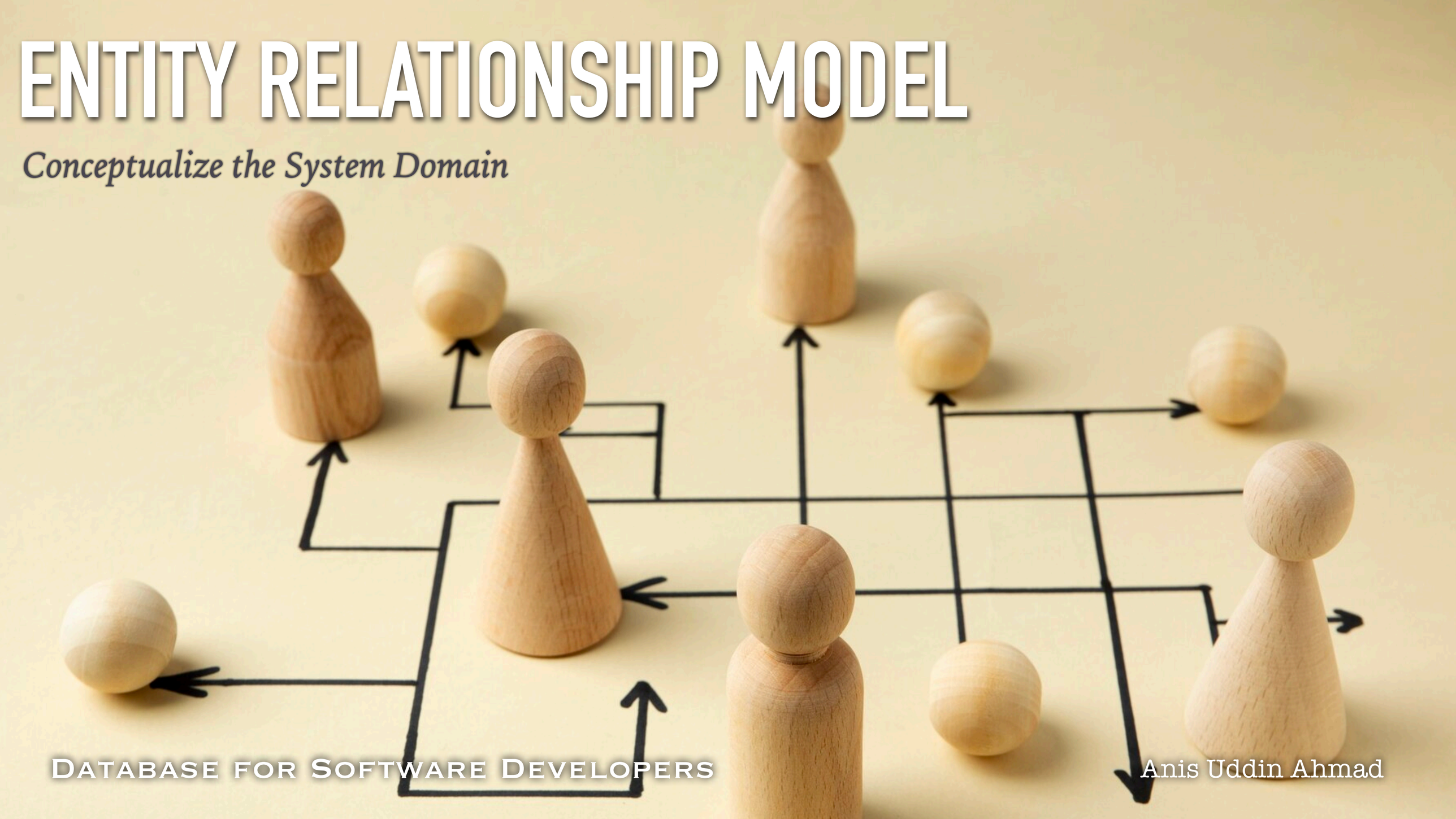
ENTITY RELATIONSHIP MODEL

Conceptualize the System Domain



Database for Software Developers

Anis Uddin Ahmad

ENTITY RELATIONSHIP MODEL

ENTITY RELATIONSHIP MODEL

Represent **things** a business **needs to remember**
in order to perform **business processes**.

ENTITY RELATIONSHIP MODEL

Represent **things** a business **needs to remember**
in order to perform **business processes**.

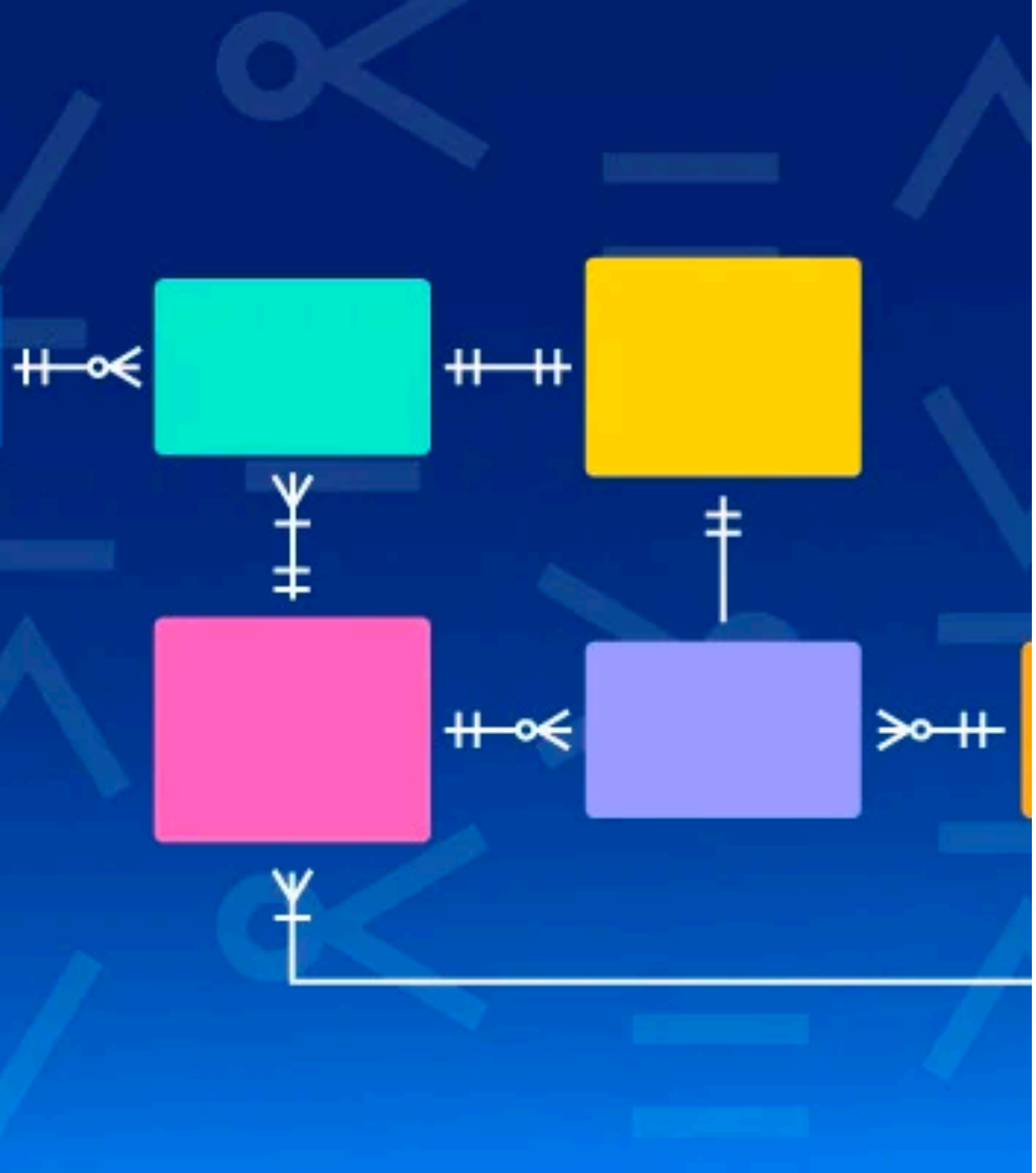
It's a conceptual view of a database.

ENTITY RELATIONSHIP MODEL

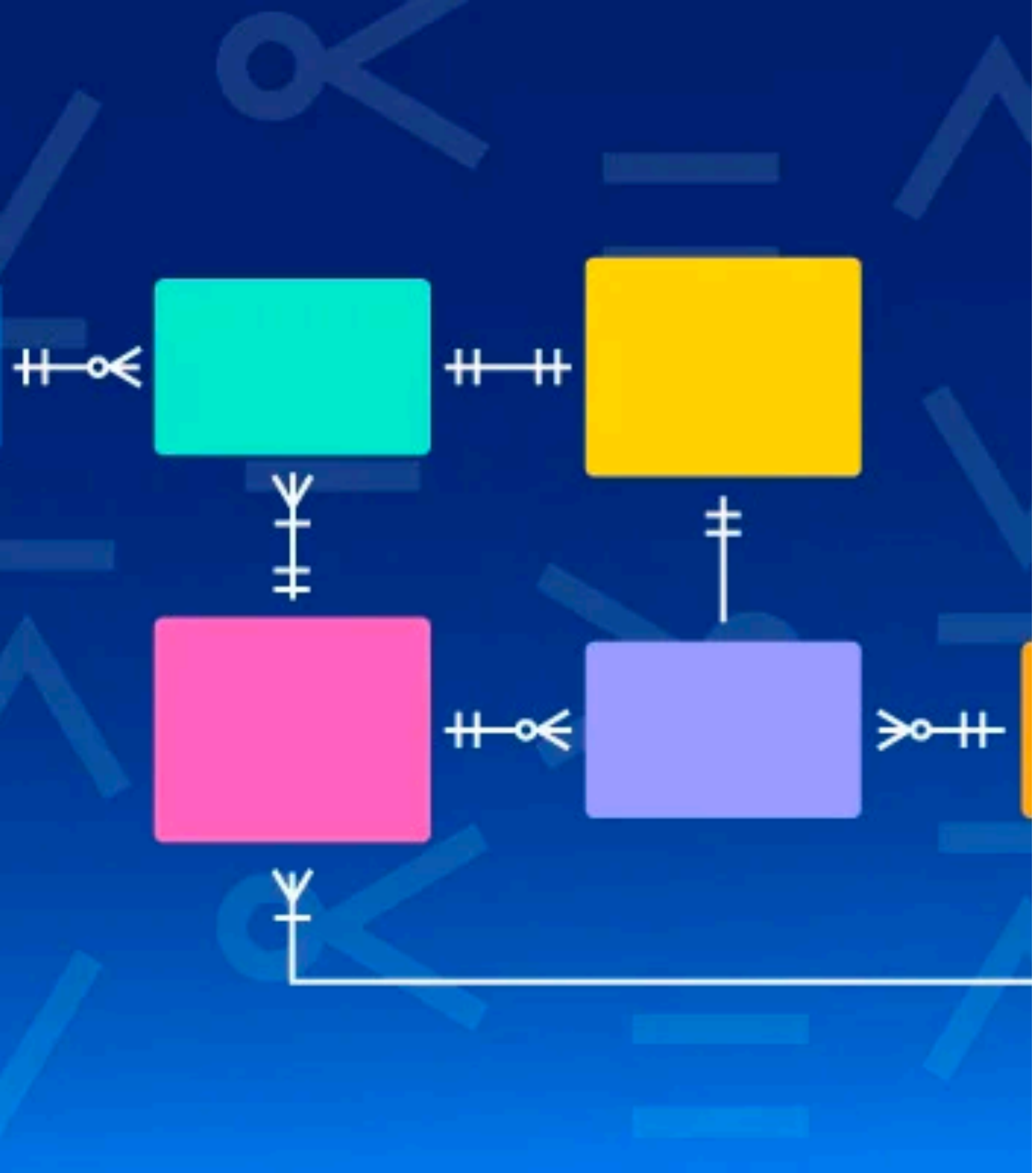
Represent **things** a business **needs to remember**
in order to perform **business processes**.

It's a conceptual view of a database.

Visualization of real-world entities and the associations among them.

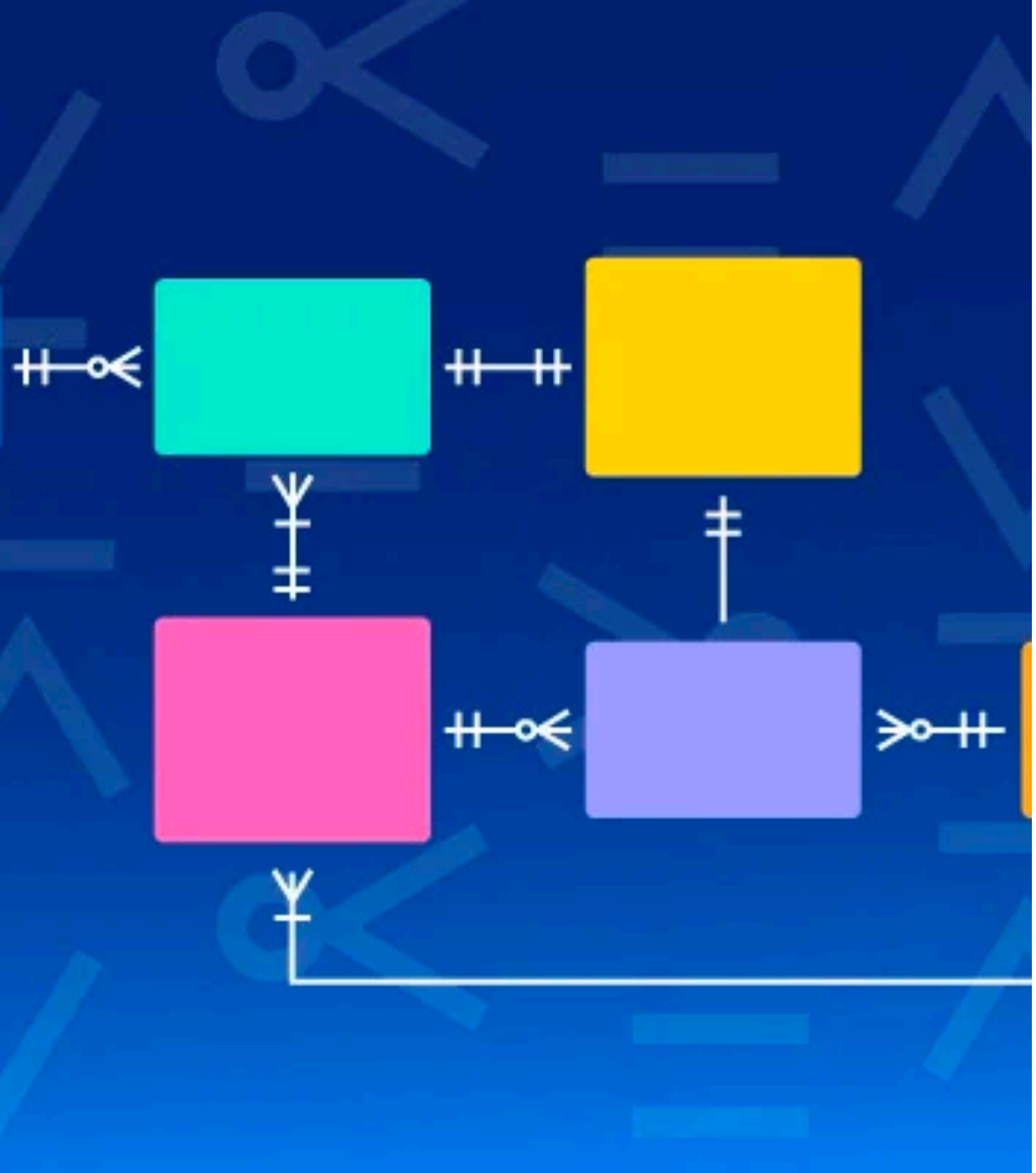


ENTITY



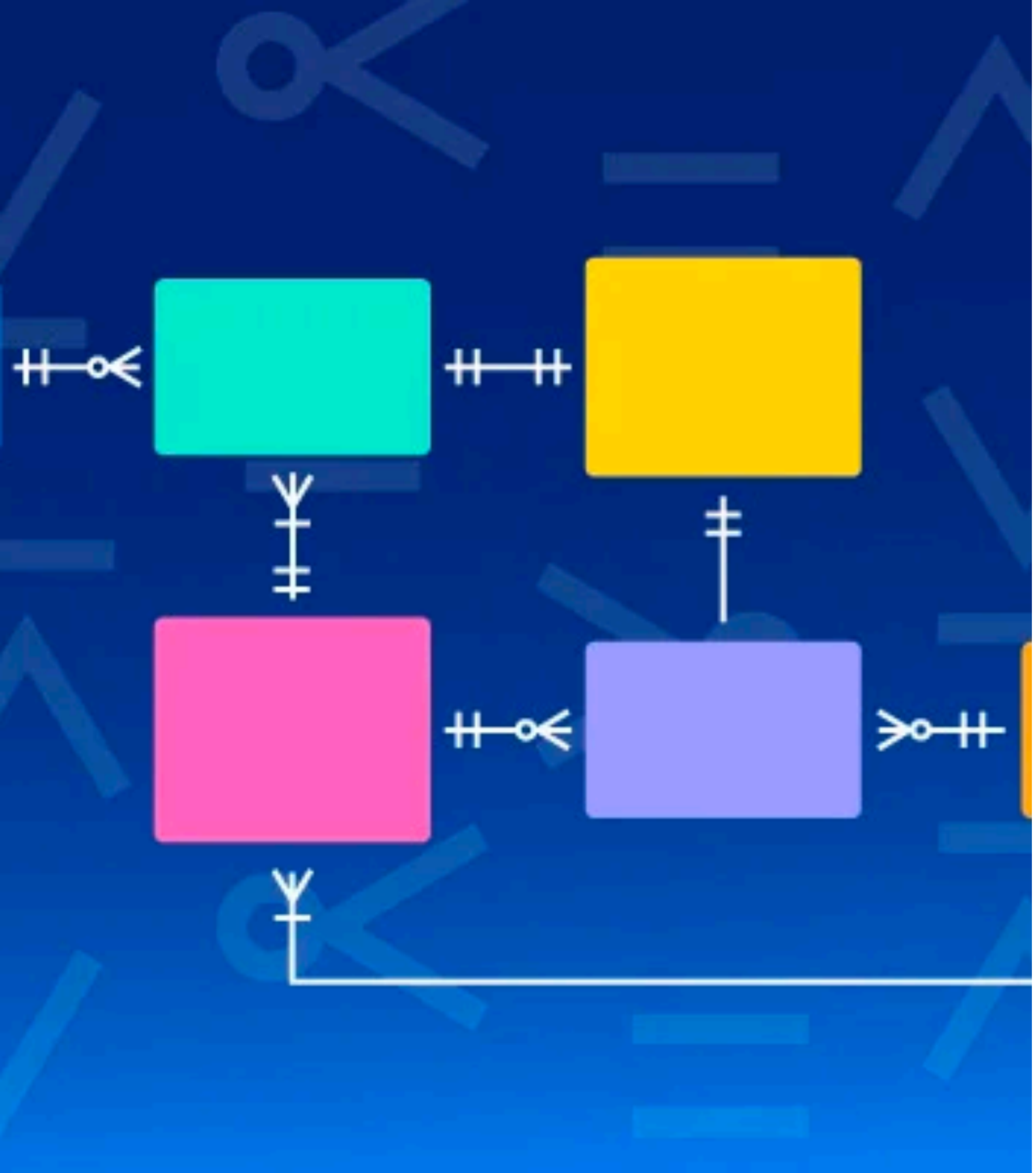
ENTITY

- A thing or object that have independent existence.



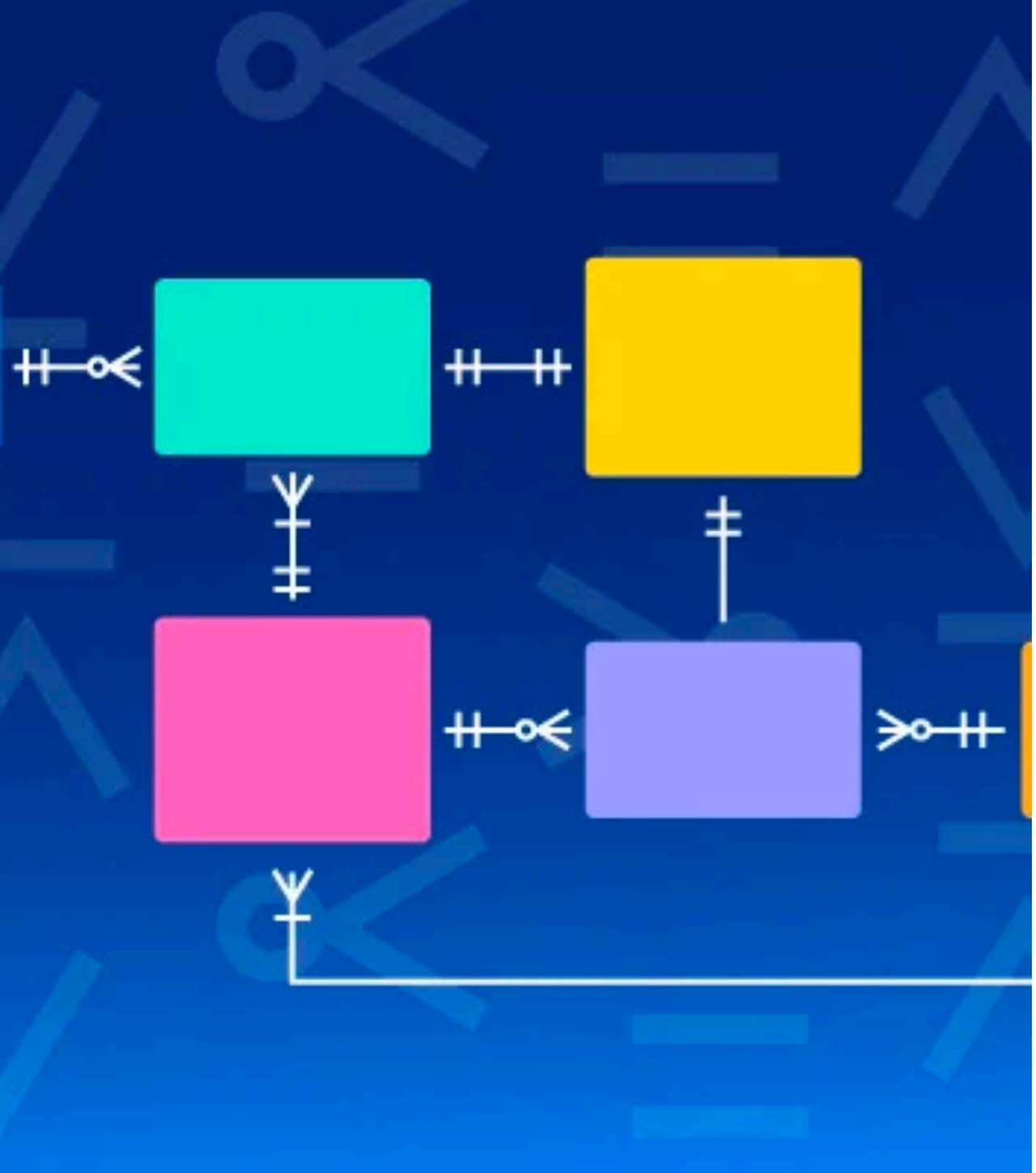
ENTITY

- A thing or object that have independent existence.
- Can have physical existence or conceptual existence only.



