Logical Database using Entity Relationship Diagram (ERD)

Mohamed Elshaikh

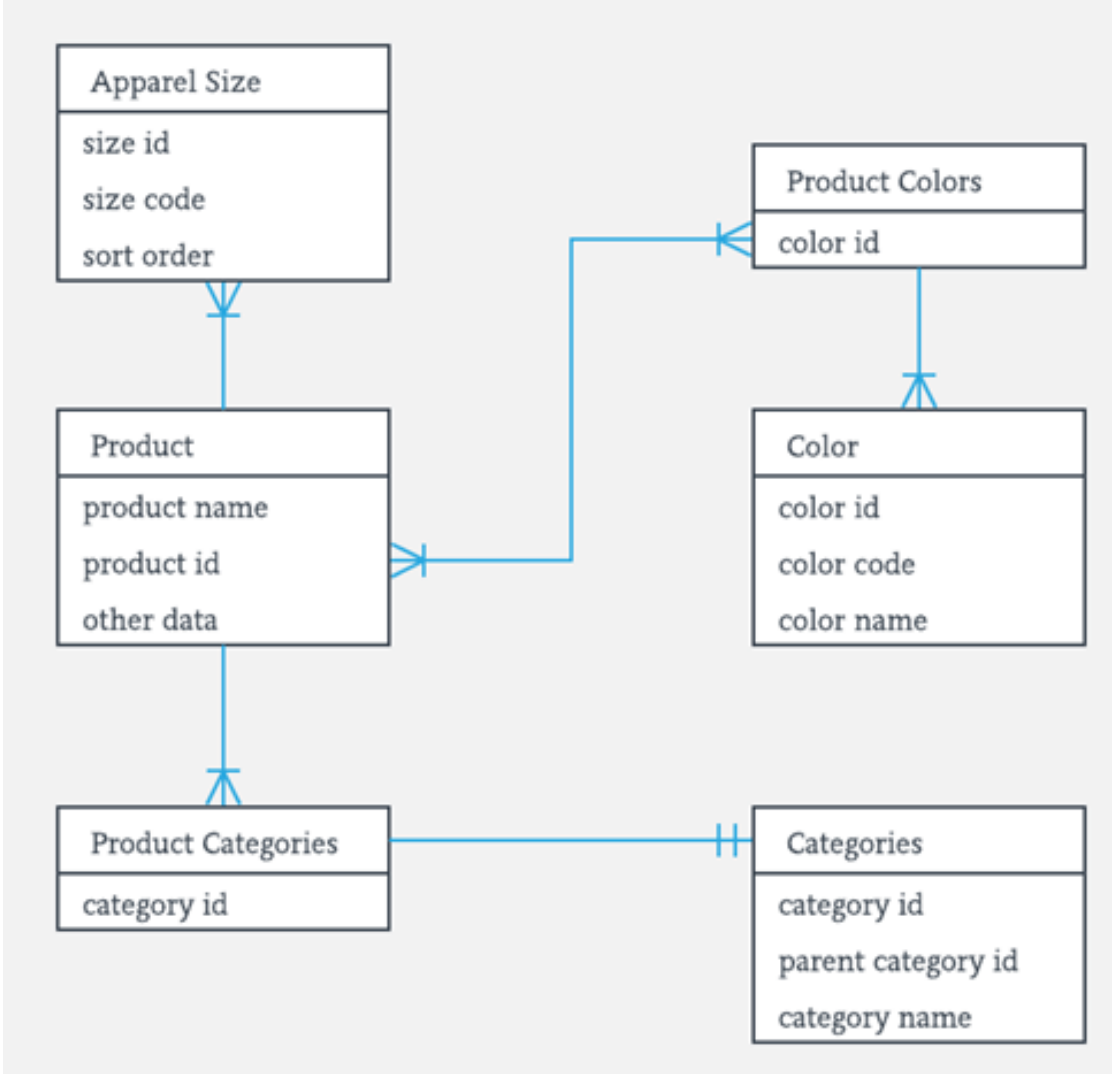
Chapter 06

Objectives

- **How to use Entity–Relationship (ER) modeling in database design.**
- **Basic concepts associated with ER model.**
- **Diagrammatic technique for displaying ER model using Unified Modeling Language (UML).**
- **How to identify and resolve problems with ER models called connection traps.**
- **How to build an ER model from a requirements specification.**

ER diagram

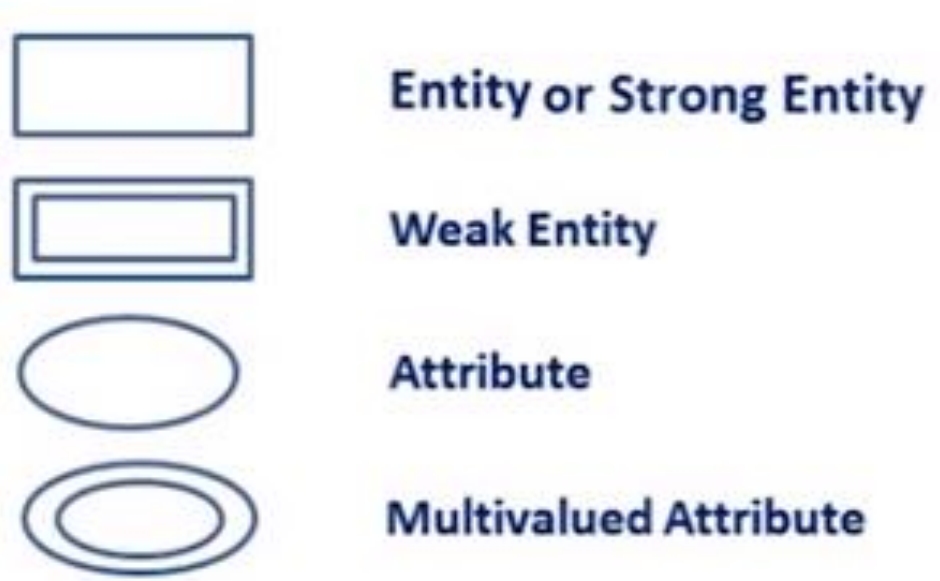
- ER Diagram stands for Entity Relationship Diagram
- Displays the relationship of entity sets stored in a database.
  - rectangles to represent entities
  - ovals to define attributes
  - diamond shapes to represent relationships
- The purpose of ER Diagram is to represent the entity framework infrastructure.

