FALL SEMESTER 21-22

CSE 2031 – PRINCIPLES OF DATABASE MANAGEMENT SYSTEMS

Course Type: ETH

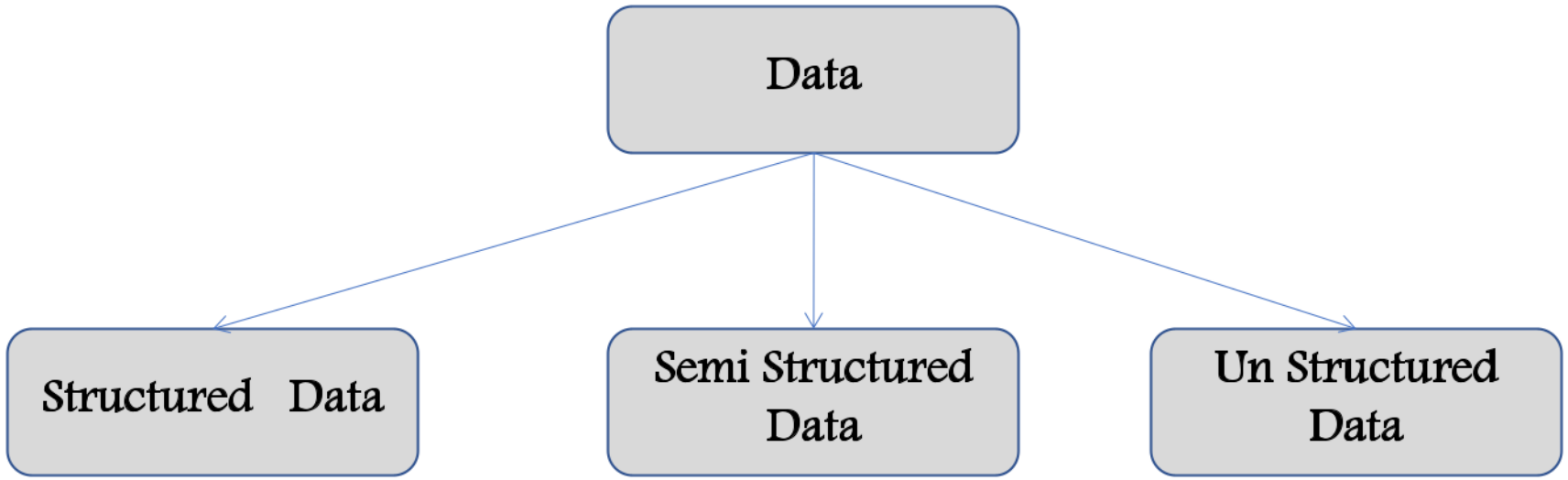
S.Vengadeswaran
Asst. Prof. (SCOPE)
VIT Vellore

❖ Data vs Information

- **Data** – is a collection of a distinct small unit of information.
- Variety of forms like text, numbers, media, bytes, etc.
- It can be stored in pieces of paper or electronic memory, etc.
- ‘Datum' that means 'single piece of information.'
- In computing, Data is information that can be translated into a form for efficient movement and processing.
- **Information?** – Meaningful data
- Types of data?
  - Structured
  - Semi – Structured
  - Unstructured

#	Country, Other	Total Cases	New Cases	Total Deaths	New Deaths	Total Recovered	New Recovered	Active Cases	Serious, Critical	Tot Cases/ 1M pop	Deaths/ 1M pop
	World	199,007,621	+32,498	4,240,312	+535	179,614,880	+19,036	15,152,429	90,255	25,531	544.0
1	USA	35,768,924		629,380		29,673,290		5,466,254	11,513	107,381	1,889
2	India	31,695,368		424,808		30,849,681		420,879	8,944	22,726	305
3	Brazil	19,938,358		556,886		18,645,993		735,479	8,318	93,085	2,600
4	Russia	6,288,677		159,352		5,625,890		503,435	2,300	43,072	1,091
5	France	6,146,619		111,885		*5,702,014*		332,720	1,137	93,942	1,710
6	UK	5,880,667		129,719		4,520,199		1,230,749	869	86,136	1,900
7	Turkey	5,747,935		51,428		5,459,899		236,608	543	67,369	603
8	Argentina	4,935,847		105,772		4,581,132		248,943	3,913	108,137	2,317
9	Colombia	4,794,414		120,998		4,587,754		85,662	8,155	93,151	2,351