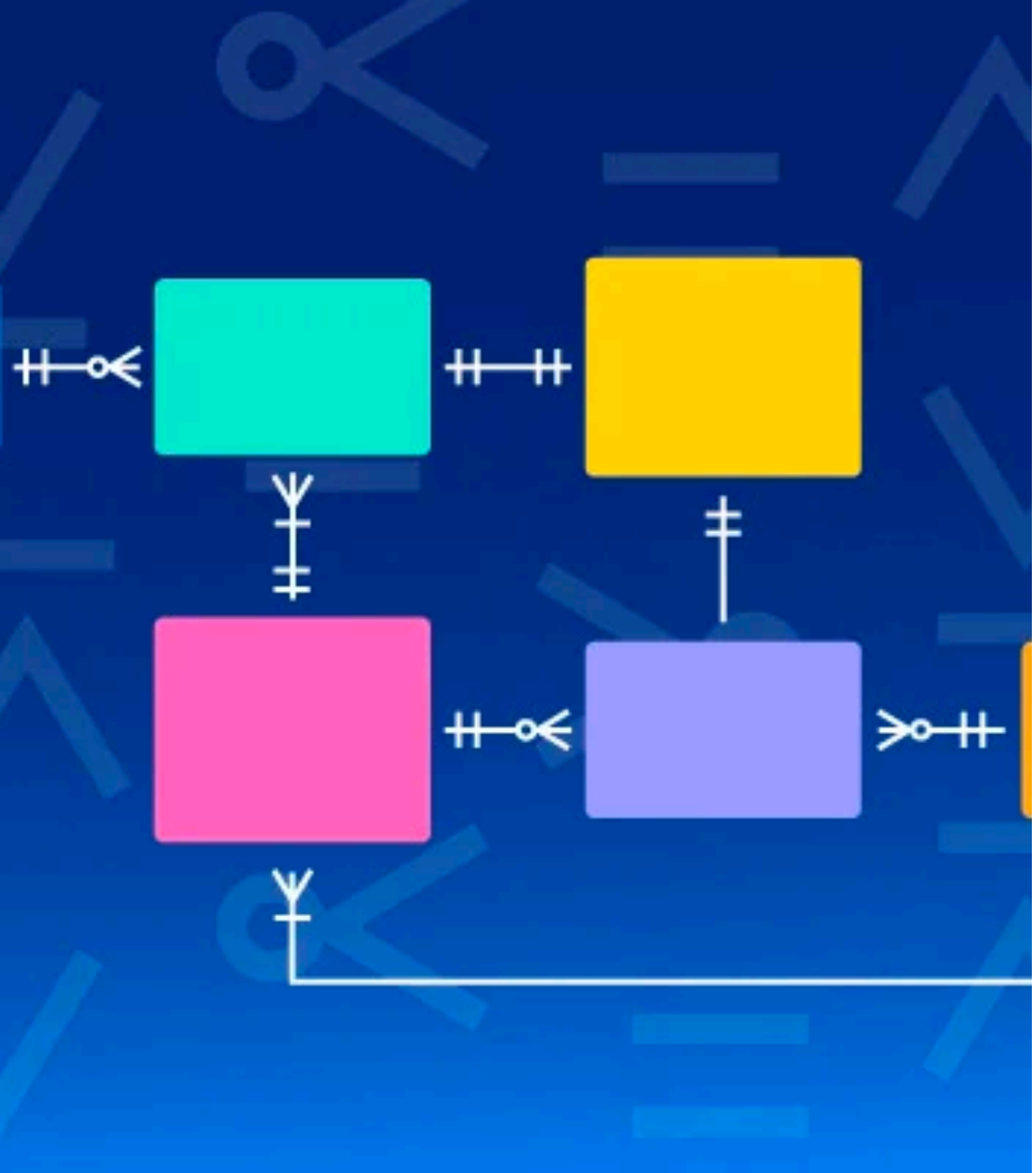
ENTITY

- A thing or object that have independent existence.
- Can have physical existence or conceptual existence only.
- Represents people, things, events, locations or concepts within the Target System.



ENTITY

- A thing or object that have independent existence.
- Can have physical existence or conceptual existence only.
- Represents people, things, events, locations or concepts within the Target System.
- In ER diagram, represented with a Rectangle.



ENTITY

- A thing or object that have independent existence.
- Can have physical existence or conceptual existence only.
- Represents people, things, events, locations or concepts within the Target System.
- In ER diagram, represented with a Rectangle.

ENTITY

ENTITY

People

- Employees
- Students
- Payroll
- Pensions
- Sick leave
- Vendors

ENTITY

People

- Employees
- Students
- Payroll
- Pensions
- Sick leave
- Vendors

Things

- Furniture
- Equipment
- Stationery
- Fire extinguishers
- Books
- Packages
- Raw Materials
- Finished Goods

ENTITY

People

- Employees
- Students
- Payroll
- Pensions
- Sick leave
- Vendors

Things

- Furniture
- Equipment
- Stationery
- Fire extinguishers
- Books
- Packages
- Raw Materials
- Finished Goods

Locations

- Offices
- Addresses
- Warehouses
- Stock rooms
- Floors
- Shelves

ENTITY

People

- Employees
- Students
- Payroll
- Pensions
- Sick leave
- Vendors

Things

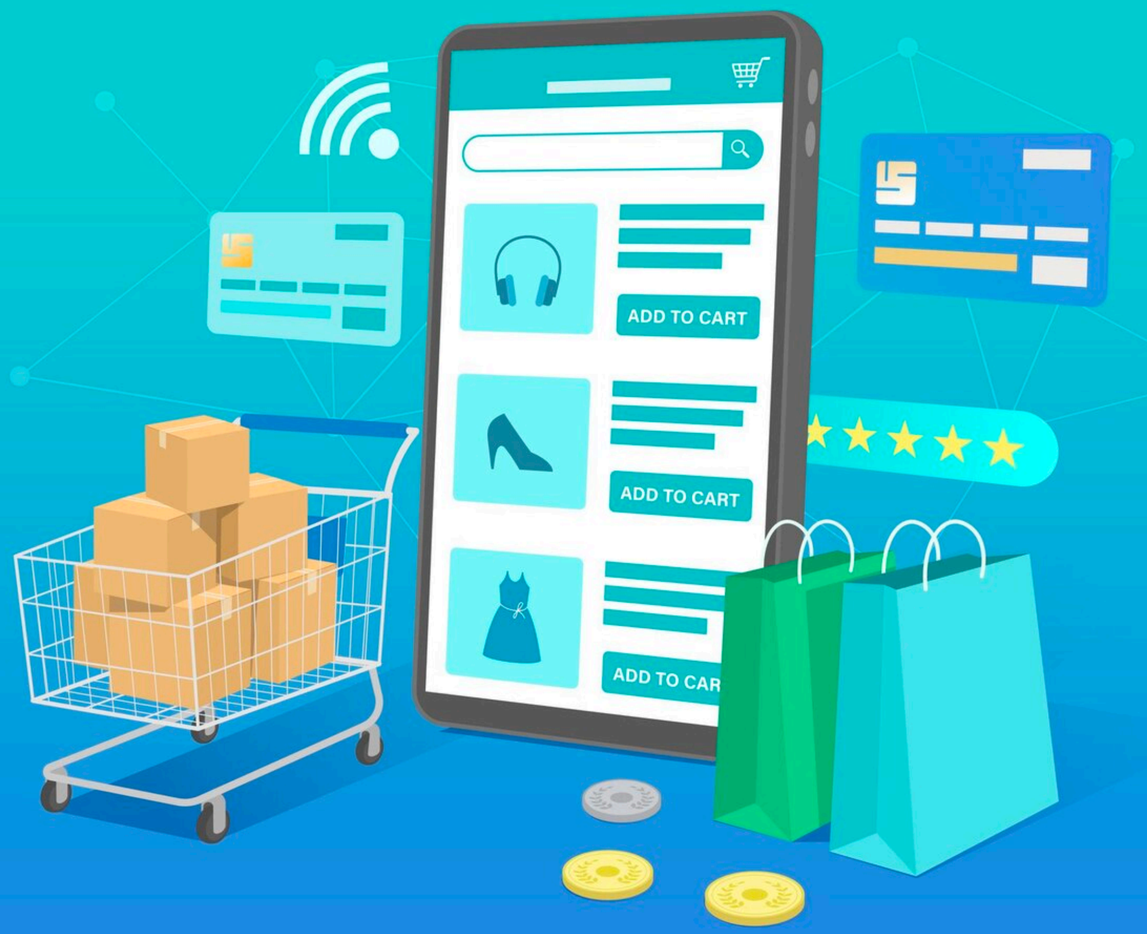
- Furniture
- Equipment
- Stationery
- Fire extinguishers
- Books
- Packages
- Raw Materials
- Finished Goods

Locations

- Offices
- Addresses
- Warehouses
- Stock rooms
- Floors
- Shelves

Events

- Sale made
- Purchase order raised
- Item hired
- Invoice issued
- Delivery received
- Check In/Out
- Meeting Happened



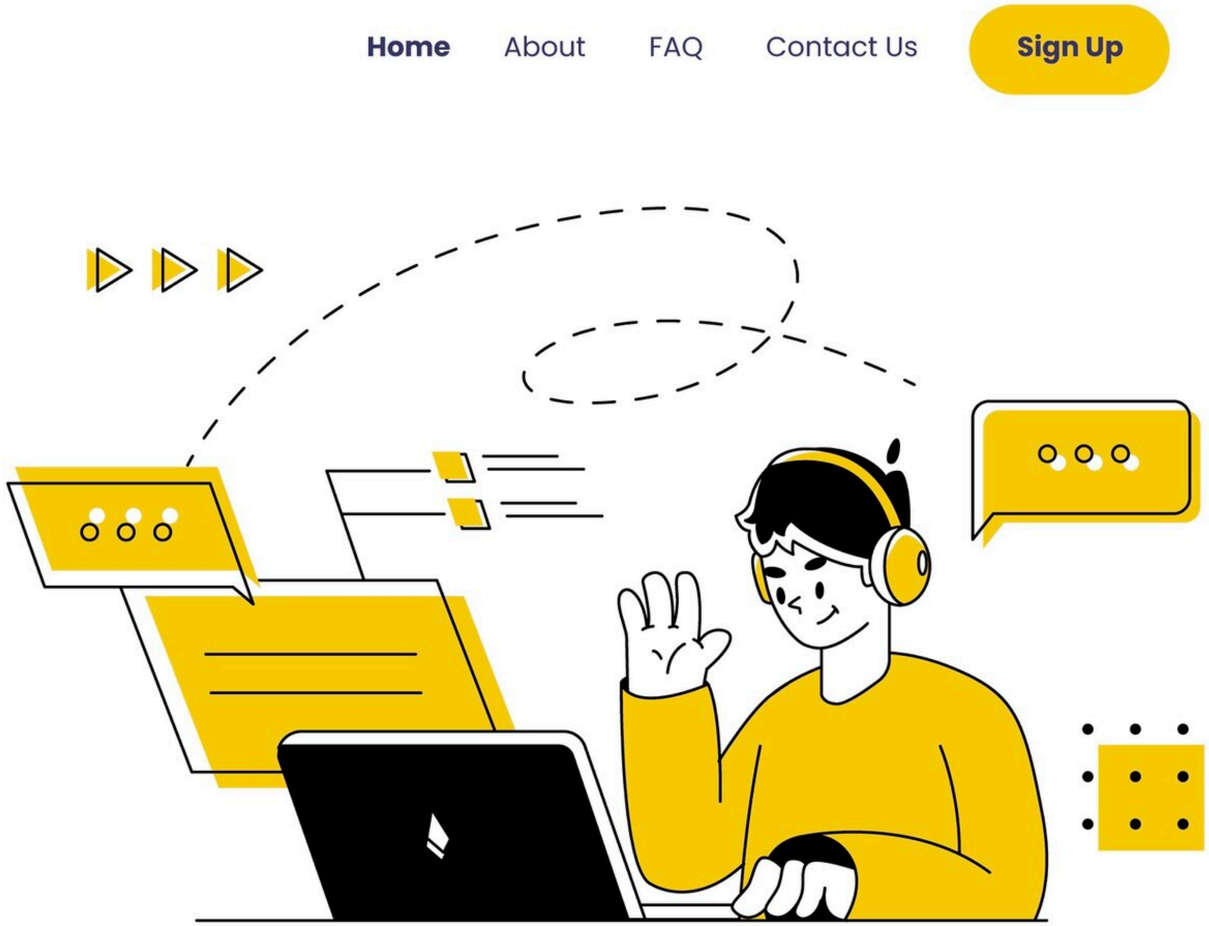
EXERCISE - FIND ENTITIES

Online Shopping Platform

Product	Cart	Delivery
Review	Order	Wishlist
User	Payment	Notification

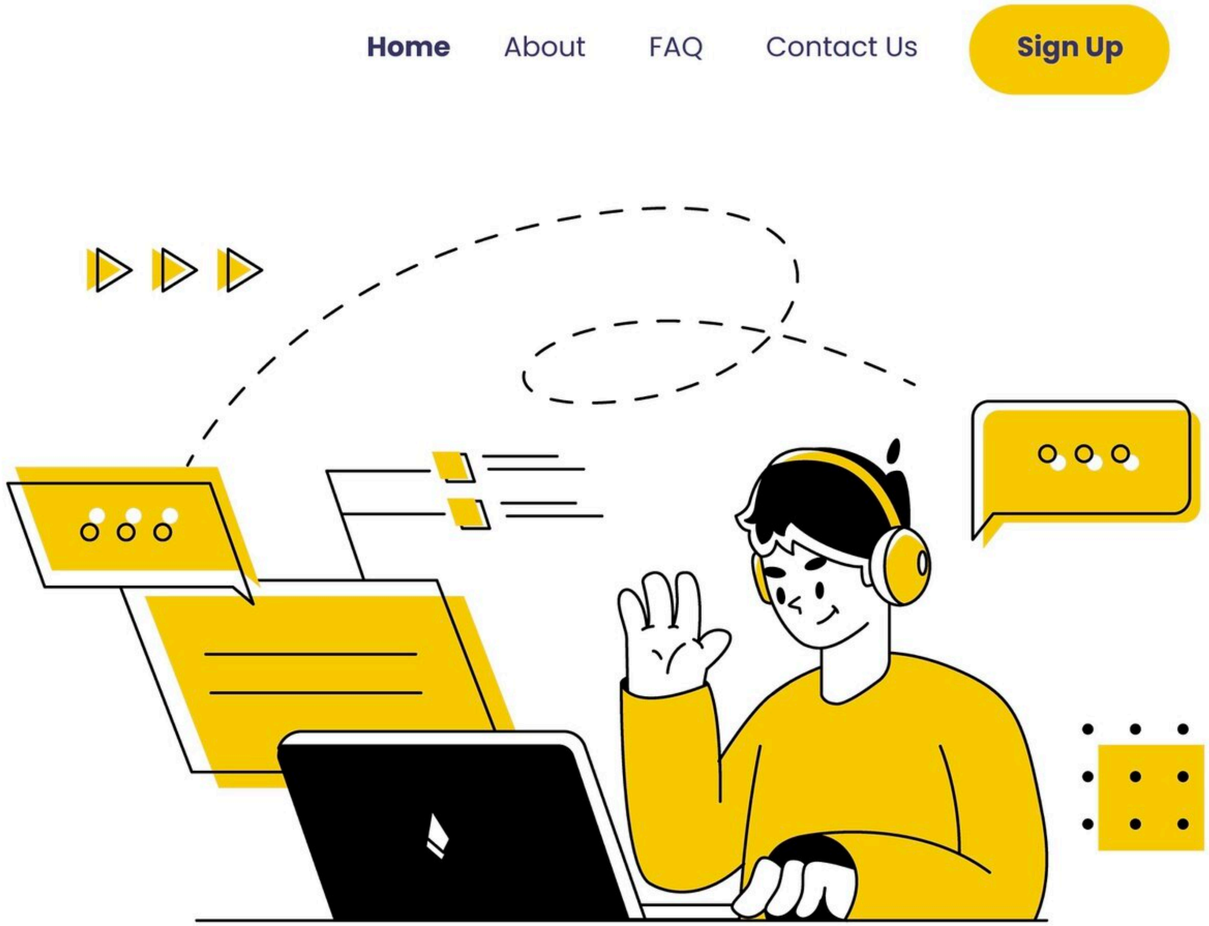
And more…

EXERCISE - FIND ENTITIES



Learning Management System (LMS)

EXERCISE - FIND ENTITIES



Learning Management System (LMS)

Student	Batch	Class
Instructor	Enrollment	Exam
Course	Payment	Certificate

And more…

WEAK ENTITY

WEAK ENTITY

- Do not have independent existence

WEAK ENTITY

- Do not have independent existence
- Always connected and dependent on other (Strong) Entity

WEAK ENTITY

- Do not have independent existence
- Always connected and dependent on other (Strong) Entity
- May not have a key attribute

WEAK ENTITY

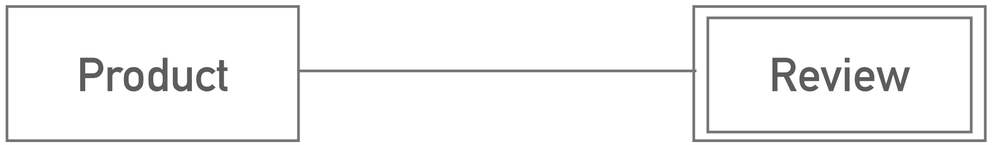
- Do not have independent existence
- Always connected and dependent on other (Strong) Entity
- May not have a key attribute
- Represented using a double rectangle

WEAK ENTITY

- Do not have independent existence
- Always connected and dependent on other (Strong) Entity
- May not have a key attribute
- Represented using a double rectangle

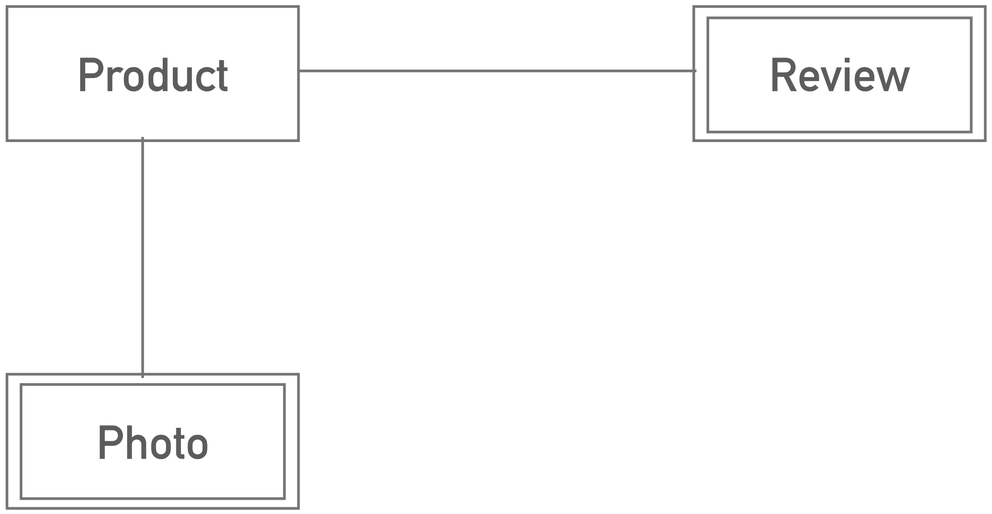
WEAK ENTITY

- Do not have independent existence
- Always connected and dependent on other (Strong) Entity
- May not have a key attribute
- Represented using a double rectangle



WEAK ENTITY

- Do not have independent existence
- Always connected and dependent on other (Strong) Entity
- May not have a key attribute
- Represented using a double rectangle



WEAK ENTITY

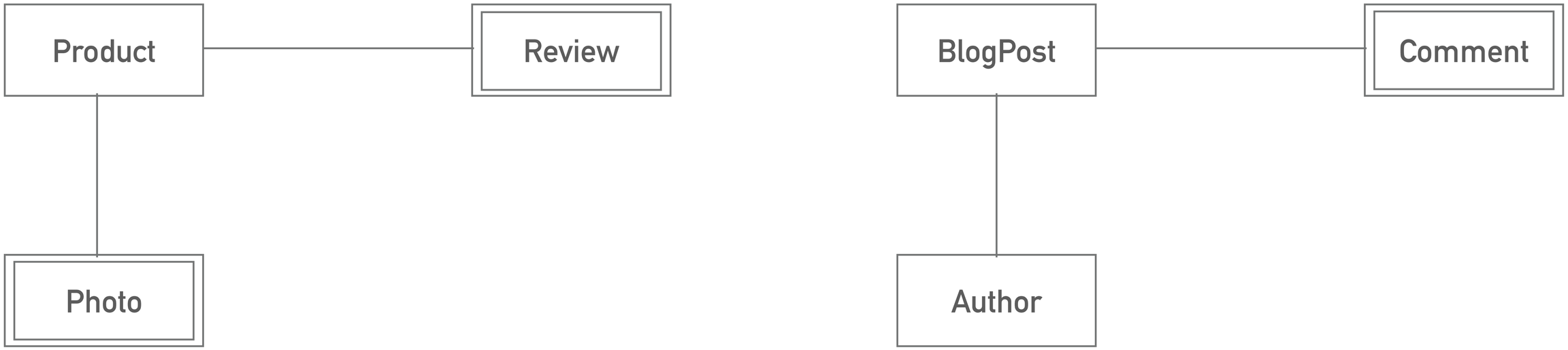
- Do not have independent existence
- Always connected and dependent on other (Strong) Entity
- May not have a key attribute
- Represented using a double rectangle

WEAK ENTITY

- Do not have independent existence
- Always connected and dependent on other (Strong) Entity
- May not have a key attribute
- Represented using a double rectangle

WEAK ENTITY

- Do not have independent existence
- Always connected and dependent on other (Strong) Entity
- May not have a key attribute
- Represented using a double rectangle



ENTITIES – TYPE AND SET

Products

Marker Pen

A4 Paper

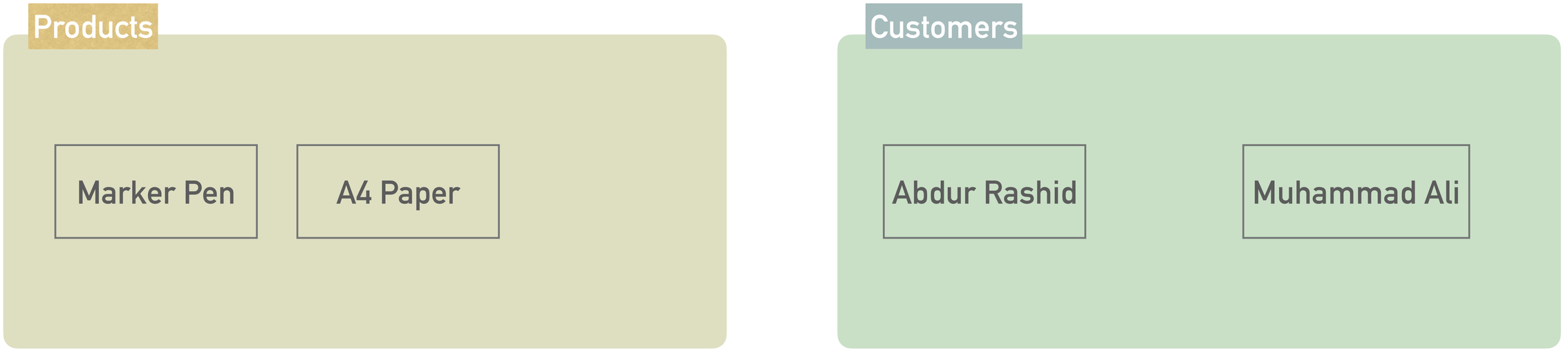
Customers

Abdur Rashid

Muhammad Ali

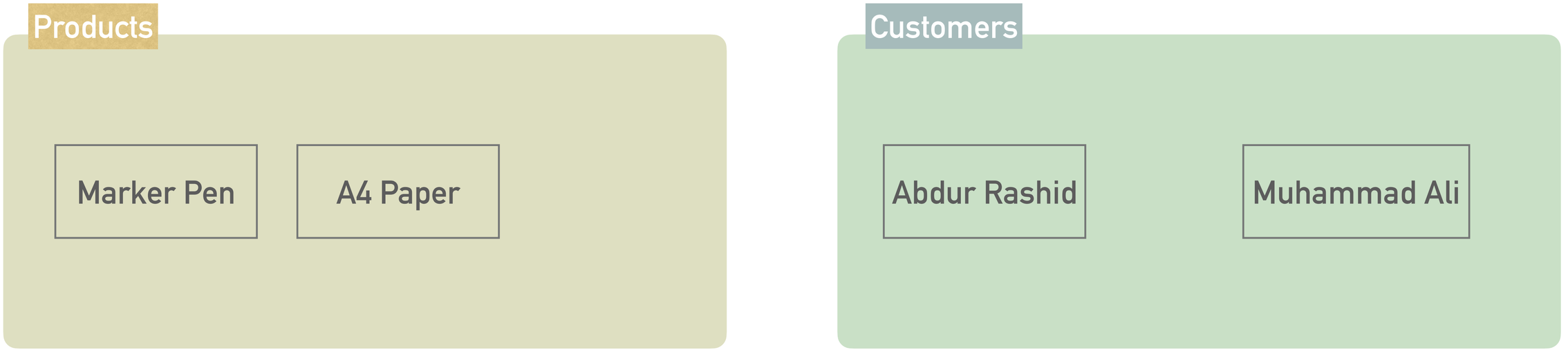
ENTITIES – TYPE AND SET

- **Entity Type:** Collection of Entities with same attributes. (e.g. Products, Customers)



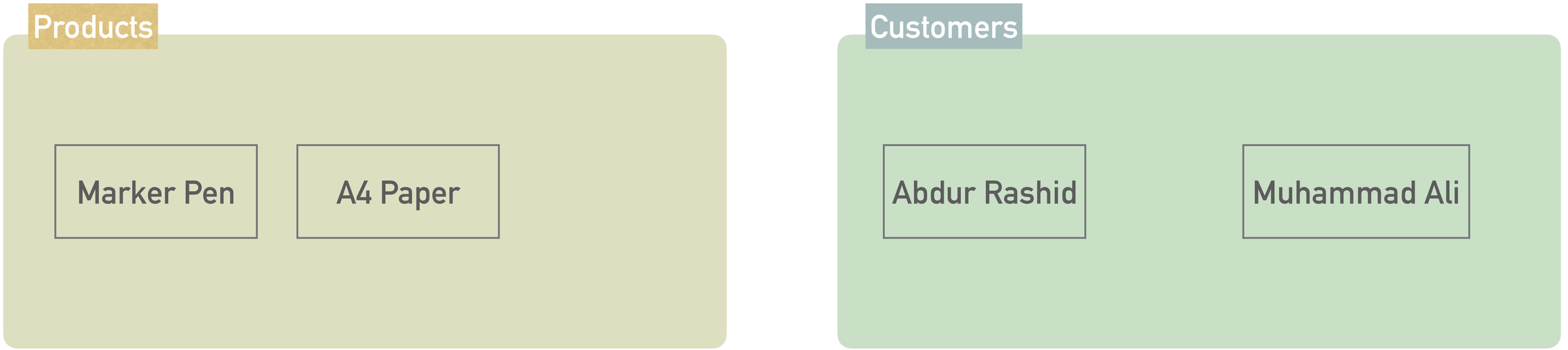
ENTITIES – TYPE AND SET

- **Entity Type:** Collection of Entities with same attributes. (e.g. Products, Customers)
- **Entity Set:** Collection of Entities (of same Entity Type) at a point of time and optionally with some conditions.