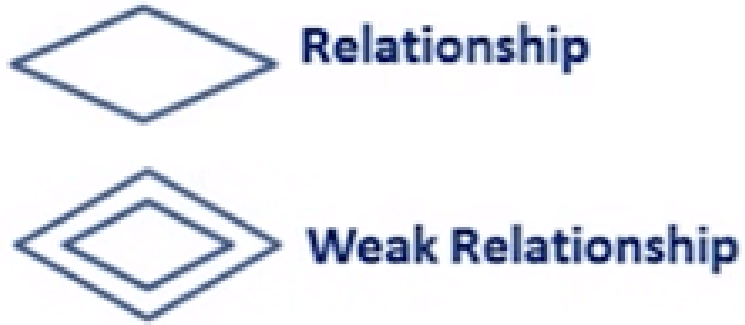


ER Diagrams Symbols & Notations

- Rectangles: Entity Relationship Diagram

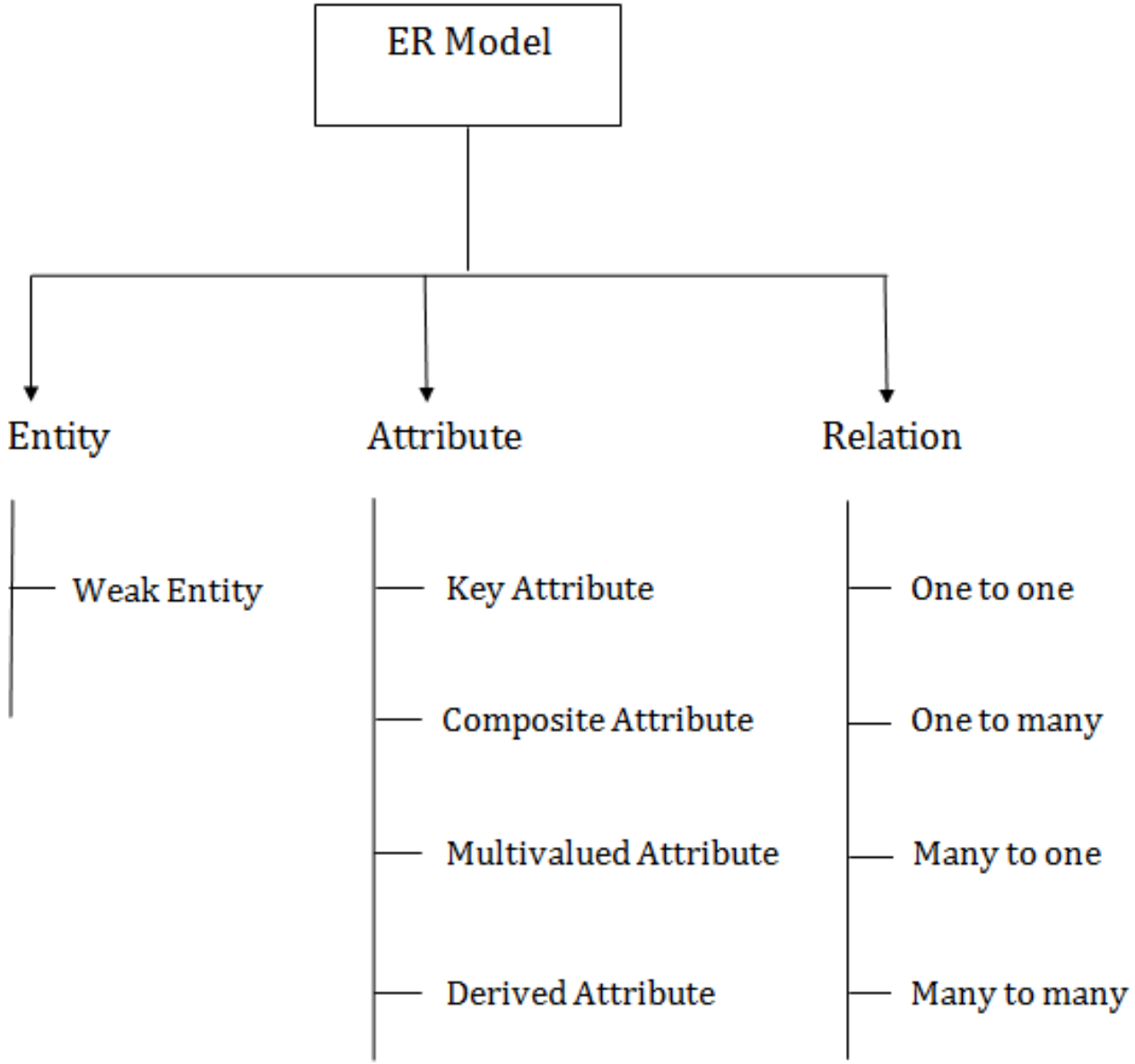
  symbol represents entity types
  - Ellipses : Symbol represent attributes
  - Diamonds: This symbol represents relationship types
  - Lines: It links attributes to entity types, and entity types with other relationship types
- Primary key: attributes are underlined
- Double Ellipses: Represent multi-valued attributes



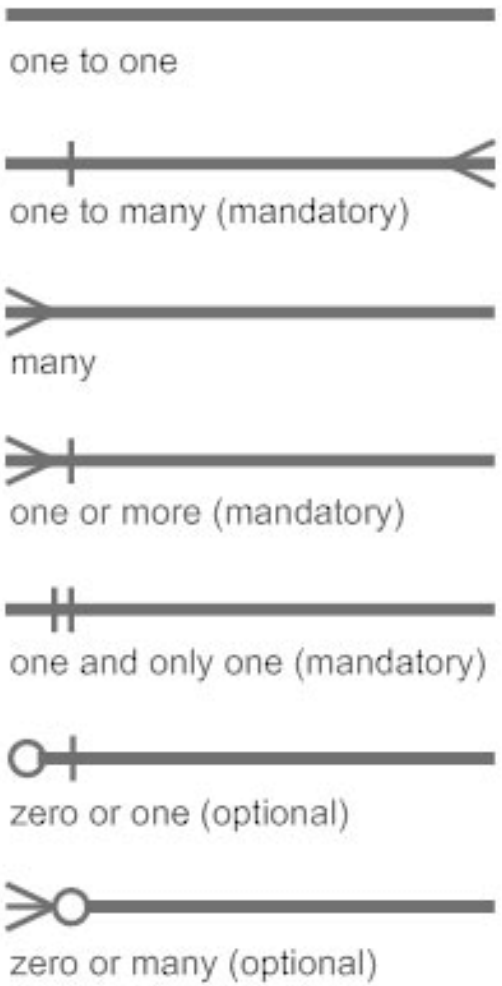


Concepts of the ER Model

- Entity types
- Relationship types
- Attributes



Notation of ER diagram



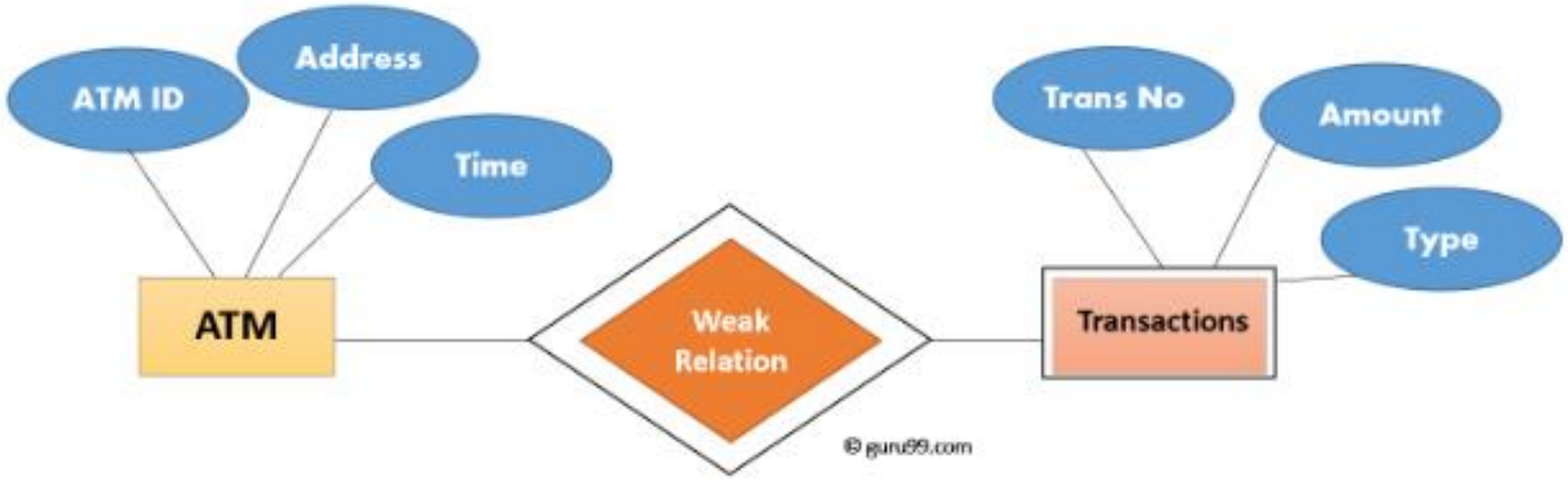
Entity Type

- A real-world thing either living or non-living that is easily recognizable and nonrecognizable.
- Anything in the enterprise that is to be represented in a database.

- A university may have some departments. All these departments employ various lecturers and offer several programs.
- Some courses make up each program. Students register in a particular program and enroll in various courses. A lecturer from the specific department takes each course, and each lecturer teaches a various group of students.

Weak Entities

- A weak entity is a type of entity which doesn't have its key attribute. It can be identified uniquely by considering the primary key of another entity. For that, weak entity sets need to have participation.



- "Trans No" is a discriminator within a group of transactions in an ATM.

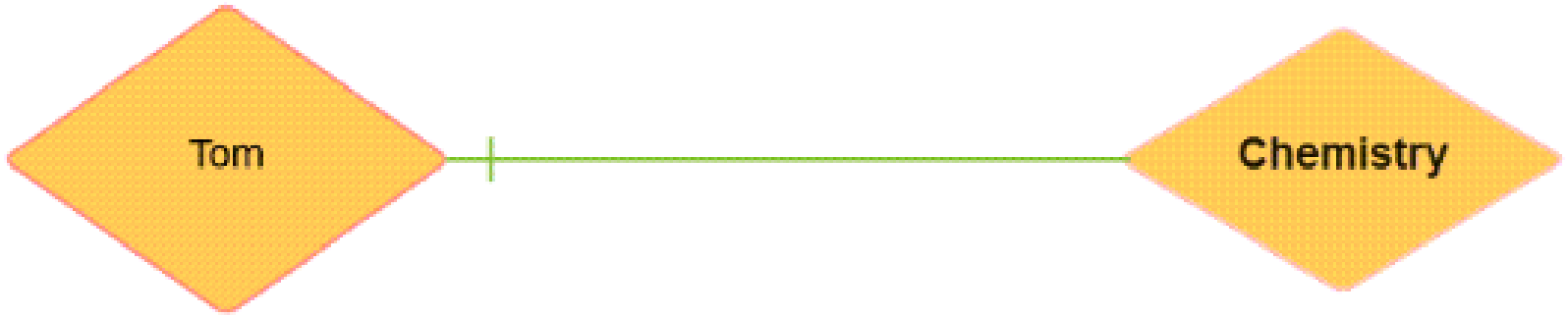
Weak and Strong Entities

STRONG ENTITIES	WEAK ENTITIES
Strong entity set always has a primary key.	It does not have enough attributes to build a primary key.
It is represented by a rectangle symbol.	It is represented by a double rectangle symbol.
It contains a Primary key represented by the underline symbol.	It contains a Partial Key which is represented by a dashed underline symbol.
The member of a strong entity set is called as dominant entity set.	The member of a weak entity set called as a subordinate entity set.
Primary Key is one of its attributes which helps to identify its member.	In a weak entity set, it is a combination of primary key and partial key of the strong entity set.
In the ER diagram the relationship between two strong entity set shown by using a diamond symbol.	The relationship between one strong and a weak entity set shown by using the double diamond symbol.
The connecting line of the strong entity set with the relationship is single.	The line connecting the weak entity set for identifying relationship is double.

