

# The E-R Model

## Weak Entity

---

# ENTITY

## ENTITY :

- The basic object that the ER model represents is an entity, which is a thing in the real world with an independent existence.
- An entity may be an object with a **physical existence** (for example, a particular person, car, house, or employee) or it may be an object with a **conceptual existence** (for instance, a company, a job, or a university course).
- There are two types of entities :
  - **Strong entity**
  - **Weak entity**

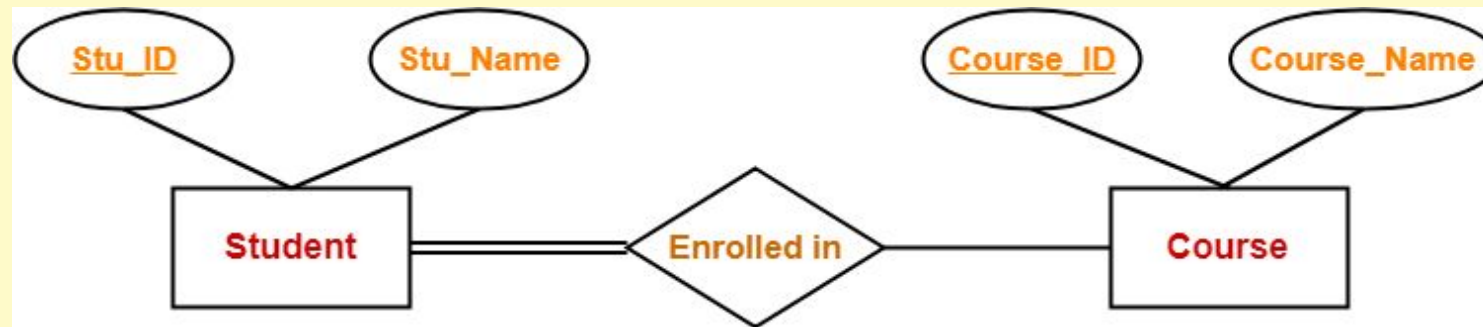
# STRONG ENTITY

## STRONG ENTITY :

- A strong entity set is an entity set that contains sufficient attributes to uniquely identify all the entities in the entity set.
- In other words, a **primary key** exists for a strong entity set.
- Primary key of a strong entity set is represented by underlining it.
- A **single rectangle** is used for representing a strong entity set.

# STRONG ENTITY

Example:



# WEAK ENTITY

- **Entity types that do not have key attributes of their own are called weak entity types.**
- In other words, **a primary key does not exist for a weak entity set.** However, it contains a partial key called as a **discriminator.**
- Weak entities are identified by being related to specific entities from another entity type (strong entity) in combination with one of their attribute values.
- **The combination of discriminator and primary key of the strong entity set makes it possible to uniquely identify all entities of the weak entity set.**

# WEAK ENTITY

- Thus, this combination serves as a primary key for the weak entity set. Clearly, this primary key is not formed by the weak entity set completely.

Primary key of weak entity set

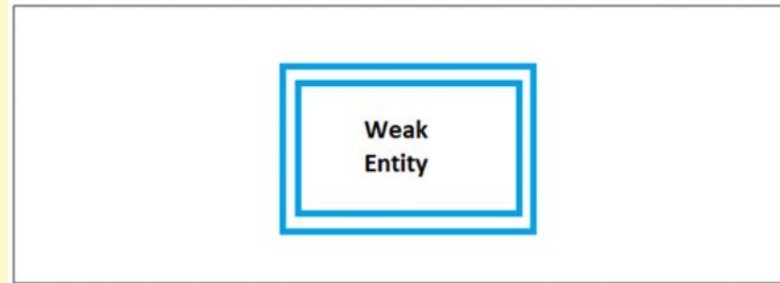
= Its own discriminator + Primary key of strong entity set

- We call this other entity type the **identifying or owner entity type**, and we call the relationship type that relates a weak entity type to its owner the **identifying relationship** of the weak entity type.

