CHANDIGARH UNIVERSITY
Discover. Learn. Empower.

APEX INSTITUTE OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Introduction to Data Science (21CST-292)

(Problems with Traditional Databases)

Faculty: Dr. Jitender Kaushal (E14621)

Associate Professor

Introduction to Data Science: Course Objectives

COURSE OBJECTIVES

The Course aims to:

- This course brings together several key big data problems and solutions.
- To recognize the key concepts of Extraction, Transformation and Loading
- To prepare a sample project in Hadoop Environment

COURSE OUTCOMES

On completion of this course, the students shall be able to:-

CO1	Identify and describe the importance of Big data analysis over Conventional Database management System.

Contents

- Describe basic file organization concepts
- Describe how a database management system organizes information and compare the principal database models
- Identify the challenges posed by data resource management and management solutions
- Problems in a traditional file environment
- Comparison of Traditional Database with Big Data

File Organization Terms and Concepts

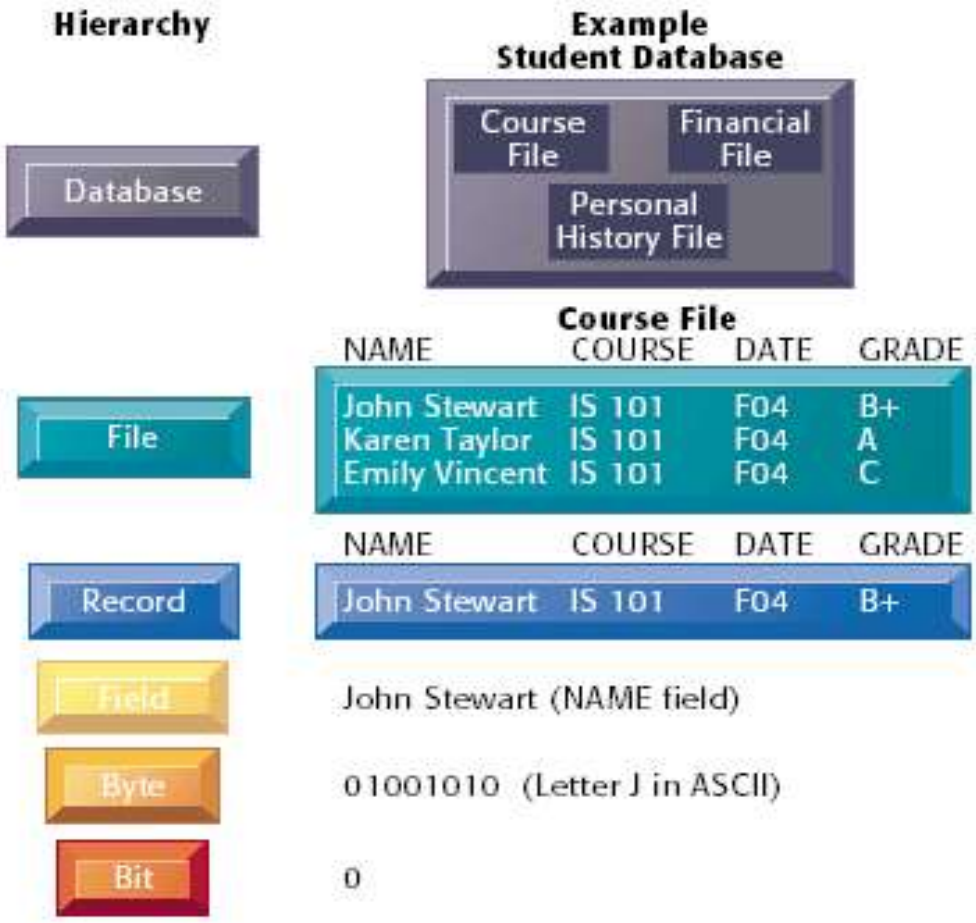
A logical relationship between distinct records is referred to as file organization. This method specifies how disc blocks are mapped to file records. The word "file organization" also refers to the method by which records are organized into blocks and then placed on a storage media.

- **Bit: Smallest unit of data; binary digit (0,1)**
- **Byte: Group of bits that represents a single character (**Historically, the byte was the number of bits used to encode a single character of text in a computer and for this reason, it is the smallest addressable unit of memory in many computer architectures.**)**
- **Field: Group of words or a complete number**
- **Record: Group of related fields**
- **File: Group of records of same type**

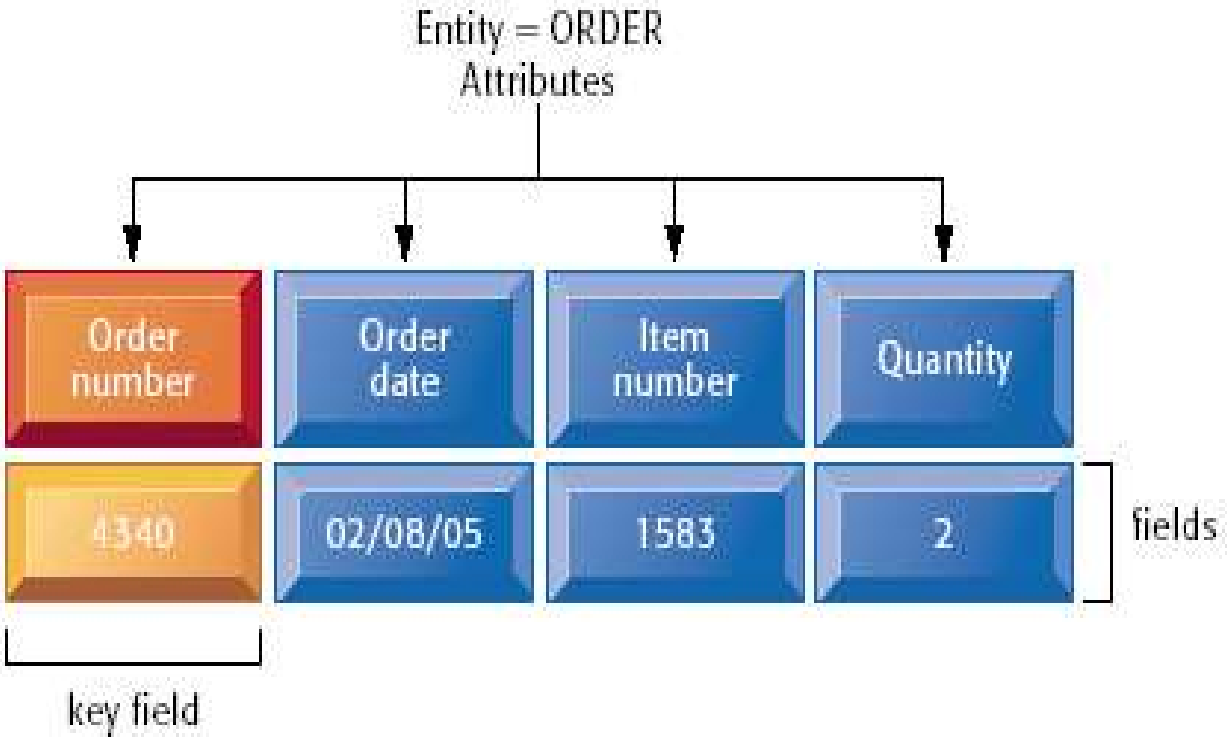
File Organization Terms and Concepts (Continued)

- **Database:** Group of related files
- **Entity:** Person, place, thing, event about which information is maintained
- **Attribute:** Description of a particular entity
- **Key field:** Identifier field used to retrieve, update, sort a record

