Pune Vidyarthi Griha's

College of Engineering, Nashik

206, Behind Reliance Petrol Pump, Dindoriroad, Mhasrul, Nashik-422004
Phone: 0253-6480000 / 36 /44 Toll Free Number: 18002665330
Website: https://pvgcoenashik.org Email: admission@pvgcoenashik.org

Approved by AICTE, New Delhi, DTE, Mumbai and
Affiliated to Savitribai Phule Pune University, Pune.

DTE Code: EN5330

“ Database Management System”

By,
Prof. Anand N. Gharu
Asst. Professor
Computer Department,
PVG COE, Nashik.

Introduction

- Atabse : https://youtu.be/d11viALaCvA
- Intro : https://youtu.be/fSWAkJz_huQ
- https://youtu.be/r8u8JuM0450?list=RDCMUC6E97LDJTFJgzWU7G3CHILw
- https://youtu.be/FR4QIeZaPeM
- https://youtu.be/fFi_0HVgLrQ

Introduction

Data : Data is a collection of facts, such as numbers, words, measurements, observations or just descriptions of things.

Database : database is an organized collection of structured information, or data, typically stored electronically in a computer system. A database is usually controlled by a database management system (DBMS)

Management : Management allows a person to organize, store and retrieve data from a computer.

System : system is a set of rules, an arrangement of things, or a group of related things that work toward a common goal.

Files : A file is a collection of data stored in one unit, identified by a filename. It can be a document, picture, audio or video stream, data library, application, or other collection of data.

Introduction

File Structure : File Structures is the Organization of Data in Secondary Storage Device in such a way that minimize the access time and the storage space.

Files Organization : File organization refers to the way records are physically arranged on a storage device.

Fields : A database field is a single piece of information from a record.

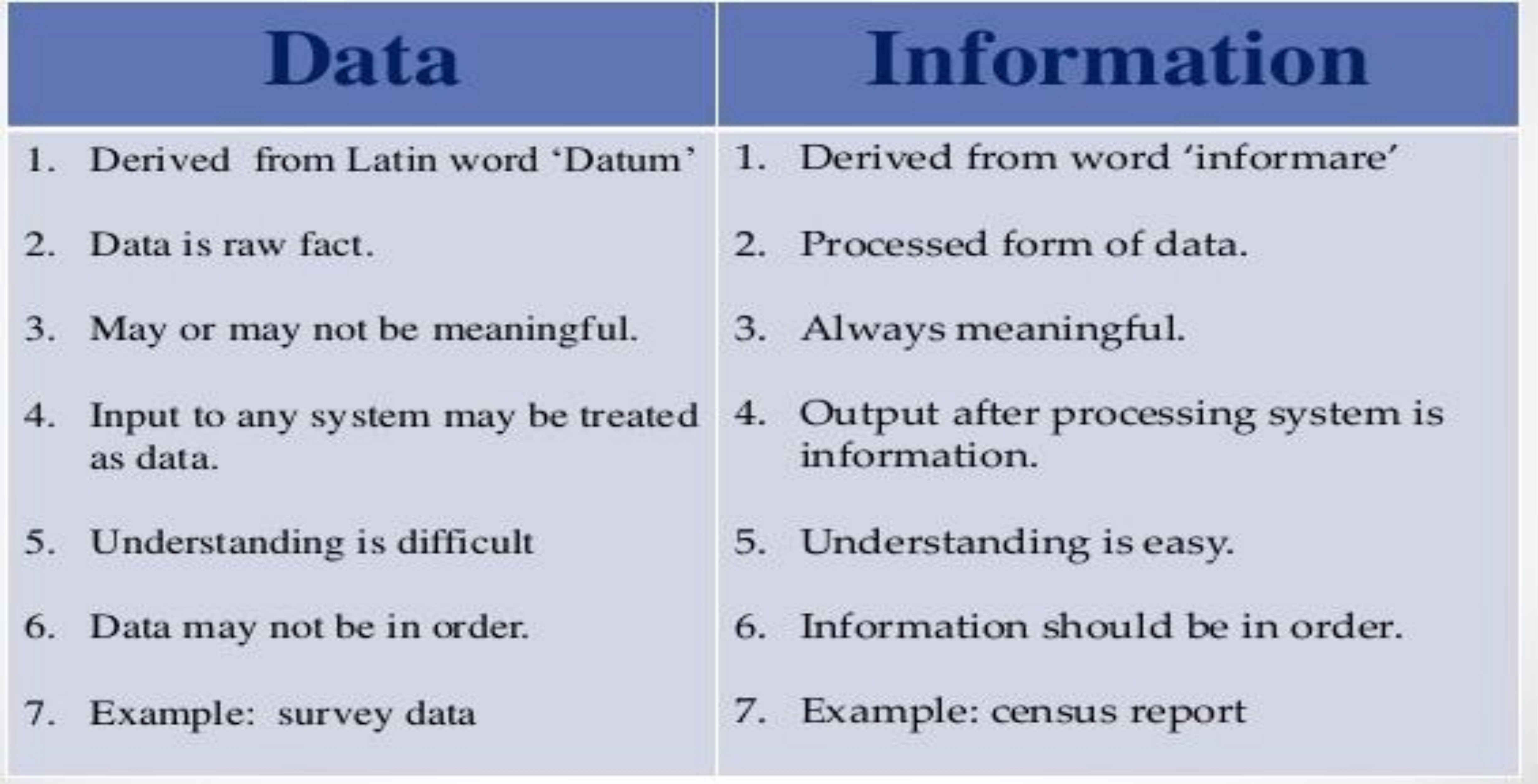
Record : A record is composed of fields and contains all the data about one particular person, company, or item in a database.

Physical Database : Physical database design translates the logical data model into a set of SQL statements that define the database.

Logical Database: logical database must be able to access and identify all files within the storage system to operate correctly.

DBMS

Data Vs Information



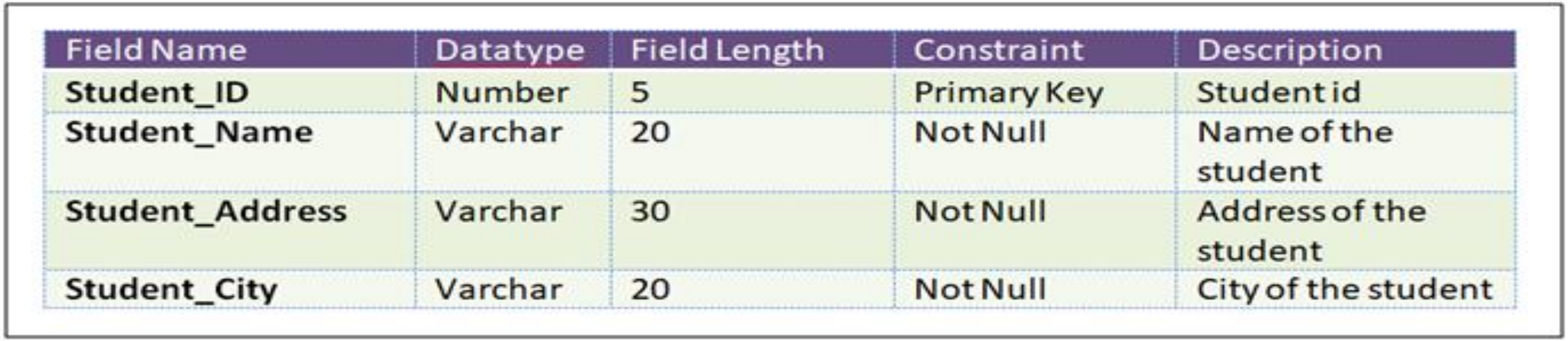
Data	Information
1. Derived from Latin word ‘Datum’	1. Derived from word ‘informare’
2. Data is raw fact.	2. Processed form of data.
3. May or may not be meaningful.	3. Always meaningful.
4. Input to any system may be treated as data.	4. Output after processing system is information.
5. Understanding is difficult	5. Understanding is easy.
6. Data may not be in order.	6. Information should be in order.
7. Example: survey data	7. Example: census report

Data Dictionary in Database

Data Dictionary :

“Data Dictionary consists of database metadata. It has records about objects in the database.”

- Data Dictionary can be defined as a DBMS component which stores the definition of characteristics of data and relationships.
- This "**data about data**" are labeled as **metadata.**
- Data Dictionary provides the DBMS with its self-describing characteristic.



Field Name	Datatype	Field Length	Constraint	Description
Student_ID	Number	5	Primary Key	Student id
Student_Name	Varchar	20	Not Null	Name of the student
Student_Address	Varchar	30	Not Null	Address of the student
Student_City	Varchar	20	Not Null	City of the student

Data Item or Field and Record

Fields : A database field is a single piece of information from a record.

Example : Roll_no

Record : A record is composed of fields and contains all the data about one particular person, company, or item in a database

Example : anand , gharu is name of rollno = 10

Definition of DBMS

Data : Data is a collection of facts, such as numbers, words, measurements, observations or just descriptions of things.

Database : database is an organized collection of structured information, or data, typically stored electronically in a computer system. A database is usually controlled by a database management system (DBMS)

Management : Management allows a person to organize, store and retrieve data from a computer.

System : system is a set of rules, an arrangement of things, or a group of related things that work toward a common goal.

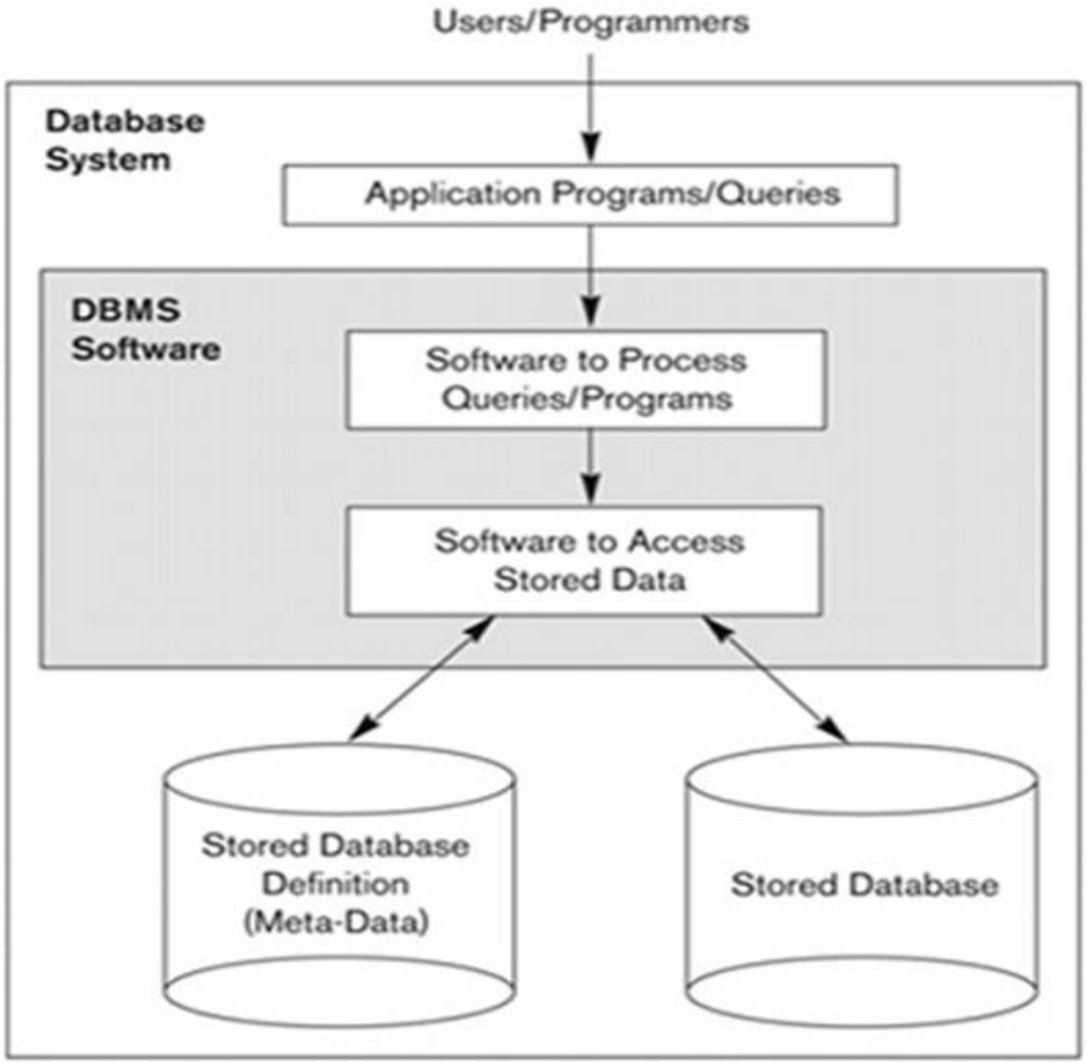
Definition of DBMS

DBMS : "Database management system is software that is used to manage the database."

example: MySQL, Oracle, IBM DB2, Microsoft Access etc

- DBMS provides an interface to perform various operations like database creation, storing data in it, updating data, creating a table in the database and a lot more.
- **The basic function of DBMS are :**

1. To store data in databse
2. To organize the dataTo control access of data
3. To Provide Security


Users/Programmers
Database System
Application Programs/Queries
DBMS Software
Software to Process Queries/Programs
Software to Access Stored Data
Stored Database Definition (Meta-Data)
Stored Database

DBMS Advantages

1. **Controls database redundancy:** It can control data redundancy because it stores all the data in one single database file and that recorded data is placed in the database.
2. **Data sharing:** In DBMS, the authorized users of an organization can share the data among multiple users.
3. **Easily Maintenance:** It can be easily maintainable due to the centralized nature of the database system.
4. **Reduce time:** It reduces development time and maintenance need.
5. **Backup:** It provides backup and recovery subsystems which create automatic backup of data from hardware and software failures and restores the data if required.
6. **Multiple user interface:** It provides different types of user interfaces like graphical user interfaces, application program interfaces

