

# El model relacional i l'àlgebra relacional

Dolors Costal Costa

P02/05002/00095



# Índex

Introducció.....	5
Objectius.....	6
1. Introducció al model relacional.....	7
2. Estructura de les dades.....	9
2.1. Visió informal d'una relació .....	9
2.2. Visió formal d'una relació .....	10
2.3. Diferències entre relacions i fitxers .....	12
2.4. Clau candidata, clau primària i clau alternativa de les relacions.....	13
2.5. Claus foranes de les relacions.....	15
2.6. Creació de les relacions d'una base de dades.....	18
3. Operacions del model relacional .....	19
4. Regles d'integritat .....	21
4.1. Regla d'integritat d'unicitat de la clau primària .....	22
4.2. Regla d'integritat d'entitat de la clau primària .....	23
4.3. Regla d'integritat referencial.....	24
4.3.1. Restricció.....	27
4.3.2. Actualització en cascada.....	27
4.3.3. Anul·lació.....	29
4.3.4. Selecció de la política de manteniment de la integritat referencial .....	31
4.4. Regla d'integritat de domini.....	31
5. L'àlgebra relacional.....	34
5.1. Operacions conjuntistes .....	37
5.1.1. Unió .....	37
5.1.2. Intersecció.....	39
5.1.3. Diferència.....	40
5.1.4. Producte cartesià.....	41
5.2. Operacions específicament relacionals.....	42
5.2.1. Selecció .....	42
5.2.2. Projecció .....	43
5.2.3. Combinació .....	44
5.3. Seqüències d'operacions de l'àlgebra relacional.....	47
5.4. Extensions: combinacions externes.....	49
Resum.....	52

Exercicis d'autoavaluació.....	53
Solucionari .....	55
Glossari .....	56
Bibliografia .....	58

## Introducció

Aquest mòdul didàctic està dedicat a l'estudi del model de dades relacional i de l'àlgebra relacional.

El concepte de *model de dades* s'ha presentat en un altre mòdul. En aquest, s'aprofundeix en un model de dades concret: el **model relacional**, que actualment té una gran rellevància. Els seus conceptes fonamentals estan ben assentats i, a més, els sistemes de gestió de bases de dades relacionals són els més estesos en la seva utilització pràctica. Per aquests motius pensem que és important conèixer-lo.

! Vegeu el concepte de *model de dades* en el mòdul "Introducció a les bases de dades" d'aquesta assignatura.

L'estudi del model relacional serveix, a més, de base per als continguts d'un altre mòdul, dedicat al llenguatge SQL. Aquest llenguatge permet definir i manipular bases de dades relacionals. Els fonaments del model relacional resulten, doncs, imprescindibles per a aconseguir un bon domini de l'SQL.

! Les construccions de l'SQL s'estudien en el mòdul didàctic "El llenguatge SQL".

En aquest mòdul s'analitzen també les **operacions de l'àlgebra relacional**, que serveixen per a fer consultes a una base de dades. Cal conèixer aquestes operacions perquè ens permeten saber quins serveis de consulta ha de proporcionar un llenguatge relacional. Una altra aportació de l'àlgebra relacional és que facilita la comprensió d'algunes de les construccions del llenguatge SQL que s'estudiaran en un altre mòdul d'aquesta assignatura. A més, constitueix la base per a l'estudi del tractament de les consultes que efectuen els SGBD internament (especialment pel que fa a l'optimització de consultes). Aquest darrer tema queda fora de l'àmbit de la present assignatura, però és rellevant per a estudis més avançats sobre bases de dades.

## Objectius

En els materials didàctics d'aquest mòdul, l'estudiant trobarà les eines indispensables per a assolir els objectius següents:

1. Conèixer els fonaments del model de dades relacional.
2. Saber distingir les característiques que ha de tenir un sistema de gestió de bases de dades relacional, per tal que sigui coherent amb els fonaments del model relacional.
3. Copsar els avantatges del model relacional que deriven de l'alt grau d'independència de les dades que proporciona, i de la simplicitat i la uniformitat del model.
4. Conèixer les operacions de l'àlgebra relacional.
5. Saber utilitzar les operacions de l'àlgebra relacional per a consultar una base de dades.

