L16: More On ER Modelling

CS1106/CS6503: Intro to Relational Databases

Dr Kieran T. Herley

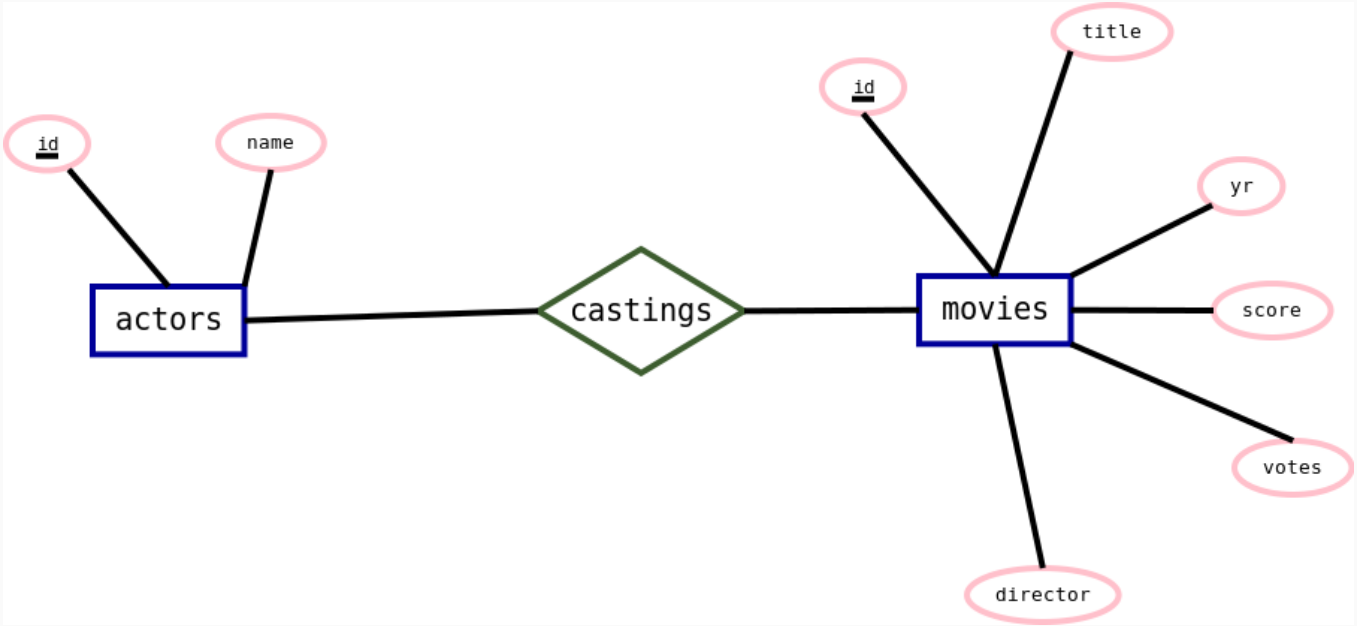
Semester One, 2023-24

School of Computer Science & Information Technology
University College Cork

Summary

Development of ER to model a simple company database.

ER Diagrams refresher



Identify and model *entities* in miniworld and *relationships* among them.

DB Design Case Study

DB1106 database-design consultants

- DB1106 has been hired to design a DB for a company *Acme Widgets Corporation*.
- DB to keep track of employees, departments and projects.
- Following exhaustive requirements collection and analysis, we have distilled needs down to "miniworld" description overleaf.

Example taken from Esmasri and Navathe

Miniworld for Acme

1. The company is organized into departments. Each department has a unique name, a unique number and a particular employee who managed the department. We keep track of the start date when that employee started managing the department. A department may have several locations.
2. A department controls a number of projects, each has a unique name, a unique number and a single location.
3. We store each employee's name, PPS number, address, salary, sex and birth date. An employee is assigned to one department but may work on several projects, which may be controlled by other departments. WE keep track of the number of hours per week that an employee works on each project. We also keep track of the direct supervisor of each employee.
4. We want to keep track of the dependents of each employee for insurance purposes. We keep track of each dependent's first name, sex, data of birth and relationship to employee.

Notes

- **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 (among other things):

- Details of each employee
- Ditto departments, projects etc.

DB1106 Consultants cont'd

Need to keep track of (among other things):

- Details of each employee
- Ditto departments, projects etc.

but also

- 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

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.