DBMS Disadvantages

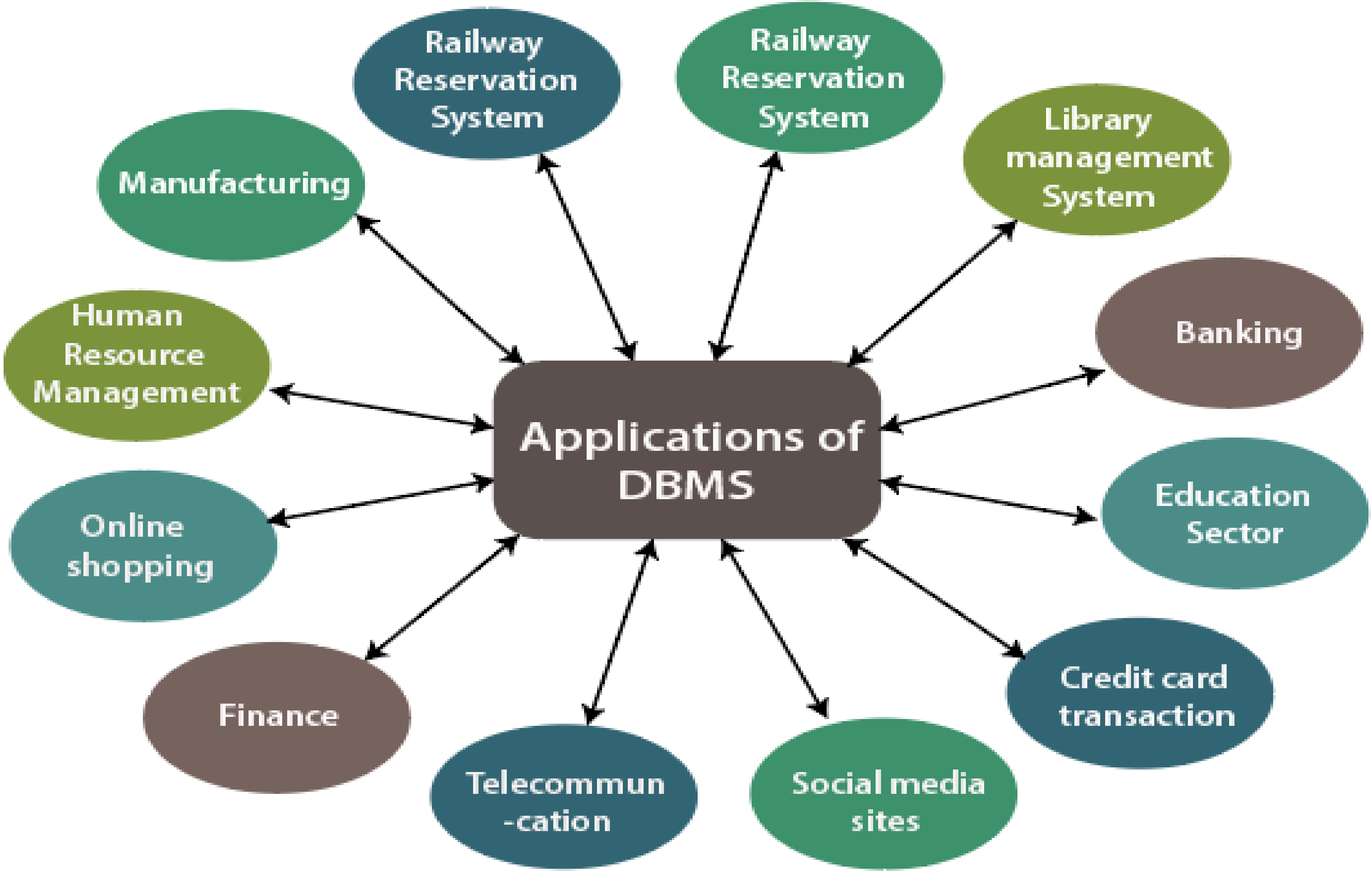
1. **Cost of Hardware and Software:** It requires a high speed of data processor and large memory size to run DBMS software.

2. **Size:** It occupies a large space of disks and large memory to run them efficiently.

3. **Complexity:** Database system creates additional complexity and requirements.

4. **Higher impact of failure:** Failure is highly impacted the database because in most of the organization, all the data stored in a single database and if the database is damaged due to electric failure or database corruption then the data may be lost forever.

Applications of DBMS



Applications of DBMS

1. **Railway Reservation System**: In the railway reservation system, the database is required to store the record or data of ticket bookings, status about train's arrival, and departure.
2. **Banking:** Database management system is used to store the transaction information of the customer in the database.
3. **Education Sector:** Presently, examinations are conducted online by many colleges and universities. They manage all examination data through the database management system (DBMS).
4. **Credit card transactions:** Database Management system is used for purchasing on credit cards and generation of monthly statements.
5. **Airline Reservation System:** This system is the same as the railway reservation system. This system also uses a database management system to store the records of flights departure, arrival, and delay status.
6. **Finance:** The database management system is used for storing information about sales, holding and purchases of financial instruments such as stocks and bonds in a database.

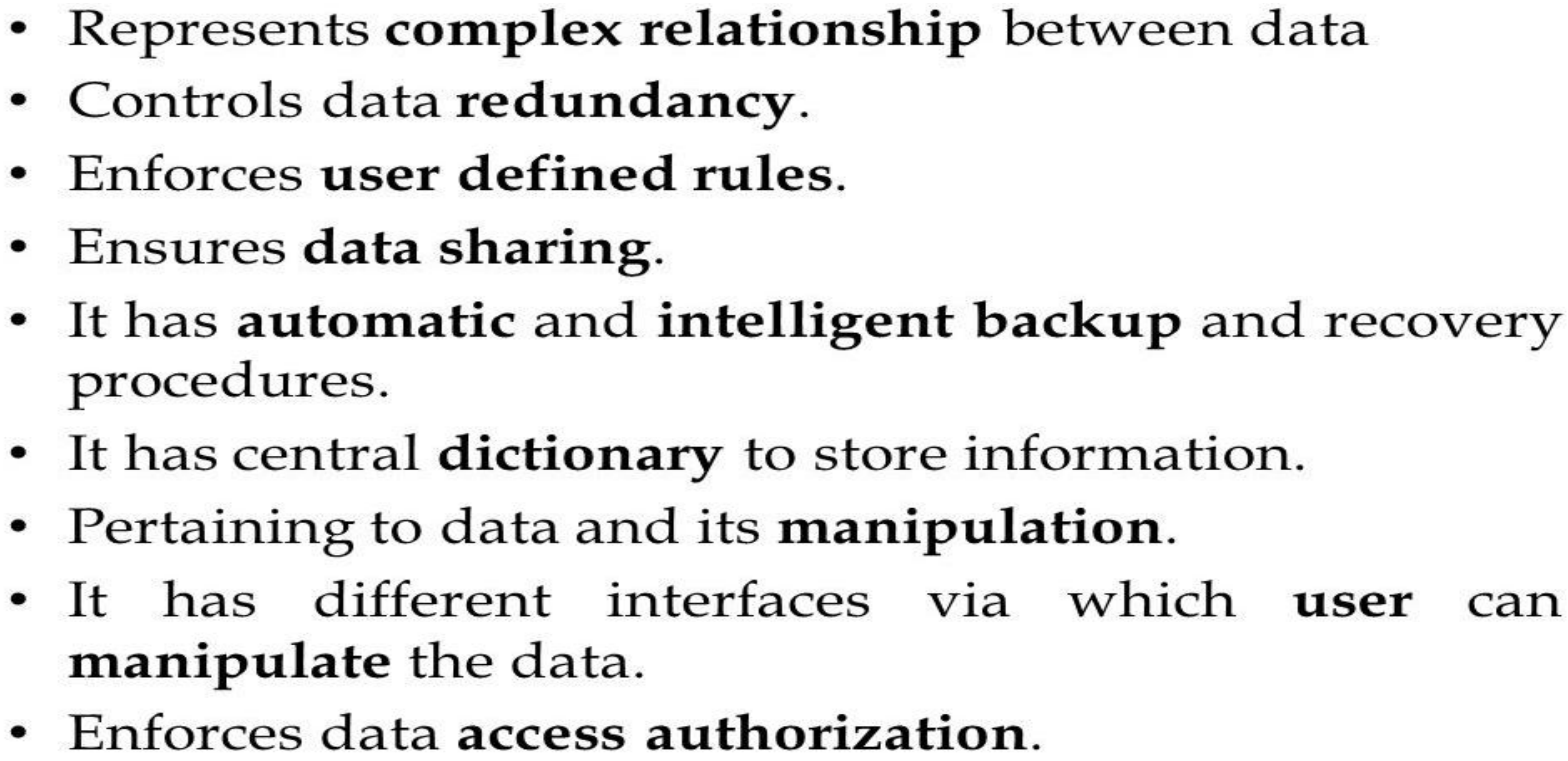
File Processing System Vs DBMS

File based system	Database system
1. It does not provid Security	**1.** It provide security.
2. data redundancy.	**2.** Database system control data redundancy.
3. The data cannot be shared	**3.** In database data is easily shared because data is stored at one place.
4. File based system is less complex system.	**4.** Database system is very complex system.
5. Data is isolated.	**5.** Data is integrated.
6. Data access takes more time	**6.** Data access takes less time.
7. File based system does not provide concurrency facility.	**7.** Database system provides concurrency facility.
8. Hardware cost is less than database system	**8.** Hardware cost is high in database than file system.
9. Single user system	**9.** Multi user System
10. Example : C++, Java	**10. Example :** Oracle, MySQL, IBM DB2.

Characteristics of Database Management System

1. Provides security and removes redundancy
2. Self-describing nature of a database system
3. Insulation between programs and data abstraction
4. Support of multiple views of the data
5. Sharing of data and multiuser transaction processing
6. DBMS allows entities and relations among them to form tables.
7. It follows the ACID concept (Atomicity, Consistency, Isolation, and Durability).
8. DBMS supports multi-user environment that allows users to access and manipulate data in parallel.

Characteristics of Database Management System



- Represents **complex relationship** between data
- Controls data **redundancy**.
- Enforces **user defined rules**.
- Ensures **data sharing**.
- It has **automatic** and **intelligent backup** and recovery procedures.
- It has central **dictionary** to store information.
- Pertaining to data and its **manipulation**.
- It has different interfaces via which **user** can **manipulate** the data.
- Enforces data **access authorization**.

Users in Database Management System

1. Database **Administrator** (DBA)
2. Naive / Parametric End Users
3. System Analyst
4. Sophisticated Users
5. Data Base Designers
6. Application Program
7. Casual Users / Temporary Users

Users in Database Management System

- **Database Administrator (DBA) :**
- Database Administrator (DBA) is a person/team who defines the schema and also controls the 3 levels of database.
- The DBA will then create a new account id and password for the user if he/she need to access the data base.
- DBA is also responsible for providing security to the data base and he allows only the authorized users to access/modify the data base.
- DBA also monitors the recovery and back up and provide technical support.
- The DBA has a DBA account in the DBMS which called a system or superuser account.
- DBA repairs damage caused due to hardware and/or software failures.
- DBA decides storage structure and access strategy

- Data dictionary is created by DBA.

Users in Database Management System

2. Naive / End Users :

Parametric End Users are the unsophisticated who don’t have any DBMS knowledge but they frequently use the data base applications in their daily life to get the desired results.

For examples, Railway’s ticket booking users are naive users. Clerks in any bank is a naive user because they don’t have any DBMS knowledge but they still use the database and perform their given task.

3. System Analyst:

System Analyst is a user who analyzes the requirements of end users. They check whether all the requirements of end users are satisfied.

Users in Database Management System

4. Sophisticated Users :

Sophisticated users can be engineers, scientists, business analyst, who are familiar with the database. They can develop their own data base applications according to their requirement. They don’t write the program code but they interact the data base by writing SQL queries directly through the query processor.

5. Data Base Designers :

Data Base Designers are the users who design the structure of data base which includes tables, indexes, views, constraints, triggers, stored procedures. He/she controls what data must be stored and how the data items to be related.

Users in Database Management System

6. Application Program :

Application Program are the back end programmers who writes the code for the application programs.They are the computer professionals. These programs could be written in Programming languages such as Visual Basic, Developer, C, FORTRAN, COBOL etc.

