SmartCliff
CAREER MOBILITY SOLUTIONS

We are on a mission to address the digital skills gap for 10 Million+ young professionals, train and empower them to forge a career path into future tech

Database Fundamentals

Data vs. Information

Data

- **Facts** and **statistics** collected together for **reference or analysis**.
- Data can be **anything** text, image audio, video etc.
- Facts are truth values of **physical property**.
- Does **not reveal its meaning**
- **Example:**
  - **4,8,12,16,…**
  - **dog, cat, cow, cockatoo,…**
  - **161.2, 175.3, 166.4, 164.7, 169.3,…**

Data vs. Information

Information

- The **processed data** is called information.
- Data to **reveal its meaning.**
- **Example:**

  - **4, 8, 12 and 16 are the first four answers in the 4 x table**
  - **dog, cat, cow is a list of household pets**
  - **165, 175.2, 186.3, 164.3, 169.3 are the height of 14-year old students.**

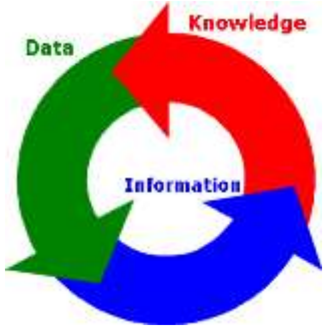
Note: “Every information is data but every data is not information”

Data Vs Information

DATA	INFORMATION
Raw facts	Processed data
No context (Meaningless)	Data with context
Just numbers and characters	Value added to data
Not organized	Organized, Summarized and Analyzed

Example:

- 4,8,12,16,…
- dog, cat, cow, cockatoo,…
- 161.2, 175.3, 166.4, 164.7, 169.3,…

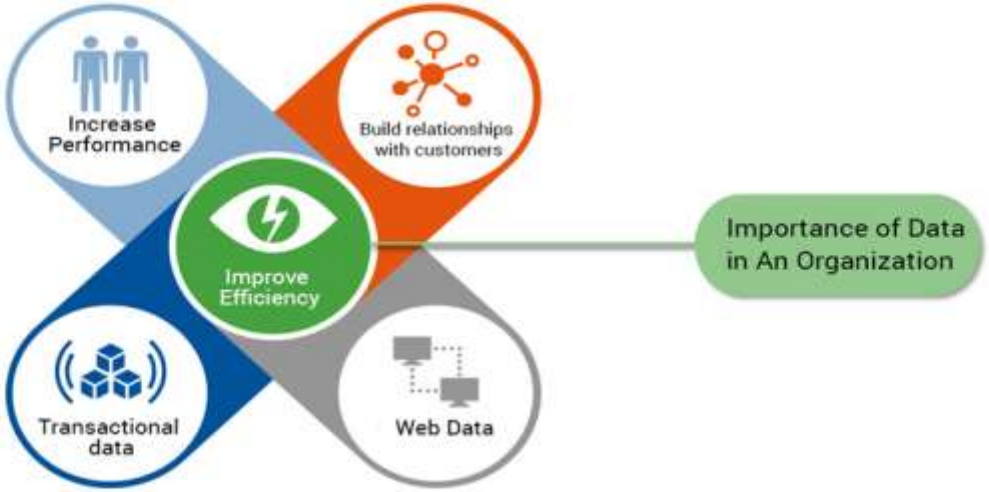


Example:

- 4, 8, 12 and 16 are the first four answers in the 4 x table
- dog, cat, cow is a list of household pets
- 165, 175.2, 186.3, 164.3, 169.3 are the height of 14-year old students.

Importance of data

- Optimize business operations.
- Better decision making.
- Improved marketing strategies.
- Reduce costs.
- Increase revenue and profit.

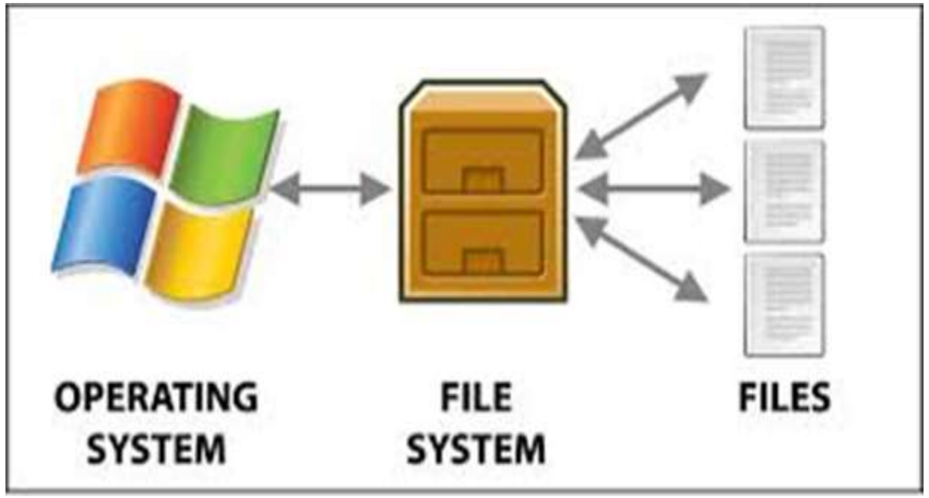

Increase Performance
Build relationships with customers
Improve Efficiency
Importance of Data in An Organization
Transactional data
Web Data

Where to store data ?

- Physically in Notebooks, ledgers, files etc.
- Digitally in a computer using applications like wordpad, notepad, excel sheet etc.
- Logically in a database.

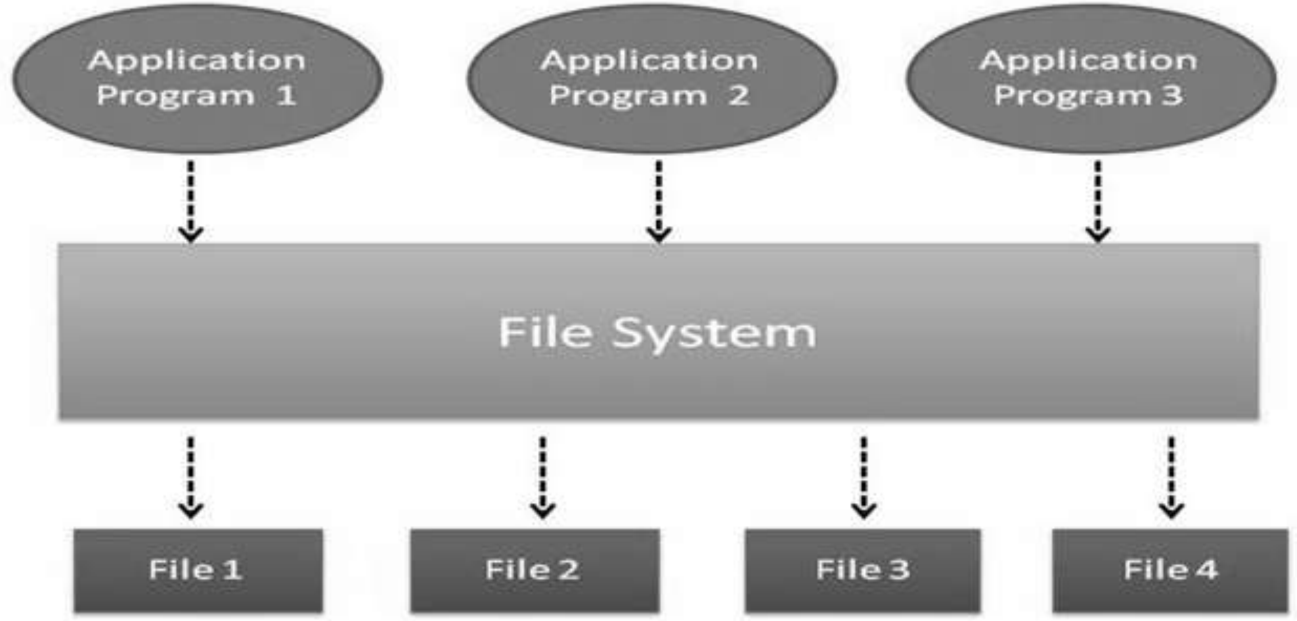
File System

- Each **file in the system** used its own application program to **store, retrieve and modify data** and each file was owned by the individual.
- In **File System** the files are named and where they are **placed logically** for **storage** and **retrieval**.
- Supported by **Conventional Operating System**.



File System

Traditional File Systems



File System : Drawbacks

1. Data Redundancy and Inconsistency

- Multiple file formats, duplication of information in different files

2. Difficulty in accessing data

- Need to write a new program to carry out each new task

3. Data Dependence

- Program and application in the file processing system are data dependent but, the problem is incompatible with file format.

4. Limited data sharing

File System : Drawbacks

5. The problem with security

6. Concurrent access by multiple users

- Concurrent access needed for performance. Uncontrolled concurrent accesses can lead to inconsistencies. **Example:** Two people reading a balance (say 100) and updating it by withdrawing money (say 50 each) at the same time

7. Time-consuming

8. Inefficient to maintain the record of the big firm having many items.

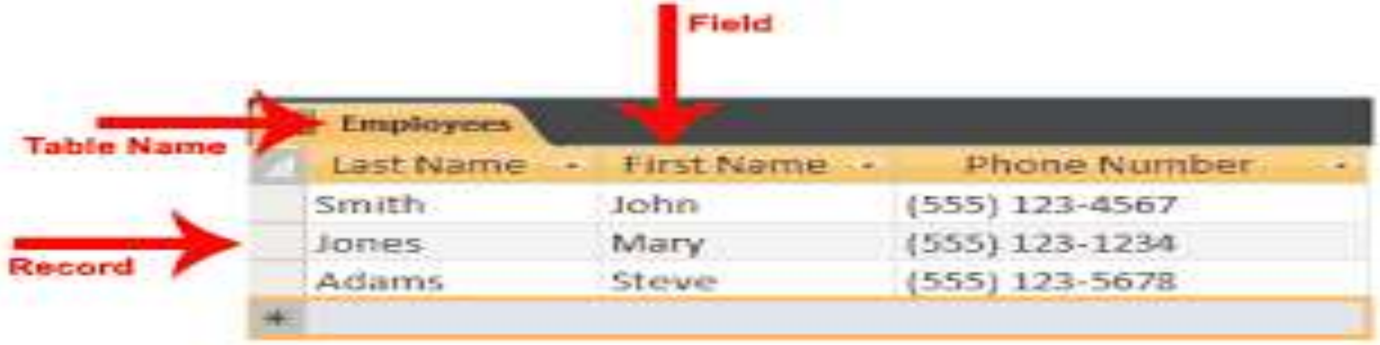
Database

- A **database** is simply a **collection of related data** that is organized.
- This may come in the form of a set of index cards (**e.g. a recipe collection**), an ordered list (**e.g. a phone book**), or a set of tables in the computer (**e.g. student records**).

