DATABASE SYSTEMS

VIJENDRA

What is Data??

Raw or unorganized facts and figures (such as alphabets, numbers, symbols).

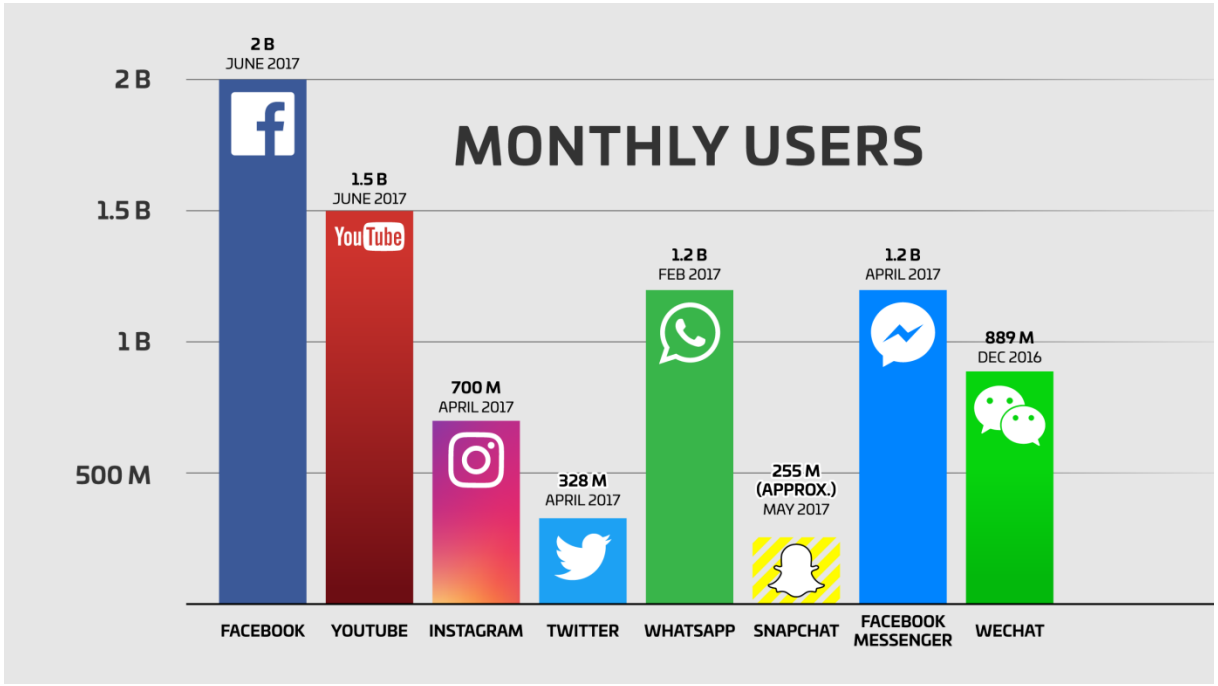
Example:

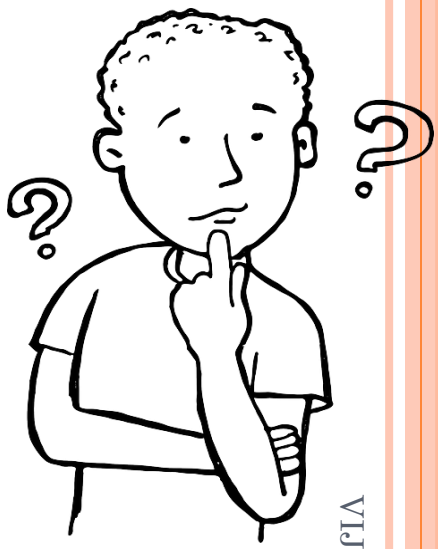
Student Data: Roll No, Name, Mobile, CGPA etc.

Faculty Data: Name, Designation, Email-id, Mobile etc.

User Profiles on Facebook, Twitter etc.

What can be done with Data





VIJENDRA

Data → Information

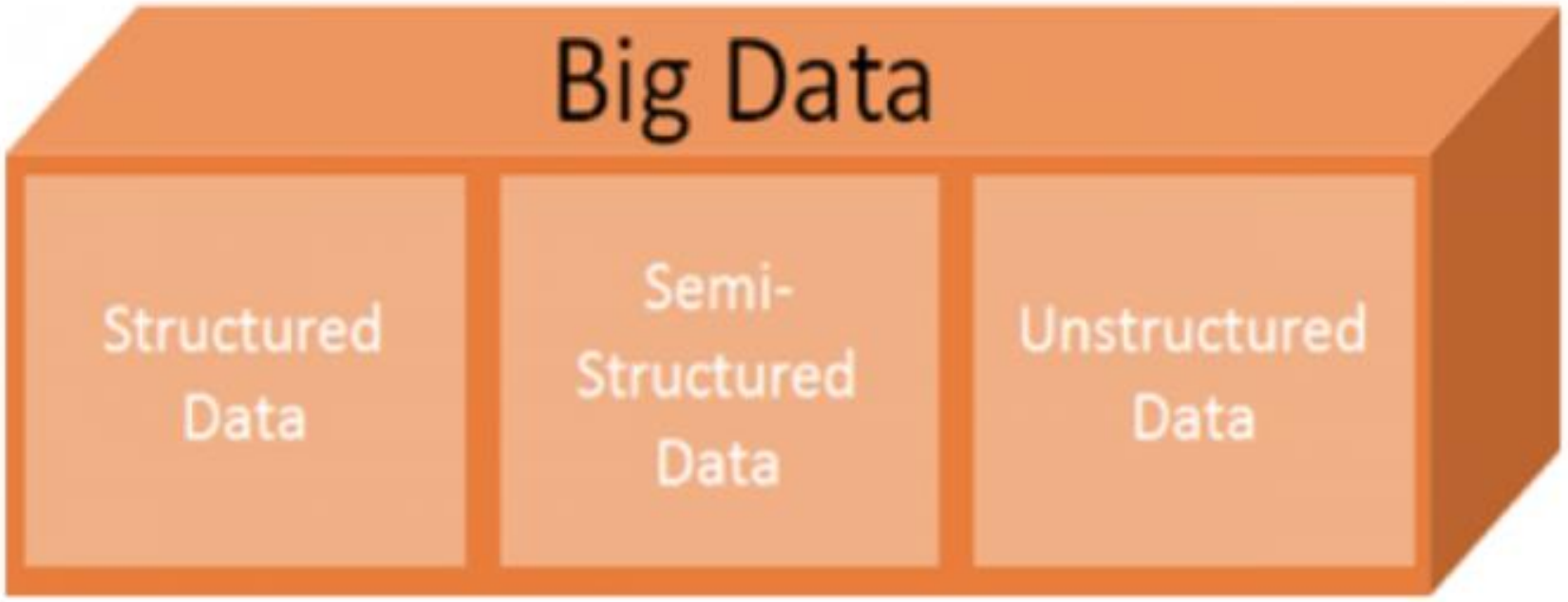
Information → Knowledge

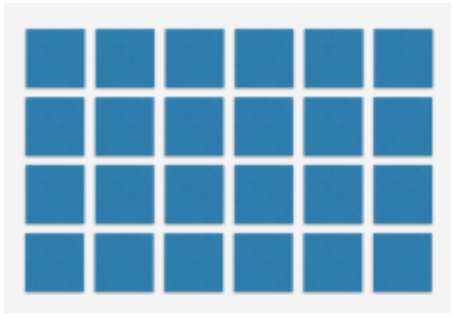
Knowledge → Growth of Business/ Organization

