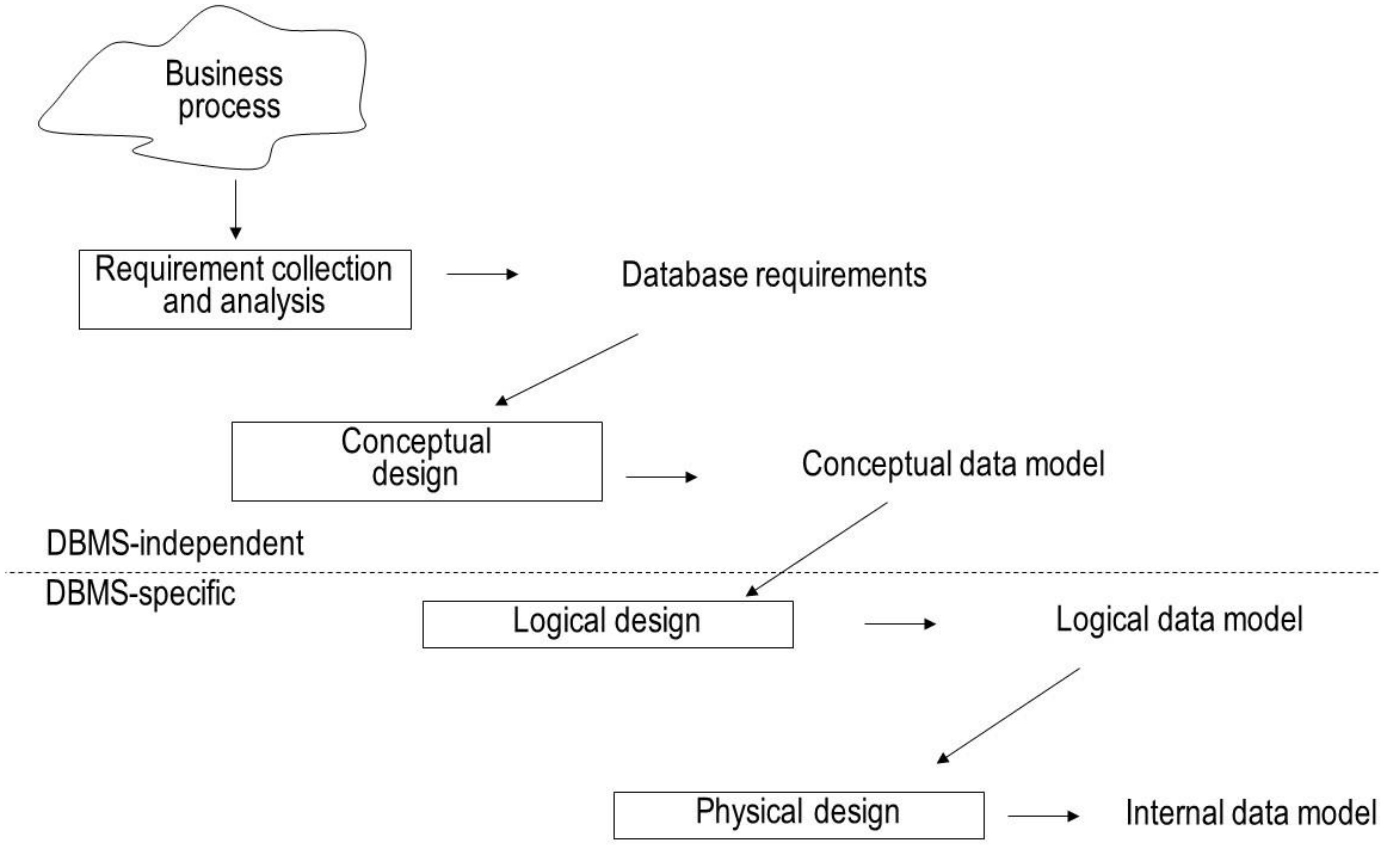
IBA

Database Design and Modelling

CS 341 Database Systems

Phases of Design Steps

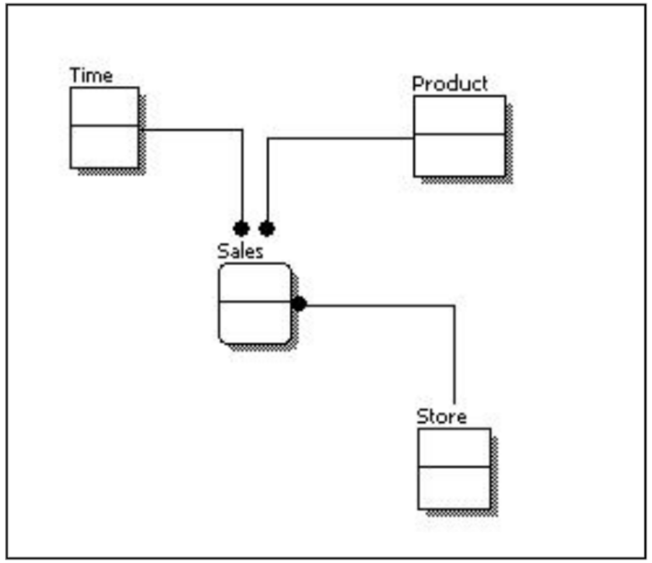


Data Model Levels

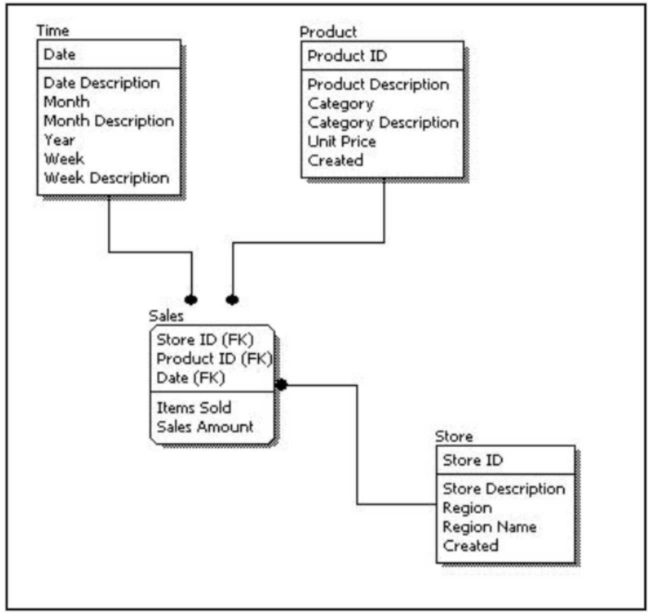
Feature	Conceptual	Logical	Physical
Entity Names	✓	✓	
Entity Relationships	✓	✓	
Attributes		✓	
Primary Keys		✓	✓
Foreign Keys		✓	✓
Table Names			✓
Column Names			✓
Column Data Types			✓

Data Model Levels

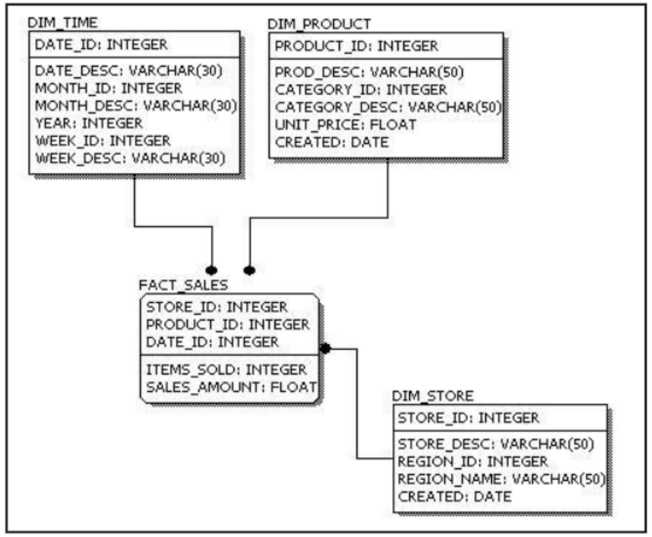
Conceptual Model Design



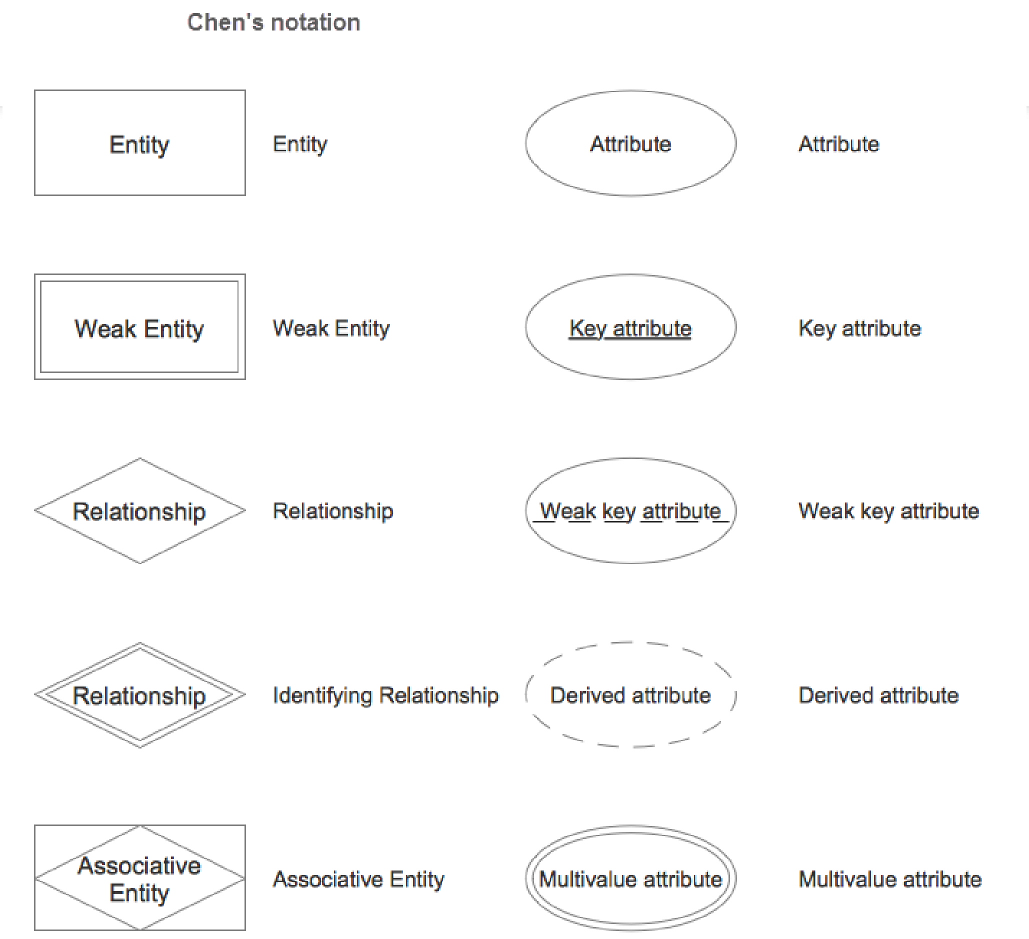
Logical Model Design

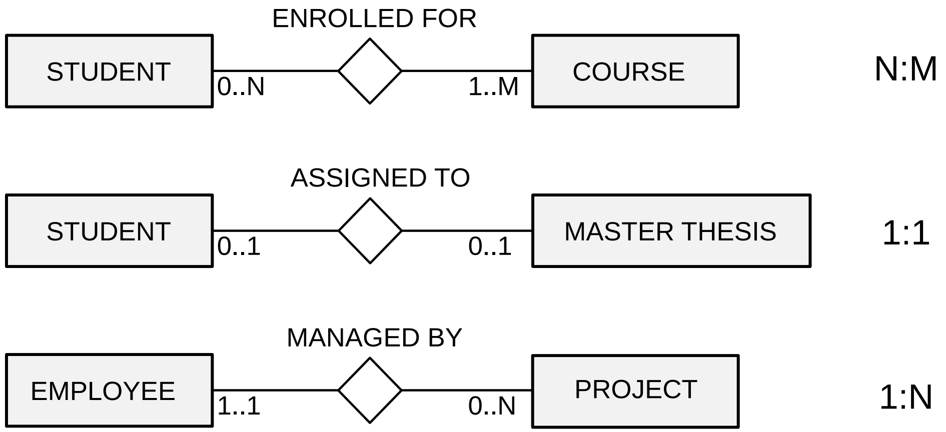


Physical Model Design

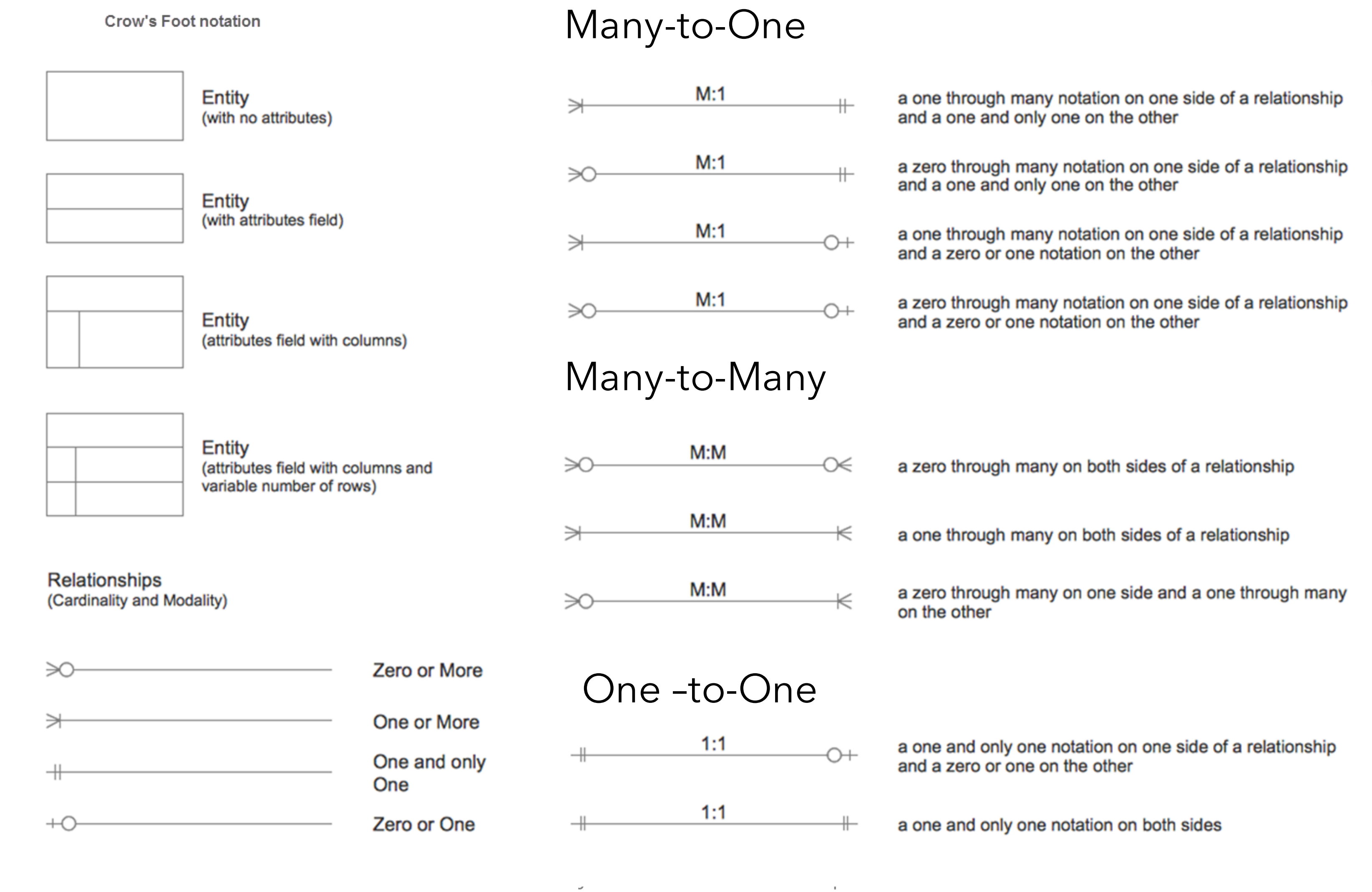


Chen’s Notation





Crow’s Foot Notation



Crow's Foot notation

Entity
(with no attributes)

Entity
(with attributes field)

Entity
(attributes field with columns)

Entity
(attributes field with columns and variable number of rows)

Relationships
(Cardinality and Modality)

Zero or More

One or More

One and only One

Zero or One

Many-to-One

M:1 — a one through many notation on one side of a relationship and a one and only one on the other

M:1 — a zero through many notation on one side of a relationship and a one and only one on the other

M:1 — a one through many notation on one side of a relationship and a zero or one notation on the other

M:1 — a zero through many notation on one side of a relationship and a zero or one notation on the other

Many-to-Many

M:M — a zero through many on both sides of a relationship

M:M — a one through many on both sides of a relationship

M:M — a zero through many on one side and a one through many on the other

One -to-One

1:1 — a one and only one notation on one side of a relationship and a zero or one on the other

1:1 — a one and only one notation on both sides

Concepts of the ER Model

Entity types

Attributes

Relationship types

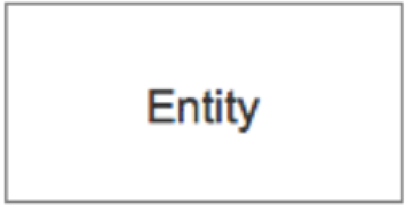
Entity Type

Entity type

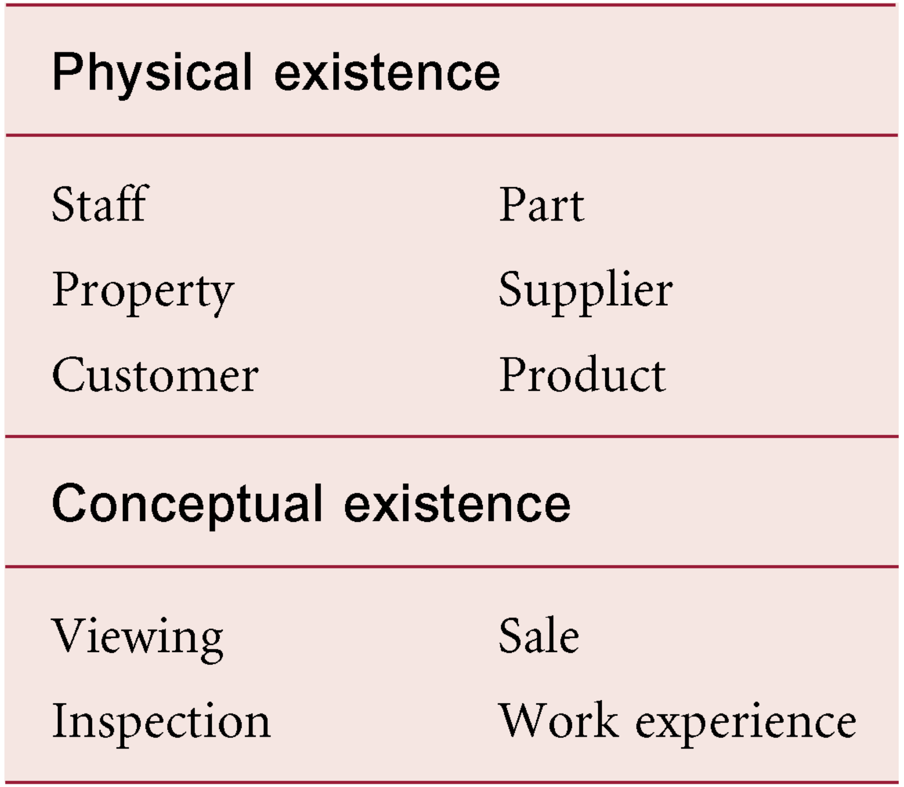
- Group of objects with same properties, identified by enterprise as having an independent existence.

