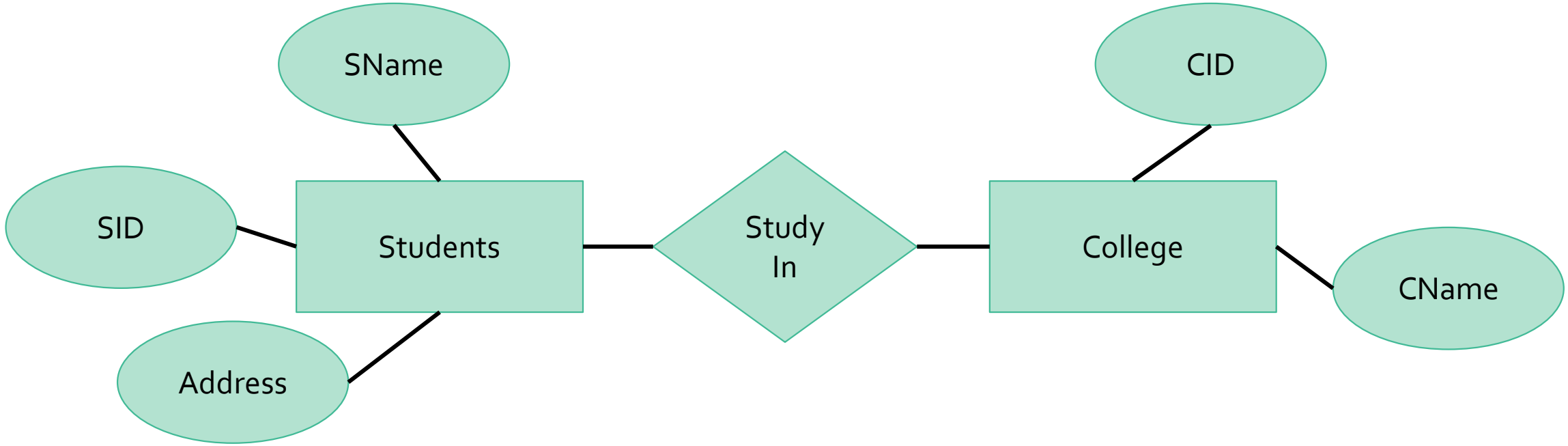
ENTITY RELATIONSHIP MODEL

Entity Relationship Diagram

An Entity Relationship Diagram (ERD) is a visual representation of different data using conventions that describe how these data are related to each other.



Purpose

- The database analyst/designer gains a better understanding of the information to be contained in the database through the process of constructing the ERD.
- The ERD serves as a documentation tool.
- Finally, the ERD is used to communicate the logical structure of the database to users. In particular, the ERD effectively communicates the logic of the database to users.

Major Components

- Entities
- Attributes
- Relationships

Entity

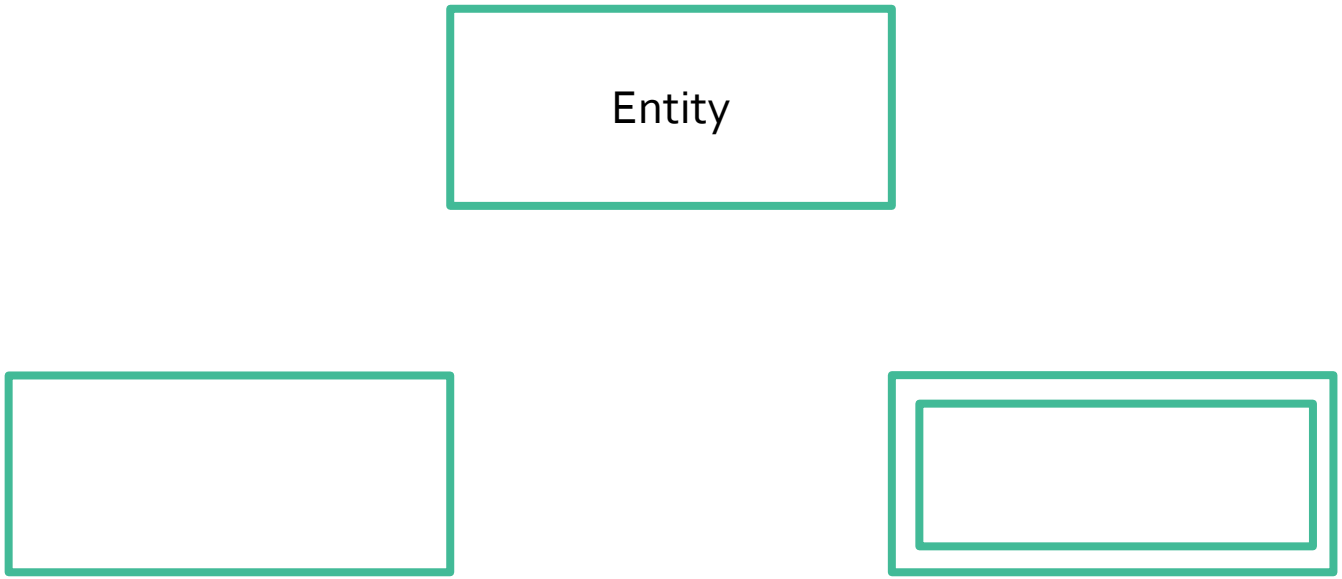
- A name/label assigned to items/objects that exist in an environment and that have similar properties.
- It could be person, place, event or even concept.

Types of Entity

- A **strong/regular entity** is the one whose instances can exist independently. They have their own identity.

- A **weak entity** is the one whose instances cannot exist without being linked with instances of some other entity. They cannot exist independently. In more technical terms it can defined as an entity that cannot be identified by its own attributes.

Symbols



Strong/regular entity

Weak entity

When naming Entity

- Singular noun recommended
- Organization specific names
- Write in capitals
- Abbreviations can be used, be consistent

Attribute

An attribute of an entity is a defining property or quality.

Domain of an attribute

- Every attribute has a domain
- Set of possible values for an attribute
- The attributes in an entity set get the values from the same domain

Types of Attributes

Single vs. composite

- Simple attributes are atomic values, which cannot be divided further.
- Composite attributes are made of more than one simple attribute.

