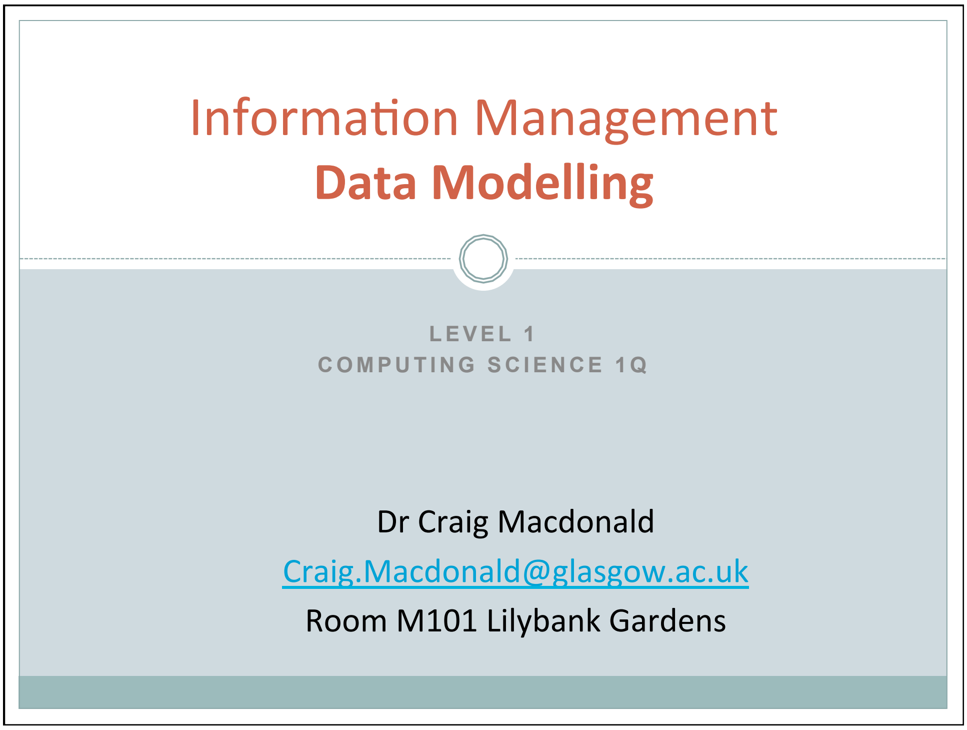
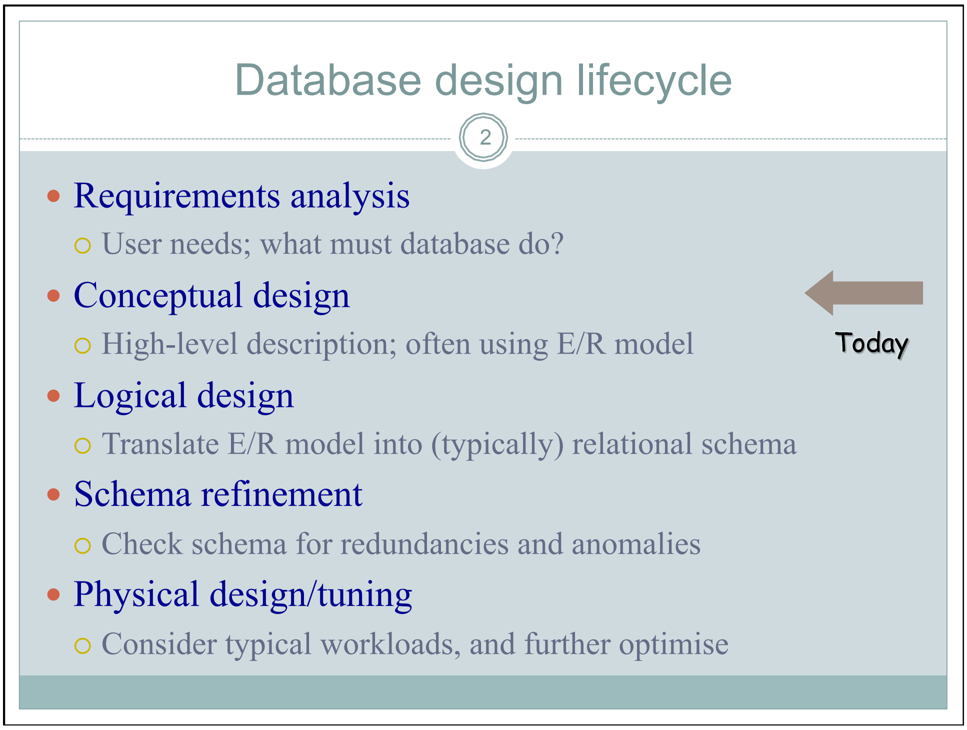


Information Management
Data Modelling

LEVEL 1
COMPUTING SCIENCE 1Q

Dr Craig Macdonald
Craig.Macdonald@glasgow.ac.uk
Room M101 Lilybank Gardens

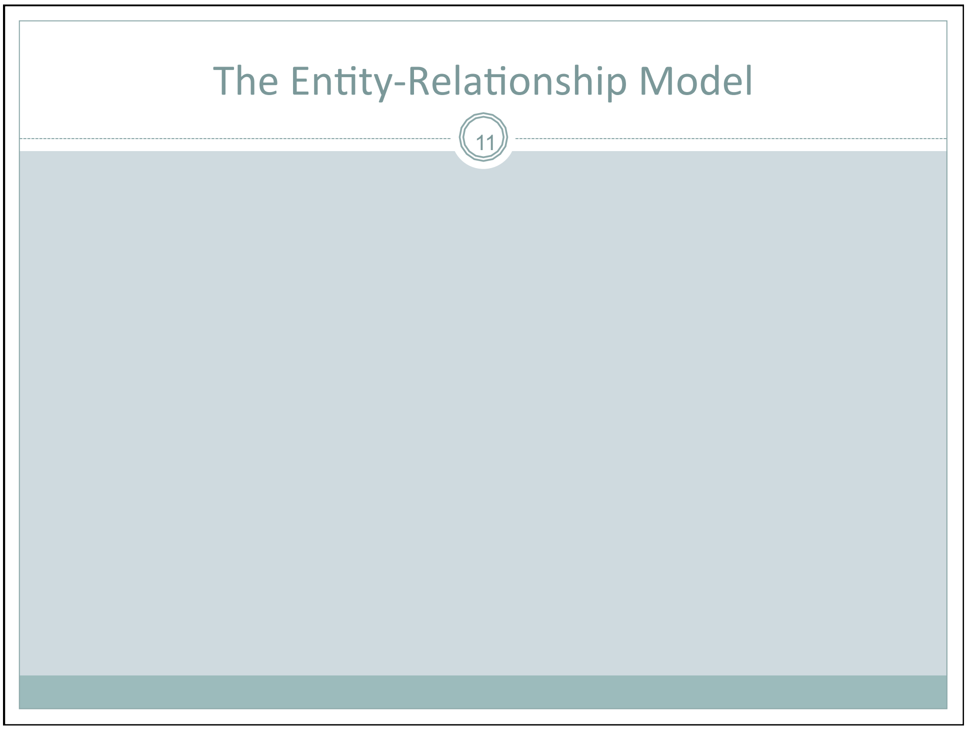


Database design lifecycle

2

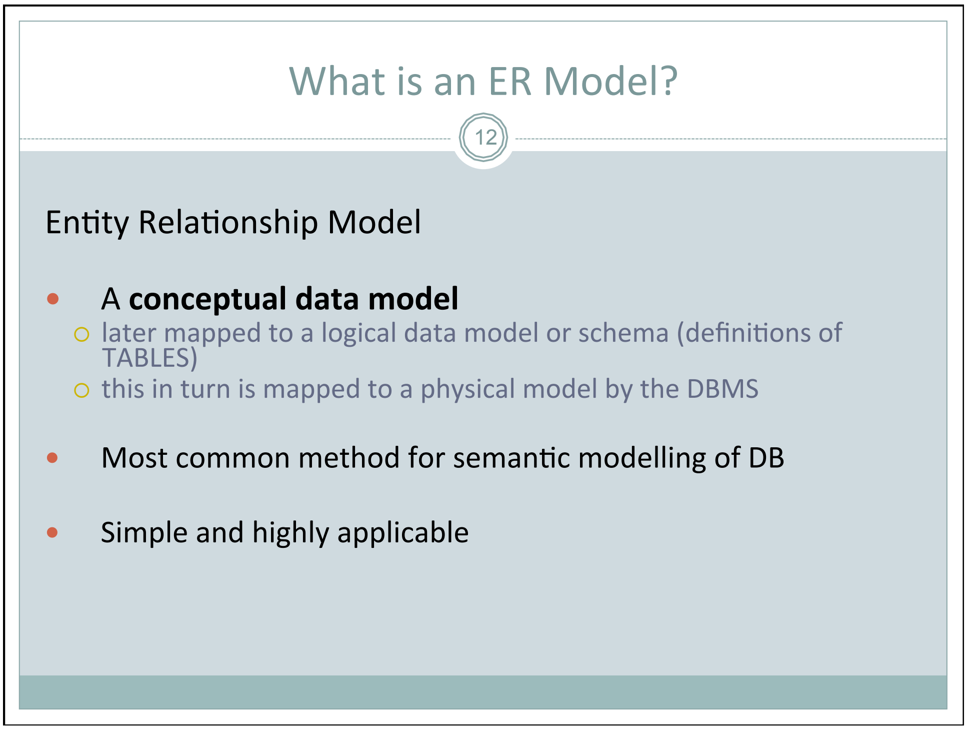
- Requirements analysis
  - User needs; what must database do?
- Conceptual design
  - High-level description; often using E/R model
- Logical design
  - Translate E/R model into (typically) relational schema
- Schema refinement
  - Check schema for redundancies and anomalies
- Physical design/tuning
  - Consider typical workloads, and further optimise

Today



The Entity-Relationship Model

11

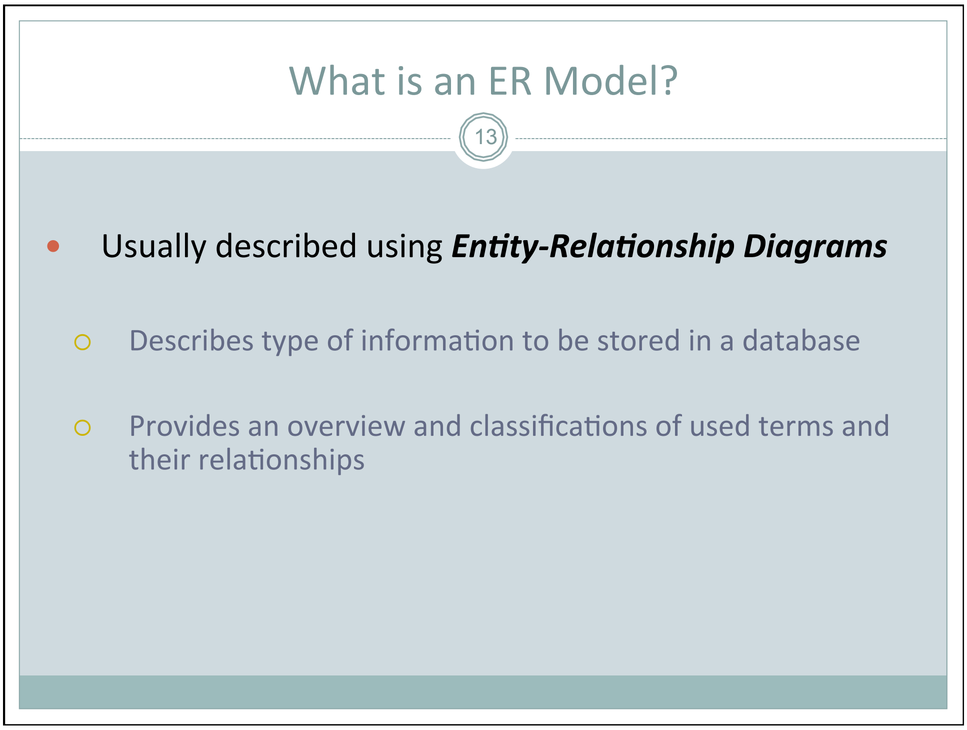


What is an ER Model?

