Entity Relationship Modelling

Peter Chen

- Introduced ER modelling in 1976

Classify informational needs in terms of

- Entity types
- Relationship types
- Attributes

Entity Relationship Modelling

Entity type

- These represent the things, person, places, events that you need to keep track of
- Recall Company database
  Entity types are Department, Project, Employee, Dependent

- Instances of entity types, commonly called **entities**:
  For Departments:
  Research
  Administration
  Headquarters

  For Employees:
  John Smith
  Franklin Wong….. etc

Entity Relationship Modelling

Relationship type

- These represent the way entity types interact or relate to one another
- Recall Company database
  An employee has dependents
  An employee works in a department
  An employee works on a project
  An employee supervises an employee
  A department is managed by an employee

- The verbs used to describe how entities relate to one another indicate relationships

- Examples of instances of a relationship type:
  Franklin Wong *supervises* John Smith
  James Borg *supervises* Franklin Wong

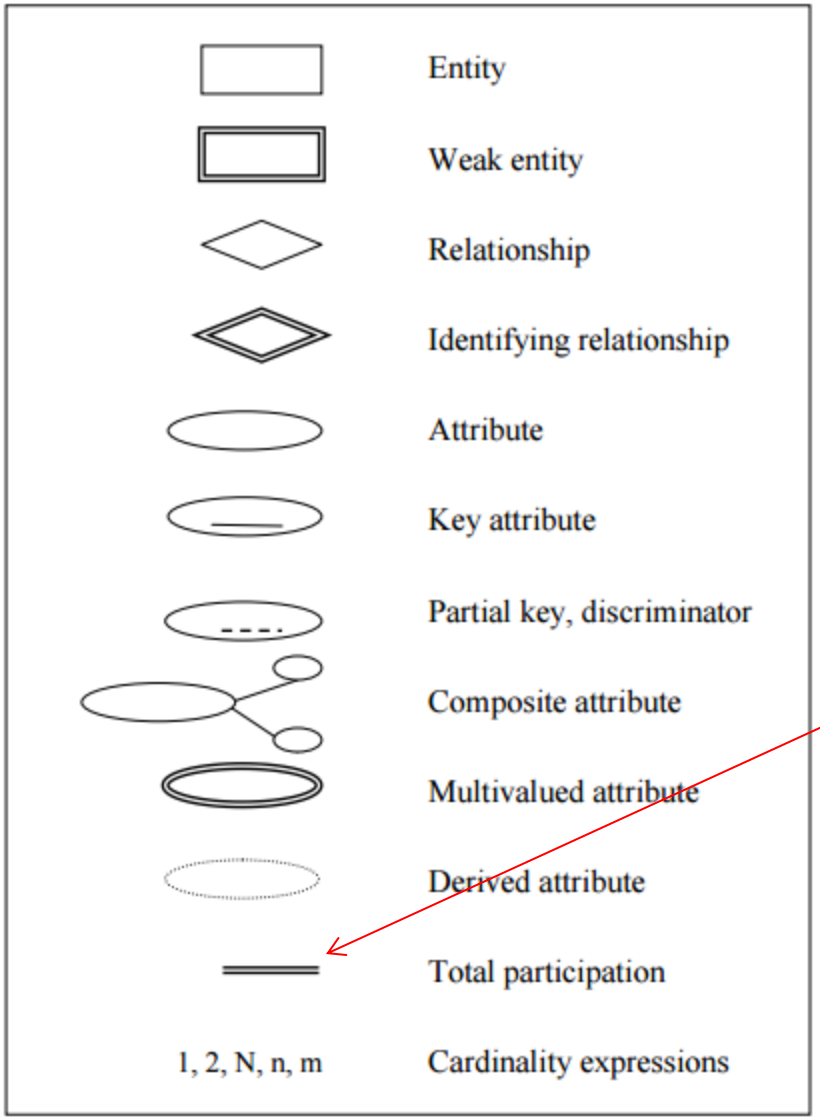
Entity Relationship Modelling

Attributes

- These are the characteristics that describe an entity type or a relationship type
- Recall Company database
- A department is identified by its number
  A department is identified by its name
  A department is in many locations

  A project is identified by its name
  A project is identified by its number
  A project is at a location

Entity Relationship Modelling



single line

Regular relationship line
(not mandatory i.e. not total participation)

double line

Entity Relationship Modelling

Example. What does an ERD look like if we have entity types Department, Project, Employee and relationships:

