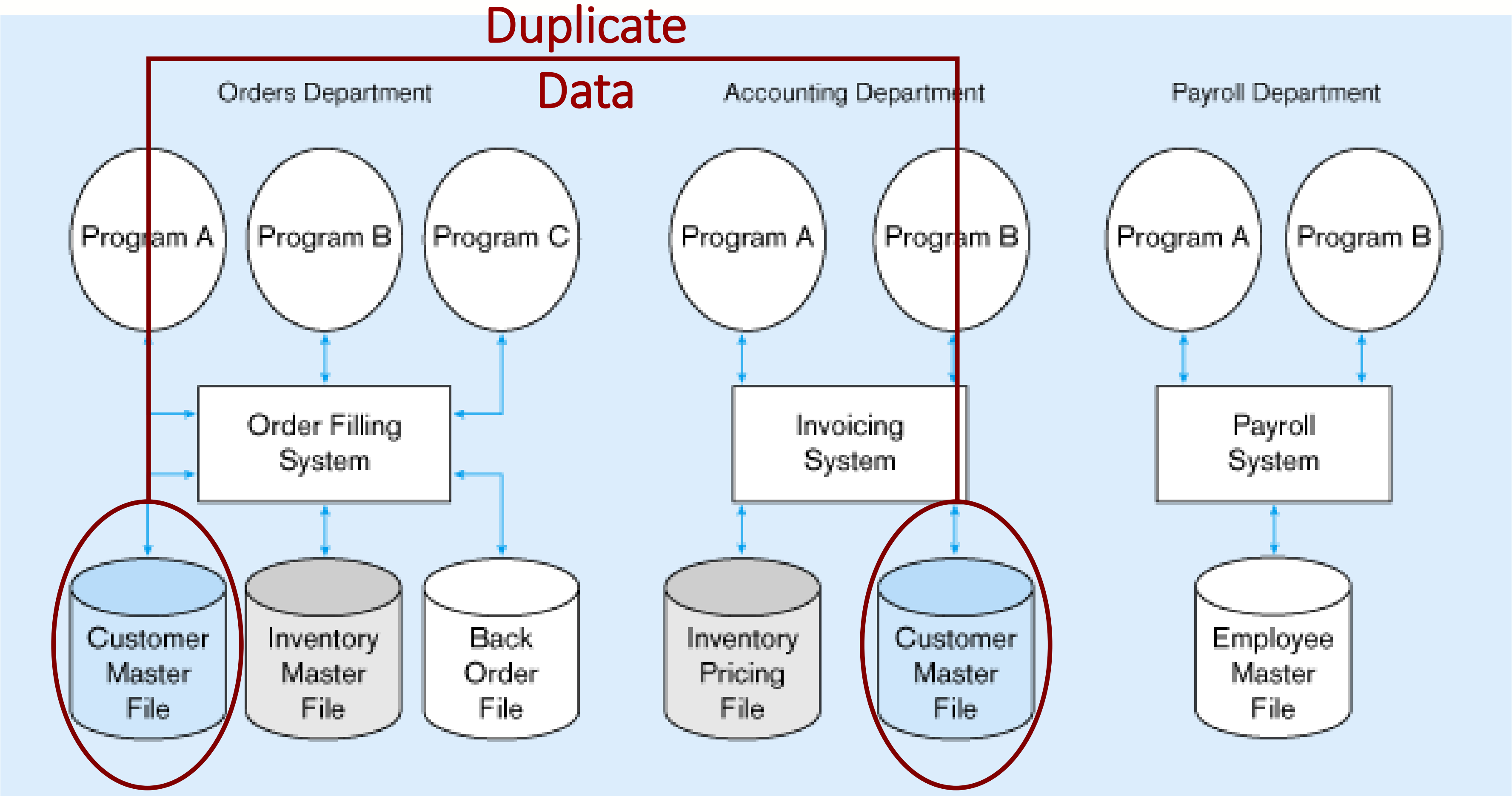
112 Davy Jones 999 Elm Street . . .
113 Peter Smith 101 Oak St . . .

- Add to file (e.g., Cell phone)
  - Write code to copy the employee file and add an empty cell phone slot.
  - Find all programs that use the employee file.
    - Modify file definitions.
    - Modify reports (as needed)
    - Recompile, and fix new bugs.
- Easy to keep to employee files?

Problem with Program-data Dependence

- The file description is stored within each application program that accesses a given file.
- Consequently, any changes to a file structure require modifications to the file description for all programs that access the file.
- It is often challenging to locate all programs affected by such changes, and consequently, errors are often introduced when making such changes.
- Each application program must have its own processing routines for reading, inserting, updating, and deleting data.
- Lack of coordination and central control.

