Data Modelling

Dr Na Yao

Objectives

- Understand basic concepts of Relational model
- Understand basic concepts associated with Entity-Relationship(ER) model.
- Be able to use Entity–Relationship (ER) modelling in database design.
- Be able to build an ER model from a requirements specification.

Data model

- In introduction we know that **Database** is a shared collection of logically related data (and a description of this data), designed to meet the information needs of an organization.
- How do we build a Database?
- A technique called **data modelling** helps you to understand the structure and meaning of data.
- A **data model** is a graphical description of the components of database.

Recap of Relational model

- E. F. Codd proposed relational data model in 1970.
- In the relational model, all data is logically structured within relations (tables).
- A **relation**, is a two-dimensional table arranged in columns and rows.
- A **relational database** is a collection of relations.
- One row of a table stores details of one case (instance) of an item. All the rows in a table store data about the same type of items.
- One column in the table contains the same type of data.

Recap of Relational model

- In a relational database, each row must be uniquely identified with **primary key**.

studentID	firstName	surname	age	programme
001	Mary	White	20	IoT
004	Tom	Hardy	19	Telecom
006	Mary	Bennet	20	E-commerce
032	John	Doe	21	Telecom
101	Ann	Martins	19	IoT

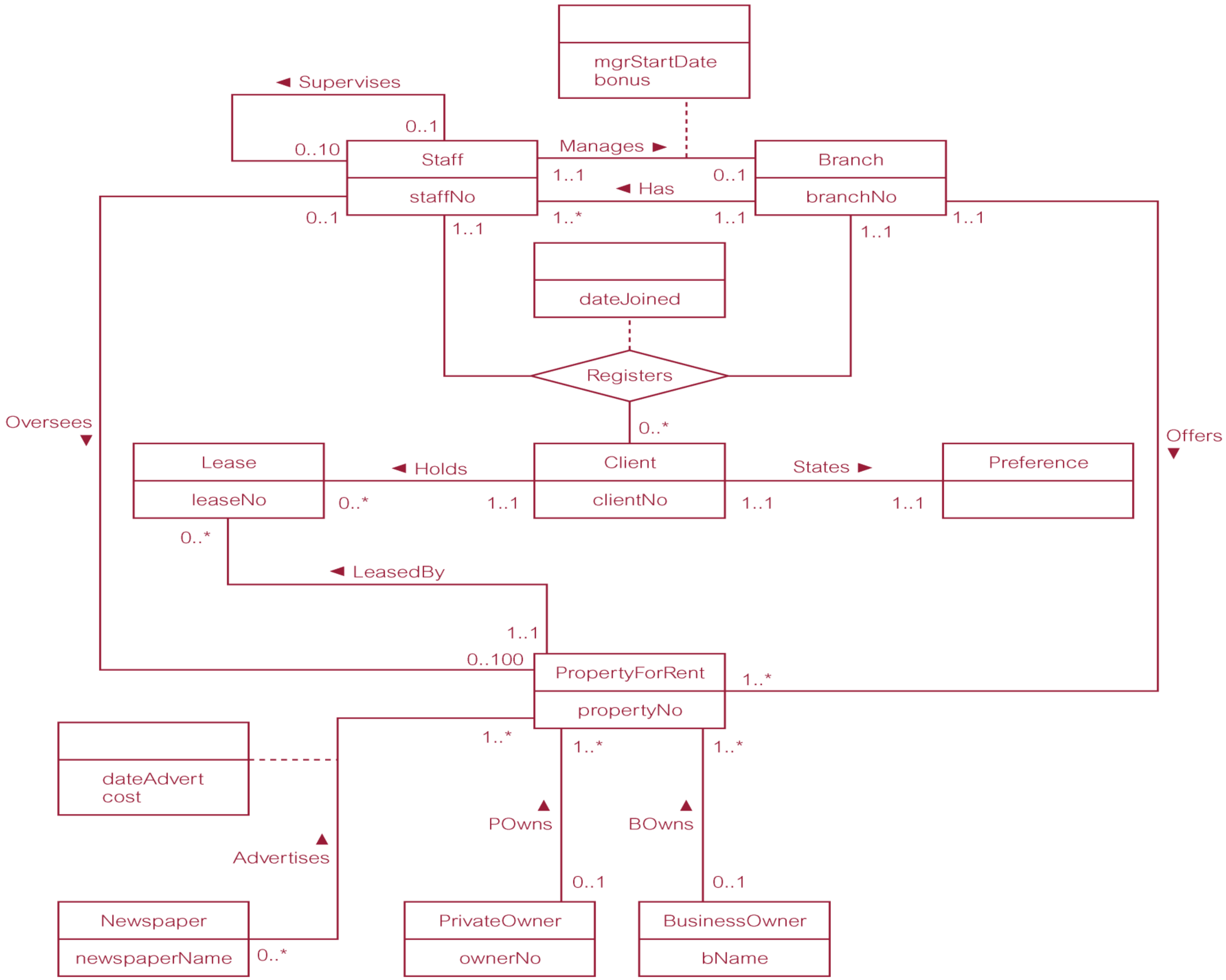
- The tables in a relational database are **connected** or **related** by means of the data in the tables.

Entity-relationship (E-R) modelling

- E-R modelling is a high-level *conceptual modelling technique for DB applications*
- Developed by Peter Chen and published in 1976 *
- Main concepts (building blocks)
  - Entity
  - Attributes
  - Relationship

* Chen, Peter (March 1976). "The Entity-Relationship Model - Toward a Unified View of Data". *ACM Transactions on Database Systems* 1 (1): 9–36.

ER diagram of Branch user view of *DreamHome*

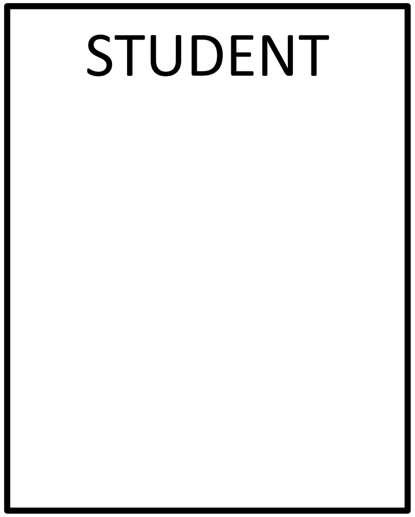


Entity

- Basic building block of a data model.
- An entity is a “thing” about which data should be stored.
- Group of objects with same properties, identified by enterprise as having an independent existence.
- Can be objects with a physical or conceptual existence
  - Physical existence:
  - Conceptual existence:

Entity

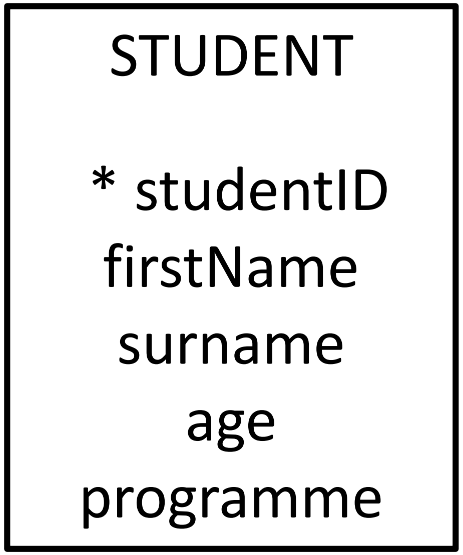
An entity is represented by a *rectangle*. The name of the entity is shown in singular form in uppercase in the top part of the rectangle.



STUDENT

Attributes

- An entity has characteristics or **attributes**.
- An attribute is a discrete element of data; it describes an entity.
- Attributes are shown below the entity's name.
- Attribute names must be carefully selected so that they are self-explanatory and unique.
- A identifier (primary key) uniquely identifies an instance of an entity. Attribute(s) that are identifiers are labelled in the entity.



STUDENT

* studentID
firstName
surname
age
programme

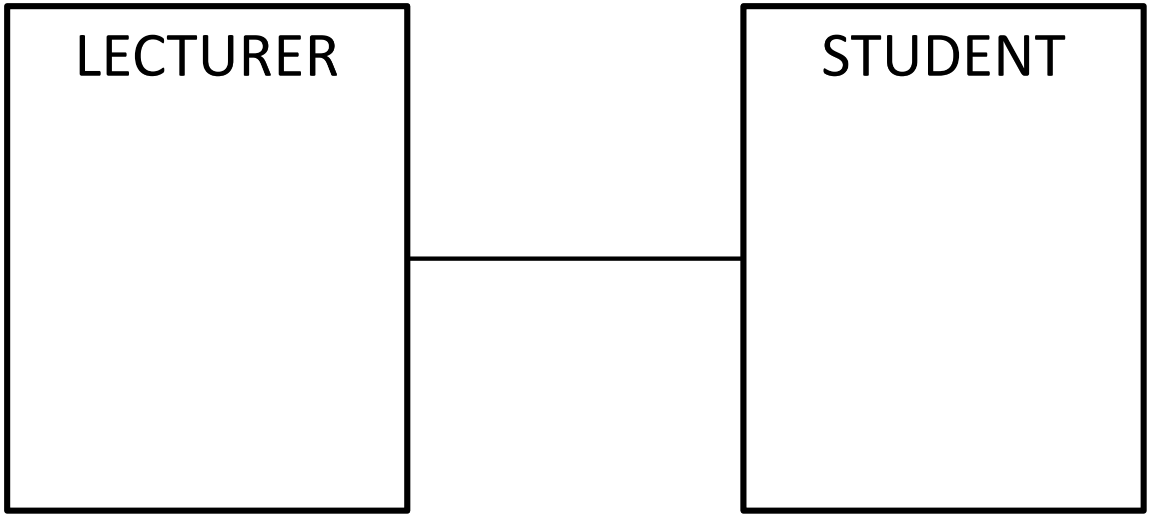
Exercise

Read the following description:

- An second hand bookshop want to keep a record of all the books in stock.
- Each book has a title, ISBN number, year of publication, price, condition of the book (like new, good condition, worn etc.)
- Design a single entity database using ER diagram. *Clearly* ***label everything***.

