

---

# Disseny lògic de bases de dades. Transformació a relacional a partir del model ER

---

PID\_00267038

Ignasi Lorente Puchades  
Manel Díaz Llobet  
Jaume Sistac Planas  
Elena Rodríguez González

---

Temps mínim de dedicació recomanat: 5 hores

---



**Ignasi Lorente Puchades**

Enginyer superior en Informàtica per la Universitat Oberta de Catalunya (UOC). Exerceix de cap de projectes d'aplicacions web en l'àmbit públic i privat. Professor col·laborador del Grau Multimèdia a la Universitat Oberta de Catalunya (UOC) i professor associat a la Universitat Pompeu Fabra (UPF).

**Manel Díaz Llobet**

Enginyer superior en informàtica per la UPC. Treballa com a professor de desenvolupament d'aplicacions, de llocs web, sistemes operatius, xarxes i bases de dades des de l'any 2000. Ha realitzat diferents projectes de gestió per a l'empresa privada i ha intervingut en projectes d'intel·ligència artificial en l'àmbit de la gestió editorial per a mitjans de comunicació.

**Jaume Sistac Planas****Elena Rodríguez González**

L'encàrrec i la creació d'aquest recurs d'aprenentatge UOC han estat coordinats per la professora: Àngels Rius Gavidia (2019)

Segona edició: setembre 2019

© Ignasi Lorente Puchades, Manel Díaz Llobet, Jaume Sistac Planas, Elena Rodríguez González

Tots els drets reservats

© d'aquesta edició, FUOC, 2019

Av. Tibidabo, 39-43, 08035 Barcelona

Realització editorial: FUOC

*Cap part d'aquesta publicació, incloent-hi el disseny general i la coberta, no pot ser copiada, reproduïda, emmagatzemada o transmesa de cap manera ni per cap mitjà, tant si és elèctric com químic, mecànic, òptic, de gravació, de fotocòpia o per altres mètodes, sense l'autorització prèvia per escrit dels titulars dels drets.*

# Índex

<b>Introducció</b> .....	5
<b>Objectius</b> .....	6
<b>1. El model relacional</b> .....	7
1.1. Claus d'una relació .....	9
1.2. Claus foranes d'una relació .....	11
<b>2. Transformació del model entitat-relació en el model relacional</b> .....	13
2.1. Interrelacions binàries .....	14
2.1.1. Interrelacions binàries amb connectivitat 1:1 .....	14
2.1.2. Interrelacions binàries amb connectivitat 1:N .....	14
2.1.3. Interrelacions binàries amb connectivitat N:M .....	15
2.2. Interrelacions ternàries i n-àries .....	16
2.2.1. Interrelacions ternàries de connectivitat N:M:P .....	16
2.2.2. Interrelacions ternàries de connectivitat N:M:1 .....	17
2.2.3. Interrelacions ternàries de connectivitat N:1:1 .....	17
2.2.4. Interrelacions ternàries de connectivitat 1:1:1 .....	19
2.2.5. Interrelacions n-àries .....	20
2.3. Transformació d'entitats dèbils .....	21
2.4. Transformació d'entitats associatives .....	22
2.5. Resum del procés de transformació .....	22
2.6. Exemple de transformació de model conceptual (ER) en model relacional .....	23
<b>3. Teoria de la normalització</b> .....	26
3.1. Anomalies de disseny .....	27
3.1.1. Anomalies de modificació .....	27
3.1.2. Anomalies d'esborrament .....	28
3.1.3. Anomalies d'inserció .....	28
3.2. El procés de normalització .....	30
3.2.1. Dependències funcionals i dependències funcionals completes (o plenes) .....	30
3.2.2. Primera forma normal .....	33
3.2.3. Segona forma normal .....	34
3.2.4. Tercera forma normal .....	36
3.2.5. Forma normal de Boyce-Codd .....	36
3.2.6. Conclusions sobre dependències funcionals en les formes normals .....	40

3.3. Aplicació de la teoria de la normalització al disseny de bases de dades relacionals .....	41
3.4. Exemple de normalització .....	43
<b>Resum</b> .....	48
<b>Activitats</b> .....	49
<b>Glossari</b> .....	56

## Introducció

En el mòdul «Disseny conceptual de bases de dades» hem après a crear un model conceptual de dades a partir de la descripció d'un problema plantejat. Hem utilitzat el diagrama entitat-relació (ER) per descriure les dades que són rellevants i com es relacionen entre si. En aquest mòdul tractarem la segona etapa del procés de disseny d'una base de dades –és a dir, la del disseny lògic– i a partir del model conceptual veurem com obtenir el lògic.

El model lògic permetrà descriure formalment l'estructura de dades que s'implementarà en l'etapa següent, la del disseny físic. A més, aquesta solució serà independent de la tecnologia i del sistema de bases de dades que puguem emprar per gestionar la base de dades. No obstant això, en aquesta fase caldrà escollir el model de dades. Nosaltres triarem el model relacional perquè resulta molt senzill fer la traducció a relacional des del diagrama ER i permet obtenir dissenys de qualitat en aplicar la teoria de la normalització.

En aquest mòdul estudiarem els elements que constitueixen el model de dades relacional, en particular aquells que són rellevants per a la transformació del model ER a relacional, com les claus i l'estructura de les relacions. A més, veurem com sistematitzar aquesta traducció i obtenir així un primer model lògic, que podrà ser millorat posteriorment amb la teoria de la normalització amb l'aplicació d'una sèrie de regles per tal d'obtenir un model lògic sense redundàncies de dades ni anomalies de disseny.

## Objectius

En els materials didàctics d'aquest mòdul trobareu les eines indispensables per assolir els objectius següents:

- 1.** Conèixer en què consisteix el model relacional i quins són els elements que ofereix per a l'estructuració i l'organització de dades.
- 2.** Conèixer els mecanismes de transformació del diagrama entitat-relació (ER) a relacional tot utilitzant els elements propis del model relacional.
- 3.** Identificar els diferents fets semàntics d'un model lògic i, amb l'aplicació de les regles de la teoria de la normalització, obtenir un model lògic sense anomalies de disseny.

## 1. El model relacional

En aquest mòdul estudiarem el model relacional, que és el model de bases de dades segons el qual s'organitzen les bases de dades relacionals. Aquest model ens serà útil per a representar el model lògic corresponent al model conceptual obtingut en l'etapa anterior.

Un model de dades està format per tres elements:

- 1) una estructura per emmagatzemar les dades,
- 2) un conjunt d'operacions per manipular-les,
- 3) un conjunt de regles o restriccions inherents al model.

Explicarem en què consisteix el model relacional sobre la base d'aquests elements.

L'estructura d'emmagatzematge que utilitza el model relacional és la relació, coneguda també com a taula. Cada taula permet emmagatzemar informació sobre un concepte i cada ocurrència o instància concreta d'aquest concepte (dada) es representa mitjançant una tupla, també anomenada registre o fila. Una tupla d'una relació té una estructura i es compon d'un conjunt d'atributs o columnes definits sobre un conjunt de dades. Així doncs, l'agrupació d'un conjunt de tuples amb una mateixa estructura, dona lloc a una relació.

Començarem a estudiar, de forma més detallada i formal, l'estructura de dades que suporta el model relacional des de l'element més petit que la compon, el tipus de dades de les columnes de la taula o relació.

Com hem vist en el mòdul «Disseny conceptual», les dades per representar són definides sobre un tipus de dades, que determina el conjunt de valors possibles que aquest pot prendre.

Un **domini**  $D$  és un conjunt de valors atòmics, indivisibles, que no es pot representar com a combinacions d'altres valors dins d'un SGDB relacional.

### Domini definit per l'usuari

Per exemple, es pot definir un domini per als diferents colors que pot tenir una capsa de llapis de colors, els colls d'una baralla de cartes, etc.

En el model relacional trobarem dos tipus de dominis:

1) **Dominis predefinitos**, corresponents als tipus de dades bàsics com els enters, les cadenes de caràcters, els booleans, les dates, etc.

2) **Dominis definits per l'usuari**, corresponents als tipus de dades específics de cada problema del món real que es vol representar

És important adonar-se que no es poden confondre les relacions (conjunts de dades agrupades) del model relacional amb les interrelacions (associacions entre entitats) que hem vist en l'estudi del model conceptual.

### Atenció!

El domini de dades de cada atribut també forma part de la descripció de l'estructura de la relació o taula però, per fer-ho més senzill, l'ometrem al representar les relacions sigui en forma textual o tabular.

Una **relació** es l'agrupació d'un conjunt de dades dins del model relacional, la qual està formada per l'**esquema** o **intensió** i l'**extensió**.

L'esquema d'una relació el podem entendre com el patró de les dades que volem emmagatzemar, mentre que la intensió cal entendre-la com les dades concretes que segueixen aquest patró.

Per exemple, si considerem el cas de les persones, de les quals volem conèixer el seu DNI, nom, cognoms i l'adreça de correu electrònic, les ocurrències de persona en el model relacional es podrien representar així:

PERSONA			
DNI	nom	cognoms	correuElectronic
00000000T	Ramon	Cosina Riera	ramon@exemple.com
11111111V	Verònica	Martí Roura	verònica@exemple.com

Intensió

Extensió

Així doncs, formalment podem definir l'**esquema de la relació** o **intensió** com el nom de la relació  $R$  i un conjunt d'atributs  $A_1, A_2, \dots, A_n$  i es representarà com  $R\{A_1, A_2, \dots, A_n\}$ .

En l'exemple anterior, el nom de la relació serà PERSONA i el conjunt d'atributs serà DNI, nom, cognoms, correuElectronic i es representarà formalment com:

PERSONA(DNI, nom, cognoms, correuElectronic)



L'**extensió de la relació** de l'esquema és un conjunt de tuples  $t_i$  en què els valors que es contenen són propis del domini o bé prenen un valor NULL i es representa com:

$$t_i = \langle v_{i1}, v_{i2}, \dots, v_{in} \rangle.$$

Seguint l'exemple de la relació PERSONA, una possible tupla seria:

$$t_1 = \langle 34901276T, Ramon, Cosina Riera, ramon@exemple.com \rangle$$

Podria succeir que una persona no tingui correu electrònic, de manera que el valor que tindria assignat en la tupla seria el valor nul:

$$t_1 = \langle 34901276T, Ramon, Cosina Riera, NULL \rangle$$

El **grau** d'una relació és el nombre d'atributs del seu esquema.

En el cas de la relació PERSONA, el seu grau serà 4.

La **cardinalitat** de la relació és el nombre de tuples que pertanyen a la seva extensió.

En el cas de la relació PERSONA, la seva cardinalitat serà 2.

El fet que el model relacional estigui basat en la teoria de conjunts ens permetrà realitzar una sèrie d'operacions sobre les relacions que facilitarà la recuperació de dades independentment de la seva ubicació.

Aquestes operacions sovint requeriran associar i recuperar dades que es troben en diferents relacions. Per poder-les referenciar de manera adequada necessitarem identificar de manera unívoca les tuples de cada relació i, per aquest motiu, serà necessari definir el concepte de *clau de la relació*.

### 1.1. Claus d'una relació

És important poder identificar de manera unívoca cada tupla de la bases de dades. La identificació de la informació no només ens servirà per recuperar únicament una dada en particular, sinó que ens permetrà relacionar les dades amb independència de la seva ubicació física. Aquest és un dels fonaments del model relacional.

#### El valor NULL

Considerarem el valor NULL com una marca que permet indicar que un valor, que és desconegut o no, pren un valor dins del domini.

#### Vegeu també

Per saber-ne més, podeu consultar l'apartat 4 d'aquest mòdul.

#### Vegeu també

Per saber més sobre aquestes claus, podeu consultar el mòdul «Disseny conceptual».

A causa de la importància de les claus en el model relacional, serà necessari establir un conjunt de claus candidates que ens permetin assolir aquest objectiu. En el model relacional introduïrem el concepte de *superclau*.

Una **superclau** d'una relació definida per l'esquema  $R\{A_1, A_2, \dots, A_n\}$  és un conjunt d'atributs per als quals no hi ha dues tuples diferents dins de la relació que tinguin els mateixos valors pels atributs d'aquest conjunt.

#### Superclau

Tota relació té, com a mínim, una superclau que està formada per tots els atributs del seu esquema.

A més a més, la resta d'atributs de l'esquema de relació són funcionalment dependents d'aquest conjunt d'atributs.

La **dependència funcional** entre dos atributs  $a$  i  $b$  ( $a \rightarrow b$ ) implica que, si es coneix el valor de l'atribut  $a$ , es pot inferir quin serà el valor de  $b$  dintre de l'extensió de la relació.

En una relació, les superclaus ens serviran per identificar les tuples d'una relació  $i$ , per aquest motiu, els altres atributs tindran una dependència funcional envers elles. A més a més, una superclau pot tenir més atributs dels estrictament necessaris per identificar una tupla. El nostre objectiu serà el de trobar el mínim conjunt possible d'atributs que ens permeti identificar una tupla.

Una **clau candidata** d'una relació és una superclau  $c$  de la relació en la qual cap dels seus subconjunts no és superclau de la relació.

Per exemple, en la relació PERSONA que estem considerant, una superclau podria ser la formada pel total d'atributs de la relació:

*PERSONA (DNI, nom, cognoms, correuElectronic)*

Tot i això, plantejarem com a claus candidates les superclaus formades pels atributs {DNI} i {correuElectronic}.

La **clau primària**, CP (en anglès *PK, primary key*), d'una relació és la clau candidata escollida pel dissenyador durant el procés de disseny com a identificador de les tuples de la relació.

Una **clau alternativa** és una clau candidata que no ha estat escollida com a clau primària.

Per exemple, en la relació PERSONA, escollirem com a clau primària la clau candidata formada pel l'atribut {DNI}, de manera que la clau candidata formada per l'atribut {correuElectronic} el considerarem com una clau alternativa.

Com veurem més endavant, una tècnica habitual en el disseny del model relacional és afegir atributs numèrics que, tot i no tenir relació conceptual amb l'entitat que es vol representar, ens permetran identificar i manipular les tuples fàcilment.

Por tant, sempre trobarem una superclau de la relació i, en conseqüència, sempre existirà una clau candidata. La clau primària serà una de les claus candidates, en el cas que n'hi hagi més d'una, i sempre la podrem designar.

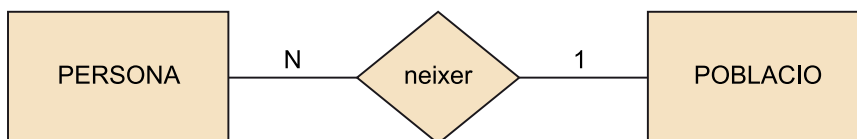
## 1.2. Claus foranes d'una relació

Les entitats s'associen entre elles mitjançant interrelacions. Aquestes interrelacions poden ser de diferent grau i tenir diferents tipus de connectivitat.

En el model relacional establirem aquestes associacions a partir de l'ús de claus foranes, que ens permetran fer referència a les altres relacions a partir de les seves claus primàries.

Una **clau forana**, CF (FK en anglès), és un conjunt d'atributs d'una relació que permet identificar unívocament una tupla d'una altra relació amb la qual s'associa, o d'ella mateixa, a partir de la seva clau primària.

Prenem per exemple el modelat de les poblacions on han nascut les persones, de manera que l'associació en el model conceptual deriva en una relació que tindrà una connectivitat 1:N.



En aquest cas tenim inicialment les relacions POBLACIO(*codi*, *nom*) i PERSONA(*DNI*, *nom*, *cognoms*, *correuElectronic*), en les quals es defineix les claus primàries {*codi*} i {*DNI*} per a cada relació.

Si volem establir una associació entre les dues relacions, afegirem un nou atribut {*poblacioNaixement*} a la relació PERSONA que faci referència a la clau primària de la relació POBLACIO i, per tant, es considerarà una clau forana.