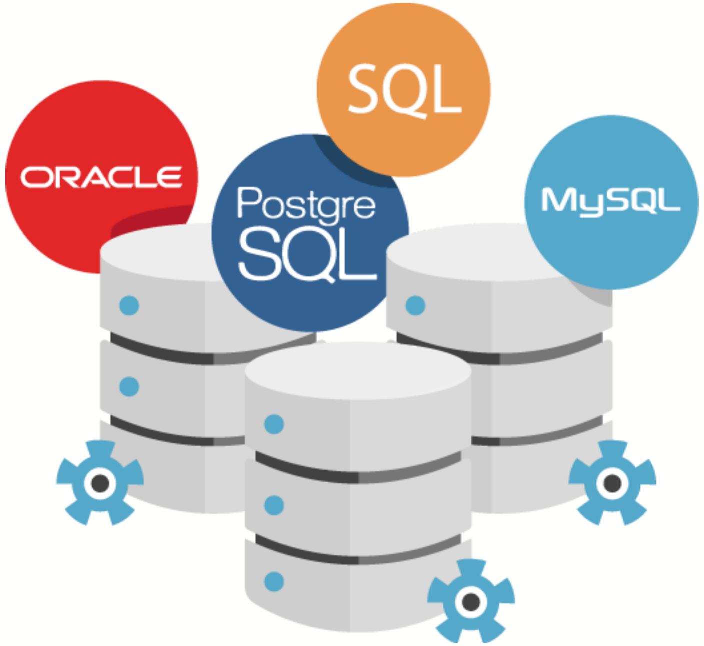
Duplication in File Processing System



Problems with Data Redundancy

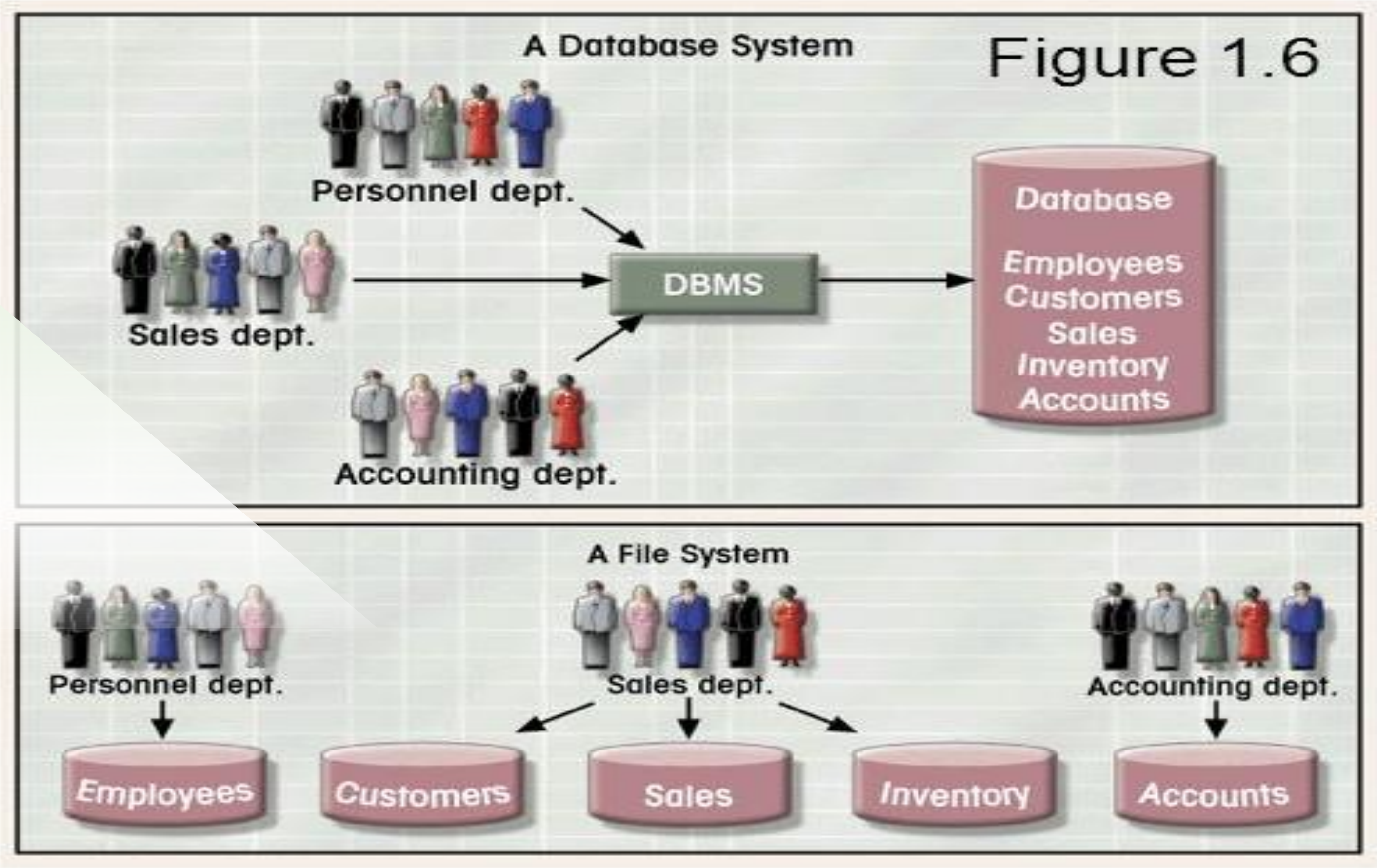
- Waste of space to have duplicate data
- Causes more maintenance headaches
- The biggest problem:
  - When data changes in one file, it could cause inconsistencies
  - Compromises data integrity.

The Database Approach



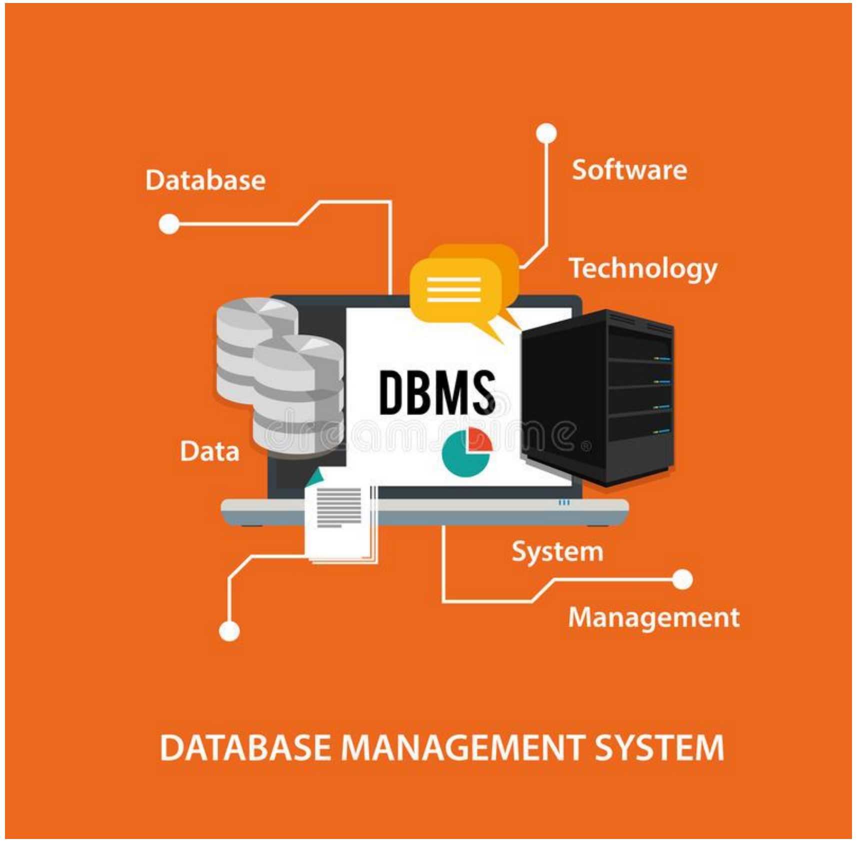
- A central repository of shared data.
- Data is managed by a controlling agent.
- Stored in a standardized, convenient form.
- Therefore, it requires a Database Management System (DBMS)

Illustration Database vs. File Processing System



Database Management System (DBMS)

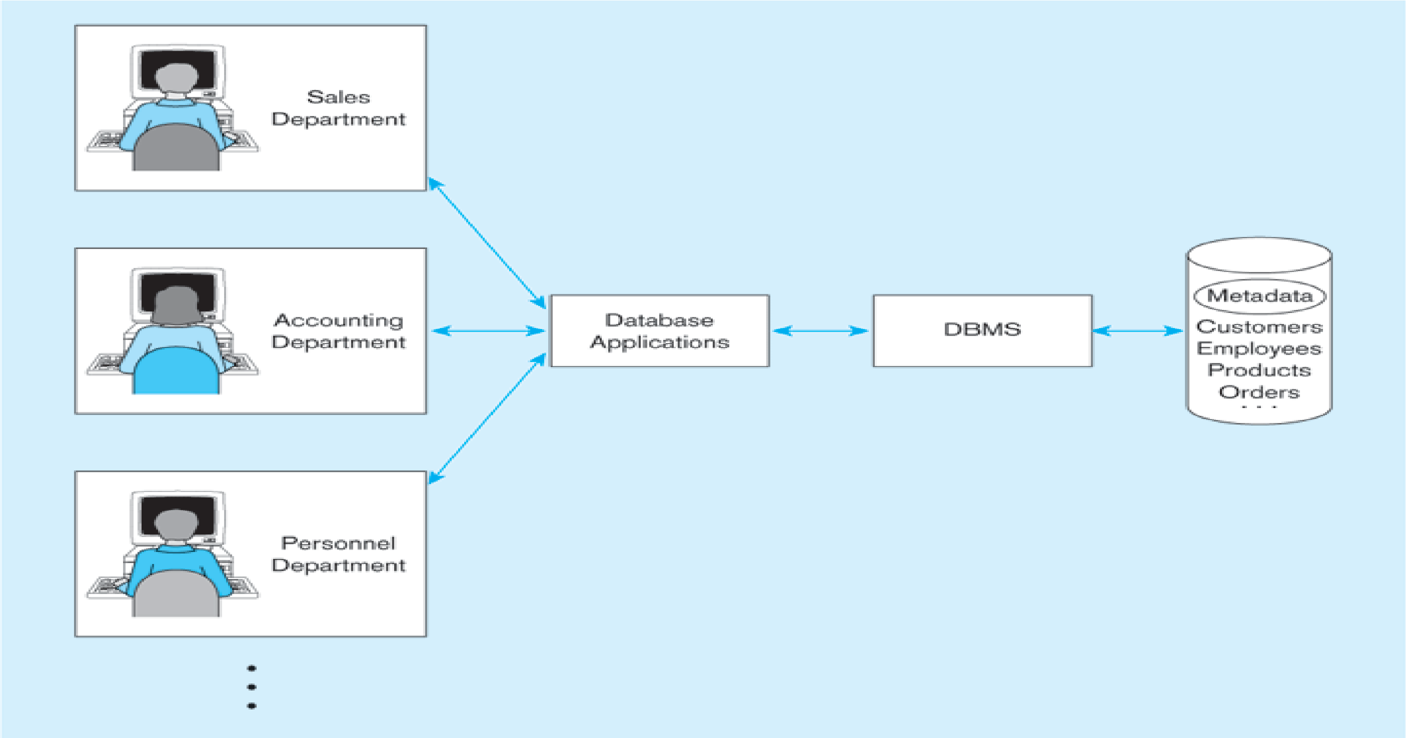
- A **software** system is used to create, maintain, and provide controlled access to user databases.
- It provides a systematic method of **creating, updating, storing, and retrieving data** in a database.
- It enables end users and application programmers to **share data**, and it enables data to be shared among multiple applications rather than propagated and stored in new files for every new application.
- It also provides facilities for **controlling data access, enforcing data integrity, managing concurrency control, and restoring** a database.