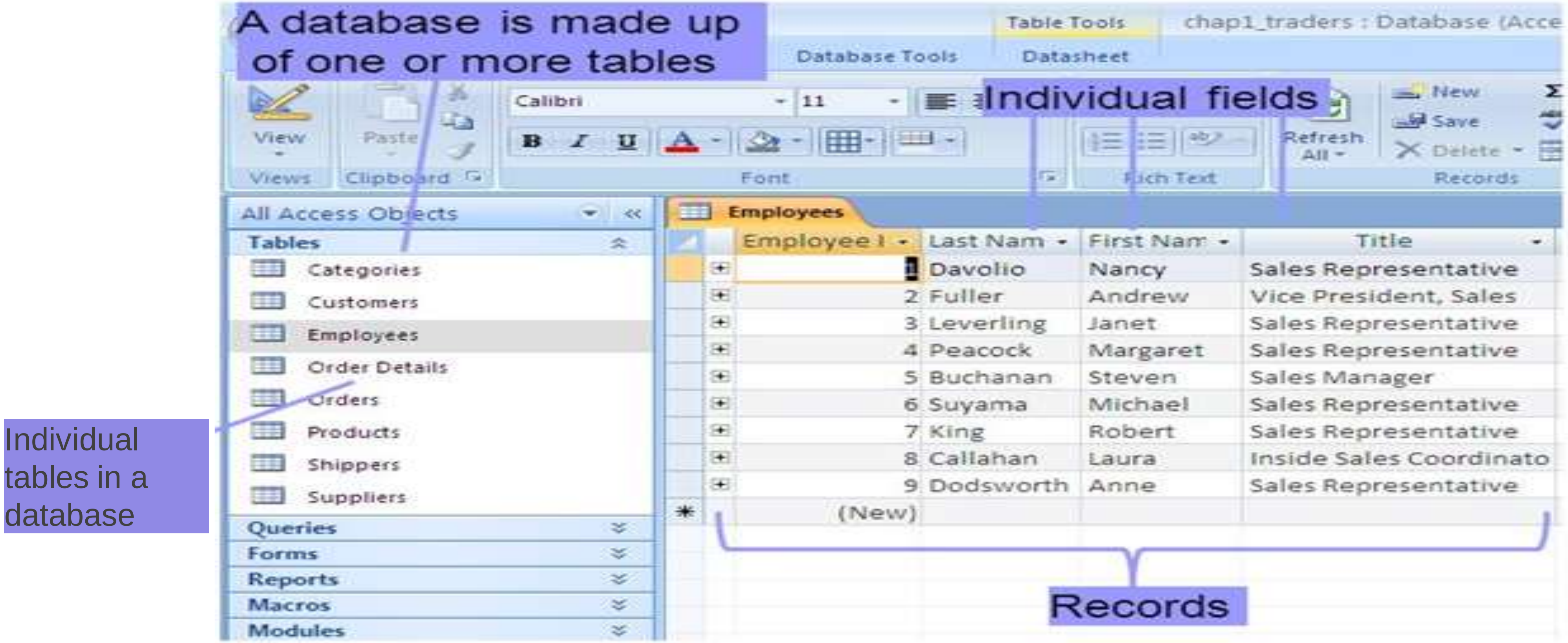
Traditional Database Terminologies

- **Database** - A database is a collection of **tables, with related data**.
- **Table** or **Relation**: A database table is **composed of records** and **fields** that hold data. Each table in a database holds **data about a different**, but **related, subject.**
- **Record or Tuple**: A record is **composed of fields** and contains all the **data about one particular** person, company, or item in a database.
- **Field or Attribute**: A field is **part of a record** and contains a single piece of data for the subject of the record.

Database Terminologies



Last Name	First Name	Phone Number
Smith	John	(555) 123-4567
Jones	Mary	(555) 123-1234
Adams	Steve	(555) 123-5678



Employee I	Last Nam	First Nam	Title
1	Davolio	Nancy	Sales Representative
2	Fuller	Andrew	Vice President, Sales
3	Leverling	Janet	Sales Representative
4	Peacock	Margaret	Sales Representative
5	Buchanan	Steven	Sales Manager
6	Suyama	Michael	Sales Representative
7	King	Robert	Sales Representative
8	Callahan	Laura	Inside Sales Coordinato
9	Dodsworth	Anne	Sales Representative
(New)			

Types of Database

Based on number of User

- Single User Database also known as Desktop Database
- Multi User Database also known as Workgroup and Enterprise Database

Based on Location

- Centralized and Distributed Database

Based on the Type

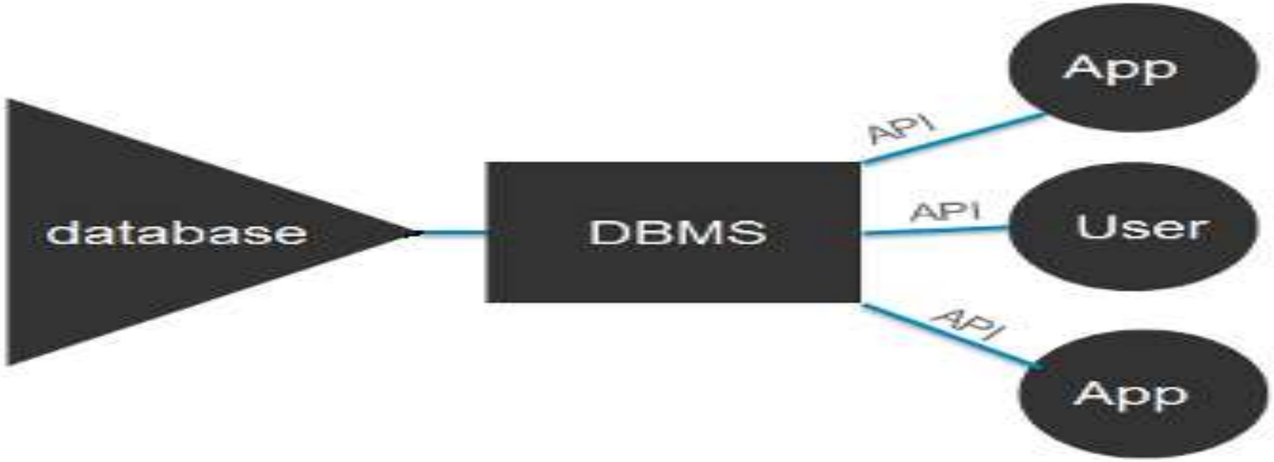
- Structured and Unstructured Database

Based on the Use

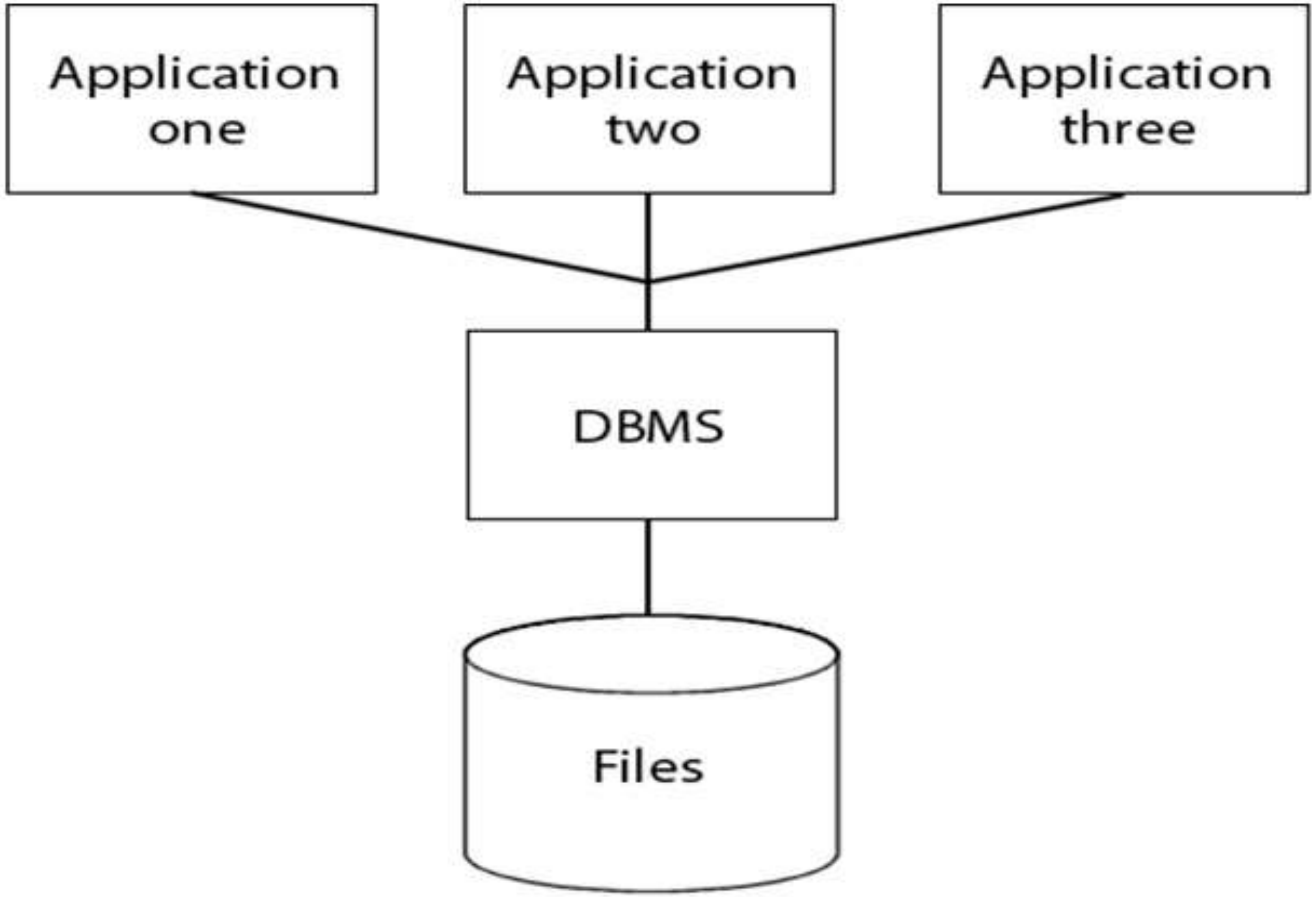
- Operational or Transactional or Procedural Database

Database Management System (DBMS)

- **DBMS** - Collection of **interrelated data** and a **set of programs** to access the data.
- **Goal**: DBMS provides an **environment** that is **both convenient and efficient** to use in retrieving and storing database information.



DBMS Approach



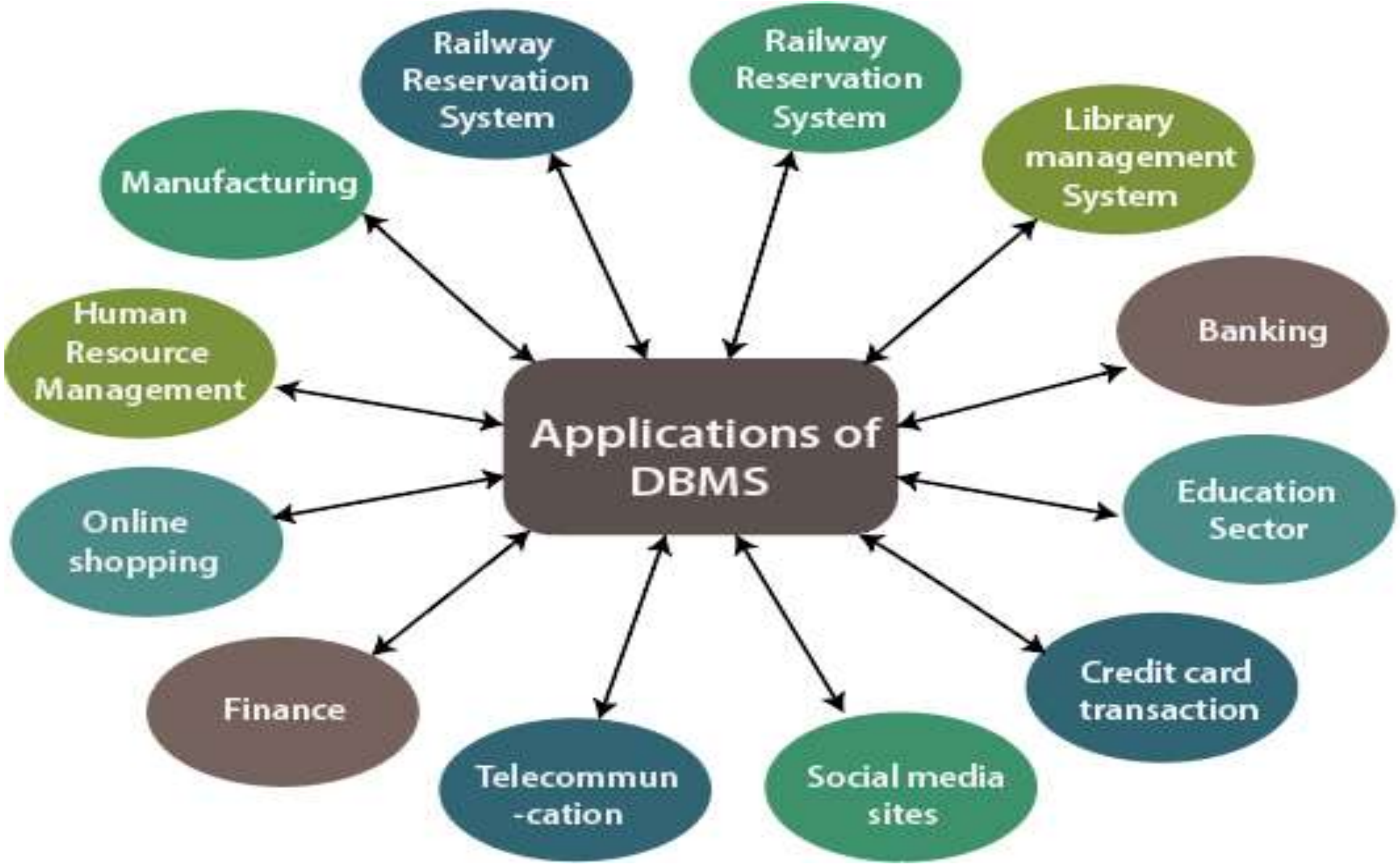
Database Systems

“Collection of DB and DBMS is called database systems”