D'aquesta manera, definirem les relacions de la manera següent:

POBLACIO(*codi*, *nom*)  
 PERSONA(*DNI*, *nom*, *cognoms*, *correuElectronic*, *poblacioNaixement*)  
 on {*poblacioNaixement*} referència POBLACIO (*codi*)

### Vegeu també

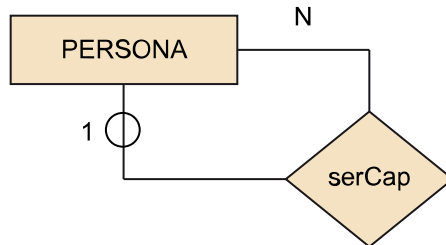
Per saber més sobre claus foranes, consulteu el mòdul «Disseny conceptual».

### Atenció!

Les relacions es podran relacionar amb d'altres o amb elles mateixes, tal com succeeix en el model conceptual, en el qual hem vist les interrelacions recursives en què una entitat s'associa amb ella mateixa.

En el cas d'associacions recursives podem establir, de la mateixa manera, un nou atribut a la relació que proporcioni la funcionalitat d'una clau forana envers la clau primària de la relació.

Per exemple, el cas d'una relació reflexiva entre persones d'una empresa que modela la relació serCap, la qual permet saber de quantes persones és cap una persona (1 o moltes) i quants caps te una persona (0 si és el càrrec més alt de l'empresa, o 1 en tots els altres casos) el representàrem així:



Així, definirem la relació PERSONA de la manera següent:

*PERSONA*(DNI, nom, cognoms, correuElectronic, DNICap)  
on {DNICap} referencia PERSONA

Tot i això, pot ser que una tupla d'una relació no tingui correspondència amb una tupla de la relació a què fa referència la seva clau forana. En aquest cas la clau forana es prendrà com un valor nul.

Per exemple, la clau forana podrà prendre valor nul quan una treballador no tingui cap o sigui el màxim responsable, ja que no haurà de fer referència a cap altra persona.

És important tenir en compte que tant el nombre d'atributs d'una clau forana com el domini de valors de cadascun dels seus atributs ha de coincidir exactament amb els dels atributs que formen la clau primària de la relació a la qual fa referència. Així doncs caldrà verificar els dominis sobre els quals estan definits els atributs referenciats per les claus foranes.

Tanmateix, es pot donar el cas que alguns atributs d'una clau forana d'una relació, o bé la seva totalitat, formin part de la seva clau primària. Aquest fet es pot donar habitualment en relacions amb entitats dèbils.

Considerem una relació que fa referència als viatges que ha fet una persona, en què una persona pot viatjar a més d'una població i, a més, una població pot ser visitada per més d'una persona.

Per representar aquest fet mitjançant una relació (VIATJAR), necessitarem una tupla diferent per a cada parella d'ocurrències de PERSONA-POBLACIÓ, per la qual cosa tots dos atributs es podrien considerar clau primària. A més, els valors d'aquests atributs ens permetran identificar les tuples associades en les relacions PERSONA i POBLACIÓ per separat.

Així doncs, podem considerar l'esquema de la relació VIATJAR com:

*VIATJAR*(codiPoblacio, DNI, data)  
on {codiPoblacio} referencia POBLACIO(codi)  
on {DNI} referencia PERSONA

### Atenció!

En definir una clau forana, indicarem explícitament el nom dels atributs als qual es fa referència quan no hi hagi coincidència de noms entre els atributs de la relació a la que referència.

### Nota

El cercle en un dels extrems de la interrelació indica opcionalitat.

## 2. Transformació del model entitat-relació en el model relacional

En els diferents exemples que hem vist al llarg d'aquest mòdul hem fet referència a la representació de les dades per resoldre un problema donada la seva representació conceptual en termes del model entitat-relació. Per tant, han servit per il·lustrar la transformació del model conceptual a model lògic.

En aquest apartat veurem tota la casuística possible i donarem pautes per fer aquesta transformació.

Comencem pels elements bàsics del model ER: les entitats, les interrelacions i els atributs.

Una **entitat** del model ER es transformarà en una relació en el model relacional.

Una **interrelació** del model ER es transformarà en una clau forana o bé en una nova relació.

En el cas de les interrelacions caldrà tenir en compte el seu grau i la seva connectivitat.

Donat un diagrama ER, el procediment que farem servir per obtenir el corresponent model lògic serà el següent:

- 1) Es transformaran totes les entitats en relacions del model relacional.
- 2) Els atributs de l'entitat es consideraran atributs de la relació i la clau primària escollida serà la clau primària de la relació.
- 3) Les interrelacions binàries 1:1 i 1:N donaran lloc a claus foranes, de manera que s'hauran d'afegir nous atributs a la relació.
- 4) Les interrelacions binàries N:M i totes les n-àries donaran lloc a noves relacions

Tot seguit veurem amb més detall com es transformaran les interrelacions del model entitat-relació en el model relacional.

## 2.1. Interrelacions binàries

En aquest subapartat tractarem les relacions binàries, que ens serviran de base a l'hora de transformar la resta d'interrelacions amb més entitats (n-àries).

### 2.1.1. Interrelacions binàries amb connectivitat 1:1

Les interrelacions binàries amb connectivitat 1:1 es transformaran en dues relacions, una per cada entitat, en què cada una de elles tindrà una clau forana que referencii l'altra relació.

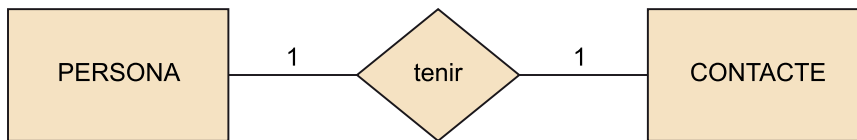
En aquest cas durant el disseny s'haurà de decidir quina de les entitats és la que tindrà la clau forana.

S'ha de tenir en compte en aquest tipus d'interrelacions els nous atributs que s'afegeixen per indicar les claus foranes, també seran claus candidates de la relació.

En el cas que una de les entitats sigui opcional, és a dir, que pugui participar o no en la interrelació, la clau forana podrà prendre valors nuls.

Pensem en una agenda en la qual per cada persona guardarem el seu DNI, nom i cognoms i, a més, tindrà associada una fitxa de contacte, en què es guardarà el seu telèfon, correu electrònic, adreça postal, etc. No hi ha persones ni contactes repetits.

En el model conceptual es representarà de la manera següent.



D'aquesta manera, tindrem dues possibles opcions de transformació. En primer lloc l'opció en què la clau forana forma part de la relació CONTACTE.

*PERSONA(DNI, nom, cognoms)*  
*CONTACTE(codi, telefon, correuElectronic, adreça, DNI)*  
 on {codi} referencia PERSONA, és únic i no pot ser Null.

En la segona opció la clau forana forma part de la relació PERSONA.

*PERSONA(DNI, nom, cognoms, codi)*  
 on {codi} referencia CONTACTE i és únic  
*CONTACTE(codi, telefon, correuElectronic, adreça)*

#### Atenció!

Si l'atribut DNI de la relació CONTACTE no és únic, podria passar que una persona (un DNI) tingués més d'un contacte (codi) i es pogués donar una situació com aquesta:  
*CONTACTE (codi, ....., DNI)*  
*C01, ....., 52111111A*  
*C02, ....., 52111111A*

#### Aclariment

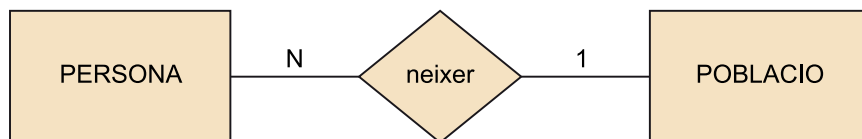
Com en el cas anterior, l'atribut *codi* de PERSONA hauria de ser únic per garantir que no existeixen dos contactes (codi) d'una mateixa PERSONA (DNI) i no null. Si pogués ser Null aleshores la interrelació 'tenir' tindria un zero, indicant opcionalitat, en un dels extrems 1.

### 2.1.2. Interrelacions binàries amb connectivitat 1:N

De manera similar a les interrelacions binàries amb connectivitat 1:1, les interrelacions binàries amb connectivitat 1:N es transformaran en dues relacions, una per cada entitat, en la qual la relació del costat N tindrà una clau forana que referencii l'altra relació del costat 1.

Suposem que volem saber en quines poblacions ha nascut cadascuna de les persones que consten en la base de dades.

Com que cada persona ha nascut en un sol lloc, en el model conceptual ho representarem com una associació entre dues entitats amb connectivitat 1:N, tal com indiquem a continuació:



La transformació a relacional correcta serà:

*POBLACIO(codi, nom)*  
*PERSONA(DNI, nom, cognoms, correuElectronic, poblacioNaixement)*  
 on {poblacioNaixement} referencia POBLACIO(codi)  
 on {DNI} és únic

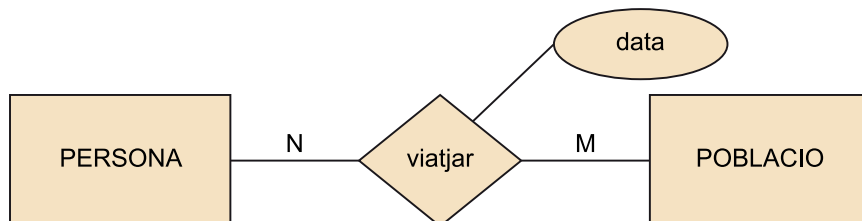
Si declaréssim la clau forana a la relació POBLACIO, ens trobaríem que cada persona podria tenir més d'una ciutat de naixement i en una ciutat només hi podria haver nascut una persona, per la qual cosa aquesta solució no és correcta.

### 2.1.3. Interrelacions binàries amb connectivitat N:M

Les interrelacions binàries amb connectivitat N:M es transformaran en tres relacions, una per cada entitat i una nova relació formada pels atributs de la clau primària de cadascuna de les dues entitats que participen en la interrelació.

En cas que les interrelacions tinguin atributs, aquests formaran part de la nova relació creada.

Si considerem els viatges fets per una persona i entenem que una persona pot viatjar a més d'un lloc (M al costat POBLACIO) i que una mateixa població pot ser visitada per diverses persones (N al costat PERSONA) la seva representació binària N-M seria aquesta:



La transformació en model relacional correcta serà:

*POBLACIO(codi, nom)*  
*PERSONA(DNI, nom, cognoms, correuElectronic)*  
*VIATJAR(codi, DNI, data)*  
 on {codi} referencia POBLACIO  
 on {DNI} referencia PERSONA

En aquest cas les claus foranes de la nova relació envers les relacions referides constituïran la clau primària de la nova relació viatjar. Serà possible, també, afegir un nou atribut que identifiqui de manera unívoca les tuples de la nova relació, de manera que el conjunt de les claus foranes passaran a ser una clau alternativa.

## 2.2. Interrelacions ternàries i n-àries

De manera similar a les interrelacions binàries de connectivitat N:M, la transformació d'una interrelació ternària o n-ària donarà sempre lloc a una nova relació, la qual tindrà com a atributs les claus primàries de les entitats interrelacionades i tots els atributs que tingui la interrelació.

La clau primària de la nova relació, si no es crea un nou atribut identificador, serà el conjunt o subconjunt de les claus foranes de les entitats interrelacionades segons sigui la connectivitat de la interrelació.

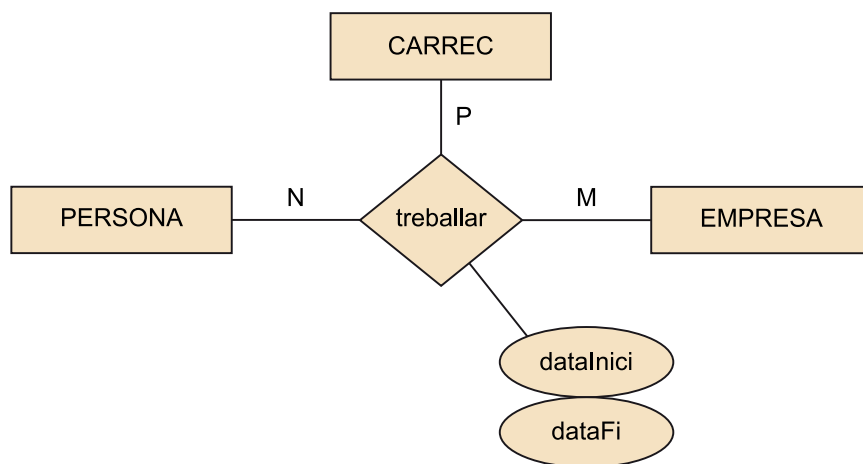
### 2.2.1. Interrelacions ternàries de connectivitat N:M:P

Les interrelacions ternàries de connectivitat N:M:P es transformaran en quatre relacions, una per cada entitat i una nova relació formada pels atributs de la clau primària de cadascuna de les tres entitats que participen en la interrelació.

En cas que les interrelacions tinguin atributs, aquests formaran part de la nova relació creada.

La nova relació tindrà com a clau primària tots els atributs que formen les claus primàries de les entitats associades per la interrelació.

Per exemple, si es vol disposar de la informació relativa a càrrecs ocupats per persones en empreses al llarg del temps, podríem representar-ho conceptualment mitjançant una associació ternària en què s'associen les entitats: PERSONA, CÀRREC i EMPRESA, de manera que els atributs de la interrelació siguin la data d'inici i la del final del càrrec.



La transformació en model relacional correcta serà:

```
PERSONA(DNI, nom, cognoms, correuElectronic)
EMPRESA(codi, nom, adreça)
CARREC(codi, nom)
TREBALLAR(DNI, codiEmpresa, codiCarrec, dataInici, dataFi)
on {codiEmpresa} referencia EMPRESA(codi)
on {codiCarrec} referencia CARREC(codi)
on {DNI} referencia PERSONA
```



### 2.2.2. Interrelacions ternàries de connectivitat N:M:1

Les interrelacions ternàries de connectivitat N:M:1 es transformaran en quatre relacions, una per cada entitat i una nova relació formada pels atributs de la clau primària de cadascuna de les tres entitats que participen en la interrelació.

En cas que les interrelacions tinguin atributs, aquests formaran part de la nova relació creada.

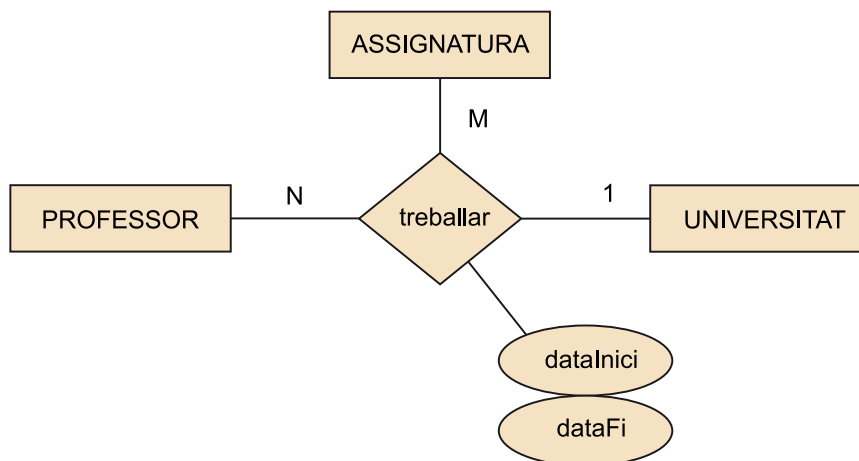
La nova relació tindrà com a clau primària tots els atributs que formen les claus primàries de les dues entitats associades per la interrelació i etiquetades amb M i amb N. Observeu que no es tenen en compte els atributs de la clau primària del costat 1.

Imaginem que volem saber quines assignatures imparteix cada professor en una universitat, com també el període de temps durant el qual les ha impartit o imparteix.

Si partim del fet que un professor tan sols pot impartir cada assignatura en una universitat, que en una universitat imparteixen assignatures molts professors i que una assignatura, en un mateix centre, la poden impartir diversos professors, en diferents moments, podríem representar-ho així:

#### Alerta!

Fixeu-vos que la clau primària del cantó 1 no forma part de la clau primària de la nova relació!