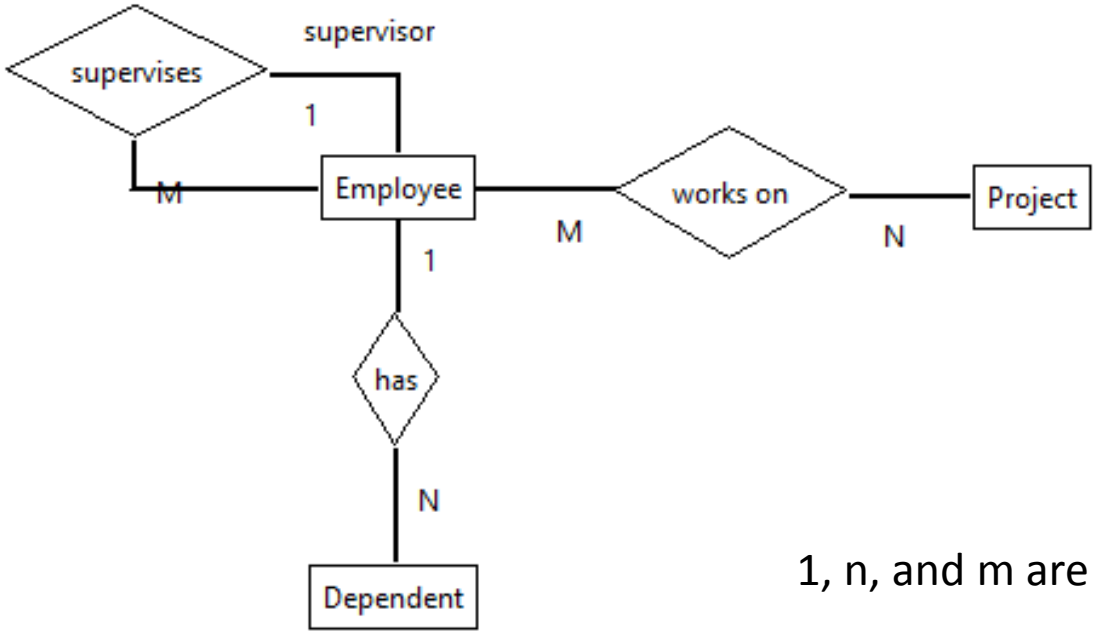
An employee may supervise many employees
An employee may have many dependents
An employee may work on many projects
A project must be sponsored by exactly one department
A department must have exactly one manager
An employee must be assigned to a department

An employee may be supervised by at most one employee
A dependent must belong to exactly one employee
A project may be worked on by many employees
A department may sponsor many projects
An employee can managed one department
A department may have many employees

Entity Relationship Modelling

Example. At first we will include:
An employee may supervise many employees
An employee may have many dependents
An employee may work on many projects

