

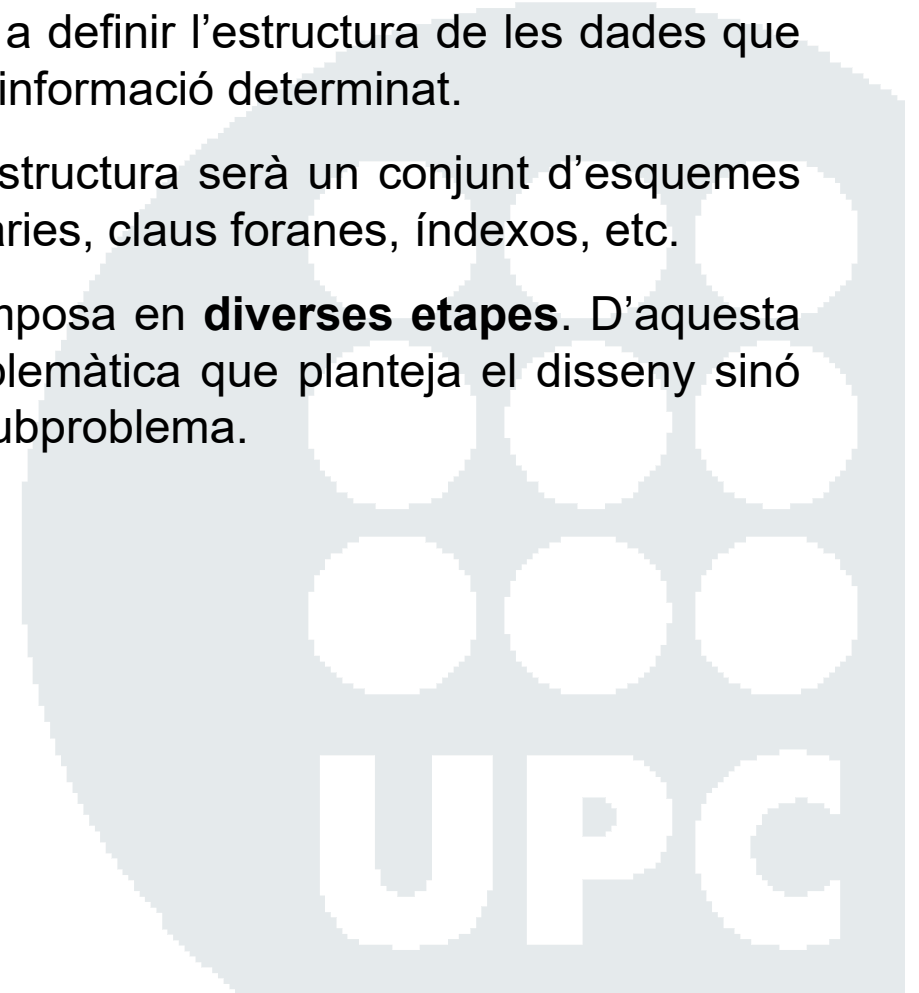
## 7. Introducció al disseny de bases de dades relacionals

- Introducció
- Punt de partida: disseny conceptual (o especificació) en UML
- Disseny lògic: transformació al model relacional



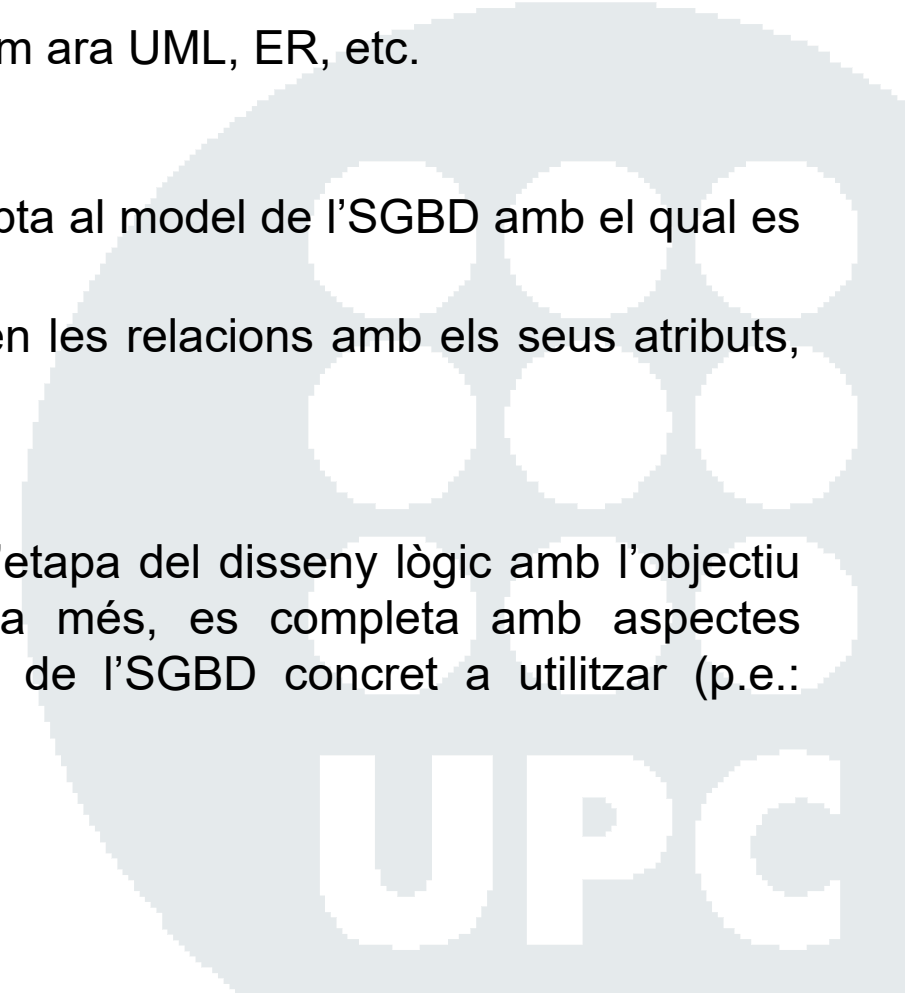
## Introducció

- El **disseny d'una BD relacional** consisteix a definir l'estructura de les dades que ha de tenir la base de dades d'un sistema d'informació determinat.
- En el cas de les BD relacionals, aquesta estructura serà un conjunt d'esquemes de relació amb els seus atributs, claus primàries, claus foranes, índexos, etc.
- El **procés de disseny** d'una BD es descomposa en **diverses etapes**. D'aquesta manera no cal resoldre de cop tota la problemàtica que planteja el disseny sinó que a cada etapa s'afronta un sol tipus de subproblema.

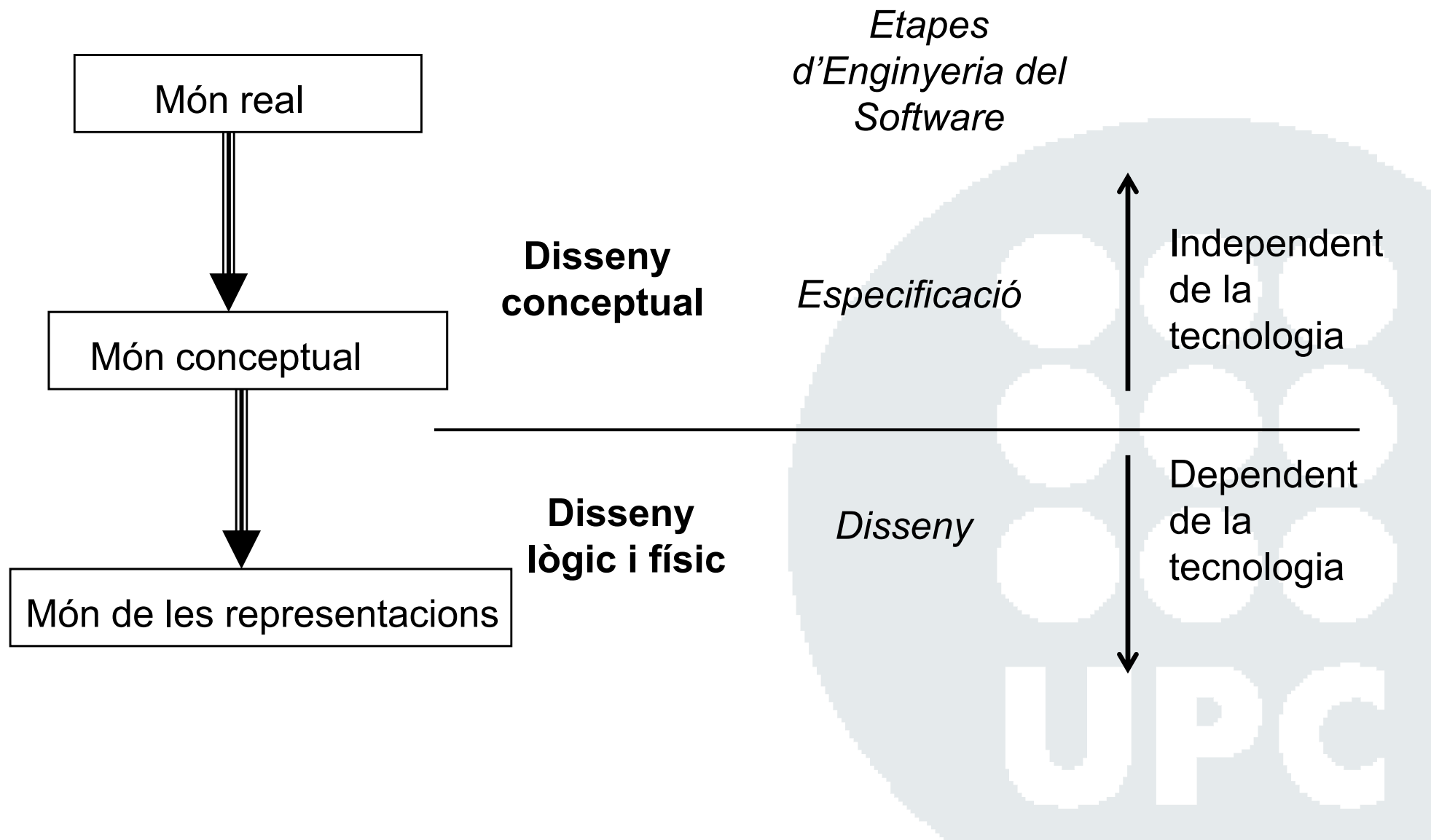


## Introducció: etapes del disseny de BD

- **Disseny conceptual:**
  - S'obté una estructura de la informació de la futura BD independent de la tecnologia que cal emprar.
  - S'obté un esquema en un llenguatge com ara UML, ER, etc.
- **Disseny lògic:**
  - El resultat del disseny conceptual s'adapta al model de l'SGBD amb el qual es desitja implementar la BD.
  - En el cas del model relacional s'obtenen les relacions amb els seus atributs, claus primàries i claus foranes.
- **Disseny físic:**
  - Es transforma l'estructura obinguda a l'etapa del disseny lògic amb l'objectiu d'aconseguir una major eficiència i, a més, es completa amb aspectes d'implementació física que dependran de l'SGBD concret a utilitzar (p.e.: índexos).



