System Analysis and Design

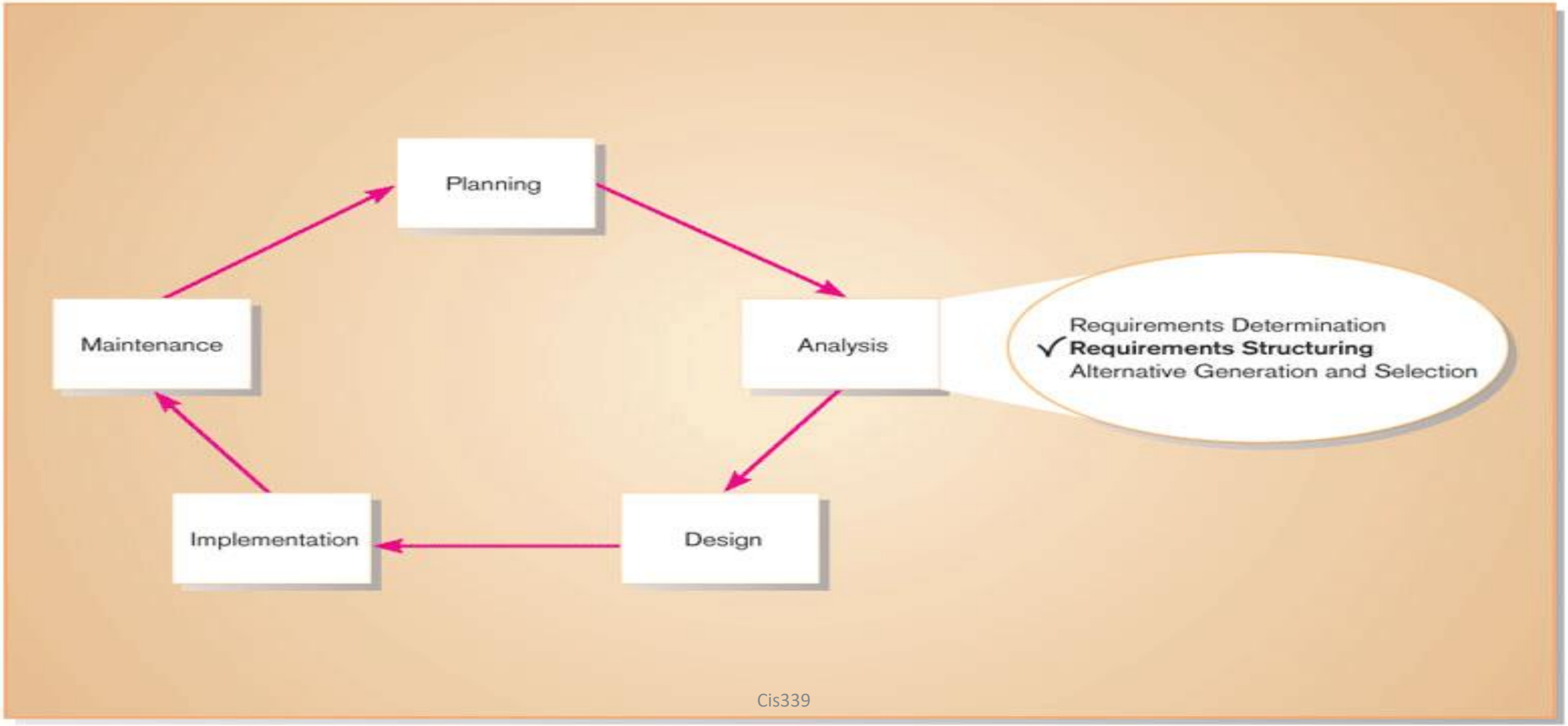
Chapter 5
System Analysis

5.2
Structuring System Requirements

5.2.2
Conceptual Data Modeling

Conceptual Data Modeling

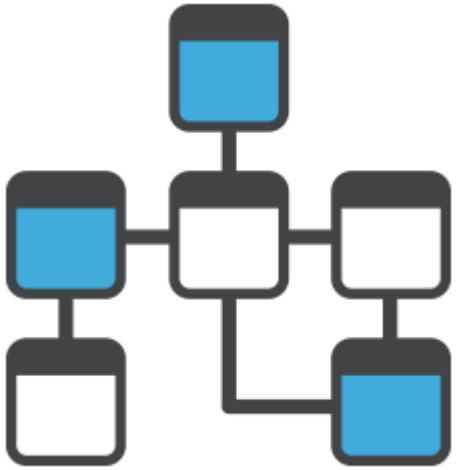
Figure 9-1 Systems development life cycle with analysis phase highlighted



Conceptual Data Model

- a detailed model that captures the overall structure of organizational data while being independent of any database management system or other implementation consideration

- typically done in parallel with other requirements analysis and structuring steps during system analysis



Process of Conceptual Data Modeling

1. First step is to develop a data model for the system being replaced (if the system exists)
2. Next, a new conceptual data model is built that includes all the requirements of the new system- evolve through various iteration.
3. In the design stage, the conceptual data model is translated into a physical design, data storage architecture is been selected, then file and DB are defined.

Deliverables and Outcome

- Primary deliverable is an entity-relationship (E-R) diagram or class diagram.
- As many as 4 E-R or class diagrams are produced and analyzed:
  - E-R diagram that covers data needed in the project's application.
  - E-R diagram for the application being replaced
  - E-R diagram for the whole database from which the new application's data are extracted.
  - E-R diagram for the whole database from which data for the application system being replaced is drawn.

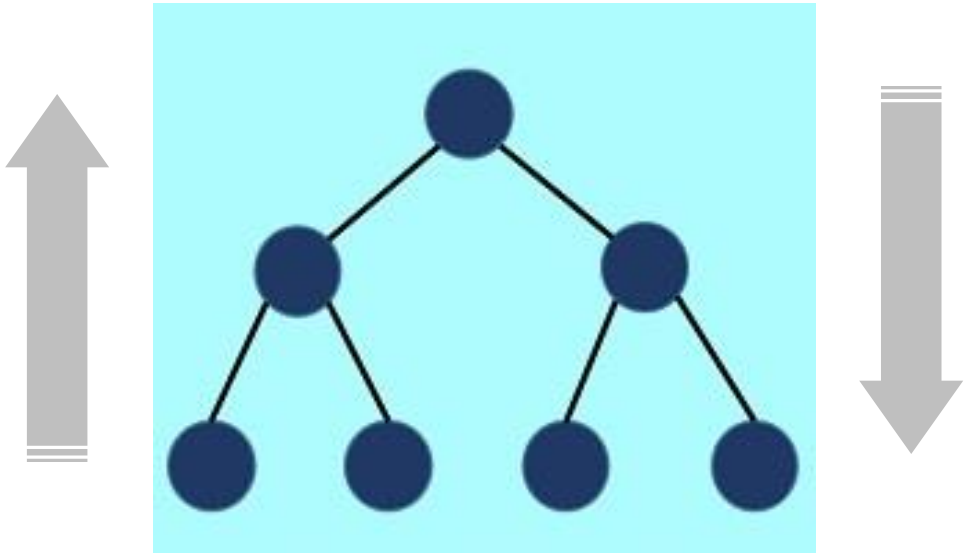
9.8

Deliverables and Outcome

- Second deliverable is a set of entries about data objects to be stored in repository or project dictionary.
  - Repository links data, process, and logic models of an information system.
  - Data elements included in the DFD must appear in the data model and vice versa.
  - Each data store in a process model must relate to business objects represented in the data model.

Gathering Information for Conceptual Data Modeling

- Two perspectives on data modeling:
  - ***Top-down approach***
    - for a data model is derived from an intimate understanding of the business.
  - ***Bottom-up approach***
    - for a data model is derived by reviewing specifications and business documents



Gathering Information for Conceptual Data Modeling

Requirements Determination Questions for Data Modeling:

1. What are subjects/objects of the business?

- **Data entities and descriptions.**

2. What unique characteristics distinguish between subjects/objects of the same type?

- **Primary key**

3. What characteristics describe each subject/object?

- **Attributes and secondary keys.**

4. How do you use the data?

- **Security controls and user access privileges.**

Gathering Information for Conceptual Data Modeling

Requirements Determination Questions for Data Modeling:

5. Over what period of time are you interested in the data?

- Cardinality and time dimensions.

6. Are all instances of each object the same?

- Supertypes, subtypes, and aggregations.

7. What events occur that imply associations between objects?

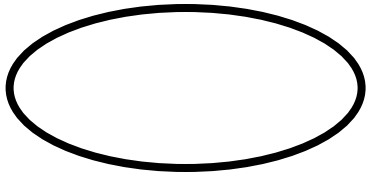
- Relationships and cardinalities.

8. Are there special circumstances that affect the way events are handled?

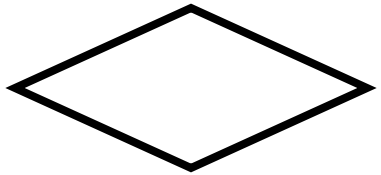
- Integrity rules, cardinalities, time dimensions.