## 1. Introducció al model relacional

El **model relacional** és un model de dades i, com a tal, té en compte els tres aspectes següents de les dades:

- 1) L'**estructura**, que ha de permetre representar la informació que ens interessa del món real.
- 2) La **manipulació**, a la qual dóna suport mitjançant les operacions d'actualització i consulta de les dades.
- 3) La **integritat**, que és facilitada mitjançant l'establiment de regles d'integritat, és a dir, condicions que les dades han de complir.

El concepte de *model de dades* s'ha explicat al mòdul "Introducció a les bases de dades" d'aquesta assignatura.

Un **sistema de gestió de bases de dades relacional** (SGBDR) és un sistema de gestió de bases de dades que dóna suport a la definició de dades mitjançant l'estructura de les dades del model relacional, i també a la manipulació d'aquestes dades amb les operacions del model, i, a més, assegura que se satisfan les regles d'integritat que el model relacional estableix.


El concepte de *SGBD* ha estat presentat al mòdul "Introducció a les bases de dades" d'aquesta assignatura.

Els principis del model de dades relacional van ser establerts per E.F. Codd els anys 1969-1970. De totes maneres, fins a la dècada dels vuitanta no es van començar a comercialitzar els primers SGBD relacionals amb rendiments acceptables. Val a dir, però, que els SGBD relacionals que es comercialitzen actualment encara no suporten tot el que estableix la teoria relacional fins al darrer detall.

El **principal objectiu del model de dades relacional** és facilitar que la base de dades sigui percebuda o vista per l'usuari com una estructura lògica que consisteix en un conjunt de relacions, i no com una estructura física d'implementació. Això ajuda a aconseguir un alt grau d'independència de les dades.

Un objectiu addicional del model és aconseguir que aquesta estructura lògica amb què es percep la base de dades sigui simple i uniforme. Per tal de proporcionar simplicitat i uniformitat, tota la informació es representa d'una única manera: mitjançant valors explícits que contenen les relacions (no es fan servir conceptes com ara apuntadors entre relacions). Amb el mateix propòsit, tots els valors de dades es consideren atòmics, és a dir, no és possible descompondre'ls.

Cal precisar que un SGBD relacional, a nivell físic, pot emprar qualsevol estructura de dades per a implementar l'estructura lògica formada per les relacions. En particular, a nivell físic, el sistema pot fer servir apuntadors, índexs, etc. Aquesta implementació física queda, però, amagada a l'usuari.

Als apartats següents estudiarem l'estructura de les dades, les operacions i les regles d'integritat del model relacional. Hi ha dues maneres possibles d'enfocar l'estudi dels continguts d'aquest mòdul. La primera és seguir-los en l'ordre d'exposició. Així es van tractant tots els elements de la teoria del model relacional de manera molt precisa i en un ordre lògic. Una altra possibilitat, però, és començar amb la lectura del resum final del mòdul i llegir després tota la resta de continguts en l'ordre normal. El resum descriu els aspectes més rellevants de la teoria relacional que s'expliquen i, així, proporciona una visió global dels continguts del mòdul que, per a alguns estudiants, pot ser útil de copsar abans d'iniciar un estudi detallat. 



## 2. Estructura de les dades

El model relacional proporciona una estructura de les dades per a representar la informació que ens interessa del món real que consisteix en un conjunt de relacions.

L'estructura de les dades del model relacional es basa, doncs, en el concepte de *relació*.

### 2.1. Visió informal d'una relació

En primer lloc, presentarem el concepte de *relació* de manera informal. Es pot obtenir una bona idea intuïtiva del que és una relació si la visualitzem com una taula o un fitxer. A la figura 1 es mostra la visualització tabular d'una relació que conté dades d'empleats. Cada fila de la taula conté una col·lecció de valors de dades relacionats entre si; en el nostre exemple, són les dades corresponents a un mateix empleat. La taula té un nom (*EMPLEATS*), i també té un nom cadascuna de les seves columnes (*DNI*, *nom*, *cognom* i *sou*). El nom de la taula i els noms de les columnes ajuden a entendre el significat dels valors que conté la taula. Cada columna conté valors d'un cert domini; per exemple, la columna *DNI* conté valors del domini *númerosDNI*.

A l'assignatura *Estructura de la informació* es va presentar un concepte semblant de *relació*, però només sobre dos dominis.