## Introducció: el disseny de BD i els tres mons



## **Punt de partida: disseny conceptual (o especificació) en UML**

- Classes d'objectes i atributs
- Associacions
- Multiplicitat de les associacions
- Associacions recursives
- Classes associatives
- Generalització/especialització



## Classes d'objectes i atributs

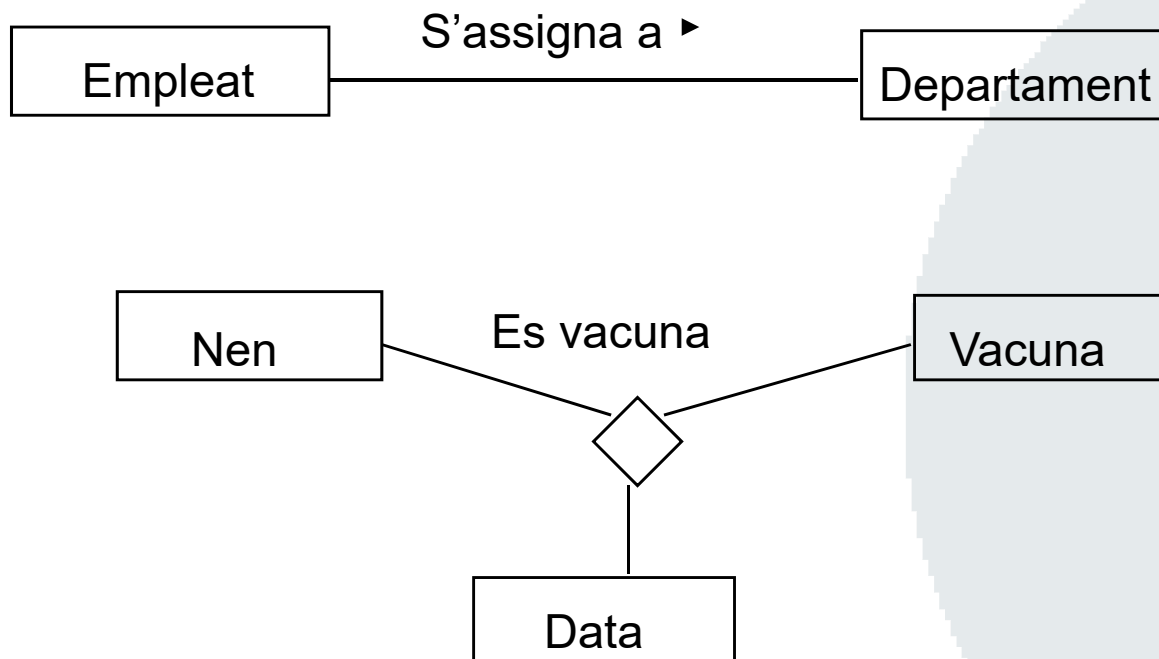
Empleat
dni nom cognom sou

Clau externa Empleat: dni

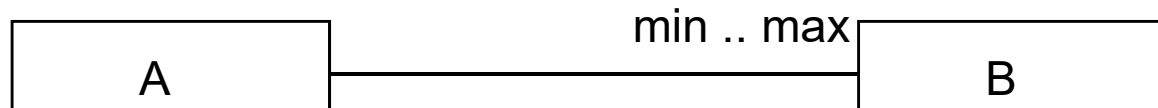
- **Classe d'objectes:** Descriu un conjunt d'objectes similars dels quals interessen les mateixes propietats.
- Els objectes tenen **identitat** (OID) i són distingibles entre ells.
- **Atribut:** Propietat compartida pels objectes d'una classe.
- **Clau externa**
  - Conjunt d'atributs (mínim) que permeten identificar els objectes d'una classe.
  - UML no proporciona notació gràfica per assenyalar les claus externes de les classes d'objectes.
  - Es poden descriure amb restriccions textuais.

## Associacions

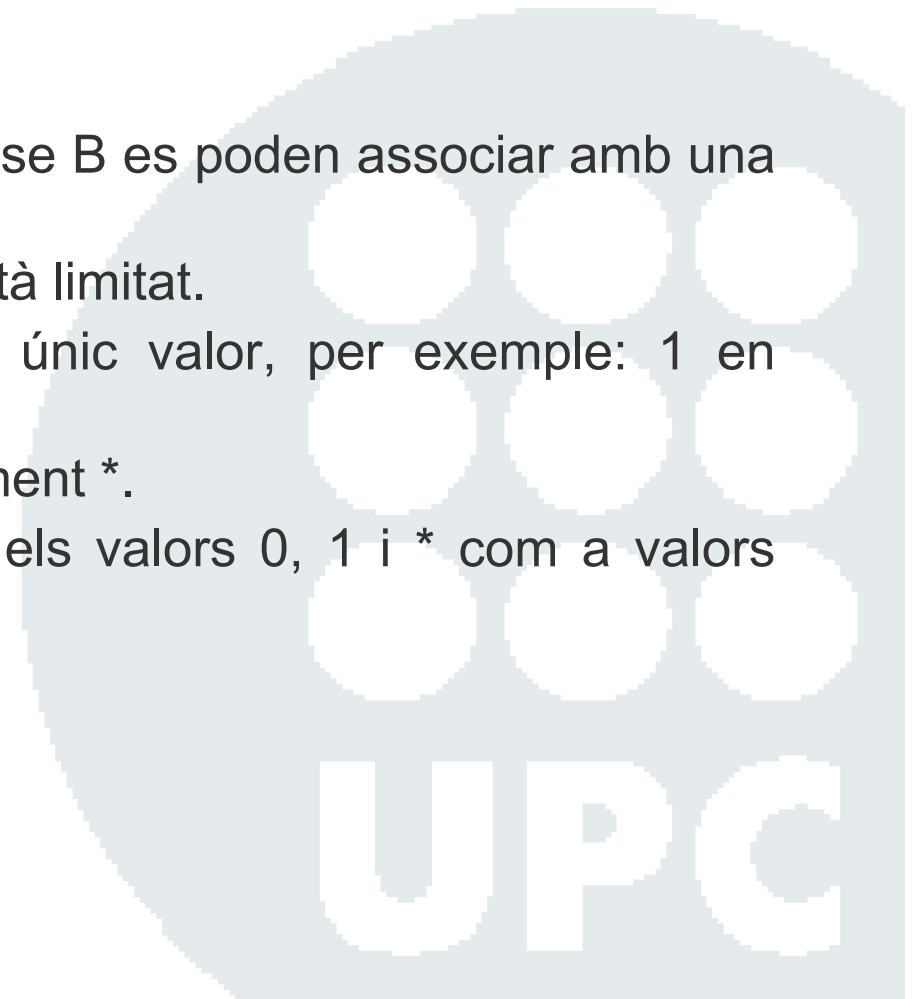
- Una associació és la representació d'una relació entre dos o més objectes.
- Poden ser binàries o d'ordre superior a dos.



## Multiplicitat de les associacions binàries (1)

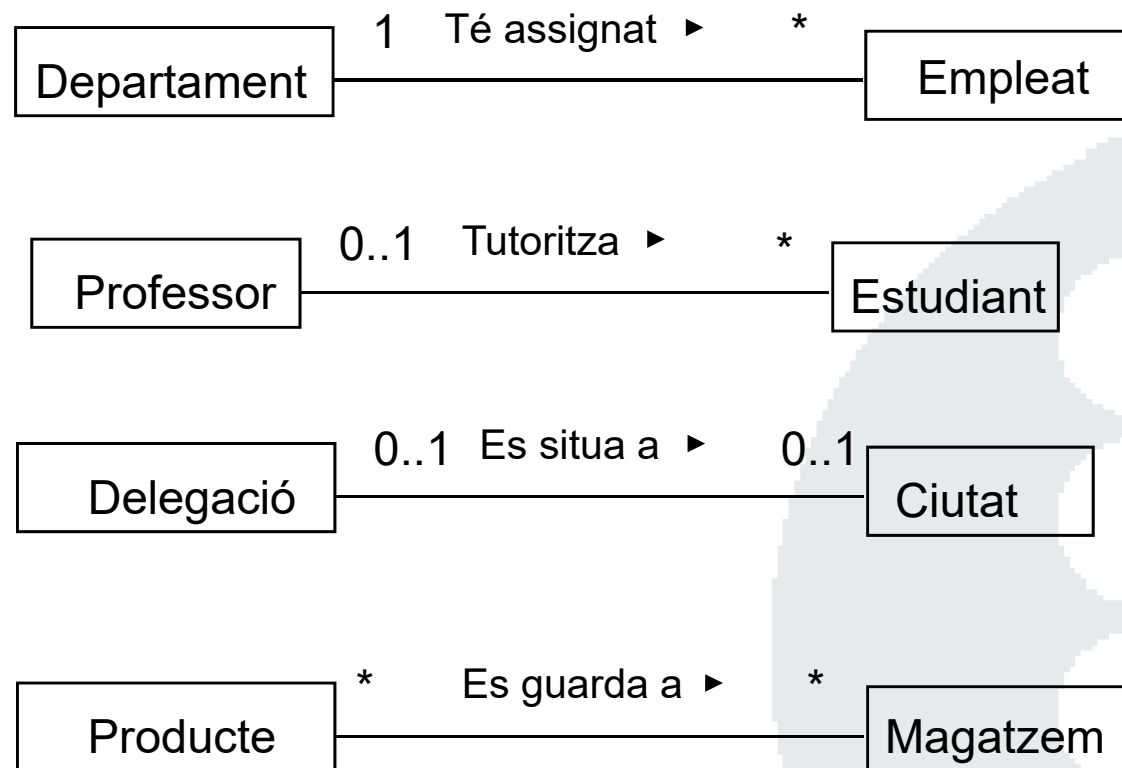


- Defineix quàntes instàncies d'una classe B es poden associar amb una instància de la classe A.
- Si max és \* indica que el màxim no està limitat.
- Si min=max es pot indicar amb un únic valor, per exemple: 1 en comptes d' 1..1.
- La multiplicitat 0..\* s'indica abreviadament \*.
- Per simplificar, considerarem només els valors 0, 1 i \* com a valors mínims i/o màxims de la multiplicitat.