Relationship Types

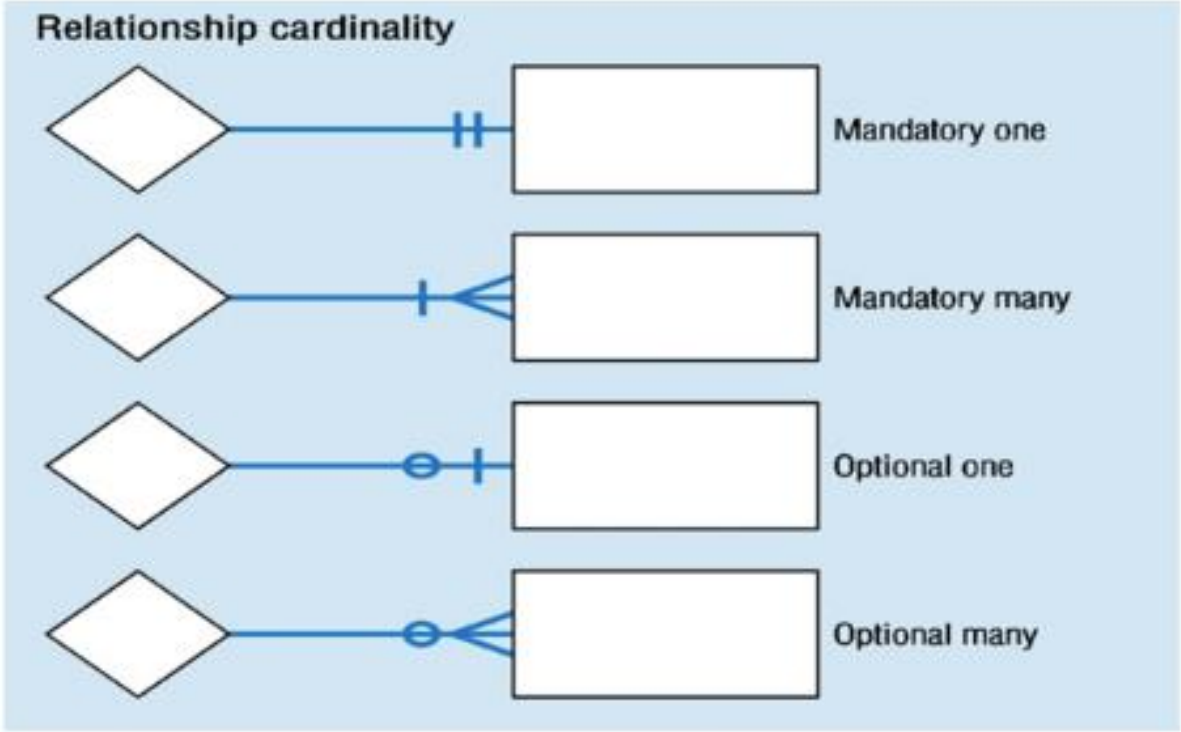
- Relationship
  - An association among two or more entities. E.g., Tom works in the Chemistry department.
- Relationship type
  - Set of meaningful associations among entity types.
- Relationship occurrence
  - Uniquely identifiable association, which includes one occurrence from each participating entity type.

- Example
  - You are attending this lecture
  - I am giving the lecture
  - Just loke entities, we can classify relationships according to relationship-types:
  - A student attends a lecture
  - A lecturer is giving a lecture.



Cardinality

- Defines the numerical attributes of the relationship between two entities or entity sets.
  - One-to-One Relationships
  - One-to-Many Relationships
  - May to One Relationships
  - Many-to-Many Relationships



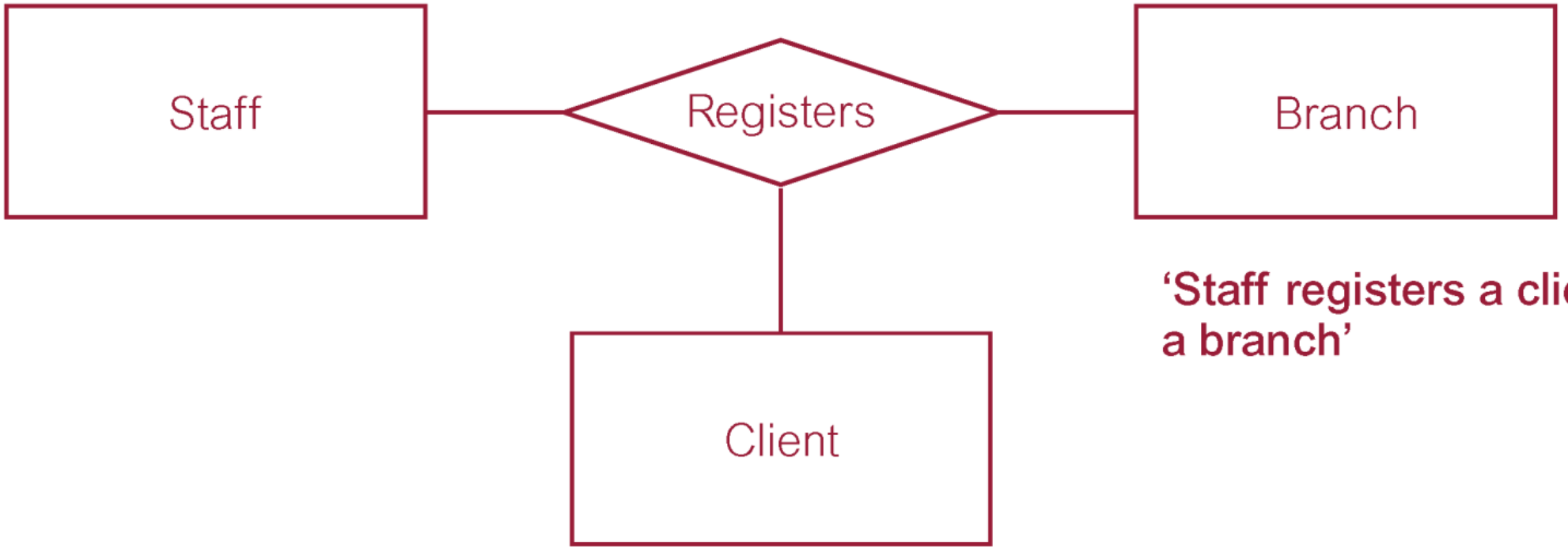
Relationship Types

- Degree of a Relationship
  - Number of participating entities in relationship.
- Relationship of degree :
  - One is recursive
  - two is binary
  - three is ternary
  - four is quaternary.