# WEAK ENTITY

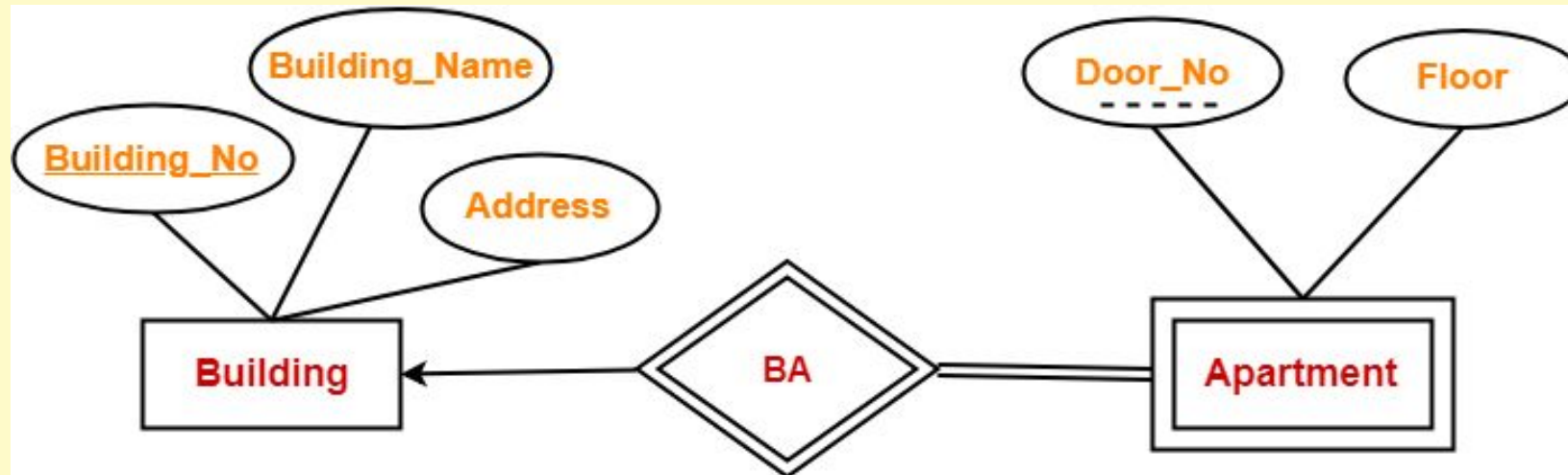
## Representation

- A **double rectangle** is used for representing a **weak entity set**
- The **double diamond** symbol is used for representing the **relationship between a strong entity and weak entity** which is known as identifying relationship
- **Double lines** are used for presenting the **connection with a weak entity set** with relationship



# WEAK ENTITY

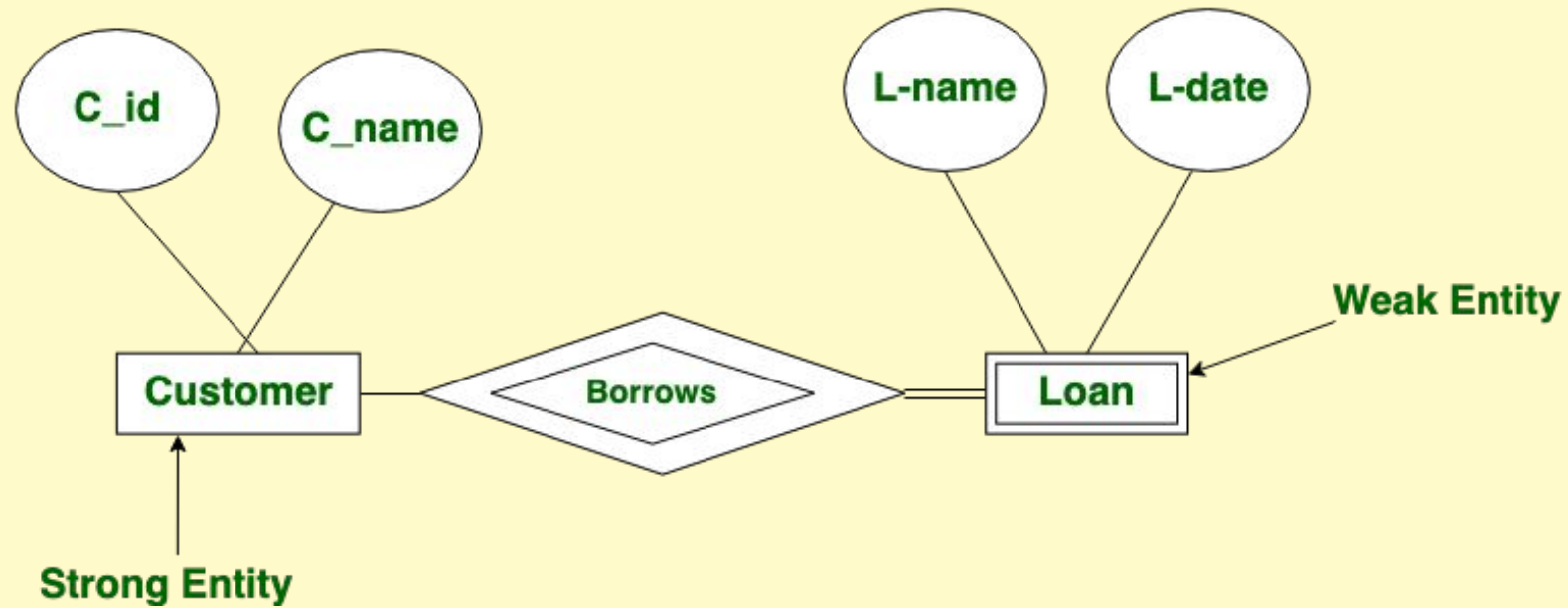
Example :