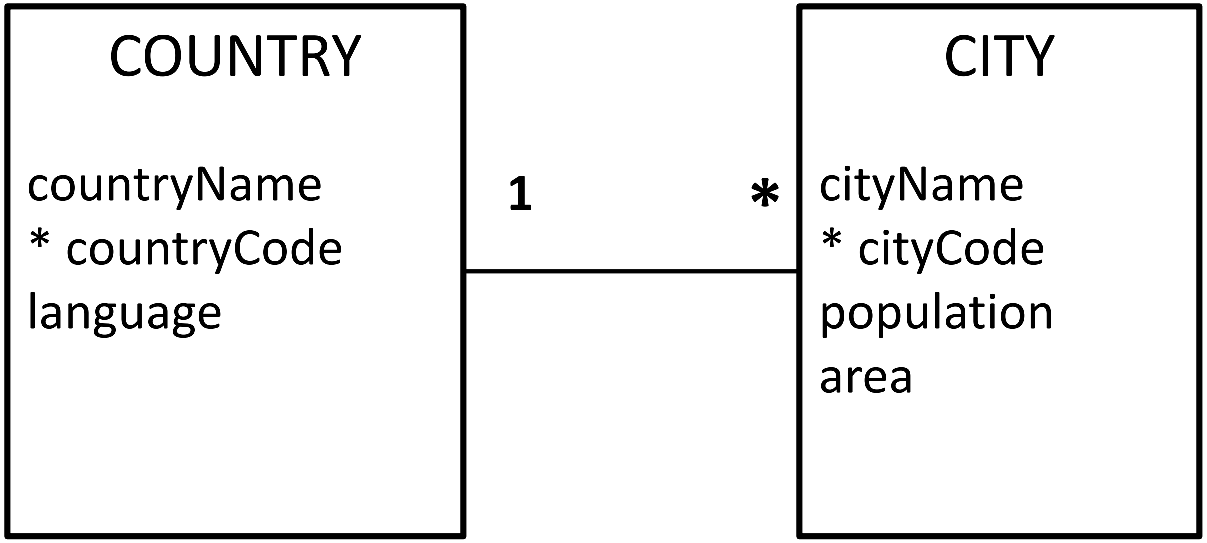
Relationship

- Entities are related to other entities.
- Relationship describes a linkage between two entities and is represented by an ***arc*** between them.



One-to-many (1:m) relationship

- Consider the database to record countries and cities.



COUNTRY

countryName
* countryCode
language

1

*

CITY

cityName
* cityCode
population
area

- This can be read as: “a country can have many cities, but a city belongs to only one country.”

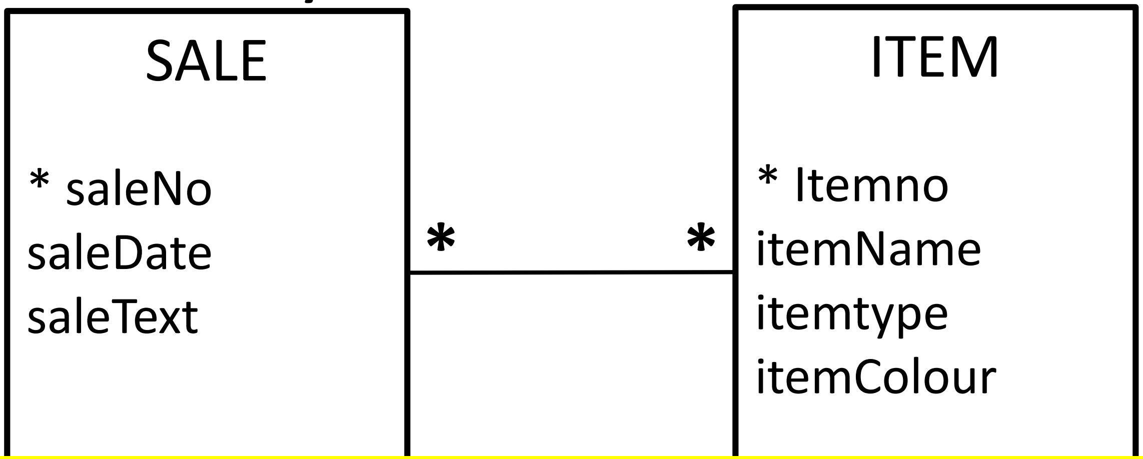
Exercise

- The second hand bookshop now realised a problem in their earlier single entity database – one book may have *many* copies and each copy has its own condition, e.g. Harry Potter and the Philosopher’s Stone has 3 copies, one nearly new condition and two worn condition; the nearly new condition copy has a higher price than the worn ones.

  Redesign your database with 1:m relationship.

Many-to-Many (m:m) relationship

- Consider the case when items are sold. We can identify two entities: SALE and ITEM. A sale can contain many items, and an item can appear in many sales.



SALE

* saleNo
saleDate
saleText

*

*

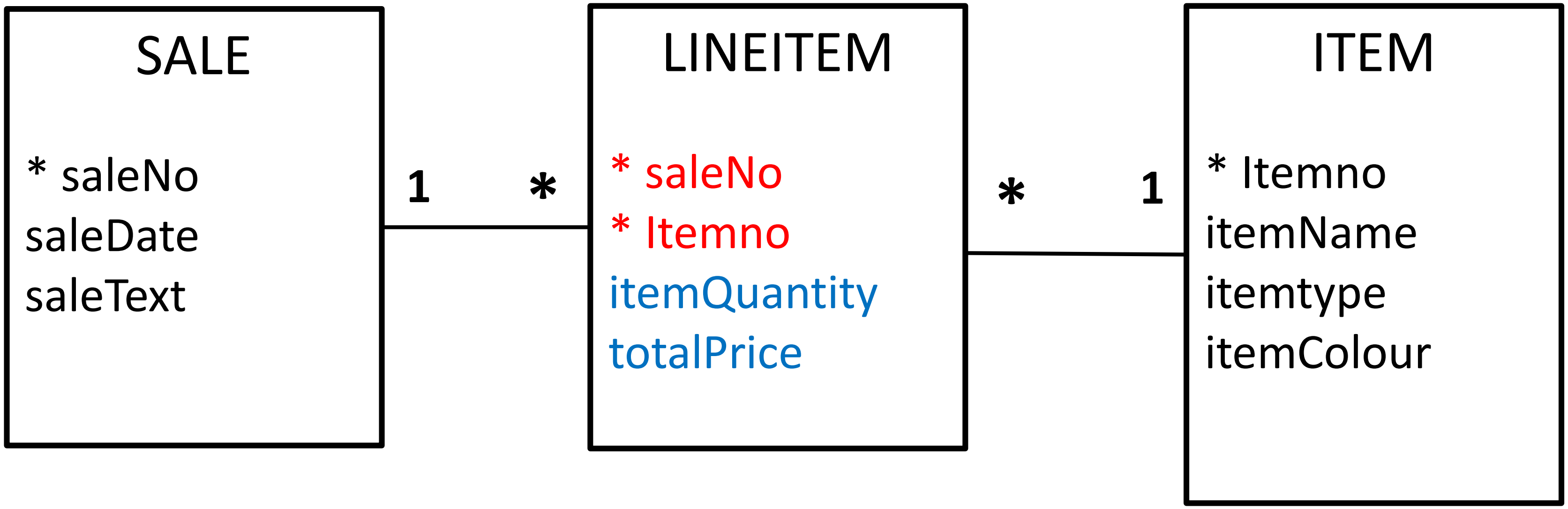
ITEM

* Itemno
itemName
itemtype
itemColour

But how do we record the information for the *m:m* relationship?

Many-to-Many (m:m) relationship

- Information missing from the relationship includes: quantity of an item being sold, total price etc.
- To store these information (attributes related to the m:m relationship), we create a third entity (**associative entity**)to link the entities through two 1:m relationships.



SALE

* saleNo
saleDate
saleText

1 *

LINEITEM

* saleNo
* Itemno
itemQuantity
totalPrice

* 1

ITEM

* Itemno
itemName
itemtype
itemColour

Relational keys

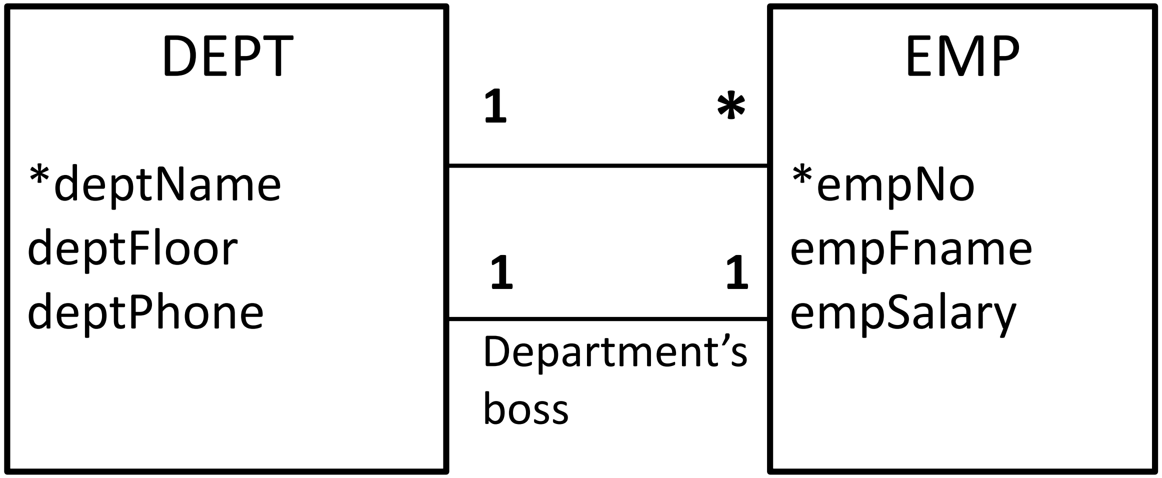
- **Candidate Key**
  - A set of attributes that uniquely identifies a tuple within a relation.
  - Uniqueness : In each tuple, candidate key uniquely identify that tuple.
  - Irreducibility: No proper subset of the candidate key has the uniqueness property.
- **Primary Key**
  - Candidate key selected to identify tuples uniquely within relation.
- **Foreign Key**
  - Attribute, or set of attributes, within one relation that matches candidate key of some (possibly same) relation.
- **Composite Key**
  - A candidate key that consists of two or more attributes.

