

# Model relacional i normalització

Isidre Guixà Miranda

**Adaptació de continguts:** Carlos Manuel Martí Hernández



# Índex

<b>Introducció</b>	<b>5</b>
<b>Resultats d'aprenentatge</b>	<b>7</b>
<b>1 Model relacional</b>	<b>9</b>
1.1 Estructuració de les dades	9
1.1.1 Domini	10
1.1.2 Esquema i extensió	11
1.1.3 Claus candidates, clau primària i claus alternatives	13
1.1.4 Claus foranes	15
1.1.5 Operacions amb relacions	17
1.2 Regles d'integritat	18
1.2.1 Unicitat de la clau primària	19
1.2.2 Entitat de la clau primària	20
1.2.3 Integritat referencial	21
1.2.4 Integritat del domini	25
1.3 Traducció del model Entitat-Relació al model relacional	26
1.3.1 Entitats	27
1.3.2 Interrelacions	28
1.3.3 Entitats febles	36
1.3.4 Generalització i especialització	37
1.3.5 Entitats associatives	38
<b>2 Normalització</b>	<b>39</b>
2.1 La relació universal	42
2.2 Dependències funcionals	43
2.3 Primera forma normal	45
2.4 Preservació d'informació i dependències en la normalització	46
2.5 Segona forma normal	47
2.6 Tercera forma normal	49
2.7 Forma normal de Boyce-Codd	51
2.8 Quarta forma normal	53
2.9 Cinquena forma normal	55
2.10 Desnormalització	58



## Introducció

Els sistemes gestors de bases de dades són grans aplicacions informàtiques pensades per facilitar la gestió de dades, i perquè aquesta gestió sigui eficient és necessari i convenient que les dades tinguin un disseny adequat. Això s'aconsegueix seguint adequadament alguns mètodes de disseny com ara el disseny Entitat-Relació.

Partim, doncs, del supòsit que tenim bases de dades ben dissenyades i ens correspon implementar-les en un sistema gestor de bases de dades. Al llarg de la història de la informàtica s'han succeït diversos models de sistemes gestors de bases de dades: jeràrquics, en xarxa i relacionals. La majoria dels sistemes gestors de bases de dades actuals es basen en el model relacional i, per tant, ens hem de centrar en el seu estudi.

En l'apartat "Model relacional", es veurà com s'estructuren les dades i quines regles té associades aquest model. A banda d'això, cal destacar l'apartat destinat a la transformació del model ER en el model relacional, ja que aquesta serà una tècnica imprescindible per al disseny lògic de BD.

En l'apartat "Normalització" s'introdueix un seguit de conceptes que formen la coneguda teoria de la normalització, la qual permet detectar errors en el disseny de la base de dades i facilita mecanismes per a la seva correcció. En principi, tot el model relacional derivat d'un model Entitat-Relació previ, correctament elaborat, hauria de ser també correcte. Però en ocasions això no és així, i aleshores, en l'explotació d'una base de dades es poden detectar problemes que fan necessari un estudi del disseny del model relacional, per si cal introduir modificacions en el mateix. Així mateix, hi ha dissenyadors que no volen modelitzar la realitat amb el model Entitat-Relació i volen dissenyar directament el model relacional. Cal dir, d'altra banda, que aquesta problemàtica és molt freqüent quan les bases de dades d'una certa complexitat han estat dissenyades directament utilitzant el model relacional, sense haver dissenyat prèviament cap model entitat-relació.

El coneixement tan del model Relacional com de la normalització de bases de dades és molt important per a qualsevol persona que analitzi, dissenyi o administri bases de dades relacionals, o que implementi aplicacions que interactuïn amb bases de dades

Evidentment, per tal d'adquirir uns coneixements òptims en aquesta matèria és imprescindible efectuar totes les activitats proposades, així com els exercicis d'autoavaluació.



## Resultats d'aprenentatge

En finalitzar aquesta unitat l'alumne/a:

1. Dissenya models lògics normalitzats interpretant diagrames entitat/relació.

- Identifica els principals elements del model relacional: relacions, atributs, domini dels atributs, diferents tipus de claus i cardinalitat de les relacions.
- Identifica i interpreta les regles d'integritat associades a cadascuna de les claus primàries.
- Identifica i interpreta les regles d'integritat associades a cadascuna de les claus foranies, tenint en compte les diferents possibilitats de modificar i/o esborrar (eliminació i/o modificació en cascada, restricció de l'eliminació i/o modificació, eliminació i/o modificació aplicant valors nuls als registres relacionats).
- Identifica les taules, camps i les relacions entre taules, d'un disseny lògic.
- Tradueix un model Entitat-Relació a model relacional aplicant les regles corresponents de traducció.
- Aplica les regles de normalització en el model relacional.
- Elabora la guia d'usuari i la documentació completa relativa al disseny físic (taules, atributs i relacions) de la base de dades relacional, de manera estructurada i clara; afegint les restriccions que no poden plasmar-se en el disseny lògic.





## 1. Model relacional

El model relacional és un model de dades basat en dues disciplines matemàtiques: la lògica de predicats i la teoria de conjunts.

Potser a causa d'aquest sòlid fonament teòric, que proporciona a aquest model una robustesa excepcional, els SGBD relacionals (o SGBDR) són actualment els que tenen una implantació més gran en el mercat.

El model relacional va ser proposat originàriament per Edgar Frank Codd en el seu treball *A Relational Model of Data for Large Shared Data Banks* ('Un model relacional de dades per a grans bancs de dades compartits') l'any 1970, tot i que no es va implementar comercialment fins al final de la dècada.

### E. F. Codd

Codd treballava per a IBM, però no va ser aquesta multinacional qui va creure abans en les possibilitats del model relacional, sinó més aviat la competència, i molt especialment *Oracle*, empresa que va néixer, justament, amb el nom de Relational Software.

### SGBD

Acrònim de Sistema Gestor de Bases de Dades. És un programari especialitzat en la gestió de bases de dades (enteses, aquestes, com un conjunt estructurat d'informació).

## 1.1 Estructuració de les dades

El **model relacional** permet construir **estructures de dades** per representar les diferents informacions del món real que tinguin algun interès.

Les estructures de dades construïdes seguint el model relacional estan formades per conjunts de relacions.

Les **relacions** poden ser concebudes com a **representacions tabulars** de les dades.

Cal precisar els extrems següents:

- Tota relació ha de tenir un nom que la identifiqui unívocament dins de la base de dades.
- Cada fila està constituïda per un tuple de dades relacionades entre elles, anomenat també *registre*, que guarda les dades que ens interessa reflectir d'un objecte concret del món real.
- En canvi, cada columna conté, en cada cel·la, dades d'un mateix tipus, i se la pot anomenar *atribut* o *camp*.

### Tuple

En l'àmbit de les BD, podem definir *tuple* com una seqüència finita d'objectes que comprèn les diferents associacions entre cada atribut de la relació i un valor concret, admissible dins del domini respectiu.

- Cada cel·la, o intersecció entre fila i columna, pot emmagatzemar un únic valor.

### Exemple de relació

La taula 1.1 reflecteix l'estructuració tabular de la relació ALUMNE, que conté les dades personals corresponents als individus matriculats en un centre docent.

Cada fila conté unes quantes dades relacionades que, en aquest cas, són les que pertanyen a un mateix alumne.



La relació té un nom (ALUMNE), com cadascuna de les columnes (DNI, Nom, Cognoms i Telefon). Si aquests noms són prou significatius, permeten copsar de seguida el sentit que tenen els valors de les dades emmagatzemades en la relació.

TAULA 1.1. Exemple de relació

ALUMNE			
DNI	Nom	Cognoms	Telefon
47126654F	Josep	Bel Rovira	453641282
51354897S	Anna	Pacheco Cuscó	723352151
56354981L	Xavier	Rius Montalvo	726922235

Tota base de dades relacional està formada per un conjunt de relacions.

Aquesta senzilla manera de visualitzar l'estructura de les bases de dades relacionals resulta molt entenedora per a la majoria d'usuaris. Però cal aprofundir en algunes característiques addicionals de les relacions, per talde poder-les distingir clarament dels fitxers tradicionals.

### 1.1.1 Domini

Pel que fa al model relacional, un **domini** consisteix en un conjunt finit de valors indivisibles.

Els atributs només poden prendre els valors que estiguin inclosos dins del domini respectiu. Altrament no són valors vàlids, i un SGBD relacional no en pot permetre l'emmagatzematge.

#### Exemples de dominis

Examinem l'atribut **Telefon** de la relació ALUMNE. Si el definim de tal manera que només pugui emmagatzemar nou caràcters (perquè els telèfons sempre consten de nou dígits) de tipus numèric (ja que les lletres no poden formar part d'un número de telèfon), el domini d'aquest atribut inclourà totes les combinacions possibles (en concret,  $10^9$ , que és una magnitud gran, però finita).

Una altra cosa és que molts d'aquests valors no es podran correspondre mai amb valors existents en el món real (per exemple, difícilment un operador assignarà a un dels seus

abonats una cadena de nou zeros com a identificador telefònic). Per aconseguir-ho, caldria restringir força més el domini de l'atribut a l'hora de definir-lo.

Centrem-nos ara en l'atribut **Cognoms**. Contindrà els valors dels dos cognoms dels alumnes que els tinguin, separats per un espai en blanc. Per tant, aquest camp està definit per tal que pugui emmagatzemar dos objectes del món real: primer cognom i segon cognom.

Conceptualment, els usuaris podran distingir entre els dos objectes representats, i els programadors d'aplicacions podran truncar, en cas necessari, el resultat obtingut en fer una consulta del camp Cognoms. Però tot SGBD relacional considerarà el valor contingut en l'atribut Cognoms de manera atòmica, sense cap estructuració interna.

Hem de considerar **dues tipologies de dominis**:

- **Dominis predefinitos.** Són els tipus de dades que admeti cada SGBD, com, per exemple (esmentats de manera genèrica, ja que hi ha moltes especificitats en funció dels diferents sistemes gestors), les cadenes de caràcters, els nombres enters, els nombres decimals, les dades de caire cronològic, etc.
- **Dominis definits pels usuaris.** Consisteixen en restriccions addicionals aplicades sobre el domini predefinit d'alguns atributs, establertes pels dissenyadors i pels administradors de bases de dades.

#### Exemple de domini definit per l'usuari

En una relació per emmagatzemar les dades dels aspirants a mosso d'esquadra, es podria establir el camp IMC, per registrar els índexs de massa corporal respectius.

Doncs bé, es podria restringir el domini d'aquest camp de tal manera que no admetés aspirants amb valors inferiors a dinou ni superiors a trenta, ja que la normativa no ho permet.

### 1.1.2 Esquema i extensió

Tota relació consta d'un esquema (també anomenat *intensió de la relació*) i de la seva extensió.

→ **L'esquema d'una relació** consisteix en un nom que la identifica unívocament dins de la base de dades, i en el conjunt d'atributs que aquella conté.

És molt recomanable, per tal d'evitar confusions en la implementació ulterior, seguir uniformement una notació concreta a l'hora d'expressar els esquemes de les relacions que formen una mateixa base de dades.

A continuació, es detallen les característiques d'un dels sistemes de notació més freqüents:

- Cal escriure el nom de les relacions amb majúscules i preferiblement en singular.

- S'ha d'escriure el nom dels atributs començant amb majúscula i continuant amb minúscules, sempre que no es tracti de sigles, ja que aleshores és més convenient deixar totes les lletres amb majúscules (com ara DNI). Per tal de fer els noms compostos més llegidors, es pot encapçalar cada paraula de les que formen el nom del camp amb una lletra majúscula (per exemple: DataNaixement, TelefonParticular, etc.).

#### Exemple d'esquema d'una relació

L'esquema de la relació que es mostra en la taula 1.2, conforme al sistema de notació proposat, quedaria com segueix:

→ ALUMNE(DNI, Nom, Cognoms, Telefon)

Cal precisar que l'ordre en què ens mostrin els atributs és indiferent, per definició del model relacional.

TAULA 1.2. Exemple de relació

ALUMNE			
DNI	Nom	Cognoms	Telefon
47126654F	Josep	Bel Rovira	453641282
51354897S	Anna	Pacheco Cuscó	723352151
56354981L	Xavier	Rius Montalvo	726922235

Els atributs d'una relació són únics dins d'aquesta. El seu nom no pot estar repetit dins d'una mateixa relació. Ara bé, diferents relacions sí que poden contenir atributs amb el mateix nom.

D'altra banda, cal dir que els dominis de diferents atributs d'una mateixa relació poden ser idèntics, malgrat que els camps respectius emmagatzemin els valors de diferents propietats de l'objecte (per exemple, seria perfectament lògic que els atributs TelefonFix, TelefonMobil i TelefonFeina, tot i pertànyer a una mateixa relació, tinguessin el mateix domini).

→ L'extensió d'una relació consisteix en els valors de les dades emmagatzemades en tots els tuples que aquesta conté.

#### Exemple d'extensió

Si prenem com a base, una vegada més, la relació amb esquema ALUMNE(DNI, Nom, Cognoms, Telefon) de la taula 2, la seva extensió seria una llista en què figurarien tots els alumnes de la base de dades:

Alumne 1: 47126654F, Josep, Bel Rovira, 453641282 Alumne 2: 51354897S, Anna, Pacheco Cuscó, 723352151 Alumne 3: 56354981L, Xavier, Rius Montalvo, 726922235

De vegades, els atributs de les relacions poden no contenir cap valor o, dit d'una altra manera, poden contenir valors nuls.

### Exemple de valor nul

Imaginem que s'hi matricula un quart alumne que no té telèfon. Les seves dades en la coneguda relació amb esquema ALUMNE(DNI, Nom, Cognoms, Telefon) reflectiran aquesta circumstància amb l'absència de valor en l'atribut Telefon del tuple que li correspongui.

En utilitzar representacions tabulars per visualitzar els valors de les extensions de les relacions (en el pla teòric, no en implementacions reals amb SGBD), per tal d'indicar que una cel·la té valor nul s'hi pot incloure el mot NUL (com en la taula 1.3), o bé es pot deixar en blanc, simplement.

TAULA 1.3. Exemple de relació amb valors nuls

ALUMNE			
DNI	Nom	Cognoms	Telefon
47126654F	Josep	Bel Rovira	453641282
51354897S	Anna	Pacheco Cuscó	723352151
56354981L	Xavier	Rius Montalvo	726922235
24583215W	Mariona	Castellví Mur	NUL

El **grau d'una relació** depèn del nombre d'atributs que inclou el seu esquema.

### Exemple de grau d'una relació

La relació amb esquema ALUMNE(DNI, Nom, Cognoms, Telefon) de la taula 3 és de grau 4, perquè té quatre atributs.

La **cardinalitat d'una relació** ve donada pel nombre de tuples que en formen l'extensió.

### Exemple de cardinalitat

Si ens fixem en la taula 1.3, la cardinalitat de la relació ALUMNE és 4, perquè la seva extensió conté quatre tuples corresponents als quatre alumnes que, de moment, hi ha matriculats.

## 1.1.3 Claus candidates, clau primària i claus alternatives

Per tal de resultar útil, l'emmagatzematge de la informació ha de permetre la identificació de les dades. En l'àmbit de les bases de dades relacionals, els tuples de les relacions s'identifiquen mitjançant les anomenades *superclaus*.

Una **superclau** és un subconjunt dels atributs que formen l'esquema d'una relació tal que no és possible que hi hagi més d'un tuple en l'extensió respectiva, amb la mateixa combinació de valors en els atributs que formen part del subconjunt esmentat.

Però una superclau pot contenir atributs innecessaris, que no contribueixen a la identificació inequívoca dels diferents tuples. El que habitualment interessa és treballar amb **superclaus mínimes**, tals que cap subconjunt propi sigui capaç per sí sol d'identificar els tuples de la relació.

Per definició, cap superclau mínima no pot admetre valors nuls en cap dels seus atributs, perquè si ho fes, no podria garantir la identificació inequívoca dels tuples que continguessin algun valor nul en alguns dels atributs de la superclau mínima en qüestió.

D'altra banda, cal dir que en una mateixa relació pot passar que hi hagi més d'una superclau mínima que permeti distingir els tuples unívocament entre ells.

S'anomenen **claus candidates** totes les superclaus mínimes d'una relació formades pels atributs o conjunts d'atributs que permeten identificar els tuples que conté la seva extensió.