Data Types

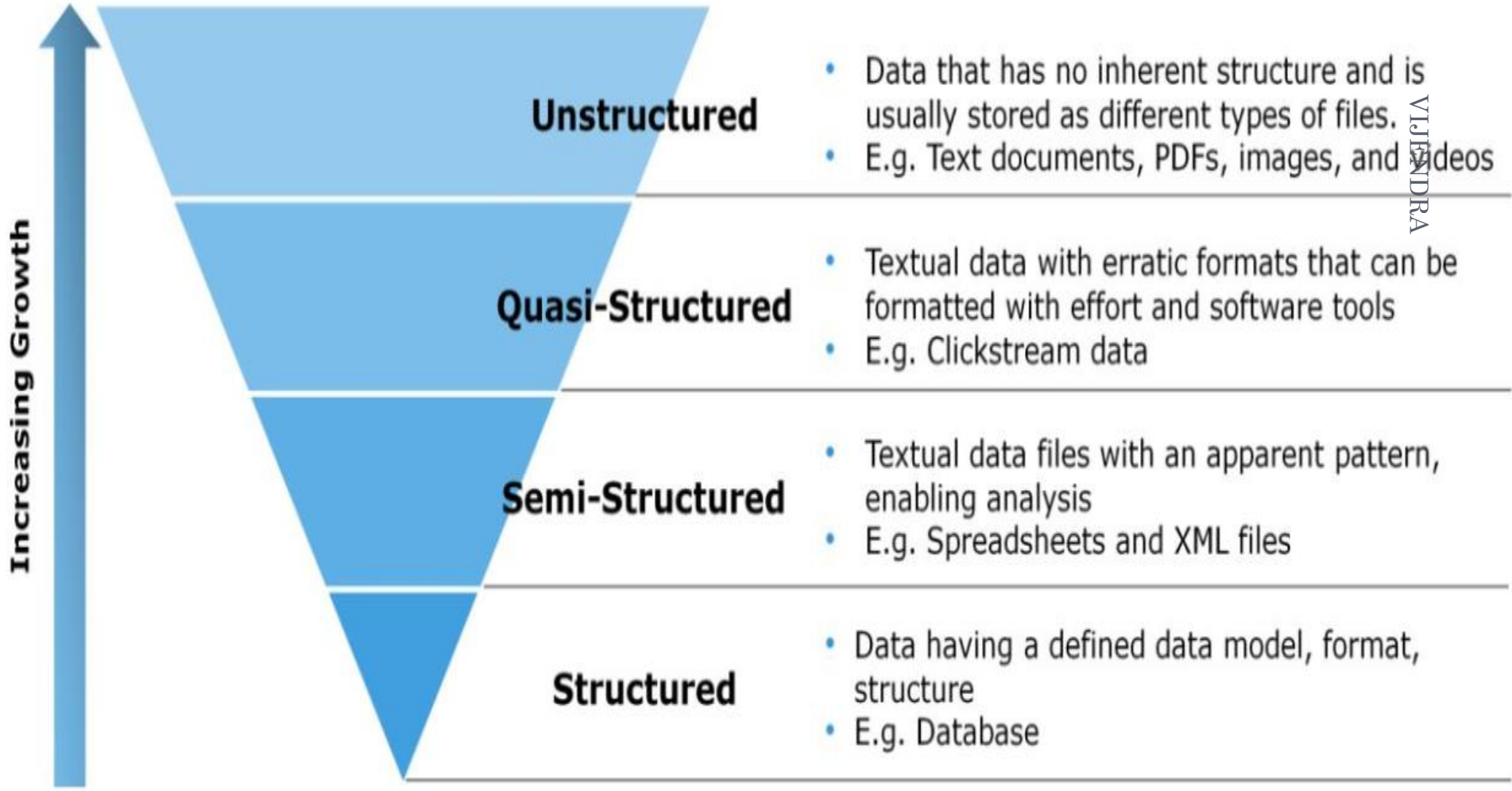
- Numeric → bit, tinyint, smallint, int, bigint, decimal, numeric, float, real
- Date/Time → Date, Time, Datetime, Timestamp, Year
- Character/String → Char, Varchar, Varchar (max), Text
- Unicode Character/String → NChar, NVarchar, NVarchar (max), NText
- Binary → Binary, Varbinary, Varbinary (max), image
- Miscellaneous → Clob, Blob, XML, JSON