La transformació en model relacional serà:

*PROFESSOR*(DNI, nom, cognoms, correuElectronic)  
*UNIVERSITAT*(codi, nom, adreça)  
*ASSIGNATURA*(codi, nom)  
*TREBALLAR*(DNI, codiAssignatura, codiUniversitat, dataInici, dataFi)  
 on {codiUniversitat} referència *UNIVERSITAT*(codi)  
 on {codiAssignatura} referència *ASSIGNATURA*(codi)  
 on {DNI} referència *PROFESSOR*

A diferència del cas anterior, l'atribut *codiUniversitat* no serà part de la clau primària de la nova relació.

### 2.2.3. Interrelacions ternàries de connectivitat N:1:1

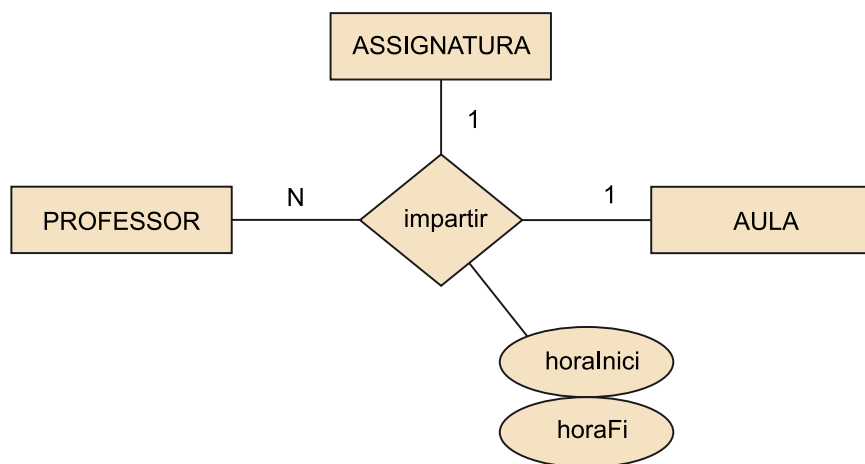
Les interrelacions ternàries de connectivitat N:1:1 es transformaran en quatre relacions, una per cada entitat i una nova relació formada pels atributs de la clau primària de cadascuna de les tres entitats que participen en la interrelació.

En cas que les interrelacions tinguin atributs, aquests formaran part de la nova relació creada.

La nova relació tindrà com a clau primària tots els atributs que formen la clau primària de l'entitat etiquetada amb *N* i els atributs que formen la clau primària d'una de les altres dues relacions.

D'aquesta manera obtindrem dues claus candidates i, per tant, hi haurà més d'una opció de transformació.

Considerem a continuació el modelat de la interrelació que s'estableix quan una assignatura pot ser impartida per més d'un professor, però només una vegada en una mateixa aula. Es representaria per mitjà del model conceptual de la manera següent:



En aquest cas, trobarem dues possibles solucions. Si bé les relacions que deriven de les entitats seran les mateixes:

*PROFESSOR*(DNI, nom, cognoms, correuElectronic)  
*AULA*(codi, aforament)  
*ASSIGNATURA*(codi, nom)

Trobarem dues possibles solucions a la representació de la interrelació:

*IMPARTIR*(DNI, codiAssignatura, codiAula, horaInici, horaFi)  
 on {codiAula} referencia *AULA*(codi)  
 on {codiAssignatura} referencia *ASSIGNATURA*(codi)  
 on {DNI} referencia *PROFESSOR*  
 i {DNI, codiAula} és únic

o bé

*IMPARTIR*(DNI, codiAssignatura, codiAula, horaInici, horaFi)  
 on {codiAula} referencia *AULA*(codi)  
 on {codiAssignatura} referencia *ASSIGNATURA*(codi)  
 on {DNI} referencia *PROFESSOR*  
 i {DNI, codiAula} és únic

D'aquesta manera, la clau primària estarà sempre formada per la clau forana envers l'entitat del costat *N* i una de les altres dues claus foranes. Observeu que {DNI, codiAssignatura} i {DNI, codiAula} pel fet de ser clau primària d'IMPARTIR identifiquen únivocament les tuples de la relació IMPARTIR.

### 2.2.4. Interrelacions ternàries de connectivitat 1:1:1

Les interrelacions ternàries de connectivitat 1:1:1 es transformaran en quatre relacions, una per cada entitat i una nova relació formada pels atributs de la clau primària de cadascuna de les tres entitats que participen en la interrelació.

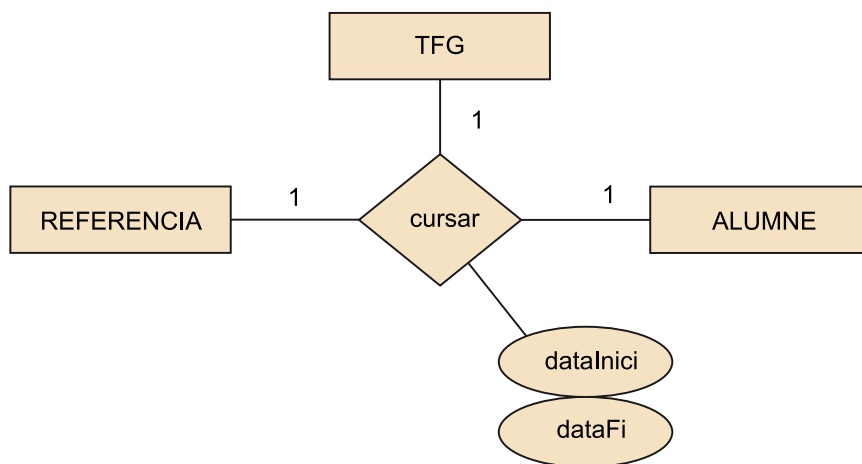
En cas que les interrelacions tinguin atributs, aquests formaran part de la nova relació creada.

La nova relació tindrà com a clau primària un únic atribut dels tres que formen, respectivament, la clau primària de les relacions que hi intervenen.

D'aquesta manera obtindrem tres claus candidates i, per tant, hi haurà més d'una opció.

Considerem finalment el modelat de la interrelació que hi ha entre el treball de final de grau (TFG) que realitza un alumne i l'entrada bibliogràfica que tindrà el treball a la biblioteca de la universitat, suposant que cada estudiant només podrà realitzar un TFG al llarg de la seva vida.

En el model conceptual es representarà de la manera següent:



En aquest cas, trobarem tres possibles solucions. Si bé les relacions que deriven de les entitats seran les mateixes:

*ALUMNE(DNI, nom, cognoms)*  
*TFG(codi, nom)*  
*REFERENCIA(codiEntrada, tematica, paraulesClau)*

Trobarem tres possibles solucions a la representació de la interrelació:

*CURSAR(codiEntrada, codiTFG, DNIALumne, dataInici, dataFi)*  
*on {DNIALumne} referencia ALUMNE(DNI) i és únic*  
*on {codiTFG} referencia TFG(codi) i és únic*  
*on {codiEntrada} referencia REFERENCIA i és únic*

o bé

*CURSAR(codiEntrada, codiTFG, DNIALumne, dataInici, dataFi)*  
*on {DNIALumne} referencia ALUMNE(DNI) i és únic*  
*on {codiTFG} referencia TFG(codi) i és únic*  
*on {codiEntrada} referencia REFERENCIA i és únic*

o bé

*CURSAR(codiEntrada, codiTFG, DNIAlumne, dataInici, dataFi)*  
*on {DNIAlumne} referencia ALUMNE(DNI) i és únic*  
*on {codiTFG} referencia TFG(codi) i és únic*  
*on {codiEntrada} referencia REFERENCIA i és únic*

D'aquesta manera, la clau primària estarà sempre formada per la combinació de dues de les claus foranes envers dues entitats de les que formen la interrelació.

En cas que volguéssim que un dels altres atributs de la interrelació formés part de la clau primària tindriem solucions anàlogues combinant aquest atribut amb algun dels altres.

### 2.2.5. Interrelacions n-àries

Les interrelacions n-àries són una generalització de les interrelacions ternàries i, per tant, es transformaran en  $n+1$  relacions, una per cada entitat i una nova relació formada pels atributs de la clau primària de cadascuna de les tres entitats que participen en la interrelació.

En el cas que totes les entitats estiguin connectades amb «molts», la nova relació tindrà com a clau primària tots els atributs que formen les claus primàries de les entitats associades per la interrelació.

En el cas que una o més entitats estiguin connectades amb «un», la nova relació tindrà com a clau primària tots els atributs que formen les claus primàries de les entitats associades per la interrelació dels costats «molts» i totes menys una de les entitats dels costats «un».

D'altra banda, com hem vist en el mòdul «Disseny conceptual», les interrelacions de grau superior a 3 sovint poden facilitar el manteniment d'un històric de valors que prengui al llarg del temps les instàncies de les entitats pròpies d'un món que volem representar.

Aquesta serà una necessitat que apareixerà habitualment en el disseny de les bases de dades i per aquest motiu és important saber representar-la en el model relacional.

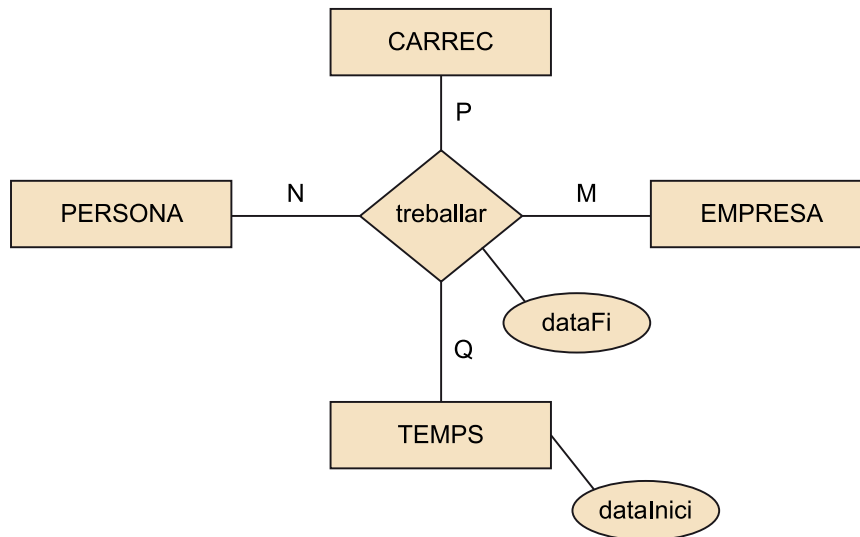
Per exemple, per poder mantenir l'històric de carrecs ocupats per persones que treballen en una empresa, ens caldria conèixer l'interval de temps en què han ocupat cada càrrec.

Per aquest motiu ens caldrà tenir la data d'inici del càrrec, que sempre estarà informada perquè si no no existiria la interrelació entre les entitats, i la data de fi, que podrà ser desconeguda o tenir valor nul si la persona actualment encara ocupa el càrrec a l'empresa.

En el model conceptual es representarà de la manera següent:

#### Atenció!

Així doncs, en les interrelacions N:M:1 hem vist que la clau primària està formada per 2+1 atributs i en les relacions 1:1:1 la clau primària està formada per 0+1 atributs.



En aquest cas totes les entitats estaran connectades amb «molts», de manera que la clau primària de la relació «treballar» estarà formada per tots els atributs que formin totes les claus primàries de les entitats que associa la interrelació «treballar».

Les relacions derivades de la transformació de les entitats seran:

*PERSONA*(DNI, nom, cognoms)  
*CARREC*(codi, nom)  
*EMPRESA*(codi, nom, CIF)  
*TEMPS*(codi, dataInici)

I la relació derivada de la transformació de la interrelació serà:

*TREBALLAR*(DNI, codiCarrec, codiEmpresa, codiTemps, dataFi)  
on {DNI} referència PERSONA  
on {codiCarrec} referència CARREC(codi)  
on {codiEmpresa} referència EMPRESA(codi)  
on {codiTemps} referència TEMPS(codi)  
on {dataFi} pot ser nul

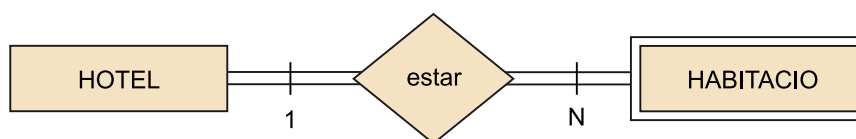
### 2.3. Transformació d'entitats dèbils

Com sabem, les entitats dèbils tenen la particularitat que no tenen un atribut que les identifiqui unívocament sinó que depenen d'una altra entitat.

Aquest tipus d'entitats es transformaran en el model relacional com si es tractés d'una entitat qualsevol, però la seva clau primària tindrà com un dels seus atributs, la clau forana envers la seva relació principal.

Considerem, per exemple, les habitacions d'un hotel que pertany a una cadena hotelera. A causa que el número de les habitacions es podrà repetir en hotels diferents, aquest número no servirà per a identificar-les de manera unívoca, essent necessari saber en quin hotel estarà situada l'habitació.

En el model conceptual es representarà de la manera següent:



La transformació en model relacional serà:

*HOTEL(codi, nom)*  
*HABITACIO(numero, codiHotel)*  
 on {codiHotel} referència *HOTEL(codi)*

Com veiem, la clau primària de la relació *HABITACIO* està formada pels atributs *numero* i *codiHotel* però, tanmateix, la seva clau forana envers la relació *HOTEL* és l'atribut *codiHotel*.

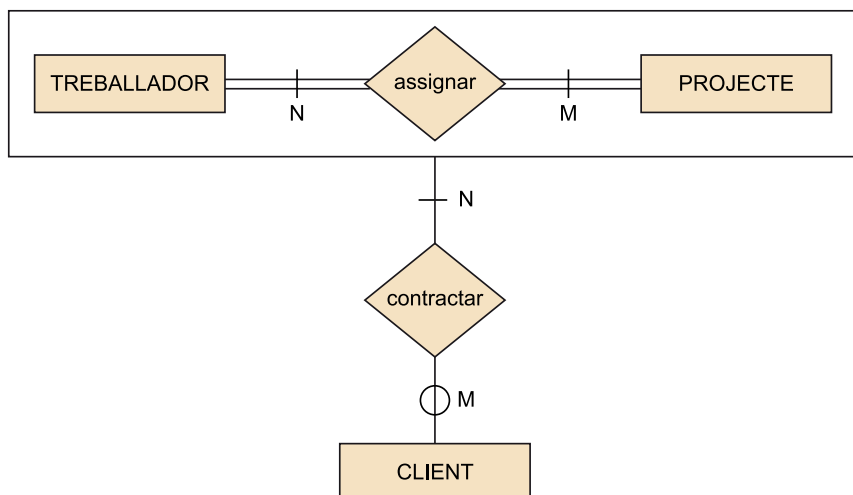
## 2.4. Transformació d'entitats associatives

Les entitats associatives vindran donades al considerar una interrelació entre entitats com si fos una entitat.

La transformació d'aquest model conceptual es realitzarà en dos passos, modelant en primer lloc l'entitat associativa com si fos una interrelació binària i, en segon lloc, la resta d'entitats i interrelacions que hi intervenen.

Pensem en el cas d'una empresa en què els treballadors estan assignats a diferents projectes i aquests projectes poden ser d'ús intern o contractats per diferents clients.

En el model conceptual es representarà de la manera següent:



La transformació en model relacional serà:

*TREBALLADOR(DNI, nom, ...)*  
*PROJECTE(codi, nom, ...)*  
*ASSIGNAR(DNI, codiProjecte)*  
 on {DNI} referència *TREBALLADOR*  
 on {codiProjecte} referència *PROJECTE(codi)*  
*CLIENT(codi, nom, ...)*  
*CONTRACTAR(codiClient, DNI, codiProjecte, ...)*  
 on {codiClient} referència *CLIENT(codi)*  
 on {DNI, codiProjecte} referència *ASSIGNAR*

## 2.5. Resum del procés de transformació

A continuació es mostra una taula de resum de les accions que s'han de realitzar a l'hora de transformar un model ER en un model relacional.