Entity occurrence

- Uniquely identifiable object of an entity type.

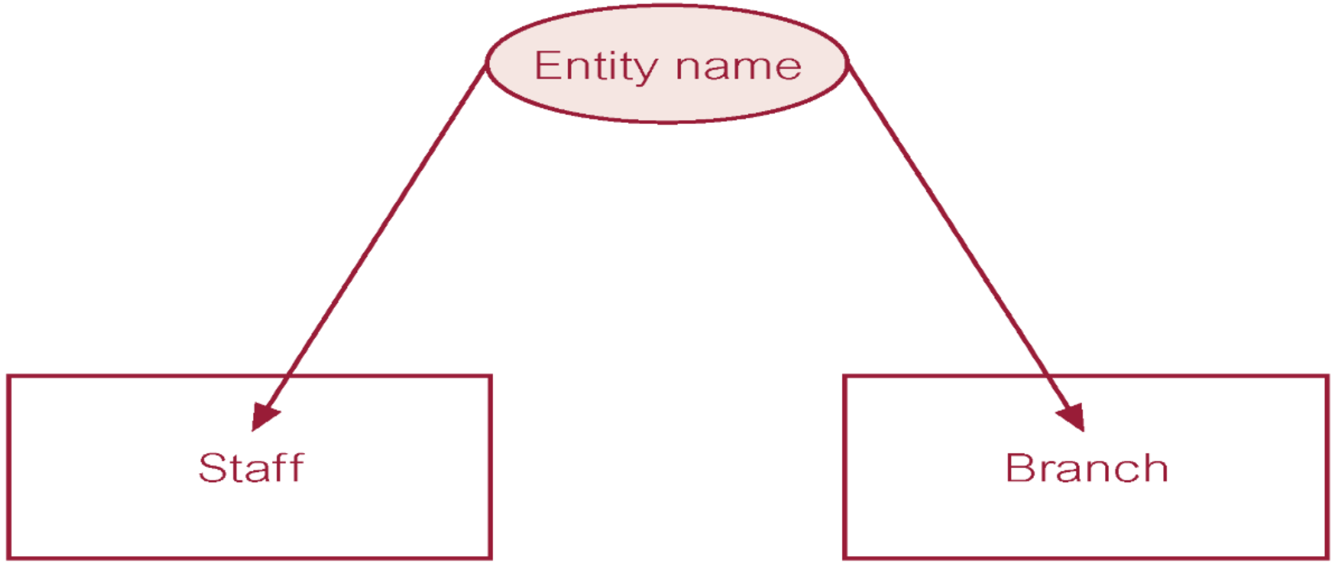


Examples of Entity Types



Physical existence	
Staff	Part
Property	Supplier
Customer	Product
Conceptual existence	
Viewing	Sale
Inspection	Work experience

ER diagram of Staff and Branch entity types



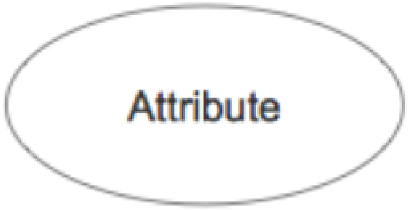
Attributes

Attribute

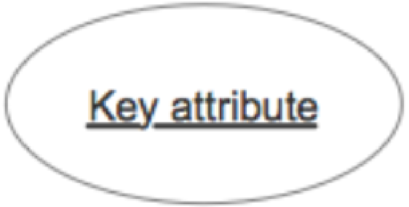
- Property of an entity or a relationship type.

Attribute Domain

- Set of allowable values for one or more attributes.
- Not displayed in an ER model



Key Attribute Types

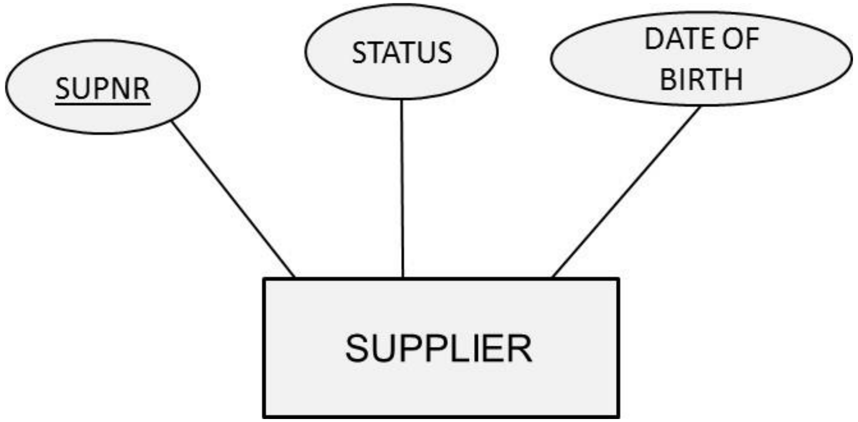


A key attribute type is an attribute type whose values are distinct for each individual entity

- Examples: supplier number, product number, social security number

A key attribute type can also be a combination of attribute types

- Example: combination of flight number and departure date



Attributes

Simple Attribute

- Attribute composed of a single component with an independent existence.

Composite Attribute

- Attribute composed of multiple components, each with an independent existence.

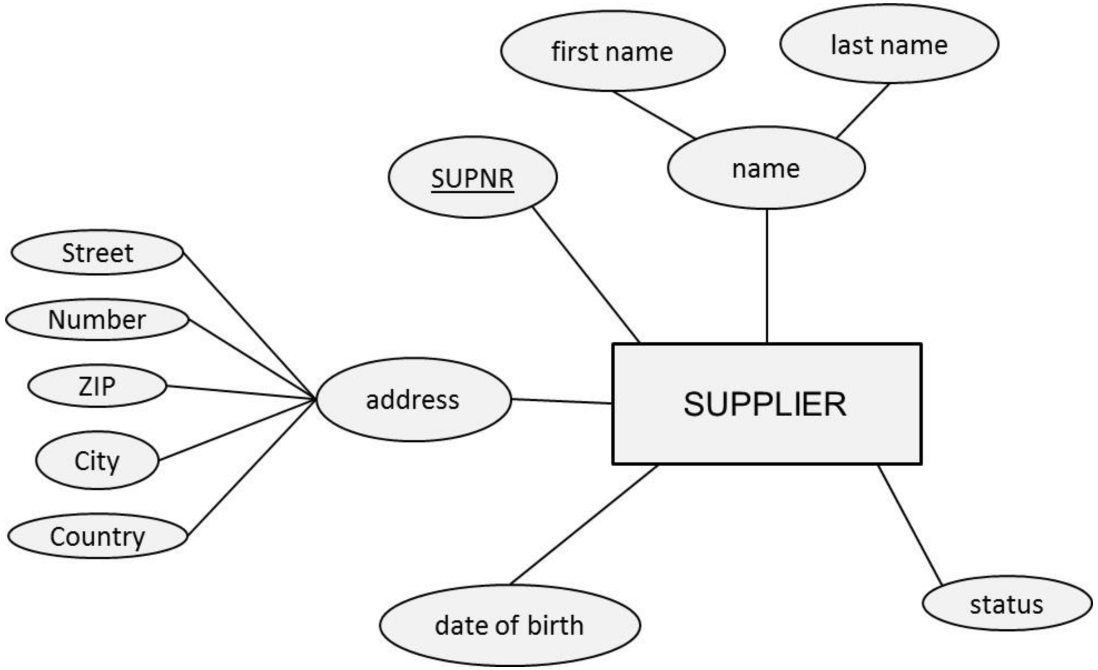
Simple versus Composite Attribute Types

A **simple or atomic** attribute type cannot be further divided into parts

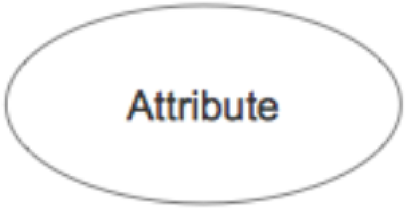
Examples: supplier number, supplier status

A **composite** attribute type is an attribute type that can be decomposed into other meaningful attribute types

Examples: address, name

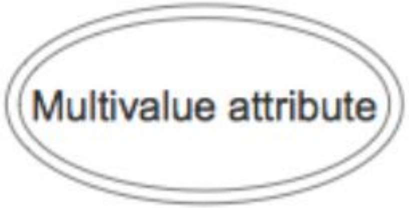


Attributes



Single-valued Attribute

- Attribute that holds a single value for each occurrence of an entity type.



Multi-valued Attribute

- Attribute that holds multiple values for each occurrence of an entity type.

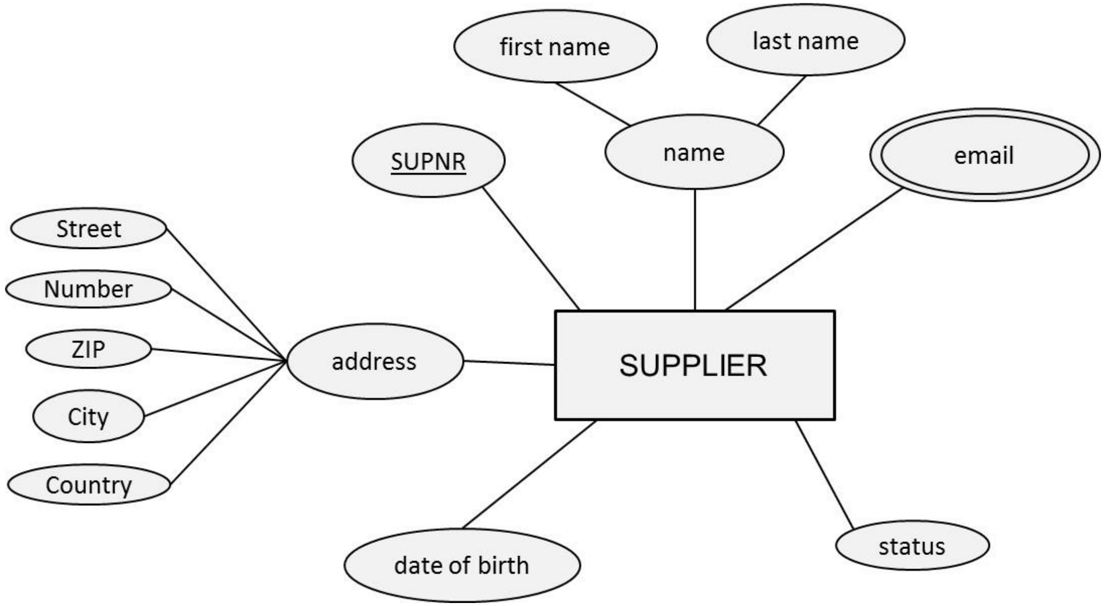
Single-Valued versus Multi-Valued Attribute Types

A **single-valued** attribute type has only one value for a particular entity

- Examples: supplier number, supplier name

A **multi-valued** attribute type is an attribute type that can have multiple values

- Example: email address

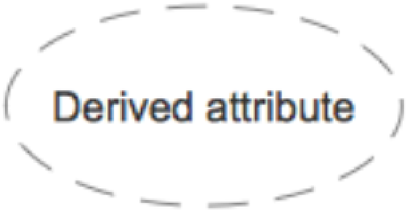


Note: this is chen notation, the crow's foot does not distinguish this.

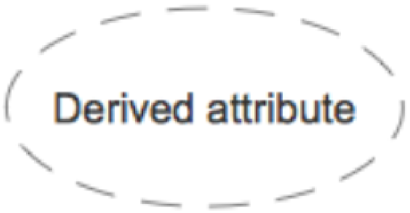
Attributes

Derived Attribute

- Attribute that represents a value that is derivable from value of a related attribute, or set of attributes, not necessarily in the same entity type.

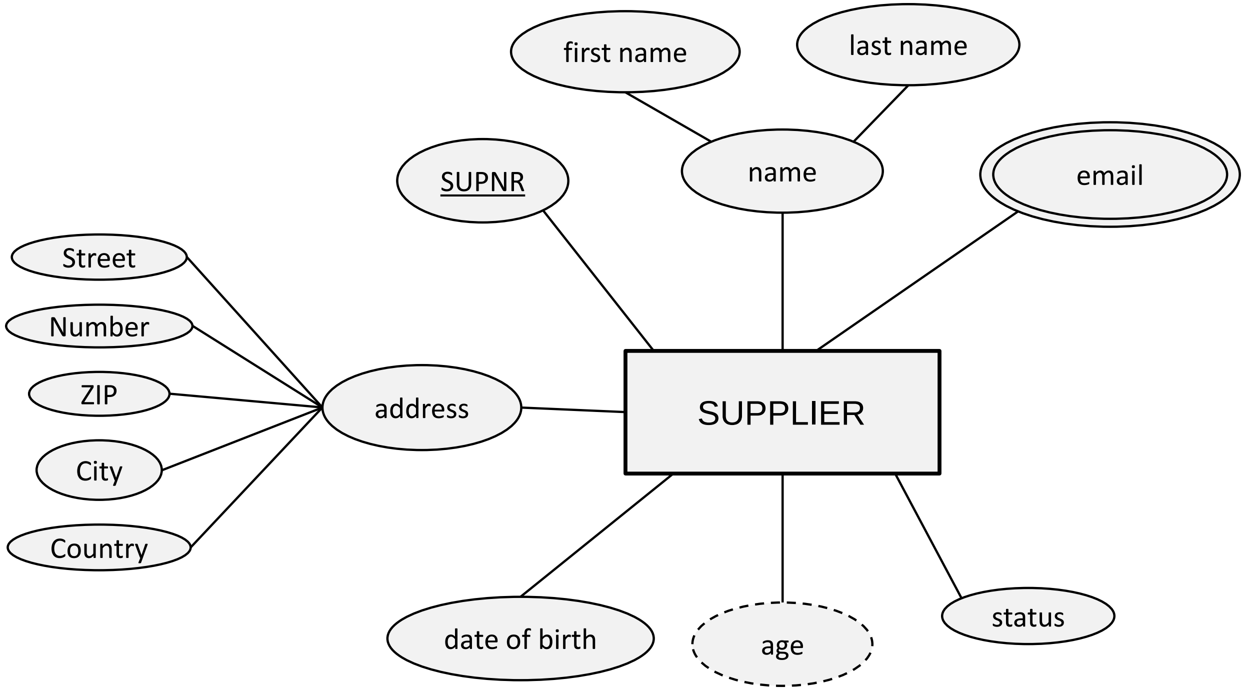


Derived Attribute Type



A derived attribute type is an attribute type which can be derived from another attribute type

- Example: age



Keys

Candidate Key

- Minimal set of attributes that uniquely identifies each occurrence of an entity type.