12

Entity Relationship Model

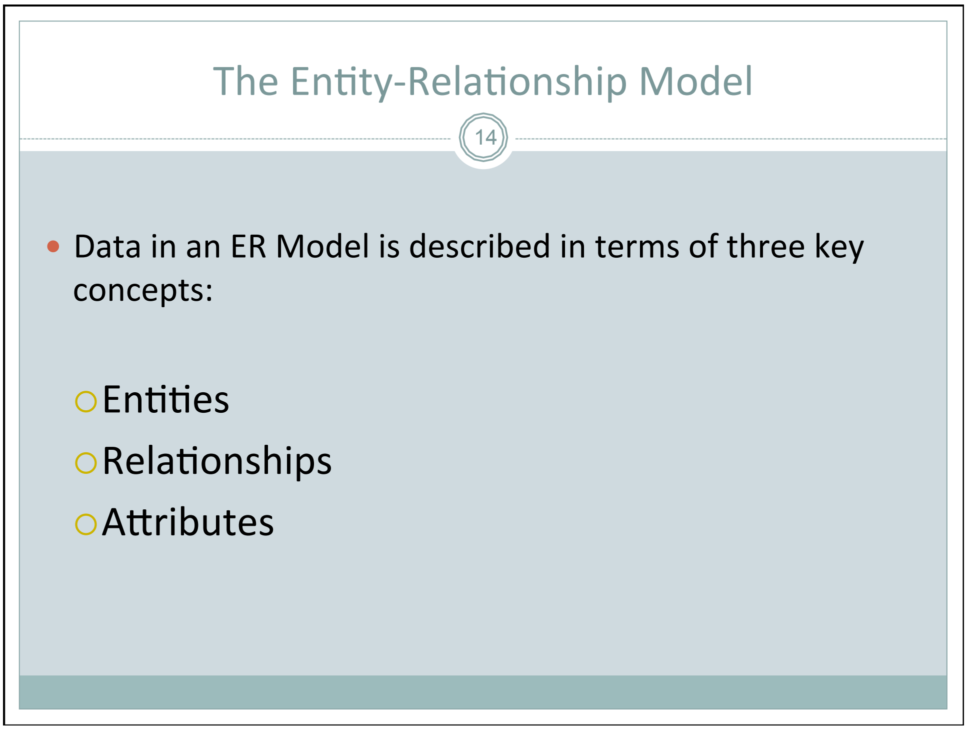
- A **conceptual data model**
  - later mapped to a logical data model or schema (definitions of TABLES)
  - this in turn is mapped to a physical model by the DBMS
- Most common method for semantic modelling of DB
- Simple and highly applicable



What is an ER Model?

13

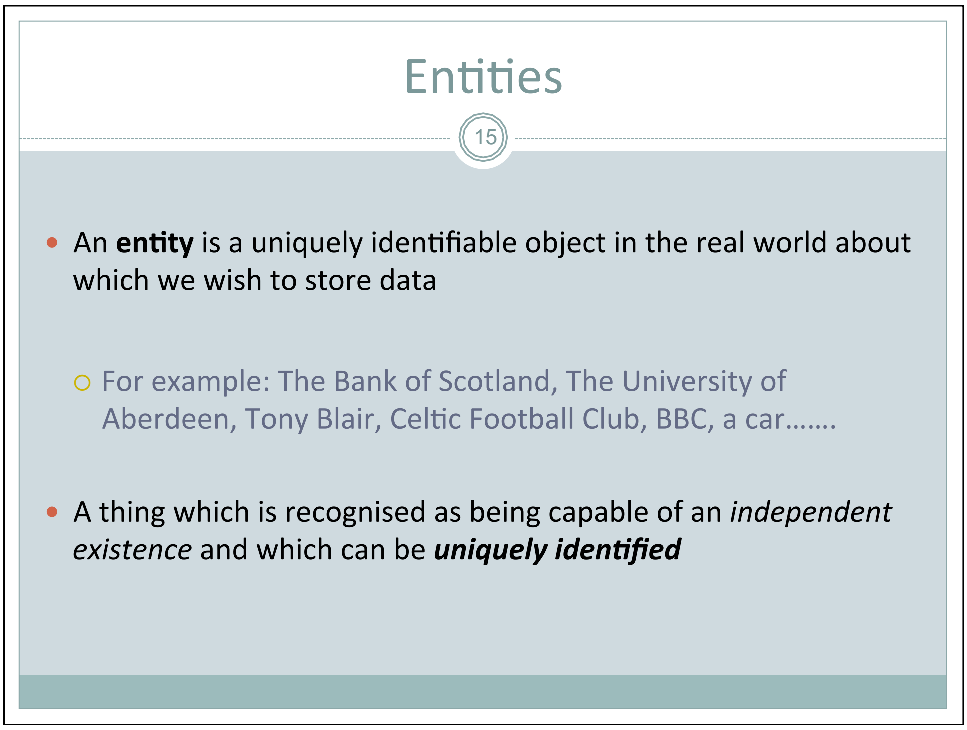
- Usually described using ***Entity-Relationship Diagrams***
  - Describes type of information to be stored in a database
  - Provides an overview and classifications of used terms and their relationships



The Entity-Relationship Model

14

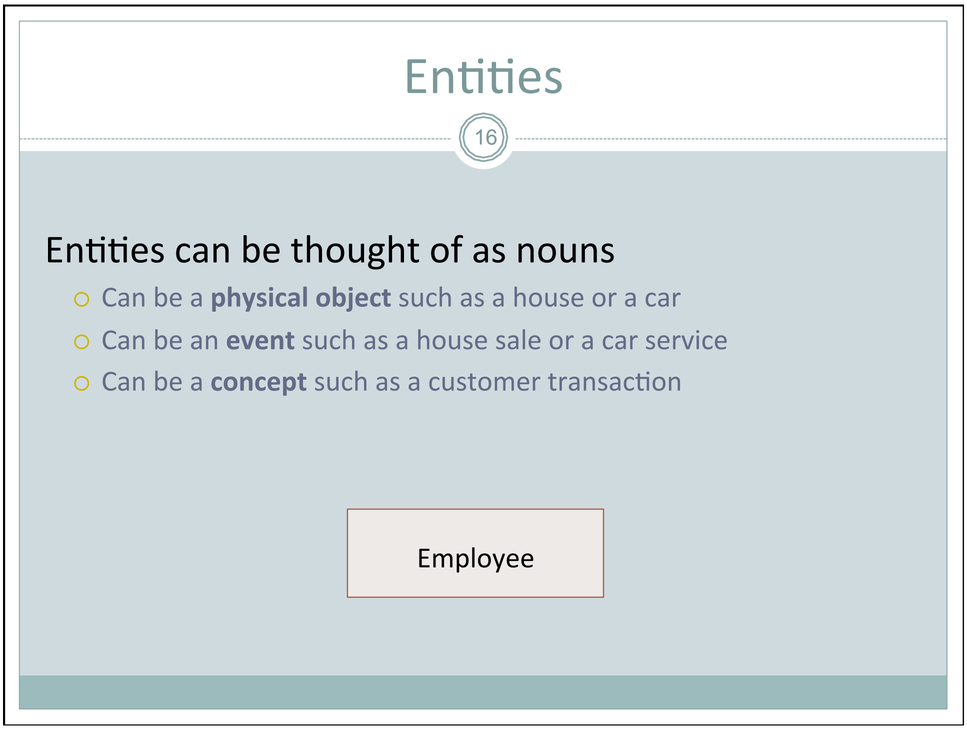
- Data in an ER Model is described in terms of three key concepts:
  - Entities
  - Relationships
  - Attributes



Entities

15

- An **entity** is a uniquely identifiable object in the real world about which we wish to store data
  - For example: The Bank of Scotland, The University of Aberdeen, Tony Blair, Celtic Football Club, BBC, a car…….
- A thing which is recognised as being capable of an *independent existence* and which can be ***uniquely identified***



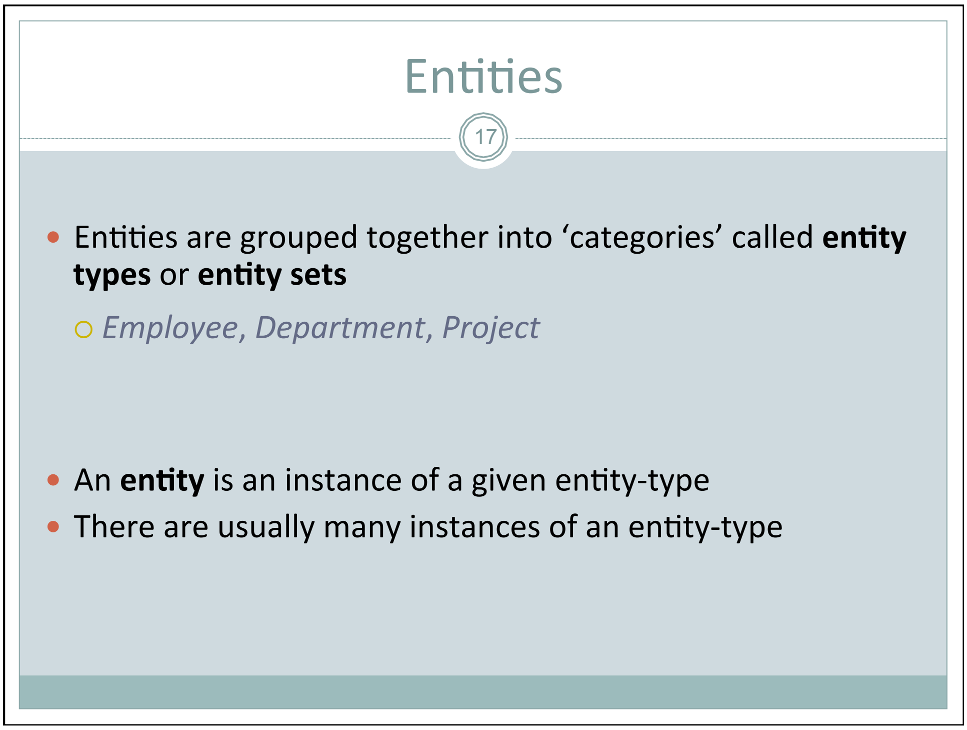
Entities

16

Entities can be thought of as nouns

- Can be a **physical object** such as a house or a car
- Can be an **event** such as a house sale or a car service
- Can be a **concept** such as a customer transaction

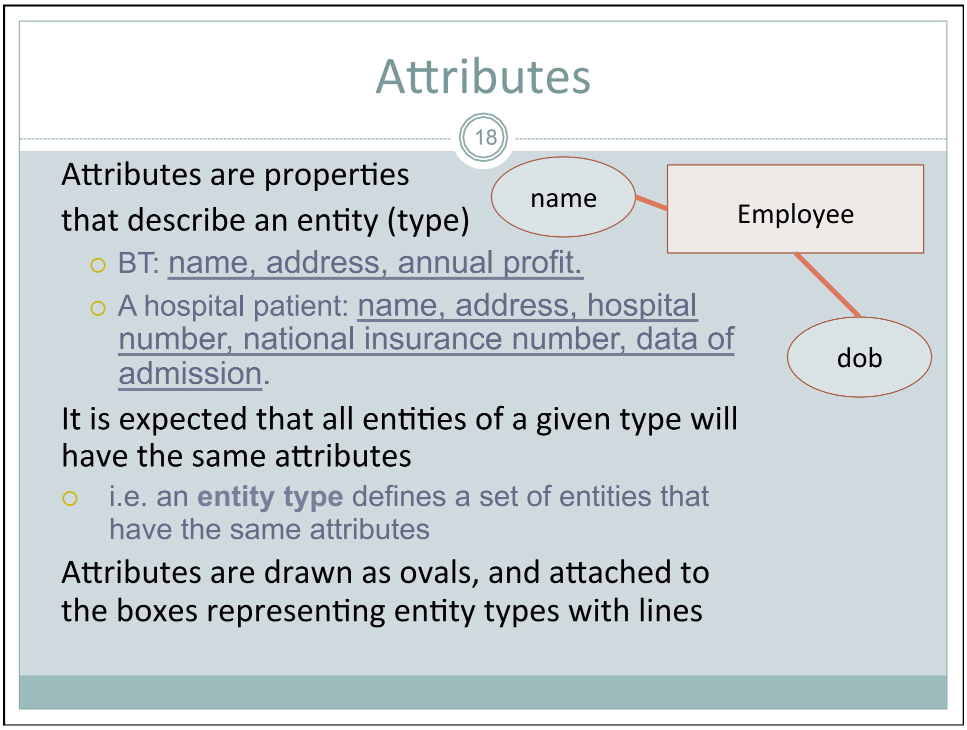
Employee



Entities

17

- Entities are grouped together into ‘categories’ called **entity types** or **entity sets**
  - *Employee, Department, Project*
- An **entity** is an instance of a given entity-type
- There are usually many instances of an entity-type



Attributes

18

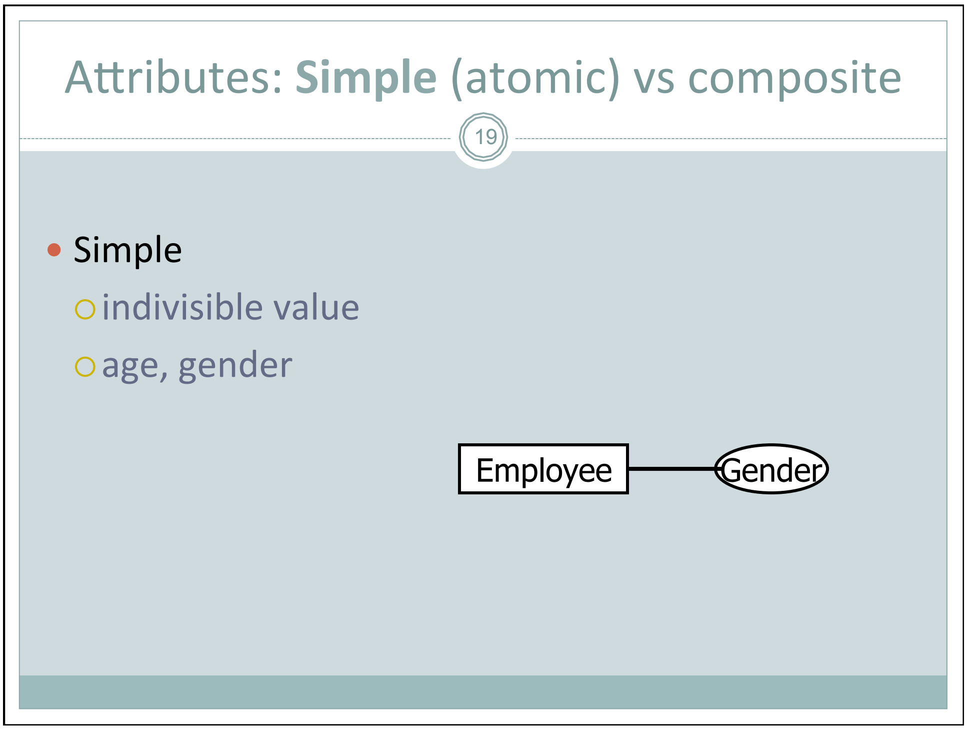
Attributes are properties that describe an entity (type)

- BT: name, address, annual profit.
- A hospital patient: name, address, hospital number, national insurance number, data of admission.