7. Casual Users / Temporary Users :

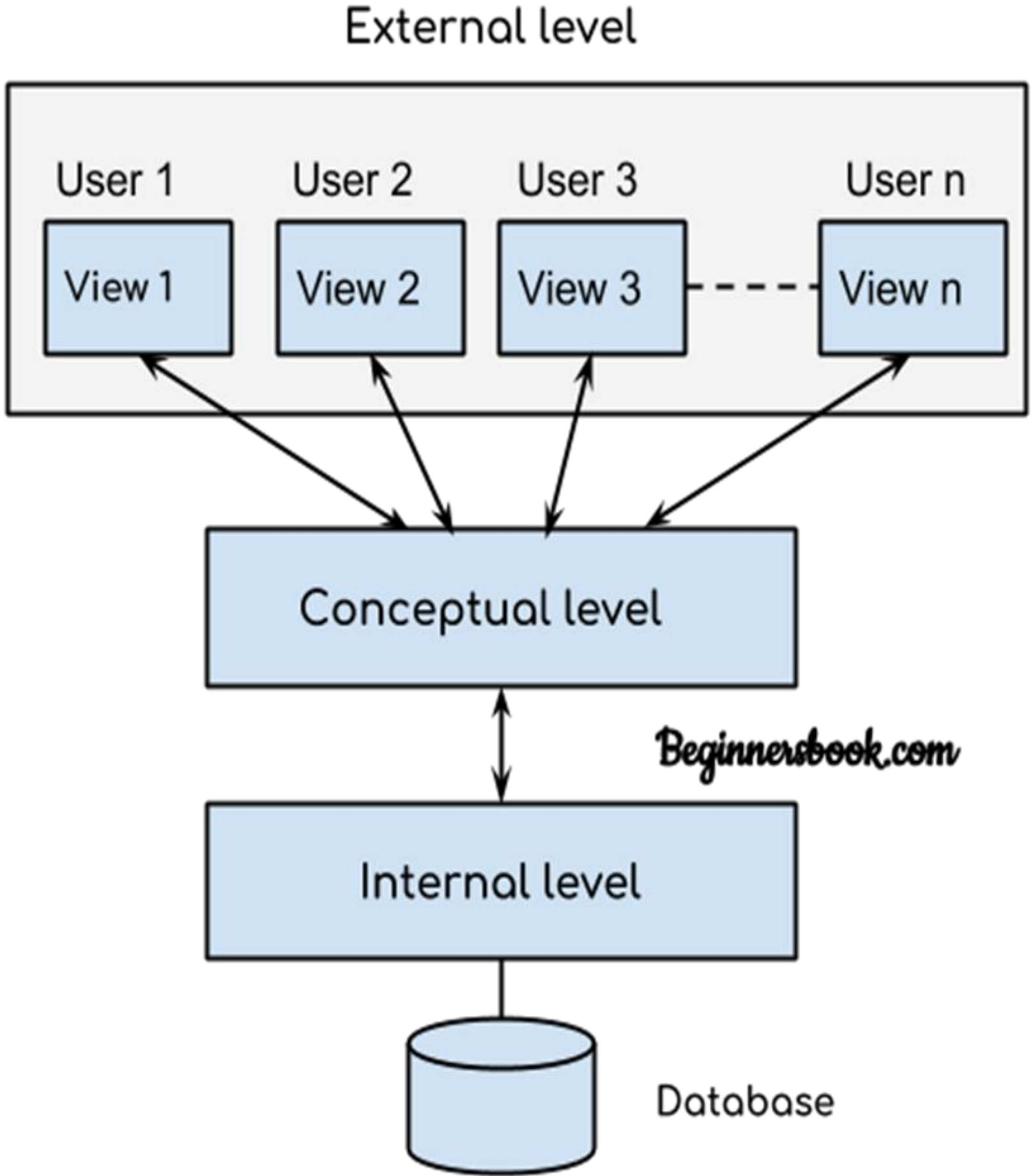
Casual Users are the users who occasionally use/access the data base but each time when they access the data base they require the new information, for example, Middle or higher level manager.

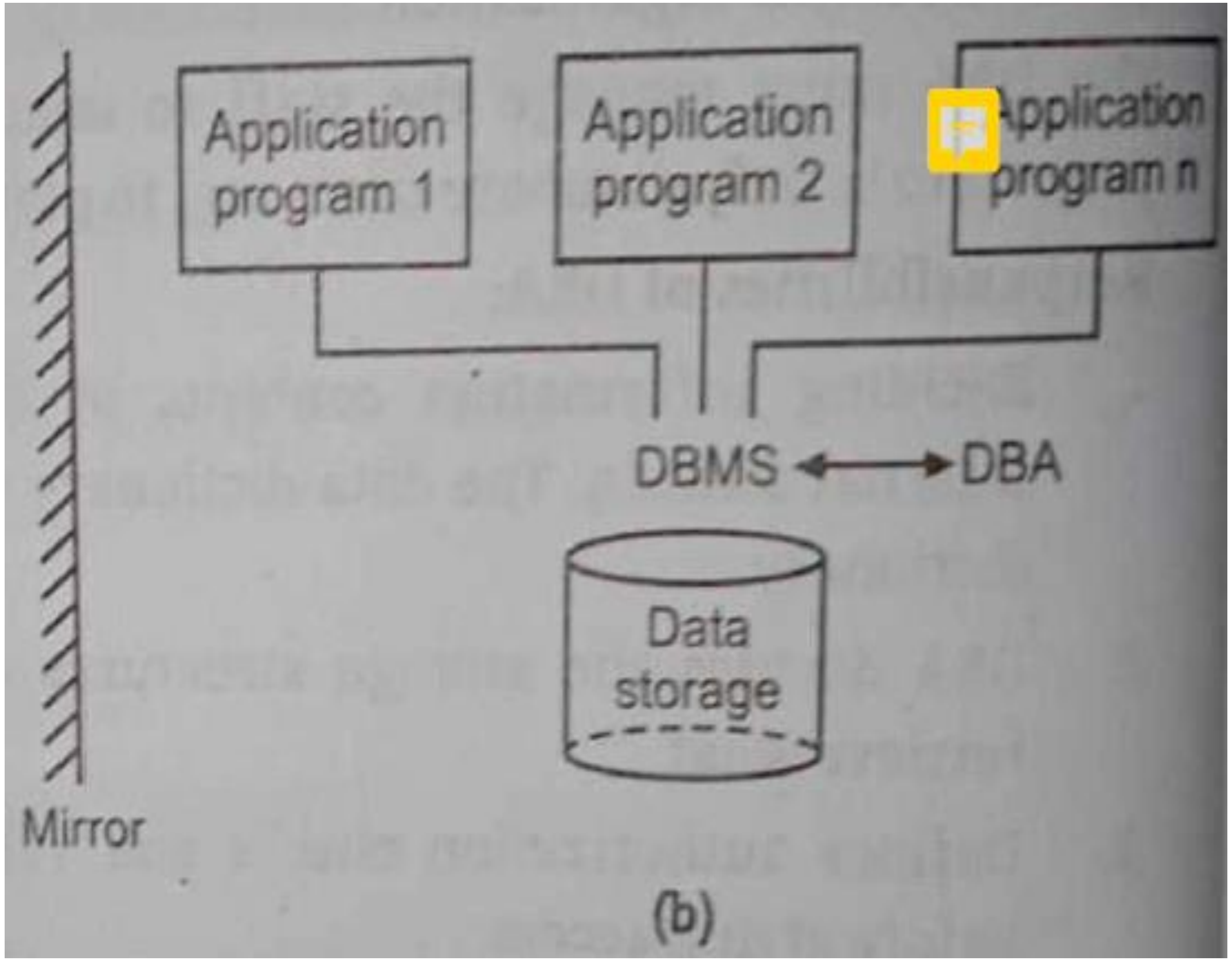
View in DBMS

View of Data in DBMS

1. **Abstraction** is one of the main features of database systems. Hiding irrelevant details from user and providing abstract view of data to users, helps in easy and efficient user-database interaction.
2. It is also called as three schema architecture.
3. The top level of that architecture is **"view level".**
4. The view level provides the **"view of data"** to the users and hides the irrelevant details such as data relationship, database schema, constraints, security etc from the user.
5. The three schema architecture is also used to separate the user applications and physical database.
6. The three schema architecture contains three-levels. It breaks the database down into three different categories

View of Data in DBMS





Architecture of DBMS

View of Data in DBMS

1. External level

- It is also called **view level**. The reason this level is called “view” is because several users can view their desired data from this level which is internally fetched from database with the help of conceptual and internal level mapping.
- The user doesn’t need to know the database schema details such as data structure, table definition etc.
- External level is the “top level” of the Three Level DBMS Architecture.

View of Data in DBMS

2. **Conceptual level**

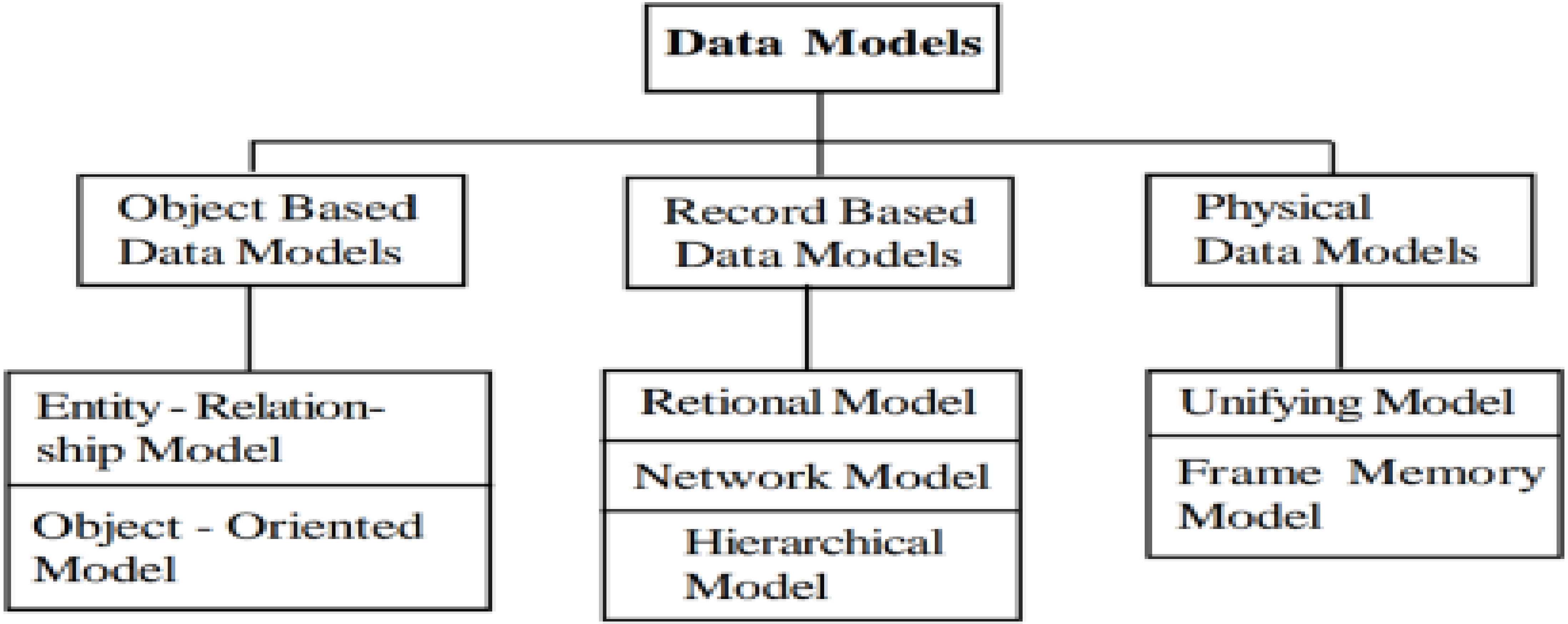
- It is also called logical level. The whole design of the database such as relationship among data, schema of data etc. are described in this level.
- Database constraints and security are also implemented in this level of architecture. This level is maintained by DBA (database administrator).

3. **Internal level**

- This level is also known as **physical level**. This level describes how the data is actually stored in the storage devices. This level is also responsible for allocating space to the data. This is the lowest level of the architecture.

Data Model in DBMS

"Data Model is a logical structure of Database. It describes the design of database to reflect entities, attributes, relationship among data, constrains etc."



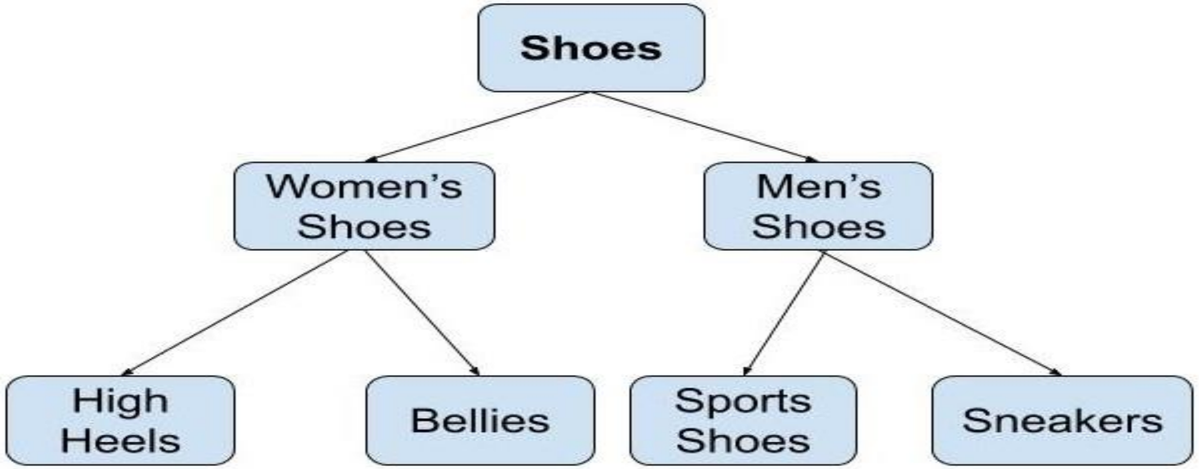
Types of Data Model in DBMS

1. Hierarchical Model
2. Network Model
3. Entity-Relationship Model
4. Relational Model
5. Object-Oriented Data Model
6. Object-Relational Data Model
7. Flat Data Model
8. Semi-Structured Data Model
9. Associative Data Model
10. Context Data Model

Types of Data Model in DBMS

1. Hierarchical Model

Hierarchical Model was the first DBMS model. This model organises the data in the hierarchical tree structure. The hierarchy starts from the root which has root data and then it expands in the form of a tree adding child node to the parent node. This model easily represents some of the real-world relationships like food recipes, sitemap of a website etc. Example: We can represent the relationship between the shoes present on a shopping website in the following way:



Hierarchical Model

Types of Data Model in DBMS

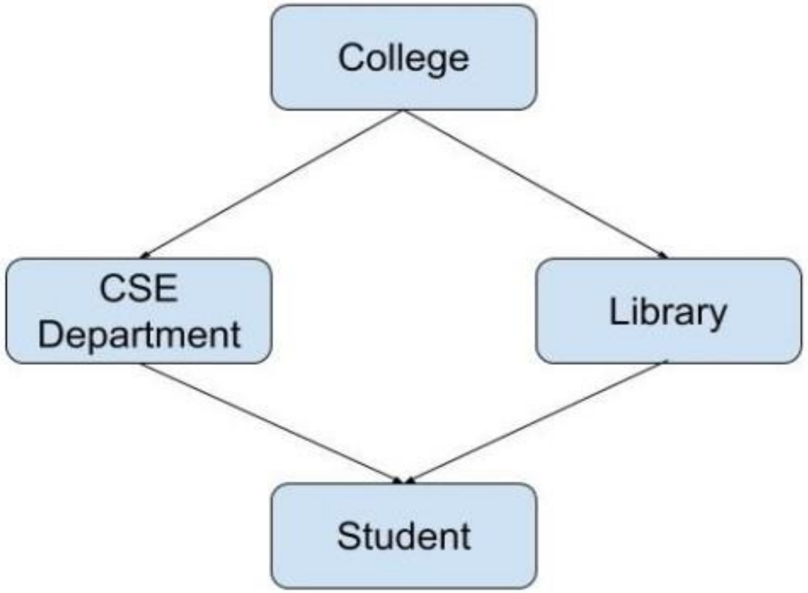
Advantages Hierarchical Model :

1. Easier database design.
2. Data independence
3. Manages large amounts of data.
4. Multi-level relationships between data sets.
5. Structures data in an upside-down tree.
6. Many children per parent.
7. Improve data sharing.
8. Tabular view improves simplicity.