Figura 1

Relació **EMPLEATS**

<i>númerosDNI</i>	<i>noms</i>	<i>cognoms</i>	<i>sous</i>
<i>DNI</i>	<i>nom</i>	<i>cognom</i>	<i>sou</i>
40.444.255	Joan	Garcia	2.000
33.567.711	Marta	Roca	2.500
55.898.425	Carles	Bonmatí	1.500

#### Conjunt de relacions

Una base de dades relacional consta d'un conjunt de relacions, cadascuna de les quals es pot visualitzar d'aquesta manera tan senzilla. L'estructura de les dades del model relacional resulta, doncs, fàcil d'entendre per a l'usuari.

Tot i això, si definim les relacions de manera més precisa, ens adonem que presenten algunes característiques importants que, en la visió superficial que hem presentat, queden amagades. Aquestes característiques són les que motiven que el concepte de *relació* sigui totalment diferent del de *fitxer*, tot i que, a primera vista, relacions i fitxers puguin semblar similars.

## 2.2. Visió formal d'una relació

A continuació definim formalment les relacions i altres conceptes que hi estan vinculats, com ara *domini*, *esquema de relació*, etc.

Un **domini**  $D$  és un conjunt de valors atòmics. Pel que fa al model relacional, *atòmic* significa indivisible. És a dir, que, per molt complex o llarg que sigui un valor atòmic, per a un SGBD relacional no té una estructura interna.

Els dominis poden ser de dos tipus: !

**1) Dominis predefinitos**, que corresponen als tipus de dades que normalment són proporcionats pels llenguatges de bases de dades, com ara els enters, les cadenes de caràcters, els reals, etc.

**2) Dominis definits per l'usuari**, que poden ser més específics. Tota definició d'un domini ha de constar, com a mínim, del nom del domini i de la descripció dels valors que en formen part.

Recordeu que s'estableix una classificació similar per als tipus de dades dels llenguatges de programació, com vam veure a *Fonaments de programació I*.

Domini definit per l'usuari

Per exemple, l'usuari pot definir un domini per a les edats dels empleats que s'anomeni *dom\_edat* i que contingui els valors enters que són entre 16 i 65.

Una **relació** es compon de l'**esquema** (o intensió de la relació) i de l'**extensió**.

Si considerem la representació tabular anterior (figura 1), l'esquema correspondria a la capçalera de la taula i l'extensió correspondria al cos:


Figura 2

EMPLEATS				Esquema
<i>DNI</i>	<i>nom</i>	<i>cognom</i>	<i>sou</i>	
40.444.255	Joan	Garcia	2.000	Extensió
33.567.711	Marta	Roca	2.500	
55.898.425	Carles	Bonmatí	1.500	

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

Nom i conjunt d'atributs de la relació *EMPLEATS*

Si prenem com a exemple la figura 1, el nom de la relació és *EMPLEATS* i el conjunt d'atributs és  $\{DNI, nom, cognom, sou\}$ .

Prendrem la convenció de denotar l'esquema de la relació de la manera següent:  $R(A_1, A_2, \dots, A_n)$ , on  $R$  és el nom de la relació i  $A_1, A_2, \dots, A_n$  és una ordenació qualsevol dels atributs que pertanyen al conjunt  $\{A_1, A_2, \dots, A_n\}$ . 

Denotació de l'esquema de la relació *EMPLEATS*

L'esquema de la relació de la figura 1 es podria denotar, per exemple, com *EMPLEATS(DNI, nom, cognom, sou)* o, també, *EMPLEATS(nom, cognom, DNI, sou)*, perquè qualsevol ordenació dels seus atributs es considera vàlida per a denotar l'esquema d'una relació.

Un **atribut**  $A_j$  és el nom del paper que exerceix un domini  $D$  en un esquema de relació.  $D$  és el **domini de**  $A_j$  i es denota com  $\text{domini}(A_j)$ .

Domini de l'atribut DNI

Segons la figura 1, l'atribut *DNI* correspon al paper o rol que exerceix el domini *númerosDNI* a l'esquema de la relació *EMPLEATS* i, aleshores,  $\text{domini}(DNI) \ni \text{númerosDNI}$ .


Convé observar que cada atribut és únic en un esquema de relació, perquè no té sentit que un mateix domini exerceixi dues vegades el mateix paper en un mateix esquema. Per consegüent, no pot passar que en un esquema de relació hi hagi dos atributs amb el mateix nom. En canvi, sí que es pot repetir un nom d'atribut en relacions diferents. Els dominis dels atributs, contràriament, no han de ser necessàriament tots diferents en una relació.