It is expected that all entities of a given type will have the same attributes

- i.e. an **entity type** defines a set of entities that have the same attributes

Attributes are drawn as ovals, and attached to the boxes representing entity types with lines



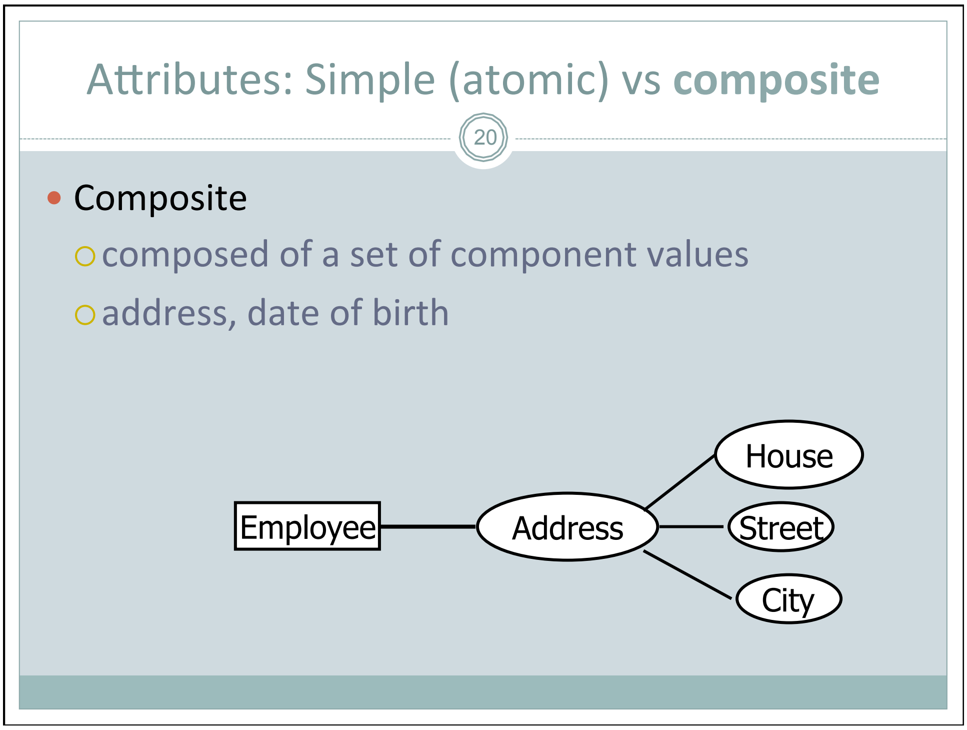
Attributes: **Simple** (atomic) vs composite

19

- Simple
  - indivisible value
  - age, gender

Employee

Gender



Attributes: Simple (atomic) vs **composite**

20

- Composite
  - composed of a set of component values
  - address, date of birth

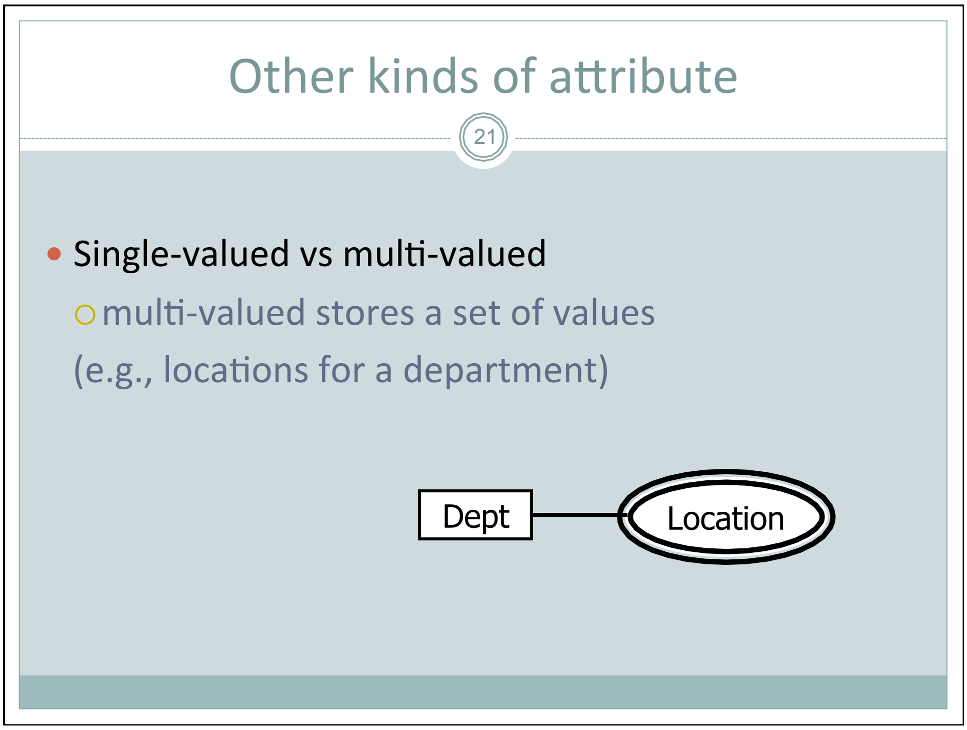
Employee

Address

House

Street

City



Other kinds of attribute

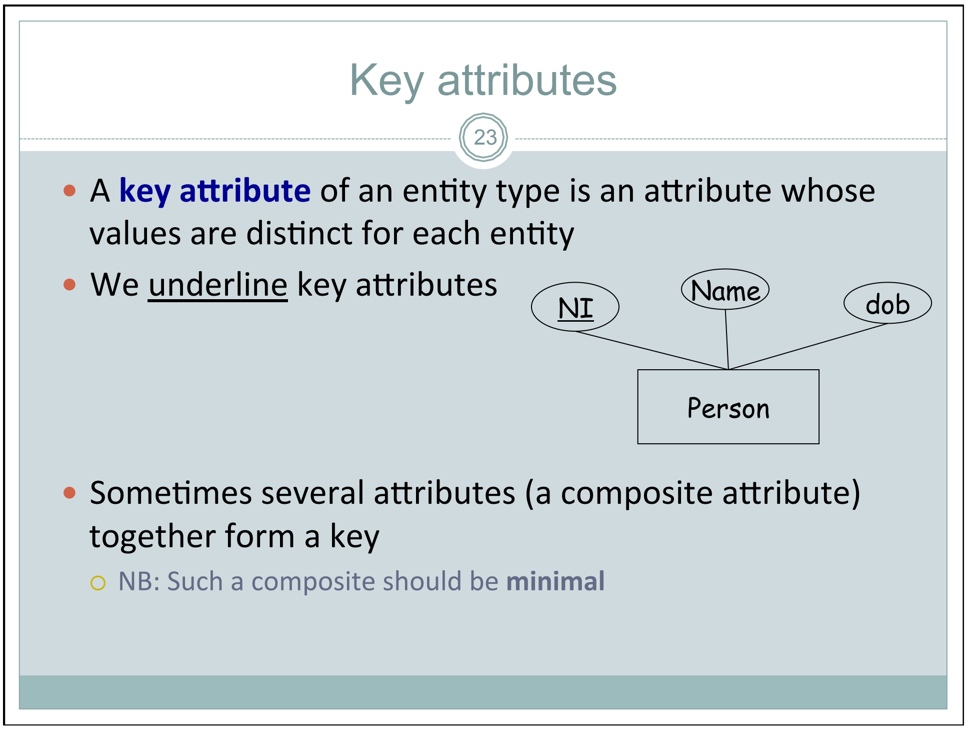
21

- Single-valued vs multi-valued
  - multi-valued stores a set of values

  (e.g., locations for a department)

Dept

Location



Key attributes

23

- A **key attribute** of an entity type is an attribute whose values are distinct for each entity
- We underline key attributes

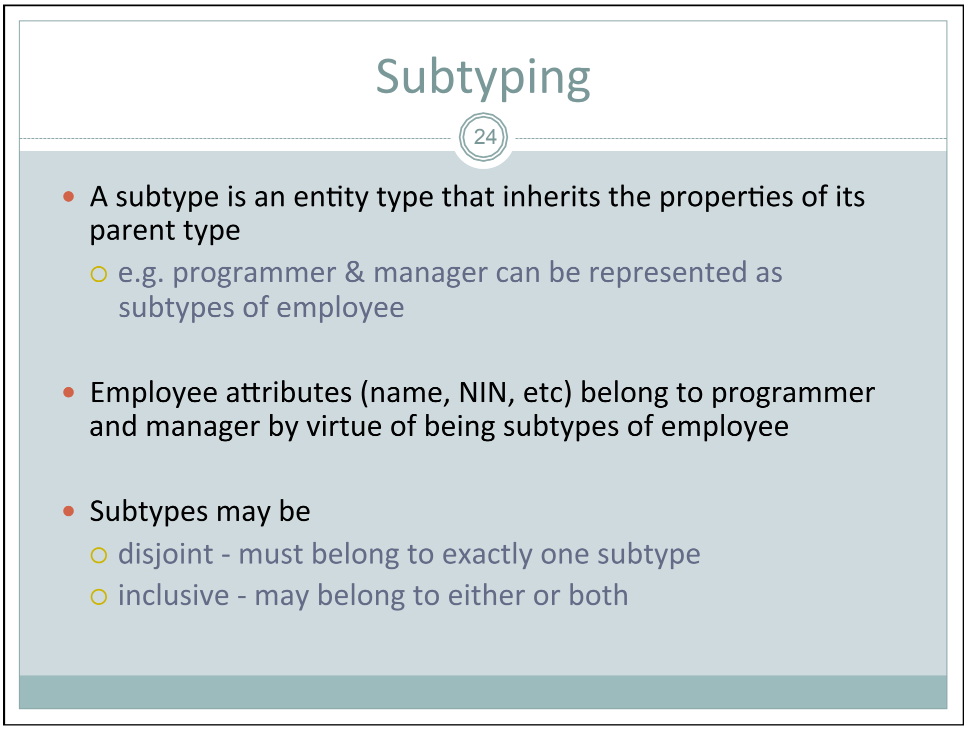
NI

Name

dob

Person

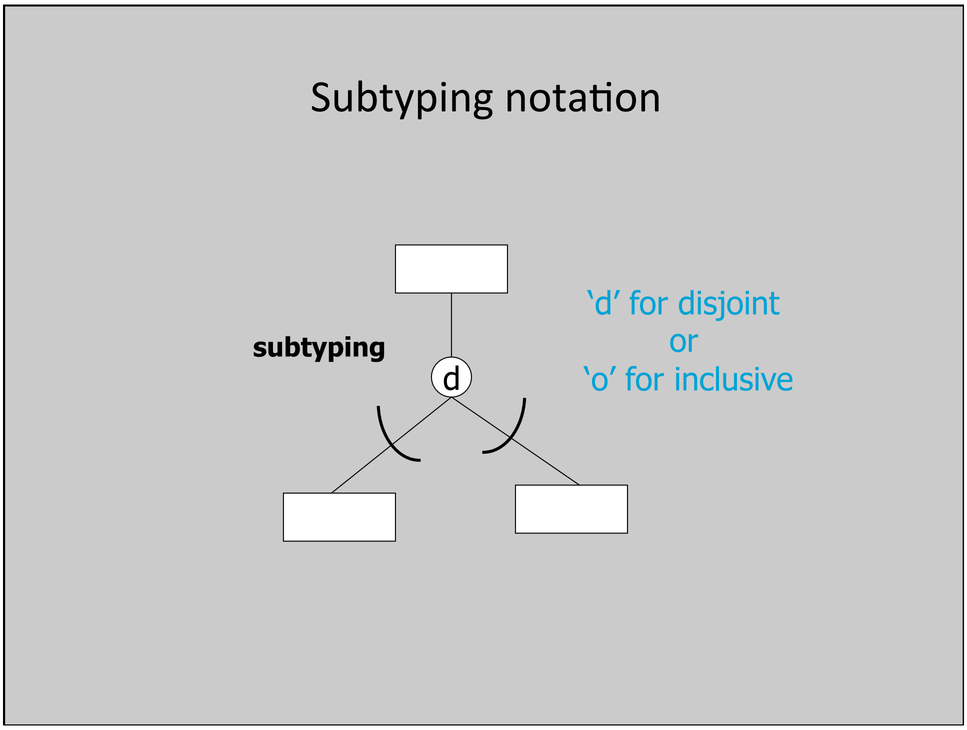
- Sometimes several attributes (a composite attribute) together form a key
  - NB: Such a composite should be **minimal**