Exercise

- The Marathoner, a monthly magazine, regularly reports the performance of professional marathon runners. It has asked you to design a database to record the details of all major marathons (e.g., London, Beijing and Paris).
- Professional marathon runners compete in several races each year. A race may have thousands of competitors, but only about 200 or so are professional runners, the ones The Marathoner tracks.
- For each race, the magazine reports a runner's time and finishing position and some personal details such as name, gender, and age.

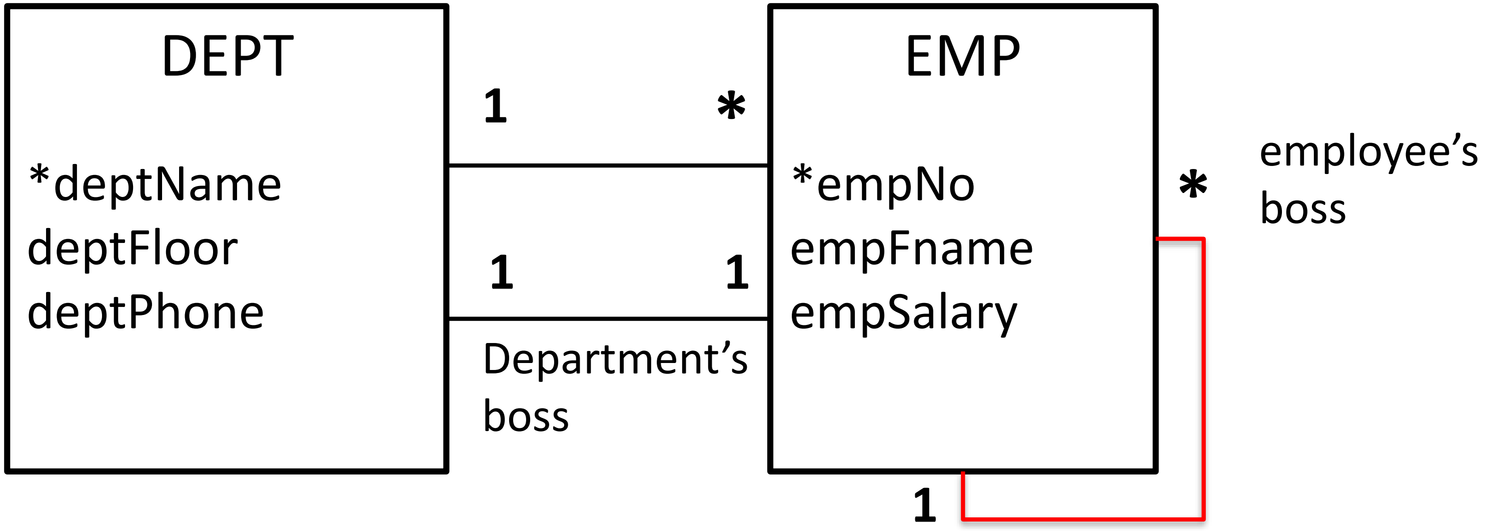
One-to-one relationship

- A department has one or more employees, and an employee belongs to one department.
- A department has one boss, and a person is boss of only one department.
- Boss is a 1:1 relationship between DEPT and EMP.



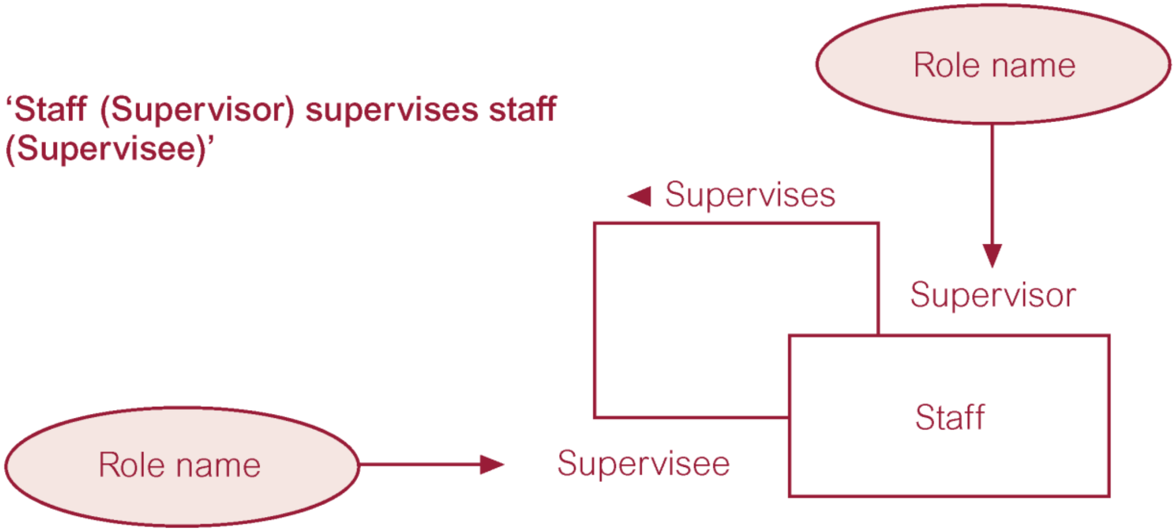
Recursive Relationships

- There is more to boss than just a department.
- People also have a boss. An employee can be boss to many other employees, and an employee has normally just one boss.
- The person-boss relationship is a recursive 1:m relationship.



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.



‘Staff (Supervisor) supervises staff (Supervisee)’

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

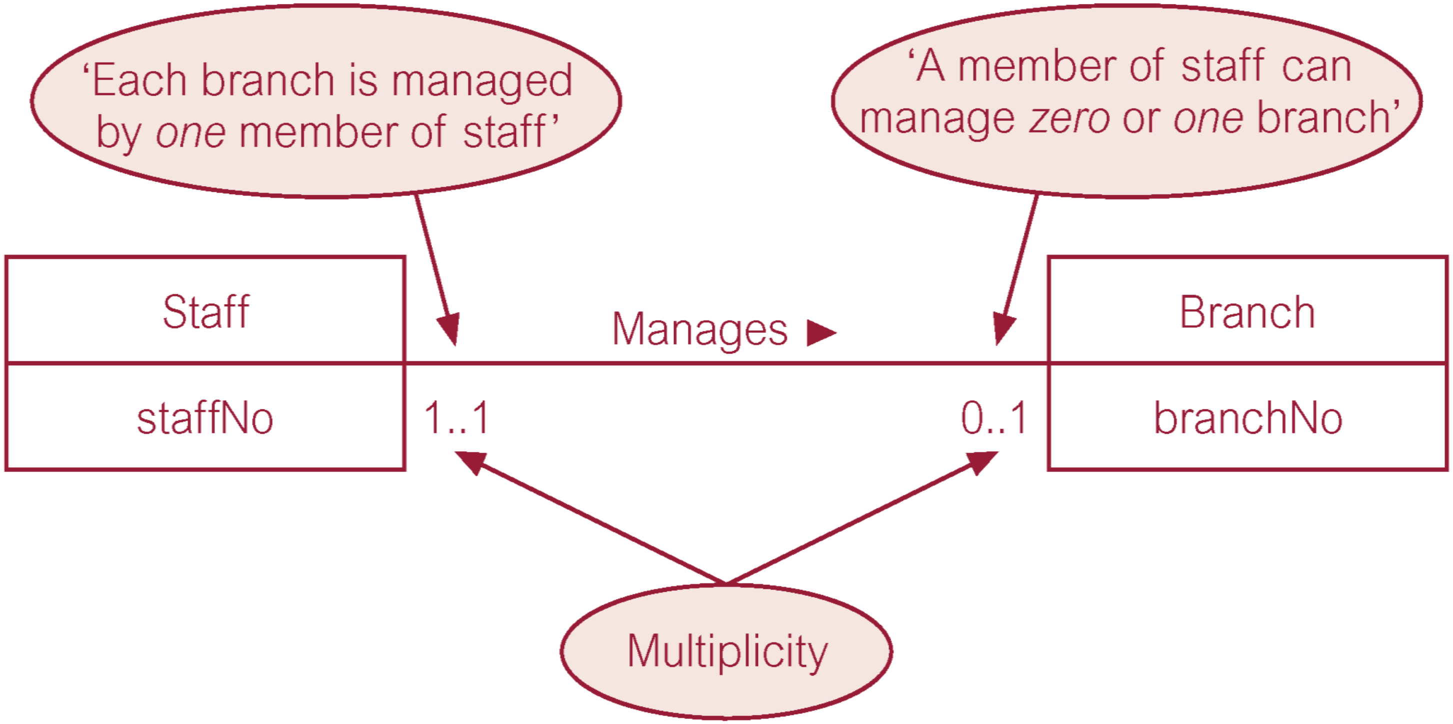
- The most common degree for relationships is binary.

- Binary relationships are generally referred to as being:
  - one-to-one (1:1)
  - one-to-many (1:*)
  - many-to-many (*:*)

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.