Exemple d'atributs diferents amb el mateix domini

Si prenem com a exemple l'esquema de relació *PERSONES(DNI, nom, cognom, telcasa, telfeina)*, els atributs *telcasa* i *telfeina* poden tenir el mateix domini:  $\text{domini}(telcasa) \ni \text{telèfon}$  i  $\text{domini}(telfeina) \ni \text{telèfon}$ .

En aquest cas, el domini *telèfon* exerceix dos papers diferents en l'esquema de relació: el d'indicar el telèfon particular d'una persona i el d'indicar el de la feina.

L'**extensió de la relació d'esquema**  $R(A_1, A_2, \dots, A_n)$  és un conjunt de tuples  $t_i$  ( $i \in 1, 2, \dots, m$ ), on cada tupla  $t_i$  és, a la vegada, un conjunt de parells  $t_i \ni \{ \langle A_1: v_{i1} \rangle, \langle A_2: v_{i2} \rangle, \dots, \langle A_n: v_{in} \rangle \}$  i, per a cada parell  $\langle A_j: v_{ij} \rangle$ , es compleix que  $v_{ij}$  és un valor de  $\text{domini}(A_j)$  o bé un valor especial que anomenem *nul*.

 Noteu la similitud entre aquest concepte de *tupla* i el que ja coneixeu dels llenguatges de programació, que heu estudiat en les assignatures de *Fonaments de programació*.

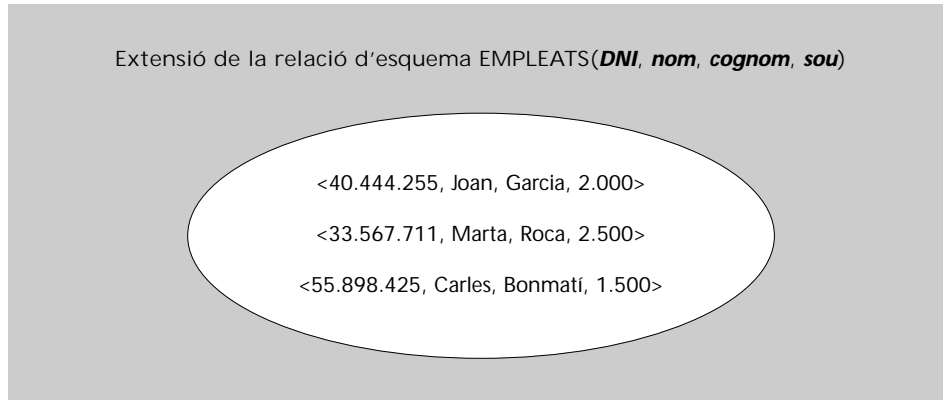
Per a simplificar, prendrem la convenció de referir-nos a una tupla  $t_i \ni \{ \langle A_1: v_{i1} \rangle, \langle A_2: v_{i2} \rangle, \dots, \langle A_n: v_{in} \rangle \}$  que pertany a l'extensió d'una relació d'esquema denotat com  $R(A_1, A_2, \dots, A_n)$ , de la manera següent:  $t_i \ni \langle v_{i1}, v_{i2}, \dots, v_{in} \rangle$ . 

Si denotem l'esquema de la relació representada a la figura 1 com *EMPLEATS(DNI, nom, cognom, sou)*, el conjunt de tuples de la seva extensió serà el de la figura de la pàgina següent:

Alguns autors...

... anomenen *taules*, *columnes* i *files* les relacions, els atributs i les tuples, respectivament.

Figura 3



Aquesta figura...

... ens mostra l'extensió d'*EMPLEATS* en forma de conjunt, mentre que les figures anteriors ens la mostraven en forma de files d'una taula. La representació tabular és més còmoda, però no reflecteix la definició d'extensió de manera tan exacta.

Si en una tupla  $t_i \in \langle v_{i1}, v_{i2}, \dots, v_{in} \rangle$ , el valor  $v_{ij}$  és un **valor nul**, això significa que el valor de l'atribut  $A_j$  és desconegut per a la tupla  $t_i$  de la relació, o bé que no és aplicable a aquesta tupla.