Subtyping

24

- A subtype is an entity type that inherits the properties of its parent type
  - e.g. programmer & manager can be represented as subtypes of employee
- Employee attributes (name, NIN, etc) belong to programmer and manager by virtue of being subtypes of employee
- Subtypes may be
  - disjoint - must belong to exactly one subtype
  - inclusive - may belong to either or both



Subtyping notation

subtyping

‘d’ for disjoint
or
‘o’ for inclusive

Relationships

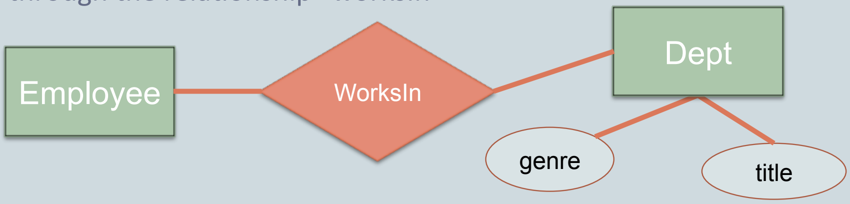
26

- Captures how two or more entities are related
- Can be thought of as verbs, linking two or more nouns
- Examples:
  - an *owns* relationship between a company and a computer
  - a *supervises* relationship between an employee and a department
  - a *performs* relationship between an artist and a song
  - a *proved* relationship between a mathematician and a theorem

Relationships

27

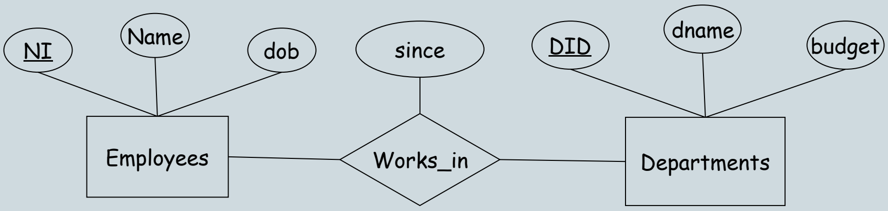
- **Relationships types** represent the *interaction* between entity types
  - For example the entities “employee” and “dept” might interact through the relationship “worksIn”



- Relationship types are represented by diamonds
- They connect the participating entity types with straight lines

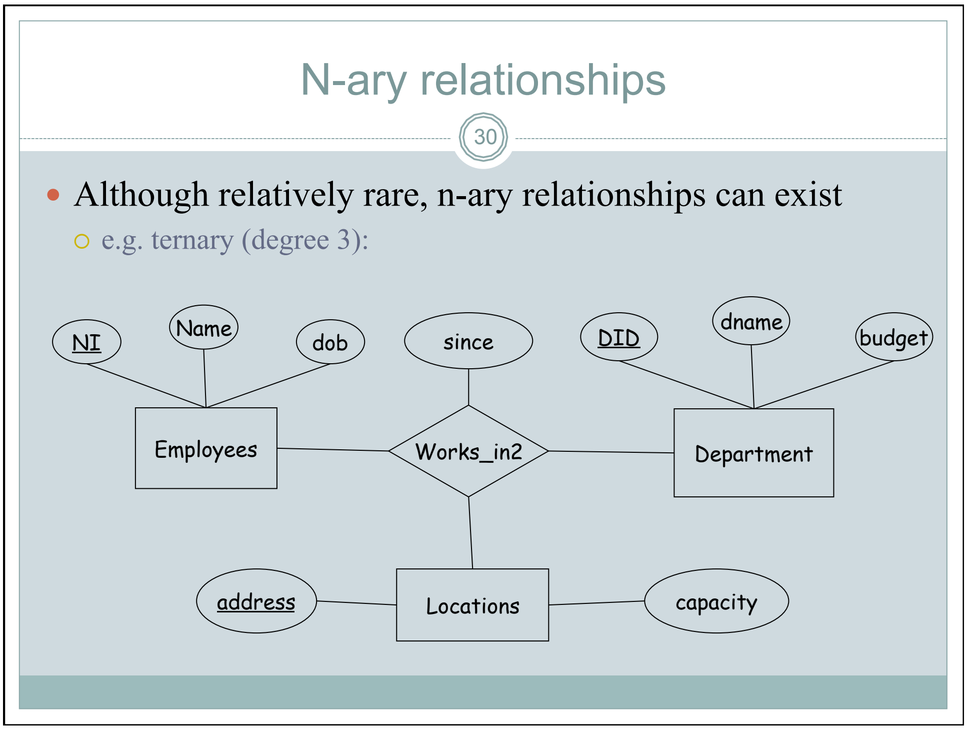
Relationship attributes

- Relationships can also have **attributes**
  - NB: A relationship must be uniquely determined by the entities, without reference to the relationship attributes



Relationship Degrees

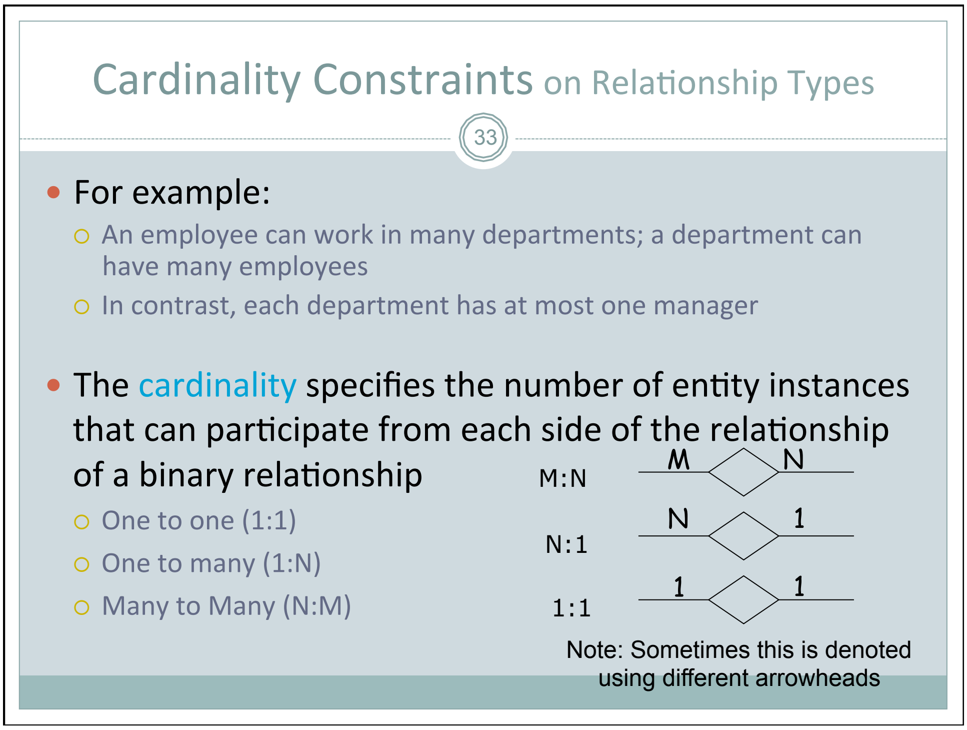
- The ***Degree*** of a relationship is the number of entity types participating
  - Binary relationships
    - 2 participating entity types
    - Employee works for Department
  - Ternary relationships
    - 3 participating entity types
    - a Manager manages a Project in a Department



N-ary relationships

30

- Although relatively rare, n-ary relationships can exist
  - e.g. ternary (degree 3):



Cardinality Constraints on Relationship Types

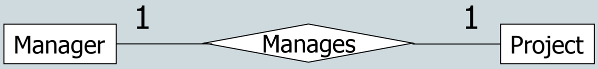
33

- For example:
  - An employee can work in many departments; a department can have many employees
  - In contrast, each department has at most one manager
- The cardinality specifies the number of entity instances that can participate from each side of the relationship of a binary relationship
  - One to one (1:1)
  - One to many (1:N)
  - Many to Many (N:M)

Note: Sometimes this is denoted using different arrowheads

Cardinality – 1:1

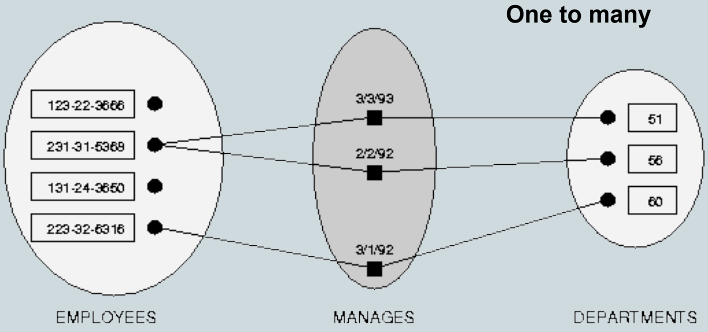
- One-to-one (1-1)



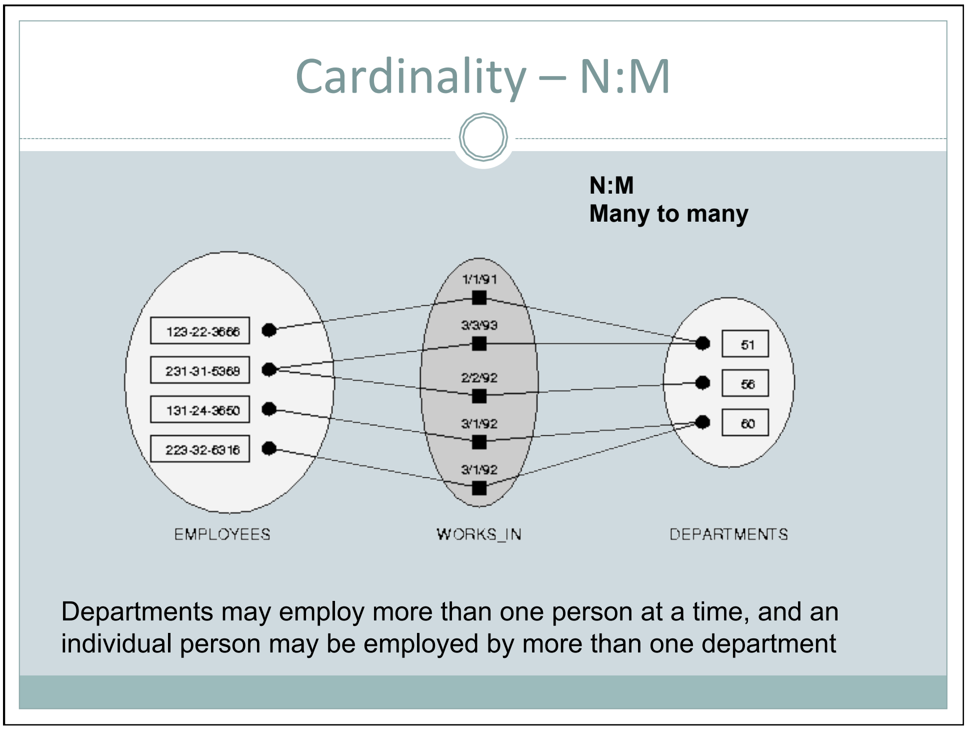
- Each manager manages ONLY one project
- Each project is managed by ONLY one manager

Cardinality – 1:N

1:N
One to many



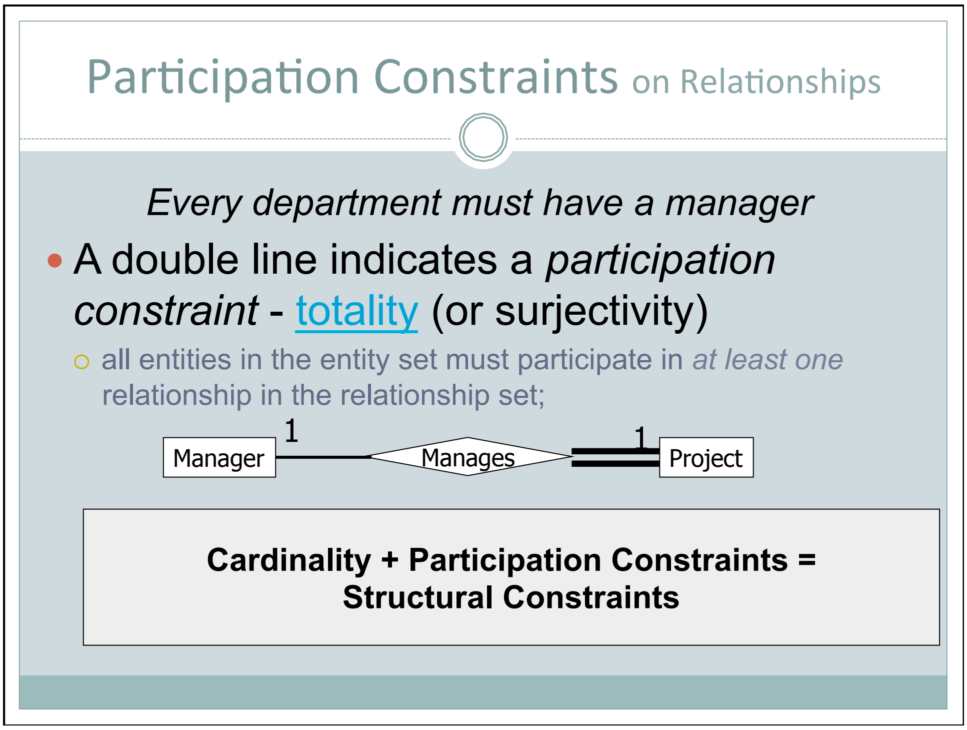
A department cannot have more than one manager
(but it may be that an individual manages multiple departments)