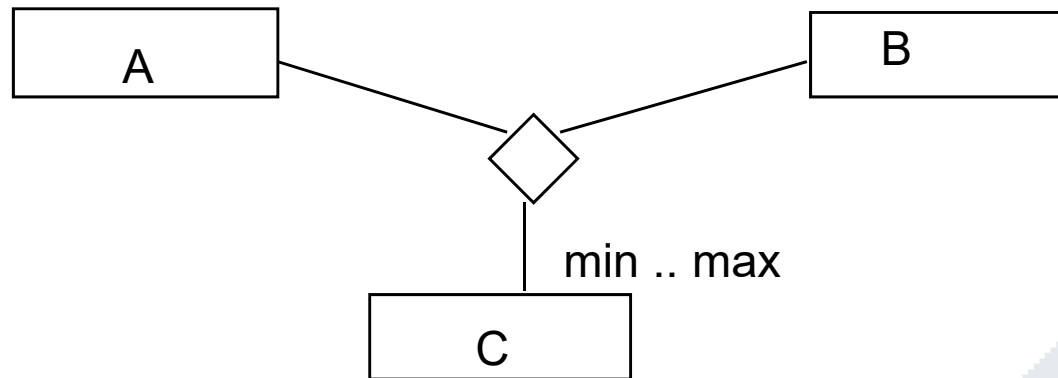




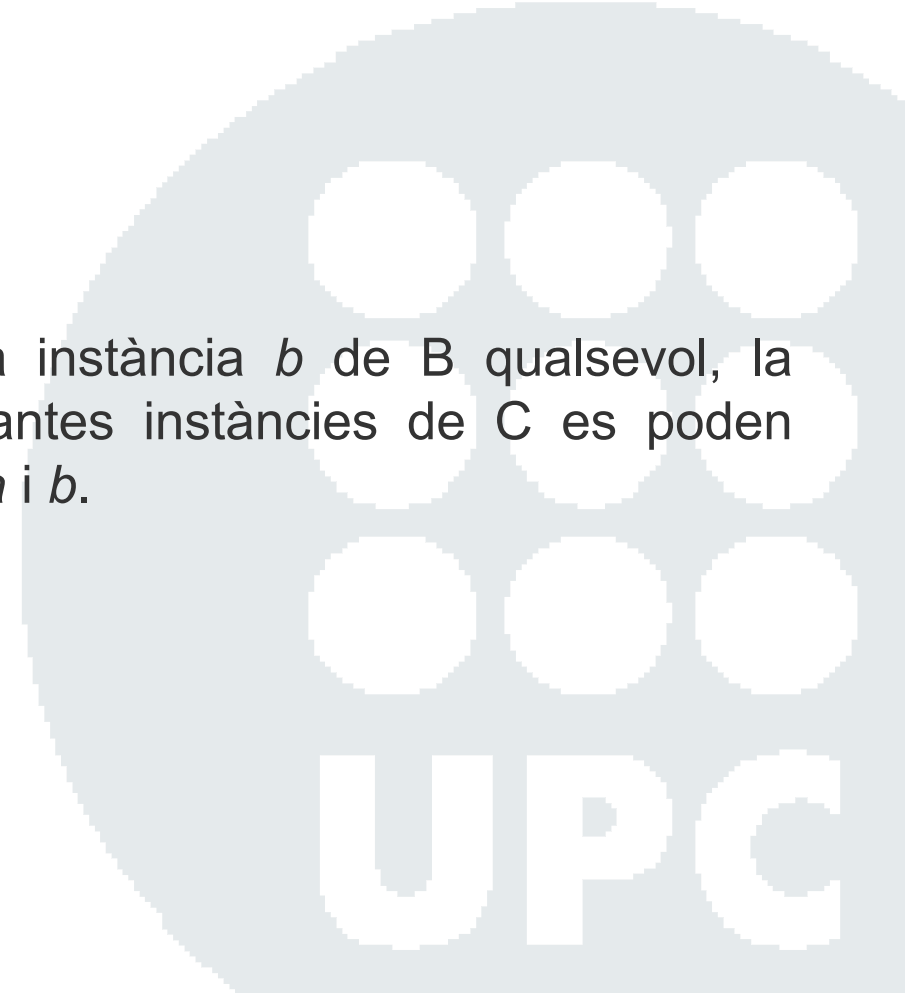
## Multiplicitat de les associacions binàries: exemples



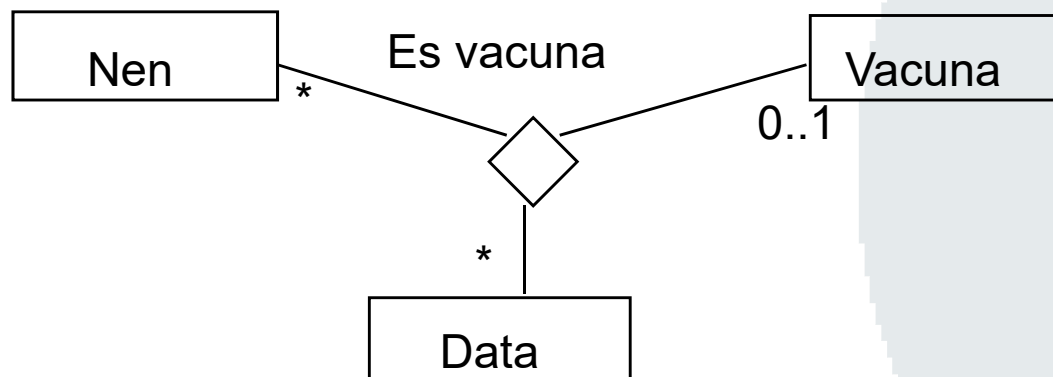
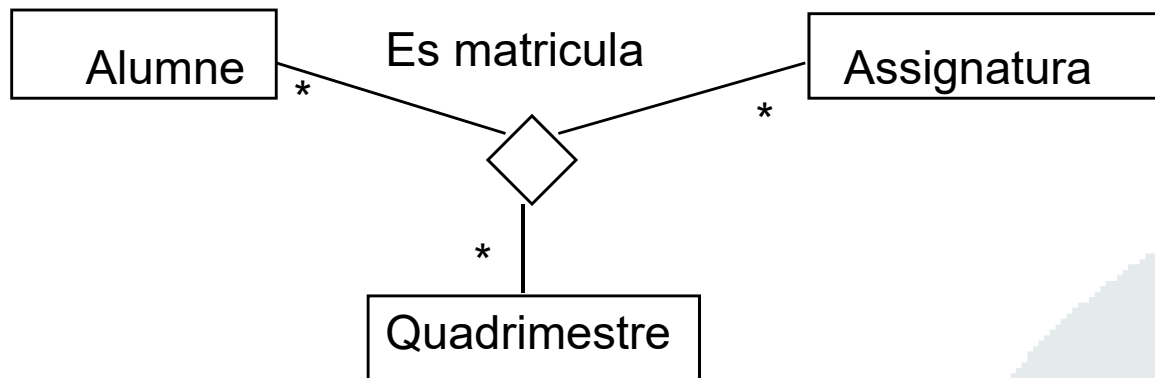
## Multiplicitat de les associacions ternàries



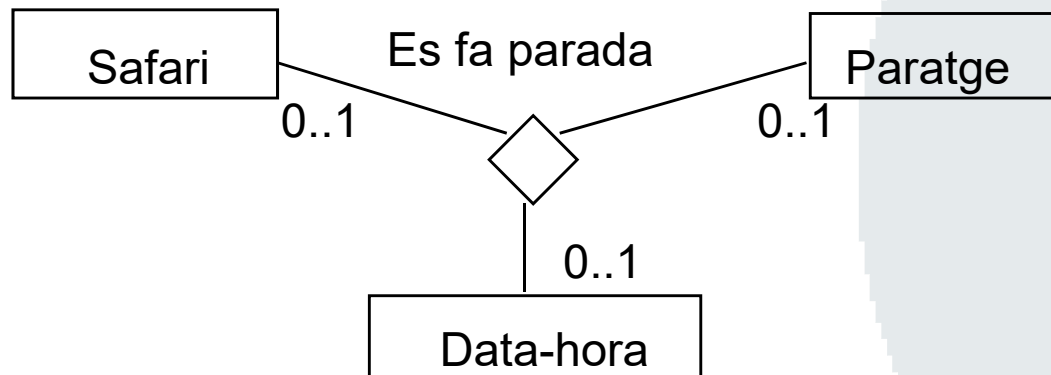
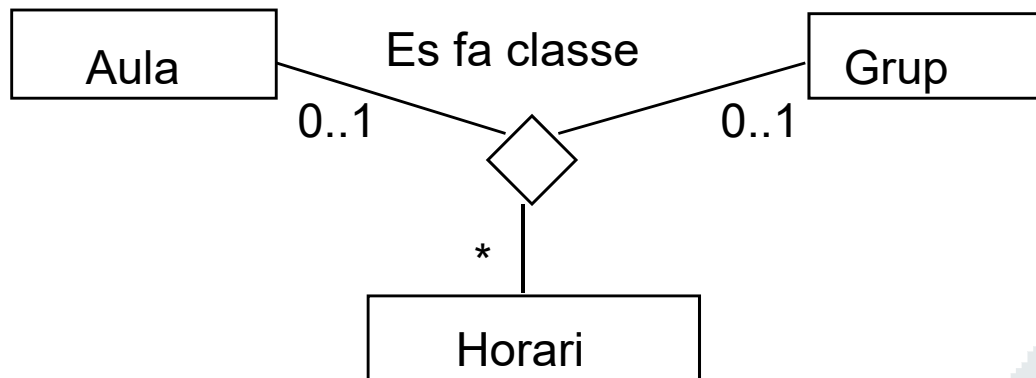
- Donades una instància  $a$  d'A i una instància  $b$  de B qualsevol, la multiplicitat al costat C defineix quantes instàncies de C es poden associar amb la parella formada per  $a$  i  $b$ .