Binary relationship called POwns

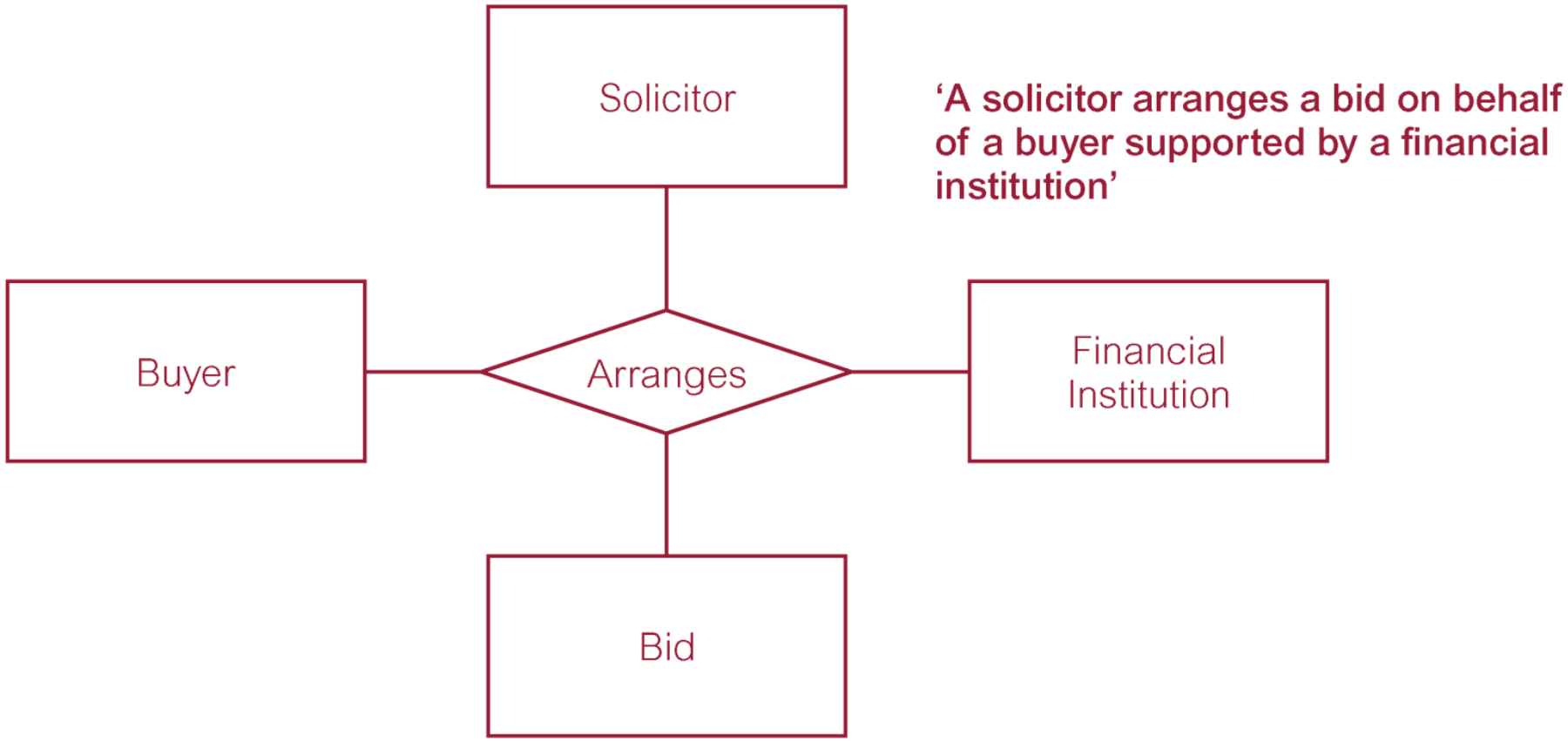
‘Private owner owns property for rent’

Ternary relationship called Registers



'Staff registers a client at a branch'

Quaternary relationship called Arranges

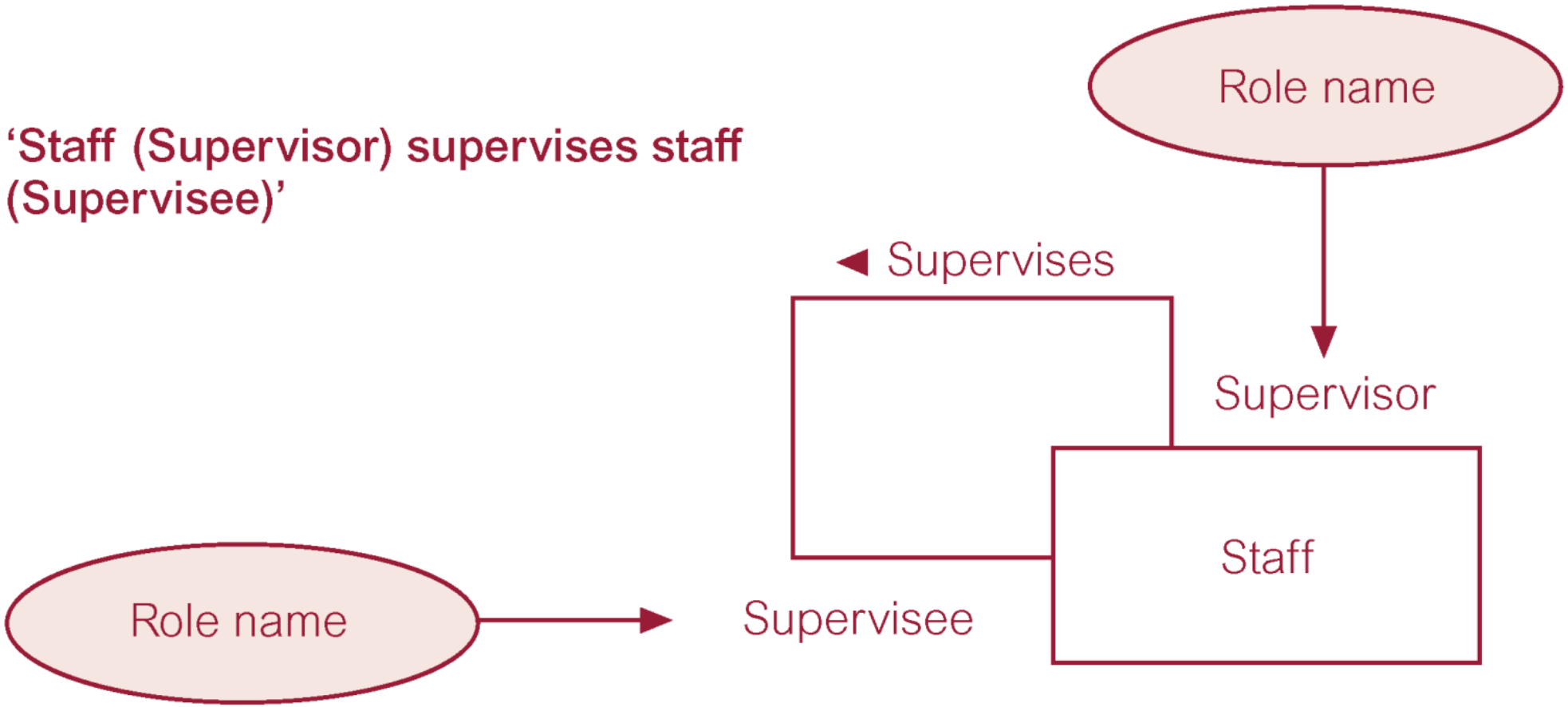


'A solicitor arranges a bid on behalf of a buyer supported by a financial institution'

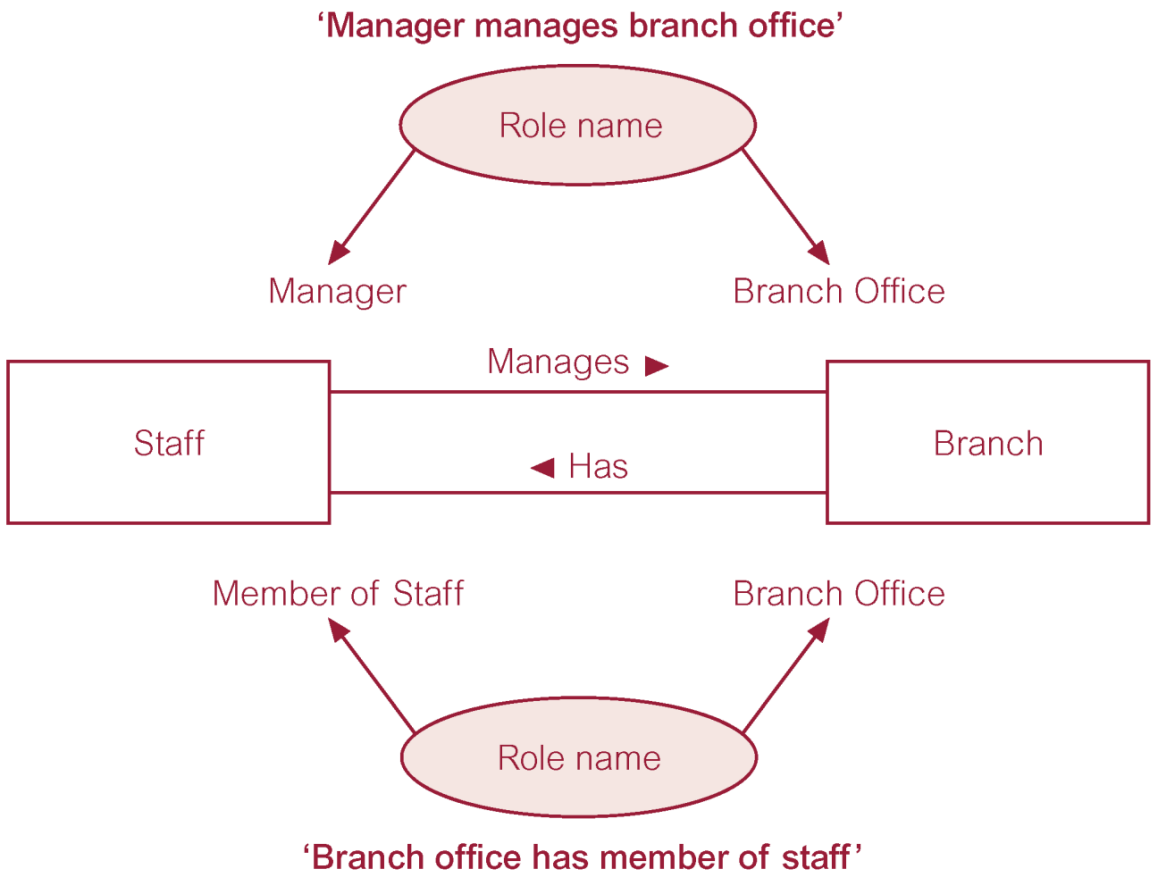
Relationship Types

- Recursive Relationship
  - Relationship type where same entity type participates more than once in different roles.
- Relationships may be given role names to indicate purpose that each participating entity type plays in a relationship.