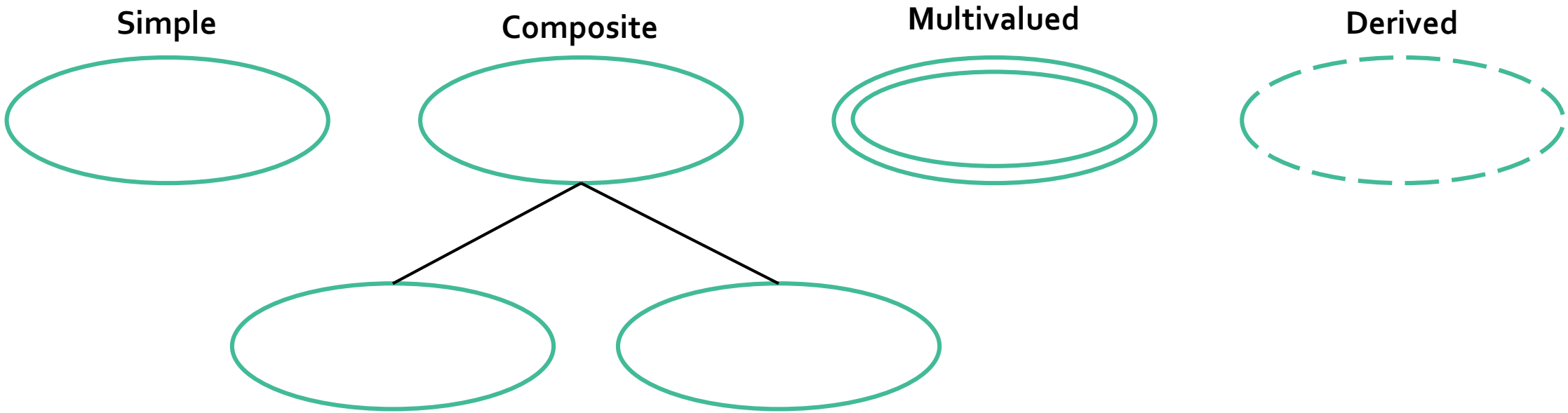
Single valued vs. multi-valued

- Single-value attributes contain single value.
- Multi-value attributes may contain more than one values.

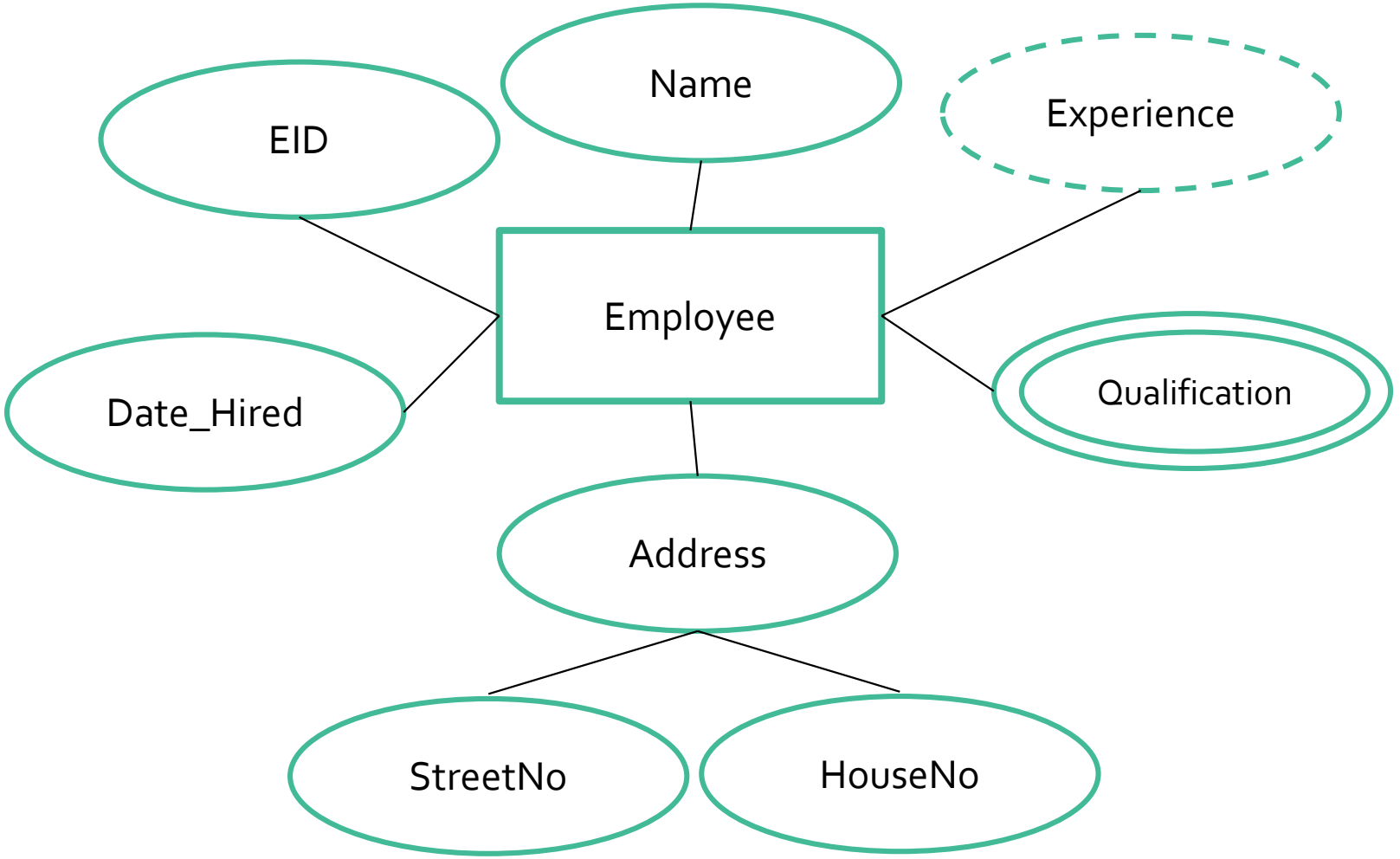
Stored vs. derived

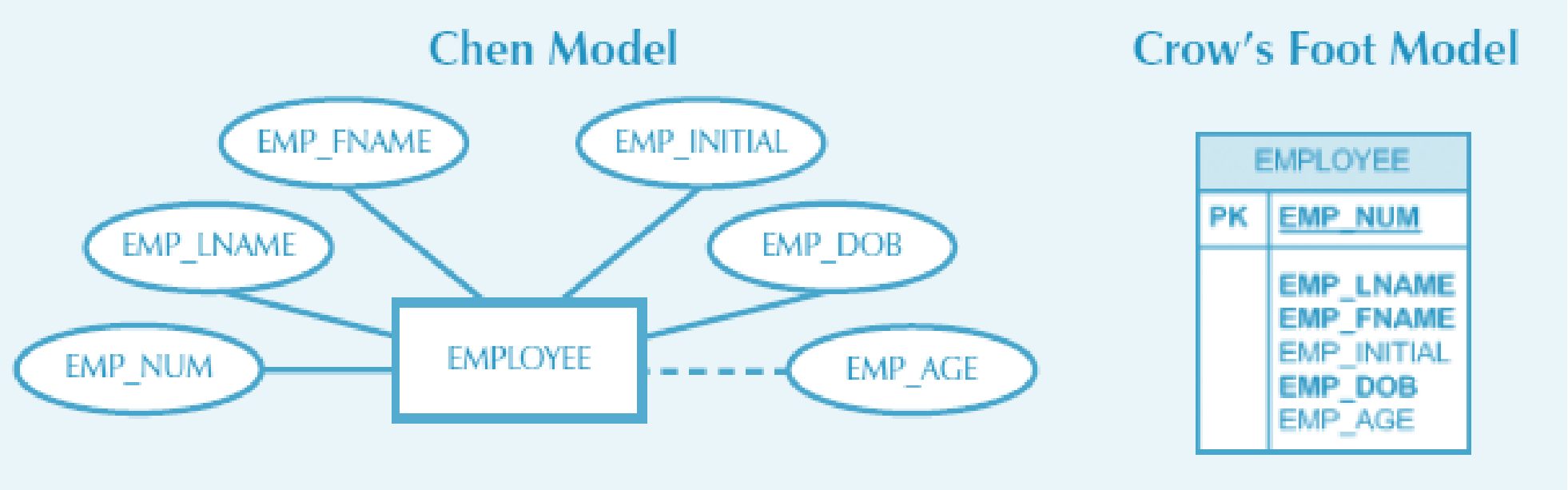
- Stored attributes are the one whose value we store in the database.
- Derived attributes are the attributes that do not exist in the physical database, but their values are derived from other attributes present in the database.