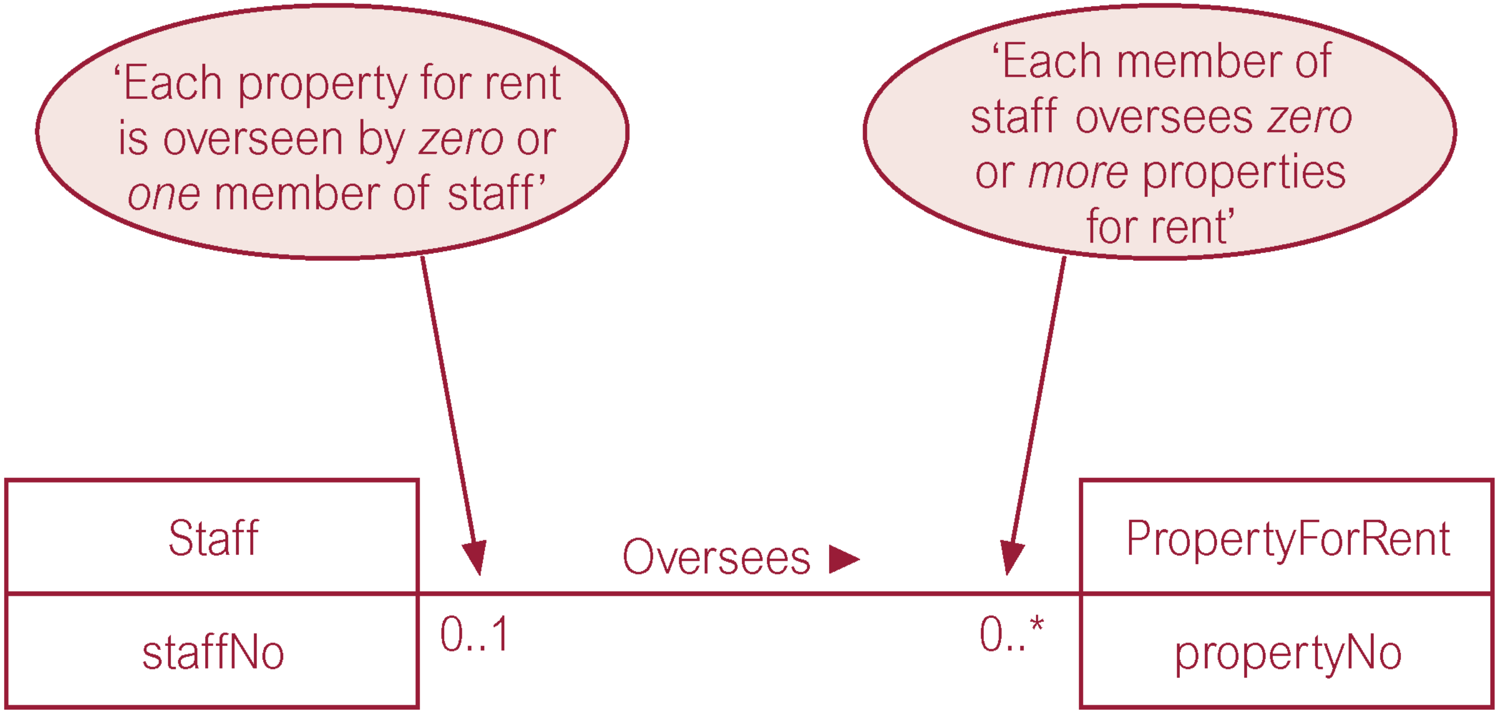
Multiplicity of Staff *Manages* Branch (1:1) relationship



‘Each branch is managed by *one* member of staff’

‘A member of staff can manage *zero* or *one* branch’

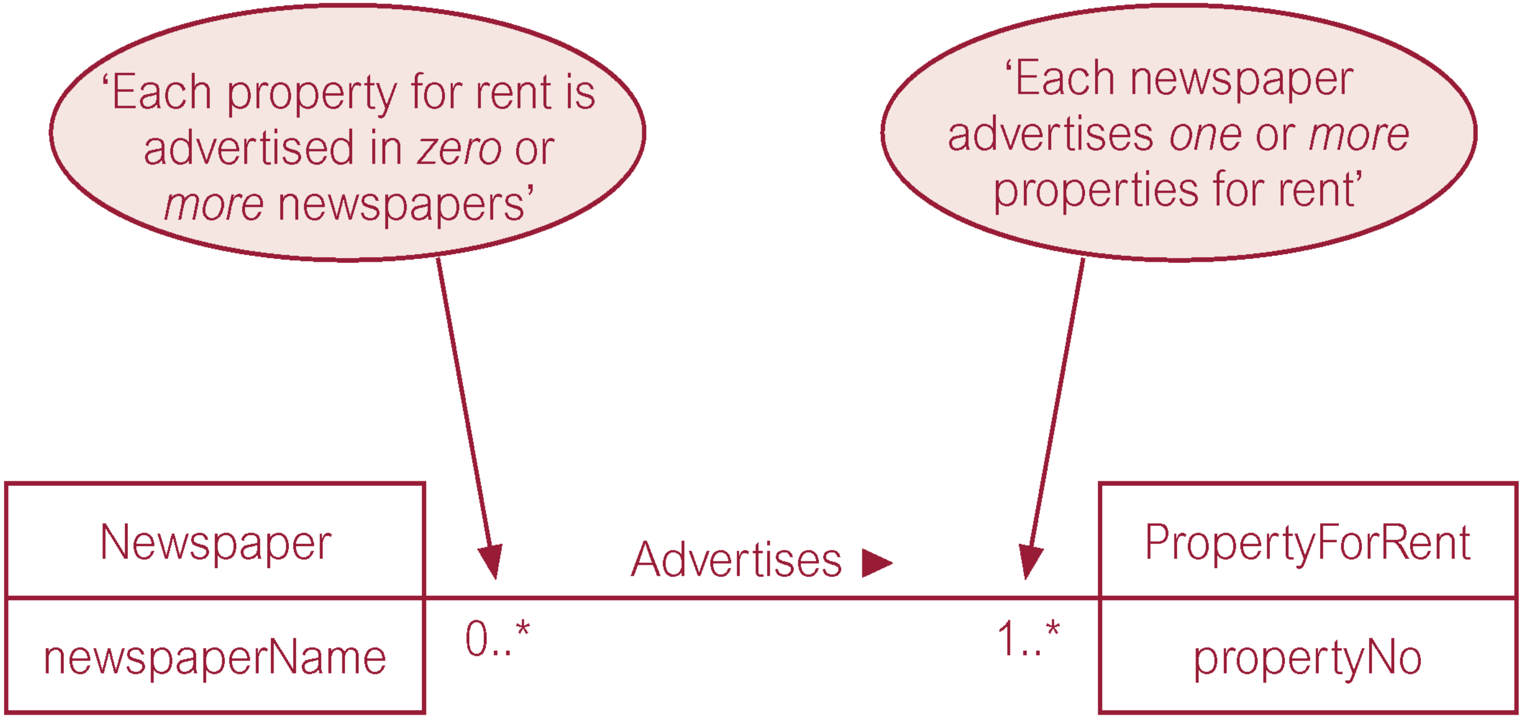
Multiplicity of Staff *Oversees* PropertyForRent (1:*) relationship type



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

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

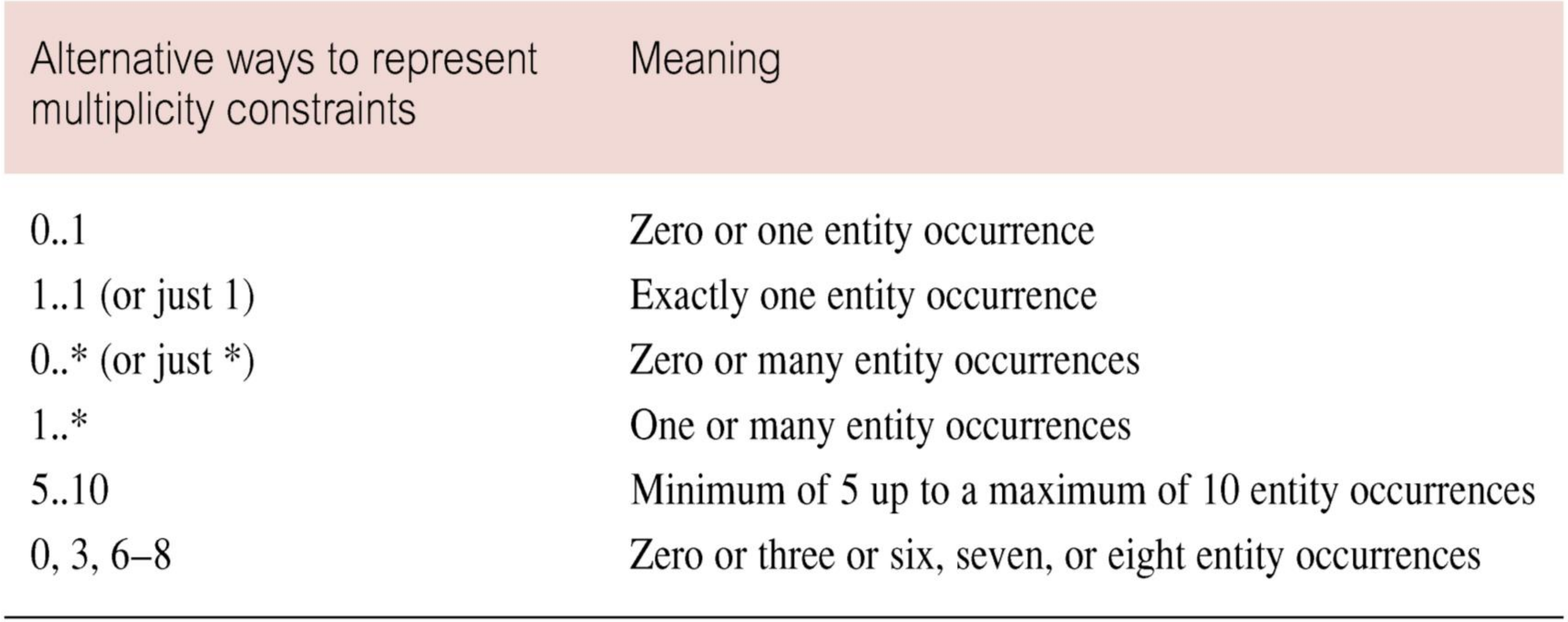
Multiplicity of Newspaper *Advertises* PropertyForRent (*:*) relationship