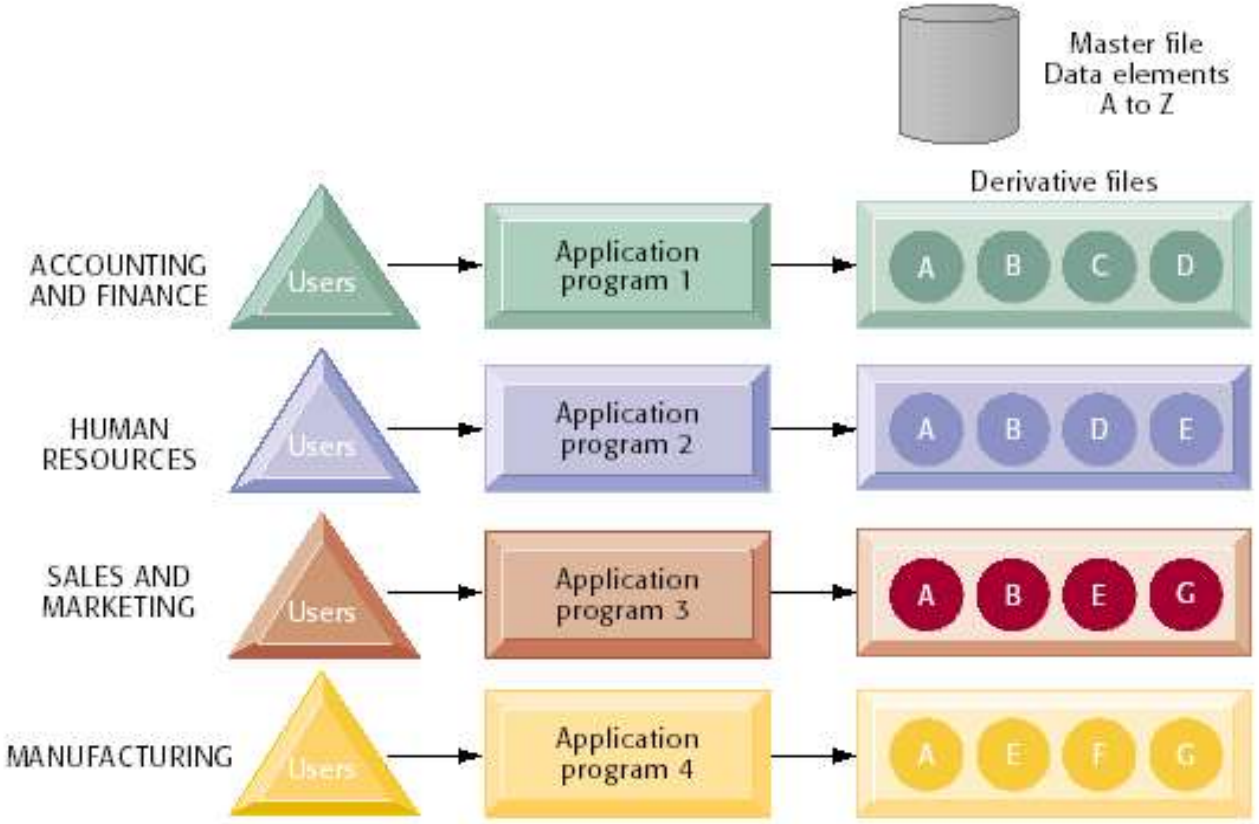
The Data Hierarchy



Entities and Attributes



DIAGRAMATICALLY ORGANIZATION OF DATA IN A TRADITIONAL FILE ENVIRONMENT CAN BE UNDERSTOOD AS:

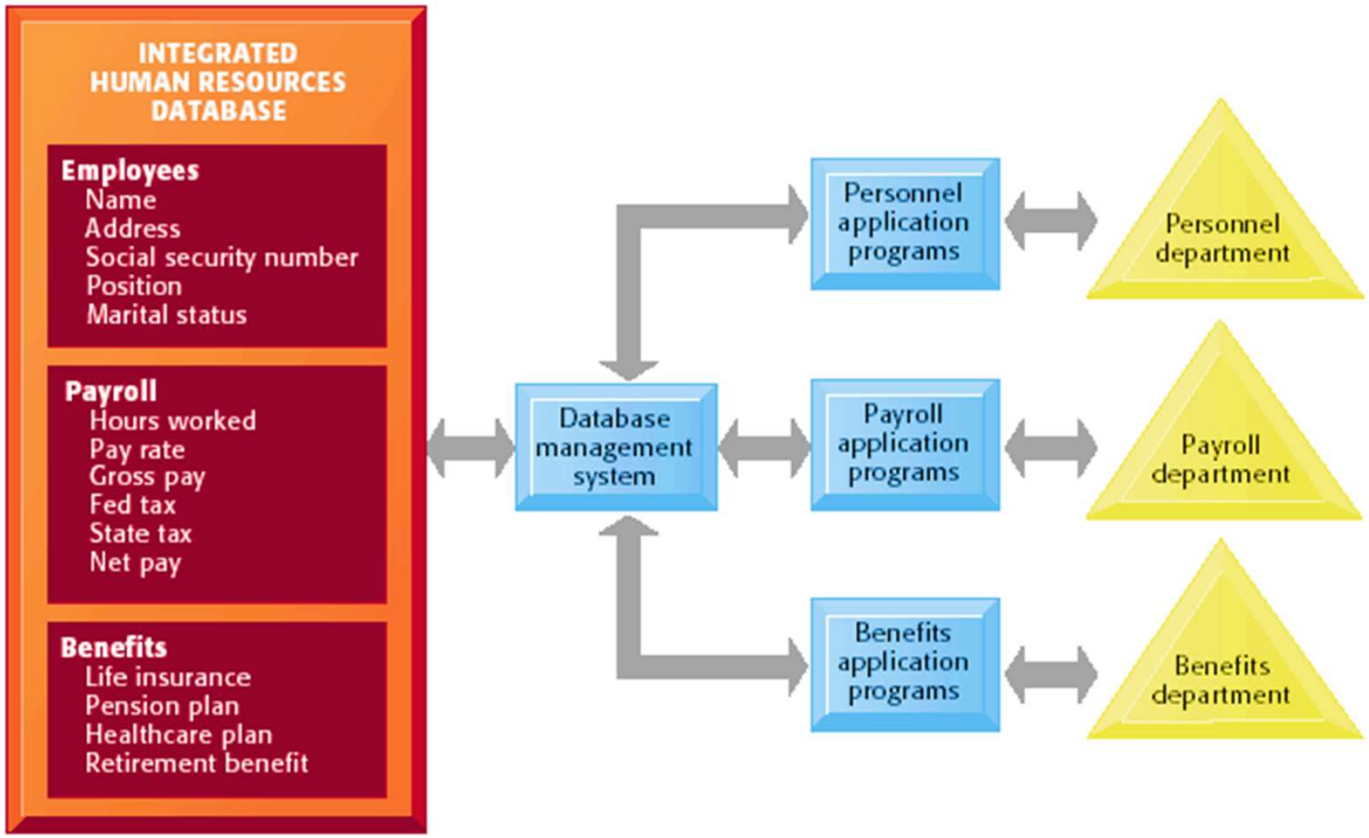


Traditional File Processing

Database Management System (DBMS)

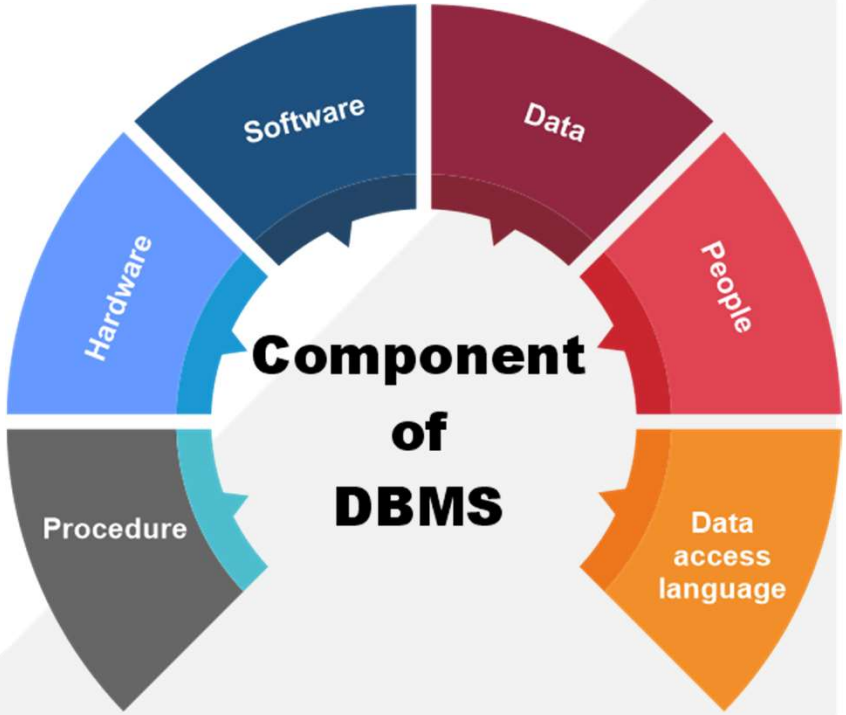
- Software for creating and maintaining databases
- Permits firms to rationally manage data for the entire firm
- Acts as interface between application programs and physical data files
- Separates logical and design views of data
- Solves many problems of the traditional data file approach

