

Model relacional. Àlgebra relacional. Normalització

1

Isidre Guixà i Miranda
INS Milà i Fontanals, d'Igualada

Sistemes gestors de bases de dades relacionals

Octubre del 2010
© Isidre Guixà i Miranda
INS Milà i Fontanals
C/. Emili Vallès, 4
08700 - Igualada

En cas de suggeriment i/o detecció d'error podeu posar-vos en contacte
via el correu electrònic iguixa@xtec.cat

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

Índex

Índex.....	3
Introducció.....	5
Objectius	7
1. El model relacional.....	8
1.1. Relacions	8
1.2. Claus	11
1.3. Esquema d'una relació	13
1.4. Diferències entre relació i arxiu.....	14
1.5. Regles d'integritat del model.....	16
1.6. El valor NUL en el model relacional	18
1.7. L'arquitectura ANSI/X3/SPARC i el model relacional.....	19
2. El model relacional a partir del model entitat - relació	23
2.1. Normes de traducció.....	23
2.2. Exemple global	35
3. Llenguatge procedural: àlgebra relacional.....	39
3.1. Operacions conjuntistes	39
3.2. Operacions relacionals	42
3.3. Operacions addicionals	44
3.4. Exemples.....	47
3.5. Conclusions.....	51
4. Normalització	52
4.1. La relació universal.....	55
4.2. Dependències funcionals	56
4.3. Primera forma normal	58
4.4. Preservació d'informació i dependències en la normalització	59
4.5. Segona forma normal	60
4.6. Tercera forma normal	62
4.7. Forma normal de Boyce-Codd	63
4.8. Quarta forma normal.....	66
4.9. Cinquena forma normal.....	67

Introducció

Els sistemes gestors de bases de dades són grans aplicacions informàtiques pensades per facilitar la gestió de dades i per a que aquesta gestió sigui eficient és necessari i convenient que les dades tinguin un disseny adequat, fet que s'assoleix seguint adequadament algunes de les metodologies de disseny existents, com ara el disseny entitat-relació.

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

En el nucli d'activitat "El model relacional" s'introdueix la teoria bàsica que dona suport al model relacional.

En el nucli d'activitat "El model relacional a partir del model entitat-relació" s'explica quines són les regles de traducció a emprar per obtenir el model relacional a partir d'un model entitat-relació.

En el nucli d'activitat "Llenguatge procedural: l'àlgebra relacional" es presenta l'àlgebra relacional com al llenguatge que incorpora el model relacional per a gestionar les dades. És un llenguatge teòric que ens ha d'ajudar a exercitar la nostra ment per a la gestió de dades en els sistemes gestors de bases de dades relacionals.

En el nucli d'activitat "Normalització" s'introdueixen un seguit de conceptes que formen la coneguda teoria de la normalització la qual permet detectar errors en el disseny de la base de dades i facilita mecanismes per a la seva correcció. En principi, un model relacional de bases de dades conseqüència d'un model entitat-relació previ, és correcte si el model entitat-relació era correcte, però en ocasions això no succeeix i en l'explotació d'una base de dades es pot detectar problemes que fan necessari un estudi del disseny del model relacional. Així mateix, hi ha dissenyadors que no volen modelar la realitat amb el model entitat-relació i volen dissenyar directament el model relacional. En qualsevol cas, doncs, es fa necessari disposar d'unes pautes de disseny en el model relacional.

El coneixement dels quatre nuclis d'activitats és important per qualsevol persona que analitzi, dissenyi i/o administri bases de dades relacionals i/o implementi aplicacions que interactuïn amb bases de dades i per a assolir uns coneixements òptims és imprescindible efectuar totes les activitats proposades així com els exercicis d'autoavaluació.

Objectius

En acabar la unitat didàctica heu de ser capaços del següent:

1. Identificar les característiques dels sistemes gestors de bases de dades relacionals (SGBDR), les prestacions dels productes existents en l'actualitat i les tendències.
2. Aplicar els processos de traducció del model entitat-relació al model relacional.
3. Diferenciar les característiques, l'objectiu i la forma de representació d'un disseny conceptual de dades i de relacions.
4. Definir un disseny lògic en un model relacional de taules, claus i condicions d'integritat, a partir del disseny conceptual.
5. Relacionar les operacions bàsiques de l'àlgebra i del càlcul relacionals amb els conceptes associats a la representació de la informació.
6. Determinar les causes i maneres de resoldre les incidències aparegudes en l'operació del SGBDR segons la informació dels manuals i de l'administrador del SGBDR.
7. Elaborar la guia d'usuari i la documentació completa relativa a les taules i als atributs de la base de dades relacional, de manera estructurada i clara.
8. Distingir el grau de normalització en el que es troben les relacions que defineixen un disseny relacional.
9. Aplicar els processos de normalització a les relacions que defineixen un disseny relacional per tal d'assolir millors nivells de normalització i així minimitzar les anomalies en la gestió de les dades emmagatzemades.

1. El model relacional

El model relacional es basa en una estructura de dades anomenada **relació** o **taula** molt semblant al concepte de tipus de dada interna *taula* o al concepte de tipus de dada externa *arxiu*.

A tall d'exemple, la taula 1 ens mostra una possible **relació** en una base de dades relacional.

Taula 1. Exemple de relació en una base de dades relacional.

Número	Cognom	Nom	Data Naixement	Pes	Altura	Sexe
7893	Ollé	Teresa	04.12.1985	45	150	D
4536	Esteve	Guifré	12.04.1980	60	170	H
2375	Miralles	Oriol	24.05.1979	62	173	H
9823	Brito	Júlia	23.09.1983	50	165	D
1287	Pelfort	Anna	15.07.1987	53	160	D
7834	Coromina	Jaume	10.10.1980	70	175	H

Oi que aquesta representació té moltes similituds amb la representació d'un *arxiu* o d'una *taula de tuples*? Alerta, una relació és semblant a una taula de tuples o a un arxiu, però amb certes precaucions que anem a explicitar.

1.1. Relacions

Com apareix el model relacional? En els models precursors, jeràrquic i en xarxa, hi havia una gran dependència entre l'estructura lògica de les dades i la seva representació interna. Així, els usuaris de bases de dades jeràrquiques veuen i accedeixen a les dades sota l'estructura del tipus de dada *arbre* i els usuaris de bases de dades en xarxa veuen i accedeixen a les dades sota l'estructura del tipus de dada *xarxa*.

El model relacional és proposat per E. F. Codd l'any 1970 abans que Chen presentés el model entitat - relació en l'any 1976. Els primers SGBDR apareixeran, però, a principis de la dècada dels 80.

El model relacional proposat per Codd és presenta per superar la dependència de l'estructura lògica de les dades i la seva representació interna. Per tant ha de ser un model força abstracte i això ho aconsegueix dotant-lo d'una base matemàtica, fet que motiva que la definició de **relació** tingui un alt component de llenguatge matemàtic.

El concepte **relació** es defineix sempre sobre un cert conjunt de dominis.

Referència bibliogràfica als treballs de Codd

E.F. Codd (Juny del 1970). A Relational Model of Data for Large Shared Data Banks. Communications of ACM. Vol. 13. N. 6. Pàgs. 377-387.

Domini. Importància en els models entitat-relació i en el model relacional.

En el model entitat-relació es defineix el **domini** com el conjunt de valors que pot prendre un determinat **atribut** d'una **entitat**. És a dir, el concepte de domini sorgeix a partir del concepte d'**atribut** i aquest del concepte d'**entitat**. La descripció del model relacional segueix un procés invers. Es parteix del concepte **domini** i s'arriba al concepte **relació**.

Un **domini** és un conjunt finit de valors **homogenis** i **atòmics** caracteritzats per un **nom**.

Homogenis per indicar que són valors del mateix tipus (dates de naixement, noms d'animals, estats civils,...)

Atòmics per indicar que són indivisibles des del punt de vista semàntic, és a dir, la seva descomposició perdria el significat associat.

El **nom** és la característica o propietat que representen (atribut).

En el darrer exemple podem observar set dominis diferents associats a set noms: *número, cognom, nom, data naixement, pes, altura* i *sexe*.

Una **relació** **R** sobre un conjunt finit de n dominis $D_1, D_2, \dots, D_n$, no necessàriament diferents, és un subconjunt del producte cartesià $D_1 \times D_2 \times \dots \times D_n$.

Recordem que el producte cartesià esmentat és el conjunt format per tots els possibles elements del tipus $(a_1, a_2, \dots, a_n)$ on a_i és un element de D_i . En ser R un subconjunt no té perquè contenir totes les possibles combinacions. El contingut de R pot anar des del subconjunt buit fins la totalitat del producte cartesià.

Anomenem **atribut d'una relació** al *paper* o *rol* que té cadascun dels dominis en què es basa la relació.

Anomenem **tuple d'una relació**, a cadascun dels elements que formen la relació.

Els conceptes formals (matemàtics) definits per **relació**, **atribut** i **tuple** són poc utilitzats i en el seu lloc s'utilitzen els conceptes informals **taula**, **columna** i **fila**. El motiu és obvi, no?

Taula 2. Relació per a exemplificar els conceptes *atributs* i *tuples*.

Número	Cognom	Nom	Data Naixement	Pes	Altura	Sexe
7893	Ollé	Teresa	04.12.1985	45	150	D
4536	Esteve	Guifré	12.04.1980	60	170	H
2375	Miralles	Oriol	24.05.1979	62	173	H
9823	Brito	Júlia	23.09.1983	50	165	D
1287	Pelfort	Anna	15.07.1987	53	160	D
7834	Coromina	Jaume	10.10.1980	70	175	H

A la relació de la taula 2 hi observem set *atributs* i sis *tuples*. Mol possiblement, és més còmode dir que en aquesta *taula* hi ha set *columnnes* i sis *files*.

Anomenem **grau d'una relació** al nombre d'atributs de la relació.

Aquest valor és constant al llarg del temps (mentre no hi hagi un canvi en el disseny de la relació). La relació de la taula 2 és de grau 7.

Anomenem **cardinalitat d'una relació** al nombre de tuples de la relació.

Aquest valor és variable al llarg del temps, ja que es poden produir altes i baixes a la relació. La relació del cas anterior té cardinalitat 6.

És clar que sempre es verifica que: $0 \leq \text{Card}(R) \leq |D_1| \times |D_2| \times \dots \times |D_n|$

Anomenem **intensió d'una relació** al significat semàntic de la relació, és a dir, al conjunt de n atributs acompanyats del corresponent domini

Aquest és un concepte invariable al llarg del temps, de la mateixa manera que ho és el grau de la relació.

Anomenem **extensió d'una relació** al conjunt de tuples que hi ha en una relació en un determinat moment.

Aquest concepte és variable al llarg del temps, de la mateixa manera que ho és la cardinalitat de la relació.

La taula 3 ens mostra la relació dels darrers quatre conceptes.

Taula 3. Relació de conceptes *intensió*, *extensió*, *grau* i *cardinalitat*.

	Invariable en el temps	Variable en el temps
Visió qualitativa	INTENSÍO	EXTENSÍO
Visió quantitativa	GRAU	CARDINALITAT

1.2. Claus

Una **clau** és un o varis atributs que conjuntament identifiquen, de manera unívoca, un tuple d'una relació.

Com a exemple, suposem que volem identificar les ciutats amb el seu nom. De tots és conegut que això no és possible doncs hi pot haver ciutats amb el mateix nom en diferents països. Fins i tot dins el mateix país. Oblidem-nos d'aquesta darrera possibilitat. Suposem que dins un país, el nom de la ciutat és atribut identificador. Però hem de gestionar ciutats de diferents països. Quin identificador prendre? Podem considerar la clau formada per la parella *nom de país* i *nom de ciutat*, parella que identifica tota ciutat.

Fixeu-vos que el concepte de clau també engloba el concepte d'atribut identificador, ja que una clau pot estar formada per un únic atribut.

En el model entitat-relació es distingeixen diferents tipus de claus que també cal distingir en el model relacional:

- **Clau candidata:** Qualsevol de les claus possibles sobre una entitat.
- **Clau mínima:** Clau que no té atributs redundants, és a dir, que tot atribut és imprescindible per a que sigui una clau.
- **Clau redundant:** Clau que té atributs redundants, és a dir, que té algun atribut no imprescindible per a que sigui una clau.
- **Clau primària:** Clau **mínima** escollida per identificar l'entitat. Només n'hi ha una.
- **Clau alternativa:** Tota clau candidata mínima no primària.

No cal dir que cap atribut d'una clau pot tenir definit el valor nul.

En el model relacional es mantenen els mateixos tipus de claus, amb l'obligatorietat de tenir una **clau primària**. Apareix, a més, un nou tipus de clau.

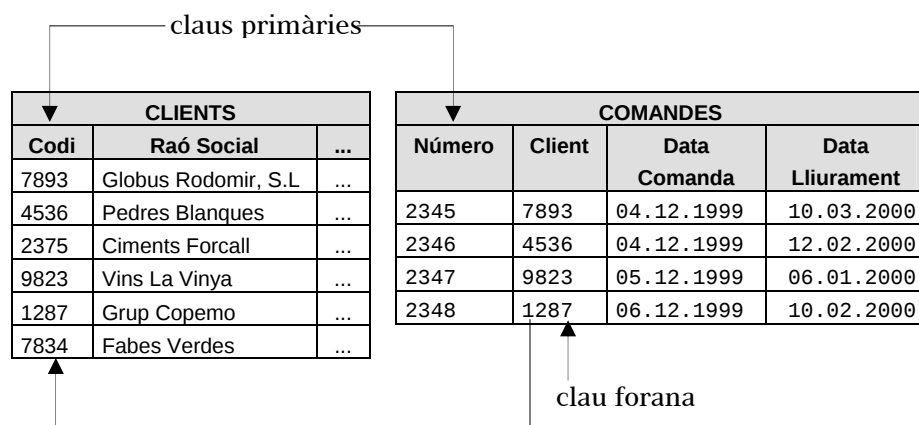
Claus primàries i foranes en alguns SGBD

Alguns SGBDR permet tenir relacions sense cap clau primària definida, cosa que està en contradicció amb les directrius del model relacional. Però davant una relació R sense clau primària, per permetre definir una altra relació R* que referenciï R mitjançant clau forana cal dir els atributs referenciats.

Anomenem **clau forana** d'una relació R a un conjunt no buit d'atributs de la relació definits sobre un domini que és el mateix domini d'una clau primària en una altra (o la mateixa) relació R*. En tal situació es diu que R* és la **relació referenciada** i R és la **relació que referencia**.

Aquest concepte és el que utilitza el model relacional per a establir la integritat referencial i que en el model entitat-relació s'estableix via relacions entre les entitats. Així, si considerem les relacions CLIENTS i COMANDES de manera que un client pot tenir varies comandes i que una comanda pertany obligatòriament a un client, tindriem la situació que ens mostra la taula 4.

Taula 4. Exemplificació de claus primàries i claus foranes.



L'atribut **client** de la relació COMANDES és clau forana respecte l'entitat CLIENTS. La relació CLIENTS és la relació referenciada i la relació COMANDES és la relació que referencia.

La definició explícita que la relació referenciada i la relació que referencia poden ser la mateixa. És el símil de la relació reflexiva sobre una entitat en el model entitat-relació. Així, si considerem la relació EMPLEATS en la que l'atribut **cap_superior** indica quin és el cap de cada empleat, el qual a la vegada és un empleat, tindriem la situació que mostra la taula 5

Taula 5. Exemple de clau forana sobre la mateixa taula.

EMPLEATS				
Número	Cognom	Nom	...	Cap Superior
7893	Ollé	Teresa	...	4536
4536	Esteve	Guifré	...	7834
2375	Miralles	Oriol	...	4536
9823	Brito	Júlia	...	7834
1287	Pelfort	Anna	...	
7834	Coromina	Jaume	...	1287

clau primària

clau forana

1.3. Esquema d'una relació

Ja hem definit els conceptes *relació* i *claus*. Hem vist que una relació no es pot entendre sense l'existència d'una obligatòria clau primària i possibles claus foranes. Interessa doncs, poder aglutinar tota aquesta informació sota un únic concepte.