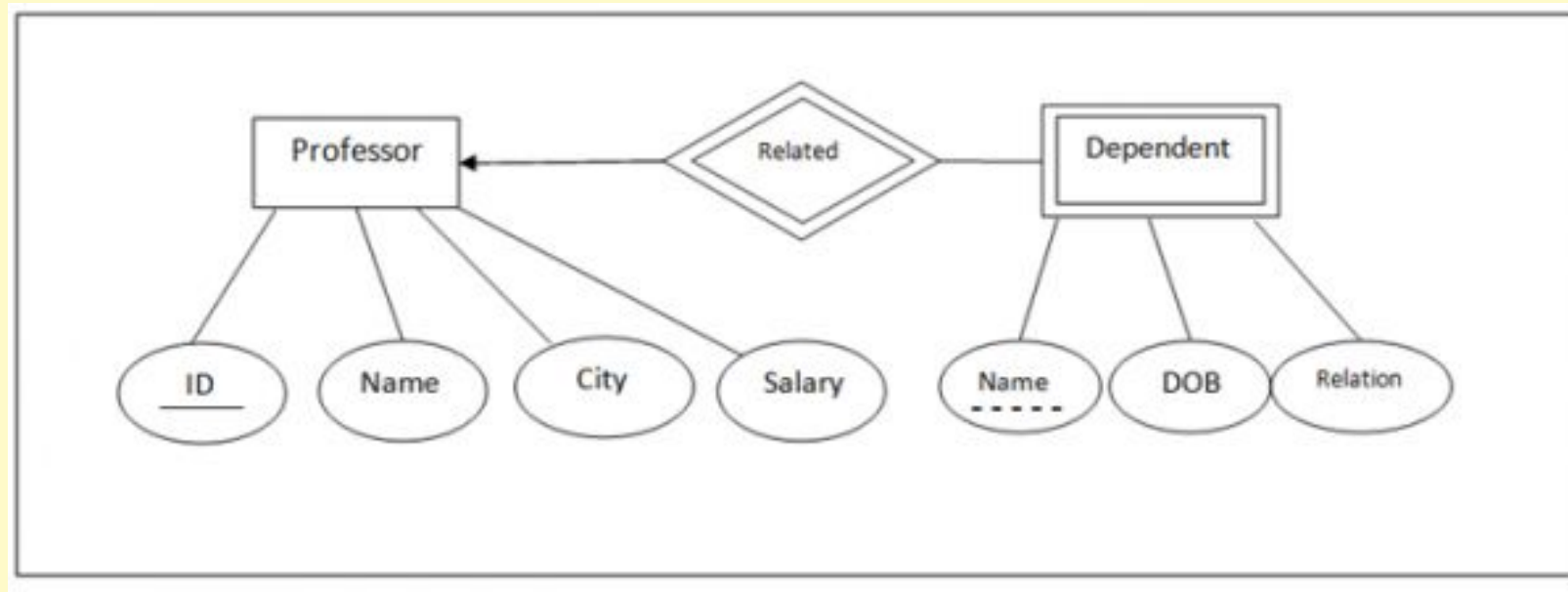
# WEAK ENTITY

Example :