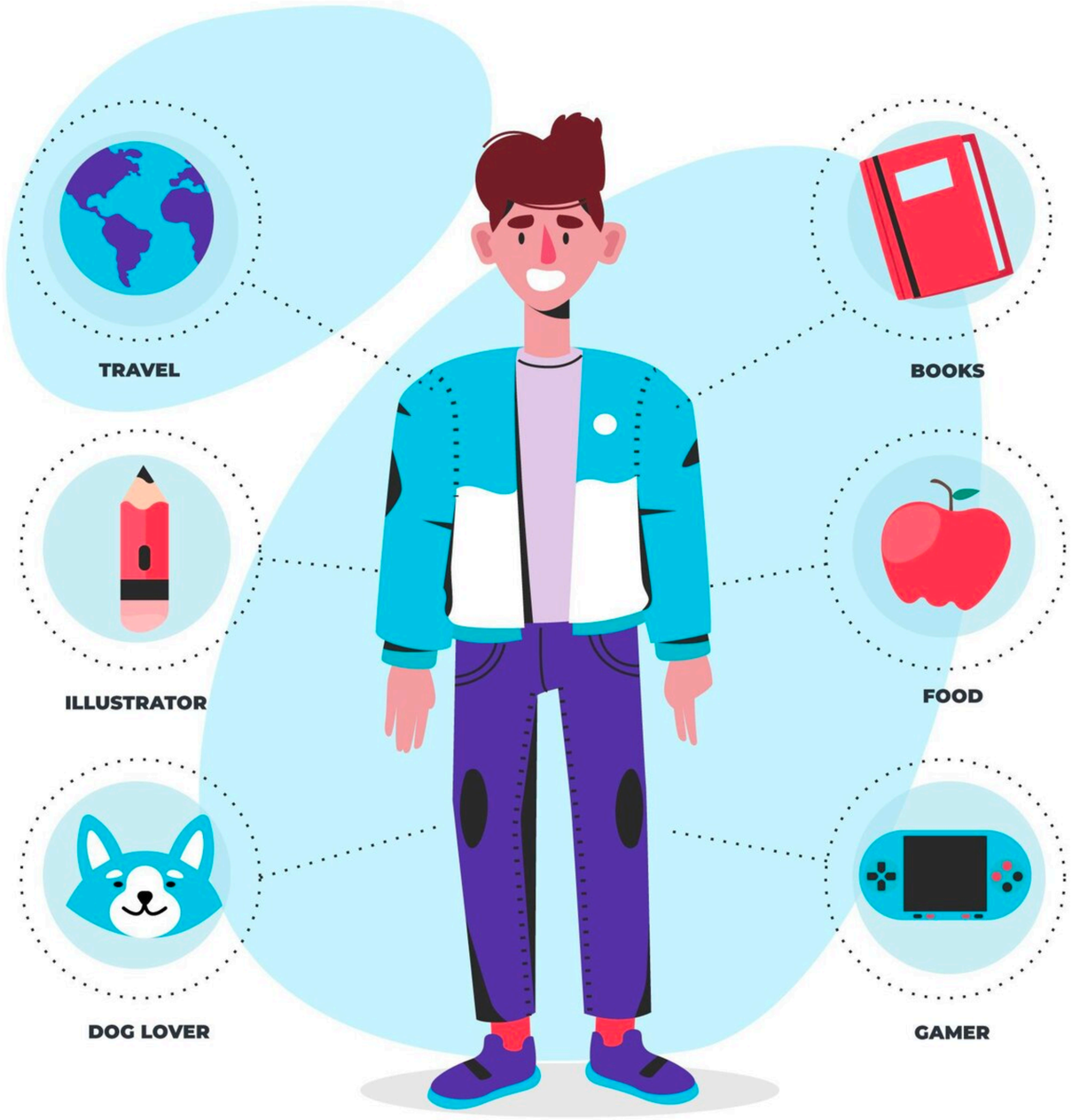


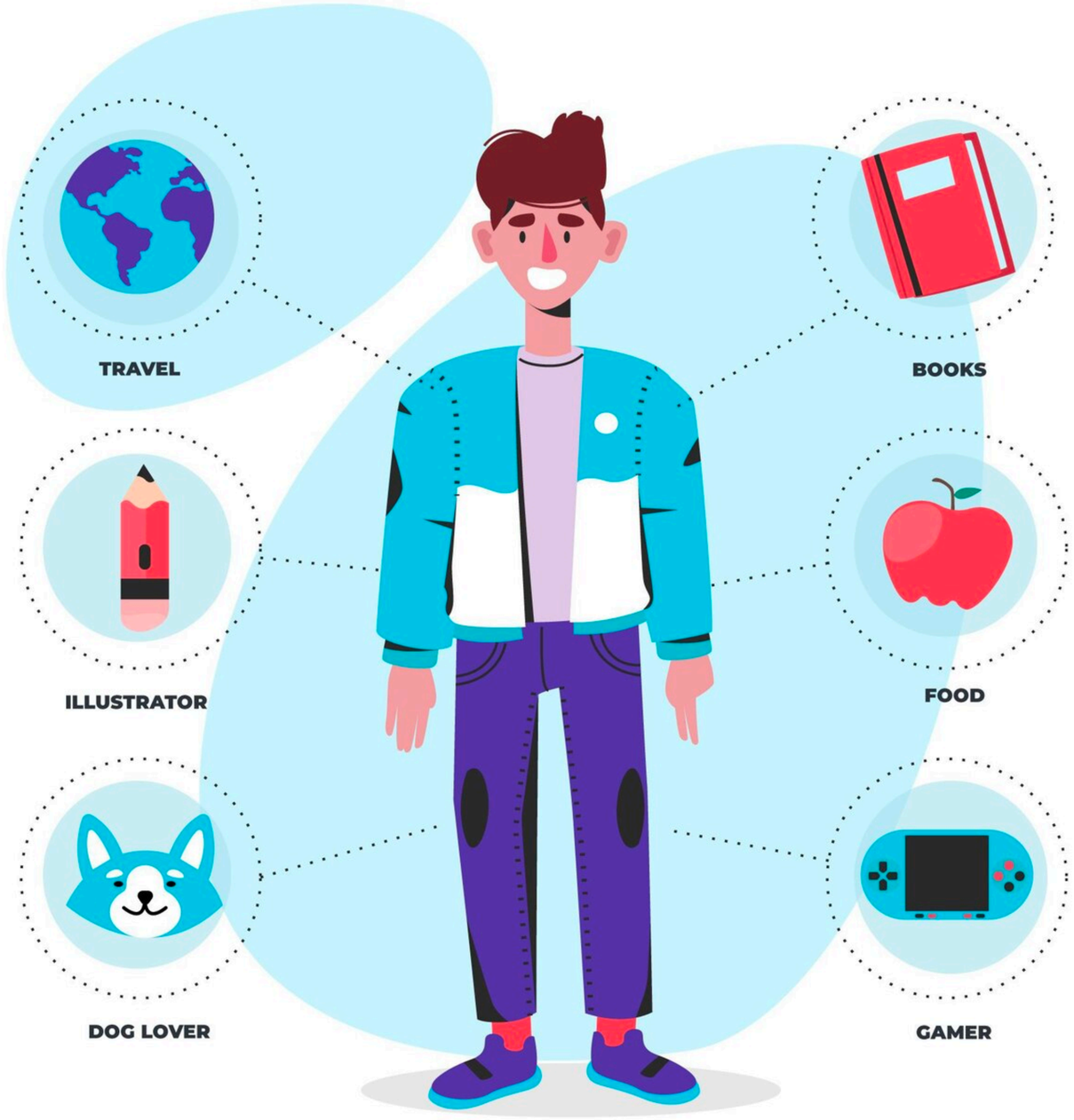
ENTITIES – TYPE AND SET

- **Entity Type:** Collection of Entities with same attributes. (e.g. Products, Customers)
- **Entity Set:** Collection of Entities (of same Entity Type) at a point of time and optionally with some conditions.
- **Weak Entity Set:** Entity Set that do not have sufficient attributes to form a *primary key*.



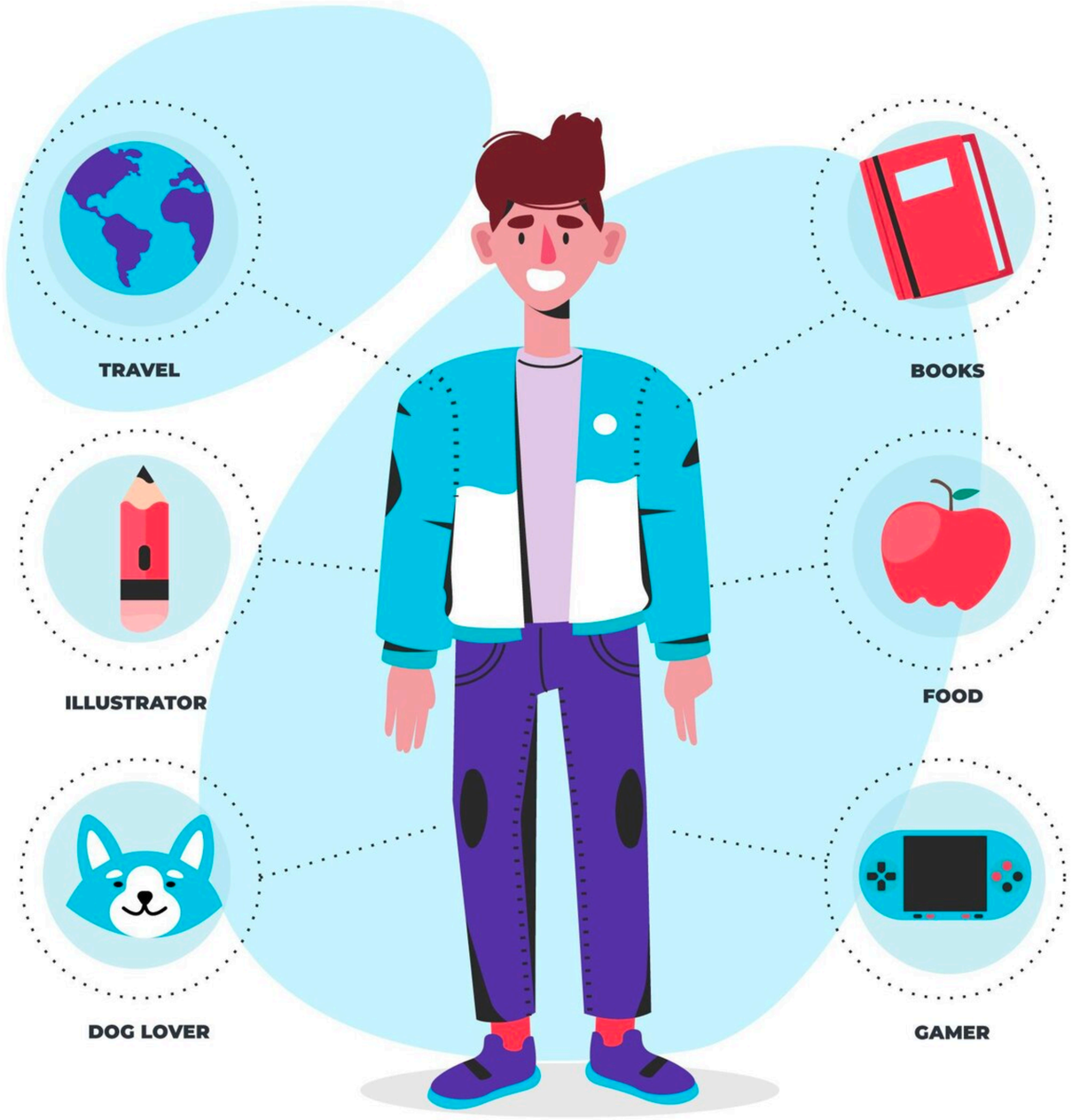
ATTRIBUTES





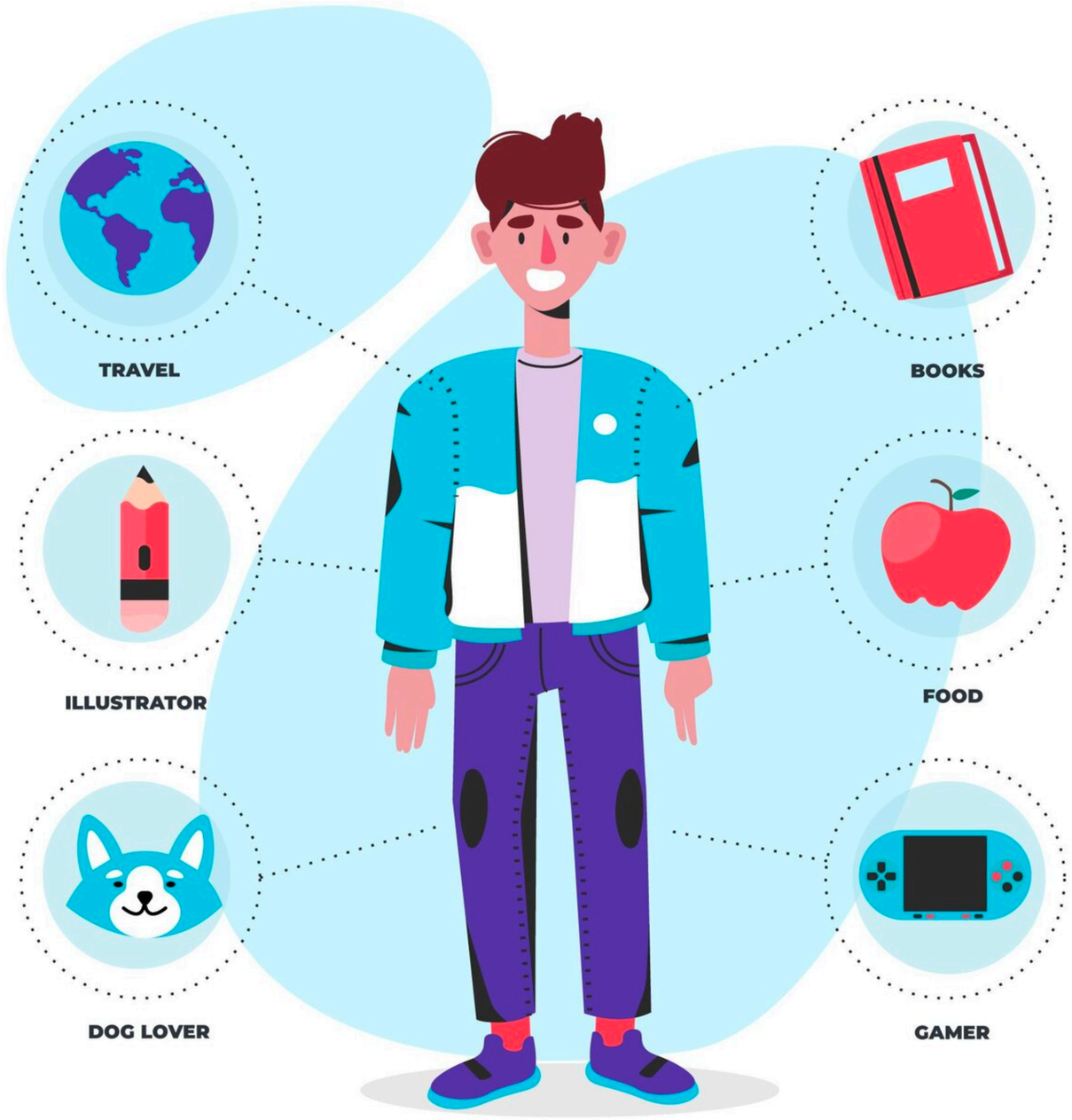
ATTRIBUTES

- Properties that describes an Entity.



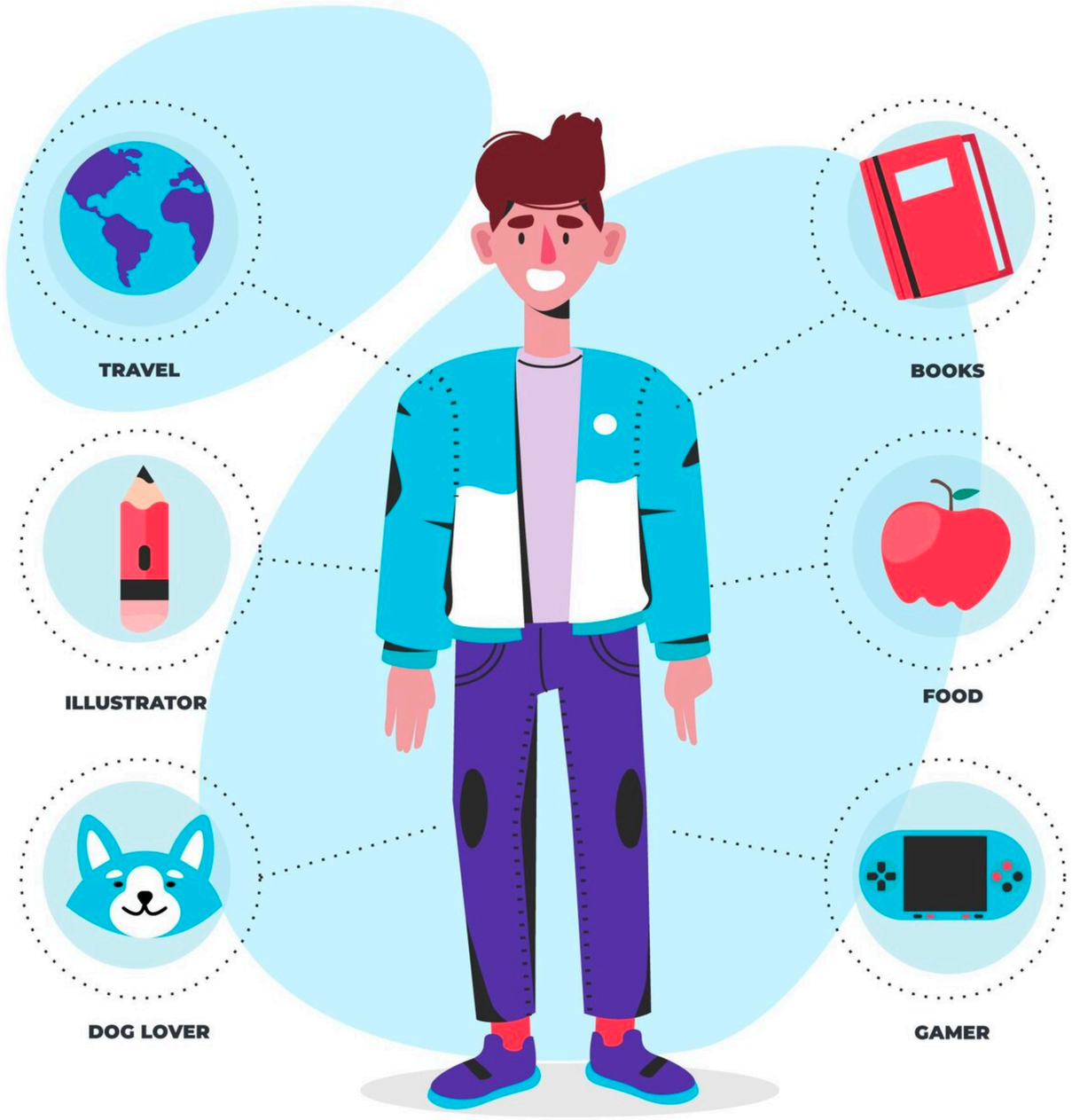
ATTRIBUTES

- Properties that describes an Entity.
- These information or properties are required to operate the **Target System**.



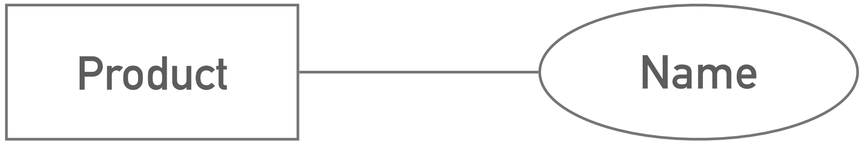
ATTRIBUTES

- Properties that describes an Entity.
- These information or properties are required to operate the **Target System**.
- Represented with an Oval in ER Diagram.



ATTRIBUTES

- Properties that describes an Entity.
- These information or properties are required to operate the **Target System**.
- Represented with an Oval in ER Diagram.