Database System=Database +DBMS

“Store and transform data into information to support making decisions”

DBMS Applications

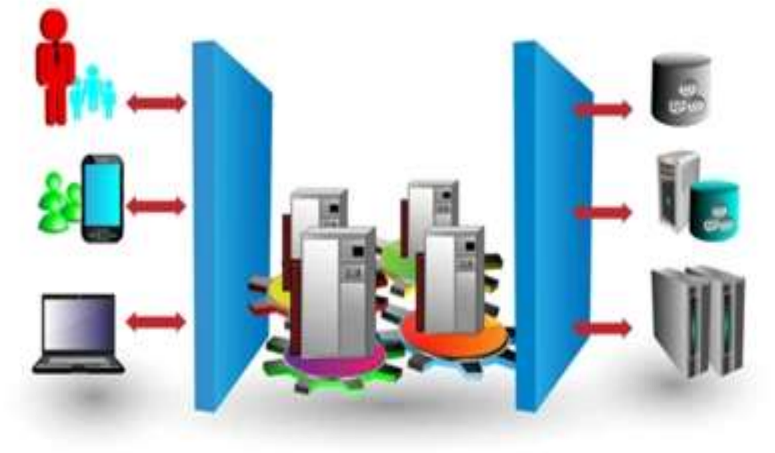


DBMS : Advantage

- **Improved Data Sharing**
- **Improved Data Security**
- **Easy retrieval of data**
- **Better Data Integration**
- **Minimized Data Inconsistency**
- **Improved Data Access**
- **Improved Decision Making**
- **Increased End-user Productivity**

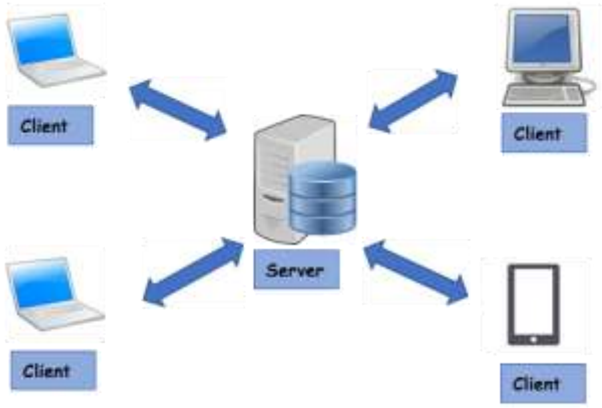
DBMS Architecture

- A database system can be centralized or decentralized.
- This depends on the architecture of the database.
- There are three types
  - Single tier or Host based
  - Two tier
  - Three tier



Client-Server Architecture

- **Client :**
  - User Interface to access a database
  - Also called as front end.
  - Sends request to the database and gets response.
- **Server:**
  - Holds database.
  - Processes the request by the client and gives response .



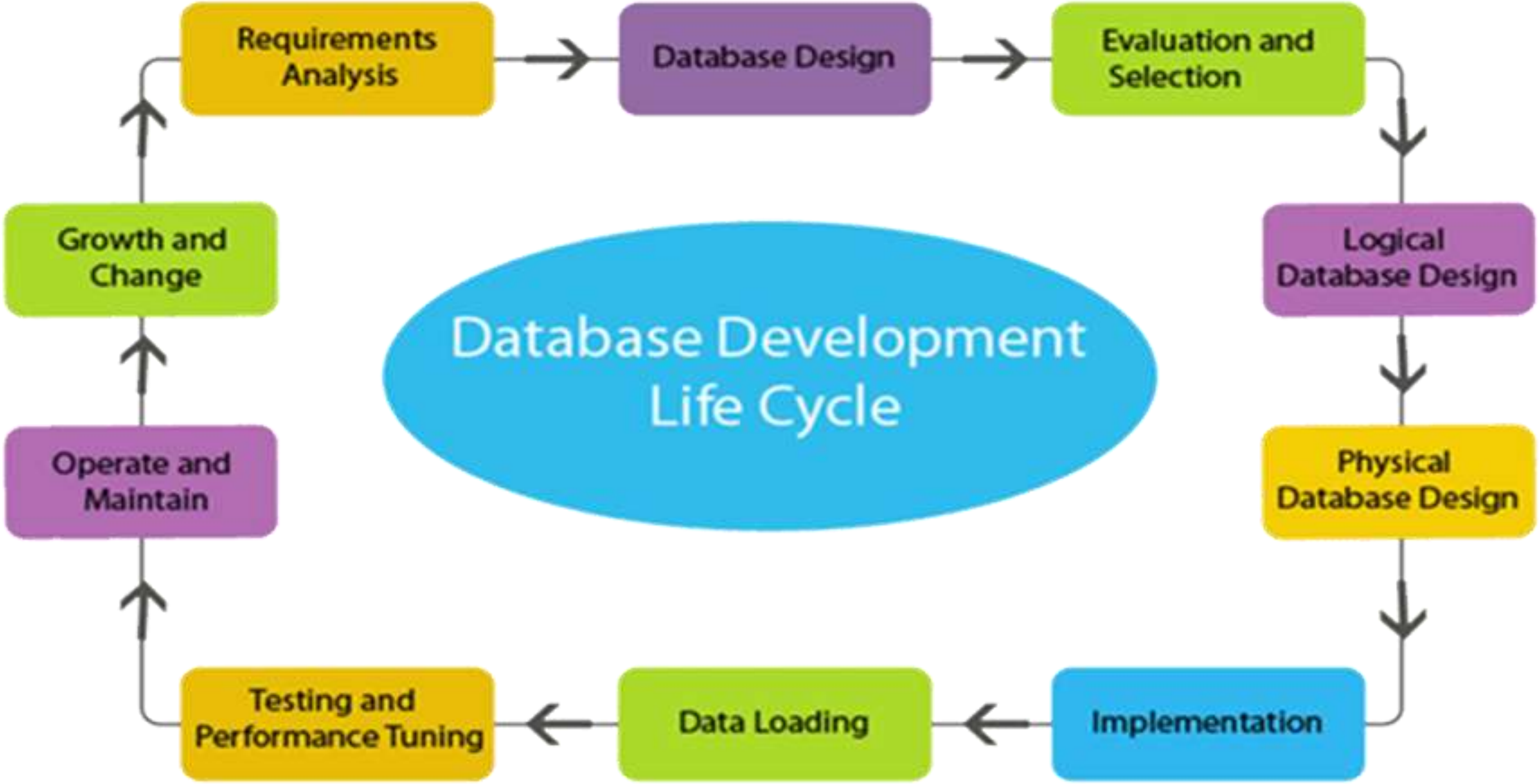
File System Vs Database

Basis	File System	DBMS
Structure	The file system is software that manages and organizes the files in a storage medium within a computer.	DBMS is software for managing the database.
Data Redundancy	Redundant data can be present in a file system.	In DBMS there is no redundant data.
Backup and Recovery	It doesn’t provide backup and recovery of data if it is lost.	It provides backup and recovery of data even if it is lost.
Query processing	There is no efficient query processing in the file system.	Efficient query processing is there in DBMS.
Consistency	There is less data consistency in the file system.	There is more data consistency because of the process of normalization.
Sharing	Data is distributed in many files. So, not easy to share data	Due to centralized nature sharing is easy

File System Vs Database

Basis	File System	DBMS
Security Constraints	File systems provide less security in comparison to DBMS.	DBMS has more security mechanisms as compared to file systems.
Cost	It is less expensive than DBMS.	It has a comparatively higher cost than a file system.
Data Independence	There is no data independence.	In DBMS data independence exists.
User Access	Only one user can access data at a time.	Multiple users can access data at a time.
Meaning	The user has to write procedures for managing databases	The user not required to write procedures.
Data Abstraction	It give details of storage and representation of data	It hides the internal details of Database
Integrity Constraints	Integrity Constraints are difficult to implement	Integrity constraints are easy to implement

Database Development process



Requirement Analysis

- It requires two steps
- **Planning**
  - The plan of the entire Database Development Life Cycle is decided, and the scope of the project and resource requirements are obtained.
- **Defining the system**
  - A layout of the proposed database system's scope is prepared.

Database Design

- Process of producing a detailed data model of a database containing all the needed logical and physical design choices
- **Logical model:**
  - Describes how the system to be implemented regardless of the DBMS.
- **Physical model:**
  - Describes how the system will be implemented using a specific DBMS.