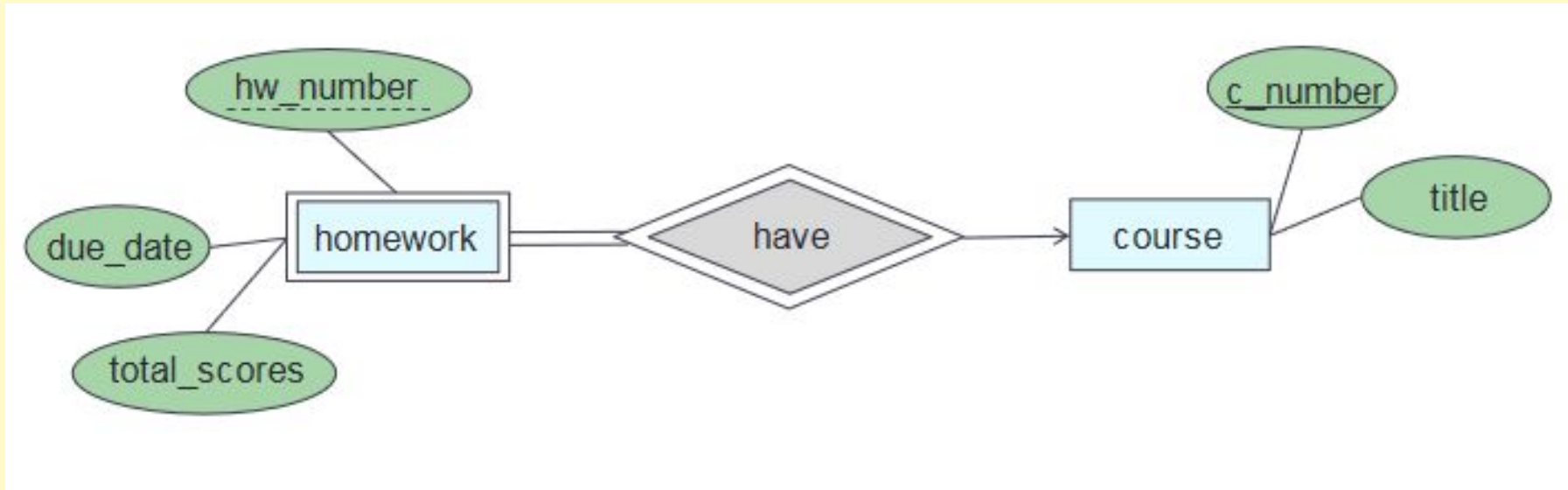
# WEAK ENTITY

Example :



# WEAK ENTITY

Example :