This ERD needs many refinements
… more rules to include

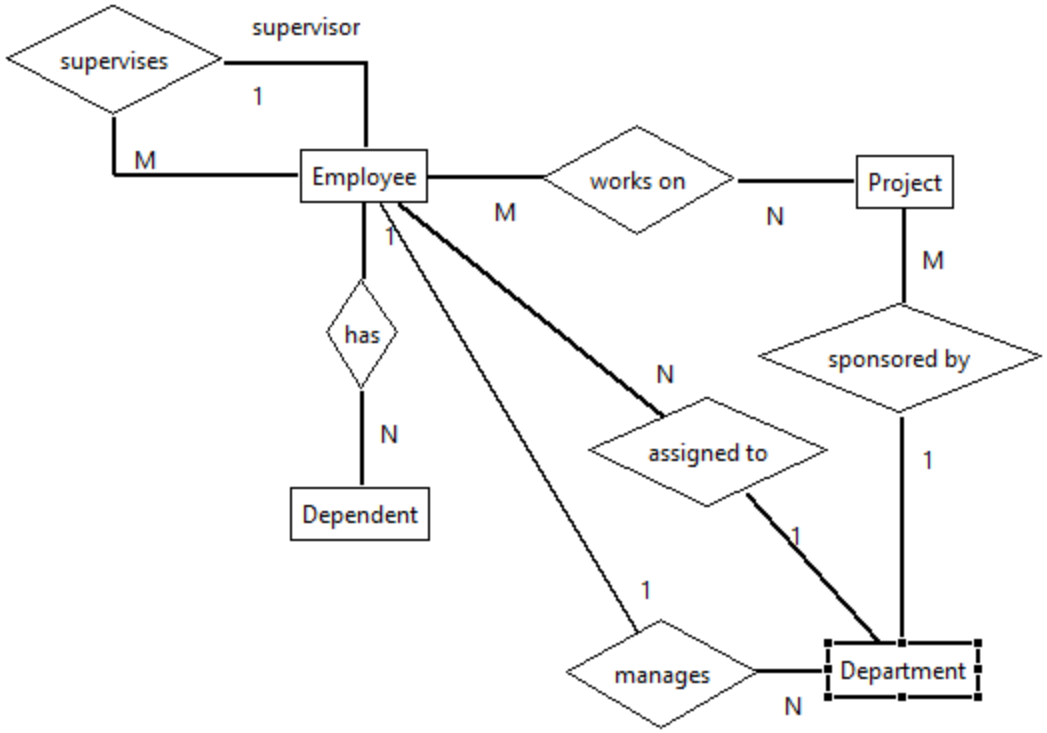


1, n, and m are cardinalities → later

Demonstration using ERD Tool found at
http://www.acs.uwinnipeg.ca/rmcfadyen/CreativeCommons/index.htm

Entity Relationship Modelling

Example. Now, we'll include:
A project must be sponsored by exactly one department
A department must have exactly one manager
An employee must be assigned to a department



Entity Relationship Modelling

Weak Entity Types.
When the existence of an entity depends on the existence of another entity, we say the entity is a weak entity.

Weak entity types are shown with a **double-lined** rectangle

Example: a dependent must not exist unless the dependent is related to an employee

Two related concerns:

There must be at least one relationship the weak entity must participate in.

Mandatory participation (**total** participation) is shown with a **double-line**.

Dependents are identified by their name along with the employee's ssn. SSN is the key of Employee and we show that this is used to partially identify a dependent using an identifying relationship.

An **identifying** relationship is shown with a **double-lined** diamond shape.

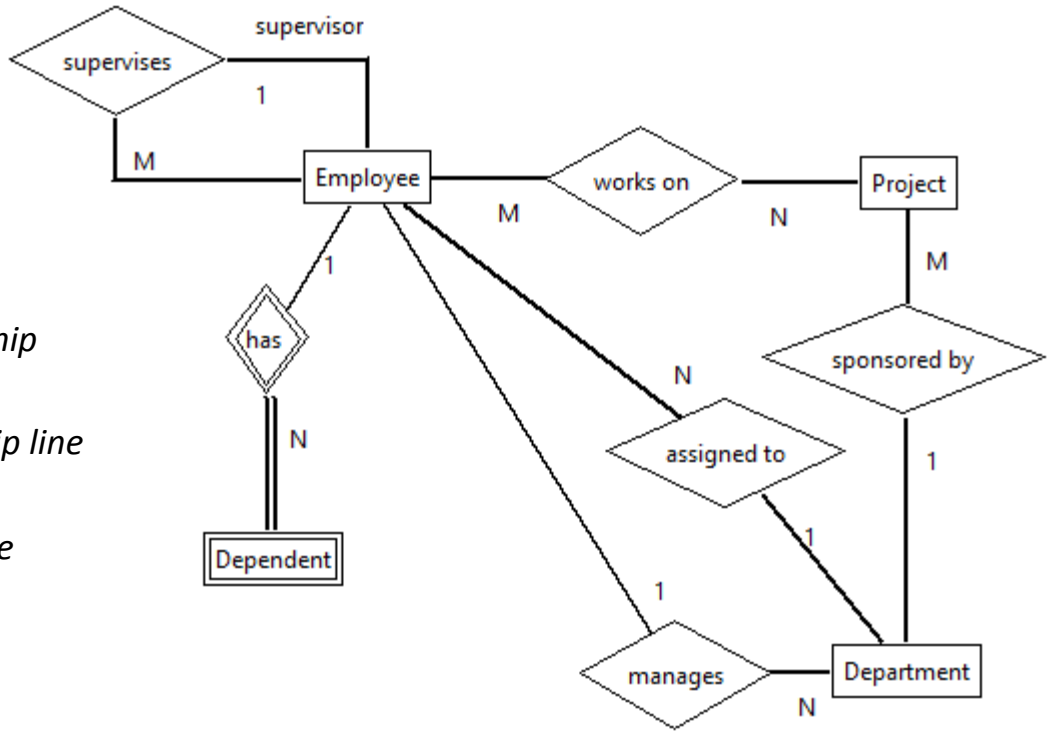
Entity Relationship Modelling

Weak Entity Types.