Data Models

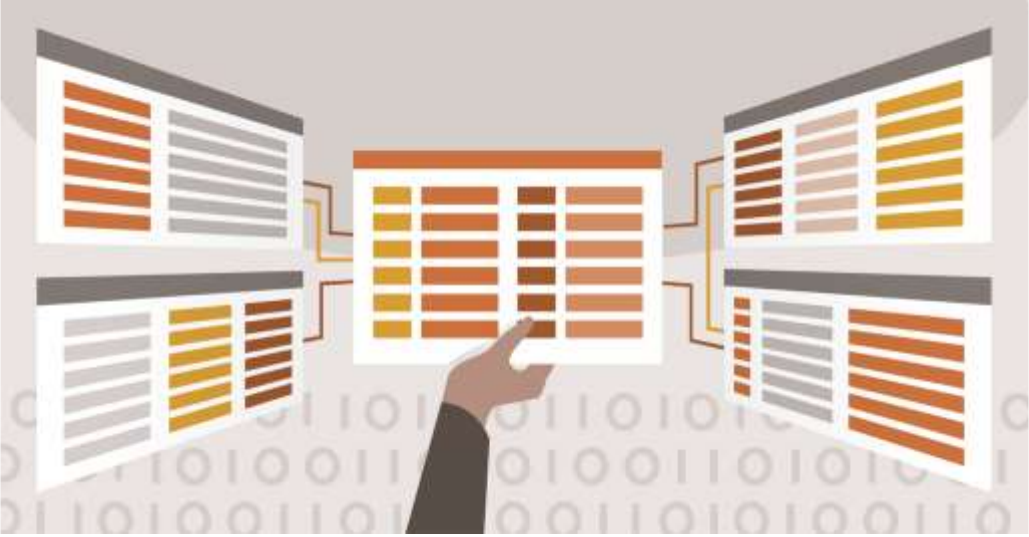
- A data model describes the logical structure of the database.
- Defines how data is stored and accessed in the database
- Types of data models
  - Hierarchical data model.
  - Network data model.
  - Object-oriented data model.
  - Entity-relationship model.
  - Relational model

The Relational Model

- Widely used model.
- Represents the database as a collection of relations.
- Relations are nothing but two-dimensional tables.
- Table comprises of rows and columns.
- Columns represent the fields or attributes, and rows represent the records.
- Relationships are represented by inter-related tables

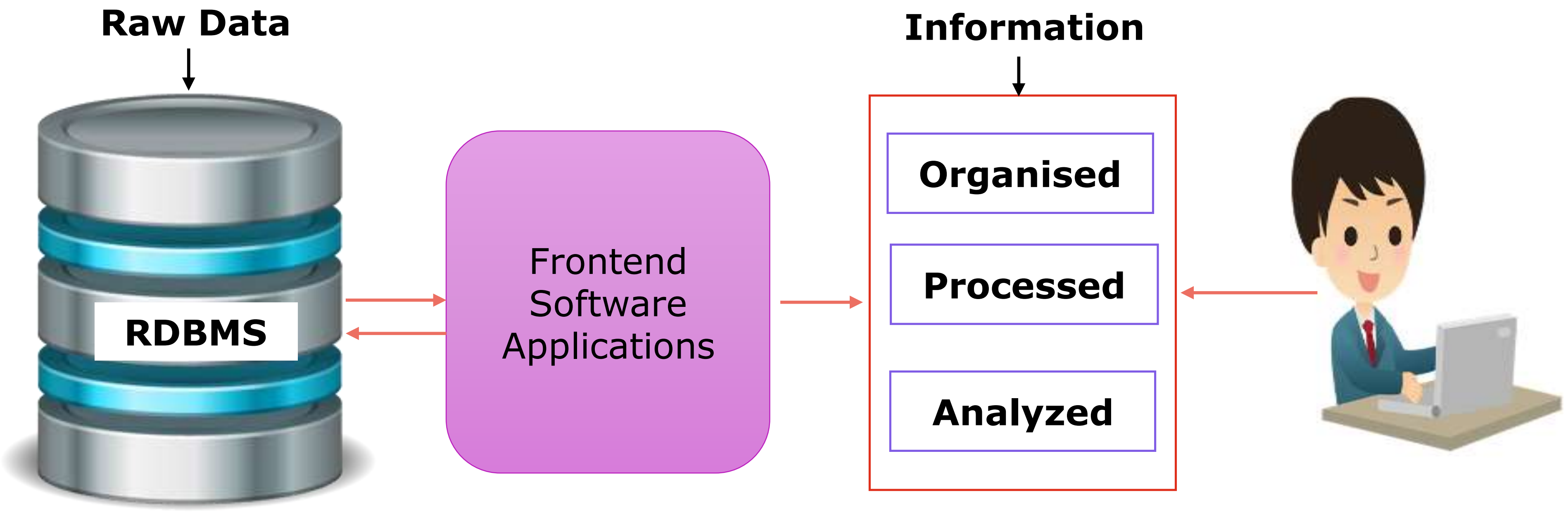
Building the Relational database

- **Steps**
  - Identify the entities.
  - Define the attributes.
  - Normalize the tables.
  - Determine the relationships between the entities.
  - Implement the model in the database



Relational Database Management System (RDBMS)

- A relational database management system is a type of database management system (DBMS) that **stores data in a row-based** table structure that **connects related data** elements.



Relational Database Management System (RDBMS)

- A RDBMS allows us to **create, read, update, Delete**(CRUD) and administer a relational database.
- Most relational database management systems use the **SQL language to access the database.**
- An RDBMS includes functions that maintain the **security, accuracy, integrity, and consistency of the data.**

Features of RDBMS

- All data stored in the **table structure** in the form of **rows and columns.**
- Facilitates with unique **identification of the rows**.
- **Index creation for retrieving data** at a higher speed
- Facilitates a **common column to be shared** amid two or more tables
- **Multi-user accessibility** is facilitated to be controlled by individual users
- A **virtual table creation** is enabled to store sensitive data and simplify queries

Popular Databases

- •Microsoft Access
- •MYSQL
- •Microsoft SQL Server
- •Oracle
- •File Maker
- •DB2
- •Ingres
- •PostgreSQL
- •SQLite
- •MongoDB
- •etc..

Levels of Abstraction

Main Purpose:

1) Provides user with an **abstract view of data**.
2) System **hides certain details** of how the data are actually stored and maintaine

Levels of Abstraction :

- **Physical level :** Lowest level of abstraction that **describes how the data** are **actually stored**.
- **Logical level:** Next higher level of abstraction that **describes what data are stored in database** and what relationships exist among those data.
- **View level:** Highest level of abstraction that **describes only part of entire database.**

Instance and Schema

- **Instance** - The data stored in **database** at a particular **moment of time** is called **instance** of **database**. It **changes over time** when we add or delete data from the database.
- **Schema – Design** of a database. Simply, schema is only a **structural view(design)** of a database, which is specified **during database design** and is **not expected to change frequently.**

Entity Relationship Model

- An Entity Relationship (ER) Model is a **type of flowchart** that illustrates how **entities** such as people, objects or concepts **relate to each other** within a system.
- ER Diagrams are most often used to **design or debug relational databases.**
- ER diagram represent the **real-world object** called **entities** and **relationship among the object.**
- They mirror grammatical structure, with **entities as nouns** and **relationships as verbs**.

Example:

/* Real World Entity : Banking System*/

Entity (Object) : Account ,Customer, Bank Branch.

Relationship : Account A-101 is held by customer Johnson

Entity Relationship Model

- ER diagrams are used to **model and design relational databases**, in terms of **logic and business rules** (in a logical data model) and in terms of the **specific technology to be implemented** (in a physical data model).
- ER diagrams are used to **analyze existing databases** to find and **resolve problems** in logic or deployment.

Symbols used for ER-Diagrams

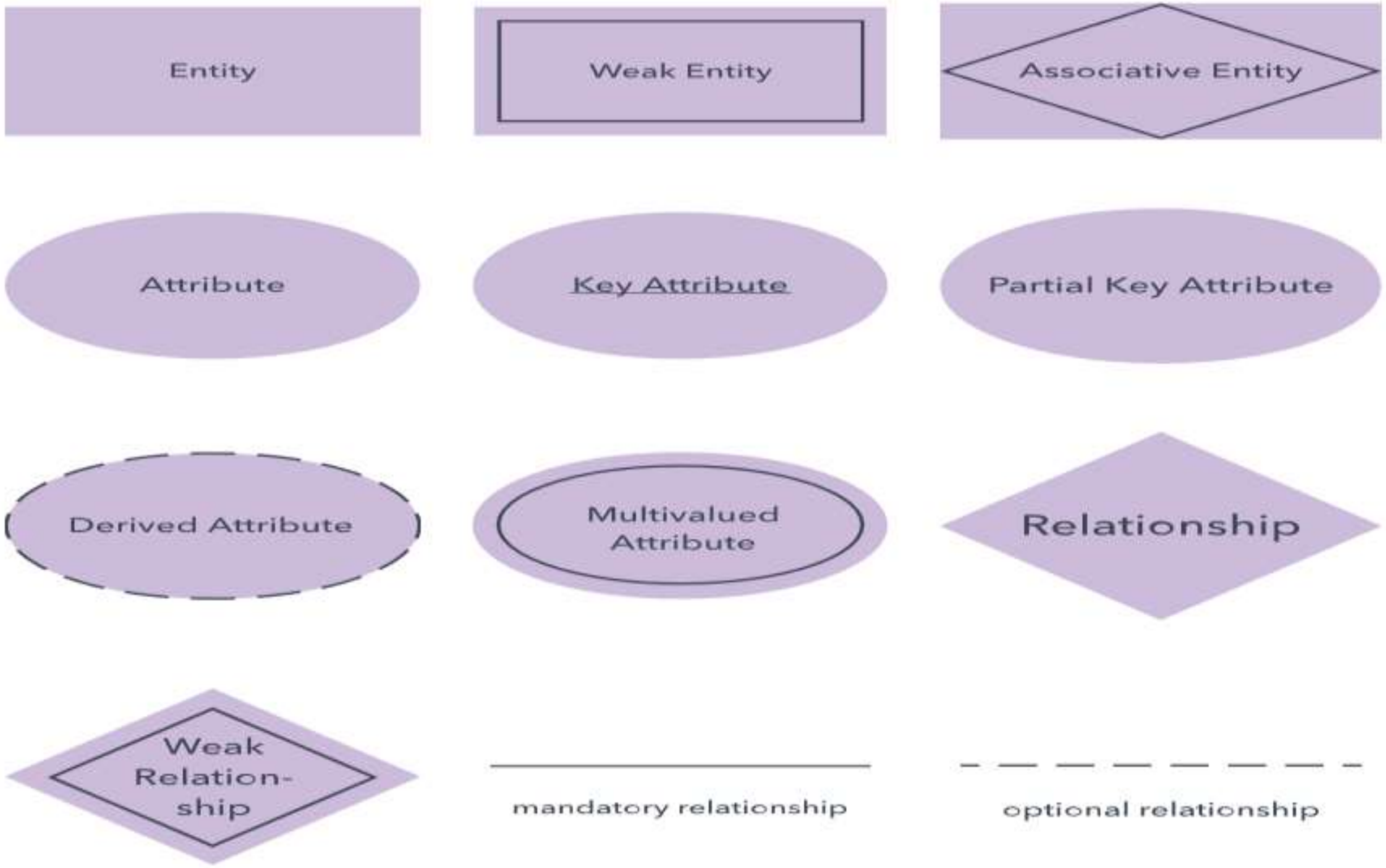
- There are **several notation systems**, which are similar but vary in a few specifics.