Another Classification of Data

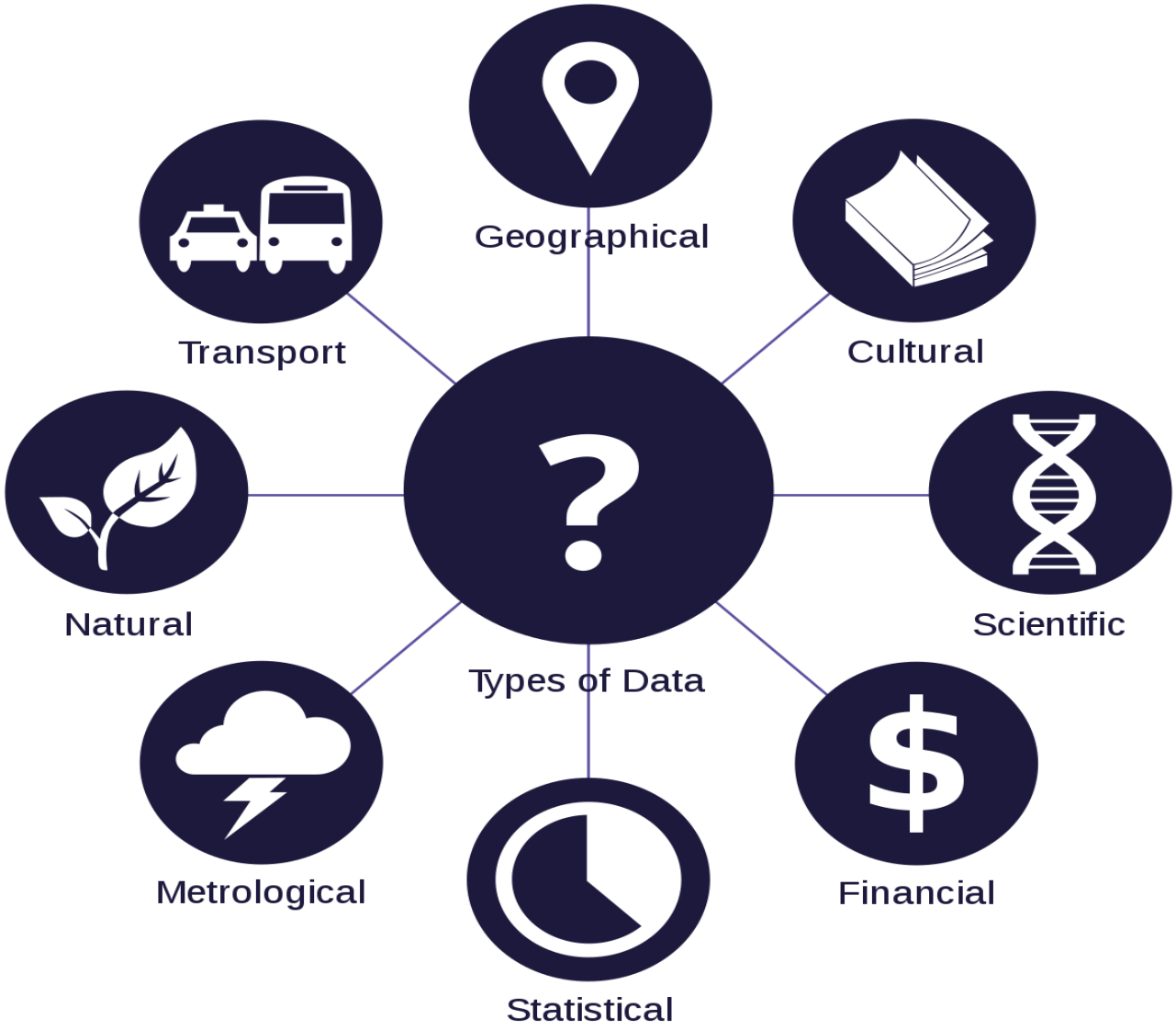




Types of Big Data



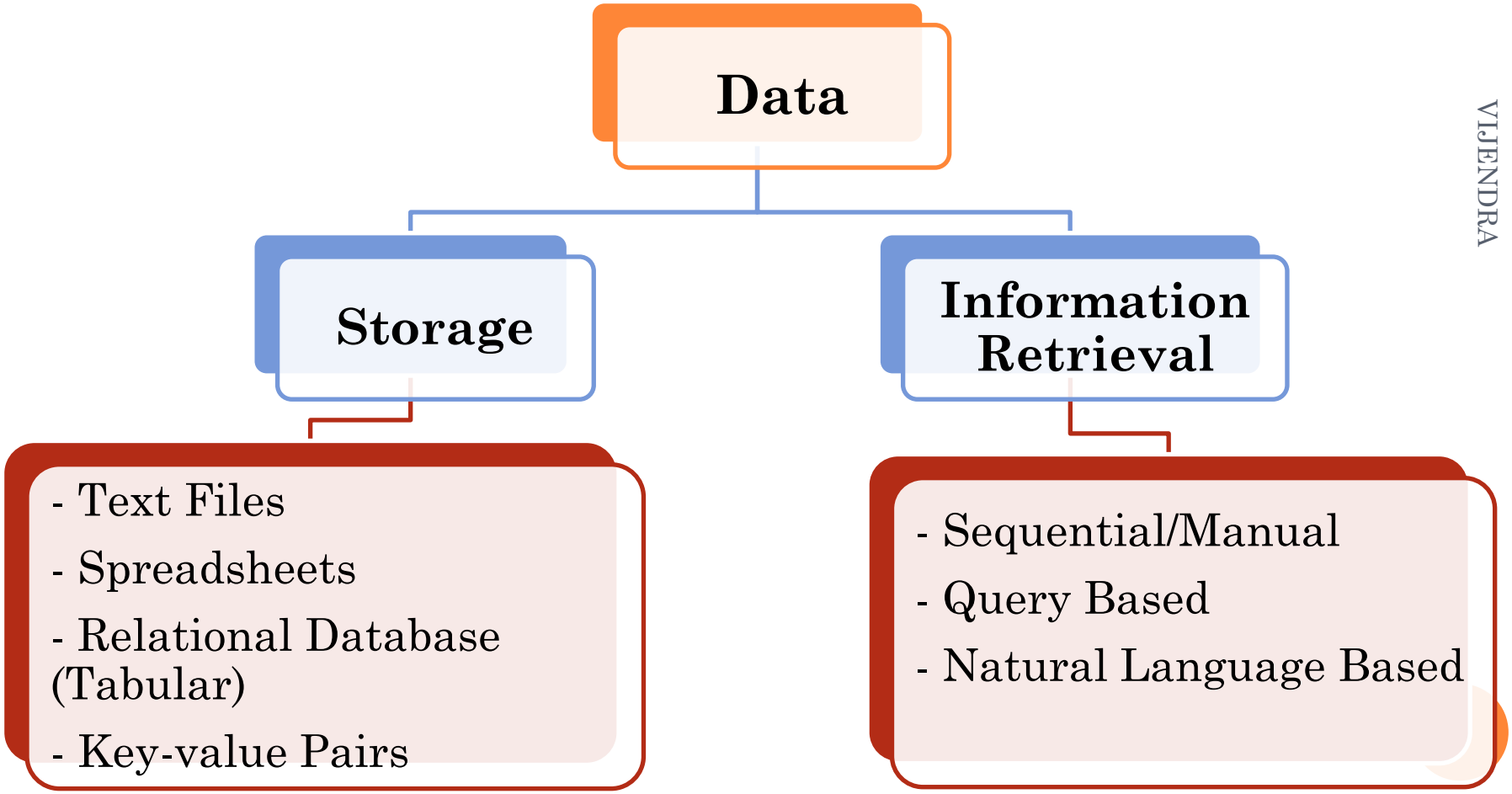
Types of Data – Based on Domain



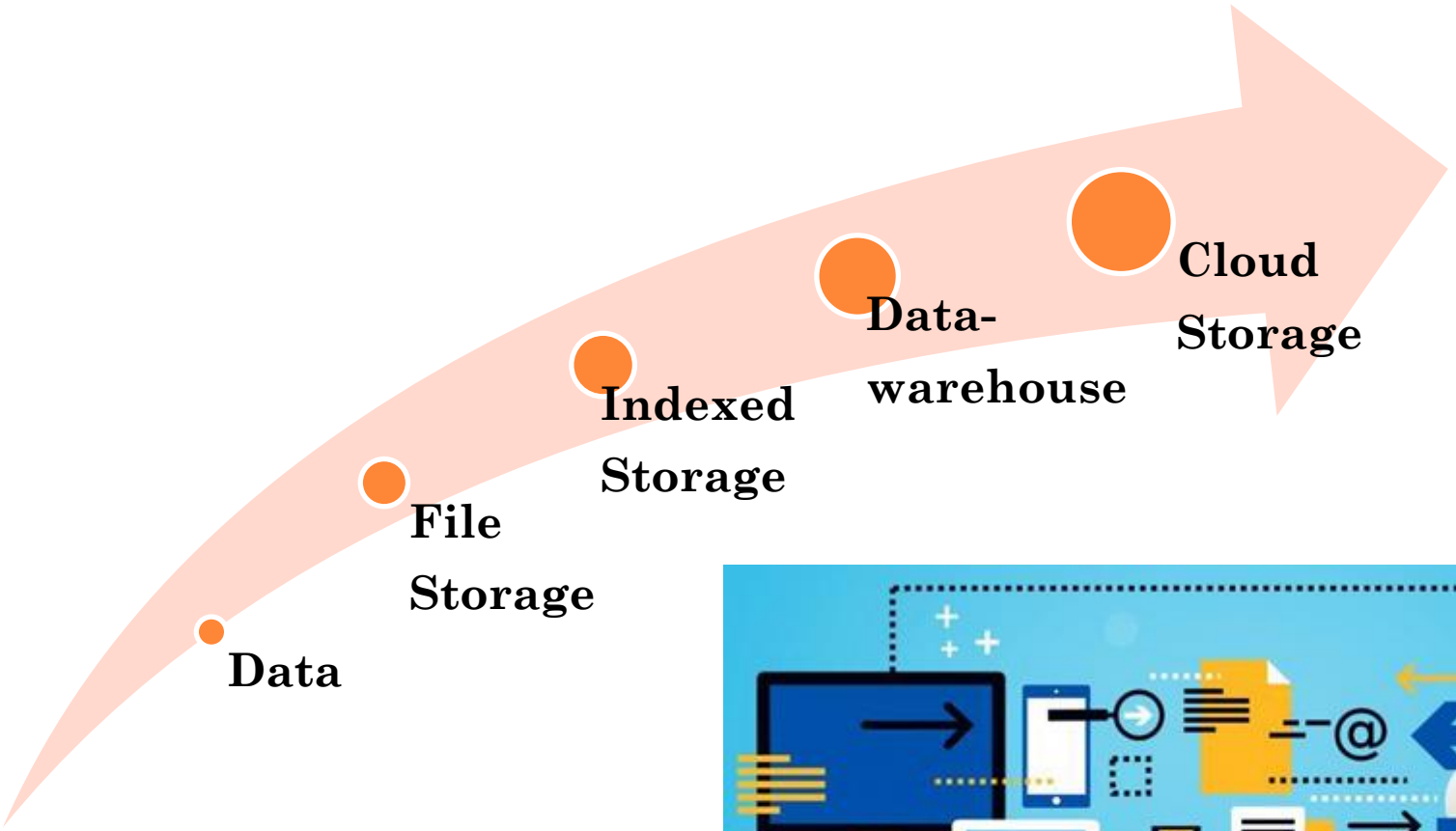
Handling Tremendous Data

- Handling this tremendous amount of data is overwhelming.
- Thus, we need some automated mechanism to perform this task.

Aspects of Data



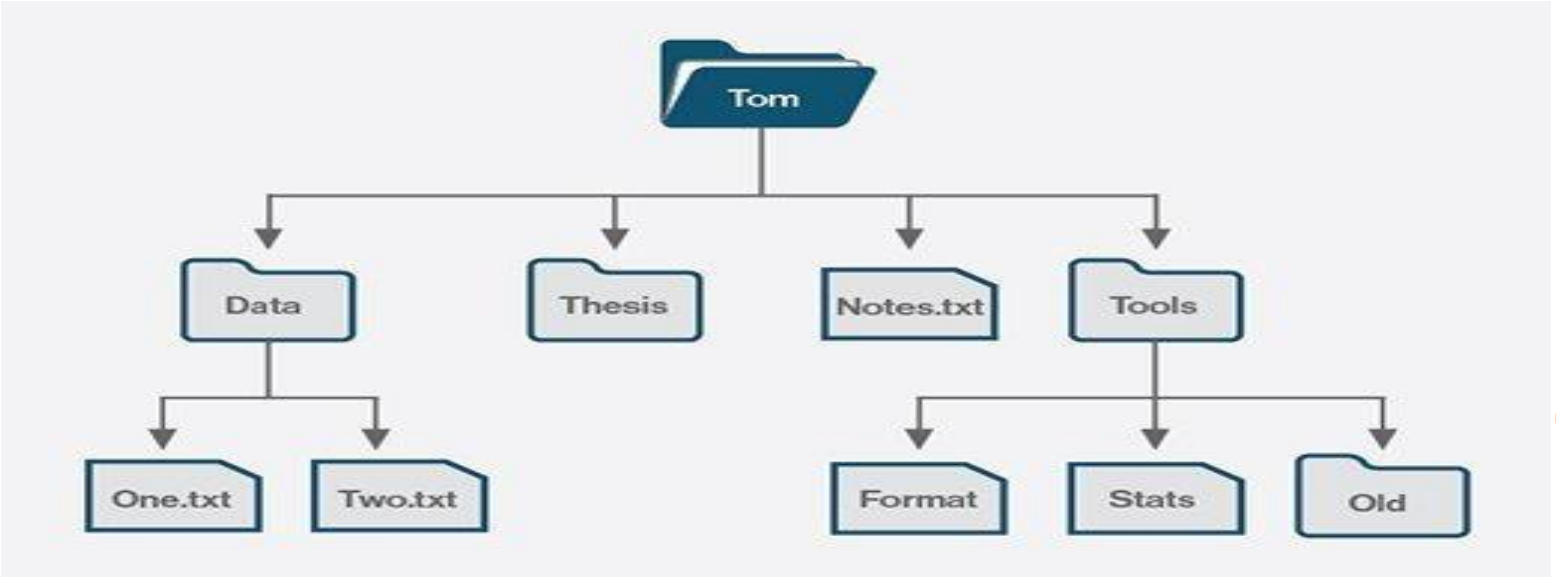
Evolution of Data Storage



VIJENDRA

File Storage

Alternatively referred to as a flat database or text database, a flat file is a file of data that does not contain links to other files or is a non-relational database. A good example of a flat file is a single text-only file that contains all the data needed for a program that is often separated by some kind of delimiter.



File Storage…

- File handling programs like reading contents of a text file, writing student records in a file are examples of traditional File Storage.

- **Problems with File 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
  - Security problems

What is a Database?