#### Tria de la clau primària

Amb molta freqüència, és l'administrador de la BD qui tria la clau primària de la relació, d'entre les claus candidates disponibles, tot realitzant, en el fons, tasques de dissenyador lògic.

Però, a l'hora d'implementar una BD, entre totes les claus candidates de cada relació només se n'ha de triar una.

Quan parlem de **clau primària** ens referim a la clau que, finalment, el dissenyador lògic de la base de dades tria per distingir unívocament cada tuple d'una relació de la resta.

Aleshores, les **claus candidates no triades** com a clau primària resten presents en la relació.

Quan una relació ja té establerta una clau primària, la resta de claus presents en aquella, i que també podrien servir per identificar els diferents tuples de l'extensió respectiva, es coneixen com a **claus alternatives**.

Una forma de diferenciar els atributs que formen la clau primària de les relacions, dels altres atributs de l'esquema respectiu, és posar-los subratllats. Per aquest motiu, normalment es col·loquen junts i abans que la resta d'atributs, dins de l'esquema. Però només es tracta d'una qüestió d'elegància, ja que el model relacional no es basa ni en l'ordre dels atributs de l'esquema, ni tampoc en l'ordre dels tuples de l'extensió de la relació.

#### Exemples de claus candidates, primària i alternatives

Observant la taula 1.4, podem imaginar que la relació ALUMNE té uns quants atributs més, de manera que el seu esquema queda com segueix:

ALUMNE(DNI, NumSS, NumMatricula, Nom, Cognoms, Telefon)

Veurem fàcilment com els atributs DNI, NumSS (número de la Seguretat Social) i NumMatricula, en ser personals i irrepetibles, ens podrien servir per identificar unívocament els alumnes. Per tant, serien claus candidates.

Aleshores, el dissenyador de BD s'haurà de decidir per una clau candidata com a clau primària. Si, per exemple, tria DNI com a clau primària, les antigues claus candidates restants es passaran a considerar claus alternatives.

En aquest cas, doncs, l'esquema resultant haurà de reflectir quina és la clau primària de la relació, tot subratllant l'atribut DNI:

→ ALUMNE(DNI, NumSS, NumMatricula, Nom, Cognoms, Telefon)

**TAULA 1.4.** Exemple de relació amb valors nuls

ALUMNE			
DNI	Nom	Cognoms	Telefon
47126654F	Josep	Bel Rovira	453641282
51354897S	Anna	Pacheco Cuscó	723352151
56354981L	Xavier	Rius Montalvo	726922235
24583215W	Mariona	Castellví Mur	NUL

Si no es disposa d'un atribut que sigui capaç d'identificar els tuples de la relació per si sol, cal buscar un subconjunt d'atributs, tals que la combinació dels valors que adoptin no es pugui repetir. Si aquesta possibilitat no existeix, cal afegir a la relació un atribut addicional que faci d'identificador.

Per definició, el model relacional no admet tuples repetits, és a dir, no permet l'existència de tuples en una mateixa relació que tinguin els mateixos valors en cadascun dels atributs.

Ara bé, les implementacions concretes dels diferents SGBD sí que permeten aquesta possibilitat, sempre que no s'estableixi cap clau primària en la relació amb tuples repetits.

Aquesta permissivitat de vegades permet solucionar certes eventualitats, però no hauria de ser la manera habitual de treballar amb BD relacionals.

### 1.1.4 Claus foranes

Qualsevol BD relacional, per petita que sigui, conté normalment més d'una relació. Per tal de reflectir correctament els vincles existents entre alguns objectes del món real, cal que els tuples de les diferents relacions d'una base de dades es puguin interrelacionar.

De vegades, fins i tot pot ser necessari relacionar els tuples d'una relació amb altres tuples de la mateixa relació.

El mecanisme que ofereixen les BD relacionals per interrelacionar les relacions que contenen es fonamenta en les anomenades *claus foranes*.



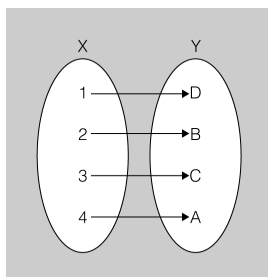
Una **clau forana** està constituïda per un atribut, o per un conjunt d'atributs, de l'esquema d'una relació, que serveix per relacionar els seus tuples amb els tuples d'una altra relació de la base de dades (o amb els tuples d'ella mateixa, en alguns casos).

Per tal d'aconseguir connectar els tuples d'una relació amb els d'una altra (o amb les seves pròpies), la clau forana utilitzada ha de referenciar la clau primària de la relació amb la qual es vol relacionar.

Les diferents combinacions de valors dels atributs de tota clau forana han d'existir en la clau primària a què fan referència, o bé han de ser valors nuls. Altrament, les referències serien errònies i, per tant, les dades serien incorrectes.

Cal parar atenció en les característiques següents de les claus foranes:

- Tota clau forana ha de tenir el mateix nombre d'atributs que la clau primària a la qual fa referència.
- Entre els atributs de l'esquema d'una clau forana i els de la clau primària respectiva s'ha de poder establir una correspondència (concretament, una bijecció).
- Els dominis dels atributs de tota clau forana han de coincidir amb els dominis dels atributs de la clau primària respectiva (o, com a mínim, cal que siguin compatibles dins d'un cert rang).



Exemple de bijecció

Una relació pot contenir més d'una clau forana, o bé no contenir-ne cap. I, en sentit invers, la clau primària d'una relació pot estar referenciada per una o més claus foranes, o bé pot no estar referenciada per cap.

Finalment, cal dir que es pot donar el cas que un mateix atribut formi part tant de la clau primària de la relació com d'alguna de les seves claus foranes.

### Exemples de claus foranes

La relació ALUMNE, tal com es mostra en la taula 1.5, incorpora dues claus foranes.

Una d'elles, CodiAula, fa referència a la clau primària de la relació AULA (formada per l'atribut Codi), exposada en la taula 1.6, per tal d'indicar quina aula correspon a cada alumne.



En canvi, DNIDelegat fa referència a la clau primària de la mateixa relació (formada per l'atribut DNI), i serveix per indicar quin és el delegat que representa cada alumne.

Fixem-nos que l'alumna Mariona Castellví encara no té assignat ni delegat ni aula i, per aquest motiu, el tuple que la representa conté, de moment, valors nuls en els atributs de les dues claus foranes.



**TAULA 1.5.** Exemple de relació amb claus foranes

ALUMNE					
DNI	Nom	Cognoms	Telefon	DNIDelegat	CodiAula
47126654F	Josep	Bel Rovira	453641282	47126654F	102
51354897S	Anna	Pacheco Cuscó	723352151	51354897S	201
56354981L	Xavier	Rius Montalvo	726922235	51354897S	201
24583215W	Mariona	Castellví Mur	NUL	NUL	NUL

**TAULA 1.6.** Exemple de relació amb clau primària referenciada

AULA	
Codi	Capacitat
101	40
102	36
201	30

La notació més habitual per designar les claus foranes de les relacions consisteix a afegir aquesta circumstància a continuació del seu esquema, incloent-hi entre dues claus el conjunt d'atributs que formen la clau forana de què es tracta, precedit de l'adverbi ON, i seguit de la forma verbal REFERENCIA i de la relació a què fa referència. Si hi ha més d'una clau forana, se separen per comes i l'última ha d'anar precedida de la conjunció I.

O bé posar: CF:{atribut} → Taula a qui fa referència

**Exemple de notació per designar claus foranes**

Les dues relacions que es mostren en les taules 5 i 6 s'expressaran de la manera següent:

ALUMNE(DNI, Nom, Cognoms, Telefon, DNIDelegat, CodiAula) ON {DNIDelegat} REFERENCIA ALUMNE i {CodiAula} REFERENCIA AULA

AULA(Codi, Capacitat)

### 1.1.5 Operacions amb relacions

El model relacional permet realitzar una sèrie d'operacions amb les dades emmagatzemades en les BD, les quals tenen diferents finalitats:

- **Actualització.** Aquestes operacions realitzen canvis en els tuples que queden reflectits en les relacions que contenen les BD. Poden ser de tres tipus:
  - **Inserció.** Consisteix a afegir un o més tuples nous a una relació determinada.
  - **Esborrat.** Consisteix a eliminar un o més tuples nous d'una relació determinada.
  - **Modificació.** Consisteix a canviar el valor d'un o més atributs d'un o més tuples d'una relació determinada.

- **Consulta.** Aquestes operacions només fan possible l'obtenció parametritzada de dades, sense que es vegin alterades les emmagatzemades en la BD.

La realització d'aquestes operacions comporta el coneixement previ de l'estructura formada per les relacions que sigui necessari utilitzar, és a dir, els esquemes de les relacions i les interrelacions entre elles, mitjançant les claus foranes.

### Exemples d'operacions

Si prenem en consideració la relació ALUMNE que mostra la taula 1.7, un exemple d'inserció consistirà a afegir un nou alumne, com ara el següent:

```
<65618724G, Lúdia, Bofarull Mora, 564628231, 47126654F, 102>
```

Un exemple d'esborrament seria eliminar el tuple que conté les dades d'un alumne donat d'alta, com ara el següent:

```
<56354981L, Xavier, Rius Montalvo, 726922235, 51354897S, 201>
```

Un exemple de modificació seria, per exemple, canviar el número de telèfon de Josep Bel Rovira que consta en la BD (453641282) per un altre (546022547), per tal de reflectir correctament la realitat, de manera actualitzada, o bé assignar-ne un de nou a algú que abans no en tenia, com la Mariona Castellví Mur, introduint 875261473 en lloc de l'anterior valor nul.

I, com a exemple de consulta, ens podria interessar obtenir una llista, ordenada alfabèticament pels cognoms, de tots els alumnes que són delegats d'aula i que, per tant, tenen valors coincidents en l'atribut DNI i en l'atribut DNIDelegat. En aquest cas, el resultat seria el següent:

```
1 <47126654F, Josep, Bel Rovira, 546022547, 47126654F, 102>
2 <51354897S, Anna, Pacheco Cuscó, 723352151, 51354897S, 201>
```

**TAULA 1.7.** Exemple de relació amb claus foranes

ALUMNE					
DNI	Nom	Cognoms	Telefon	DNIDelegat	CodiAula
47126654F	Josep	Bel Rovira	453641282	47126654F	102
51354897S	Anna	Pacheco Cuscó	723352151	51354897S	201
56354981L	Xavier	Rius Montalvo	726922235	51354897S	201
24583215W	Mariona	Castellví Mur	NUL	NUL	NUL

## 1.2 Regles d'integritat

Els valors que emmagatzemen les BD han de reflectir en tot moment, de manera correcta, la porció de la realitat que volem modelitzar.

Anomenem **integritat** la propietat de les dades que consisteix a representar correctament les situacions del món real que modelitzen.

Per tal que les dades siguin íntegres, cal garantir que siguin correctes, i també que estiguin senceres.

En atenció a l'objectiu esmentat, doncs, les dades han de complir certes condicions, que podem agrupar en dues tipologies diferents:

- **Restriccions d'integritat de l'usuari.** Són condicions específiques de cada BD. Els SGBD han de permetre als administradors establir certes restriccions aplicables a casos concrets, i han de garantir que es respectin durant l'explotació habitual del sistema.
- **Regles d'integritat del model.** Són condicions de caire general que han de complir totes les BD que segueixin el model relacional. No cal definir-les en implementar cada BD, perquè es consideren preestablertes.

#### Exemple de restricció d'integritat de l'usuari

En donar d'alta un nou alumne de la relació ALUMNE, que es mostra en la taula 8, podríem exigir al sistema que el validés, mitjançant l'algorisme corresponent, si la lletra introduïda del NIF es correspon amb les xifres introduïdes prèviament, i que denegué la inserció en cas contrari, per tal de no emmagatzemar una situació en principi no admissible en el món real.

#### Exemple de regla d'integritat del model

Com que la relació ALUMNE, que es mostra en la taula 1.8, té definit l'atribut DNI com a clau primària, el sistema validarà automàticament que no s'introdueixi més d'un alumne amb el mateix carnet d'identitat, ja que aleshores la clau primària no compliria el seu objectiu de garantir la identificació inequívoca de cada tuple, diferenciant-lo de la resta.

TAULA 1.8. Exemple de relació amb claus foranes

ALUMNE					
DNI	Nom	Cognoms	Telefon	DNIDelegat	CodiAula
47126654F	Josep	Bel Rovira	453641282	47126654F	102
51354897S	Anna	Pacheco Cuscó	723352151	51354897S	201
56354981L	Xavier	Rius Montalvo	726922235	51354897S	201
24583215W	Mariona	Castellví Mur	NUL	NUL	NUL

### 1.2.1 Unicitat de la clau primària

El valor d'una clau primària, globalment considerada, no pot estar repetit en més d'un tuple de la mateixa relació, ja que aleshores la clau primària no estaria en condicions d'assegurar la identificació inequívoca dels diferents tuples.



En cap moment, no hi pot haver dos o més tuples amb la mateixa combinació de valors en el conjunt dels atributs que formen la clau primària d'una relació.

Els SGBD relacionals han de garantir la regla d'unicitat de la clau primària en totes les insercions de nous tuples, i també en totes les modificacions que afectin el valor d'algun dels atributs que formen part de la clau primària.

#### Exemple d'unicitat de la clau primària

En la relació AULA, que es mostra en la taula 1.9, no s'hauria de poder inserir un nou tuple amb els valors <102, 40>, perquè la clau primària ja emmagatzema el valor 102, corresponent a un altre tuple.



Si volem donar d'alta una altra aula al primer pis de l'edifici amb capacitat per a quaranta alumnes, haurem d'utilitzar com a clau primària un altre valor no present en l'atribut Codi, i resultarà, per exemple, <103, 40>.

Tampoc no hauria de ser possible modificar la clau del tuple <101, 40> i assignar-li el valor 102, perquè aquest valor ja el té assignat la clau primària d'un altre tuple.

TAULA 1.9. Exemple de relació amb clau primària referenciada

AULA	
Codi	Capacitat
101	40
102	36
201	30

### 1.2.2 Entitat de la clau primària

Les claus primàries serveixen per diferenciar cada tuple d'una relació de la resta de tuples de la mateixa relació. Per garantir la consecució d'aquesta finalitat, cal que els atributs que formen part d'una clau primària no puguin tenir valor nul ja que, si s'admetés aquesta possibilitat, els tuples amb valors nuls en la clau primària no es podrien distingir d'alguns altres.

Cap atribut que formi part d'una clau primària no pot contenir mai valors nuls en cap tuple.

Els SGBD relacionals han de garantir la regla d'entitat de la clau primària en totes les insercions de nous tuples, com també en totes les modificacions que afectin el valor d'algun dels atributs que formen part de la clau primària.

#### Exemple d'entitat de la clau primària

En la relació AULA, que es mostra en la taula 1.10, no s'hauria de poder inserir un nou tuple amb els valors <NUL, 26>, perquè la clau primària, per definició, no pot contenir valors nuls.

Si volem donar d'alta una altra aula amb capacitat per a vint-i-sis alumnes, haurem d'utilitzar com a clau primària un altre valor no nul, i resultarà, per exemple, <202, 26>.

Tampoc no hauria de ser possible, per la mateixa raó que hem exposat més amunt, modificar la clau del tuple <101, 40> i assignar-li el valor nul.

**TAULA 1.10.** Exemple de relació amb clau primària referenciada

AULA	
Codi	Capacitat
101	40
102	36
201	30

### 1.2.3 Integritat referencial

El model relacional no admet, per definició, que la combinació de valors dels atributs que formen una clau forana no sigui present en la clau primària corresponent, ja que això implicaria una connexió incorrecta.

La **integritat referencial** implica que, per a qualsevol tuple, la combinació de valors que adopta el conjunt dels atributs que formen la clau forana de la relació o bé ha de ser present en la clau primària a la qual fa referència, o bé ha d'estar constituïda exclusivament per valors nuls (si els atributs implicats admeten aquesta possibilitat, i així s'ha estipulat en definir-ne les propietats).

Els SGBD relacionals hauran de fer les comprovacions pertinents, de manera automàtica, per tal de garantir la integritat referencial, quan es produeixin dos tipus d'operacions amb relacions que tinguin claus foranes:

- Insercions de nous tuples.
- Modificacions que afectin atributs que formin part de qualsevol clau forana.