- Here we are going to illustrate the examples based on the following style

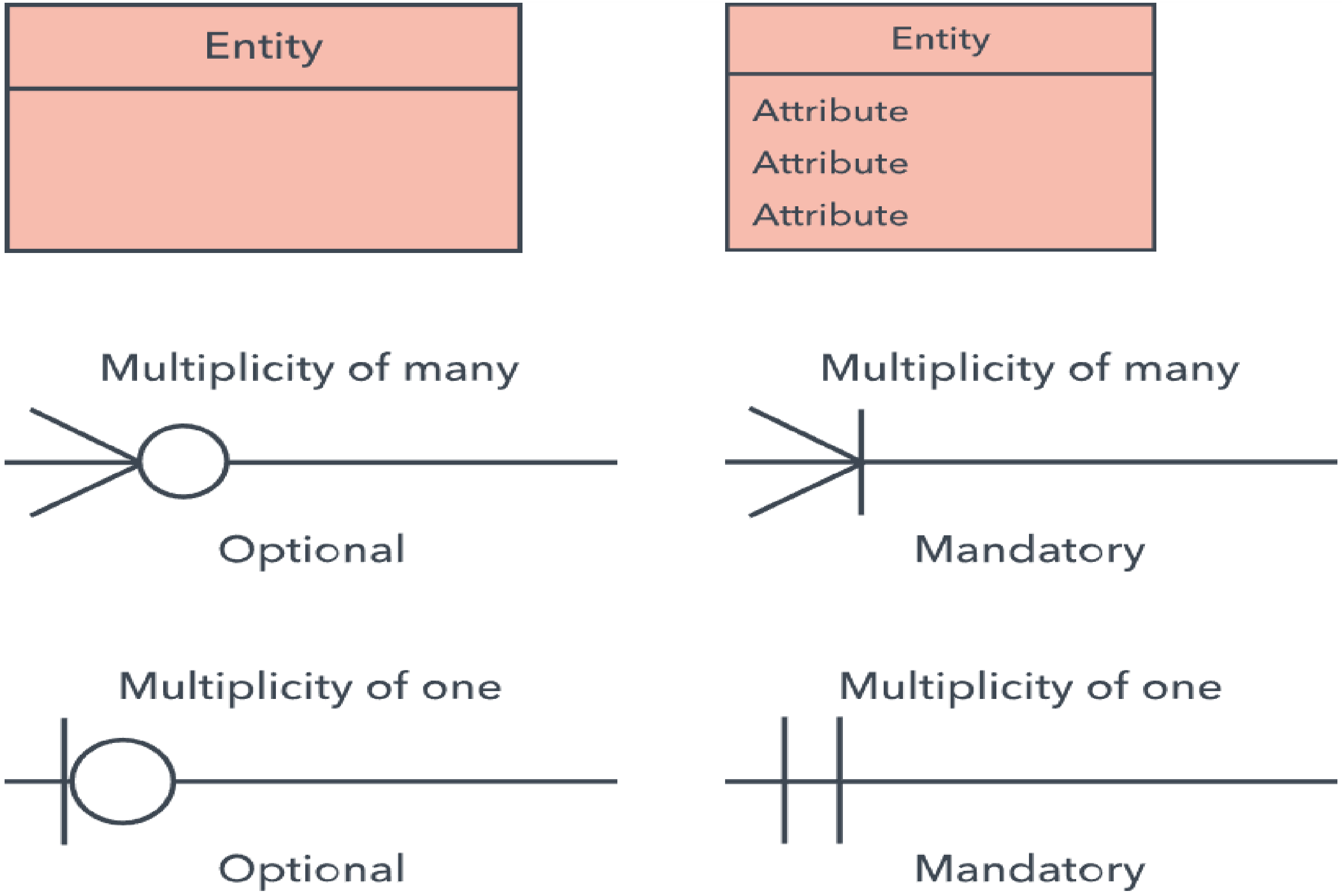
*** Chen Notation style**

* **Crow’s foot or Information engineering style.**

Symbols used for ER-Diagrams : Chen Notational Style



Symbols used for ER-Diagrams : Crow’s Foot or Information Engineering - style



Entity

Entity

Attribute

Attribute

Attribute

Multiplicity of many

Optional

Multiplicity of many

Mandatory

Multiplicity of one

Optional

Multiplicity of one

Mandatory

Component and Features of ER Diagram

- ER Diagrams are composed of following components

  1. **Entities.**

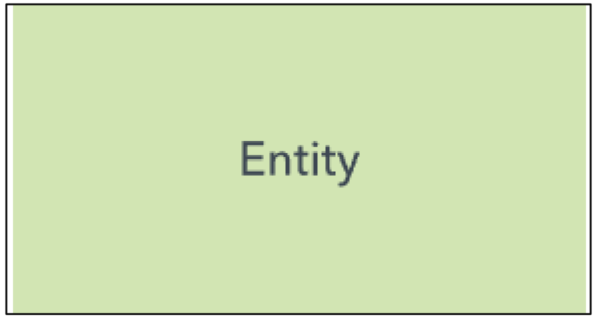
  2. **Attributes**.

  3. **Relationships.**

Component and Features of ER Diagram

1. **Entity:**

- A definable thing such as a **person, object, concept or event** that can have **data stored** about it.
- Define your entities as **nouns.**
- **Examples:** a customer, student, car or product.
- Entity in Entity relationship diagram will be denoted as **rectangle.**



Component and Features of ER Diagram

Entity type:

- A group of **definable things**, such as students or athletes, whereas the entity would be the specific student or athlete.
- **Other Examples:** customers, cars or products.

Entity Set :

- An entity set is a **set of entities of the same type.**
- **Examples:** Customers who purchased last month.

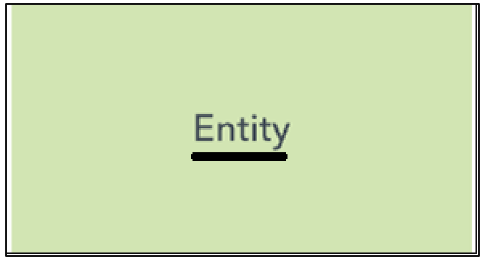
Component and Features of ER Diagram

Entity Categories:

- Entities are categorized as **strong, weak or associative**.

Strong Entity :

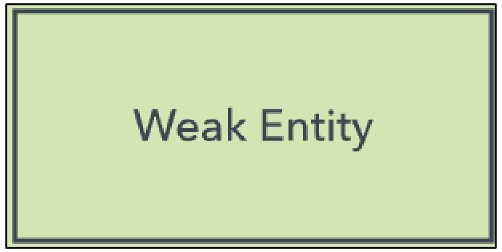
- Strong Entity is **independent of any other entity** in the schema.
- **Example:** A **student entity** can exist **without needing any other entity in the schema** or a **course entity** can exist **without needing any other entity in the schema.**



Component and Features of ER Diagram

Weak Entity :

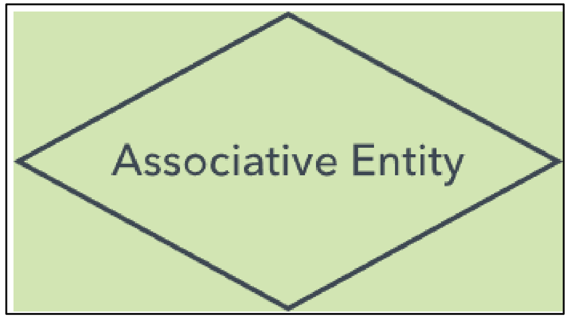
- A weak entity is an **entity set** that **does not have sufficient attributes** for **Unique Identification of its records.**
- **Example :**A **loan entity can not be created** for a customer **if the customer doesn’t exist**.
- **A double rectangle is used for representing a weak entity set.**



Component and Features of ER Diagram

Associative Entity:

- An **associative entity** associates **entities (or elements) within an entity set.**
- **Example:** Customer borrows loan.



Component and Features of ER Diagram

Entity keys:

- Entity keys refers to an **attribute that uniquely defines** an **entity** in an **entity set**.
- Entity keys are categories as follows.

1. **Primary key:** A candidate key chosen by the database designer to **uniquely identify** the entity set.
2. **Foreign key:** Identifies the **relationship** between entities.

Component and Features of ER Diagram

- **A primary key** as an **attribute** (or a collection of attributes) that can **uniquely identify** each **row of a database table.**
- **T**he primary keys can **uniquely identify** the other columns of a table **therefore** we **cannot have two records having the same value of the primary key field that is no duplicate records.**
- **T**he primary keys are used to **fetch the records** from the database, the **primary keys cannot be NULL**.
- That is primary key **should follow** the **combination of two constraints** namely **NOT NULL and UNIQUE**.

Component and Features of ER Diagram

- The **selection** of the right **primary key** is one of the most important aspects of designing the **overall database**.
- The primary key is used in the **retrieval** of query results in the **fastest** way possible.
- **Example: Student Table**

Roll_No	CGPA	Name	Section
1	9.21	Aman	A
2	8.86	Mohit	A
13	9.21	Sushant	C
24	9.71	Mohit	-
5	8.56	Saumya	C

Component and Features of ER Diagram

- A **foreign key** is an **attribute** in a table whose **values are referenced from a primary key** in **another table.**
- The foreign key establishes the **relationship between** the **two tables** and enforces **referential integrity in the SQL Server**.
- **Example:** The Department table has a primary key Department number, and this will be the foreign key of the Employee table.
- The **table** containing the **primary key** is known as the **parent table**, whereas the table which contains the **foreign key** is known as the **child table**.

Component and Features of ER Diagram

- Let's say we have a table named **student** which contains the **Names, Roll Numbers, and Percentages** of the students in a University.
- The table is as shown below:

Roll No	Name	Percentage
2019021001	Shefali	88
2019021002	Atul	92
2019021003	Koyal	87
2019021004	Aditya	94

Component and Features of ER Diagram

The Roll No column acts as a **Primary key** in the **student** table. It uniquely identifies a particular row in the table. We have another table named **Course** which contains the **Roll No** and the **Name** of the Course in which a particular student is enrolled.

The **Course** table is as shown below-

SNo	Roll No	Course	Duration
1	2019021001	CSE	4
2	2019021002	ECE	6
3	2019021003	ME	3
4	2019021004	CE	2

The **Roll No column** in the **Course table** is used as a **foreign key** that references the **student** table's primary key. Therefore, the **Roll No** column acts as a foreign key.

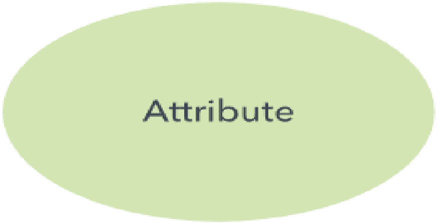
Component and Features of ER Diagram

- Now, **in the student table,** there are **no duplicate values, and no null value** present in the **Roll_No** column.
- But, the Name column, the CGPA column, and the Section column contain some duplicate values.
- So, we should use the **Roll_No column** as the **primary key** because Roll_No has both the properties of a primary key. (i.e. UNIQUE and NOT NULL).

Component and Features of ER Diagram

2. Attribute:

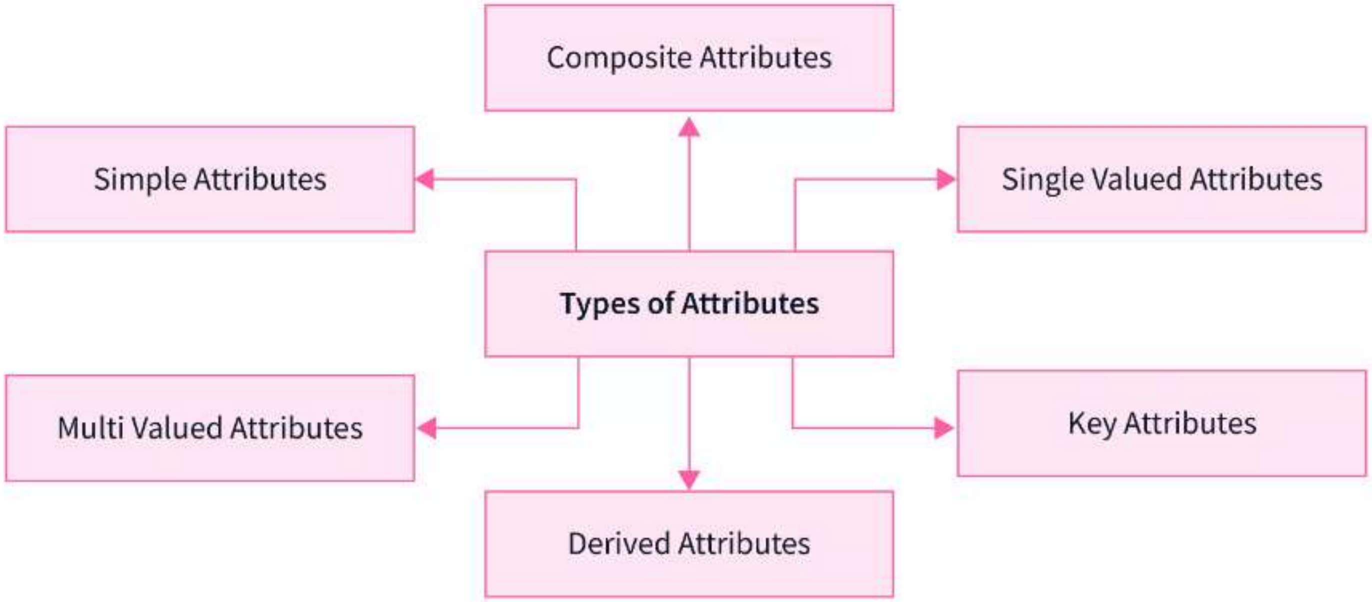
- An attribute describes the **property of an entity.**
- An attribute is represented as **Oval** in an ER diagram.



