Lecture 12: More On ER Modelling

CS1106/CS6503– Introduction to Relational Databases

Dr Kieran T. Herley

2019-2020

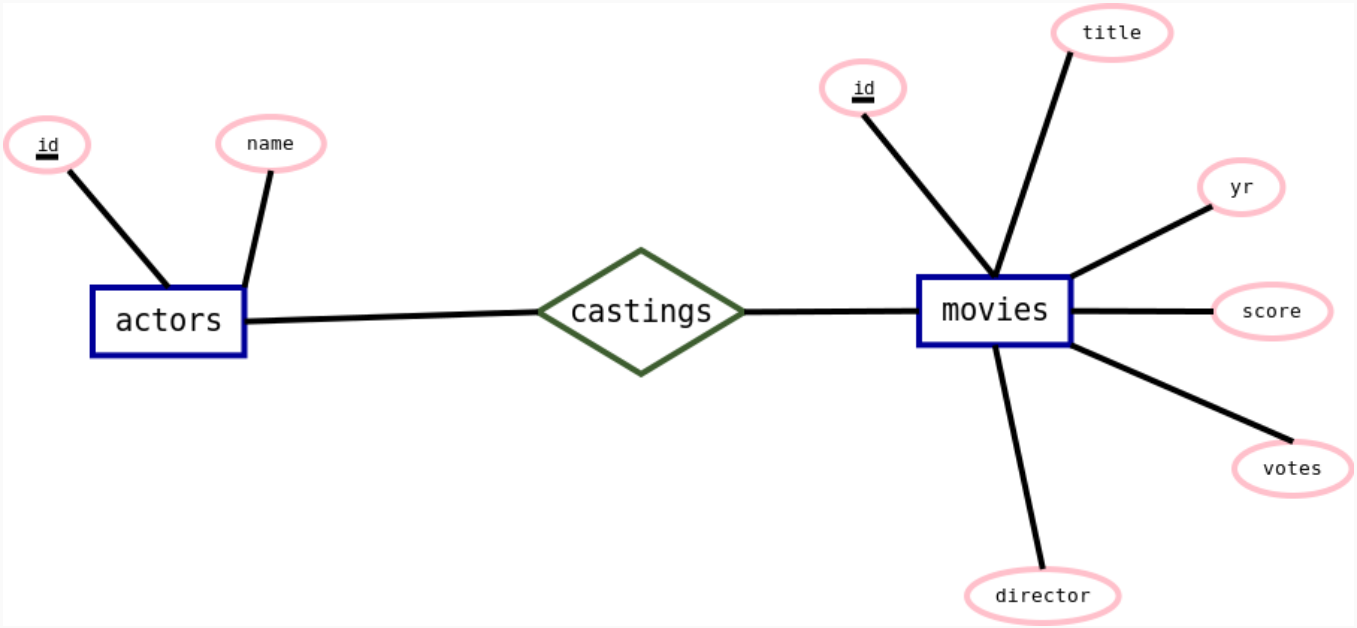
School of Computer Science & Information Technology
University College Cork

Summary

Development of ER to model a simple company database.

ER Diagrams

ER Diagram for Our Movies DB



DB Design Case Study

Database for DB1106 Consultants

- **Company** organized into **departments**, each spread over several **locations**
- Each department has several **projects**
- **Employees** are assigned to departments and work on several projects (possibly from other departments)
- Each employee may have a number of **dependents**

DB1106 Consultants cont'd

Need to keep track of:

- Which employees work in which departments
- Which employees manage which departments
- Which employees supervise which employees
- Which departments are located where
- Which projects are located where
- Which employees work on which projects (and for how many hours)
- Which employees have which dependents

Notes

Data to be maintained:

Employees name, pps number (unique), gender, data of birth and salary; Also his department, project (and hours worked), direct supervisor

Departments name, number (unique); Also employee who manages, department (multiple) locations

Projects name, number (unique); Also controlling department and (single) location