Symbols



Example


Name
EID
Experience
Employee
Date_Hired
Qualification
Address
StreetNo
HouseNo


Chen Model
EMP_FNAME
EMP_INITIAL
EMP_LNAME
EMP_DOB
EMP_NUM
EMPLOYEE
EMP_AGE
Crow's Foot Model
EMPLOYEE
PK
EMP_NUM
EMP_LNAME
EMP_FNAME
EMP_INITIAL
EMP_DOB
EMP_AGE

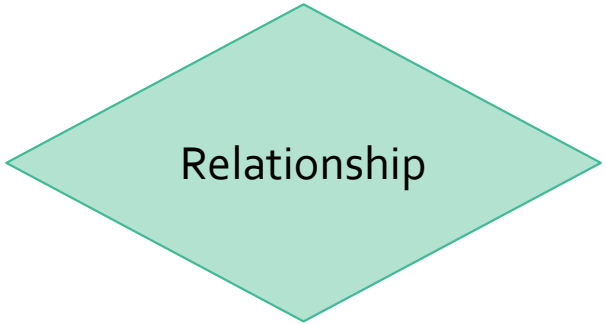
Advantages and Disadvantages of Storing Derived Attributes

TABLE 4.2 Advantages and Disadvantages of Storing Derived Attributes

	DERIVED ATTRIBUTE	
	STORED	NOT STORED
Advantage	Saves CPU processing cycles Saves data access time Data value is readily available Can be used to keep track of historical data	Saves storage space Computation always yields current value
Disadvantage	Requires constant maintenance to ensure derived value is current, especially if any values used in the calculation change	Uses CPU processing cycles Increases data access time Adds coding complexity to queries

Relationship

A relationship describes how entities interact. It is a association among entities. Relationships are represented by diamond shapes and are labeled using verbs.



Types of Relationship

Binary

- The one that links two entity .

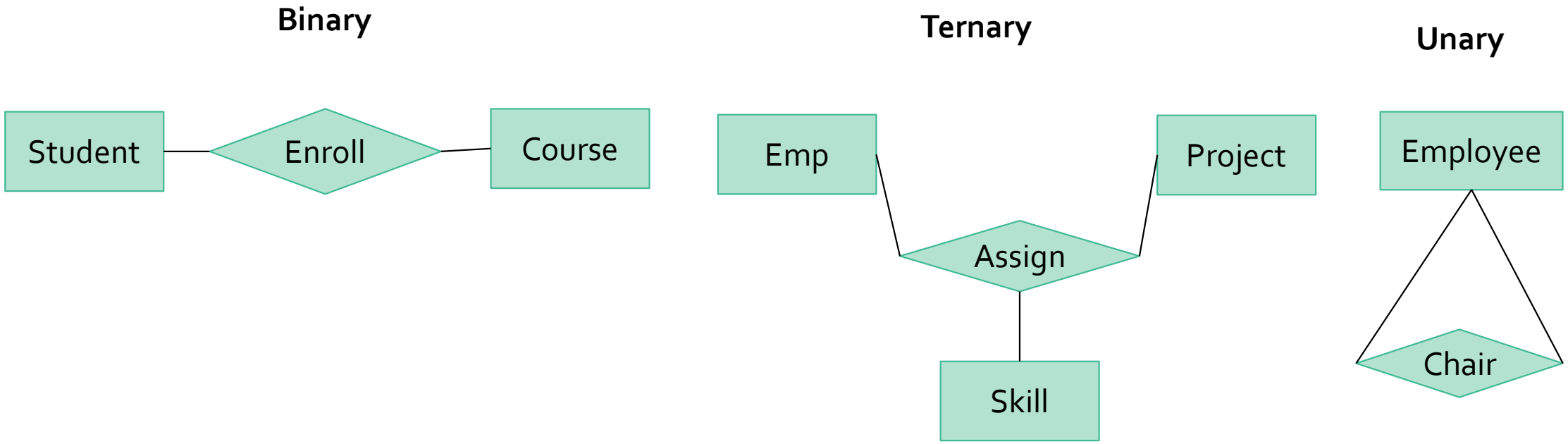
Ternary

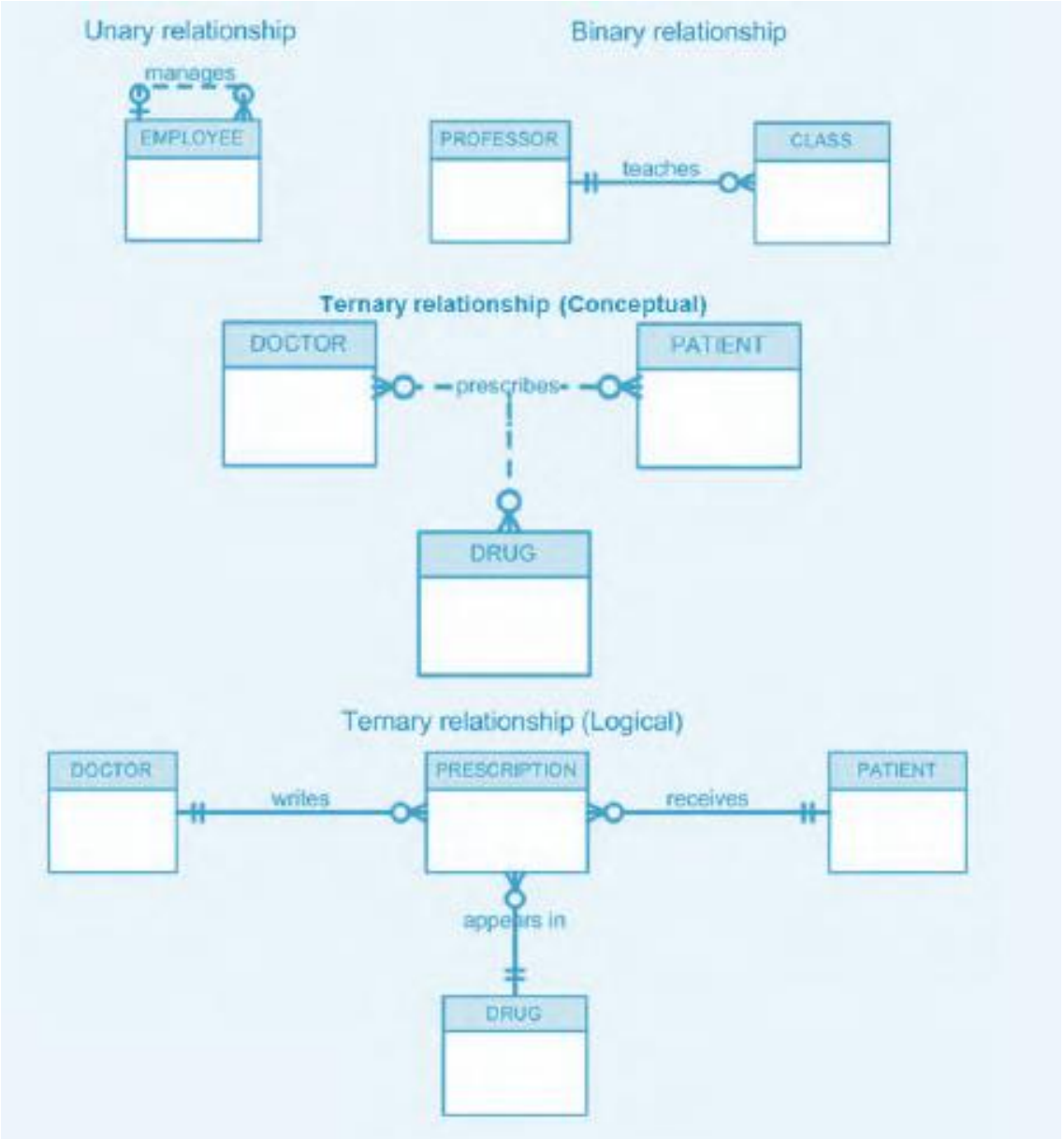
- One that involves three entities.