Component and Features of ER Diagram

Types of attribute:

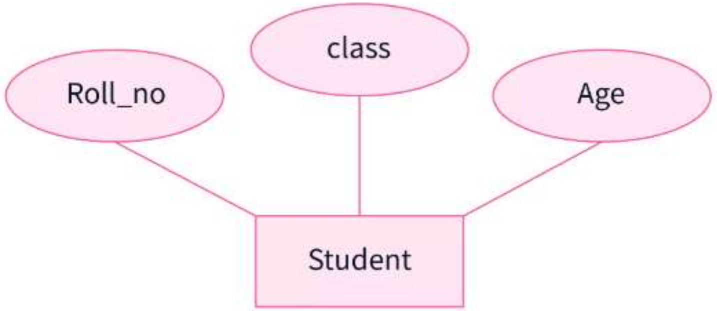
- Attribute can be categorized as follows



Component and Features of ER Diagram

a) Simple Attribute :

- Simple attributes in an ER model diagram are **independent attributes** that **can't be classified** further and also, can't be subdivided into any other component.
- These attributes are also known as **atomic attributes.**
- **Example:**



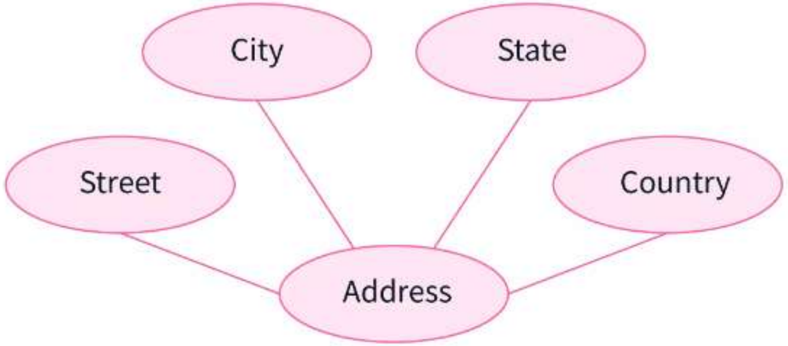
Component and Features of ER Diagram

- Student is an entity represented by a rectangle, and it consists of attributes: Roll_no, class, and Age.
- Also, there is a point to be noted that we can't further subdivide the Roll_no attribute and even the other two attributes into sub-attributes.

Component and Features of ER Diagram

b) Composite Attribute:

- Composite attributes have opposite functionality to that of simple attributes as we can further **subdivide composite attributes** into **different components** or sub-parts that form simple attributes.
- In simple terms, **composite attributes are composed of one or more simple attributes.**
- **Example:**



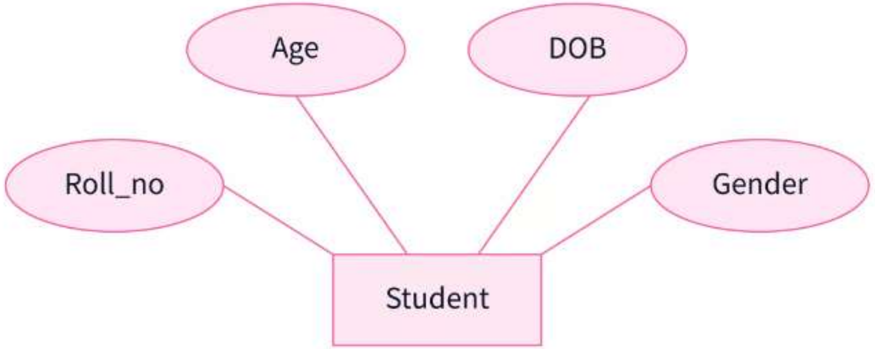
Component and Features of ER Diagram

- **Address is a composite attribute** represented by an elliptical shape, and it can be further subdivided into many simple attributes like Street, City, State, Country, Landmark, etc.

Component and Features of ER Diagram

c) Single valued Attribute:

- Single valued attributes are those attributes that consist of a **single value** for each entity instance and can't store more than one value.
- The value of these single-valued attributes always remains the same, just like the name of a person.
- **Example:**



Component and Features of ER Diagram

- Student is an entity instance, and it consists of attributes: Roll_no, Age, DOB, and Gender.
- These attributes can store only one value from a set of possible values.
- Each entity instance can have only one Roll_no, which is a unique, single DOB by which we can calculate age and also fixed gender.
- Also, we can't further subdivide these attributes, and hence, they are **simple as well as single-valued attributes.**

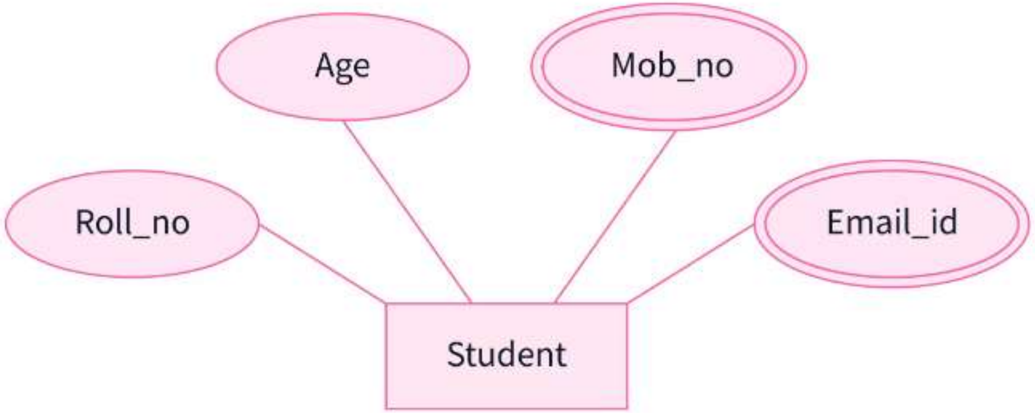
Component and Features of ER Diagram

d) Multi valued Attribute:

- Multi-valued attributes have **opposite functionality** to that of **single-valued attributes**, and as the name suggests, multi-valued attributes can take up and **store more than one value at a time** for an entity instance from a set of possible values.
- These attributes are represented by **co-centric elliptical shape**, and we can also use **curly braces { }** to represent multi-valued attributes inside it.

Component and Features of ER Diagram

- **Example:**



- The Student entity has four attributes: Roll_no and Age are simple as well as single-valued attributes as discussed above but Mob_no and Email_id are represented by co-centric ellipse **are multi-valued attributes.**

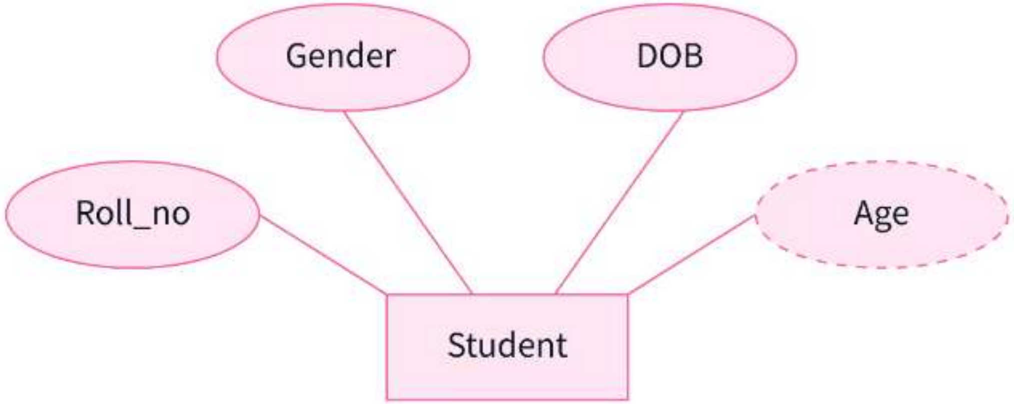
Component and Features of ER Diagram

- Each student in the real world can provide **more than one email-id** as well as a **mobile contact number,** and therefore, we need these attributes to be **multi-valued** so that they **can store multiple values at a time for an entity instance.**

Component and Features of ER Diagram

d) Derived Attribute:

- Derived attributes are those attributes whose **values can be derived** from the **values of other attributes.**
- They are always **dependent upon other attributes for their value.**
- **Example:**



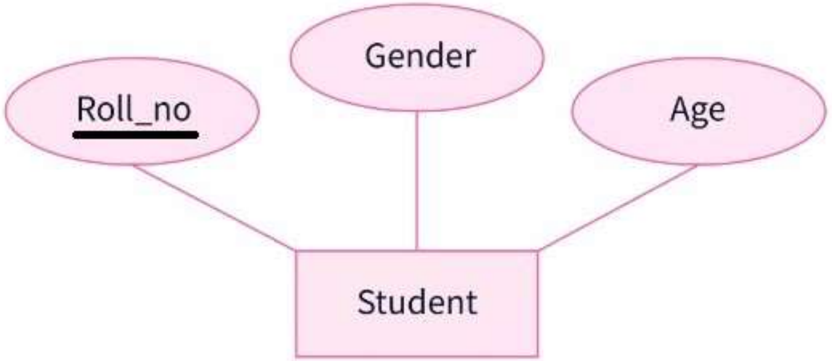
Component and Features of ER Diagram

- DOB is a **single-valued attribute** and remains constant for an entity instance.
- From DOB, we can **derive the Age attribute**, which changes every year, and can easily calculate the age of a person from his/her date of birth value.
- Hence, the Age attribute here is derived attribute from the DOB single-valued attribute.

Component and Features of ER Diagram

e) Key Attribute:

- Key attributes are special types of attributes that act as the **primary key** for an entity, and they can **uniquely identify an entity from an entity set**.
- The values that key attributes store must be **unique and non-repeating**.
- **Example:**



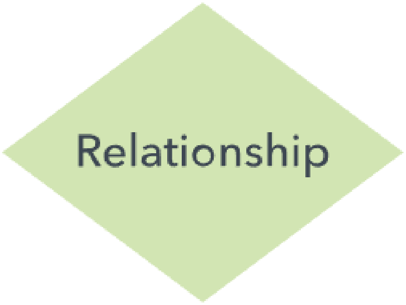
Component and Features of ER Diagram

- **Roll_no** attribute of the **Student entity** is not only simple and **single-valued attribute** but also, a key valued attribute as well.
- Roll_no of a student will always be **unique to identify the student**.
- Also note that the Gender and Age of two or more persons can be same and overlapping in nature and obviously, we can't identify a student on the basis of them.
- Hence, gender and age are not key-valued attributes.

Component and Features of ER Diagram

3. Relationship:

- Relationship describes how **entities act upon each other** or are **associated with each other**.
- Relationships can be described as verbs.
- For **example**, the **student might register for a course**. The two entities would be the student and the course, and the relationship depicted is the act of enrolling, connecting the two entities in that way.
- Relationships are typically shown as **diamonds** or labels directly on the connecting lines.



Component and Features of ER Diagram

Degree of a Relationship Set

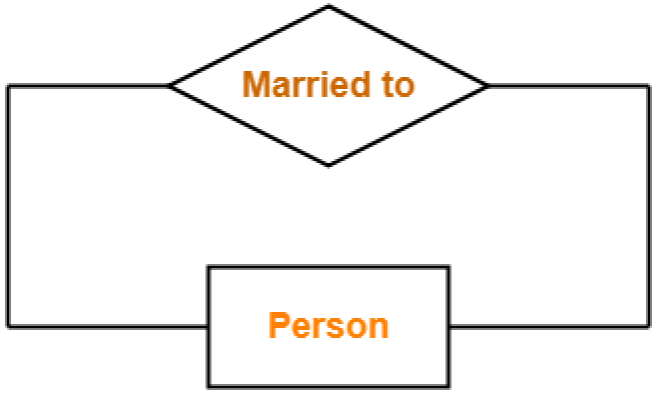
- Number of entity sets that **participate in a relationship set**.
- Relationship sets that involve **two entity sets are binary** (or degree two). Generally, **most relationship** sets in a database system are **binary.**
- Relationship sets may involve **more than two entity sets**.