Anomenem **esquema d'una relació** al conjunt d'informació que especifica la relació, format pel nom de la relació, el nom dels atributs, la clau primària i les claus foranes que puguin existir.

Els diferents autors que tracten el model relacional proposen diferents notacions per aquest concepte, totes elles, però, similars. Així, tenim:

nom_relació (nom_atr₁, nom_atr₂, nom_atr₃, ..., nom_atr_n)

on apareix el nom de la relació i el nom dels atributs. Manca, però, la informació referent a les claus.

- La clau primària es pot representar precedida del símbol # o subratllada. Si està formada per varis atributs, tots ells aniran precedits del símbol # o tots ells estaran subratllats.

Així, en els exemples anteriors, utilitzant el símbol #, tindríem:

clients (#codi, raó_social, ...)
comandes (#número, client, data_comanda, data_lliurament)

i utilitzant el subratllat, tindríem:

clients (codi, raó_social, ...)
comandes (número, client, data_comanda, data_lliurament)

- Les claus foranes les indicarem amb un subratllat discontinu, sota el qual hi apareixerà la relació de la qual n'és clau forana.

Així, en l'exemple anterior:

```
clients (codi, raó_social, ...)
comandes (número, client, data_comanda, data_lliurament)
           clients
```

Observem que en certes ocasions, un atribut pot ser part de clau primària i de clau forana. Imaginem-nos la situació en que cada país té codificats uns certs departaments (províncies) i que diferents països poden tenir departaments amb la mateixa codificació (cas d'entitat forta - dèbil en model entitat-relació). En el model relacional tindriem les dues relacions:

```
països (#codi, nom, ...)
departaments (#codipaís, #codidept, nom, ...)
              països
```

En aquest material utilitzarem el símbol # per les claus primàries i el subratllat discontinu per les claus foranes i, d'aquesta manera, quan un atribut formi part d'ambdós tipus de clau, no tindrem problemes amb els diferents subratllats.

Alguns autors incorporen un darrer concepte.

Anomenem **matriu relacional** a la col·lecció dels esquemes de les relacions en el disseny relacional d'una base de dades.

1.4. Diferències entre relació i arxiu

Els exemples de relacions presentats fins el moment ens podrien portar a pensar que no hi ha diferència entre una relació i el concepte que es té d'un arxiu -independentment del tipus d'accés que proporcioni el corresponent SGF-.

Hi ha, però, diferències molt importants, aparegudes a partir de diferents consideracions.

a) Basant-se en la teoria matemàtica de conjunts i en el producte cartesià en què es basa la definició de les relacions:

- No hi pot haver tuples repetits. (⚠)

- Els tuples no estan ordenats. (!!)

b) Basant-se en una clàusula afegida amb posterioritat a la definició matemàtica i que aprofitem aquest moment per introduir-la:

- Els atributs no estan ordenats, fet que està en contradicció amb la definició del producte cartesià i que per aquest motiu s'ha d'explicitar.

c) Basant-se en la definició que s'ha efectuat de domini:

- Els valors dels atributs són atòmics, és a dir, si es descomponen perden el seu significat.

Aquesta darrera observació no sembla una diferència entre relació i arxiu si en el disseny d'arxius es té present l'atomicitat dels camps. Això, però, no acostuma a passar. Recordeu el concepte utilitzat en molts SGF d'*atributs multivalor*, el qual és contrari al que s'assegura en el model relacional.

d) Basant-se en les regles d'integritat del model. Aquestes són tan importants que aquí únicament les enumerem i els hi dediquem un apartat específic.

- Els valors de la clau primària no poden prendre valors nuls.
- Cal mantenir concordança entre els valors que prenen els atributs d'una clau forana i els valors dels atributs de la clau primària de referència o, si és permès, deixar els atributs de la clau forana amb valors nuls.

Totes aquestes diferències entre relació i arxiu ens facilitaran la possibilitat, davant d'un producte comercial destinat a gestionar dades, de classificar-lo com a SGF o com a SGBDR. En l'actualitat és impossible trobar productes que utilitzin el terme SGBDR (o semblant) i en realitat no siguin res més que un SGF avançat, que incorpora algunes característiques de SGBDR. Així, recordo per exemple, els productes dBase (fins la versió dBase IV) que es definien com a *Gestors de Bases de Dades*. Aquesta definició era del tot incorrecte i moltes vegades m'havia de barallar amb "informàtics" amb poca o nul·la formació sobre SGBD que el defensaven aferrissadament com a SGBDR. Una única consideració deixava els seus raonaments per terra: on és la integritat referencial? Heu de tenir molt present que els diferents canvis que s'han produït en el món de la informàtica han suposat autèntiques revolucions conceptuais, per les que no tothom està preparat per assumir o disposat a digerir.

Informàtics dels anys 70

Jo diria que cada 10-15 anys es produeix una revolució dins el món de la informàtica a nivell comercial. Quan aquesta es produeix, els conceptes que arriben al món comercial ja va una dècada, com a mínim, que existeixen. Així tenim:

- Anys 70: SGF amb 3GL
- Anys 80: SGBD amb 3GL i 4GL textuais
- Anys 90: SGBDR amb 4GL visuals i Internet
- Segle XXI ???

1.5. Regles d'integritat del model

Tot model té unes regles d'integritat que cal conèixer. En el model relacional cal distingir-ne dues:

1) Regla d'integritat d'entitat

Els atributs que componen la clau primària d'una relació no poden prendre valors nuls.

Recordem que el concepte de valor nul té dues aplicacions:

- Quan l'atribut no és aplicable per alguna tuple, com pel cas d'un atribut que correspongui al *número de parts* que ha tingut una persona i es tracti d'una tuple corresponent a una persona de sexe masculí. Evidentment no té cap sentit, fins el moment present, aquest atribut.
- Quan l'atribut no és conegut per alguna tuple, com pel cas d'una data de naixement per una persona.

2) Regla d'integritat referencial

Els atributs que formen una clau forana d'una relació no poden prendre valors inexistents en els atributs de clau primària que referencien, és a dir, o prenen valors ja existents a la clau primària de referència o prenen valors nuls (si els tenen permesos).

No confondre el fet que una clau forana tingui permesa l'existència de valor nul amb la regla d'integritat d'entitat que obliga a no permetre valors nuls en la clau primària. En algunes situacions que veurem més endavant, una clau forana pot ser, a la vegada, clau primària de la pròpia relació. En tal cas, aquesta clau forana no podrà tenir valor nuls, per la regla d'integritat d'entitat. Però en una situació en què la clau forana no sigui part de la clau primària es pot decidir si pot o no tenir valors nuls.

Assegurar que un model té unes determinades regles d'integritat vol dir que els productes basats en el model les han de verificar. Però, el model no dona les pautes d'actuació per a verificar-les. Únicament marca que s'han de complir.

Així, ens trobem davant diferents possibilitats d'actuació per part dels SGBDR de cara a mantenir les regles d'integritat.

3) Per la regla d'integritat d'entitat tots els SGBDR estan d'acord: cap atribut de la clau primària pot prendre valors nuls i, per tant, ja no permeten en el moment d'efectuar una alta ni en modificacions de la clau primària.

4) Per la regla d'integritat referencial hi ha diferents situacions a controlar.

a) Quan es produeix una **alta** en la relació que té definida alguna clau forana.

Tots els SGBDR tenen una actuació similar: la clau forana es queda amb valor nul (si té permès aquest valor) o pren un valor existent en la clau primària de la relació referenciada.

e) Quan es produeix una **modificació en la clau primària d'una tuple** o una **baixa d'una tuple** que tingui tuples que la referencien en altres relacions.

En aquest punt hi ha diferents actuacions possibles i alguns SGBDR permeten escollir el tipus de gestió mentre que altres obliguen a utilitzar una única possibilitat. A continuació presentem les diferents possibilitats.

- **Actuació restringida**, consistent en no permetre la modificació de la clau primària en una tuple ni la baixa d'una tuple que tingui tuples que la referencien en altre relacions, de manera que no es violi la integritat referencial.
- **Actuacions permissives**, que permetin efectuar l'operació que viola la integritat referencial i que actuïn de manera automàtica per a restablir la integritat referencial. En aquest tipus de funcionament cal haver informat al SGBDR de l'actuació a seguir en cada possibilitat. En tenim varies:
 - **Propagació en cascada**, consistent en què la modificació en la clau primària d'una tuple referenciada provoca la mateixa modificació en totes les tuples que referencien la tuple modificada i en què la baixa d'una tuple referenciada provoca l'eliminació de totes les tuples que referencien la tuple donada de baixa.
 - **Propagació amb posta a valors nuls**, consistent en què la modificació en la clau primària d'una tuple referenciada o la baixa d'una tuple

referenciada provoca la posta a valors nuls en les claus foranes de totes les tuples que referencien la tuple modificada o donada de baixa.

- **Propagació amb posta a valors per defecte**, consistent en què la modificació en la clau primària d'una tuple referenciada o la baixa d'una tuple referenciada provoca la posta a valors definits per defecte en les claus foranes de totes les tuples que referencien la tuple modificada o donada de baixa.
- **Execució d'una rutina dissenyada específicament**, la qual és accessible pel SGBDR i que aquest executarà de manera automàtica *immediatament abans* de produir-se la modificació de la clau primària d'una tuple referenciada i/o la baixa d'una tuple referenciada. L'actuació ha de ser *immediatament abans* degut a que del contrari no es produiria la baixa i/o modificació per violar-se la integritat referencial.

Els SGBDR que disposen d'aquesta darrera possibilitat la utilitzen per a més menesters que per a mantenir la integritat referencial. Aquesta circumstància és coneguda com a **gestió de disparadors** i és molt útil quan es vol garantir que el SGBD actuï de manera automàtica davant certs esdeveniments (abans i després d'altres, baixes i modificacions).

Terminologia anglesa

El concepte **disparador** és conegut en la parla anglesa com a **trigger**, i per aquest motiu s'acostuma a utilitzar en l'argot informàtic.

1.6. El valor NUL en el model relacional

L'existència del valor NUL no ens és estranya. En diversos SGF existeix la possibilitat de definir-lo per a certs atributs dels arxius per a representar aquelles situacions en les que el valor de l'atribut és desconegut o no aplicable.

En els SGF seqüencial - indexats, la possibilitat de definir valors NULS en els atributs que constitueixen una clau d'accés permet ignorar, utilitzant la via d'accés, els registres amb valor nul en els atributs de la clau. És a dir, els registres afectats no formen part de l'índex, gestió efectuada pel SGF. Els programadors, en efectuar les aplicacions, són els responsables de la gestió dels valors NULS. El valor NUL és diferent per cada atribut i és un valor especial del domini en què es basa l'atribut.

En el model relacional la necessitat de valors NULS continua existint per les mateixes causes que en SGF. Però la gestió que el SGBDR n'ha d'efectuar és diferent i més complicada, per diversos motius:

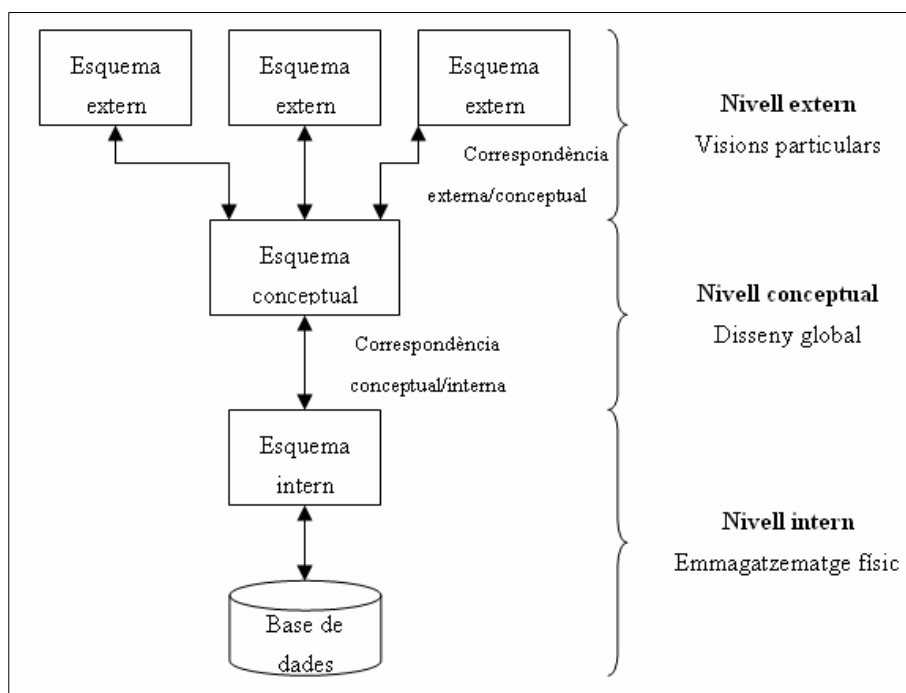
- En primer lloc, el SGBDR haurà de permetre la possibilitat de definir la obligatorietat o no d'un atribut. El fet de no ser obligatori implica l'existència de valor NUL per l'atribut en qüestió.
- El valor NUL no és un valor especial del domini de cada atribut, sinó que és un valor universal pel SGBDR.
- Els llenguatges de consulta i manipulació (LC i LMD), hostatjats o no, han d'incorporar instruccions especials per la manipulació del valor NUL i els usuaris n'han de ser conscients.
- La regla d'integritat d'entitat del model relacional no permet que els atributs de la clau primària prenguin el valor NUL.

Les anteriors causes provoquen que el SGBDR hagi de gestionar el valor NUL, però no hi ha cap normativa estàndard al respecte, de manera que cada fabricant de SGBDR l'implementa a la seva manera. Per aquest motiu és molt important esbrinar la gestió que del valor NUL incorpora qualsevol SGBDR abans d'utilitzar-lo.

1.7. L'arquitectura ANSI/X3/SPARC i el model relacional

Els SGBDR segueixen l'arquitectura de bases de dades anomenada ANSI/X3/SPARC, que estructura una base de dades en tres nivells: **intern**, **conceptual** i **extern**, cadascun dels quals correspon a un punt de vista diferent de la base de dades: emmagatzematge físic de la BD, disseny global de la BD i visions particulars de la BD. Vegeu la figura 1.

Figura 1. Arquitectura de tres nivells en els SGBDR



Tots els conceptes exposats fins el moment en el model relacional formen es seu esquema conceptual: dominis, atributs, relacions, claus primària i forana, grau, cardinalitat, intensió, extensió, valor NUL,...

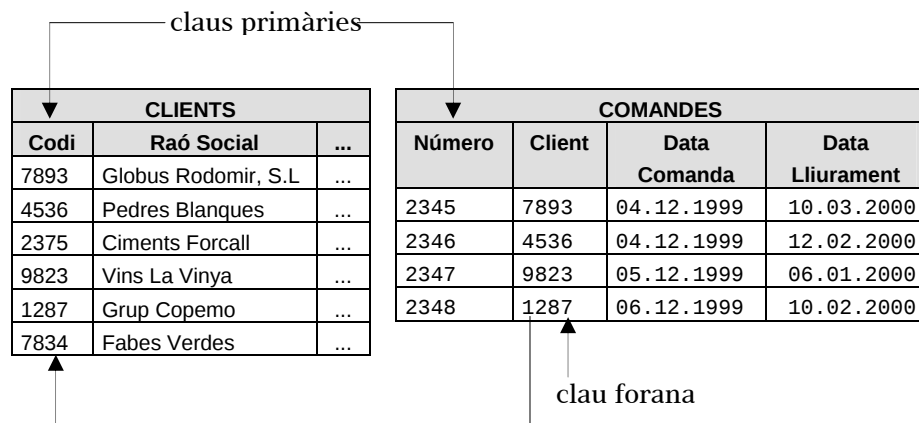
En els models jeràrquics i en xarxa de bases de dades, hi ha una gran dependència entre l'estructura lògica de les dades i la seva representació interna, és a dir, la base de dades està plena d'informació corresponent a la implementació física (els punters d'arbres i xarxes es palpen dins les entitats). En l'esquema conceptual del model relacional no hi ha, tal i com ha de ser, cap referència a la implementació física; per això hi ha l'esquema intern, del qual el model relacional no en diu res i per tant cada SGBDR l'implementa de la forma que creu més adient.