The Contemporary Database Environment



Components of DBMS

- **Data definition language:** Specifies content and structure of database and defines each data element
- **Data manipulation language:** Used to process data in a database
- **Data dictionary:** Stores definitions of data elements and data characteristics



Sample Data Dictionary Report

NAME: AMT-PAY-BASE
FOCUS NAME: BASEPAY
PC NAME: SALARY

DESCRIPTION: EMPLOYEE'S ANNUAL SALARY

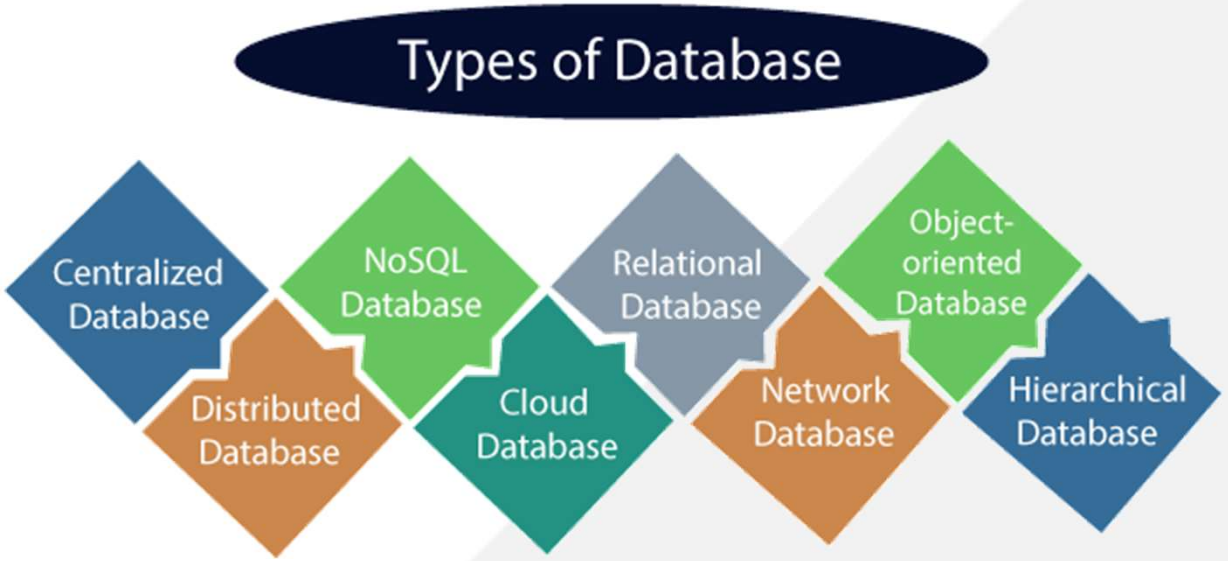
SIZE: 9 BYTES
TYPE: N (NUMERIC)
DATE CHANGED: 01/01/04
OWNERSHIP: COMPENSATION
UPDATE SECURITY: SITE PERSONNEL
ACCESS SECURITY: MANAGER, COMPENSATION PLANNING AND RESEARCH
MANAGER, JOB EVALUATION SYSTEMS
MANAGER, HUMAN RESOURCES PLANNING
MANAGER, SITE EQUAL OPPORTUNITY AFFAIRS
MANAGER, SITE BENEFITS
MANAGER, CLAIMS PAYING SYSTEMS
MANAGER, QUALIFIED PLANS
MANAGER, SITE EMPLOYMENT/EEO
BUSINESS FUNCTIONS USED BY: COMPENSATION
HR PLANNING
EMPLOYMENT
INSURANCE
PENSION
401K

PROGRAMS USING: PI01000
PI02000
PI03000
PI04000
PI05000

REPORTS USING: REPORT 124 (SALARY INCREASE TRACKING REPORT)
REPORT 448 (GROUP INSURANCE AUDIT REPORT)
REPORT 452 (SALARY REVIEW LISTING)
PENSION REFERENCE LISTING

Types of Databases

- **Relational DBMS**
- **Hierarchical and network DBMS**
- **Object-oriented databases**

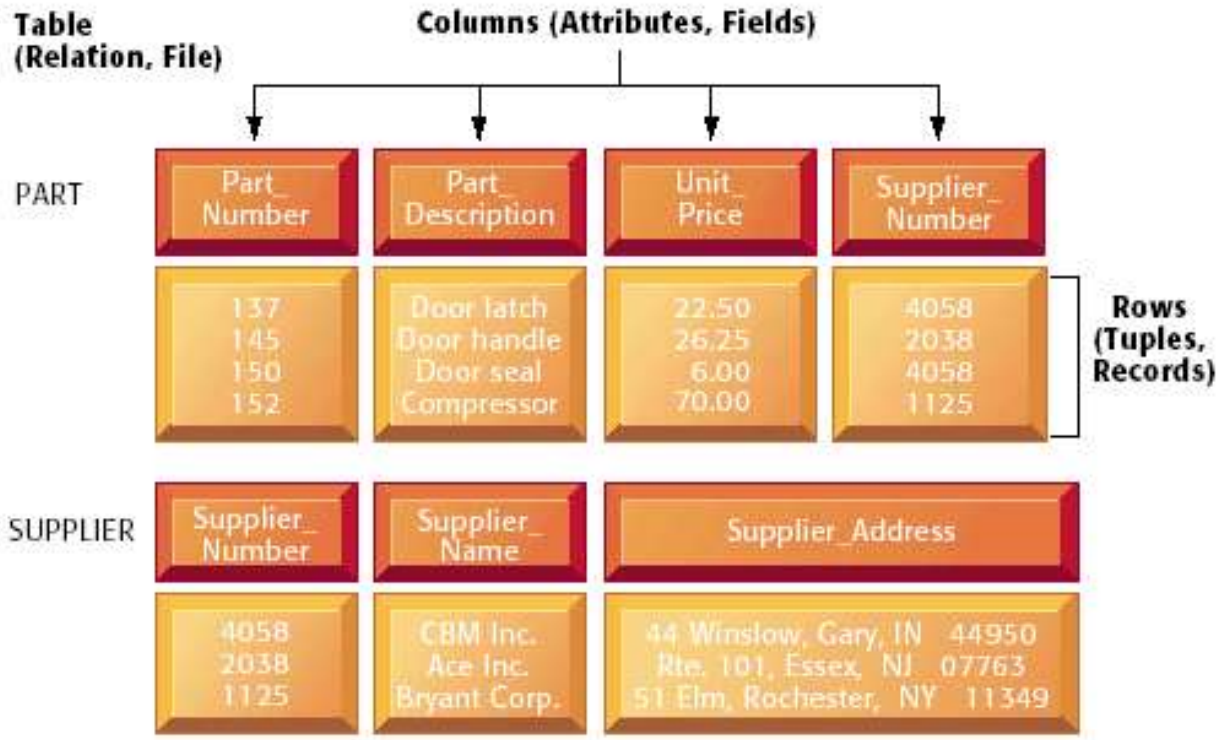


Relational DBMS:

- **Represents data as two-dimensional tables called relations**
- **Relates data across tables based on common data element**
- **Examples: DB2, Oracle, MS SQL Server**

DB2 (Database 2), is **a set of relational database products built and offered by IBM**.