‘Each property for rent is advertised in *zero* or *more* newspapers’

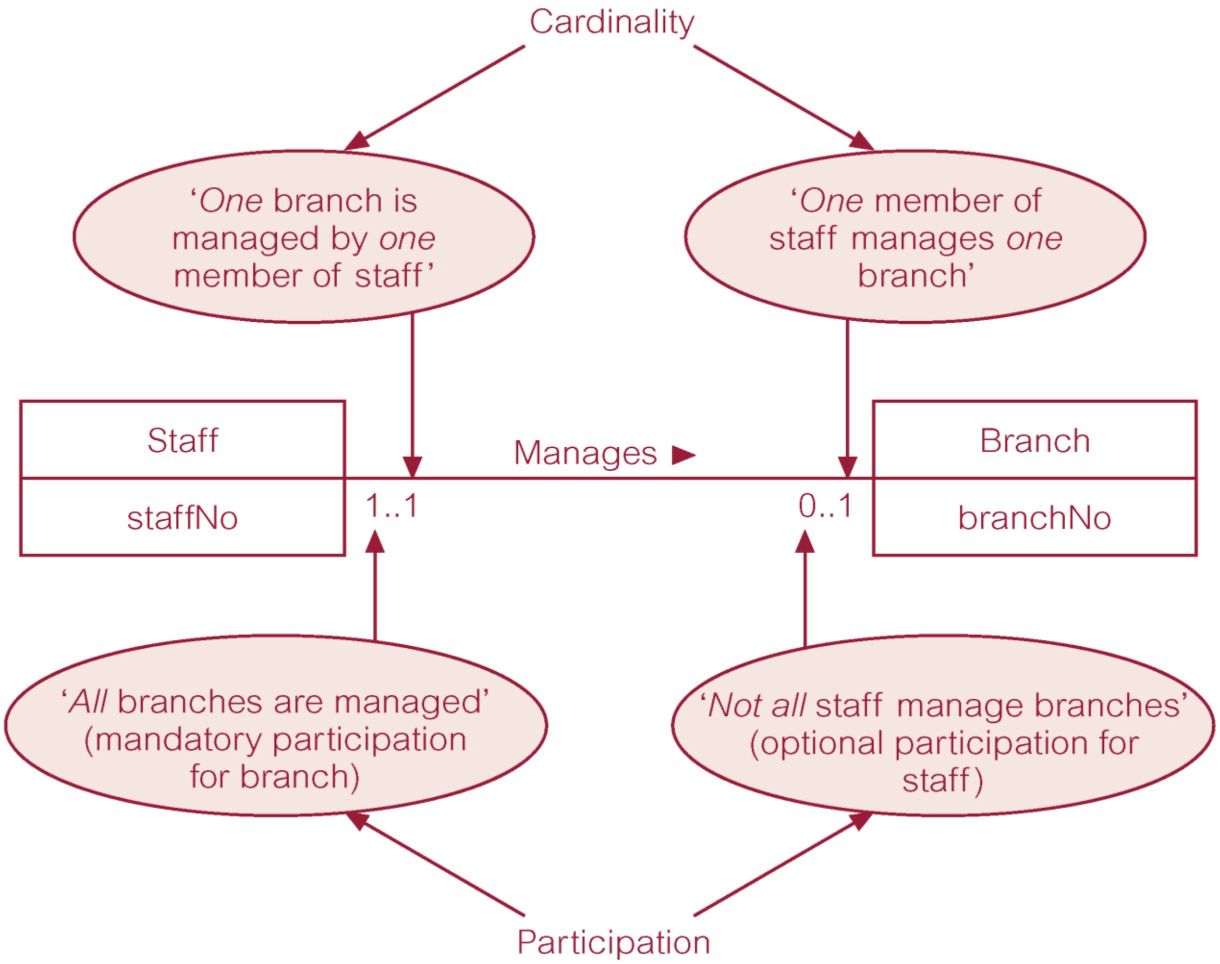
‘Each newspaper advertises *one* or *more* properties for rent’

Summary of 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 as cardinality and participation constraints



Cardinality

'*One* branch is managed by *one* member of staff'

'*One* member of staff manages *one* branch'

'*All* branches are managed' (mandatory participation for branch)

'*Not all* staff manage branches' (optional participation for staff)

Participation

Relationship Types

- Degree of a Relationship
  - Number of participating entities in a relationship.

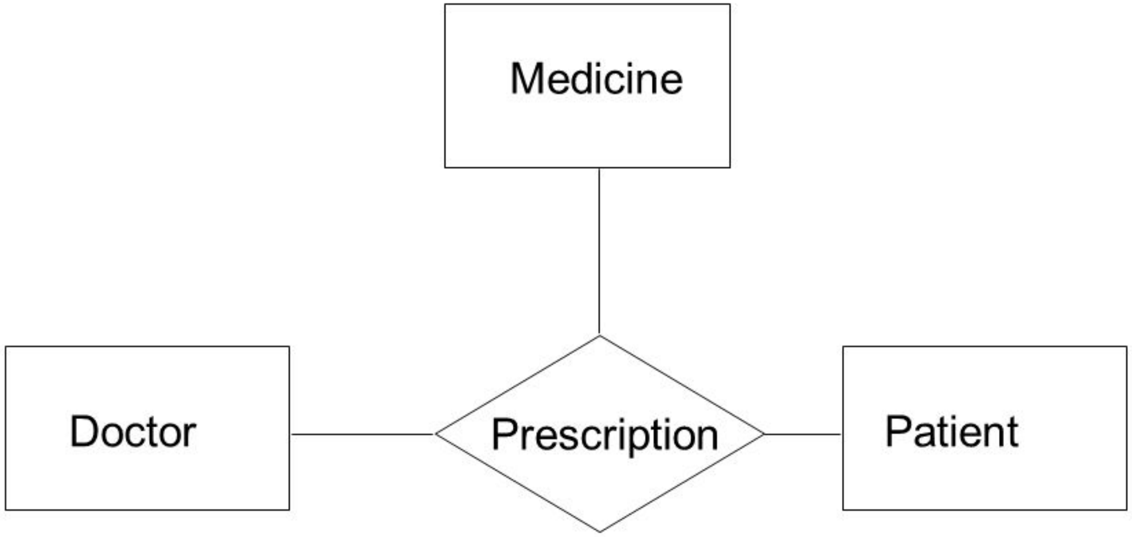
- Relationship of degree :
  - two is binary (what we have seen so far are all binary relationships)
  - three is ternary
  - four is quaternary

Ternary relationship 1

- In a ternary relationship, three entities are ***simultaneously*** involved.

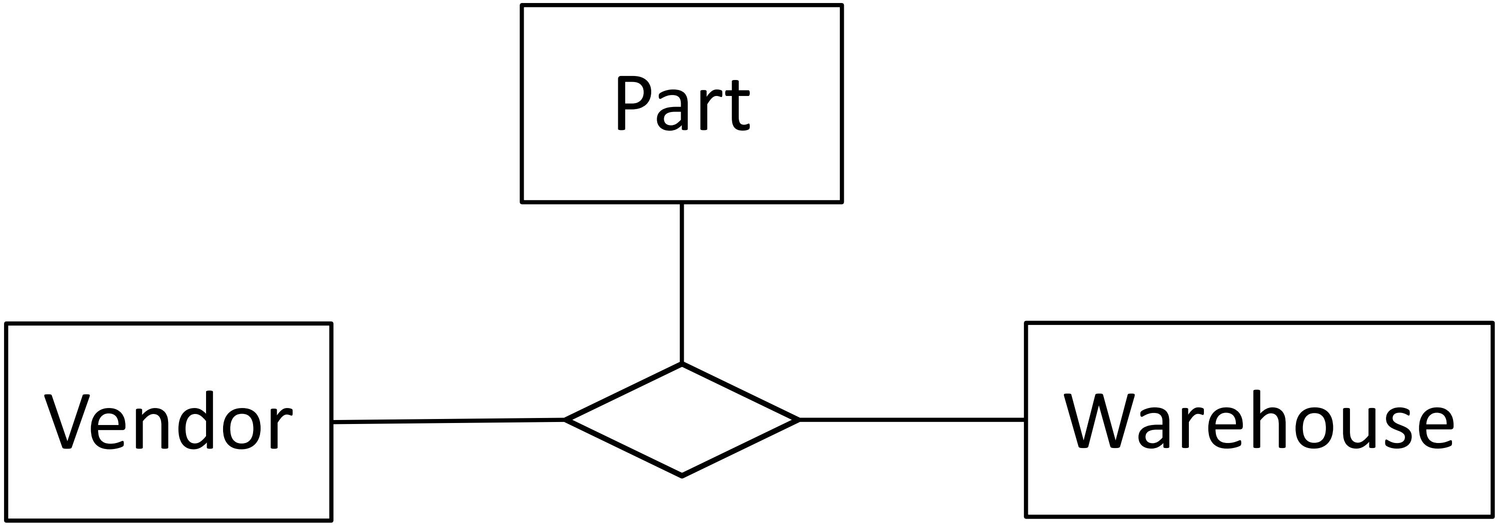
Ternary relationship of

“Doctor prescribes patients medicines”