Basic Notations for ER Diagram

Entity

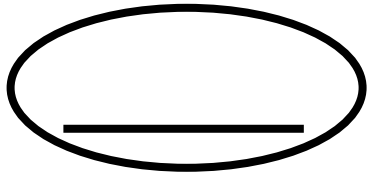


Attribute

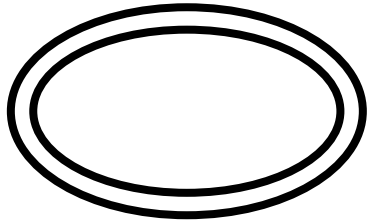


Relationship

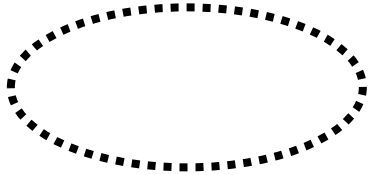
Line



Key Attribute

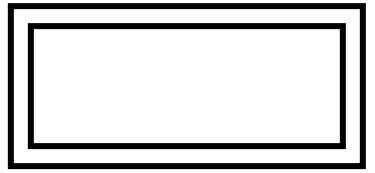


Multivalued Attribute

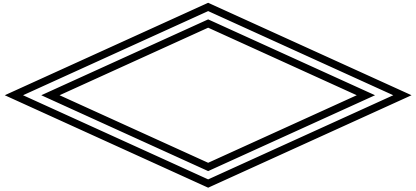


Derived Attribute

Composite Attribute

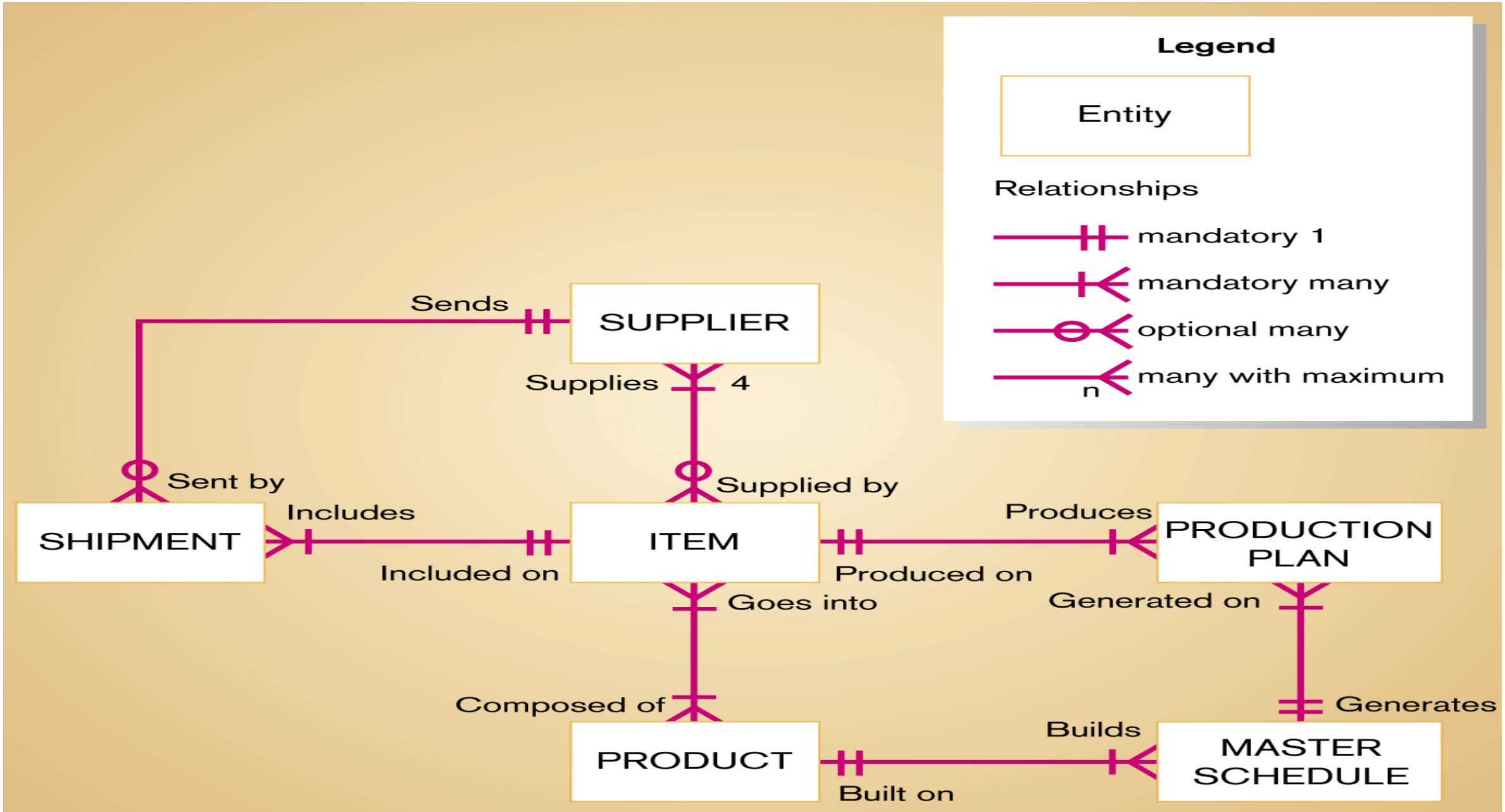


Weak Entity

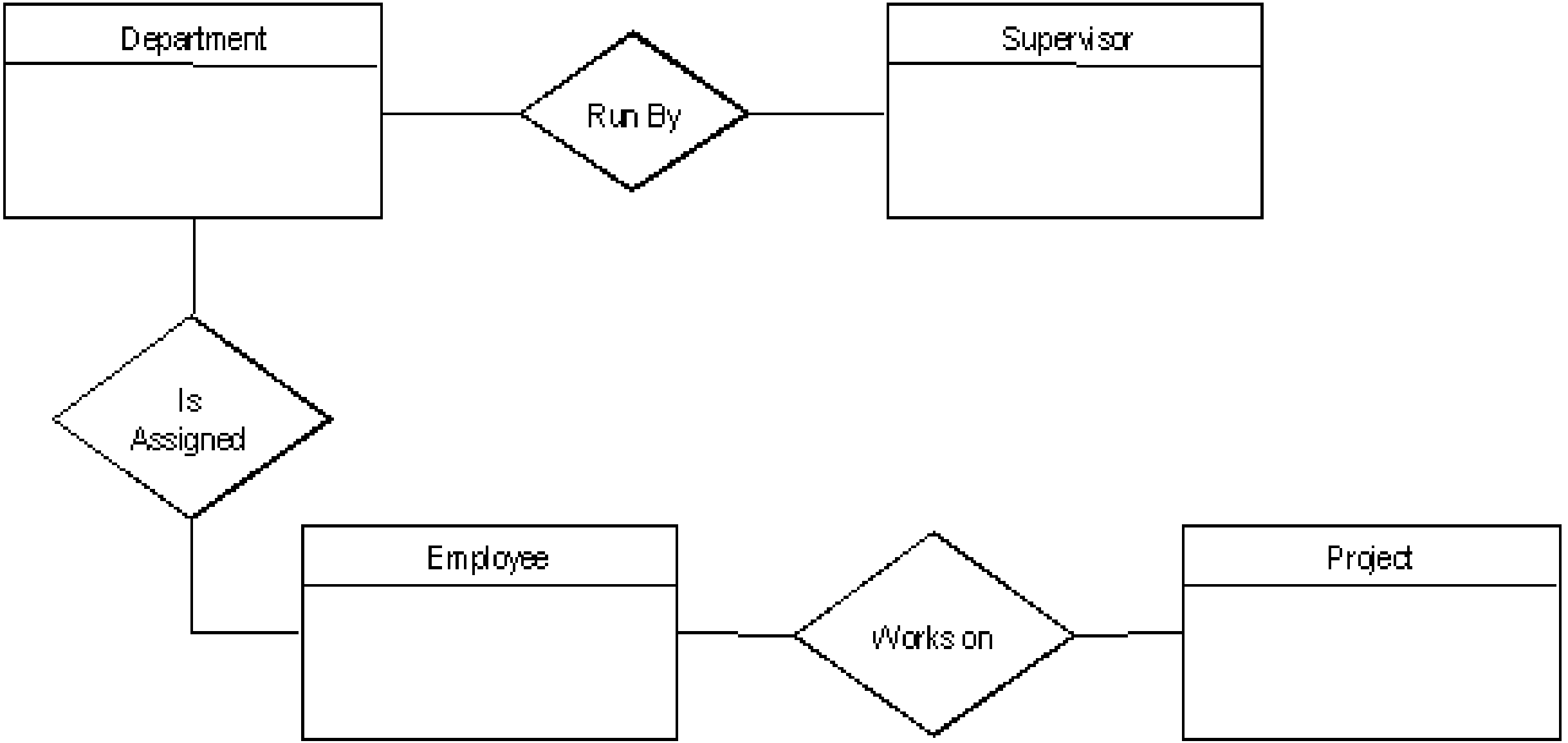


Weak Relationship

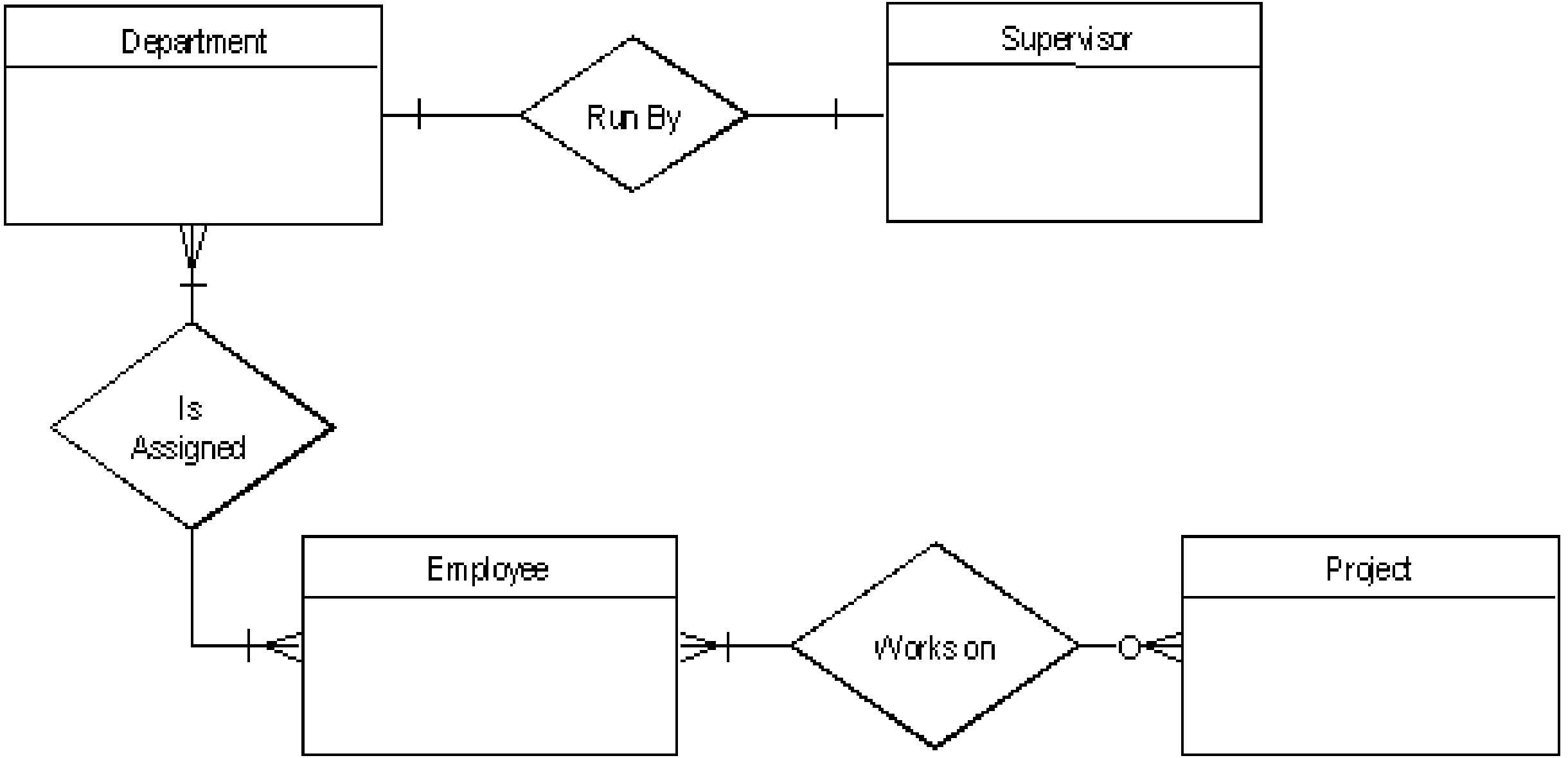
Sample conceptual data model diagram



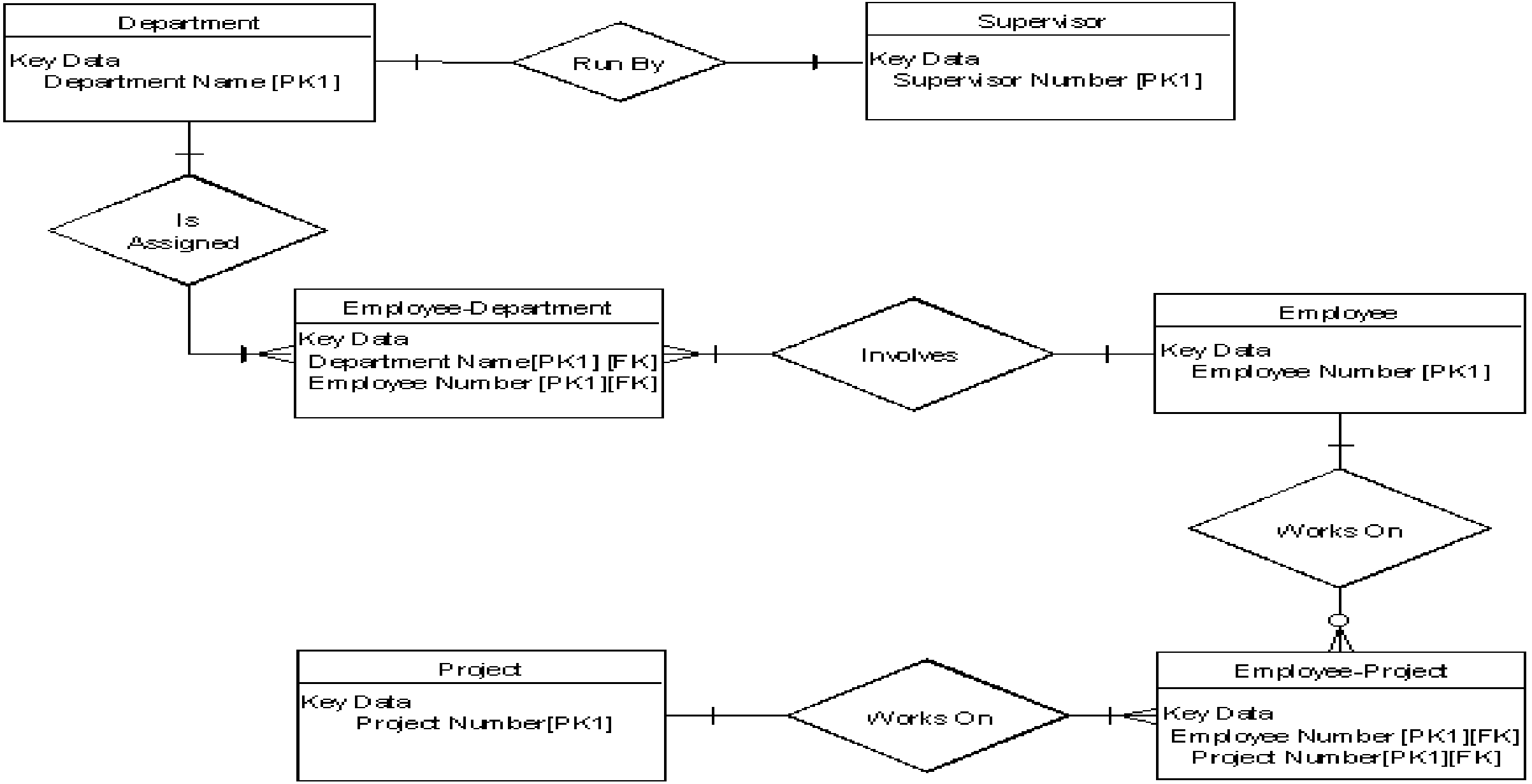
Rough ERD


Department
Run By
Supervisor
Is Assigned
Employee
Works on
Project

Fill cardinality in ERD



Fill attribute in ERD



Introduction to Entity-Relationship (E-R) Modeling

Entity-Relationship data model (**E-R model**):

- a detailed, logical representation of the entities, associations and data elements for an organization or business area.

Entity-relationship diagram (**E-R diagram**):

- a graphical representation of an E-R model.
- The E-R model is expressed in terms of:
  - **Data entities** in the business environment.
  - **Relationships** or associations among those entities.
  - **Attributes** or properties of both the entities and their relationships.

Entity-Relationship (E-R) Modeling

Key Terms

- **Entity**
  - A person, place, object, event or concept in the user environment about which the organization wishes to maintain data (person, place, object, event)
  - Represented by a **rectangle** in E-R diagrams
- **Entity Type**
  - A collection of entities that share common properties or characteristics- called entity class (employee, course, account)
- **Entity Instance**
  - A single occurrence of an entity is called entity instance
  - Example: **Ramesh** can be an instance of **Student** entity

Entity-Relationship (E-R) Modeling

Key Terms

- **Attribute**
  - a named property or characteristic of an entity that is of interest to the organization.
  - Naming an attribute: i.e. Vehicle_ID.
    - Place its name inside the rectangle for the associated entity in the E-R diagram.

Entity-Relationship (E-R) Modeling

Key Terms

- **Candidate keys and identifiers**
  - **Candidate key**: an attribute (or combination of attributes) that uniquely identifies each instance of an entity type.
  - **Identifier**: a candidate key that has been selected as the unique, identifying characteristic for an entity type.

Rules for Selecting an identifier

- Choose a candidate key that will not change its value.
- Choose a candidate key that will never be null.
- Avoid using intelligent keys.
- Consider substituting single value surrogate keys for large composite keys.