## Multiplicitat de les associacions ternàries: exemples (1)

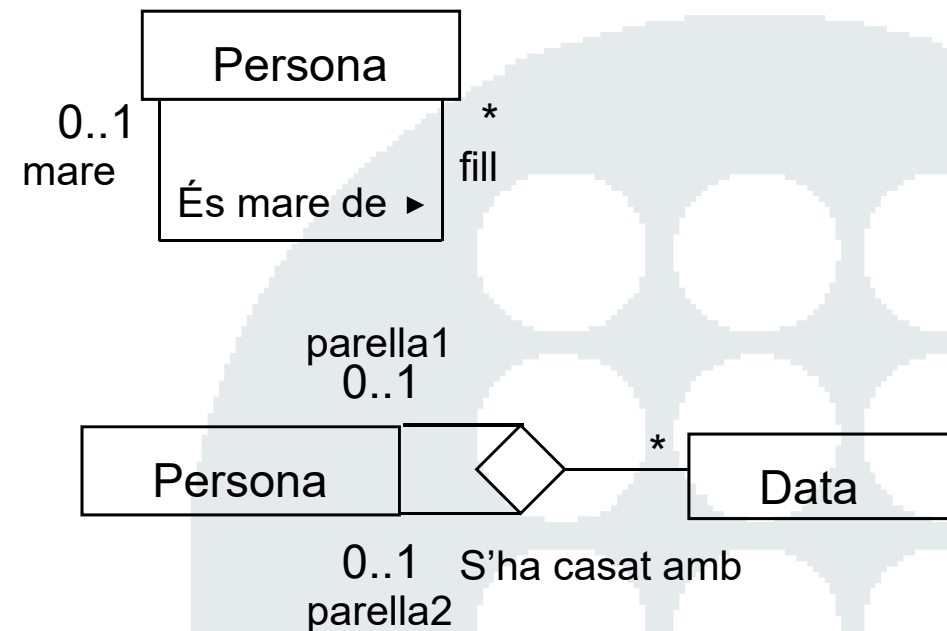
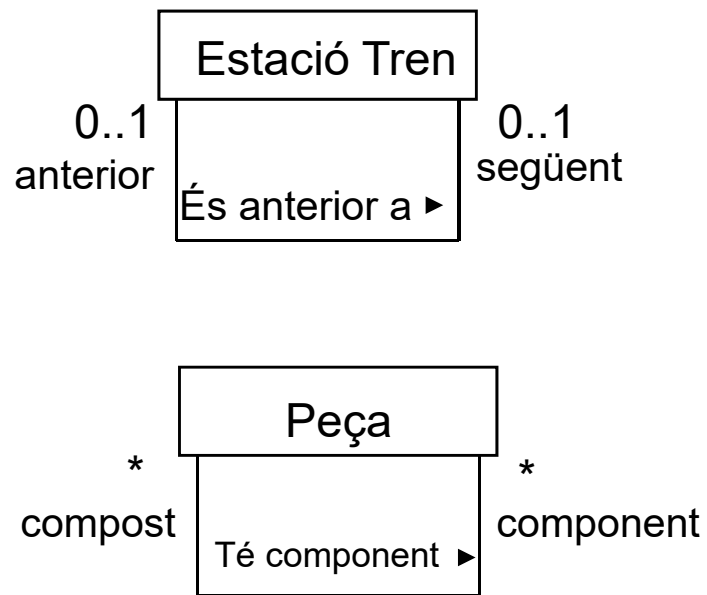


## Multiplicitat de les associacions ternàries: exemples (2)



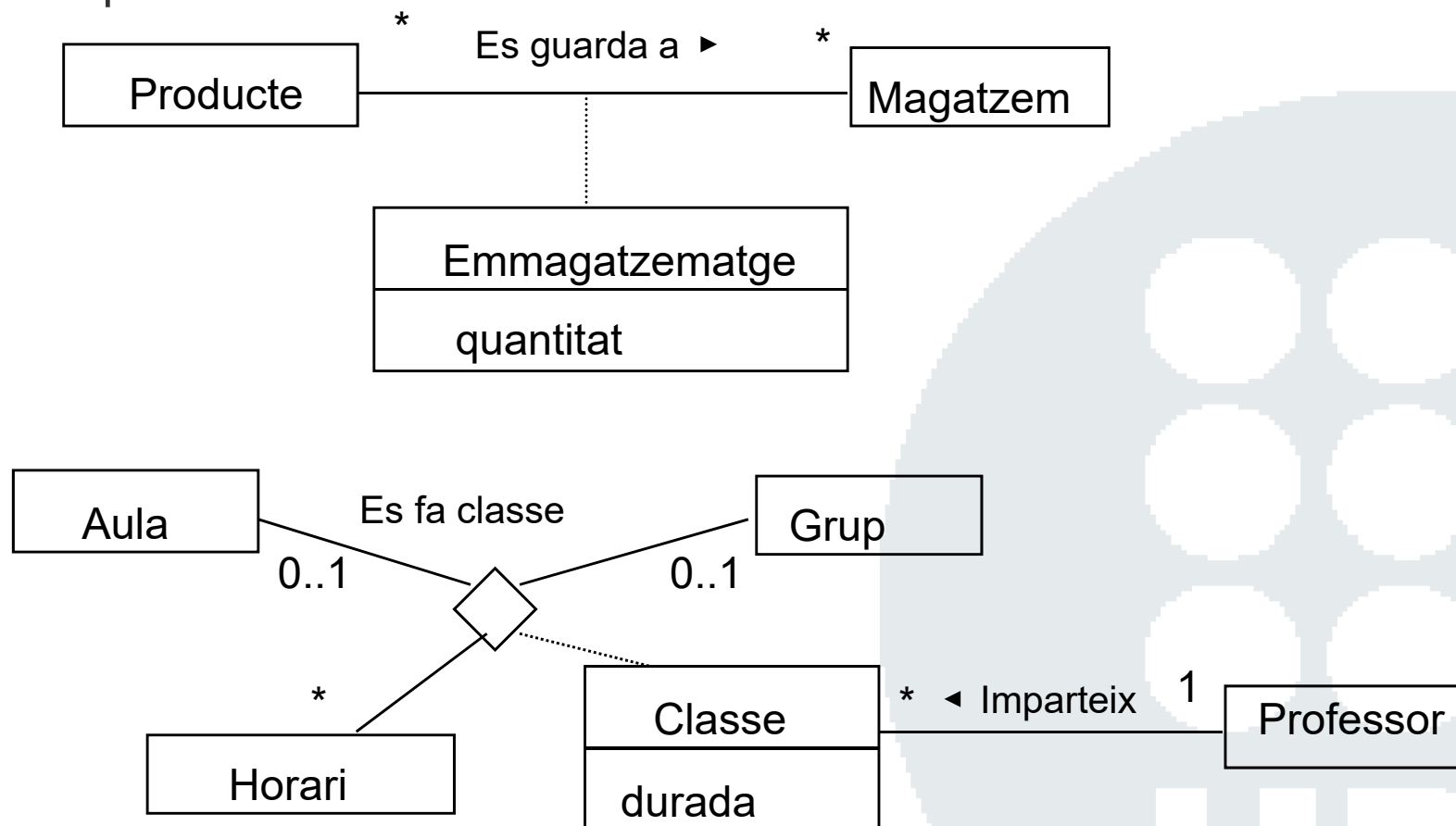
## Associacions recursives

- Associacions en les que una mateixa classe d'objectes hi participa més d'una vegada.
- Exemples:



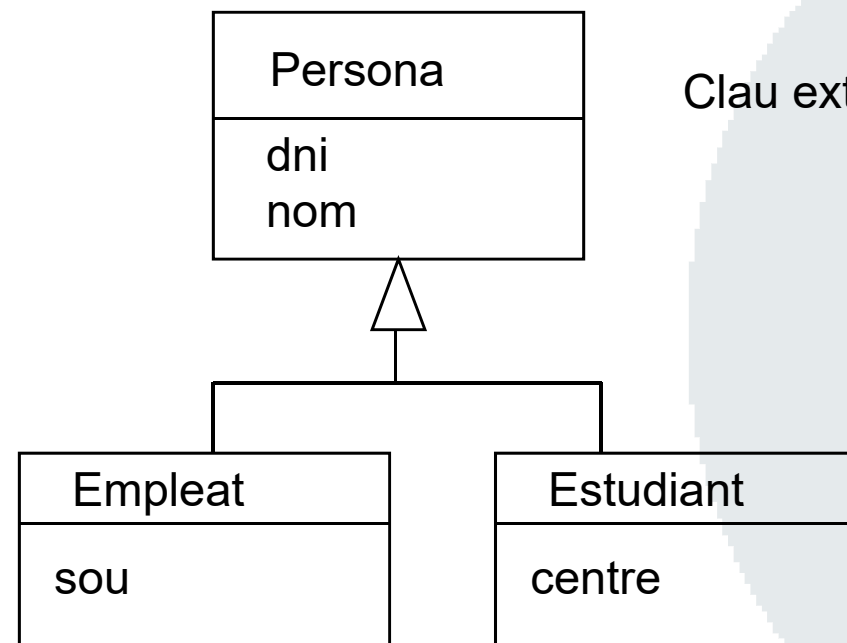
## Classes associatives

- Representa una associació que es pot veure com una classe (així pot tenir atributs i associacions).
- Exemples:

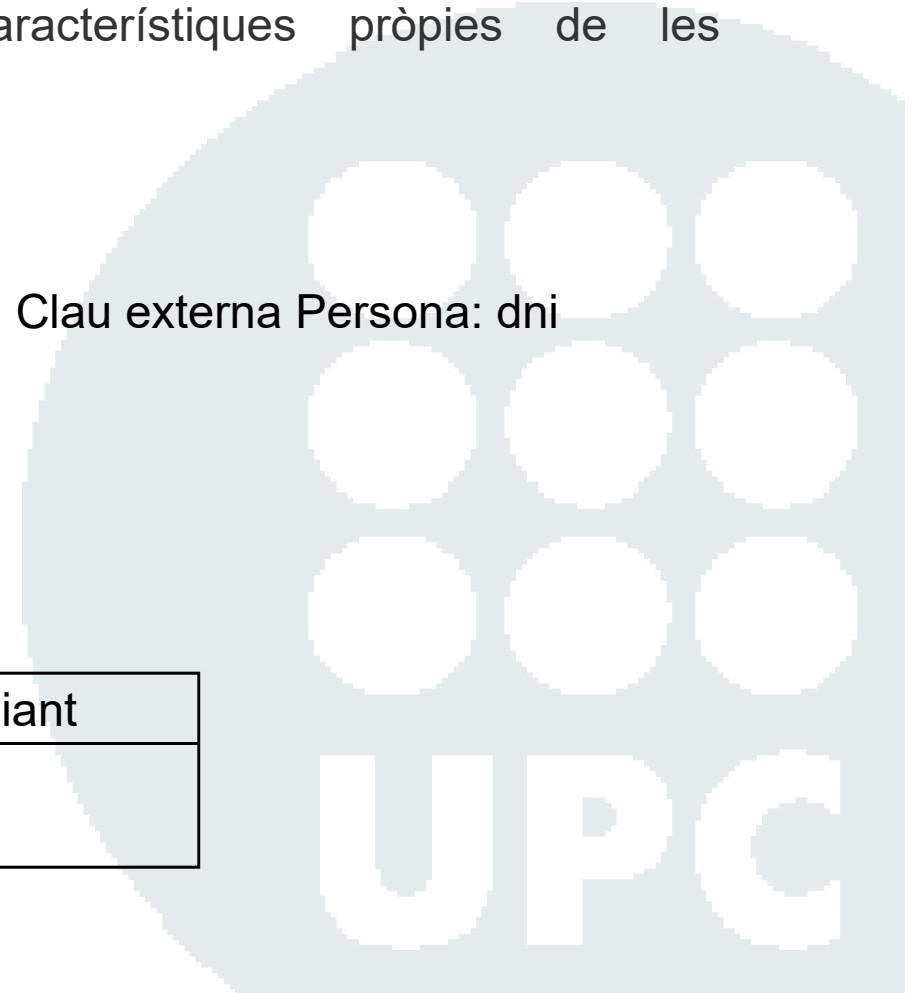


## Generalització/especialització

- Permet reflectir el fet que hi ha una classe d'objectes general, que anomenem superclasse, que es pot especialitzar en entitats subclasse.
  - La superclasse representa les característiques comunes a totes les subclasses.
  - Les subclasses representen les característiques pròpies de les especialitzacions.