❖ Types of Data



Structured Data	Semi Structured Data	Un Structured Data
Predefined Schema	No predefine Formal schema	No predefine Formal schema
Predefined format	Predefined format	No Predefined format
Eg: Databases	Xml File	Text Files, Multimedia

Sample XML file

```
<?xml version="1.0"?>
- <birds>
  - <owl id="1201">
      <species>Bubo bubo</species>
      <name>Eagle Owl</name>
      <region>Eurasia</region>
    </owl>
  - <owl id="1202">
      <species>Strix occidentalis</species>
      <name>Spotted Owl</name>
      <region>North America</region>
    </owl>
</birds>
```

Introduction

- **Key Terminologies on Database concepts:**
  - **Database** is a collection/organization of coherent data with some inherent meaning.
  - In database the data is organized as Tables so that it can be easily accessed, managed or updated efficiently.
  - **Table (Relation)** is a collection of data organized as rows and columns.
    - **Rows (Tuples)** represents specific information of each individual entry in the table, and every row in the table has the same structure.
    - **Column (Attributes)** is a descriptive property of a particular type
    - **Domain** is set of permissible value of each attribute

Database 1: VIT University

Domain

Table 1: Faculty Details		
Faculty ID	Faculty Name	Department
101	Vengadesh	Analytics
102	Rajesh	Software System
103	Kumar	Information Security

Table/ Relation

Row/ Tuple

Column/ Attribute

Table 2: Student Details		
Student ID	Student Name	Address
BCE500	Sushanth	50, A1 Block
BCE501	Mihir Kumar	65, A11 Block
BCE502	Calpana	52, B2 Block

Table 3: Course Details				
Course Code	Course Title	Course Type	No of Students	Course Faculty ID
CSE 1003	Digital Logic and Design	Theory only	80	102
CSE 2005	Operating System	Theory and Lab	100	101
CSE 3001	Software Engineering	Lab only	70	103

CSE 2031 –DATABASE MANAGEMENT SYSTEMS – Outline

Module 1 **DATABASE SYSTEMS CONCEPTS AND ARCHITECTURE**	• History and motivation for database systems –characteristics of database approach – Actors on the scene – Workers behind the scene – Advantages of using DBMS approach– Data Models, Schemas, and Instances– Three–Schema Architecture and Data Independence– The Database System Environment– Centralized and Client/Server Architectures for DBMSs– Classification of database management systems.
Module 2 **DATA MODELING**	• Entity Relationship Model : Types of Attributes, Relationship, Structural Constraints – Relational Model, Relational model Constraints – Mapping ER model to a relational schema – Integrity constraints
Module 3 **SCHEMA REFINEMENT**	• Guidelines for Relational Schema – Functional dependency; Normalization, Relational Decomposition, Boyce Codd Normal Form, Multi–valued dependency and Fourth Normal form; Join dependency and Fifth Normal form.

CSE 2031 –DATABASE MANAGEMENT SYSTEMS – Outline

Module 4 **PHYSICAL DATABASE DESIGN**	• Indexing and Hashing: Single level indexing, multi-level indexing, dynamic multilevel Indexing, Ordered Indices – B+ tree Index Files – Static Hashing – Dynamic Hashing.
Module 5 **QUERY PROCESSING AND TRANSACTION PROCESSING**	• Translating SQL Queries into Relational Algebra – heuristic query optimization – cost based query optimization. Introduction to Transaction Processing – Transaction and System concepts – Desirable properties of Transactions–Characterizing schedules based on recoverability –Characterizing schedules based on serializability.
Module 6 **CONCURRENCY CONTROL AND RECOVERY TECHNIQUES**	• Two-Phase Locking Techniques for Concurrency Control – Concurrency Control based on timestamp. Recovery Concepts – Recovery based on deferred update – Recovery techniques based on immediate update – Shadow Paging
Module 7 **NoSQL DBMS**	• Introduction, Need of NoSQL, CAP Theorem, different NoSQL data models: Key-value stores, Column families, Document databases, Graph databases

❖ Course Objective

1. To understand the concept of DBMS and ER Modeling.
2. To explain the normalization, Query optimization and relational algebra.
3. To apply the concurrency control, recovery, security and indexing for the real time data