Cardinality – N:M

N:M
Many to many

Departments may employ more than one person at a time, and an individual person may be employed by more than one department



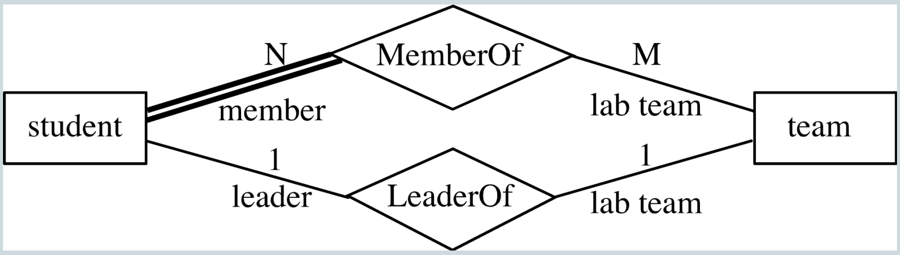
Participation Constraints on Relationships

Every department must have a manager

- A double line indicates a *participation constraint* - totality (or surjectivity)
  - all entities in the entity set must participate in *at least one* relationship in the relationship set;

Cardinality + Participation Constraints = Structural Constraints

Total Participation



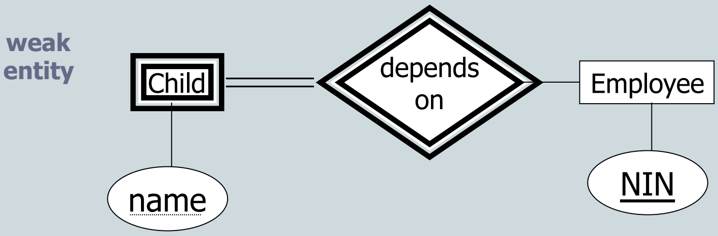
Every student must be a member of a team

A double line indicates the total participation constraint in an ER model

Note - the participation of *student* in *LeaderOf* is **partial**,
because a student *might* be a team leader

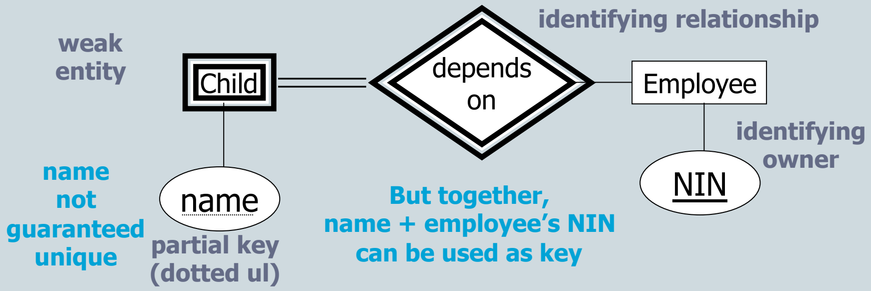
Weak Entity Types

- Depend on other entities to guarantee uniqueness
- Do not have primary key (attributes) of their own



Weak Entity Types

- Depend on other entities to guarantee uniqueness
- Do not have primary keys of their own



weak entity

Child

depends on

identifying relationship

Employee

identifying owner

NIN

name not guaranteed unique

name

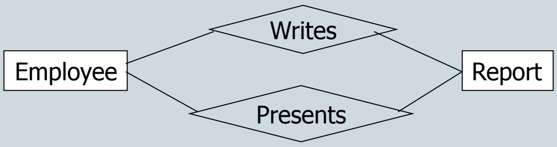
partial key (dotted ul)

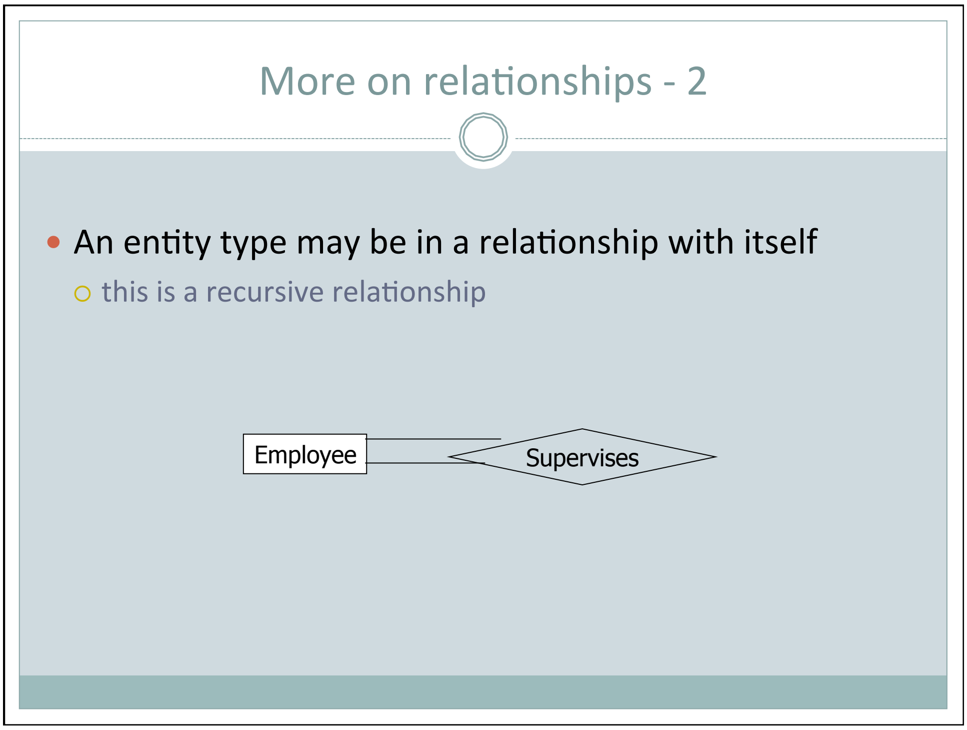
But together, name + employee's NIN can be used as key

- Weak entity set must have total participation in this identifying relationship set.

More on relationships - 1

- There may be more than one relationship between entity types



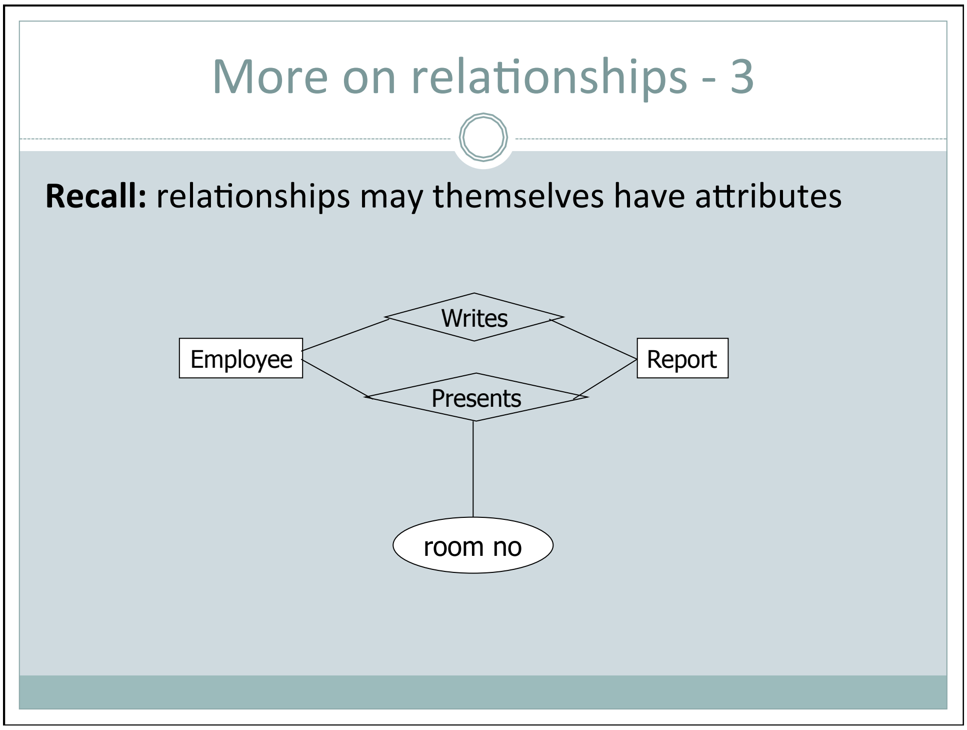


More on relationships - 2

- An entity type may be in a relationship with itself
  - this is a recursive relationship