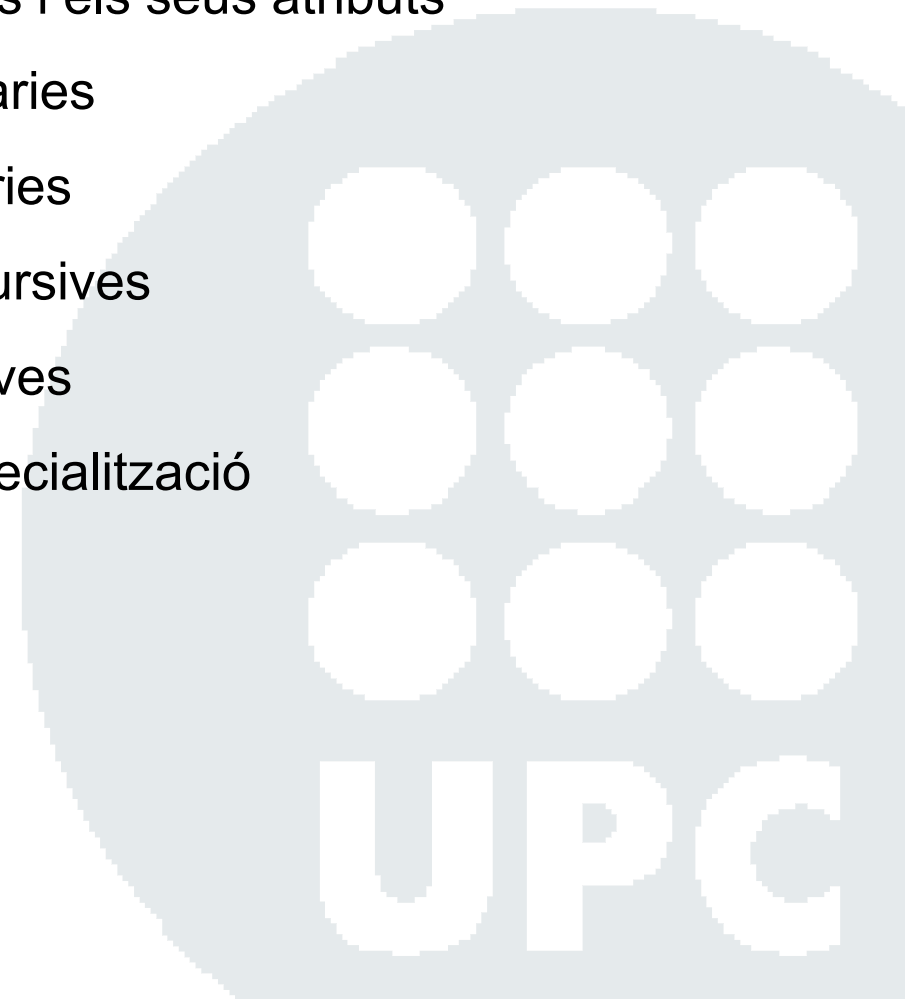


Clau externa Persona: dni



## Disseny lògic: transformació al model relacional

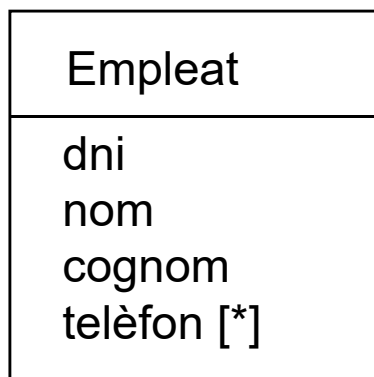
- Transformacions inicials del model de partida
- Transformació de les classes d'objectes i els seus atributs
- Transformació de les associacions binàries
- Transformació de les associacions n-àries
- Transformació de les associacions recursives
- Transformació de les classes associatives
- Transformació de la generalització/especialització
- Resum de les transformacions
- Consideracions finals



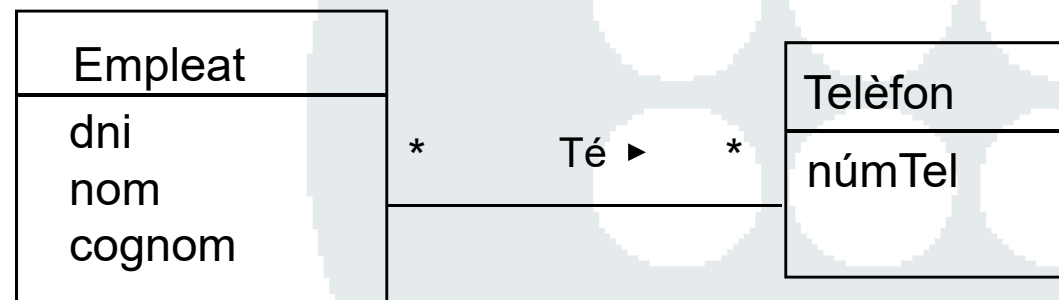


## Transformacions inicials del model de partida

- Per poder aplicar les transformacions que veurem més endavant cal que el model de partida compleixi els requisits següents:
  - Totes les classes d'objectes (no associatives ni subclasses d'altres classes) han de tenir restricció de clau externa.
  - Tots els atributs han de ser univaluats (no poden ser multivaluats).
- Si algun d'aquests requisits no es compleix cal transformar el model de partida de manera que es compleixi:
  - Afegir atributs amb restriccions de clau externa a les classes que calgui.
  - Transformar els atributs multivaluats en associacions.
- Exemple de transformació d'un atribut multivaluat:



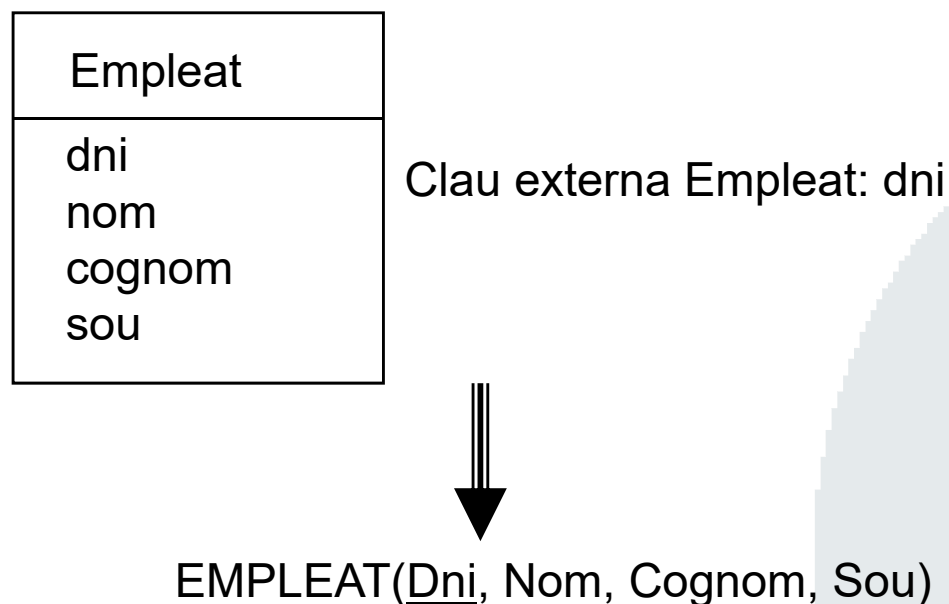
Clau externa Empleat: dni



Clau externa Empleat: dni  
Telèfon: númTel

## Transformació de les classes d'objectes i els seus atributs

- Cada classe d'objectes es transforma en una relació
  - La clau primària de la relació serà la clau externa de la classe d'objectes.
  - Els atributs de la relació seran els atributs de la classe d'objectes.



- Observació: és possible que calgui afegir atributs a la relació quan es faci la transformació de les associacions on participa la classe (es veurà a la transformació de les associacions).

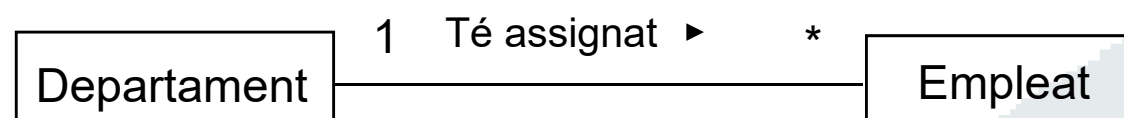
## Transformació de les associacions binàries

- Per transformar una associació (binària o n-ària) cal prèviament haver transformat totes les classes d'objectes que associa.
- Casos a considerar per les associacions binàries:
  - Cas un a molts.
  - Cas un a un.
  - Cas molts a molts.

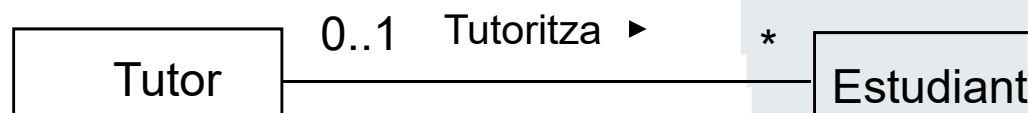


## Associacions binàries: cas un a molts

- Cas un a molts: la multiplicitat és 0..1 o 1 a l'extrem 'un' i \* a l'extrem 'molts' de l'associació.
- La transformació consisteix en afegir una clau forana a la relació que correspon a la classe de l'extrem 'molts' de l'associació per tal de referenciar a l'altra relació.