- Collection of related data
- *Example: Employee Information*

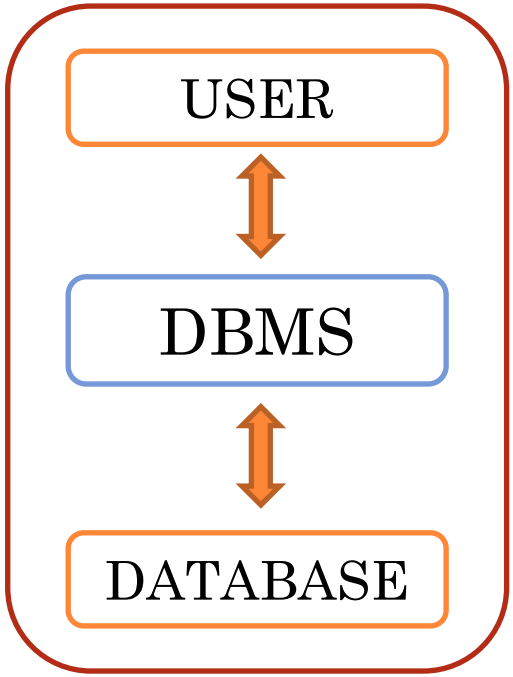
EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

Database Management Systems

- DBMS contains information about a particular enterprise
  - Collection of interrelated data
  - Set of programs to create and maintain a database
  - An environment that is both *convenient* and *efficient* to use

Database System



User does not interact directly with the database. DBMS acts as a interface between the user and the DB.

Levels of Abstraction

- **External Level:** application programs hide details of data types. Views can also hide information (such as an employee's salary) for security purposes.

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

Accounts View

Fname	Lname	Ssn
John	Smith	333445555
Franklin	Wong	888665555
…	…	…

Manager View

Fname	Lname	Dno
John	Smith	5
Franklin	Wong	5
…	…	…

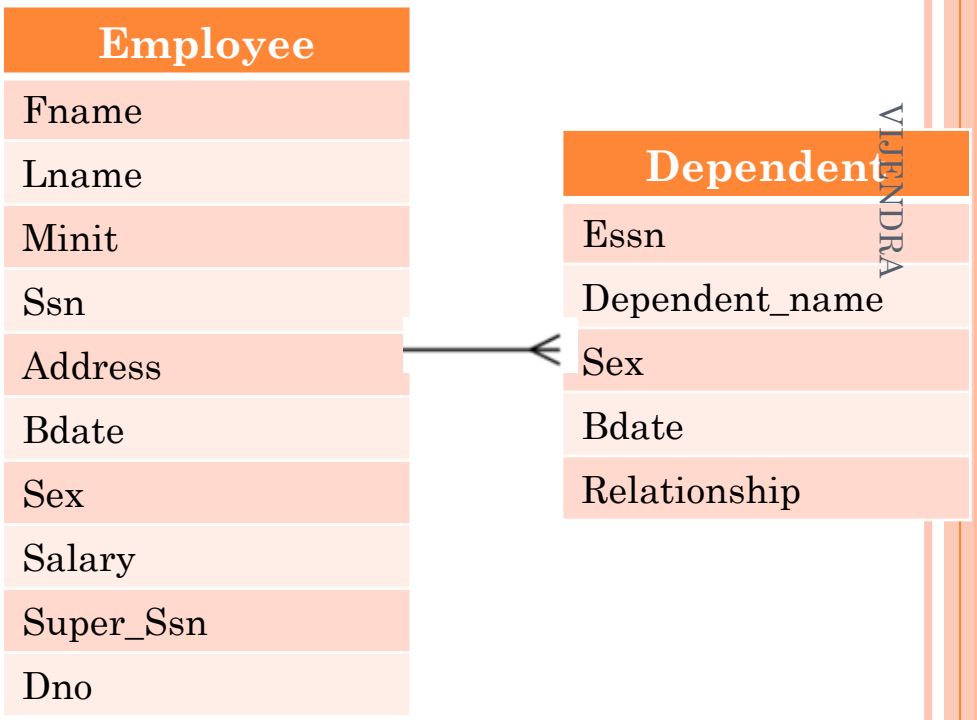
Levels of Abstraction

- **Logical level:** describes data stored in database, and the relationships among the data.

- **Physical level:** describes how a record

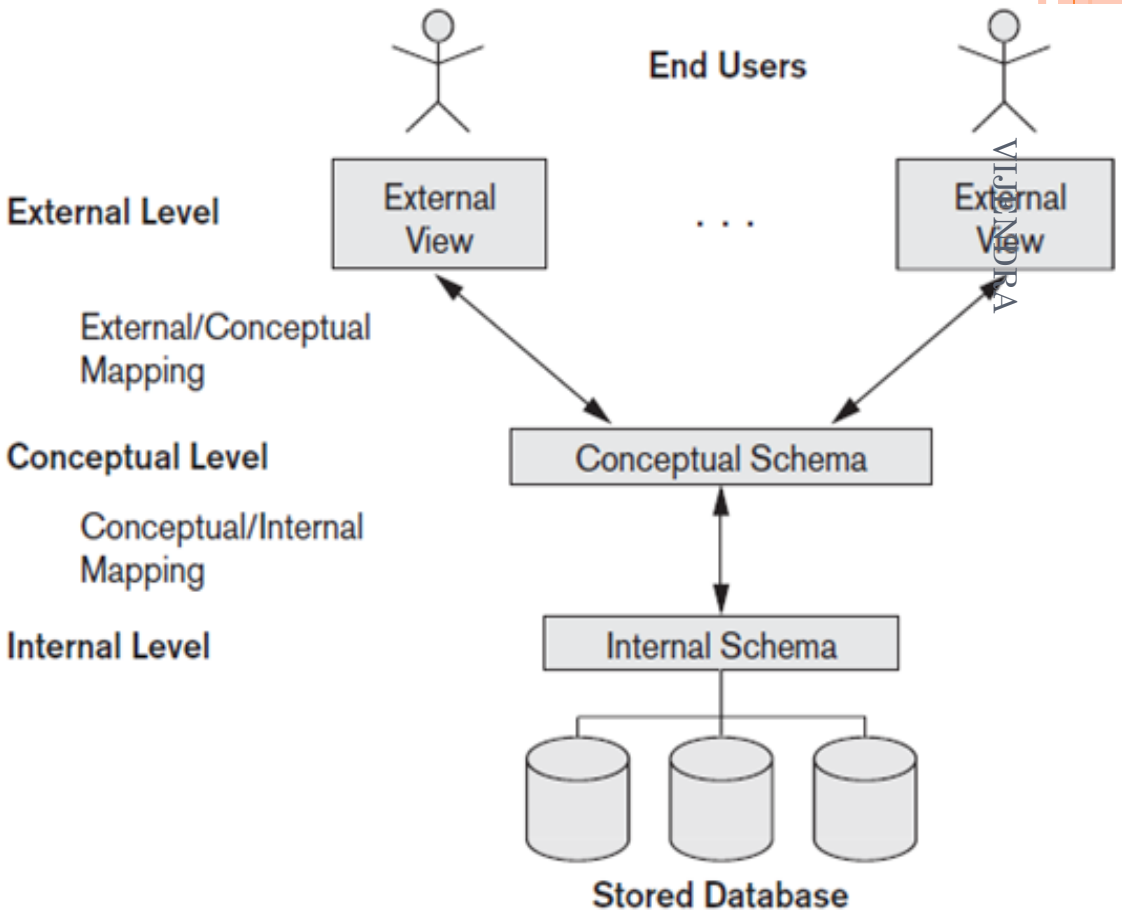
(e.g., customer) is stored.

```
type          Employee = record
                Fname : string;
                Lname : string;
                Address : string;
                Ssn : integer;
                Bdate: date;
                Sex: char
              end;
```