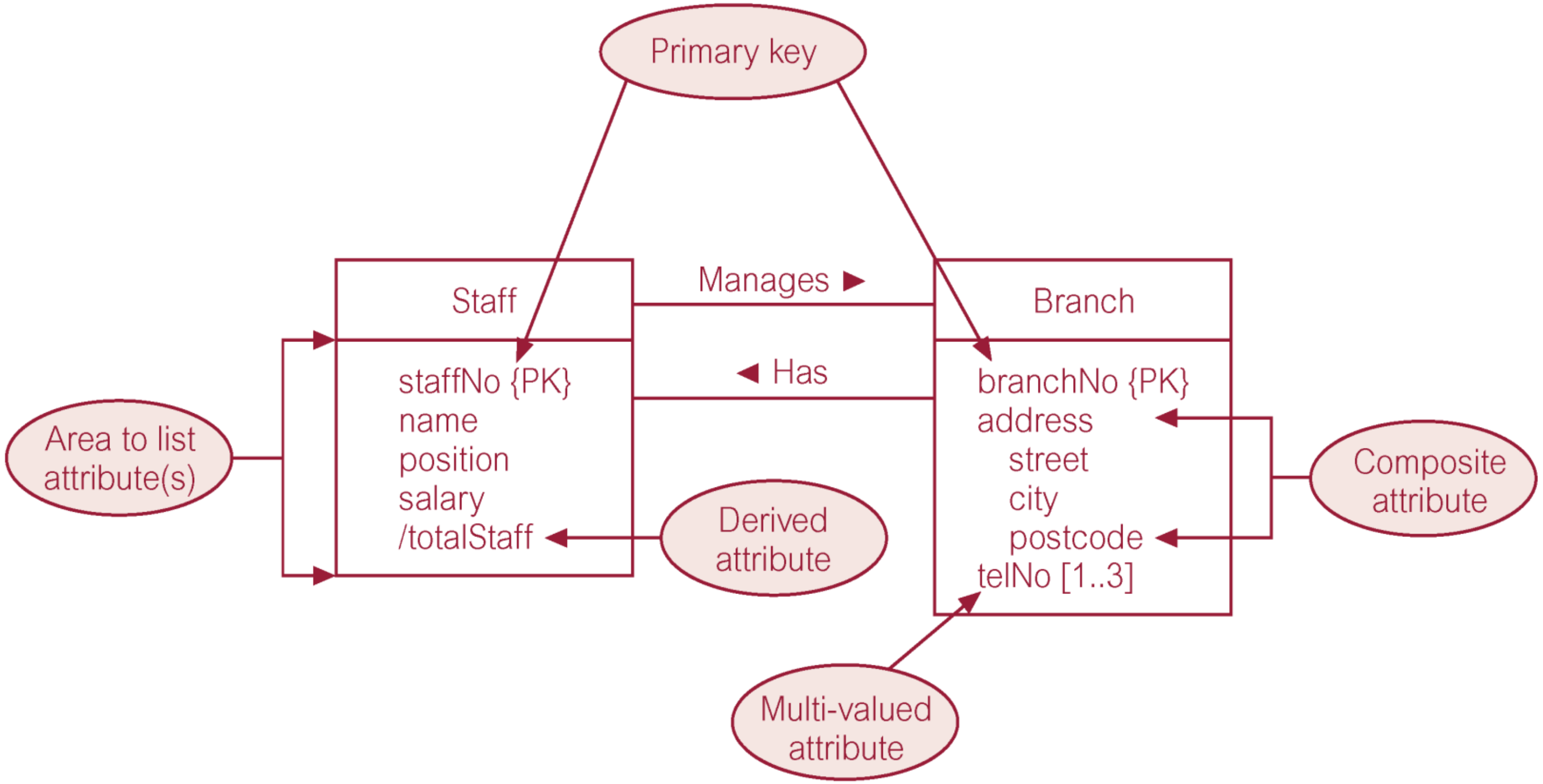
Primary Key

- Candidate key selected to uniquely identify each occurrence of an entity type.

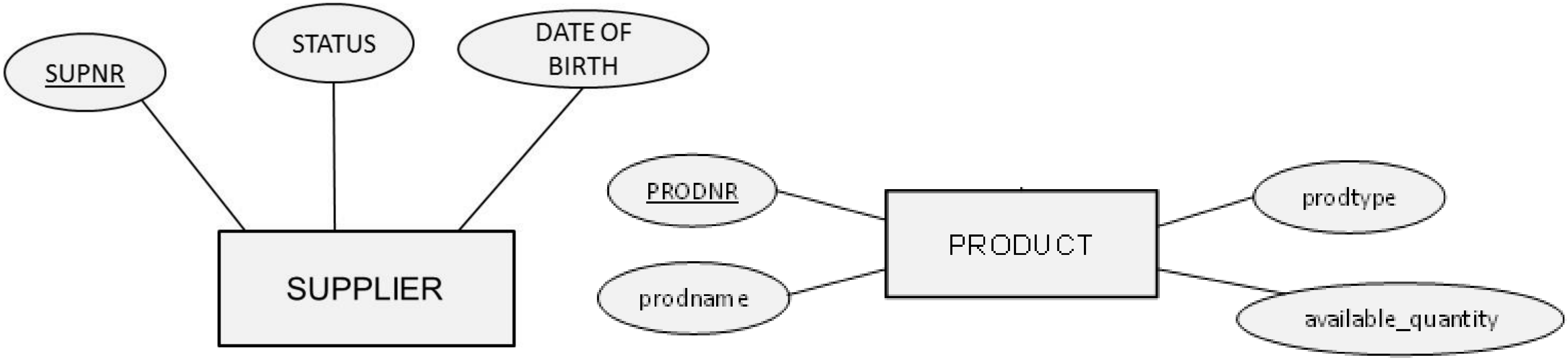
Composite Key

- A candidate key that consists of two or more attributes.

ER diagram of Staff and Branch entities and their attributes



ER diagram of Supplier and Products entity types



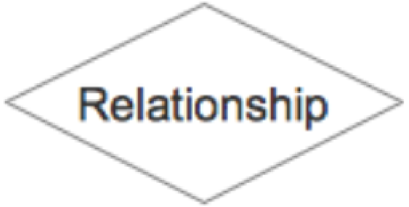
Relationship Types

Relationship type

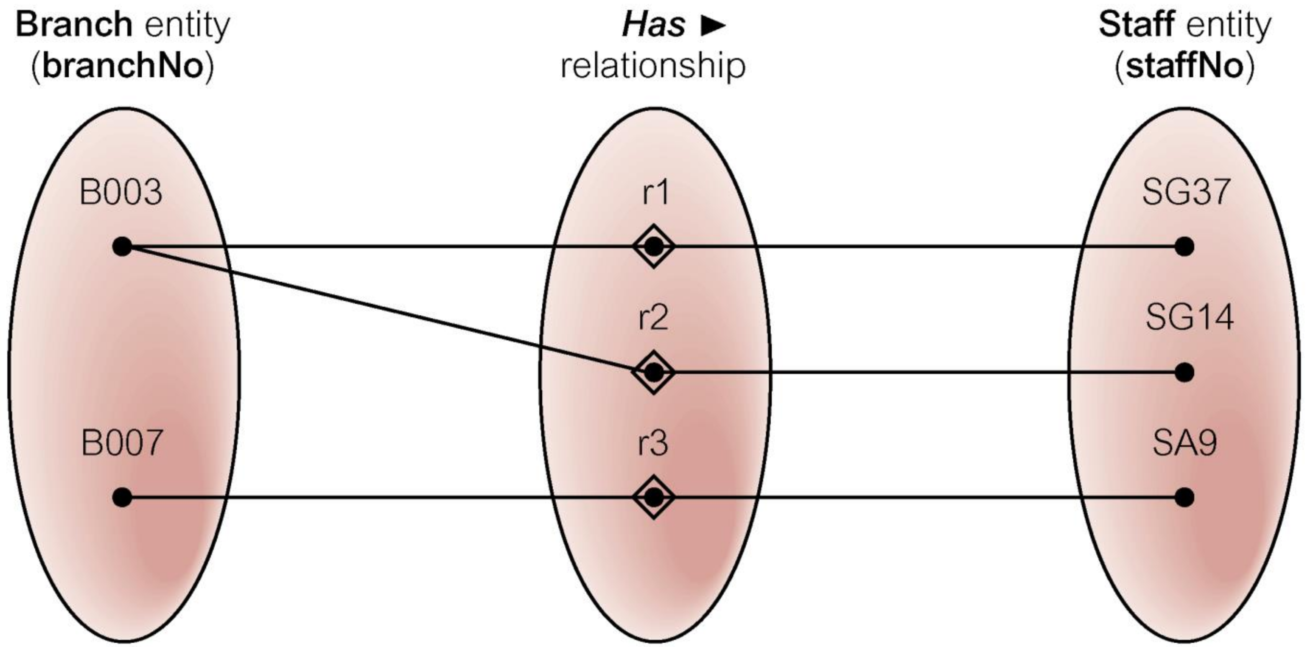
- Set of meaningful associations among instances of one, two or more entity types.

Relationship occurrence

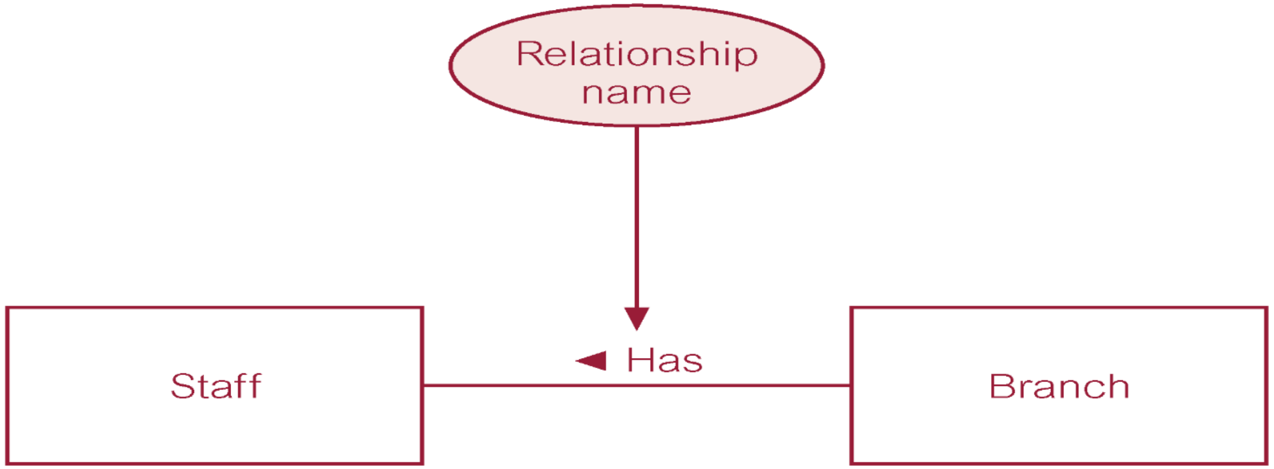
- Uniquely identifiable association, which includes one occurrence from each participating entity type.



Semantic net of Has relationship type

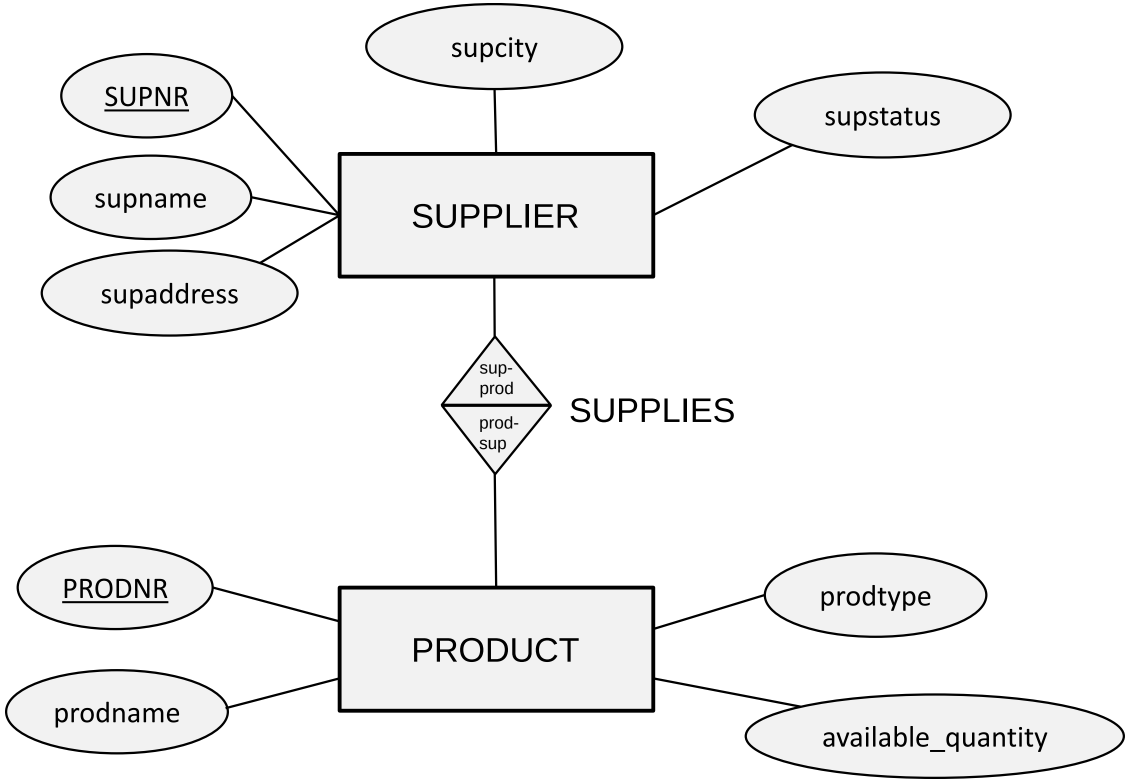


ER diagram of Branch Has Staff relationship



'Branch has staff'

ER diagram of Supplier *supplies* Product relationship



Degree of a Relationship

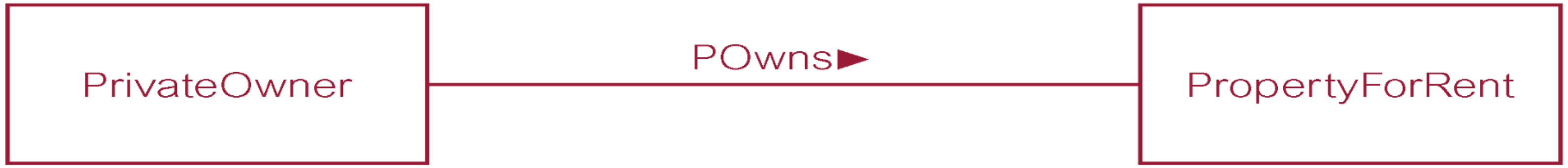
- **Degree of a Relationship**

Number of participating entities in relationship.

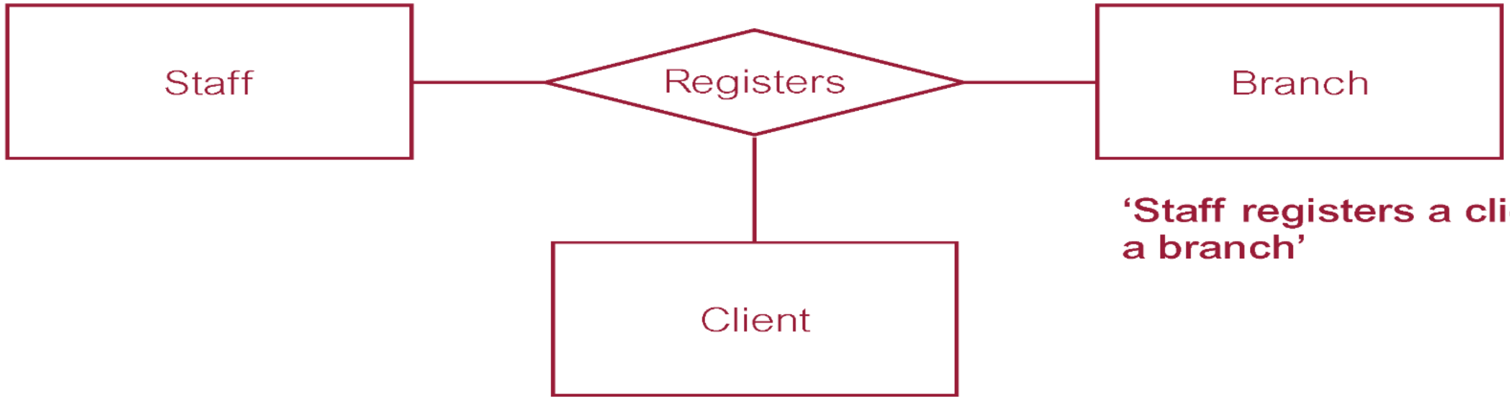
  - Two is Binary
  - Three is Ternary
  - Four is Quaternary

Binary relationship called *POwns*

‘Private owner owns property for rent’