D'altra banda, els SGBD relacionals també hauran de validar la correcció d'uns altres dos tipus d'operacions amb relacions que tinguin la clau primària referenciada des d'alguna clau forana:

- Esborraments de tuples.
- Modificacions que afectin atributs que formin part de la clau primària.

Per tal de garantir la integritat referencial en aquests dos últims tipus d'operació, es pot seguir alguna de les tres polítiques següents: restricció, actualització en cascada i anul·lació.

#### Exemple de violació de la integritat referencial

Continuem especulant amb les relacions ALUMNE i AULA (reflectides en la taula 1.11 i taula 1.12 respectivament).

El tuple que conté les dades de Mariona Castellví Mur té un valor nul en l'atribut que forma la clau forana que fa referència a la relació AULA.

Si el volguéssim actualitzar amb el valor 316, per exemple, el sistema no ens ho hauria de deixar fer, perquè aquest valor no és present en la clau primària de cap tuple de la relació AULA i, per tant, aquesta operació contravindria la regla d'integritat referencial.

**TAULA 1.11.** Relació amb claus foranes

ALUMNE					
DNI	Nom	Cognoms	Telefon	DNIDelegat	CodiAula
47126654F	Josep	Bel Rovira	453641282	47126654F	102
51354897S	Anna	Pacheco Cuscó	723352151	51354897S	201
56354981L	Xavier	Rius Montalvo	726922235	51354897S	201
24583215W	Mariona	Castellví Mur	NUL	NUL	NUL

**TAULA 1.12.** Relació amb clau primària referenciada

AULA	
Codi	Capacitat
101	40
102	36
201	30

### Política de restricció

La política de restricció consisteix a prohibir l'operació d'actualització de què es tracti:

- En cas d'esborrament, no permetrà eliminar un tuple si la seva clau primària està referenciada des d'alguna clau forana.
- En cas de modificació, no permetrà alterar el valor de cap dels atributs que formen la clau primària d'un tuple, si aquesta està referenciada des d'alguna clau forana.

### Exemples de restriccions

Considerem una vegada més les relacions ALUMNE i AULA que es mostren en la taula 1.13 i taula 1.14 respectivament.

Aplicant la restricció tant en cas d'esborrament com de modificació, aquestes operacions no seran possibles amb l'aula 102 de la relació AULA perquè hi ha alumnes matriculats que han d'assistir a classe dins d'aquest espai i, per tant, la referencien des de la clau forana dels tuples que els representen. Sí seria possible, en canvi, esborrar l'aula 101, perquè no està referenciada des de la relació ALUMNE.

**TAULA 1.13.** Relació amb claus foranes

ALUMNE						
DNI		Nom	Cognoms	Telefon	DNIDelegat	CodiAula
47126654F	Josep	Bel Rovira	453641282	47126654F	102	
51354897S	Anna	Pacheco Cuscó	723352151	51354897S	201	
56354981L	Xavier	Rius Montalvo	726922235	51354897S	201	
24583215W	Mariona	Castellví Mur	NUL	NUL	NUL	

**TAULA 1.14.** Relació amb clau primària referenciada

AULA	
Codi	Capacitat
101	40
102	36
201	30

### Actualització en cascada

La política d'actualització en cascada consisteix a permetre l'operació d'actualització de què es tracti sobre un tuple determinat, però disposant al mateix temps una sèrie d'operacions compensatòries que propaguin en cascada les actualitzacions necessàries per tal que es mantingui la integritat referencial dels tuples que referencien, des dels atributs que en formen la clau forana, el tuple objecte d'actualització:

- En cas d'esborrament, s'eliminaran tots els tuples que facin referència al tuple esborrat.
- En cas de modificació, els valors dels atributs que formin part de la clau forana dels tuples que facin referència al tuple modificat s'alteraran per tal de continuar coincidint amb els nous valors de la clau primària del tuple al qual fan referència.

#### Exemples d'actualització en cascada

Tornem a prendre com a punt de partida dels exemples les relacions ALUMNE i AULA que es mostren en la taula 1.15 i taula 1.16 respectivament.

Si apliquem l'actualització en cascada tot esborrant el tuple <201, 30> de la relació AULA, també s'esborraran els dos tuples de la relació ALUMNE que hi fan referència des de la clau forana respectiva (CodiAula).

En canvi, si apliquem l'actualització en cascada tot modificant el tuple <201, 30> de la relació AULA, canviant el valor de la seva clau primària per un altre, com ara 203, els dos tuples de la relació ALUMNE que hi fan referència actualitzaran en cascada el valor de l'atribut CodiAula de 201 a 203, per tal de mantenir la connexió correcta entre els tuples d'ambdues relacions.

**TAULA 1.15.** Relació amb claus foranes

ALUMNE					
DNI	Nom	Cognoms	Telefon	DNIDelegat	CodiAula
47126654F	Josep	Bel Rovira	453641282	47126654F	102
51354897S	Anna	Pacheco Cuscó	723352151	51354897S	201
56354981L	Xavier	Rius Montalvo	726922235	51354897S	201
24583215W	Mariona	Castellví Mur	NUL	NUL	NUL

**TAULA 1.16.** Relació amb clau primària referenciada

AULA	
Codi	Capacitat
101	40
102	36
201	30

### Política d'anul·lació

La política d'anul·lació consisteix a permetre l'operació d'actualització de què es tracti en un tuple determinat, però disposant al mateix temps una sèrie d'operacions compensatòries que posin valors nuls en tots els atributs que formin part de les claus foranes dels tuples que facin referència al tuple objecte d'actualització:

- En cas d'esborrament, els atributs de la clau forana dels tuples que facin referència al tuple esborrat passaran a tenir valor nul, i no indicaran cap tipus de connexió.
- En cas de modificació, els atributs de la clau forana dels tuples que facin referència al tuple modificat passaran a tenir valor nul, i no indicaran cap tipus de connexió.

La política d'anul·lació només es pot aplicar si els atributs de les claus foranes implicades admeten els valors nuls.

### Exemple d'anul·lació

Prenem una vegada més com a punt de partida dels exemples les relacions ALUMNE i AULA que es mostren en la taula 1.17 i taula 1.18 respectivament.

Aplicant la política d'anul·lació, tant si esborrem el tuple <102, 36> de la relació AULA, com si només canviem el valor de l'atribut de la seva clau primària (Codi) per un altre (com, per exemple, 105), el tuple de la relació ALUMNE que hi fa referència actualitzarà el valor de l'atribut CodiAula de 102 a valor nul, per tal d'evitar una connexió incorrecta entre els tuples d'ambdues relacions, i aleshores resultarà el tuple següent:

<47126654F, Josep, Bel Rovira, 453641282, 47126654F, NUL>



**TAULA 1.17.** Relació amb claus foranes

ALUMNE					
DNI	Nom	Cognoms	Telefon	DNIDelegat	CodiAula
47126654F	Josep	Bel Rovira	453641282	47126654F	102
51354897S	Anna	Pacheco Cuscó	723352151	51354897S	201
56354981L	Xavier	Rius Montalvo	726922235	51354897S	201
24583215W	Mariona	Castellví Mur	NUL	NUL	NUL

**TAULA 1.18.** Relació amb clau primària referenciada

AULA	
Codi	Capacitat
101	40
102	36
201	30

### Selecció de la política que s'ha de seguir

Serà el dissenyador de cada BD qui escollirà la política més adequada que s'ha de seguir en cada cas concret. Com a orientació, convé saber que les opcions més freqüents, sempre que no calgui fer consideracions addicionals, són les següents:

- En cas d'esborrament, normalment s'opta per la restricció.
- En cas de modificació, el més habitual és optar per l'actualització en cascada.

La política d'anul·lació és molt menys freqüent, i es posa en pràctica quan es volen conservar certes dades, encara que hagin perdut la connexió que tenien abans, de vegades amb l'esperança que la puguin recuperar més endavant.

### 1.2.4 Integritat del domini

La regla d'integritat del domini implica que tots els valors no nuls que contenen els atributs de les relacions de qualsevol BD han de pertànyer als respectius dominis declarats per als atributs en qüestió.

Aquesta condició és aplicable tant pel que fa als dominis predefinitos, com també pel que fa als dominis definits per l'usuari.

La regla d'integritat del domini també comporta que els operadors que és possible aplicar sobre els valors depenen dels dominis dels respectius atributs que els emmagatzemen.

### Exemples d'integritat de domini

Fixem-nos, per últim cop, en la relació AULA que es mostra en la taula 1.19.

**TAULA 1.19.** Exemple de relació amb clau primària referenciada

AULA	
Codi	Capacitat
101	40
102	36
201	30

Si en la relació amb esquema AULA(Codi, Capacitat) definim el domini de l'atribut Codi com el dels nombres enters de 0 fins a 999, aleshores no podrem inserir, per exemple, un valor en l'atribut que forma la clau primària que no pertanyi al seu domini, com ara INF, o LAB.

Tampoc no podrem aplicar determinats operadors per comparar valors de la clau primària amb valors que no pertanyin al seu domini. Així, no podrem consultar les característiques d'una aula amb Codi='INF', ja que 'INF' és una cadena de caràcters.

#### Model Entitat Relació o model ER

El model ER és un model de dades que té com a resultat un diagrama ER o diagrama Chen on gràficament es poden identificar els principals elements de dades, les seves característiques més importants i les interrelacions entre els mateixos.

#### Fases del disseny de BD

Les fases del disseny de BD són:

1. Disseny conceptual
2. Disseny lògic
3. Disseny físic

### 1.3 Traducció del model Entitat-Relació al model relacional

Una vegada conegudes les característiques d'un model de bases de dades relacional, caldrà partir del model conceptual general (del model Entitat-Relació) i fer un estudi del disseny lògic de bases de dades en aquest àmbit, el relacional.

En tots els exemples, pressuposarem que prèviament ha tingut lloc una fase de disseny conceptual de la qual ha resultat un model Entitat-Relació (o model ER) recollit en els diagrames Chen de què es tracti en cada cas.

Abans d'implementar pròpiament la BD dins de l'entorn ofert pel SGBD utilitzat, cal transformar aquests diagrames en estructures de dades relacionals.

El model ER es basa en les entitats i en les interrelacions existents entre aquestes. Podem avançar uns quants aspectes generals sobre com s'han de traduir aquests elements al model relacional:

- Les entitats sempre donen lloc a relacions, siguin del tipus que siguin.
- Les interrelacions binàries de connectivitat 1-1 o 1-N originen claus foranes en relacions ja existents.
- Les interrelacions binàries de connectivitat M-N i totes les n-àries d'ordre superior a 2 sempre es transformen en noves relacions.

És convenient seguir un cert ordre a l'hora de dissenyar lògicament una base de dades. Una bona pràctica pot consistir a procedir de la manera següent:

1. En primer lloc, cal transformar les entitats del diagrama amb el qual treballem en relacions.
2. Després s'ha de continuar transformant en relacions les entitats que presenten algun tipus d'especificitat (és a dir, les febles, les associatives, o les derivades d'un procés de generalització o especialització).
3. A continuació, s'han d'afegir a les anteriors relacions els atributs necessaris per formar les claus foranes derivades de les interrelacions binàries amb connectivitat 1-1 i 1-N presents en el diagrama ER.
4. I, finalment, ja pot començar la transformació de les interrelacions binàries amb connectivitat M-N i de les interrelacions n-àries.



Cap SGBD no pot resoldre una referència a una taula que encara no ha estat creada.

D'aquesta manera evitem que hi hagi claus foranes que facin referència a relacions que encara no s'han descrit. Això fa més llegidor el model relacional obtingut, certament, però també estalvia la feina d'haver d'ordenar les relacions a l'hora d'escriure (típicament en llenguatge SQL) i les instruccions pertinents per tal que el SGBD utilitzat creï les taules de la base de dades.

Les tècniques necessàries per realitzar correctament el disseny lògic de bases de dades, segons el tipus de conceptualització de què es tracti en cada cas.

### 1.3.1 Entitats

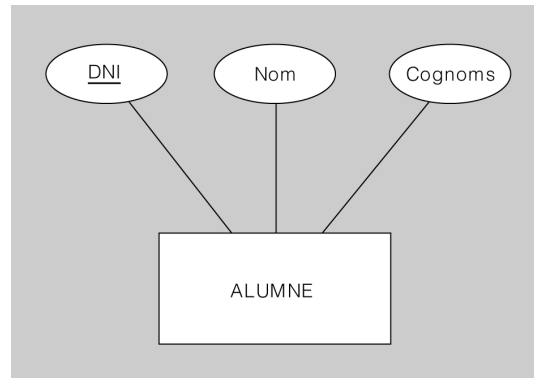
Cada entitat del model ER es transforma en una relació del model relacional:

- Els atributs de l'entitat originària seran els atributs de la relació resultant.
- La clau primària de l'entitat originària serà la clau primària de la relació resultant.
- Quan una entitat intervé en alguna interrelació binària 1-1 o 1-N, pot ser necessari afegir ulteriorment nous atributs, per tal que actuï com a claus foranes de la relació.

#### Exemple de transformació d'entitat

El diagrama ER de la figura 1.1 es tradueix al model relacional de la manera següent:

ALUMNE(DNI, Nom, Cognoms)

**FIGURA 1.1.** Entitat

### 1.3.2 Interrelacions

Un cop transformades totes les entitats en relacions, cal traduir les interrelacions en què aquelles participen.

**1. Binàries.** Per traduir les interrelacions binàries cal tenir en compte la seva connectivitat, així com també les dependències d'existència.

**a. Connectivitat 1-1 i dependències d'existència.** Cal afegir a qualsevol de les dues relacions una clau forana amb valor únic que faci referència a l'altra relació.

#### Dependències d'existència

De vegades, una entitat instància només té sentit si hi ha com a mínim una altra entitat instància que hi està associada mitjançant una interrelació binària determinada. En aquests casos, es diu de la darrera entitat que és una entitat obligatòria en la interrelació. Altrament, es diu que es tracta d'una entitat opcional en la interrelació.

Però si una de les dues entitats és opcional en la relació, aleshores és ella qui ha d'acollir la clau forana, per tal d'evitar, en cas contrari, l'emmagatzematge de valors nuls en aquesta, i estalviar-se així espai d'emmagatzematge.

Els atributs de la interrelació (si n'hi ha) acompanyen la clau forana.

#### Exemple de transformació d'interrelació binària amb connectivitat 1-1

El diagrama ER de la figura 1.2 representa una interrelació binària amb connectivitat 1-1. Per tant, en principi hi hauria dues possibilitats de transformació, segons si es col·loca la clau forana en l'entitat PROFESSOR o en l'entitat DEPARTAMENT:

DEPARTAMENT(Codi, Descripció)

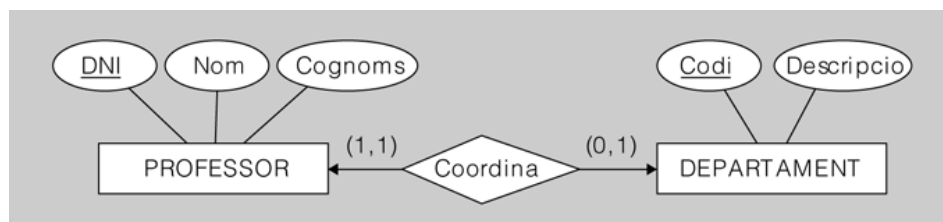
PROFESSOR(DNI, Nom, Cognoms, CodiDepartament) ON {CodiDepartament} REFERENCIA DEPARTAMENT i CodiDepartament ADMET VALORS NULS

O bé:

PROFESSOR(DNI, Nom, Cognoms)

DEPARTAMENT(Codi, Descripció, DNIProfessor) ON {DNIProfessor} REFERENCIA PROFESSOR

Ara bé, l'entitat DEPARTAMENT és opcional en la interrelació Coordina. Això vol dir que hi pot haver professors que no coordinin cap departament. Per tant, l'opció més correcta consisteix a afegir la clau forana a la relació DEPARTAMENT, ja que si s'afegís a la relació PROFESSOR hauria de prendre el valor nul en molts casos, i ocuparia un espai d'emmagatzematge innecessari.