❖ Expected Course Outcome

1. Explain the basic concept and role of DBMS in an organization.
2. Illustrate the design principles for database design, ER model and normalization.
3. Demonstrate the basics of query evaluation and heuristic query optimization techniques.
4. Apply Concurrency control and recovery mechanisms for the desirable database problem.
5. Compare the basic database storage structure and access techniques including B Tree, B+ Tress and hashing
6. Review the fundamental view on unstructured data and its management.
7. Design and implement the database system with the fundamental concepts of DBMS

❖ Books

Text Books

1. Ramez Elmasri, Shamkant B. Navathe, "Fundamentals of Database Systems", Seventh Edition, Pearson Education, 2016.

Reference Books

1. Raghu Ramakrishnan, Johannes Gehrke, "Database Management Systems", Fourth Edition, Tata McGraw Hill, 2014.
2. Thomas Connolly, Carolyn Begg, Database Systems: A Practical Approach to Design, Implementation and Management, 6thEdition, Pearson, 2015
3. Meier, Andreas, Kaufmann, Michael, "SQL & NoSQL Databases – Models, Languages, Consistency Options and Architectures for Big Data Management", Springer, 2019

❖ Assessments

Kindly Note:
This includes participation in technical events like Hack-a-Thon, completion of on-line courses, publication of articles in scientific journals, and any other related activity.

As.No.	Assessment Title	Question Upload	Answer Upload	Due Date	Activity Date	Max. Mark	Weightage %
1	Digital Assignment - Digital Assignment - I	Mandatory	Mandatory	-	-	10	10
2	QUIZ - Quiz - I	Not Applicable	Not Applicable	-	-	20	10
3	QUIZ - Quiz - II	Not Applicable	Not Applicable	-	-	20	10
4	CAT - Continuous Assessment Test - I	Not Applicable	Not Applicable	-	-	30	15
5	CAT - Continuous Assessment Test - II	Not Applicable	Not Applicable	-	-	30	15
6	FAT - Final Assessment Test	Currently info not available	Currently info not available	-	-	100	40
						Total Weightage Mark	100

Confirm

(This will freeze the rubrics)

- **GATE CSE Database Management System (DBMS) 2019 Paper**
  - DBMS Weightage = 8 Marks
- **GATE CSE Database Management System (DBMS) 2018 Paper**
  - DBMS Weightage =: 6 Marks
- **GATE CSE Database Management System (DBMS) 2017**
  - DBMS Weightage in Set 1 = 8 Marks
  - DBMS Weightage in Set 2 = 8 Marks
- **GATE CSE Database Management System (DBMS) 2016 Paper**
  - DBMS Weightage in Set 1 = 5 Marks
  - DBMS Weightage in Set 2 = 5 Marks
- **GATE CSE Database Management System (DBMS) 2015 Paper**
  - DBMS Weightage in Set 1 = 6 Marks
  - DBMS Weightage in Set 2 = 5 Marks
  - DBMS Weightage in Set 3 = 6 Marks
- **GATE CSE Database Management System (DBMS) 2014 Paper**
  - DBMS Weightage in Set 1 = 3 Marks
  - DBMS Weightage in Set 2 = 4 Marks
  - DBMS Weightage in Set 3 = 6 Marks
- **GATE CSE Database Management System (DBMS) 2013 Paper**
  - DBMS Weightage = 7 Marks
- **GATE CSE Database Management System (DBMS) 2012 Paper**
  - DBMS Weightage = 11 Marks
- **GATE CSE Database Management System (DBMS) 2011 Paper**
  - DBMS Weightage = 7 Marks
- **GATE CSE Database Management System (DBMS) 2010 Paper**
  - DBMS Weightage = 7 Marks

Companies recruiting databases

Introduction

- **Database Management System (DBMS):** Database + Management System.
  - Database is a collection of data and Management System is a set of programs to store and retrieve those data.
  - Hence DBMS is a collection of inter-related data and set of programs to store & access those data in an easy and effective manner.
- **Relational Database Management System (RDBMS)** is a collection of programs or general-purpose software that enables user to create and maintain a database.
  - It is extension of DBMS, but the key extensions are the RDBMS able to store the relationship among the tables. Examples
    - **Licensed:** Oracle, Microsoft SQL Server, IBM DB2.
    - **Opensource:** PostgreSQL, MariaDB, SQLite, MySQL
    - **NoSQL:** MongoDB (Open Source)
    - **Cloud:** Amazon Aurora (SQL) and Amazon DynamoDB (NoSQL)