Finding entities

1. The company is organized into departments. Each department has a unique name, a unique number and a particular employee who manages the department. We keep track of the start date when that employee started managing the department. A department may have several locations.
2. A department controls a number of projects, each has a unique name, a unique number and a single location.
3. We store each employee's name, PPS number, address, salary, sex and birth date. An employee is assigned to one department but may work on several projects, which may be controlled by other departments. WE keep track of the number of hours per week that an employee works on each project. We also keep track of the direct supervisor of each employee.
4. We want to keep track of the dependents of each employee for insurance purposes. We keep track of each dependent's first name, sex, data of birth and relationship to employee.

departments

locations

projects

employees

dependents

Entities and associated info.

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

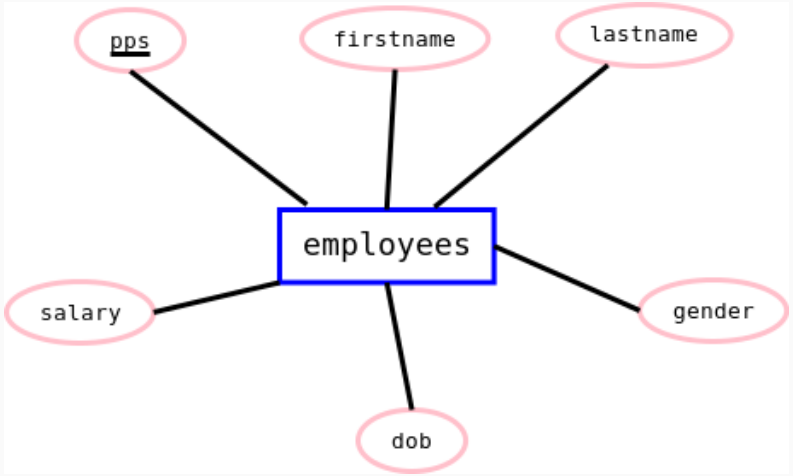
Departments name, number (unique); Also employee who manages, department (multiple) locations

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

Locations name

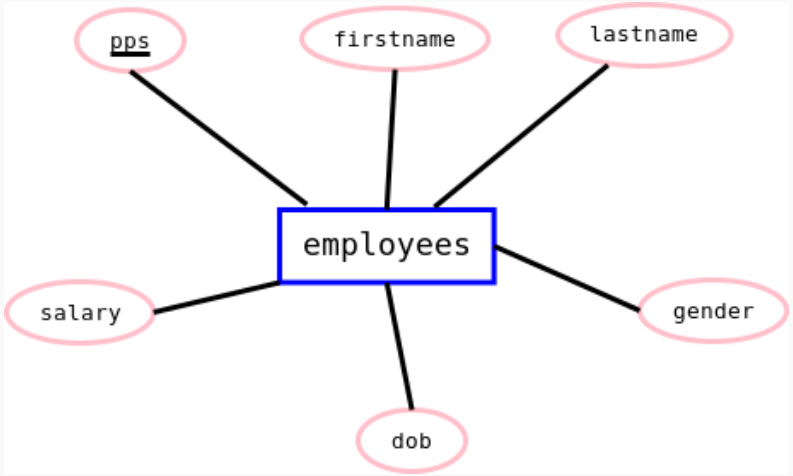
Dependents name, gender, data of birth, relationship to employee; Also employee

Entities in DB

- 

- Others:

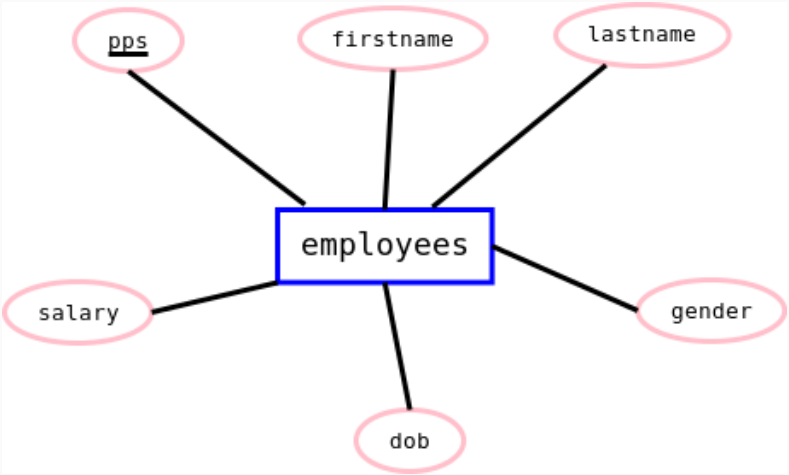
Entities in DB

- 

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

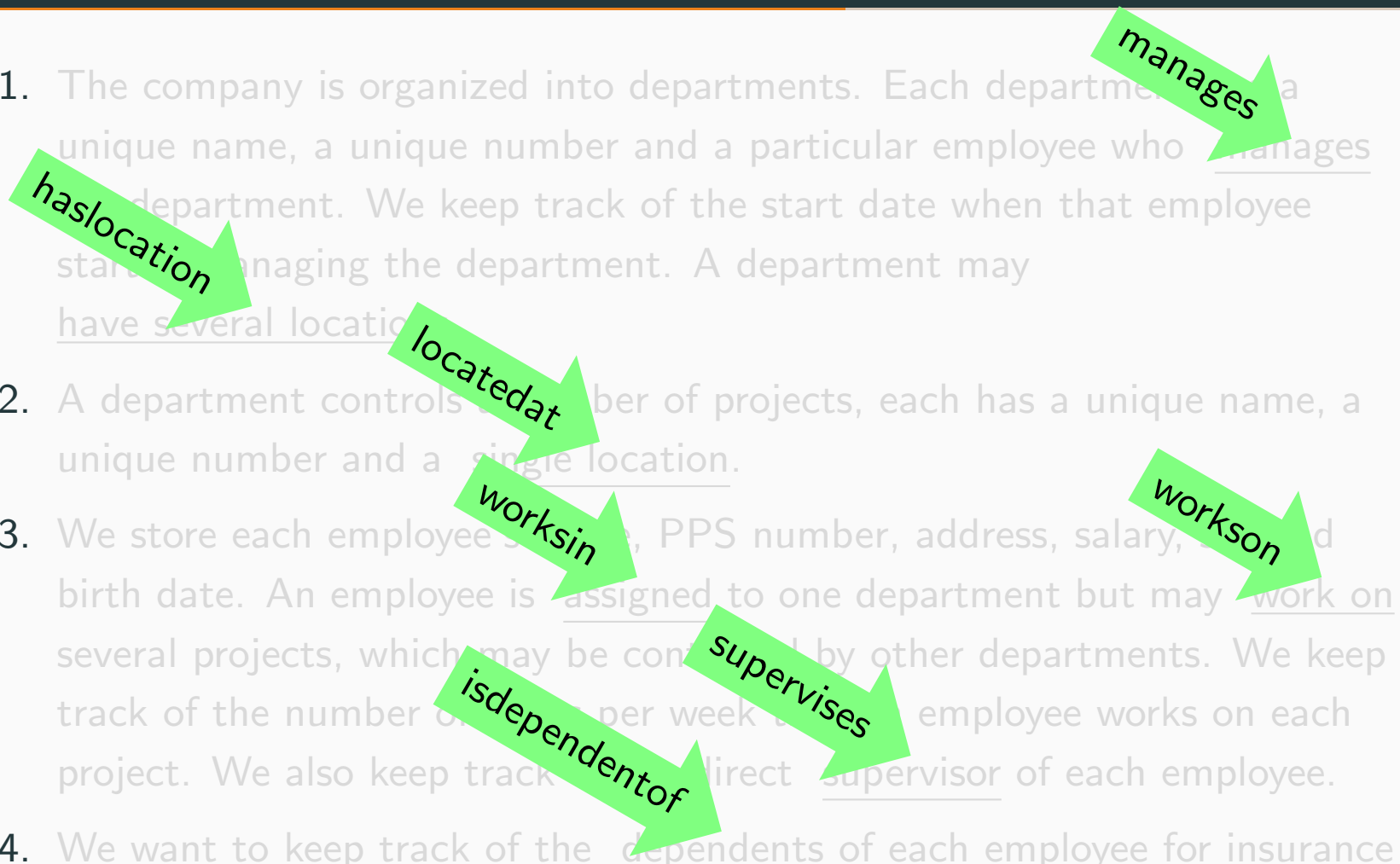
Keys in ER Diagrams

- 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.

Finding relationships



1. The company is organized into departments. Each department has a unique name, a unique number and a particular employee who manages the department. We keep track of the start date when that employee started managing the department. A department may have several locations.
2. A department controls a number of projects, each has a unique name, a unique number and a single location.
3. We store each employee's name, PPS number, address, salary, sex and birth date. An employee is assigned to one department but may work on several projects, which may be controlled by other departments. We keep track of the number of hours per week that an employee works on each project. We also keep track of the direct supervisor of each employee.
4. We want to keep track of the dependents of each employee for insurance purposes. We keep track of each dependent's first name, sex, data of birth and relationship to employee.