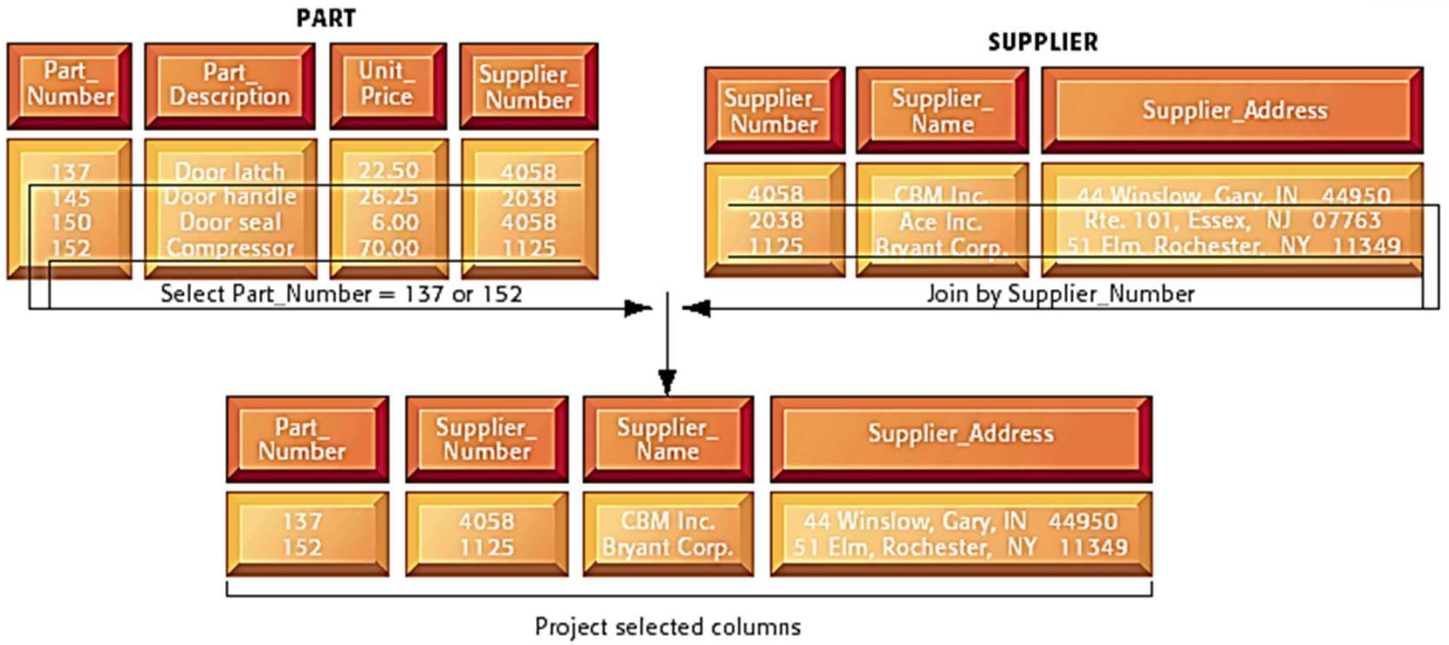
The Relational Data Model



Three Basic Operations in a Relational Database:

- **Select:** Creates subset of rows that meet specific criteria
- **Join:** Combines relational tables to provide users with information
- **Project:** Enables users to create new tables containing only relevant information

The Three Basic Operations of a Relational DBMS

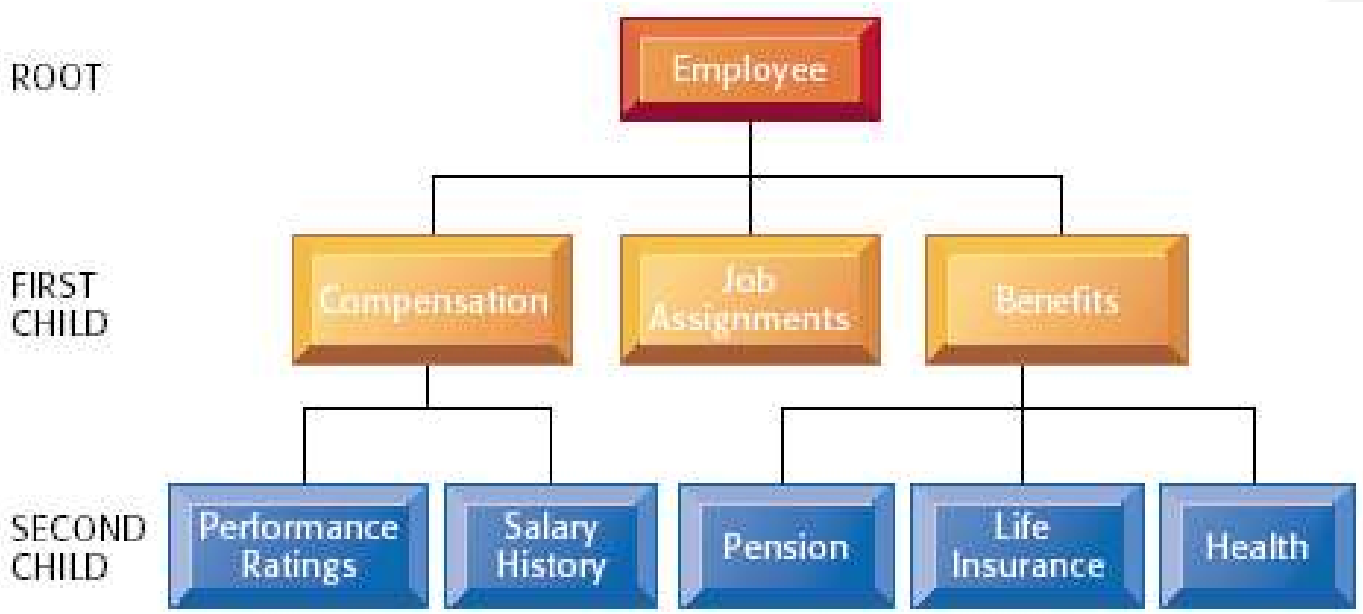


Hierarchical and Network DBMS

Hierarchical DBMS:

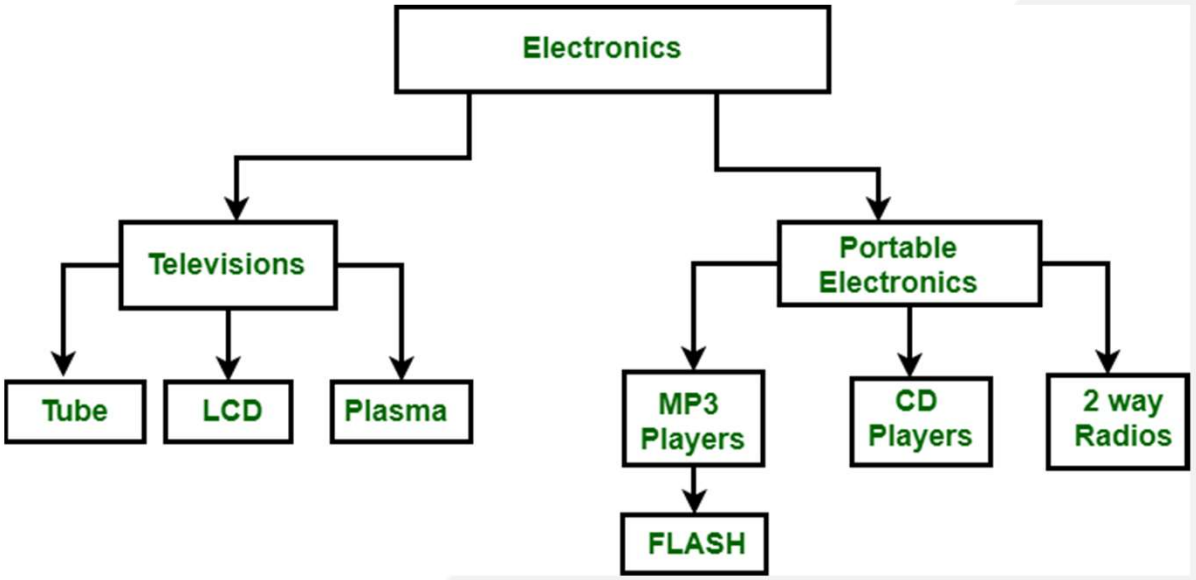
- **Organizes data in a tree-like structure**
- **Supports one-to-many parent-child relationships**
- **Prevalent in large legacy systems**