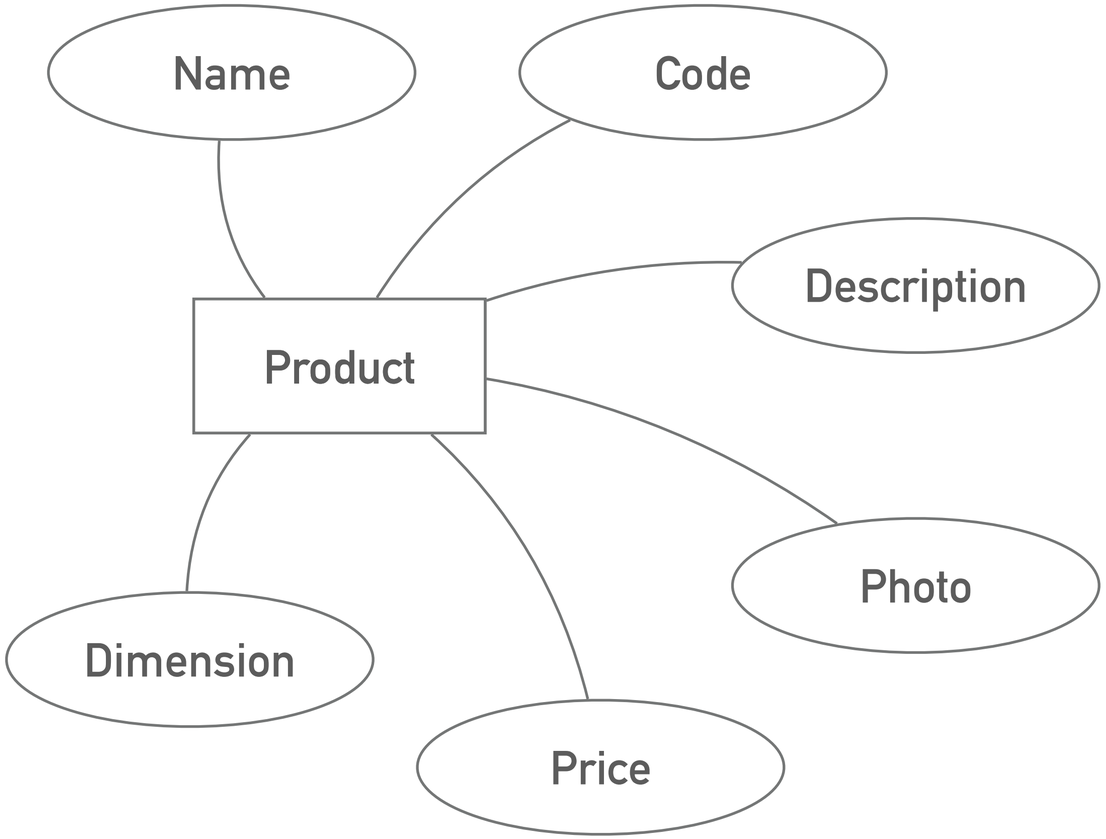


EXERCISE – FINDING ATTRIBUTES

Product

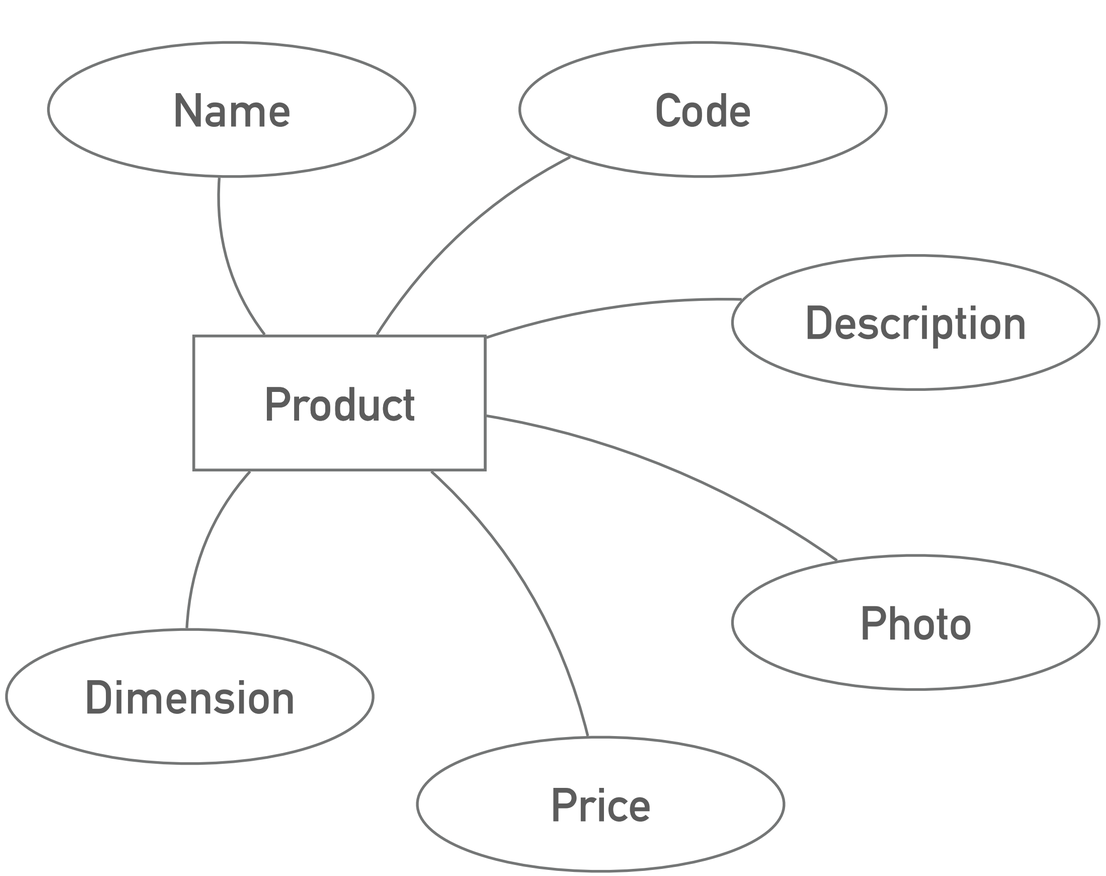
Customer

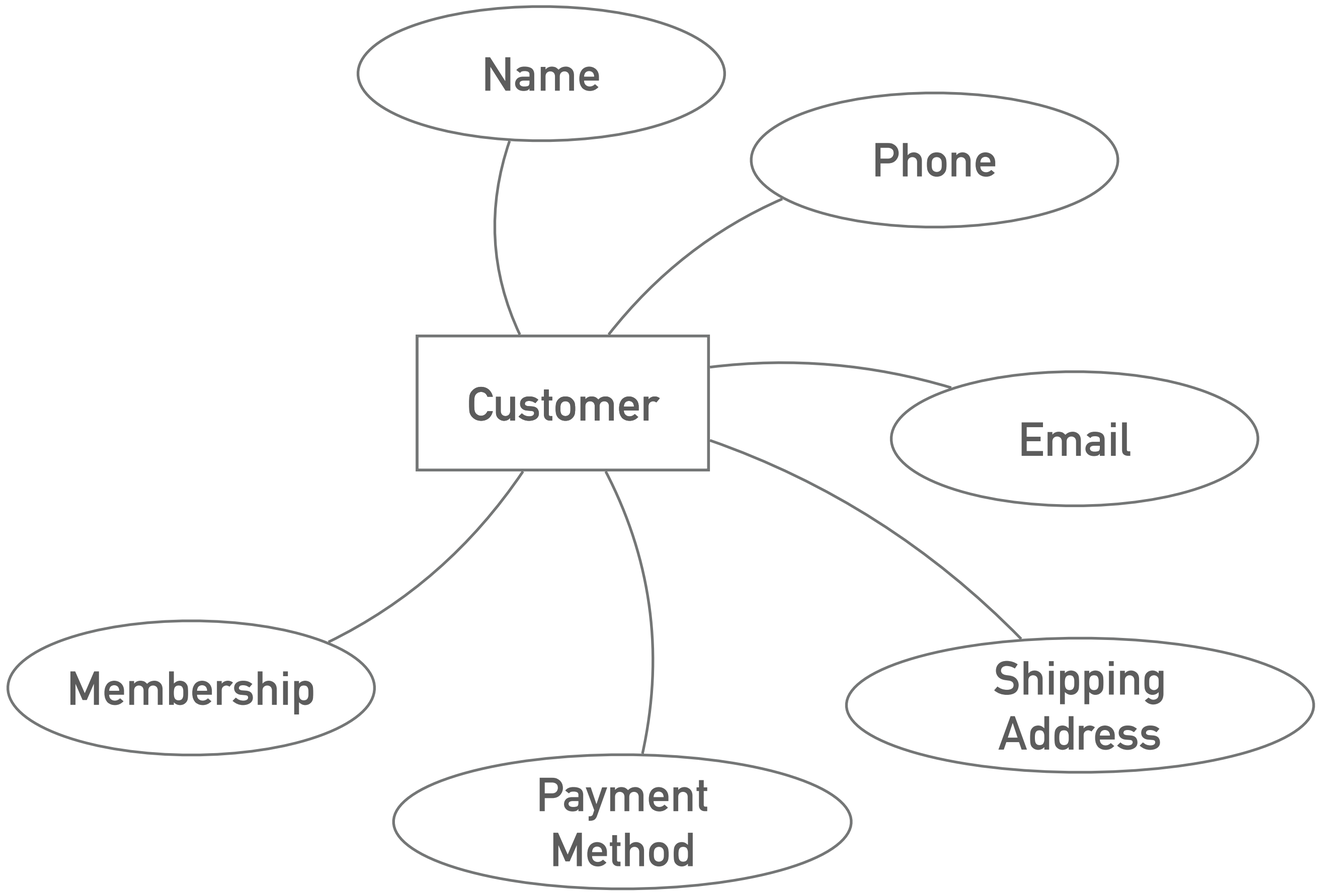
EXERCISE – FINDING ATTRIBUTES



Customer

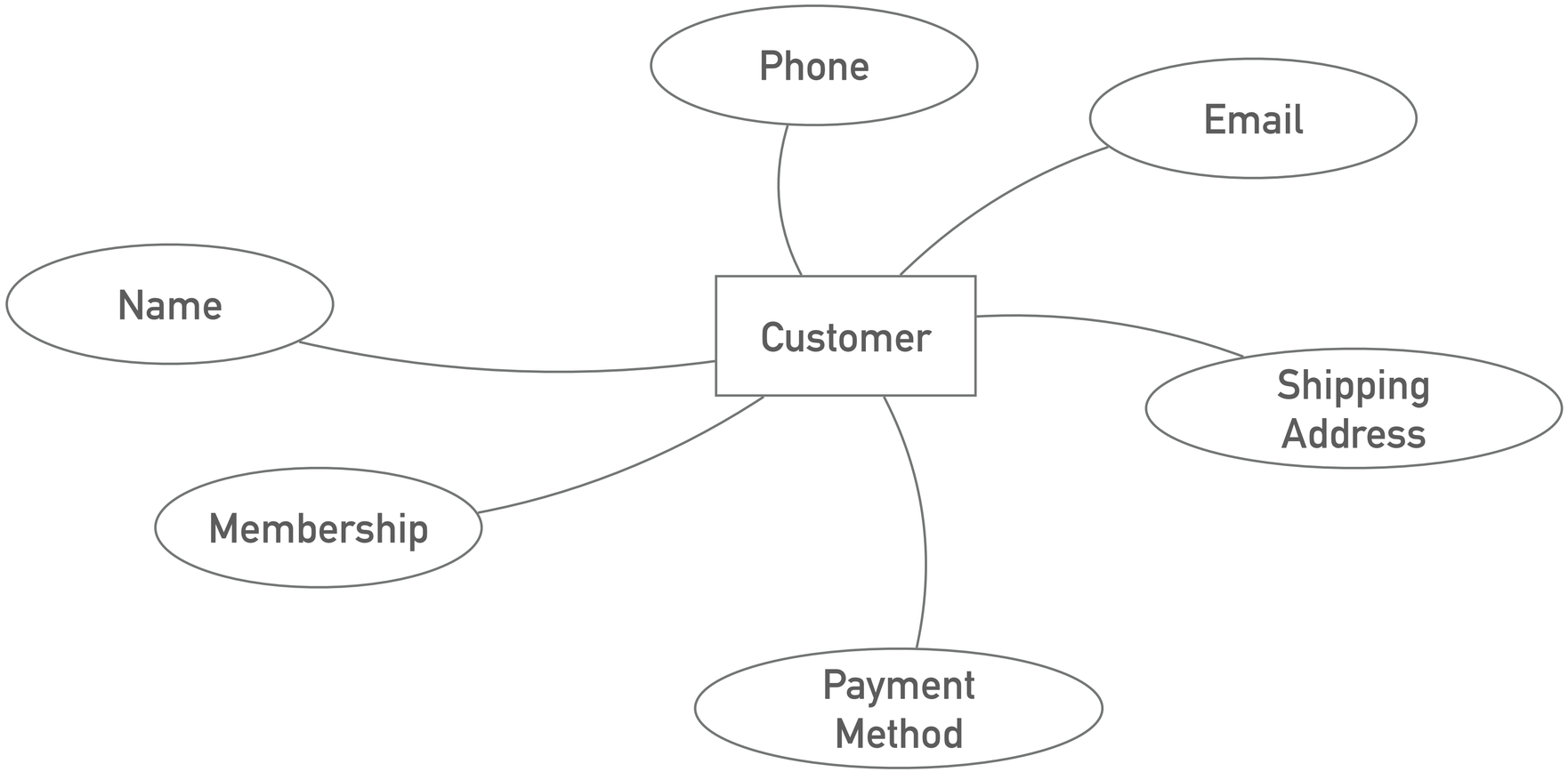
EXERCISE – FINDING ATTRIBUTES





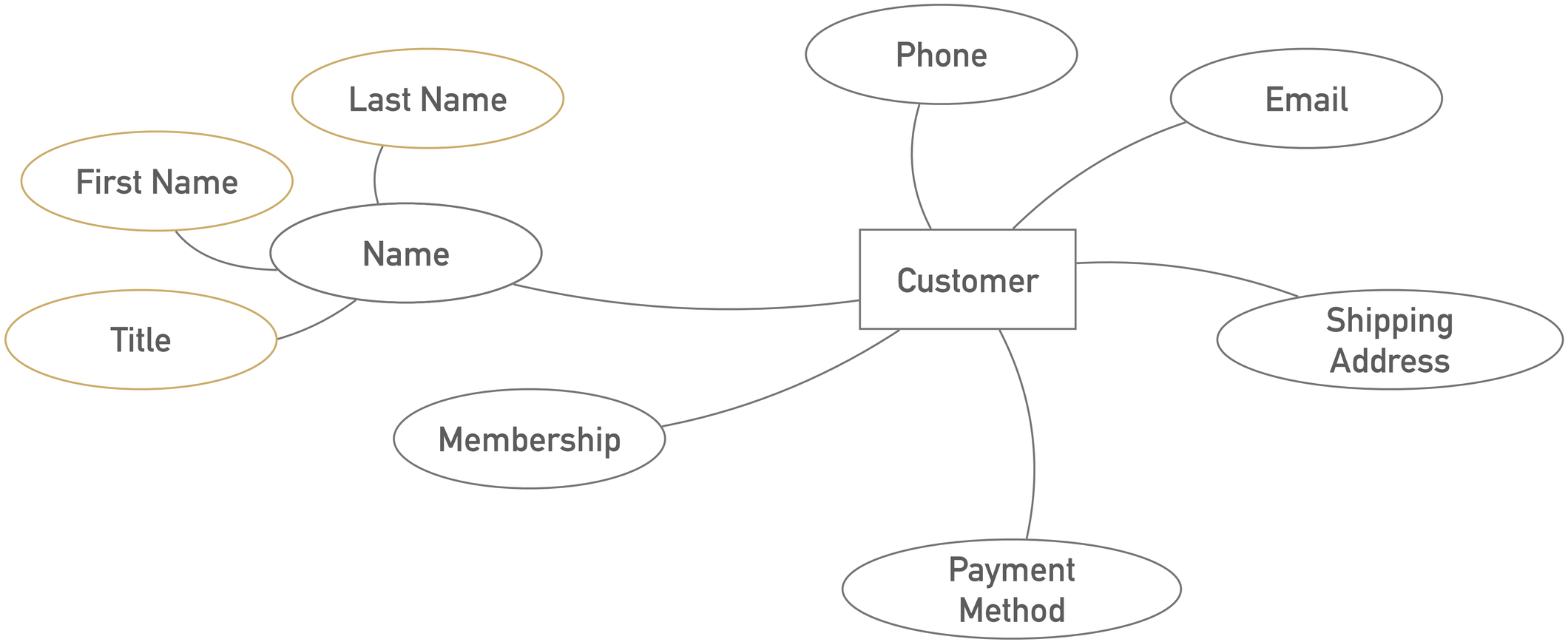
ATTRIBUTES – SIMPLE VS COMPOSITE

- Composite: Can be divided into some parts. e.g. Name, Address
- Simple: cannot be divided: Age, Height



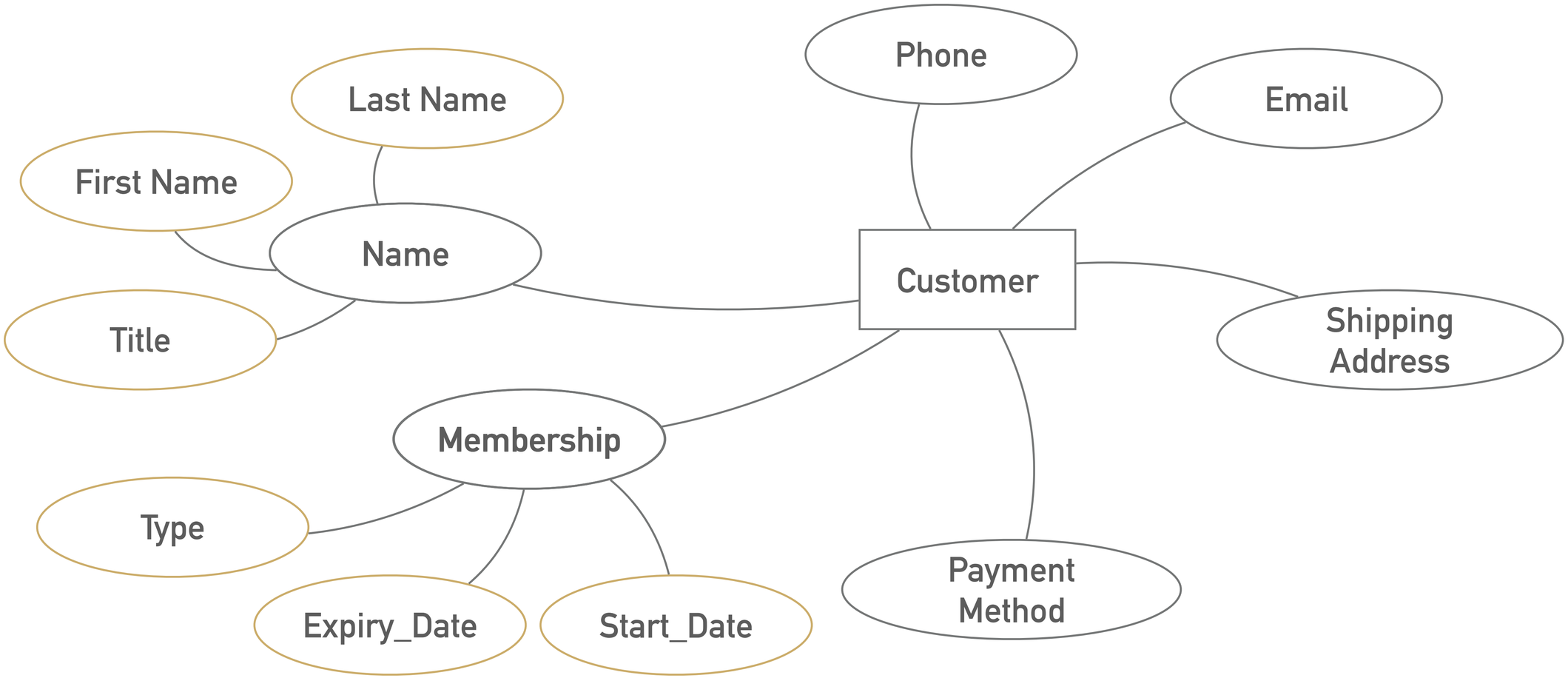
ATTRIBUTES – SIMPLE VS COMPOSITE

- Composite: Can be divided into some parts. e.g. Name, Address
- Simple: cannot be divided: Age, Height



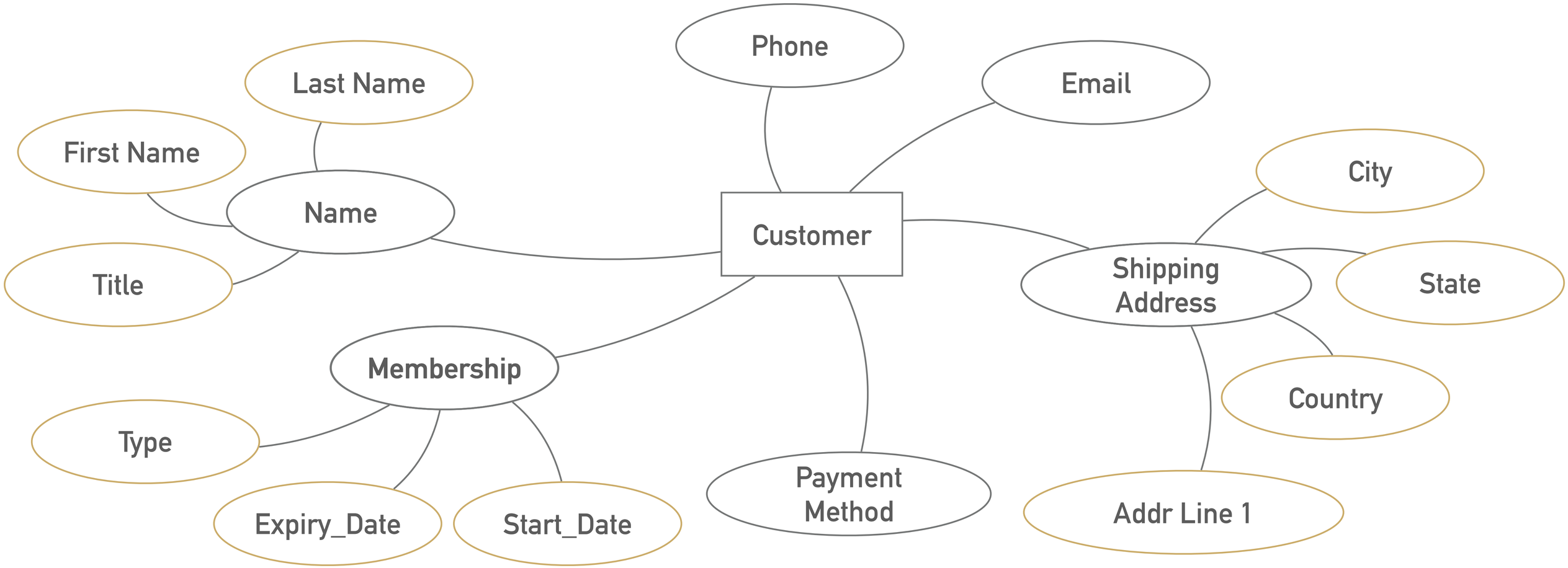
ATTRIBUTES – SIMPLE VS COMPOSITE

- Composite: Can be divided into some parts. e.g. Name, Address
- Simple: cannot be divided: Age, Height



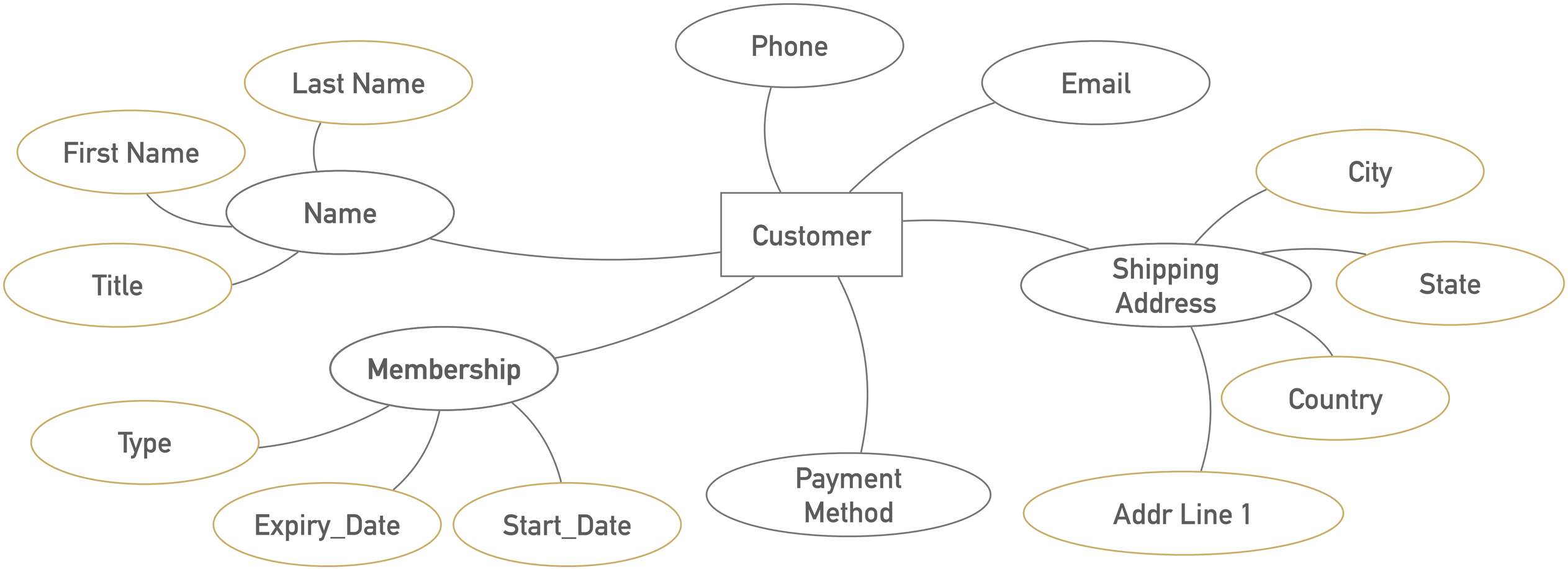
ATTRIBUTES – SIMPLE VS COMPOSITE

- Composite: Can be divided into some parts. e.g. Name, Address
- Simple: cannot be divided: Age, Height