Recursive relationship called Supervises with role names

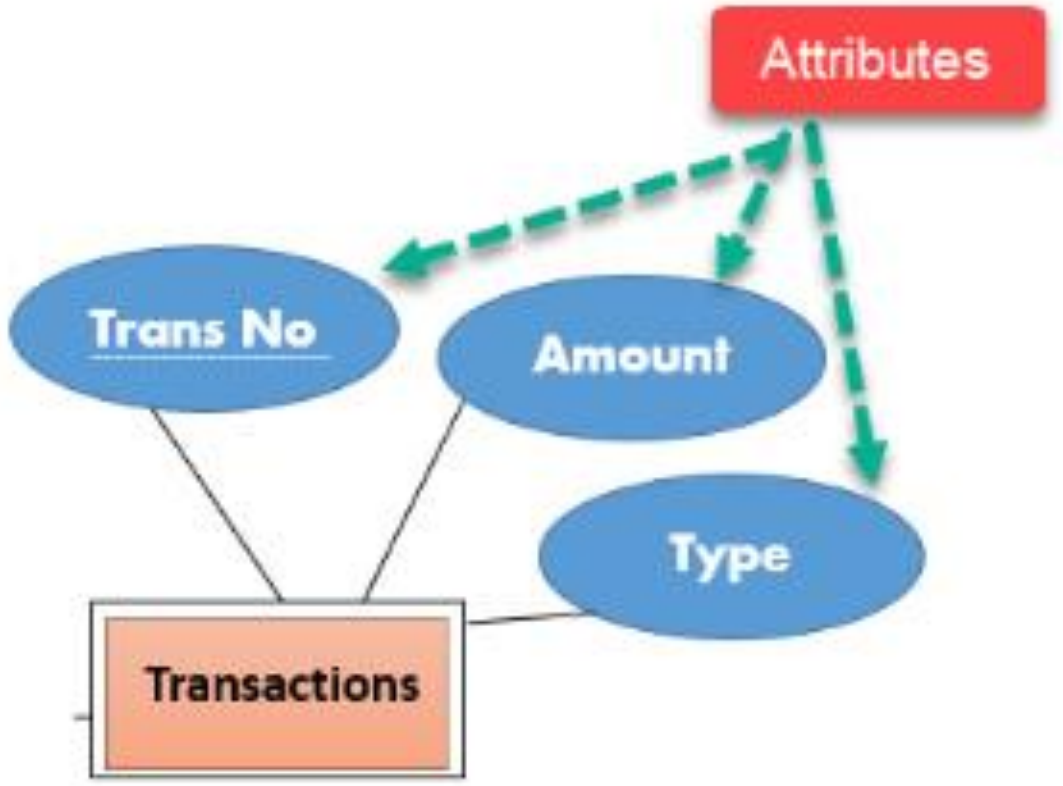


Entities associated through two distinct relationships with role names



Attributes

- Attribute
  - Property of an entity or a relationship type.
- Attribute Domain
  - Set of allowable values for one or more attributes.
- An attribute in ER Diagram examples, is represented by an Ellipse



Attributes

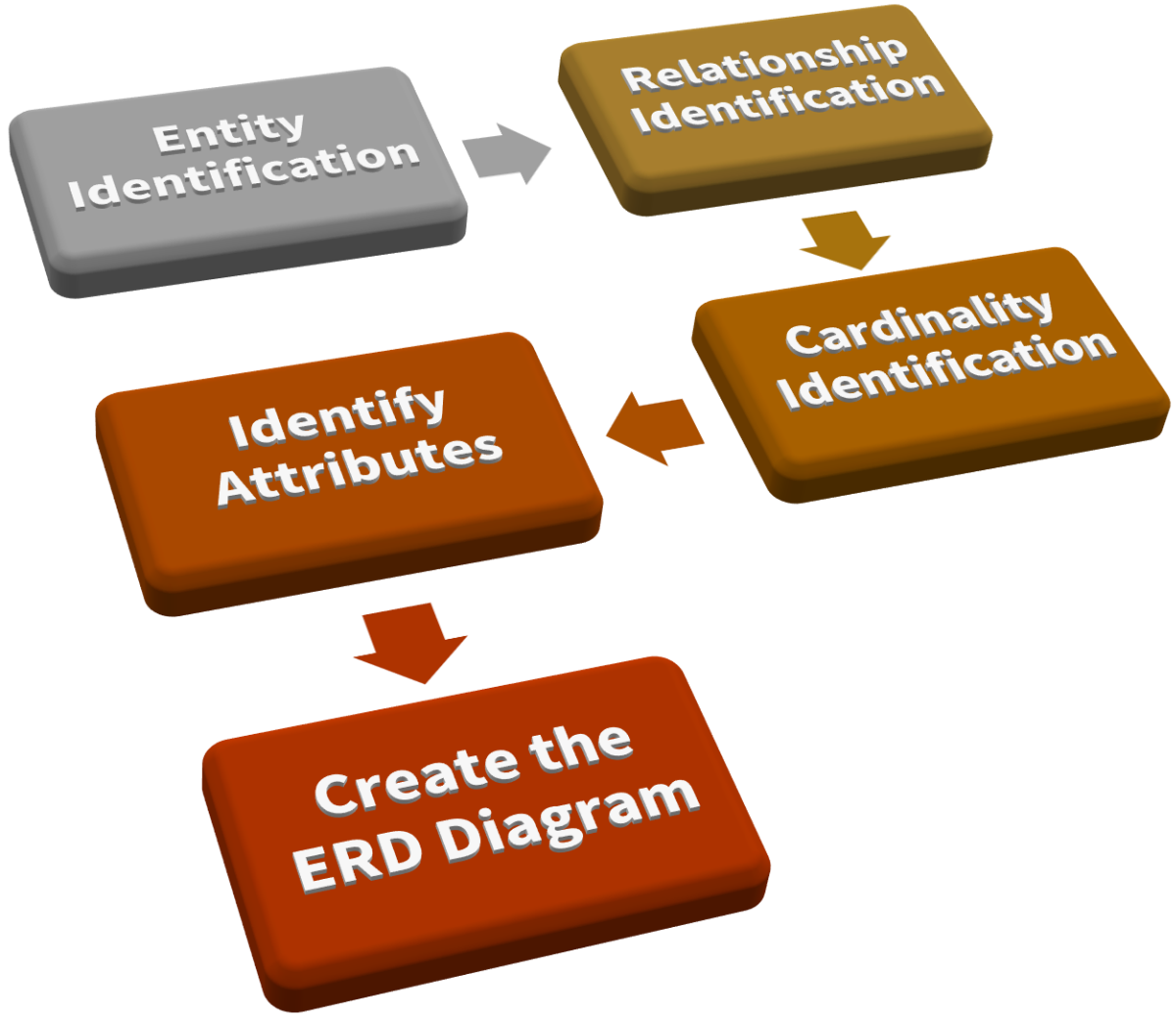
Types of Attributes	Description
Simple attribute	Simple attributes can’t be divided any further. For example, a student’s contact number. It is also called an atomic value.
Composite attribute	It is possible to break down composite attribute. For example, a student’s full name may be further divided into first name, second name, and last name.
Derived attribute	This type of attribute does not include in the physical database. However, their values are derived from other attributes present in the database. For example, age should not be stored directly. Instead, it should be derived from the DOB of that employee.
Multivalued attribute	Multivalued attributes can have more than one values. For example, a student can have more than one mobile number, email address, etc.

Keys

- Primary key: identify one and only one instance of an entity uniquely
- Candidate key:A candidate key is an attribute or set of attributes that can uniquely identify a tuple.
- Super Key: Super key is an attribute set that can uniquely identify a tuple. A super key is a superset of a candidate key.
- Foreign key: point to the primary key of another table.
- Alternate key: one or more attributes or a combination of attributes that uniquely identify each tuple in a relation.
- Composite key: Whenever a primary key consists of more than one attribute, it is known as a composite key.
- Artificial key: The key created using arbitrarily assigned data are known as artificial keys.