Database Management System (DBMS)

- It provides an **interface** between the various database applications for organizational users and the database.

- It allows users to **share** data and query, access, and update the stored data.



Elements of Database Approach

1. Data Model

Graphical model showing high-level entities and relationships for the organization.

2. Relational Database

Database technology involving tables (relations) representing entities and primary/foreign keys representing relationships.

3.Use of Internet Technology

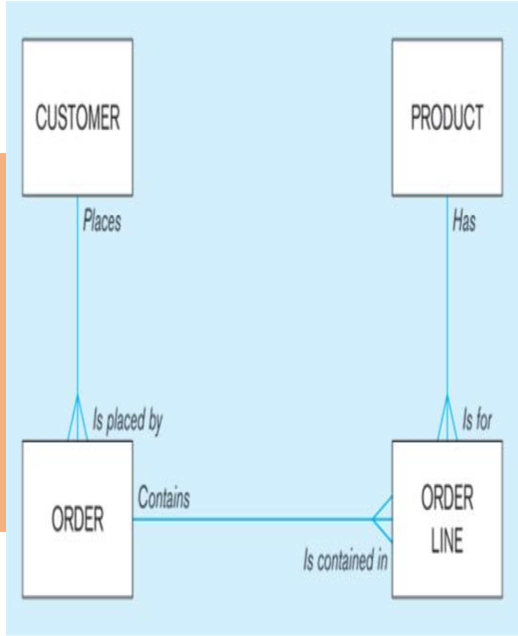
Networks and telecommunications, distributed databases, client-server and 3-tier architectures.

4. Database Application

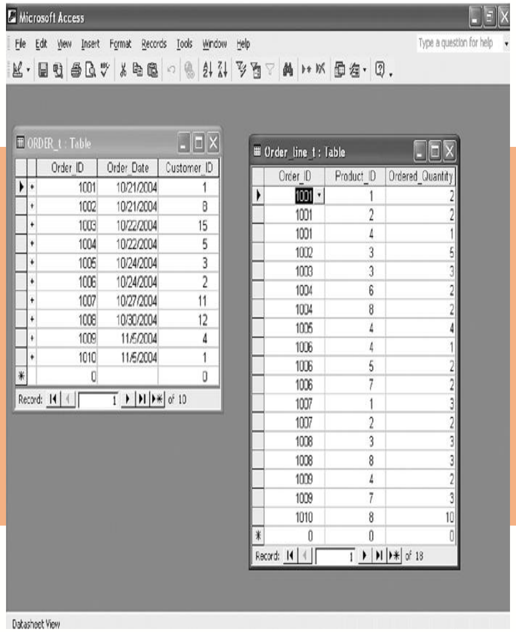
Application programs used to perform database activities (create, read, update, and delete) for database users.

Elements of Database Approach

1. Data Model



2. Relational Database



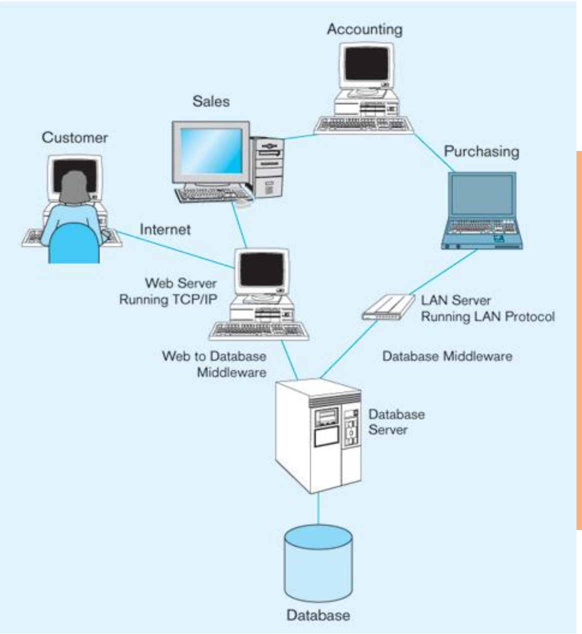
Order ID	Order Date	Customer ID
1001	10/21/2004	1
1002	10/21/2004	8
1003	10/22/2004	15
1004	10/22/2004	5
1005	10/24/2004	3
1006	10/24/2004	2
1007	10/27/2004	11
1008	10/30/2004	12
1009	11/5/2004	4
1010	11/5/2004	1
0		0

Record: 1 of 10

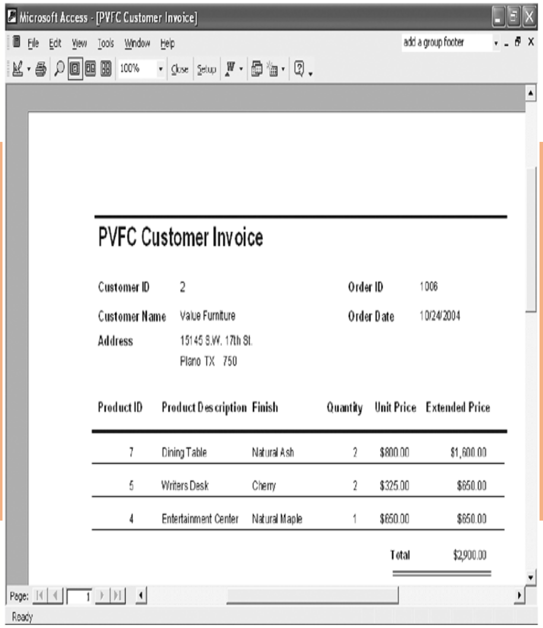
Order_ID	Product_ID	Ordered_Quantity
1001	1	2
1001	2	2
1001	4	1
1002	3	5
1003	3	3
1004	6	2
1004	8	2
1005	4	4
1006	4	1
1006	5	2
1006	7	2
1007	1	3
1007	2	2
1008	3	3
1008	8	3
1009	4	2
1009	7	3
1010	8	10
0	0	0