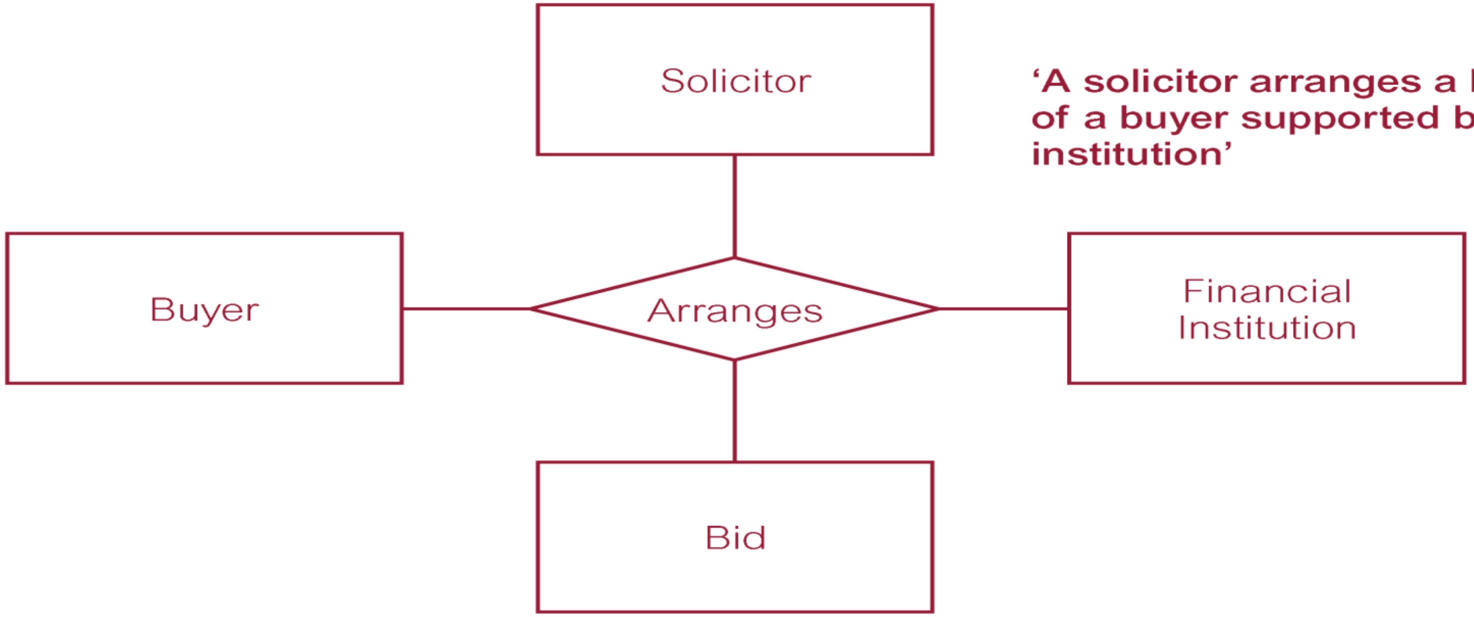


Ternary relationship called *Registers*



'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

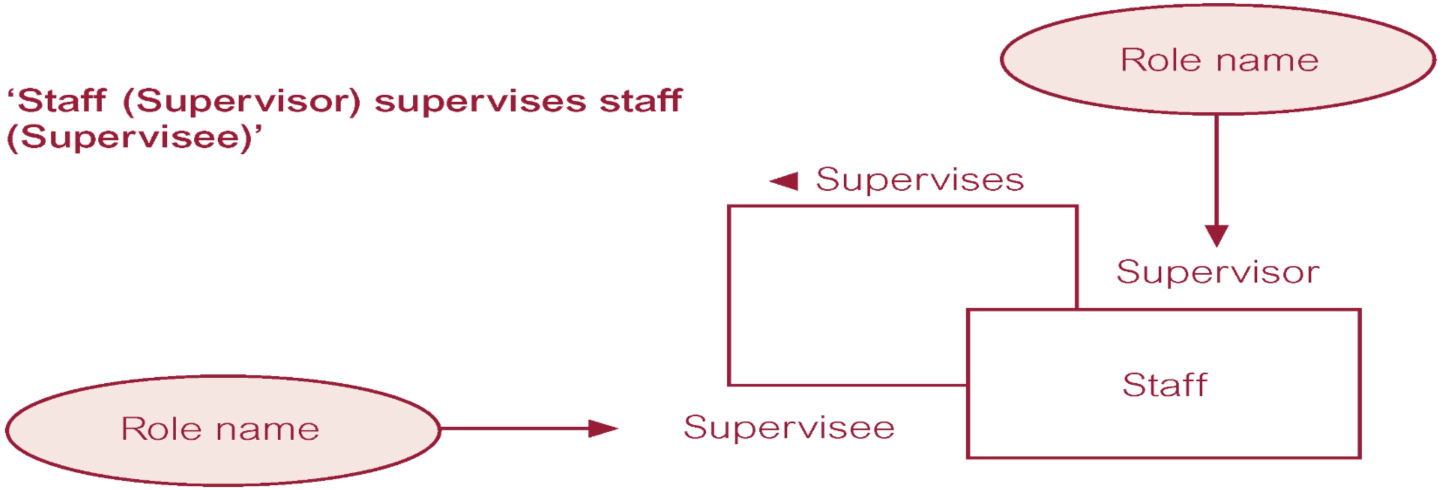
Recursive Relationship

- Relationship type where same entity type participates more than once in different roles.

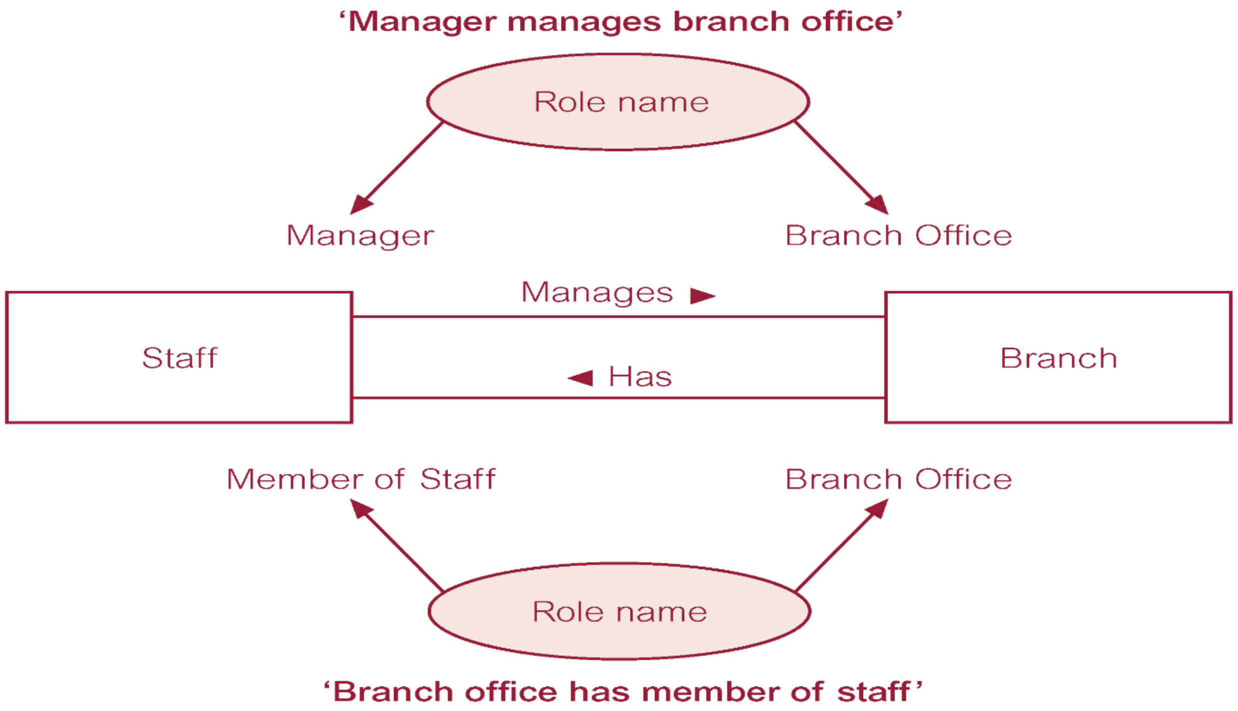
- Relationships may be given role names to indicate purpose that each participating entity type plays in a relationship.

Recursive relationship called *Supervises* with role names

'Staff (Supervisor) supervises staff (Supervisee)'



Entities associated through two distinct relationships with role names



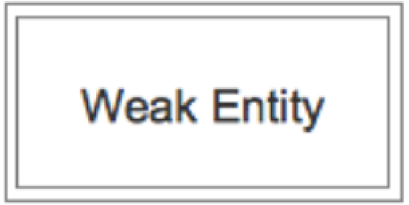
Entity Type

Strong Entity Type

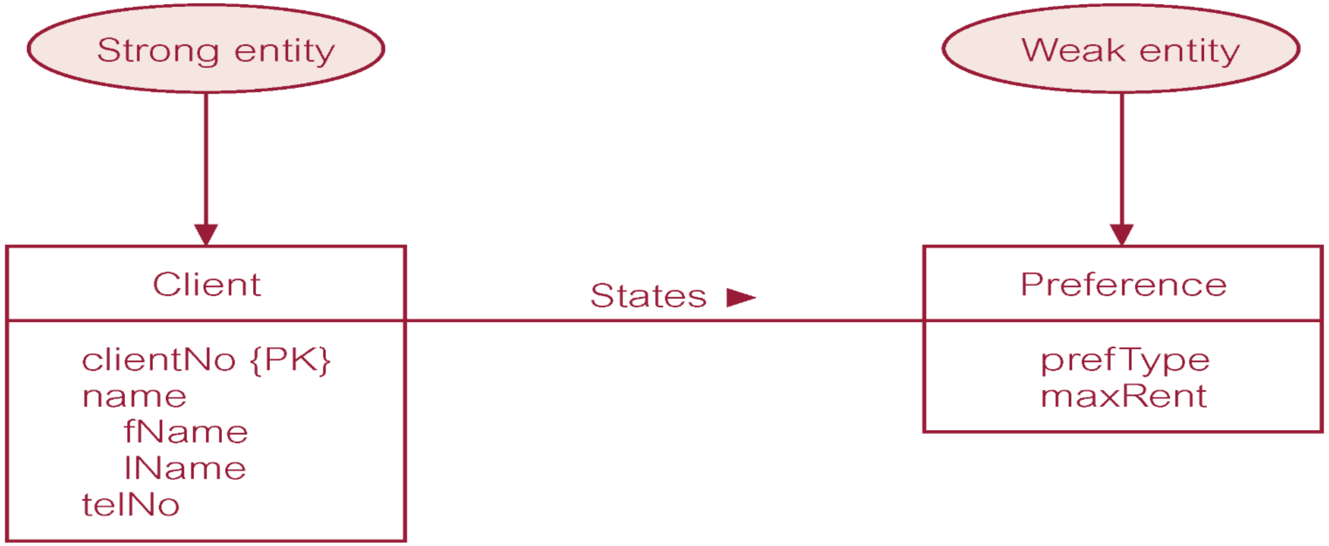
- Entity type that is not existence-dependent on some other entity type.

Weak Entity Type

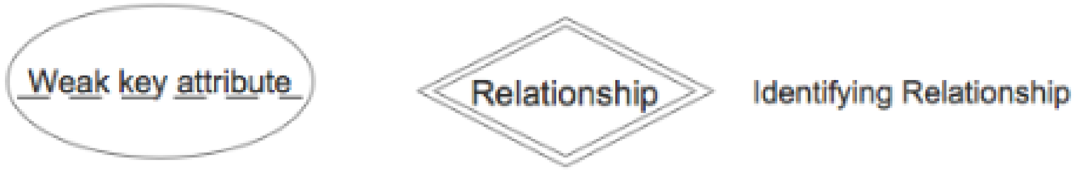
- Entity type that is existence-dependent on some other entity type.

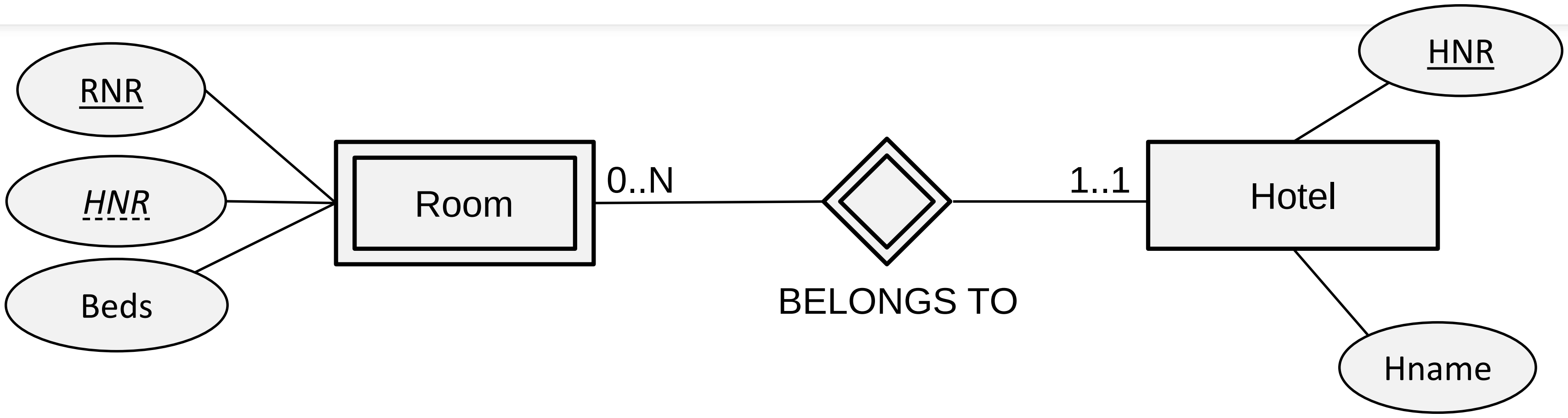


Strong entity type called Client and weak entity type called Preference



Weak Entity Type

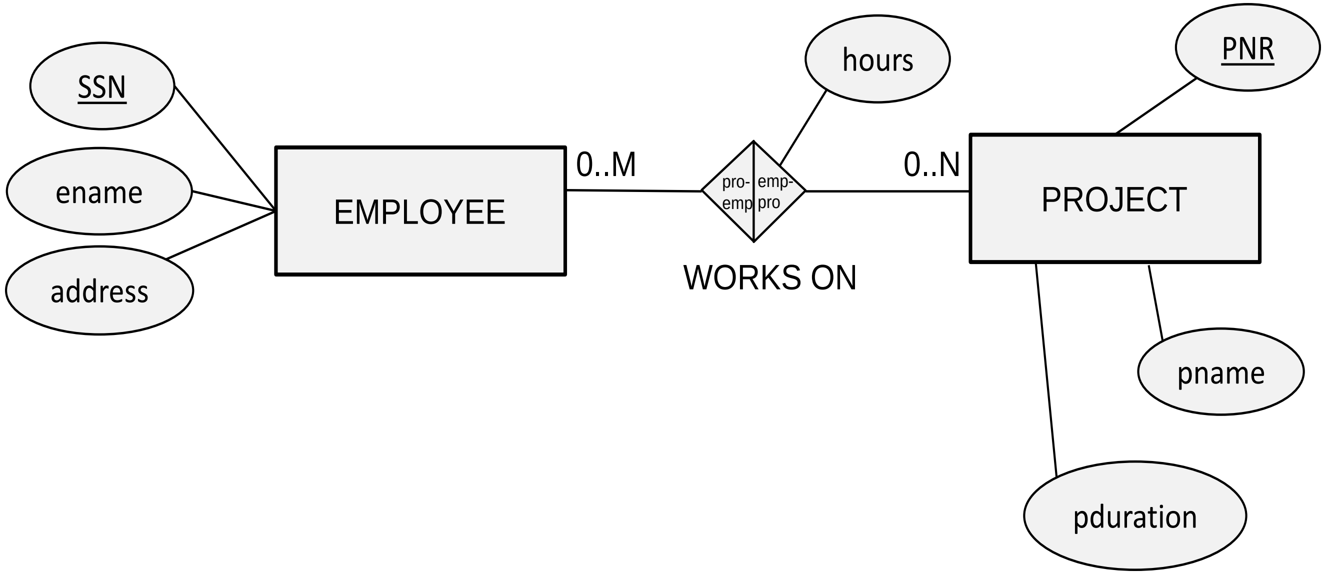




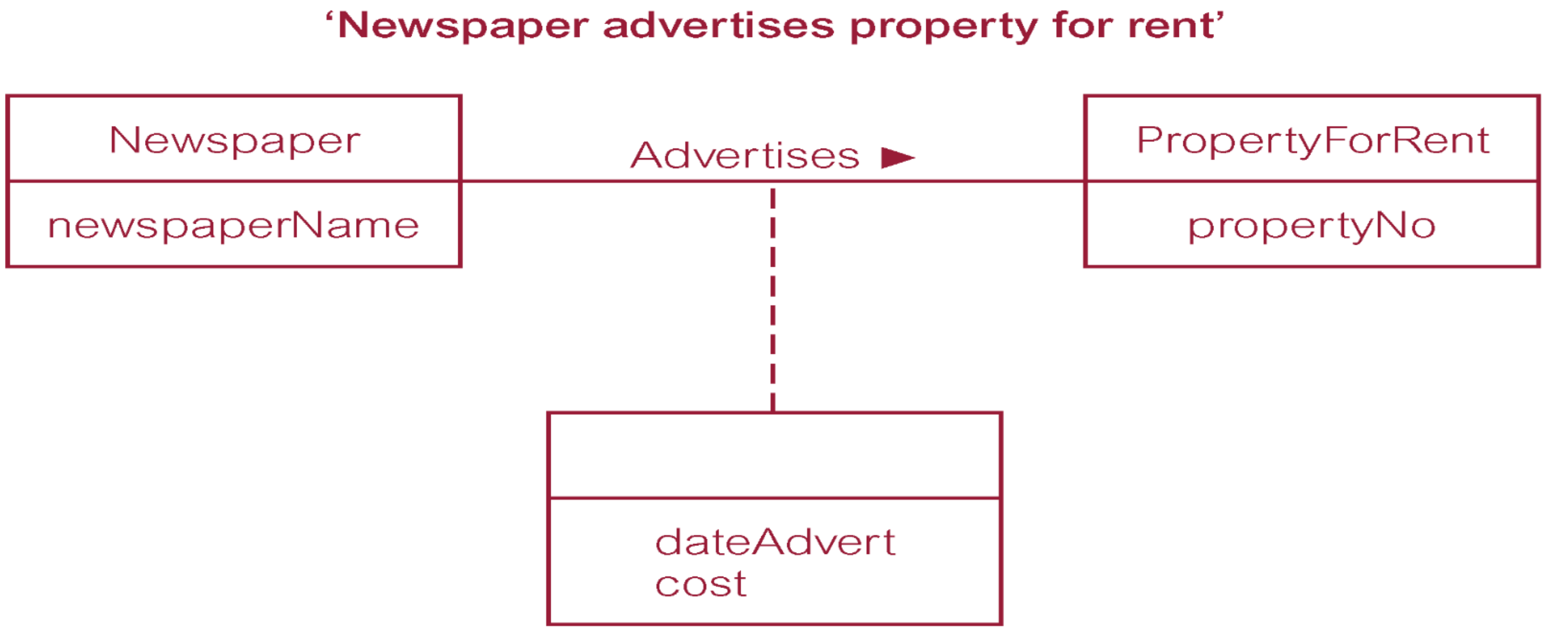
- A strong entity type is an entity type that has a key attribute type
- A weak entity type is an entity type that does not have a key attribute type of its own
    - related to owner entity type from which it borrows an attribute type to make up a key attribute type

Relationship Attribute Types

- Relationship type can also have attribute types
- These attribute types can be migrated to one of the participating entity types in case of a 1:1 or 1:N relationship type



Relationship called *Advertises* with attributes



Structural Constraints

- Main type of constraint on relationships is called multiplicity.