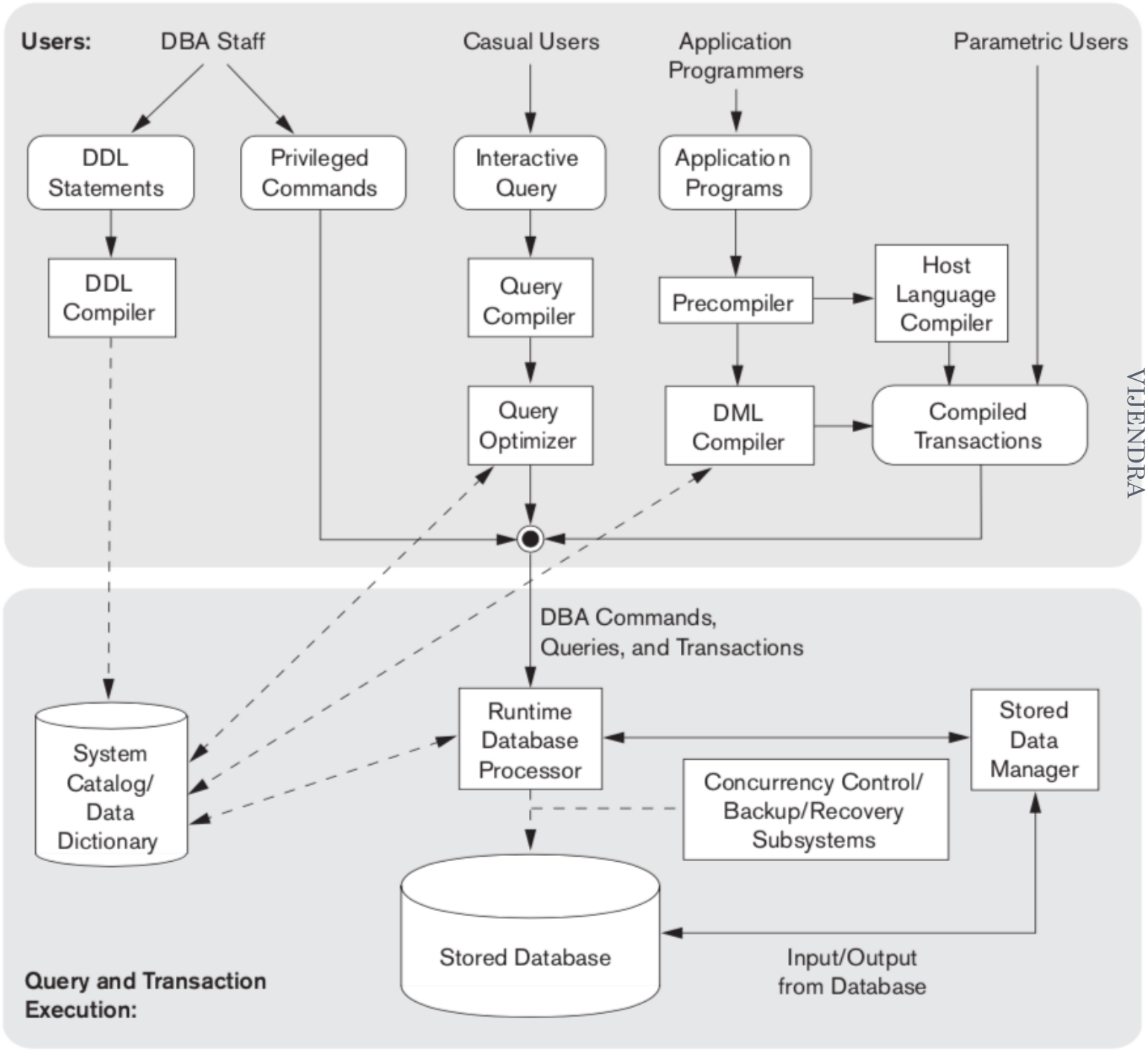


THREE SCHEMA ARCHITECTURE

- **External/view Level** → number of external user views. Each external schema describes the part of the database that a particular user group is interested in.
- **Conceptual Level Schema** → structure (entities, data types, relationships, user operations, constraints) of whole database for a community of users.
- **Internal Level Schema** → physical storage structure of database ie. complete details of data storage and access paths for the database.



Component Modules of a DBMS



VIJENDRA

RELATIONAL DBMS

- Integrity Constraints of RDMS → Tabular Form

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

Record/ Tuple

Attribute/Field

No. of Records = Cardinality **No. of Attributes = Arity**

Relational Schema v/s Instance

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

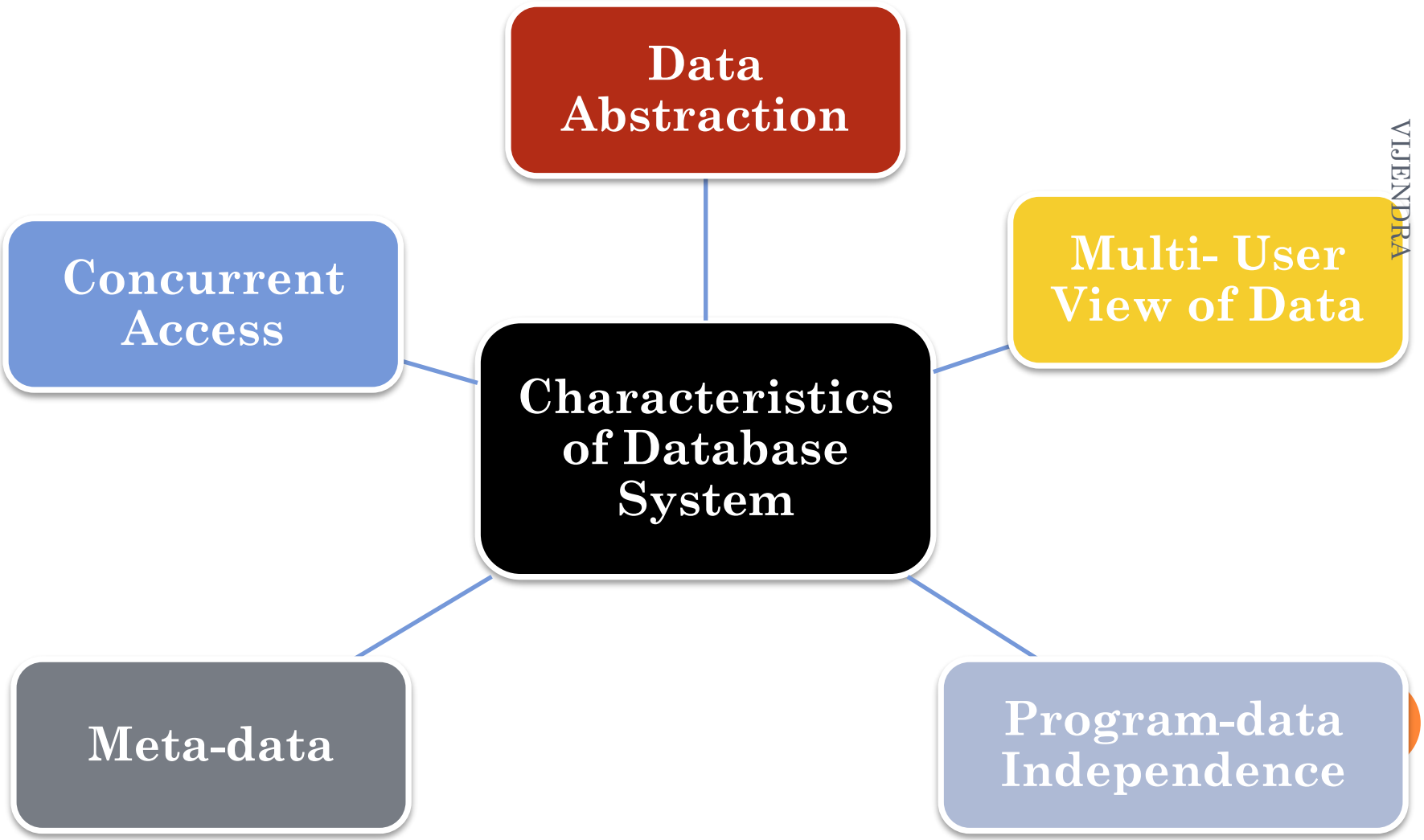
- **Relational Schema:** definition/structure of DB Table

  Example:

  EMPLOYEE (Fname, Minit, Lname, Ssn, Bdate, Address, Sex, Salary, Super_ssn, Dno)

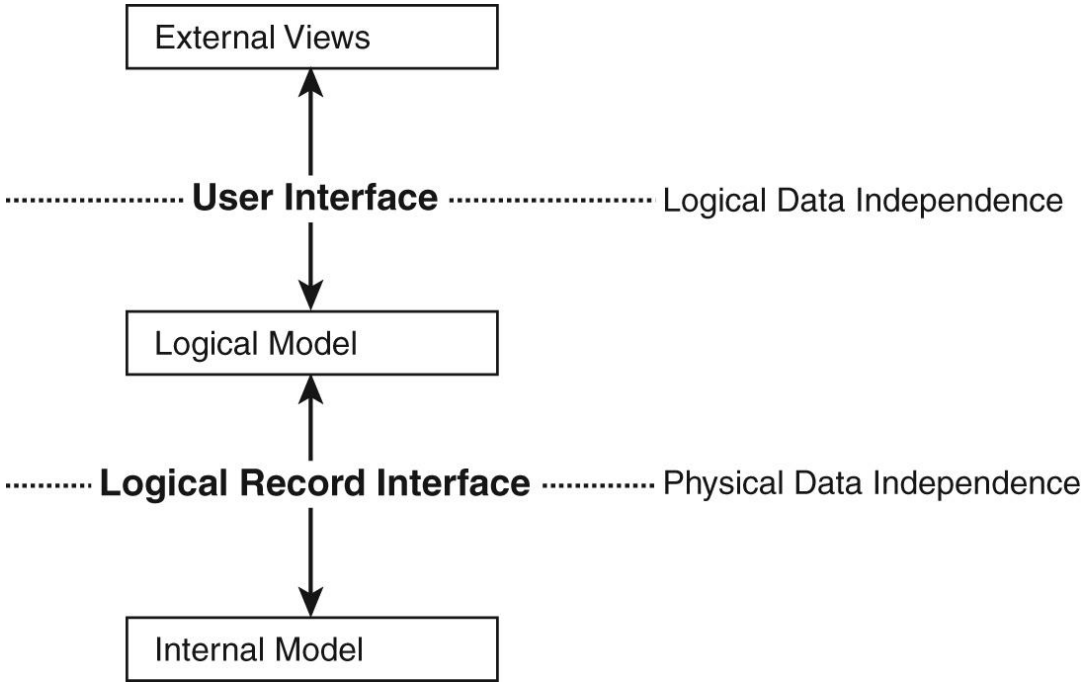
- **Relational Instance:** set of records of relation at the point of time