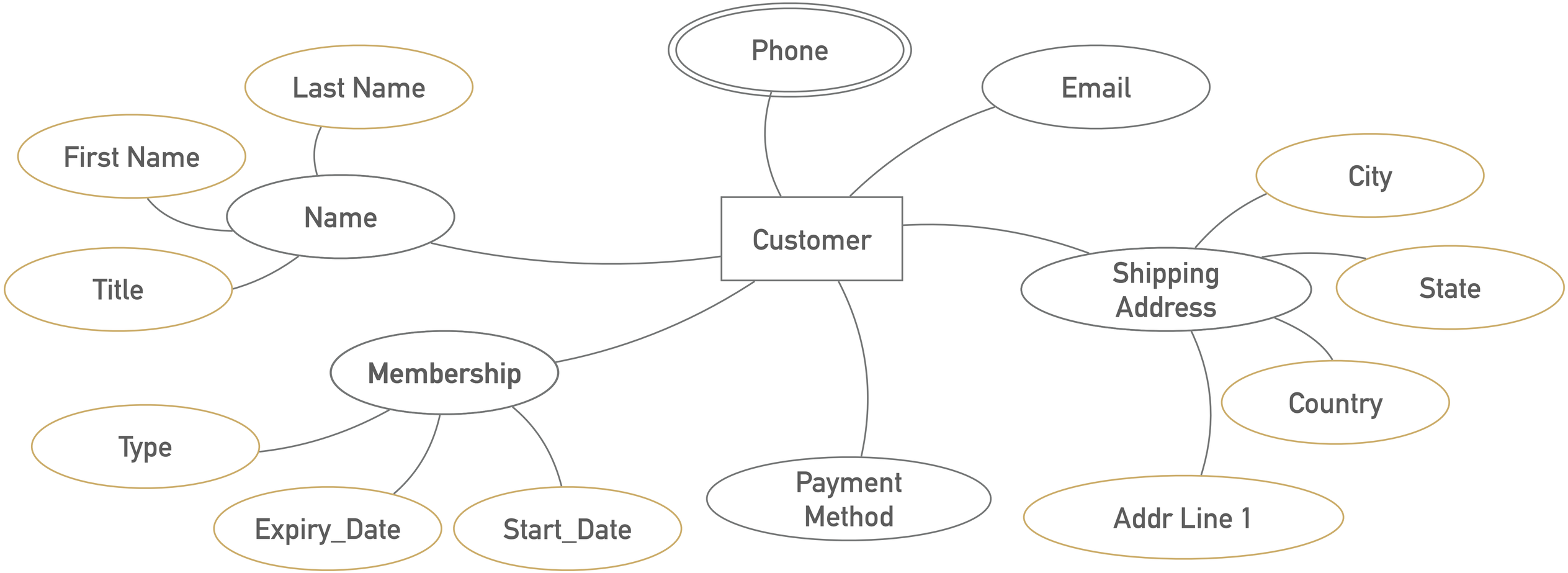
ATTRIBUTES – SINGLE-VALUED VS MILTI-VALUED

- **Single-Valued:** Can have only one value. e.g. Age
- **Multi-Valued:** Can have a set of values. e.g. Educational Certificate



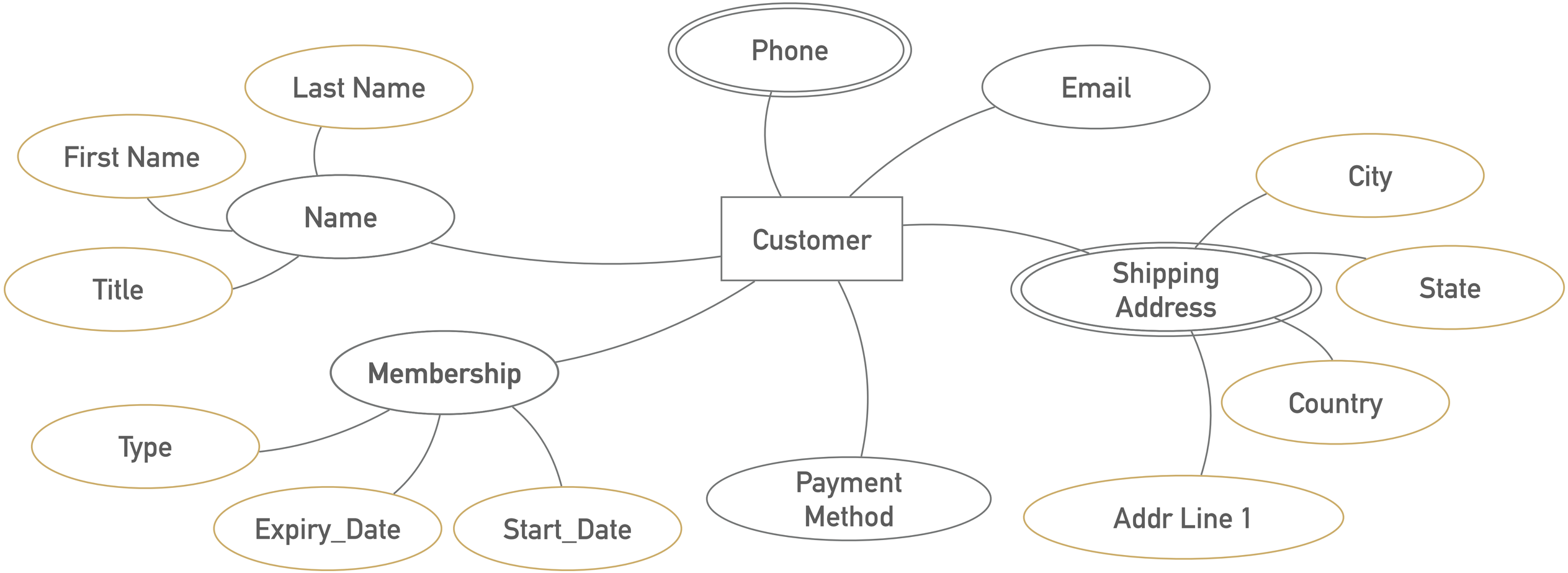
ATTRIBUTES – SINGLE-VALUED VS MILTI-VALUED

- **Single-Valued:** Can have only one value. e.g. Age
- **Multi-Valued:** Can have a set of values. e.g. Educational Certificate



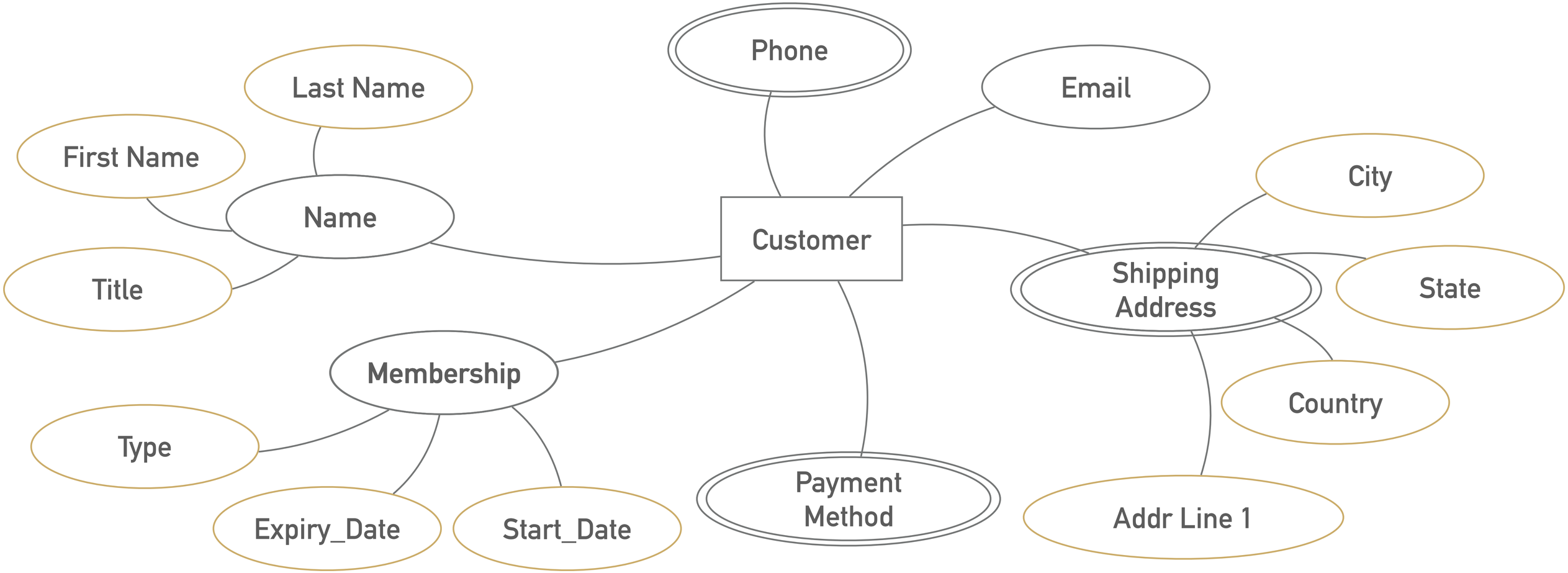
ATTRIBUTES – SINGLE-VALUED VS MILTI-VALUED

- **Single-Valued:** Can have only one value. e.g. Age
- **Multi-Valued:** Can have a set of values. e.g. Educational Certificate



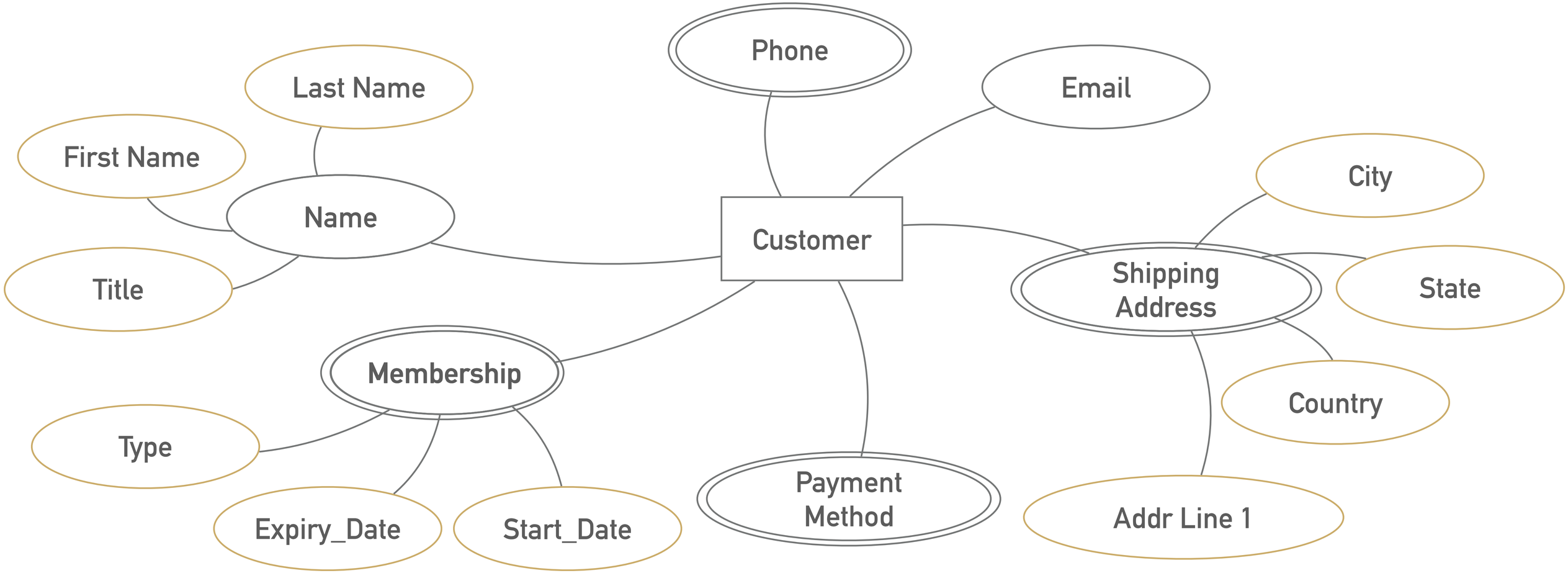
ATTRIBUTES – SINGLE-VALUED VS MILTI-VALUED

- **Single-Valued:** Can have only one value. e.g. Age
- **Multi-Valued:** Can have a set of values. e.g. Educational Certificate



ATTRIBUTES – SINGLE-VALUED VS MILTI-VALUED

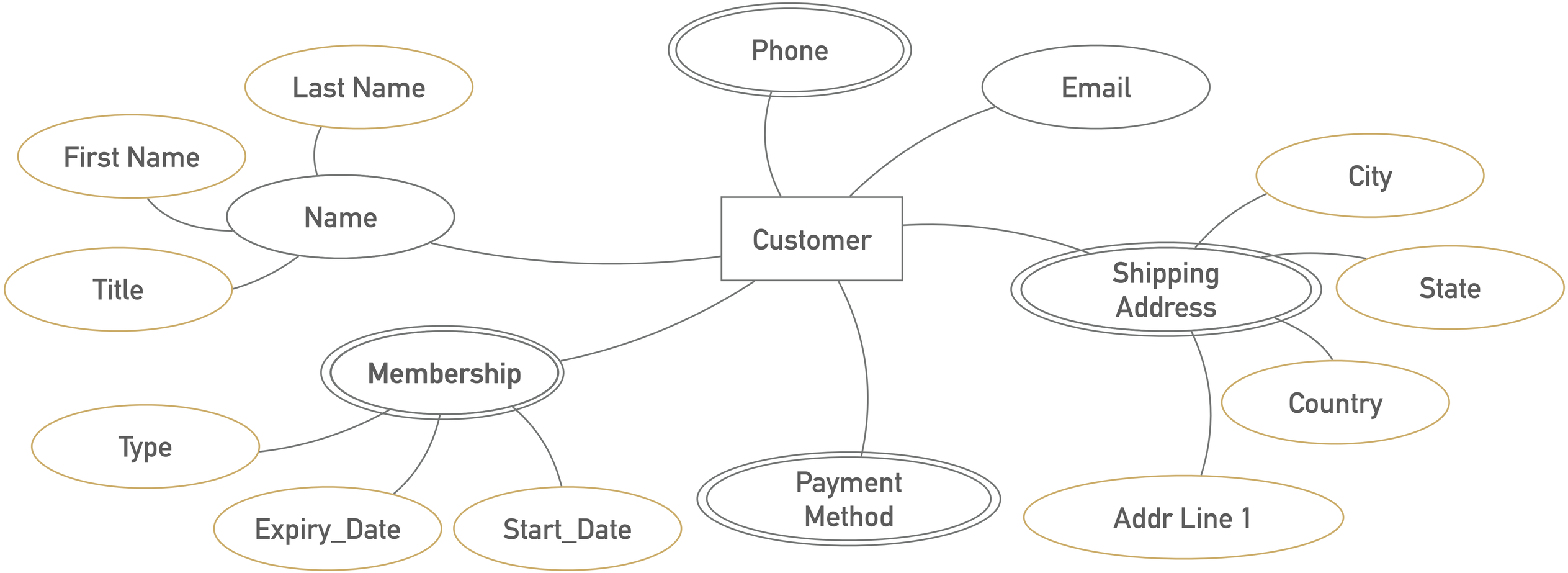
- **Single-Valued:** Can have only one value. e.g. Age
- **Multi-Valued:** Can have a set of values. e.g. Educational Certificate



ATTRIBUTES - COMPLEX

Complex = Composite + Multivalued

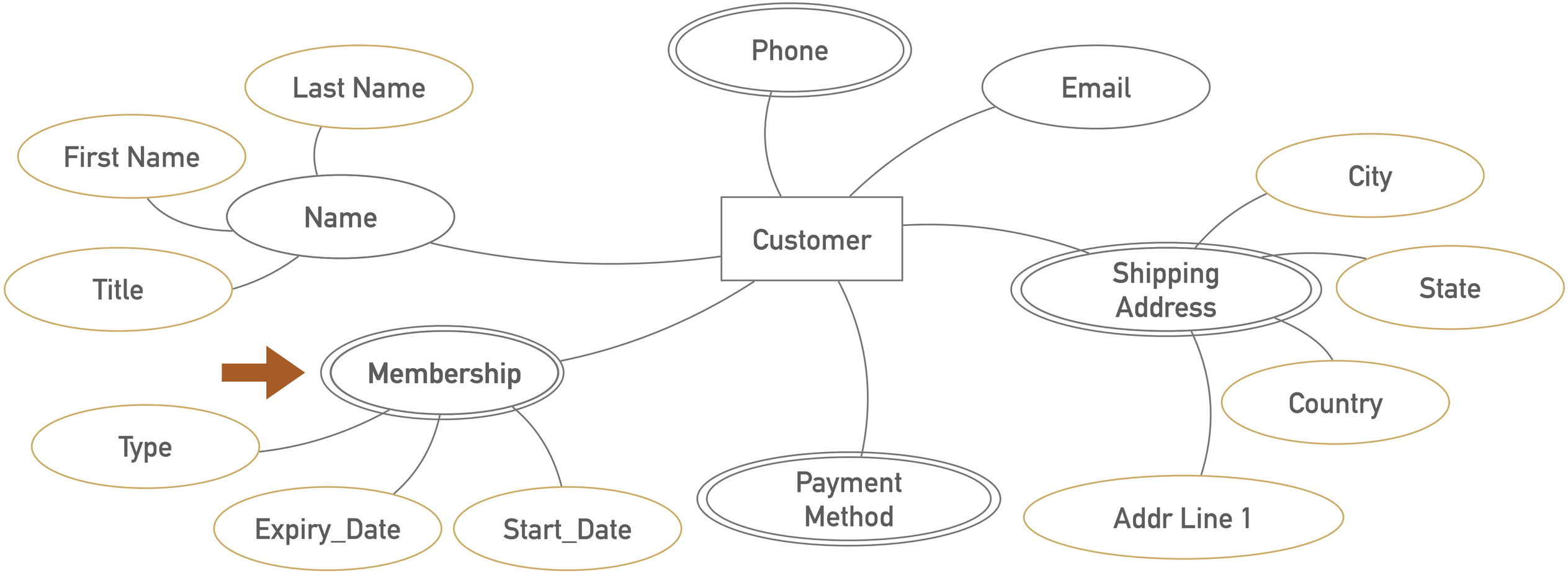
{Addresses(Title, District, Thana, Vill, Post, Road)}



ATTRIBUTES - COMPLEX

Complex = Composite + Multivalued

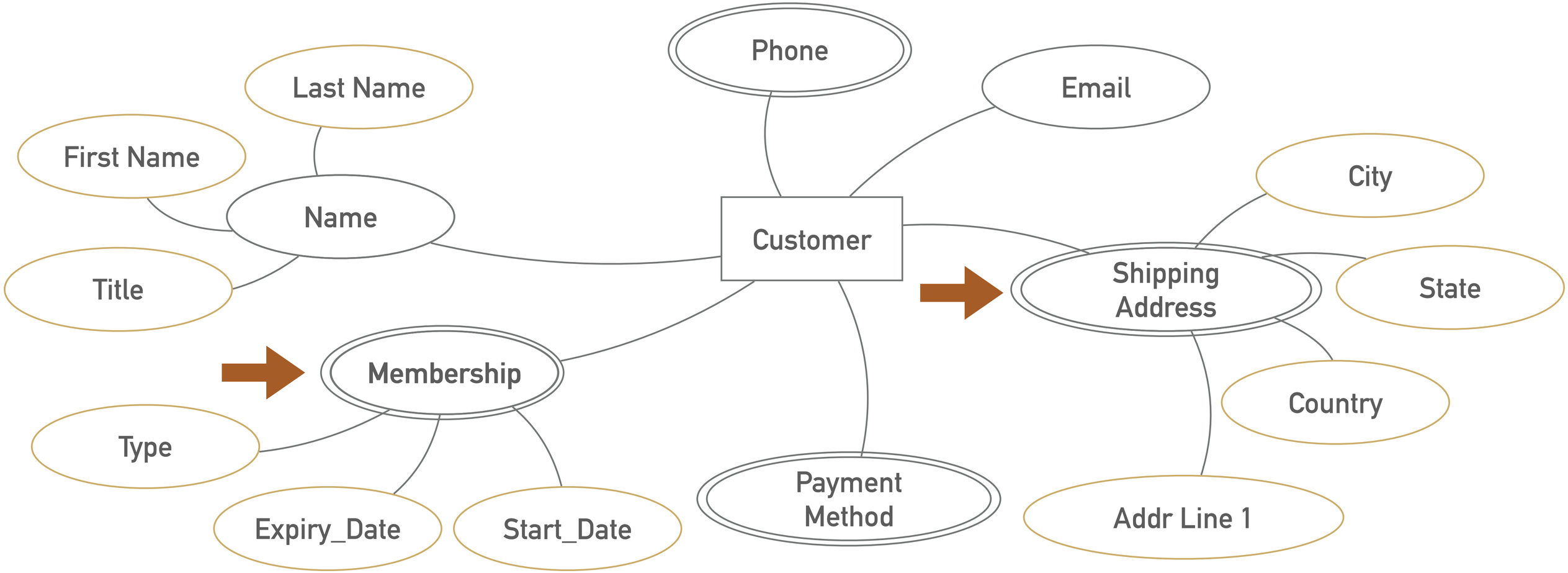
{Addresses(Title, District, Thana, Vill, Post, Road)}