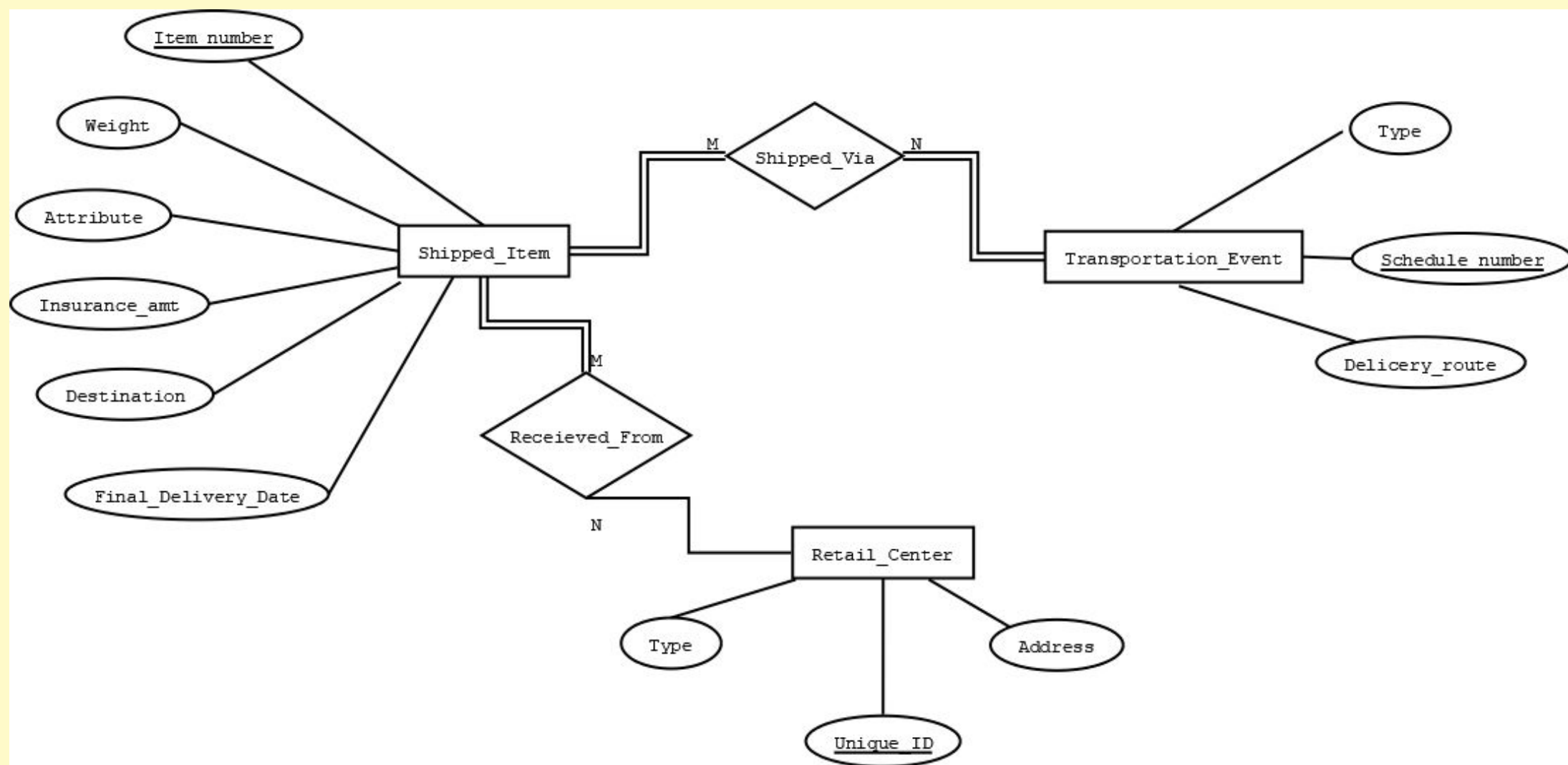
# STRONG V/s WEAK ENTITY

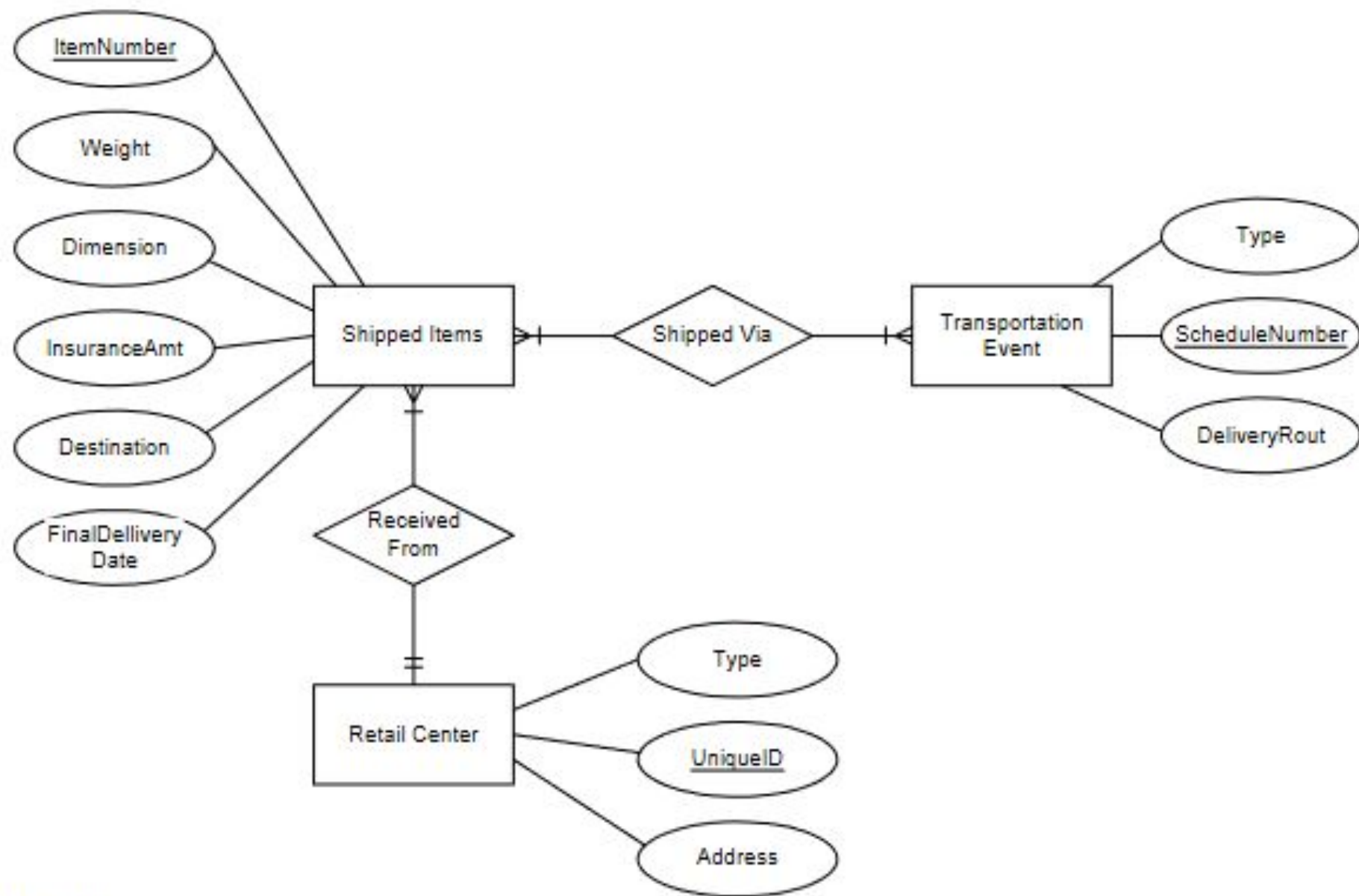
---

S.NO	Strong Entity	Weak Entity
1.	Strong entity always has primary key.	While weak entity has partial discriminator key.
2.	Strong entity is not dependent of any other entity.	Weak entity is depend on strong entity.
3.	Strong entity is represented by single rectangle.	Weak entity is represented by double rectangle.
4.	Two strong entity's relationship is represented by single diamond.	While the relation between one strong and one weak entity is represented by double diamond.
5.	Strong entity have either total participation or not.	While weak entity always has total participation.

# EXAMPLE

UPS prides itself on having up-to-date information on the processing and current location of each shipped item. To do this, UPS relies on a company-wide information system. Shipped items are the heart of the UPS product tracking information system. Shipped items can be characterized by item number (unique), weight, dimensions, insurance amount, destination, and final delivery date. Shipped items are received into the UPS system at a single retail center. Retail centers are characterized by their type, uniqueID, and address. Shipped items make their way to their destination via one or more standard UPS transportation events (i.e., flights, truck deliveries). These transportation events are characterized by a unique scheduleNumber, a type (e.g, flight, truck), and a deliveryRoute. Please create an Entity Relationship diagram that captures this information about the UPS system. Be certain to indicate identifiers and cardinality constraints.





# EXAMPLE

## Requirements of the Company

- The company is organized into DEPARTMENTS. Each department has a name, number and an employee who *manages* the department. We keep track of the start date of the department manager.
- Each department *controls* a number of PROJECTs. Each project has a name, number and is located at a single location.
- We store each EMPLOYEE's social security number, address, salary, sex, and birthdate. Each employee *works for* one department but may *work on* several projects. We keep track of the number of hours per week that an employee currently works on each project. We also keep track of the *direct supervisor* of each employee.
- Each employee may *have* a number of DEPENDENTs. For each dependent, we keep track of their name, sex, birthdate, and relationship to employee.