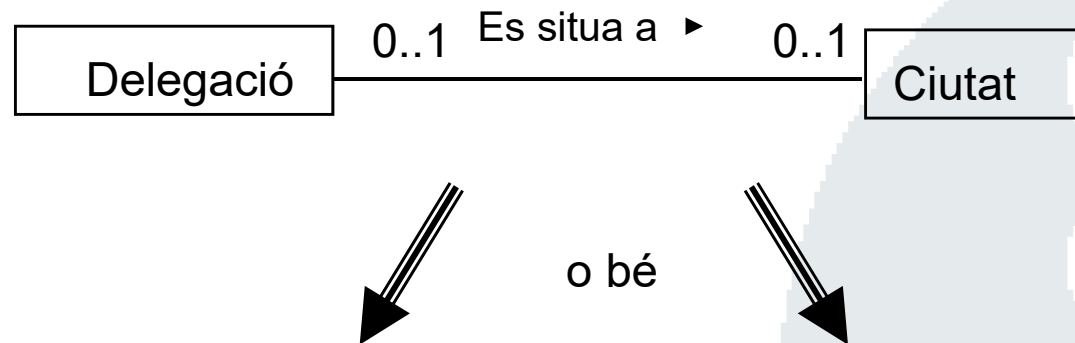
DEPARTAMENT(Dep, ...)  
EMPLEAT(Emp, ..., Dep)  
on {Dep} referencia DEPARTAMENT  
Dep NOT NULL



TUTOR(Tut, ...)  
ESTUDIANT(Est, ..., Tut)  
on {Tut} referencia TUTOR

## Associacions binàries: cas un a un

- Cas un a un significa que la multiplicitat és 0..1 a ambdós extrems de l'associació.
- La transformació consisteix en afegir a qualsevol de les dues relacions (corresponents a les classes associades) una clau forana que referenciï a l'altra relació.



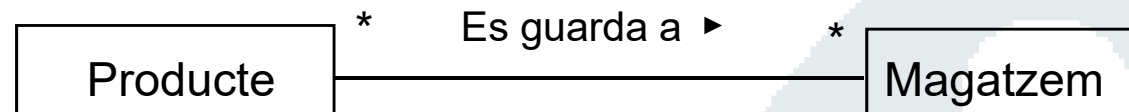
o bé

DELEGACIÓ(Nom-del, ..., Nom-ciutat)  
 on {Nom-ciutat} referencia CIUTAT  
 Nom-ciutat UNIQUE  
 CIUTAT(Nom-ciutat, ...)

DELEGACIÓ(Nom-del, ...)  
 CIUTAT(Nom-ciutat, ..., Nom-del)  
 on {Nom-del} referencia DELEGACIÓ  
 Nom-del UNIQUE

## Associacions binàries: cas molts a molts

- Cas molts a molts significa que la multiplicitat és \* a ambdós extrems de l'associació.
- La transformació consisteix en definir una nova relació. La seva clau primària estarà formada pels atributs de la clau primària de les relacions corresponents a les classes dels dos extrems de l'associació.



PRODUCTE(Pro, ...)

MAGATZEM(Mag, ...)

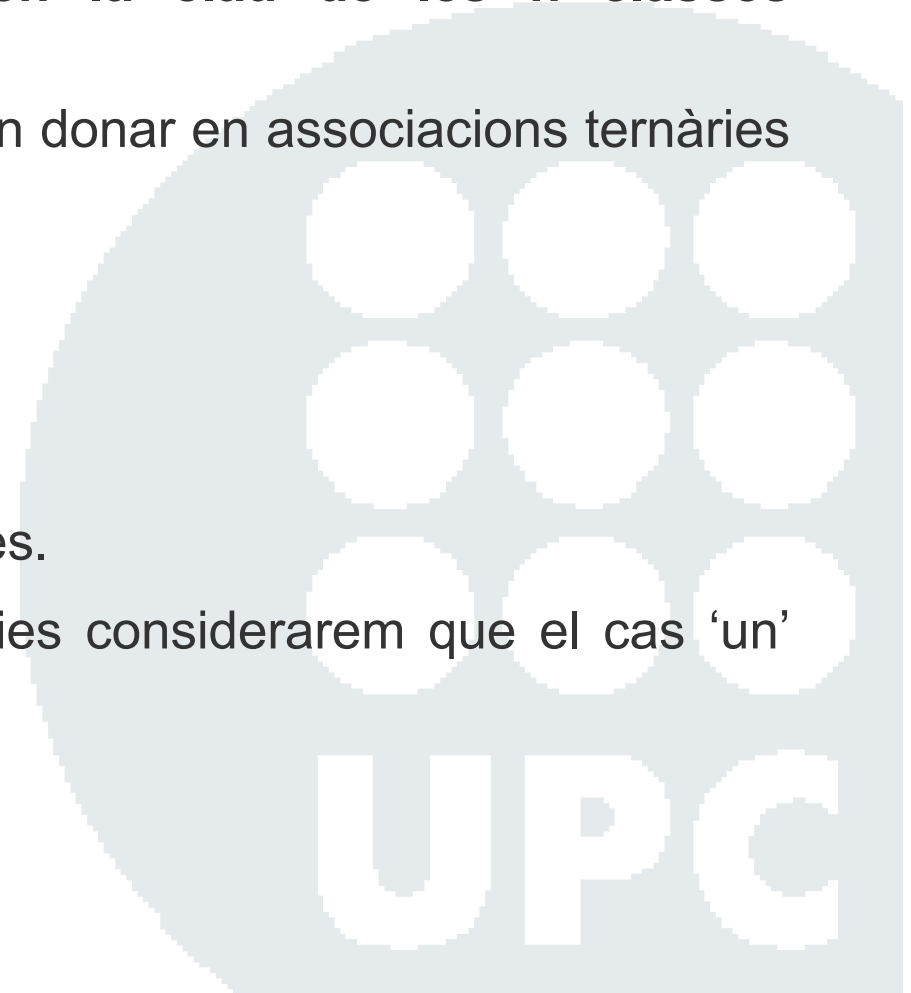
EMMAGATZEMATGE(Pro, Mag)

on {Pro} referencia PRODUCTE i

{Mag} referencia MAGATZEM

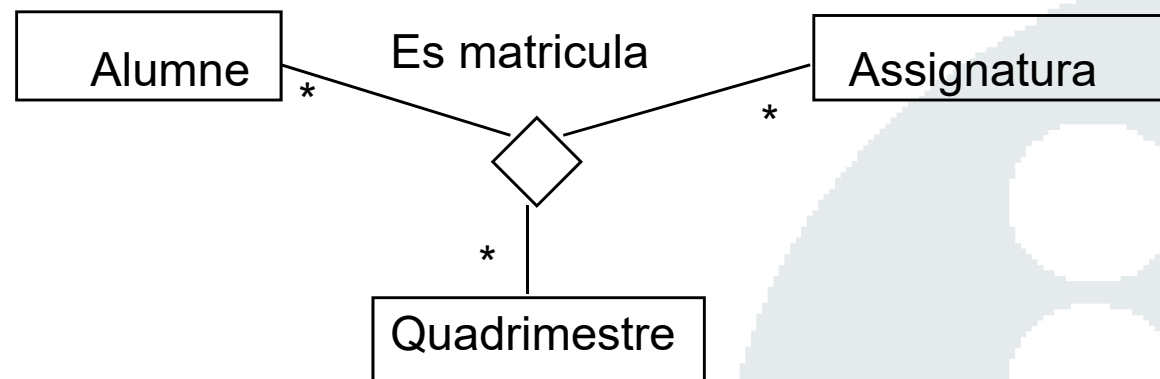
## Transformació de les associacions n-àries

- Per transformar una associació (binària o n-ària) cal prèviament haver transformat totes les classes d'objectes que associa.
- Les associacions n-àries sempre es transformen en una **nova relació que conté els atributs que formen la clau de les n classes associades**.
- Analitzarem alguns casos que es poden donar en associacions ternàries i després descriurem el cas general:
  - Ternàries molts-molts-molts.
  - Ternàries molts-molts-un.
  - Ternàries molts-un-un.
  - Transformació general de les n-àries.
- Observació: per les associacions n-àries considerarem que el cas 'un' correspon només a 0..1 (no 1).



## Associacions ternàries: molts-molts-molts

- En el cas molts-molts-molts la multiplicitat és \* als tres extrems de l'associació.
- La clau primària de la nova relació està formada pels atributs de les claus de les classes corresponents als tres extrems de l'associació.



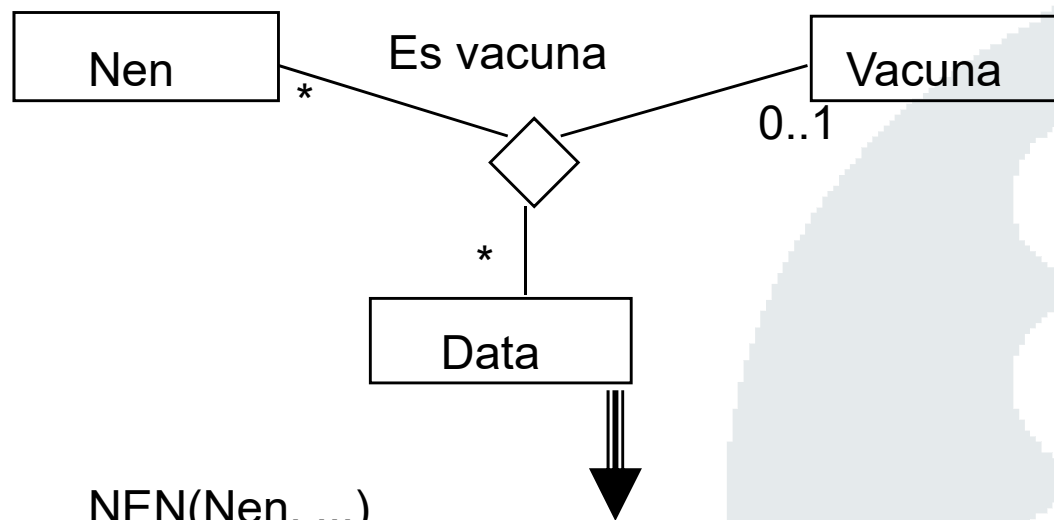
↓

ALUMNE(Alum, ...)  
ASSIGNATURA(Assig, ...)  
QUADRIMESTRE(Quad)  
**MATRICULACIÓ(Alum, Assig, Quad)**  
on {Alum} referencia ALUMNE,  
{Assig} referencia ASSIGNATURA i  
{Quad} referencia QUADRIMESTRE



## Associacions ternàries: molts-molts-un

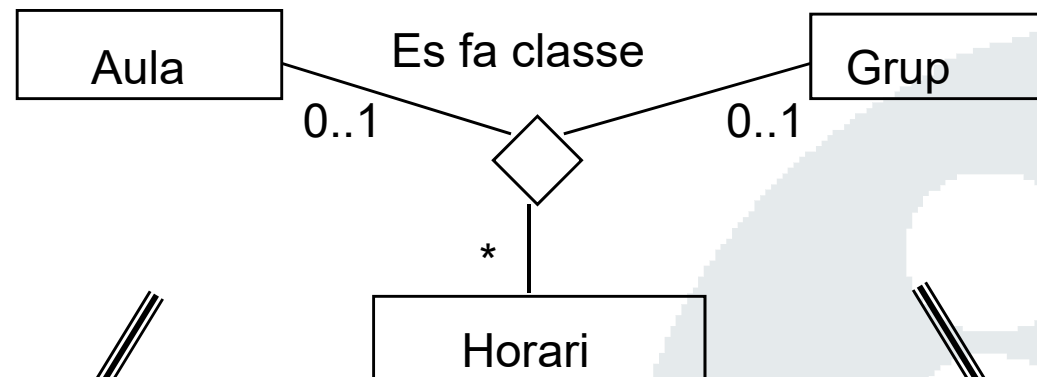
- En el cas molts-molts-un la multiplicitat és \* als dos extrems 'molts' i 0..1 a l'extrem 'un' de l'associació.
- La clau primària està formada pels atributs de les claus de les classes corresponents als dos extrems 'molts' de l'associació.