Record: 1 of 18

3.Use of Internet Technology



4. Database Application



PVFC Customer Invoice

Customer ID 2 Order ID 1006

Customer Name Value Furniture Order Date 10/24/2004

Address 15145 S.W. 17th St.
Plano TX 750

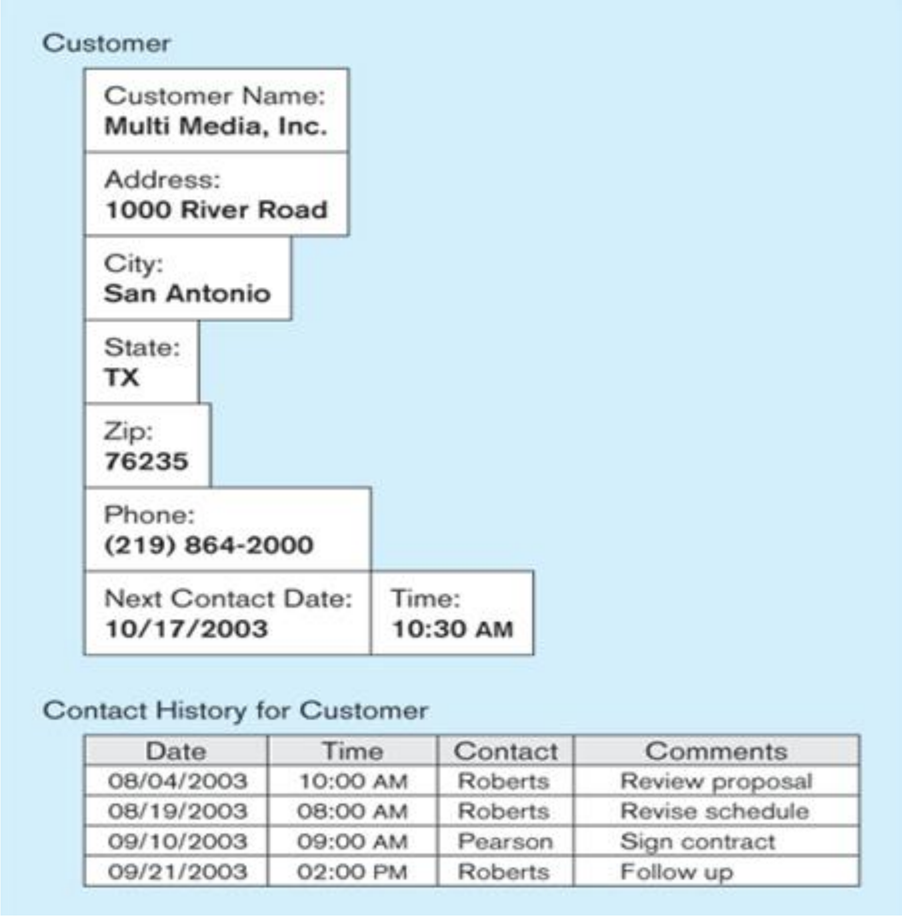
Product ID	Product Description	Finish	Quantity	Unit Price	Extended Price
7	Dining Table	Natural Ash	2	$800.00	$1,600.00
5	Writers Desk	Cherry	2	$325.00	$650.00
4	Entertainment Center	Natural Maple	1	$650.00	$650.00
				Total	$2,900.00

The **Range** of Database Application

- The range of database applications can be divided into three categories based on the client's location (i.e., application) and the database software itself.
- The categories are:
  - Personal Database
  - Two-tier Client-Server Database
  - Multitier Client-Server Database

Personal Database

- It is designed to support **one user**.
- It resided on a personal computer, including laptops, smartphones, and PDAs.
- The purpose of this kind of database is to provide the user with the ability to manage a small amount of data efficiently.


Customer

Customer Name: **Multi Media, Inc.**

Address: **1000 River Road**

City: **San Antonio**

State: **TX**

Zip: **76235**

Phone: **(219) 864-2000**

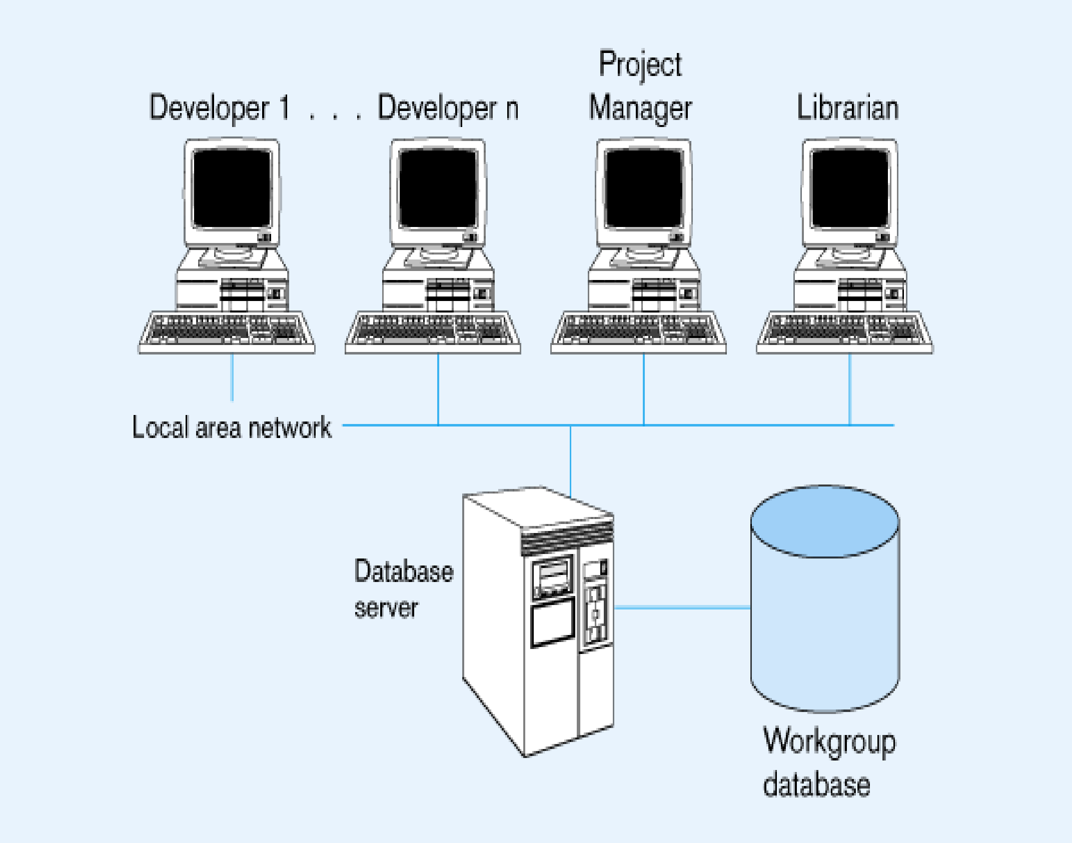
Next Contact Date: **10/17/2003** Time: **10:30 AM**

Contact History for Customer

Date	Time	Contact	Comments
08/04/2003	10:00 AM	Roberts	Review proposal
08/19/2003	08:00 AM	Roberts	Revise schedule
09/10/2003	09:00 AM	Pearson	Sign contract
09/21/2003	02:00 PM	Roberts	Follow up