Unary

- Entities linked with itself, also called recursive relationship.

Types of Relationship




Unary relationship
manages
EMPLOYEE
Binary relationship
PROFESSOR
teaches
CLASS
Ternary relationship (Conceptual)
DOCTOR
prescribes
PATIENT
DRUG
Ternary relationship (Logical)
DOCTOR
writes
PRESCRIPTION
receives
PATIENT
appears in
DRUG

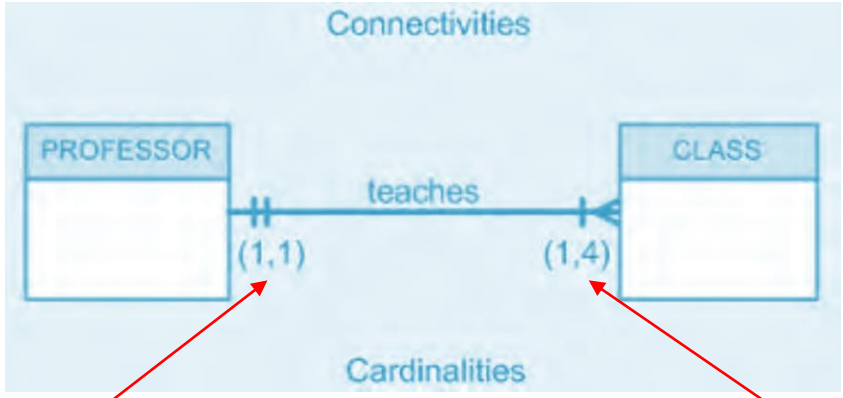
Cardinality

Express the minimum and maximum number of occurrences of one entity for a single occurrence of the other

- One-to-One (1:1)
- One-to-many (1:N)
- Many-to-Many (M:N)

Cardinality

- In the ERD, cardinality is indicated by placing the appropriate numbers beside the entities, using the format (x,y).
- The first value represents the minimum number of associated entities, while the second value represents the maximum number of associated entities.



Each class is taught by one and only one professor.

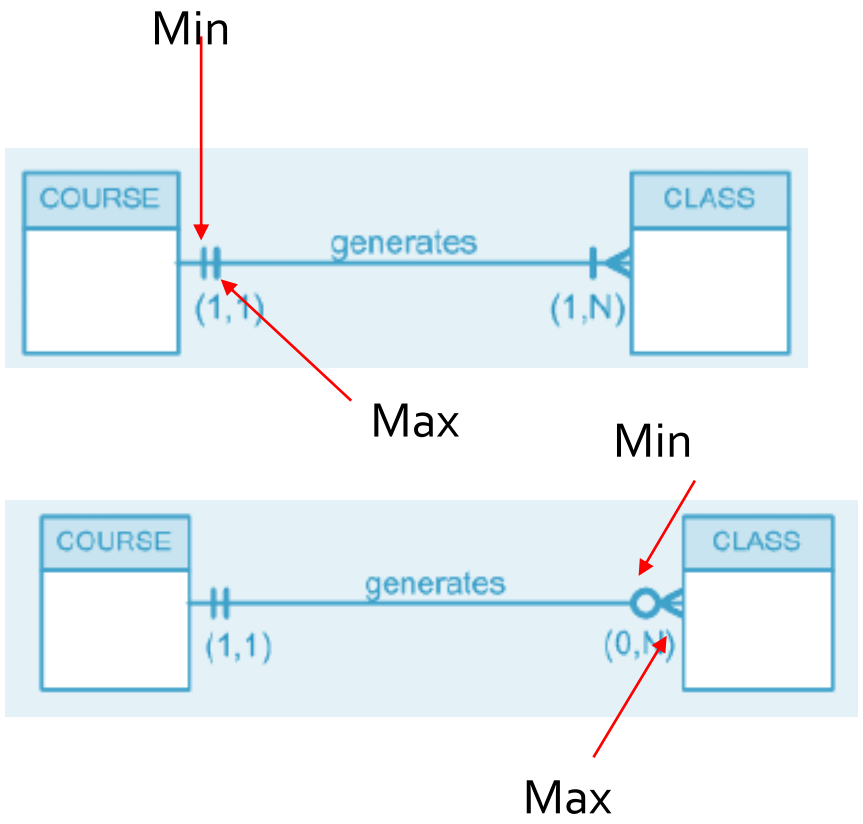
Each professor teaches up to four classes.

Cardinality

TABLE 4.3 Crow's Foot Symbols

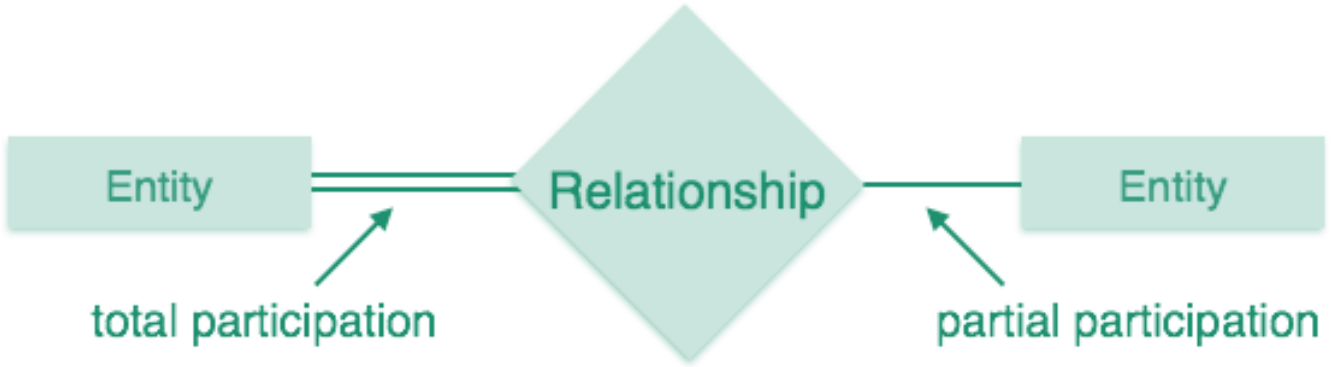
CROW'S FOOT SYMBOL	CARDINALITY	COMMENT
	(0,N)	Zero or many. Many side is optional.
	(1,N)	One or many. Many side is mandatory.
	(1,1)	One and only one. 1 side is mandatory.
	(0,1)	Zero or one. 1 side is optional.