Ens manca comentar l'esquema extern. Com facilita el model relacional la representació de les diferents visions de la base de dades? La solució està en les **vistes**, també anomenades **relacions virtuals**.

Una **vista** és una relació, amb nom, derivada d'altres relacions i/o vistes.

Considerem, les relacions **CLIENTS** i **COMANDES** de la taula 6. Pot interessar una visió de les comandes on consti el número de comanda, el codi de client, la raó social i la data de comanda, com mostra la taula 7.

Taula 6. Relacions per a definir-hi vistes.



Taula 7. Exemple de vista a partir de taules.

COMANDES_RESUMIDES			
Número	Client	Raó Social	Data Comanda
2345	7893	Globus Rodomir, S.L	04.12.1999
2346	4536	Pedres Blanques	04.12.1999
2347	9823	Vins La Vinya	05.12.1999
2348	1287	Grup Copemo	06.12.1999

Observem que hem presentat quelcom semblant a una relació, amb un nom, `COMANDES_RESUMIDES`, que no pot coincidir amb cap dels noms de les relacions i/o vistes ja existents. És aconsellable que el nom tingui algun significat. Potser podríem haver trobat un nom més adient.

Aquesta espècie de relació és una vista. No conté cap dada, sinó que és el resultat d'una particular visió de les comandes que s'obté a partir de les relacions `COMANDES` i `CLIENTS`, les quals sí contenen les dades.

L'arquitectura ANSI/X3/SPARC presenta els esquemes conceptuals com a nexa entre les aplicacions i usuaris i la base de dades. Les aplicacions i usuaris tenen accés a l'esquema conceptual que els correspon i és el SGBD l'encarregat de traduir les sol·licituds formulades sobre l'esquema extern cap l'esquema conceptual amb la utilització del corresponent mapeig.

L'equivalència, en el model relacional, de l'esquema extern, és la definició de les vistes, de manera que l'usuari hauria d'accedir sempre a vistes i seria el SGBDR l'encarregat de traspassar les sol·licituds formulades sobre les vistes a les corresponents relacions. La realitat, però, és que el model relacional permet l'accés directe a les relacions (sempre i quan hi hagi els corresponents permisos).

El model relacional permet l'accés, amb les pertinents autoritzacions, de les aplicacions i usuaris a les relacions (que contenen les dades) i a les vistes (que no contenen dades però es refereixen a dades existents en altres relacions). Les aplicacions i usuaris no tenen per què saber si accedeixen a relacions i/o vistes, és a dir, no poden percebre la diferència.

Per últim, notem que una vista es defineix en termes d'altres relacions i/o vistes, és a dir, es poden definir vistes que es derivin de relacions i altres vistes.

La gestió de les dades a través de vistes té avantatges i inconvenients.

- Entre les avantatges cal destacar que una vista és definible en els termes que necessita l'aplicació o l'usuari, de manera que es modela la realitat a les particulars necessitats i així l'accés a les corresponents dades és més còmode.
- Entre els inconvenients estan les operacions d'actualització d'una vista, les quals en moltes ocasions no es poden executar ja que el SGBDR es troba amb ambigüitats. Així, en la vista `COMANDES_RESUMIDES`, si es demana eliminar una tuple, el SGBDR pot interpretar que es tracta d'eliminar només la tuple de la relació

COMANDES o aquesta i la corresponent tuple de la relació CLIENTS.
Quina decisió ha de prendre? No té prou informació.

No hi ha cap normativa estàndard respecte a l'actuació dels SGBDR davant les actualitzacions sobre vistes. Per aquest motiu cal esbrinar detalladament quin és el funcionament que aporta cada producte. N'hi ha que són molt restrictius i no permeten cap actualització i d'altres que són més permissius i faciliten actualitzacions sota certes condicions.

2. El model relacional a partir del model entitat - relació

Hi ha dues maneres d'obtenir el model relacional corresponent a una base de dades:

- Dissenyar-lo directament a partir de l'estudi de la situació a modelar aplicant les tècniques de normalització.
- Dissenyar el model entitat-relació i aplicar un procés de transformació entre models.

En aquest apartat veurem com obtenir el model relacional a partir del model entitat-relació.

2.1. Normes de traducció

Recordem que en el model entitat-relació hi ha entitats i relacions entre entitats acompanyades, ambdues, d'atributs, mentre que el model relacional hi ha relacions amb atributs.

Per tant, les normes de traducció han de consistir en com passar de les entitats i les relacions del model entitat-relació a les relacions del model relacional.

1) Transformació de les entitats.

Aquesta és una transformació directa, és a dir, cada entitat del model entitat-relació provoca l'aparició d'una relació en el model relacional, amb els mateixos atributs. La clau primària també es manté.

2) Transformació de les relacions entre entitats.

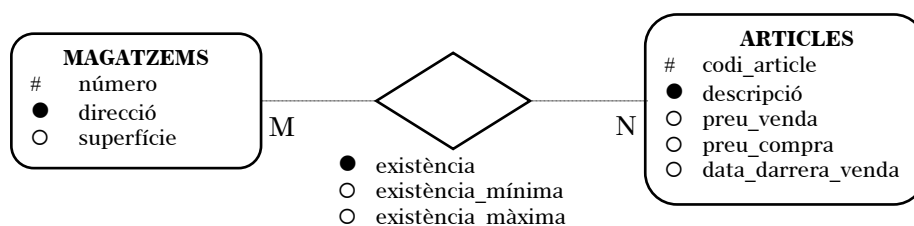
Recordem que les relacions del model entitat-relació connectaven entitats i portaven associat els conceptes de grau i d'obligatorietat. Hi ha diversos casos de transformació de relacions del model entitat-relació a model relacional.

a) Relacions binàries (M:N).

El procés a aplicar consisteix en:

- Es genera una nova relació, en el model relacional, que per clau primària, té els atributs de les claus primàries de les dues entitats que participen amb grau N a la relació del model entitat-relació.
- Els subconjunts d'atributs de la nova clau primària provinents de les claus primàries de les entitats són claus foranes de les relacions generades per les corresponents entitats.
- Els atributs existents (si n'hi ha) a la relació del model entitat-relació passen a ser atributs de la nova relació.

El model entitat-relació que presentem a continuació correspon al control d'existències en una organització que disposa d'articles que es poden trobar en diferents magatzems i interessa tenir constància de l'existència de cada article en els diferents magatzems. Així mateix interessa poder mantenir les existències mínimes i màximes de cada article en cada magatzem.



En l'exemple superior podem observar dues entitats i una relació de grau (M:N). La seva traducció al model relacional és:

```

magatzems (#número, direcció, superfície)
articles (#codi_article, descripció, preu_venda, preu_compra, data_darrera_venda)
stock (#magatzem, #article, existència, existència_mínima, existència_màxima)
           magatzems    articles

```

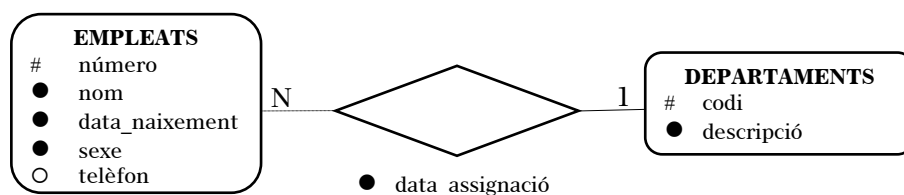
Les relacions MAGATZEMS i ARTICLES apareixen com a traducció de les entitats d'igual nom. La relació STOCK, però, apareix com a traducció de la relació d'igual nom entre les entitats MAGATZEMS i ARTICLES degut a que la relació en el model entitat-relació té dues entitats que participen amb grau N.

La clau primària de la nova relació STOCK està formada per (*magatzem*, *article*), és a dir, les claus primàries de les relacions MAGATZEMS i ARTICLES. L'atribut *magatzem* és part de la clau primària i és clau forana de MAGATZEMS. L'atribut *article* és part de la clau primària i és clau forana d'ARTICLES.

b) Relacions binàries (1:1) o (1:N).

- En aquesta situació mai apareixerà una nova entitat.
- Si hi ha una entitat X que participa amb grau N (cas 1:N), a la corresponent relació X del model relacional s'afegeix els atributs de la clau primària de l'altra entitat (la que participa amb grau 1 a la relació).
- Si no hi ha cap entitat que participi amb grau diferent d'1 (cas 1:1), s'escull una entitat (suposem X) de les que participen amb grau 1 i la seva traducció al model relacional passa a ser la relació X a la que, com en el paràgraf anterior, s'afegeix els atributs de la clau primària de l'altra entitat.
- Els atributs afegits a la relació X a causa dels dos anteriors paràgrafs passen a ser claus foranes de les entitats de les que provenen. El caràcter d'obligatorietat de l'entitat dins la relació en el model entitat-relació es correspon amb l'existència de *valor nul* pels nous atributs, de manera que "obligatorietat es correspon amb *no valor nul*" i "*no obligatorietat* es correspon amb *valor nul*".
- Els atributs existents (si n'hi ha) a la relació del model entitat-relació passen, també, a ser nous atributs de la relació X.
- Exemple per una relació (1:N)

En el següent model entitat-relació es modela la situació existent entre els departaments i els empleats d'una determinada organització en què, obligatòriament cada empleat és assignat a un departament i un departament pot tenir molts empleats assignats.



En el model entitat-relació apareixen dues entitats i una relació (N:1). La seva traducció al model relacional és:

```
empleats (#número, nom, data_naixement, sexe, telèfon, departament, data_assignació)
                                                    departaments
departaments (#codi, descripció)
```

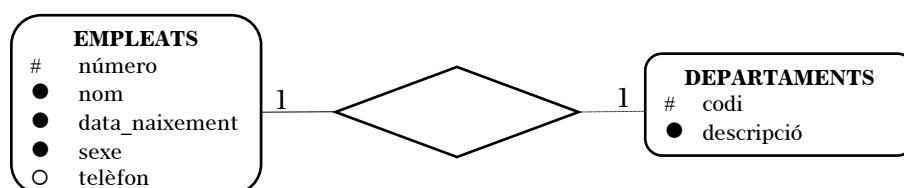
La relació DEPARTAMENTS apareix com a traducció de l'entitat d'igual nom, a l'igual que la relació EMPLEATS, però en aquesta, a més dels atributs corresponents a l'entitat (número, nom, data_naixement, sexe i telèfon) hi apareixen els atributs causats per la relació (1:N) entre les

entitats en el model entitat-relació: l'atribut departament com a clau forana de la relació DEPARTAMENTS i l'atribut data_assignació.

En l'exemple anterior l'atribut departament de la relació EMPLEATS no permet *valor nul* degut a que cada empleat és assignat obligatòriament a un departament, tal i com es representa en el model entitat-relació.

- Exemple per una relació (1:1)

En el següent model entitat-relació es modela la situació que s'estableix entre les mateixes entitats de l'apartat anterior sota el concepte de DIRECCIÓ de cada departament a càrrec d'un empleat.



En el model entitat-relació apareixen dues entitats i una relació (1:1). Per efectuar la traducció al model relacional tenim dues possibilitats, ja que hem d'escollir una de les entitats que participa amb grau 1 per a incloure, en la corresponent relació del model relacional, el conjunt d'atributs de la clau primària de l'altra entitat com a clau forana.

Si escollim l'entitat EMPLEATS obtenim les següents relacions on es pot observar que cada ocurrència d'EMPLEATS va acompanyada del departament del qual és director.

```
empleats (#número, nom, data_naixement, sexe, telèfon, dir_departament)
                                     departaments
departaments (#codi, descripció)
```

Si escollim l'entitat DEPARTAMENTS obtenim les següents relacions on es pot observar que cada ocurrència de DEPARTAMENTS va acompanyada de l'empleat que n'és el director.

```
empleats (#número, nom, data_naixement, sexe, telèfon)
departaments (#codi, descripció, director)
                                     empleats
```

Quina representació us sembla millor? Hi ha una resposta clara a aquesta pregunta: la segona. El motiu és que en la primera representació cada empleat va acompanyat del departament del qual n'és el director i, és clar, no tot empleat és director de departament, per la qual cosa en moltes ocasions hi haurà d'haver el *valor nul* en l'atribut dir_departament. En canvi, en la segona

representació és el departament qui va acompanyat per l'empleat que n'és el director i, per tant, no hi haurà *valors nuls*, és a dir, s'aprofitarà molt millor l'espai.

Dels raonaments de l'anterior paràgraf i de l'observació del model entitat-relació se'n dedueix que en una relació entre entitats en les que totes participen amb grau 1 escollirem com a entitat tal que la seva corresponent relació assumeixi les claus foranes de les altres entitats, si és possible, una entitat que hi participi amb caràcter no obligatori, de manera que es minimitzi la utilització de valors nuls. (!!)

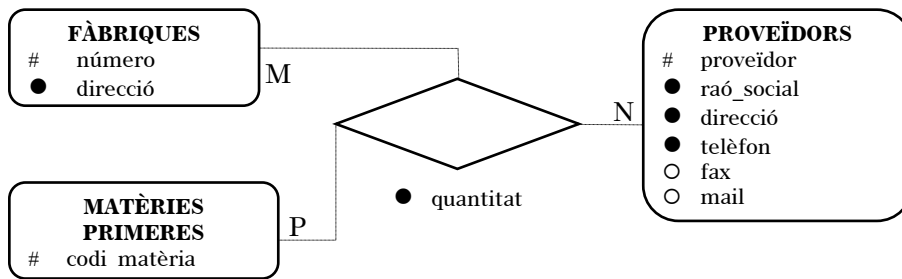
c) Relacions ternàries.

La transformació de les relacions ternàries té grans semblances amb la transformació de les binàries M:N, en el sentit que no és possible representar la relació amb l'aparició únicament de claus foranes (com succeeix en les relacions 1:1 o 1:N) sinó que es fa necessària l'aparició d'una nova relació.

Així doncs, la transformació d'una relació ternària sempre dona lloc a una nova relació, que contindrà com atributs els atributs de les claus primàries de les tres entitats relacionades i tots els atributs existents (si n'hi ha) a la pròpia relació. La clau primària de la nova relació depèn del grau en que les diverses entitats participen a la relació.

- Relació M:N:P
 - Es genera una nova relació, en el model relacional, que per clau primària, té els atributs de les claus primàries de les tres entitats que participen amb grau N a la relació del model entitat-relació.
 - Els subconjunts d'atributs de la nova clau primària provinents de les claus primàries de les entitats són claus foranes de les relacions generades per les corresponents entitats.
 - Els atributs existents (si n'hi ha) a la relació del model entitat-relació passen a ser atributs de la nova relació.

El següent model entitat-relació correspon a un petit control dels subministres que els diferents proveïdors de matèries primeres efectuen a les fàbriques gestionades per l'organització. Es suposa que donat un proveïdor pot subministrar diferents matèries primeres a diferents fàbriques, les quals poden ser subministrades, a la vegada, per diferents proveïdors.



En aquest exemple podem observar, en el model entitat-relació, tres entitats i una relació de grau (M:N:P). La seva traducció al model relacional és:

```

fàbriques (#número, direcció)
proveïdors (#proveïdor, raó_social, direcció, telèfon, fax, mail)
matèries_primeres (#codi_matèria, descripció)
subministres (#fàbrica, #proveïdor, #matèria, quantitat)
              fàbriques   proveïdors   matèries
              primeres
  
```

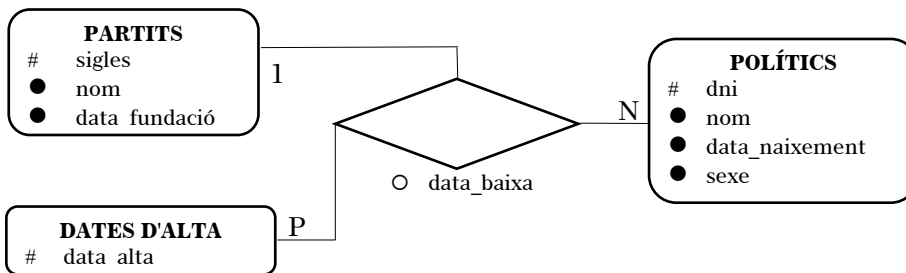
Les relacions FÀBRIQUES, PROVEÏDORS i MATÈRIES PRIMERES apareixen com a traducció de les entitats d'igual nom. La relació SUBMINISTRES però, apareix com a traducció de la relació d'igual nom entre les entitats.