Two-Tier Client-Server Database

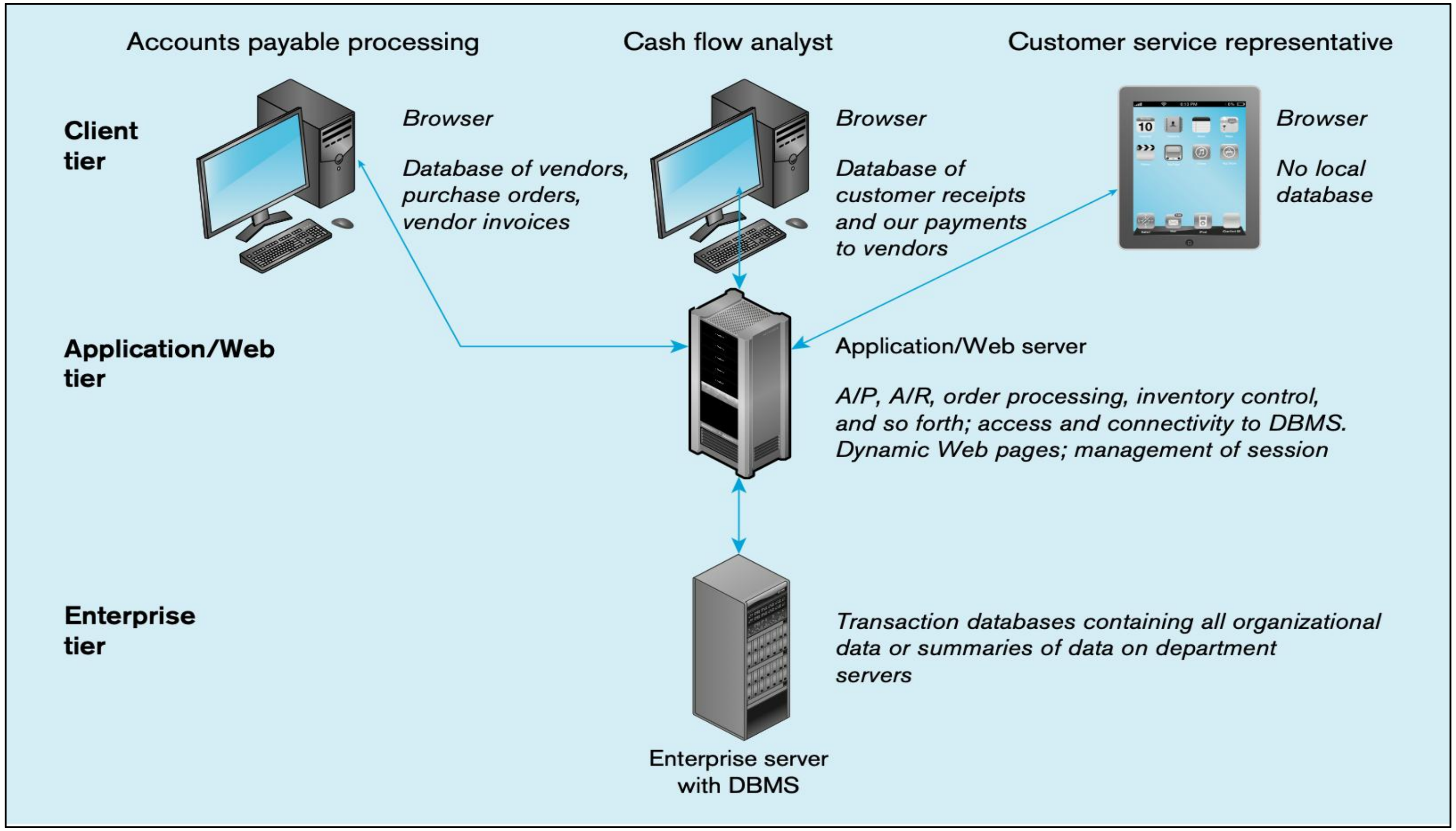
- A relatively small team of people collaborated on the project that required a shared database amongst the group.
- The most common method of sharing data for this type of need is a two-tier client-server application.
- In this method, the client manages the main business and data processing logic and user interface.
- While the server manages and controls access to the database.



Multitier Client-Server Database

- The emergence of the Web changed the roles of client and server, leading to the three-tier architecture.
- Three-tier architecture can enhance security:
  - Database server only accessible via a middle tier
  - Clients cannot directly access the database server
- In three-tier architecture:
  - User interface layer – runs on the client.
  - Business logic and data processing layer – middle tier runs on a server (*application or web server*).
  - DBMS – stores data required by the middle tier. This tier may be on a separate server (*database server*).
- If the bottom layer is split into two layers (a web server and a database server), then it is a 4-tier architecture (possible to the n-tier)

Illustration of Three-tier Client-Server Database

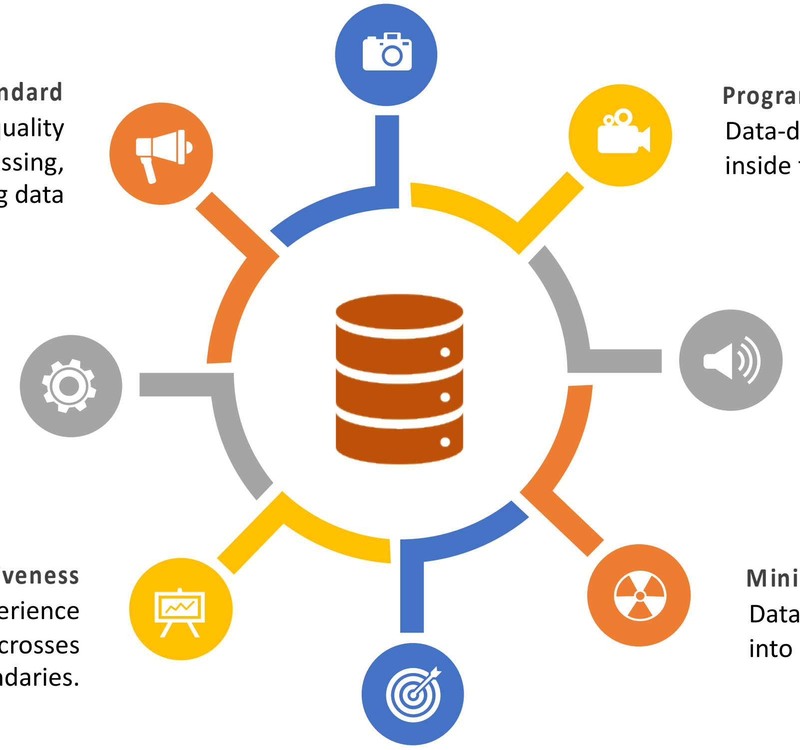


The Advantages of The Database Approach

Increased productivity of application development
New development without worry the file design.
DBMS provides high level productivity tools such as forms and reports generator.

Enforcement of standard
Such as naming convention, data quality standard and uniform procedures for accessing, updating and protecting data

Program-data independence
Data-description are stored in repository not inside the application program



Improved data sharing
A database is designed to shared corporate resource. Authorized internal and external users are granted permission to use the database.

Improved data consistency
By eliminating redundancy, reduce the opportunities for inconsistencies.