Types of Data Model in DBMS

2. Network Model :

This model is an extension of the hierarchical model. It was the most popular model before the relational model. This model is the same as the hierarchical model, the only difference is that a record can have more than one parent. It replaces the hierarchical tree with a graph. Example: In the example below we can see that node student has two parents i.e. CSE Department and Library. This was earlier not possible in the hierarchical model.



Network Model

Types of Data Model in DBMS

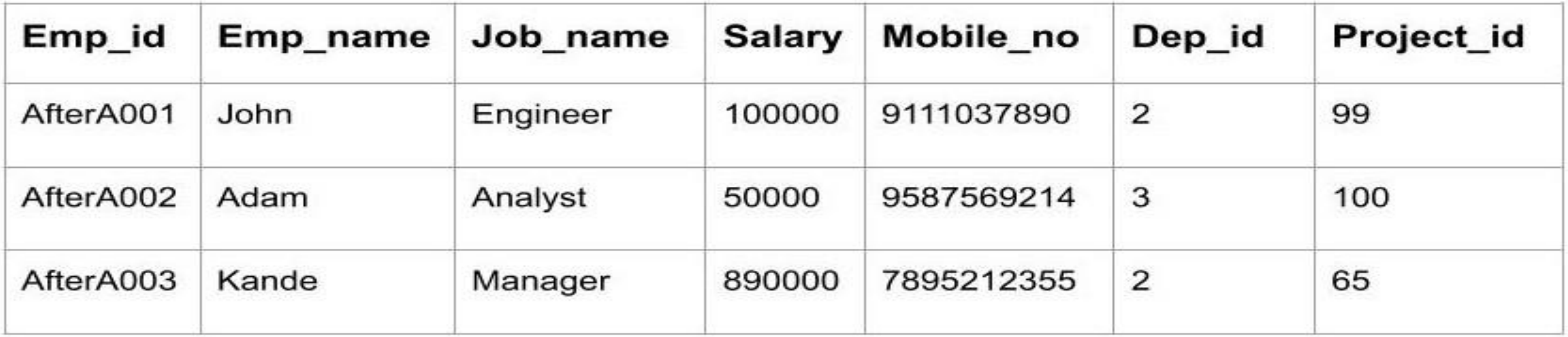
Advantages Network Model :

1. Multi-parent support.
2. Somewhat same simplicity as the hierarchical model.
3. More useful than the hierarchical data model.
4. Deals with even larger amounts of information than the hierarchical model.
5. Promotes data integrity.
6. Many too many relationships support.
7. Data independence.
8. Improved data access.

Types of Data Model in DBMS

3. Relational Model

Relational Model is the most widely used model. In this model, the data is maintained in the form of a two-dimensional table. All the information is stored in the form of row and columns. The basic structure of a relational model is tables. So, the tables are also called relations in the relational model. Example: In this example, we have an Employee table..



Emp_id	Emp_name	Job_name	Salary	Mobile_no	Dep_id	Project_id
AfterA001	John	Engineer	100000	9111037890	2	99
AfterA002	Adam	Analyst	50000	9587569214	3	100
AfterA003	Kande	Manager	890000	7895212355	2	65

EMPLOYEE TABLE

Types of Data Model in DBMS

Advantages Relational Model :

1. Easier database design
2. Users do not have to know the physical representation of the database.
3. Use of SQL language to access data.
4. Tabular view improves simplicity.
5. Support large amounts of data.
6. Data independence.
7. Multi-level relationships between data sets
8. No need to predefined data relationships.

Types of Data Model in DBMS

4. Object Oriented Model :

Object oriented data model is based upon real world situations. These situations are represented as objects, with different attributes. All these object have multiple relationships between them.

Elements of Object oriented data model

1. Objects :

The real world entities and situations are represented as objects in the Object oriented database model.

Attributes and Method :

Every object has certain characteristics. These are represented using Attributes. The behaviour of the objects is represented using Methods.

Types of Data Model in DBMS

4. Object Oriented Model :

Class

Similar attributes and methods are grouped together using a class. An object can be called as an instance of the class.

Inheritance

A new class can be derived from the original class. The derived class contains attributes and methods of the original class as well as its own.

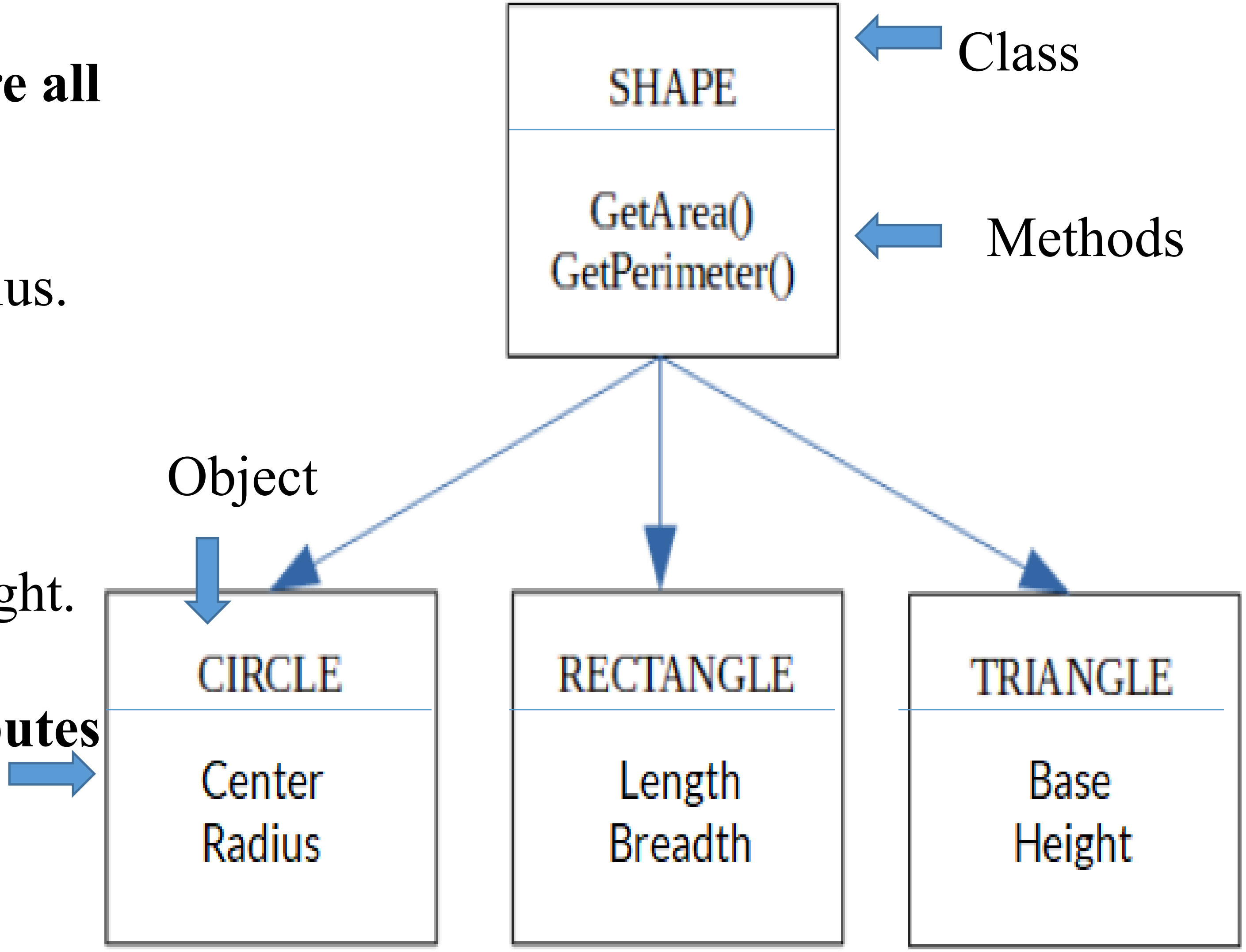
Types of Data Model in DBMS

4. Object Oriented Model :

Shape, Circle, Rectangle and Triangle are all objects in this model.

- Circle has the attributes Center and Radius.
- Rectangle has the attributes Length and Breath
- Triangle has the attributes Base and Height.

The objects Circle, Rectangle and Triangle inherit from the object Shape.



Network Vs Relational Model

NETWORK DATA MODEL	RELATIONAL DATA MODEL
1. This model is difficult to understand	1. This model is very simple to understand.
2. It organizes records in form of directed graphs.	2. It organizes records in form of tables.
3. In this relationship between various records is represented physically via linked list.	3. In this relationship between various records is represented logically via tables.
4. There is lack of declarative querying facilities.	4. It provides declarative query facility using SQL.
5. It organizes records to one another through links or pointers.	5. It organizes records in form of table and relationship between tables are set using common fields.
6. Retrieval of informations are complex	6. Retrieval of informations are simple
7. There is partial data independence in this model.	7. This model provides data independence.
8. VAX-DBMS, DMS-1100 of UNIVAC and SUPRADBMS's use this model.	8. It is mostly used in real world applications. Oracle, SQL.

Network Vs Relational Model

SR. NO	HIERARCHICAL DATA MODEL	NETWORK DATA MODEL	RELATIONAL DATA MODEL
1	In this model, to store data hierarchy method is used.	It organizes records to one another through links or pointers.	It organizes records in the form of table.
2	To organize records, it uses tree structure.	It organizes records in the form of directed graphs.	It organizes records in the form of tables.
3	It implements 1:1 and 1:n relations. Not support many to many	In addition to 1:1 and 1:n it also implements many to many relationships.	In addition to 1:1 and 1:n it also implements many to many relationships.
4	Deletion anomaly exists in this model i.e. it is difficult to delete the parent node.	There is no deletion anomaly.	There is no deletion anomaly.
5	This model lacks data independence.	There is partial data independence in this model.	This model provides data independence.
6	It is used to access the data which is complex.	It is used to access the data which is complex.	It is used to access the data which is complex
7	&XML and XAML use this model.	VAX-DBMS, DMS-1100 of UNIVAC and SUPRADBMS's use this model.	It is mostly used in real world applications. Oracle, SQL.

Entity Relationship Diagram

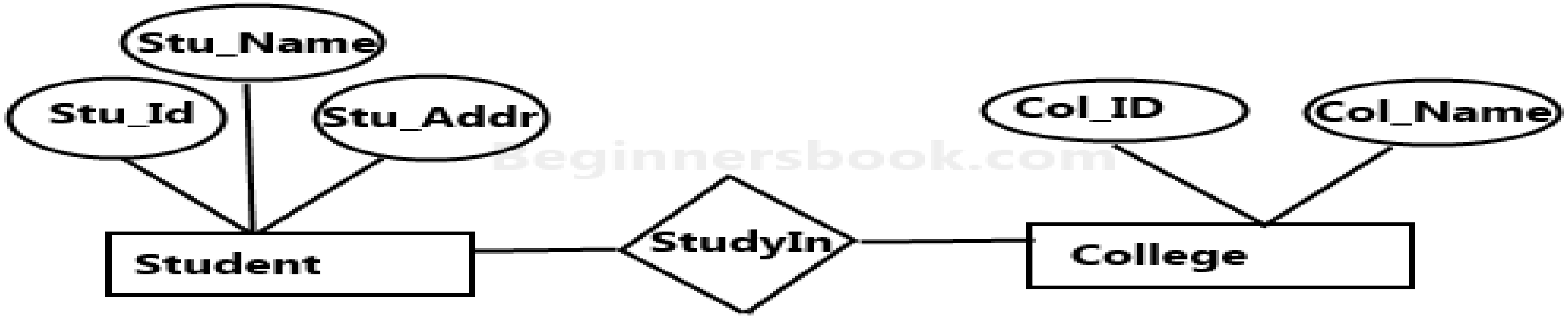
An Entity–relationship model (ER model) describes the structure of a database with the help of a diagram, which is known as Entity Relationship Diagram (ER Diagram).

- It was developed by Peter Chen in 1976.
- An Entity Relationship (ER) Diagram is a type of flowchart that illustrates how **“entities”** such as people, objects or concepts relate to each other within a system.
- The main components of E-R model are: entity set and relationship set.
- ER-diagram is graphical representation of Database.
- An ER diagram shows the relationship among entity sets. An entity set is a group of similar entities and these entities can have attributes.
- An entity is a table or attribute of a table in database, so by showing relationship among tables and their attributes, ER diagram shows the complete logical structure of a database.

Entity Relationship Diagram

A simple ER Diagram:

In the following diagram we have two entities Student and College and their relationship. The relationship between Student and College is many to one as a college can have many students however a student cannot study in multiple colleges at the same time. Student entity has attributes such as Stu_Id, Stu_Name & Stu_Addr and College entity has attributes such as Col_ID & Col_Name.