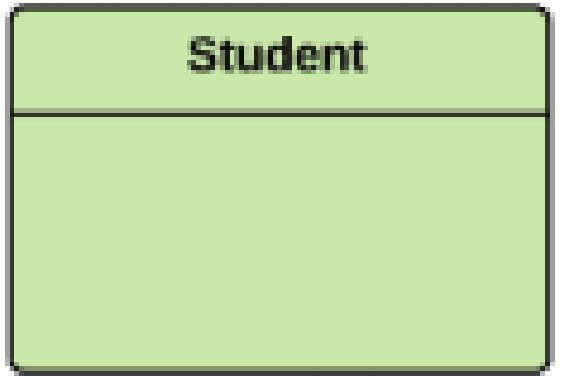
ERD in Steps

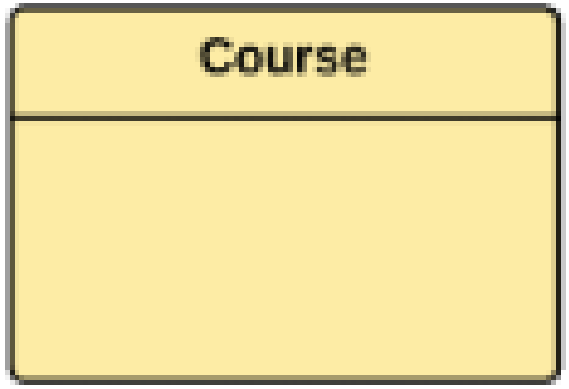
In a university, a Student enrolls in Courses. A student must be assigned to at least one or more Courses. Each course is taught by a single Professor. To maintain instruction quality, a Professor can deliver only one course

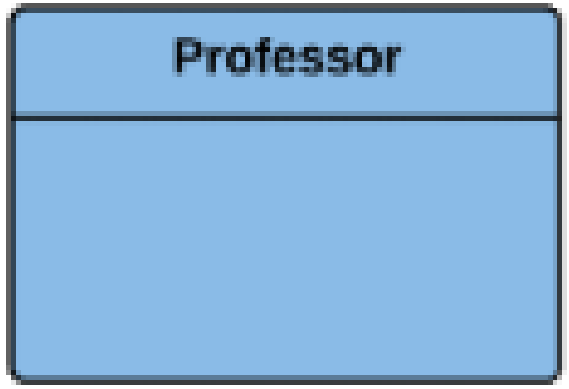


Step1: Entity Identification

- We have three entities
  - Student
  - Course
  - Professor







Step2: Relationship Identification

- Two relationships
  - The student is **assigned** a course
  - Professor **delivers** a course

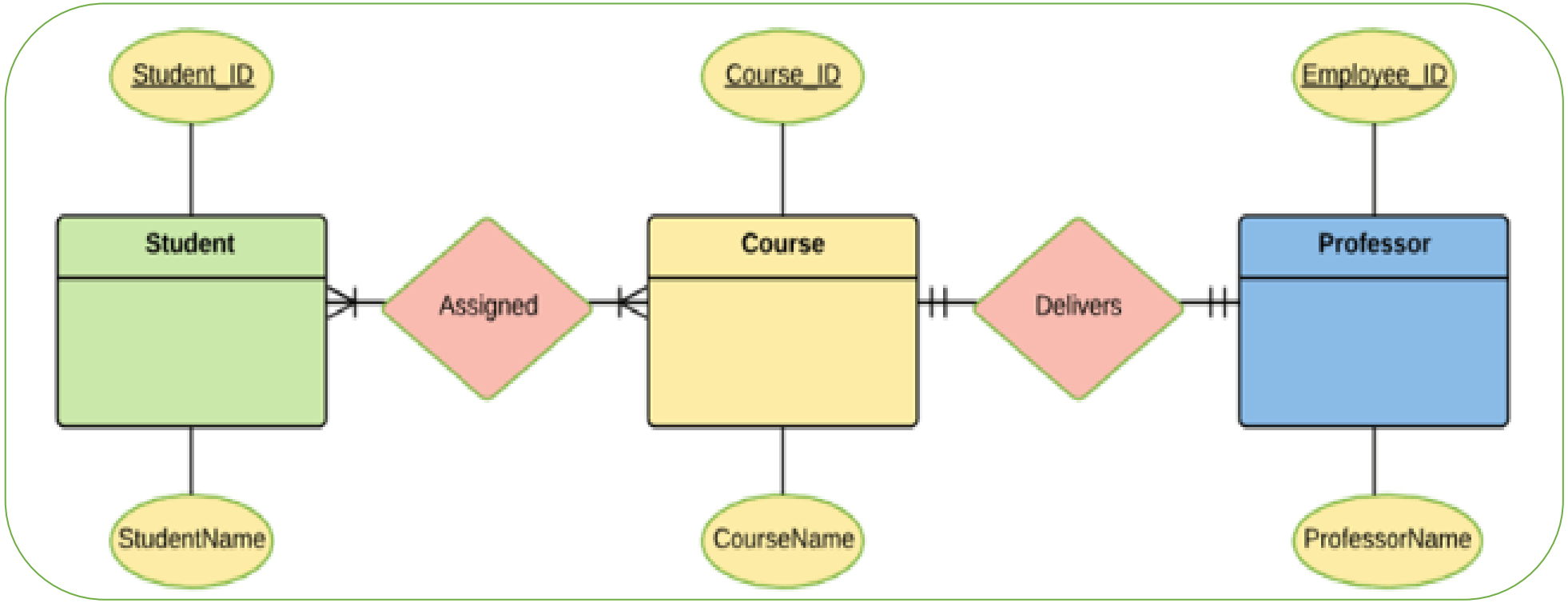
Step3: Cardinality Identification

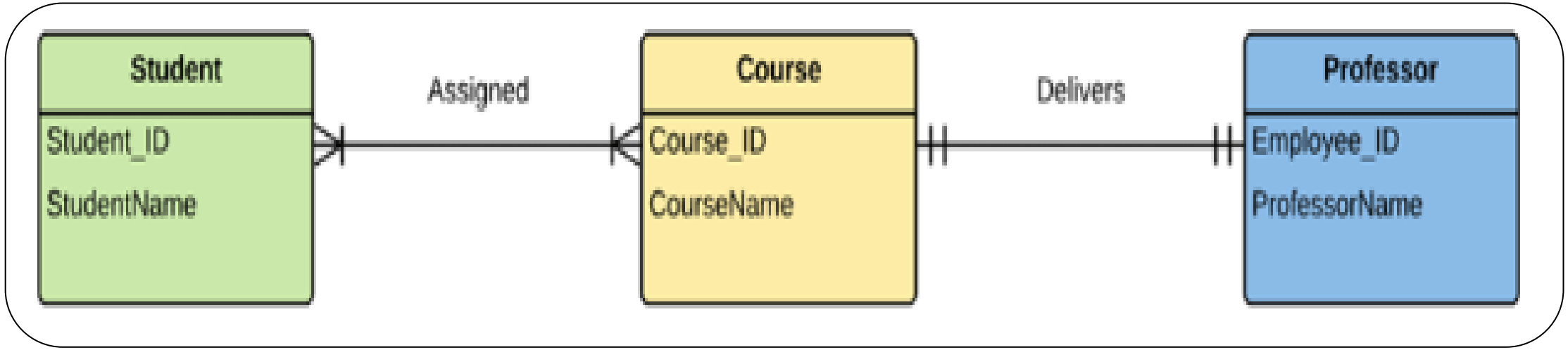
- For this problem statement we know that,
  - A student can be assigned **multiple** courses
  - A Professor can deliver only **one** course

Step4: Identify Attributes

- Read the documents or any data maintained by the organization
  - Files
  - Reports
  - Form
- Interview
- Attribute is paired with one entity or use modifier to make unique.
- Identify primary keys.