Weak entity types are shown with a double-lined rectangle

Mandatory participation (**total** participation) is shown with a double-line.

An **identifying** relationship is shown with a double-lined diamond shape.



double-lined relationship

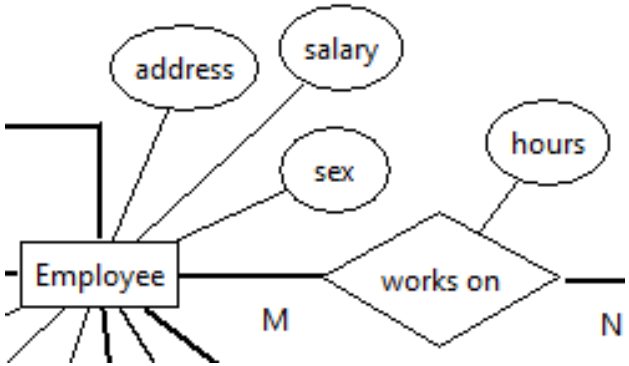
double-line relationship line

double-lined entity type

Entity Relationship Modelling

Attributes. For each entity and relationship type we must ascertain their attributes (What describes an instance?).

e.g. an employee is described by their *address*, *salary*, *sex*
e.g. and employee works *hours* on a project



Attributes are shown with a single-lined oval (with exceptions)

Entity Relationship Modelling

For each lowest level attribute, it must be

Atomic – cannot meaningfully be decomposed any further

e.g. an employee works some hours on a project (not broken down to hours & minutes)

We must determine if the following applies:

Composite – for purpose of business rules we can show an attribute composed of other attributes

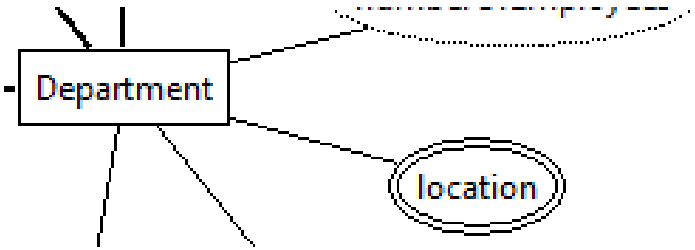
e.g. the name of an employee comprises their first name, initial, and last name

*Attributes are connected as in a **tree**.*

Entity Relationship Modelling

Single valued – an instance of the entity or relationship has at most one value for the attribute
e.g. pnumber, pname, plocation for a project
e.g. name, sex, birthDate, relationship for a dependent
e.g. startDate for an employee as manager of a department

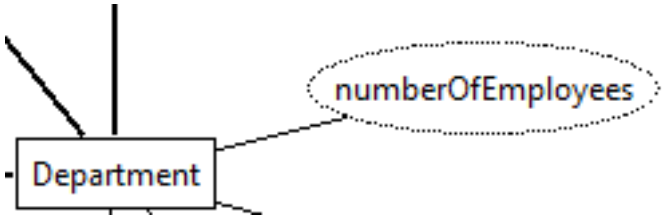
Multi-valued – an instance ….. can have many values
e.g. a department is in many locations



Multi-valued attribute has a double-lined oval

Entity Relationship Modelling

Derived – the value of an attribute can be obtained from other sources in the database
e.g. a department has "number of employees"



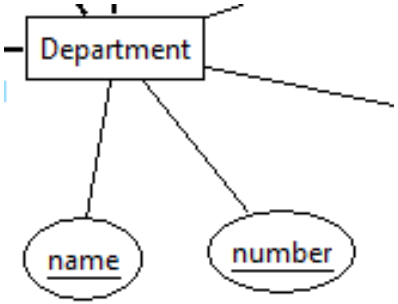
Derived attribute has a dotted-lined oval

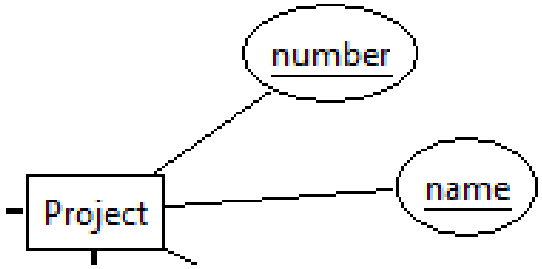
Entity Relationship Modelling

Key – an attribute (may be composite) distinguishes instances of the entity type or relationship type (no two instances can have the same value for a key attribute)

e.g. Project has two keys: number and name

e.g. Department has two keys: number and name





Two keys

Each key is underlined

This is not a composite key

Entity Relationship Modelling

Partial Key

– an attribute that distinguishes instances of a weak entity type relative to a strong entity type