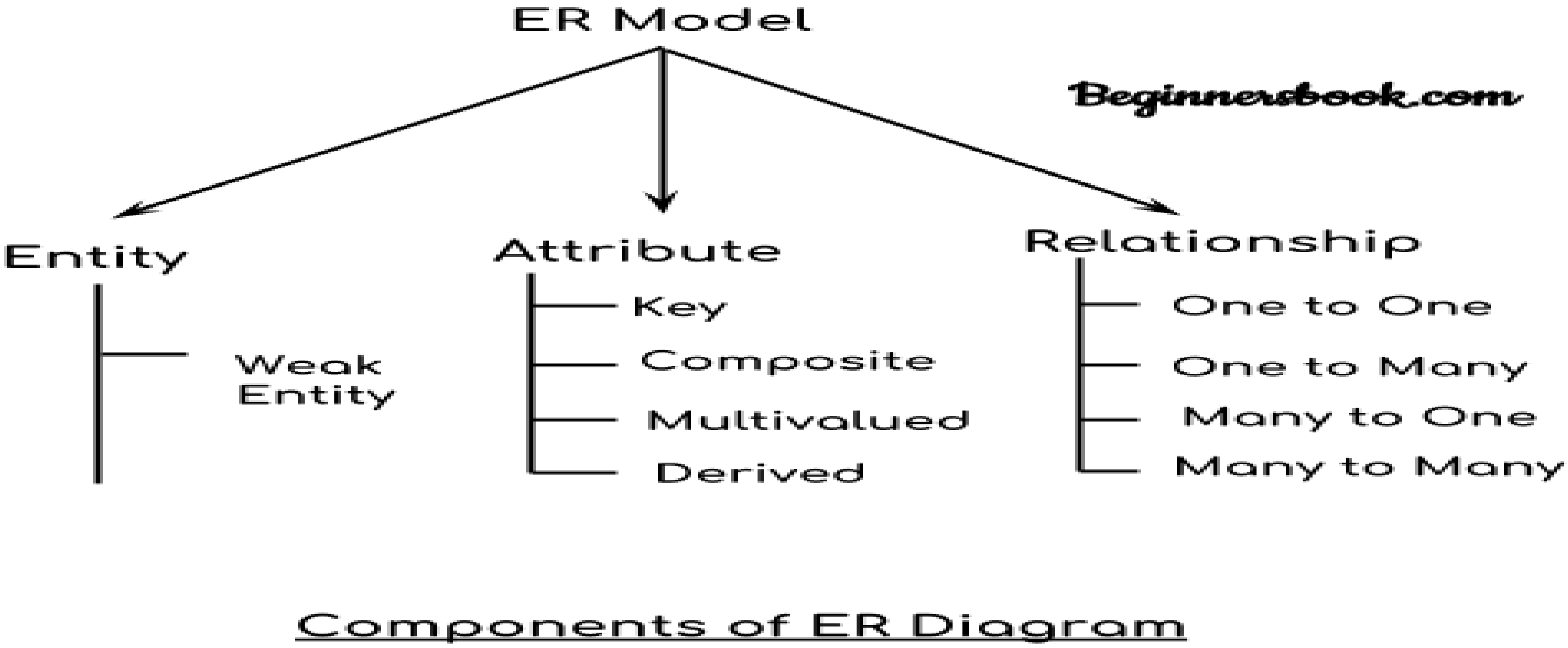


Sample E-R Diagram

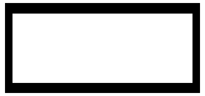
Entity Relationship Diagram

1. **Rectangle:** Represents Entity sets.
2. **Ellipses:** Attributes
3. **Diamonds:** Relationship Set
4. **Lines:** They link attributes to Entity Sets and Entity sets to Relationship Set
5. **Double Ellipses:** Multivalued Attributes
6. **Dashed Ellipses:** Derived Attributes
7. **Double Rectangles:** Weak Entity Sets
8. **Double Lines:** Total participation of an entity in a relationship set

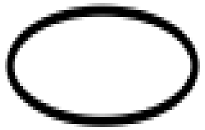
Components of a ER Diagram



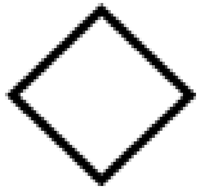
Components of a ER Diagram



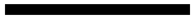
Represents Entity



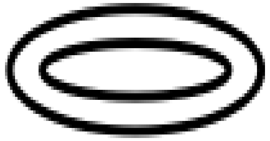
Represents Attribute



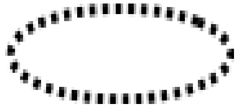
Represents Relationship



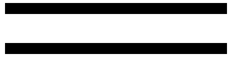
Links Attribute(s) to entity set(s) or
Entity set(s) to Relationship set(s)



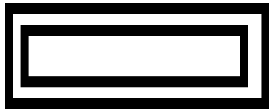
Represents Multivalued Attributes



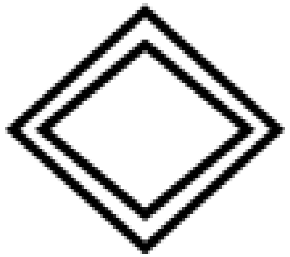
Represents Derived Attributes



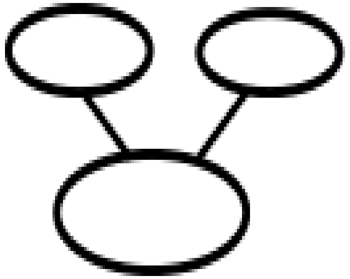
Represents Total Participation of Entity



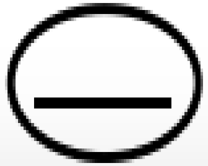
Represents Weak Entity



Represents Weak Relationships



Represents Composite Attributes



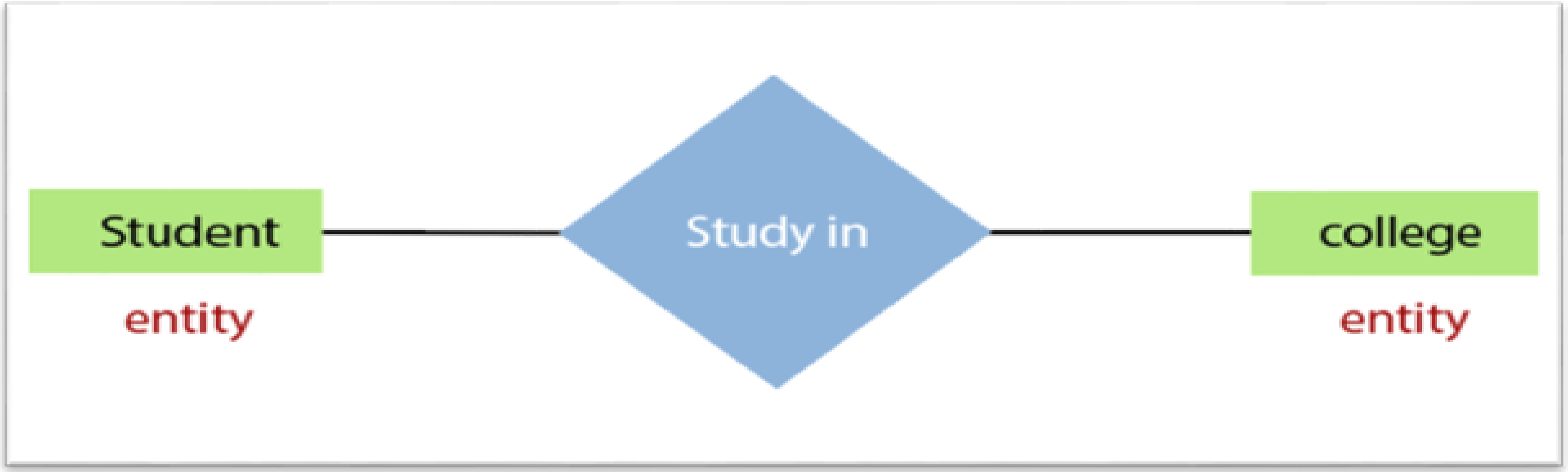
Represents Key Attributes / Single Valued Attributes

1. Entity

1. Entity

An entity is an object or component of data. An entity is represented as rectangle in an ER diagram.

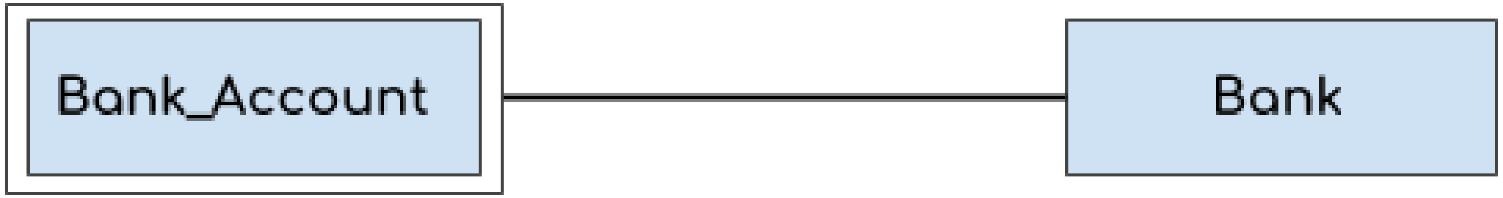
For example: In the following ER diagram we have two entities Student and College and these two entities have many to one relationship as many students study in a single college. We will read more about relationships later, for now focus on entities.



Weak entity

- **Weak Entity:**

An entity that cannot be uniquely identified by its own attributes and relies on the relationship with other entity is called weak entity. The weak entity is represented by a double rectangle. For example – a bank account cannot be uniquely identified without knowing the bank to which the account belongs, so bank account is a weak entity.



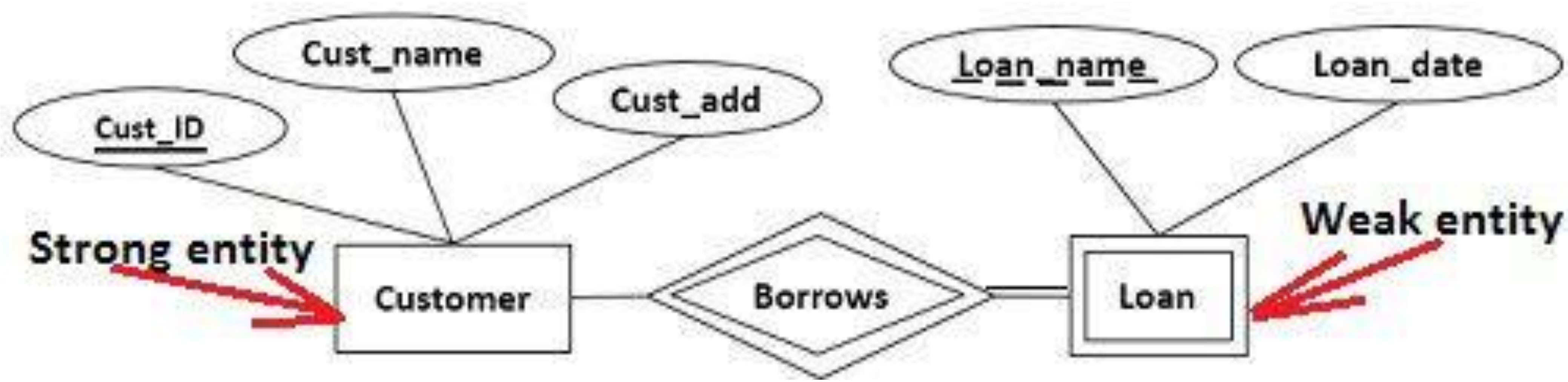
Beginnersbook.com

Weak entity

An entity can be characterized into two types:

Strong entity: This type of **entity has a primary key** attribute which uniquely identifies each record in a table. In the ER diagram, a strong entity is usually represented by a single rectangle.

Weak entity: An **entity does not have a primary key** attribute and depends on another strong entity via foreign key attribute. In the ER diagram, a weak entity is usually represented by a double rectangle.



Attributes

2. Attribute :

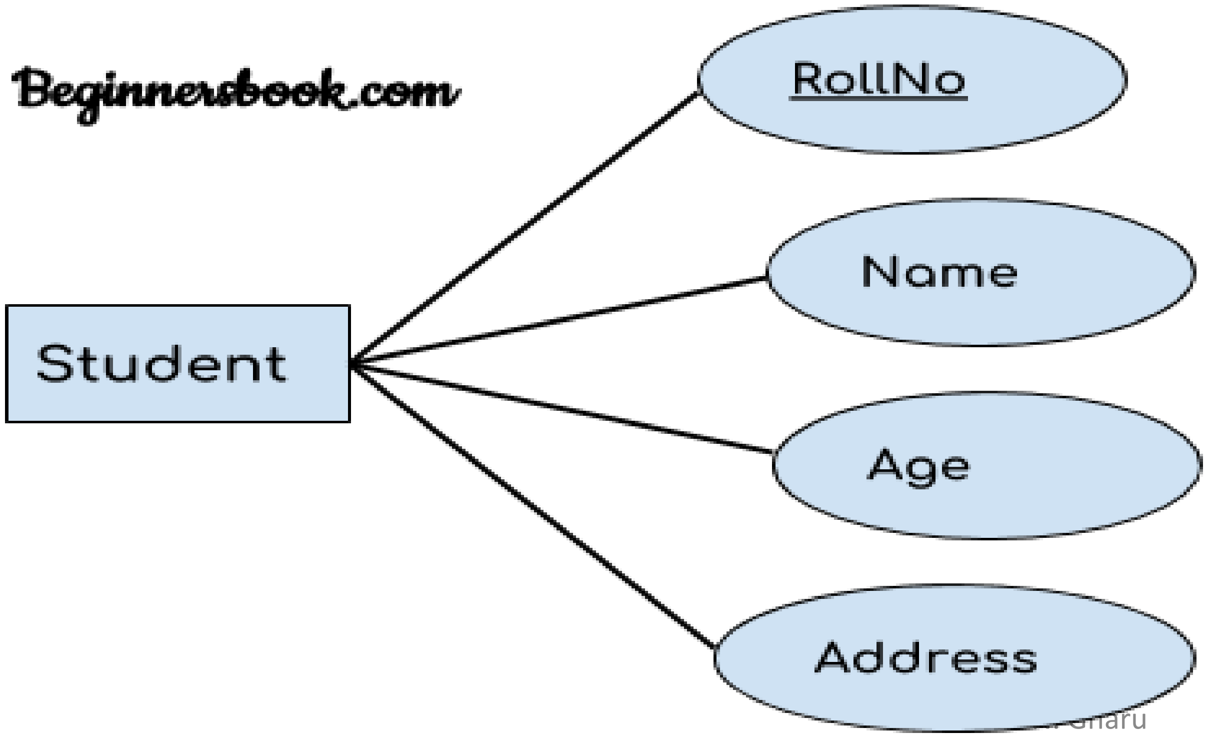
- An attribute describes the property of an entity. An attribute is represented as Oval in an ER diagram. There are four types of attributes:

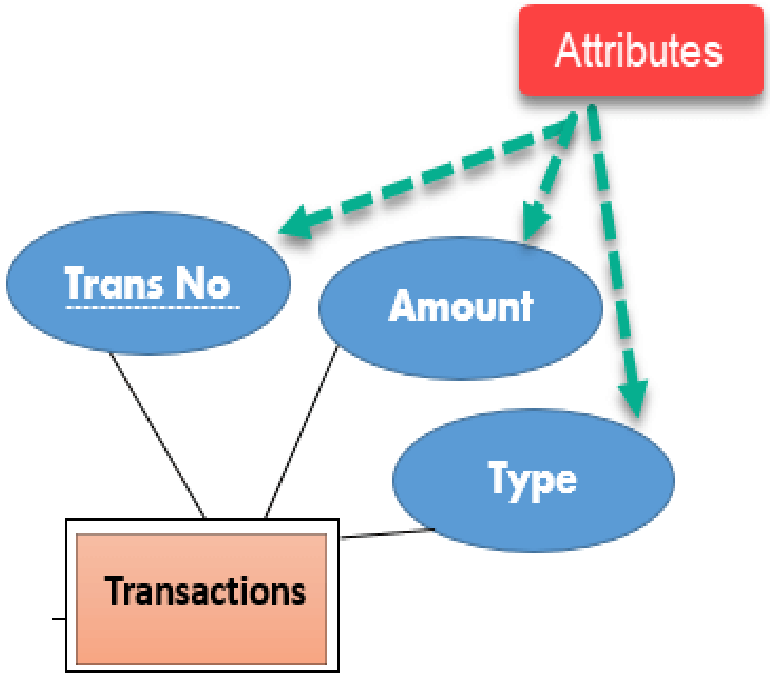
1. Key attribute
2. Composite attribute
3. Multivalued attribute
4. Derived attribute

Types of Attributes

1. Key attribute :

A key attribute can uniquely identify an entity from an entity set. For example, student roll number can uniquely identify a student from a set of students. Key attribute is represented by oval same as other attributes however the text of key attribute is underlined.

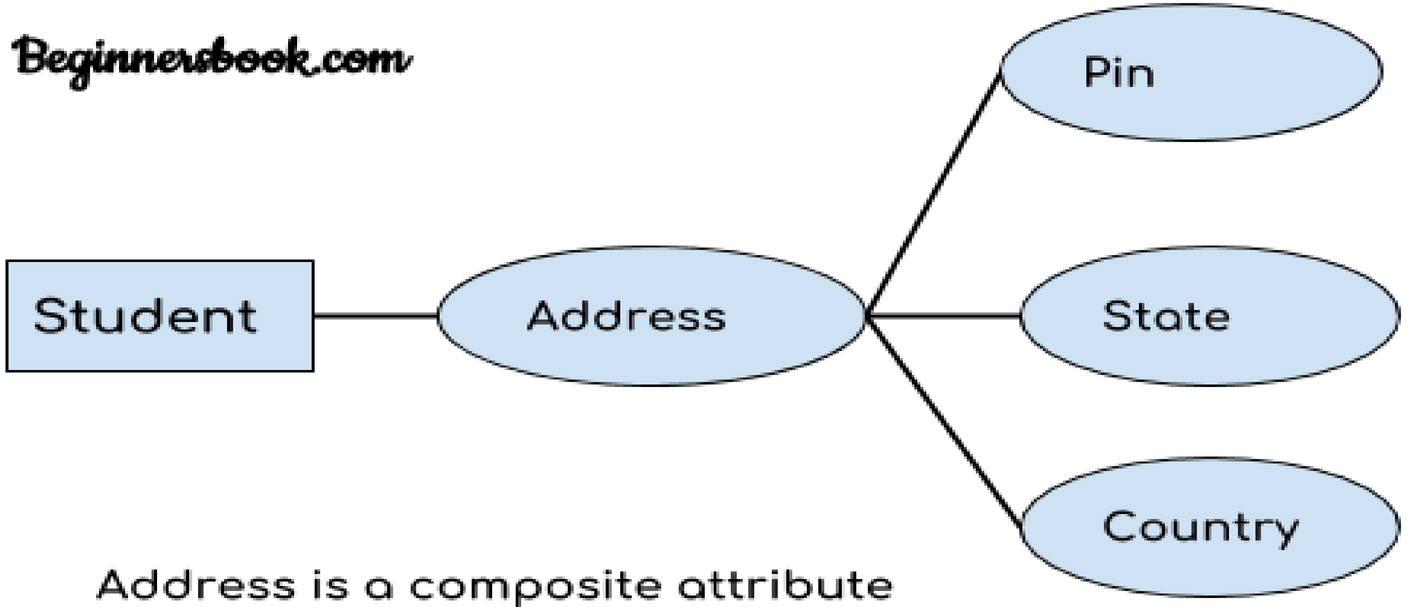




Types of Attributes

2. Composite attribute:

An attribute that is a combination of other attributes is known as composite attribute. For example, In student entity, the student address is a composite attribute as an address is composed of other attributes such as pin code, state, country.



Types of Attributes

3. Multivalued attribute:

An attribute that can hold multiple values is known as multivalued attribute. It is represented with **double ovals** in an ER Diagram. For example – A person can have more than one phone numbers so the phone number attribute is multivalued

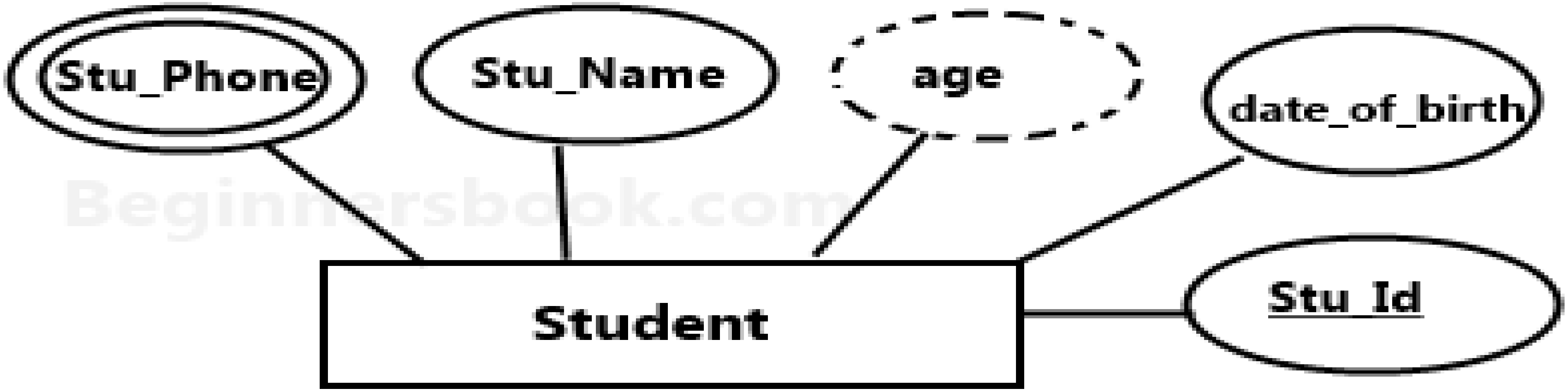
.

4. Derived attribute:

A derived attribute is one whose value is dynamic and derived from another attribute. It is represented by **dashed oval** in an ER Diagram. For example – Person age is a derived attribute as it changes over time and can be derived from another attribute (Date of birth).

Types of Attributes

E-R Diagram for Multivalued attribute & Derived attribute :



3. Relationship

3. Relationship :

A relationship is represented by diamond shape in ER diagram, it shows the relationship among entities. There are four types of relationships:

1. One to One

2. One to Many

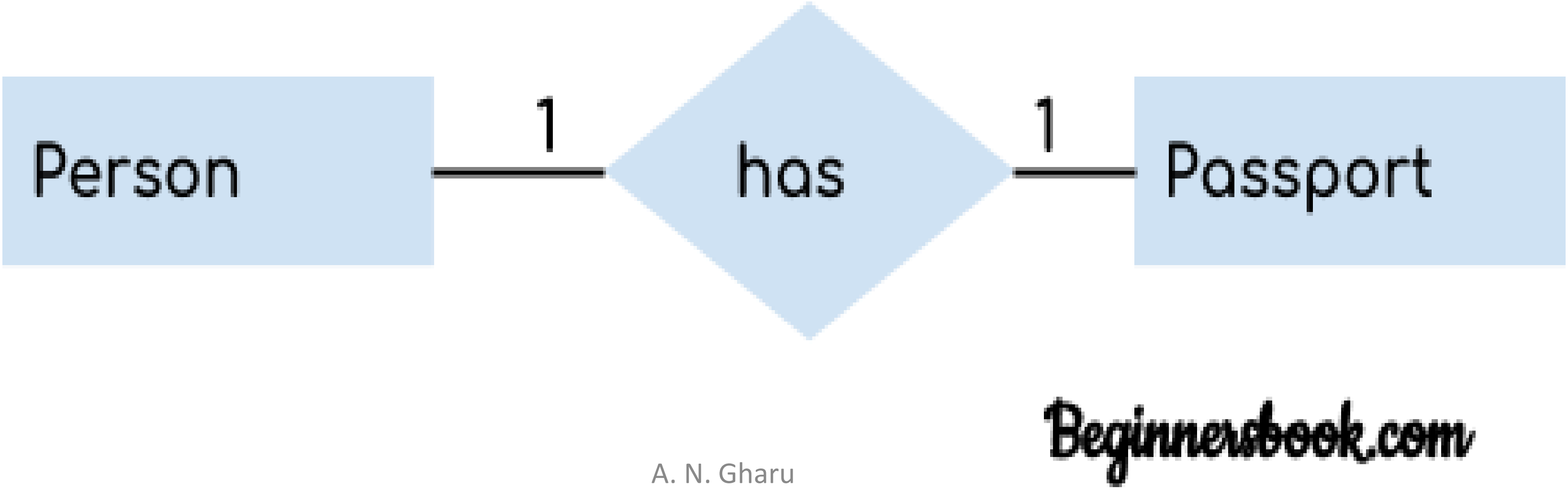
3. Many to One

4. Many to Many

Types of Relationship

1. One to One Relationship :

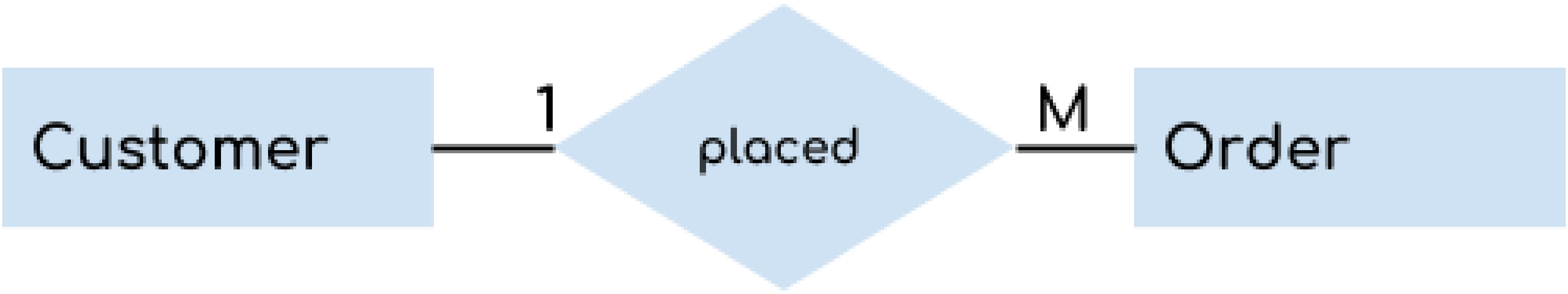
When a single instance of an entity is associated with a single instance of another entity then it is called one to one relationship. For example, a person has only one passport and a passport is given to one person.



Types of Relationship

2. One to Many Relationship

When a single instance of an entity is associated with more than one instances of another entity then it is called one to many relationship. For example – a customer can place many orders but a order cannot be placed by many customers.

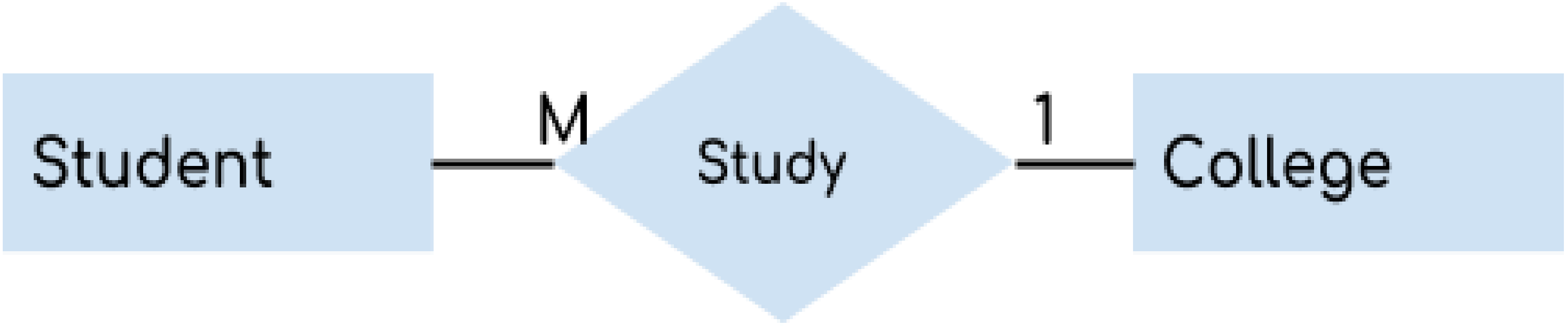


Beginnersbook.com

Types of Relationship

3. Many to One Relationship

When more than one instances of an entity is associated with a single instance of another entity then it is called many to one relationship. For example – many students can study in a single college but a student cannot study in many colleges at the same time.