Types:

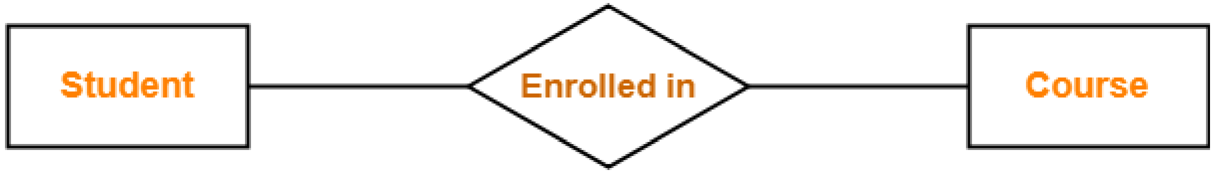
1) Unary Relationship
2) Binary Relationship
3) Ternary Relationship
4) Quaternary Relationship

Component and Features of ER Diagram

Unary Relationship

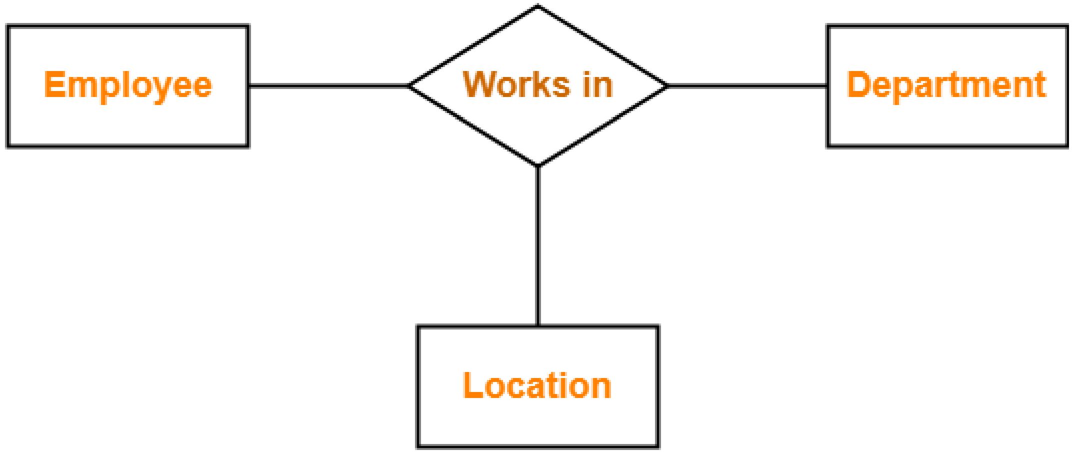


Binary Relationship



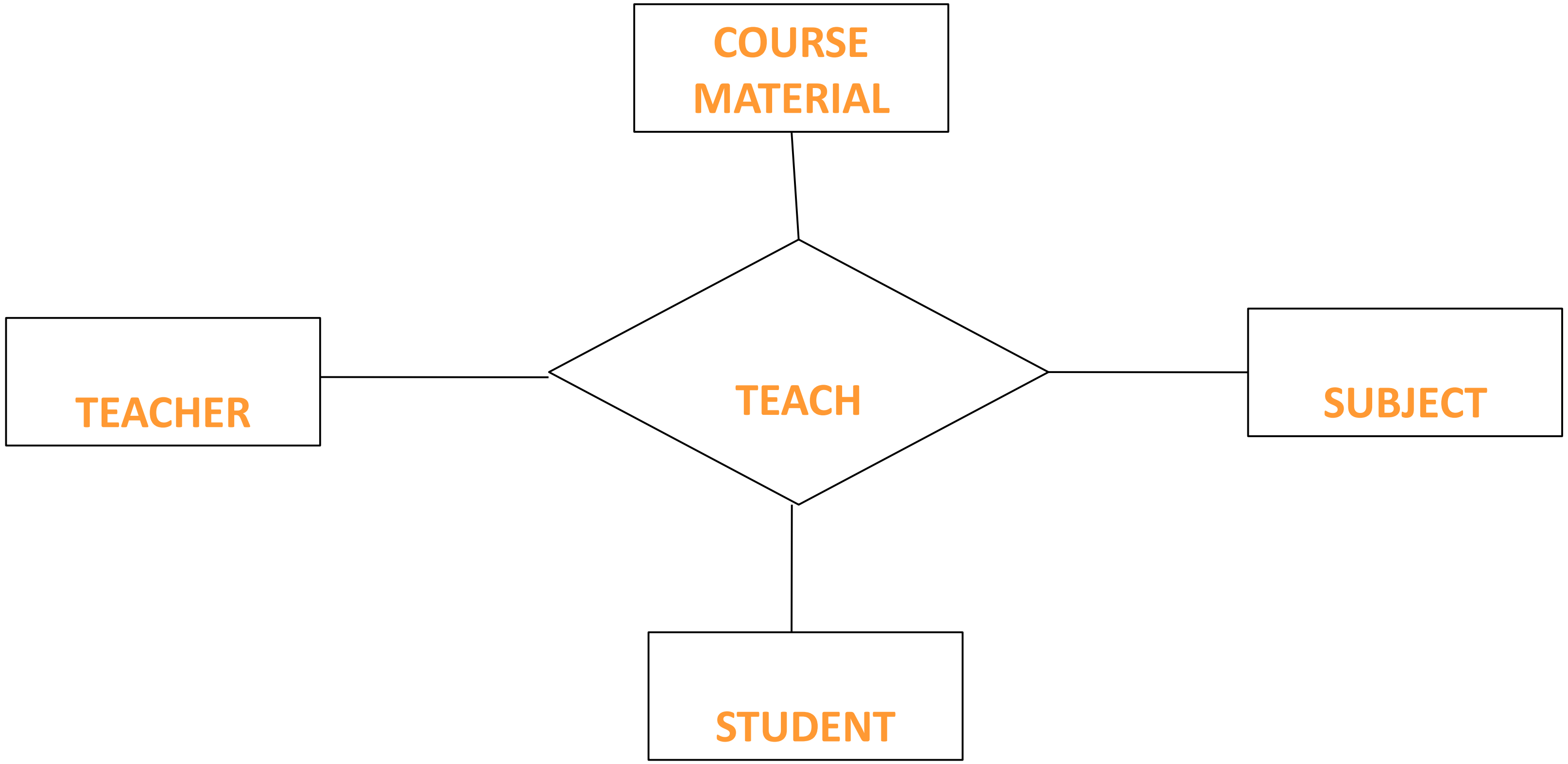
Component and Features of ER Diagram

Ternary Relationship



Component and Features of ER Diagram

Quaternary Relationship



Component and Features of ER Diagram

Cardinality Constraint:

- Express the **number of entities** to which another entity can be associated via a relationship set.
- **Cardinality constraint** defines the **maximum number of relationship** instances in which an entity can participate.
- For a **binary relationship set the mapping cardinality** must be one of the following **types**:

1) **One to one (1:1)**
2) **One to many (1:n)**
3) **Many to one (m:1)**
4) **Many to many (m:n)**

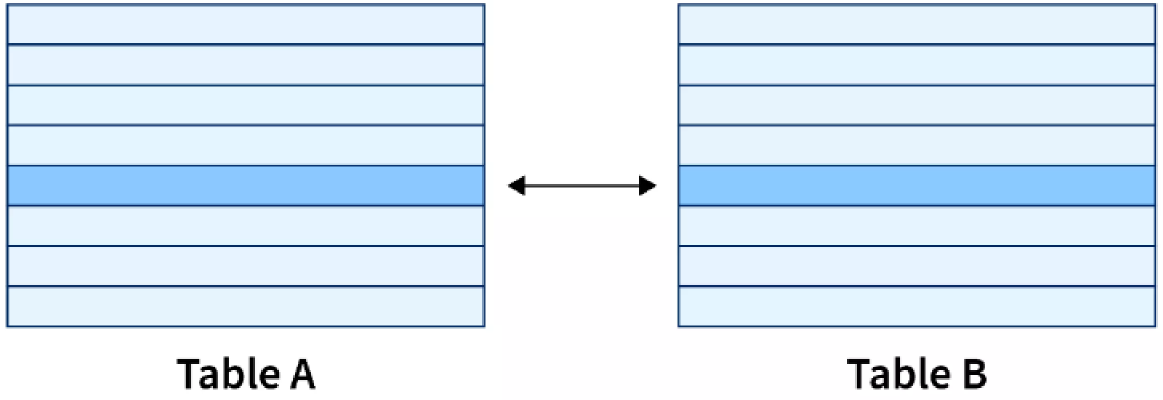
Component and Features of ER Diagram

- Cardinality constraints are expressed by drawing either a **directed line (→), signifying** "**one**," or an **undirected line (—), signifying "many**," between the relationship **set and the entity set**.

Component and Features of ER Diagram

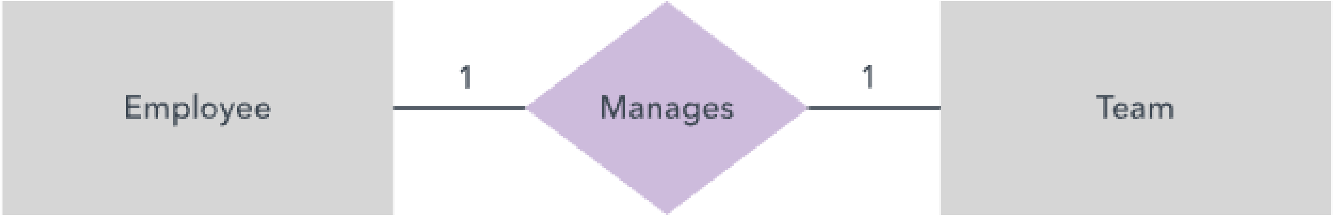
1) One to one (**1:1**):

- **One entity associated with at most one entity** that is one-to-one relationship means a single record in Table A is related to a single record in Table B and vice-versa.
- We can visualize the one-one relationship as below



Component and Features of ER Diagram

- **For example,** If there are two entities, Employee and Team both the entities have the association of one to one ,(i.e)., Each employee manages a team.



Component and Features of ER Diagram

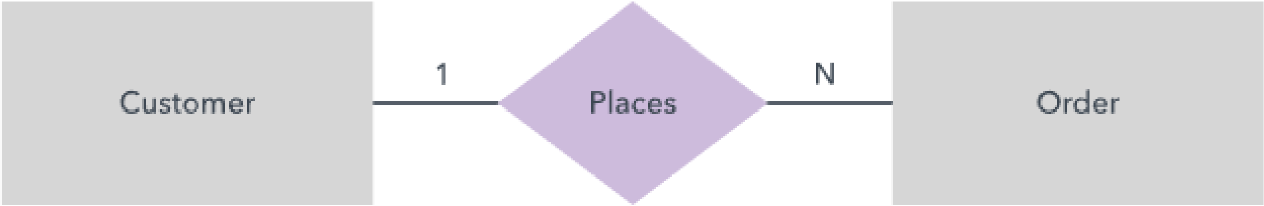
2) One to Many (**1:M**):

- **One entities** associated with **many entity.**
- This relationship exists when **each record of table A** can be related to **one or more records of another table, i.e., table B.**
- However, a single record in table B will have a link to a single record in table A.
- A one-to-many relationship in DB can also be named a **many-to-one relationship**, depending on how we view it.
- The one-to-many relationship in DB exists between the pair of tables if a single record in one table is related to one or more records of the other table.