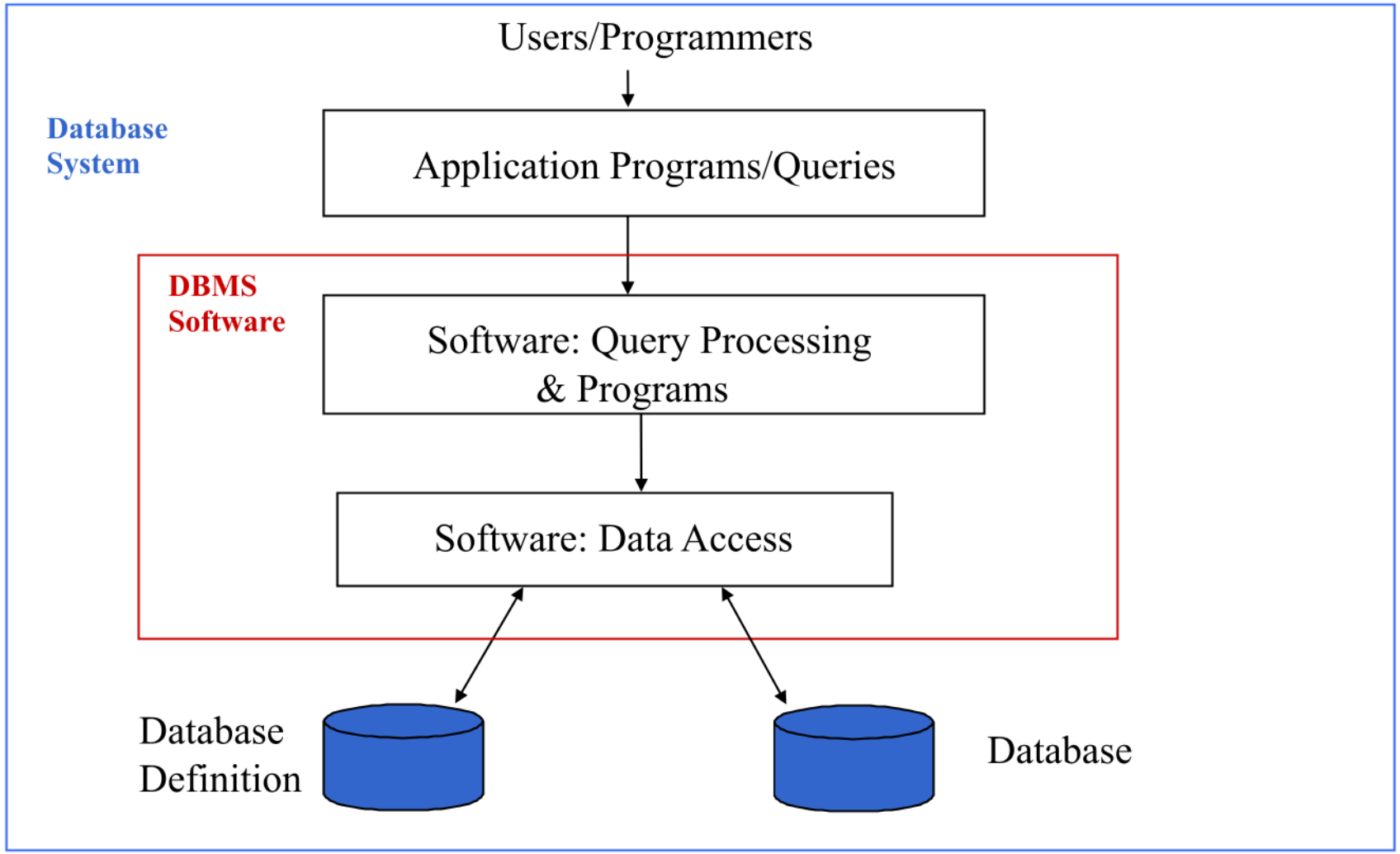
Thank you

DBMSs provide...

Facilities to:

- **Define** – specify data types, structures & constraints for the data to be stored in the database
- **Construct** – store the data
- **Manipulate** – pose queries to retrieve specific data, update data or generate reports based on the data

Database System Environment



Application of DBMS

Sector	Use of DBMS
Banking	For customer information, account activities, payments, deposits, loans, etc.
Airlines, Railways	For reservations and schedule information.
Universities, Colleges	For student information, course registrations, colleges and grades.
Telecommunications	It helps to keep call records, monthly bills, maintaining balances, etc.
Finance	For storing information about stock, sales, and purchases of financial instruments like stocks and bonds.
Sales	Use for storing customer, product & sales information.
HR	For information about employees, salaries, payroll, deduction, generation of paychecks, etc.