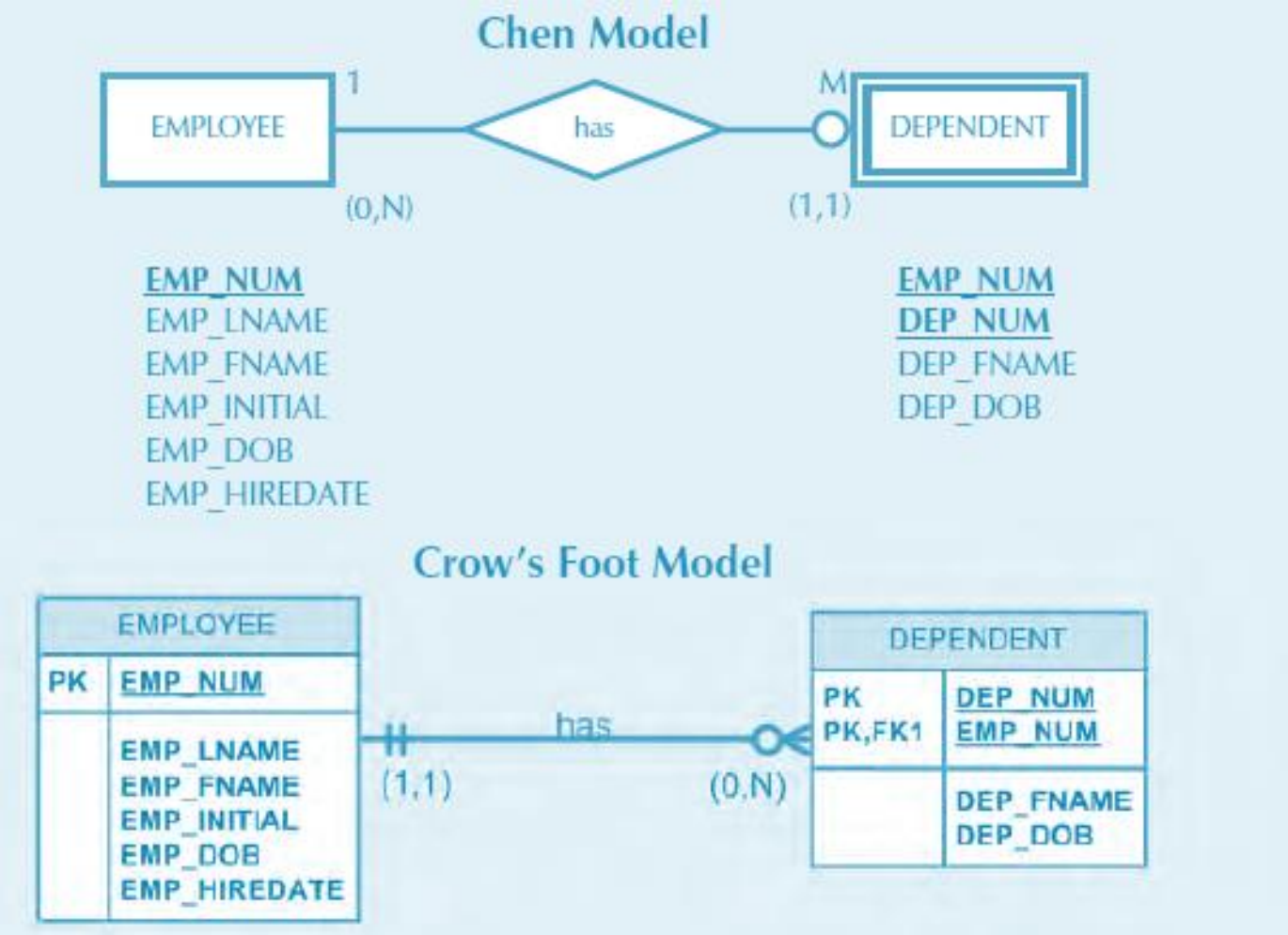
Cardinality



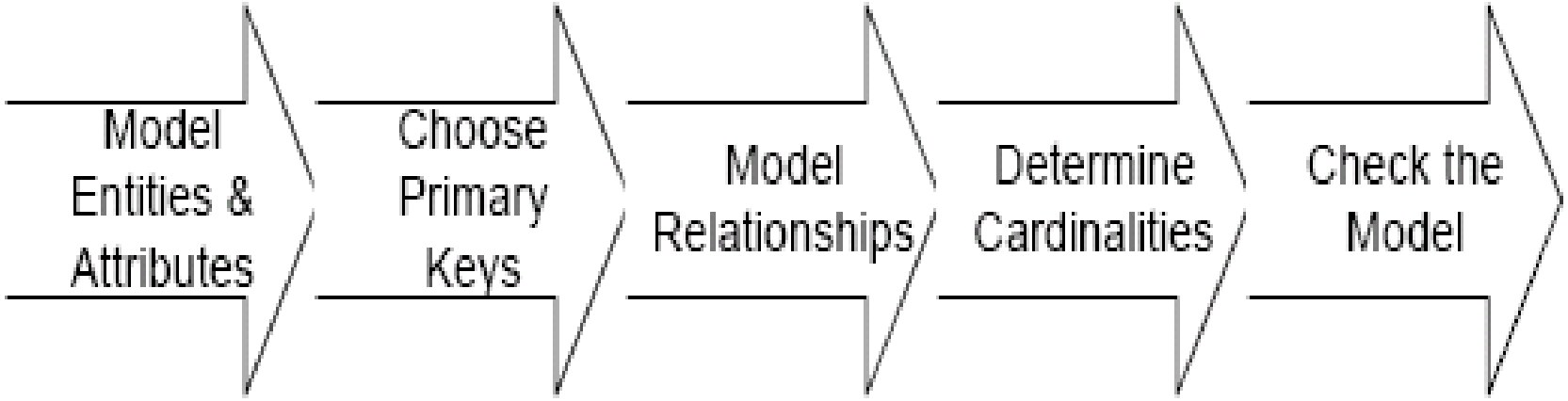
Participation Constraints

- **Total Participation** – Each entity is involved in the relationship. Total participation is represented by double lines.
- **Partial participation** – Not all entities are involved in the relationship. Partial participation is represented by single lines.