- **Multiplicity** - number (or range) of possible occurrences of an entity type that may relate to a single occurrence of an associated entity type through a particular relationship.

- Represents **policies** (called **business rules**) established by user or company.

Structural Constraints

The most common degree for relationships is binary.

Binary relationships are generally referred to as being:

- one-to-one (1:1)
- one-to-many (1:*) or (1:M)
- many-to-many (*:*) or (M:M)

Summary of multiplicity constraints

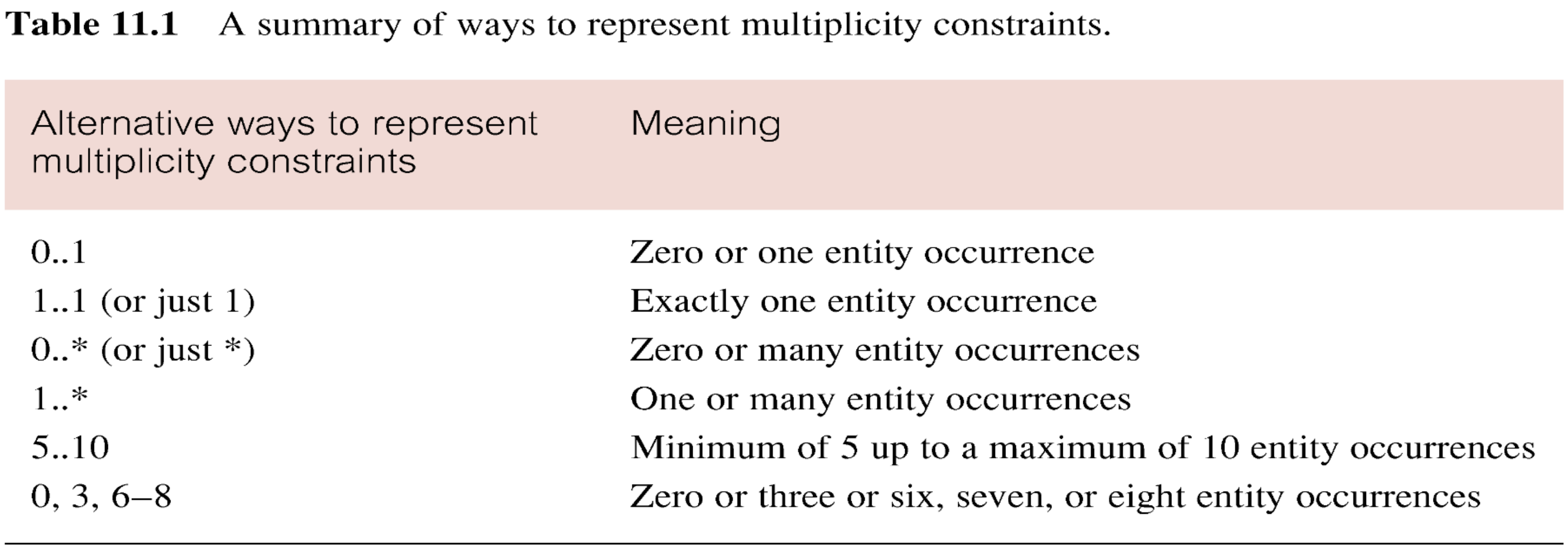
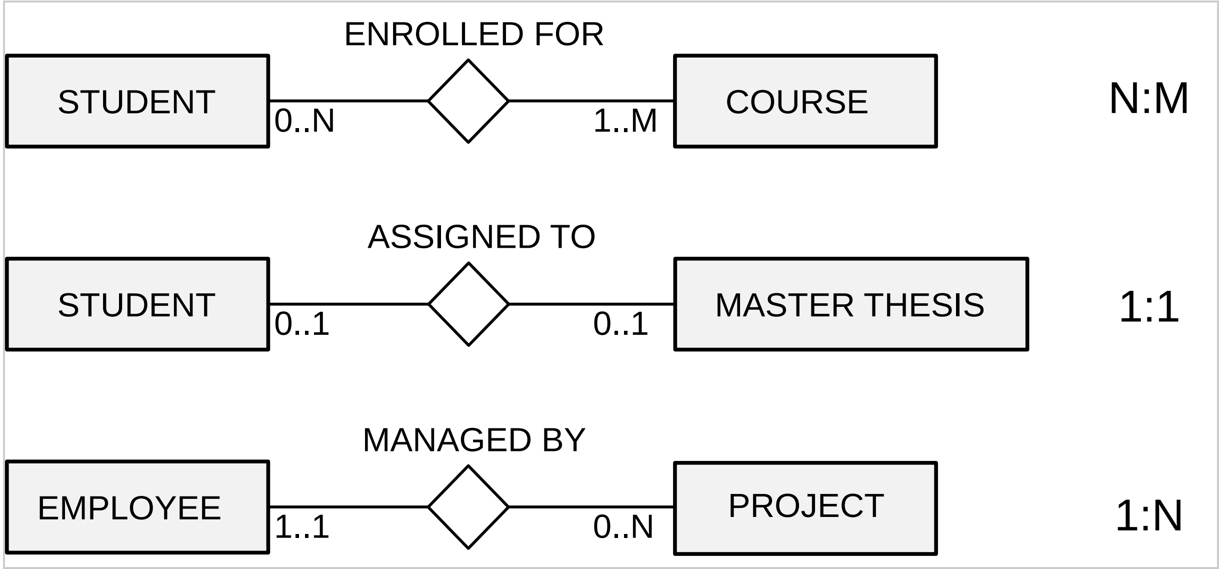


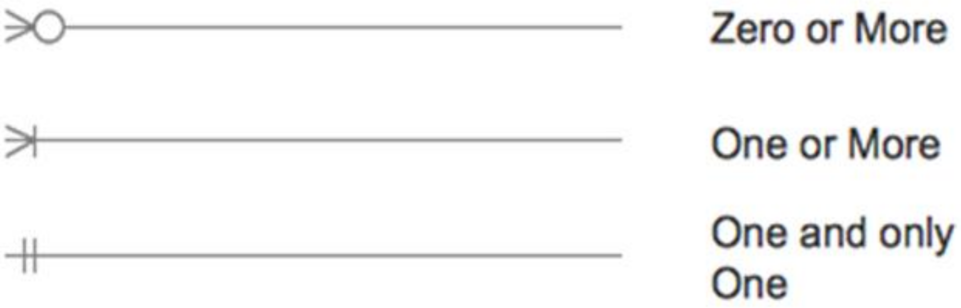
Table 11.1 A summary of ways to represent multiplicity constraints.

Alternative ways to represent multiplicity constraints	Meaning
0..1	Zero or one entity occurrence
1..1 (or just 1)	Exactly one entity occurrence
0..* (or just *)	Zero or many entity occurrences
1..*	One or many entity occurrences
5..10	Minimum of 5 up to a maximum of 10 entity occurrences
0, 3, 6–8	Zero or three or six, seven, or eight entity occurrences

Multiplicity

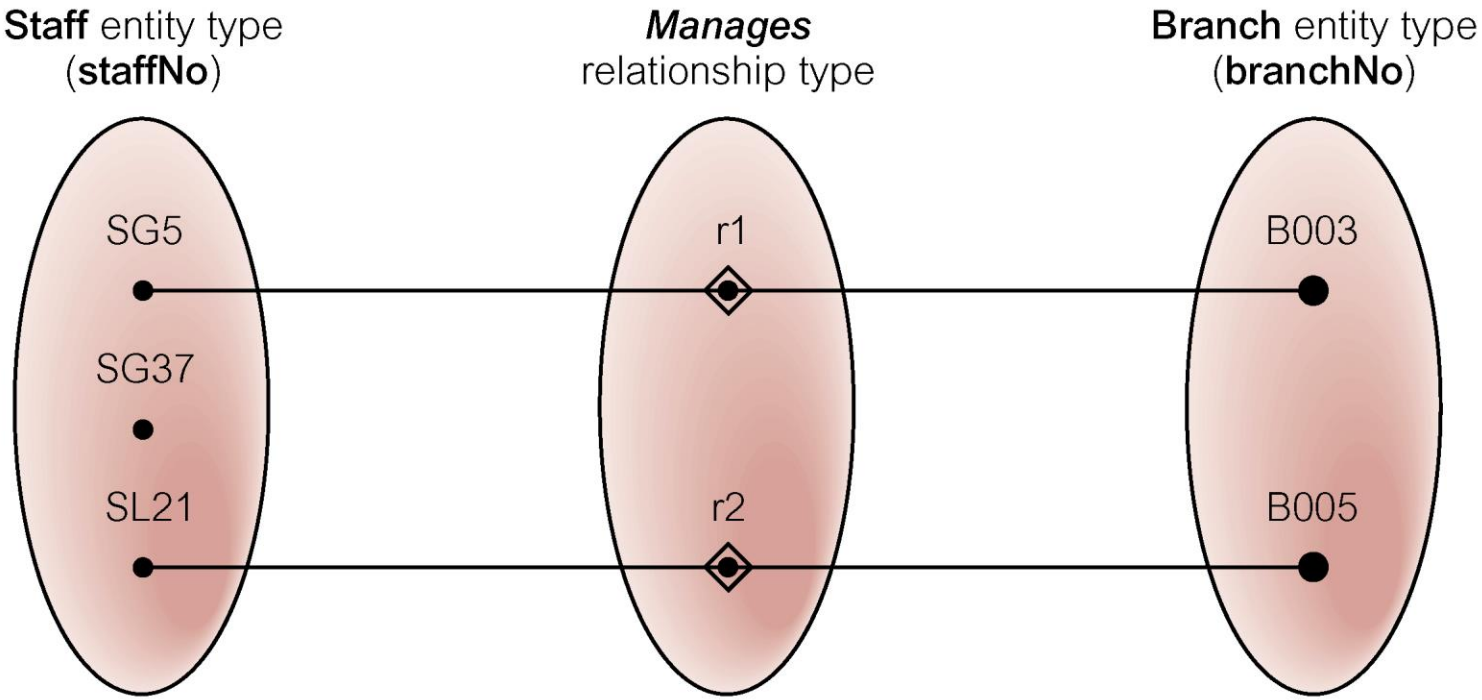
Chen's VS Crow's Foot



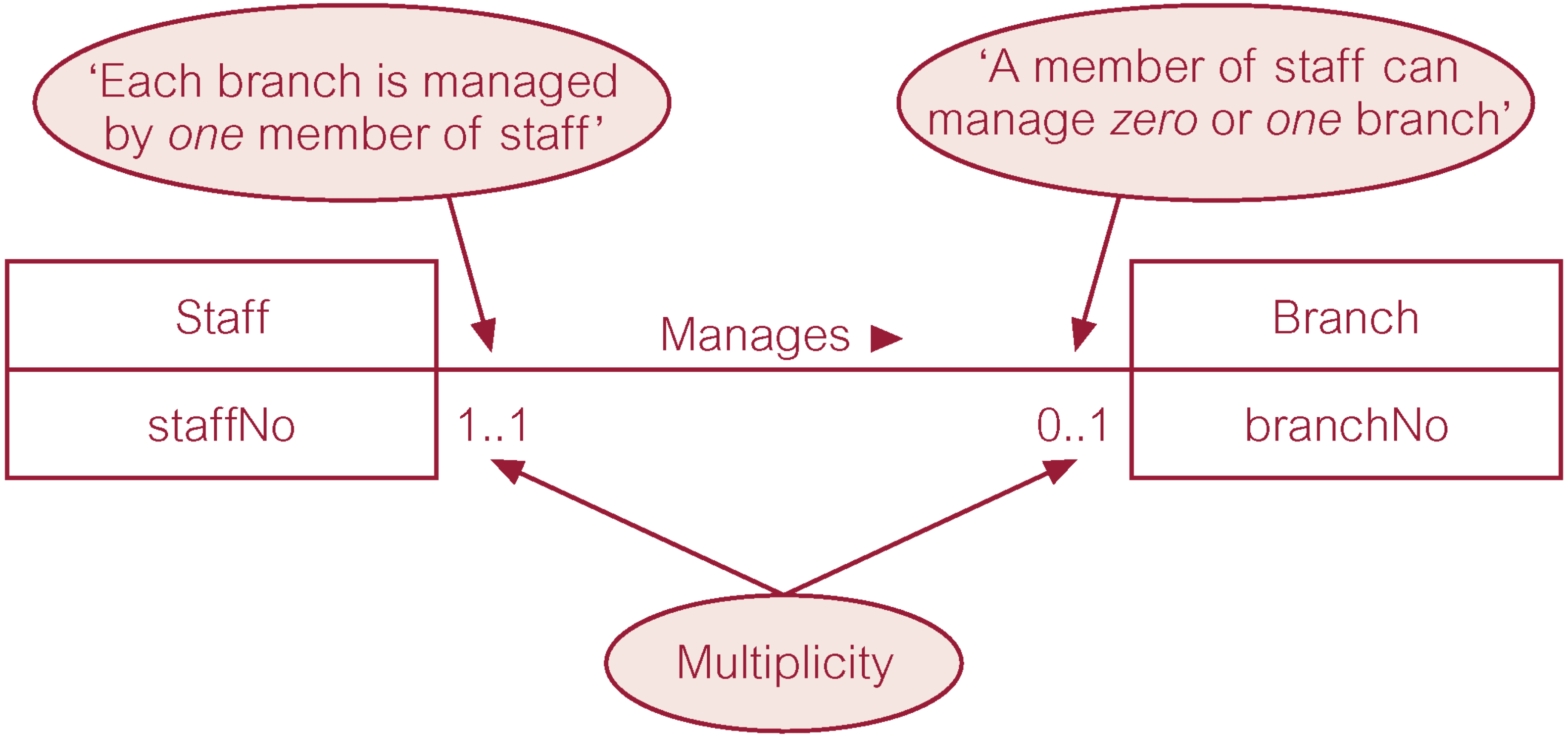


Session 02

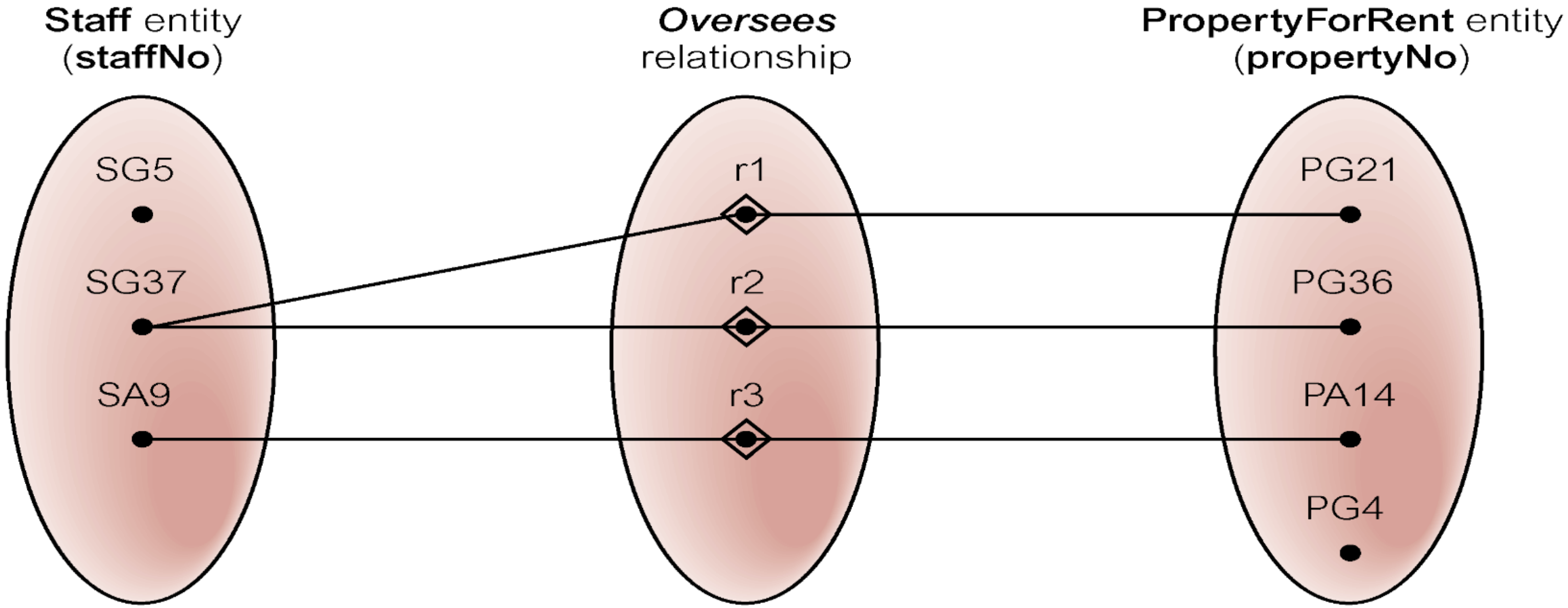
Semantic net of Staff *Manages* Branch relationship type