A Hierarchical Database for a Human Resources System



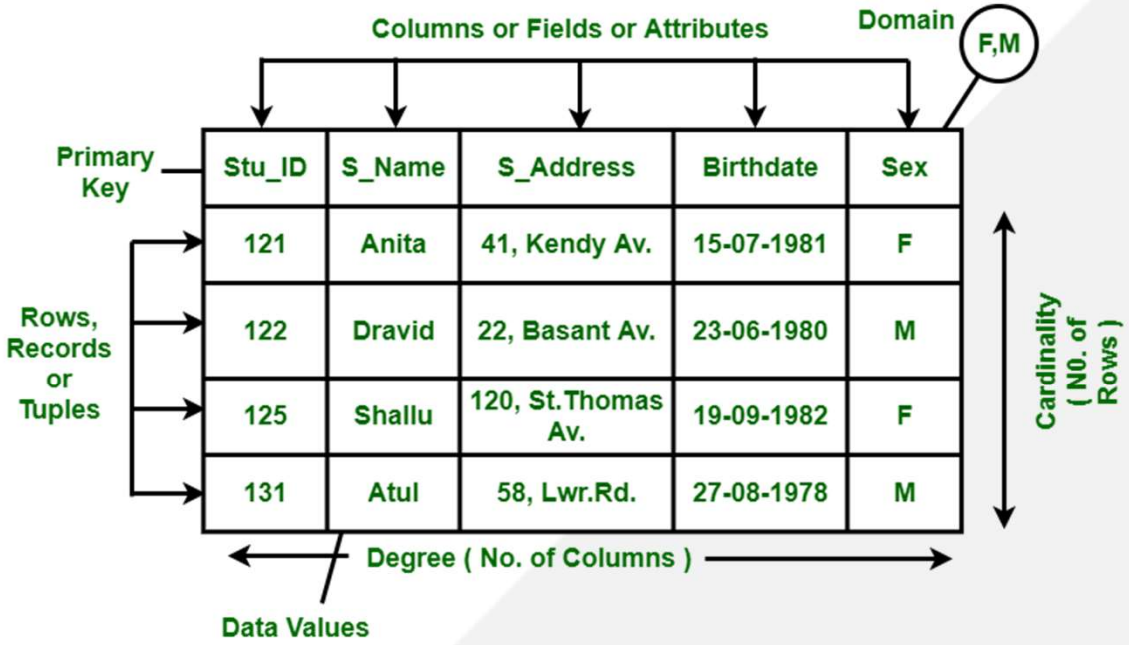
Hierarchical data model is the oldest type of data model. It was developed by IBM in 1968. It organizes data in a tree-like structure. Hierarchical model consists of the following :

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



Relational Data Model: Relational data model was developed by E.F. Codd in 1970. There are no physical links as they are in the hierarchical data model. Following are properties of the relational data model :

- Data is represented in form of a table only.
- It deals only with data, not with physical structure.
- It provides information regarding metadata.
- At the intersection of row and column there will be only one value for the tuple.
- It provides a way to handle queries with ease.

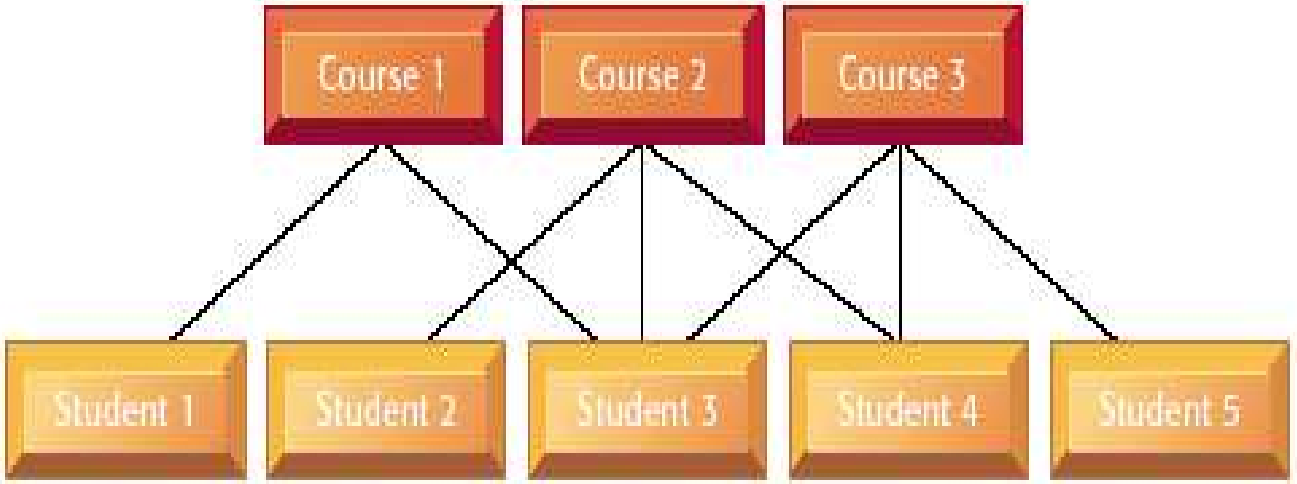


Hierarchical and Network DBMS

Network DBMS:

- **Depicts data logically as many-to-many relationships**

The Network Data Model



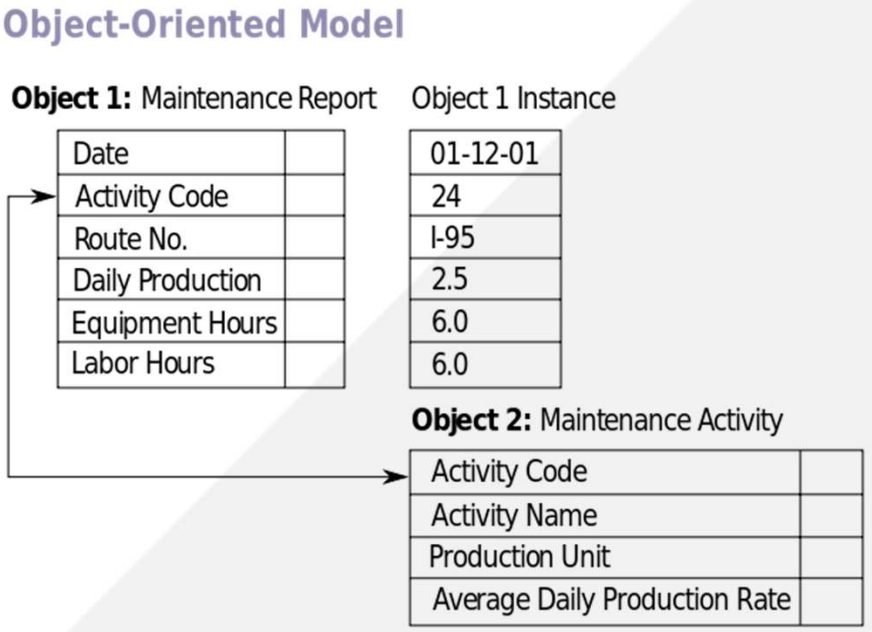
Hierarchical and Network DBMS

Disadvantages:

- **Outdated**
- **Less flexible compared to RDBMS**
- **Lack support for ad-hoc and English language-like queries**

Object-Oriented Databases

- **Object-oriented DBMS: Stores data and procedures as objects that can be retrieved and shared automatically**
- **Object-oriented databases** are a type of database management system. Different database management systems provide additional functionalities. Object-oriented databases add database functionality to object programming languages, creating more manageable code bases.



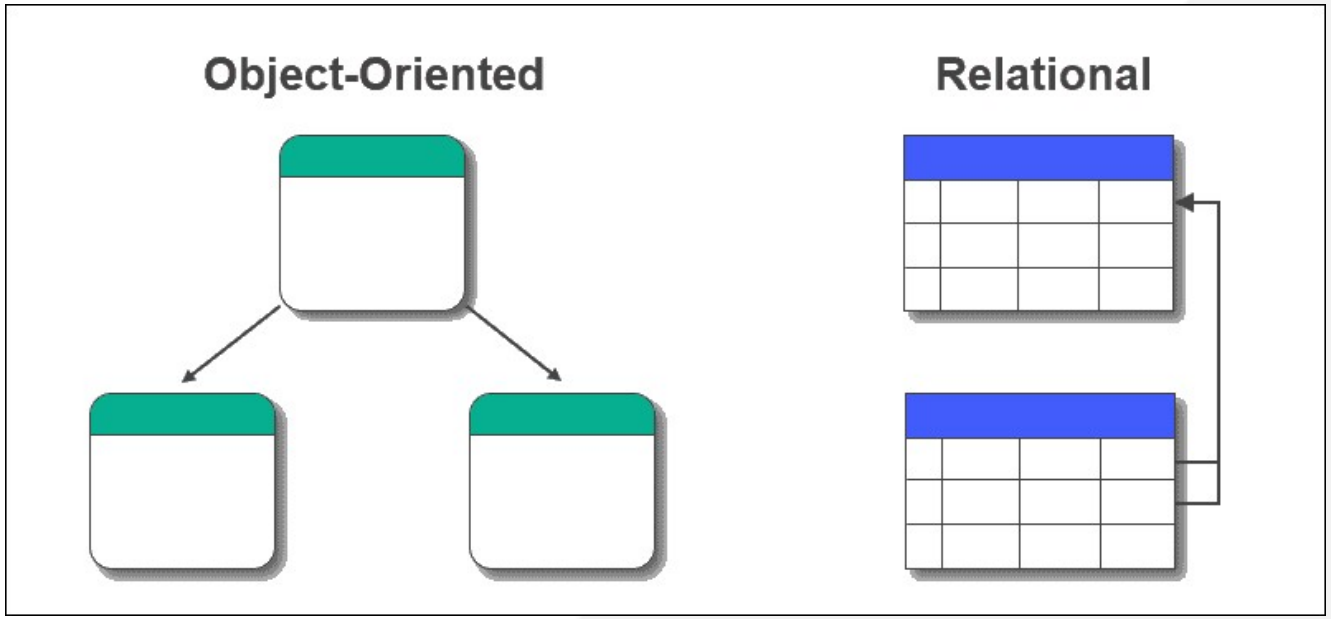
- **Object-relational DBMS: Provides capabilities of both object-oriented and relational DBMS**