Chen Model
EMPLOYEE
1
has
M
DEPENDENT
(0,N)
(1,1)
EMP_NUM
EMP_LNAME
EMP_FNAME
EMP_INITIAL
EMP_DOB
EMP_HIREDATE
EMP_NUM
DEP_NUM
DEP_FNAME
DEP_DOB
Crow's Foot Model
EMPLOYEE
PK
EMP_NUM
EMP_LNAME
EMP_FNAME
EMP_INITIAL
EMP_DOB
EMP_HIREDATE
has
(1,1)
(0,N)
DEPENDENT
PK
PK,FK1
DEP_NUM
EMP_NUM
DEP_FNAME
DEP_DOB

Steps in building an ERD



A Simple Example

A company has several departments. Each department has a supervisor and at least one employee. Employees must be assigned to at least one, but possibly more departments. At least one employee is assigned to a project, but an employee may be on vacation and not assigned to any projects. The important data fields are the names of the departments, projects, supervisors and employees, as well as the supervisor and employee number and a unique project number.

Identify entities

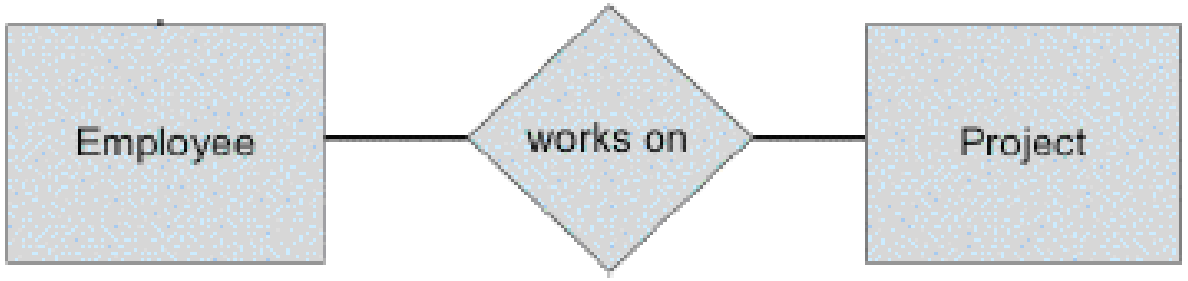
- One approach to this is to work through the information and highlight those words which you think correspond to entities.

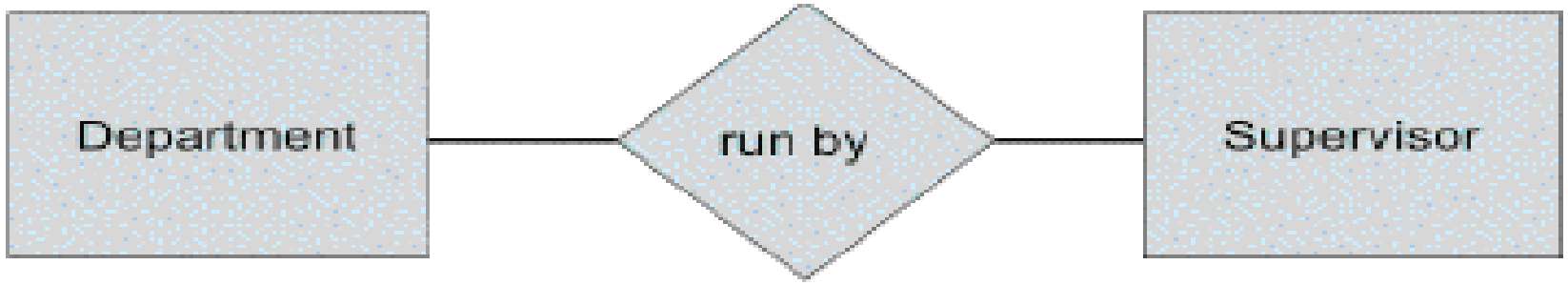
- A company has several departments. Each department has a supervisor and at least one employee. Employees must be assigned to at least one, but possibly more departments. At least one employee is assigned to a project, but an employee may be on vacation and not assigned to any projects. The important data fields are the names of the departments, projects, supervisors and employees, as well as the supervisor and employee number and a unique project number.

Identified Relationships

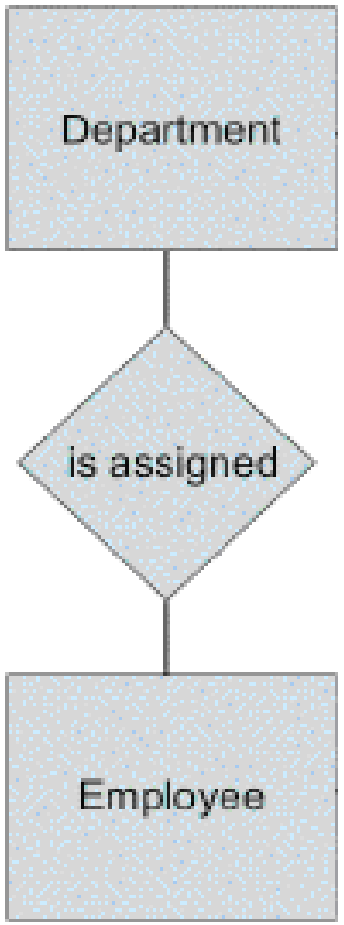
- A Department is assigned an employee
- A Department is run by a supervisor
- An employee belongs to a department
- An employee works on a project
- A supervisor runs a department
- A project uses an employee