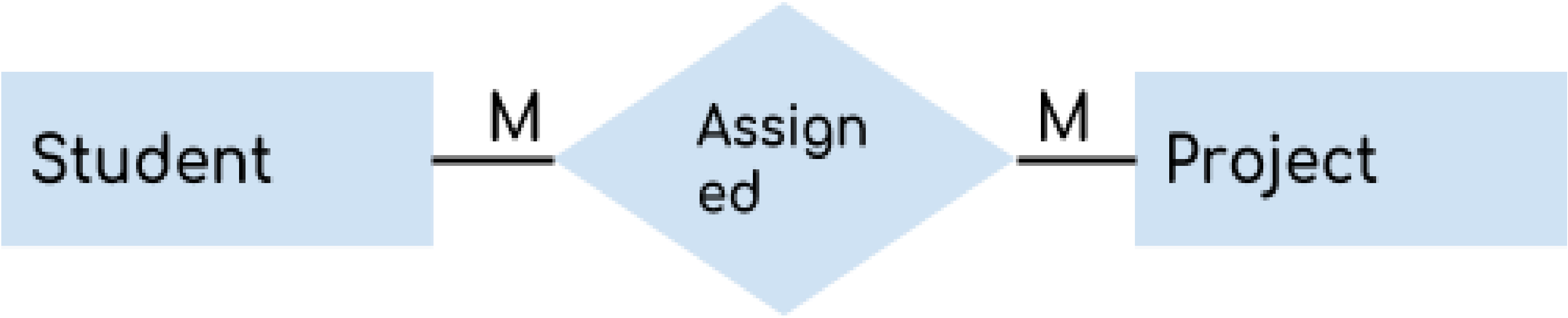


Beginnersbook.com

Types of Relationship

4. Many to Many Relationship

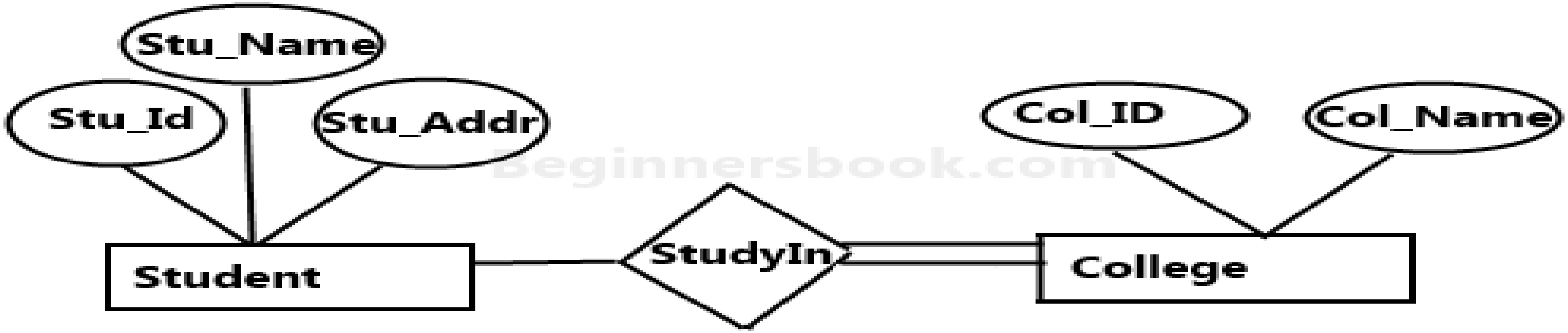
When more than one instances of an entity is associated with more than one instances of another entity then it is called many to many relationship. For example, a can be assigned to many projects and a project can be assigned to many students.



Beginnersbook.com

Total Participation of an Entity set

A Total participation of an entity set represents that each entity in entity set must have at least one relationship in a relationship set. For example: In the below diagram each college must have at-least one associated Student.



E-R Digram with total participation of College entity set in StudyIn relationship Set - This indicates that each college must have atleast one associated Student.

Extended Features of ER Diagram

EER is a high-level data model that incorporates the extensions to the original ER model. Enhanced ERD are high level models that represent the requirements and complexities of complex database.

There are three types of EER :

1. Generalization
2. Specialization
3. Aggregration

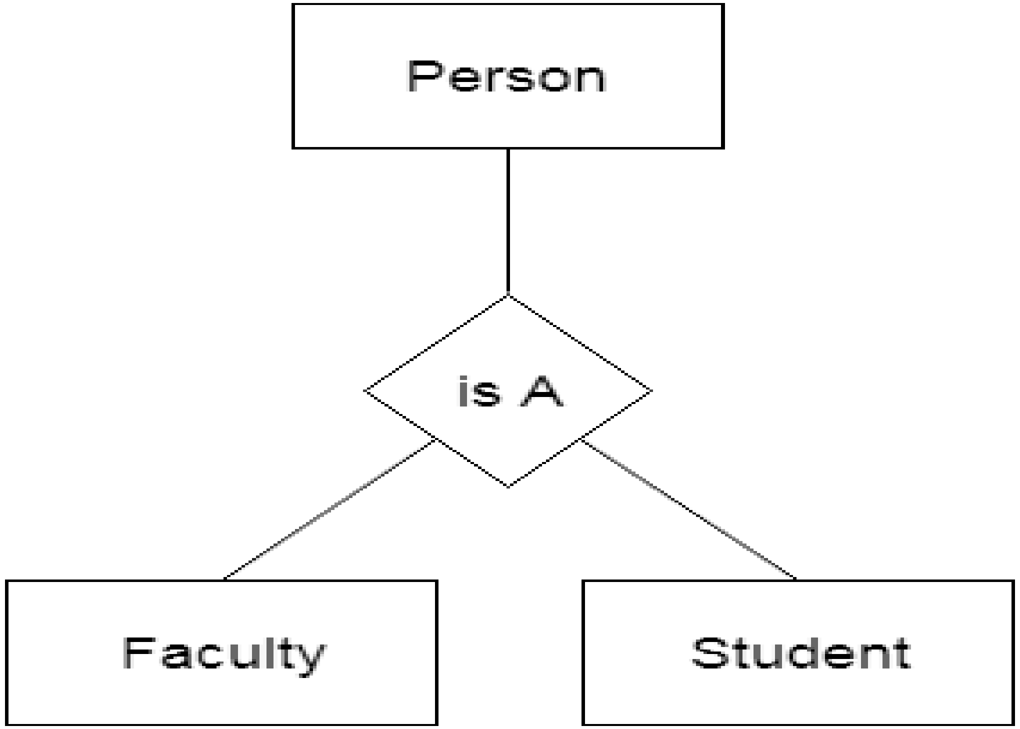
Extended Features of ER Diagram

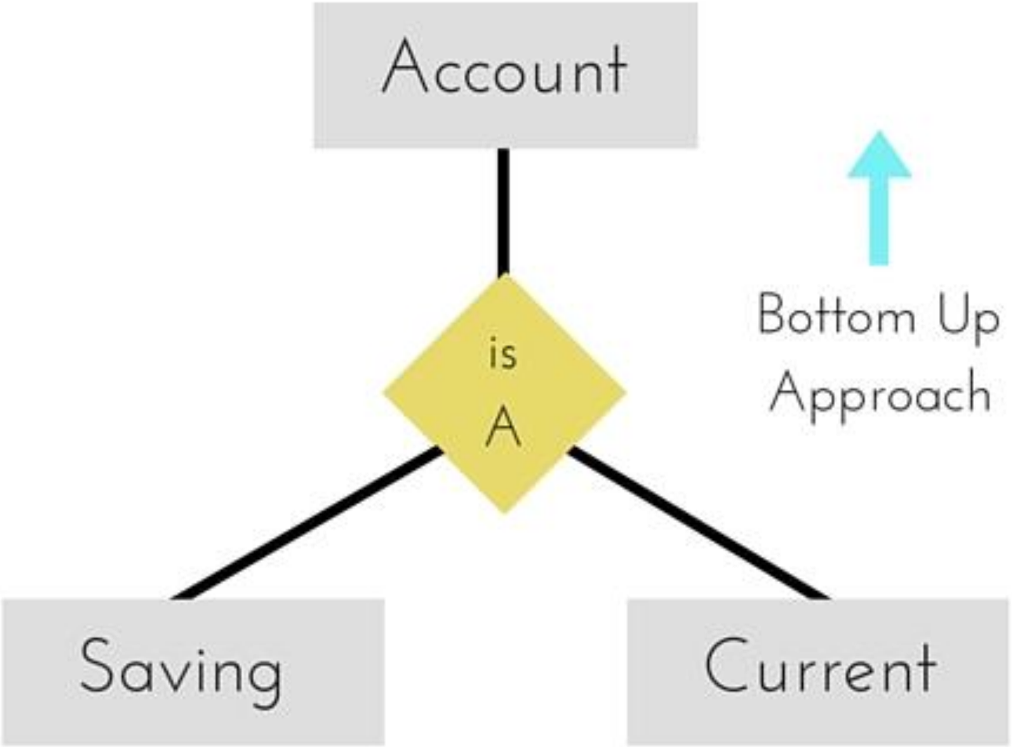
1. Generalization : (↑)

- Generalization is like a **bottom-up approach** in which two or more entities of lower level combine to form a higher level entity if they have some attributes in common.
- In generalization, an entity of a higher level can also combine with the entities of the lower level to form a further higher level entity.
- Generalization is more like subclass and superclass system, but the only difference is the approach. Generalization uses the bottom-up approach.
- In generalization, entities are combined to form a more generalized entity, i.e., subclasses are combined to make a superclass.

Extended Features of ER Diagram

1. Generalization :





For example, **Saving** and **Current** account types entities can be generalised and an entity with name **Account** can be created, which covers both.

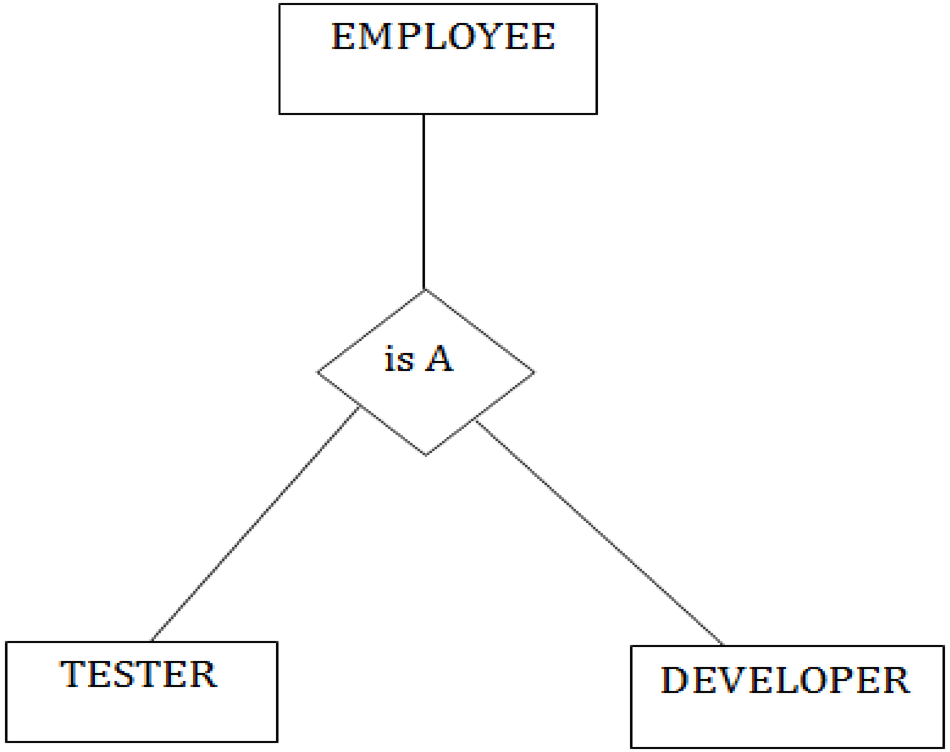
Extended Features of ER Diagram

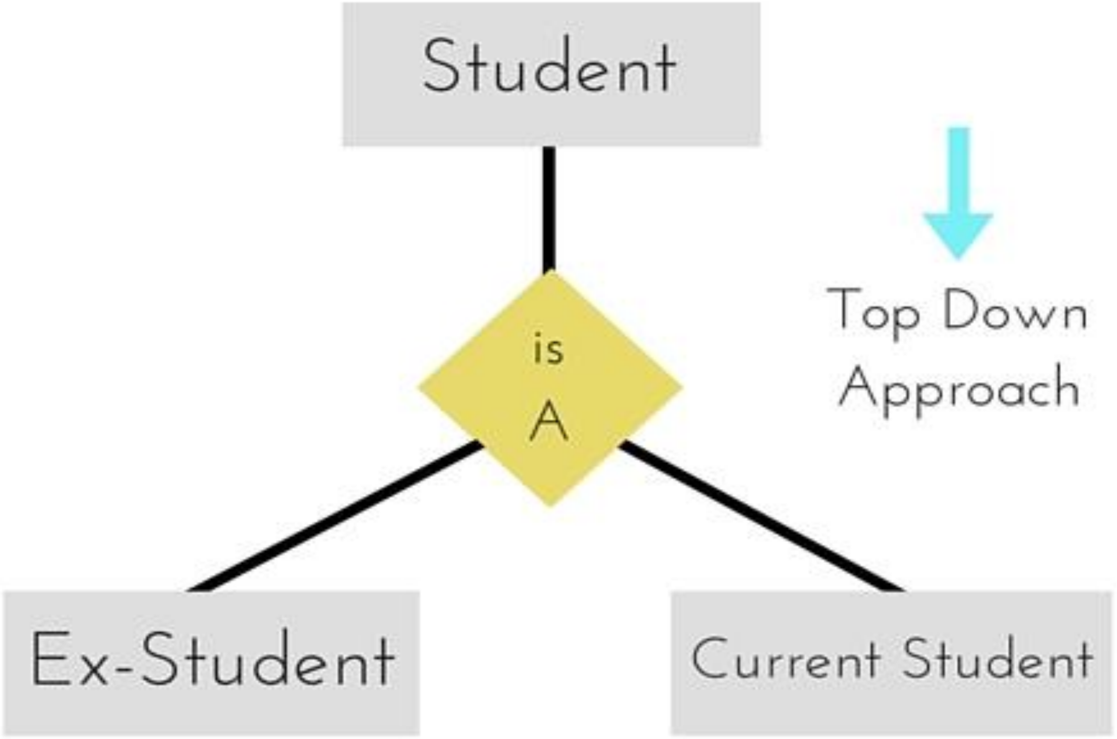
2. Specialization :

- Specialization is a top-down approach, and it is opposite to Generalization. In specialization, one higher level entity can be broken down into two lower level entities.
- Specialization is used to identify the subset of an entity set that shares some distinguishing characteristics.
- Normally, the superclass is defined first, the subclass and its related attributes are defined next, and relationship set are then added

Extended Features of ER Diagram

2. **Specialization :**





For example: In an Employee management system, EMPLOYEE entity can be specialized as TESTER or DEVELOPER based on what role they play in the company.