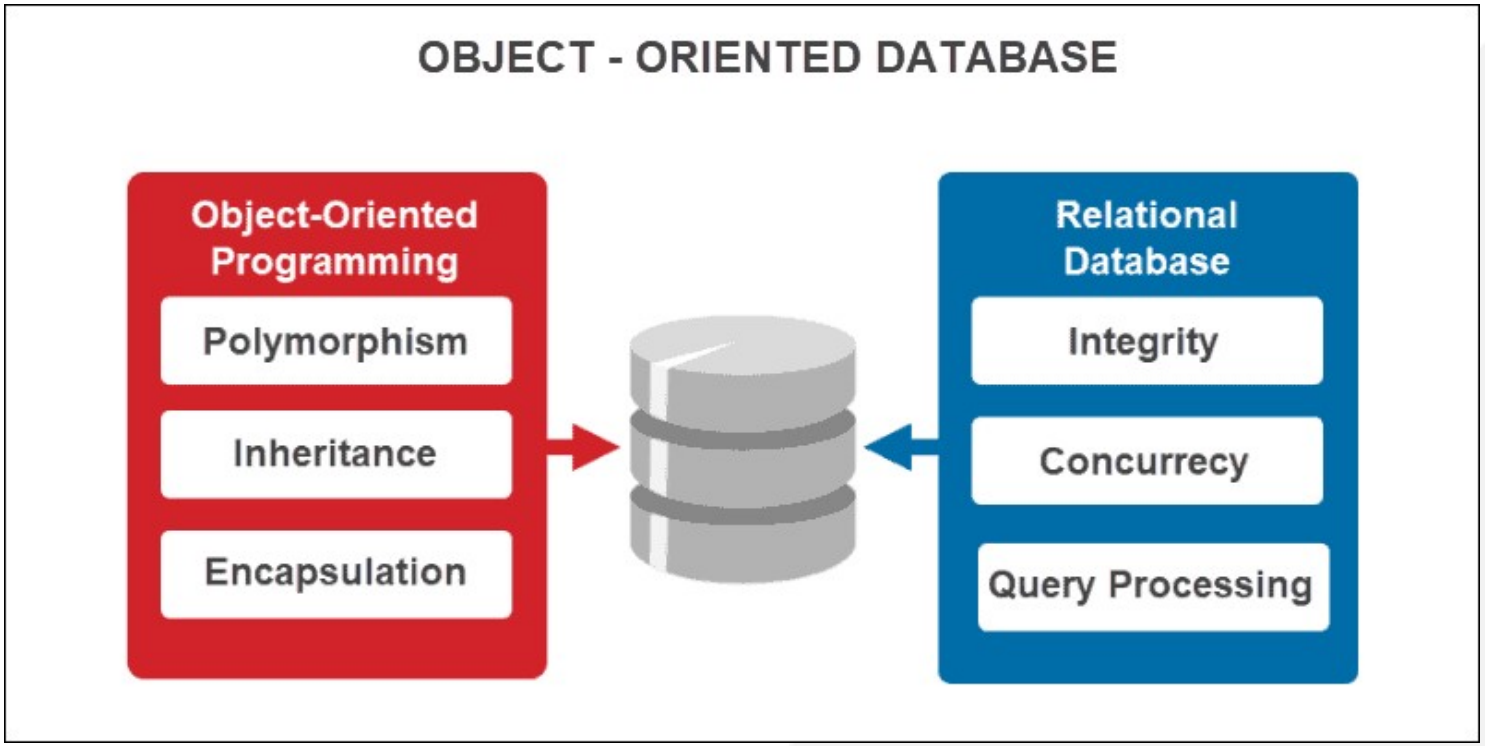
Object Database Definition

An object database is managed by an **o**bject-**o**riented **d**ata**b**ase **m**anagement **s**ystem (OODBMS). The database combines object-oriented programming concepts with relational database principles.

- **Objects** are the basic building block and an instance of a class, where the type is either built-in or user-defined.
- **Classes** provide a schema or blueprint for objects, defining the behavior.
- **Methods** determine the behavior of a class.
- **Pointers** help access elements of an object database and establish relations between objects.

- The main characteristic of objects in OODBMS is the possibility of **user-constructed types**. An object created in a project or application saves into a database as is.
- Object-oriented databases directly deal with data as complete objects. All the information comes in one instantly available object package instead of multiple tables.




OBJECT - ORIENTED DATABASE
Object-Oriented Programming
Polymorphism
Inheritance
Encapsulation
Relational Database
Integrity
Concurrecy
Query Processing

Object-Oriented Programming Concepts

Object-oriented databases closely relate to object-oriented programming concepts. The four main ideas of object-oriented programming are:

- **Polymorphism**
- **Inheritance**
- **Encapsulation**
- **Abstraction**

These four attributes describe the critical characteristics of object-oriented management systems.

1. Polymorphism

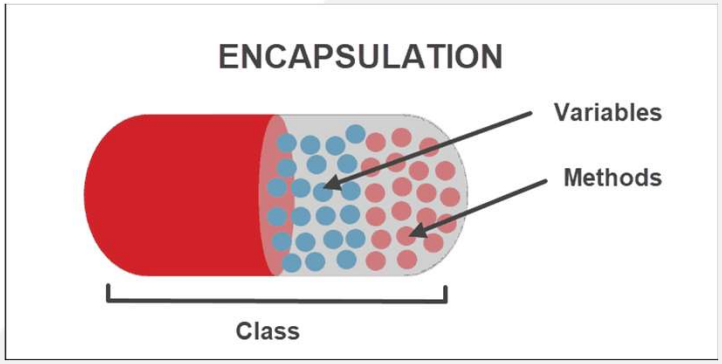
- **Polymorphism** is the capability of an object to take **multiple forms.** This ability allows the same program code to work with different data types.
- Both a car and a bike are able to *break*, but the mechanism is different. In this example, the action break is a polymorphism. The defined action is **polymorphic** — the result changes depending on which vehicle performs.

2. Inheritance

- **Inheritance** creates a **hierarchical relationship between related classes** while making parts of **code reusable.**
- Defining new types inherits all the existing class fields and methods plus further extends them.
- The existing class is the **parent** class, while the **child** class extends the parent.
- For example, a parent class called *Vehicle* will have child classes *Car* and *Bike*. Both child classes **inherit** information from the parent class and **extend** the parent class with new information depending on the vehicle type.

3. Encapsulation

- **Encapsulation** is the ability to group data and mechanisms into a single object to provide access protection. Through this process, pieces of information and details of how an object works are **hidden**, resulting in data and function security.
- Classes interact with each other through methods without the need to know how particular methods work.
- As an example, a car has descriptive characteristics and actions. You can change the color of a car, yet the model or make are examples of properties that cannot change.
- A class **encapsulates** all the car information into one entity, where some elements are modifiable while some are not.



4. Abstraction

- **Abstraction** is the procedure of representing only the essential data features for the needed **functionality**. The process selects vital information while unnecessary information stays hidden.
- Abstraction helps reduce the complexity of modeled data and allows reusability.
- For example, there are different ways for a computer to connect to the network. A web browser needs an internet connection. However, the connection type is irrelevant.
- An established connection to the internet represents an **abstraction**, whereas the various types of connections represent different implementations of the abstraction.