**FIGURA 1.2.** Interrelació binària amb connectivitat 1-1

**b. Connectivitat 1-N.** En aquests casos cal afegir una clau forana a la relació que resulta de traduir l'entitat ubicada al costat N de la interrelació, que faci referència a l'altra relació.

Si es col·loqués la clau forana en l'altra relació, l'atribut que la forma hauria de ser multivalent per tal de poder representar totes les connexions possibles, i això no està permès dins del model relacional.

Els atributs de la interrelació (si n'hi ha) acompanyen la clau forana.

#### Exemple de transformació d'interrelació binària amb connectivitat 1-N

El diagrama ER de la figura 1.3 representa una interrelació binària amb connectivitat 1-N. Per tant, la clau forana s'haurà d'afegir necessàriament a l'entitat derivada de l'entitat del costat N, i resulta el model següent:

DEPARTAMENT(Codi, Descripcio)

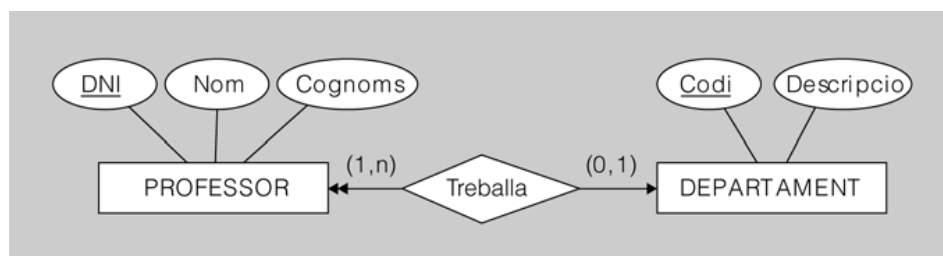
PROFESSOR(DNI, Nom, Cognoms, CodiDepartament) ON {CodiDepartament} REFERENCIA DEPARTAMENT i CodiDepartament ADMET VALORS NULS

→ CF:{CodiDepartament} --> DEPARTAMENT

L'entitat del costat 1 (DEPARTAMENT) és opcional en la interrelació Treballa. Això implica que l'entitat PROFESSOR admetrà valors nuls en la seva clau forana que fa referència a DEPARTAMENT, ja que hi podrà haver professors no assignats a cap departament. Però, al contrari del que passava amb les interrelacions 1-1, aquí no es podran evitar aquests valors nuls, ja que la clau forana ha d'anar necessàriament a l'entitat que resulta de traduir al model relacional l'entitat ubicada al costat N de la interrelació.

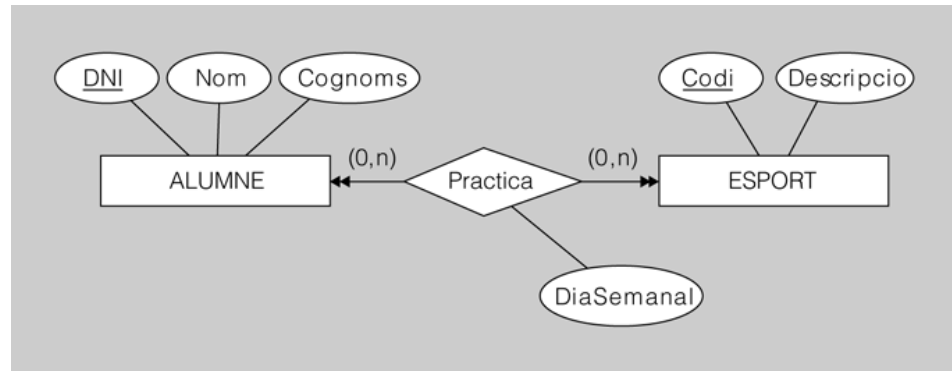
No cal posar que admet valors nuls perquè la clau forana ja té eixa definició.

Si hi ha restricció d'existència diríem que el CodiDepartament seria VNN.

**FIGURA 1.3.** Interrelació binària amb connectivitat 1-N

**c. Connectivitat M-N.** Cada interrelació M-N es transforma en una nova relació amb les característiques següents:

- La seva clau primària estarà formada pels atributs de les claus primàries de les dues entitats interrelacionades.
- Els atributs de la interrelació (si n'hi ha) es convertiran en atributs de la nova relació.

**FIGURA 1.4.** Interrelació binària amb connectivitat M-N**Exemple de transformació d'interrelació binària amb connectivitat M-N**

El diagrama ER de la figura 1.4 es tradueix al model relacional de la manera següent:

ALUMNE(DNI, Nom, Cognoms)

ESPORT(Codi, Descripcio)

PRACTICA(DNIAlumne, CodiEsport, DiaSemanal) ON {DNIAlumne} REFERENCIA ALUMNE i {CodiEsport} REFERENCIA ESPORT

**2. Ternàries.** Tota interrelació ternària es transforma en una nova relació, que tindrà per atributs els de les claus primàries de les tres entitats interrelacionades, més els atributs propis de la interrelació, si en té.

La composició de clau primària de la nova relació depèn de la connectivitat de la interrelació ternària originària.

**a. Connectivitat M-N-P.** En aquest cas, la clau primària està formada per tots els atributs que formen les claus primàries de les tres entitats interrelacionades (si no fos així, la clau primària hauria de repetir algunes combinacions dels seus valors per tal de modelitzar totes les possibilitats, però aquesta possibilitat no està permesa dins del model relacional).

**Exemple de transformació d'interrelació ternària amb connectivitat M-N-P**

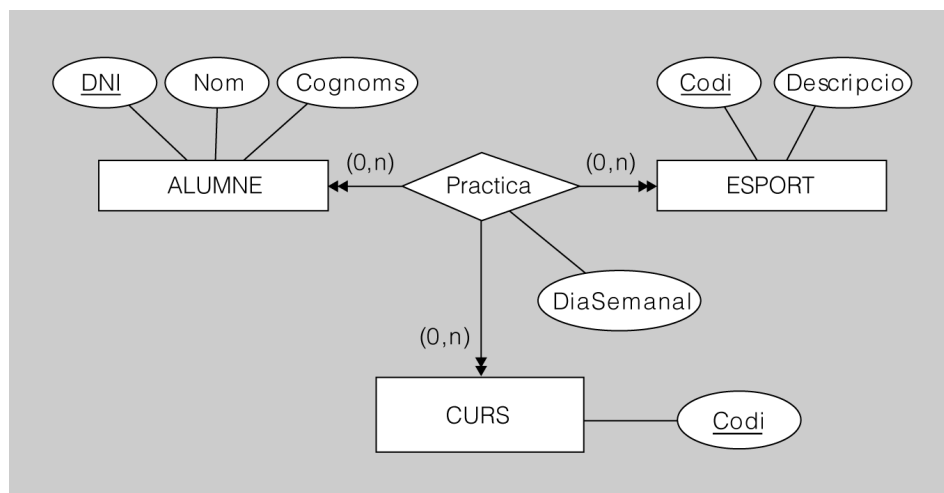
El diagrama ER de la figura 1.5 es tradueix al model relacional de la manera següent:

ALUMNE(DNI, Nom, Cognoms)

ESPORT(Codi, Descripcio)

CURS(Codi)

PRACTICA(DNIAlumne, CodiEsport, CodiCurs, DiaSemanal) ON {DNIAlumne} REFERENCIA ALUMNE {CodiEsport} REFERENCIA ESPORT i {CodiCurs} REFERENCIA CURS

**FIGURA 1.5.** Interrelació ternària amb connectivitat M-N-P

**b. Connectivitat 1-M-N.** La clau primària està composta per tots els atributs que formen les claus primàries de les dues entitats que són a tots dos costats de la interrelació etiquetats amb una N (o amb el que és equivalent, una fletxa de punta doble). Hem d'indicar que l'atribut sobrant sigui VNN per tal que queden ben relacionades les 3 entitats.

**Exemple de transformació d'interrelació ternària amb connectivitat 1-M-N**

El diagrama ER de la figura 1.6 es tradueix al model relacional de la manera següent:

ALUMNE(DNI, Nom, Cognoms)

ESPORT(Codi, Descripció)

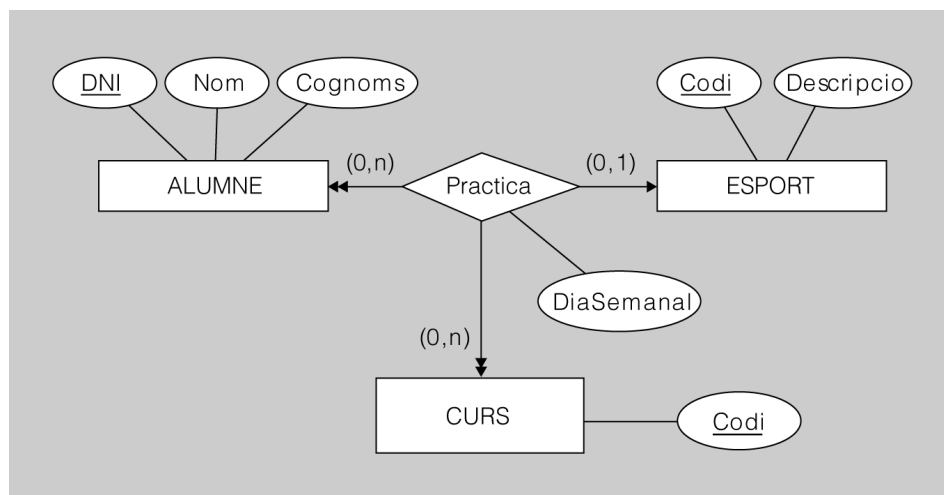
CURS(Codi)

PRACTICA(DNIAlumne, CodiEsport, CodiCurs, DiaSemanal) ON {DNIAlumne} REFERENCIA ALUMNE {CodiEsport} REFERENCIA ESPORT, I {CodiCurs} REFERENCIA CURS i VNN: {CodiEsport}

Fixem-nos que, en aquest cas, un alumne només pot practicar un esport en cada curs acadèmic i, per tant, no cal incorporar la clau de l'entitat ESPORT a la clau de la relació PRACTICA.

Per a representar qualsevol RE en una relació ternària s'ha d'afegir una RI. Per exemple, en este cas, si hi ha RE, és a dir, si l'entitat amb cardinalitat 1 és (1,1) haurem d'afegir a la relació PRACTICA una RI:

ESPORT[CodiEsport] **C** PRACTICA[CodiEsport]

**FIGURA 1.6.** Interrelació ternària amb connectivitat 1-M-N

**c. Connectivitat 1-1-N.** En aquests casos, la clau primària està composta pels atributs que formen la clau primària de l'entitat del costat N de la interrelació, més els atributs que formen la clau primària de qualsevol de les altres dues entitats connectades amb cardinalitat 1.

Així, doncs, tota nova relació derivada d'una interrelació ternària amb connectivitat 1-1-N disposarà de dues claus candidates. L'elecció d'una d'aquestes com a clau primària de la nova relació quedarà al criteri del dissenyador lògic de BD.

**Exemple de transformació d'interrelació ternària amb connectivitat 1-1-N**

El diagrama ER de la figura 1.7 es pot traduir al model relacional de dues maneres:

↓  
Llavors l'altra ha de ser CAIt

ALUMNE(DNI, Nom, Cognoms)

ESPORT(Codi, Descripció)

CURS(Codi)

COORDINACIO(CodiCurs, DNIAlumne, CodiEsport, DiaSemanal) ON {DNIAlumne} REFERENCIA ALUMNE {CodiEsport} REFERENCIA ESPORT i {CodiCurs} REFERENCIA CURS i CAIt:{CodiCurs,CodiEsport}

O bé:

ALUMNE(DNI, Nom, Cognoms)

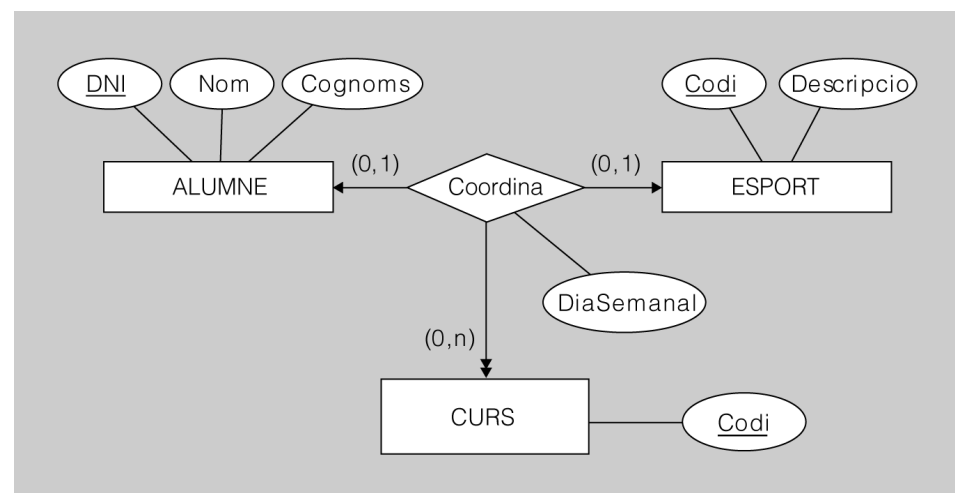
ESPORT(Codi, Descripció)

CURS(Codi)

COORDINACIO(CodiCurs, CodiEsport, DNIAlumne, DiaSemanal) ON {DNIAlumne} REFERENCIA ALUMNE {CodiEsport} REFERENCIA ESPORT i {CodiCurs} REFERENCIA CURS i CAIt:{CodiCurs,DNIAlumne}

Es pot veure que hem modificat el nom de la relació derivada de la interrelació per tal de convertir el verb originari en un substantiu, que normalment és més adequat per designar relacions.

**FIGURA 1.7.** Interrelació ternària amb connectivitat 1-1-N



**d. Connectivitat 1-1-1.** En aquests casos, la clau primària està composta pels atributs que formen la clau primària de dues entitats qualssevol, ja que totes tres estan connectades amb cardinalitat 1.



Així, doncs, tota nova relació derivada d'una interrelació ternària amb connectivitat 1-1-1 disposarà de tres claus candidates. L'elecció d'una d'aquestes com a clau primària de la nova relació quedarà al criteri del dissenyador lògic de BD.

→ Les altres 2 quedaran com a claus alternatives

#### Exemple de transformació d'interrelació ternària amb connectivitat 1-1-1

El diagrama ER de la figura 1.8 es pot traduir al model relacional de tres maneres:

ALUMNE(DNI, Nom, Cognoms)

ESPORT(Codi, Descripció)

CURS(Codi)

COORDINACIO(CodiCurs, DNIALumne, CodiEsport, DiaSemanal) ON {DNIALumne} REFERENCIA ALUMNE {CodiEsport} REFERENCIA ESPORT i {CodiCurs} REFERENCIA CURS

O bé:

ALUMNE(DNI, Nom, Cognoms)

ESPORT(Codi, Descripció)

CURS(Codi)

COORDINACIO(CodiCurs, CodiEsport, DNIALumne, DiaSemanal) ON {DNIALumne} REFERENCIA ALUMNE {CodiEsport} REFERENCIA ESPORT i {CodiCurs} REFERENCIA CURS

O bé:

ALUMNE(DNI, Nom, Cognoms)

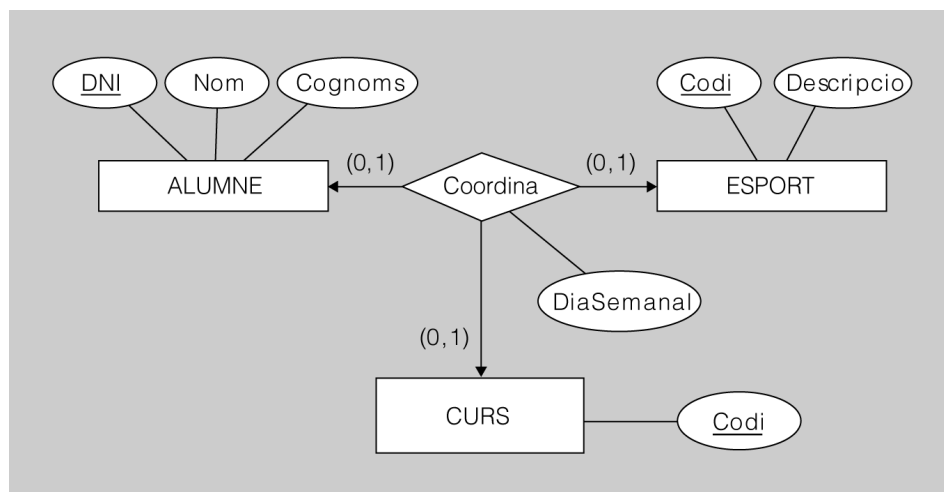
ESPORT(Codi, Descripció)

CURS(Codi)

COORDINACIO(CodiEsport, DNIALumne, CodiCurs, DiaSemanal) ON {DNIALumne} REFERENCIA ALUMNE {CodiEsport} REFERENCIA ESPORT i {CodiCurs} REFERENCIA CURS

Fixem-nos que hem canviat el significat del diagrama respecte al que hem representat en la figura 1.7: ara un alumne només pot coordinar la pràctica d'un esport durant un sol curs acadèmic, al llarg dels seus estudis, per tal d'afavorir la rotació en els càrrecs de coordinació del centre.