Popular DBMS Software

- MySQL
- Microsoft Access
- Oracle
- PostgreSQL
- SQLite
- IBM DB2
- LibreOffice Base
- MariaDB
- Microsoft SQL Server etc.

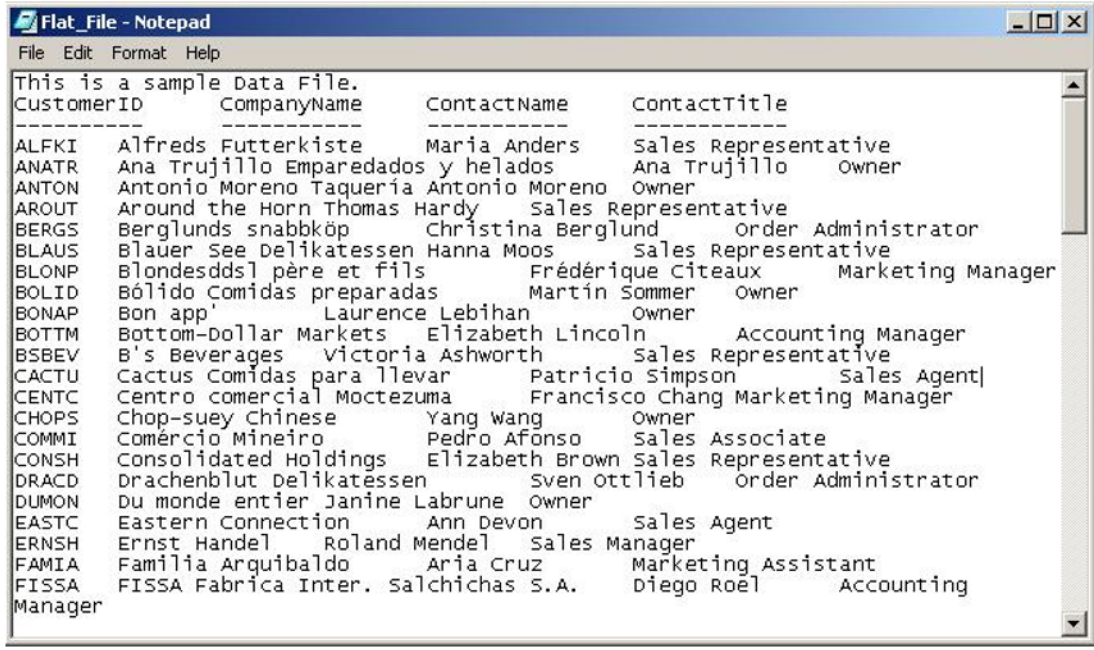
History of DBMS

1. File-Based

- 50 years of journey of its evolution
- 1968 – File-Based database were introduced.
- Data was maintained in a flat file.