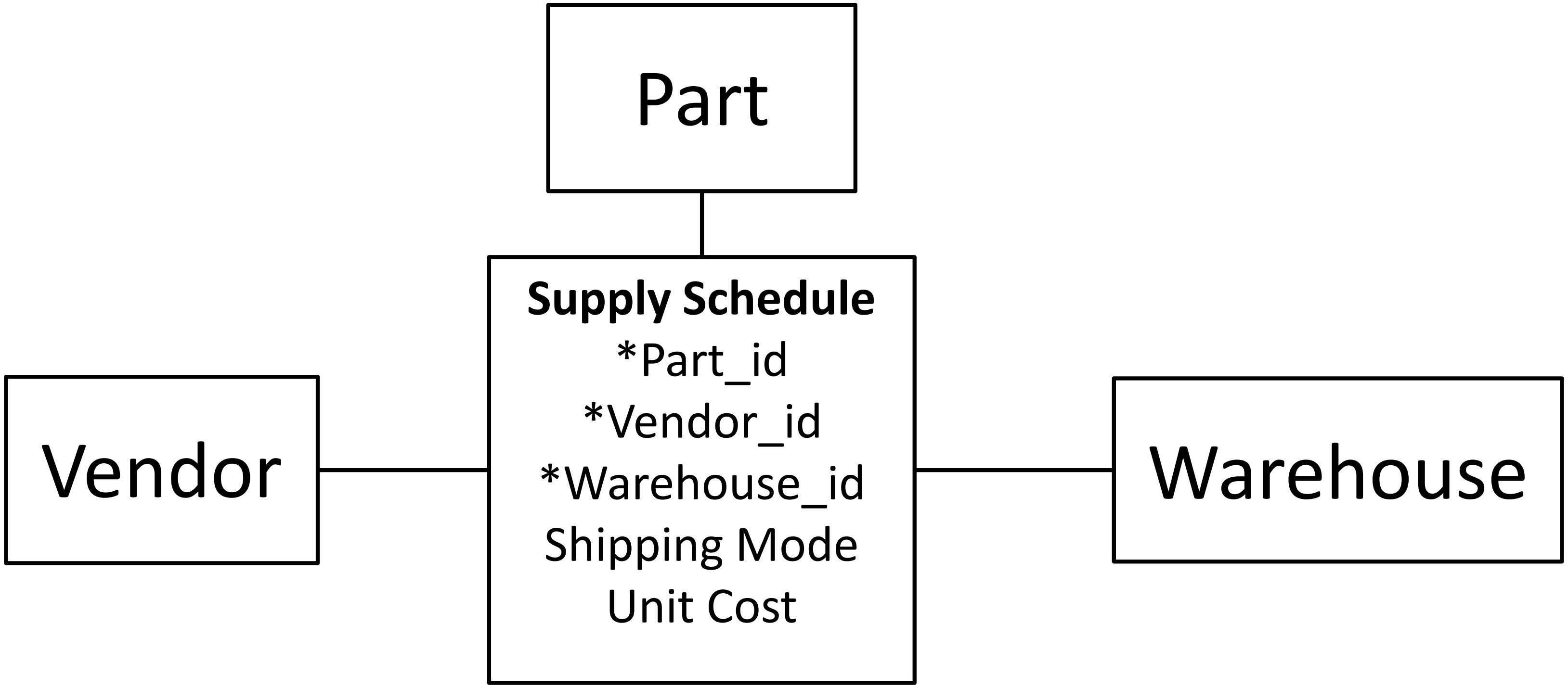
Ternary relationship 2

- Vendors can supply various parts to warehouses.
- Three entity types: Vendor, Part and Warehouse



Ternary Relationship 3

- Attributes associated with the "*Supply Schedule*" ternary relationship



Part

Vendor

Supply Schedule
*Part_id
*Vendor_id
*Warehouse_id
Shipping Mode
Unit Cost

Warehouse

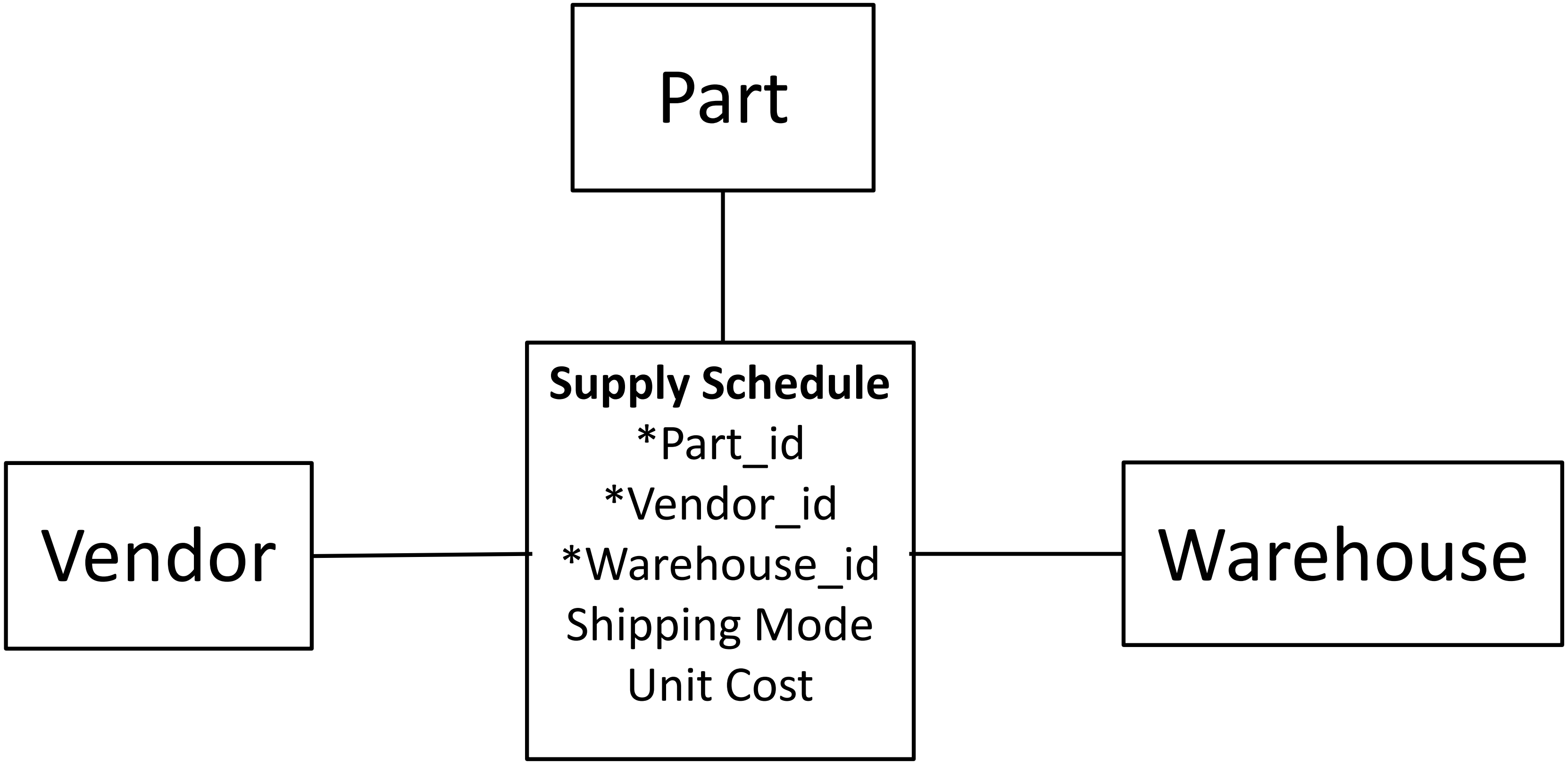
Note: a ternary relationship is ***not the same*** as three binary relationships!
e.g. Unit cost cannot be properly associated with any one of the three possible binary relationships among the three entity types.

Ternary relationship 4

- Business rules
  - Each vendor can supply many parts to any number of warehouses but need not supply any parts.
  - Each part can be supplied by any number of vendors to more than one warehouses, but each part must be supplied by at least one vendor to a warehouse.
  - Each warehouse can be supplied with any number of parts from more than one vendor, but each warehouse must be supplied with at least one part.

Ternary relationship 5

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



ER model summary

Data modelling

Data modelling

- A technique for modelling data
- A graphical representation of a database
- The goal is to identify the facts to be stored in a database
  - not concerned with how the data will be stored
  - not concerned with how the data will be processed
- Data modeling is a partnership between the client and designer
- Drawing a data model is an **iterative** process of trial and revision.