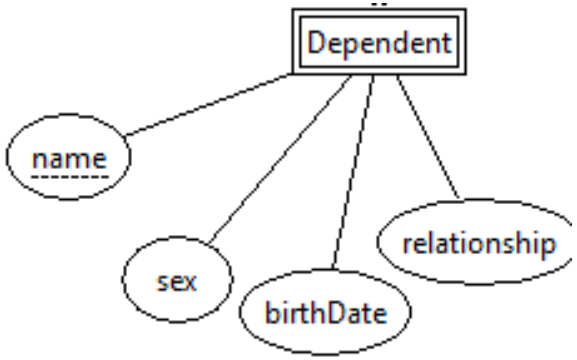
– an attribute that distinguishes instances of a relationship type relative to the associated entity type instances.

e.g. consider the dependent's name

– a dependent is partially identified by their name

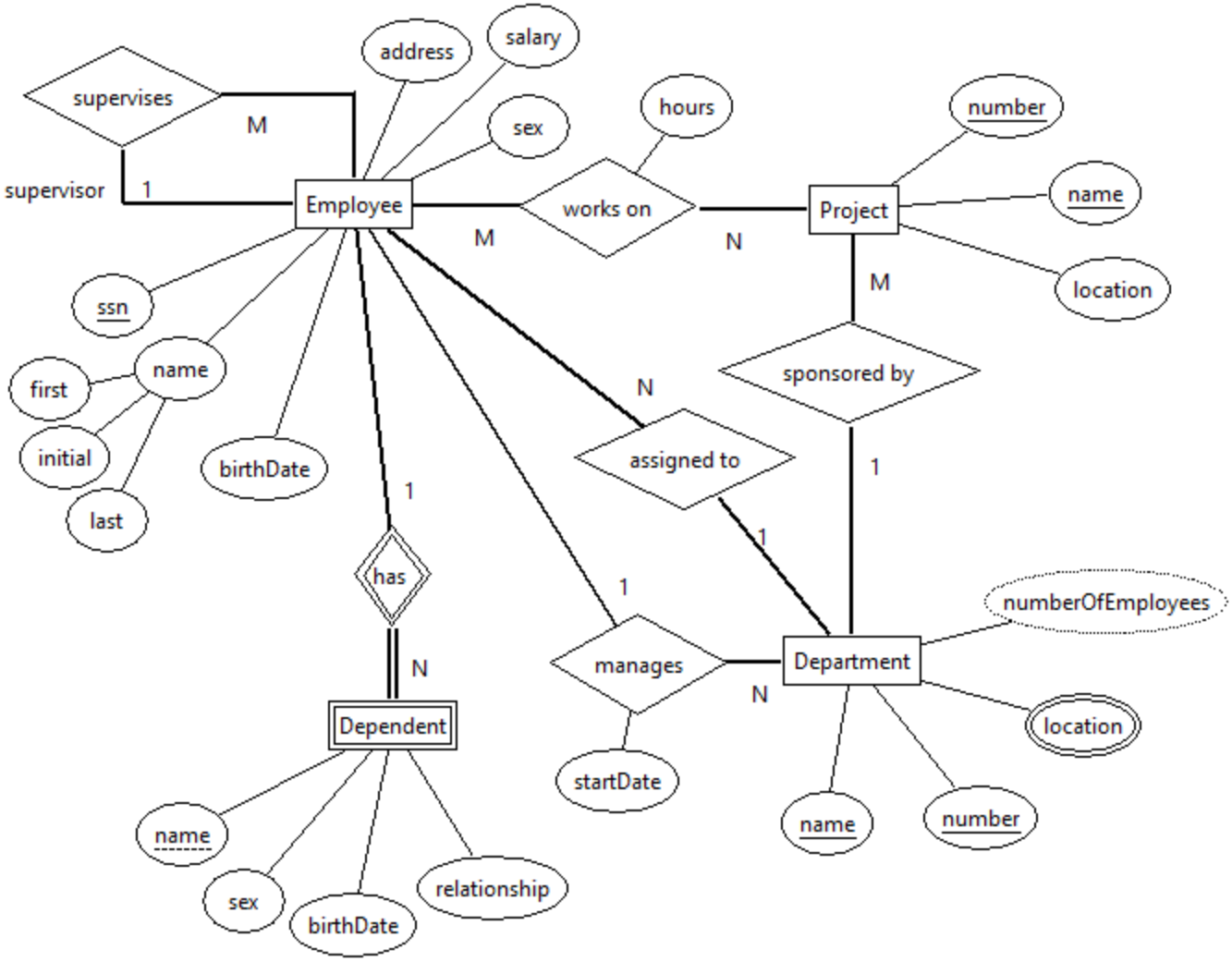
– partial because other dependents, of other employees, can have the same name