La clau primària de la nova relació SUBMINISTRES està formada per (fàbrica, proveïdor, matèria), és a dir, les claus primàries de les entitats que participen amb grau N a la relació. L'atribut fàbrica és part de la clau primària i és clau forana de FÀBRIQUES. L'atribut proveïdor és part de la clau primària i és clau forana de PROVEÏDORS. L'atribut matèria és part de la clau primària i és clau forana de MATÈRIES PRIMERES.

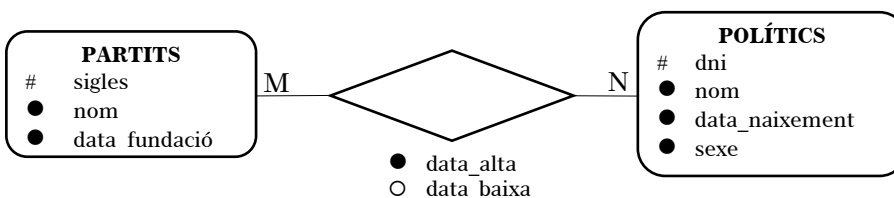
- Relació M:N:1
 - Es genera una nova relació, en el model relacional, que per clau primària, té els atributs de les claus primàries de les dues entitats que participen amb grau N a la relació del model entitat-relació. Els atributs de la clau primària de l'entitat que participa amb grau 1, passen a ser atributs de la nova relació.
 - Tots els atributs afegits a la nova relació provinents de les claus primàries de les diverses entitats, passen a ser claus foranes de les entitats de les que provenen. El caràcter d'obligatorietat de l'entitat que participa amb grau 1 a la relació en el model entitat-relació es correspon amb l'existència de *valor nul* pels atributs que genera la seva clau primària a la nova relació, de manera que "obligatorietat es correspon amb *no valor nul*" i "no obligatorietat es correspon amb *valor nul*".

- Els atributs existents (si n'hi ha) a la relació del model entitat-relació passen a ser atributs de la nova relació.

En el següent model entitat-relació es presenta un control històric i actual de les afiliacions dels polítics als diferents partits polítics. En concret es pretén tenir constància de totes les afiliacions d'un polític al llarg de la seva carrera política. Per cada afiliació es vol gestionar la data d'alta i la data de baixa, la qual no contindrà valor en cas que l'afiliació encara estigui activa. S'ha de tenir en compte que un polític pot afiliar-se a un mateix partit en diferents èpoques. Suposarem que en una mateixa data un polític no s'afilia a diferents partits.



Aquest exemple és un cas típic de situació en la que és necessària una entitat per representar la data d'alta ja que un polític pot estar afiliat diferents vegades a un mateix partit polític. Sense l'entitat DATES D'ALTA no es podria representar aquesta situació entre les entitats PARTITS i POLÍTICS. En efecte, si ho intentéssim apareixeria el model entitat-relació següent.



Però en aquest model, no hi ha manera de representar que un polític s'afilia a un mateix partit en més d'una ocasió. És a dir, un partit es pot aparellar amb molts polítics, i a l'inrevés, un polític es pot aparellar amb molts partits, però un aparellament partit - polític és únic i té una data d'alta i una data de baixa quan l'afiliació es dona per finalitzada.

Situacions com les de l'exemple s'acostumen a produir quan s'ha de gestionar informació en la que un mateix fet es pot produir en més d'una ocasió i la data en què es produeix és l'única forma d'identificar l'ocasió en què té lloc dit fet. La solució està en fer aparèixer una entitat DATA. A vegades, cal precisar molt més i no n'hi ha prou amb la data sinó que cal arribar a precisar l'hora i/o el minut i/o el segon. En tal situació

es fa aparèixer una entitat **MOMENT TEMPORAL**, on l'atribut arriba a la precisió necessària. **!!**

Així doncs, considerem el primer model entitat-relació en el què apareixen les tres entitats i la relació (M:N:1). La seva traducció al model relacional és:

```

partits (#sigles, nom, data_fundació)
polítics (#dni, nom, data_naixement sexe)
dates_alta (#data_alta)
afiliació (#dni, #data_alta, sigles, data_baixa)
           polítics dates_alta partits

```

Les relacions **PARTITS**, **POLÍTICS** i **DATES_ALTA** apareixen com a traducció de les entitats d'igual nom. La relació **AFILIACIÓ** apareix com a traducció de la relació d'igual nom entre les entitats esmentades.

La clau primària de la nova relació **AFILIACIÓ** està formada per (dni, data_alta), és a dir, les claus primàries de les entitats que participen amb grau diferent d'1 a la relació. L'atribut dni és part de la clau primària i és clau forana de **POLÍTICS**. L'atribut data_alta és part de la clau primària i és clau forana de **DATES_ALTA**. L'atribut sigles no és part de la clau primària ja que l'entitat **PARTITS** participa amb grau 1 a la relació entre les entitats. En canvi l'atribut sigles és clau forana de **PARTITS**.

Més amunt hem comentat la necessitat de l'entitat **DATES D'ALTA** en el model entitat-relació. L'estricta aplicació de la traducció del model entitat-relació al model relacional fa aparèixer la relació **DATES_ALTA** formada per un únic atribut que és la clau primària. Es pot observar que l'existència d'aquesta relació no aporta cap informació. Per tant, la podem obviar i obtenim:

```

partits (#sigles, nom, data_fundació)
polítics (#dni, nom, data_naixement sexe)
afiliació (#dni, #data_alta, sigles, data_baixa)
           polítics          partits

```

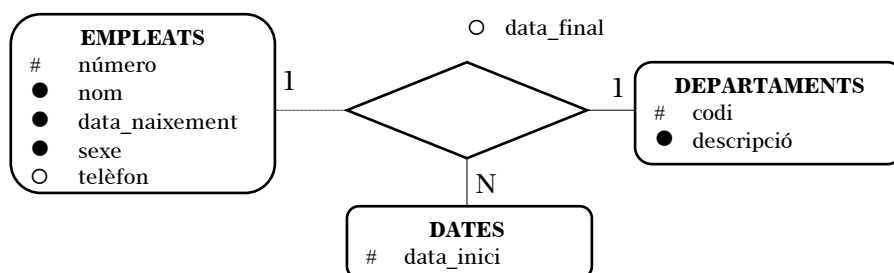
Així doncs, la relació **AFILIACIÓ** manté la clau primària (dni, data_alta) però el segon atribut deixa de ser clau forana de cap entitat.

D'aquest exemple en deduïm que l'entitat **MOMENT TEMPORAL** que en ocasions ens veiem obligats a introduir en el model entitat-relació, tal i com més amunt hem raonat, es pot suprimir en la traducció al model relacional si no aporta més informació que el propi *moment temporal*.

- Relació N:1:1

- Es genera una nova relació, en el model relacional, que per clau primària, té els atributs de la clau primària de l'entitat que participa amb grau N a la relació del model entitat-relació i els atributs d'una de les dues entitats que participen amb grau 1 (tenim, per tant, dues claus candidates). Els atributs de la clau primària de l'altra entitat que participa amb grau 1, passen a ser atributs de la nova relació.
- Tots els atributs afegits a la nova relació provinents de les claus primàries de les diverses entitats, passen a ser claus foranes de les entitats de les que provenen. El caràcter d'obligatorietat de l'entitat que participa amb grau 1 a la relació en el model entitat-relació i no forma part de la clau primària de la nova relació, es correspon amb l'existència de *valor nul* pels atributs que genera la seva clau primària a la nova relació, de manera que "obligatorietat es correspon amb *no valor nul*" i "no obligatorietat es correspon amb *valor nul*".
- Els atributs existents (si n'hi ha) a la relació del model entitat-relació passen a ser atributs de la nova relació.

El següent model entitat-relació modela el concepte de DIRECCIÓ d'un departament en una empresa, tenint en compte que es vol tenir constància de la direcció dels departaments en els diversos períodes de la seva història, tenint en compte que un departament pot ser dirigit pel mateix cap en diversos períodes.



En el model entitat-relació apareixen tres entitats i una relació (1:1:N).

En la traducció al model relacional, com ja s'ha comentat més amunt, l'entitat DATES no té cap interès si no aporta més informació i, per tant, no la considerarem en la traducció. Tenim dues possibles traduccions:

```

empleats (#número, nom, data_naixement, sexe, telèfon,
departaments (#codi, descripció)
direcció (#data_inici, #departament, empleat, data_final)
                                departaments  empleats
  
```

```

empleats (#número, nom, data_naixement, sexe, telèfon,
departaments (#codi, descripció)
  
```

Se'ns presenta un problema, però, quan les dues entitats que participen amb grau 1 en el model entitat-relació, ho fan amb caràcter no obligatori.

En tal situació, és millor canviar el disseny en el model entitat-relació i transformar la relació ternària en dues relacions binàries.

Suposem el cas de la relació per representar, per a cada infant, qui és el seu pare i qui és la seva mare. Es tracta de la relació:

INFANT(N) – PARE(1) – MARE(1)
que ha de permetre el fet de deixar sense informar el pare o la mare d'un infant.

La seva traducció al model relacional, on el pare o la mare han de formar part de la clau primària, no admet valors nuls per aquest atribut. La solució radica en considerar dues relacions binàries:

INFANT(N) – PARE(1)
INFANT(N) – MARE(1)

La traducció d'aquestes relacions sí permeten el fet de deixar sense informar el pare o la mare d'un infant o, fins i tot, ambdós.

defensa (#estudiant, #tribunal, projecte, data_defensa)
 estudiants tribunals projectes

defensa (#estudiant, #projecte, tribunal, data_defensa)
 estudiants projectes tribunals

defensa (#projecte, #tribunal, estudiant, data_defensa)
 projectes tribunals estudiants

d) Relacions n-àries.

La transformació de les relacions n-àries es pot veure com una generalització del que hem vist per les ternàries:

- En aquesta situació sempre apareix una nova entitat.
- Si totes les entitats hi participen amb grau N (cas N:N:N:N:...), es crea una nova relació que té per clau primària, els atributs de les claus primàries de totes les entitats.
- Si hi ha una o més entitats (del total n) que hi participen amb grau 1, es crea una nova relació que té per clau primària, els atributs de les claus primàries de n-1 de les entitats relacionades, amb la condició que l'entitat per la que no s'inclou la seva clau primària, ha de ser una de les que participa amb grau 1. Els atributs de la clau primària de l'entitat que hi participa amb grau 1, passen a ser atributs de la nova relació.
- Tots els atributs afegits a la nova relació a causa dels dos anteriors paràgrafs passen a ser claus foranes de les entitats de les que provenen. El caràcter d'obligatorietat de l'entitat que participa amb grau 1 a la relació en el model entitat-relació i no forma part de la clau primària de la nova relació es correspon amb l'existència de *valor nul* pels nous atributs, de manera que "*obligatorietat*" es correspon amb *no valor nul*" i "*no obligatorietat*" es correspon amb *valor nul*".
- Els atributs existents (si n'hi ha) a la relació del model entitat-relació passen, també, a ser nous atributs de la nova relació.

3) Transformació de les entitats fortes-dèbils

La traducció de les entitats fortes-dèbils del model entitat-relació al model relacional és molt semblant a la traducció d'una relació de grau (1:N).

- L'entitat forta es tradueix en una relació que manté els atributs de l'entitat d'origen.

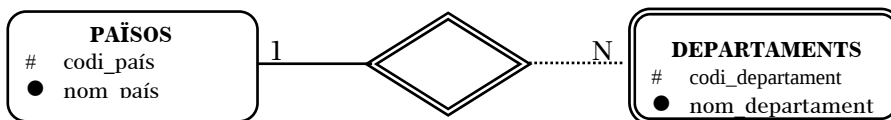
Se'ns presenta un problema, però, quan hi ha més d'una entitat que participa amb grau 1 a la relació en el model entitat-relació que ho fa amb caràcter no obligatori, ja que això obligaria a que els atributs de la nova clau primària heretats de les entitats que participen amb caràcter no obligatori a la relació en el model relacional, admetessin valors nuls, fet no possible en una clau primària.

En tal situació, és millor canviar el disseny en el model entitat-relació i transformar la relació n-ària en un conjunt d'altres relacions on aparegui alguna relació binària que permeti representar la no obligatorietat de l'entitat que participa amb grau 1.

No s'ha de caure en el parany de pensar que cal utilitzar sempre relacions binàries... Hi ha ocasions en les que la única utilització de relacions binàries no permet modelar convenientment la realitat.

- L'entitat dèbil, en canvi, es tradueix en una relació que manté tots els atributs de l'entitat d'origen però que té per clau primària la parella formada pel conjunt d'atributs de la clau primària de l'entitat forta (que es constitueixen en clau forana) i pel conjunt d'atributs de la clau primària de l'entitat dèbil. No es pot oblidar que una entitat és dèbil quan les seves ocurrences necessiten d'una entitat de referència (entitat forta) per a identificar-se.

Considerem, com exemple, la traducció del següent model entitat-relació on apareix l'entitat DEPARTAMENTS com a entitat feble de l'entitat PAÏSOS.



La traducció al model relacional obté:

```

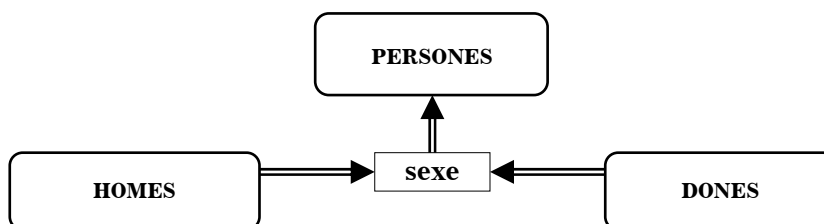
països (#codi_país, nom_país)
departaments (#codi_país, #codi_departament, nom_departament)
                països
  
```

4) Transformació de les particions en el model entitat-relació estàs

La traducció d'una partició d'una entitat en altres entitats, la qual es produeix a partir d'un cert atribut de l'entitat mare que permet distribuir totes les seves ocurrences entre les entitats que formen la partició, consisteix en:

- L'entitat mare es tradueix en una relació que manté els atributs de l'entitat mare i, evidentment, l'atribut en què es basa la partició.
- Cada entitat de la partició es tradueix en una relació que manté els atributs però que té com a clau primària la mateixa clau primària que l'entitat mare, la qual serà clau forana.

Considerem, com exemple, l'entitat PERSONES particionada en les entitats DONES i HOMES, sota l'atribut *sexe*.



De la traducció al model relacional s'obté:

```

persones (#codi_persona, nom, cognom1, cognom2, sexe, direcció,...)
dones (#codi_persona, "atributs específics per les dones")
      persones
homes (#codi_persona, "atributs específics pels homes")
      persones

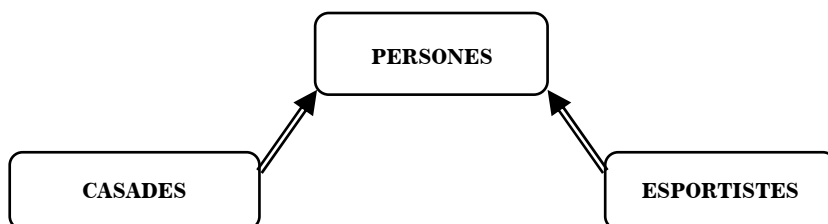
```

5) Transformació dels subconjunts d'entitats en el model entitat-relació estès

La traducció de subconjunts (disjunts o no) d'una entitat, consisteix en:

- L'entitat mare es tradueix en una relació que manté els atributs de l'entitat mare.
- Cada entitat subconjunt es tradueix en una relació que manté els atributs però que té com a clau primària la mateixa clau primària que l'entitat mare, la qual serà clau forana.