ATTRIBUTES - COMPLEX

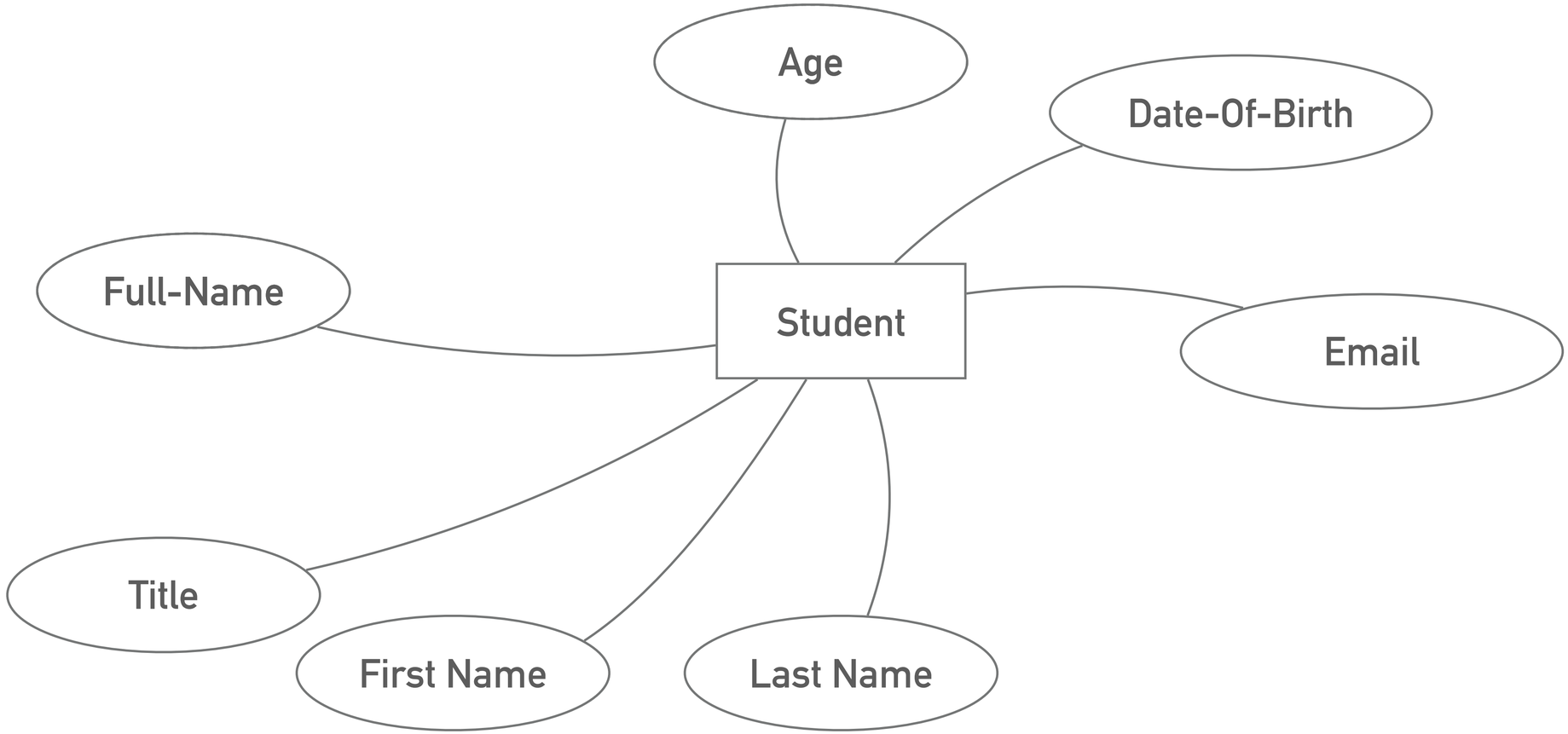
Complex = Composite + Multivalued

{Addresses(Title, District, Thana, Vill, Post, Road)}



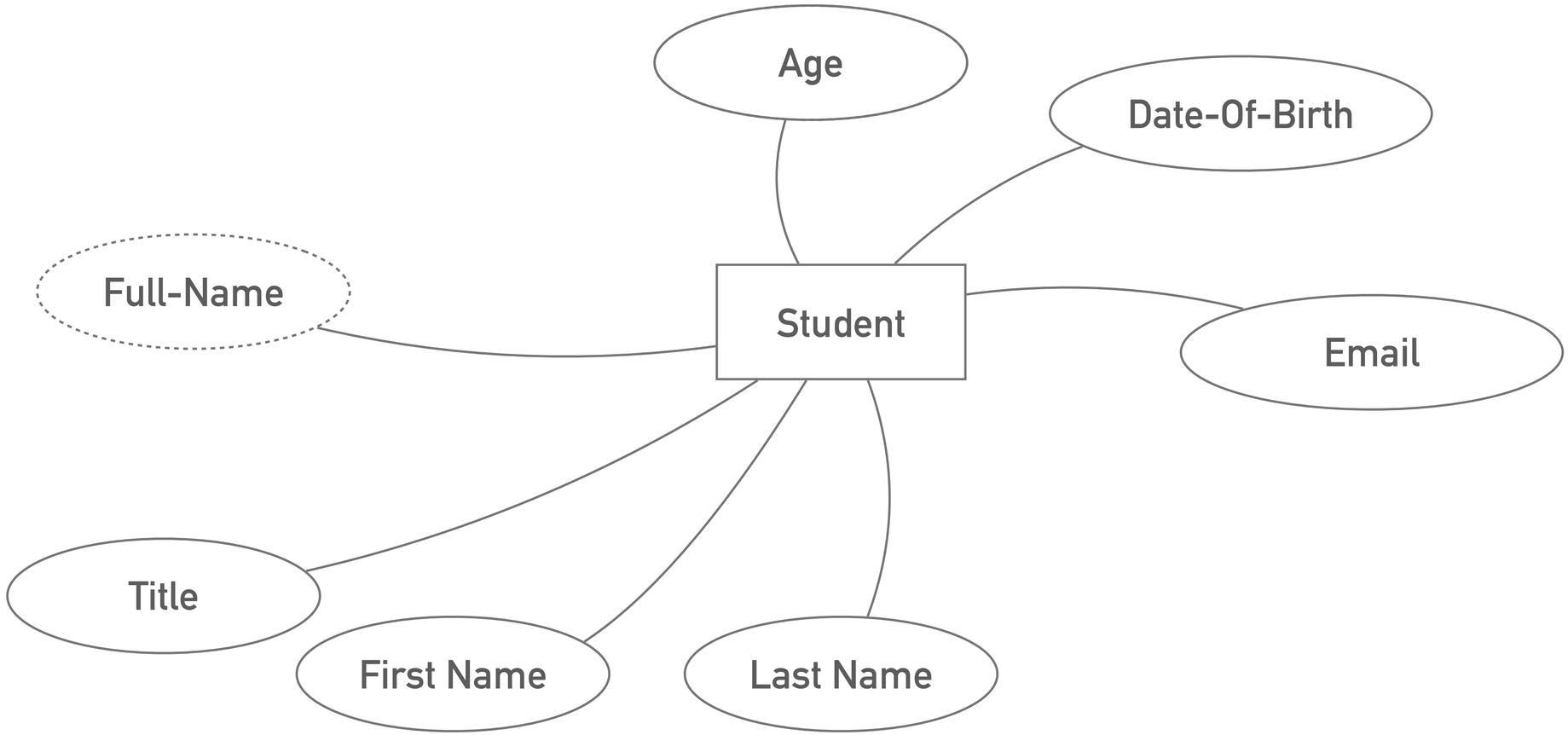
ATTRIBUTES – DERIVED VS STORED

- **Stored:** Independent information. Need to be stored. e.g. Date of Birth
- **Derived:** Can/should be calculated from other attribute. e.g. Age



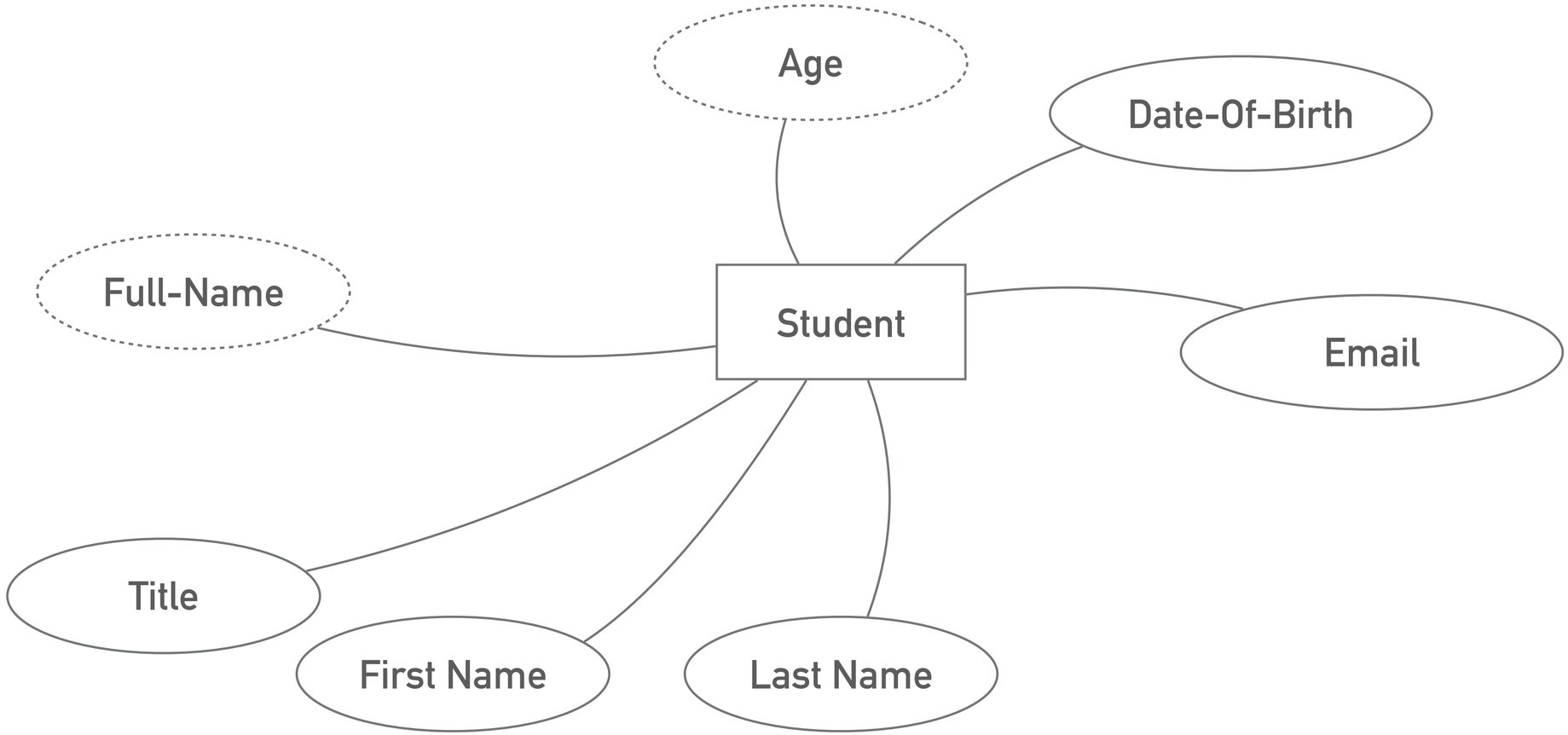
ATTRIBUTES – DERIVED VS STORED

- **Stored:** Independent information. Need to be stored. e.g. Date of Birth
- **Derived:** Can/should be calculated from other attribute. e.g. Age



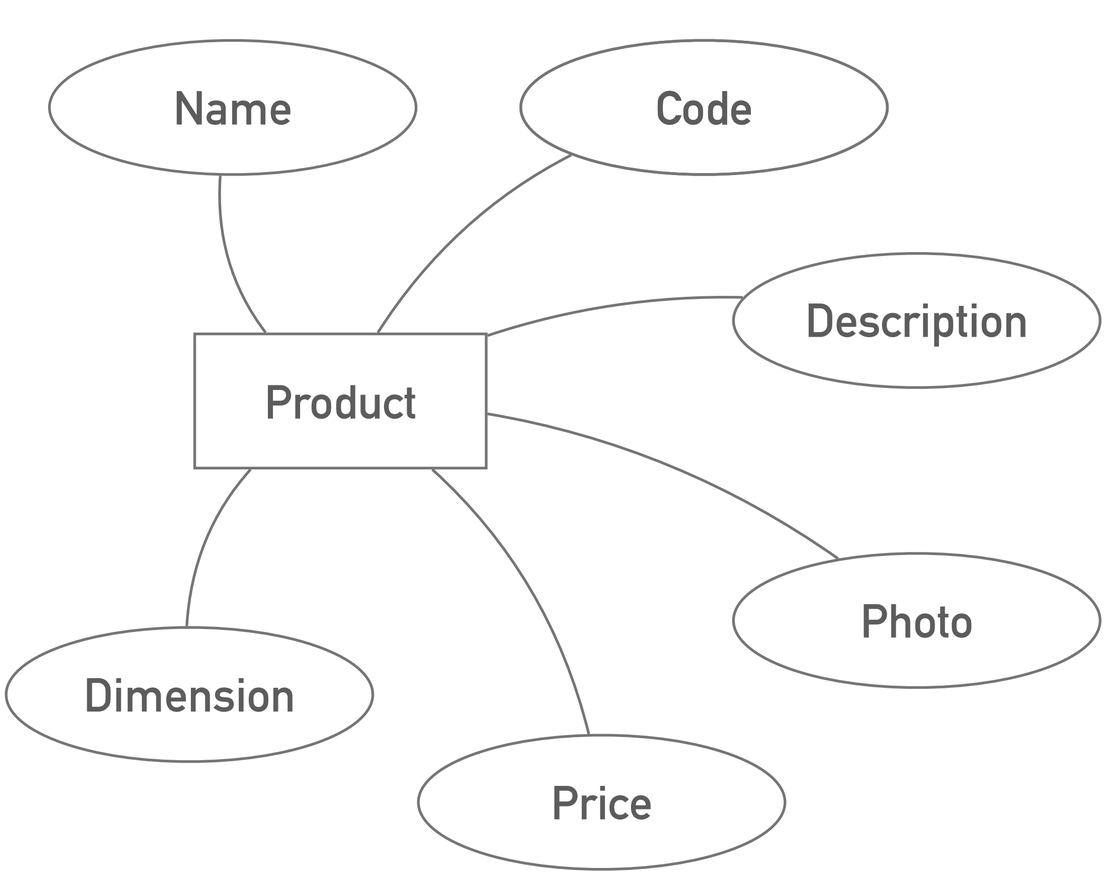
ATTRIBUTES – DERIVED VS STORED

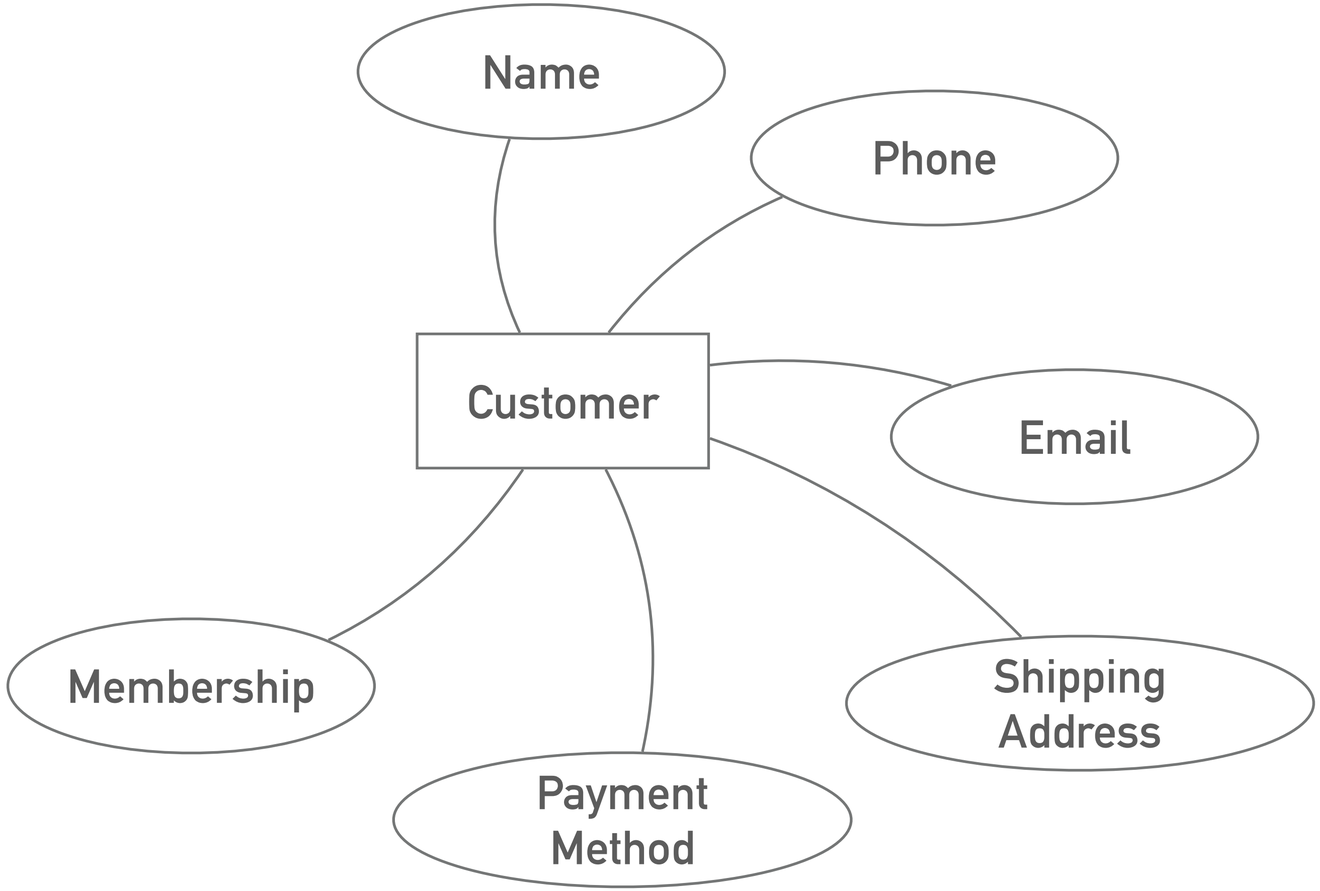
- **Stored:** Independent information. Need to be stored. e.g. Date of Birth
- **Derived:** Can/should be calculated from other attribute. e.g. Age



ATTRIBUTES – KEY ATTRIBUTES

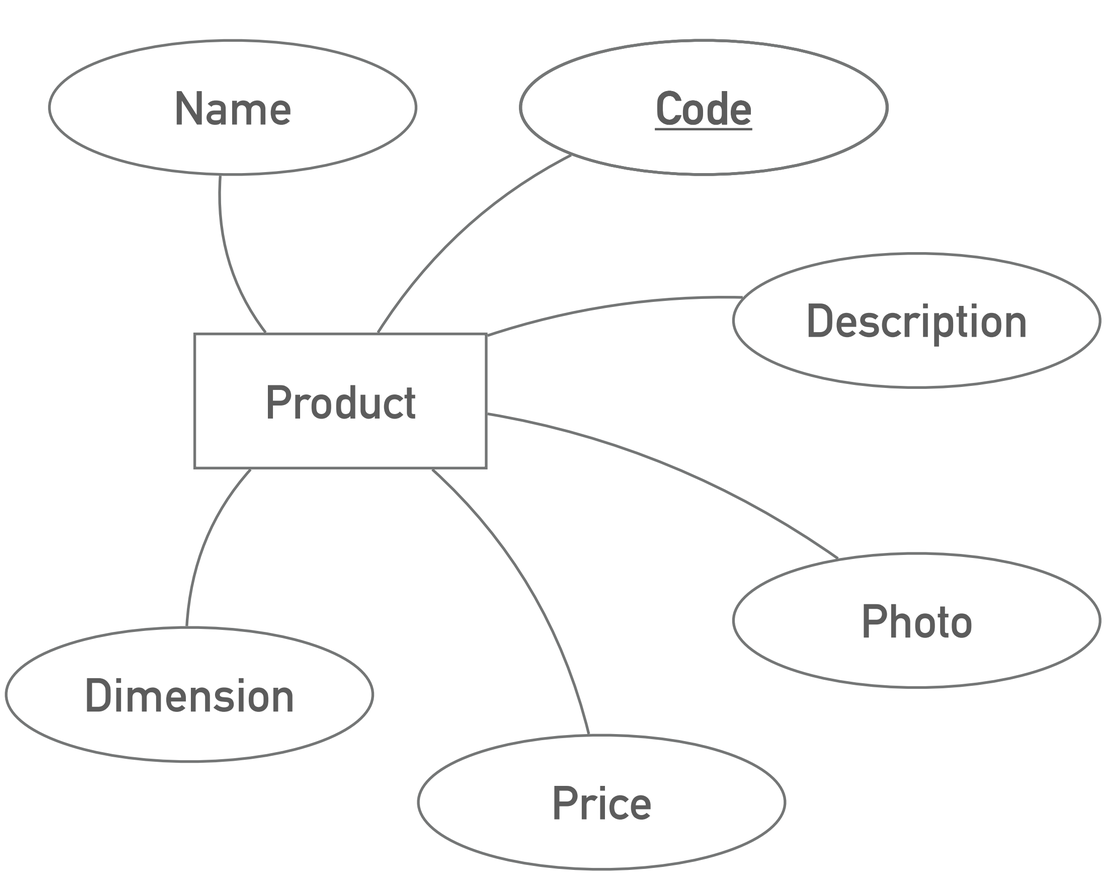
Key Attribute: Attribute that identifies Entity *uniquely* - NID, Roll Number,

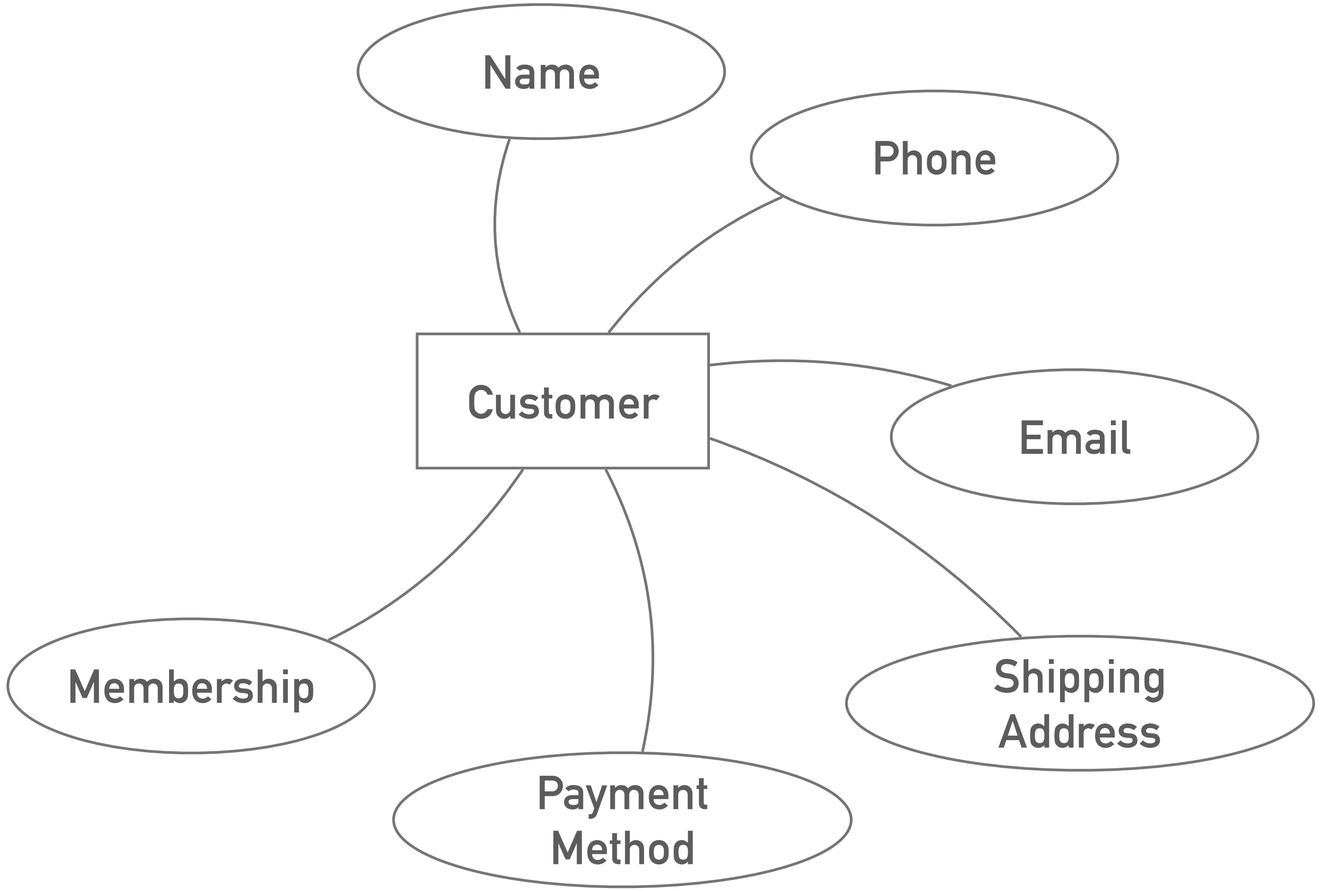




ATTRIBUTES – KEY ATTRIBUTES

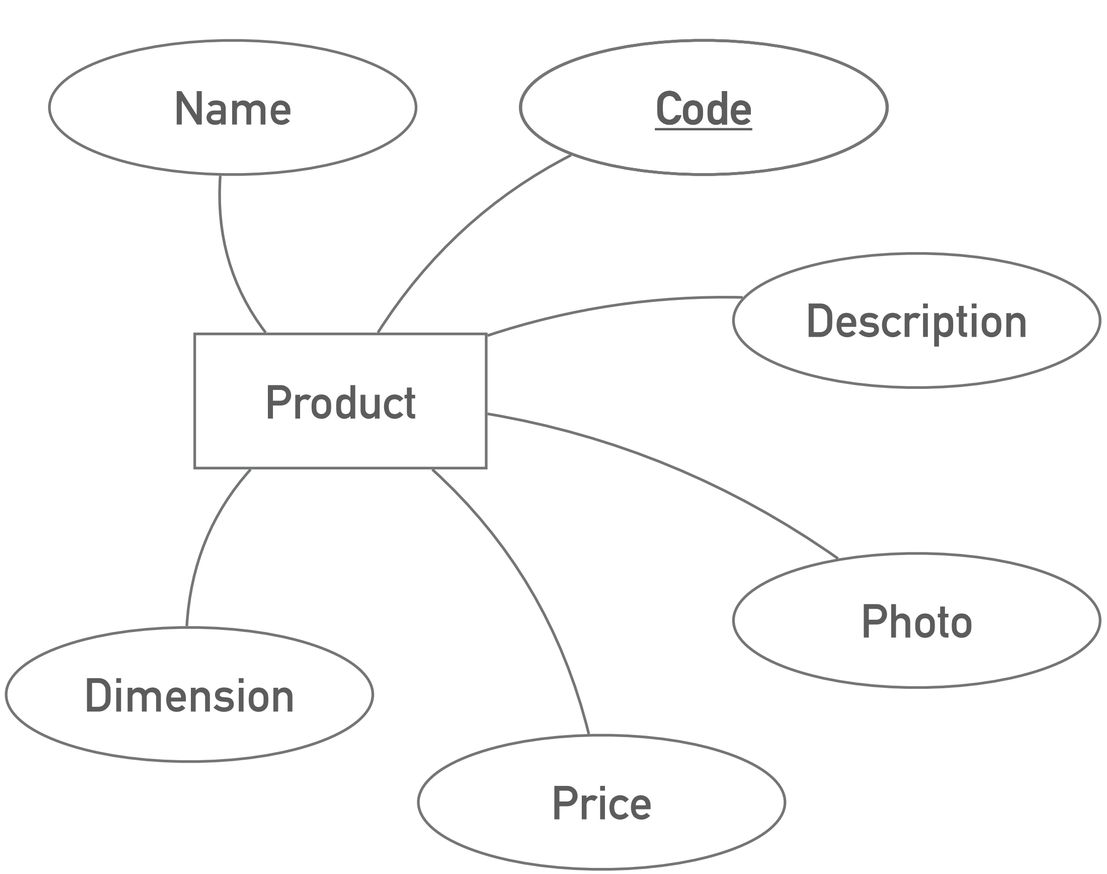
Key Attribute: Attribute that identifies Entity *uniquely* - NID, Roll Number,