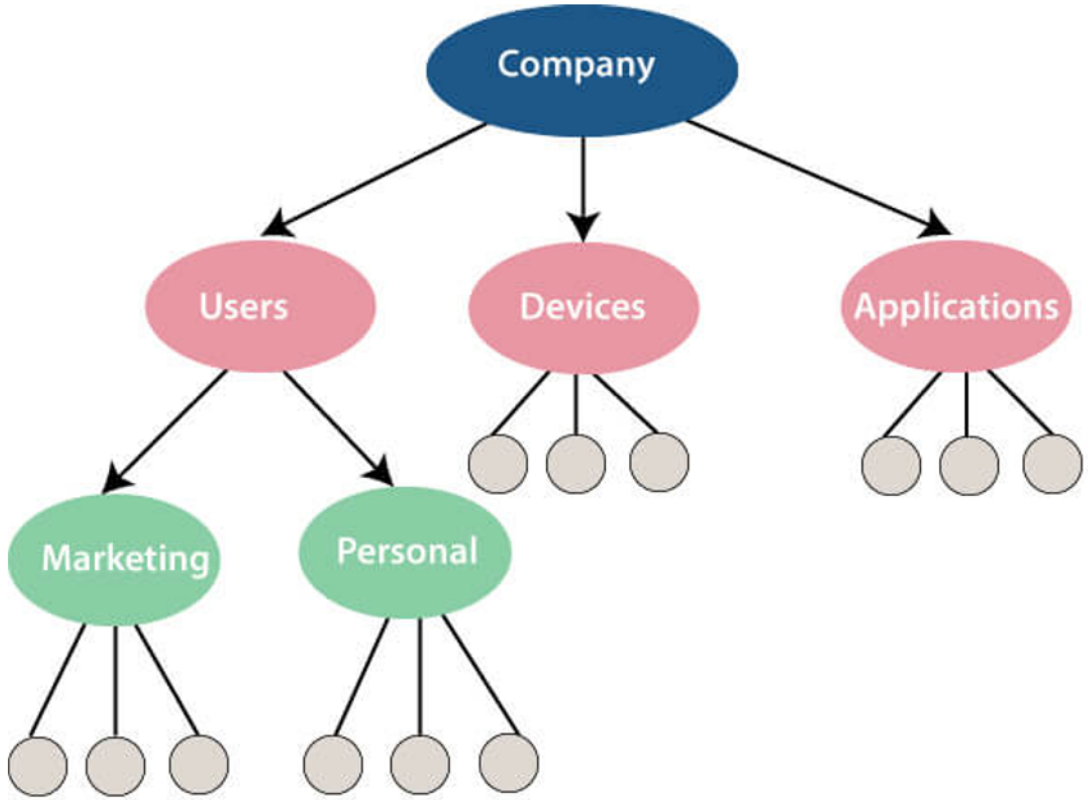
Advantages

- File system has various access methods,
  - e.g., sequential, indexed, and random.
- It requires extensive programming language such as COBOL, BASIC.



2. Hierarchical DB

- 1968-1980 - era of the Hierarchical DB.
- Prominent - IBM's first DBMS IMS (Information Management System).
- In this model, files are related in a parent/child manner.
- Like file system, this model also had some limitations
  - complex implementation,
  - lack structural independence,
  - can't handle a many-many relationship,



3. Network DB

- Charles Bachman – first N/W DBMS at Honeywell – Integrated Data Store (IDS).
- Developed in the early 1960s, Standardized in 1971 by the CODASYL group (Conference on Data Systems Languages).
- In this model, files are related as owners and members
- Network data model identified the following components:
  - Network schema (Database organization)
  - Sub-schema (views of database per user)
  - Data management language (procedural)
- This model also had some limitations like system complexity and difficult to design and maintain.

4. Relational Database

- 1970 – Present: Era of Relational Database – proposed by E.F. Codd.
- Relational database model has two main terminologies called instance and schema.
- The instance is a table with rows or columns
- Schema specifies the structure like name of the relation, type of each column and name.
- This model uses some mathematical concept like set theory and predicate logic.
- During the era of the relational database, many more models had introduced like OO DB, Graph DB, NOSQL DB, Cloud DB etc

5. Cloud database

- Cloud database facilitates you to store, manage, and retrieve their structured, unstructured data via a cloud platform.
- Accessible over the Internet.
- Cloud databases are also called a database as service (DBaaS) because they are offered as a managed service.
- Some best cloud options are
- AWS (Amazon Web Services – aurora, dynamo db
  - Snowflake Computing
  - Oracle Database Cloud Services
  - Microsoft SQL server
  - Google cloud spanner

5. Cloud database (Contd.)

Advantages of cloud database

Lower costs

- Generally, company provider does not have to invest in databases.

Automated

- Cloud databases are enriched with a variety of automated processes such as recovery, failover, and auto-scaling.

Increased accessibility

- You can access your cloud-based database from any location, anytime. All you need is just an internet connection.

6. NoSQL Database

A NoSQL database is an approach to design such databases that can accommodate a wide variety of data.

NoSQL stands for "not only SQL."

NoSQL databases are useful for a large set of distributed data.

Examples:

- MongoDB, CouchDB, Cloudant (Document-based)
- Memcached, Redis, Coherence (key-value store)
- HBase, Big Table, Accumulo (Tabular)

6. NoSQL Database (contd.)

Advantage of NoSQL

High Scalability

- NoSQL can handle an extensive amount of data because of scalability.
- If the data grows, NoSQL database scale it to handle that data in an efficient manner.