FIGURA 1.8. Interrelació ternària amb connectivitat 1-1-1



**3. n-àries.** Cada interrelació n-ària es transforma en una nova relació, que té com a atributs les claus primàries de totes les entitats relacionades, més els atributs propis de la interrelació originària, si en té.

La composició de la clau primària de la nova relació depèn de la connectivitat de la interrelació n-ària.

**a. Connectivitat de totes les entitats amb cardinalitat N.** La clau primària està formada per tots els atributs que formen les claus primàries de totes les entitats interrelacionades (n).

Cal seguir el mateix mecanisme que amb les interrelacions ternàries amb connectivitat M-N-P.

**b. Connectivitat d'una o més entitats amb cardinalitat 1.** La clau primària està formada per tots els atributs que formen les claus primàries de totes les entitats interrelacionades excepte una (n-1). L'entitat que no incorpora la seva clau primària a la de la nova relació ha d'estar forçosament connectada amb un 1.

Cal seguir el mateix mecanisme que amb les interrelacions ternàries amb connectivitat 1-M-N, 1-1-N i 1-1-1.

**4. Recursives.** Les interrelacions recursives traduïdes es comporten de la mateixa manera que la de la resta d'interrelacions:

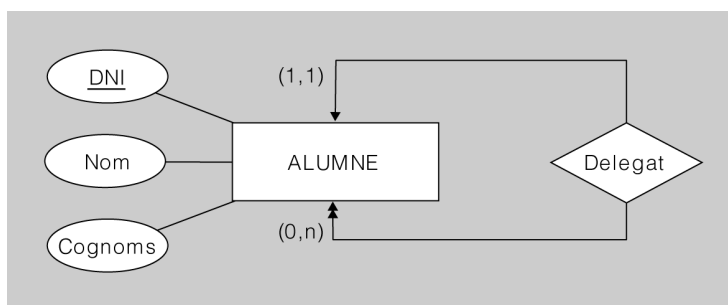
- Les binàries amb connectivitat 1-1 i 1-N donen lloc a una clau forana.
- Les binàries amb connectivitat M-N i les n-àries originen una nova relació.

**a. Binàries amb connectivitat 1-1 o 1-N.** En aquestes situacions, cal afegir a la relació sorgida de l'entitat originària que es relaciona amb ella mateixa una clau forana que faci referència a la pròpia clau primària.

Evidentment, els atributs de la clau forana no poden tenir els mateixos noms que els de la clau primària als quals fan referència, ja que tots dos es troben en la mateixa relació, i això atemptaria contra els principis del model relacional.

**FIGURA 1.9.** Interrelació recursiva binària amb connectivitat 1-N

Es comporten com les binàries.



**Exemple de transformació d'interrelació recursiva binària amb connectivitat 1-N**

El diagrama ER de la figura 1.9 es tradueix al model relacional de la manera següent:

ALUMNE(DNI, Nom, Cognoms, DNIDelegat) ON {DNIDelegat} REFERENCIA ALUMNE

**b. Binàries amb connectivitat M-N.** Quan la interrelació recursiva binària té connectivitat M-N s'origina una nova relació, la qual té com a clau primària els atributs que formen la clau primària de l'entitat originària, però dos cops, ja que cal modelitzar el fet que l'única entitat que intervé en la conceptualització prevista s'interrelaciona amb ella mateixa (i no pas amb una altra de diferent).

Cal modificar convenientment els noms d'aquests atributs que són presents dos cops en la nova relació perquè no coincideixin, i respectar així les directrius del model relacional.

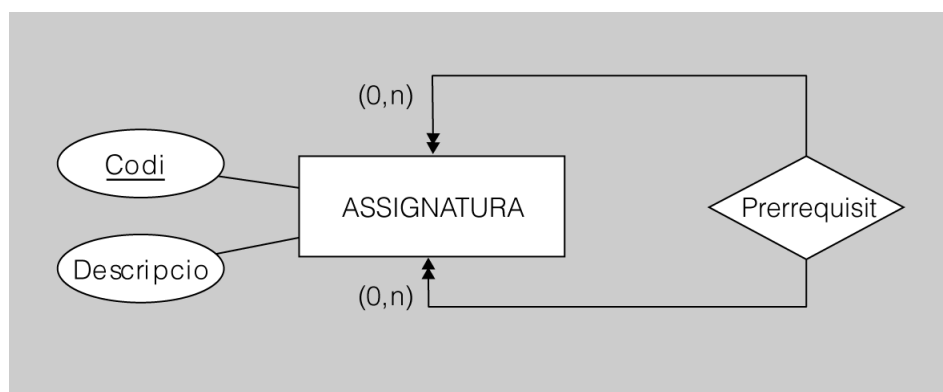
#### Exemple de transformació d'interrelació recursiva binària amb connectivitat M-N

El diagrama ER de la figura 1.10 es tradueix al model relacional de la manera següent:

ASSIGNATURA (Codi, Descripció)

PRERREQUISIT(CodiAssignatura, CodiPrerrequisit) ON {CodiAssignatura} REFERENCIA ASSIGNATURA i {CodiPrerrequisit} REFERENCIA ASSIGNATURA

**FIGURA 1.10.** Interrelació recursiva binària amb connectivitat M-N



**c. n-àries.** S'origina una nova relació, la clau primària de la qual es construeix de manera diferent en funció de la connectivitat:

Quan la connexió de totes les entitats es produeix amb cardinalitat N, la clau primària de la nova relació es compon de tots els atributs que formen part de les claus primàries de totes les entitats interrelacionades (n).

Quan la connexió d'una o més de les entitats es produeix amb cardinalitat 1, la clau primària de la nova relació es compon de tots els atributs que formen les claus primàries de totes les entitats interrelacionades excepte una (n-1). L'entitat que no incorpora la seva clau primària a la de la nova relació ha d'estar forçosament connectada amb un 1.

#### Exemple de transformació d'interrelació recursiva n-ària

El diagrama ER de la figura 1.11 es tradueix al model relacional de la manera següent:

ALUMNE(DNI, Nom, Cognoms)

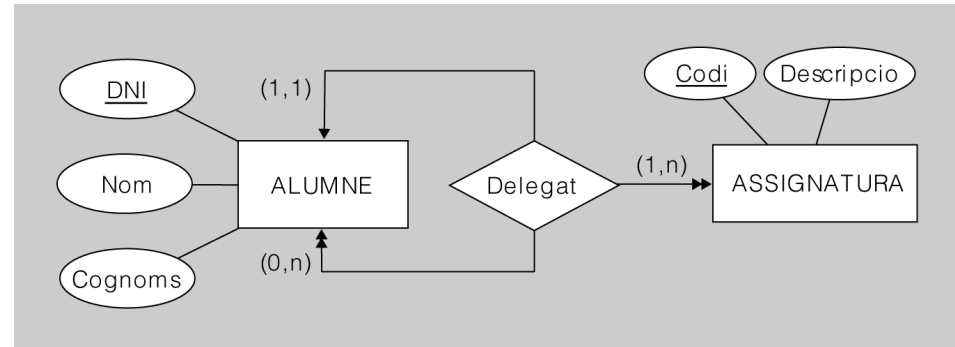
ASSIGNATURA(Codi, Descripció)

DELEGAT(DNIAlumne, CodiAssignatura, DNIDelegat) ON {DNIAlumne} REFERENCIA ALUMNE, {CodiAssignatura} REFERENCIA ASSIGNATURA i {DNIDelegat} REFERENCIA ALUMNE

Fixem-nos que hem incorporat a la clau primària de la nova relació els atributs que formen les claus primàries de les dues entitats connectades amb cardinalitat N, és a dir, ASSIGNATURA i ALUMNE, però des de la posició dels alumnes que no són delegats.

D'aquesta manera, es modelitza el fet que cada alumne té un delegat per a cada assignatura, i que el delegat de cada assignatura representa una pluralitat d'alumnes.

**FIGURA 1.11.** Interrelació recursiva n-ària



Aquest cas no es vorà.

### 1.3.3 Entitats febles

Com que les entitats febles sempre estan situades en el costat N d'una interrelació 1-N que els serveix per completar la identificació inequívoca de les seves instàncies, la relació derivada de l'entitat feble ha d'incorporar a la seva clau primària els atributs que formen la clau primària de l'entitat de la qual són tributàries. Els atributs esmentats constitueixen, simultàniament, una clau forana que fa referència a l'entitat de la qual depenen.

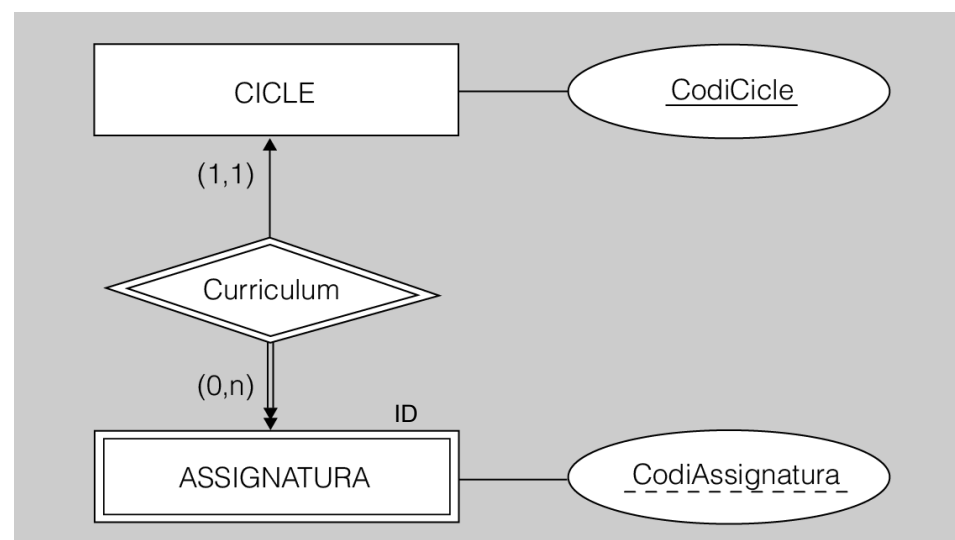
#### Exemple de transformació d'entitat feble

El diagrama ER de la figura 1.12 es tradueix al model relacional de la manera següent:

CICLE(CodiCicle)

ASSIGNATURA(CodiCicle, CodiAssignatura) ON {CodiCicle} REFERENCIA CICLE

**FIGURA 1.12.** Entitat feble



Es comporta com una relació binària 1:N però la clau forana formarà part també de la clau primària.

### 1.3.4 Generalització i especialització

En aquests casos, tant l'entitat superclasse com les entitats de tipus subclasse es transformen en noves relacions.

La relació derivada de la superclasse hereta d'aquesta la clau primària. A més, s'encarrega d'emmagatzemar els atributs comuns a tota l'especialització o generalització.

Les relacions derivades de les entitats de tipus subclasse també tenen, com a clau primària, la clau de l'entitat superclasse, que al mateix temps actua com a clau forana, en referenciar l'entitat derivada de la superclasse.

#### Exemple de transformació de generalització o especialització

La figura 1.13 mostra un encadenament de generalitzacions o especialitzacions. Si el traduïm a un model relacional obtenim el resultat següent:

PERSONA(DNI, Nom, Cognoms, Telefon)

PROFESSOR(DNI, Sou) ON {DNI} REFERENCIA PERSONA

ALUMNE(DNI) ON {DNI} REFERENCIA PERSONA

INFORMATIC(DNI, EspecialitatMaquinari, EspecialitatProgramari) ON {DNI} REFERENCIA PROFESSOR

ADMINISTRATIU(DNI, Titulacio, Especialitat) ON {DNI} REFERENCIA PROFESSOR

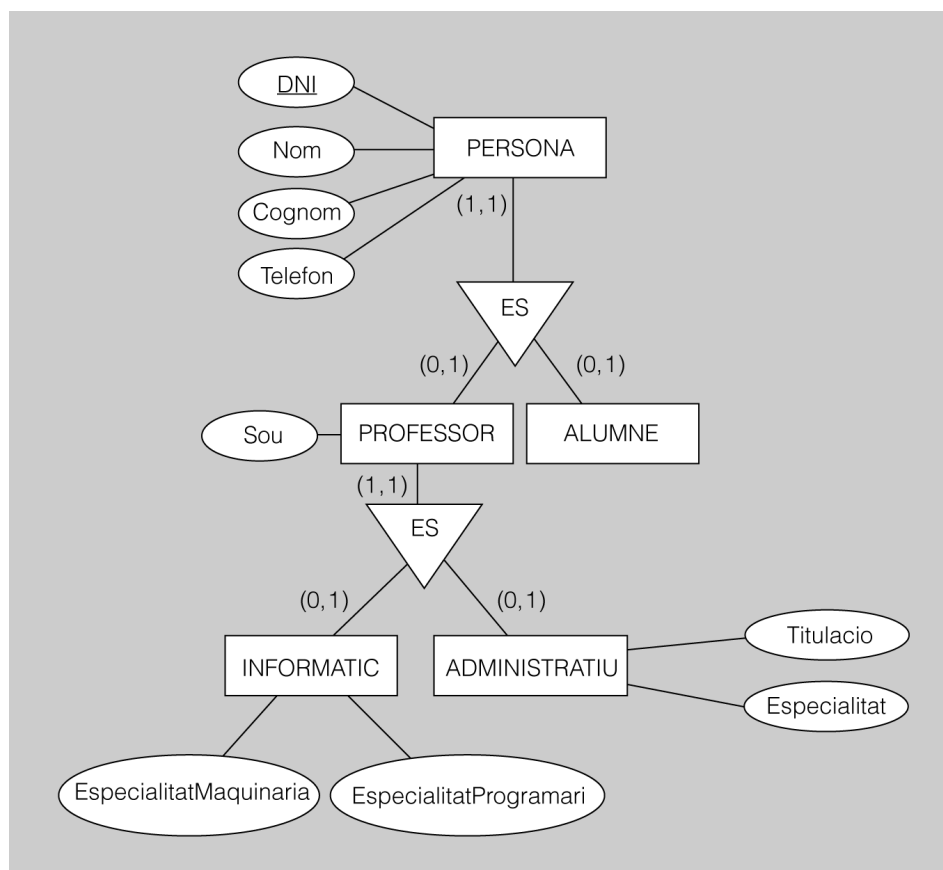
Aquesta correspon a No Disjunta + Parcial, a partir d'aquesta afegirem RI per a representar els altres 3 tipus.

Disjunta Parcial: L'intersecció de cada par d'entitats subclasse ha de ser buida, per ser Disjunta.  
 $PROFESSOR[DNI] \cap ALUMNE[DNI] = \text{buit}$

No Disjunta Total: Per ser Total hem d'afegir la RI:  
 $PERSONA[DNI] = PROFESSOR[DNI] \cup ALUMNE[DNI]$

Disjunta Total: Es la més restrictiva i serà com el primer esquema + les 2 anteriors RI.