Data modelling: The building blocks

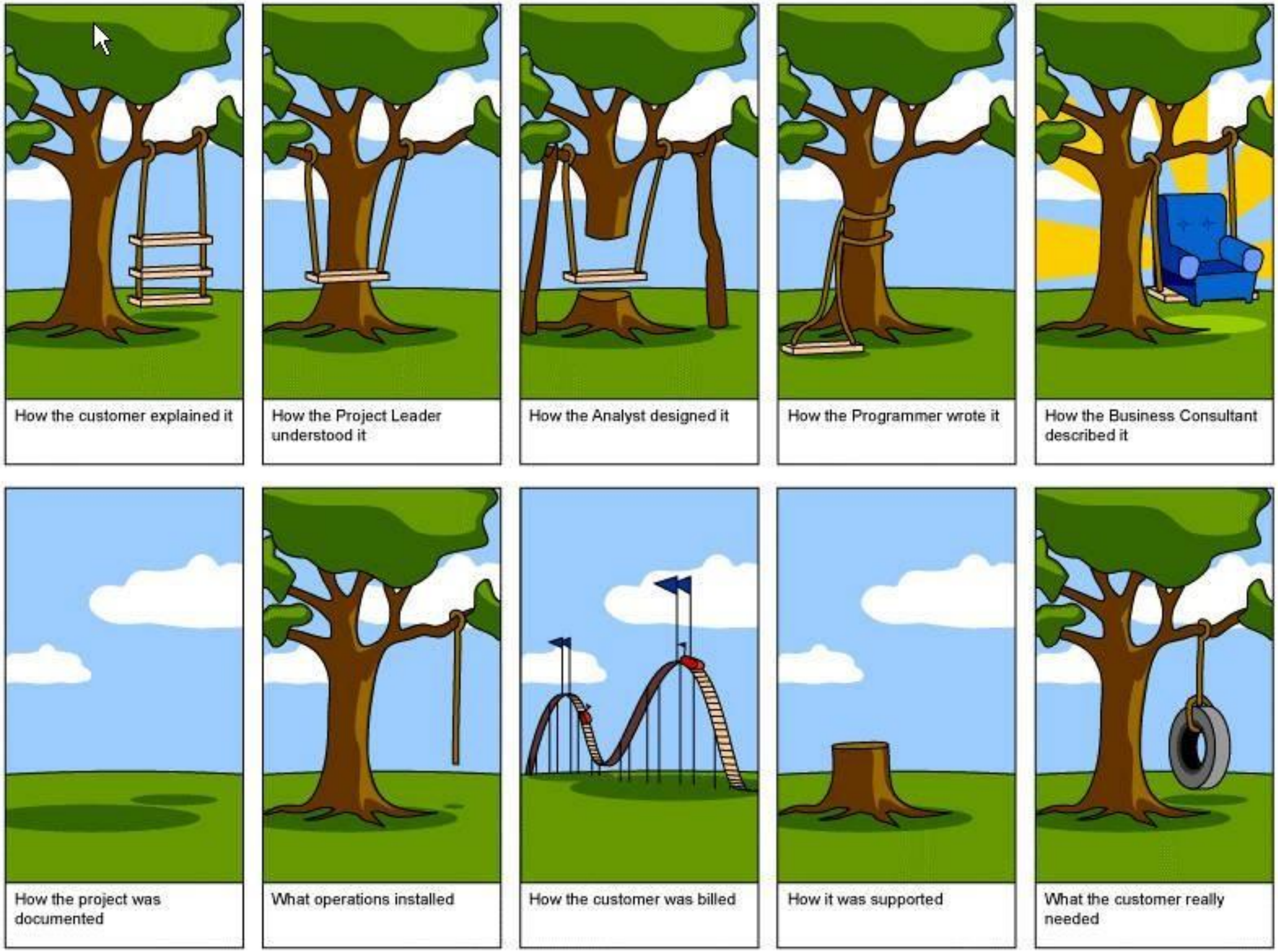
- Entity
  - Find any *nouns* in the system description or provided documentation.
- Attribute
  - Data required to describe fully an entity.
- Relationship
- Identifier (key)
  - No part of an identifier can be *null*.

Data model quality

- A well-formed data model

- A high fidelity image

Without data model


How the customer explained it
How the Project Leader understood it
How the Analyst designed it
How the Programmer wrote it
How the Business Consultant described it
How the project was documented
What operations installed
How the customer was billed
How it was supported
What the customer really needed

http://gerardnico.com/

Data model quality: A well-formed data model

- Construction rules obeyed
- No ambiguity
  - All entities, attributes, relationships, and identifiers are defined
  - All relationships are represented, using the correct notation
  - Relationships are labeled to avoid misunderstanding
  - All attribute names are meaningful and unique
  - Names are meaningful to the client

Data model quality: A high fidelity image

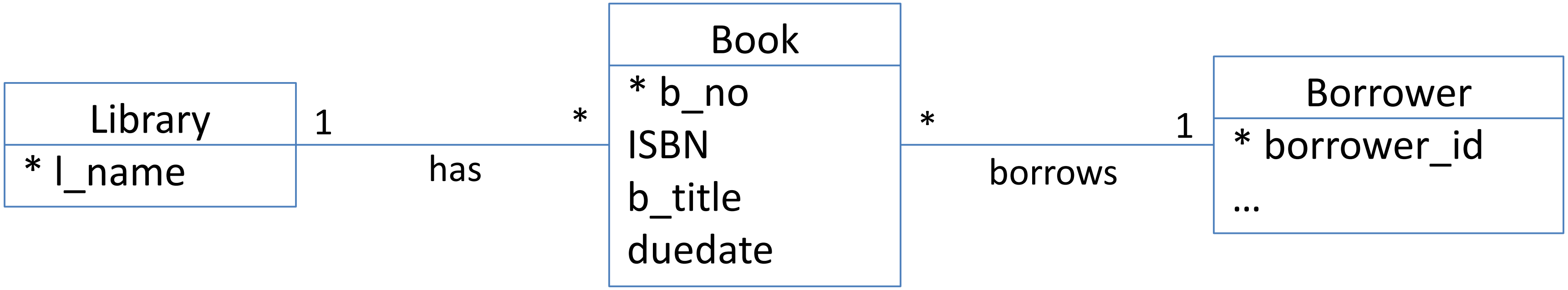
- Faithfully describes the world it is supposed to represent
- Relationships are of the correct degree
- Data model is complete, understandable, and accurate
- The data model makes sense to the client

Data model: Quality improvement

- Drawing a data model is an **iterative** process of trial and revision.
  - Is the level of detail correct?
  - Are all exceptions handled?
  - Is the model accurate?

Data model Quality improvement: Library

- Consider the following data model for library.

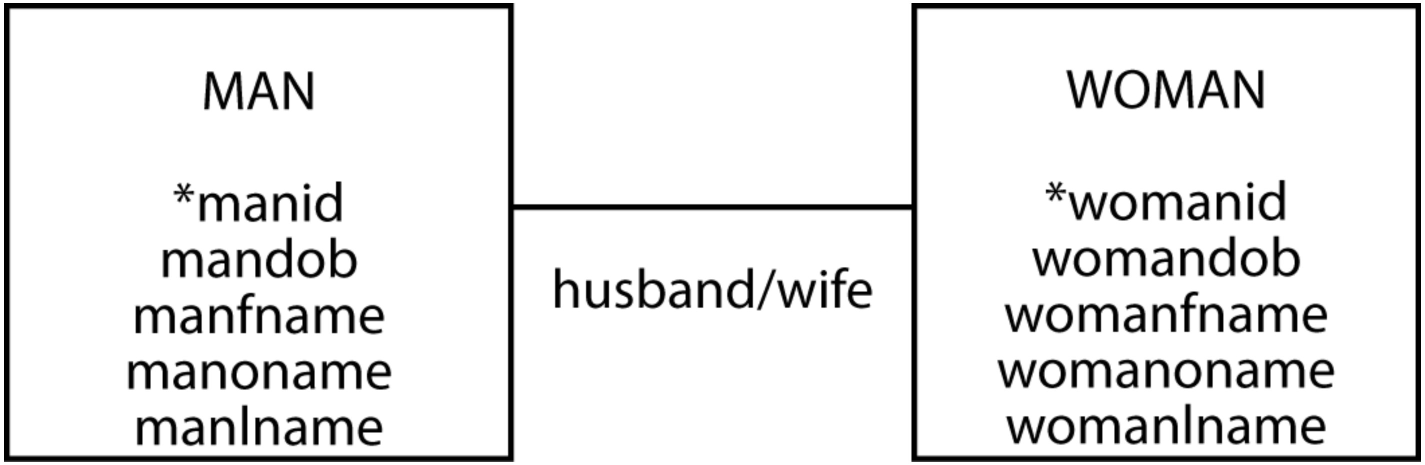


- What happens if the library has two copies of the book?
- Add an attribute to Book called *copy number*?
- ISBN vs b_no?

Revised library data model

Data model Quality improvement: Family matters - take 1

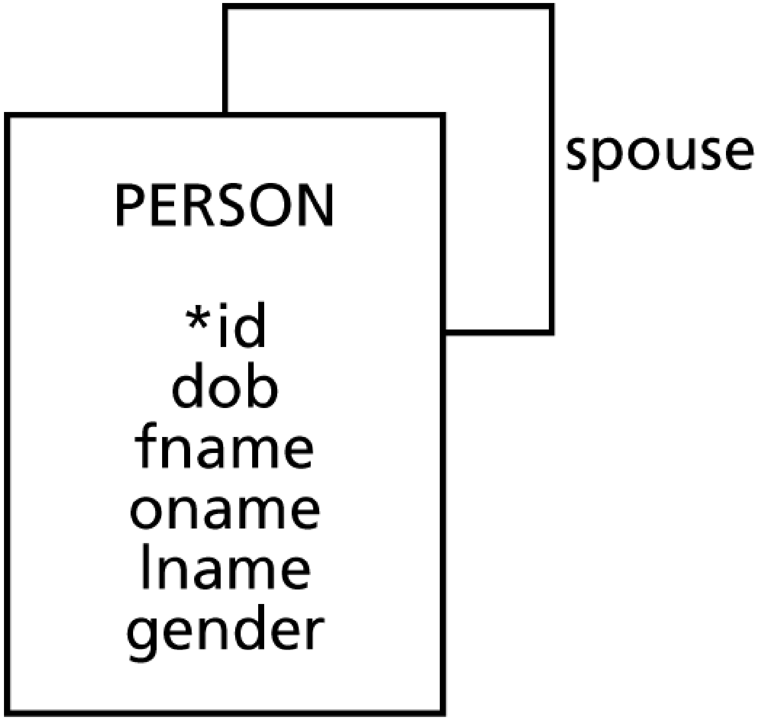
- Families can be very complicated. We start with a very limited view of marriage and gradually ease the restrictions to demonstrate how any aspects of a relationship can be modelled.



- What’s the problem of this model?