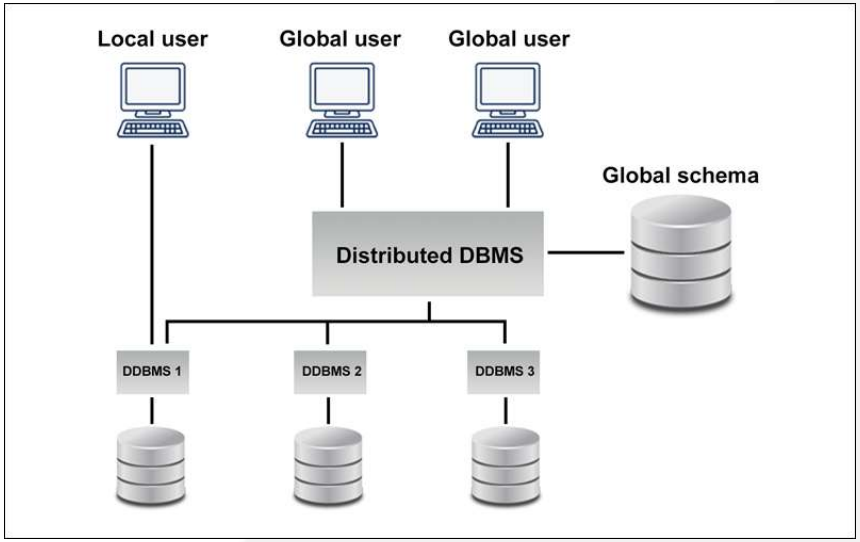
Centralized database:

- Used by single central processor or multiple processors in client/server network
- There are advantages and disadvantages to having all corporate data in one location.
- Security is higher in central environments, risks lower.
- If data demands are highly decentralized, then a decentralized design is less costly, and more flexible.

Distributed database:

- **Databases can be decentralized either by partitioning or by replicating.**

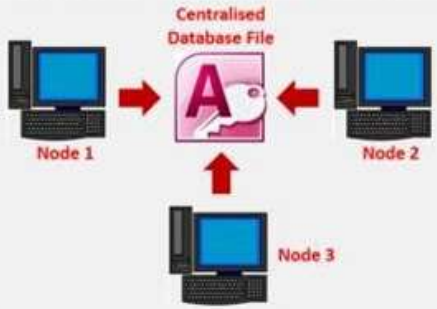
- **Partitioned database: Database is divided into segments or regions. For example, a customer database can be divided into Eastern customers and Western customers, and two separate databases maintained in the two regions.**



Centralised vs. Distributed Databases

Centralised Databases

A **single** database located at **1 site** on a network



Advantages:

Since there is only 1 database file, it is easier to:

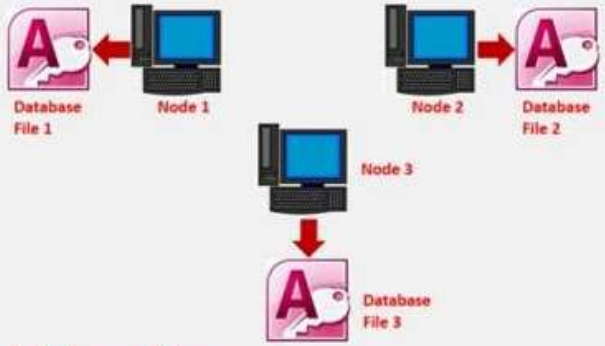
- Get a complete view of Data
- Manage, update an backup Data

Disadvantages:

- Bottle necking from multiple users accessing the same file – slowing down productivity

Distributed Databases

Consists of **2 or more files** located at different sites on a network



Advantages:

Having multiple database files means:

- Users wont interfere with each other when accessing / manipulating Data
- Speed since files are retrieved from nearest location
- If one site fails, the system can still run

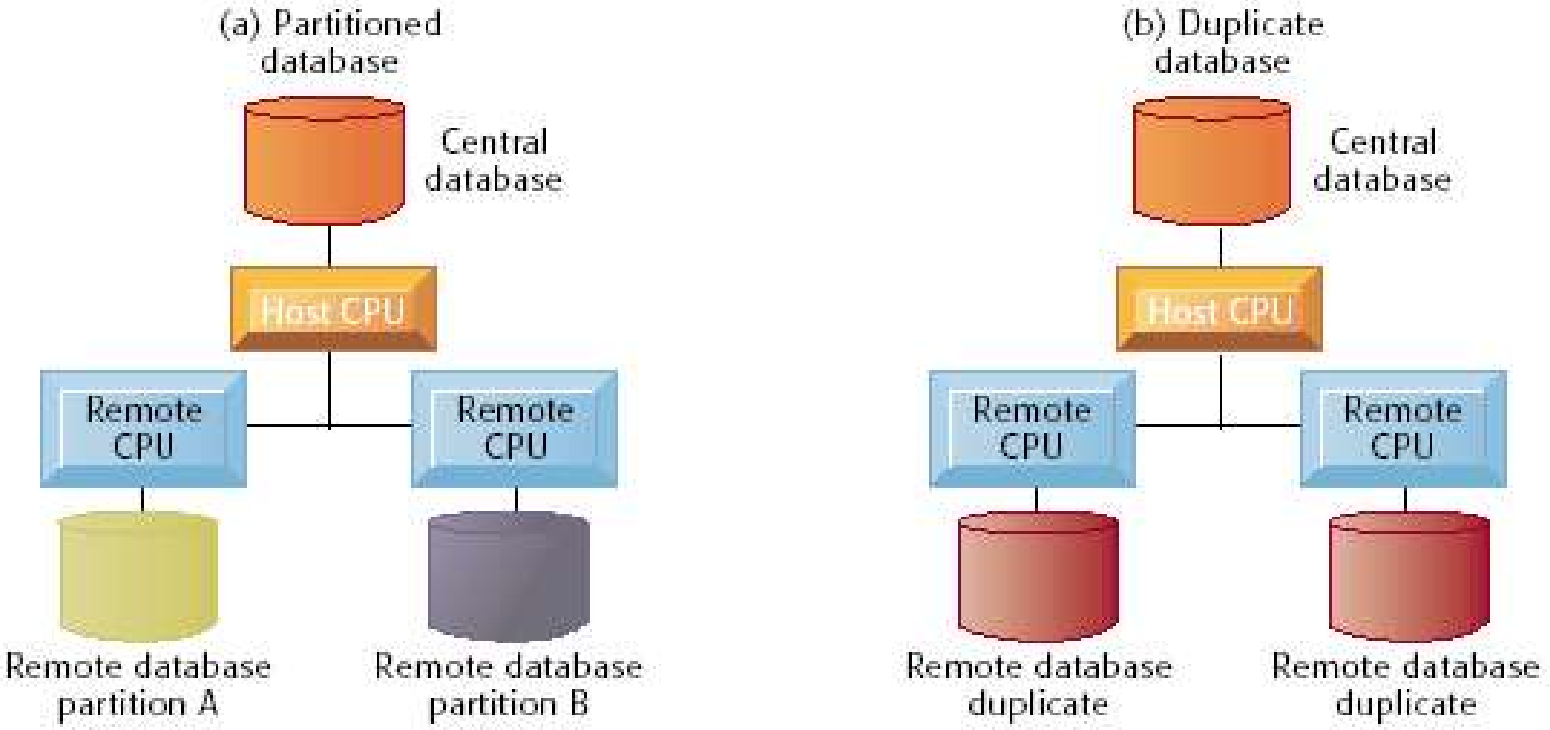
Disadvantages:

- Time for Synchronisation of the multiple databases
- Data Replication for each different database file

- **Duplicated database:** The database is completely duplicated at two or more locations. The separate databases are synchronized in off hours on a batch basis.

- Regardless of which method is chosen, data administrators and business managers need to understand how the data in different databases will be coordinated and how business processes might be affected by the decentralization.

Distributed Databases



Ensuring Data Quality

- **Corporate and government databases have unexpectedly poor levels of data quality.**
- **National consumer credit reporting databases have error rates of 20-35%.**
- **32% of the records in the FBI's Computerized Criminal History file are inaccurate, incomplete, or ambiguous.**
- **Gartner Group estimates that consumer data in corporate databases degrades at the rate of 2% a month.**

Ensuring Data Quality: (Continued)

- **The quality of decision making in a firm is directly related to the quality of data in its databases.**

- **Data Quality Audit: Structured survey of the accuracy and level of completeness of the data in an information system.**

- **Data Cleansing: Consists of activities for detecting and correcting data in a database or file that are incorrect, incomplete, improperly formatted, or redundant.**