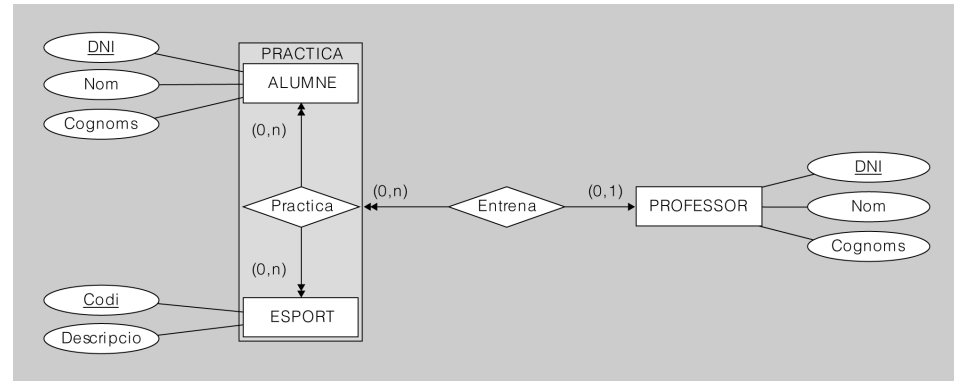
FIGURA 1.13. Generalització i especialització



### 1.3.5 Entitats associatives

Les entitats associatives es basen en una interrelació entre entitats. La traducció d'aquesta interrelació a un model relacional equival a la traducció de l'entitat associativa.

**FIGURA 1.14.** Entitat associativa



#### Exemple de transformació d'entitat associativa

El diagrama ER de la figura 1.14 es tradueix al model relacional de la manera següent:

ALUMNE(DNI, Nom, Cognoms)

ESPORT(Codi, Descripció)

PROFESSOR(DNI, Nom, Cognoms)

PRACTICA(DNIAlumne, CodiEsport, DNIProfessor) ON {DNIAlumne} REFERENCIA ALUMNE, {CodiEsport} REFERENCIA ESPORT i {DNIProfessor} REFERENCIA PROFESSOR

Fixem-nos com la relació PRACTICA, derivada de l'entitat associativa, incorpora una clau forana que fa referència a la relació PROFESSOR, ja que l'entitat associativa originària és al costat N d'una interrelació binària amb l'entitat PROFESSOR.

## 2. Normalització

El disseny d'una base de dades pot ser una tasca extremadament complexa. Hi ha diferents metodologies que permeten abordar el problema de trobar l'esquema relacional que representi millor la realitat que es vol modelitzar.

Coneixem el model Entitat-Relació per establir models per a qualsevol realitat, del qual s'obté, com a resultat, el diagrama Entitat-Relació, altrament anomenat *diagrama de Chen*. També coneixem el procés de traducció d'un diagrama Entitat-Relació a un esquema relacional.

Per tant, si per arribar a l'esquema relacional que ha de modelitzar la realitat hem seguit el camí que consisteix a, primerament, efectuar el diagrama Entitat-Relació per després efectuar-ne la traducció al model relacional, i el diagrama Entitat-Relació era correcte, haurem obtingut un esquema relacional del tot correcte. Aquest seria el camí aconsellable.

Però no sempre és així i ens trobem dissenys efectuats directament en l'esquema relacional. Hi ha diferents causes que ho provoquen:

- D'entrada, el model Entitat-Relació és posterior al model relacional i, per tant, hi ha bases de dades que van ser formulades directament en la terminologia relacional. No hi havia cap altra opció!
- Hi ha dissenyadors que “no volen perdre el temps” en un model Entitat-Relació i dissenyen directament en el model relacional. Quin error més gran!
- De vegades, s'ha de modificar la base de dades a causa de noves necessitats, i el disseny s'efectua directament sobre aquesta en lloc d'analitzar-se i realitzar-se sobre el model Entitat-Relació per després transferir els canvis a l'esquema relacional. Quin error més gran!

Fixeu-vos que donem un suport absolut al fet d'utilitzar el model Entitat-Relació per obtenir-ne posteriorment el model relacional. Un bon disseny en el model Entitat-Relació acostuma a proporcionar una base de dades relacional ben dissenyada, cosa que no passarà si el disseny Entitat-Relació incorpora errors. D'altra banda, si no hi ha hagut el disseny Entitat-Relació previ, hi ha més possibilitats de tenir una base de dades relacional mal dissenyada.

En l'apartat del “Model relacional” d'aquesta unitat, es presenta el procés de traducció d'un diagrama Entitat-Relació a un esquema relacional.

La **teoria de la normalització** és un mètode que permet assegurar si un disseny relacional (tant si prové de la traducció d'un diagrama Entitat-Relació com si s'ha efectuat directament en el model relacional) és més o menys correcte.

En general, els mals dissenys poden originar les situacions següents:

- Repetició de la informació
- Impossibilitat de representar certa informació:
  - Anomalies en les insercions
  - Anomalies en les modificacions
  - Anomalies en els esborraments

Un bon disseny ha d'aconseguir el següent:

- Emmagatzemar tota la informació necessària amb el mínim d'informació redundant.
- Mantenir el mínim de lligams entre les relacions de la base de dades per tal de facilitar-ne la utilització.
- Millorar la consultabilitat de les dades emmagatzemades.
- Minimitzar els problemes d'actualització (altes, baixes i modificacions) que poden sorgir en haver d'actualitzar simultàniament dades de diferents relacions.

#### Exemple de disseny relacional inadequat

Considerem el disseny relacional de la taula 2.1 per enregistrar la informació dels professors amb els alumnes de cadascun i la qualificació que han obtingut en els diversos crèdits.

**TAULA 2.1.** Exemple de disseny relacional inadequat

<u>DniProf</u>	<u>NomProfessor</u>	<u>DniAlum</u>	<u>NomAlumne</u>	<u>Edat</u>	<u>Credit</u>	<u>Nota</u>
33.333.333	Joan Finestra	77.777.777	Anna Taula	20	ADBD	4.5
33.333.333	Joan Finestra	88.888.888	Miquel Cadira	19	ADBD	5.7
33.333.333	Joan Finestra	77.777.777	Anna Taula	20	SGBD	6
33.333.333	Joan Finestra	88.888.888	Miquel Cadira	19	SGBD	7
44.444.444	Maria Porta	77.777.777	Anna Taula	20	MET	6
44.444.444	Maria Porta	88.888.888	Miquel Cadira	19	MET	5
44.444.444	Maria Porta	77.777.777	Anna Taula	20	LLC	4
44.444.444	Maria Porta	88.888.888	Miquel Cadira	19	LLC	3

Oi que convindreu que aquest disseny està pensat amb els peus? Ràpidament, hi veiem els problemes següents:

- Hi ha informació repetida, fet que pot provocar inconsistències. Fixem-nos que en cas d'haver de modificar qualsevol dels valors dels camps que formen la clau primària (*DniProf*,



*NomProfessor, DniAlum, NomAlumne, Edat, Credit*), el canvi s'ha d'efectuar en totes les files en què apareix aquest valor.

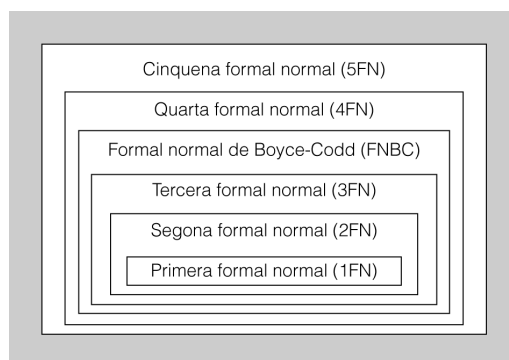
- No hi pot haver valors nuls en les columnes que formen la clau primària.
- Així, si no coneixem l'edat d'un alumne, tenim un greu problema.
- En cas d'arribar a la conclusió que necessitem emmagatzemar més informació dels professors o dels alumnes, caldrà afegir més columnes i repetir la informació per a cada fila en què aparegui el professor o alumne.
- Consultar la informació en la taula 2.1 pot esdevenir feixuc atesa la gran quantitat d'informació diferent que conté.

El mètode que proposa la teoria de la normalització per determinar si un disseny relacional és correcte consisteix a avaluar el disseny de totes les relacions (taules) per tal de veure en quin grau de normalitat es troba cadascuna i, així, poder decidir si el disseny ja és correcte o si cal refinar-lo.

La teoria de la normalització defineix les **formes normals** com a indicadors per avaluar el grau de normalitat de les relacions, i es diu que una relació està en una forma normal determinada quan satisfà un conjunt determinat de condicions.

Hi ha diferents graus de normalitat i, per tant, de formes normals, les quals compleixen la relació d'inclusió de la figura 2.1, que s'ha d'interpretar en el sentit que a mesura que augmenta el nivell de la forma normal, la relació ha de complir un conjunt de condicions més restrictiu i, per tant, continua verificant les condicions de les formes normals de nivell inferior.

**FIGURA 2.1.** Relació d'inclusió entre les diverses formes normals



Així, doncs, l'objectiu hauria de ser aconseguir un esquema relacional en què totes les relacions tinguessin el grau màxim de normalitat, és a dir, en què totes es trobessin en la cinquena forma normal (5FN).

El **procés de normalització** per aconseguir que una relació que es troba en una forma normal X passi a estar en una forma normal Y superior a X consisteix sempre en la descomposició o subdivisió de la relació original (forma normal X) en dues o més relacions que verifiquin el nivell de forma normal Y.

Per tant, el procés de normalització augmenta el nombre de relacions presents en la base de dades. Amb això, segur que s'aconsegueix una disminució de redundàncies i una disminució de les anomalies en els problemes d'actualització de la informació, però, en canvi, es penalitzen les consultes, ja que la seva execució haurà d'anar a cercar la informació en moltes taules relacionades entre elles.

Així, doncs, cal trobar un equilibri, i de vegades pot ser convenient renunciar al nivell màxim de normalització (5FN) i, per tant, permetre una certa redundància en els esquemes amb la finalitat d'alleugerir els costos de les consultes. En aquestes situacions, es parla d'un procés de desnormalització.

El nostre objectiu final és conèixer les condicions que han de complir les relacions per assolir cadascun dels nivells de forma normal, i el procés per dividir les relacions en noves relacions que verifiquin les condicions desitjades. Per aconseguir-ho, hem de conèixer els conceptes de *relació universal* i *dependència funcional*.

## 2.1 La relació universal

En efectuar directament el disseny relacional d'una base de dades, el dissenyador es troba amb un conjunt de conceptes que tradueix en atributs, els quals, pel seu significat, agruparà en una o més relacions.

Anomenem **relació universal** la relació consistent en l'agrupament dels atributs corresponents a **tots** els conceptes que constitueixen una base de dades relacional.

**TAULA 2.2.** Relació universal per a un esquema relacional ideat per a una gestió de comandes de compra

Num	DataComanda	Article	Descripció	Qtat	Preu	DataPrevista	NomProv	PaisProv	Moneda
22.523	25-05-2000	PC3-500	PC Pentium III a 500	5	150	1-06-2000	ARKANSAS	XINA	EUR
22.523	25-05-2000	PRO-15	Protector pantalla 15"	5	8	1-06-2000	ARKANSAS	XINA	EUR
22.524	27-05-2000	PC3-500	PC Pentium III a 500	15	145	5-06-2000	MELISSA	ITÀLIA	USD
22.524	27-05-2000	PRO-15	Protector pantalla 15"	15	50	5-06-2000	MELISSA	ITÀLIA	USD
22.525	27-05-2000	INK430	Cartutx de tinta 430	20	25	31-5-2000	ARKANSAS	XINA	EUR

Així, imaginem que es vol dissenyar una base de dades per al control de les comandes de compra d'una organització determinada. Imaginem que cal incloure-hi els conceptes corresponents a número i data de la comanda; codi, descripció,

quantitat i preu pactat per cada article sol·licitat; data prevista de lliurament de la comanda; nom (*NomProv*) i país (*PaisProv*) del proveïdor; i moneda en què es pacta la comanda. La relació universal es representa en la taula 2.2.

Oi que hi ha molta redundància i poca organització? Evidentment, el disseny relacional d'una base de dades basat en la relació universal acostuma a ser del tot incorrecte, i fa necessari aplicar un procés de normalització per tal d'anar dividint la relació en altres relacions de manera que assoleixin graus de normalitat millors, és a dir, compleixin les restriccions corresponents a les formes normals més elevades.

Molt poques vegades es parteix de la relació universal. L'experiència dels dissenyadors provoca que, d'entrada, ja es pensi en relacions que assoleixen un cert grau de normalitat.

## 2.2 Dependències funcionals

Les definicions de les diferents formes normals, és a dir, el conjunt de condicions que les defineixen, es basen en el concepte de **dependència funcional**.

Donats dos atributs (o conjunts d'atributs) A i B d'una relació R, direm que **B depèn funcionalment de A** si per cada valor de A existeix un, i només un, valor de B associat amb ell. També direm que **A implica B**. Ho simbolitzarem per  $A \rightarrow B$ .

El concepte de dependència funcional estableix lligams entre atributs o conjunt d'atributs d'una mateixa relació.

**TAULA 2.3.** Relació universal per a un esquema relacional ideat per a una gestió de comandes de compra

Num	DataComanda	Article	Descripció	Qtat	Preu	DataPrevista	NomProv	PaisProv	Moneda
22.523	25-05-2000	PC3-500	PC Pentium III a 500	5	150	1-06-2000	ARKANSAS	XINA	EUR
22.523	25-05-2000	PRO-15	Protector pantalla 15"	5	8	1-06-2000	ARKANSAS	XINA	EUR
22.524	27-05-2000	PC3-500	PC Pentium III a 500	15	145	5-06-2000	MELISSA	ITÀLIA	USD
22.524	27-05-2000	PRO-15	Protector pantalla 15"	15	50	5-06-2000	MELISSA	ITÀLIA	USD
22.525	27-05-2000	INK430	Cartutx de tinta 430	20	25	31-5-2000	ARKANSAS	XINA	EUR

En la relació universal de la taula 2.3 diríem que, entre d'altres, la data de comanda depèn funcionalment del número de comanda, igual que la data prevista, el nom i

el país del proveïdor i la moneda. Ho podríem escriure com segueix:

$Num \rightarrow DataComanda$

$Num \rightarrow DataPrevista$

$Num \rightarrow NomProv$

$Num \rightarrow PaisProv$

$Num \rightarrow Moneda$

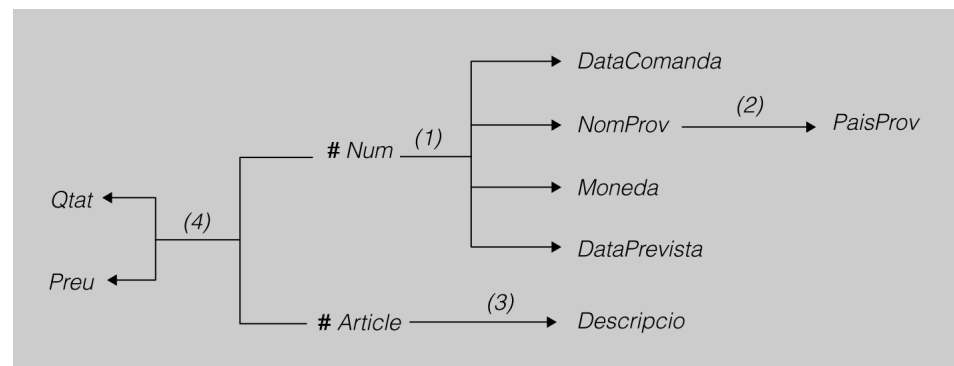
En tractar-se de diferents atributs que depenen funcionalment d'un mateix atribut, escriurem:

$Num \rightarrow DataComanda, DataPrevista, NomProv, PaisProv, Moneda$

Donats dos atributs (o conjunts d'atributs) A i B d'una relació R, direm que B té una **dependència funcional completa o total** de A si B depèn funcionalment de A però no depèn funcionalment de cap subconjunt de A.

És molt convenient representar les dependències funcionals d'una relació mitjançant un **esquema de dependències funcionals**. L'esquema per a la relació universal de la taula 2.3 seria el que es mostra en la figura 2.2.

**FIGURA 2.2.** Exemple d'esquema de dependències funcionals



Fixem-nos que hem marcat els atributs que són clau d'alguna de les entitats que formen part de la relació: *Article* identifica l'article i *Num* identifica la comanda. Fixem-nos, també, que la parella (*Num*, *Article*) identifica la quantitat i preu dels articles demanats en la comanda.