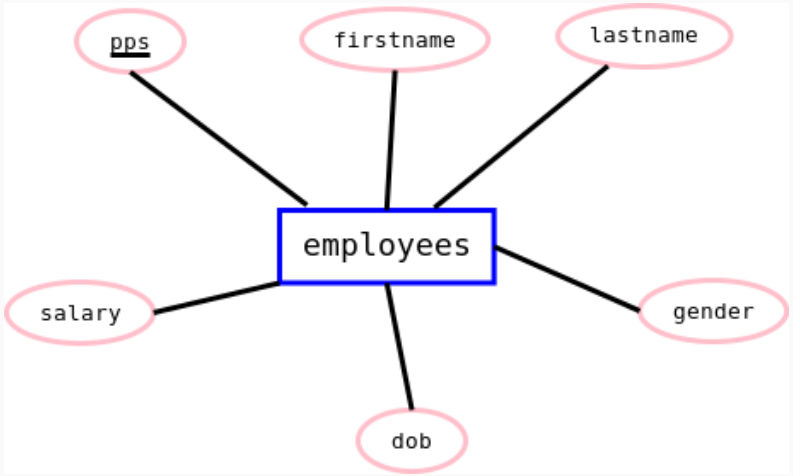
Locations name

Dependents name, gender, data of birth, relationship to employee; Also employee

Towards An ER Diagram

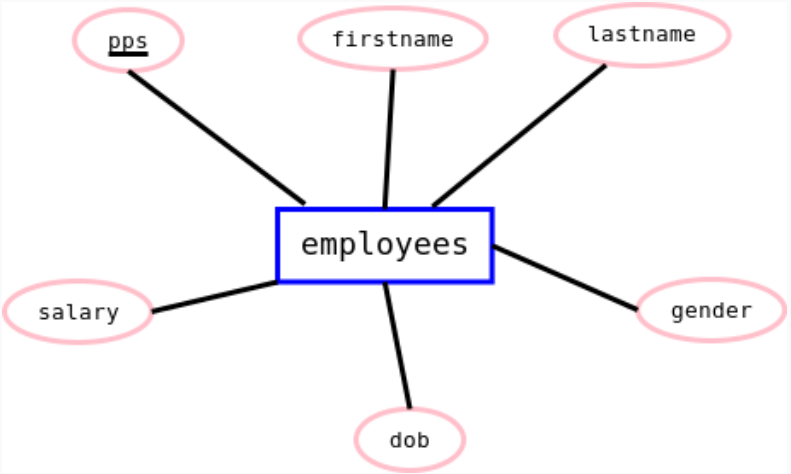
- Judgement required to determine appropriate entities and relationships
- Rough rule of thumb:
  - Candidates for entity sets often conveniently describable using nouns e.g. employees, departments, etc.
  - Candidates for relationships often conveniently describable using verbs e.g. supervises, works on, etc.

Entities in DB

- 

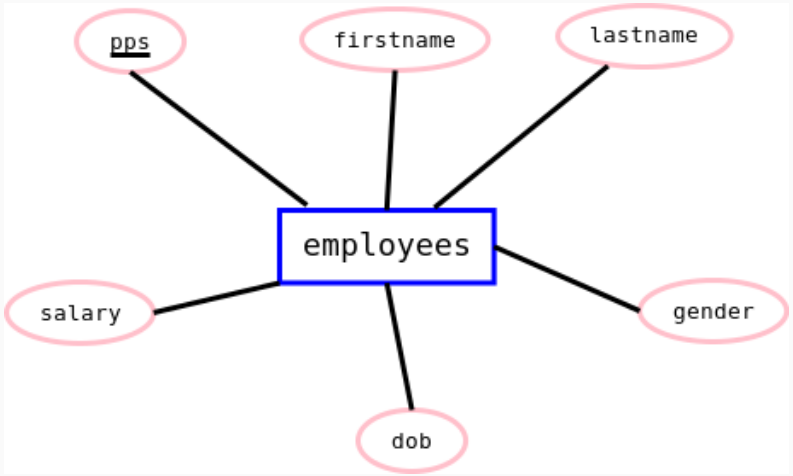
- Others:

Entities in DB

- 

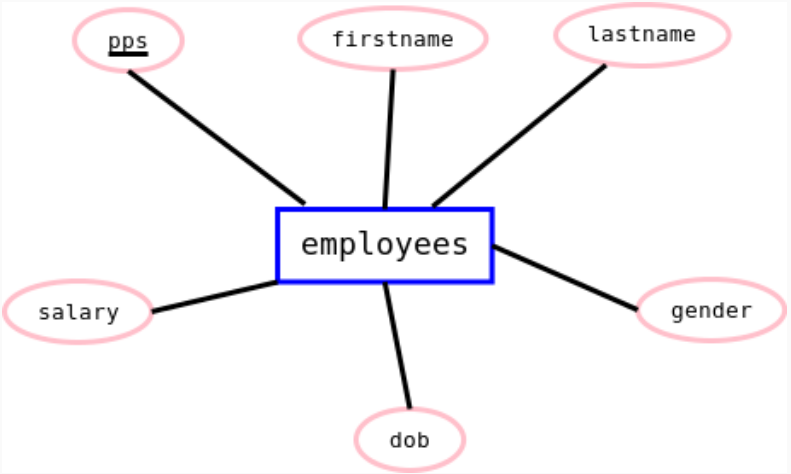
- Others:
  - `departments`

Entities in DB

- 

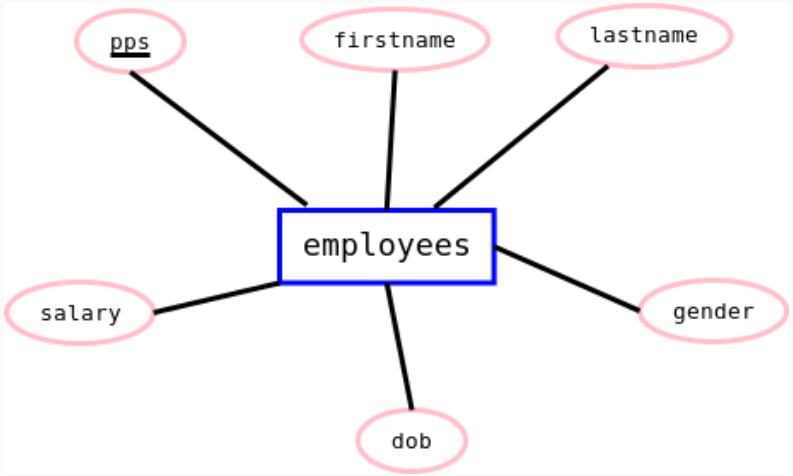
- Others:
  - `departments`
  - `projects`

Entities in DB

- 

- Others:
  - `departments`
  - `projects`
  - `locations`

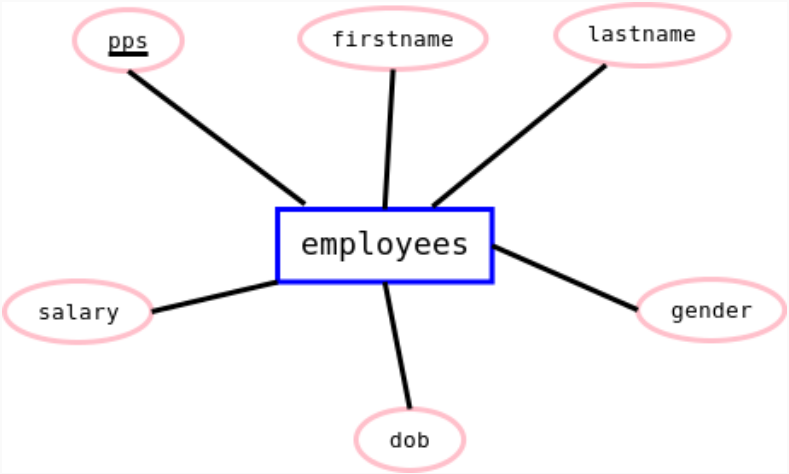
Entities in DB

- 

- Others:
  - `departments`
  - `projects`
  - `locations`
  - `dependents`

Keys in ER Diagrams

- A *key* is a set of attributes such that no two entities in the entity set can have exactly the same key values; every entity set must have key