Data model Quality improvement: Family matters - take 2

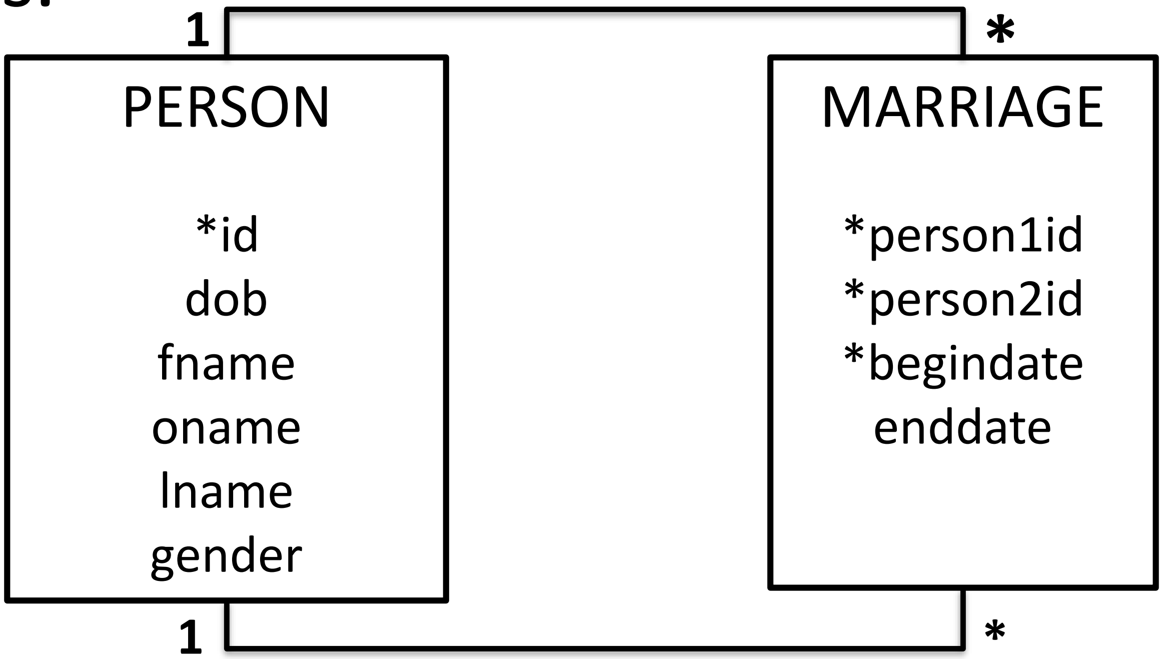
- Usually when entities are combined, you have to create a new attribute to distinguish between different types.
- Generalise the relationship label to *spouse*.



- Problems?

Data model Quality improvement: Family matters - take 3

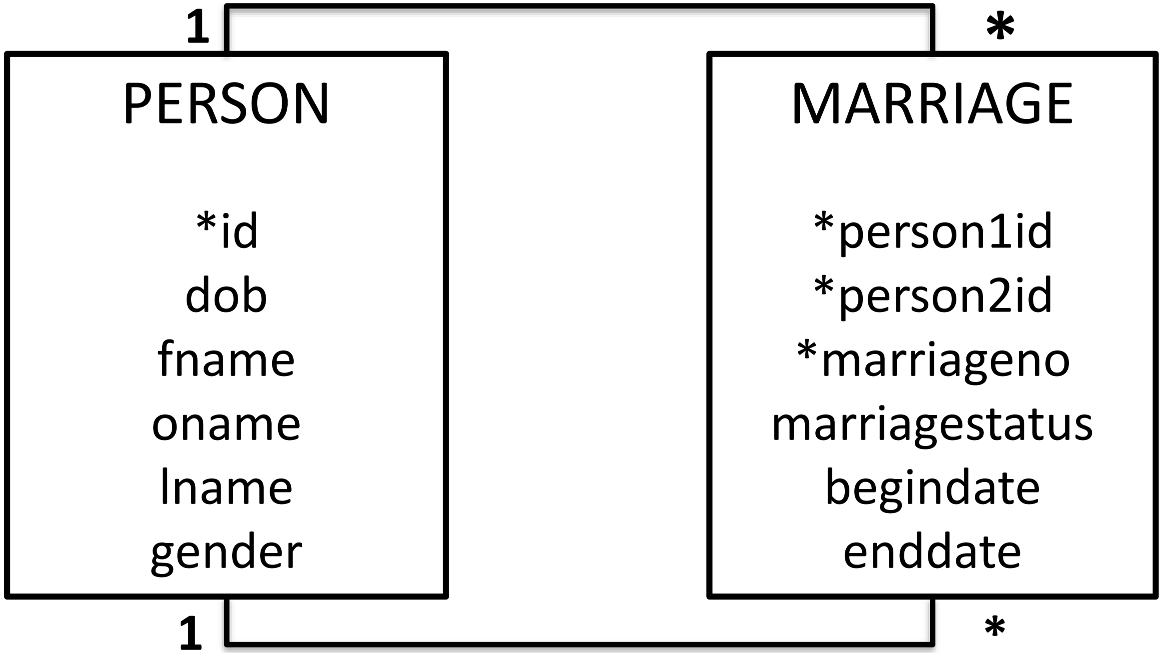
- Marriage is an m:m relationship between two persons.



What about couples who are not officially married but have cohabited for an extended period?

Data model Quality improvement: Family matters - take 4

- Two new attributes added to handle a common-law marriage.
  - *Marriageno* can count the number of times a couple has been married to each other.
  - *Marriagestatus* can record whether a marriage is current or ended.



PERSON

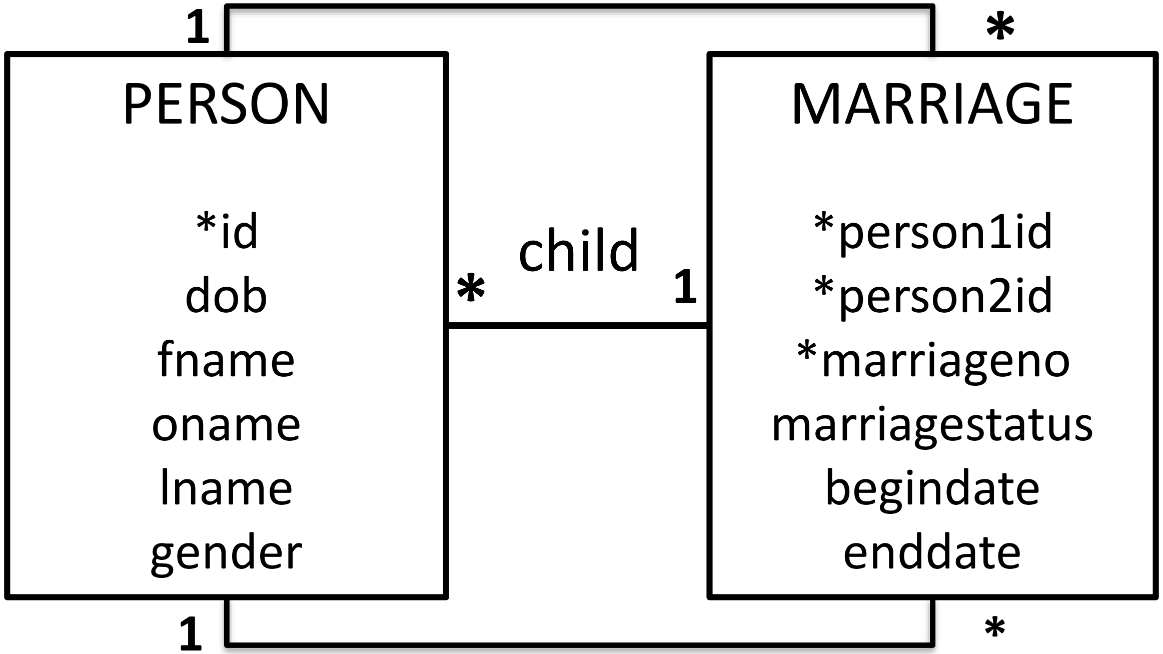
*id
dob
fname
oname
lname
gender

MARRIAGE

*person1id
*person2id
*marriageno
marriagestatus
begindate
enddate

Data model Quality improvement: Family matters - take 5

- Now that we have the couple successfully married, we need to start thinking about children.
- A marriage has zero or more children, and let's start with the assumption a child belongs to only one marriage.



PERSON

*id
dob
fname
oname
lname
gender

MARRIAGE

*person1id
*person2id
*marriageno
marriagestatus
begindate
enddate

Any other situations to consider?

Hints on data modeling

- The model will expand and contract
- Invent identifiers(keys) where necessary
- Keys should have only one purpose – identification
- A data model does not imply ordering
- Create an attribute if ordering of instances is required
- An attribute’s meaning must be consistent

Hints on data modeling

- Single instance entities are OK
- Select names carefully
- Synonyms—different words have the same meaning
  - Get clients to settle on a common word or use views
- Homonyms—same word has different meanings
  - Clarify to avoid confusion
- Naming associative entities
  - Concatenate entity names if there is no obvious real world name

Hints on data modeling

- Uncover all exceptions
- Label relationships to avoid ambiguity
- Keep the data model well-formed and accurate

Making assumptions

- Data model is all about documenting rules and policies of an organisation.
- Database analyst should:
  - Identify and understand those rules that govern data
  - Represent those rules so that they can be unambiguously understood by information systems developers and users
  - Implement those rules in database technology
- Business rules can be gathered by *interviews* and *organisation documents*(policies, manuals, procedures etc)
- Sometimes a data analyst has to ask questions to clarify business rules.
- Occasionally a data analyst has to **make assumptions**.

Extra reading on business rules: Modern Database Management, Chapter 2, “Modelling the rules of the organization”

Making assumptions

- Different assumptions can result in different data models.
- Assumptions have to be sensible and reasonable.
- You must **clearly state the assumptions** if you make any – important for your coursework and exam!

Exercise

- A cinema has multiple theatres. Movies are shown through the day starting at 11am and finishing at 11am. Each movie is given a two-hour time slot.
- One movie is never shown in more than one theatre at a time, but movies can be shifted among theatres because seating capacity varies.
- The cinema boss also want to store the data of how many people, classified by adults and children, attended each showing of a movie.
- Ticket prices vary by movie and time slot. For example, X-men Apocalypse is £10 for everyone at 11am but is £15 at 9pm.

Clearly state the assumptions if you make any.