NEN(Nen, ...)  
 DATA(Dat)  
 VACUNA(Vac, ...)  
 VACUNACIÓ(**Nen**, **Dat**, Vac)  
 on {Nen} referencia NEN,  
 {Dat} referencia DATA i  
 {Vac} referencia VACUNA  
 Vac NOT NULL

## Associacions ternàries: molts-un-un

- En el cas molts-un-un la multiplicitat és \* al l'extrem 'molts' i 0..1 als extrems 'un' de l'associació.
- La relació té dues claus candidates. Cadascuna està formada pels atributs de les claus de dues de les classes associades una de les quals és necessàriament la del costat 'molts'.

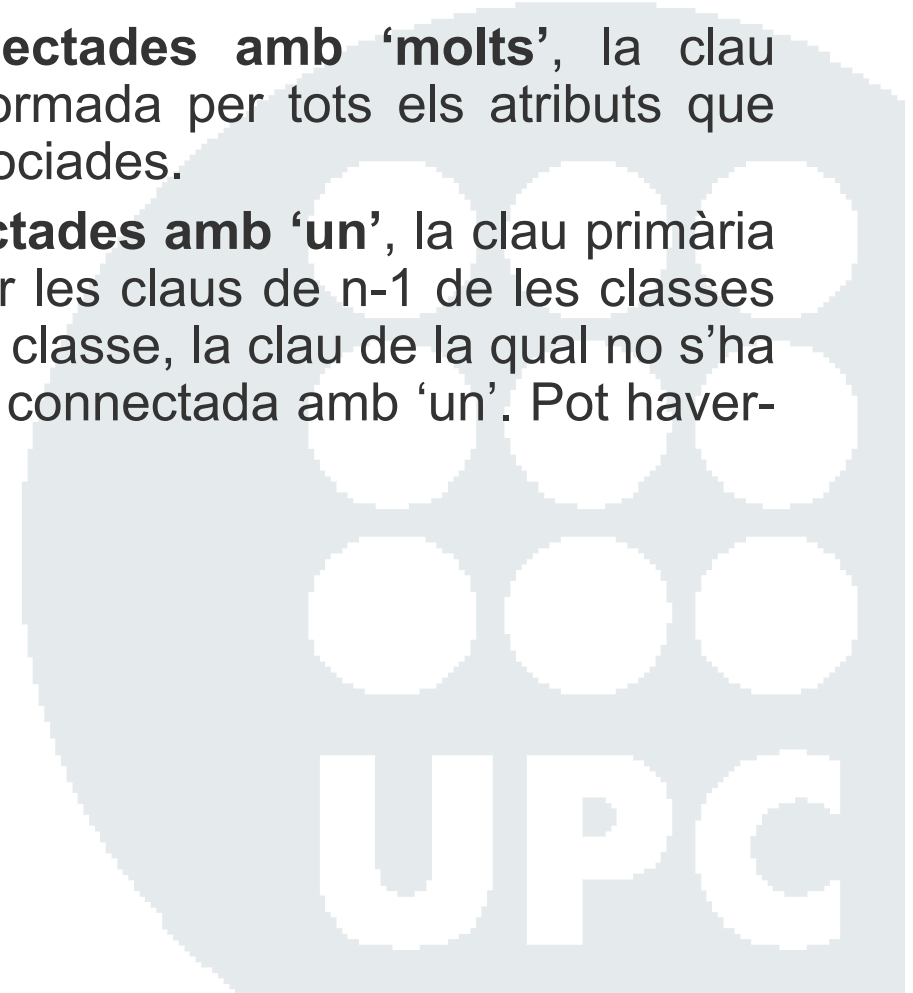


HORARI(Dia-set, Hora)  
 AULA(Codi-aula, ...)  
 GRUP(Gr, ...)  
 CLASSE (**Dia-set, Hora**, **Codi-aula**, Gr)  
 on {Dia-set, Hora} referencia HORARI,  
 {Codi-aula} referencia AULA i  
 {Gr} referencia GRUP  
 Gr NOT NULL  
 UNIQUE(Dia-set, Hora, Gr)

HORARI(Dia-set, Hora)  
 AULA(Codi-aula, ...)  
 GRUP(Gr, ...)  
 CLASSE (**Dia-set, Hora**, Codi-aula, **Gr**)  
 on {Dia-set, Hora} referencia HORARI,  
 {Codi-aula} referencia AULA i  
 {Gr} referencia GRUP  
 Codi-aula NOT NULL  
 UNIQUE(Dia-set, Hora, Codi-aula)

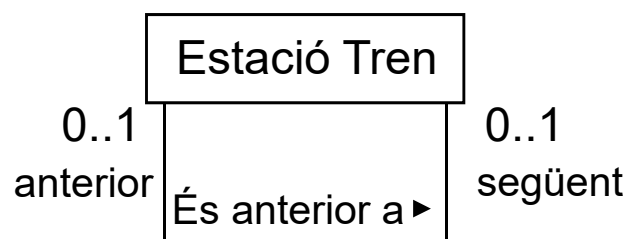
## Transformació general de les n-àries

- En tots els casos, la transformació d'una interrelació n-ària consistirà en una **nova relació** que conté tots els atributs que formen les claus de les n classes associades.
- **Si totes les classes estan connectades amb 'molts'**, la clau primària de la nova relació estarà formada per tots els atributs que formen les claus de les n classes associades.
- **Si una o més classes estan connectades amb 'un'**, la clau primària de la nova relació estarà formada per les claus de n-1 de les classes associades amb la condició de que la classe, la clau de la qual no s'ha inclòs, ha de ser una de les que està connectada amb 'un'. Pot haver-hi diverses claus candidates.



## Transformació de les associacions recursives (1)

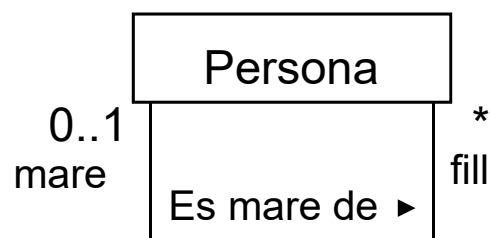
- Les **associacions recursives** es transformen de la mateixa manera que la resta d'associacions, és a dir, que cal tenir en compte si són binàries o n-àries i també la seva multiplicitat i aplicar les transformacions corresponents.
- Exemples:



ESTACIO (Línia,Nom, ..., LíniaEstSeg,NomEstSeg)  
 on {LíniaEstSeg,NomEstSeg} referencia ESTACIO  
 UNIQUE(LíniaEstSeg,NomEstSeg)

o bé

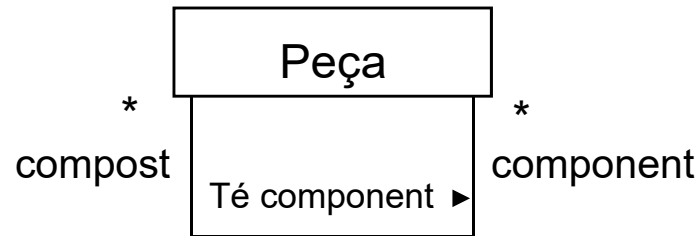
ESTACIO (Línia,Nom, ..., LíniaEstAnt,NomEstAnt)  
 on {LíniaEstAnt,NomEstAnt} referencia ESTACIO  
 UNIQUE(LíniaEstAnt,NomEstAnt)



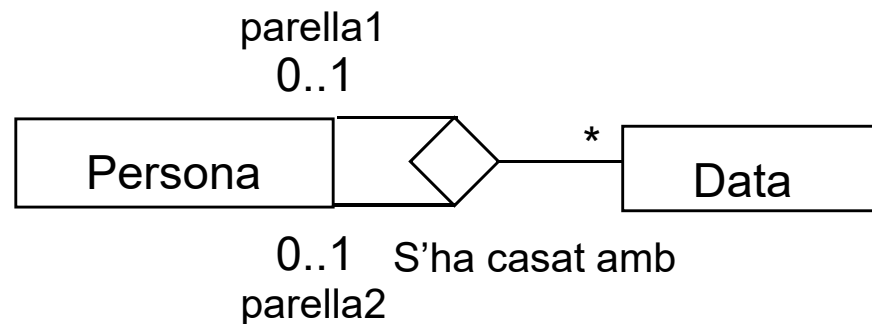
PERSONA(Codi-per, ..., Codi-mare)  
 on {Codi-mare} referencia PERSONA

## Transformació de les associacions recursives (2)

### ■ Exemples:



PEÇA(Codi-peça, ...)  
 COMPONENT(Codi-compost, Codi-component)  
 on {Codi-compost} referencia PEÇA  
 i {Codi-component} referencia PEÇA