Considerem, com exemple, l'entitat PERSONES amb les entitats subconjunts CASADES i ESPORTISTES.



De la traducció al model relacional s'obté:

```

persones (#codi_persona, nom, cognom1, cognom2, sexe, direcció,...)
casades (#codi_persona, "atributs específics per les persones casades")
      persones
esportistes (#codi_persona, "atributs específics per les persones esportistes")
      persones

```

2.2. Exemple global

Volem informatitzar el circuit de vendes d'una empresa que ven uns determinats productes.

Una venda s'inicia quan un client efectua una comanda la qual és atesa pels empleats del departament comercial. Totes les vendes s'efectuen a clients de l'empresa i, per tant, els clients estan codificats. De cada client és obligat conèixer la seva raó social i és convenient conèixer la seva direcció, telèfon, fax i el nom i càrrec de la persona de contacte. Ara bé, un client pot efectuar la comanda i no ser ell el destinatari de la mercaderia, tot i que serà ell el responsable del facturat i del corresponent pagament.

Les comandes acceptades pels empleats del departament comercial reben, de manera automàtica, un número correlatiu que las identifica dins els circuits d'informació. Interessa deixar constància de la data en què s'ha efectuat la comanda així com l'empleat que l'ha acceptat i que en serà l'encarregat del posterior seguiment.

Les comandes acceptades estan formades per un o més articles dels que comercialitza l'empresa. Els productes comercialitzats tenen un codi identificatiu designat per la pròpia empresa i, obligatòriament, una descripció. Poden ser fabricats per la pròpia empresa o subministrats per diferents proveïdors. Interessa, com a molt, conèixer el darrer proveïdor que ha subministrat un determinat producte. Així mateix, l'empresa té definides unes determinades categories de productes, de manera que cada article pot pertànyer, com a molt, a una d'aquestes categories. Cada categoria té, obligatòriament, un nom i disposa d'un espai per a efectuar-ne una breu explicació, així com la possibilitat d'incloure un logotip -per cada categoria- que pugui acompanyar els corresponents articles i així fer-los més ràpidament identificables.

L'empresa assigna, a cada empleat, un codi i, possiblement, un càrrec i un tractament (Sra., Sr., Srta., D.,...). Interessa també poder mantenir les dates de naixement i de contractació, així com la seva direcció, el telèfon personal i, a ser possible, la seva foto. Evidentment, cada empleat disposa d'una extensió telefònica dins l'empresa, que ha de ser consultable. Seria molt interessant poder disposar d'un espai per incloure diverses observacions per cada empleat.

L'empresa també té codificats els seus proveïdors de manera semblant als clients. És a dir, de cada proveïdor és obligat conèixer la seva raó social i és convenient conèixer la seva direcció, telèfon, fax i el nom i càrrec de la persona de contacte. La majoria dels proveïdors de l'empresa disposen, en l'actualitat, de pàgines WEB, de manera que es vol mantenir la direcció de la seva pàgina principal en el WEB.

Dels productes sol·licitats en una comanda interessa conèixer el preu unitari que s'estableix, la quantitat que se'n demana i el descompte que s'efectuarà, si és el cas. En acceptar una comanda, l'empleat encarregat li assigna una data prevista de lliurament que cal intentar complir. Es vol tenir constància d'aquest fet i es demana poder emmagatzemar la data en què s'efectua la tramesa de la mercaderia així com l'agència de transport que l'efectua. En aquest punt cal saber que l'empresa treballa amb unes determinades agències de transport, les quals estan codificades i de les que cal saber el nom i, si és possible, el telèfon de contacte.

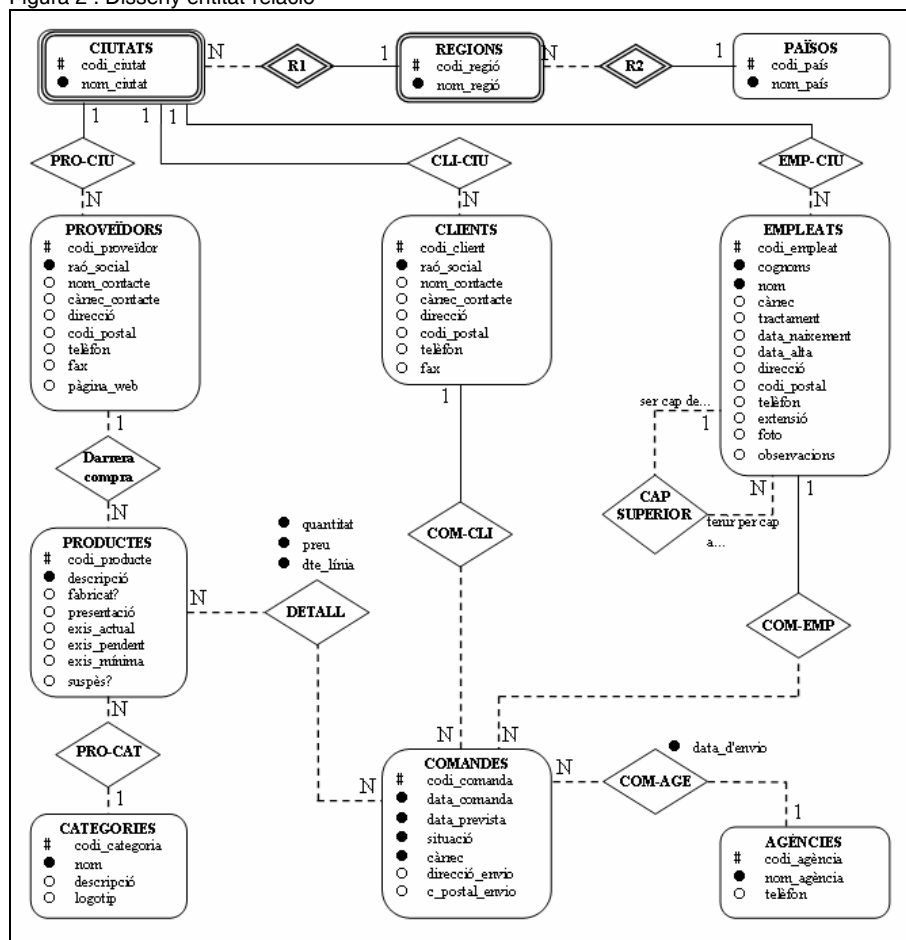
Tota comanda es serveix en la seva totalitat o no es serveix, és a dir, no hi ha la possibilitat de fer lliuraments parcials de mercaderia. A més, tota comanda pot portar un càrrec afegit causat pel total d'impostos a aplicar.

El departament comercial està estructurat jeràrquicament, de manera que cada empleat pot tenir un cap que ha de ser, a la vegada, empleat de l'empresa. Aquesta estructura jeràrquica té, evidentment, un(s) cap(s) superior(s) que no depenen de ningú. Interessa mantenir, per cada empleat, el seu cap, de manera que donat un cap es pugui saber tots els empleats que té al seu càrrec.

Dels productes que l'empresa comercialitza interessa tenir, també, la presentació (és a dir, el contingut d'una unitat de compra - venda) i l'existència actual (en unitats de compra - venda) així com la quantitat demanada a proveïdor i que està pendent de recepció. Es vol tenir, també, la quantitat mínima a comprar als proveïdors. A la vegada, un article -tingui o no existència- ha de poder-se marcar com article suspès per a no permetre la seva compra - venda fins que sigui desmarcat.

La figura 2 ens mostra el disseny entitat-relació facilitat pels analistes per donar resposta a la situació.

Figura 2 . Disseny entitat-relació



El model relacional que s'obté de l'anterior ER és:

```

països (#codi_pais, nom_país)
regions (#codi_país, #codi_regió, nom_regió)
    països
ciutats (#codi_país, #codi_regió, #codi_ciutat, nom_ciutat)
    regions
proveïdors (#codi_proveïdor, raó_social, nom_contacte, càrrec_contacte, direcció,
    codi_postal, telèfon, fax, pàgina_web, país, regió, ciutat)
    ciutats
clients (#codi_client, raó_social, nom_contacte, càrrec_contacte, direcció, codi_postal,
    telèfon, fax, país, regió, ciutat)
    ciutats
empleats (#codi_empleat, cognoms, nom, càrrec, tractament, data_naixement, data_alta,
    direcció, codi_postal, telèfon, extensió, foto, observacions, país, regió, ciutat,
    tenir_per_cap)
    ciutats
categories (#codi_categoria, nom, descripció, logotip)
productes (#codi_producte, descripció, fabricat?, presentació, exis_actual, exis_pendent,
    exis_mínima, suspès?, darrer_proveïdor, categoria)
    proveïdors categories
agències (#codi_agència, nom_agència, telèfon)
comandes (#codi_comanda, data_comanda, data_prevista, situació, càrrec, direcció_tramesa,
    c_postal_tramesa, client, empleat, agència, data_tramesa)
    clients empleats agències
detall (#producte, #comanda, quantitat, preu, dte_línia)
    productes comandes

```

3. Llenguatge procedural: àlgebra relacional

En les bases de dades podem trobar dos tipologies de llenguatges en funció de la manera que l'usuari haurà d'accedir a la base de dades:

- *implícits*, on l'usuari demana les dades a accedir però no diu la forma d'accedir-hi; també s'anomenen *no procedurals* ja que no s'indica el procediment per accedir a les dades,
- *explícits*, on l'usuari indica la forma d'accedir a les dades desitjades; també s'anomenen *procedurals* ja que s'indica el procediment a seguir per accedir a les dades.

El model relacional incorpora l'**àlgebra relacional** com a llenguatge procedural.

L'àlgebra relacional és un conjunt d'operacions amb relacions que permeten obtenir altres relacions. És a dir, són **operacions tancades**, ja que operen elements del conjunt de les relacions i obtenen, com a resultat, un altre element del mateix conjunt. Això permet que el resultat d'una operació pugui ser utilitzada com a operand de qualsevol altre operació.

L'àlgebra relacional fou introduïda per E.F. Codd i estava formada, originalment, per vuit operacions, quatre de les quals es podrien anomenar **conjuntistes** degut a que es corresponen amb operacions tradicionals de la teoria matemàtica de conjunts i les altres es podrien anomenar **relacionals** per ser específiques del model relacional. Amb aquestes vuit operacions no es pot efectuar totes les operacions de consulta i, per aquest motiu, alguns autors han afegit unes operacions **addicionals**.

Referència bibliogràfica als treballs de Codd

E.F. Codd (1972). Relational Completeness of Data Base Languages. Data Base Systems, Courant Computer Science Symposia Series. Vol. 6. Prentice-Hall. Pàgs. 65-98.

3.1. Operacions conjuntistes

Les operacions de l'àlgebra relacional que es corresponen amb la teoria matemàtica de conjunts són: **unió**, **intersecció**, **diferència** i **producte cartesià**. Les operacions **unió**, **intersecció** i **diferència** del model relacional són més restrictives que les existents en la teoria matemàtica de conjunts, ja que en el model relacional s'obliga a que les relacions a operar siguin **compatibles**.

Anomenem **atributs compatibles** aquells atributs definits sobre el mateix domini semàntic. Anomenem **relacions compatibles** aquelles que tenen el mateix grau i els atributs compatibles dos a dos.

Que dos atributs estiguin definits sobre el mateix domini semàntic vol dir que el domini, com a conjunt de valors, sigui comú i que el significat que poden prendre dits valors sigui també comú. Així, podem tenir els dos atributs *pes* i *altura* definits sobre el conjunt dels nombres reals positius (comú per tots dos) però no són compatibles doncs el significat dels dos atributs no és el mateix.

En operar relacions compatibles, la nova relació tindrà per esquema un esquema compatible amb el de les relacions operades. Pot prendre's l'esquema de qualsevol de les relacions operades com a esquema de la nova relació. (!)

Donades dues **relacions R i S compatibles**, es defineix la **unió de R i S**, i es nota per $R \cup S$, com la relació constituïda pels tuples que formen part de la relació R, de la relació S o d'ambdues relacions.

És a dir, són els tuples resultat de la unió dels conjunts de tuples de R i S tal i com defineix la teoria matemàtica de conjunts. Cal recordar que la unió de conjunts elimina els tuples repetits (és a dir, no és equivalent a la suma de conjunts).

Donades dues **relacions R i S compatibles**, es defineix la **intersecció de R i S**, i es nota per $R \cap S$, com la relació constituïda pels tuples que formen part de les dues relacions R i S simultàniament.

És a dir, són els tuples resultat de la intersecció dels conjunts de tuples de R i S com defineix la teoria matemàtica de conjunts.

Donades dues **relacions R i S compatibles**, es defineix la **diferència de R menys S**, i es nota per $R - S$, com la relació constituïda pels tuples de R que no formen part de la relació S.

És a dir, són els tuples resultat de la diferència dels conjunts de tuples de R i S com es defineix en la teoria matemàtica de conjunts.

Per definir la darrera de les operacions conjuntistes, no es necessita que les relacions a operar siguin compatibles, però es necessita, en canvi, haver introduït el concepte de concatenació de tuples.

Donats dos tuples $t_a = (a_1, a_2, a_3, \dots, a_n)$ i $t_b = (b_1, b_2, b_3, \dots, b_m)$ es defineix la concatenació de t_a i t_b , i ho notarem $t_a \circ t_b$, com el tuple format pels elements de t_a seguits dels elements de t_b . És a dir:

$$t_a \circ t_b = (a_1, a_2, a_3, \dots, a_n, b_1, b_2, b_3, \dots, b_m)$$

Ara ja estem en disposició de definir la darrera operació conjuntista.

Donades dues **relacions** R i S , es defineix el **producte cartesià de R i S** , i es nota per $R \times S$, com el conjunt de tuples formats per la concatenació dels tuples de R amb els tuples de S .

Recordem que en el model relacional, l'ordre dels atributs és irrellevant. Per tant, en el producte cartesià el que importa és el conjunt d'atributs que el formen.

El fet que dos atributs siguin o no compatibles no té res a veure amb el nom dels atributs. Suposarem, però, en els exemples que segueixen, que atributs amb igual nom són atributs compatibles.

Considerem les relacions R i S . Utilitzem-les per a observar el funcionament de les quatre operacions conjuntistes presentades.

$R \cap S$	A	B	C
	a	1	α
	a	2	β

$R - S$	A	B	C
	b	1	α

R	A	B	C
	a	1	α
	a	2	β
	b	1	α

$R \cup S$	A	B	C
	a	1	α
	a	2	β
	b	1	α
	c	2	γ
	d	1	γ
	a	1	β

S	A	B	C
	c	2	γ
	d	1	γ
	a	2	β
	a	1	α
	a	1	β

$R \times S$	A _R	B _R	C _R	A _S	B _S	C _S
	a	1	α	c	2	γ
	a	2	β	c	2	γ
	b	1	α	c	2	γ
	a	1	α	d	1	γ
	a	2	β	d	1	γ
	b	1	α	d	1	γ
	a	1	α	a	2	β
	a	2	β	a	2	β
	b	1	α	a	2	β
	a	1	α	a	1	α
	a	2	β	a	1	α
	b	1	α	a	1	α
	a	1	α	a	1	β
	a	2	β	a	1	β
	b	1	α	a	1	β

3.2. Operacions relacionals

Les operacions purament relacionals que va introduir E. F. Codd són: **selecció, projecció, combinació i divisió**.

La **selecció**, també anomenada **restricció**, és l'operació que permet efectuar "talls horitzontals" en una relació, és a dir, és l'operació que permet seleccionar un subconjunt de tuples d'entre els tuples que formen la relació. Els tuples seleccionats s'identifiquen per una certa condició que han de verificar. Les condicions permeses són les provinents dels operadors relacionals $<$, $<=$, $>$, $>=$, $=$, $\neq$ i la utilització d'ells juntament amb els operadors lògics $\wedge$, $\vee$, $\neg$.

Donada una **relació R**, es defineix la **selecció de R segons una condició c**, i es nota per **$R(c)$** , com la relació constituïda pels tuples de R que verifiquen la condició c.

La **projecció** és l'operació que permet efectuar "talls verticals" en una relació, és a dir, és l'operació que permet seleccionar un subconjunt d'atributs (columnes) per tots els tuples de la relació.