Entity-Relationship (E-R) Modeling

Key Terms

- **Multi-valued Attribute**
  - An attribute that may take on more than one value for each entity instance (multi skills for one employee)
  - Represented on E-R Diagram in two ways:
    - double-lined ellipse
    - weak entity (employee and dependent name, age, and relation).
- **Repeating group**: a set or two or more multivalued attributes that are logically related.

Entity-Relationship (E-R) Modeling

Key Terms

- **Relationship**
  - An association between the instances of one or more entity types that is of interest to the organization
  - Association indicates that an event has occurred or that there is a natural link between entity types
  - Relationships are always labeled with verb phrases (student *completes* course)

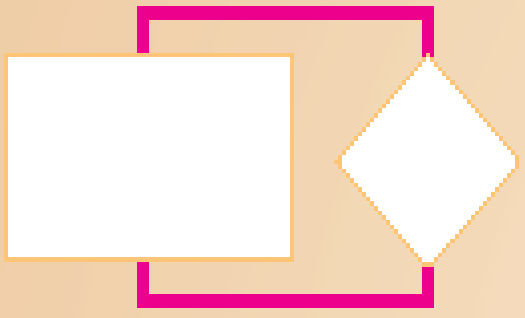
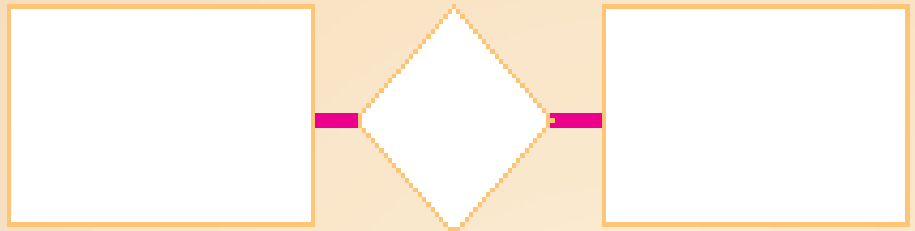
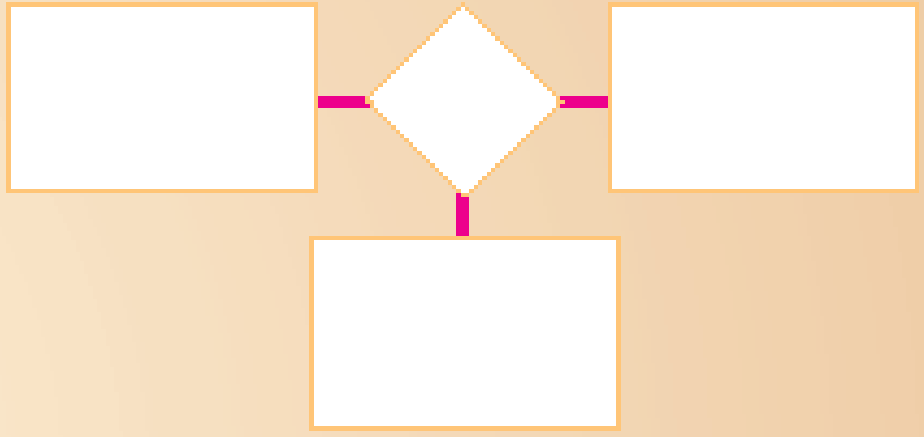
Degree of Relationship

- Degree of relationship is the number of entity types that participate in a relationship

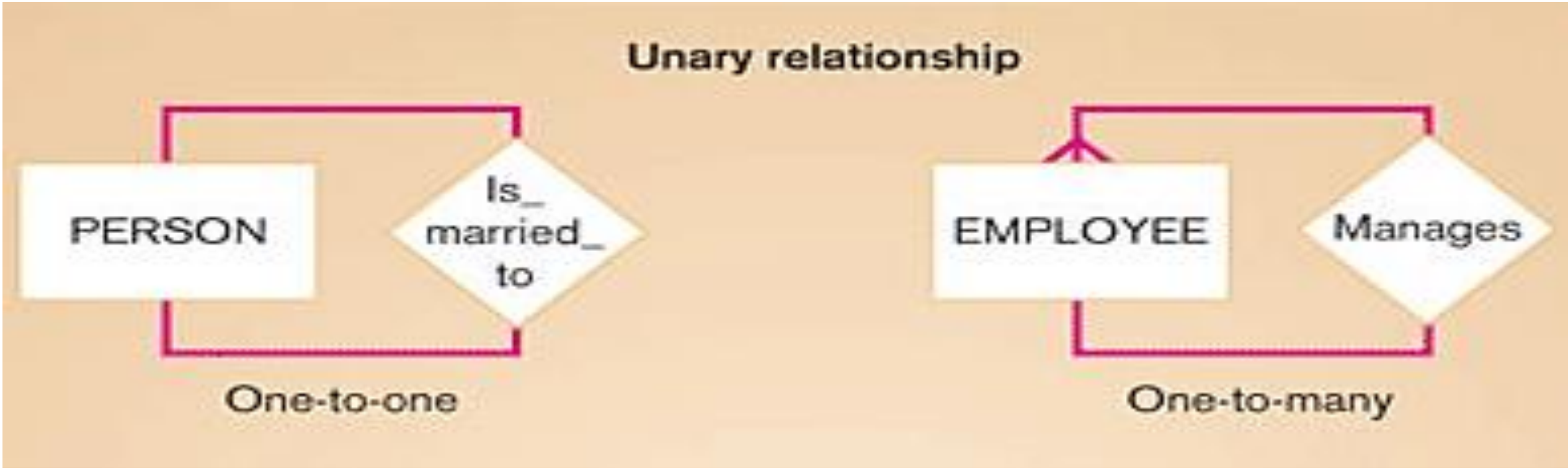
- For example, the relationship *Student Completes Course* is of degree 2 since it has two entities *Student* and *Course*

Degree of Relationship

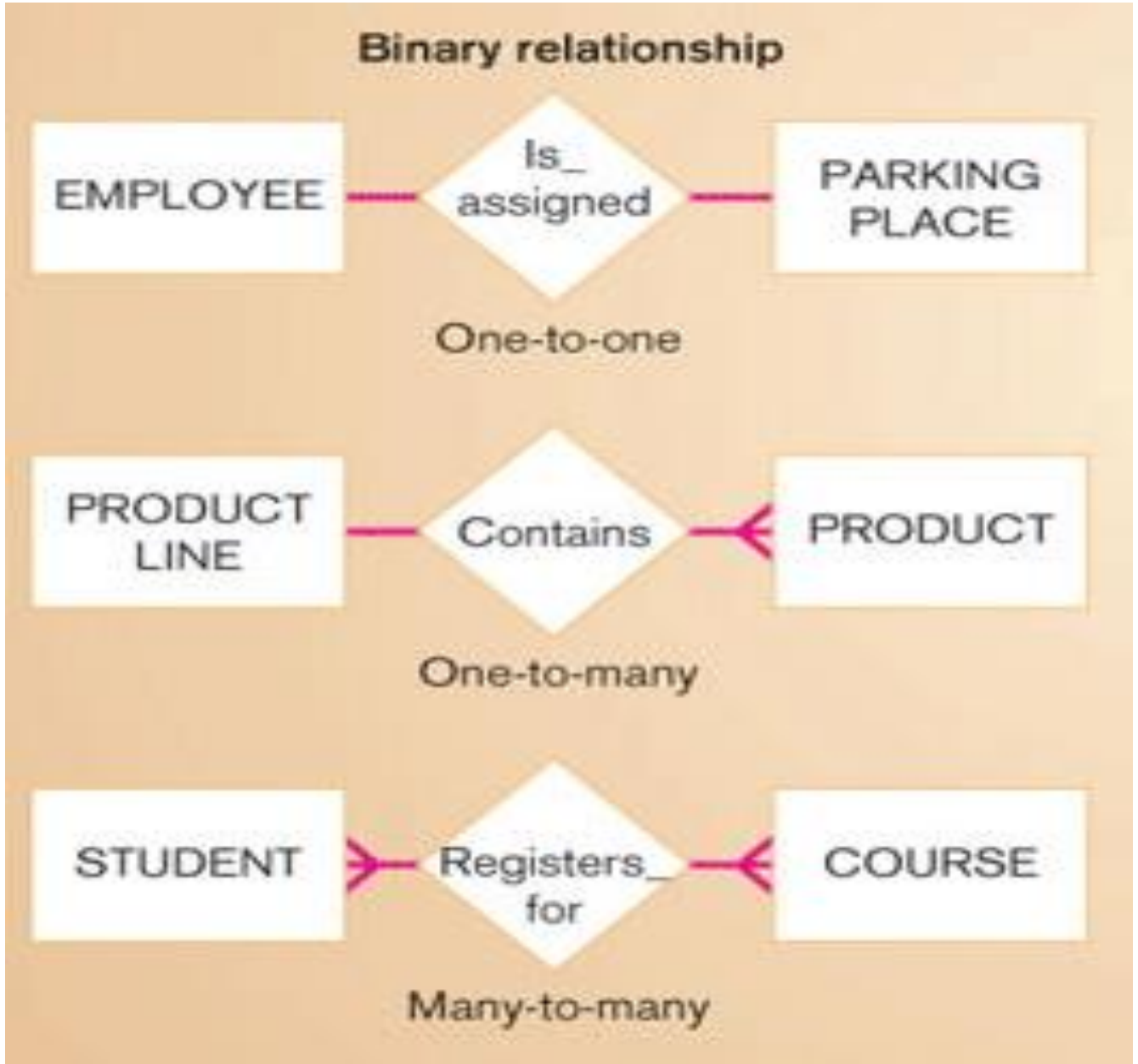
There are three types of relationships based on the degree