- Note the the attributes that form the keys of the entity set are underlined

Relationships in DB

- 

  (Attributes omitted for brevity)
- Other relationships:

Relationships in DB

- 

  (Attributes omitted for brevity)
- Other relationships:
  - `manages` who manages which departments

Relationships in DB

- 

  (Attributes omitted for brevity)
- Other relationships:
  - `manages` who manages which departments
  - `worksin` who works in which department

Relationships in DB

- 

  (Attributes omitted for brevity)
- Other relationships:
  - `manages` who manages which departments
  - `worksin` who works in which department
  - `workson` who works on which project

Relationships in DB

- 

  (Attributes omitted for brevity)
- Other relationships:
  - `manages` who manages which departments
  - `worksin` who works in which department
  - `workson` who works on which project
  - `supervises` who supervises who

Relationships in DB

- 

  (Attributes omitted for brevity)
- Other relationships:
  - `manages` who manages which departments
  - `worksin` who works in which department
  - `workson` who works on which project
  - `supervises` who supervises who
  - `haslocation` which departments have which locations

Relationships in DB

- 

  (Attributes omitted for brevity)
- Other relationships:
  - `manages` who manages which departments
  - `worksin` who works in which department
  - `workson` who works on which project
  - `supervises` who supervises who
  - `haslocation` which departments have which locations
  - `locatedat` which projects are located where

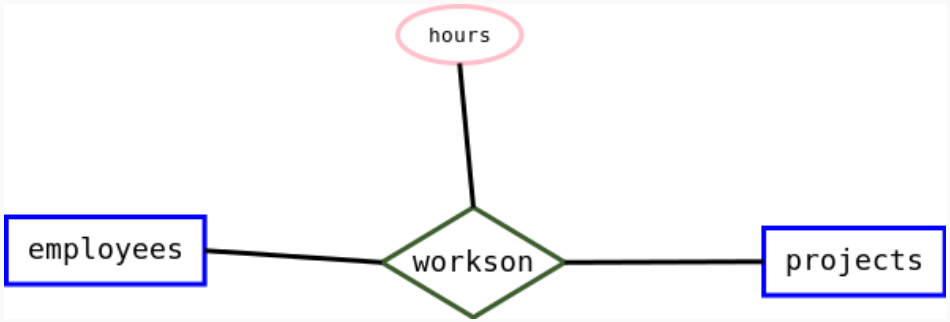
Relationships in DB

- 

  (Attributes omitted for brevity)
- Other relationships:
  - `manages` who manages which departments
  - `worksin` who works in which department
  - `workson` who works on which project
  - `supervises` who supervises who
  - `haslocation` which departments have which locations
  - `locatedat` which projects are located where
  - `isdependentof` which individuals are dependents of which employees