Extended Features of ER Diagram

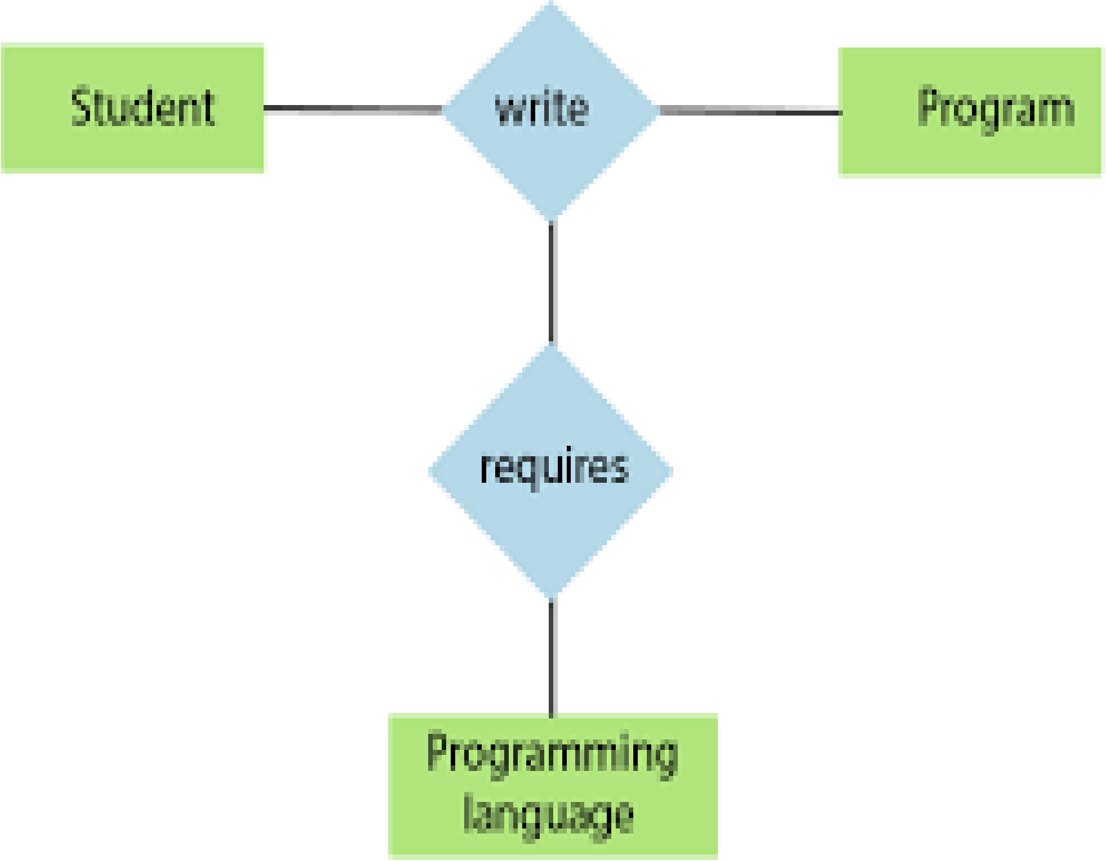
1. **Aggregration :**

- In aggregation, the relation between two entities is treated as a single entity.
- In aggregation, relationship with its corresponding entities is aggregated into a higher level entity.

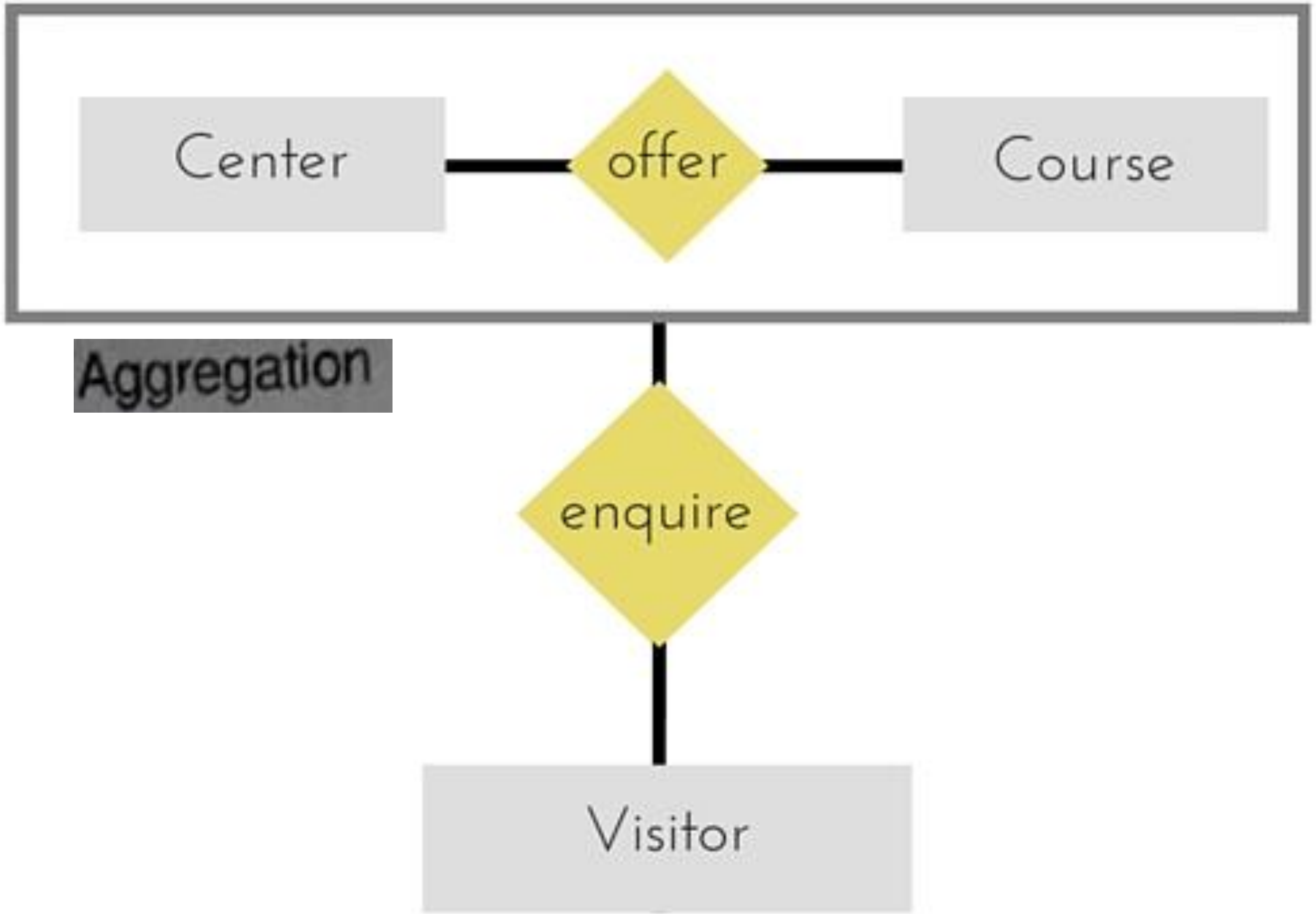
For example: Center entity offers the Course entity act as a single entity in the relationship which is in a relationship with another entity visitor. In the real world, if a visitor visits a coaching center then he will never enquiry about the Course only or just about the Center instead he will ask the enquiry about both.

Extended Features of ER Diagram

1. Aggregration :



Aggregation

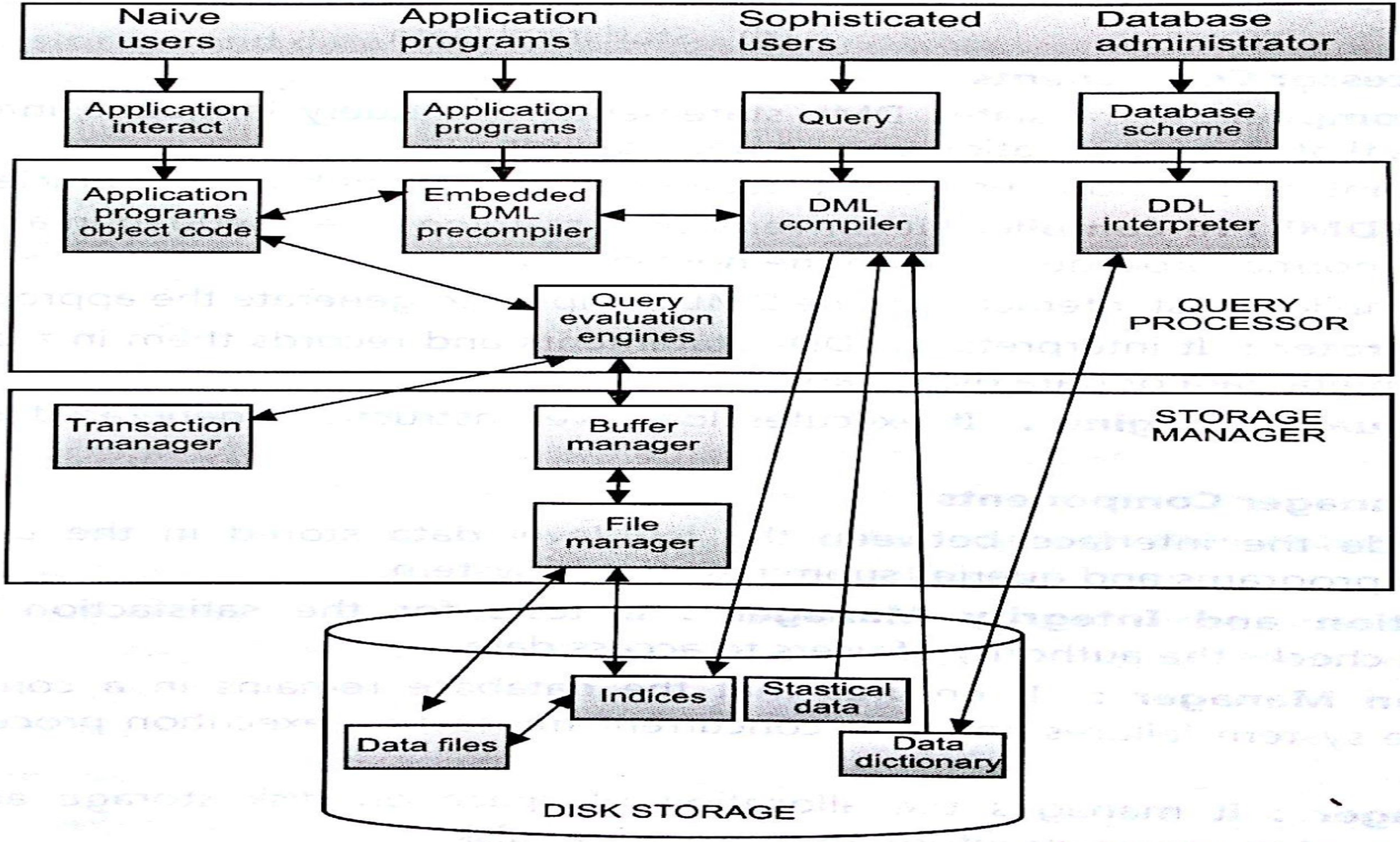


Generalization Vs Specialization

GENERALIZATION	SPECIALIZATION
1. Generalization works in **Bottom-Up approach**.	1. Specialization works in **top-down approach**.
2. In Generalization, **size of schema gets reduced.**	2. In Specialization, **size of schema gets increased.**
3. Generalization is normally **applied to group of entities.**	3. We can apply Specialization **to a single entity.**
4. Generalization can be defined as a process of **creating groupings from various entity** sets	4. Specialization can be defined as process of **creating subgrouping within an entity set**
5. Generalization process starts with the number of entity sets and it creates high-level entity with the help of some common features.	5. Specialization process starts from a single entity set and it creates a different entity set by using some different features.
6. There is no inheritance in Generalization.	6. There is inheritance in Specialization.

Overall structure of DBMS

Overall System Structure of DBMS



System structure

Overall System Structure of DBMS

1. Query processor

- **DML compiler**

It processes the DML statements into low level instruction (machine language), so that they can be executed.

Embedded DML precompiler

Converts the DML statements in the an application program to normal procedure calls in the host language.

- **DDL interpreter**

It processes the DDL statements into a set of table containing meta data (data about data)

Query Evaluation Engine

Overall System Structure of DBMS

2. Storage Manager

- **Authorization and integrity manager**

Tests for the satisfaction of integrity constraints and Checks the authority of user to perform various action.

- **Transaction Manager**

Ensures the database remains in a consistent (correct) state despite system failures.

- **File manager**

Responsible for the allocation of space on the disk storage system.

- **Buffer manager**

Manages the data coming into and out of the system, Including the caching of data.

Overall System Structure of DBMS

3. Data structures

- **Data files**

It stores the data.

Data dictionary

the metadata about the structure of the database.

- **Indices**

Used to provide fast access to the data.

- **Statistical data**

The query processor uses this to optimize queries.

THANK YOU!!!

My Blog : https://anandgharu.wordpress.com/

Email : gharu.anand@gmail.com