A relationship between two instances of one entity type	A relationship between the instances of two entity types	A simultaneous relationship among the instances of three entity types Not the same as three binary relationships
 Unary	 Binary	 Ternary

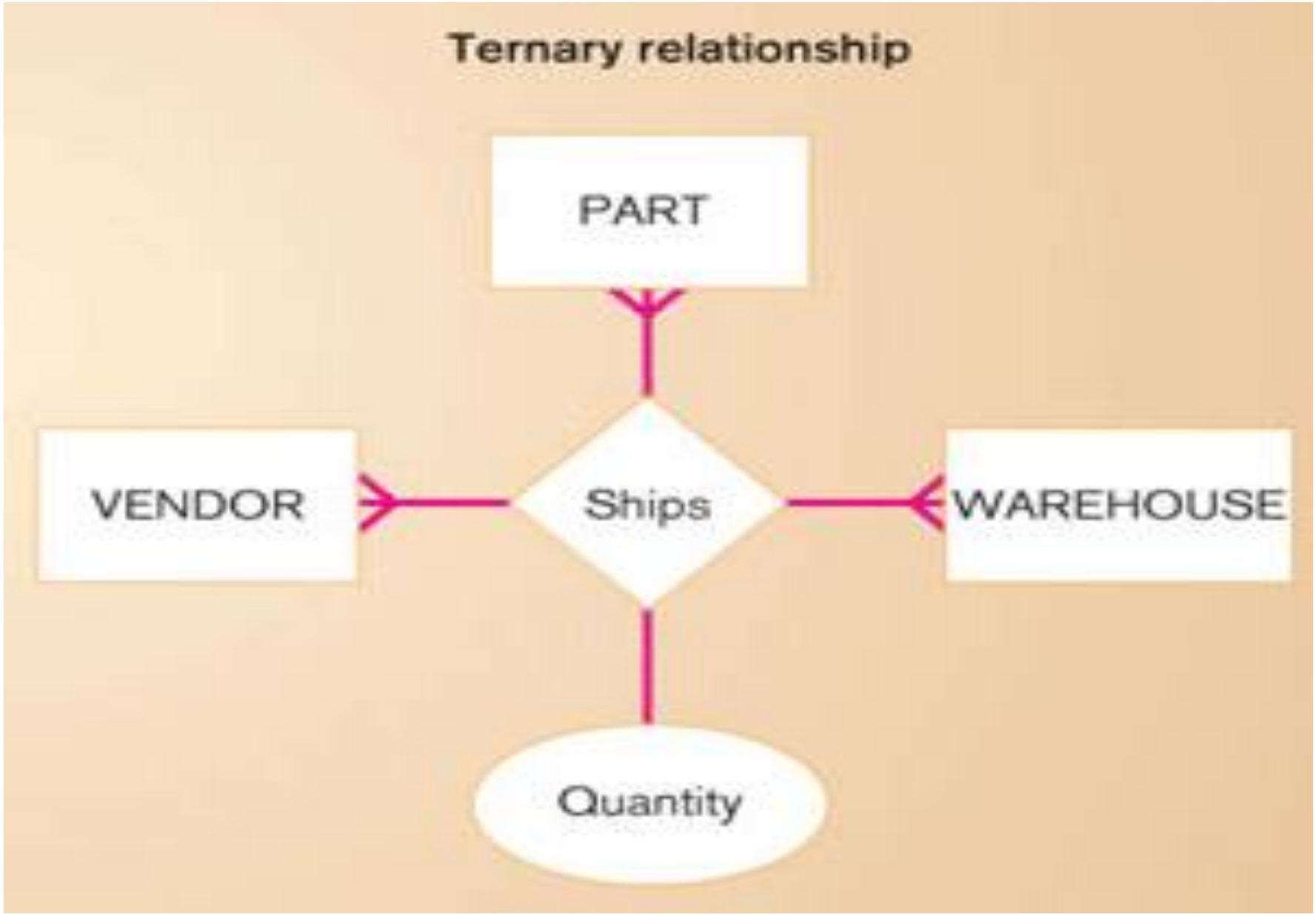
Degree of Relationship



Degree of Relationship



Degree of Relationship



Cardinality

- **Cardinality is** the number of instances of entity B that can (or must) be associated with each instance of entity A.
- Minimum Cardinality
  - The minimum number of instances of entity B that may be associated with each instance of entity A
- Maximum Cardinality
  - The maximum number of instances of entity B that may be associated with each instance of entity A.
- Mandatory vs. Optional Cardinalities
  - Specifies whether an instance must exist or can be absent in the relationship.

Cardinality

There are three types of relationships based on cardinality

1. **One to One –**

- When each entity in each entity set can take part **only once in the relationship**, the cardinality is one to one.
- Let us assume that a driver can drive one car a car can be driven by only one driver.

Cardinality

2. **One to Many**

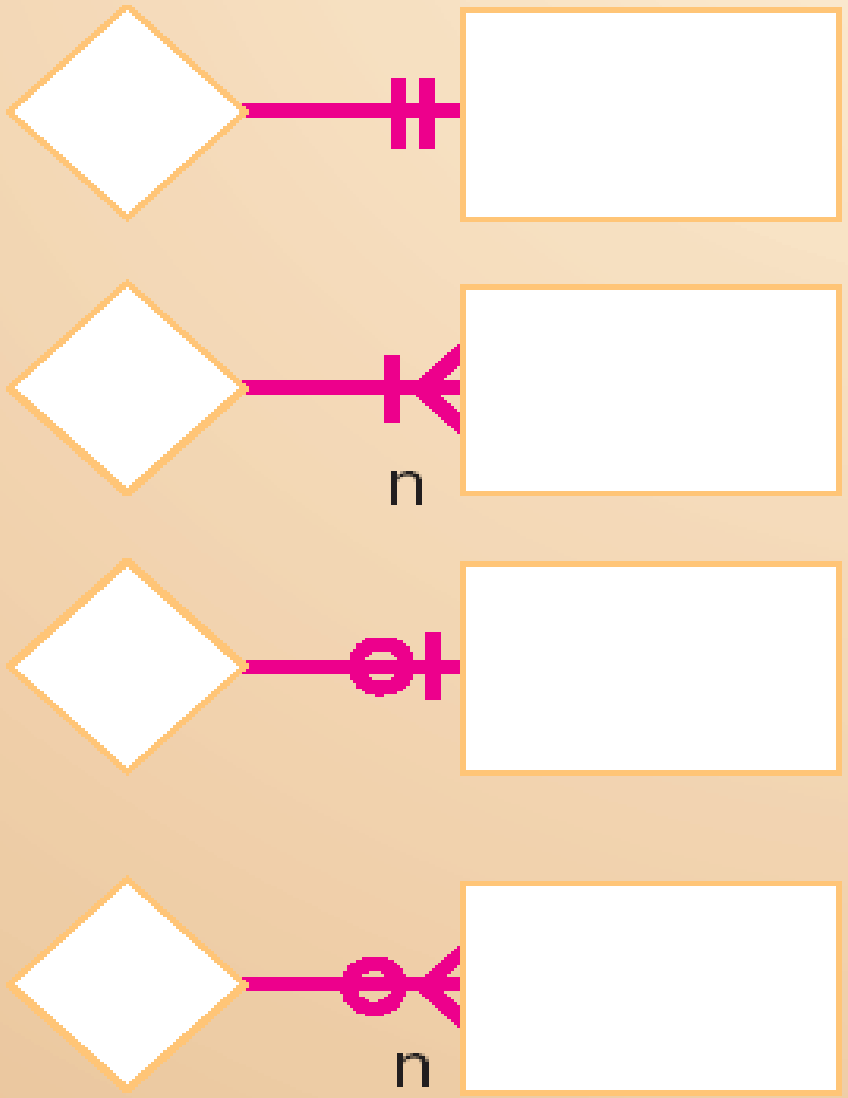
- When entities in one entity set **can take part only once in the relationship and entities in other entity set can take part more than once in the relationship,** cardinality is one to many.
- Let us assume that a student can take only one course but one course can be taken by many students. So the cardinality will be n to 1. It means that for one course there can be n students but for one student, there will be only one course.

Cardinality

3. Many to many –

- When entities in all entity sets can **take part more than once in the relationship** cardinality is many to many.
- Let us assume that a student can take more than one course and one course can be taken by many students. So the relationship will be many to many.