Relationships With Attributes

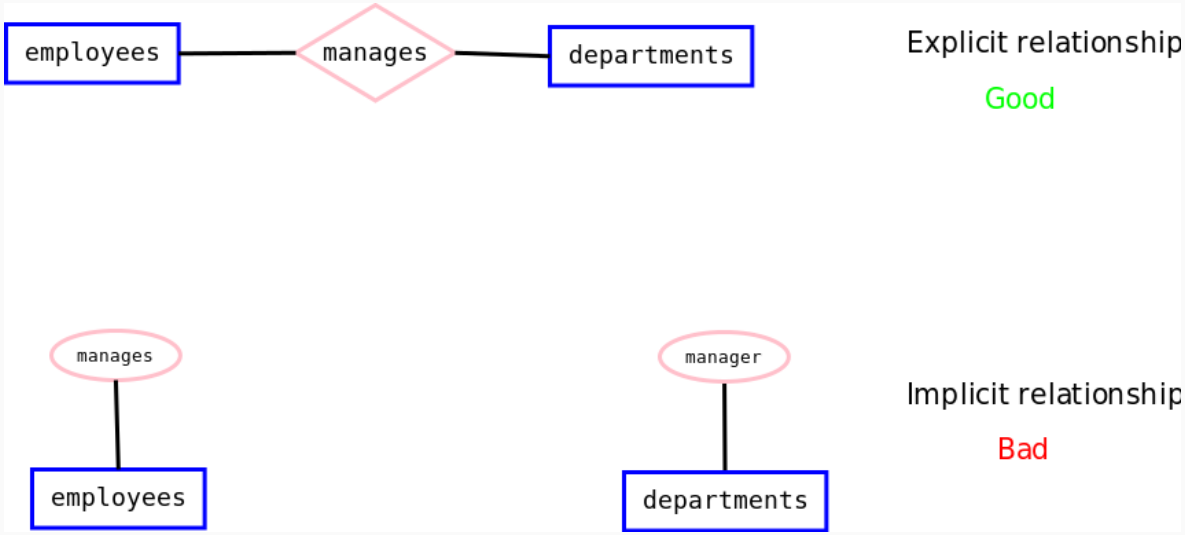
- Often cleaner to associate attributes with relationships rather than entity sets



- Note: employee works different number of hours on each project

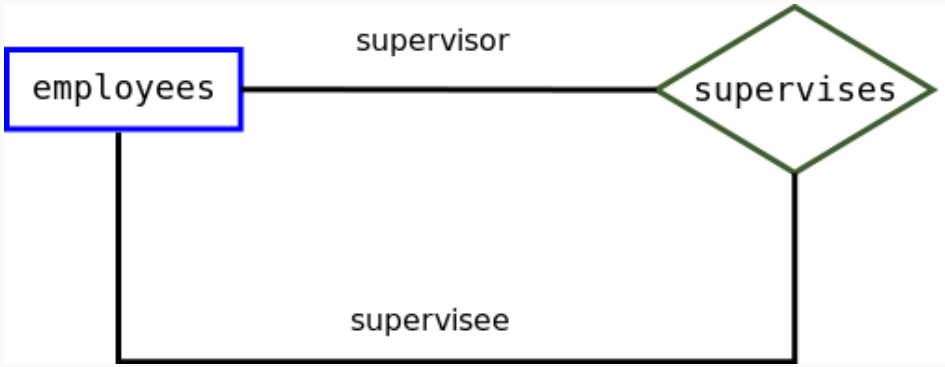
Implicit vs Explicit Relationships

- Could "encode" relationships implicitly using attribute values



- This is generally a bad idea– redundant and error-prone

Reflexive Relationships



- Elements of relationship sets are *pairs* of employees
- Each has different *role* in relationship (supervisor, supervisee)