Element del model ER	Transformació al model relacional
Entitat	Relació

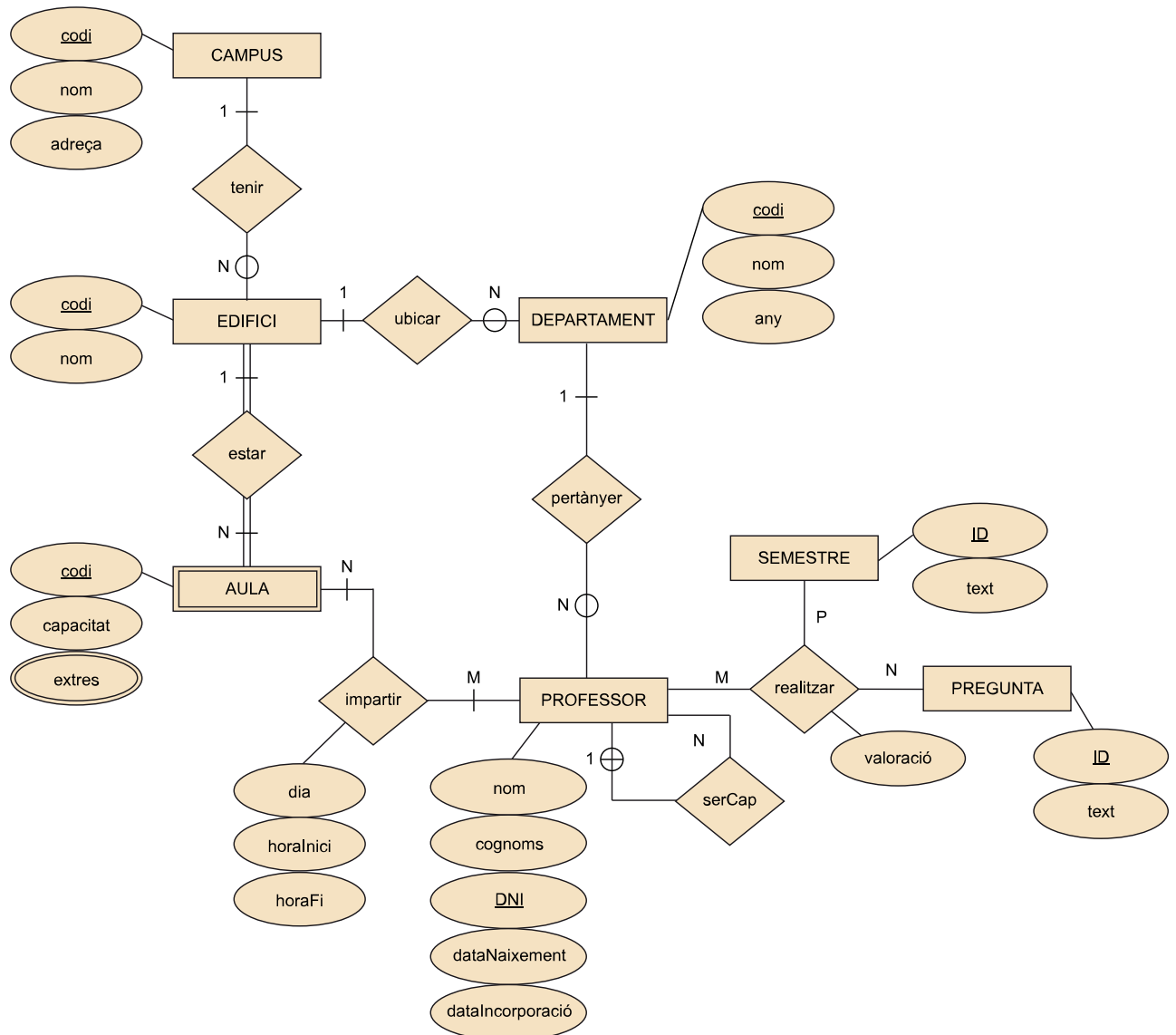
Element del model ER	Transformació al model relacional
Interrelació 1:1	Clau forana
Interrelació 1:N	Clau forana
Interrelació M:N	Nova Relació
Interrelació n-ària	Nova Relació
Interrelació recursiva	Com a les interrelacions no recursives: <ul style="list-style-type: none"> <li>• Clau forana per a binàries 1:1 i 1:N</li> <li>• Nova relació per a binàries M:N i n-àries</li> </ul>
Entitat dèbil	La clau forana de la interrelació identificadora forma part de la clau primària
Generalització/especialització	<ul style="list-style-type: none"> <li>• Nova relació per a l'entitat superclasse</li> <li>• Nova relació per a cadascuna de les entitats subclasse</li> </ul>
Entitat associativa	La transformació de la interrelació que l'origina és alhora la seva transformació

És important fer notar que el model lògic relacional no té tanta riquesa semàntica como el model conceptual i que, per tant, de vegades les restriccions modelades en el model conceptual que no són representables en el lògic relacional han de ser tractades en el disseny físic a nivell de programació.

## 2.6. Exemple de transformació de model conceptual (ER) en model relacional

Una universitat vol implementar una aplicació que els permeti gestionar les activitats que realitzen els seus professors, de manera que es pugui tenir un control de la seva docència en els diferents campus en què es divideix. Per aquest motiu, serà necessari definir una base de dades que permeti mantenir i tractar aquesta informació.

Després de revisar els seus requeriments, s'ha dissenyat el model conceptual representat al següent diagrama ER:



En primer lloc identificarem les relacions provinents de les entitats fortes, identificant els seus atributs i les claus primàries:

*CAMPUS*(codi, nom, adreça)

*EDIFICI*(codi, nom)

*DEPARTAMENT*(codi, nom, any)

*PROFESSOR*(nom, cognoms, DNI, dataNaixement, dataIncorporació)

*SEMESTRE*(id, text)

*PREGUNTA*(id, text)

En segon lloc, transformarem l'entitat feble AULA que dependrà de l'entitat EDIFICI:

*AULA*(codi, codiEdifici, capacitat, extres)

on {codiEdifici} referencia EDIFICI(codi)



A continuació transformarem les interrelacions binàries de connectivitat 1:N, que modificaran la definició de les relacions ja existents:

*EDIFICI(codi, nom, adreça, codiCampus)*

*on {codiCampus} referencia CAMPUS(codi)*

*DEPARTAMENT(codi, nom, any, codiEdifici)*

*on {codiEdifici} referencia EDIFICI(codi)*

*PROFESSOR(nom, cognoms, DNI, dataNaixement, dataIncorporacio, codiDepartament, DNICap)*

*on {codiDepartament} referencia DEPARTAMENT(codi) i no pot ser null*

*on {DNICap} referencia PROFESSOR(DNI)*

Tot seguit transformarem les interrelacions binàries de connectivitat N:M i les ternàries. En aquest cas la interrelació binària pertany a una entitat d'associació, de manera que farà referència a la relació corresponent a l'entitat dèbil.

*IMPARTIR(codiAula, codiEdifici, professorDNI, dia, horaInici, horaFi)*

*on {codiAula, codiEdifici} referencia AULA(codi, codiEdifici)*

*on {professorDNI} referencia PROFESSOR(DNI)*

*REALITZAR(professorDNI, preguntaID, semestreID, valoracio)*

*on {professorDNI} referencia PROFESSOR(DNI)*

*on {preguntaID} referencia PREGUNTA(id)*

*on {semestreID} referencia SEMESTRE(id)*

### 3. Teoria de la normalització

L'objectiu de la teoria de la normalització és formalitzar un conjunt d'idees simples que guien un bon disseny de bases de dades relacionals clàssiques.

La teoria de la normalització es fonamenta en el principi bàsic següent: tota relació ha de descriure un concepte semàntic únic.

La teoria de la normalització ens permet reconèixer els casos en els quals aquest principi no es compleix. El mecanisme que la teoria de la normalització fa servir per a això són les formes normals (FN). Una relació està en una forma normal determinada, si satisfà un conjunt de restriccions determinades que són pròpies d'aquesta forma normal. La violació d'aquestes restriccions origina que la relació tingui les anomalies i les redundàncies que hem esmentat abans.

Hi ha diverses formes normals, de les quals nosaltres estudiarem les quatre més importants. Cada forma normal indica unes restriccions específiques que ha de complir una relació.

Quan una relació no compleix una forma normal, és perquè viola la restricció associada amb aquella forma normal. Per aconseguir que es verifiqui la forma normal, caldrà evitar la condició que fa que es violi la restricció en qüestió. El procediment que aplicarem per aconseguir que no es violi la restricció associada a la forma normal rep el nom de **normalització**.

Nosaltres analitzarem una a una cada forma normal i resoldrem les condicions que fan que es violi una forma normal determinada.

La primera forma normal (1FN) es basa en el concepte de *valor atòmic*, mentre que la resta de formes normals es basen en el concepte de *dependència*. La segona forma normal (2FN), la tercera forma normal (3FN) i la forma normal de Boyce-Codd (FNBC) es basen en el concepte de **dependència funcional**. Hi ha però altres formes normals que es basen en altres conceptes i que no estudiarem.

Un model lògic que no està normalitzat, presenta una sèrie d'anomalies perjudicials per a la bona gestió de les dades que emmagatzema. Tot seguit veurem amb un exemple quines són aquestes anomalies i els problemes que generen.

#### Nomenclatura

Abreugem forma normal amb les sigles FN.

#### Graus de les formes normals

Si una relació està en tercera forma normal (3FN), també estarà en segona forma normal (2FN) i en primera forma normal (1FN). Per indicar que una relació no està en 1FN, s'acostuma a fer servir l'acrònim FN2. (NF NF, Non First Normal Form).

### 3.1. Anomalies de disseny

Com a conseqüència d'un disseny dolent, podem tenir relacions que presenten un alt grau de redundància, és a dir, presenten repeticions que són evitables. Aquest fet en complica el manteniment, atès que es produeixen anomalies. A continuació analitzem aquestes anomalies sobre la següent relació de *SUBMINISTRAMENTS*.

Taula 1. Relació de *SUBMINISTRAMENT*

SUBMINISTRAMENT			
CodiProveïdor	codiArticle	quantitat	ciutatProveïdor
P1	A1	100	Reus
P1	A2	150	Reus
P2	A1	200	Vic
P2	A2	250	Vic
P3	A2	100	Vic

Font: elaboració pròpia.

La clau primària de la relació de l'esquema *SUBMINISTRAMENT* (*codiProveïdor*, *codiArticle*, *quantitat*, *ciutatProveïdor*) està formada pels atributs *codiProveïdor* i *codiArticle*. Aquesta relació representa quins articles subministren els diferents proveïdors i en quina quantitat. A més, també enregistra la ciutat de cada proveïdor.

#### 3.1.1. Anomalies de modificació

Si un proveïdor canvia de ciutat, caldrà posar la nova ciutat del proveïdor a totes les tuples que facin referència al proveïdor en qüestió si no volem que la base de dades sigui inconsistent. Per exemple, si el proveïdor amb *codiProv = P1* canvia de ciutat i marxa a Salou, haurem de modificar dues tuples; si aquest proveïdor subministrés mil articles diferents, hauríem de modificar mil tuples, etc. La situació ideal fora que només haguéssim de fer una modificació.

Taula 2. Modificacions

SUBMINISTRAMENT			
CodiProveïdor	codiArticle	quantitat	ciutatProveïdor
P1	A1	100	Salou
P1	A2	150	Salou
P2	A1	200	Vic
P2	A2	250	Vic

Font: elaboració pròpia.

SUBMINISTRAMENT			
P3	A2	100	Vic

Font: elaboració pròpia.

Les anomalies de modificació obliguen a modificar totes les tuples que guarden un fet determinat si aquest canvia.

### 3.1.2. Anomalies d'esborrament

Si un proveïdor que només subministra un producte (per exemple, el proveïdor amb *codiProv* = P3) deixa de subministrar-lo, caldrà esborrar la tupla de la relació *SUBMINISTRAMENT*. Com a conseqüència, haurem perdut les dades personals del proveïdor en qüestió, en aquest cas, el codi de proveïdor i la ciutat. Aquest proveïdor ens ha subministrat 100 unitats de l'article A2 que tenim al magatzem. Si s'elimina el proveïdor, també s'elimina aquesta informació de l'estoc, que pot provocar desquadraments al magatzem.

Taula 3. Anomalies d'esborrament

SUBMINISTRAMENT			
CodiProveïdor	codiArticle	quantitat	ciutatProveïdor
P1	A1	100	Reus
P1	A2	150	Reus
P2	A1	200	Vic
P2	A2	250	Vic

Font: elaboració pròpia.

Sense voler, a causa de les anomalies d'esborrament, es poden perdre fets elementals.

### 3.1.3. Anomalies d'inserció

No podem emmagatzemar dades personals d'un proveïdor amb, per exemple, *codiProv* = P4 i de la ciutat de Mollet si no sabem els articles que subministra, atès que caldria afegir una tupla a la relació de *SUBMINISTRAMENT* amb valor *null* per a l'atribut *codiArticle*, i es violaria la regla d'integritat d'entitat de la clau primària:

#### **Integritat d'entitat de la clau primària**

Recordeu que la regla d'integritat d'entitat de la clau primària disposa que els atributs de la clau primària d'una relació no poden tenir valor *null*.

Taula 4. Anomalies d'inserció

SUBMINISTRAMENT			
CodiProveïdor	codiArticle	quantitat	ciutatProveïdor
P1	A1	100	Reus
P1	A2	150	Reus
P2	A1	200	Vic
P2	A2	250	Vic
P3	A2	100	Vic
P4	NULL	NULL	Mollet

Font: elaboració pròpia.

Les anomalies d'inserció consisteixen a no poder inserir fets elementals de manera independent.

L'origen de totes aquestes anomalies és que la relació *SUBMINISTRAMENT* descriu dos fets elementals, del món real, diferents: d'una banda, els articles que subministra cada proveïdor; i de l'altra, el proveïdor en si mateix. A més, aquests fets són completament independents entre si, atès que els articles concrets que subministra cada proveïdor no mantenen cap relació directa amb el fet que el proveïdor sigui, per exemple, d'una ciutat o d'una altra, i a l'inrevés. En tot cas, entre aquests dos fets hi ha una relació indirecta perquè afecten un mateix individu del món real, és a dir, el mateix proveïdor.

En conclusió, tota relació que no representa un concepte (o fet elemental) únic del món real està subjecta a presentar redundàncies, anomalies de manteniment i inconsistències potencials.

Aquest és el cas de la nostra relació *SUBMINISTRAMENT*.

Aquest serà l'objectiu del procés de normalització; separar els diversos conceptes o fets elementals del món real que estan en una mateixa relació, de forma que cada concepte o fet elemental acabi estant en una relació diferent tot aplicant les regles de les diferents formes normals.

## 3.2. El procés de normalització

### 3.2.1. Dependències funcionals i dependències funcionals completes (o plenes)

Com ja sabeu, sobre una base de dades podem imposar restriccions o regles d'integritat. Aquestes restriccions regulen els possibles estats vàlids, és a dir, els estats íntegres d'una base de dades. A continuació, presentem un nou tipus de restricció, les dependències funcionals.

Intentarem explicar el significat de la definició de dependència funcional mitjançant l'exemple de relació que es mostra a continuació:

Taula 5. *SUBMINISTRAMENT* (*codiProveïdor*, *codiArticle*, *quantitat*, *ciutatProveïdor*)

SUBMINISTRAMENT			
CodiProveïdor	codiArticle	quantitat	ciutatProveïdor
P1	A1	100	Reus
P1	A2	150	Reus
P2	A1	200	Vic
P2	A2	250	Vic
P3	A2	100	Vic

Font: elaboració pròpia.

Alguns exemples de dependències funcionals que existeixen en aquesta relació serien:

$\{codiProveïdor, codiArticle\} \rightarrow quantitat$

$\{codiProveïdor, codiArticle\} \rightarrow ciutatProveïdor$

Atès que  $\{codiProveïdor, codiArticle\}$  és la clau primària de la relació *SUBMINISTRAMENT*, estem segurs que cada valor de la parella  $\{codiProveïdor, codiArticle\}$  determina de manera unívoca els valors dels atributs *quantitat* i *ciutatProveïdor*.

També seria dependència funcional:

$codiProveïdor \rightarrow ciutatProveïdor$

En aquest cas també podem veure que, cada cop que l'atribut *codiProveïdor* pren el mateix valor, el valor de l'atribut *ciutatProveïdor* es repeteix.