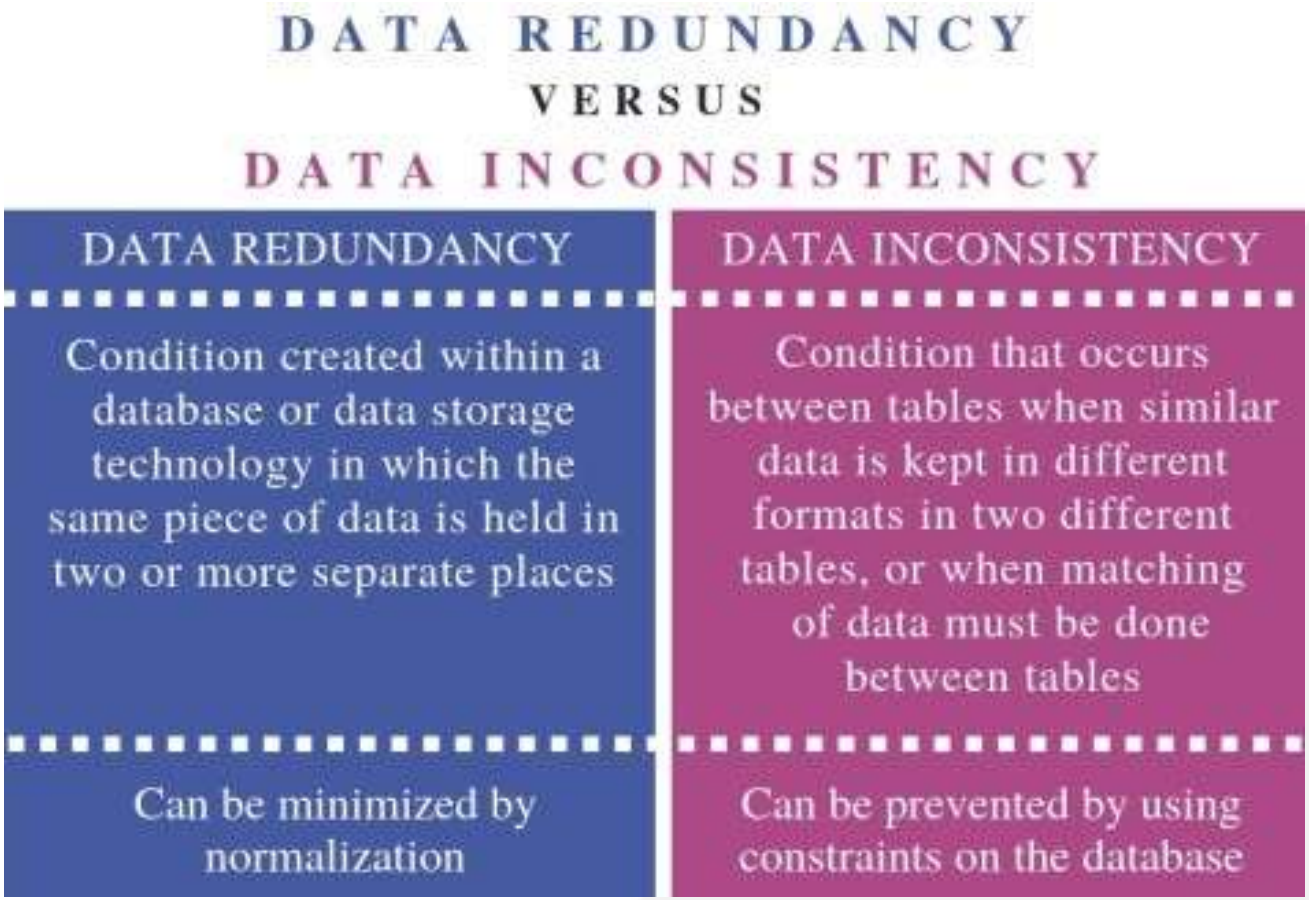
Problems with the Traditional File Environment

Data Redundancy and Inconsistency:

- **Data redundancy: The presence of duplicate data in multiple data files so that the same data are stored in more than one place or location.**

- **Data inconsistency: The same attribute may have different values.**

Problems with the Traditional File Environment



Problems with the Traditional File Environment

Data Consistency

Data consistency is ensuring that data is correct after it has been processed.

For example, if you had to calculate someone's age from their date of birth, and their age was calculated incorrectly, then you would say that the data has become inconsistent.

This may have happened because the data was entered incorrectly, or calculated incorrectly or for another reason.

If you had to convert a measurement in one unit into another unit, and they were in fact incorrect, then the data has become inconsistent. This could have happened for the same reasons as the date of birth and age error.

Data Redundancy

Data redundancy in a database means that the same data is present in more than one table. Or in the case of a flat file database, there are records with partly duplicated data. For example

Jones, 48, Male, Teacher
Jones, 48, 3 Advent Drive
Jones, employee number 22345

As you can see in the records above, the name is repeated three times and the age is duplicated twice. A relational database can avoid this duplication.

This is usually a mark of an inefficient database and people go to great lengths to avoid it.

In order to reduce duplicated data, you can use the three 'normal forms' of database design i.e. First Normal, Second Normal and the most efficient (but complex) Third Normal Form.

Problems with the Traditional File Environment (Continued)

Program-data dependence:

- The coupling of data stored in files and the specific programs required to update and maintain those files such that changes in programs require changes to the data.

Lack of flexibility:

- A traditional file system can deliver routine scheduled reports after extensive programming efforts, but it cannot deliver ad-hoc reports or respond to unanticipated information requirements in a timely fashion.

*[Ad-hoc reporting is when reports are generated on request or created on request. They are usually created for a specific use or to answer a precise question. For example, you may change a report to view current sales metrics from a specific product, rather than viewing sales metrics for all products.]

Problems with the Traditional File Environment (Continued)

Poor security:

- **Because there is little control or management of data, management will have no knowledge of who is accessing or even making changes to the organization's data.**

Lack of data sharing and availability:

- **Information cannot flow freely across different functional areas or different parts of the organization. Users find different values of the same piece of information in two different systems, and hence they may not use these systems because they cannot trust the accuracy of the data.**

Efficiency Issues with Real Databases

Indexing:

- How to efficiently find all songs written by Paul Simon in a database with 10,000,000 entries?
- Data structures for representing sorted order on fields

Disk management:

- Databases are often too big to fit in RAM, leave most of it on disk and swap in blocks of records as needed – could be slow

Concurrency:

- Transaction semantics: either all updates happen *in batch* or none (commit or rollback)
- Like delete one record and simultaneously add another but guarantee not to leave in an inconsistent state
- Other users might be blocked till done

Query optimization: The overall process of choosing the most efficient

- The order in which you JOIN tables can drastically affect the size of the intermediate tables.

Real-world issues with databases

- It's all about scaling up to many records (and many users)
- Data warehousing:
  - Full database is stored in secure, off-site location
  - Slices, snapshots, or views are put on interactive query servers for fast user access ("staging")
  - Might be processed or summarized data
- Databases are often distributed:
  - Different parts of the data held in different sites
  - Some queries are local, others are "corporate-wide"
  - How to do distributed queries?
  - How to keep the databases synchronized?

Traditional Data	Big Data
Traditional data is generated in enterprise level.	Big data is generated outside the enterprise level.
Its volume ranges from Gigabytes to Terabytes.	Its volume ranges from Petabytes to Zettabytes or Exabytes.
Traditional database system deals with structured data.	Big data system deals with structured, semi-structured,database, and unstructured data.
Traditional data is generated per hour or per day or more.	But big data is generated more frequently mainly per seconds.
Traditional data source is centralized and it is managed in centralized form.	Big data source is distributed and it is managed in distributed form.

Continued…..

Traditional Data	Big Data
Data integration is very easy.	Data integration is very difficult.
Normal system configuration is capable to process traditional data.	High system configuration is required to process big data.
The size of the data is very small.	The size is more than the traditional data size.
Traditional data base tools are required to perform any data base operation.	Special kind of data base tools are required to perform any database schema-based operation.
Normal functions can manipulate data.	Special kind of functions can manipulate data.

Continued…..

Traditional Data	Big Data
Its data model is strict schema based and it is static.	Its data model is a flat schema based and it is dynamic.
Traditional data is stable and inter relationship.	Big data is not stable and unknown relationship.
Traditional data is in manageable volume.	Big data is in huge volume which becomes unmanageable.
It is easy to manage and manipulate the data.	It is difficult to manage and manipulate the data.
Its data sources includes ERP transaction data, CRM transaction data, financial data, organizational data, web transaction data etc.	Its data sources includes social media, device data, sensor data, video, images, audio etc.

FAQs

- What are the discrepancies of traditional databases?
- How data was arranged in various previous databases?
- What is data redundancy and inconsistency?
- Why bigdata management is most prior as compared to the others?
- Differences between traditional databases and big data.

References

- RamezElmasri and Shamkant B. Navathe, “Fundamentals of Database System”, The Benjamin / Cummings Publishing Co.
- https://www.geeksforgeeks.org/
- Sinan Ozdemir, “Principles of data science”, Packt> Publications, 2016..

THANK YOU