#### Exemple de valor nul

Podríem tenir un atribut *telcasa* a la relació *EMPLEATS* i es podria donar el cas que un empleat no tingués telèfon a casa seva, o bé que en tingués, però no se'n conegués el número. En ambdues situacions, el valor de l'atribut *telcasa* per a la tupla corresponent a l'empleat seria el valor nul.

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

#### Grau de la relació *EMPLEATS*

A la relació d'esquema *EMPLEATS*(*DNI, nom, cognom, sou*), el grau és 4.

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

#### Cardinalitat de la relació *EMPLEATS*

Observant la figura 3 es dedueix que la cardinalitat de la relació *EMPLEATS* és 3.

### 2.3. Diferències entre relacions i fitxers

A primera vista, relacions i fitxers resulten similars. Els registres i els camps que formen els fitxers s'assemblen a les tuples i als atributs de les relacions, respectivament.

Tot i aquesta similitud superficial, la visió formal de relació que hem presentat estableix algunes característiques de les relacions que les fan diferents dels fitxers clàssics. A continuació descrivim aquestes característiques: !

Recordeu que hem estudiat els fitxers i les seves característiques en el mòdul didàctic "Els fitxers" d'aquesta assignatura. !

**1) Atomicitat dels valors dels atributs:** els valors dels atributs d'una relació han de ser atòmics, és a dir, no han de tenir estructura interna. Aquesta característica prové del fet que els atributs sempre han de prendre un valor del seu domini o bé un valor nul, i del fet que s'ha establert que els valors dels dominis han de ser atòmics en el model relacional.

L'objectiu de l'atomicitat dels valors és donar simplicitat i uniformitat al model relacional.

**2) No-repetició de les tuples:** en un fitxer clàssic pot passar que dos registres del fitxer siguin exactament iguals, és a dir, que continguin les mateixes dades. En el cas del model relacional, en canvi, no és possible que una relació contingui tuples repetides. Aquesta característica es dedueix de la mateixa definició de l'extensió d'una relació. L'extensió és un conjunt de tuples i, en un conjunt, no hi pot haver elements repetits.

**3) No-ordenació de les tuples:** de la definició de l'extensió d'una relació com un conjunt de tuples es dedueix també que aquestes tuples no estaran ordenades, atès que no és possible que hi hagi una ordenació entre els elements d'un conjunt.

La finalitat d'aquesta característica és aconseguir que, mitjançant el model relacional, es puguin representar els fets en un nivell abstracte que sigui independent de la seva estructura física d'implementació. Més concretament, encara que els SGBD relacionals han de proporcionar una implementació física que emmagatzemarà les tuples de les relacions en un ordre concret, aquesta ordenació no és visible si ens situem en el nivell conceptual.

Exemple de no-ordenació de les tuples

En una base de dades relacional, per exemple, no té sentit consultar la "primera tupla" de la relació *EMPLEATS*.

**4) No-ordenació dels atributs:** l'esquema d'una relació consta d'un nom de relació  $R$  i un conjunt d'atributs  $\{A_1, A_2, \dots, A_n\}$ . Així, doncs, no hi ha un ordre entre els atributs d'un esquema de relació, atès que aquests atributs formen un conjunt.

Com en el cas anterior, l'objectiu d'aquesta característica és representar els fets en un nivell abstracte, independent de la seva implementació física.

Exemple de no-ordenació dels atributs

L'esquema de relació *EMPLEATS*(*DNI*, *nom*, *cognom*, *sou*) denota el mateix esquema de relació que *EMPLEATS*(*nom*, *cognom*, *DNI*, *sou*).

El concepte d'*extensió* d'una relació s'ha explicat al subapartat 2.2 d'aquest mòdul didàctic.

El concepte d'*esquema* d'una relació s'ha explicat al subapartat 2.2 d'aquest mòdul didàctic.

## 2.4. Clau candidata, clau primària i clau alternativa de les relacions

Tota la informació que conté una base de dades s'ha de poder identificar d'alguna manera. En el cas particular de les bases de dades que segueixen el model rela-

cional, per a identificar les dades que la base de dades conté, es poden utilitzar les claus candidates de les relacions. A continuació definim què s'entén per *clau candidata* i, també, què s'entén per *clau primària* i *clau alternativa* d'una relació. Per a fer-ho, però, abans caldrà definir el concepte de *superclau*.