Entity	Primary Key	Attribute
Student	**Student_ID**	**StudentName**
Professor	**Employee_ID**	**ProfessorName**
Course	**Course_ID**	**CourseName**


Student_ID
Course_ID
Employee_ID
Student
Assigned
Course
Delivers
Professor
StudentName
CourseName
ProfessorName


Student
Student_ID
StudentName
Assigned
Course
Course_ID
CourseName
Delivers
Professor
Employee_ID
ProfessorName

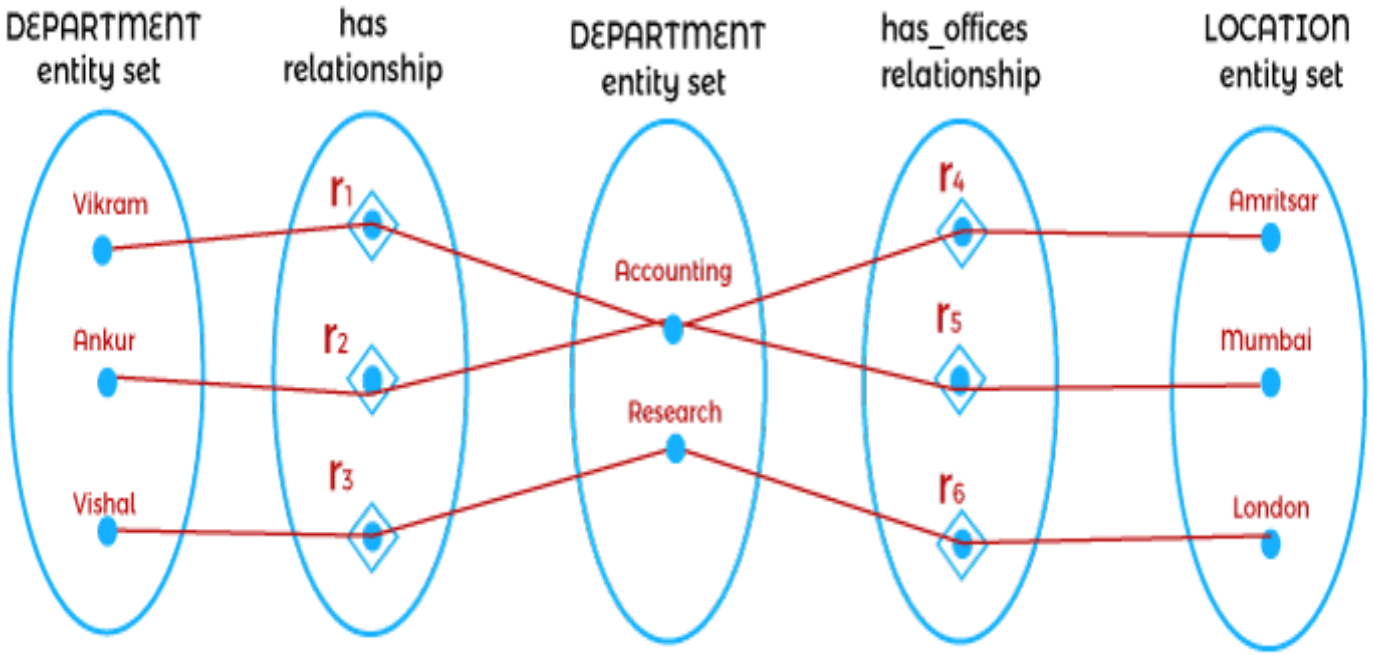
Problems with ER Models

- Problems may arise when designing a conceptual data model called connection traps.
- Often due to a mis-interpretation of the meaning of certain relationships.
- Two main types of connection traps are called fan traps and chasm traps.

Problems with ER Models

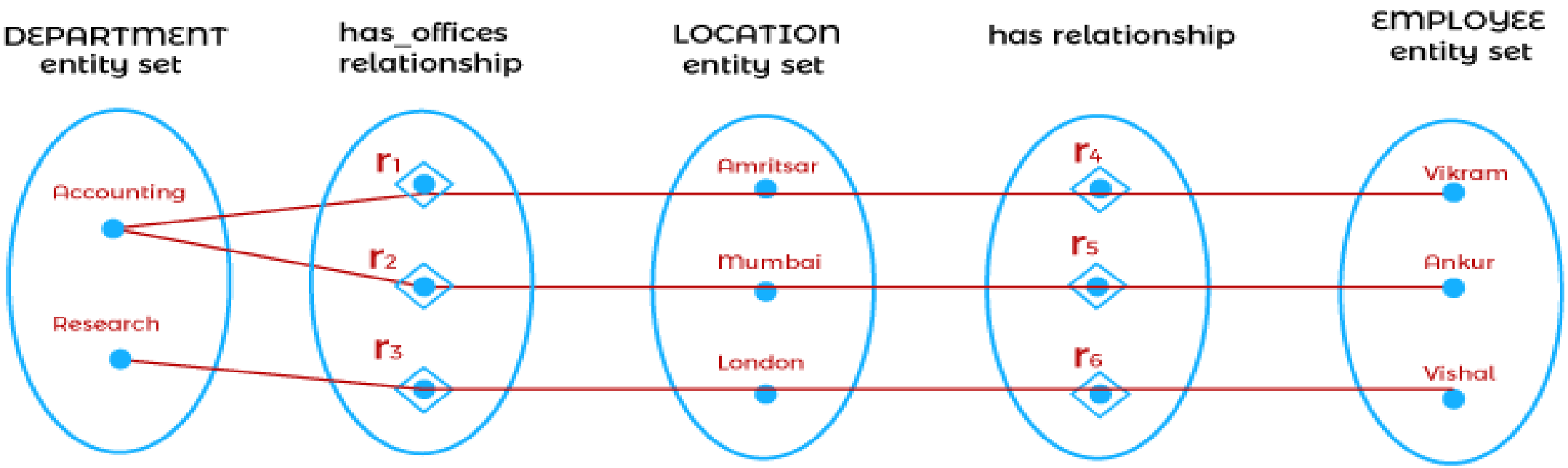
- Fan Trap
  - Where a model represents a relationship between entity types, but pathway between certain entity occurrences is ambiguous.
- Chasm Trap
  - Where a model suggests the existence of a relationship between entity types, but pathway does not exist between certain entity occurrences.

Fan Trap



- One-to-many (1:N):
  - Dep-to-Emp
  - Dep-to-Loc
- Converge to single entity [Bank]

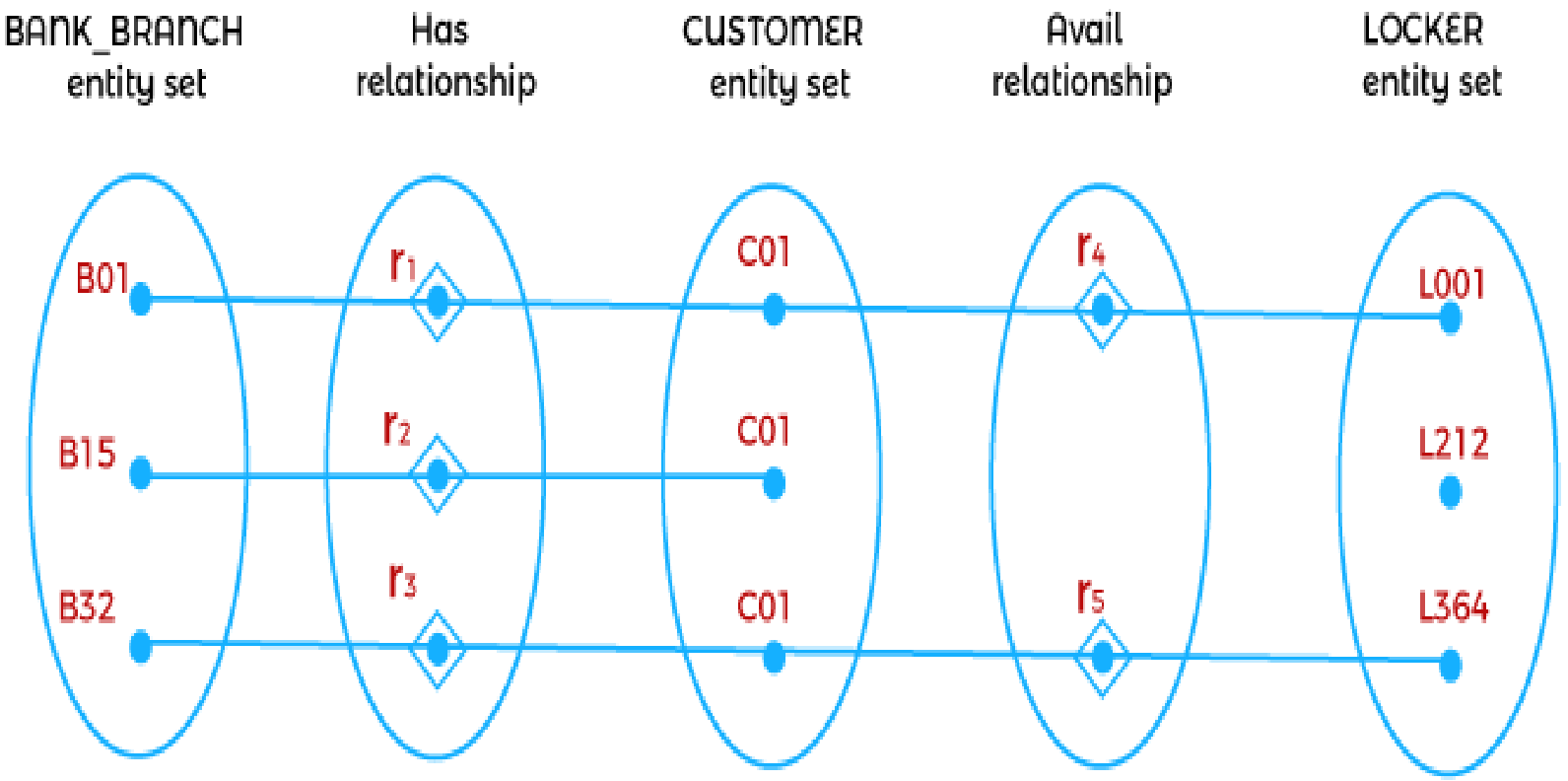
- A problem arises if we want to determine which employee works at which location. For example. If we want to know in which city does Vikram works.