Més concretament:

- Per a les dues primeres files que corresponen al proveïdor P1, el valor associat a l'atribut *ciutatProv* és sempre el mateix i indica que la ciutat del proveïdor és Reus.
- Per a les files 3 i 4 que corresponen al proveïdor P2, el valor associat a l'atribut *ciutatProv* es repeteix i indica que la ciutat del proveïdor és Vic.
- Per a la cinquena fila que correspon al proveïdor P3, el valor associat a l'atribut *ciutatProv* indica que la ciutat del proveïdor és Vic.

Dit d'una altra manera, per conèixer el valor de l'atribut *ciutatProveïdor* només cal fixar-se en el valor de l'atribut *codiProveïdor*.

El significat que té aquest fet en el món real és que un proveïdor està localitzat sempre a la mateixa ciutat i que aquest fet NO depèn del codi de l'article que proveeixi.

D'altra banda, fixem-nos en la primera dependència funcional:

$\{codiProveïdor, codiArticle\} \rightarrow quantitat$

En aquest cas, per conèixer la quantitat proveïda d'un cert article, és necessari conèixer tant el codi del proveïdor que ens el proveeix, com en el codi de l'article que es proveeix.

Si només es té en compte el *codiProveïdor*, es pot veure que per al proveïdor amb codi P1, hi ha dues quantitats diferents, 100 i 150 (o sigui, que el *codiProveïdor* sol no determina el valor de l'atribut *quantitat*).

D'altra banda, si ens fixem només en el *codiArticle*, es pot veure que per a l'article amb codi A1, també hi ha dues quantitats diferents, 100 i 200 (o sigui, que el *codiArticle* sol no determina el valor de l'atribut *quantitat*).

Per determinar el valor de l'atribut *quantitat* de forma unívoca, és necessari conèixer tant el *codiProveïdor* com el *codiArticle*. El que és el mateix, és necessari conèixer el valor de tota la clau primària.

Aquests tipus de dependències funcionals s'anomenen «dependències funcionals completes» o «dependències funcionals plenes».

En canvi, considerem el següent:

$codiArticle \rightarrow quantitat$

Això no seria dependència funcional, atès que no sempre que l'atribut *codiArticle* pren el mateix valor es repeteix el valor associat a l'atribut *quantitat*. Per exemple, la fila 1 i 3 tenen el mateix valor associat (A1) per l'atribut *codiArticle*, però en cada cas el valor que pren l'atribut *quantitat* és diferent (100 a la primera fila i 200 a la tercera fila, respectivament).

Així, un atribut (o conjunt d'atributs) {Y} depèn funcionalment d'un altre atribut (o conjunt d'atributs) {X}, expressat d'una altra manera  $\{X\} \rightarrow \{Y\}$ ; si donat un valor  $v_1$  per a {X} només es pot tenir un sol valor  $v_2$  per a {Y}, i a més, sempre que es doni  $v_1$  a {X} s'haurà de donar  $v_2$  a {Y}.

A l'exemple:

{X} = *codiProveïdor*

{Y} = *ciutatProveïdor*

$v_1$  = «P1»

$v_2$  = «Reus»

Sempre que l'atribut *codiProveïdor* tingui el valor «P1», el valor de l'atribut *ciutatProveïdor* haurà de ser obligatòriament Reus.

Les conclusions principals que podem extreure de la definició i l'exemple de les dependències funcionals són aquestes:

- a) Per identificar les dependències funcionals d'una relació, cal analitzar detingudament el significat dels atributs que hi intervenen.
- b) La clau primària d'una relació sempre determina funcionalment la resta d'atributs de la relació. Aquesta conclusió es pot estendre a totes les claus alternatives que la relació pugui tenir.

La primera conclusió és particularment important. A l'exemple, nosaltres hem deduït que el codi d'un proveïdor en determina funcionalment la ciutat. A més, aparentment, hem arribat a aquesta conclusió per l'examen de l'extensió de la relació *SUBMINISTRAMENT*. Però això no és del tot exacte o, dit d'una altra manera, no és suficient. Examinar l'extensió d'una relació ens ajuda a descartar dependències funcionals (a l'exemple, hem descartat que el codi dels articles determini el nombre d'unitats que un proveïdor és capaç de subministrar d'aquest article), però no ens permet d'assegurar rotundament que hi ha dependència funcional. Realment, hem d'anar més enllà i preguntar-nos si el comportament que inferim a partir de l'examen de l'extensió es dona, o no, en la realitat.



En resum, per deduir dependències funcionals no necessitem en absolut examinar l'extensió de la relació, atès que és una propietat d'intensió. En tot cas, examinar l'extensió ens pot ajudar a corroborar les dependències funcionals. El que necessitem és conèixer el significat dels atributs a la realitat per poder assegurar que les dependències funcionals calculades són les correctes.

Una vegada identificades les dependències funcionals i les dependències funcionals completes (o plenes), una forma d'indicar-les és la següent:

*SUBMINISTRAMENTS* (*codiProveïdor*, *codiArticle*, *quantitat*, *ciutatProveïdor*)  
 {*codiProveïdor*, *codiArticle*} → *quantitat*  
*codiProveïdor* → *ciutatProveïdor*

Com que {*codiProveïdor*, *codiArticle*} és la clau primària de la relació i determinen el valor de *quantitat*, la dependència funcional de *quantitat* és plena.

Com que *codiProveïdor* determina el valor de *ciutatProveïdor* i *codiProveïdor* no és la clau primària de la relació, la dependència funcional de *ciutatProveïdor* NO és plena.

Quan fem el disseny conceptual d'una base de dades i a continuació trobem el disseny lògic corresponent, totes les dependències funcionals que hi pugui haver en les diferents relacions que formen part del disseny lògic han de ser dependències funcionals completes.

### 3.2.2. Primera forma normal

Una relació està en **primera forma normal (1FN)** si i només si cap atribut de la relació és en si mateix una relació, és a dir, si tot atribut de la relació és atòmic, no es pot descompondre i no és un grup repetitiu.

#### Informació

La primera forma normal va ser proposada per Codd l'any 1970.

La relació *SUBMINISTRAMENT* que mostrem tot seguit no està en 1FN, atès que els atributs *codiArticle* i *quantitat* no són atòmics perquè són, en si mateixos, una relació, és a dir, es poden descompondre.

Taula 6. Relació no normalitzada en 1FN

SUBMINISTRAMENT			
<u>CodiProveïdor</u>	<u>codiArticle</u>	quantitat	ciutatProveïdor
P1	A1	100	Reus
	A2	150	
P2	A1	200	Vic
	A2	250	

SUBMINISTRAMENT			
P3	A1	200	Vic

Font: elaboració pròpia.

Per aconseguir normalitzar una relació en 1FN, caldrà aplicar una operació que es coneix amb el nom d'*aplanar*. Com a conseqüència d'aquesta operació, la relació *SUBMINISTRAMENT* quedaria de la manera següent:

Taula 7. Relació normalitzada en 1FN

SUBMINISTRAMENT			
<u>CodiProveïdor</u>	<u>codiArticle</u>	quantitat	ciutatProveïdor
P1	A1	100	Reus
P1	A2	150	Reus
P2	A1	200	Vic
P2	A2	250	Vic
P3	A2	100	Vic

Font: elaboració pròpia.

És important que ens adonem que, quan s'aplana la relació *SUBMINISTRAMENT* original, la clau primària de la relació canvia i passa a ser composta. També cal destacar que el model relacional sempre garanteix que les relacions estan en 1FN, atès que només hi ha una estructura per representar les dades (la relació) i, a més, cada dada es representa uniformement mitjançant valors atòmics.

### 3.2.3. Segona forma normal

Una relació està en **segona forma normal (2FN)** si i només si està en 1FN, i tot atribut no-clau depèn funcionalment i de manera completa de la clau primària.

#### Informació

La segona forma normal va ser proposada per Codd l'any 1970.

Hi ha una excepció: un atribut pot dependre funcionalment de part de la clau primària, si aquest atribut és part d'una clau alternativa.

D'altra banda, també es pot donar el cas que un atribut {Z} que no forma part de la clau primària depengui funcionalment d'un altre atribut {Y} que tampoc no forma part de la clau primària, però aquest segon atribut sí que té una dependència funcional completa amb la clau primària {X}:

$\{Z\} \rightarrow \{Y\} \rightarrow \{X\}$ , sent  $\{X\}$  la clau primària de la relació, i  $\{Y\}$  i  $\{Z\}$  atributs de la relació que no formen part de la clau primària.

En aquest cas, s'aplica un dels axiomes d'Armstrong, anomenat *transitivitat*, que diu:

Si  $\{Z\} \rightarrow \{Y\}$  i  $\{Y\} \rightarrow \{X\}$  llavors  $\{Z\} \rightarrow \{X\}$

En aquest cas es considera, per aplicació de l'axioma de transitivitat, que l'atribut  $\{Z\}$  també té una dependència funcional plena amb la clau primària  $\{X\}$ .

És important destacar que la 2FN es basa en el concepte de «dependència funcional completa» i, per tant, aquesta forma normal només es pot violar quan la clau primària d'una relació és composta. En resum, aquelles relacions amb clau primària simple, és a dir, amb clau primària formada per un únic atribut sempre estaran en 2FN.

Per exemple, la relació de *SUBMINISTRAMENT* que mostrem tot seguit no està en 2FN:

*SUBMINISTRAMENT* (*codiProveïdor*, *codiArticle*, *quantitat*, *ciutatProveïdor*)  
 $\{codiProveïdor, codiArticle\} \rightarrow quantitat$   
 $codiProveïdor \rightarrow ciutatProveïdor$

La idea que hi ha al darrera de la 2FN és molt simple. Si un atribut no-clau (en el nostre exemple, *ciutatProveïdor*) depèn funcionalment d'una part de la clau primària (*codiProveïdor*, a l'exemple) és perquè representa un fet relacionat amb aquesta part de la clau primària i, per tant, tots dos (l'atribut no-clau i els atributs de la clau del qual depèn) són un altre concepte semàntic (en el nostre cas, la ciutat de residència del proveïdor). La redundància que hi ha en una relació que no està en 2FN és immediata: els valors de l'atribut no-clau es repetiran per a tots els valors diferents de la part de la clau de la qual no depèn.

Per aconseguir passar una relació a 2FN, cal que evitem que hi hagi dependències funcionals no completes respecte a la clau. Per tant, tots els atributs que participen en la dependència funcional no completa s'hauran de projectar en una nova relació que correspon al concepte semàntic que representen, tal com mostrem a continuació:

*SUBMINISTRAMENT* (*codiProveïdor*, *codiArticle*, *quantitat*)  
 On  $\{codiProveïdor\}$  es refereix a *CIUTATS\_PROVEÏDOR* (*codiProveïdor*)  
 $\{codiProveïdor, codiArticle\} \rightarrow quantitat$   
*CIUTATS\_PROVEÏDOR* (*codiProveïdor*, *ciutat*)  
 $codiProveïdor \rightarrow ciutatProveïdor$

#### Axiomes d'Armstrong

Són un conjunt de regles d'inferència que permeten obtenir totes les dependències funcionals d'una relació.

D'aquesta manera, els diferents conceptes semàntics es descriuen en diferents relacions tot seguint el principi bàsic de la normalització.

### 3.2.4. Tercera forma normal

Una relació està en **tercera forma normal (3FN)** si i només si està en 2FN i cap atribut no-clau depèn funcionalment de cap altre conjunt d'atributs no-clau.

#### Informació

La tercera forma normal va ser proposada per Codd l'any 1970.

L'excepció aplicada a la 2FN es propaga també a la tercera forma normal. Considerem la relació *CLIENT* que mostrem a continuació:

*CLIENT* (*codiClient*, *carrer*, *número*, *ciutat*, *província*)

*codiClient* → *carrer*

*codiClient* → *número*

*codiClient* → *ciutat* → *província*

Aquesta relació està en segona forma normal, però no en tercera forma normal, atès que hi ha una dependència funcional *ciutat* → *província* entre dos atributs no-clau i, a més, cap dels dos atributs és clau alternativa de la relació i, per tant, l'excepció no s'aplica. De nou, el fet que l'atribut *província* depengui funcionalment de l'atribut *ciutat* és independent del client en si mateix. Les redundàncies i les anomalies que presenta aquesta relació són immediates.

#### Redundàncies associades a 3FN

Per a cada client que estigui localitzat a la ciutat de Mataró, tindrem repetit el fet que aquesta ciutat pertanyi a la província de Barcelona.

Per normalitzar una relació que viola la 3FN caldrà evitar la dependència funcional entre atributs no-clau. Per tant, els atributs que participen en la dependència funcional s'hauran de projectar en una nova relació que correspongui al concepte semàntic que representen, tal com mostrem a continuació:

*CLIENT* (*codiClient*, *carrer*, *número*, *ciutat*)

On {*ciutat*} es refereix a *CIUTAT* (*ciutat*)

*codiProveïdor* → *carrer*

*codiProveïdor* → *número*

*codiProveïdor* → *ciutat*

*CIUTAT* (*ciutat*, *província*)

*ciutat* → *província*

### 3.2.5. Forma normal de Boyce-Codd

Analitzem ara la relació *NOTA* següent:

*NOTA* (*dniAlumne*, *codiAssignatura*, *codiMatricula*, *nota*)

*dniAlumne* → *codiMatricula*

*codiMatricula* → *dniAlumne*

$\{dniAlumne, codiAssignatura\} \rightarrow nota$

La relació *NOTA* verifica la 1FN (recordem que el model relacional garanteix sempre per defecte aquesta forma normal). Aquesta relació també està en 2FN, malgrat que hi ha una dependència funcional no completa de part de la clau cap a un atribut no-clau ( $\{dniAlumne\} \rightarrow \{codiMatricula\}$ ). Atès que aquesta dependència involucra un atribut no-clau  $\{codiMatricula\}$  que forma part d'una clau alternativa de la relació  $\{codiMatricula, codiAssignatura\}$ , s'aplica l'excepció. Per acabar, la relació també està en 3FN encara que hi ha una dependència funcional entre atributs no-clau  $\{codiMatricula\} \rightarrow \{dniAlumne\}$ . De nou, atès que involucra atributs que formen part de claus alternatives, s'aplica l'excepció.

Malgrat això, tal com podem veure a continuació, la relació *NOTA* presenta redundàncies i anomalies:

Taula 8. Relació *NOTA*

NOTA			
<u>dniAlumne</u>	<u>codiAssignatura</u>	codiMatricula	nota
52111111A	05.001	123	A
52111111A	04.002	123	B
52222222B	05.002	215	B
52222222B	05.001	215	B
52222222B	04.002	215	C

Font: elaboració pròpia.

a) Redundàncies: per cada assignatura diferent en què estigui matriculat un alumne, es repetirà el seu codi de matrícula (per exemple, l'alumne amb DNI 52222222B té repetit tres cops el seu codi de matrícula).

b) Anomalies de modificació: si volem modificar, per exemple, el codi de matrícula de l'estudiant amb DNI 52222222B perquè sigui 312 en comptes de 215, haurem de modificar tres tuples. Com ja sabem, l'ideal seria modificar només una tupla.

L'origen d'aquestes anomalies és històric i es deu a un error d'omissió. Quan Codd va enunciar la 2FN i la 3FN l'any 1970, no va considerar la possibilitat que en una relació hi pogués haver diverses claus candidates que fossin compostes, i tampoc no va considerar la possibilitat que entre aquestes hi pogués haver encavalcaments. Per això, l'any 1974, Boyce i Codd van proposar la

forma normal de Boyce-Codd que soluciona les limitacions de la 2FN i la 3FN. De fet, sovint es normalitza una relació directament a la forma normal de Boyce-Codd sense passar prèviament per la 2FN i la 3FN.

Una relació està en forma normal de Boyce-Codd (FNBC)\* si i només si està en 1FN, i si tots els determinants són una clau candidata de la relació.

**\* Observació**

BCNF en anglès.

En la definició apareix un concepte nou: el concepte *determinant*.

Donada una dependència funcional  $\{X\} \rightarrow \{Y\}$ , un *determinant* és el conjunt  $\{X\}$ .