Una **superclau** d'una relació d'esquema  $R(A_1, A_2, \dots, A_n)$  és un subconjunt dels atributs de l'esquema tal que no hi pot haver dues tuples en l'extensió de la relació que tinguin la mateixa combinació de valors per als atributs del subconjunt.

Una superclau, per tant, ens permet identificar totes les tuples que conté la relació.

Algunes superclaus de la relació *EMPLEATS*

A la relació d'esquema *EMPLEATS*(*DNI, NSS, nom, cognom, telèfon*), algunes de les superclaus de la relació serien els subconjunts d'atributs següents: {*DNI, NSS, nom, cognom, telèfon*}, {*DNI, cognom*}, {*DNI*} i {*NSS*}.

Una **clau candidata** d'una relació és una superclau *C* de la relació que compleix que cap subconjunt propi de *C* no és superclau.

És a dir, *C* compleix que l'eliminació de qualsevol dels seus atributs dóna un conjunt d'atributs que no és superclau de la relació. Intuïtivament, una clau candidata permet identificar qualsevol tupla d'una relació, de manera que no sobri cap atribut per a fer la identificació.

Claus candidates de *EMPLEATS*

A la relació d'esquema *EMPLEATS*(*DNI, NSS, nom, cognom, telèfon*), només hi ha dues claus candidates: {*DNI*} i {*NSS*}.

Habitualment, una de les claus candidates d'una relació es designa clau primària de la relació. La **clau primària** és la clau candidata, els valors de la qual s'utilitzaran per a identificar les tuples de la relació.

El dissenyador de la base de dades és qui escull la clau primària d'entre les claus candidates.

Les claus candidates no escollides com a primària s'anomenen **claus alternatives**.

Per exemple,...

... si s'emmagatzema informació dels empleats d'una empresa, cal poder distingir quines dades corresponen a cadascun dels diferents empleats.

Observeu que...

... tota relació té, almenys, una superclau, que és la formada per tots els atributs del seu esquema. Això es deu a la propietat que compleix tota relació de no tenir tuples repetides.


En l'exemple *EMPLEATS*(*DNI, NSS, nom, cognom, telèfon*) aquesta superclau seria: {*DNI, NSS, nom, cognom, telèfon*}.

Fixeu-vos que,...

... com que tota relació té almenys una superclau, podem garantir que tota relació té com a mínim una clau candidata.

Relació amb una clau candidata

Si una relació només té una clau candidata, llavors aquesta clau candidata n'ha de ser també la clau primària. Com que totes les relacions tenen una clau candidata com a mínim, podem garantir que, per a tota relació, serà possible designar una clau primària.

Usarem la convenció de subratllar els atributs que formen part de la clau primària en l'esquema de la relació. Així, doncs,  $R(\underline{A_1}, \underline{A_2}, \dots, \underline{A_i}, \dots, A_n)$  indica que els atributs  $A_1, A_2, \dots, A_i$  formen la clau primària de  $R$ . 

Elecció de la clau primària de *EMPLEATS*

A la relació d'esquema *EMPLEATS*(*DNI*, *NSS*, *nom*, *cognom*, *telèfon*), en què hi ha dues claus candidates,  $\{DNI\}$  i  $\{NSS\}$ , es pot escollir com a clau primària  $\{DNI\}$ . Ho indicarem subratllant l'atribut *DNI* a l'esquema de la relació *EMPLEATS*(*DNI*, *NSS*, *nom*, *cognom*, *telèfon*). En aquest cas, la clau  $\{NSS\}$  serà una clau alternativa d'*EMPLEATS*.

És força possible que una clau candidata o una clau primària consti de més d'un atribut.

Clau primària de la relació *DESPATXOS*

A la relació d'esquema *DESPATXOS*(*edifici*, *número*, *superfície*), la clau primària és formada pels atributs *edifici* i *número*. En aquest cas, podrà passar que dos despatxos diferents siguin al mateix edifici, o bé que tinguin el mateix número, però mai no podrà passar que tinguin la mateixa combinació de valors per a *edifici* i *número*.

## 2.5. Claus foranes de les relacions

Fins ara hem estudiat les relacions de manera individual, però hem de tenir en compte que una base de dades relacional normalment conté més d'una relació, per tal de poder representar diversos tipus de fets que succeeixen al món real. Per exemple, podríem tenir una petita base de dades que contingués dues relacions: una que tingués per nom *EMPLEATS*, que emmagatzemaria dades dels empleats d'una empresa, i una altra amb el nom *DESPATXOS*, que emmagatzemaria les dades dels despatxos que té l'empresa.