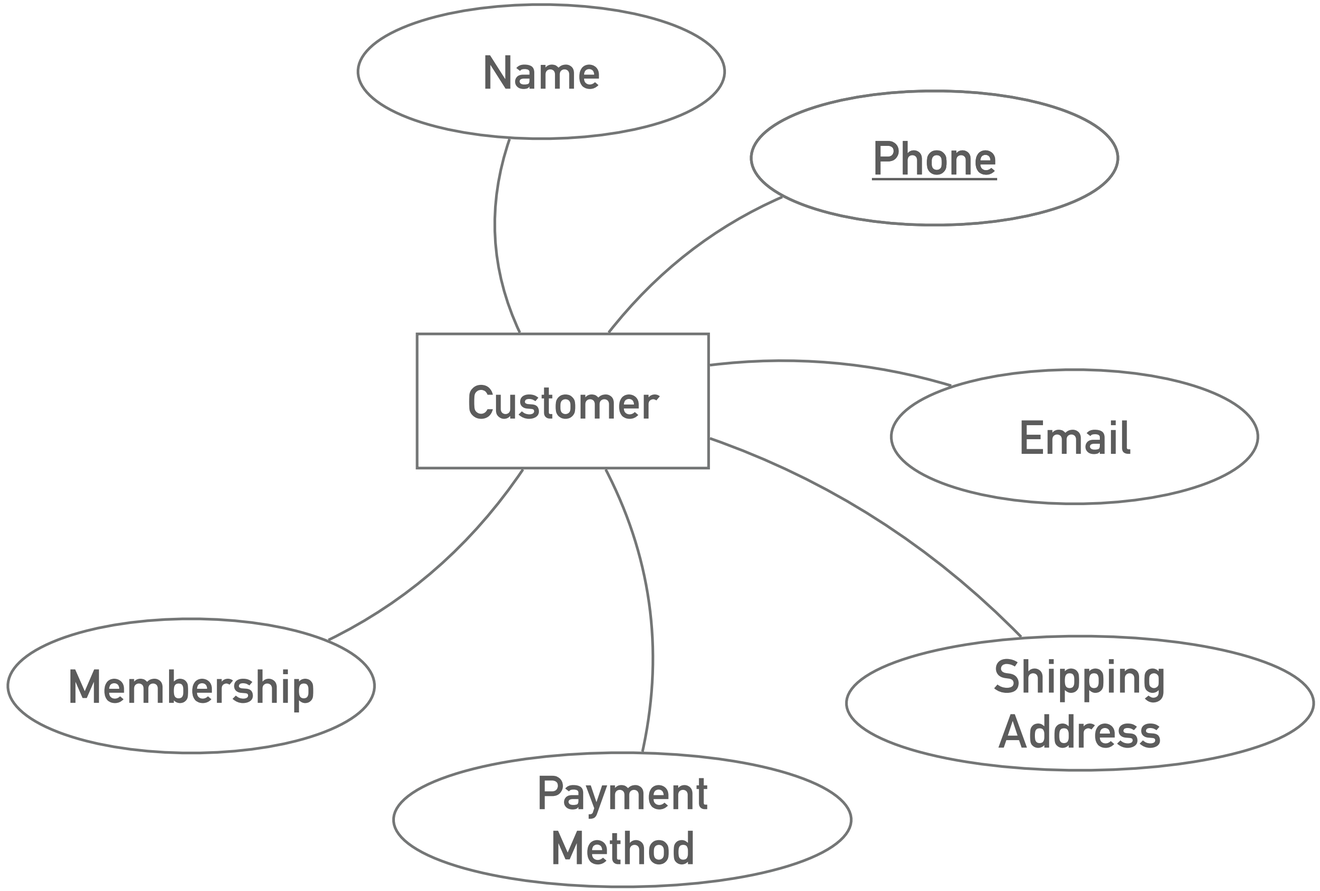




ATTRIBUTES – KEY ATTRIBUTES

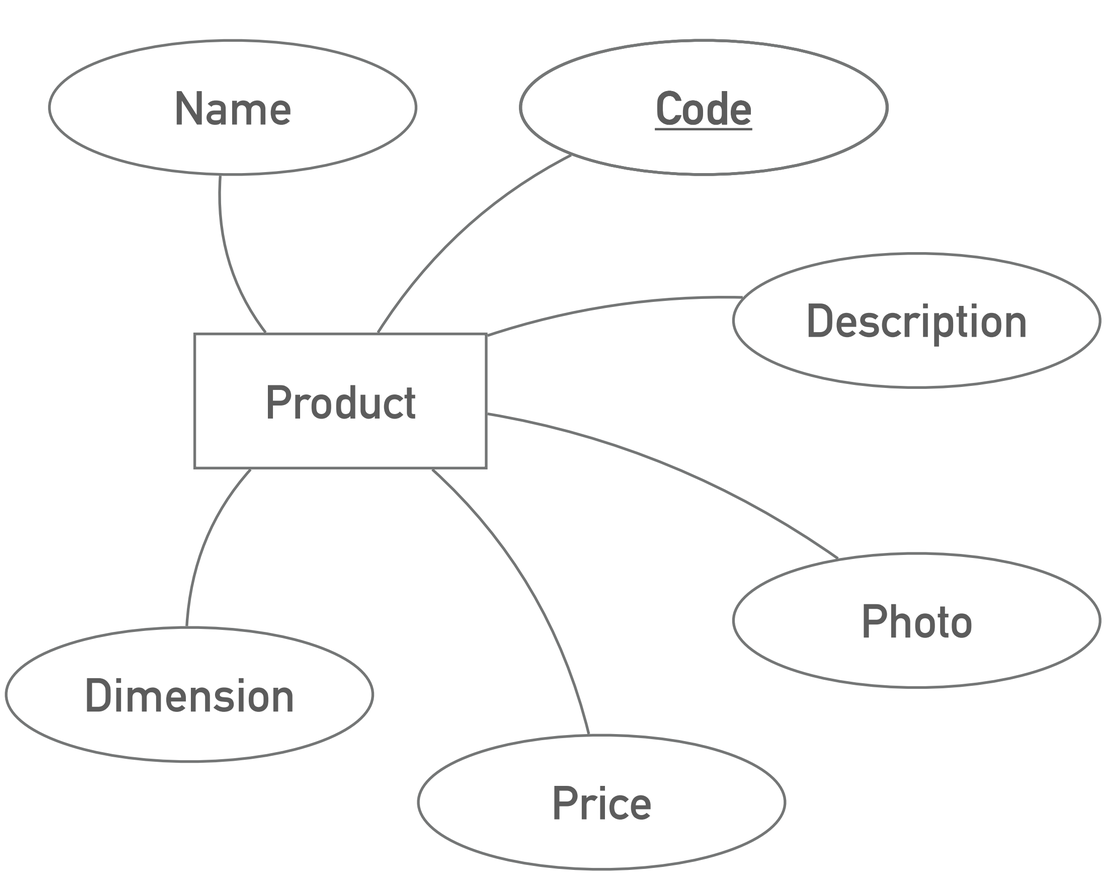
Key Attribute: Attribute that identifies Entity *uniquely* - NID, Roll Number,

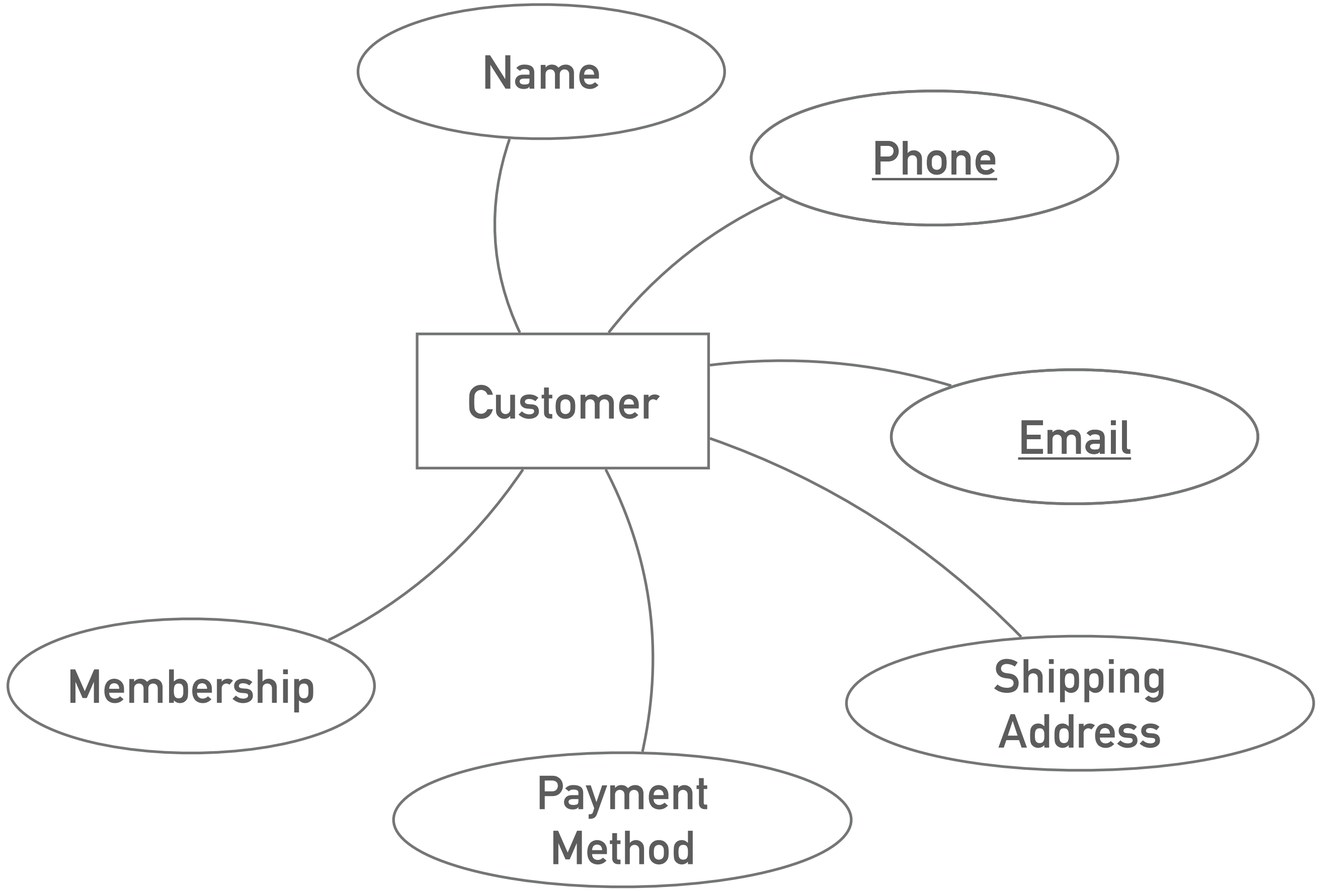




ATTRIBUTES – KEY ATTRIBUTES

Key Attribute: Attribute that identifies Entity *uniquely* - NID, Roll Number,





ATTRIBUTES – KEY ATTRIBUTES

- **Primary Key:** The key attribute that is used to refer an Entity *uniquely*. *e.g.* NID, Student-ID
- **Candidate Key:** Key attributes other than the Primary Key.
- **Composite Key:** Primary key that was build with multiple attributes. e.g., ISBN+Member-ID

ATTRIBUTES – NULL VALUES

When an Attribute value -

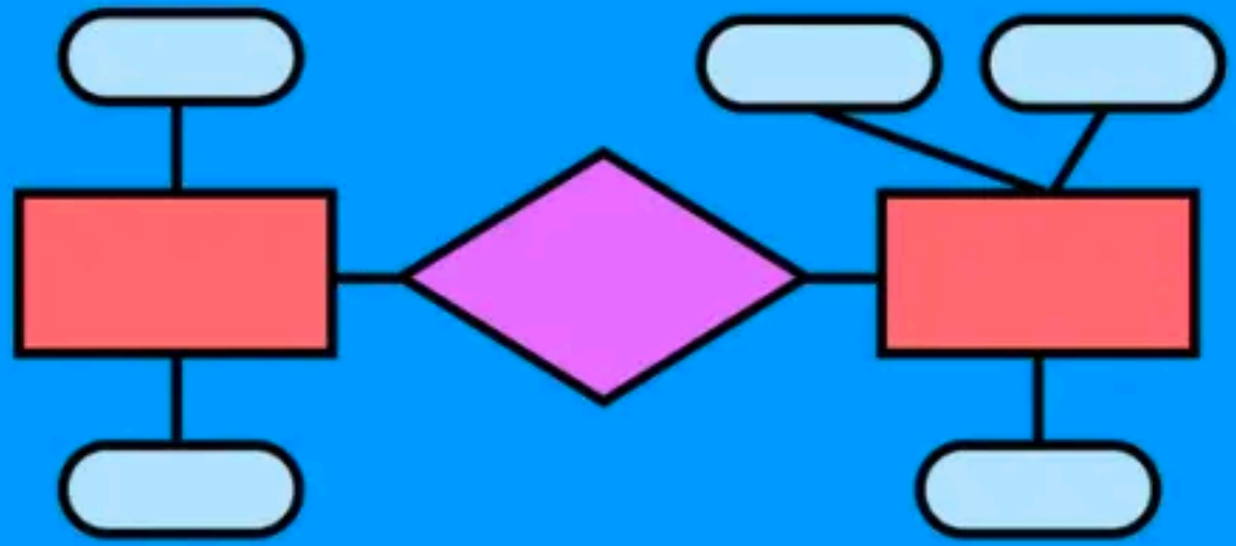
- Don’t exists
- Existence Unknown
- Exists but missing

Non-zero value

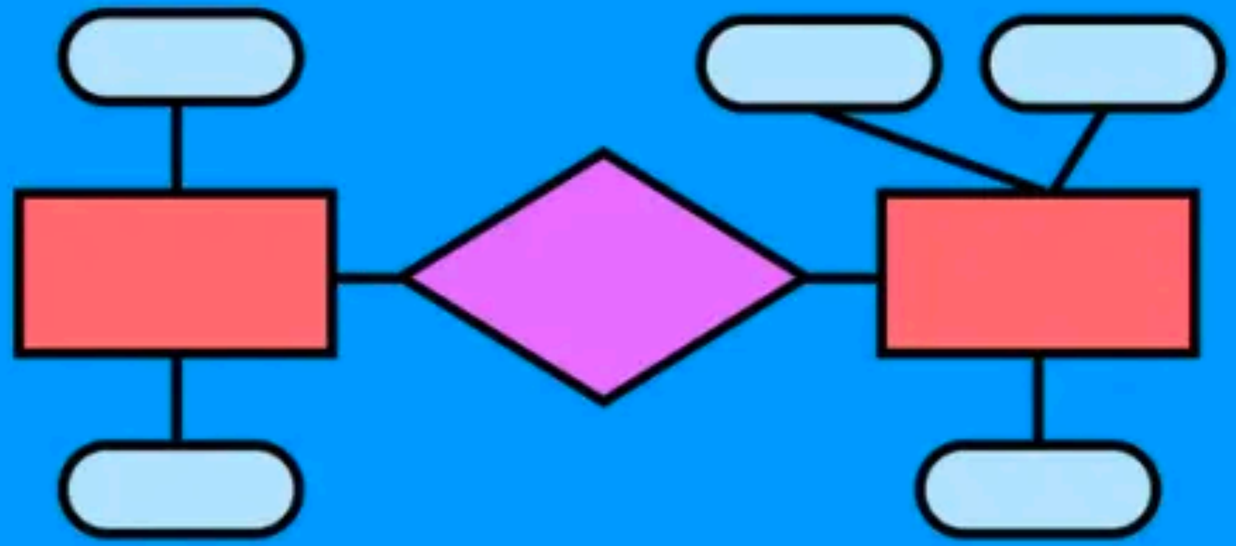
0

null

undefined

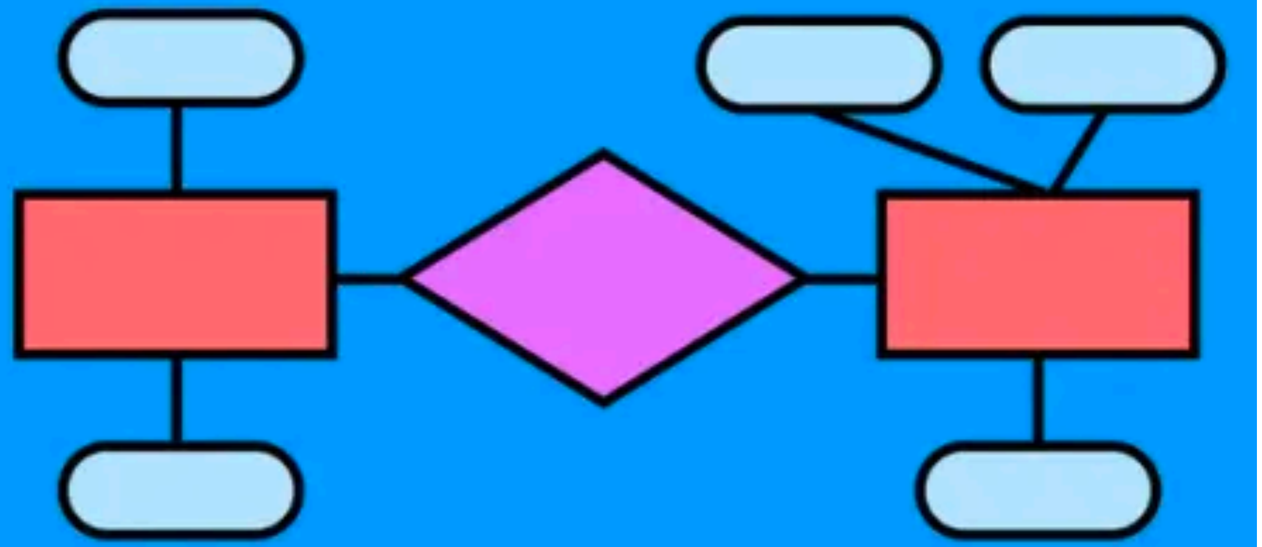


RELATIONSHIP



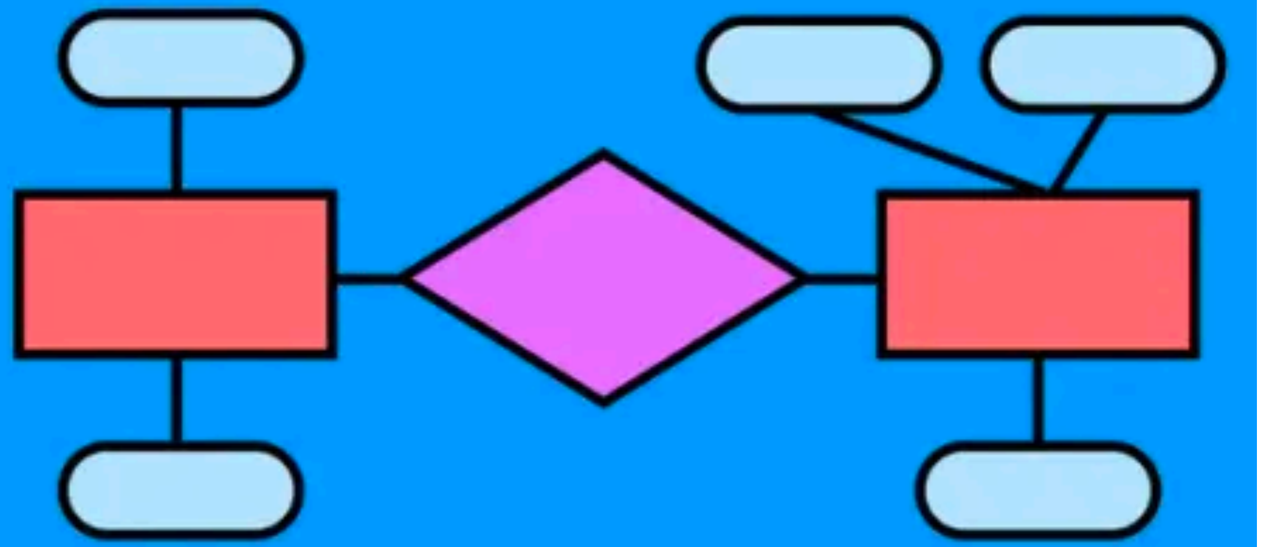
RELATIONSHIP

- Describes purposeful connection between Entities



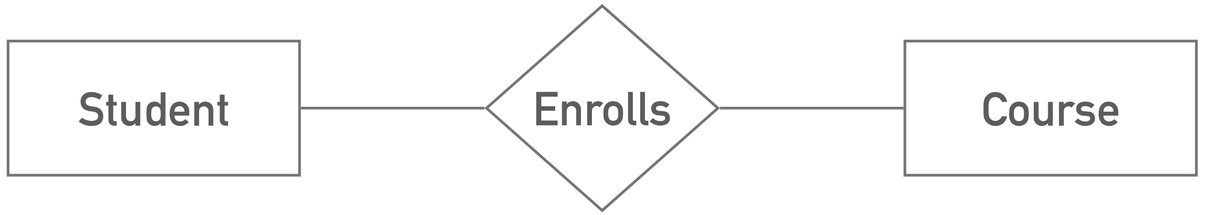
RELATIONSHIP

- Describes purposeful connection between Entities
- Represented with a Diamond in ER Diagram.



RELATIONSHIP

- Describes purposeful connection between Entities
- Represented with a Diamond in ER Diagram.