– the employee's id along with the dependent's name completely identifies a dependent



Partial key is underlined with dashes

Entity Relationship Modelling



Entity Relationship Modelling

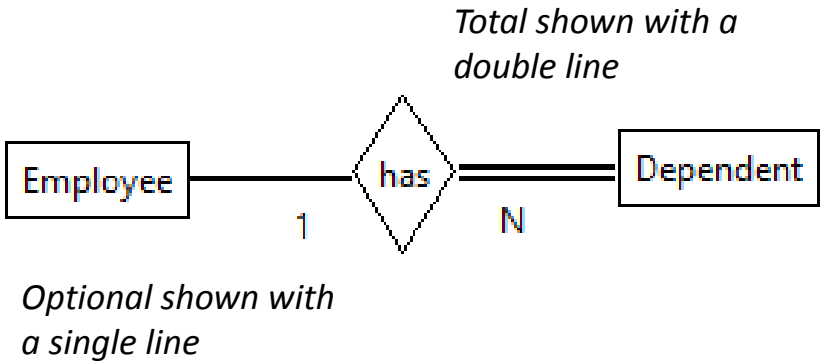
Relationships

Degree – the number of entity types involved

Mostly, these are binary … i.e. involving two entity types

Participation is either total or optional

e.g. a department must have a manager → mandatory (i.e. total)
e.g. an employee may have dependents → optional
e.g. a dependent must be related to an employee → mandatory



Entity Relationship Modelling

Cardinality of relationships

Binary relationships are one of: 1 to 1, or 1 to many (many to 1), or many-to-many

Lots of ways to indicate cardinality:

1-1, 1-many, many-1, many-many

1:1, 1:m, m:1, m:n

1 to many

An employee may supervise many employees

An employee is supervised by at most one employee

many-to-many

An employee may work on many projects

A project may have many employees working on it

1 to 1

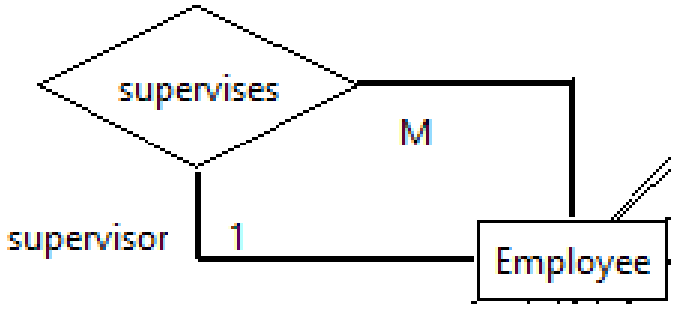
A department must have exactly one manager

An employee may manage at most one department

Entity Relationship Modelling

Recursive relationships
A relationship is recursive if an entity type plays more than one role

e.g. an employee supervises an employee



Note: The ERD Tool on the remote desktop does not save the positioning of the relationship lines

- The lines are one on top of the other
- You must select and re-position them to see the different lines

Entity Relationship Modelling

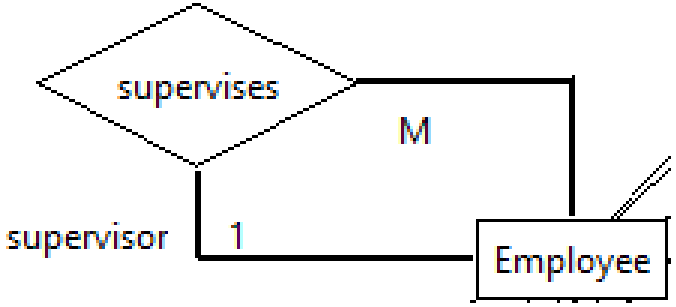
Role names

The role an entity type plays can be given a name

The employee who manages a department is the manager

Particularly useful for recursive relationships

A supervisor supervises other employees



Entity Relationship Modelling

Identifying Relationships
The key of one entity type is required to identify instances of another

e.g. dependents are identified by utilizing the employee's SSN.

e.g. UW courses are identified by department code + course number
- ACS-3902
- MATH-3902
- ENG-3902