Relationships in DB

- 

  (Attributes omitted for brevity)
- Other relationships:

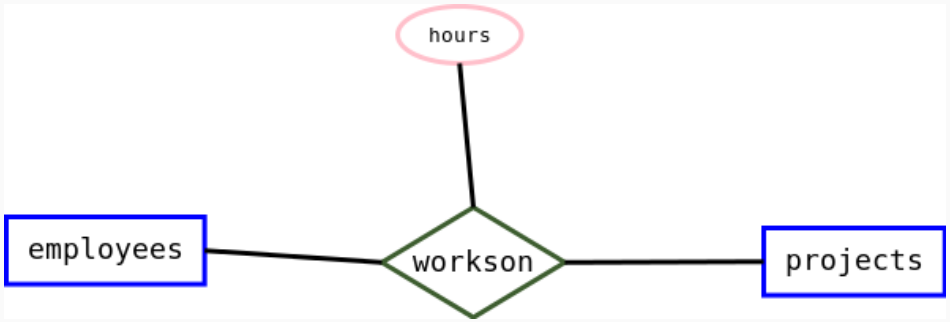
Relationships in DB

- 

  (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

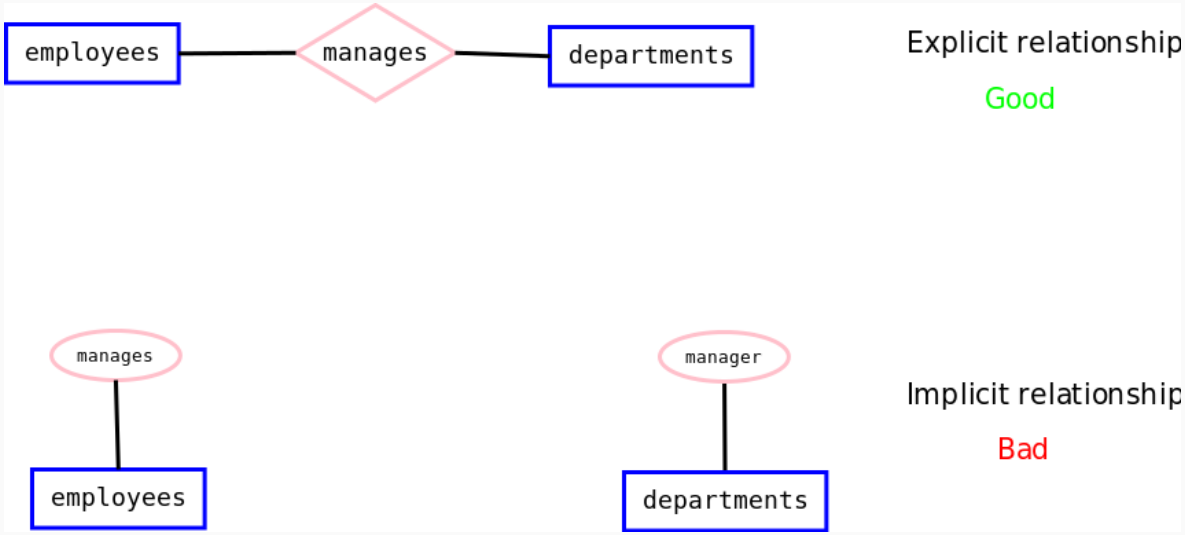
- Often cleaner to associate attributes with relationships rather than entity sets