Characteristics of Database Systems



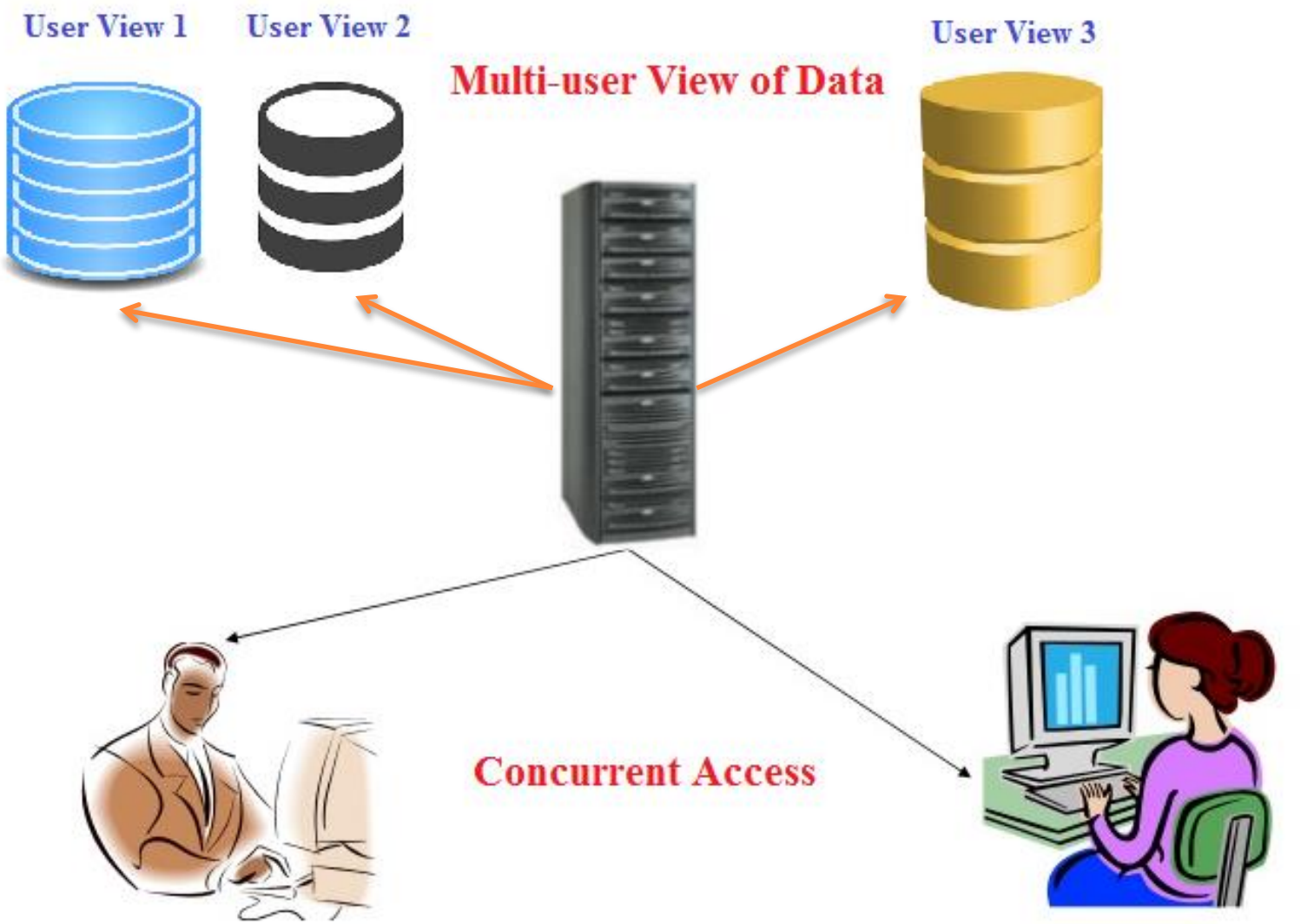
Data Abstraction & Program-Data Independence

- **Data Abstraction:** A **data model** is used to hide storage details and present the users with a *conceptual view* of the database.



- **Program-data Independence**: Allows changing data storage structures and operations without having to change the DBMS access programs.

Concurrent Access & Multi-user View of Data



Main Characteristics of the Database Approach

- **Self-describing nature of a database system:**
  - A DBMS **catalog** stores the *description* of a particular database (e.g. data structures, types, and constraints)
  - The description is called **meta-data** (see next slide).
  - This allows the DBMS software to work with *different* database applications (university, bank, airlines, etc.)
- **Insulation between programs and data:**
  - Called **program-data independence**.
  - Allows changing data structures and data storage organization without having to change the DBMS access programs.

EXAMPLE OF META-DATE IN 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

Note: Major_type is defined as an enumerared type with all known majors. XXXXNNNN is used to define a type with four alpha characters followed by four digits

Figure 1.3
An example of a database catalog for the database in Figure 1.2.

Main Characteristics of the Database Approach (cont.)

- **Insulation between programs and data (cont.):**
  - Accomplished through **data abstraction**
  - A **data model** is used to hide storage details and present the users with a conceptual view of the database.
  - Programs refer to the data model constructs rather than data storage details
- **Support of multiple views of the data:**
  - Each user may see a different view of the database, which describes **only** the data of interest to that user.

Main Characteristics of the Database Approach (cont.)

- **Sharing of data and multi-user transaction processing:**
  - Allowing a set of **user transactions** to access and update the database ***concurrently*** (at the same time).
  - *Concurrency control* within the DBMS guarantees that each **transaction** is correctly executed or aborted
  - *Recovery* subsystem ensures each completed transaction has its effect permanently recorded in the database
  - **OLTP** (Online Transaction Processing) is a major part of database applications (allows hundreds of concurrent transactions to execute per second)

Advantages of Using the Database Approach

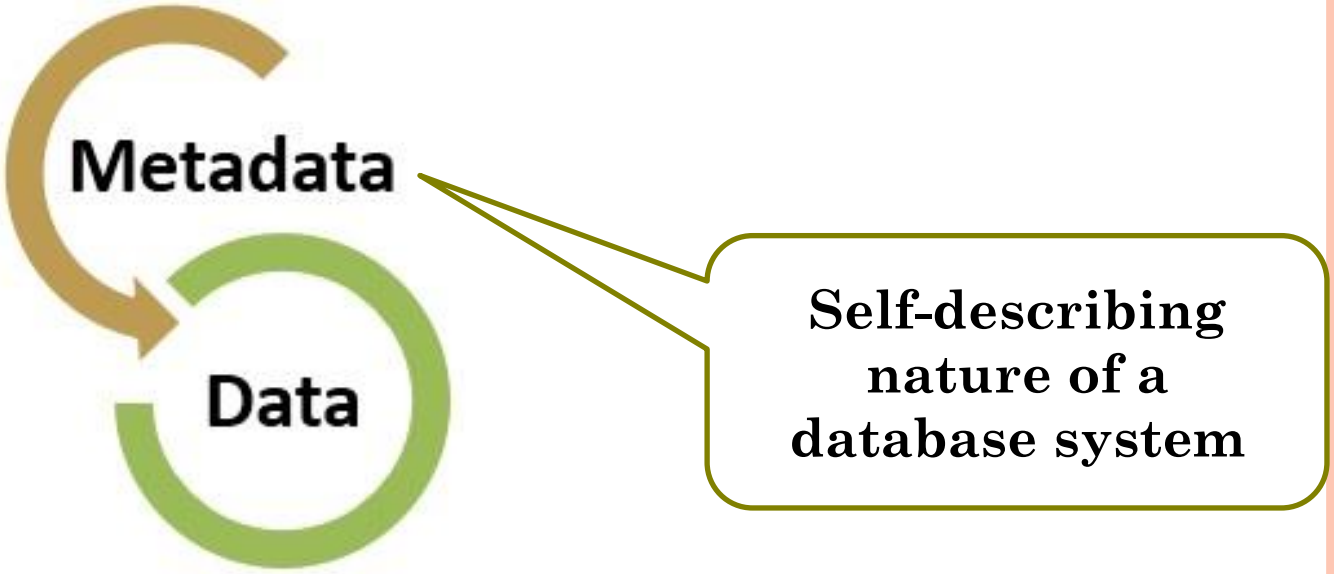
- Controlling redundancy in data storage and in development and maintenance efforts.
  - Sharing of data among multiple users.
- Restricting unauthorized access to data.
- Providing persistent storage for program Objects
  - In Object-oriented DBMSs
- Providing Storage Structures (e.g. indexes) for efficient Query Processing

Advantages of Using the Database Approach (cont.)