- We can only determine that the employee Vikram works in either Amritsar or Mumbai from the current structure.
- because of mutual exclusion of Employee and Location entity occurrences with department entity.
- We can resolve this fan trap by reconstructing the ER model so that they now represent the correct association between these entities.

Chasm Trap

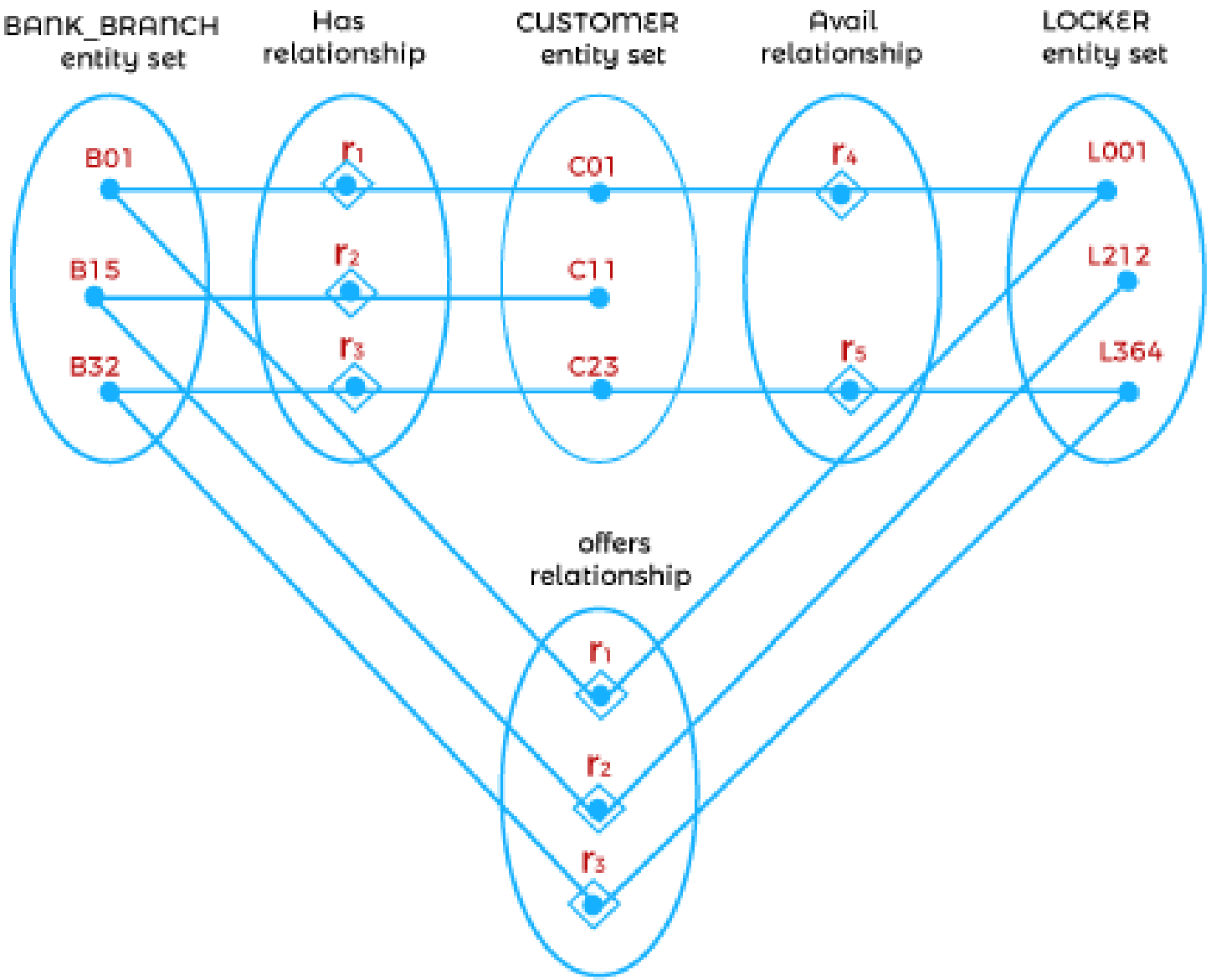
- Occurs when a pathway does not exist in all the related entities.

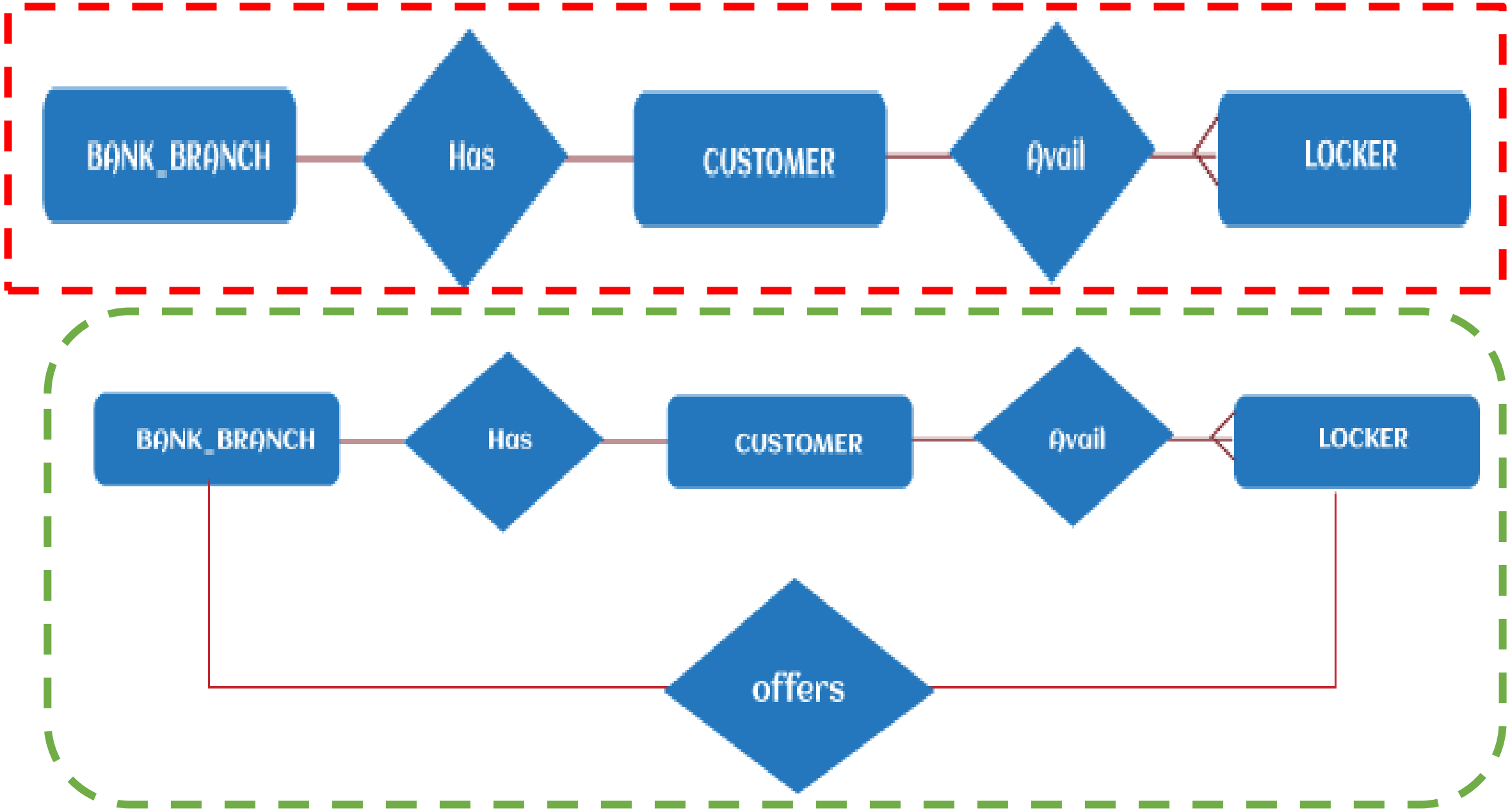


- Suppose we want to know which bank's branch locker L212 is available.
- locker L212 has not yet been allocated.

Chasm Solution

- Identify the missing relationship
- Reconstruct ER.




BANK_BRANCH
Has
CUSTOMER
Avail
LOCKER
BANK_BRANCH
Has
CUSTOMER
Avail
LOCKER
offers

Questions