Employee

Supervises



More on relationships - 3

Recall: relationships may themselves have attributes

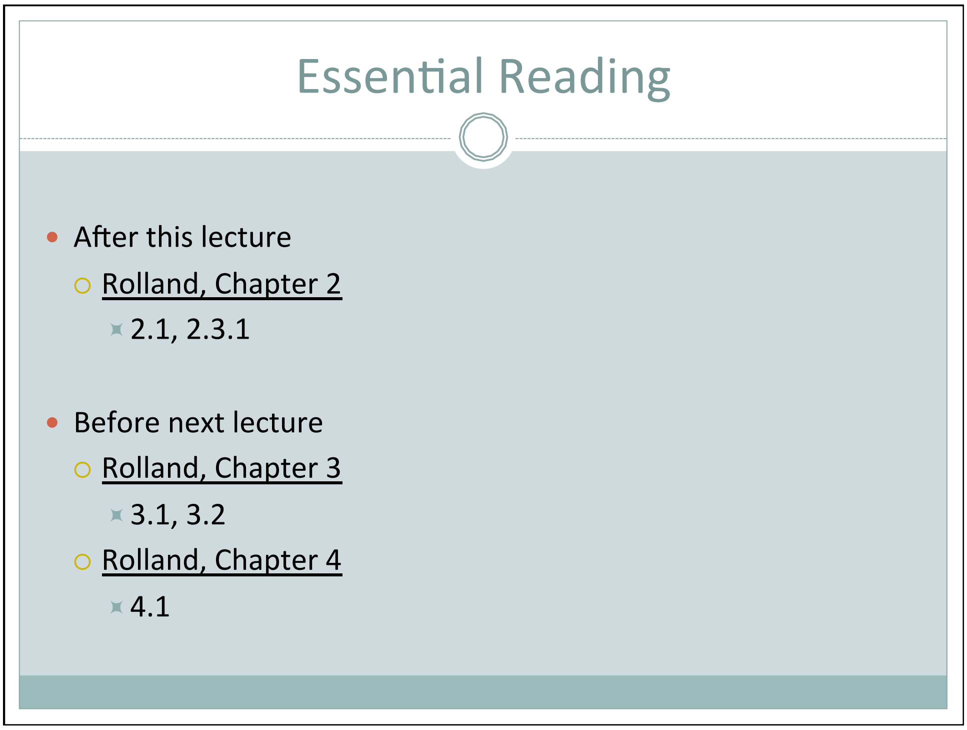
Employee

Writes

Presents

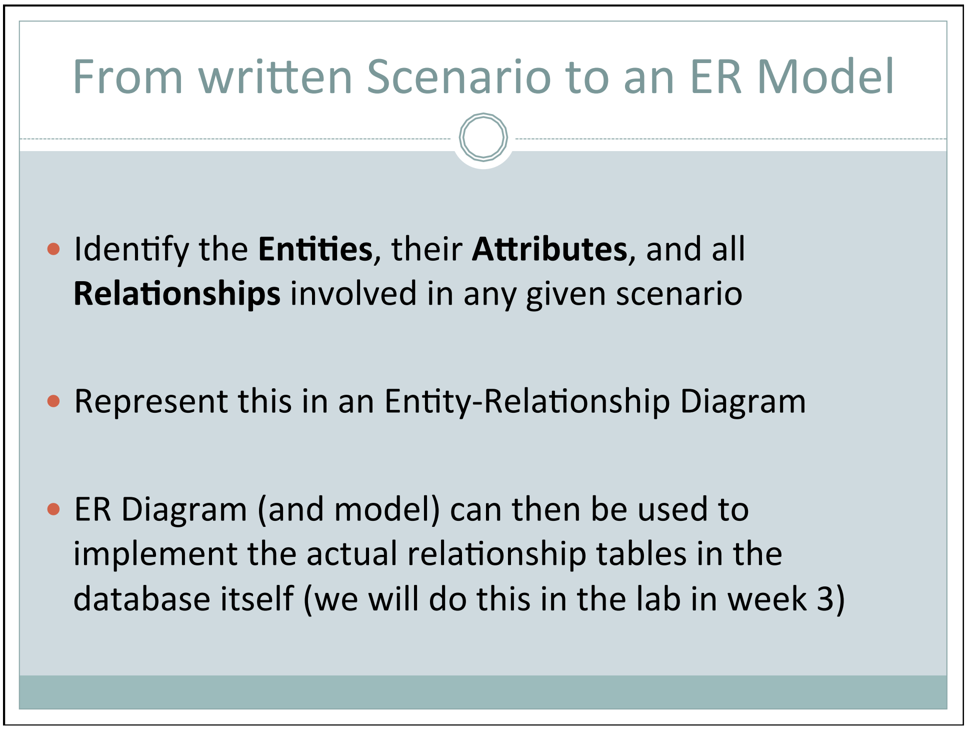
Report

room no



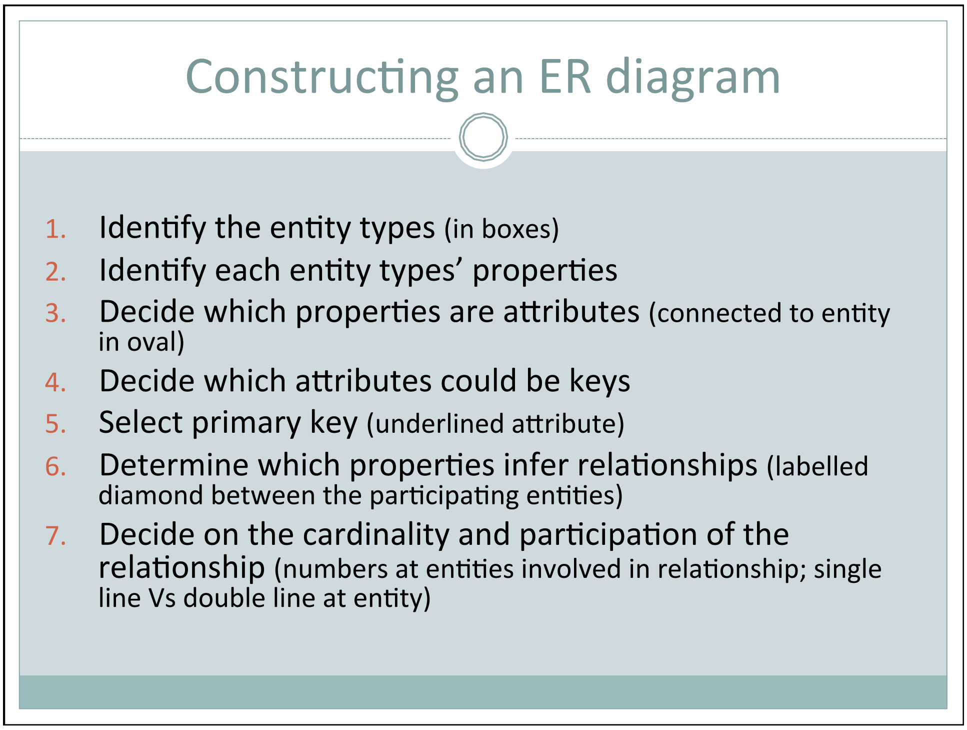
Essential Reading

- After this lecture
  - Rolland, Chapter 2
    - 2.1, 2.3.1
- Before next lecture
  - Rolland, Chapter 3
    - 3.1, 3.2
  - Rolland, Chapter 4
    - 4.1



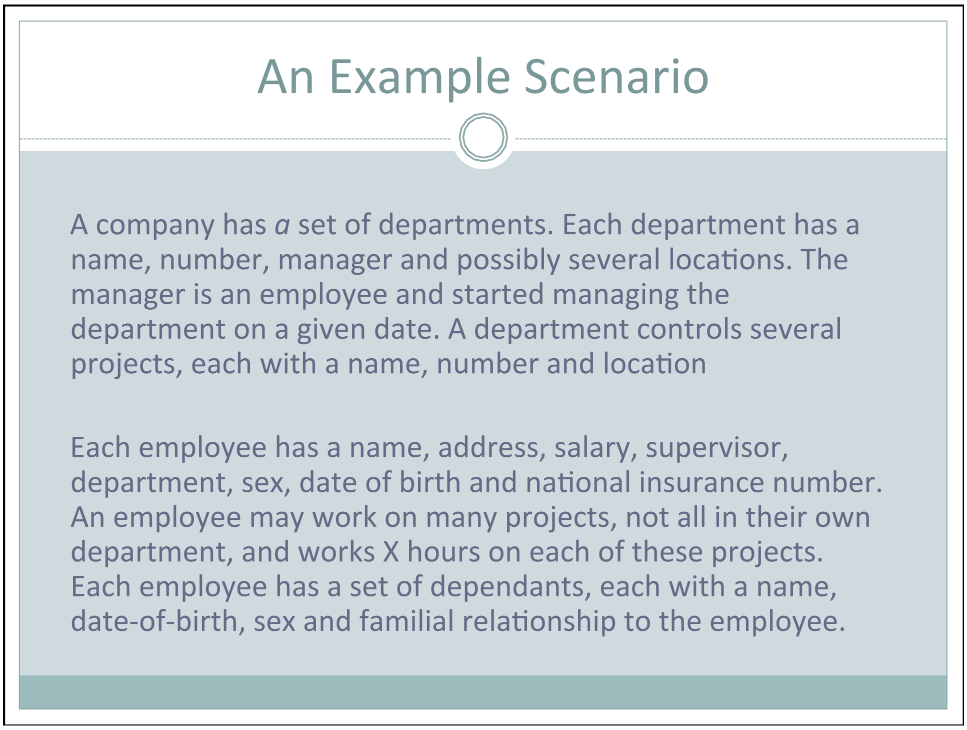
From written Scenario to an ER Model

- Identify the **Entities**, their **Attributes**, and all **Relationships** involved in any given scenario
- Represent this in an Entity-Relationship Diagram
- ER Diagram (and model) can then be used to implement the actual relationship tables in the database itself (we will do this in the lab in week 3)



Constructing an ER diagram

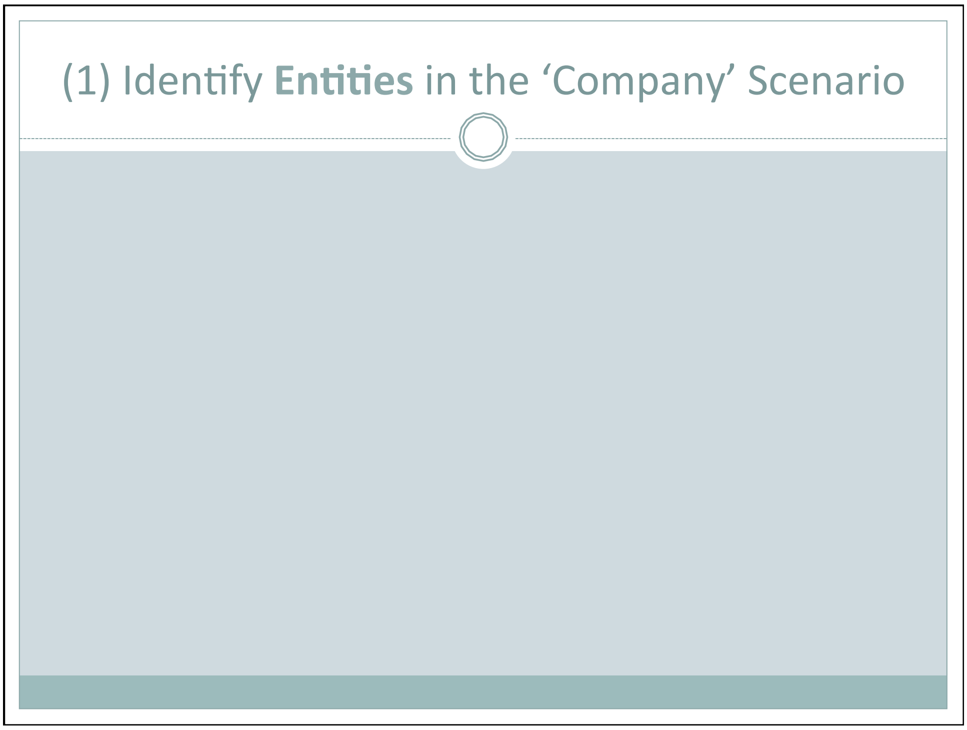
1. Identify the entity types (in boxes)
2. Identify each entity types' properties
3. Decide which properties are attributes (connected to entity in oval)
4. Decide which attributes could be keys
5. Select primary key (underlined attribute)
6. Determine which properties infer relationships (labelled diamond between the participating entities)
7. Decide on the cardinality and participation of the relationship (numbers at entities involved in relationship; single line Vs double line at entity)



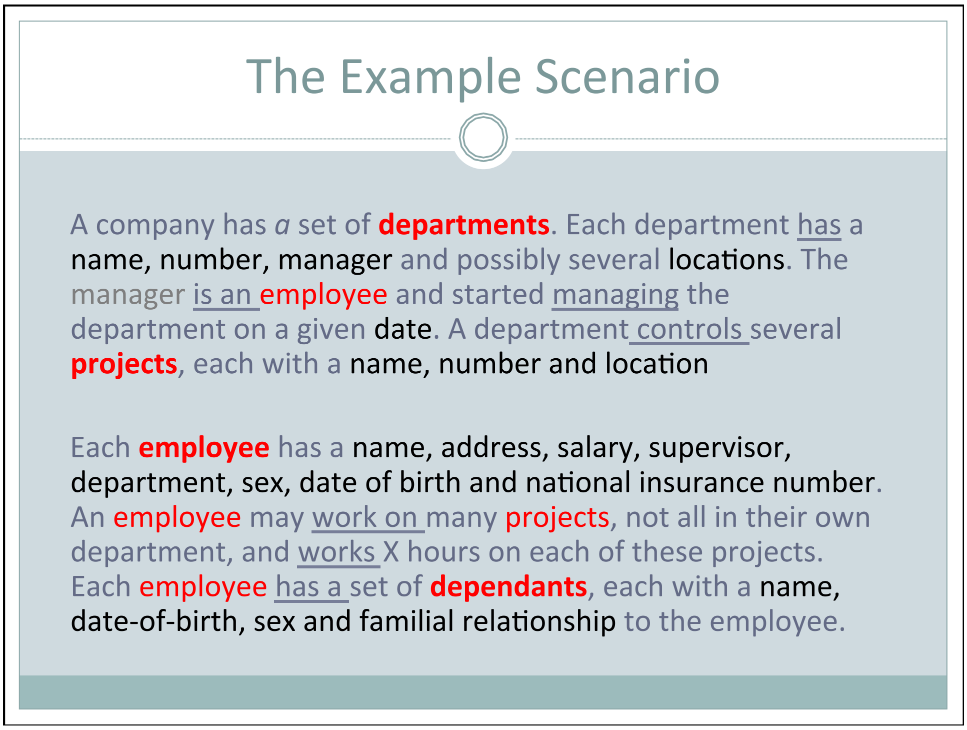
An Example Scenario

A company has *a* set of departments. Each department has a name, number, manager and possibly several locations. The manager is an employee and started managing the department on a given date. A department controls several projects, each with a name, number and location

Each employee has a name, address, salary, supervisor, department, sex, date of birth and national insurance number. An employee may work on many projects, not all in their own department, and works X hours on each of these projects. Each employee has a set of dependants, each with a name, date-of-birth, sex and familial relationship to the employee.



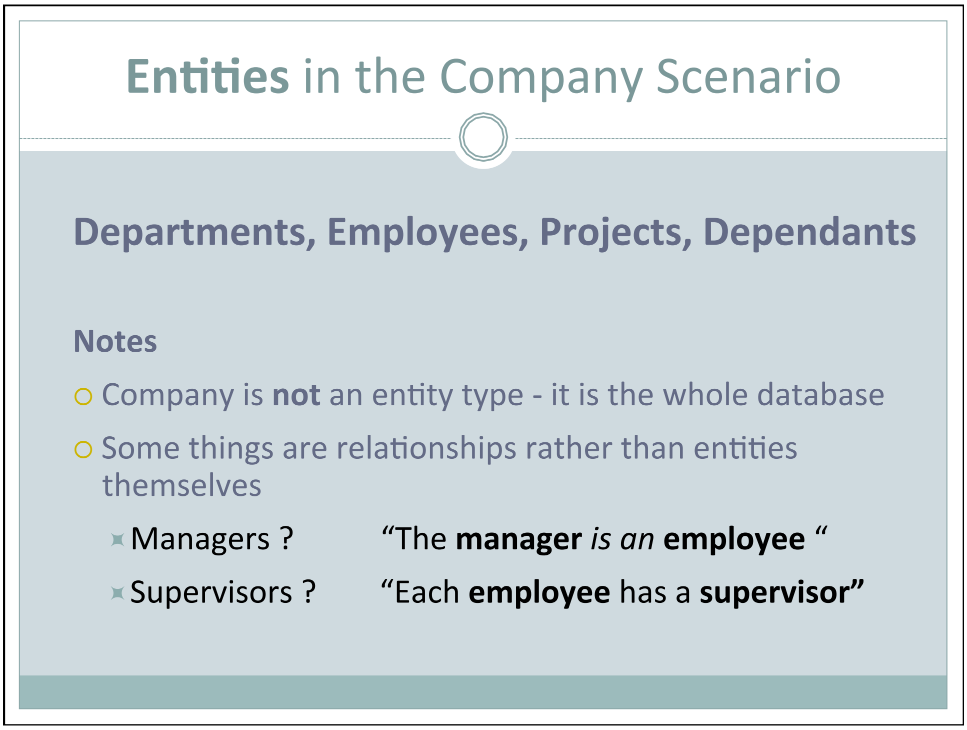
(1) Identify **Entities** in the 'Company' Scenario



The Example Scenario

A company has *a* set of **departments**. Each department has a name, number, manager and possibly several locations. The manager is an employee and started managing the department on a given date. A department controls several **projects**, each with a name, number and location

Each **employee** has a name, address, salary, supervisor, department, sex, date of birth and national insurance number. An employee may work on many projects, not all in their own department, and works X hours on each of these projects. Each employee has a set of **dependants**, each with a name, date-of-birth, sex and familial relationship to the employee.