Drawing Rough ERD

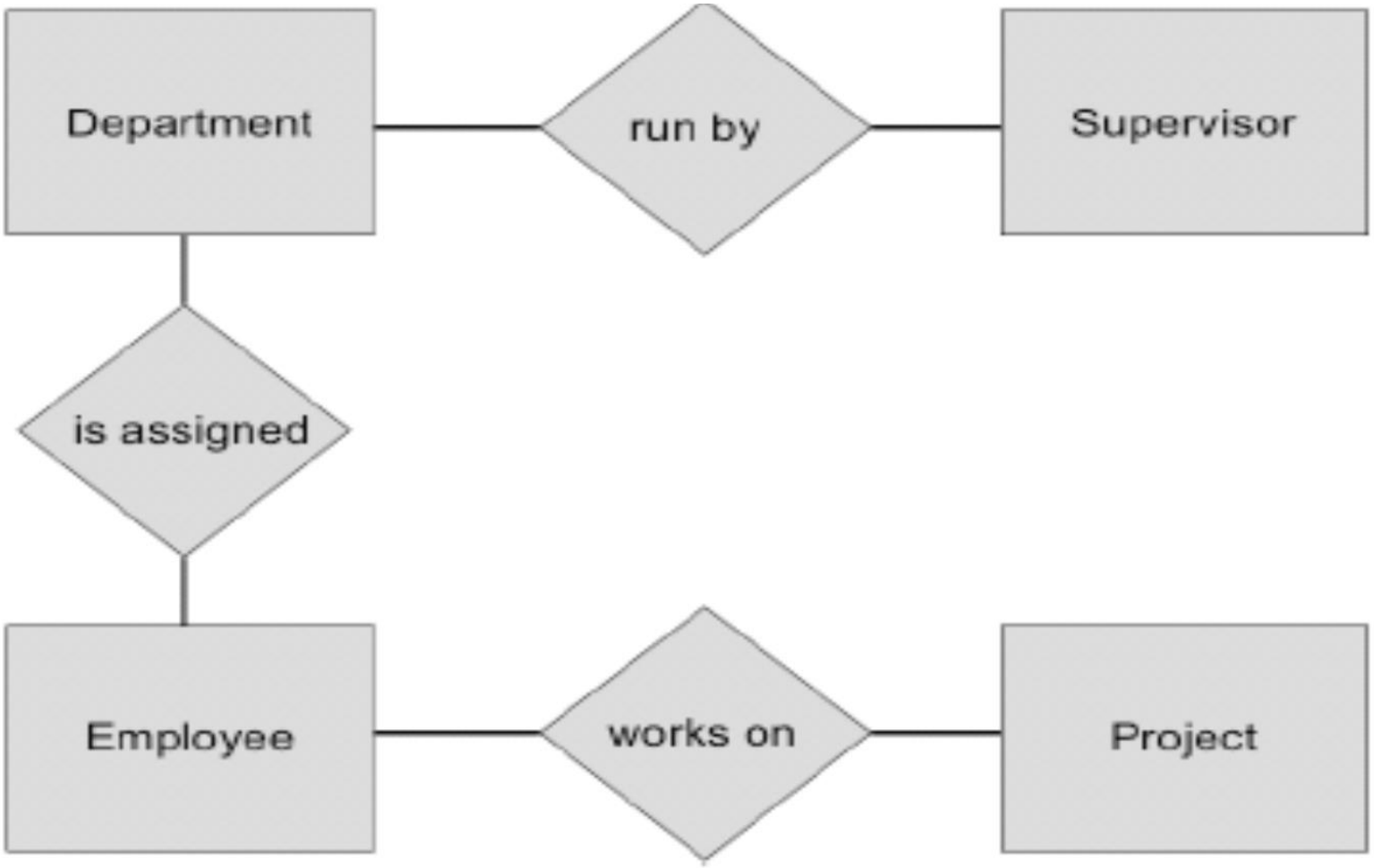




Drawing Rough ERD (Contd.)



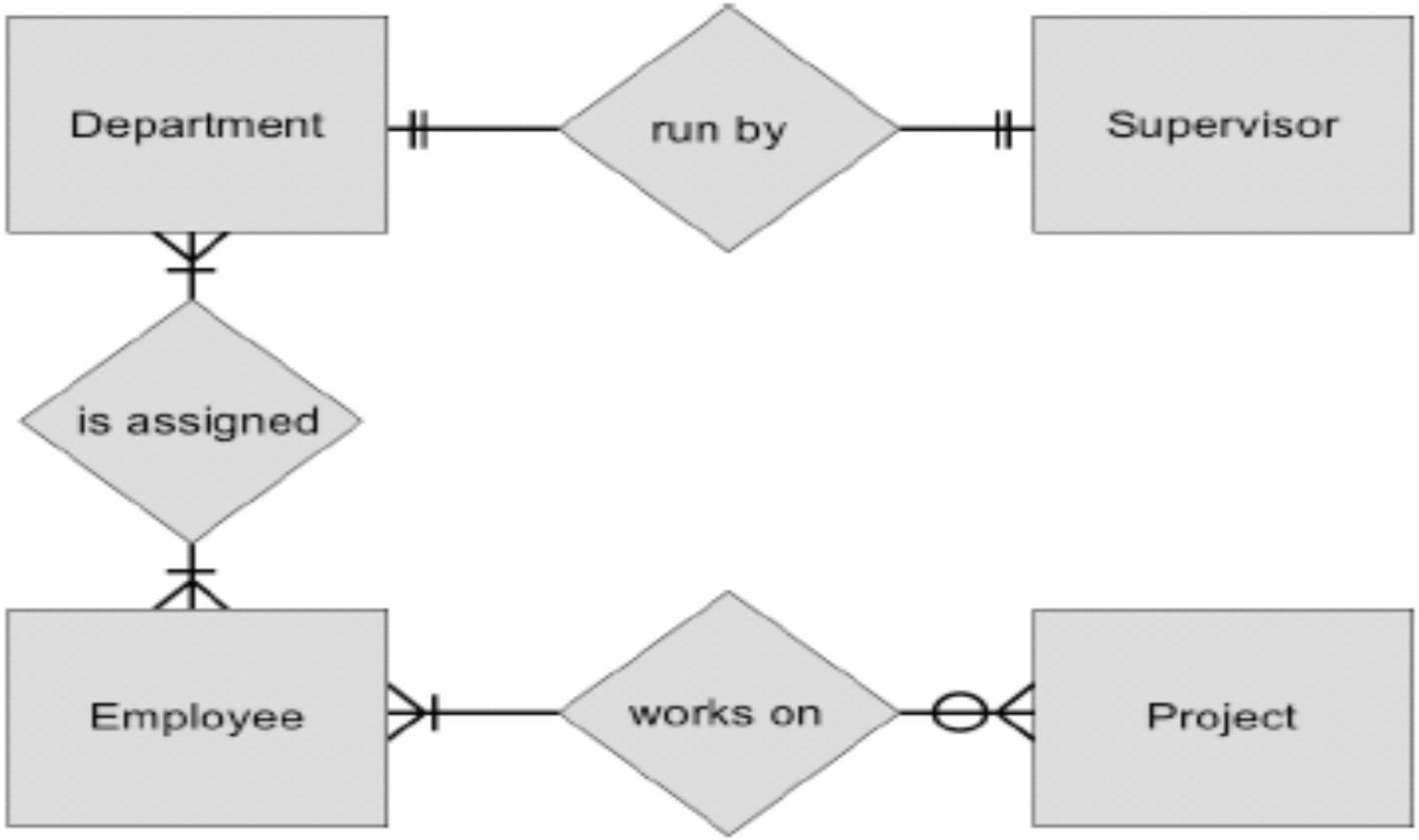
Drawing Rough ERD (Contd.)