General Design Principles

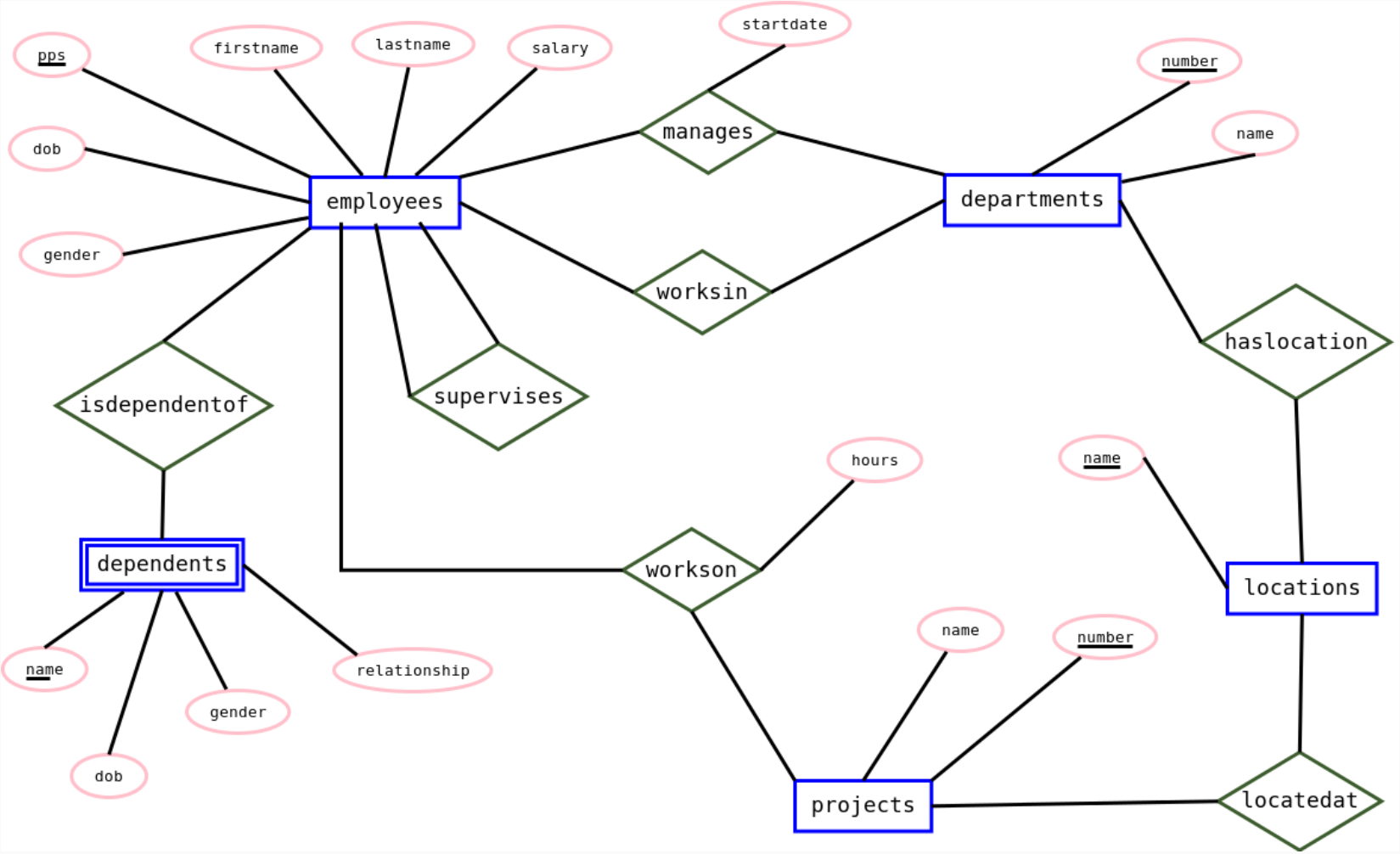
Faithfulness should accurately reflect pertinent aspects of real-world problem domain the DB data is intended to represent

Simplicity should be as simple as possible– but no simpler

Avoid Redundancy

- Each piece of information should be represented only once
- Duplicate information is wasteful of space and encourages errors and inconsistencies when info. is added, modified, or removed

ER Diagram


pps
firstname
lastname
salary
startdate
number
dob
manages
name
employees
departments
gender
worksin
haslocation
isdependentof
supervises
hours
name
dependents
workson
locations
name
relationship
gender
name
number
dob
projects
locatedat

Generating DB Schema From ER Diagram

Basic Idea

- Each entity set is represented by a table
- Each relationship is also represented by a table

Caution

- Further refinements needed to weed out poor designs

Notes and Acknowledgements

The company database design is adapted from Elmasri and Navathe, “Fundamentals of Database Systems”. Addison-Wesley.