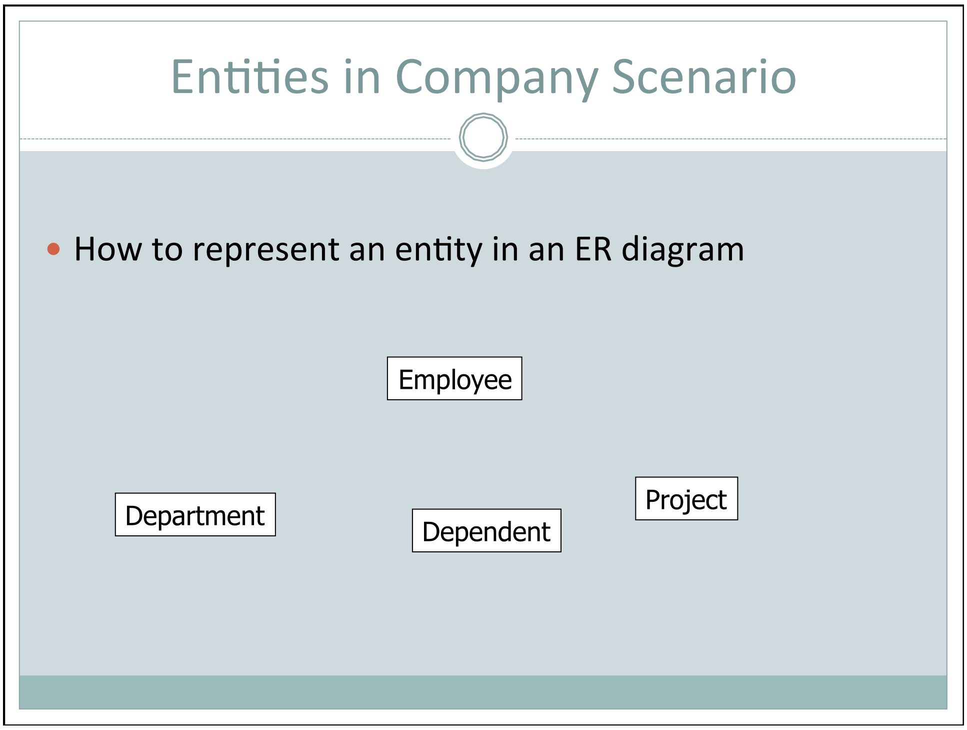


Entities in the Company Scenario

Departments, Employees, Projects, Dependants

Notes

- Company is **not** an entity type - it is the whole database
- Some things are relationships rather than entities themselves
  - Managers ? “The **manager** *is an* **employee** “
  - Supervisors ? “Each **employee** has a **supervisor”**



Entities in Company Scenario

- How to represent an entity in an ER diagram

Employee

Department

Dependent

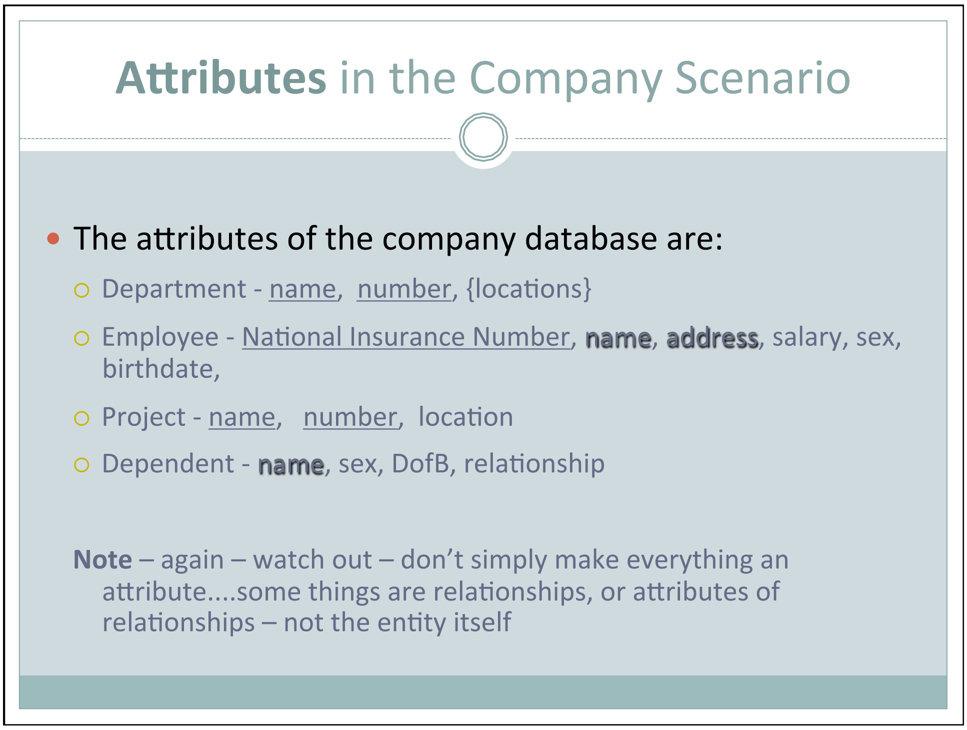
Project

(2) Identify **Attributes** in Company Scenario

The Example Scenario

A company has *a* set of departments. Each department has a **name**, **number**, manager and possibly several **locations**. The manager is an employee and started managing the department on a **given date**. A department controls several projects, each with a **name**, **number** and **location.**

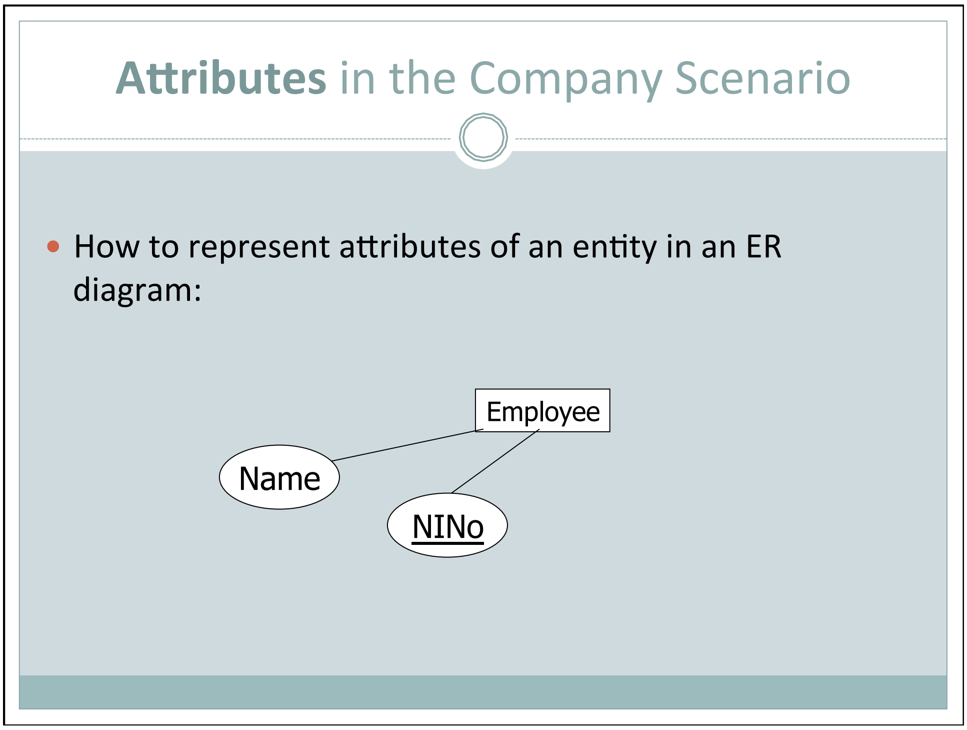
Each employee has a **name**, **address**, **salary**, supervisor, department, **sex**, **date of birth** and **national insurance number**. An employee may work on many projects, not all in their own department, and works X hours on each of these projects. Each employee has a set of dependants, each with a **name**, **date-of-birth**, **sex** and **familial relationship** to the employee.



Attributes in the Company Scenario

- The attributes of the company database are:
  - Department - name, number, {locations}
  - Employee - National Insurance Number, **name**, **address**, salary, sex, birthdate,
  - Project - name, number, location
  - Dependent - **name**, sex, DofB, relationship

Note – again – watch out – don't simply make everything an attribute....some things are relationships, or attributes of relationships – not the entity itself



Attributes in the Company Scenario

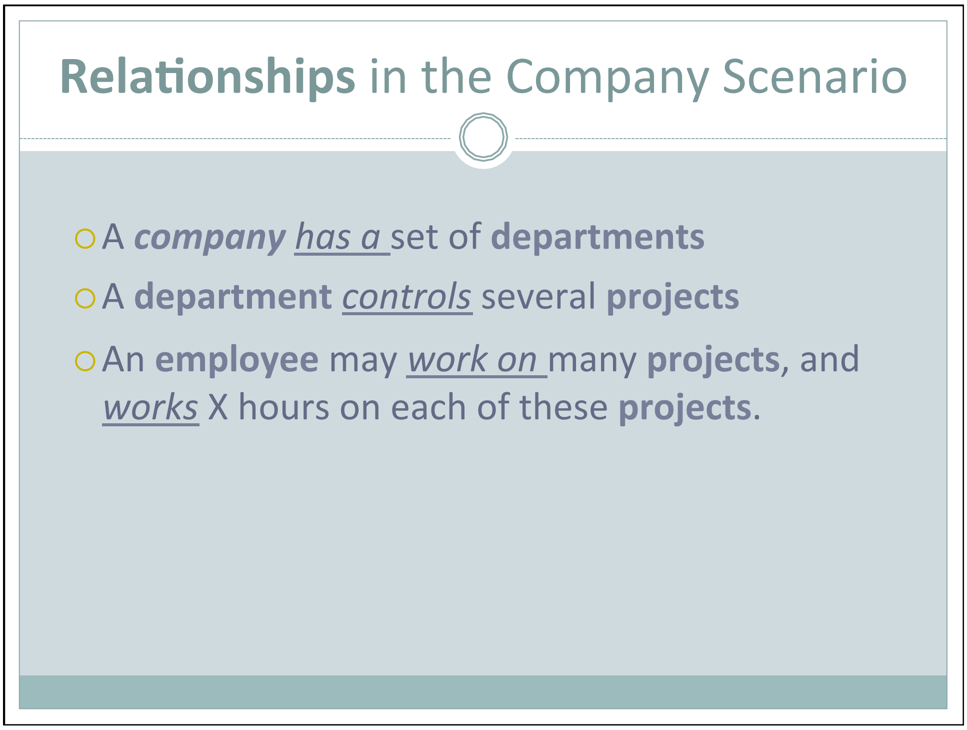
- How to represent attributes of an entity in an ER diagram:

(3) Identify Relationships in Company Scenario

The Example Scenario

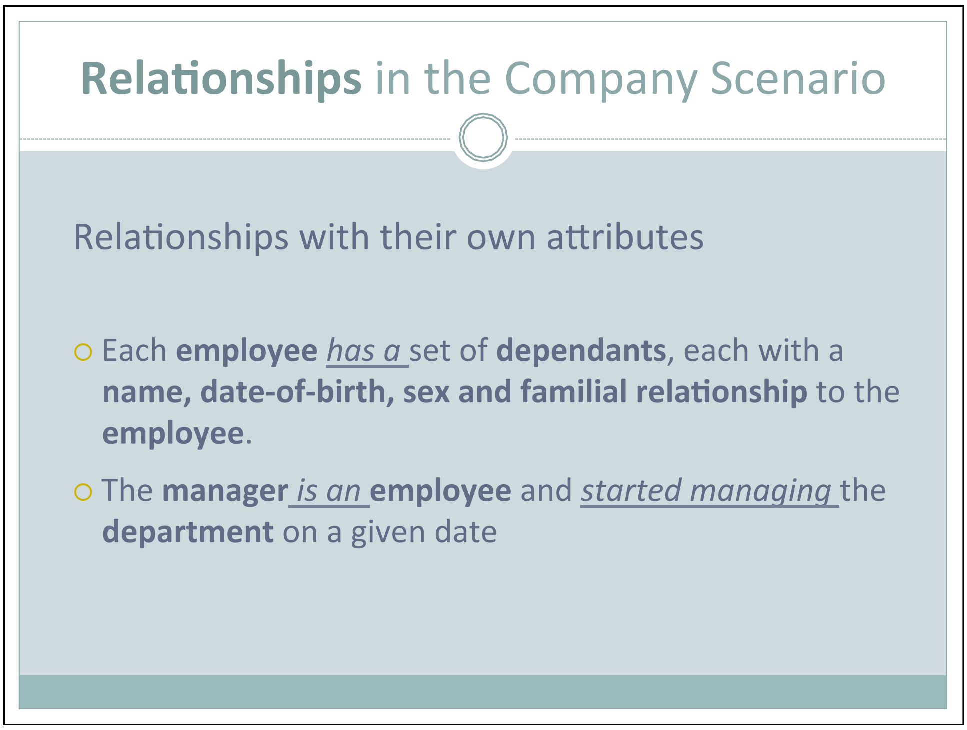
A ***company*** has *a* set of ***departments***. Each department has a name, number, manager and possibly several locations. The manager is an employee and started managing the department on a given date. A ***department*** controls several ***projects***, each with a name, number and location

Each employee has a name, address, salary, supervisor, department, sex, date of birth and national insurance number. An ***employee*** may work on many ***projects***, not all in their own department, and works X hours on each of these projects. Each employee has a set of dependants, each with a name, date-of-birth, sex and familial relationship to the employee.