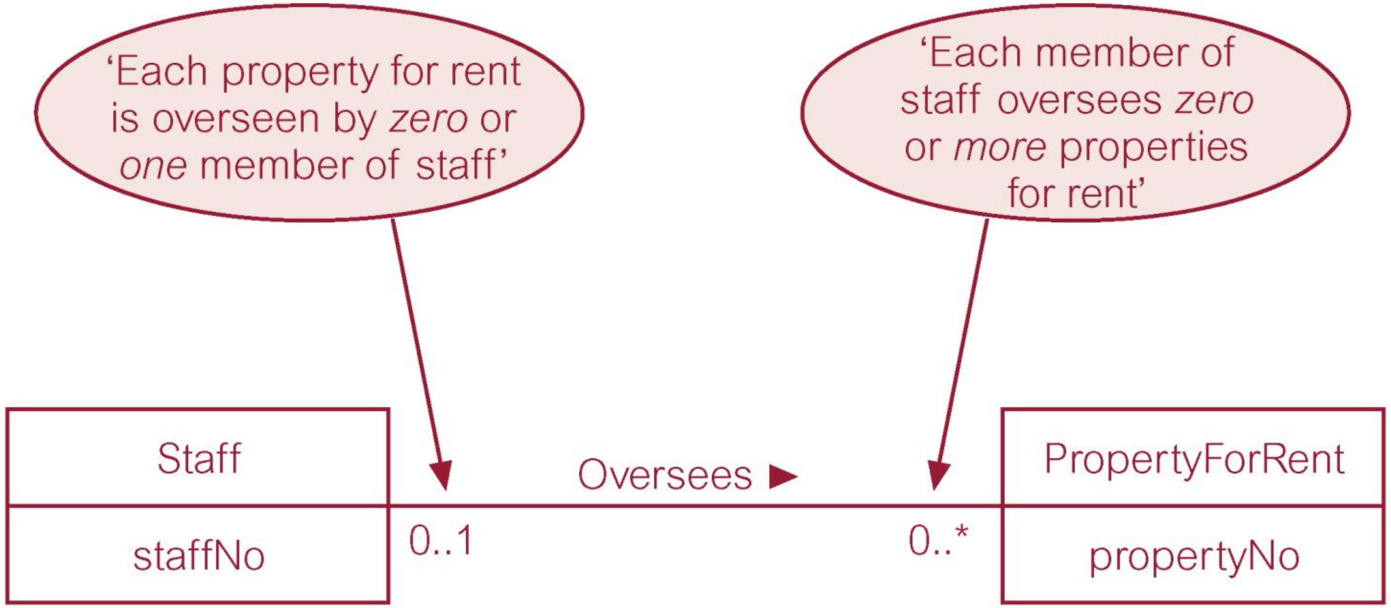
Multiplicity of Staff *Manages* Branch (1:1) relationship


‘Each branch is managed by *one* member of staff’
‘A member of staff can manage *zero* or *one* branch’
Staff
staffNo
Manages ►
1..1
0..1
Branch
branchNo
Multiplicity

Semantic net of Staff *Oversees* PropertyForRent relationship type


Staff entity (**staffNo**)
Oversees relationship
PropertyForRent entity (**propertyNo**)
SG5
SG37
SA9
r1
r2
r3
PG21
PG36
PA14
PG4

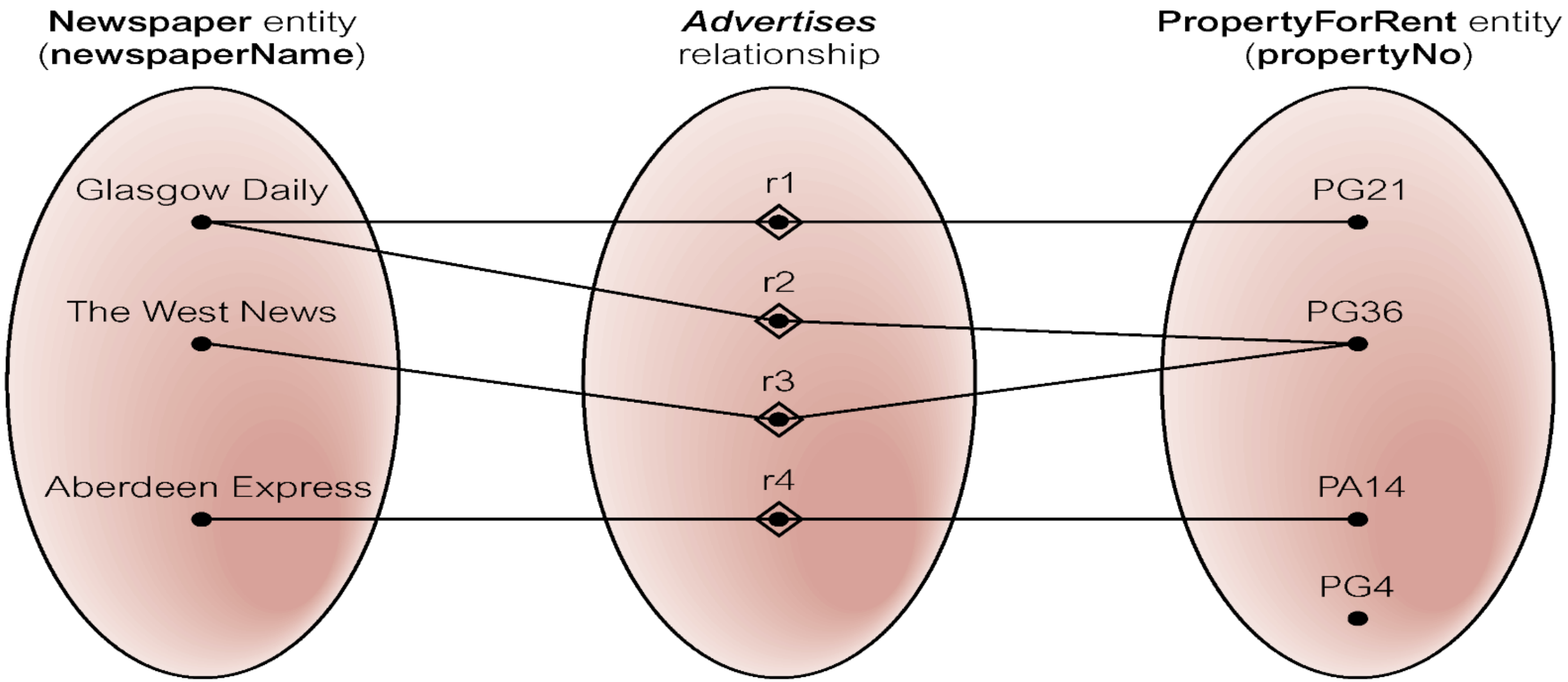
Multiplicity of Staff *Oversees* PropertyForRent (1:*) relationship type



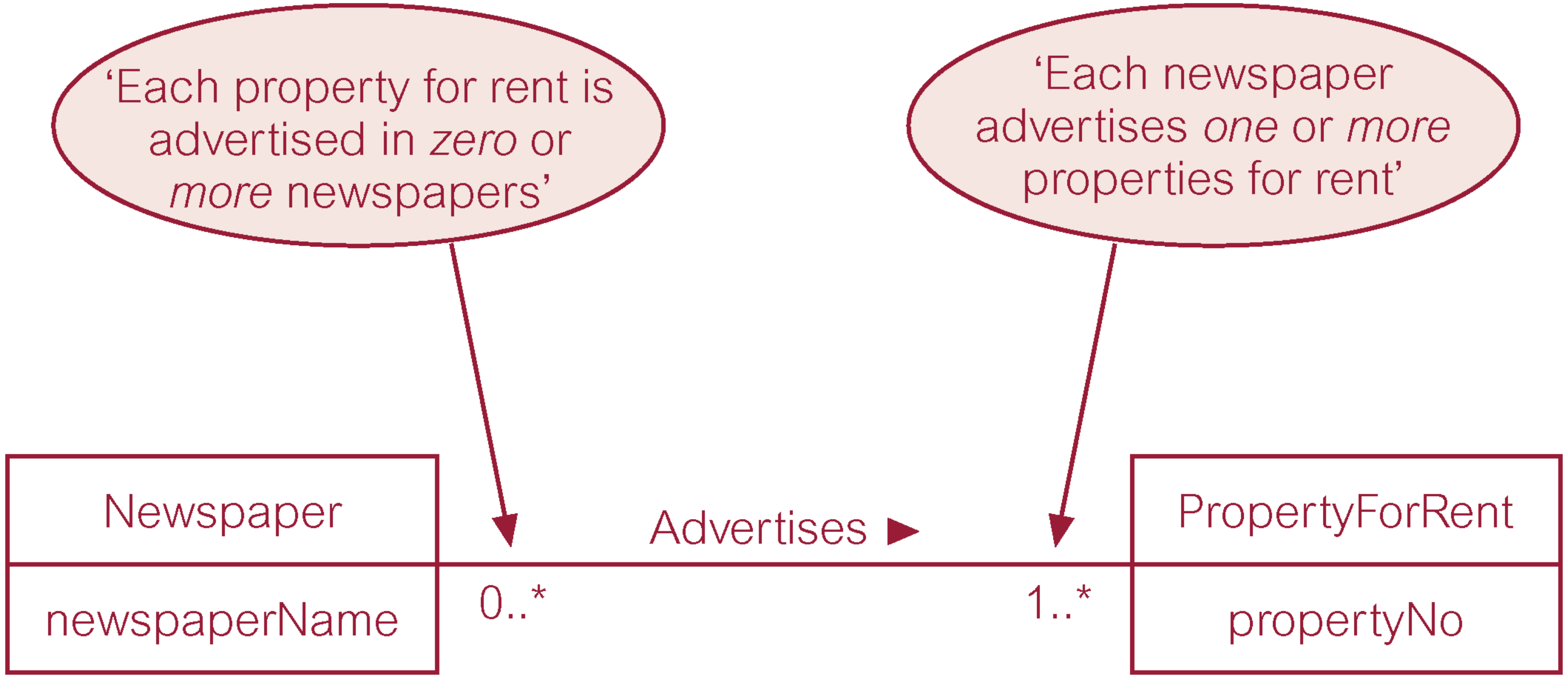
‘Each property for rent is overseen by *zero* or *one* member of staff’

‘Each member of staff oversees *zero* or *more* properties for rent’

Semantic net of Newspaper *Advertises* PropertyForRent relationship type



Multiplicity of Newspaper *Advertises* PropertyForRent (*:*) relationship



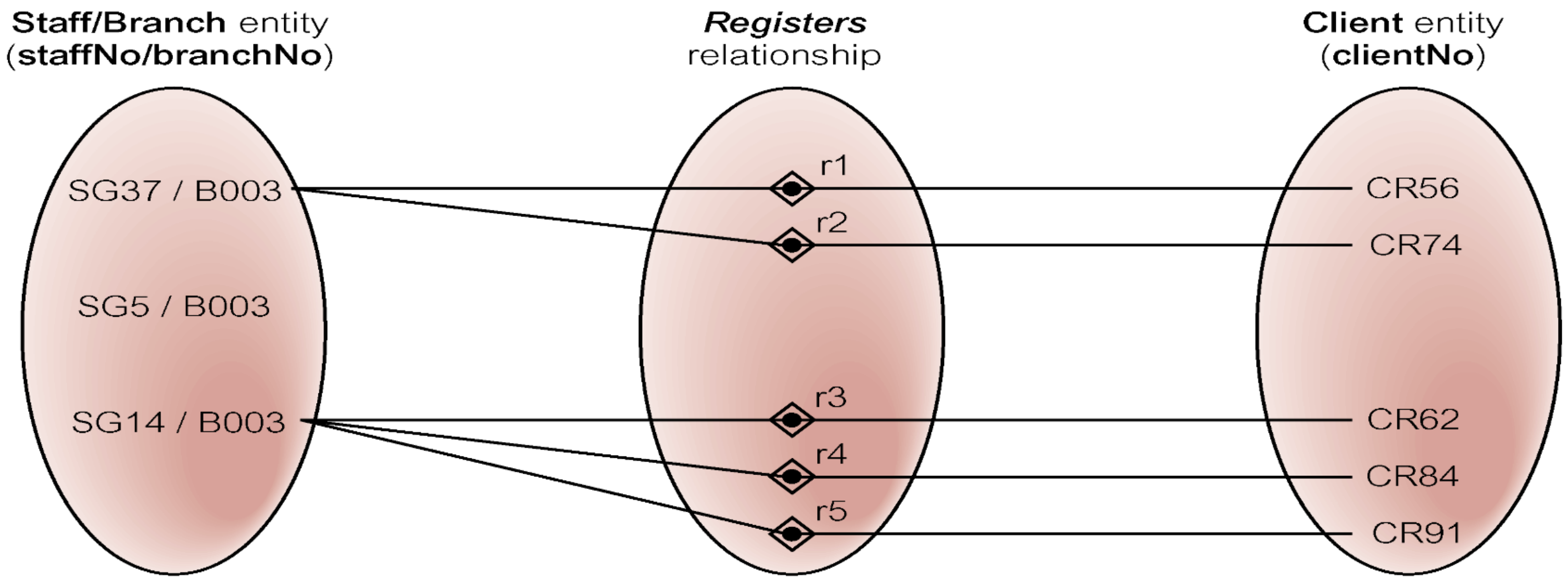
‘Each property for rent is advertised in *zero* or *more* newspapers’
‘Each newspaper advertises *one* or *more* properties for rent’
Newspaper
newspaperName
Advertises ►
0..*
1..*
PropertyForRent
propertyNo

Structural Constraints

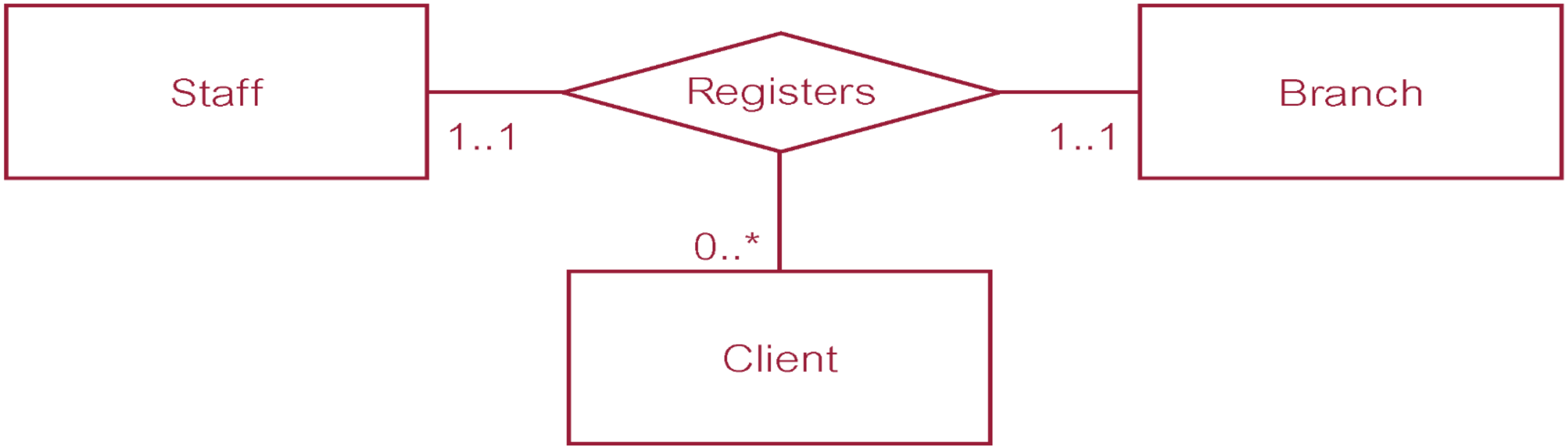
Multiplicity for Complex Relationships

- Number (or range) of possible occurrences of an entity type in an n-ary relationship when other (n-1) values are fixed.

Semantic net of ternary *Registers* relationship with values for Staff and Branch entities fixed



Multiplicity of ternary *Registers* relationship



Multiplicity of ternary *Supply* relationship

Assume that we have a situation

a) *Suppliers* can **supply** *products* for *projects*.