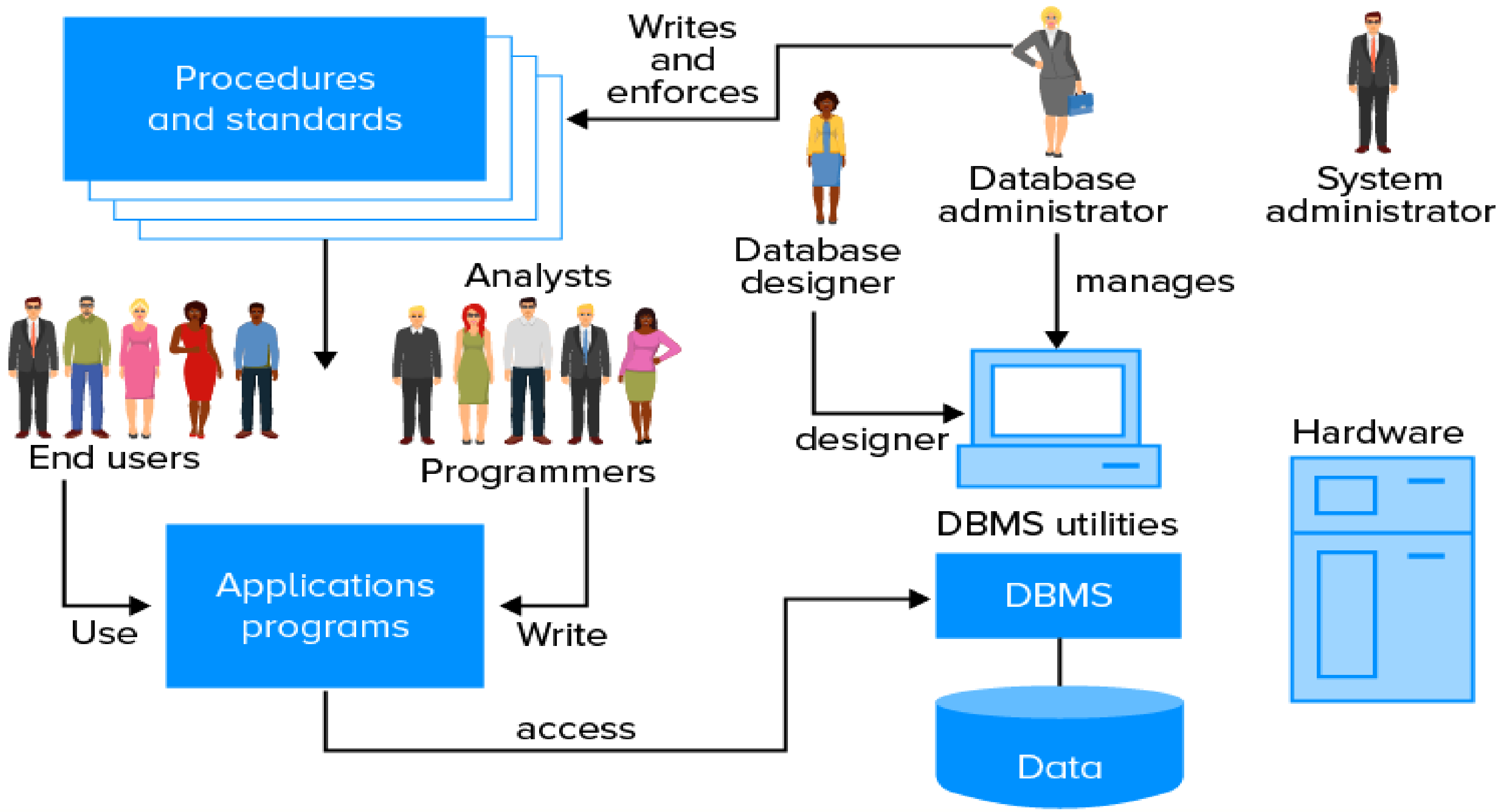
Improved data accessibility and responsiveness
Without the knowledge of programming experience can retrieve and display data, even it crosses departmental boundaries.

Minimal data redundancy
Database design goal to integrate data files into a single, logical structure.

Improved data quality
Through integrity constraint

The Database System Environment

- Hardware
- Software
- People



The Database System Environment

- Data modeling and design tools – automated tools used to design databases and applications programs.
- Repository – centralized storehouse of metadata
- Database Management System (DBMS) – software for managing the database
- Database – a storehouse of the data
- Application Programs – software using the data
- User Interface – languages, menus, and other facilities by which users interact with various system components such as CASE tools, DBMS, etc.
- Data/Database Administrators – personnel responsible for maintaining the database
- System Developers – personnel responsible for designing databases and software
- End Users – people who use the applications and databases.

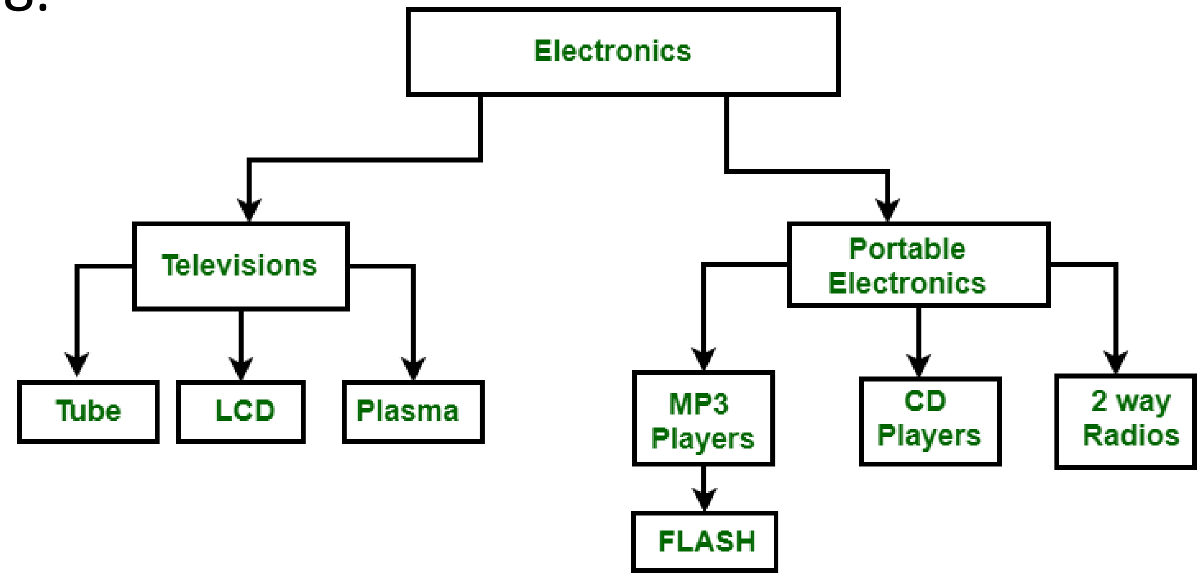
Evolution of Database Systems

- Flat files - 1960s - 1980s
- Hierarchical – 1970s - 1990s
- Network – 1970s - 1990s
- Relational – 1980s - present
- Object-oriented – 1990s - present
- Object-relational – 1990s - present
- Data warehousing – 1980s - present
- Web-enabled – 1990s - present