Per tant, si ho expressem d'una manera informal, els determinants en una dependència funcional són els orígens de fletxa.

Si tornem a examinar la relació *NOTA* i busquem totes les dependències funcionals que hi ha en la relació, arribem a la situació següent:

*NOTA* (*dniAlumne*, *codiAssignatura*, *codiMatricula*, *nota*)

$dniAlumne \rightarrow codiMatricula$

$codiMatricula \rightarrow dniAlumne$

$\{dniAlumne, codiAssignatura\} \rightarrow nota$

$\{codiMatricula, codiAssignatura\} \rightarrow nota$

La taula següent mostra tots els determinants (primera columna) i totes les claus candidates de la relació *NOTA* (segona columna):

Taula 9. Determinants i claus candidates de la relació *NOTA*

NOTA	
Determinants	Claus candidates
dniAlumne	dniAlumne, codiAssignatura
codiMatricula	codiMatricula, codiAssignatura
dniAlumne, codiAssignatura	
codiMatricula, codiAssignatura	

Font: elaboració pròpia.

Atès que no tots els determinants són claus candidates de la relació *NOTA*, arribem a la conclusió que *NOTA* no està en FNBC. De nou, per normalitzar la relació caldrà evitar la condició que causa la violació de la restricció. Per tant, haurèm d'evitar les dependències funcionals  $\{dniAlumne\} \rightarrow \{codiMatricula\}$  i  $\{codiMatricula\} \rightarrow \{dniAlumne\}$  de la relació *NOTA*.

Això s'aconsegueix si es projecten els atributs involucrats en una nova relació que correspon al concepte semàntic que representen. Ara bé, en aquest cas i a diferència dels casos anteriors, tenim diverses alternatives:

**a)**

*NOTA* (*dniAlumne*, *codiAssignatura*, *nota*)

On  $\{dniAlumne\}$  es refereix a *ALUMNES*  $\{dniAlumne\}$

$\{dniAlumne, codiAssignatura\} \rightarrow nota$

*ALUMNE* (*dniAlumne*, *codiMatricula*)

$dniAlumne \rightarrow codiMatricula$

$codiMatricula \rightarrow dniAlumne$

**b)**

*NOTA* (*codiMatricula*, *codiAssignatura*, *nota*)

On  $\{codiMatricula\}$  es refereix a *ALUMNES* (*codiMatricula*)

$\{codiMatricula, codiAssignatura\} \rightarrow nota$

*ALUMNE* (*codiMatricula*, *dniAlumne*)

$codiMatricula \rightarrow dniAlumne$

$dniAlumne \rightarrow codiMatricula$

**c)**

*NOTA* (*dniAlumne*, *codiAssignatura*, *nota*)

$\{dniAlumne, codiAssignatura\} \rightarrow nota$

*ALUMNE* (*codiMatricula*, *dniAlumne*)

$codiMatricula \rightarrow dniAlumne$

$dniAlumne \rightarrow codiMatricula$

**d)**

*NOTA* (*codiMatricula*, *codiAssignatura*, *nota*)

$\{codiMatricula, codiAssignatura\} \rightarrow nota$

*ALUMNE* (*dniAlumne*, *codiMatricula*)

$dniAlumne \rightarrow codiMatricula$

*codiMatricula* → *dniAlumne*

Les diferents possibilitats de projecció d'atributs entre la relació *NOTA* i la nova relació *ALUMNE* s'obtenen com a resultat de combinar les claus candidates de la relació original *NOTA*. Com a dissenyadors, som lliures de triar qualsevol de les opcions presentades. Ara bé, les més naturals són les opcions a) o b), atès que les opcions c) i d) són mixtes. Entre les opcions a) i b), des del punt de vista de la secretaria del centre, possiblement l'opció b) seria més coherent.\*

**\* Observació**

Pensem que hi ha alumnes d'altres països que no tenen DNI.

Observem que, independentment de l'opció triada, les relacions *NOTA* i *ALUMNE* estan en FNBC, atès que tots els determinants són claus candidates de la relació. Això és particularment important per a la nova relació *ALUMNE*.

### 3.2.6. Conclusions sobre dependències funcionals en les formes normals

Les conclusions principals que es poden extreure del procés de normalització són les que es presenten a continuació:

a) Sempre és possible normalitzar fins a la forma normal de Boyce-Codd.

b) El procés de normalització no és únic, tot i que hi ha solucions millors, en funció de la utilització posterior que s'hagi de fer de la base de dades normalitzada.

c) Com a conseqüència de la normalització, el model lògic final és millor que el model lògic inicial pels següents motius:

- Elimina redundàncies i anomalies.
- Separa fets semànticament diferents.

d) Donat un model lògic inicial d'una base de dades, si hi apliquem un procés de normalització, el model lògic final obtingut equival sempre al model lògic inicial. És a dir, el procés de normalització preserva la semàntica (o significat) del model lògic inicial. Cal tenir en compte que les dependències funcionals i les dependències funcionals completes representen els diferents conceptes semàntics relacionats que s'emmagatzemen a les diferents relacions de la base de dades. És per això que, en el procés de normalització, mai no es pot trencar una dependència funcional (completa o no). Si un atribut {Y} depèn d'un altre atribut {X}, i l'atribut {X} s'ha de reubicar en una altra relació, forçosament l'atribut {Y} també s'ha de reubicar en la mateixa relació que {X} per no perdre el concepte semàntic que indica la seva dependència funcional (es veurà aquest cas a l'exemple exposat més endavant).



### 3.3. Aplicació de la teoria de la normalització al disseny de bases de dades relacionals

El disseny de bases de dades relacionals comença sempre amb la captura i l'abstracció dels requisits de dades dels usuaris de l'organització que encarreguen el disseny de la base de dades. Això implica que el dissenyador ha de representar, amb la màxima exactitud possible, els objectes (o les entitats) del món real que hi intervenen, les propietats (o els atributs), les interrelacions entre entitats i, també, les regles d'integritat que s'han de complir. Tanmateix, és important distingir entre un disseny d'una base de dades partint de zero i un disseny d'una base de dades que parteixi de dissenys parcials ja existents.

En el segon cas, els dissenys parcials s'integren en un disseny únic i nou per a l'organització. La utilització d'aquests dissenys és freqüent en organitzacions que volen centralitzar diferents sistemes d'informació que fins ara funcionaven de manera independent, i també en el disseny de grans bases de dades; aleshores, en el disseny de la base de dades intervenen diferents equips que fan dissenys parcials amb l'objectiu d'obtenir un disseny global. En ambdós casos, és necessària una etapa d'integració de vistes per obtenir un esquema conceptual global.

En canvi, en el primer cas, el disseny de la base de dades es fa a partir de zero i es pot tractar teòricament des de dos punts de vista extrems:

1) Es pot partir d'una única relació, també coneguda com a *relació universal*, que conté tots els atributs d'interès per a una organització concreta, i també totes les dependències funcionals entre aquests atributs. A partir d'aquests elements, caldrà aplicar de manera reiterada un procés de normalització amb l'objectiu d'obtenir un esquema relacional (o conjunt d'esquemes de relacions) que caracteritzarà la base de dades de l'organització.

2) Es parteix d'un conjunt d'atributs d'interès per a una determinada organització i de les respectives dependències funcionals, i després es construeix un esquema relacional que caracteritza la base de dades de l'organització.

El primer enfocament sobre el disseny teòric de la base de dades dona lloc als mètodes d'anàlisi o descomposició, també coneguts com a **mètodes descendents**,\* mentre que el segon dona lloc als mètodes de síntesi o composició, també coneguts com a **mètodes ascendents**.\*

#### \* Observació

Mètodes descendents, en anglès, top-down.  
Mètodes ascendents, en anglès, bottom-up.

Ambdós enfocaments fan servir mètodes que podríem denominar *purs*. Tanmateix, a la pràctica, s'utilitzen mètodes mixtos que combinen anàlisi i síntesi.

En els **mètodes mixtos**, el dissenyador haurà d'elaborar el disseny conceptual de la base de dades mitjançant un model semàntic de dades.

Després de l'elaboració del model conceptual mitjançant els mètodes mixtos, caldrà traduir el model obtingut a un disseny equivalent, però ara expressat en el model relacional. Com a resultat d'aquest procés, obtenim el que es coneix com a *disseny lògic de la base de dades*. Per acabar, caldrà implementar el disseny lògic obtingut sobre l'SGBD relacional concret que emprà l'organització i, així, donar lloc al disseny físic de la base de dades.

Per què és més natural la utilització dels mètodes mixtos? La resposta és senzilla. Quan un dissenyador detecta un atribut, també detecta a quina o quines entitats pertany l'atribut en qüestió. O a l'inrevés: quan es detecta una entitat, també es poden detectar fàcilment una bona part dels seus atributs. Això significa que implícitament s'identifiquen algunes de les relacions que formen part del disseny lògic de la base de dades, precisament aquelles que corresponen a entitats en el disseny conceptual.

Explicitar les interrelacions entre les entitats és una tasca que està contemplada en qualsevol model semàntic de dades. En el disseny lògic posterior, aquestes interrelacions donaran lloc a relacions noves o a atributs nous que han de complir unes regles d'integritat determinades.

Així com la normalització sempre és necessària en els dissenys elaborats amb els mètodes d'anàlisi 1 (basats en la relació universal), també és convenient aplicar-la al final del procés dels mètodes mixtos per al disseny de bases de dades relacionals clàssiques. D'aquesta manera eliminarem redundàncies i evitarem les anomalies de disseny que hem estudiat a l'inici de la teoria de la normalització. Malgrat això, convé tenir present que el seguiment correcte de qualsevol metodologia de disseny condueix a un disseny lògic compost per relacions que verifiquen, com a mínim, la tercera forma normal.

En definitiva, la normalització resulta útil com a comprovació d'aspectes semàntics difícils, però poques vegades és realment necessària, atès que un dissenyador mínimament experimentat és capaç de distingir fets semànticament independents i, per tant, ja té en compte el principi bàsic en el qual es fonamenta la teoria de la normalització.

En casos molt particulars, pot interessar que una relació concreta no verifiqui una forma normal determinada. Aquesta tècnica es coneix en el disseny físic de bases de dades com a **desnormalització**. Pensem que la normalització minimitza l'existència de redundàncies i preveu les anomalies de disseny d'una relació per millorar la integritat de les dades; amb això, però, es paga el preu

d'una degradació en el rendiment perquè les dades que abans estaven en una única relació, després de la normalització, estan disperses en diferents relacions.

La normalització, doncs, tendeix a penalitzar la recuperació o la consulta de les dades perquè exigeix operacions de combinació per recuperar dades que originalment estaven en una única relació. Quan els atributs d'una relació que representen un altre concepte semàntic no s'actualitzen amb freqüència, pot ser convenient fer servir la *desnormalització*.

### 3.4. Exemple de normalització

Tenim la següent intenció de relació que emmagatzema informació sobre visites de pacients a metges:

*VISITA (numColegiat, numTS, dadesPacient, dadesHospital, dataVisita)*

Per donar context a la informació que conté la relació, direm que: un pacient que és identificat pel seu número de la targeta sanitària (*numTS*) es visita amb un metge, identificat pel seu número de col·legiat (*numColegiat*) en una certa data (*dataVisita*) i en un cert hospital (*dadesHospital*) d'una ciutat (*dadesHospital*). Només ens interessa conèixer la darrera data de visita de cada pacient, no l'historial de visites. Per això, per a cada parell (*numColegiat, numTS*), hi ha una sola data de visita i un sol hospital, i aquest parell d'atributs determina la clau primària. A més, sabem que un metge pot visitar diferents pacients en diferents hospitals.

De cada pacient es vol saber el seu nom complet (*dadesPacient*) –entès com a nom i cognoms– i el seu DNI (*dadesPacient*). La data de visita tindrà el format dd/mm/aaaa.

L'extensió de la relació vista en aquest moment és:

Taula 10. Extensió de la relació

<b>numColegiat</b>	<b>numTS</b>	<b>dadesPacient</b>	<b>dadesHospital</b>	<b>dataVisita</b>
C001	TS001	52111111A, Joan Pi Dot	del Mar, Barcelona	03/03/2019
C001	TS002	52222222B, Blai Morell Pi	Arnau de V., Lleida	05/03/2019
C002	TS001	Joan Pi Dot, 52111111A	del Mar, Barcelona	08/03/2019

Font: elaboració pròpia.

Està en 1FN?

Com es pot veure amb les dades informades, els atributs *dadesPacient* i *dadesHospital* no són atòmics, ja que aquestes columnes emmagatzemen diverses informacions diferents a les quals, a més, interessa accedir-hi de forma individual, segons el context.

### Normalització a 1FN

Per normalitzar a 1FN cal atomitzar els atributs, això vol dir que només poden tenir un valor. Això fa que es generin nous atributs perquè cada atribut emmagatzemi una única informació.

*VISITA* (*numColegiat*, *numTS*, *nomPacient*, *nomHospital*, *dataVisita*, *dniPacient*, *ciutatHospital*)

La nova extensió de la relació vista en aquest moment és:

Taula 11. Nova extensió de la relació

<b>numCo- legiat</b>	<b>numTS</b>	<b>nomPacient</b>	<b>nomHospital</b>	<b>dataVisita</b>	<b>dniPacient</b>	<b>ciutatHospital</b>
C001	TS001	Joan Pi Dot	del Mar	03/03/2019	52111111A	Barcelona
C001	TS002	Blai Morell Pi	Arnau de V.	05/03/2019	52222222B	Lleida
C002	TS001	Joan Pi Dot	del Mar	08/03/2019	52111111A	Barcelona

Font: elaboració pròpia.

### Està en 2FN?

Per saber si la relació està en segona forma normal cal comprovar que tots els atributs que no formen part de la clau primària tenen una dependència funcional plena (DFP) amb tota la clau primària, excepte les claus alternatives, que es consideren normalitzades en segona forma normal.

Veiem quines són les dependències:

*VISITA* (*numColegiat*, *numTS*, *nomPacient*, *nomHospital*, *dataVisita*, *dniPacient*, *ciutatHospital*)

- *nomPacient* té dependència funcional (DF) amb *numTS*.
- *nomHospital* té DFP amb tota la clau primària, perquè un metge pot visitar diferents pacients en diferents hospitals.
- *dataVisita* té DFP amb tota la clau primària.
- *dniPacient* és clau alternativa (es podria substituir a la clau primària el *numTS* pel *dniPacient*, i la clau primària de la relació *VISITA* també identificaria correctament totes les ocurrences).

- *ciutatHospital* té DF amb *nomHospital*, però com que *nomHospital* té DFP amb la clau primària, per la regla de transitivitat, es considera que *ciutatHospital* també té una DFP amb la clau primària.

Per tant, tenim aquestes dependències:

- $numTS \rightarrow nomPacient$
- $\{numColegiat, numTS\} \rightarrow nomHospital$
- $\{numColegiat, numTS\} \rightarrow dataVisita$
- $dniPacient \rightarrow numTS$
- $numTS \rightarrow dniPacient$
- $\{numColegiat, numTS\} \rightarrow nomHospital \rightarrow ciutatHospital$

Així, ràpidament podem veure que aquesta relació NO està en 2FN perquè hi ha atributs que no són clau alternativa, que no depenen de tota la clau, com ara *nomPacient*.

### Normalització a 2FN

Per normalitzar a 2FN, separarem relacions de manera que tinguem dependències funcionals plenes en cada taula. Així doncs, per una banda separarem les dades relatives a visites que depenen de tota la clau i crearem noves taules per a la resta. En aquest cas, tindrem:

*VISITA* (*numColegiat*, *numTS*, *nomHospital*, *dataVisita*, *dniPacient*, *ciutatHospital*)

On  $\{numTS\}$  es refereix a *PACIENT* (*numTS*)

*PACIENT* (*numTS*, *nomPacient*)

### Està en 3FN?

Per saber si la relació està en tercera forma normal, cal comprovar que cap atribut no-clau depengui de cap altre atribut no-clau o de cap conjunt d'atributs no-clau. S'aplica també l'excepció de les claus alternatives, com a la segona forma normal.