Donada una **relació R**, es defineix la **projecció de R pels atributs $A_1, \dots, A_k$** , i es nota per **$R[A_1, \dots, A_k]$** , com la relació constituïda pels atributs $A_1, \dots, A_k$ de tots els tuples de R.

Observem que les relacions no contenen tuples duplicats i que, en efectuar talls verticals -projeccions- en poden aparèixer. La pròpia operació de projecció elimina els duplicats. **!!**

Considerem la relació S següent i observem uns exemples de seleccions i projeccions sobre S.

S	C	D
	α	5
	γ	2
	α	7
	ε	8

S (C $\geq$ α i D $\geq$ 6)			C	D
			α	7
			ε	8

S [C]	C
	α
	γ
	ε

La tercera operació relacional és la **combinació**, anomenada en moltes ocasions **reunió** o **join** (terme anglès molt utilitzat). En realitat hi ha diferents tipus de combinacions, sent el més comú el **θ -join**, (pronunciat theta-join). Els altres tipus són l'**equi-join** i el **natural-join**.

Les combinacions són les operacions que permeten obtenir subconjunts de tuples del producte cartesià de dues relacions segons una certa condició relacional entre un atribut de cada relació.

Donades dues relacions **R** i **S**, es defineix el **θ -join de R segons l'atribut A i S segons l'atribut Z**, i es nota per **$R[A\theta Z]S$** com el subconjunt de tuples del producte cartesià $R \times S$ que verifiquen $A \theta Z$ on θ és qualsevol dels operadors relacionals ($>$, $>=$, $<$, $<=$, $=$, $\neq$).

L'**equi-join** és un **θ -join** en el que l'operador θ és la igualtat. Es nota com **$R[A=Z]S$** .

El **natural-join** és un equi-join en el que l'atribut pel que s'executa la combinació només apareix una vegada. Es nota com **$R[A*Z]S$** .

La notació **$R*S$** indica el natural-join per tots els atributs del mateix nom en les dues relacions. (!!)

Exemplifiquem diferents tipus de combinació a partir de les relacions R i S següents. Per l'atribut C, cal considerar l'ordre $\alpha < \beta < \gamma < \delta < \epsilon$.

R	A	B	C
	1	a	α
	2	a	β
	3	a	γ
	1	b	δ

S	C	D
	α	5
	γ	2
	α	7
	ϵ	8

$R[C<C]S$	A	B	C_R	C_S	D
	1	a	α	γ	2
	1	a	α	ϵ	8
	2	a	β	γ	2
	2	a	β	ϵ	8
	3	a	γ	ϵ	8
	1	b	δ	ϵ	8

$R[C=C]S$	A	B	C_R	C_S	D
	1	a	α	α	5
	1	a	α	α	7
	3	a	γ	γ	2

$R[C*C]S$	A	B	C	D
	1	a	α	5
	1	a	α	7
	3	a	γ	2

L'operació relacional **divisió** de dues relacions obté una relació tal que el producte cartesià d'ella per la relació divisora és la relació dividend.

Donades dues relacions **$R(A_1, A_2, \dots, A_M, B_1, B_2, \dots, B_N)$** i **$S(B_1, B_2, \dots, B_N)$** on els atributs $B_1, B_2, \dots, B_N$ de les dues relacions són compatibles, es defineix la relació **divisió de R entre S**, i es nota

per $R[B_1, B_2, \dots, B_N : B_1, B_2, \dots, B_N]S$, com la relació formada pels atributs $A_1, A_2, \dots, A_M$ de R i el major conjunt de tuples de R (atributs $A_1, A_2, \dots, A_M$) de manera que el producte cartesià d'aquesta nova relació amb la relació S obté tuples incloses dins la relació R .

Recordem que en el model relacional l'ordre dels atributs és irrellevant (a tenir en compte en comprovar si el producte cartesià de les tuples de la relació resultat per les tuples de la relació divisor obté tuples incloses dins la relació dividend). (!!)

A l'exemple de la dreta observem que la divisió $R[C:D]S$ està formada només pel tuple (α) . El producte cartesià de S amb aquesta relació obté tuples inclosos dins R . El tuple (β) no pot formar part de la relació ja que del producte cartesià n'apareixeria (x, β) que no és tuple de R .

S	D	R	C	B	$R[C:D]S$	B
	x		x	α		α
	y		y	α		
			z	α		

A la situació de la dreta, la divisió $R[B,C:D,E]S$ és una relació amb l'atribut A però amb cap tuple, és a dir, una relació buida.

R	A	B	C	S	D	E
	α	10	x		10	x
	α	20	y		20	y
	α	30	z		10	y
	β	40	x			

3.3. Operacions addicionals

Ja hem introduït més amunt que algunes operacions de consulta no són factibles amb l'única utilització de les operacions conjuntistes i relacionals introduïdes per E. F. Codd i, per aquest motiu, alguns autors presenten algunes operacions addicionals. En considerarem quatre: **tornar a anomenar**, **ampliar**, **resumir** i **combinació externa**.

Donat un atribut A d'una relació R , l'operació **tornar a anomenar** A , que ho notarem com $R(\text{tornar a anomenar } A \text{ com } A')$, consisteix en obtenir una nova relació R' , còpia de R , amb el nom de l'atribut A canviat per A' .

L'operació **tornar a anomenar** permet canviar el nom dels atributs de les relacions, fet necessari abans d'efectuar operacions com el producte cartesià o com les combinacions quan en les relacions a operar hi ha atributs amb el mateix nom.

Donada una relació R , l'operació **ampliar**, que ho notarem com **$R(\text{ampliar } \langle \text{expr} \rangle \text{ com } A)$** , consisteix en obtenir una nova relació R' , amb els atributs existents a R i amb un nou atribut A calculat a partir de l'execució de l'expressió $\langle \text{expr} \rangle$ sobre cada tuple de R .

L'operació **ampliar** permet ampliar les columnes de R amb columnes calculables a partir de les columnes existents.

Donat un atribut A d'una relació R , l'operació **resumir**, que ho notarem com **$R(\text{per } A \text{ resumir } \langle \text{oper} \rangle (B) \text{ com } B')$** , consisteix en obtenir una nova relació R' formada únicament pels atributs A i B' , on els tuples s'obtenen de totes les agrupacions dels tuples de R pels diferents valors d' A calculant B' a partir de l'operació $\langle \text{oper} \rangle$ (suma, mitja, màxim, mínim, comptar) sobre tots els valors de B per cada agrupació.

L'operació **resumir** permet efectuar agrupacions de tuples que tenen un mateix valor per un determinat atribut (o conjunt d'atributs) i calcular-ne unes operacions "resum" sobre uns nous atributs que emmagatzemen els resums obtinguts. Com operacions "resum" tenim **suma**, **mitja**, **màxim** i **mínim** que actuen sobre un atribut numèric i **comptar** que no actua sobre cap atribut i es simplement compta la quantitat de tuples de cada grup.

Vegem alguns exemples de les operacions **tornar a anomenar**, **ampliar** i **resumir**.

R	A	B	C
	α	10	x
	α	20	y
	α	30	z
	β	40	x

$R(\text{tornar a anomenar } A \text{ com } Z)$	Z	B	C
	α	10	x
	α	20	y
	α	30	z
	β	40	x

$R(\text{ampliar } B*2 \text{ com } Z)$	A	B	C	Z
	α	10	x	20
	α	20	y	40
	α	30	z	60
	β	40	x	80

$R(\text{per } A \text{ resumir suma}(B) \text{ com } S, \text{màxim}(B) \text{ com } M)$	A	S	M
	α	60	30
	β	40	40

L'operació **combinació externa**, també anomenada **outer join**, és una ampliació de l'equi-join proposat Codd. En certes ocasions és menester tenir el resultat de l'equi-join ampliat amb tots els tuples d'una de les relacions que no tenen corresponent tuple en l'altre relació.

Donades dues relacions **R** i **S**, es defineix el **left-outer-join de R segons l'atribut A i S segons l'atribut Z**, i es nota per **$R[\bullet A=Z]S$** , com el subconjunt de tuples del producte cartesià $R \times S$ que verifiquen $A = Z$ (resultat de $R[A=Z]S$) més els tuples de **R** que no tenen, per l'atribut A correspondència amb cap tuple de **S** segons l'atribut Z, els quals presenten valors NULS en els atributs provinents de **S**.

Donades dues relacions **R** i **S**, es defineix el **right-outer-join de R segons l'atribut A i S segons l'atribut Z**, i es nota per **$R[A=Z\bullet]S$** , com el subconjunt de tuples del producte cartesià $R \times S$ que verifiquen $A = Z$ (resultat de $R[A=Z]S$) més els tuples de **S** que no tenen, per l'atribut Z correspondència amb cap tuple de **R** segons l'atribut A, els quals presenten valors NULS en els atributs provinents de **R**.

Observem exemples de combinacions externes:

R	A	B	C
	1	a	α
	2	a	β
	3	a	γ
	1	b	δ

S	C	D
	α	5
	γ	2
	α	7
	ε	8

$R[\bullet C=C]S$	A	B	C_R	C_S	D
	1	a	α	α	5
	1	a	α	α	7
	3	a	γ	γ	2
	2	a	β	NUL	NUL
	1	b	δ	NUL	NUL

$R[\bullet C^*C]S$	A	B	C	D
	1	a	α	5
	1	a	α	7
	3	a	γ	2
	2	a	β	NUL
	1	b	δ	NUL

$R[C=C\bullet]S$	A	B	C_R	C_S	D
	1	a	α	α	5
	1	a	α	α	7
	3	a	γ	γ	2
	NUL	NUL	NUL	ε	8

$R[C^*C\bullet]S$	A	B	C	D
	1	a	α	5
	1	a	α	7
	3	a	γ	2
	NUL	NUL	ε	8

3.4. Exemples

Reprenem un model relacional ja presentat amb anterioritat:

```

països (#codi_país, nom_país)
regions (#codi_país, #codi_regió, nom_regió)
      països
ciutats (#codi_país, #codi_regió, #codi_ciutat, nom_ciutat)
      regions
proveïdors (#codi_proveïdor, raó_social, nom_contacte, càrrec_contacte, direcció,
      codi_postal, telèfon, fax, pàgina_web, país, regió, ciutat)
      ciutats
clients (#codi_client, raó_social, nom_contacte, càrrec_contacte, direcció, codi_postal,
      telèfon, fax, país, regió, ciutat)
      ciutats
empleats (#codi_empleat, cognoms, nom, càrrec, tractament, data_naixement, data_alta,
      direcció, codi_postal, telèfon, extensió, foto, observacions, país, regió, ciutat,
      tenir_per_cap)
      ciutats
categories (#codi_categoria, nom, descripció, logotip)
productes (#codi_producte, descripció, fabricat?, presentació, exis_actual, exis_pendent,
      exis_mínima, suspès?, darrer_proveïdor, categoria)
      proveïdors      categories
agències (#codi_agència, nom_agència, telèfon)
comandes (#codi_comanda, data_comanda, data_prevista, situació, càrrec, direcció_tramesa,
      c_postal_tramesa, client, empleat, agència, data_tramesa)
      clients      empleats      agències
detall (#producte, #comanda, quantitat, preu, dte_línia)
      productes      comandes

```

Anem a proposar algunes consultes i a resoldre-les amb la utilització de l'àlgebra relacional. Notarem per RR la relació resultat per cada consulta.

a) Quins són els productes (només codi i descripció) actualment suspesos.

Aquesta informació es troba a la relació *productes*, en la que hem de filtrar els tuples pels que l'atribut *suspès?* té el valor *cert* i únicament hem de presentar els atributs *codi_producte* i *descripció*. Per tant, primer cal efectuar una selecció dels tuples corresponents i, posteriorment, una projecció dels atributs a presentar.

```

R1 = productes (suspès?==cert)
RR = R1 [codi_producte, descripció]

```

Observem que la solució s'ha obtingut amb dos passos, però s'hagués pogut incloure tot en una sola línia, de la manera que segueix.

```

RR = (productes (suspès?==cert))[codi_producte, descripció]

```

En els exemples que seguiran es presentaran els diferents passos fins arribar a la solució RR per tal de facilitar-ne la llegibilitat.

b) Quins regions de "França" existeixen a la base de dades.

La informació que se'ns demana resideix a la relació *regions*, on hi ha el *codi* i *nom* de les regions dels diferents països. Ara bé, el *país* hi apareix codificat i nosaltres no sabem el codi que correspon a "França". Necessitem, per tant, vincular la informació de *regions* amb la informació de *països* on sí existeix el *nom* del país.

Tenim diferents possibilitats per assolir el resultat desitjat.

- Possibilitat 1.

```
R1 = països (majúscules(nom_país) == "FRANÇA")
/* S'obté, via selecció, els països que tenen per nom "França". */
R2 = R1 [codi_país]
/* Es restringeix, via projecció, a la columna codi_país. */
R3 = R2 (tornar a anomenar codi_país com país)
/* Es renombre la columna codi_país a país. */
R4 = R3 x regions
/* S'obté totes les combinacions de R3 amb regions. */
R5 = R4 (país==codi_país)
/* Es selecciona les combinacions que interessin. */
RR = R5 [codi_regió, nom_regió]
/* Es restringeix les columnes que interessin. */
```

- Possibilitat 2

```
R1 = (països (majúscules(nom_país) == "FRANÇA")) [codi_país]
R2 = R1 (tornar a anomenar codi_país com país)
RR = ((R2 x regions) (país==codi_país))[codi_regió, nom_regió]
```

Aquesta segona possibilitat és una calca de l'anterior minimitzant la generació de relacions intermitges en base a aprofitar el resultat que s'obté d'una operació de l'àlgebra relacional com a operando d'una altra operació.

- Possibilitat 3.

En les versions anteriors s'obté el producte cartesià de dues relacions per posteriorment seleccionar les tuples que verifiquen una condició d'igualtat entre atributs de les relacions sobre les que es realitza el producte cartesià. L'àlgebra relacional proporciona les operacions de *join* per a tal menester, de manera que es minimitzen molts passos.

```

R1 = (països (majúscules(nom_país) == "FRANÇA"))[codi_país]
R2 = R1[codi_país * codi_país]regions
/* En ser un natural-join la columna codi_país només apareixerà una vegada. */
RR = R2 [codi_regió, nom_regió]

```

- Possibilitat 4

```

R1 = (països (majúscules(nom_país) == "FRANÇA"))[codi_país]
R2 = R1 * regions
/* És un natural-join que s'efectua sobre totes les columnes amb mateix nom: codi_país */
RR = R2 [codi_regió, nom_regió]

```

- Possibilitat 5

```

R1 = països * regions
RR = (R1 (majúscules(nom_país) == "FRANÇA"))[codi_regió, nom_regió]

```

En aquest cas hem optat per efectuar en primer lloc el *natural-join* i posteriorment executar la selecció de les tuples corresponents a "França", mentre que en els casos anteriors en efectuar-se el *natural-join* la relació *països* ja hi participava prèviament filtrada.