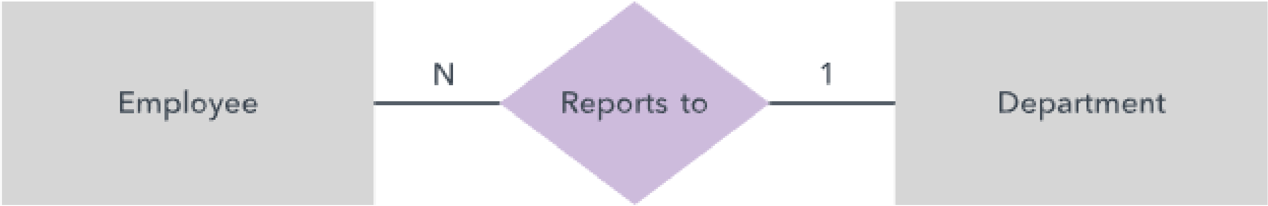
Component and Features of ER Diagram

- **For example**, if there are two entities, 'Customer' and ‘Order', then each customer can place more than one order, and many employee can belong to one department.

One to Many Relationship



Many to One Relationship



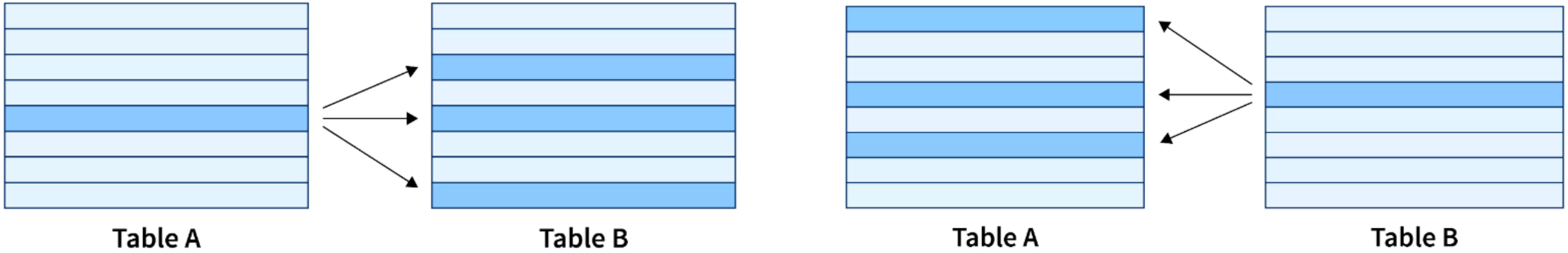
Component and Features of ER Diagram

Many to Many (**M:M**):

- **Many** entities associated with **any number of** entities**.**
- A many-to-many relationship exists between the tables if **a single record of the first table** is related to **one or more records of the second table** and **a single record in the second table is related to one or more records of the first table.**
- Consider the tables A and B. In a many-to-many relationship, each record in table A can be linked to one or more records in table B and vice-versa. It is also represented as an N: N relationship.

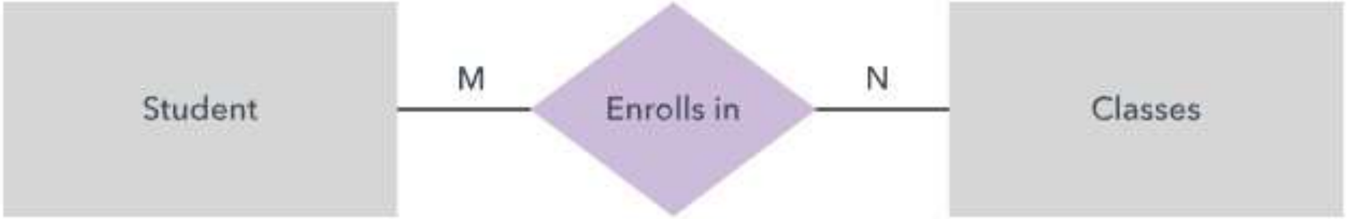
Component and Features of ER Diagram

- We can visualize the many-to-many relationship as below
- **A single record of the Table A** is related to **one or more records of the Table B** and **a single record in the Table B is related to Table A.**



Component and Features of ER Diagram

- **For example,** consider the two tables, i.e., a student table and a classes table.
- A particular student may enroll himself in one or more than one course, while a course also may have one or more students.



Entity Relationship Diagram : Example

- To Illustrate the Entity relationship diagram we have taken the application – **Online Shopping cart.**

A shopping cart on an online retailer's site facilitates the purchase of a product or service. The customer can be able to have a look at all products or search for the needed product that they want to pick up in one place. A shopping cart serves the customer to add the product that they are likely to buy and store them for purchase in the future. The shopping cart accepts the customer's payment and organizes the distribution of that information to the merchant, payment processor, and other parties.

Entity Relationship Diagram : Example

Step 1: Complete analysis of the requirement to build the Entity relationship diagram.

1. The system allows multiple customers to log in and browse the products.
2. Each product in the online shop has a unique ID or code.
3. Customers can purchase products from multiple sections and add them to their carts.
4. Customers can pay for their purchases through debit/ credit cards or cash.
5. The system should keep track of each item with its unique code, price and product name.

Entity Relationship Diagram : Example

6. The system can track each customer and record their phone number, address and payment information.

7. The user using the software should be able to download the customer’s information and shipments along with payment receipts.

8. The customer should be able to track their order through the unique shipment code. They should be able to see and download their receipts.

9.The customers should be able to contact support in case of any delays or problems.

Entity Relationship Diagram : Example

Step 2 : Based on your analysis identify the entity set related to the ERD .

- The entities for the online shopping cart is

1. Categories.
2. Shopping order.
3. Deliveries.
4. Products.
5. Customers.
6. Transaction Reports.
7. Seller.
8. Payment

Entity Relationship Diagram : Example

Step 3 : We need to define attribute to each identified entities .

1. Categories : category_ID, category_name, category_type

2. Shopping order :Order_ID,date

3.Deliveries: Deliveries_ID ,date

4.Products: Product_ID,Product_Name

5.Customers: Customer_ID,name,Contact_no,Address.

6.Transaction Report: Report_ID

7.Seller: Seller_ID,Name.

8.Payment:Payment_ID,Date.

Entity Relationship Diagram : Example

Step 4 : Identify the key attribute to specify the uniquely identifies one entity from another in entity set and Identify the foreign key to establish the relationship between the entities in entity set.

- The key attribute of the entities are as follows

1. Categories : category_ID
2. Shopping order :Order_ID
3. Deliveries: Deliveries_ID
4. Products: Product_ID
5. Customers: Customer_ID.
6. Transaction Report: Report_ID.
7. Seller: Seller_ID.
8. Payment:Payment_ID.

Entity Relationship Diagram : Example

- The Foreign key attribute to establish the relationship between entities(Based on the requiremnent) are as follows

1. Shopping order :Customer_ID.

2. Deliveries: Customer_ID.

3. Products: category _ID

4.Transaction Report: Customer_ID, Order_ID,Product_ID,Payment _ID.

5. Seller:Product_ID

6. Payment: category_ID

Entity Relationship Diagram : Example

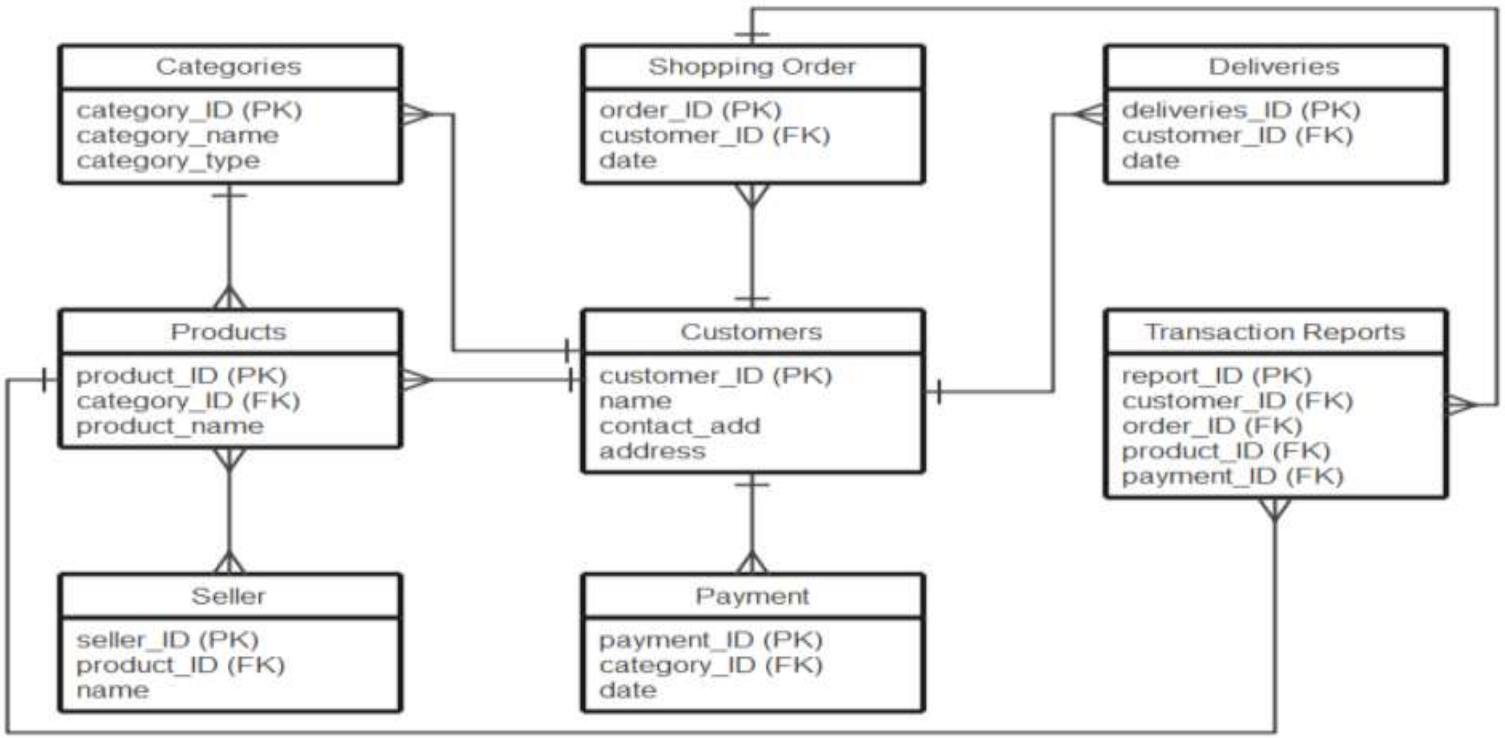
Step 5 : Identify the relationship that exists between the entities.

- Based on the association find the relationship between the entities with cardinality constraint as one to one, one to many, many to one, many to many.
- **Example:** Category {M..1} Customer,

  Products {M..N} Seller

Entity Relationship Diagram : Example

Step 6 : Draw the ER diagram based on the above identification of entities ,attributes and relationship as follows



Categories
category_ID (PK)
category_name
category_type

Shopping Order
order_ID (PK)
customer_ID (FK)
date

Deliveries
deliveries_ID (PK)
customer_ID (FK)
date

Products
product_ID (PK)
category_ID (FK)
product_name

Customers
customer_ID (PK)
name
contact_add
address

Transaction Reports
report_ID (PK)
customer_ID (FK)
order_ID (FK)
product_ID (FK)
payment_ID (FK)

Seller
seller_ID (PK)
product_ID (FK)
name

Payment
payment_ID (PK)
category_ID (FK)
date

THANK YOU

SmartCliff
Career Mobility Solutions