Cardinality Symbols



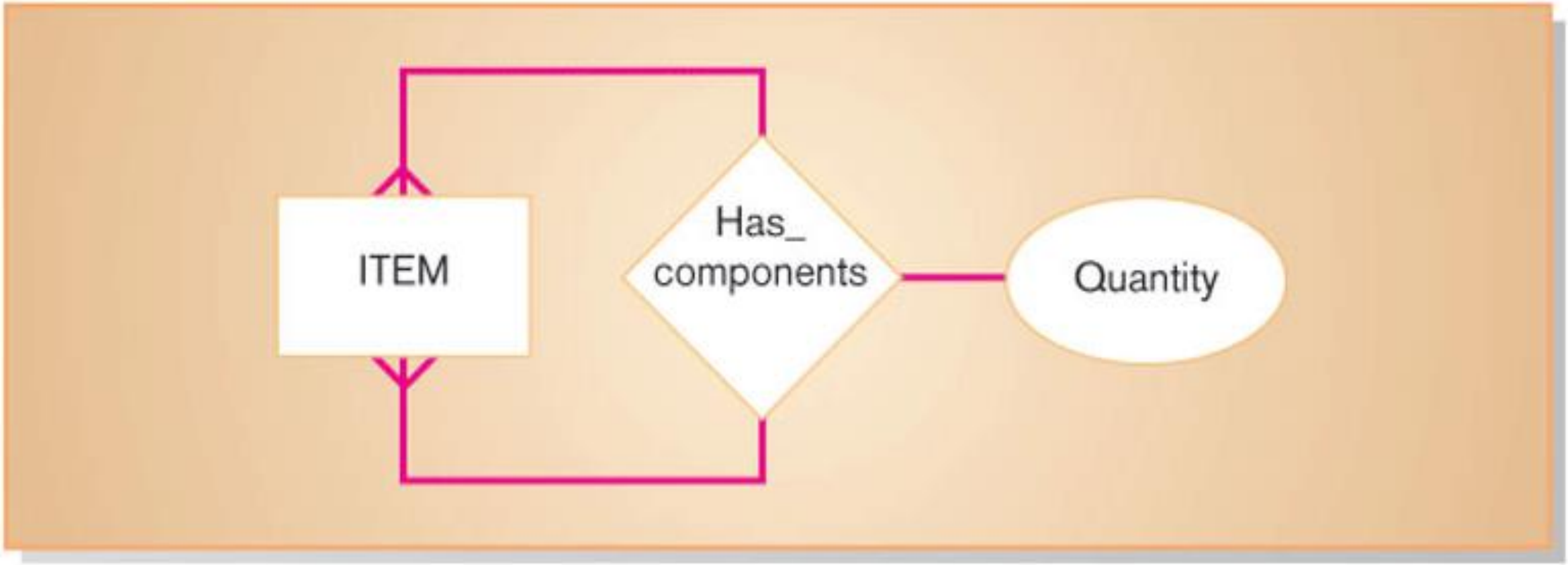
Mandatory 1 cardinality

Mandatory many (M) cardinality (1, 2, …, many)
(*n* is a number for an upper limit, if one exists)

Optional 0 or 1 cardinality

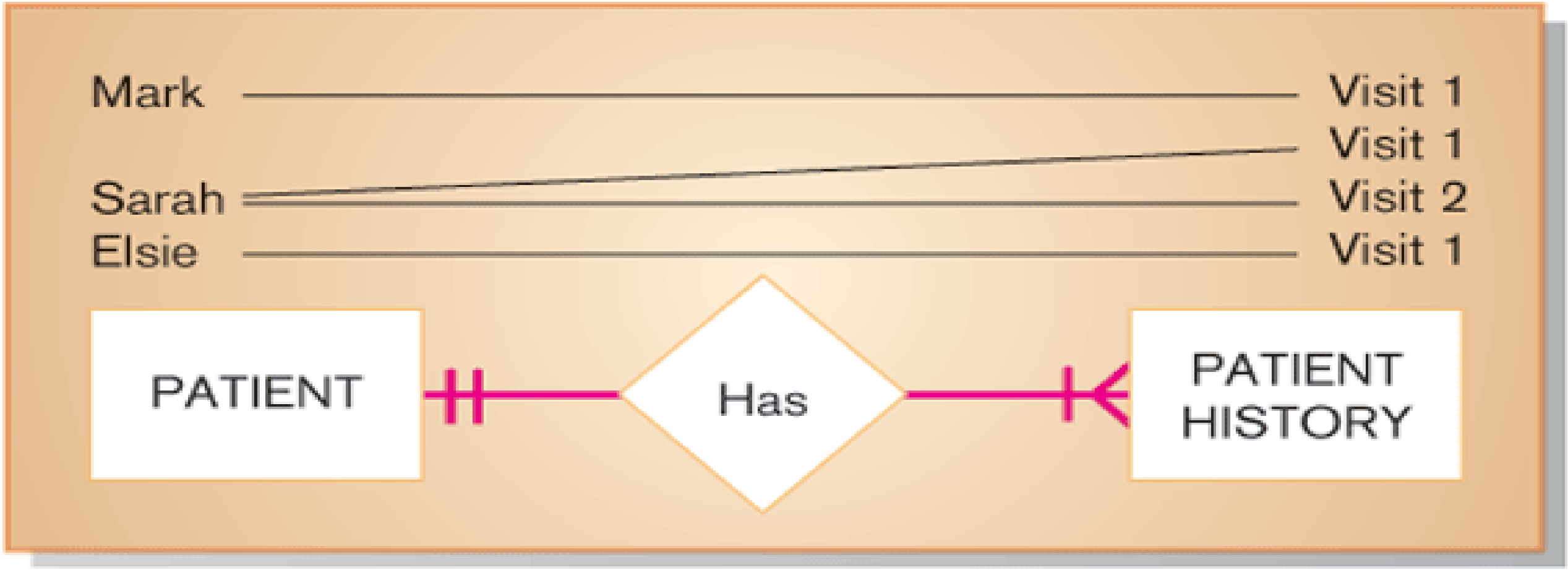
Optional zero-many cardinality (0, 1, 2, …, many)

Unary Relationship Example



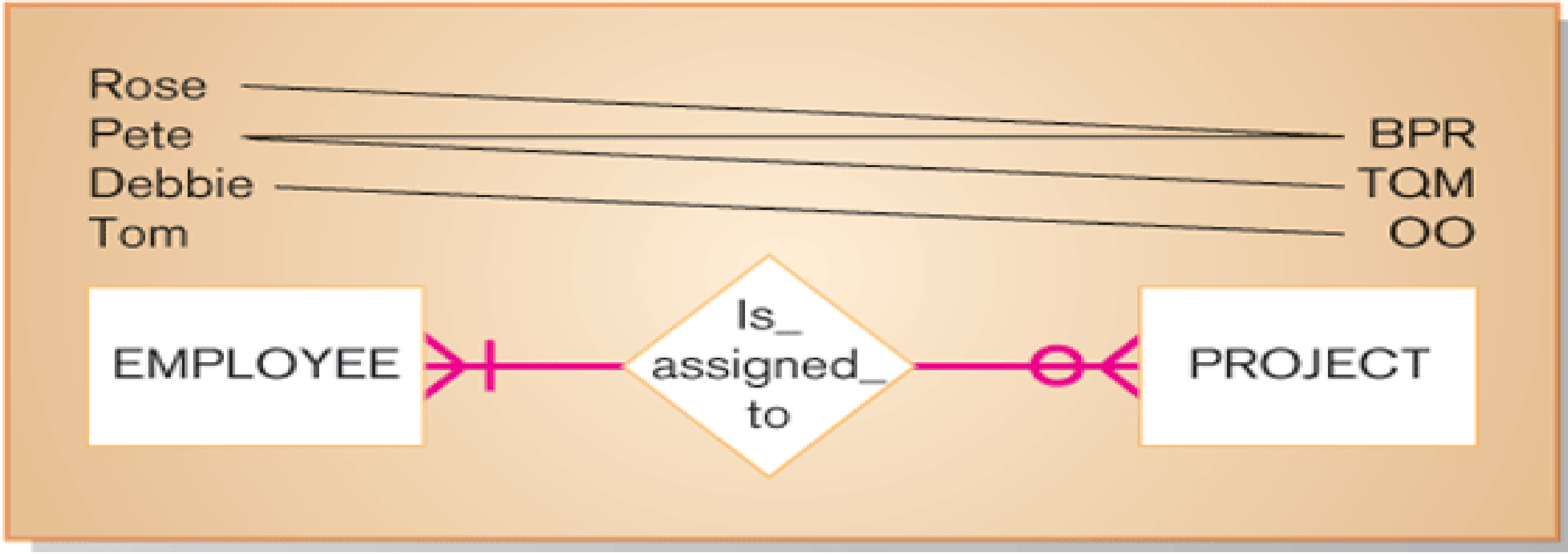
Binary Relationship Example

Example of binary relationship with mandatory cardinalities



Binary Relationship Example

Example of binary relationship with one optional and one mandatory cardinality



Naming and Defining Relationships

- A relationship name is a *verb phrase* and *avoid vague names*.
- A relationship definition:
  - Explains *what action is to be taken and possibly why it is important*.
  - *Gives examples to clarify the action*.
  - Explain any *optional participation.*
  - *Explain the reason for any explicit maximum cardinality* other than many.
  - *Explain any restrictions on participation in the relationship*.
  - Explain the extent of history that is kept in the relationship.
  - *Explain the extent of history that is kept in the relationship*.
  - *Explain whether an entity instance* involved in a relationship instance *can transfer participation to another relationship instance*

Steps to Create and ERD

EXAMPLE

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

Step 1) Entity Identification

We have three entities

- Student
- Course
- Professor

Step 2) Relationship Identification

We have the following two relationships

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

Step 3) Cardinality Identification

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

Step 4) Identify Attributes

- You need to study the files, forms, reports, data currently maintained by the organization to identify attributes.
- You can also conduct interviews with various stakeholders to identify entities. Initially, it's important to identify the attributes without mapping them to a particular entity.
- Once, you have a list of Attributes, you need to map them to the identified entities.
- Ensure an attribute is to be paired with exactly one entity.
- If you think an attribute should belong to more than one entity, use a modifier to make it unique.
- Once the mapping is done, identify the primary Keys.
- If a unique key is not readily available, create one.