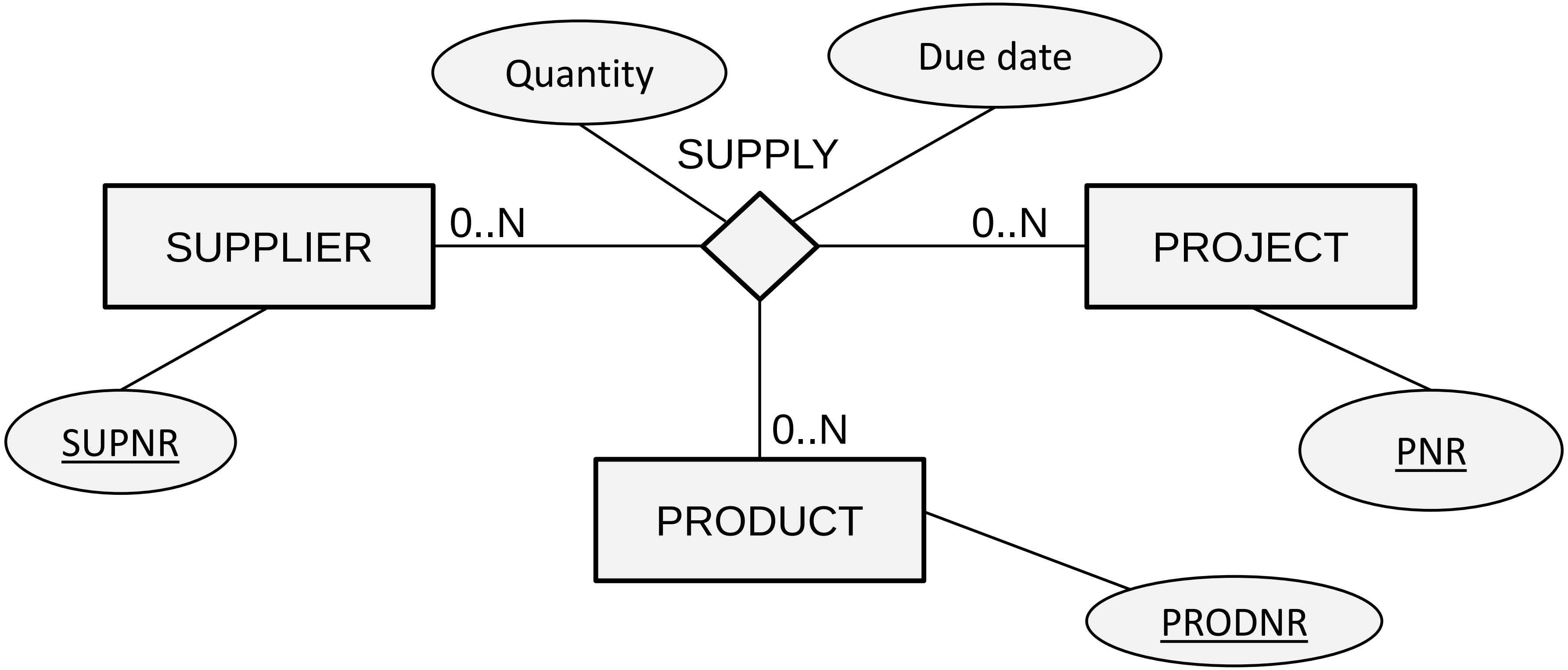
b) A *supplier* can supply a particular *product* for multiple *projects*.

c) A *product* for a particular *project* can be supplied by multiple *suppliers*.

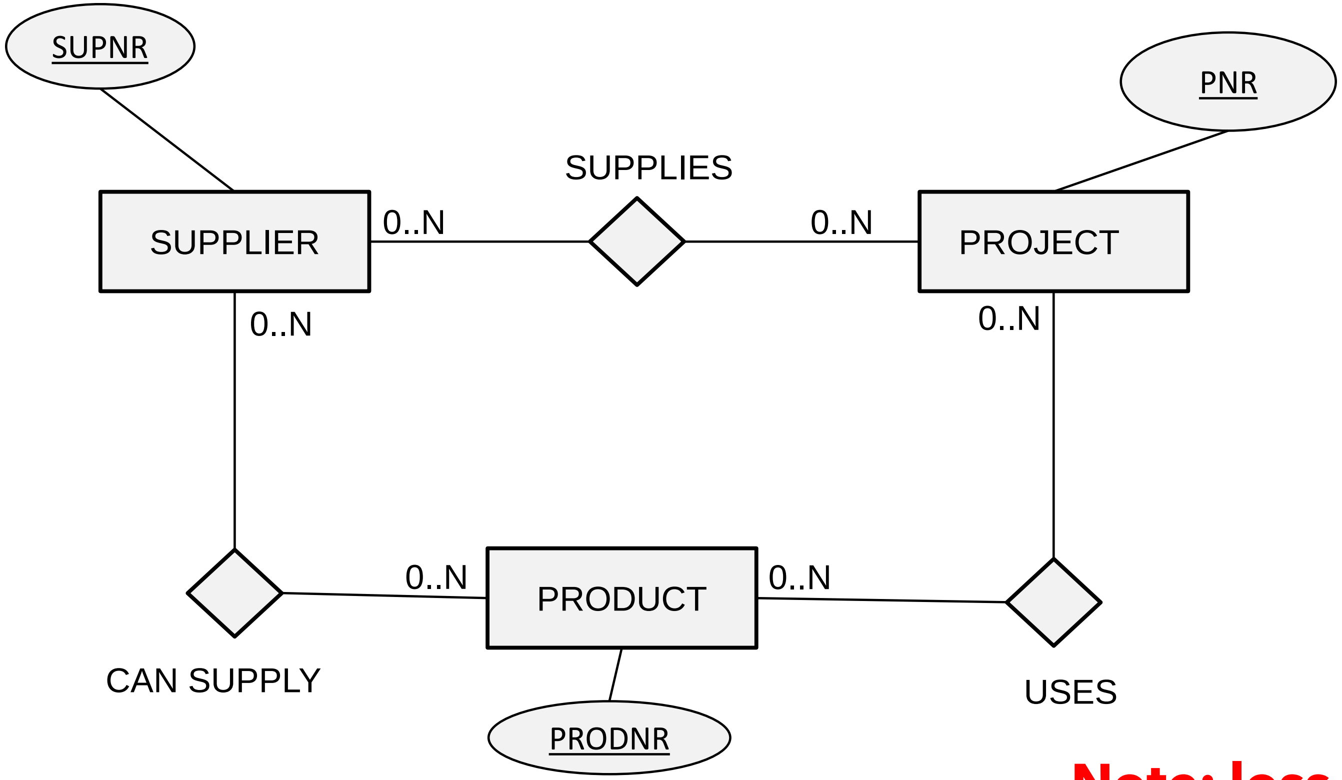
d) A *project* can have a particular *supplier* supply *multiple* products.

e) The model must also include the quantity and due date for ***supplying*** a particular **product** to a particular **project** by a particular **supplier**.

Multiplicity of ternary *Supply* relationship



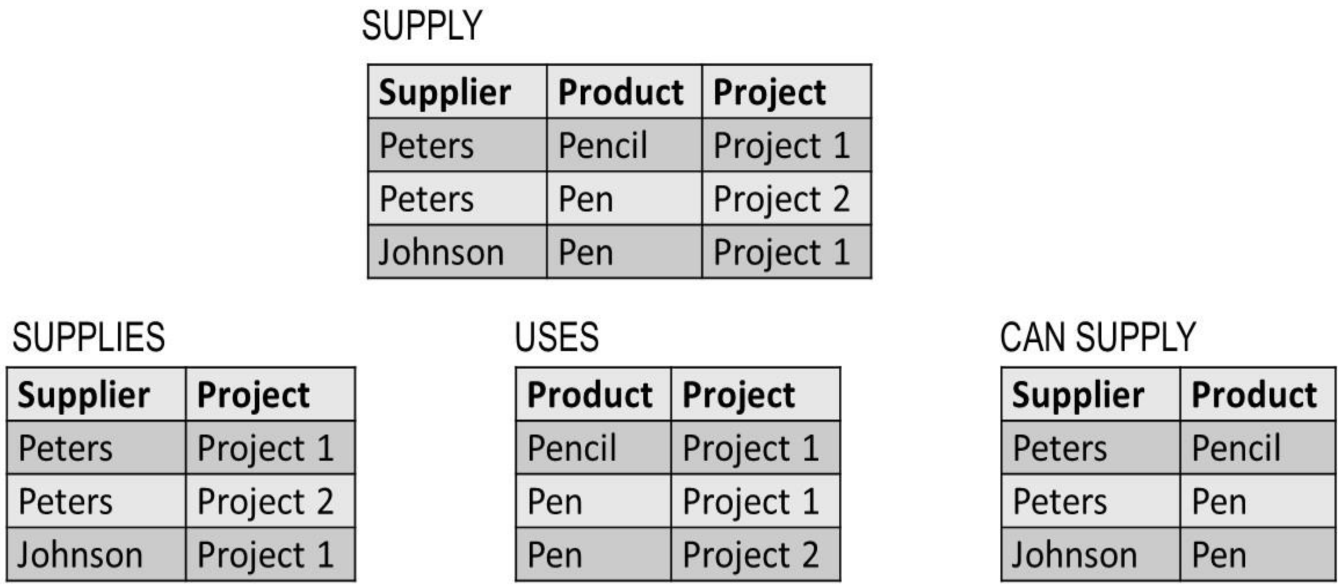
Multiplicity of ternary *Supply* relationship



Note: loss of semantics!

Ternary Relationship Types

- Say we have two projects: project 1 uses a pencil and a pen, and project 2 uses a pen. Supplier Peters supplies the pencil for project 1 and the pen for project 2 whereas supplier Johnson supplies the pen for project 1.
- From the binary relationship types, it is not clear who supplies the pen for project 1!



SUPPLY

Supplier	Product	Project
Peters	Pencil	Project 1
Peters	Pen	Project 2
Johnson	Pen	Project 1

SUPPLIES

Supplier	Project
Peters	Project 1
Peters	Project 2
Johnson	Project 1

USES

Product	Project
Pencil	Project 1
Pen	Project 1
Pen	Project 2

CAN SUPPLY

Supplier	Product
Peters	Pencil
Peters	Pen
Johnson	Pen

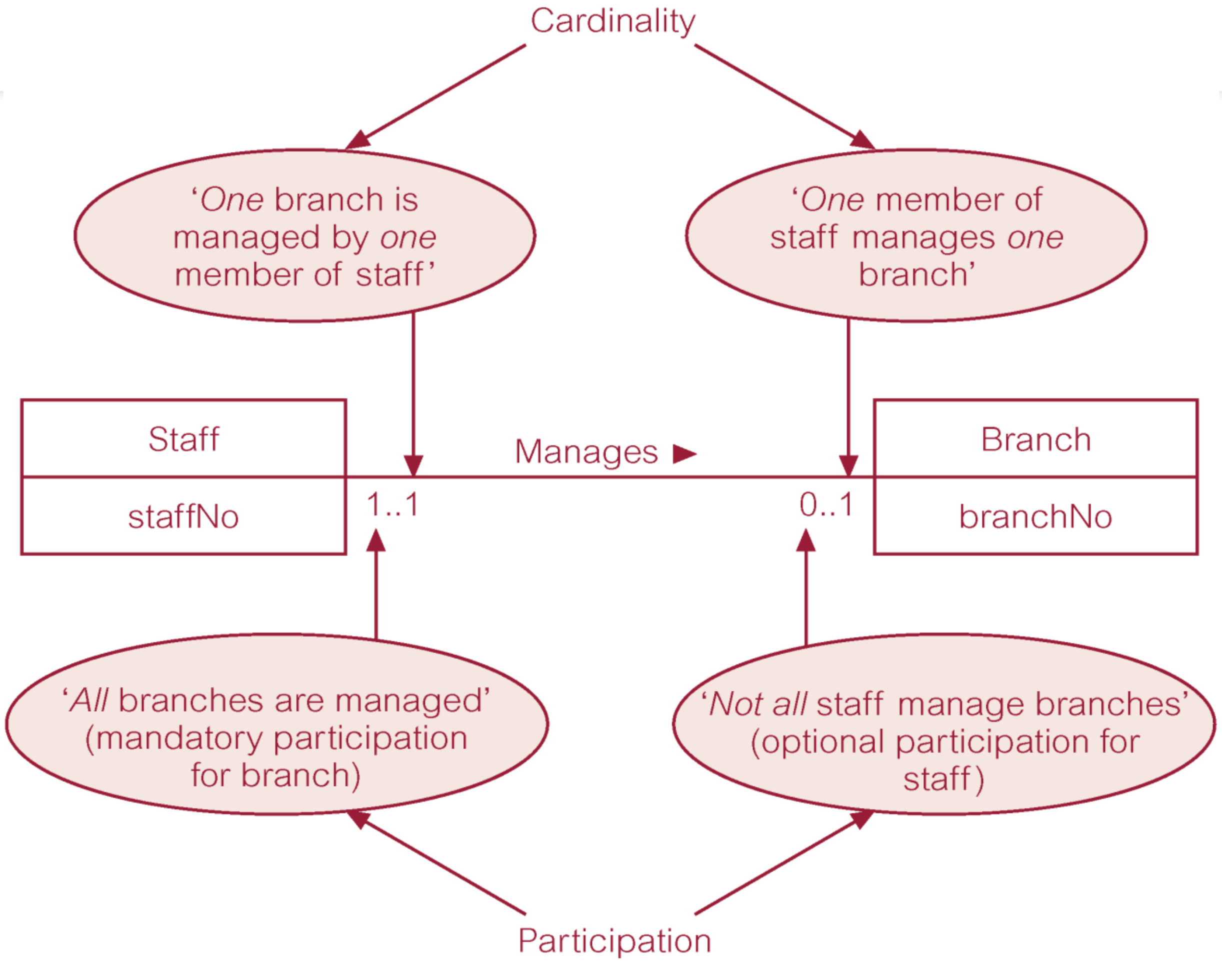
Structural Constraints

- **Multiplicity**
  - is made up of two types of restrictions on relationships: **cardinality and participation**.
- **Cardinality**
  - Describes *maximum number of possible relationship occurrences* for an entity participating in a given relationship type.
- **Participation**
  - Determines whether all or *only some entity occurrences participate* in a relationship.

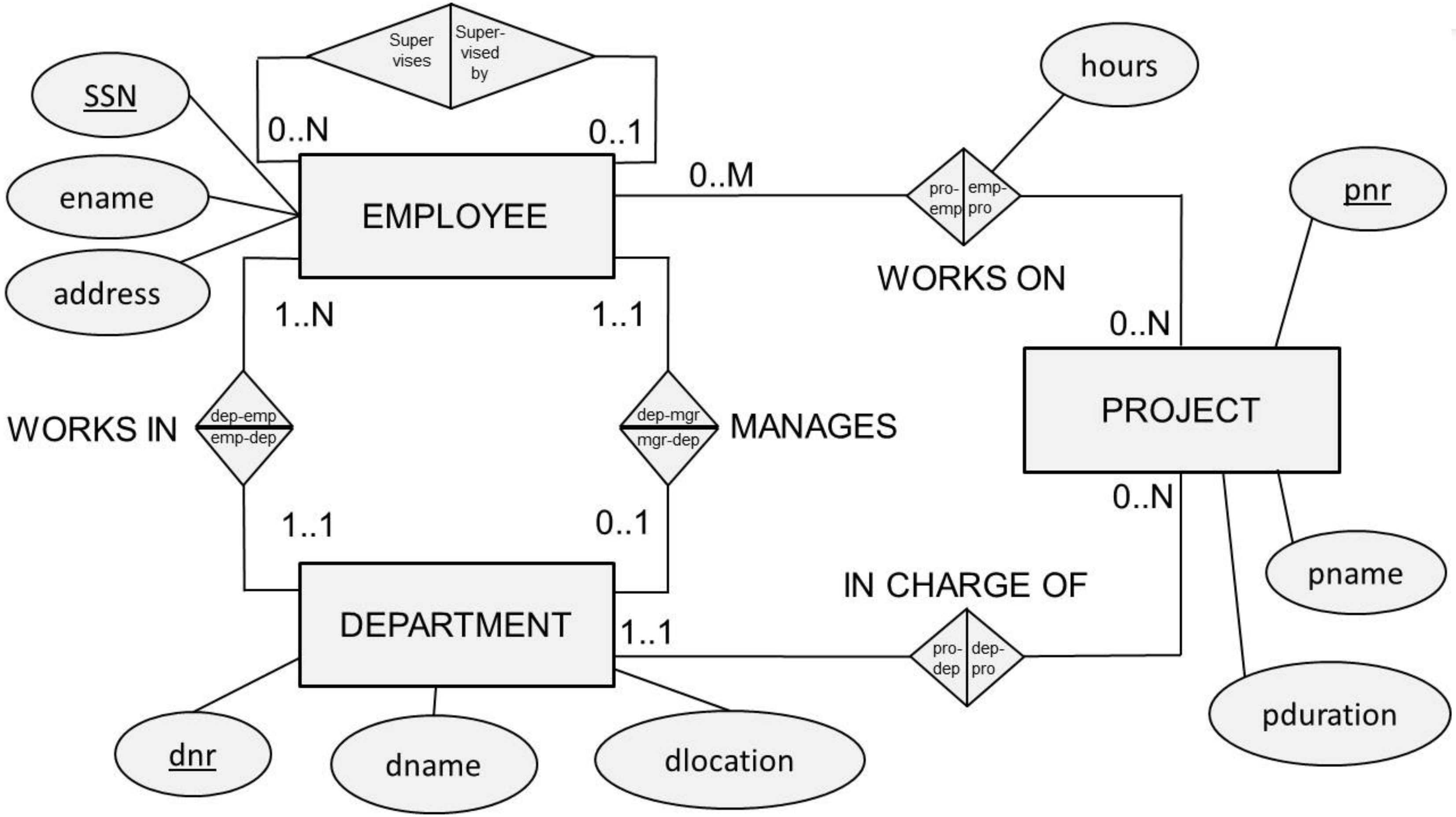
Participation

- **Partial Participation**
  - a situation in which **some** entities may not participate in the relationship.
- **Total Participation or Existence Dependency**
  - a situation in which **all** entities need to participate in the relationship; the existence of an entity depends upon the existence of another.