High Availability

- NoSQL supports auto replication. Auto replication makes it highly available because, in case of any failure, data replicates itself to the previous consistent state.

7. Graph Databases

- A graph database is a NoSQL database.
- It is a graphical representation of data.
- It contains nodes and edges.
- A node represents an entity, and each edge represents a relationship between two edges.
- Graph databases are beneficial for searching the relationship between data.
- Graph databases are very useful when the database contains a complex relationship and dynamic schema

Example

Example of a Database (with a Conceptual Data Model)

- Example:
  - Part of a UNIVERSITY environment
- Some entities:
  - STUDENTs
  - COURSEs
  - SECTIONs (of COURSEs)
  - (academic) DEPARTMENTs
  - INSTRUCTORs

Example of a simple database

Example of a simple database

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

Example of a simple database catalog

Example of a simplified database catalog

RELATIONS

Relation_name	No_of_columns
STUDENT	4
COURSE	4
SECTION	5
GRADE_REPORT	3
PREREQUISITE	2

COLUMNS

Column_name	Data_type	Belongs_to_relation
Name	Character (30)	STUDENT
Student_number	Character (4)	STUDENT
Class	Integer (1)	STUDENT
Major	Major_type	STUDENT
Course_name	Character (10)	COURSE
Course_number	XXXXNNNN	COURSE
....	....	.....
....	....	.....
....	....	.....
Prerequisite_number	XXXXNNNN	PREREQUISITE

DBMSs VS File Processing

DBMSs VS File Processing

Why do we need a DBMS?

Why not just use files to store data?

Limitations of File System

- **Data redundancy and inconsistency**
  - Multiple file formats, duplication of information in different files
- **Difficulty in accessing data**
  - Need to write a new program to carry out each new task

Contract#	Contract Date	Client #	Client Name	Status
H88203	01/01/2002	378	Bellevue Humane Society	Closed
H89014	01/01/2002	142	Furniture Showroom	Closed
H89247	04/01/2002	142	Furniture Showcase	Closed
H90032	07/01/2002	221	Dome World Recreation	Closed
H90103	07/01/2002	142	Future Showcase	Closed
J00180	07/01/2002	442	Carpets of Distinction	Ongoing

BONUS PAY						
TransactionID	Amount	Date	EmployeeID	Name	Department	Location
11	3000.00	1-Aug-2011	168	John Doe	Accounting	US
14	3000.00	1-Sep-2011	168	John Doe	Accounting	US
17	3000.00	1-Oct-2011	168	John Doe	Accounting	US
18	3000.00	1-Nov-2011	168	John Doe	Accounting	US

- **Data isolation — multiple files and formats**

Integrity constraints

Hard to add new constraints or change existing ones

Limitations of File System

- **Atomicity of updates**
  - Failures may leave database in an inconsistent state with partial updates carried out
  - E.g., transfer of funds from one account to another should either complete or not happen at all
- **Concurrent access by multiple users**
  - Concurrent accessed needed for performance
  - Uncontrolled concurrent accesses can lead to inconsistencies
  - E.g., two people reading a balance and updating it at the same time
- **Security problems**

DBMS vs. Flat File

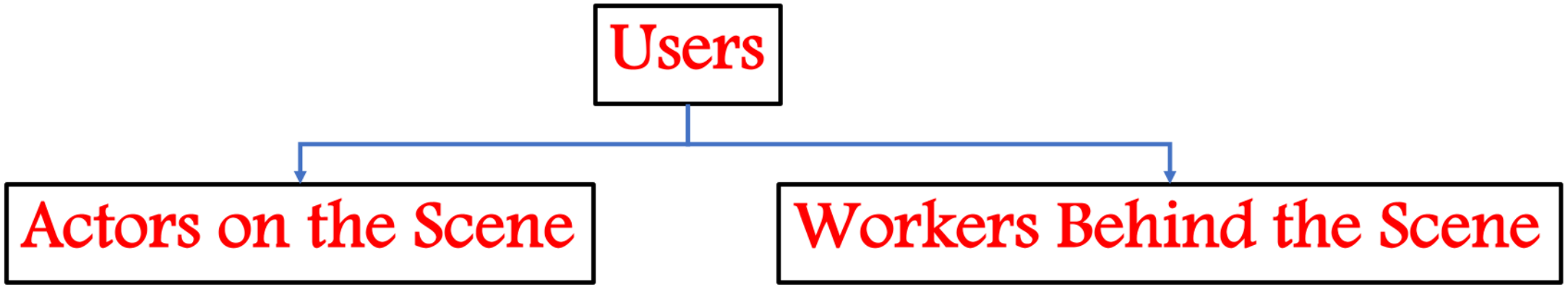
DBMS	File System
• DBMS is a collection of data. In DBMS, the user is not required to write the procedures.	• File system is a collection of data. In this system, the user has to write the procedures for managing the database.
• DBMS gives an abstract view of data that hides the details.	• File system provides the detail of the data representation and storage of data.
• DBMS provides a crash recovery mechanism, i.e., DBMS protects the user from the system failure.	• File system doesn't have a crash mechanism, i.e., if the system crashes while entering some data, then the content of the file will lost.
• DBMS provides a good protection mechanism.	• It is very difficult to protect a file under the file system.
• DBMS contains a wide variety of sophisticated techniques to store and retrieve the data.	• File system can't efficiently store and retrieve the data.
• DBMS takes care of Concurrent access of data using some form of locking.	• In the File system, concurrent access has many problems like redirecting the file while other deleting some information or updating some information.