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

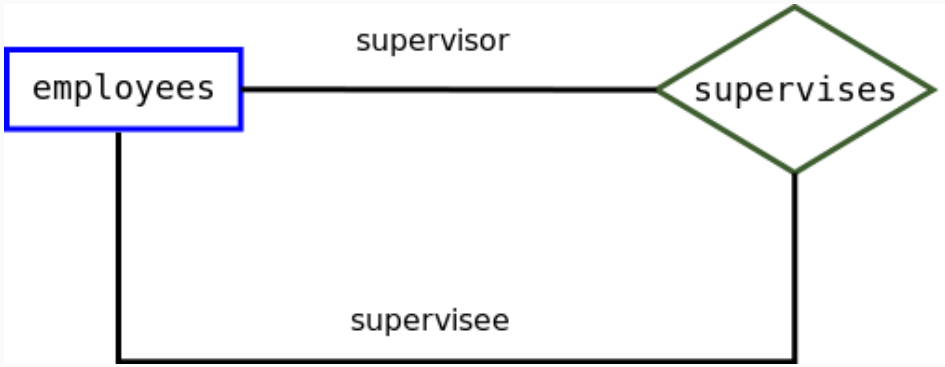
Implicit vs Explicit Relationships

- 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

General Design Principles

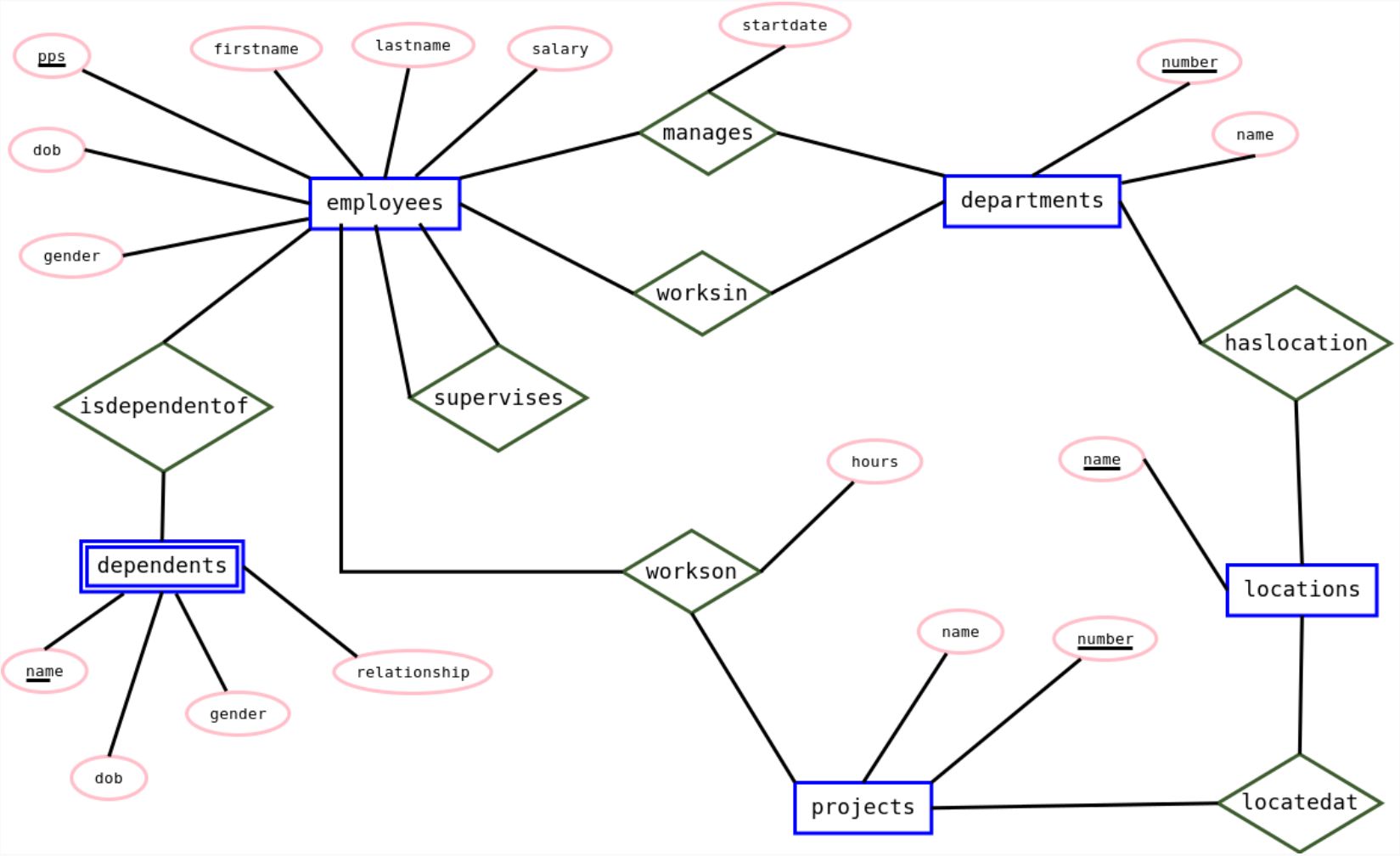
Faithfulness should accurately reflect pertinent aspects of real-world problem domain the DB data is intended to represent

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
dob
name
number
projects
locatedat

From ER to DB Schema

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.