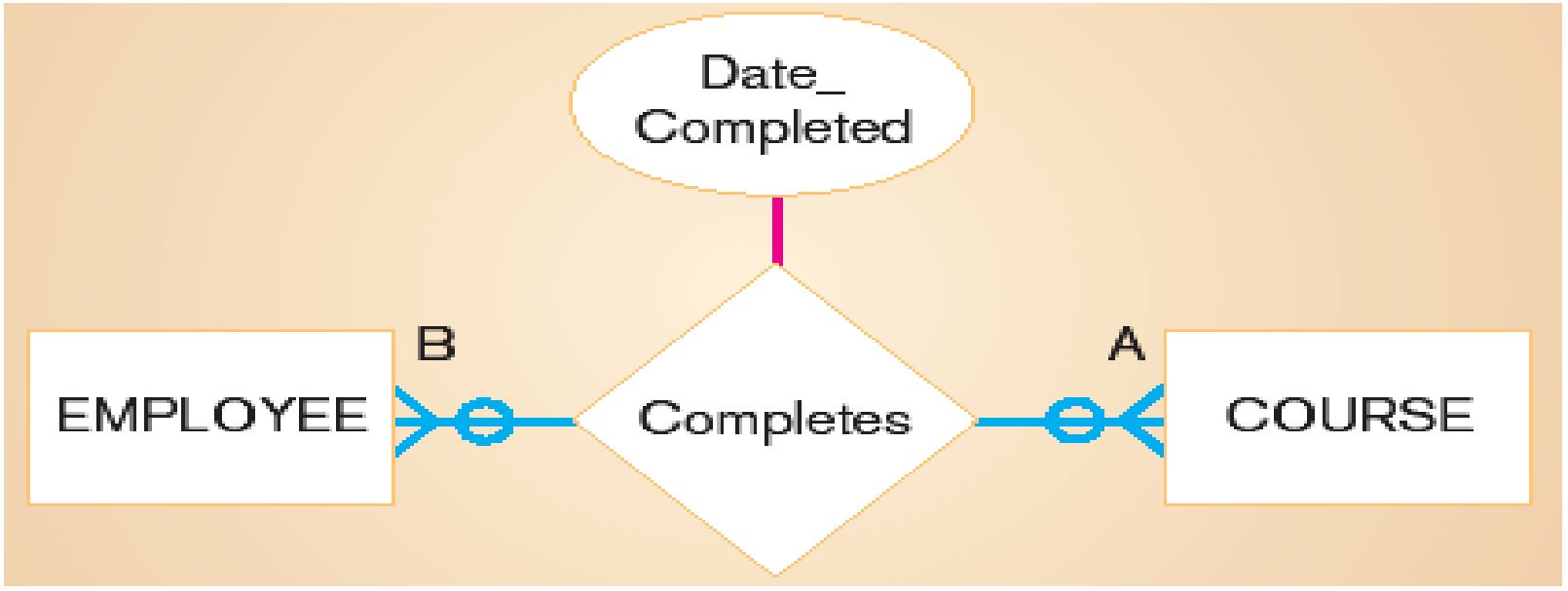
Entity	Primary Key	Attributes
Student	Student_ID	StudentName
Professor	Employee_ID	ProfessorName
Course	Course_ID	CourseName

Associative Entity (gerund)

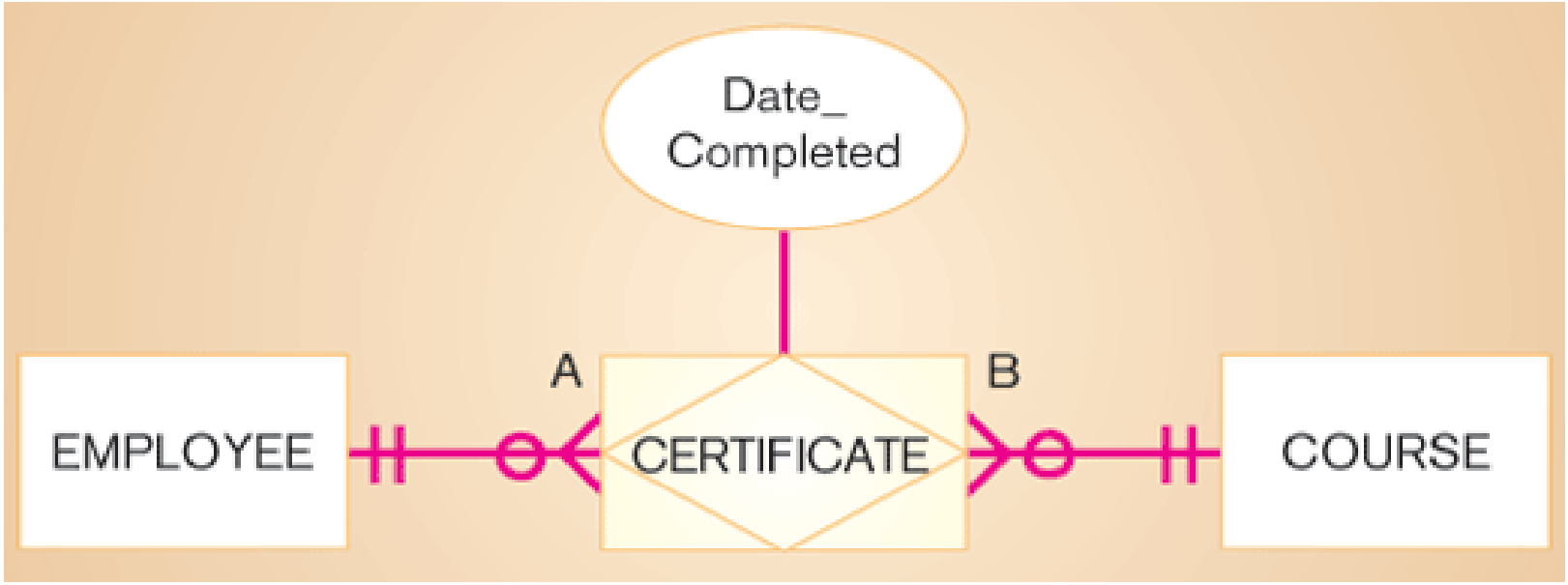
- **Associative Entity**: an entity type that associates the instances of one or more entity types and contains attributes that are peculiar to the relationship between those entity instances.
  - Sometimes called a gerund.
- The data modeler chooses to model the relationship as an entity type.

Associative Entity example

A relationship with a attribute



Representing as an associative entity



Super types and subtypes

Supertype: a generic entity type that has a relationship with one or more subtypes. *(Patient, Person, Student)*

Subtype: a subgrouping of the entities in an entity type

- Is meaningful to the organization.
- Shares common attributes or relationships distinct from other subgroupings.
- *(outpatient, inpatient, student, employee, alumnus, undergraduate, graduate*