- Providing backup and recovery services.
- Providing multiple interfaces to different classes of users.
- Representing complex relationships among data.
- Enforcing integrity constraints on the database.
- Drawing inferences and actions from the stored data using deductive and active rules
- Allowing multiple “views” of the same data

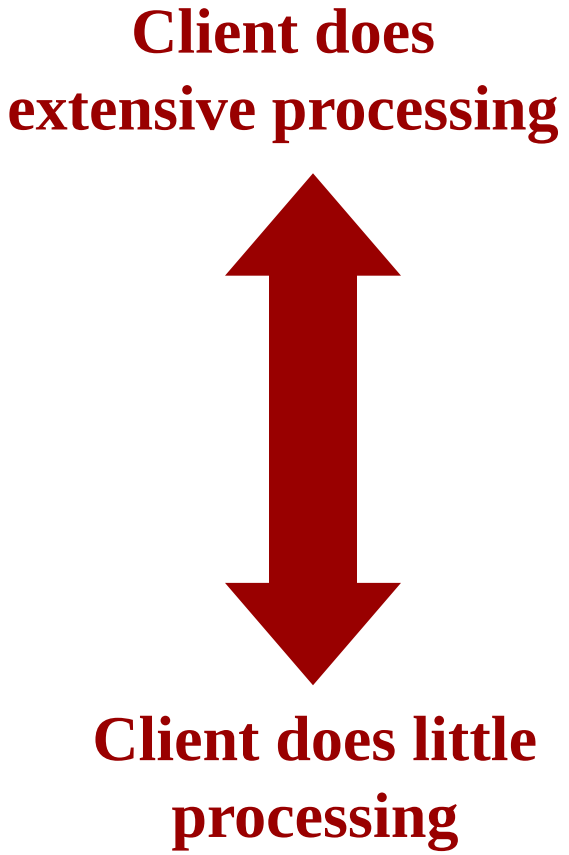
META-DATA

- **Meta-data:** A DBMS **catalog** stores the *description* of the database. This description is called meta-data. This allows the DBMS software to work with different databases.



Client/Server Architectures

- **Two-tier Architecture**
- **Three-tier Architecture**

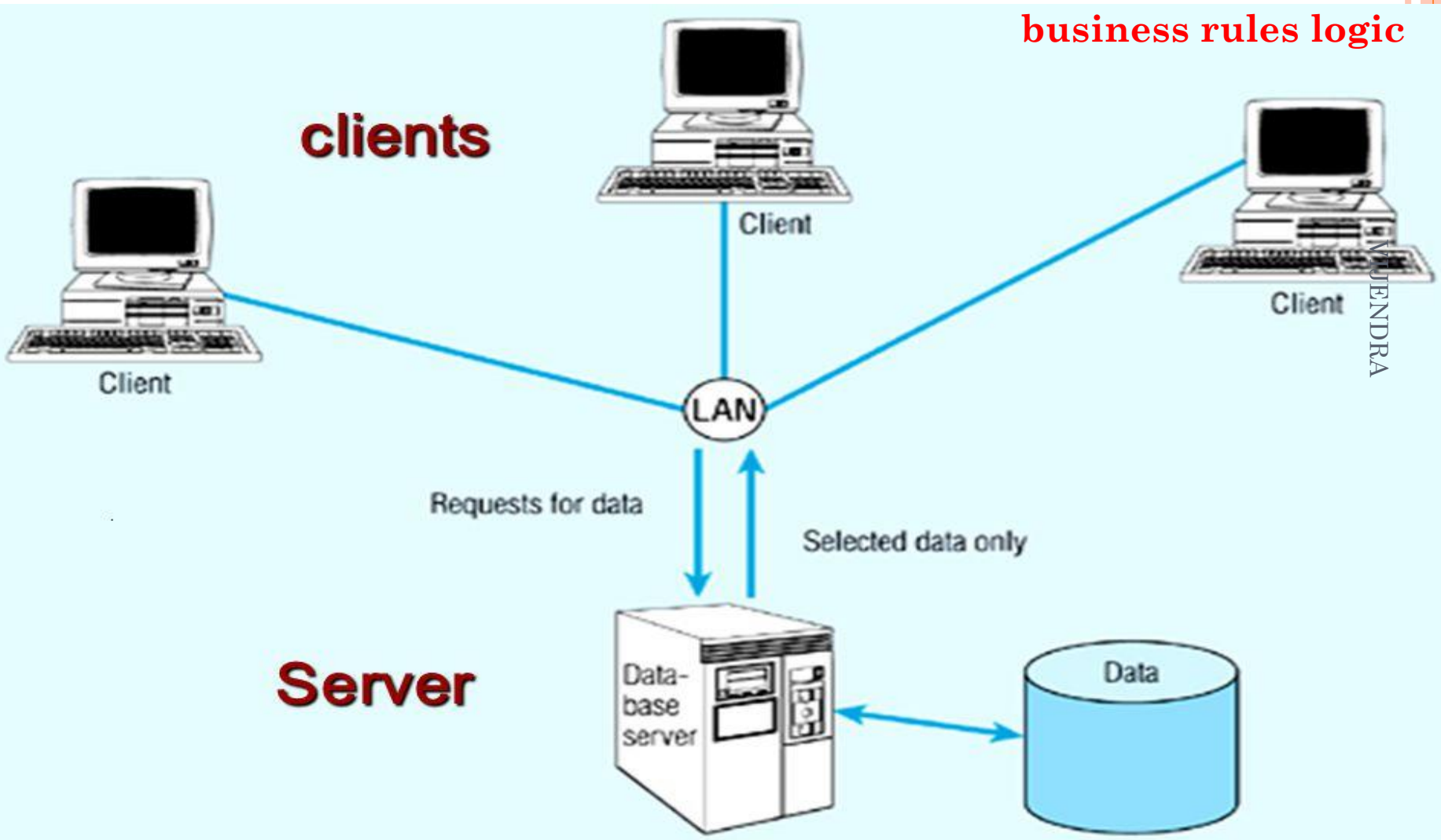


Application Logic in C/S Systems

Presentation Logic Input – keyboard/mouse Output – monitor/printer	**GUI Interface**
Processing Logic I/O processing Business rules Data management	**Procedures, functions, programs**
Storage Logic Data storage/retrieval	**DBMS Activities**