*VISITA* (*numColegiat*, *numTS*, *nomHospital*, *dataVisita*, *dniPacient*, *ciutatHospital*)

- *nomHospital* té DFP amb tota la clau primària, perquè un metge pot visitar diferents pacients en diferents hospitals.
- *dataVisita* té DFP amb tota la clau primària.
- *dniPacient* és clau alternativa (es podria substituir a la clau primària el *numTS* pel *dniPacient*, i la clau primària de la relació *VISITA* també identificaria correctament totes les ocurrències).

- *ciutatHospital* té una DF de *nomHospital* ja que donat un nom d'hospital, podem saber del cert en quina ciutat està.

Així doncs, tenim les dependències següents:

- $\{numColegiat, numTS\} \rightarrow nomHospital$
- $\{numColegiat, numTS\} \rightarrow dataVisita$
- $dniPacient \rightarrow numTS$
- $numTS \rightarrow dniPacient$
- $nomHospital \rightarrow ciutatHospital$

Veiem, doncs, que aquesta relació NO està en tercera forma normal, atès que *ciutatHospital* té una DF amb un atribut que no forma part de la clau primària, *nomHospital*.

### Normalització a 3FN

Es crea una relació per guardar la informació relativa als hospitals.

*VISITA* (*numColegiat*, *numTS*, *nomHospital*, *dataVisita*, *dniPacient*)

On  $\{numTS\}$  es refereix a *PACIENT* (*numTS*)

On  $\{nomHospital\}$  es refereix a *HOSPITAL* (*nomHospital*)

*PACIENT* (*numTS*, *nomPacient*)

*HOSPITAL* (*nomHospital*, *ciutatHospital*)

### Està en BCNF?

Com que la relació està en 1FN, caldrà comprovar que tots els determinants també són claus candidates de la relació.

Tal com diu la teoria, un determinant és l'atribut origen d'una dependència funcional. Fem dues llistes, una de determinants i una altra de claus candidates. Si les llistes coincideixen, la relació està en FNBC. Si no, caldrà normalitzar.

*VISITA* (*numColegiat*, *numTS*, *nomHospital*, *dataVisita*, *DNIPacient*)

- *dniPacient* determina el valor de *numTS*.
- *numTS* determina el valor de *dniPacient*.
- $\{numColegiat, numTS\}$  determina els valors de *nomHospital* i *dataVisita*.
- $\{numColegiat, dniPacient\}$  determina els valors de *nomHospital* i *dataVisita*.

Tenim aquestes dependències:

- $dniPacient \rightarrow numTS$
- $numTS \rightarrow dniPacient$
- $\{numCol-legiat, numTS\} \rightarrow nomHospital$
- $\{numCol-legiat, numTS\} \rightarrow dataVisita$

- $\{numCol.legiat, dniPacient\} \rightarrow nomHospital$
- $\{numCol.legiat, dniPacient\} \rightarrow dataVisita$

I, per tant, els determinants són els conjunts d'atributs que hi ha a l'esquerra de la dependència i les claus candidates són:  $\{numColegiat, numTS\}$  i  $\{numColegiat, dniPacient\}$ .

Fent dues llistes, tenim:

Taula 12. Determinants i claus candidates en la relació VISITA

Determinants	Claus candidates
<i>numTS</i>	$\{numColegiat, numTS\}$
<i>dniPacient</i>	$\{numColegiat, dniPacient\}$
$\{numColegiat, numTS\}$	
$\{numColegiat, dniPacient\}$	

Font: elaboració pròpia.

Així, podem observar que la llista de determinants NO és igual a la llista de claus candidates, la qual cosa permet deduir que la relació VISITA NO està en FNBC

### Normalització a FNBC

Per normalitzar fins a FNBC cal eliminar de la relació VISITA l'atribut *dniPacient*, que fa que la relació no estigui en FNBC.

Com que ja es disposa d'una relació per emmagatzemar les dades dels pacients, només cal moure de lloc l'atribut *dniPacient*, de VISITA cap a PACIENT.

VISITA (*numColegiat*, *numTS*, *nomHospital*, *dataVisita*)

On  $\{numTS\}$  es refereix a PACIENT (*numTS*)

I  $\{nomHospital\}$  es refereix a HOSPITAL (*nomHospital*)

PACIENT (*numTS*, *nomPacient*, *DNIPacient*)

HOSPITAL (*nomHospital*, *ciutatHospital*)

## Resum

En aquest mòdul s'ha introduït el model lògic de les bases de dades considerant com a model de dades el model relacional.

Com que partíem del model conceptual representat en notació ER, hem vist les pautes que cal seguir en la transformació del model conceptual, representat en un diagrama ER, en el model relacional tot utilitzant els elements propis d'aquest model.

Hem vist el concepte de clau primària, candidata i forana; com definir una relació, els seus components i les regles del model per assegurar la consistència de les dades. El resultat és un model lògic que podrà ser millorat amb la teoria de la normalització i implementat en l'etapa de disseny físic.

Finalment, també hem vist com aplicar les regles de la teoria de la normalització per obtenir un model lògic lliure d'anomalies de disseny, on cada fet semàntic correspon a una relació diferent.



## Activitats

**Exercici 1.** Una discogràfica vol poder gestionar la informació de les diferents cançons de les quals té adquirits els drets i dels músics que la van compondre o interpretar en els discos que ha editat.

De cada cançó es vol guardar el seu nom i el nombre de vegades que ha estat editada en diferents discos. A més, es voldrà guardar informació sobre qui és l'autor (tenint en compte que una cançó pot haver estat creada per més d'un músic).

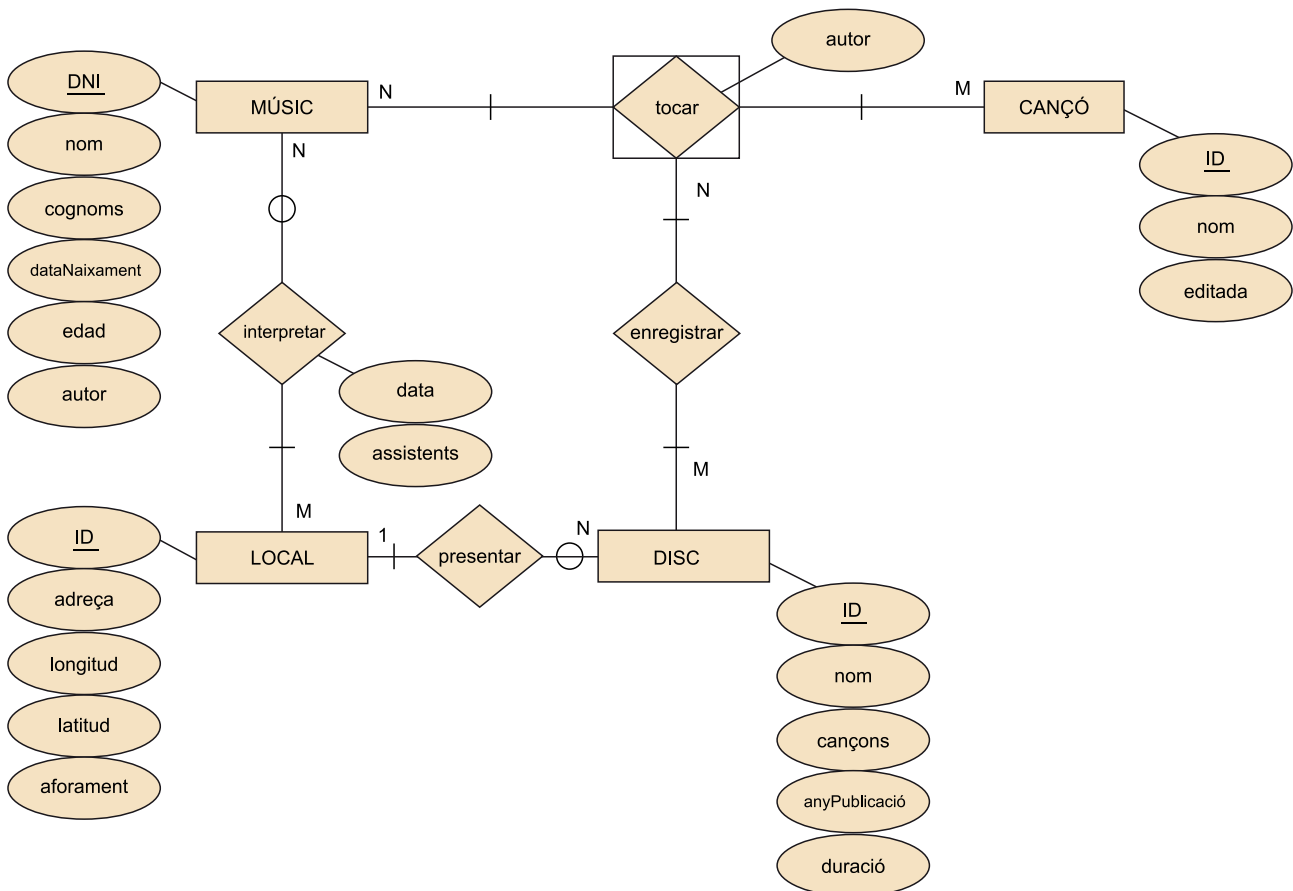
D'altra banda, es voldrà tenir un recompte dels músics que l'han enregistrat en un disc (ja sigui perquè n'és l'autor o perquè en fa una versió).

Com en el cas anterior, de cada músic es voldrà guardar el nom i cognoms, el DNI (que s'utilitzarà com a identificador), la data de naixement, l'edat i, també, el nombre de cançons de les quals és autor i que la discogràfica li ha editat.

De cada disc es voldrà guardar, a més de les cançons que conté i els músics que hi han participat, el nom, el nombre de cançons que té, l'any de publicació i la seva duració. D'altra banda, de cada disc es voldrà saber en quin local es va presentar.

Per cada local es voldrà saber el nom, l'adreça postal, l'aforament i la posició geogràfica (longitud i latitud). D'altra banda, es voldrà guardar una llista dels músics que han interpretat en aquell local, guardant la data d'aquella ocasió i els assistents al concert.

El model ER resultant durant el disseny conceptual és:



Es demana transformar el model ER en model relacional seguint les pautes indicades al mòdul.

**Exercici 2.** Es vol crear una pàgina web de notícies per difondre les notícies més rellevants de successos nacionals i internacionals. Les notícies es classificaran per categories (per exemple: «Educació», «Política», «Economia» i/o «Societat»). Dins de cada categoria es poden establir subcategories, així els lectors podran disposar d'informació més catalogada (per exemple: dins de la categoria «Educació» podríem tenir les subcategories «Universitats», «Escoles bressol» i «Recerca») si ho desitgen.

Les notícies presentades al portal en línia podran ser votades pels lectors. Les més votades podran aparèixer a la portada principal del lloc web. Hi ha dos tipus de votacions, les positives i les negatives. Per tal de poder calcular el valor final de la notícia, les votacions positives sumen un punt (+1) a la valoració i les negatives hi resten un punt (-1). Interessa emmagatzemar de cada votació qui l'ha feta i quin ha estat el valor de la votació.

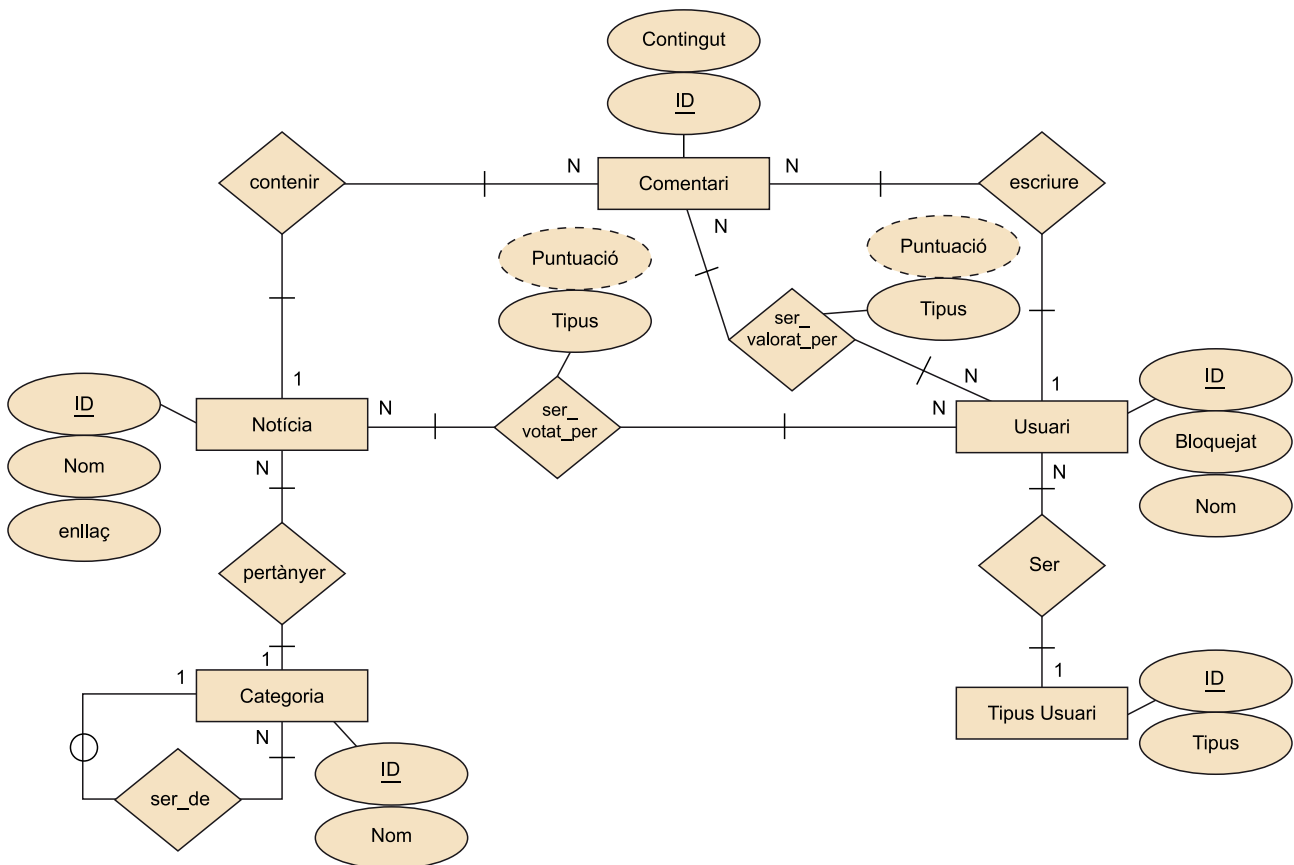
Els usuaris, en cas d'estar registrats, a més de votar les notícies, poden comentar-les. Això permet crear un fil de discussió sobre la notícia i, per tant, aporta més informació per als usuaris que en vulguin més. Els comentaris, així com les notícies, poden tenir vots positius i vots negatius, però només d'altres usuaris registrats.

Si es donés el cas que un usuari obtingués puntuacions negatives a molts dels seus comentaris, s'estudiaria el bloqueig del seu compte. L'acció de bloqueig de comptes només la podran realitzar els usuaris administradors. Dels usuaris administradors no està previst guardar informació a la base de dades.

El portal, com es comentava al principi, té com objectiu principal la difusió de notícies, i és per això que no manté el seu contingut sinó que només enllaça a la notícia original, ja sigui d'un blog, d'un diari en línia o d'una entitat pública com per exemple les seccions del BOC.

Es demana transformar el model ER en model relacional seguint les pautes indicades al mòdul.

El model ER resultant durant el disseny conceptual és:



Es demana transformar el model ER en model relacional seguint les pautes indicades al mòdul.

**Exercici 3.** Considereu la relació *COMANDA* amb l'esquema següent:

*COMANDA* (codiComanda, codiProducte, quantitat, codiClient, dataComanda)

La intenció d'aquesta relació és representar les dades dels productes que els clients demanen en les seves comandes.

Respecte a aquesta relació, es demana:

- a) En quina forma normal està la relació? Per què?
- b) Normalitzeu la relació, tant com es pugui, fins a FNBC.