Com podeu observar hi ha moltes possibilitats d'efectuar una consulta amb la utilització de l'àlgebra relacional.

c) Donat el client de codi 100000 obtenir les comandes pendents de servir. De cada comanda es vol codi, data de comanda, data prevista i nom de l'empleat responsable.

```

R1 = comandes (client==100000 i majúscules(situació)=='P')
R2 = R1[codi_comanda, data_comanda, data_prevista, empleat]
R3 = empleats [codi_empleat, nom, cognoms]
RR = R2[empleat*codi_empleat]R3

```

La solució presentada és vàlida quan cada comanda té obligatòriament un empleat, tal i com succeeix en el nostre cas. Ara bé, suposem que això no fos així i hi poguéssim haver comandes no vinculades a cap empleat, és a dir, l'atribut *empleat* de la relació *comandes* permet assolir el valor NUL. En tal situació, una comanda sense empleat vinculat no apareixeria a la relació resultant. Per aquests casos es necessita l'*outer join* i obtindríem:

```

R1 = comandes (client==100000 i majúscules(situació)=='P')
R2 = R1[codi_comanda, data_comanda, data_prevista, empleat]
R3 = empleats [codi_empleat, nom, cognoms]
RR = R2[•empleat*codi_empleat]R3

```

d) Per cada client es vol saber la quantitat de comandes servides durant l'any 2000.

```

R1 = comandes (majúscules(situació)=='P' i data_tramesa>=#01-01-2000# i
              data_tramesa<=#31-12-2000#)
R2 = R1[client]
RR = R2(per client resumir comptar() com quants)

```

La darrera operació agrupa els tuples de R2 (que només tenen la columna *client*) per l'atribut *client* i n'efectua el comptatge sobre un nou atribut *quants*, de manera que el resultat RR és una relació formada pels atributs *client* i *quants*.

e) Per cada client es vol saber l'import total de les comandes efectuades durant l'any 1999.

```

R1 = comandes (data_comanda>=#01-01-1999# i data_comanda<=#31-12-1999#)
R2 = R1[codi_comanda*comanda]detall
R3 = R2[client,quantitat,preu,dte_línia]
R4 = R3(ampliar quantitat*preu*(100-dte_línia)/100 com import)
R5 = R4[client,import]
RR = R5(per client resumir suma(import) com total)

```

En aquest exemple s'obté una relació amb els dos atributs *client* i *total*, solució a la pregunta presentada. Observem que en el procés es crea un atribut *import* que conté el càlcul de l'import de cada línia de comanda, el qual s'utilitza posteriorment per a obtenir el càlcul total a nivell de cada client.

f) Es desitja conèixer els productes pels que no s'ha efectuat cap comanda dins de l'any 2000.

```

R1 = comandes (data_comanda>=#01-01-2000# i data_comanda<=#31-12-2000#)
R2 = R1 [codi_comanda]
R3 = (detall [comanda*codi_comanda] R2) [producte]
R4 = productes [•codi_producte = producte] R3
RR = R4 (producte == NUL) [codi_producte, descripció]

```

R3 té els productes apareguts en alguna comanda efectuada dins l'any 2000.

R4 és una relació que conté tots els atributs de la relació productes i l'atribut *producte* corresponent a la relació R3. Els tuples són els de la relació *productes* i l'atribut *producte* conté valor NUL pels productes que no estaven a R3, és a dir, pels productes que no apareixen en cap comanda efectuada dins l'any 2000, que són precisament els productes que interessen.

g) Es desitja obtenir quines comandes de l'any 2000 tenen productes de totes les categories.

```
R1 = comandes (data_comanda>=#01-01-2000# i data_comanda<=#31-12-2000#)
R2 = R1 [codi_comanda]
R3 = (detall [comanda*codi_comanda] R2) [comanda, producte]
R4 = (R3 [producte*codi_producte] productes) [comanda, categoria]
R5 = categories [codi_categoria]
RR = R4 [categoria : codi_categoria] R5
```

R4 conté les comandes de l'any 2000 amb les diferents categories de productes que inclou cada comanda. És a dir, si una comanda està constituïda per 10 articles que pertanyen a tres categories, a la relació R4 la comanda apareixerà tres vegades, una per cadascuna de les tres categories.

En efectuar la *divisió* de R4 entre R5, s'obté les comandes de R4 vinculades a totes les categories de R5, que són totes les categories de la base de dades.

3.5. Conclusions

En la introducció de l'apartat 3 hem comentat que l'àlgebra relacional és un llenguatge procedural que permet accedir a les dades. Amb l'experiència dels exemples anem a matisar una mica aquesta afirmació.

En primer lloc, en els exemples hem pogut observar que l'àlgebra relacional és un llenguatge d'accés i també de definició dels esquemes externs (❗), ja que aquests no són altra cosa que un conjunt de vistes creades a partir de les relacions existents a la base de dades i l'àlgebra relacional és un llenguatge que defineix relacions a partir d'altres relacions.

En segon lloc, també ha quedat demostrat que és un llenguatge procedural, ja que per obtenir la relació resultat s'explicita, en qualsevol cas, el conjunt de passos a seguir. (❗)

Per finalitzar únicament manca comentar que no hi ha implementacions comercials de llenguatges basats en l'àlgebra relacional. Tan sols se n'ha efectuat alguns prototipus.

4. Normalització

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

Coneixem el model entitat-relació per a establir models per a qualsevol realitat, del qual se n'obté, com a resultat, el diagrama entitat-relació, altrament anomenat diagrama CHEN. També coneixem el procés de traducció d'un diagrama entitat-relació a un esquema relacional.

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

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

- D'entrada el model entitat-relació és posterior al model relacional i, per tant, hi ha bases de dades que varen ser formulades directament en la terminologia relacional. No hi havia altra opció !
- Hi ha dissenyadors que "no volen perdre el temps" en un model entitat-relació i dissenyen directament en el model relacional. Quin gran error !
- En ocasions s'ha de reformar la base de dades a causa de noves necessitats i el disseny s'efectua directament sobre ella enlloc d'analitzar-se i realitzar-se sobre el model entitat-relació per després transferir els canvis a l'esquema relacional. Quin gran error !

Observeu que estem donant total suport al fet d'utilitzar el model entitat-relació per a obtenir-ne posteriorment el model relacional. Un bon disseny en el model entitat-relació acostuma a proporcionar una base de dades relacional ben dissenyada, fet que no succeirà si el disseny entitat-relació incorpora errors. Per altra banda, si no hi ha hagut el disseny entitat-relació previ, hi ha més possibilitats de tenir una base de dades relacional mal dissenyada.



En el crèdit "Anàlisi i disseny de bases de dades" s'introdueix el model entitat-relació per a modelar la realitat.



En el nucli d'activitat "El model relacional" d'aquesta unitat didàctica es presenta el procés de traducció d'un diagrama entitat-relació a un esquema relacional.

La **teoria de la normalització** és un mètode que permet assegurar si un disseny relacional (tant si prové de la traducció d'un diagrama entitat-relació com si s'ha efectuat directament en el model relacional) és correcte en major o menor grau.

En general, els mals dissenys poden originar:

- Repetició de la informació.
- Impossibilitat de representar certa informació.
 - Anomalies en les insercions.
 - Anomalies en les modificacions.
 - Anomalies en els esborrats.

Un bon disseny ha d'aconseguir:

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

Exemple de disseny relacional inadequat

Considerem el disseny relacional de la taula 8 per a enregistrar la informació dels professors amb els alumnes de cadascun i la qualificació que han obtingut en els diversos crèdits.

Taula 8. Exemple de disseny relacional inadequat

#dniProf	#nomProfessor	#dniAlum	#nomAlumne	#edat	#crèdit	nota
33.333.333	Joan Finestra	77.777.777	Anna Taula	20	ADBD	4.5
33.333.333	Joan Finestra	88.888.888	Miquel Cadira	19	ADBD	5.7
33.333.333	Joan Finestra	77.777.777	Anna Taula	20	SGBD	6
33.333.333	Joan Finestra	88.888.888	Miquel Cadira	19	SGBD	7
44.444.444	Maria Porta	77.777.777	Anna Taula	20	MET	6
44.444.444	Maria Porta	88.888.888	Miquel Cadira	19	MET	5
44.444.444	Maria Porta	77.777.777	Anna Taula	20	LLC	4
44.444.444	Maria Porta	88.888.888	Miquel Cadira	19	LLC	3

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

- Hi ha informació repetida, fet que pot provocar possibles inconsistències. Fixem-nos que en cas d'haver de modificar qualsevol dels valors dels camps que formen la clau primària (dniProf, nomProfessor, dniAlum, nomAlumne, edat, crèdit), el canvi s'ha d'efectuar en totes les files on aquell valor apareix.
- No és possible valors nuls en les columnes que formen la clau primària.

Així, si no coneixem l'edat d'un alumne, tenim un greu problema...

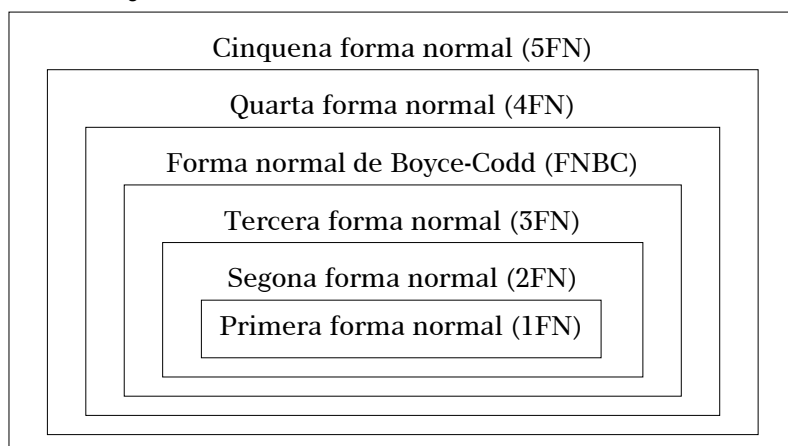
- En cas d'arribar a la conclusió que necessitem emmagatzemar més informació dels professors i/o dels alumnes, caldrà afegir més columnes i repetir la informació per a cada fila en la que aparegui el professor i/o alumne.
- Consultar la informació en la taula 8 pot esdevenir feixuc donat la gran quantitat d'informació diferent que conté.

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

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

Hi ha diferents graus de normalitat i, per tant, de formes normals, les quals compleixen la relació d'inclusió de la figura 3, que s'ha d'interpretar en el sentit que a mida que augmenta el nivell de la forma normal, la relació ha de complir un conjunt de condicions més restrictiu i, per tant, continua verificant les condicions de les formes normals de nivell inferior.

Figura 3. Relació d'inclusió entre les diverses formes normals.



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

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

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

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

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

4.1. La relació universal

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

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

Així, suposem que es vol dissenyar una base de dades pel control de les comandes de compra d'una determinada organització. Suposem que cal

incloure-hi els conceptes corresponents a: número i data de la comanda; codi, descripció, quantitat i preu pactat per cada article sol·licitat; data prevista de lliurament de la comanda; nom (*nomProv*) i país (*paísProv*) del proveïdor; moneda en la que es pacta la comanda. La relació universal és representada a la taula 9.

Taula 9. Relació universal per a un esquema relacional ideat per a una gestió de comandes de compra.

núm	dataComanda	Article	descripció	qtat	preu	dataPrevista	nomProv	paísProv	moneda
22.523	25-05-2000	PC3-500	PC Pentium III a 500	5	150	1-06-2000	ARKANSAS	XINA	EUR
22.523	25-05-2000	PRO-15	Protector Pantalla 15"	5	8	1-06-2000	ARKANSAS	XINA	EUR
22.524	27-05-2000	PC3-500	PC Pentium III a 500	15	145	5-06-2000	MELISSA	ITÀLIA	USD
22.524	27-05-2000	PRO-15	Protector Pantalla 15"	15	50	5-06-2000	MELISSA	ITÀLIA	USD
22.525	27-05-2000	INK430	Cartutx de tinta 430	20	25	31-5-2000	ARKANSAS	XINA	EUR

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

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

4.2. Dependències funcionals

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

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

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

En la relació universal de la taula 9 diríem que, entre d'altres, data de comanda depèn funcionalment del número de comanda, a l'igual que la data prevista, el nom i el país del proveïdor i la moneda. Podríem escriure-ho com:

$núm \rightarrow dataComanda$
 $núm \rightarrow dataPrevista$
 $núm \rightarrow nomProv$
 $núm \rightarrow paísProv$
 $núm \rightarrow moneda$

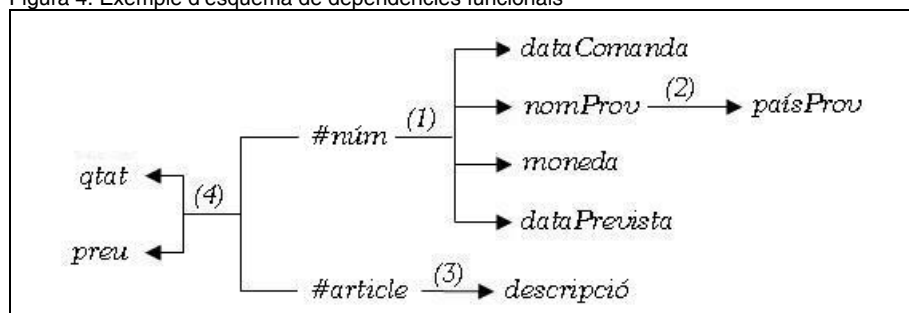
En tractar-se de diferents atributs que depenen funcionalment d'un mateix atribut, escriurem:

$núm \rightarrow dataComanda, dataPrevista, nomProv, paísProv, moneda$

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

És molt convenient representar les dependències funcionals d'una relació mitjançant un **esquema de dependències funcionals**. L'esquema per la relació universal de la taula 9 seria el que es mostra a la figura 4.

Figura 4. Exemple d'esquema de dependències funcionals



Observem que hem marcat els atributs que són clau d'alguna de les entitats que formen part de la relació: *article* identifica l'article i *núm* identifica la comanda. Observem, també, que la parella (*núm*, *article*) identifica la quantitat i preu dels articles demanats a la comanda.

En aquest esquema es poden observar les dependències funcionals entre els atributs. S'observa que *dataComanda*, *nomProv*, *moneda* i *dataPrevista* depenen funcionalment (1) de *núm* i que *paísProv* depèn funcionalment (2) de *nomProv* i que *descripció* depèn funcionalment (3) d'*article*. Així mateix, *qtat* i *preu* depenen funcionalment (4) de *núm* i *article*.

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

(l'**implicat**) pugui dependre d'un subconjunt de l'implicador. La dependència (4) també és total ja que *qtat* i *preu* depenen de la parella (*núm*, *article*) i no pas de cap subconjunt d'aquesta.

Una darrera apreciació sobre la dependència funcional (4) de la figura 5. Observem que en una mateixa comanda (*núm*) no és possible tenir varies vegades el mateix article (*article*), ja que els atributs *qtat* i *preu* depenen funcionalment de (*núm*, *article*). En cas que fos necessari tenir varies vegades el mateix article en una comanda caldria utilitzar algun altre atribut per identificar l'article dins la comanda com, per exemple, el *númeroDeLínia* de la comanda.

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

Observem, en el cas anterior, que *núm*, *article*, *nomProv* i la parella (*núm*, *article*) són determinants de la relació.

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

En l'exemple de la figura 4, podem dir que *paísProv* depèn transitivament de *núm* a través de *nomProv*.

4.3. Primera forma normal

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

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

Considerem la relació universal de la taula 10.

Taula 10. Relació que té atributs multivalor i, per tant, no es troba en 1FN.

núm	dataComanda	article	Descripció	qtat	preu	dataPrevista	nomProv	paísProv	moneda
22.523	25-05-2000	PC3-500	PC Pentium III a 500	5	150	1-06-2000	ARKANSAS	XINA	EUR
		PRO-15	Protector Pantalla 15"	5	8				
22.524	27-05-2000	PC3-500	PC Pentium III a 500	15	145	5-06-2000	MELISSA	ITÀLIA	USD
		PRO-15	Protector Pantalla 15"	15	50				
22.525	27-05-2000	INK430	Cartutx de tinta 430	20	25	31-5-2000	ARKANSAS	XINA	EUR

La relació de la taula 10 no està en 1FN ja que té atributs que poden contenir més d'un valor. Observem que aquest exemple contempla tres files (comandes 22.523, 22.524 i 22.525) i alguns atributs (*article*, *descripció*, *qtat* i *preu*) per algunes de les files tenen varis valors.

El procés a seguir per assolir una 1FN és afegir tantes files com sigui necessari per cadascun dels diferents valors del(s) camp(s) que tingui(n) valors no atòmics.

Així, en el nostre cas, obtenim la relació en 1FN de la taula 11.