Hem de considerar també que entre els diversos fets que es donen al món real hi sol haver lligams o vincles entre ells. Per exemple, els empleats que treballen per a una empresa poden estar lligats amb els despatxos de l'empresa perquè a cada empleat se li assigna un despatx concret per a treballar.

En el model relacional, per a reflectir aquests tipus de vincles, hi ha la possibilitat d'expressar connexions entre les diferents tuples de les relacions. Per exemple, en la base de dades anterior, que té les relacions *EMPLEATS* i *DESPATXOS*, pot ser necessari connectar tuples d'*EMPLEATS* amb tuples de *DESPATXOS* per a indicar quin despatx té assignat cada empleat.

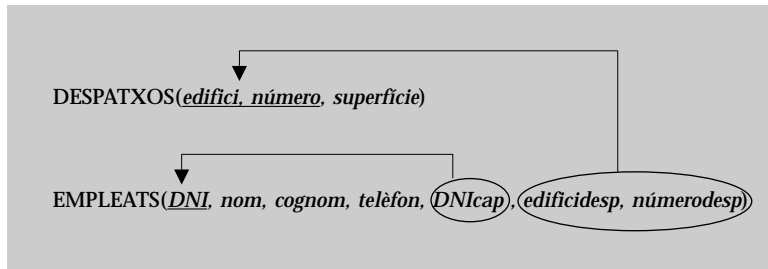
De vegades, fins i tot pot ser necessari reflectir lligams entre tuples que pertanyen a una mateixa relació. Per exemple, en la mateixa base de dades anterior, pot ser necessari connectar determinades tuples d'*EMPLEATS* amb altres tuples d'*EMPLEATS* per a indicar, per a cada empleat, qui fa de cap.

El mecanisme que proporcionen les bases de dades relacionals per a connectar tuples són les claus foranes de les relacions. Les **claus foranes** permeten establir connexions entre les tuples de les relacions. Per a fer la connexió, una clau fora-

na té un conjunt d'atributs d'una relació que referencien la clau primària d'una altra relació (o fins i tot de la mateixa relació).

#### Claus foranes de la relació *EMPLEATS*

A la figura següent, la relació *EMPLEATS*(*DNI*, *nom*, *cognom*, *telèfon*, *DNicap*, *edificidesp*, *númerodesp*), té una clau forana formada pels atributs *edificidesp* i *númerodesp* que es refereix a la clau primària de la relació *DESPATXOS*(*edifici*, *número*, *superfície*). Aquesta clau forana indica, per a cada empleat, el despatx on treballa. A més, l'atribut *DNicap* és una altra clau forana que referencia la clau primària de la mateixa relació *EMPLEATS*, i indica, per a cada empleat, quin és l'empleat que li fa de cap.



Les claus foranes tenen per objectiu establir una connexió amb la clau primària que referencien. Per tant, els valors d'una clau forana han de ser presents a la clau primària corresponent, o bé han de ser valors nuls. En cas contrari, la clau forana representaria una referència o connexió incorrecta.

#### Exemple

A la relació d'esquema *EMPLEATS*(*DNI*, *nom*, *cognom*, *DNicap*, *edificidesp*, *númerodesp*), la clau forana {*edificidesp*, *númerodesp*} referencia la relació *DESPATXOS*(*edifici*, *número*, *superfície*). Així, doncs, es compleix que tots els valors que no són nuls dels atributs *edificidesp* i *númerodesp* són valors que existeixen per als atributs *edifici* i *número* de *DESPATXOS*, tal com es pot veure a continuació:

- Relació *DESPATXOS*:

DESPATXOS		
<i>edifici</i>	<i>número</i>	<i>superfície</i>
Marina	120	10
Marina	122	15
Marina	230	20
Diagonal	120	10

- Relació *EMPLEATS*:

EMPLEATS					
<i>DNI</i>	<i>nom</i>	<i>cognom</i>	<i>DNicap</i>	<i>edificidesp</i>	<i>númerodesp</i>
40.444.255	Joan	Garcia	NULL	Marina	120
33.567.711	Marta	Roca	40.444.255	Marina	120
55.898.425	Carles	Bonmatí	40.444.255	Diagonal	120
77.232.144	Elena	Pla	40.444.255	NULL	NULL



Suposem que hi hagués un empleat amb els valors <55.555.555, Maria, Casagran, NULL, París, 400>. Com que no hi ha cap despatx amb els valors París i 400 per a *edifici* i *número*, la tupla d'aquest empleat fa una referència incorrecta, és a dir, indica un despatx per a l'empleat que, de fet, no existeix.