Multiplicity as cardinality and participation constraints

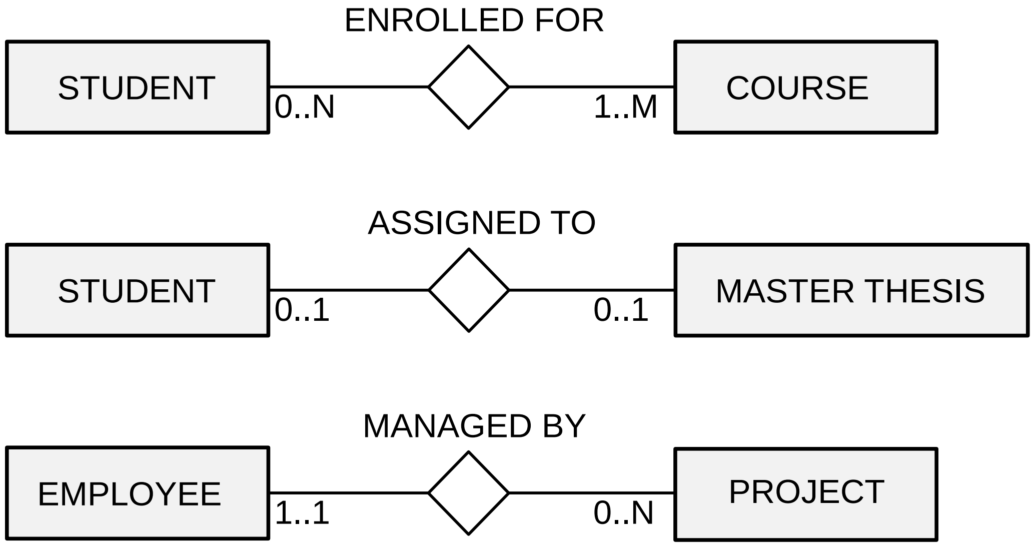

Cardinality
'*One* branch is managed by *one* member of staff'
'*One* member of staff manages *one* branch'
Staff
staffNo
Manages ►
1..1
0..1
Branch
branchNo
'*All* branches are managed' (mandatory participation for branch)
'*Not all* staff manage branches' (optional participation for staff)
Participation

ER Model (Chen’s Notation)



Multiplicity

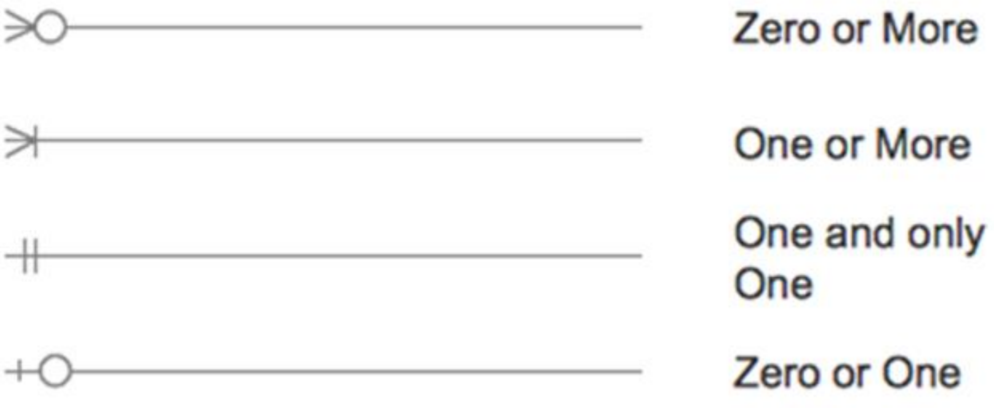
Chen’s VS Crow’s Foot



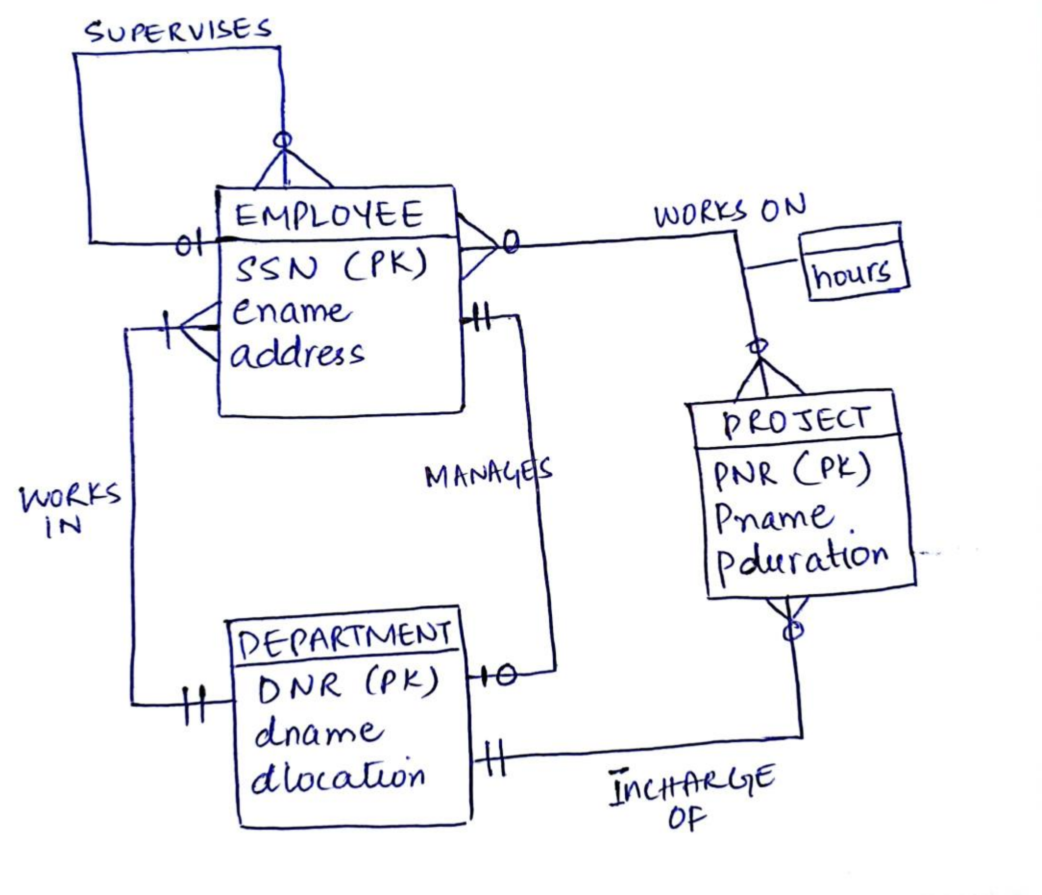
N:M

1:1

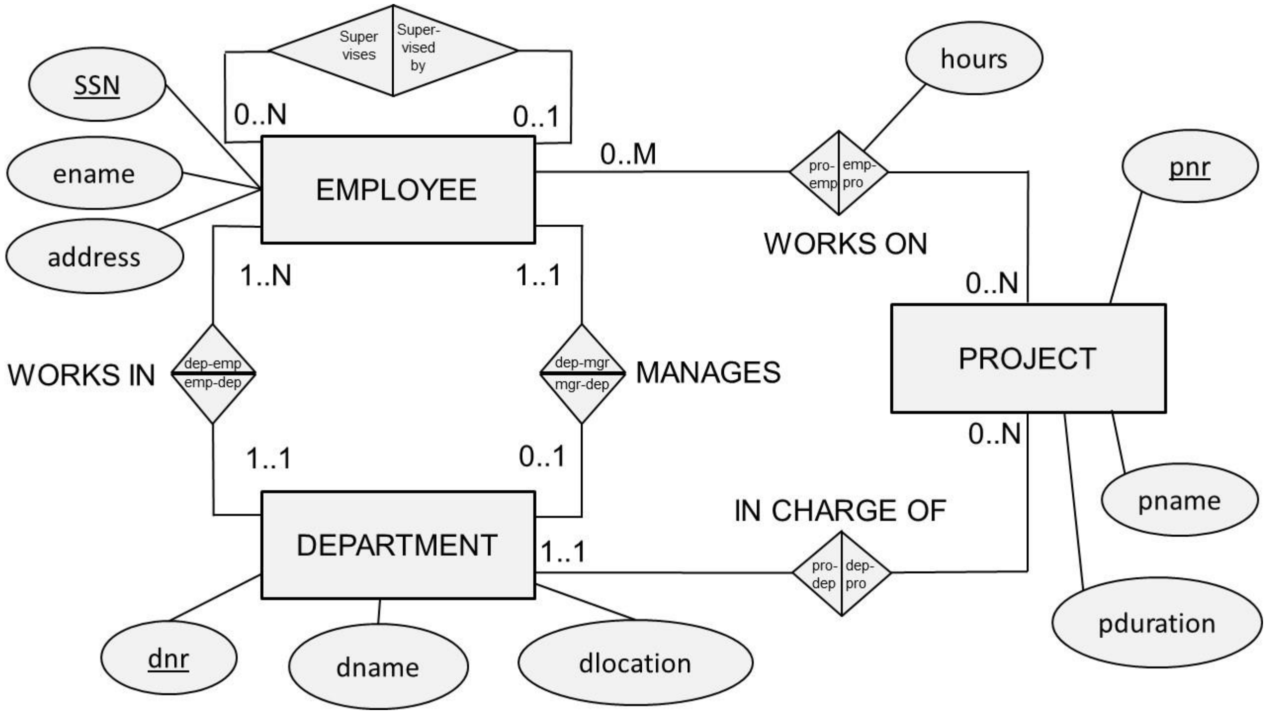
1:N

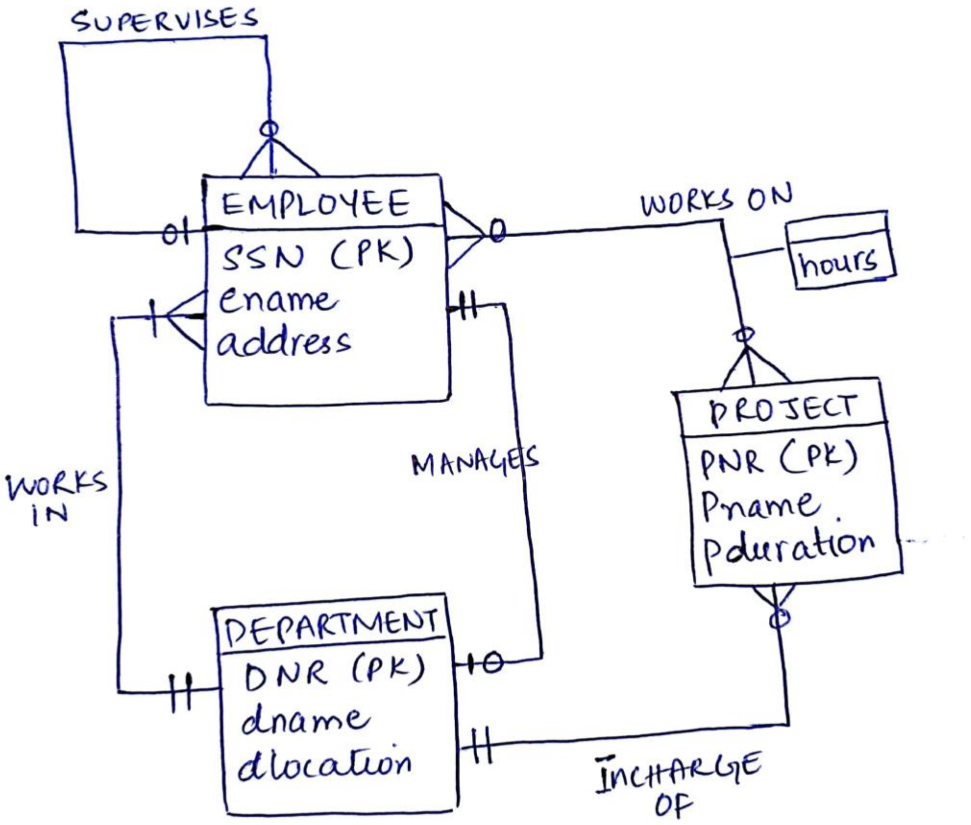


ER Model (Crow's Foot Notation)

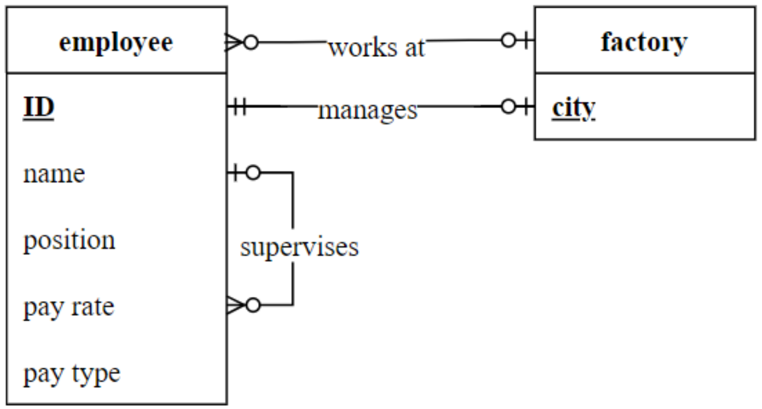


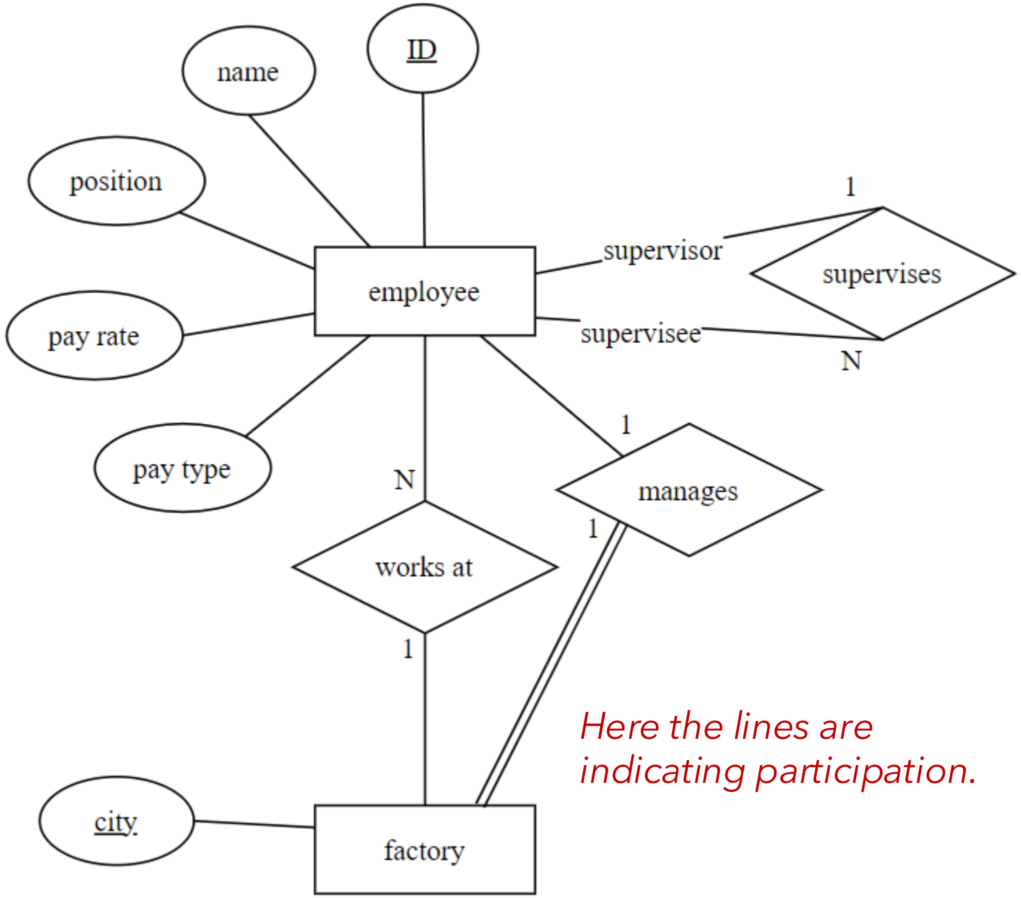
Comparison


Super vises
Super- vised by
0..N
0..1
SSN
ename
address
EMPLOYEE
0..M
pro- emp
emp- pro
hours
WORKS ON
0..N
pnr
PROJECT
1..N
1..1
WORKS IN
dep-emp
emp-dep
dep-mgr
mgr-dep
MANAGES
1..1
0..1
0..N
pname
IN CHARGE OF
DEPARTMENT
1..1
pro- dep
dep- pro
pduration
dnr
dname
dlocation


SUPERVISES
EMPLOYEE
SSN (PK)
ename
address
WORKS ON
hours
WORKS IN
MANAGES
PROJECT
PNR (PK)
Pname
Pduration
DEPARTMENT
DNR (PK)
dname
dlocation
INCHARGE OF

Comparison





Here the lines are indicating participation.

Scenario 1

- Consider a student enrollment system.
- Identify major entities, their attributes
- Identify relations and multiplicity (participation + cardinality)
- Draw an ER model in
    - Chen's Notation
    - Crow's Foot Notation