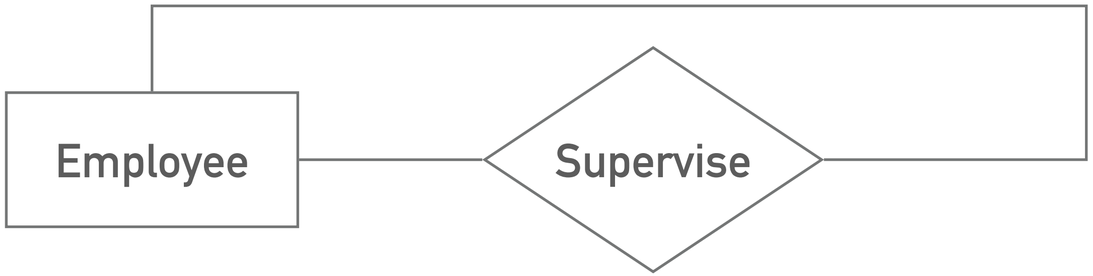


RELATIONSHIP – DEGREE OF RELATIONSHIP

- **Unary:** Linked to the Entity Type. e.g. *Employee* supervises *Employee*
- **Binary:** Association among two entities. e.g. *Publisher* publishes *Book*
- **Ternary:** Primary key that was build with multiple attributes. e.g., *Teacher* teaches *Subject* to *Student*

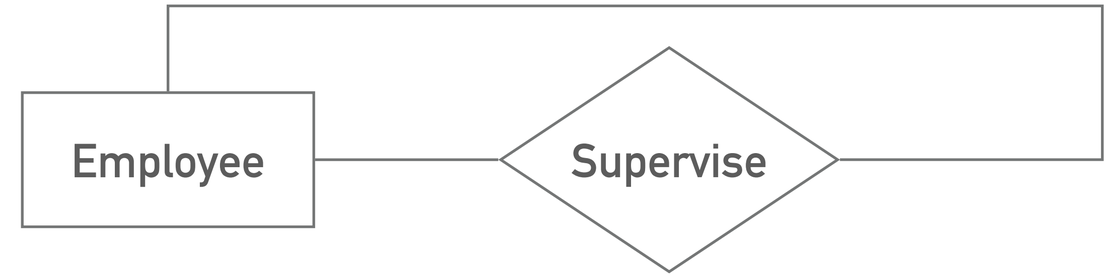
RELATIONSHIP – DEGREE OF RELATIONSHIP

- **Unary:** Linked to the Entity Type. e.g. *Employee* supervises *Employee*
- **Binary:** Association among two entities. e.g. *Publisher* publishes *Book*
- **Ternary:** Primary key that was build with multiple attributes. e.g., *Teacher* teaches *Subject* to *Student*



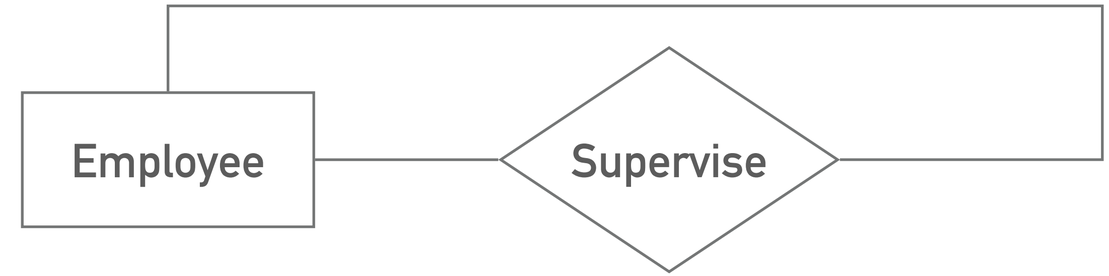
RELATIONSHIP – DEGREE OF RELATIONSHIP

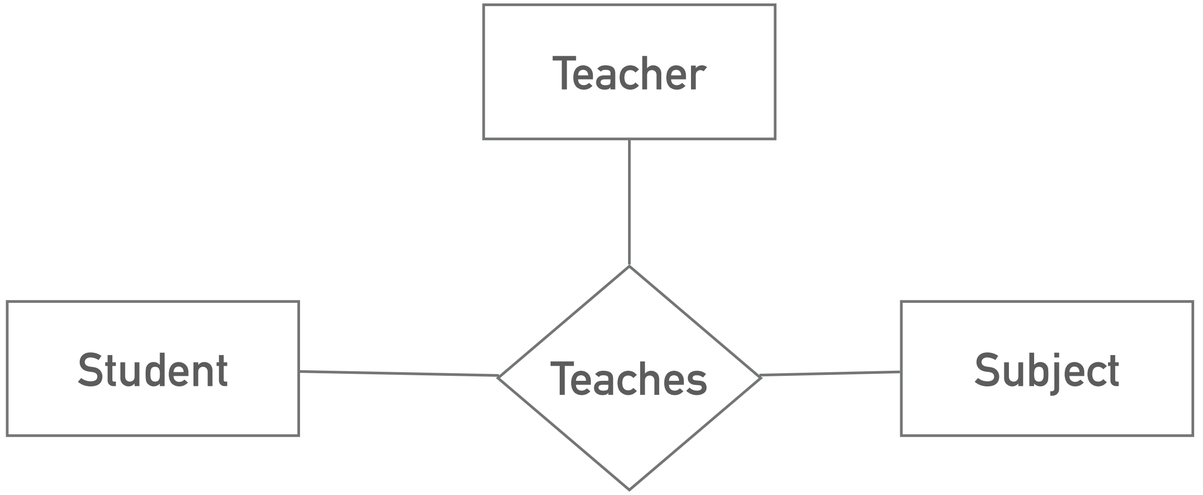
- **Unary:** Linked to the Entity Type. e.g. *Employee* supervises *Employee*
- **Binary:** Association among two entities. e.g. *Publisher* publishes *Book*
- **Ternary:** Primary key that was build with multiple attributes. e.g., *Teacher* teaches *Subject* to *Student*



RELATIONSHIP – DEGREE OF RELATIONSHIP

- **Unary:** Linked to the Entity Type. e.g. *Employee* supervises *Employee*
- **Binary:** Association among two entities. e.g. *Publisher* publishes *Book*
- **Ternary:** Primary key that was build with multiple attributes. e.g., *Teacher* teaches *Subject* to *Student*





RELATIONSHIP – CARDINALITY RATIO

Maximum number of relationship instances that an entity can participate in.

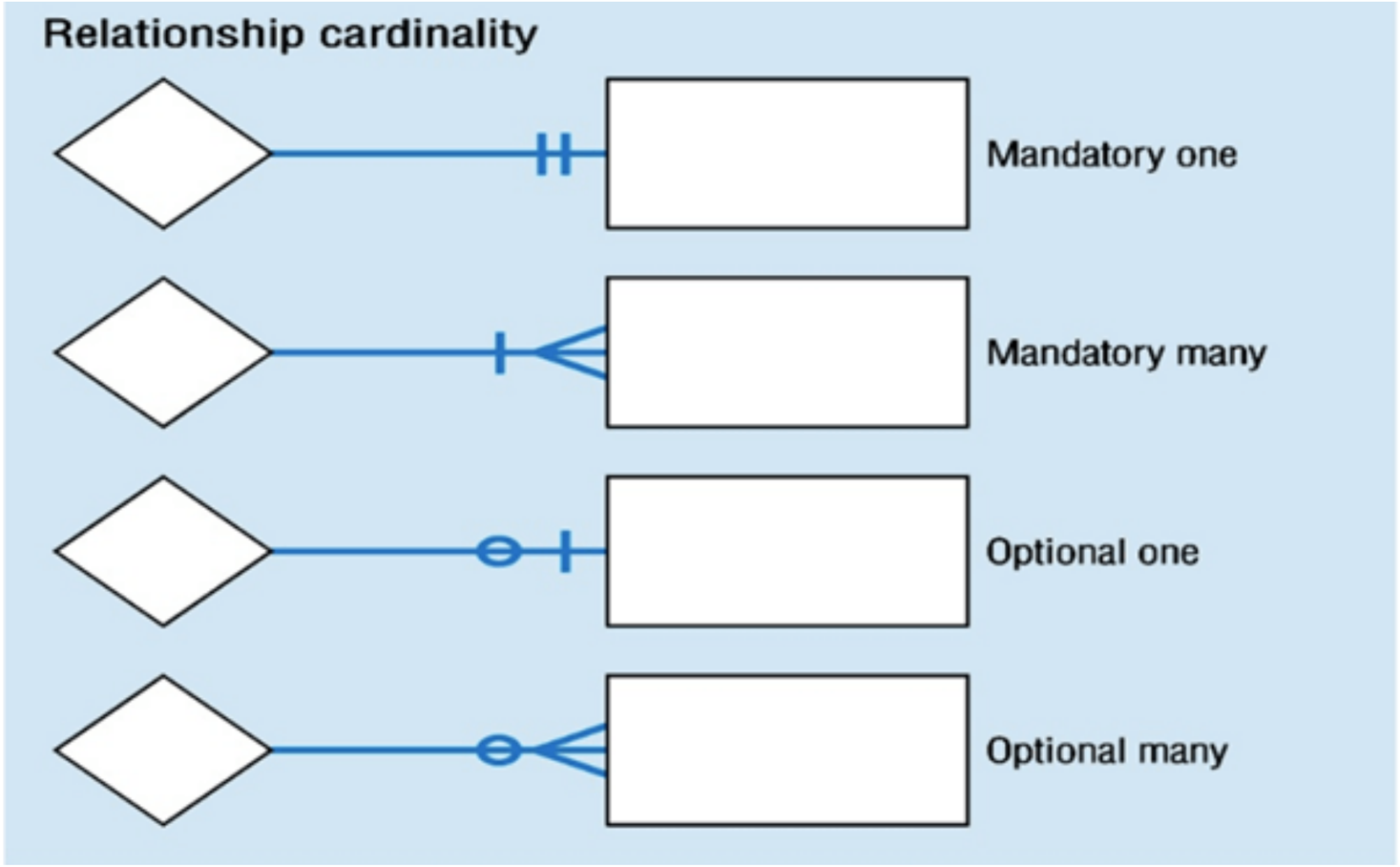
One To One:

One To Many:

Many To Many:

RELATIONSHIP – CARDINALITY RATIO (CROWS FOOT)

Maximum number of relationship instances that an entity can participate in.



RELATIONSHIP – PARTICIPATION CONSTRAINTS

Do ALL entities participates in this relationship?

NO - Partial Participation

Some Employees manages a Department.

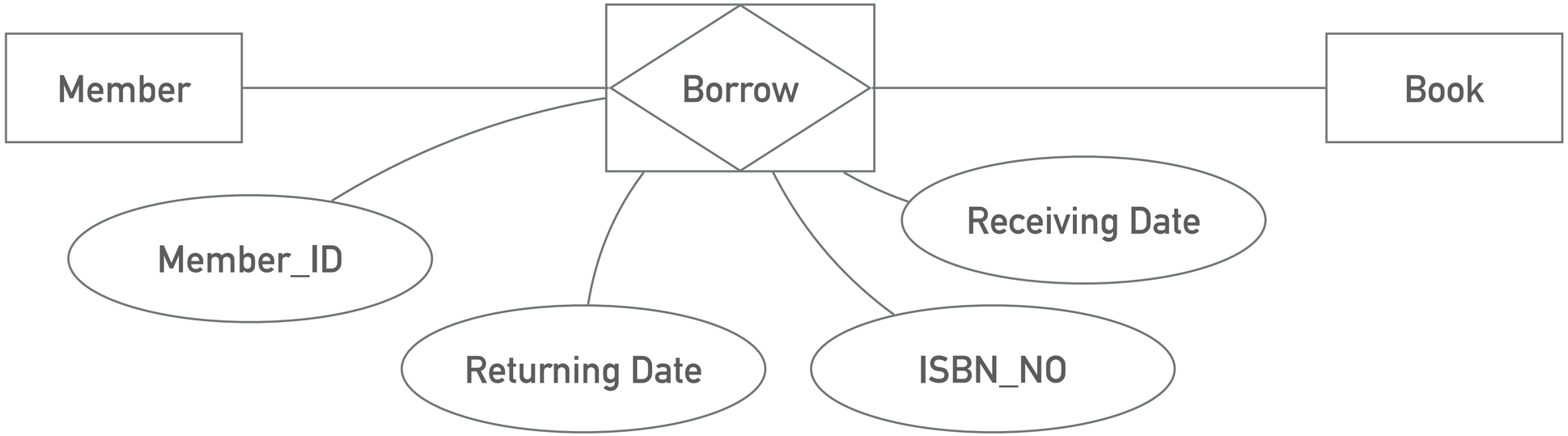
Some employees don't.

YES - Total Participation

Every Department MUST be managed by an Employee,

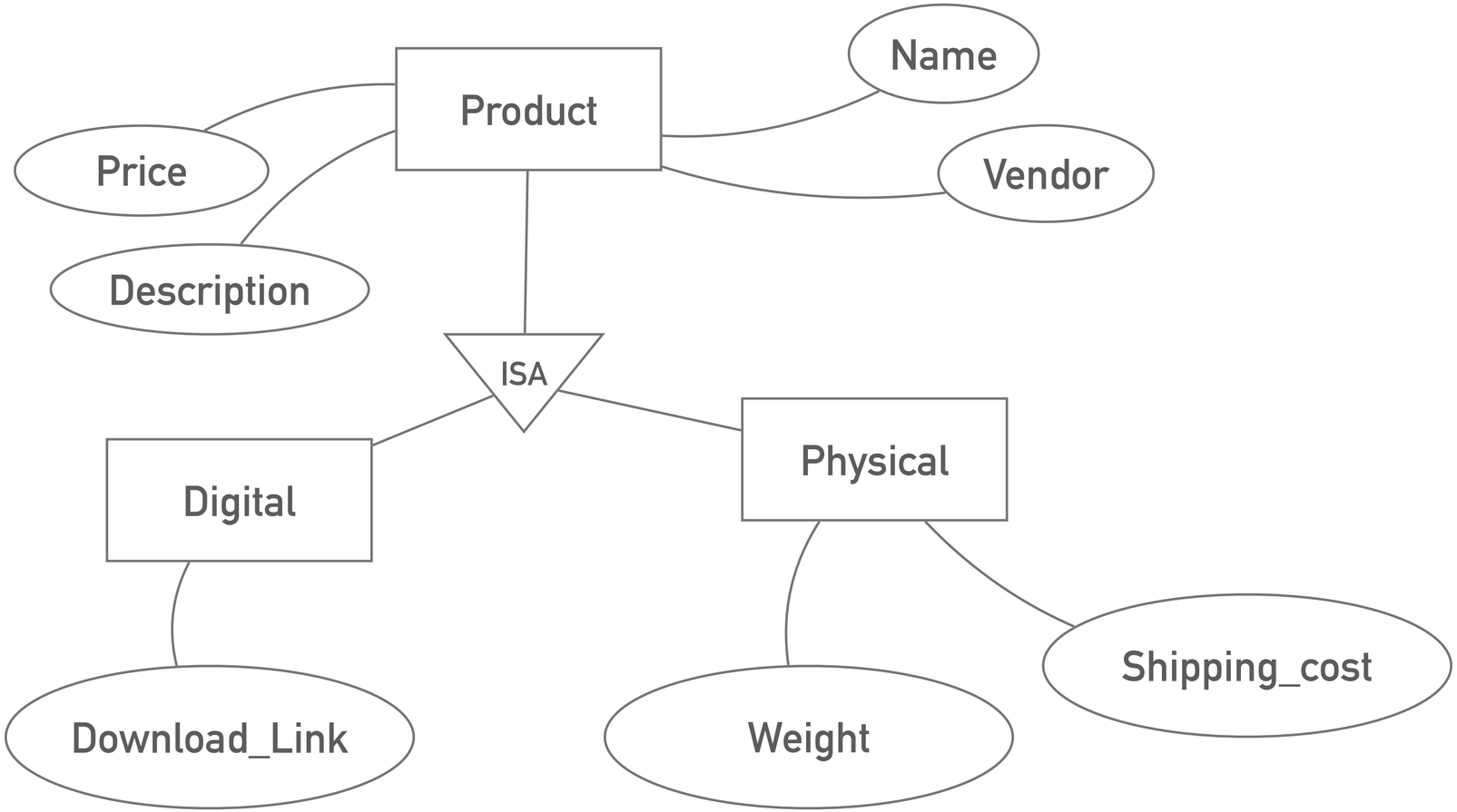
RELATIONSHIP – ASSOCIATIVE / INTERSECTION ENTITY

- Generally occurs in **Many to Many** and **Ternary Relationship**
- Can have unique identifier and other attributes
- Can have independent meaning



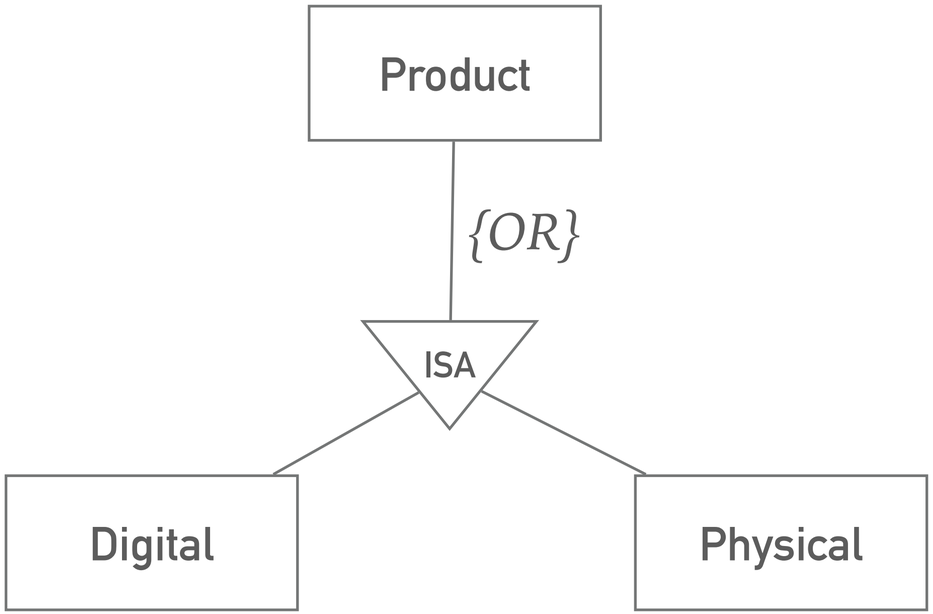
ENTITY – GENERALIZATION / SPECIALIZATION

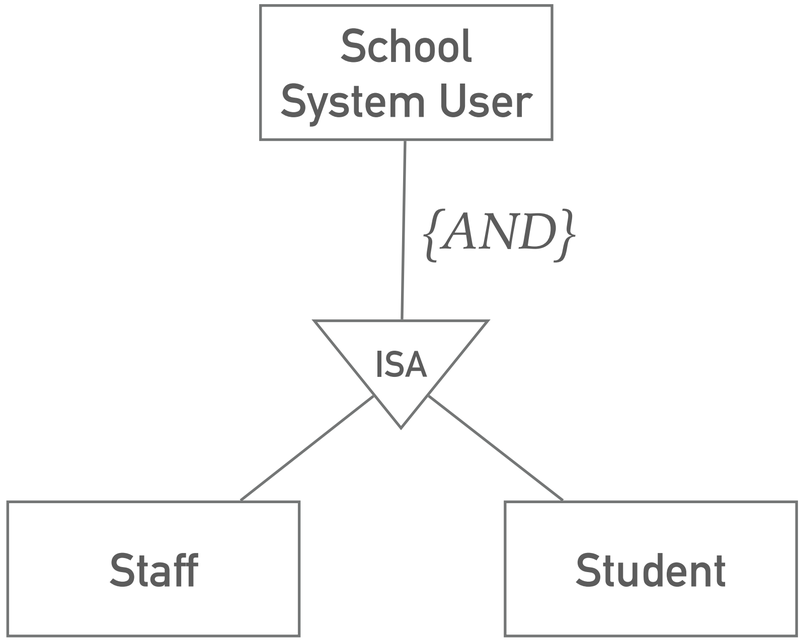
- **Generalization:** An entity type that represents a general concept at a high level. (Superclass)
- **Specialization:** An entity type that represents a specific concept at lower levels. (Subclass)



ENTITY – GENERALIZATION / SPECIALIZATION

- **Disjoint:** An entity occurrence can be a member of only one of the subclasses. (OR)
- **Overlapping:** An entity occurrence can be a member of more then one of the subclasses. (AND)

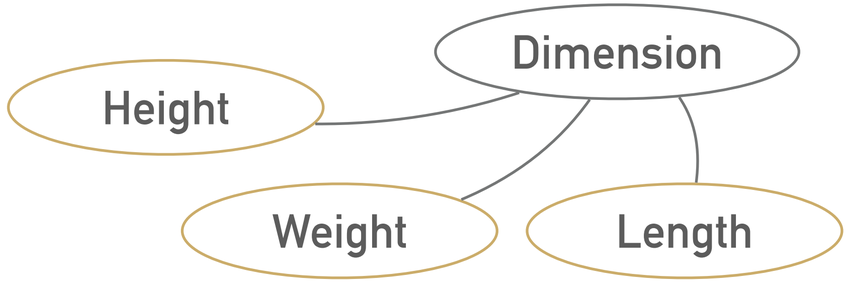




RECAP – DIAGRAM NOTATIONS

ENTITY		Rectangle
WEAK ENTITY		Double Rectangle
ATTRIBUTE		Oval
KEY ATTRIBUTE		Oval + Underline
MULTIVALUED ATTRIBUTE		Double Oval

RECAP – DIAGRAM NOTATIONS

COMPOSITE ATTRIBUTE		Oval from Oval
DERIVED ATTRIBUTE		Dashed Oval
RELASHONSHIP		Diamond
IDENTIFYING RELATIONSHIP		Double Diamond

ERD – UNIVERSITY DATABASE EXAMPLE

Consider the following requirements:

1. The university offers one or more programs.
2. Program is made up of one or more courses.
3. Student must enroll in a program.
4. Student takes the courses that are part of his program.

Source: Learning MySQL by Saied M.M. Tahaghoghi, Hugh E. Williams (Publisher: Oreilly)

ERD – UNIVERSITY DATABASE EXAMPLE (CONT.)

Consider the following requirements:

1. The university offers one or more **programs**.
2. **Program** is *made up* of one or more **courses**.
3. **Student** must *enroll in* a **program**.
4. **Student** *takes* the **courses** that are part of his **program**.

Source: Learning MySQL by Saied M.M. Tahaghoghi, Hugh E. Williams (Publisher: Oreilly)

ERD – UNIVERSITY DATABASE EXAMPLE (CONT.)

Consider the following requirements:

5. A program has a name, a program identifier, the total credit points required to graduate, and the year it commenced.

6. A course has a name, a course identifier, a credit point value, and the year it commenced.

7. Students have a given name, a surname, a student identifier, a date of birth, and the year they first enrolled.

Source: Learning MySQL by Saied M.M. Tahaghoghi, Hugh E. Williams (Publisher: Oreilly)

ERD – UNIVERSITY DATABASE EXAMPLE (CONT.)

Consider the following requirements:

8. When a **student** *takes* a **course**, the year and semester he attempted it are recorded.

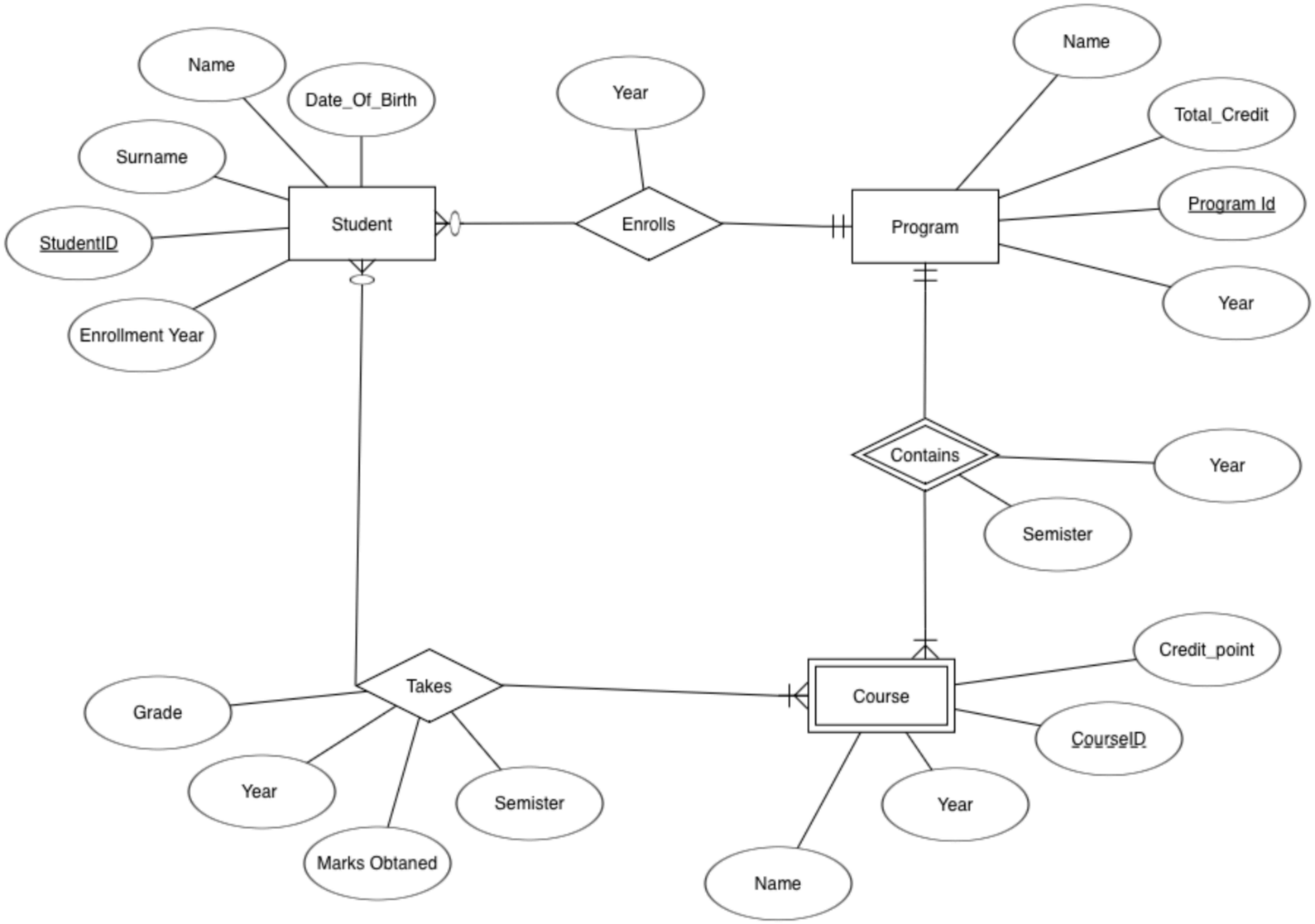
9. When he finishes the **course**, a grade (such as A or B) and a mark (such as 60 percent) are recorded.

10. Each course in a program is sequenced into a year (for example, year 1) and a semester (for example, semester 1).

Source: Learning MySQL by Saied M.M. Tahaghoghi, Hugh E. Williams (Publisher: Oreilly)

ERD – UNIVERSITY DATABASE EXAMPLE (CONT.)

Here is the diagram we drew live in class:



QUESTIONS?