En aquest esquema, es poden veure les dependències funcionals entre els atributs. Es veu que *DataComanda*, *NomProv*, *Moneda* i *DataPrevista* depenen funcionalment (1) de *Num*, que *PaisProv* depèn funcionalment (2) de *NomProv*, i que *Descripció* depèn funcionalment (3) d'*Article*. Així mateix, *Qtat* i *Preu* depenen funcionalment (4) de *Num* i *Article*.

És evident que les dependències (1), (2) i (3) són totals, ja que la part esquerra de la dependència (l'**implicador**) està formada per un únic atribut i, per tant, és impossible que la part dreta de la dependència (l'**implicat**) pugui dependre d'un

subconjunt de l'implicador. La dependència (4) també és total, ja que *Qtati Preu* depenen de la parella (*Num*, *Article*) i no pas de cap subconjunt d'aquesta.

Una darrera apreciació sobre la dependència funcional (4) de la figura 2.2: veiem que en una mateixa comanda (*Num*) no és possible tenir diverses vegades el mateix article (*Article*), ja que els atributs *Qtat* i *Preu* depenen funcionalment de (*Num*, *Article*). En cas que fos necessari tenir diverses vegades el mateix article en una comanda, caldria utilitzar algun altre atribut per identificar l'article dins la comanda com, per exemple, el *NumeroDeLinia* de la comanda.

Donat un atribut o conjunt d'atributs *A* d'una relació, direm que **A és un determinant** de la relació si hi ha algun altre atribut o conjunt d'atributs *B* que té dependència funcional total de *A*.

En el cas anterior, veiem que *Num*, *Article*, *NomProv* i la parella (*Num*, *Article*) són determinants de la relació.

Donats *A*, *B* i *C* atributs o conjunts d'atributs d'una relació, direm que **C depèn transitivament** de *A* a través de *B* si *B* depèn funcionalment de *A*, *C* depèn funcionalment de *B*, i *A* no depèn funcionalment de *B*.

En l'exemple de la figura 2.2, podem dir que *PaisProv* depèn transitivament de *Num* a través de *NomProv*.

## 2.3 Primera forma normal

En aplicar el procés de normalització a una relació, es comença per comprovar si la relació està en primera forma normal i, si no és el cas, s'efectuen les modificacions oportunes per aconseguir-ho.

Una relació està en **primera forma normal (1FN)** si cap atribut pot contenir valors no atòmics (indivisible).

Considerem la relació universal de la taula 2.4.

La relació de la taula 2.4 no està en 1FN, ja que té atributs que poden contenir més d'un valor. Veiem que aquest exemple inclou tres files (comandes 22.523, 22.524 i 22.525), i alguns atributs (*Article*, *Descripcio*, *QtatiPreu*) tenen diversos valors per a algunes de les files.

El procés que s'ha de seguir per assolir una 1FN és afegir tantes files com sigui necessari per a cadascun dels diferents valors del camp o camps que tinguin valors no atòmics.

Així, en el nostre cas, obtenim la relació en 1FN de la taula 2.5.

**TAULA 2.4.** Relació que té atributs multivalor i, per tant, no es troba en 1FN

Num	DataComanda	Article	Descripció	Qtat	Preu	DataPrevista	NomProv	PaisProv	Moneda
22.523	25-05-2000	PC3-500 , PRO-15	PC Pentium III a 500 , Protector Pantalla 15"	5, 5	150, 8	1-06-2000	ARKANSAS	XINA	EUR
22.524	27-05-2000	PC3-500 , PRO-15	PC Pentium III a 500 , Protector Pantalla 15"	15, 15	145, 50	5-06-2000	MELISSA	ITÀLIA	USD
22.525	27-05-2000	INK430	Cartutx de tinta 430	20	25	31-5-2000	ARKANSAS	XINA	EUR

**TAULA 2.5.** Relació en 1FN

Num	DataComanda	Article	Descripció	Qtat	Preu	DataPrevista	NomProv	PaisProv	Moneda
22.523	25-05-2000	PC3-500	PC Pentium III a 500	5	150	1-06-2000	ARKANSAS	XINA	EUR
22.523	25-05-2000	PRO-15	Protector pantalla 15"	5	8	1-06-2000	ARKANSAS	XINA	EUR
22.524	27-05-2000	PC3-500	PC Pentium III a 500	15	145	5-06-2000	MELISSA	ITÀLIA	USD
22.524	27-05-2000	PRO-15	Protector pantalla 15"	15	50	5-06-2000	MELISSA	ITÀLIA	USD
22.525	27-05-2000	INK430	Cartutx de tinta 430	20	25	31-5-2000	ARKANSAS	XINA	EUR

De fet, la restricció que persegueix la 1FN forma part de la definició del model relacional i, per tant, tota relació, per definició, ha d'estar en 1FN. És a dir, aquesta forma normal és redundant amb la definició del model relacional i no caldria considerar-la. Es manté, però, per assegurar que les relacions dissenyades tenen un punt de partida correcte.

En general, les relacions en 1FN poden tenir molta informació redundant. Això no ens ha de preocupar, ja que la solució rau en les formes normals de nivell superior.

#### Natural Join de R1\*R2

L'operació de Natural Join és la combinació de totes les dades de la primera relació (R1) amb totes les de la segona (R2), sempre que els valors de les dades de les columnes anomenades de forma idèntica a les dues relacions coincideixin.

## 2.4 Preservació d'informació i dependències en la normalització

En el procés de normalització d'una relació R s'apliquen processos de descomposició per aconseguir relacions R1, R2, ..., Rn que verifiquin un nivell de

normalització superior al de la relació R. La descomposició consisteix a efectuar projeccions de la relació R sobre atributs que verifiquen certes condicions, la qual cosa dóna lloc a l'aparició de R1, R2, ..., Rn.

Cal garantir que la descomposició de R en R1, R2, ..., Rn preservi la informació existent, és a dir, que el *natural-join*  $R1 * R2 * \dots * Rn$  proporcioni exactament la mateixa informació que tenia la relació R original, tant en intensió (quantitat d'atributs) com en extensió (quantitat de files).

De la mateixa manera caldria garantir la conservació de les dependències, és a dir, el conjunt de dependències associades a la relació R original ha de ser equivalent al conjunt de dependències associat a les relacions R1, R2, ..., Rn.

Ja podem avançar que la conservació de les dependències no es pot garantir en tots els processos de normalització.

## 2.5 Segona forma normal

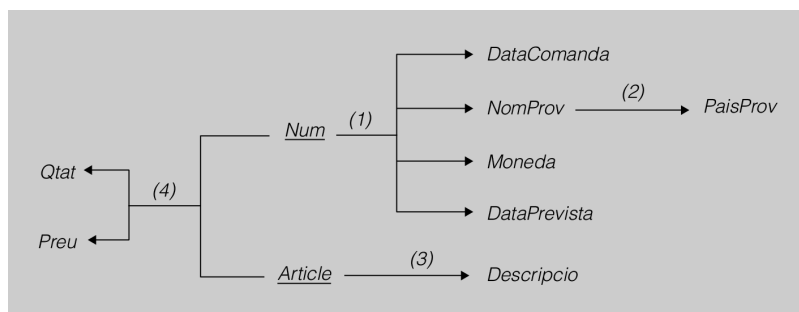
La segona forma normal persegueix l'eliminació dels problemes motivats per la presència de dependències funcionals no totals dels atributs que no formen part de la clau primària respecte a la clau primària.

Una relació està en **segona forma normal (2FN)** si està en 1FN i tot atribut que no pertanyi a la clau té dependència funcional total de la clau.

Considerem la relació següent en 1FN ideada per a la gestió de comandes de compra en una organització:

1	R ( __Num__, DataComanda, __Article__, Descripcio, Qtat, Preu,
2	DataPrevista, NomProv, PaisProv, Moneda)

**FIGURA 2.3.** Esquema de dependències funcionals



Veiem que la clau primària està formada per la parella (Num, Article) i que, en l'esquema de dependències funcionals associat (figura 2.3), hi ha atributs fora de la clau primària que no tenen dependència funcional completa de la clau.

En efecte, les dependències funcionals (1), (2) i (3) ens presenten atributs que no tenen dependència funcional total de la clau, formada per la parella (*Num*, *Article*).

El procés que s'ha de seguir per assolir una 2FN és dividir la relació (conservant la informació i les dependències) en tantes relacions com sigui necessari de manera que cada relació verifiqui que els seus atributs no-clau tenen dependència funcional total de la clau.

L'esquema de dependències funcionals ajuda a veure les relacions que han d'aparèixer. Així, en el nostre cas, de les dependències (1), (3) i (4) obtenim les relacions en 2FN:

```

1 COMANDA (__Num__, DataComanda, DataPrevista, Moneda, NomProv, PaisProv)
2 ARTICLE (__Article__, Descripcio)
3 DETALL (__Num__, __Article__, Qtat, Preu)
4      comanda  article

```

És molt probable que aquest disseny fos el proposat com a punt de partida, és a dir: sovint, en efectuar un disseny ja obtindrem relacions que estan en 2FN i, fins i tot, en formes normals de nivell superior.

La informació que hi ha en la taula 2.6 corresponent a la relació que acabem de normalitzar ara passa a estar repartida en tres taules (taula 2.7, taula 2.8 i taula 2.9).

**TAULA 2.6.** Relació en 1FN

Num	DataComanda	Article	Descripcio	Qtat	Preu	DataPrevista	NomProv	PaisProv	Moneda
22.523	25-05-2000	PC3-500	PC Pentium III a 500	5	150	1-06-2000	ARKANSAS	XINA	EUR
22.523	25-05-2000	PRO-15	Protector pantalla 15"	5	8	1-06-2000	ARKANSAS	XINA	EUR
22.524	27-05-2000	PC3-500	PC Pentium III a 500	15	145	5-06-2000	MELISSA	ITÀLIA	USD
22.524	27-05-2000	PRO-15	Protector pantalla 15"	15	50	5-06-2000	MELISSA	ITÀLIA	USD
22.525	27-05-2000	INK430	Cartutx de tinta 430	20	25	31-5-2000	ARKANSAS	XINA	EUR

**TAULA 2.7.** Relació en 2FN que emmagatzema les comandes

COMANDA					
Num	DataComanda	DataPrevista	NomProv	PaisProv	Moneda
22.523	25-05-2000	1-06-2000	ARKANSAS	XINA	EUR
22.524	27-05-2000	5-06-2000	MELISSA	ITÀLIA	USD
22.525	27-05-2000	31-5-2000	ARKANSAS	XINA	EUR



**TAULA 2.8.** Relació en 2FN pels articles

ARTICLE	
Article	Descripció
PC3-500	PC Pentium III a 500
PRO-15	Protector pantalla 15"
INK430	Cartutx de tinta 430

**TAULA 2.9.** Relació en 2FN pel detall de comanda

DETALL			
Num	Article	Qtat	Preu
22.523	PC3-500	5	150
22.523	PRO-15	5	8
22.524	PC3-500	15	145
22.524	PRO-15	15	50
22.525	INK430	20	25

Per acabar, fixem-nos que amb el nou disseny s'ha aconseguit eliminar molta redundància i, per tant, es redueixen els problemes en les operacions d'actualització i consulta. Però no desapareixen tots.

## 2.6 Tercera forma normal

Considerem el disseny de les relacions següents en 2FN ideades per a la gestió de les comandes de compra d'una organització:

```

1 COMANDA (__Num__, DataComanda, DataPrevista, Moneda, NomProv, PaisProv)
2 ARTICLE (__Article__, Descripcio)
3 DETALL (__Num__, __Article__, Qtat, Preu)
4         comanda  article

```

Fixem-nos que el país del proveïdor apareix en cada comanda. Si partim de la base que el país on resideix el proveïdor és únic, oi que encara hi ha informació redundant?

La tercera forma normal persegueix l'eliminació dels problemes motivats per la presència de dependències transitives dels atributs que no formen part de la clau primària, respecte de la clau primària.

Una relació està en **tercera forma normal (3FN)** si està en 2FN i cap atribut que no pertany a la clau depèn transitivament de la clau.

Les relacions *ARTICLE* i *DETALL* que ens ocupen ja estan en 3FN, però considerem la relació *comanda* que conté el país del proveïdor.

```

1 COMANDA (__Num__, DataComanda, DataPrevista, Moneda, NomProv, PaisProv)

```

L'atribut *PaisProv* depèn transitivament de *Num* a través de *NomProv*. Per tant, aquesta relació no està en 3FN.

El procés que s'ha de seguir per assolir una 3FN és dividir la relació (conservant la informació i les dependències) en noves relacions més simples, de manera que cada relació verifiqui que cap dels seus atributs no-clau depèn transitivament de la clau.

En el nostre cas, obtenim les relacions en 3FN:

```

1 PROVEIDOR (__CodProv__, NomProv, PaisProv)
2 COMANDA (__Num__, DataComanda, DataPrevista, Moneda, CodProv)
3                                     proveïdors

```

Fixem-nos que, en trencar la relació inicial *COMANDA*, ha semblat oportú considerar un nou atribut (*CodProv*) que identifiqui millor la nova relació *PROVEÏDOR*. Aquest fet no és imprescindible i no sempre serà convenient. Podríem haver considerat el trencament següent:

```

1 PROVEIDOR (__NomProv__, PaisProv)
2 COMANDA (__Num__, DataComanda, DataPrevista, Moneda, NomProv)
3                                     proveïdors

```

Ara bé, per a aquesta darrera possibilitat hem escollit el nom del proveïdor com a clau primària de la nova relació *PROVEÏDOR*, i l'experiència ens aconsella definir un codi que ens permeti identificar-los de manera més clara que la que proporciona el seu nom.

Tenint en compte el disseny que incorpora l'atribut *CodProv*, tindríem la conversió de la taula 2.7 en la taula 2.10 i taula 2.11.

**TAULA 2.10.** Relació en 3FN que emmagatzema les comandes

COMANDA				
Num	DataComanda	DataPrevista	CodProv	Moneda
22.523	25-05-2000	1-06-2000	ARK	EUR
22.524	27-05-2000	5-06-2000	MEL	USD
22.525	27-05-2000	31-5-2000	ARK	EUR

**TAULA 2.11.** Relació en 3FN pels proveïdors

PROVEIDOR		
CodProv	NomProv	PaisProv
ARK	ARKANSAS	XINA
MEL	MELISSA	ITÀLIA

## 2.7 Forma normal de Boyce-Codd

Considerem les naus d'emmagatzematge que hi ha en un gran mercat dedicades a guardar les mercaderies dels venedors del mercat. Imaginem que cada nau guarda mercaderia d'un tipus concret (carn fresca, peix fresc, congelats, vegetals, basar...) i que cada venedor pot dipositar mercaderia en diferents naus segons el tipus de mercaderia de cada nau (una parada de peix del mercat es pot dedicar a vendre peix fresc i peix congelat, per exemple). Ara bé, tota la mercaderia d'unes mateixes característiques d'un venedor es troba concentrada en una ubicació dins una mateixa nau per minimitzar al màxim els desplaçaments del venedor.

Per tenir constància de quin tipus de material hi ha a cada nau, es dissenya aquesta relació:

1 **DIPOSIT** (\_\_Venedor\_\_, \_\_TipusMaterial\_\_, Nau, Ubicacio)

La taula 2.12 ens exemplifica la situació. Fixem-nos que es troba en 3FN:

- Tots els atributs no pertanyents a la clau (*Nau* i *Ubicacio*) tenen dependència funcional total de la clau (2FN).

**TAULA 2.12.** Relació en 3FN

DIPOSIT			
Venedor	TipusMaterial	Nau	Ubicacio
JOSEP	Peix fresc	15	S4-C3-U1:10
MARIA	Peix fresc	25	S3-C5-U5:22
RAMON	Congelats	17	S2-C4-U1:25
ANNA	Vegetals	20	S1-C6-U7:10
ANNA	Peix fresc	25	S2-C5-U12:15
MARIA	Basar	10	S3-C4-U20:25

En efecte, la *Nau* i la *Ubicacio* dins la nau depenen del *Venedor* i del *TipusMaterial*, ja que hi pot haver diverses naus dedicades a un tipus de material, però tot el material similar d'un venedor es troba en una nau determinada. Al mateix temps, hi pot haver diverses naus amb material d'un venedor a causa de la diferent tipologia del material.