Hierarchical Database

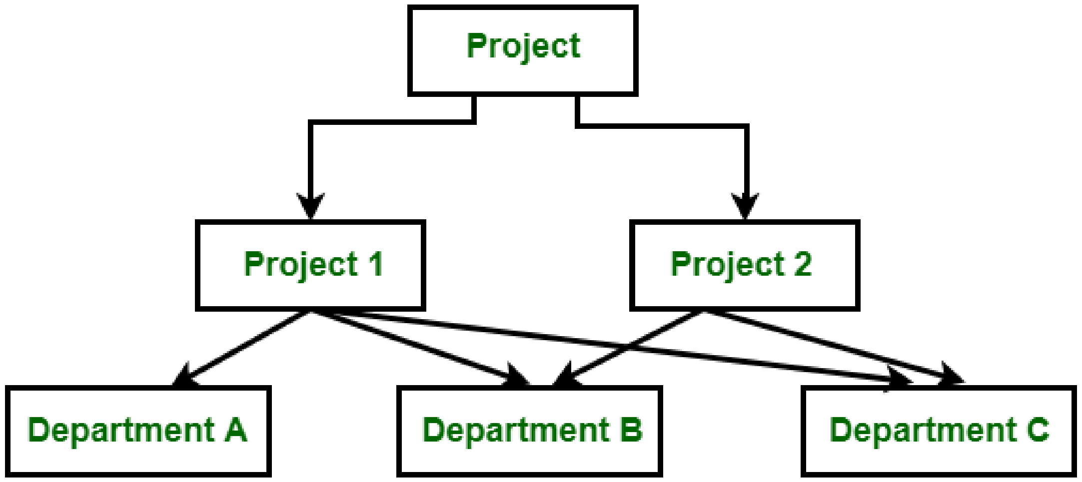
The hierarchical data model is the oldest type of the data model. It was developed by IBM in 1968. It organizes data in a tree-like structure. Hierarchical model consists of the following :

- It contains nodes which are connected by branches.
- The topmost node is called the root node.
- If there are multiple nodes appear at the top level, then these can be called root segments.
- Each node has exactly one parent.
- One parent may have many children.



Network Database

- It is the advance version of the hierarchical data model.
- To organize data it uses directed graphs instead of the tree-structure.
- In this child can have more than one parent. It uses the concept of the two data structures i.e. Records and Sets.



Relational Database

Customer(<u>CustomerID</u>, Name, …

Order(<u>OrderID</u>, <u>CustomerID</u>, OrderDate, …

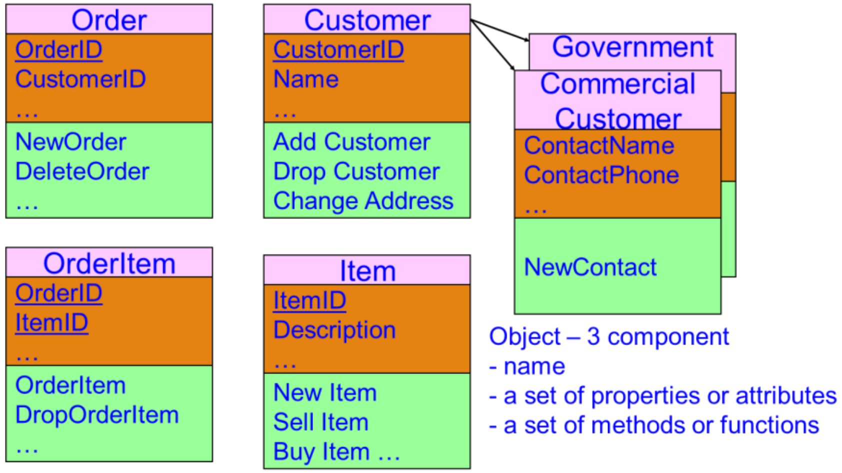
ItemsOrdered(<u>OrderID</u>, <u>ItemID</u>, Quantity, …

Items(<u>ItemID</u>, Description, Price, …

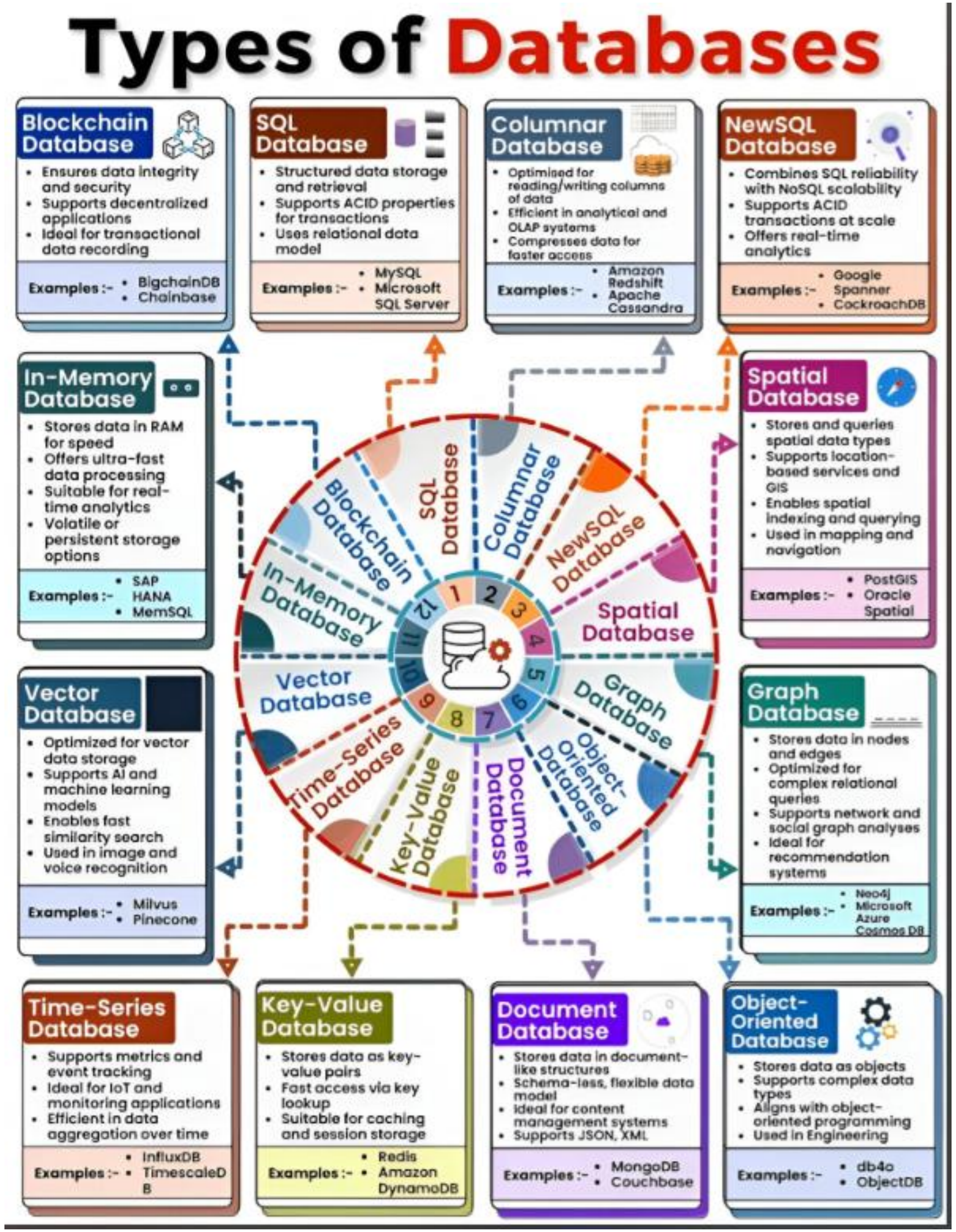
- Data is stored in separate sets of tables/relations
- The table are connected using Primary key and Foreign key
- Eg: to retrieve an order, database can link a customer and an order by <u>customerID</u>

Object-Oriented Database

- An Object-Oriented Database (OODB) is a system that stores data as **objects**, similar to how data is handled in **object-oriented programming**.
- Unlike relational databases that use **tables with rows and columns**, OODBs keep both **data (attributes)** and the **functions (methods)** that work on the data together in one place.