Relationships in the Company Scenario

- A ***company*** *has a* set of **departments**
- A **department** *controls* several **projects**
- An **employee** may *work on* many **projects**, and *works* X hours on each of these **projects**.

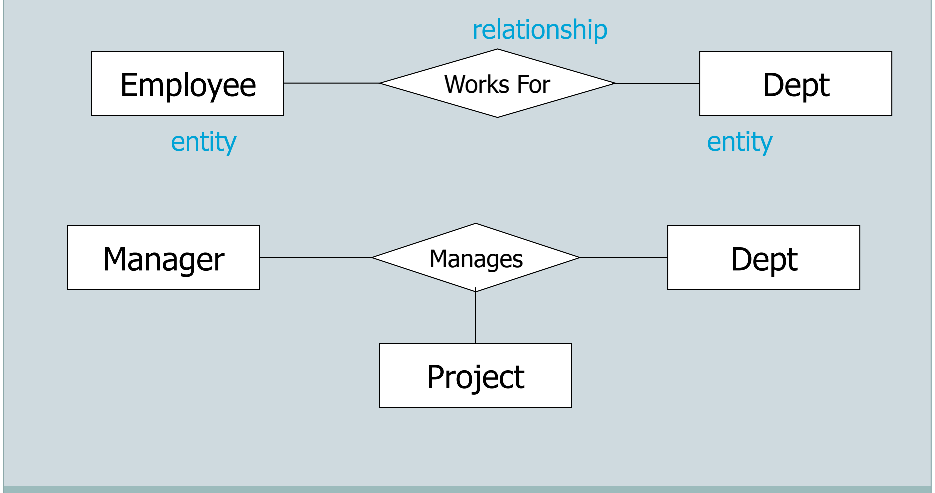


Relationships in the Company Scenario

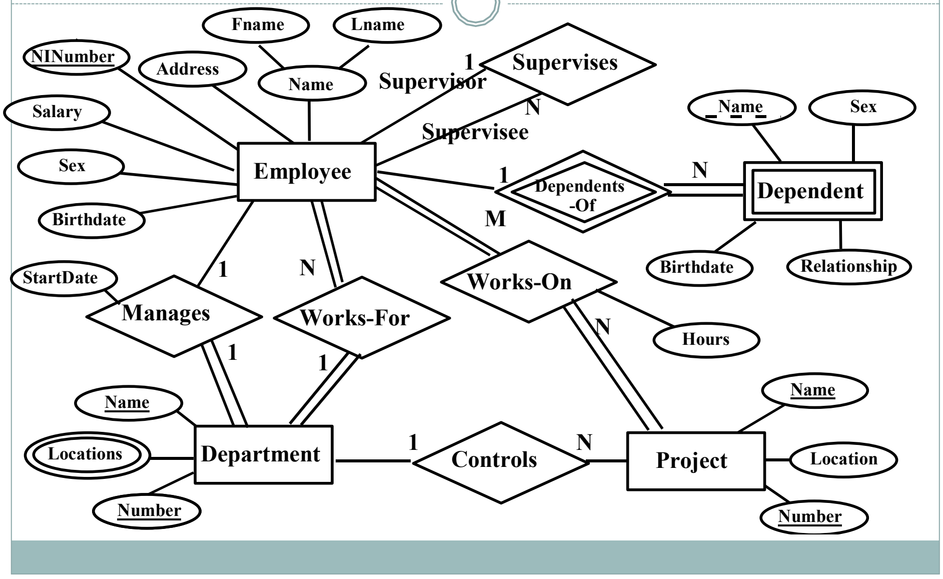
Relationships with their own attributes

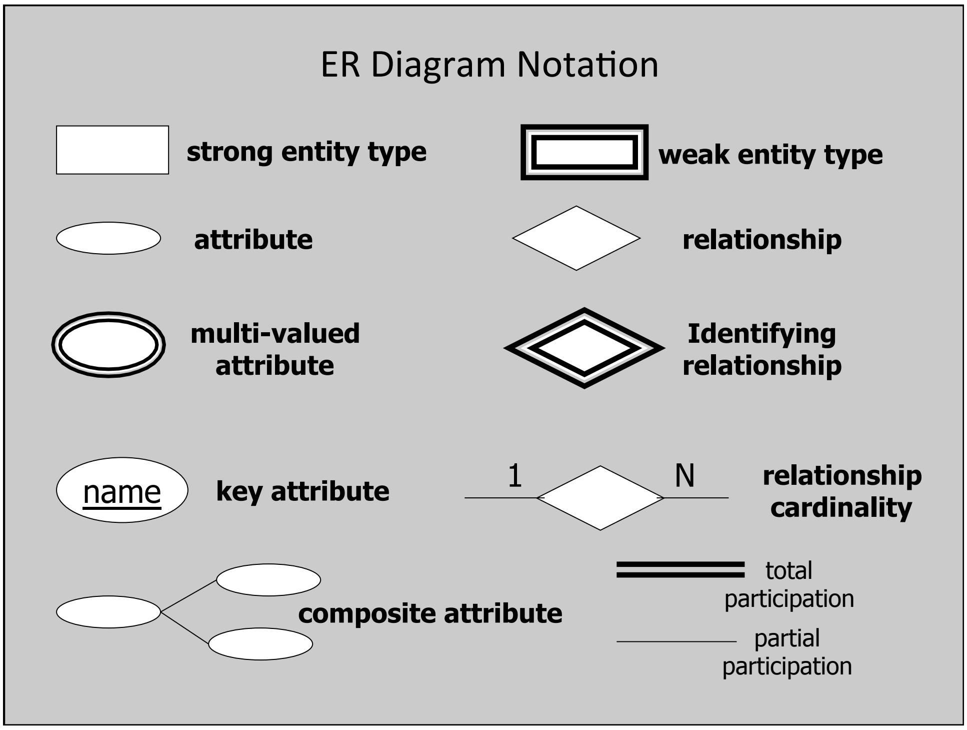
- Each **employee** *has a* set of **dependants**, each with a **name, date-of-birth, sex and familial relationship** to the **employee**.
- The **manager** *is an* **employee** and *started managing* the **department** on a given date

Representing Relationships in an ER Diagram

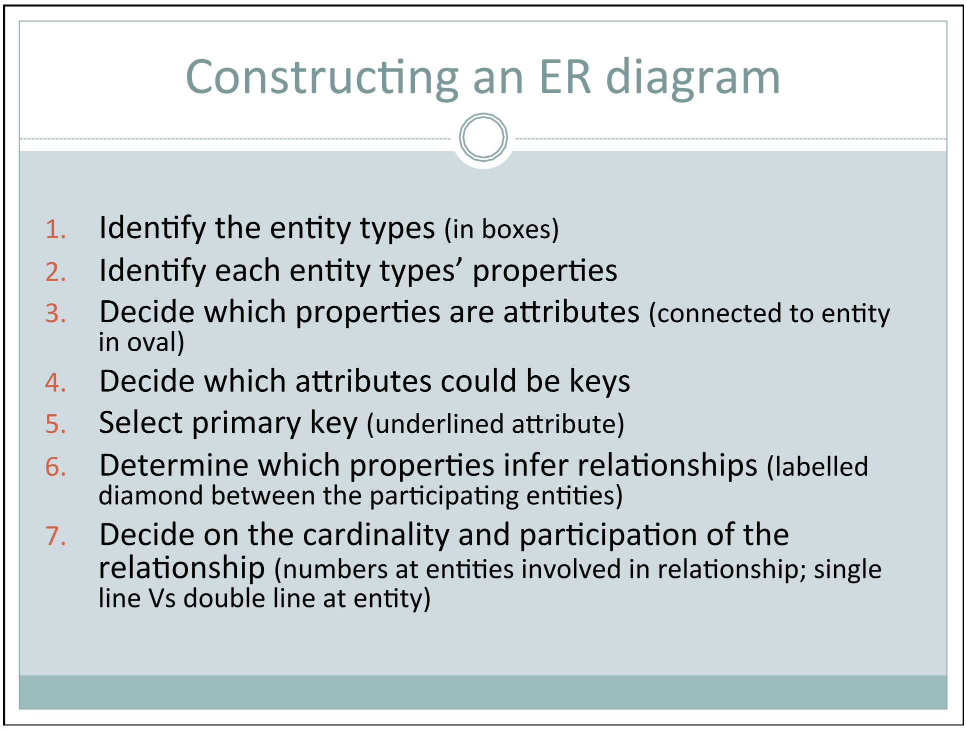


ER Diagram for the Company Database



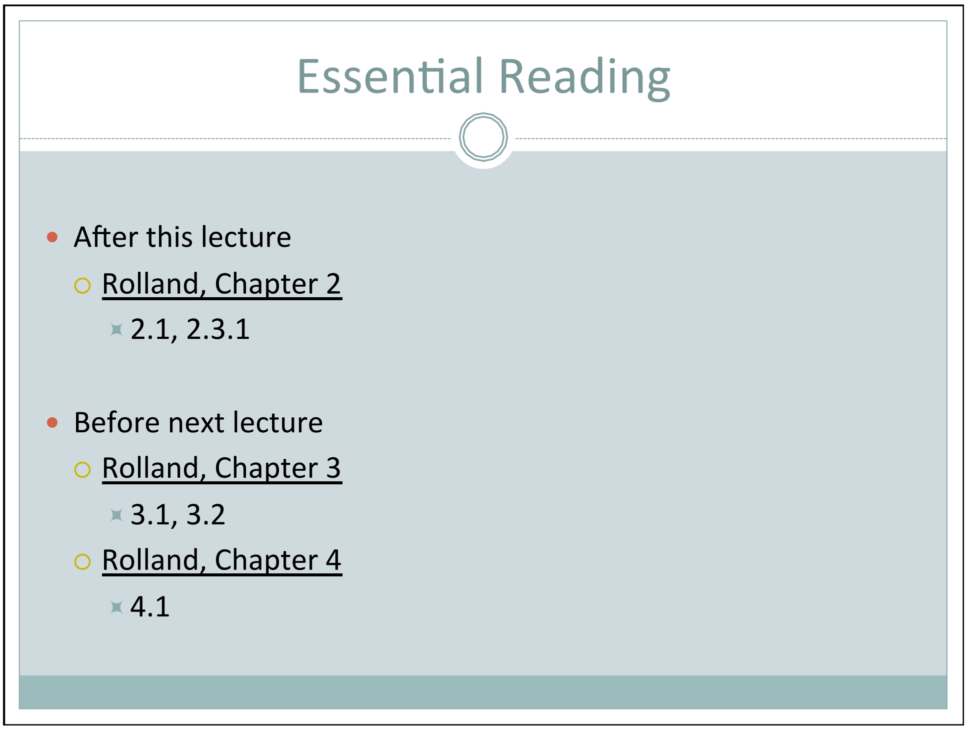


ER Diagram Notation



Constructing an ER diagram

1. Identify the entity types (in boxes)
2. Identify each entity types' properties
3. Decide which properties are attributes (connected to entity in oval)
4. Decide which attributes could be keys
5. Select primary key (underlined attribute)
6. Determine which properties infer relationships (labelled diamond between the participating entities)
7. Decide on the cardinality and participation of the relationship (numbers at entities involved in relationship; single line Vs double line at entity)



Essential Reading

- After this lecture
  - Rolland, Chapter 2
    - 2.1, 2.3.1
- Before next lecture
  - Rolland, Chapter 3
    - 3.1, 3.2
  - Rolland, Chapter 4
    - 4.1