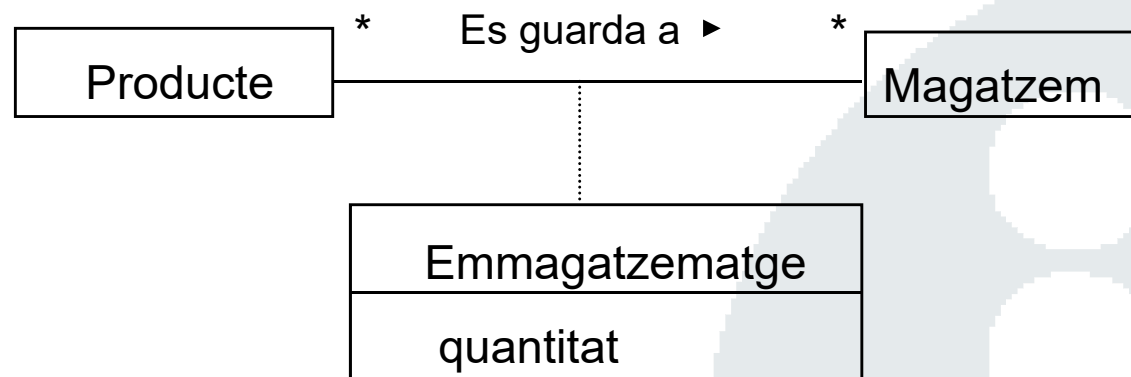
PERSONA(Codi-per, ...)  
 DATA(Dat)  
 CASAMENT(Codi-parella1, Codi-parella2, Dat)  
 on {Codi-parella1} referencia PERSONA,  
 {Codi-parella2} referencia PERSONA  
 {Dat} referencia DATA  
 Codi-parella2 NOT NULL  
 UNIQUE(Codi-parella2, Dat)

o bé

CASAMENT(Codi-parella1, Codi-parella2, Dat)  
 on {Codi-parella1} referencia PERSONA,  
 {Codi-parella2} referencia PERSONA  
 {Dat} referencia DATA  
 Codi-parella1 NOT NULL  
 UNIQUE(Codi-parella1, Dat)

## Transformació de les classes associatives (1)

- La transformació de la seva associació és, ahora, la transformació de la classe associativa.
- Si la classe associativa té atributs, aleshores s'afegeixen com a atributs de la relació corresponent a la seva transformació.
- Exemple:

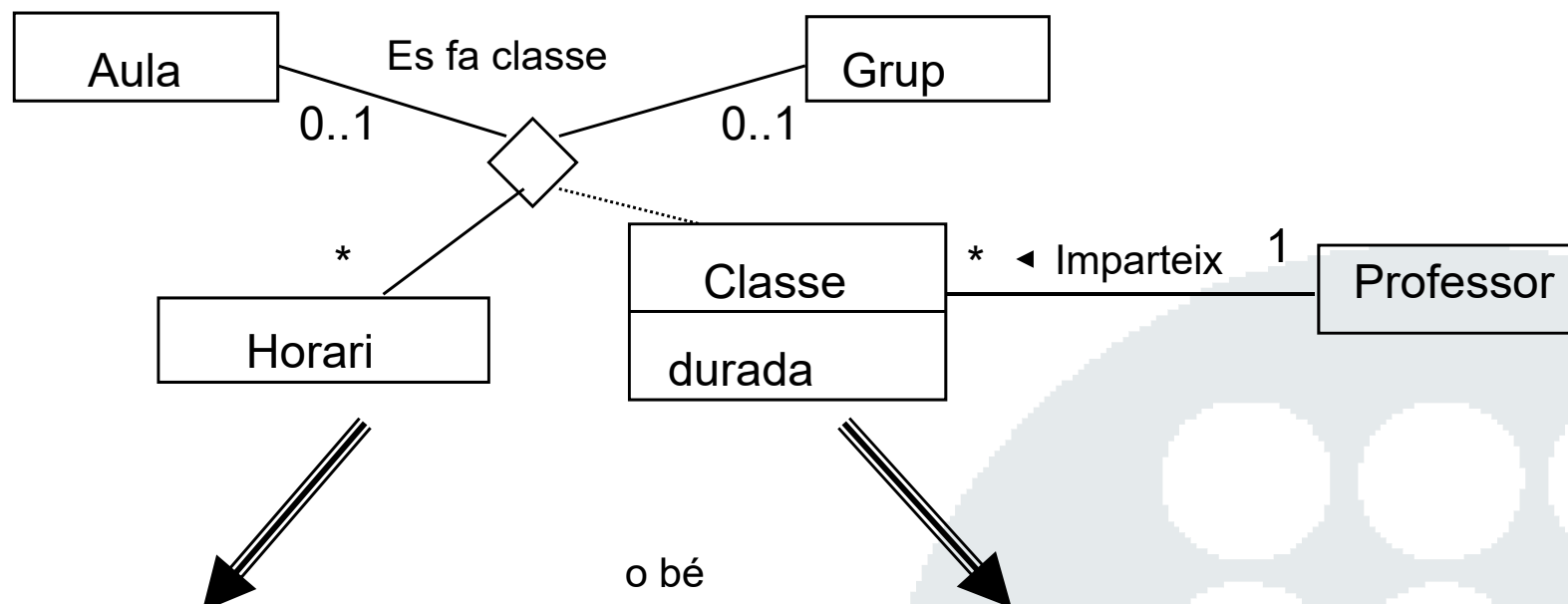


↓

PRODUCTE(Pro, ...)  
MAGATZEM(Mag, ...)  
EMMAGATZEMATGE(Pro, Mag, Quantitat)  
on {Pro} referencia PRODUCTE i  
{Mag} referencia MAGATZEM

## Transformació de les classes associatives (2)

### ■ Exemple:

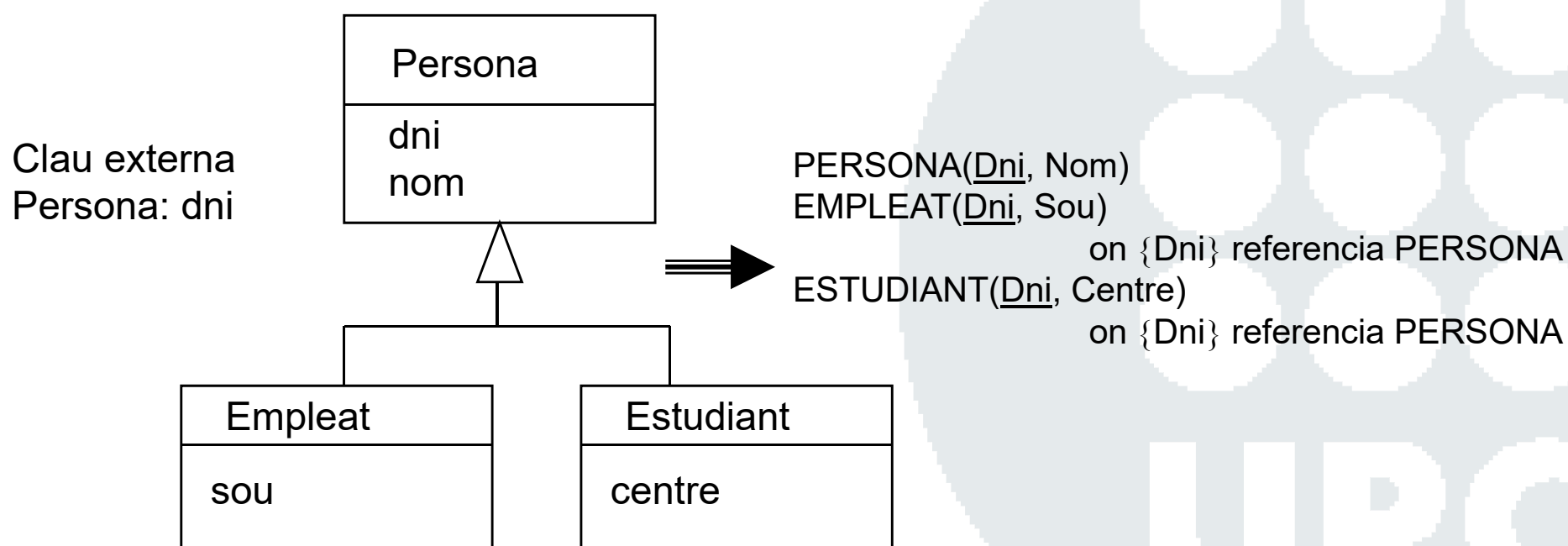


HORARI(Dia-set, Hora)  
 AULA(Codi-aula, ...)  
 GRUP(Gr, ...)  
 PROFESSOR(Prof, ...)  
 CLASSE (Dia-set, Hora, Codi-aula, Gr, Durada, Prof)  
   {Dia-set, Hora} referencia HORARI  
   {Codi-aula} referencia AULA  
   {Gr} referencia GRUP  
   Gr NOT NULL  
   UNIQUE(Dia-set, Hora, Gr)  
   {Prof} referencia Professor  
   Prof NOT NULL

HORARI(Dia-set, Hora)  
 AULA(Codi-aula, ...)  
 GRUP(Gr, ...)  
 PROFESSOR(Prof, ...)  
 CLASSE (Dia-set, Hora, Codi-aula, Gr, Durada, Prof)  
   {Dia-set, Hora} referencia HORARI  
   {Codi-aula} referencia AULA  
   {Gr} referencia GRUP  
   Codi-aula NOT NULL  
   UNIQUE(Dia-set, Hora, Codi-aula)  
   {Prof} referencia Professor  
   Prof NOT NULL

## Transformació de la generalització/especialització

- **Cadascuna de les classes d'objectes** (superclasse i subclasses) que formen part d'una generalització/especialització **es transforma en una relació**.
- La relació corresponent a la **superclasse** té com a clau primària la clau de superclasse i conté tots els atributs de la superclasse (atributs comuns).
- Les relacions corresponents a les **subclasses** tenen com a clau primària la clau de la superclasse i contenen els atributs específics de la subclasse.
- Exemple:





## Resum de les transformacions

Elements del model de partida	Transformació al model relacional
Classe d'objectes	Relació
Associació binària: un a molts	Clau forana al costat molts
Associació binària: un a un	Clau forana a qualsevol costat
Associació binària: molts a molts	Relació
Associació n-ària	Relació
Associació recursiva	Com les no recursives: Clau forana per binàries un a un i un a molts Relació per binàries molts a molts i n-àries
Classe associativa	La transformació de la seva associació és, ahora, la transformació de la classe associativa.
Generalització/especialització	Relació per la superclasse i per cadascuna de les subclasses

## Consideracions finals

- Les relacions corresponents a classes temporals com ara Data, Hora, Any, etc que no tenen atributs a part de la clau es poden eliminar.
- Quan hi ha diverses claus candidates per una relació, caldria prohibir les repeticions de valors per les claus alternatives (no primàries). En SQL es pot usar la clàusula UNIQUE.
- Restriccions que no es poden controlar mitjançant la definició de claus primàries i foranes de les relacions i que cal controlar amb altres mecanismes ('triggers', 'assertions', etc.).
  - Multiplicitats diferents de les considerades.
  - Restriccions de 'disjoint' i 'complete' que poden especificar-se en una generalització/especialització.
  - Restriccions textuais no expressables gràficament en UML poden requerir altres mecanismes de control (no és el cas de les restriccions de clau externa que es poden transformar en restriccions de clau primària de la relació corresponent).