**Exercici 4.** Considereu la relació *HORARI* amb l'esquema següent:

*HORARI* (*codiProf*, *codiAssig*, *dia*, *hora*, *aula*, *nomAssig*)

La seva intenció és representar dades dels horaris de les assignatures que s'imparteixen en un centre universitari presencial. A més, sabem que una assignatura es pot identificar tant pel codi d'assignatura com pel nom d'assignatura.

Respecte a la relació *HORARIS*, es demana:

- a) En quina forma normal està la relació? Per què?
- b) Normalitzeu la relació tant com es pugui, fins a FNBC.

**Exercici 5.** Una empresa dedicada al transport internacional de mercaderies vol registrar les dades dels viatges que fan els seus empleats. Per cada viatge que es fa s'utilitza un únic camió, on hi viatgen un conductor o més, segons la distància del viatge. Addicionalment, cada viatge té com a destinació una única ciutat del continent. Es vol enregistrar les dietes dels conductors en cada viatge. Per això s'ha creat una relació *DIETA* amb l'esquema següent:

*DIETA* (*codiViatge*, *codiConductor*, *matriculaCamio*, *importDieta*, *ciutatDestinacio*, *paisDestinacio*)

Es demana:

- a) En quina forma normal està la relació? Per què?
- b) Normalitzeu la relació tant com es pugui, fins a FNBC.

## Solucionari

### Exercici 1

MUSIC(DNI, nom, cognoms, dataNaixement)  
 LOCAL(ID, nom, adreça, longitud, latitud, aforament)  
 CANÇÓ(ID, nom, editada)  
 DISC(ID, nom, cançons, anyPublicació, durada, localID)  
 on {localID} referència LOCAL(ID)  
 on {localID} pot ser nul  
 INTERPRETAR(musicID, localID, data, assistents)  
 on {musicID} referència MUSIC(DNI)  
 on localID referència LOCAL(ID)  
 on {musicID} pot ser nul  
 TOCAR(musicID, cançoID, autor)  
 on {musicID} referència MUSIC(DNI)  
 on cançoID referència CANÇO(ID)  
 ENREGISTRAR(musicID, cançoID, discID)  
 on {musicID}, {cançoID} referència tocar  
 on {discID} referència DISC(ID)

### Exercici 2

COMENTARI(ID, contingut, usuariID, noticiaID)  
 on {usuariID} referència USUARI(ID)  
 on {noticiaID} referència NOTICIA(ID)  
 USUARI(ID, nom, bloquejat, tipusUsuariID)  
 on {tipusUsuariID} referència TIPUS\_USUARI(ID)  
 NOTICIA(ID, nom, enllaç, categoriaID)  
 on {categoriaID} referència CATEGORIA (ID)  
 CATEGORIA(ID, nom, categoriaID)  
 on {categoriaID} referència CATEGORIA (ID)  
 on {categoriaID} pot ser nul  
 TIPUS\_USUARI(ID, tipus)  
 SER\_VOTAT\_PER(noticiaID, usuariID, tipus)  
 on {noticiaID} referència NOTICIA (ID)  
 on {usuariID} referència USUARI (ID)

### Exercici 3

Per a solucionar aquest exercici, cal que busquem les dependències funcionals de la relació COMANDA sabent que {codiComanda, codiProducte} és la clau primària de la relació COMANDA. D'això se'n deriven aquestes dependències:

{codiComanda, codiProducte} → {quantitat}  
 {codiComanda, codiProducte} → {codiClient}  
 {codiComanda, codiProducte} → {dataComanda}

Adicionalment, atès que cada comanda és d'un únic client i es tramita en una data determinada, arribem a la conclusió que també hi ha les dependències funcionals següents:

{codiComanda} → {codiClient}  
 {codiComanda} → {dataComanda}

a) Per tant, en la relació COMANDA hi ha atributs que no depenen totalment de la clau primària i, a més, aquests atributs no són clau alternativa de la relació ni formen part de cap clau alternativa de la relació. En conseqüència, la relació COMANDA està en 1FN.

Per normalitzar la relació COMANDA a 2FN caldrà evitar les dependències funcionals no completes respecte de la clau, tal com mostrem a continuació:

COMANDA (codiComanda, codiClient, dataComanda)  
 PRODUCTECOMANDA (codiComanda, codiProducte, quantitat)  
 On {codiComanda} es refereix a COMANDA

Si representem el conjunt de dependències funcionals de cada relació, obtenim:

COMANDA (codiComanda, codiClient, dataComanda)  
 {CodiComanda} → {CodiClient}  
 {CodiComanda} → {DataComanda}

*PRODUCTECOMANDA* (*codiComanda*, *codiProducte*, *quantitat*)

On *{codiComanda}* es refereix a *COMANDA*  
*{codiComanda, codiProducte}* → *{quantitat}*

Fixeu-vos que, conseqüència de la normalització, hem separat els dos conceptes semàntics que hi havia representats en la relació original *COMANDA*:

- Les comandes en si mateixes.
- Els productes que es demanen en cada comanda.

Si examinem les dues relacions obtingudes (*COMANDA* o *PRODUCTECOMANDA*) arribem ràpidament a la conclusió que totes dues estan en FNBC.

#### Exercici 4

Per solucionar aquest exercici, cal que busquem les dependències funcionals de la relació *HORARI*. Com sabem que la clau primària és composta i està formada pels atributs (*codiProf*, *codiAssig*, *dia*, *hora*), tenim les següents dependències:

*{codiProf, codiAssig, dia, hora}* → *{aula}*  
*{codiProf, codiAssig, dia, hora}* → *{nomAssig}*

A més, també hi ha les dependències funcionals següents:

*{codiAssig}* → *{nomAssig}*  
*{nomAssig}* → *{codiAssig}*

Observem, doncs, que en la relació hi ha un atribut que no depèn totalment de la clau primària:

*{codiAssig}* → *{nomAssig}*

Aquest fet podria violar la 2FN, però aquest atribut forma part d'una clau alternativa de la relació *HORARI*. Per tant, la relació *HORARI* té la següent clau alternativa:

*{codiProf, nomAssig, dia, hora}*

S'aplica l'excepció associada a la 2FN i per tant podem afirmar que com a mínim la relació està en 2FN.

Si tornem a examinar el conjunt de dependències funcionals, ens adonem que hi ha un atribut no-clau que determina funcionalment un altre atribut no-clau, el que ens fa considerar la dependència funcional:

*{nomAssig}* → *{codiAssig}*

Aquest fet podria violar la 3FN i, atès que involucra un atribut que forma part d'una clau alternativa, s'aplica l'excepció. En conseqüència, la relació *HORARI* també verifica la 3FN.

Per saber si la relació *HORARI* està en FNBC, cal que refinem el nostre conjunt de dependències funcionals. Com que (*codiProf*, *nomAssig*, *dia*, *hora*) és clau alternativa de la relació *HORARI*, estem segurs que han d'existir les dependències funcionals següents:

*{codiProf, nomAssig, dia, hora}* → *{aula}*  
*{codiProf, nomAssig, dia, hora}* → *{codiAssig}*

Si ara comparem tots els determinants obtinguts i les claus candidates de la relació *HORARI*, veiem que no tots els determinants són claus candidates, i per tant, la relació no està en FNBC. A continuació mostrem els determinants i les claus candidates de la relació *HORARI*:

Horari	
Determinants	Claus candidates
codiProf, codiAssig, dia, hora	codiProf, codiAssig, dia, hora
codiProf, nomAssig, dia, hora	codiProf, nomAssig, dia, hora
codiAssig	

Horari	
nomAssig	

En conclusió, la relació *HORARI* està en 3FN.

Per aconseguir normalitzar la relació *HORARI* a la FNBC, caldrà evitar les dependències funcionals següents:

$\{codiAssig\} \rightarrow \{nomAssig\}$   
 $\{nomAssig\} \rightarrow \{codiAssig\}$

Això es pot aconseguir de diferents maneres. A continuació mostrem totes les possibilitats:

a)

*HORARI* (*codiProf*, *codiAssig*, *dia*, *hora*, *aula*)  
 $\{codiProf, codiAssig, dia, hora\} \rightarrow \{aula\}$   
*ASSIGNATURA* (*codiAssig*, *nomAssig*)  
 $\{codiAssig\} \rightarrow \{nomAssig\}$

b)

*HORARI* (*codiProf*, *codiAssig*, *dia*, *hora*, *aula*)  
 $\{codiProf, codiAssig, dia, hora\} \rightarrow \{aula\}$   
*ASSIGNATURA* (*nomAssig*, *codiAssig*)  
 $\{nomAssig\} \rightarrow \{codiAssig\}$

c)

*HORARI* (*codiProf*, *nomAssig*, *dia*, *hora*, *aula*)  
 $\{codiProf, nomAssig, dia, hora\} \rightarrow \{aula\}$   
*ASSIGNATURA* (*nomAssig*, *codiAssig*)  
 $\{nomAssig\} \rightarrow \{codiAssig\}$

d)

*HORARI* (*codiProf*, *nomAssig*, *dia*, *hora*, *aula*)  
 $\{codiProf, nomAssig, dia, hora\} \rightarrow \{aula\}$   
*ASSIGNATURA* (*codiAssig*, *nomAssig*)  
 $\{codiAssig\} \rightarrow \{nomAssig\}$

Fixeu-vos que, conseqüència de la normalització, hem separat els dos conceptes semàntics que hi havia representats en la relació original *HORARI*:

- Els horaris en si mateixos.
- Les assignatures impartides.

### Exercici 5

Per solucionar aquest exercici cal que busquem les dependències funcionals de la relació *DIETA*. Aquestes dependències són:

$\{codiViatge, codiConductor\} \rightarrow \{matriculaCamio\}$   
 $\{codiViatge, codiConductor\} \rightarrow \{importDieta\}$   
 $\{codiViatge, codiConductor\} \rightarrow \{ciutatDestinacio\}$   
 $\{codiViatge, codiConductor\} \rightarrow \{paisDestinacio\}$

Aquestes dependències funcionals són conseqüència del fet que (*codiViatge*, *codiConductor*) és la clau primària de la relació *DIETA*.

Atès que en cada viatge només es fa servir un camió i la destinació és única, també hi ha les dependències funcionals següents:

$\{codiViatge\} \rightarrow \{matriculaCamio\}$   
 $\{codiViatge\} \rightarrow \{ciutatDestinacio\}$   
 $\{codiViatge\} \rightarrow \{paisDestinacio\}$

Per acabar, la darrera dependència funcional que trobem és aquesta:

$\{ciutatDestinacio\} \rightarrow \{paisDestinacio\}$

La veracitat d'aquesta dependència funcional roman supeditada al fet que no es fan viatges a ciutats que tenen el mateix nom, però estan en països diferents. A partir d'aquestes dependències funcionals podem dir el següent:

a) Atès que en la relació *DIETA* hi ha atributs que no depenen totalment de la clau primària i, a més, aquests atributs no són clau alternativa ni formen part de claus alternatives de la relació de *DIETA*, podem afirmar que la relació *DIETA* està en 1FN.

b) Hi ha diferents alternatives per normalitzar la relació. A continuació en mostrem dues:

1. Normalitzem la relació *DIETA* original a 2FN. Cal descompondre la relació original en dues:

*VIATGE* (*codiViatge*, *matriculaCamio*, *ciutatDestinacio*, *paisDestinacio*)

$\{codiViatge\} \rightarrow \{matriculaCamio\}$

$\{codiViatge\} \rightarrow \{ciutatDestinacio\}$

$\{codiViatge\} \rightarrow \{paisDestinacio\}$

$\{ciutatDestinacio\} \rightarrow \{paisDestinacio\}$

*DIETA* (*codiViatge*, *codiConductor*, *importDieta*)

On  $\{codiViatge\}$  es refereix a *VIATGE*

$\{codiViatge, codiConductor\} \rightarrow \{importDieta\}$

La nova relació *DIETA* ja està en FNBC; en canvi, la relació *VIATGE* viola la 3FN, atès que hi ha dependències funcionals entre atributs que no són clau ni formen part de claus alternatives. Per normalitzar la relació *VIATGE*, cal descompondre-la en dues relacions, tal com mostrem tot seguit:

*VIATGE* (*codiViatge*, *matriculaCamio*, *ciutatDestinacio*)

On  $\{ciutatDestinacio\}$  es refereix a *CIUTAT*  $\{ciutat\}$

$\{codiViatge\} \rightarrow \{matriculaCamio\}$

$\{codiViatge\} \rightarrow \{ciutatDestinacio\}$

*CIUTAT* (*ciutat*, *país*)

$\{ciutat\} \rightarrow \{país\}$

Ara, tant la relació *VIATGE* com la relació *CIUTAT* estan en FNBC.

2. En aquest cas, i després d'examinar detingudament la relació *DIETA*, ens adonem que en la relació es representen tres fets diferents del món real: els viatges, les dietes dels diferents viatges i les ciutats. Per aconseguir separar aquests fets, normalitzem directament fins a la FNBC, que ens garanteix que qualsevol determinant a una dependència funcional ha de ser una clau candidata de la relació on es dona la dependència funcional. Com a conseqüència, obtenim tres relacions: la relació *VIATGE*, la relació *CIUTAT* i, finalment, la relació *DIETA*:

*VIATGE* (*codiViatge*, *matriculaCamio*, *ciutatDestinacio*)

On  $\{ciutatDestinacio\}$  es refereix a *CIUTAT*  $\{ciutat\}$

$\{codiViatge\} \rightarrow \{matriculaCamio\}$

$\{codiViatge\} \rightarrow \{ciutatDestinacio\}$

*CIUTAT* (*ciutat*, *país*)

$\{ciutat\} \rightarrow \{país\}$

*DIETA* (*codiViatge*, *codiConductor*, *importDieta*)

On  $\{codiViatge\}$  es refereix a *VIATGE*

$\{codiViatge, codiConductor\} \rightarrow \{importDieta\}$

## Glossari

**atribut d'una relació** *m* nom del rol que exerceix un domini en un esquema de relació.

**cardinalitat** *f* En una relació, el grau és el nombre de tuples que pertanyen a la seva extensió.

**clau candidata d'una relació** *f* Superclau *C* de la relació en la qual cap subconjunt propi de *C* és superclau.

**clau forana d'una relació** *f* Subconjunt dels atributs de l'esquema de la relació *CF*, en la qual existeix una relació *S* (que pot ser ella mateixa) que té per clau primària *CP*, i es compleix que, per a tota tupla *t* de l'extensió de *R*, els valors per a *CF* de *T* són o bé valors nuls, o bé valors que coincideixen amb els valors per a *CP* d'alguna tupla *s* de *S*.

**clau primària d'una relació** *f* Clau candidata de la relació escollida per a identificar les tuples de la relació.

**disseny lògic** *m* Procés de transformació d'un esquema conceptual en un esquema lògic de base de dades.

**domini** *m* Conjunt de valors atòmics.

**esquema de la relació** *m* Component d'una relació que consisteix en un nom de relació *R* i en un conjunt d'atributs  $\{A_1, A_2, \dots, A_n\}$ .

**extensió d'una relació d'esquema *R* (*A*)** *f* Conjunt de tuples  $t_i$  ( $i = 1, 2, \dots, m$ ) en què cada tupla  $t_i$  és un conjunt de parells  $t_i = \{ \langle A_1: V_{i1} \rangle, \langle A_2: V_{i2} \rangle, \dots, \langle A_n: V_{in} \rangle \}$  i, per a cada parell  $\langle A_j: V_{ij} \rangle$ , es compleix que  $v_{ij}$  és un valor de domina ( $A_{ij}$ ) o bé un valor nul.

**grau** *m* En una relació, el seu grau és el nombre d'atributs que pertanyen al seu esquema.

**inserció** *f* Fet d'afegir una o més tuples a una relació.

**integritat** *f* Propietat de les dades de correspondre a representacions plausibles del món real.

**normalització** *f* Procés pel qual, a partir d'un conjunt de relacions, s'obté un conjunt de relacions equivalent que satisfà la condició de la forma normal desitjada.

**relació** *f* Element de l'estructura de les dades d'una BD relacional format per un esquema (o intensió) i una extensió.

**superclau d'una relació** *f* Subconjunt dels atributs de l'esquema en el qual no hi pot haver dues tuples en l'extensió de la relació que tinguin la mateixa combinació de valors per als atributs del subconjunt.