- Cap atribut no pertanyent a la clau (*Nau* i *Ubicacio*) no depèn transitivament de la clau. En efecte, és impossible que hi hagi cap dependència transitiva de la clau, ja que no hi ha cap atribut que pugui servir de pont per a la transitivitat. Però aquesta relació, tot i estar en 3FN, presenta anomalies:
  - Si en un moment donat una nau no té material de cap venedor, es perd la informació referent al tipus de mercaderia que correspon a la nau.
  - Si canvia la descripció del tipus de mercaderia assignada a una nau,

cal modificar tantes files com venedors amb dipòsits d'aquell tipus de mercaderia hi hagi a la nau.

Una relació està en la **forma normal de Boyce-Codd (FNBC)** si està en 2FN i tots els seus determinants són claus candidates.

L'anterior relació *DIPOSIT* no es troba en FNBC, ja que l'atribut *Nau* és un determinant de la relació i *TipusMaterial* té dependència funcional total de *Nau*, i en canvi *Nau* no és clau candidata.

És a dir, es dona aquesta situació:

*Nau* → *TipusMaterial*

Es verifica que tota relació en FNBC està en 3FN però no a l'inrevés, com hem pogut comprovar en el nostre cas.

El procés que s'ha de seguir per assolir una FNBC és apartar de la relació els atributs que depenen dels determinants que no són claus candidates, i formar noves relacions que recullen els atributs apartats i que preserven la informació inicial.

En el nostre cas, apartarem de la relació l'atribut *TipusMaterial* i obtindrem les relacions en FNBC:

1	DIPOSIT ( __venedor__, nau, ubicacio)
2	nauTipus
3	NAU_TIPUS ( __nau__, tipusMaterial)

En els trencaments efectuats sobre una relació no normalitzada per assolir relacions 2FN i 3FN, cal efectuar la divisió de manera que es preservin la informació i les dependències funcionals, fet sempre possible en el pas a 2FN i 3FN. En el pas a FNBC, també sempre és possible efectuar la divisió mantenint la informació, però no sempre és possible el manteniment de les dependències funcionals.

En el nostre cas, la relació inicial *DIPOSIT* contenia les dependències funcionals següents:

*Venedor, TipusMaterial* → *Nau, Ubicacio*

*Nau* → *TipusMaterial*

I, en les relacions finals *DIPOSIT* i *NAU\_TIPUS*, s'ha perdut la dependència funcional que indicava que la *Nau* depenia de la parella *Venedor* i *TipusMaterial*:

*DIPOSIT*: *Venedor* → *Nau, Ubicacio*

*NAU\_TIPUS*: *Nau* → *TipusMaterial*

La dependència funcional s'ha perdut perquè el concepte de dependència funcional en el model relacional es defineix únicament entre atributs d'una mateixa

relació, i en aquest s'hauria de poder definir entre atributs de relacions diferents. De tota manera, el concepte d'integritat referencial intenta superar aquesta limitació.

A causa de la pèrdua de dependències funcionals, que no sempre es produeix, sovint no es normalitza a FNBC i es treballa amb relacions en 3FN.

## 2.8 Quarta forma normal

Considerem la relació *ESTUDIANT* (taula 2.13) dissenyada per emmagatzemar els diversos crèdits que cursa i les diverses activitats esportives que practica.

1 ESTUDIANT(\_\_Dni\_\_, \_\_Credit\_\_, \_\_Esport\_\_)

És a dir, la relació estudiant recull la possibilitat que un estudiant cursi diversos crèdits i practiqui diverses activitats esportives.

**TAULA 2.13.** Relació FNBC amb redundància

ESTUDIANT		
Dni	Credit	Esport
10.000.000	SGBD	Bàsquet
10.000.000	ADBD	Bàsquet
10.000.000	SGBD	Futbol
10.000.000	ADBD	Futbol
20.000.000	PEM	Natació
20.000.000	ADBD	Natació
20.000.000	SGBD	Natació
20.000.000	PEM	Esgrima
20.000.000	ADBD	Esgrima
20.000.000	SGBD	Esgrima
15.000.000	PEM	Natació
15.000.000	PEM	Bàsquet

Aquesta relació es troba en FNBC i, tot i així, hi ha redundància, provocada per un nou concepte: les dependències multivalents.

Donats A i B atributs o conjunts d'atributs d'una relació, direm que B té una **dependència multivalent** de A si un valor de A pot determinar un conjunt de valors de B. Ho simbolitzarem amb la notació  $A \twoheadrightarrow B$ .

Les dependències multivalents no són dependències funcionals. En canvi, però, una dependència funcional es pot arribar a considerar una dependència multivalent en què per cada valor de l'implicant hi ha un únic valor de l'implicat.

Els problemes provocats per les dependències funcionals han causat la definició de la 1FN, 2FN, 3FN i FNBC. La redundància provocada per les dependències multivalents ens porten a definir la 4FN.

Una relació es troba en **quarta forma normal (4FN)** si està en 3FN i l'implicant de tota dependència multivalent és una clau candidata.

Quan té lloc una dependència multivalent  $A \twoheadrightarrow B$ , també existeix la dependència multivalent  $A \twoheadrightarrow X - (A \cup B)$ , on  $X$  indica el conjunt de tots els atributs de la relació. És a dir, les dependències multivalents es presenten per parelles.

En el nostre cas (taula 2.13) es verifica el següent:

$Dni \twoheadrightarrow Credit$

$Dni \twoheadrightarrow Esport$

Per assolir la 4FN a partir d'una relació  $R(\underline{A}, B, C)$  que té una dependència multivalent  $A \twoheadrightarrow B$ , cal descompondre la relació  $R$  en dues relacions  $R1(\underline{A}, B)$  i  $R2(\underline{A}, C)$ .

**TAULA 2.14.** Relació en 4FN

CREDIT_EN_CURS	
Dni	Credit
10.000.000	SGBD
10.000.000	ADBD
20.000.000	PEM
20.000.000	ADBD
20.000.000	SGBD
15.000.000	PEM

**TAULA 2.15.** Relació en 4FN

ESPORT_EN_PRACTICA	
Dni	Esport
10.000.000	Bàsquet
10.000.000	Futbol
20.000.000	Natació
20.000.000	Esgrima
15.000.000	Natació
15.000.000	Bàsquet

En aquest cas, obtenim les relacions següents (taula 2.14 i taula 2.15):

- 1 CREDIT\_EN\_CURS (\_\_Dni, Credit\_\_)
- 2 ESPORT\_EN\_PRACTICA (\_\_Dni, Esport\_\_)

Sovint, la descomposició causada per les dependències multivalents s'efectua abans de les descomposicions per assolir els nivells 2FN, 3FN i FNBC. En aquesta situació, en les relacions obtingudes, cal aplicar les comprovacions per aconseguir que estiguin en 2FN, 3FN i FNBC.

## 2.9 Cinquena forma normal

Considerem la relació *PROFESSOR* (taula 2.16) dissenyada per gestionar els professors d'una determinada institució escolar que té diferents centres de docència. Cada professor està autoritzat a impartir unes determinades especialitats docents que pot posar en pràctica en qualsevol dels centres docents de la institució escolar. Així mateix, cada professor pot exercir, a més de la docència, diferents tasques (càrrecs, tutoria pedagògica, tutoria tècnica...) en diversos centres de la institució escolar.

1 PROFESSOR (\_\_CodiProf, Centre, Especialitat, Tasca\_\_)

**TAULA 2.16.** Relació en FNBC amb dependències multivalents

PROFESSOR			
CodiProf	Centre	Especialitat	Tasca
P1	C1	Matemàtiques	Tutor
P1	C2	Matemàtiques	Tutor
P1	C2	Informàtica	Aula Informàtica
P2	C1	Català	Coordinador
P2	C2	Castellà	Tutor

Aquesta relació és FNBC i s'hi aprecia una espècie de dependència multivalent, la qual no es pot solucionar per la via de la descomposició. En efecte, és molt fàcil pensar en una descomposició en les tres relacions següents (exemplificades en la taula 2.17, taula 2.18 i taula 2.19):

1 CENTRE\_DE\_PROFESSOR (\_\_CodiProf, Centre\_\_)  
 2 ESPECIALITAT\_DE\_PROFESSOR (\_\_CodiProf, Especialitat\_\_)  
 3 TASCA\_DE\_PROFESSOR (\_\_CodiProf, Tasca\_\_)

**TAULA 2.17.** Relació en 4FN

CENTRE_DE_PROFESSOR	
CodiProf	Centre
P1	C1
P1	C2
P2	C1
P2	C2

TAULA 2.18. Relació en 4FN

ESPECIALITAT_DE_PROFESSOR	
CodiProf	Especialitat
P1	Matemàtiques
P1	Informàtica
P2	Català
P2	Castellà

TAULA 2.19. Relació en 4FN

TASCA_DE_PROFESSOR	
CodiProf	Tasca
P1	Tutor
P1	Aula informàtica
P2	Coordinador
P2	Tutor

Aquesta descomposició és errònia, ja que si apliquem el *natural-join* de les tres relacions (taula 2.17, taula 2.18 i taula 2.19) no obtenim la relació inicial (taula 2.16) sinó que obtenim una relació (taula 2.20) amb moltes més instàncies.

TAULA 2.20. Relació obtinguda del “natural-join” de les relacions de la :table:Taula36:, :table:Taula37: i :table:Taula38:

PROFESSOR			
CodiProf	Centre	Especialitat	Tasca
P1	C1	Matemàtiques	Tutor
P1	C1	Matemàtiques	Aula informàtica
P1	C1	Informàtica	Tutor
P1	C1	Informàtica	Aula informàtica
P1	C2	Matemàtiques	Tutor
P1	C2	Matemàtiques	Aula informàtica
P1	C2	Informàtica	Tutor
P1	C2	Informàtica	Aula informàtica
P2	C1	Català	Coordinador
P2	C1	Català	Tutor
P2	C1	Castellà	Coordinador
P2	C1	Castellà	Tutor
P2	C2	Català	Coordinador
P2	C2	Català	Tutor
P2	C2	Castellà	Coordinador
P2	C2	Castellà	Tutor

Queda clar, doncs, que el mètode utilitzat en aquest cas no és correcte i això és degut al fet que, en aquesta situació, hi ha el que s'anomena **dependències mútues** entre els atributs de la relació. Les dependències mútues provoquen que la descomposició de la relació en altres relacions (projeccions de l'original) no verifiqui que el seu *natural-join* coincideix amb la relació original.



Direm que una relació  $R$  descomposta en relacions  $R_1, R_2, \dots, R_n$  satisfà una **dependència de reunió**, també anomenada **dependència de projecció-join**, respecte a  $R_1, R_2, \dots, R_n$  únicament si  $R$  és igual al *natural-join* de  $R_1, R_2, \dots, R_n$ . La notarem com a  $DR^*(R_1, R_2, \dots, R_n)$ .

Tornem a l'exemple de descomposició anterior: la relació *PROFESSOR* s'ha descompost en tres relacions, *CENTRE\_DE\_PROFESSOR*, *ESPECIALITAT\_DE\_PROFESSOR* i *TASCA\_DE\_PROFESSOR*, i hem pogut comprovar que la relació *professor* no satisfà una dependència de reunió respecte a *CENTRE\_DE\_PROFESSOR*, *ESPECIALITAT\_DE\_PROFESSOR* i *TASCA\_DE\_PROFESSOR*.

Ens cal trobar una dependència de reunió per a la relació *PROFESSOR*, és a dir, trobar una descomposició tal que el seu *natural-join* recuperi la relació original. Fixem-nos en la descomposició següent (taula 2.21, taula 2.22 i taula 2.23):

```

1 PROFESSOR ( __CodiProf, Centre, Especialitat, Tasca__ )
2 PCE = PROFESSOR [ __CodiProf, Centre, Especialitat__ ]
3 PCT = PROFESSOR [ __CodiProf, Centre, Tasca__ ]
4 PET = PROFESSOR [ __CodiProf, Especialitat, Tasca__ ]

```

On PCE, PCT i PET són seccions de la taula original.

**TAULA 2.21.** Relació 4FN

PCE		
CodiProf	Centre	Especialitat
P1	C1	Matemàtiques
P1	C2	Matemàtiques
P1	C2	Informàtica
P2	C1	Català
P2	C2	Castellà

**TAULA 2.22.** Relació 4FN :table:Taula42:. Relació 4FN

PCT		
CodiProf	Centre	Tasca
P1	C1	Tutor
P1	C2	Tutor
P1	C2	Aula informàtica
P2	C1	Coordinador
P2	C2	Tutor

TAULA 2.23. Relació 4FN

PET		
Professor	Especialitat	Tasca
P1	Matemàtiques	Tutor
P1	Informàtica	Aula informàtica
P2	Català	Coordinador
P2	Castellà	Tutor

En aquesta situació, veiem que si efectuem el *natural-join* de les tres relacions *PCE*, *PCT* i *PET* obtenim la relació original.

Direm que una relació està en **cinquena forma normal (5FN)**, també anomenada **forma normal projecció-join (FNPJ)**, si està en 4FN i tota dependència de reunió és conseqüència de claus candidates.

Per tant, la relació *PROFESSOR* del nostre exemple no es troba en 5FN, ja que hem trobat la dependència de reunió  $PROFESSOR^*(PCE, PCT, PET)$  en què les relacions *PCE*, *PCT* i *PET* no estan constituïdes per claus candidates de *PROFESSOR*.

Una relació 4FN que no sigui 5FN a causa d'una dependència de reunió es pot descompondre, sense pèrdua d'informació, en les relacions sobre les quals es defineix la dependència de reunió, les quals estan en 5FN.

Així, en el nostre exemple, la relació *PROFESSOR* desapareixeria per donar pas a les tres relacions *PCE*, *PCT* i *PET* en què es basa la dependència de reunió trobada.

- 1 PCE ( \_\_CodiProf, Centre, Especialitat\_\_ )
- 2 PCT ( \_\_CodiProf, Centre, Tasca\_\_ )
- 3 PET ( \_\_CodiProf, Especialitat, Tasca\_\_ )

## 2.10 Desnormalització

Pot semblar estrany plantejar-se la desnormalització d'una base de dades, després d'argumentar substancialment la importància que té disposar de bases de dades normalitzades, però en alguns casos una desnormalització controlada pot ser molt útil i, fins i tot, desitjable.

La desnormalització és pot definir com la introducció de redundàncies de forma controlada en una bases de dades, per tal de fer més eficients alguns processos que, altrament, farien que globalment el rendiment del sistema resultés poc òptim.

Tot i que per a un sistema donat, es podrien preveure certes modificacions sobre la BD per tal de millorar el rendiment una vegada en funcionament, el més típic és detectar aquestes necessitats a posteriori. Així, doncs, aquests tipus de modificacions sobre la base de dades es duen a terme, habitualment, després d'haver observat certes ineficiències en algunes operacions habituals sobre una base de dades.

A continuació es descriuen algunes situacions, a mode d'exemple, on pot ser útil la desnormalització:

- En BD on hi ha consultes intensives sobre dades d'una taula que tenen referenciades altres taules on s'hi emmagatzemen descripcions, en algun cas pot ser útil mantenir la descripció en la mateixa taula original, per tal de fer més ràpida la consulta mencionada. Pensem, per exemple, en una taula d'adreces d'empleats que té el codi de la província i que fa referència a una altra taula que té la descripció de la província. Cada cop que calgui consultar la província dels empleats caldrà processar l'operació de join de les dues taules. En canvi, si es disposa del nom de la província en la taula d'adreces, s'evitarà aquesta operació que resulta costosa.
- Afegir camps que són calculables, com ara el total d'una factura, pot reduir, també, el temps de consulta de dades d'aquesta taula, per exemple.

#### Trigger

Un **trigger** és un procediment de BD que s'executa automàticament quan s'esdevenen situacions donades. Per exemple, inserció d'un nou registre en una taula, modificació d'una dada en un camp donat, d'una taula, etc.

La introducció de redundàncies que desnormalitzin la BD habitualment va lligat a la implementació de certs mecanismes que intenten solucionar els possibles problemes que es podrien generar d'aquestes accions. La forma més típica de controlar els canvis per a evitar inconsistències en en aquests casos és la programació de triggers. Així doncs, es pot programar un trigger, per exemple, perquè en inserir, modificar o eliminar una línia de factura, es recalculi i actualitzi l'import del total de la factura, per així tenir aquest camp calculat sempre consistent.