FUNCTIONAL DEPENDANCY:- The functional dependency is a relationship that exists between two attributes. It typically exists between the primary key and non-key attribute within a table.

X->Y

The left side of FD is known as a determinant, the right side of the production is known as a dependent.

FOR EX :- Assume we have an employee table with attributes: Emp\_Id, Emp\_Name, Emp\_Address.

Here Emp\_Id attribute can uniquely identify the Emp\_Name attribute of employee table because if we know the Emp\_Id, we can tell that employee name associated with it.

Functional dependency can be written as:

1. Emp\_Id → Emp\_Name

We can say that Emp\_Name is functionally dependent on Emp\_Id.

TYPES OF FUNCTIONL DEPENDANCY:

1. TRIVAL

* A → B has trivial functional dependency if B is a subset of A.
* The following dependencies are also trivial like: A → A, B → B

Example:- Consider a table with two columns Employee\_Id and Employee\_Name.

{Employee\_id, Employee\_Name}   →    Employee\_Id is a trivial functional dependency as

Employee\_Id is a subset of {Employee\_Id, Employee\_Name}.

Also, Employee\_Id → Employee\_Id and Employee\_Name   →    Employee\_Name are trivial dependencies too.

1. NON-TRIVAL

* A → B has a non-trivial functional dependency if B is not a subset of A.
* When A intersection B is NULL, then A → B is called as complete non-trivial.

Example:-

ID   →    Name,

Name   →    DOB

Inference Rule (IR):

* The Armstrong's axioms are the basic inference rule.
* Armstrong's axioms are used to conclude functional dependencies on a relational database.
* The inference rule is a type of assertion. It can apply to a set of FD(functional dependency) to derive other FD.
* Using the inference rule, we can derive additional functional dependency from the initial set.

The Functional dependency has 6 types of inference rule:

## **1. Reflexive Rule (IR1)**

In the reflexive rule, if Y is a subset of X, then X determines Y.

Ex :- If X ⊇ Y then X  →    Y

## **2. Augmentation Rule (IR2)**

The augmentation is also called as a partial dependency. In augmentation, if X determines Y, then XZ determines YZ for any Z.

1. If X    →  Y then XZ   →   YZ

**Example:**

1. For R(ABCD),  **if** A   →   B then AC  →   BC

## **3. Transitive Rule (IR3)**

In the transitive rule, if X determines Y and Y determine Z, then X must also determine Z.

1. If X   →   Y and Y  →  Z then X  →   Z

## **4. Union Rule (IR4)**

Union rule says, if X determines Y and X determines Z, then X must also determine Y and Z.

1. If X    →  Y and X   →  Z then X  →    YZ

**Proof:**

1. X → Y (given)  
2. X → Z (given)  
3. X → XY (using IR2 on 1 by augmentation with X. Where XX = X)  
4. XY → YZ (using IR2 on 2 by augmentation with Y)  
5. X → YZ (using IR3 on 3 and 4)

## **5. Decomposition Rule (IR5)**

Decomposition rule is also known as project rule. It is the reverse of union rule.

This Rule says, if X determines Y and Z, then X determines Y and X determines Z separately.

1. If X   →   YZ then X   →   Y and X  →    Z

**Proof:**

1. X → YZ (given)  
2. YZ → Y (using IR1 Rule)  
3. X → Y (using IR3 on 1 and 2)

## **6. Pseudo transitive Rule (IR6)**

In Pseudo transitive Rule, if X determines Y and YZ determines W, then XZ determines W.

1. If X   →   Y and YZ   →   W then XZ   →   W

**Proof:**

1. X → Y (given)  
2. WY → Z (given)  
3. WX → WY (using IR2 on 1 by augmenting with W)  
4. WX → Z (using IR3 on 3 and 2)

**NORMALIZATION:-**

* Normalization is the process of organizing the data in the database.
* Normalization is used to minimize the redundancy from a relation or set of relations. It is also used to eliminate the undesirable characteristics like Insertion, Update and Deletion Anomalies.
* Normalization divides the larger table into the smaller table and links them using relationship.
* The normal form is used to reduce redundancy from the database table.

## **Types of Normal Forms**

There are the four types of normal forms:

1. 1NF
2. 2NF
3. 3NF
4. BCNF

1NF:- A relation is in 1NF if it contains an atomic value.

2NF:- A relation will be in 2NF if it is in 1NF and all non-key attributes are fully functional dependent on the primary key.

3NF:- A relation will be in 3NF if it is in 2NF and no transition dependency exists.

# ER model

* ER model stands for an Entity-Relationship model. It is a high-level data model. This model is used to define the data elements and relationship for a specified system.
* It develops a conceptual design for the database. It also develops a very simple and easy to design view of data.
* In ER modeling, the database structure is portrayed as a diagram called an entity-relationship diagram.

## **Component of ER Diagram**

1. Entity
2. Weak entity
3. Relation

a)one to one

b)one to many

c)many to one

d) many to many

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

### **1. Entity:**

An entity may be any object, class, person or place. In the ER diagram, an entity can be represented as rectangles.

Consider an organization as an example- manager, product, employee, department etc. can be taken as an entity.

**WEAK ENTITY:-** An entity that depends on another entity called a weak entity. The weak entity doesn't contain any key attribute of its own. The weak entity is represented by a double rectangle.

### **2. Attribute**

The attribute is used to describe the property of an entity. Eclipse is used to represent an attribute.

**a. Key Attribute**

The key attribute is used to represent the main characteristics of an entity. It represents a primary key. The key attribute is represented by an ellipse with the text underlined.

**b. Composite Attribute**

An attribute that composed of many other attributes is known as a composite attribute. The composite attribute is represented by an ellipse, and those ellipses are connected with an ellipse.

Ex name has three parts first name, middle name. last name.

**c. Multivalued Attribute**

An attribute can have more than one value. These attributes are known as a multivalued attribute. The double oval is used to represent multivalued attribute.

**For example,** a student can have more than one phone number.

**d. Derived Attribute**

An attribute that can be derived from other attribute is known as a derived attribute. It can be represented by a dashed ellipse.

**For example,** A person's age changes over time and can be derived from another attribute like Date of birth.

### **3. Relationship**

A relationship is used to describe the relation between entities. Diamond or rhombus is used to represent the relationship.

**a. One-to-One Relationship**

When only one instance of an entity is associated with the relationship, then it is known as one to one relationship.

**For example,** A female can marry to one male, and a male can marry to one female.

**b. One-to-many relationship**

When only one instance of the entity on the left, and more than one instance of an entity on the right associates with the relationship then this is known as a one-to-many relationship.

**For example,** Scientist can invent many inventions, but the invention is done by the only specific scientist.

**c. Many-to-one relationship**

When more than one instance of the entity on the left, and only one instance of an entity on the right associates with the relationship then it is known as a many-to-one relationship.

**For example,** Student enrolls for only one course, but a course can have many students.

**d. Many-to-many relationship**

When more than one instance of the entity on the left, and more than one instance of an entity on the right associates with the relationship then it is known as a many-to-many relationship.

**For example,** Employee can assign by many projects and project can have many employees.