Taula 11. Relació en 1FN.

núm	dataComanda	article	Descripció	qtat	preu	dataPrevista	nomProv	paísProv	moneda
22.523	25-05-2000	PC3-500	PC Pentium III a 500	5	150	1-06-2000	ARKANSAS	XINA	EUR
22.523	25-05-2000	PRO-15	Protector Pantalla 15"	5	8	1-06-2000	ARKANSAS	XINA	EUR
22.524	27-05-2000	PC3-500	PC Pentium III a 500	15	145	5-06-2000	MELISSA	ITÀLIA	USD
22.524	27-05-2000	PRO-15	Protector Pantalla 15"	15	50	5-06-2000	MELISSA	ITÀLIA	USD
22.525	27-05-2000	INK430	Cartutx de tinta 430	20	25	31-5-2000	ARKANSAS	XINA	EUR

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

Les relacions en 1FN poden tenir, en general, molta informació redundant. No ens ha de preocupar, ja que la solució radica en les formes normals de nivell superior.

4.4. Preservació d'informació i dependències en la normalització

En el procés de normalització d'una relació R s'apliquen processos de descomposició per aconseguir relacions $R_1, R_2, \dots, R_n$ que verifiquin un nivell de normalització superior al de la relació R . La descomposició

consisteix en efectuar projeccions de la relació R sobre atributs que verifiquen certes condicions, donant lloc a l'aparició de $R_1, R_2, \dots, R_n$.

Cal garantir que la descomposició de R en $R_1, R_2, \dots, R_n$ preservi la informació existent, és a dir, que el natural join $R_1 * R_2 * \dots * R_n$ proporcioni exactament les mateixa informació que tenia la relació R original, tan en intensió (quantitat d'atributs) com en extensió (quantitat de files). **!!**

D'igual forma caldria garantir la conservació de les dependències, és a dir, que el conjunt de dependències associades a la relació R original ha de ser equivalent al conjunt de dependències associat a les relacions $R_1, R_2, \dots, R_n$. **!!**

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

4.5. Segona forma normal

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

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

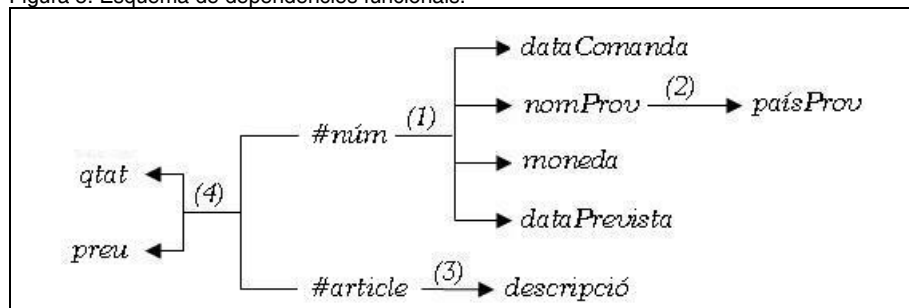
Considerem la següent relació en 1FN ideada per a la gestió de comandes de compra en una organització:

R (#núm, dataComanda, #article, descripció, qtat, preu, dataPrevista, nomProv, paísProv, moneda)

Observem que la clau primària està formada per la parella (*núm*, *article*) i, en l'esquema de dependències funcionals associat (figura 5) s'observa que hi ha atributs fora de la clau primària que no tenen dependència funcional completa de la clau.

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

Figura 5. Esquema de dependències funcionals.



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

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

comanda (#núm, dataComanda, dataPrevista, moneda, nomProv, paísProv)

article (#article, descripció)

detall (#núm, #article, qtat, preu)
comanda article

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

La informació de la taula 11 corresponent a la relació que acabem de normalitzar, passa ara a estar repartides en tres taules (12, 13 i 14).

Taula 14. Relació en 2FN que emmagatzema les comandes.

comanda					
Núm	dataComanda	dataPrevista	nomProv	paísProv	moneda
22.523	25-05-2000	1-06-2000	ARKANSAS	XINA	EUR
22.524	27-05-2000	5-06-2000	MELISSA	ITÀLIA	USD
22.525	27-05-2000	31-5-2000	ARKANSAS	XINA	EUR

Taula 14. Relació en 2FN pels articles

article	
article	descripció
PC3-500	PC Pentium III a 500
PRO-15	Protector Pantalla 15"
INK430	Cartutx de tinta 430

Taula 14. Relació en 2FN pel detall de comanda

detall			
Núm	article	qtat	preu
22.523	PC3-500	5	150
22.523	PRO-15	5	8
22.524	PC3-500	15	145
22.524	PRO-15	15	50
22.525	INK430	20	25

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

4.6. Tercera forma normal

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

comanda	(#núm, dataComanda, dataPrevista, moneda, nomProv, paísProv)
article	(#article, descripció)
detall	(#núm, #article, qtat, preu)
	comanda article

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

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

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

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

comanda	(#núm, dataComanda, dataPrevista, moneda, nomProv, paísProv)
----------------	--

L'atribut *paísProv* depèn transitivament de *núm* a través de *nomProv*. Per tant, aquesta relació no està en 3FN.

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

En el nostre cas obtenim les relacions en 3FN:

proveïdor (#codProv, nomProv, paísProv)
comanda (#núm, dataComanda, dataPrevista, moneda, codProv)
proveïdors

Observem que en efectuar el trencament de relació inicial *comanda* ha semblat oportú considerar un nou atribut (*codProv*) que identifiqui millor la nova relació *proveïdor*. Aquest fet no és imprescindible i no sempre serà convenient. Podríem haver considerat el següent trencament:

proveïdor (#nomProv, paísProv)
comanda (#núm, dataComanda, dataPrevista, moneda, nomProv)
proveïdors

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

Tenint en compte el disseny que incorpora l'atribut *codProv* tindriem la conversió de la taula 12 en les taules 15 i 16.

Taula 16. Relació en 3FN que emmagatzema les comandes

comanda				
núm	dataComanda	dataPrevista	codProv	Moneda
22.523	25-05-2000	1-06-2000	ARK	EUR
22.524	27-05-2000	5-06-2000	MEL	USD
22.525	27-05-2000	31-5-2000	ARK	EUR

Taula 16 . Relació en 3FN pels proveïdors

Proveïdor		
codProv	nomProv	paísProv
ARK	ARKANSAS	XINA
MEL	MELISSA	ITÀLIA

4.7. Forma normal de Boyce-Codd

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

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

dipòsit (#venedor, #tipusMaterial, nau, ubicació)

La taula 17 ens exemplifica la situació. Observem que es troba en 3FN:

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

En efecte, la *nau* i la *ubicació* dins la *nau* depenen del *venedor* i del *tipusMaterial*, ja que hi pot haver varies naus dedicades a un tipus de material però tot el material similar d'un venedor es troba en una determinada nau. A la vegada, hi pot haver varies naus amb material d'un venedor degut a la diferent tipologia del material.

Taula 17. Relació en 3FN

dipòsit			
venedor	tipusMaterial	nau	ubicació
JOSEP	Peix fresc	15	S4-C3-U1:10
MARIA	Peix fresc	25	S3-C5-U5:22
RAMON	Congelats	17	S2-C4-U1:25
ANNA	Vegetals	20	S1-C6-U7:10
ANNA	Peix fresc	25	S2-C5-U12:15
MARIA	Basar	10	S3-C4-U20:25

- Cap atribut no pertanyent a la clau (*nau* i *ubicació*) depèn transitivament de la clau.

En efecte, és impossible que hi hagi cap dependència transitiva de la clau ja que no hi ha cap atribut que pugui servir de pont per la transitivitat.

Però aquesta relació, tot i estar en 3FN, presenta anomalies:

- Si en un moment donat un nau no té material de cap venedor, es perd la informació referent al tipus de mercaderia que correspon a la nau.
- Si canvia la descripció del tipus de mercaderia assignada a una nau cal modificar tantes files com venedors amb dipòsits d'aquell tipus de mercaderia hi hagi a la nau.

Una relació està en la **forma normal de Boyce-Codd (FNBC)** si està en 2FN i tot determinant d'ella és clau candidata.

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

És a dir, s'està donant la situació:

nau $\longrightarrow$ *tipusMaterial*

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

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

En el nostre cas apartarem de la relació l'atributa *tipusMaterial* i obtenim les relacions en FNBC:

dipòsit (#venedor, nau, ubicació)
 nauTipus

nauTipus (#nau, tipusMaterial)

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

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

$$\begin{array}{lcl} \text{venedor, tipusMaterial} & \longrightarrow & \text{nau, ubicació} \\ \text{nau} & \longrightarrow & \text{tipusMaterial} \end{array}$$

i en les relacions finals *dipòsits* i *nauTipus*, s'ha perdut la dependència funcional que indicava que la *nau* depenia de la parella *venedor* i *tipus_material*:

$$\begin{array}{lcl} \textbf{dipòsit:} & \text{venedor} & \longrightarrow \text{nau, ubicació} \\ \textbf{nauTipus:} & \text{nau} & \longrightarrow \text{tipusMaterial} \end{array}$$

La dependència funcional s'ha perdut degut a que el concepte de dependència funcional en el model relacional es defineix únicament entre atributs d'una mateixa relació i en aquest cas caldria poder-la definir entre atributs de relacions diferents. De tota manera, el concepte d'integritat referencial intenta superar aquesta limitació.

Degut a la pèrdua de dependències funcionals, que no sempre té per què passar, en moltes ocasions no es normalitza a FNBC i es treballa amb relacions en 3FN.

4.8. Quarta forma normal

Considerem la relació *estudiant* (taula 18) dissenyada per emmagatzemar els diversos crèdits que està cursant així com les diverses activitats esportives que desenvolupa.

estudiant(#dni, #crèdit, #esport)

És a dir, la relació *estudiant* recull la possibilitat de que un estudiant estigui cursant diversos crèdits i practicant diverses activitats esportives.

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

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

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

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

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

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

En el nostre cas (taula 18) es verifica:

$dni \twoheadrightarrow crèdit$
 $dni \twoheadrightarrow esport$

Taula 18 . Relació FNBC amb redundància

estudiant		
dni	crèdit	esport
10.000.000	SGBD	Bàsquet
10.000.000	ADBD	Bàsquet
10.000.000	SGBD	Futbol
10.000.000	ADBD	Futbol
20.000.000	PEM	Natació
20.000.000	ADBD	Natació
20.000.000	SGBD	Natació
20.000.000	PEM	Esgrima
20.000.000	ADBD	Esgrima
20.000.000	SGBD	Esgrima
15.000.000	PEM	Natació
15.000.000	PEM	Bàsquet

Per assolir la 4FN a partir d'una relació R ($\#A, B, C$) que té una dependència multivalent $A \twoheadrightarrow B$, cal descompondre la relació R en dues relacions R_1 ($\#A, B$) i R_2 ($\#A, C$).

En el cas obtenim les relacions següents (taules 19 i 20):

crèditEnCurs ($\#dni, \#crèdit$)
esportEnPràctica ($\#dni, \#esport$)

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

4.9. Cinquena forma normal

Considerem la relació *professor* (taula 21) dissenyada per gestionar els professors d'una determinada institució escolar que té diferents centres de docència. Cada professor està autoritzat a impartir unes determinades especialitats docents les quals pot posar en pràctica en qualsevol dels centres docents de la institució escolar. Així mateix, cada professor pot exercir, a més de la docència, diferents tasques (càrrecs, tutoria pedagògica, tutoria tècnica,...) en diversos centres de la institució escolar.

professors ($\#codiProf, \#centre, \#especialitat, \#tasca$)

Taula 21. Relació en FNBC amb dependències multivalents

professor			
codiProf	centre	especialitat	tasca
P1	C1	Matemàtiques	Tutor
P1	C2	Matemàtiques	Tutor
P1	C2	Informàtica	Aula Informàtica
P2	C1	Català	Coordinador
P2	C2	Castellà	Tutor

Aquesta relació és FNBC i s'hi aprecia una espècie de dependència multivalent, la qual no es pot solucionar per via de la descomposició. En efecte, és molt fàcil pensar en una descomposició en les tres relacions següents (exemplificades en les taules 22, 23 i 24):

Taula 20. Relació en 4FN

crèditEnCurs	
dni	crèdit
10.000.000	SGBD
10.000.000	ADBD
20.000.000	PEM
20.000.000	ADBD
20.000.000	SGBD
15.000.000	PEM

Taula 20. Relació en 4FN

esportEnPràctica	
dni	esport
10.000.000	Bàsquet
10.000.000	Futbol
20.000.000	Natació
20.000.000	Esgrima
15.000.000	Natació
15.000.000	Bàsquet

centreDeProfessor (#codiProf, #centre)
especialitatDeProfessor (#codiProf, #especialitat)
tascaDeProfessor (#codiProf, #tasca)

Taula 24. Relació 4FN

centreDeProfessor	
codiProf	centre
P1	C1
P1	C2
P2	C1
P2	C2

Emmagatzema els centres de cada professor

Taula 24 . Relació 4FN

especialitatDeProfessor	
codiProf	especialitat
P1	Matemàtiques
P1	Informàtica
P2	Català
P2	Castellà

Emmagatzema les especialitats de cada professor

Taula 24 . Relació 4FN

tascaDeProfessor	
codiProf	tasca
P1	Tutor
P1	Aula Informàtica
P2	Coordinador
P2	Tutor

Emmagatzema les tasques de cada professor

Aquesta descomposició és errònia, doncs si apliquem el natural-join de les tres relacions (taules 22, 23 i 24) no obtenim la relació inicial (taula 21) sinó que obtenim una relació (taula 25) amb moltes més instàncies.

Taula 25. Relació obtinguda del natural-join de les relacions de les taules 22, 23 i 24

Professor			
codiProf	centre	especialitat	tasca
P1	C1	Matemàtiques	Tutor
P1	C1	Matemàtiques	Aula Informàtica
P1	C1	Informàtica	Tutor
P1	C1	Informàtica	Aula Informàtica
P1	C2	Matemàtiques	Tutor
P1	C2	Matemàtiques	Aula Informàtica
P1	C2	Informàtica	Tutor
P1	C2	Informàtica	Aula Informàtica
P2	C1	Català	Coordinador
P2	C1	Català	Tutor
P2	C1	Castellà	Coordinador
P2	C1	Castellà	Tutor
P2	C2	Català	Coordinador
P2	C2	Català	Tutor
P2	C2	Castellà	Coordinador
P2	C2	Castellà	Tutor

La marca < a la part dreta de la taula indica les files que apareixen en el natural-join i que no existien en la taula original.

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

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

Tornem a l'exemple de descomposició anterior: la relació *professor* s'ha descompost en tres relacions *centreDeProfessor*, *especialitatDeProfessor* i *tascaDeProfessor* i hem pogut comprovar que la relació *professor* no satisfà una dependència de reunió respecte *centreDeProfessor*, *especialitatDeProfessor* i *tascaDeProfessor*.

Ens cal trobar una dependència de reunió per la relació *professor*, és a dir, trobar una descomposició tal que el seu natural-join recuperi la relació original. Observem la següent descomposició (taules 39, 40 i 41)

```
professor (#codiProf, #centre, #especialitat, #tasca)
pce = professor [#codiProf, #centre, #especialitat]
pct = professor [#codiProf, #centre, #tasca]
pet = professor [#codiProf, #especialitat, #tasca]
```

Taula 27. Relació 4FN

pce		
codiProf	centre	especialitat
P1	C1	Matemàtiques
P1	C2	Matemàtiques
P1	C2	Informàtica
P2	C1	Català
P2	C2	Castellà

Taula 26 Relació 4FN

pct		
codiProf	centre	tasca
P1	C1	Tutor
P1	C2	Tutor
P1	C2	Aula Informàtica
P2	C1	Coordinador
P2	C2	Tutor

Taula 28. Relació 4FN

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

En aquesta situació, observem que si efectuem el natural-join de les tres relacions *pce*, *pct* i *pet* obtenim la relació original.

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

La relació *professor* del nostre exemple no es troba, per tant, en 5FN, ja que hem trobat la dependència de reunió $D_{professor}^*(pce, pct, pet)$ en la que les relacions *pce*, *pct* i *pet* no estan constituïdes per claus candidates de *professor*.

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

Així, en el nostre exemple, la relació *professor* desapareixeria per donar pas a les tres relacions *pce*, *pct* i *pet* en les que es basa la dependència de reunió trobada.

pce (#codiProf, #centre, #especialitat)
pct (#codiProf, #centre, #tasca)
pet (#codiProf, #especialitat, #tasca)