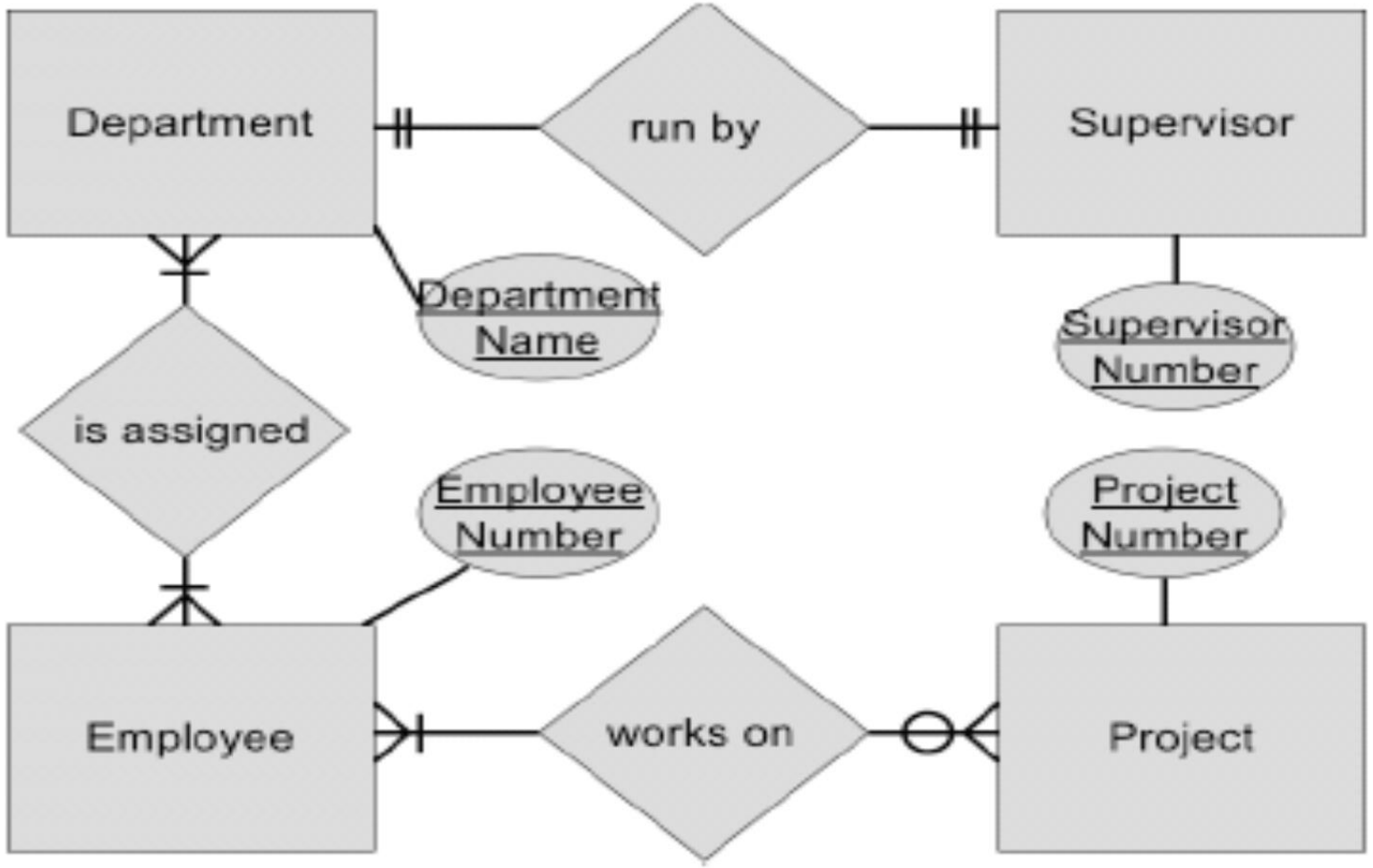
Fill in Cardinality

- Supervisor
  - Each department has one supervisor.
- Department
  - Each supervisor has one department.
  - Each employee can belong to one or more departments
- Employee
  - Each department must have one or more employees
  - Each project must have one or more employees
- Project
  - Each employee can have o or more projects.

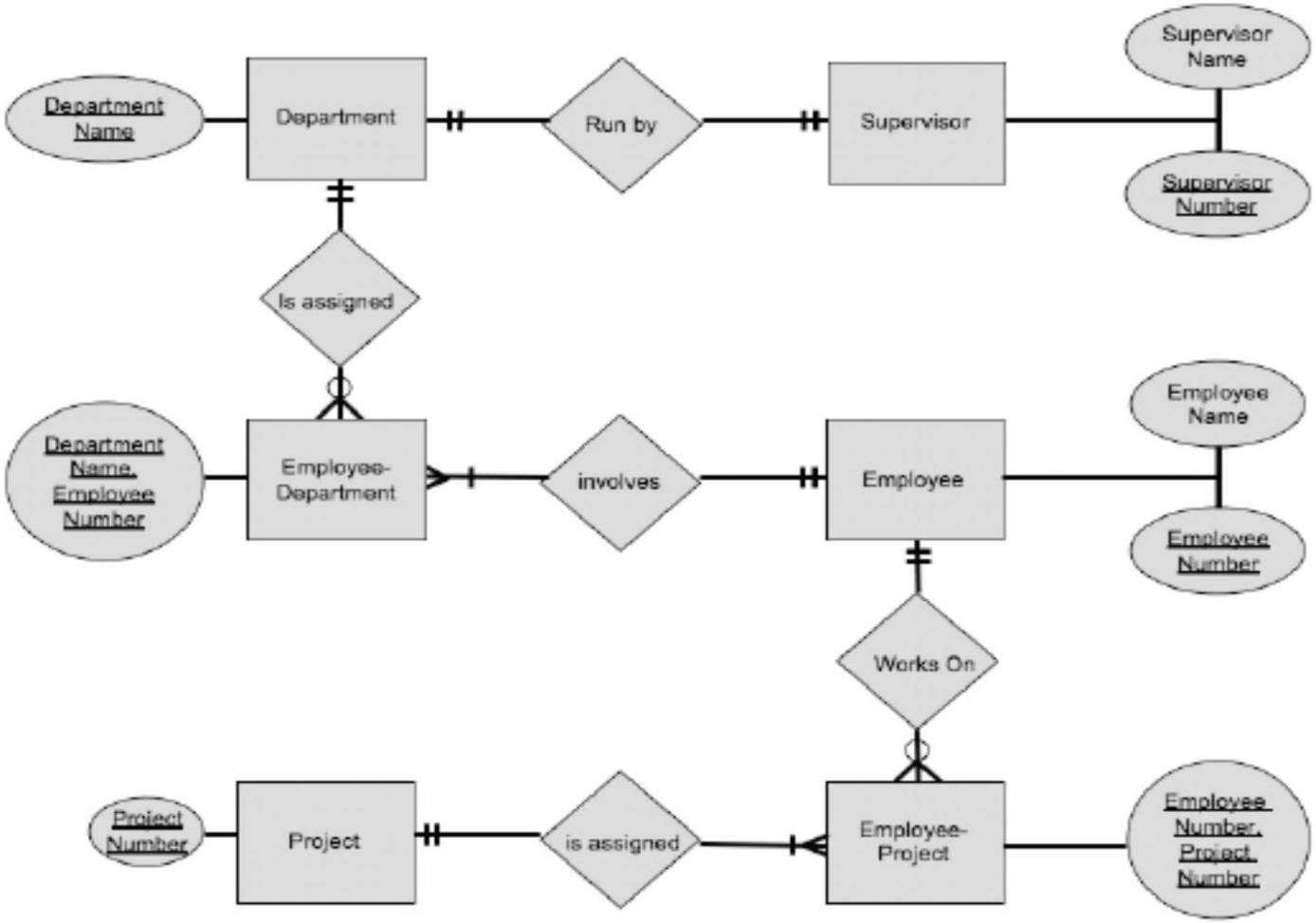
ERD with cardinality



Rough ERD Plus Primary Keys



Draw Fully Attributed ERD



Check ERD Results

- Look at your diagram from the point of view of a system owner or user. Is everything clear?
- Check through the Cardinality pairs.
- Also, look over the list of attributes associated with each entity to see if anything has been omitted.

Exercise

A company database needs to store information about employees (identified by ssn, with salary and phone as attributes), departments (identified by dno, with dname and budget as attributes), and children of employees (with name and age as attributes). Employees work in departments; each department is managed by an employee; a child must be identified uniquely by name when the parent (who is an employee; assume that only one parent works for the company) is known. We are not interested in information about a child once the parent leaves the company.