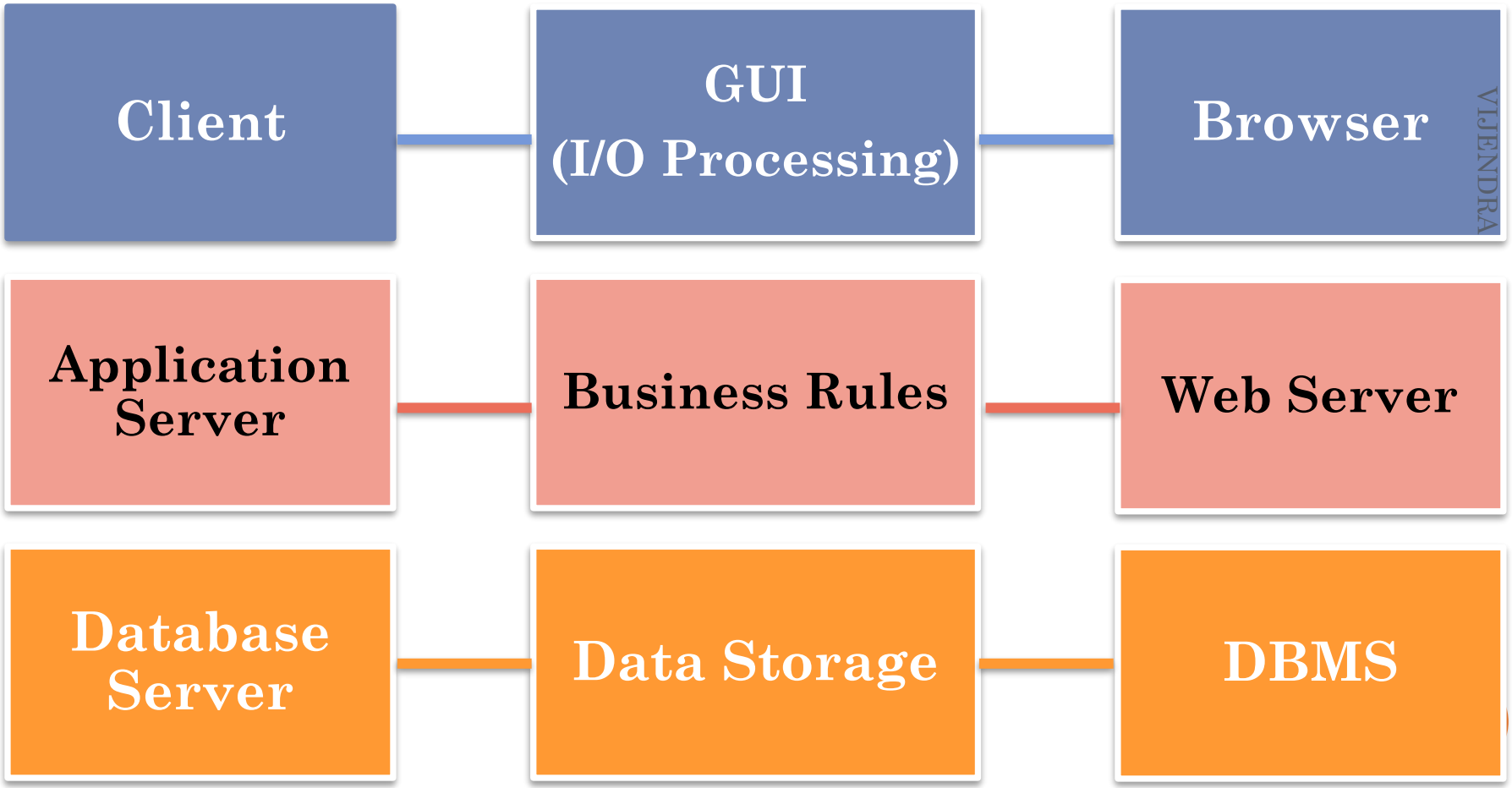
TWO-TIER ARCHITECTURE

Responsible for I/O processing and business rules logic

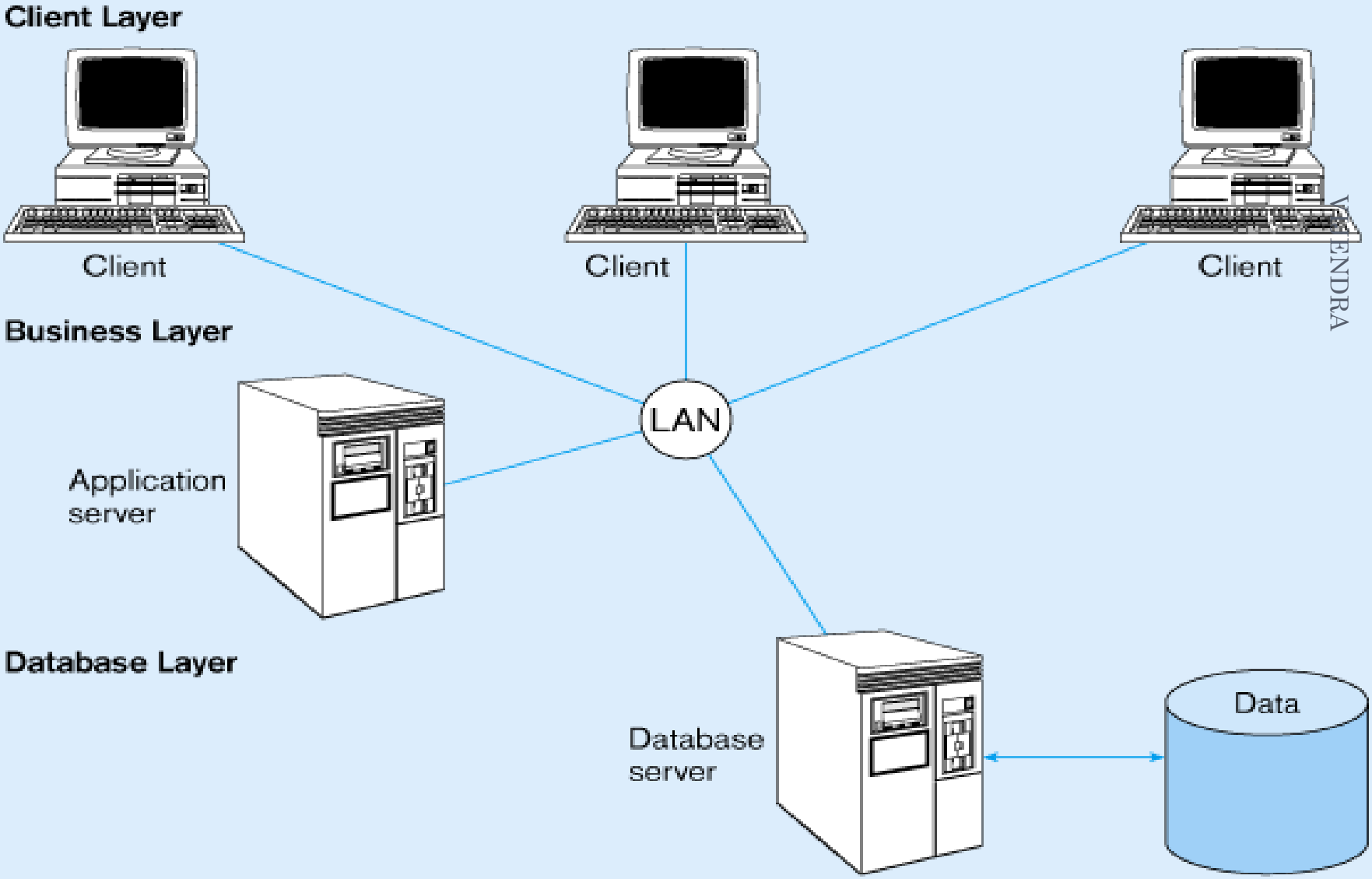


Server performs all data storage and access processing →DBMS is only on server

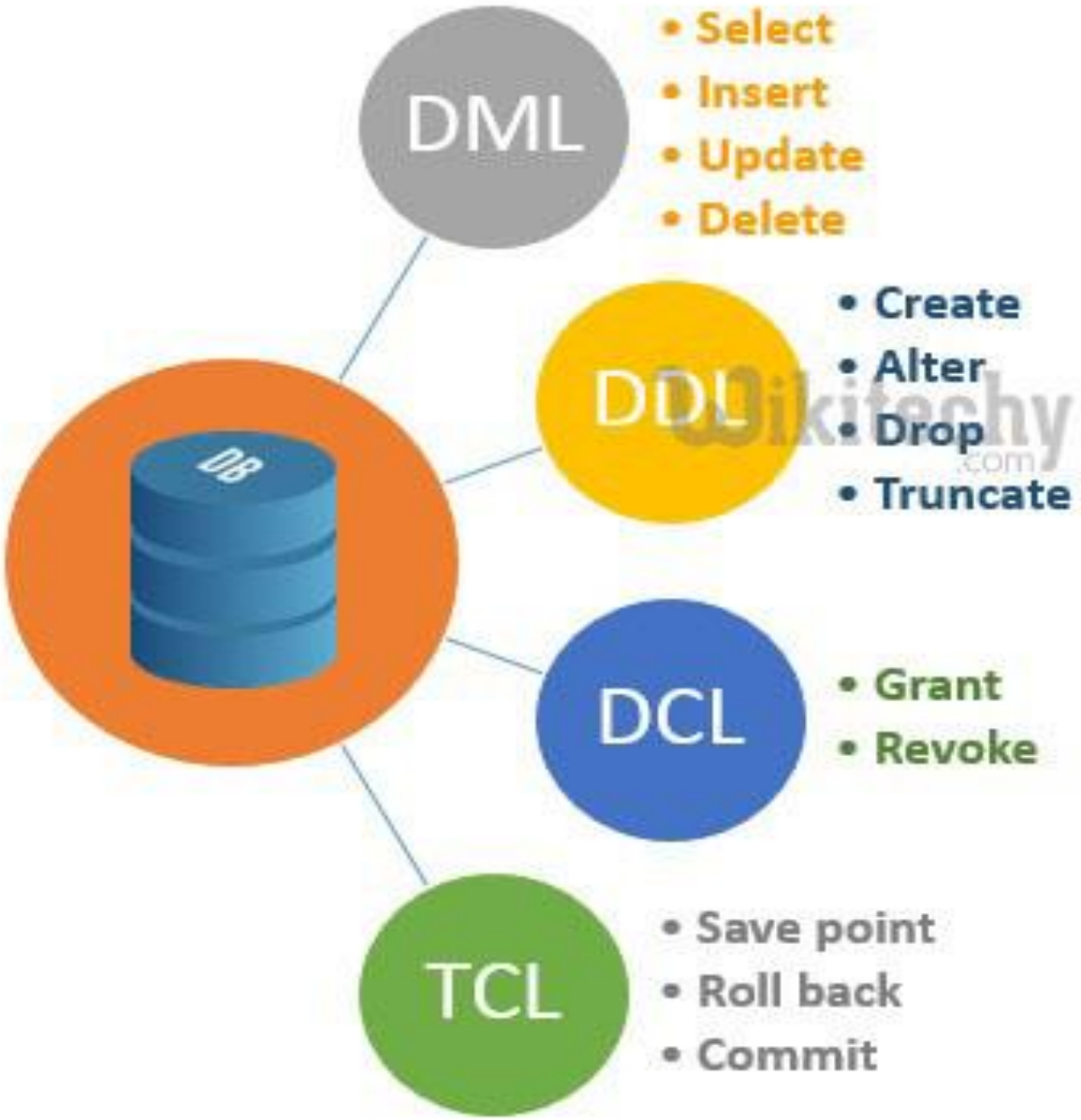
Three-tier Architecture



THREE-TIER ARCHITECTURE



Database Query Languages



VIJENDRA

Data Definition language (DDL)

DDL - define the database structure or schema

CREATE - to create objects in the database

ALTER - alters the structure of the database

DROP - delete objects from the database

Data Manipulation language (DML)

DML – manages data within schema objects

SELECT - retrieve data from the a database

INSERT - insert data into a table

UPDATE - updates existing data within a table

DELETE - deletes all records from a table, the space for the records remain

Data Control language (DCL)

DCL - used to control data access

GRANT - gives user's access privileges to database

REVOKE - withdraw access privileges given with the GRANT command

Transaction Control Language (TCL)

TCL - used to manage the changes made by DML. It allows statements to be grouped together into logical transactions.

COMMIT - save work done

SAVEPOINT - identify a point in a transaction to which you can later roll back

ROLLBACK - restore database to original since the last COMMIT rollback segment to use