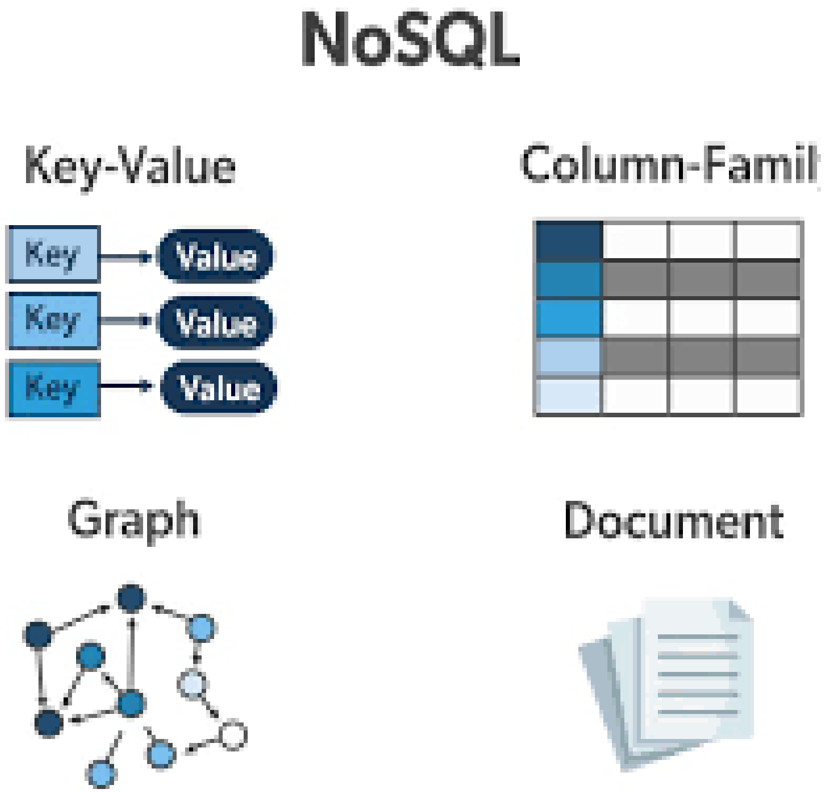


Type of Modern Database



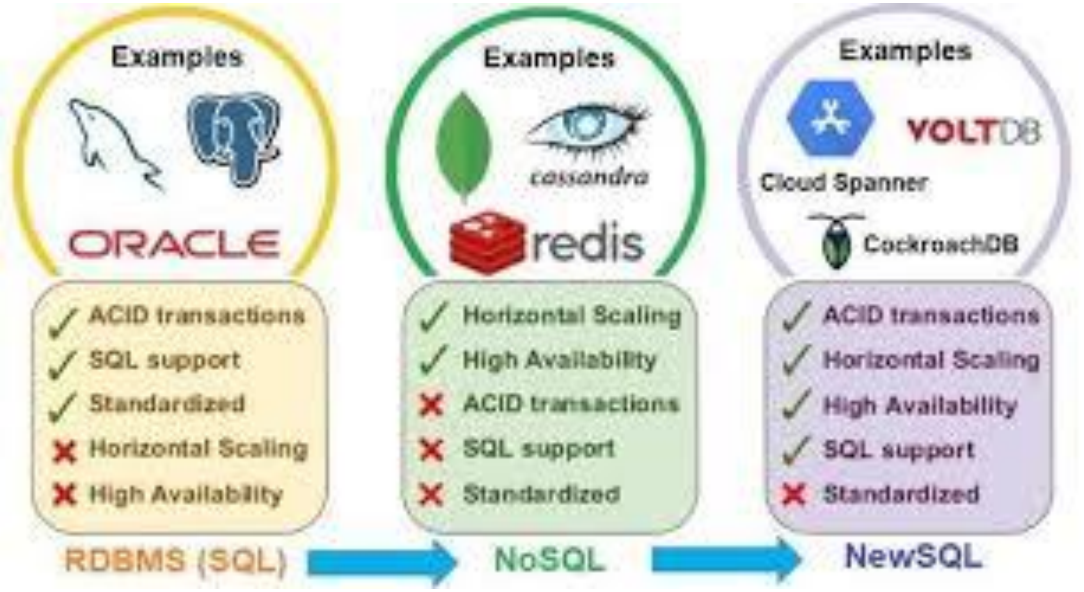
Modern Database: NoSQL Database

- A **NoSQL database** is a type of database designed to handle **large volumes of unstructured or semi-structured data** that don't fit well into traditional tables.
- Unlike relational databases, NoSQL systems are **more flexible**, **scalable**, and can manage **different data formats** such as documents, key-value pairs, columns, or graphs.
- They are commonly used in **big data**, **real-time web applications**, and **cloud environments**, where high speed and scalability are essential.
  Examples include **MongoDB**, **Cassandra**, **Redis**, and **Neo4j**.



Modern Database: NewSQL Database

- **NewSQL databases** are modern database systems that combine the **strengths of relational databases** (such as ACID transactions and SQL support) with the **scalability and performance** of NoSQL systems.
- They are designed to handle **large-scale, high-traffic applications** while maintaining data consistency and reliability.
  In short, NewSQL offers the **best of both worlds** — the structure of SQL with the speed and scalability needed for today's big data and cloud applications.
  Examples include **Google Spanner**, **CockroachDB**, and **VoltDB**.



Comparison Summary

Model	**Era / Year Introduced**	**Data Structure**	**Flexibility**	**Example Systems / Use**
Hierarchical Model	**1960s**	Tree (Parent–Child)	Low	IBM IMS (1968), Windows Registry, early banking systems
Network Model	**Late 1960s – 1970s**	Graph (Set relationships)	Medium	CODASYL DBTG (1971), IDMS, IDS
Relational Model (RDBMS)	**1970s – 1980s**	Tables (Rows & Columns)	High	IBM System R (1974), Oracle (1979), MySQL, PostgreSQL
Object-Oriented Database (OODBMS)	**1980s – 1990s**	Objects (with classes)	High	ObjectDB, db4o, Versant
NoSQL Databases	**2000s – present**	Key-Value, Document, Column, Graph	Very High	MongoDB, Cassandra, Redis, Neo4j
NewSQL Databases	**2010s – present**	Relational + Distributed	Very High	Google Spanner, CockroachDB

Summary

- Databases are essential for storing, managing, and sharing data efficiently across users and applications.
- Traditional file processing systems caused **data redundancy** and **program–data dependence**.
- The **database approach** offers centralized control, data integrity, and easier access.
- A **Database Management System (DBMS)** provides tools for creating, maintaining, and securing databases.
- Database applications range from **personal systems** to **enterprise-wide, multi-tier architectures**.
- Different database models evolved over time:
- **Hierarchical (1960s) → Network (1970s) → Relational (1980s–present) → Object-Oriented (1990s) → NoSQL & NewSQL (2000s–present)**
- Modern databases focus on **scalability, flexibility, and big data management** through **NoSQL** and **NewSQL** systems.
- Understanding database principles is key to building efficient, reliable, and data-driven applications.

End of Chapter

UniPutraMalaysia @uputramalaysia uniputramalaysia universitiputramalaysia

PERTANIAN • INOVASI • KEHIDUPAN

BERILMU BERBAKTI
WITH KNOWLEDGE WE SERVE