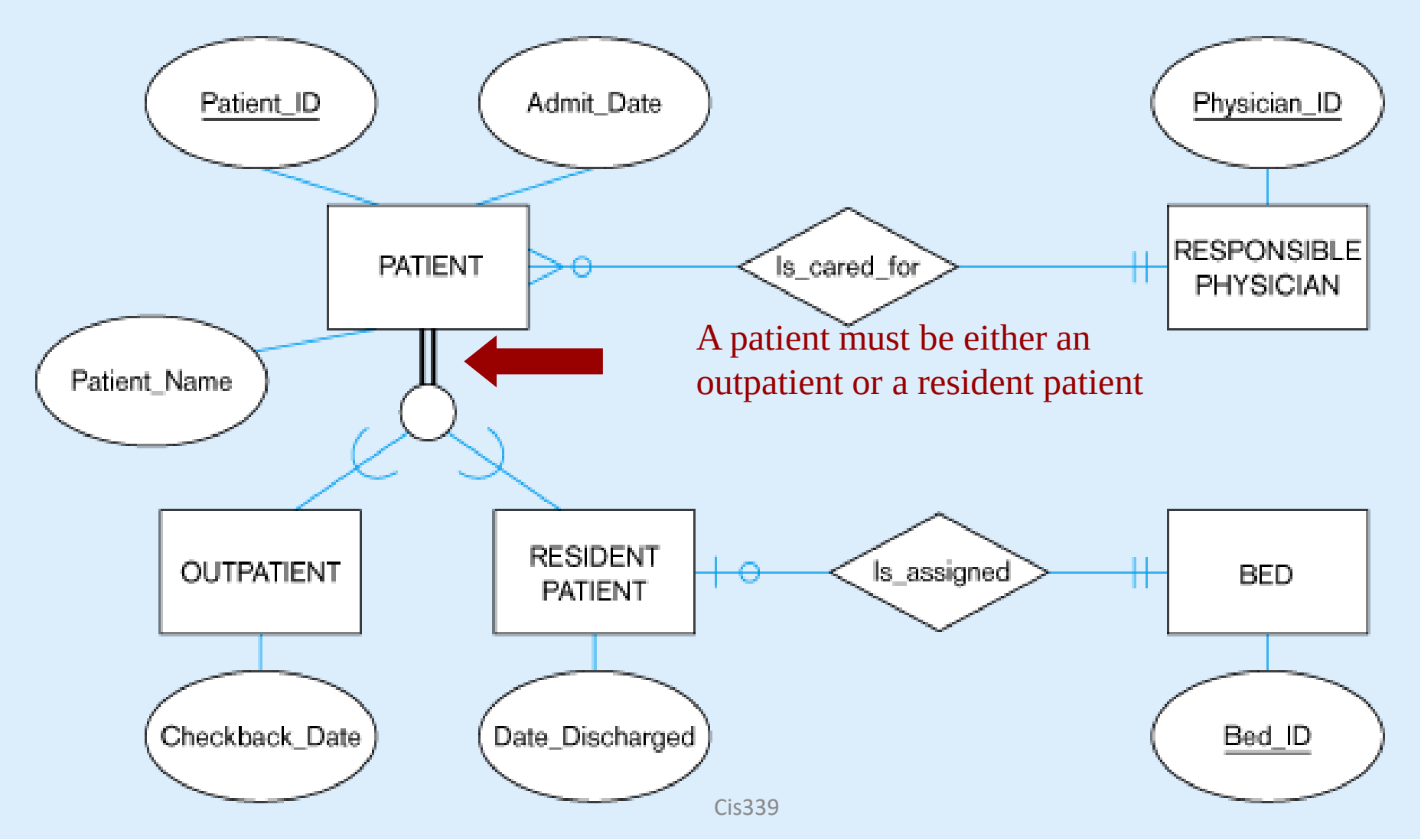
Rules for super types and subtypes

❑ ***Completeness Constraints*:**

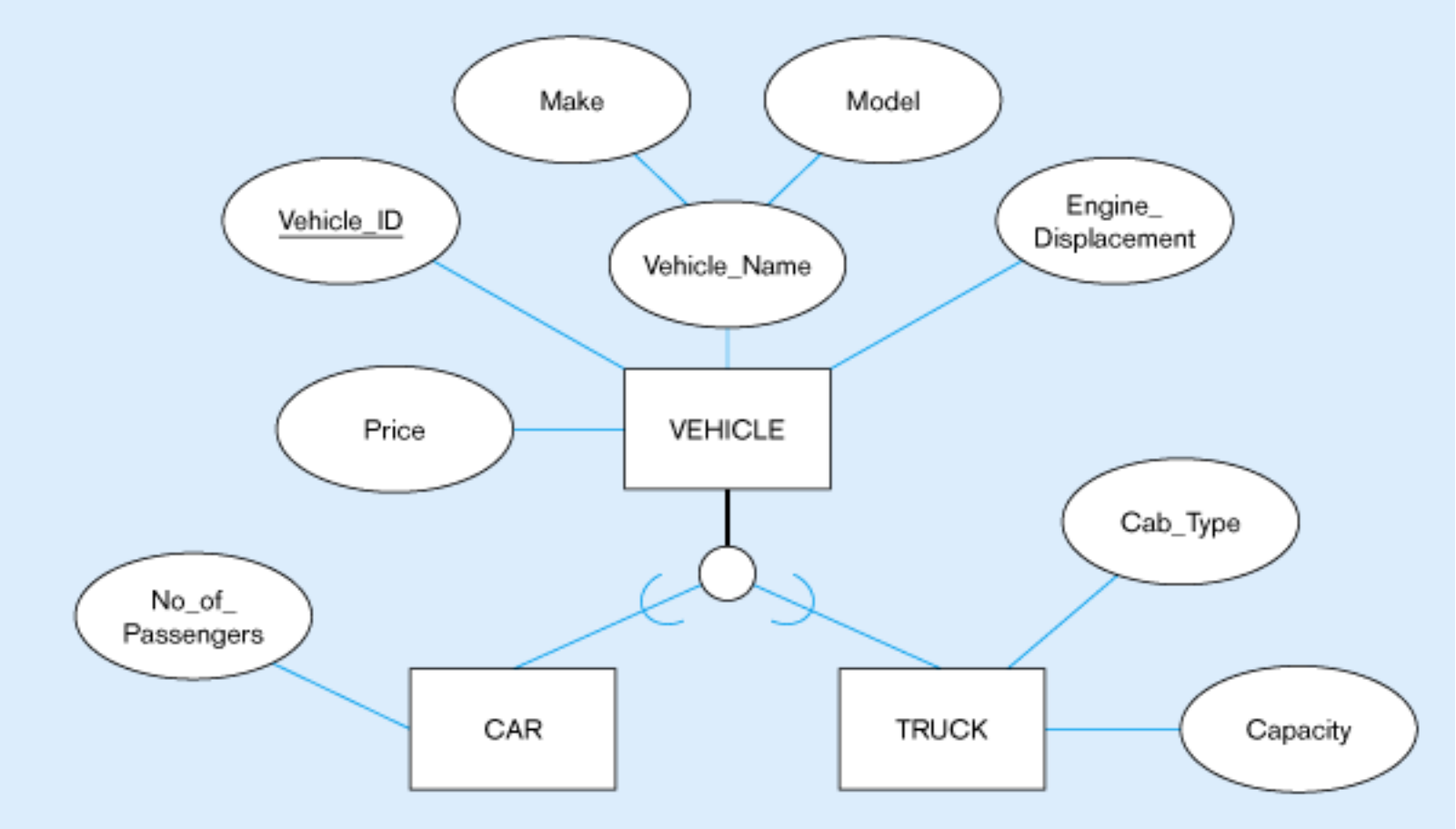
Whether an instance of a super type *must* also be a member of at least one subtype.

- **Total specialization**, specifies that each entity instance of the supertype must be a member of some subtype in the relationship., **(*ex. a student can be only a grad or undergrad or person must be student, employee or alumnus)***
- **Partial specialization**, specifies that an entity instance of the supertype does not have to belong to any subtype. may or may not be an instance of one of the subtypes. ***ex employee can be a faculty or staff or just an employee.***

Example.. Total specialization



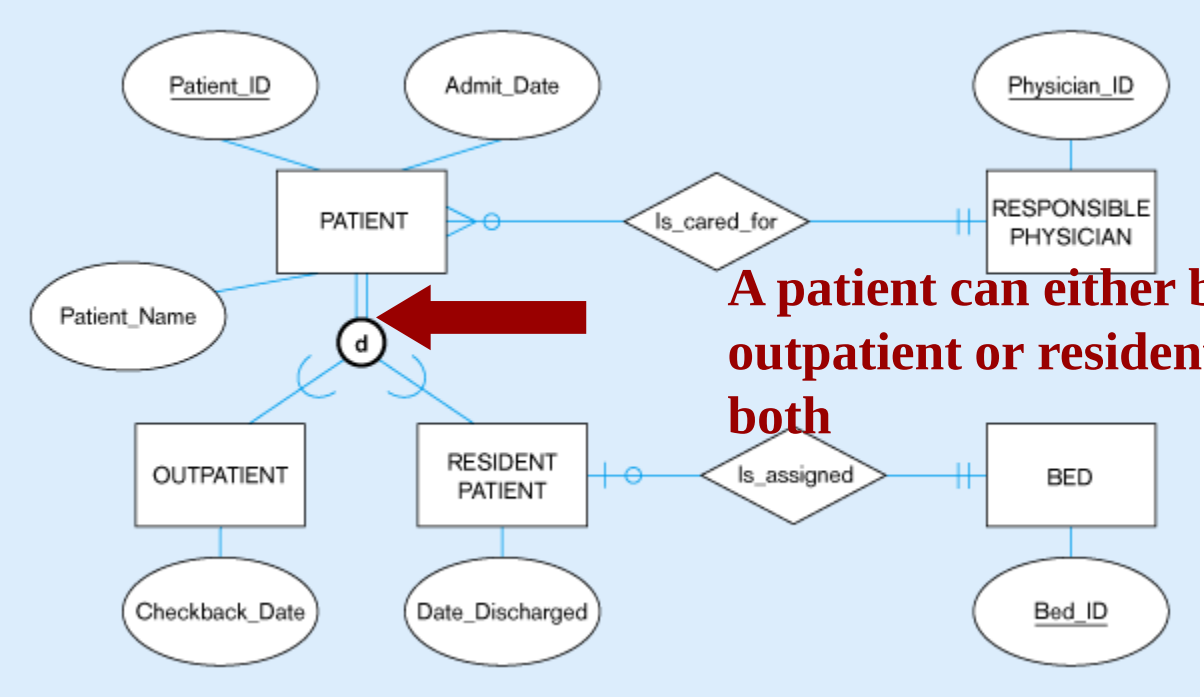
Example- Partial specialization



Rules for super types and subtypes

- ***Disjointness Constraints***:
- **Whether an instance of a super type may *simultaneously* be a member of two (or more) subtypes**
- **Disjoint rule**: specifies that if an entity instance of the supertype is a member of one subtype, it cannot simultaneously be a member of any other subtype. **. ex, student is grad or undergrad.**
- **Overlap**, specifies that an entity instance can simultaneously be a member of two (or more) subtypes**, ex person can be employee, alumnus, or student or any combination of these types.**

Disjoint rule



A patient can either be outpatient or resident, but not both

Overlap rule