Advantages of DBMS

Advantages of DBMS

- Controlling Redundancy
- Restricting Unauthorized Access
- Providing persistent storage for program objects and data structures
- Providing backup and recovery
- Providing concurrent access
- Representing complex relationship among data
- Enforcing Integrity constraints

Database Users –Actors on the Scene



- **Actors on the Scene** – Those who actually use and control the database content, and those who design, develop and maintain database applications
- **Workers Behind the Scene** – Those who design and develop the DBMS software and related tools, and the computer systems operators.

Actors on the Scene

1. Database administrators:

- Responsible for authorizing access to the database, for coordinating and monitoring its use, acquiring software and hardware resources, controlling its use and monitoring efficiency of operations.

2. Database Designers:

- Responsible to define the content, the structure, the constraints, and functions or transactions against the database. They must communicate with the end-users and understand their needs.

Actors on the Scene

3. End-users:

They use the data for queries, reports and some of them update the database content. End-users can be categorized into:

- **Casual:** access database occasionally when needed
- **Naïve or Parametric:** they make up a large section of the end-user population.
  - They use previously well-defined functions in the form of "transactions" against the database.
  - Users of Mobile Apps mostly fall in this category
  - Bank-tellers or reservation clerks are parametric users who do this activity for an entire shift of operations.
  - Social Media Users post and read information from websites

Actors on the Scene

End-users: (contd.)

- **Sophisticated:**
  - These include business analysts, scientists, engineers, others thoroughly familiar with the system capabilities.
  - Many use tools in the form of software packages that work closely with the stored database.
- **Stand-alone:**
  - Mostly maintain personal databases using ready-to-use packaged applications.
  - An example is the user of a tax program that creates its own internal database.
  - Another example is a user that maintains a database of personal photos and videos.

Actors on the Scene

4. System Analysts and Application Developers

- This category currently accounts for a very large proportion of the IT work force.

System Analysts:

- They understand the user requirements of naïve and sophisticated users and design applications including transactions to meet those requirements.

Application Programmers:

- Implement the specifications developed by analysts and test and debug them before deployment.

Workers behind the Scene

1. System Designers and Implementors:

- Design and implement DBMS packages in the form of modules, interfaces, test and debug them. The DBMS must interface with applications, language compilers, operating system components, etc.

2. Tool Developers:

- Design and implement software systems called tools for modeling and designing databases, performance monitoring, prototyping, test data generation, user interface creation, simulation etc. that facilitate building of applications and allow using database effectively.

3. Operators and Maintenance Personnel:

- They manage the actual running and maintenance of the database system hardware and software environment.

Additional Implications of using DB approach

- Potential for enforcing standards:
  - This is very crucial for the success of database applications in large organizations. **Standards** refer to data item names, display formats, screens, report structures, meta-data (description of data), Web page layouts, etc.
- Reduced application development time:
  - Incremental time to add each new application is reduced.

Additional Implications of using DB approach

- Flexibility to change data structures:
  - Database structure may evolve as new requirements are defined.
- Availability of current information:
  - Extremely important for on-line transaction systems such as airline, hotel, car reservations.
- Economies of scale:
  - Wasteful overlap of resources and personnel can be avoided by consolidating data and applications across departments.