Cal assenyalar que a la relació *EMPLEATS* hi ha una altra clau forana,  $\{DNIcap\}$ , que referencia la mateixa relació *EMPLEATS*, i aleshores es compleix que tots els valors que no són nuls de l'atribut *DNIcap* són valors que existeixen per a l'atribut *DNI* de la mateixa relació *EMPLEATS*.

A continuació establim de manera més precisa què s'entén per *clau forana*.


Una **clau forana d'una relació  $R$**  és un subconjunt dels atributs de l'esquema de la relació, que anomenem *CF*, que compleix les condicions següents:

- 1) Existeix una relació *S* (*S* no ha de ser necessàriament diferent de *R*) que té per clau primària *CP*.
- 2) Es compleix que, per a tota tupla *t* de l'extensió de *R*, els valors per a *CF* de *t* són o bé valors nuls, o bé valors que coincideixen amb els valors per a *CP* d'alguna tupla *s* de *S*.

I, llavors, es diu que la clau forana *CF* referencia la clau primària *CP* de la relació *S*, i també que la clau forana *CF* referencia la relació *S*.

Convé subratllar que,...

... tal com ja hem esmentat, el model relacional permet representar tota la informació mitjançant valors explícits que contenen les relacions, i no li cal res més. Així, doncs, les connexions entre tuples de les relacions s'expressen amb els valors explícits de les claus foranes de les relacions i no són necessaris conceptes addicionals, per exemple apuntadors entre tuples, per a establir aquestes connexions. Aquesta característica dona simplicitat i uniformitat al model.

De la noció que hem donat de *clau forana* es poden extreure diverses conseqüències: 

- 1) Si una clau forana *CF* referencia una clau primària *CP*, el nombre d'atributs de *CF* i de *CP* ha de coincidir.

Exemple de coincidència del nombre d'atributs de *CF* i *CP*

A l'exemple anterior, tant la clau forana  $\{edificidesp, númerodesp\}$  com la clau primària que referencia  $\{edifici, número\}$ , tenen dos atributs. Si això no fos així, no seria possible que els valors de *CF* existissin a *CP*.

- 2) Pel mateix motiu, es pot establir una correspondència (en concret, una bijectió) entre els atributs de la clau forana i els atributs de la clau primària que referencia.


Exemple de correspondència entre els atributs de *CF* i els de *CP*

A l'exemple anterior, a *edificidesp* li correspon l'atribut *edifici* i a *númeroesp* li correspon l'atribut *número*.

- 3) També es dedueix de la noció de *clau forana* que els dominis dels seus atributs han de coincidir amb els dominis dels atributs corresponents a la clau primària que referencia. Aquesta coincidència de dominis fa que sigui possible que els valors de la clau forana coincideixin amb valors de la clau primària referenciada.

### Exemple de coincidència dels dominis

A l'exemple anterior, s'ha de complir que  $\text{domini}(\text{edificidesp}) \subseteq \text{domini}(\text{edifici})$  i també que  $\text{domini}(\text{númerodesp}) \subseteq \text{domini}(\text{número})$ .

Observeu que, de fet, aquesta condició es podria relaxar i es podria permetre que els dominis no fossin exactament iguals, sinó que fossin, només, i d'alguna manera que convindria precisar, dominis “compatibles”. Per a simplificar-ho, nosaltres suposarem que els dominis han de ser iguals en tots els casos en què, segons Date (1995), s'acceptarien dominis “compatibles”. 

#### Lectura recomanada

Trobareu explicacions detallades sobre la coincidència de dominis a l'obra següent:

**C.J. Date** (2001). *Introducción a los sistemas de bases de datos* (7a ed., cap. 19). Prentice-Hall.


### Exemple d'atribut que forma part de la clau primària i d'una clau forana


Es pot donar el cas que algun atribut d'una relació formi part tant de la clau primària com d'una clau forana de la relació. Aquest cas es dona en les relacions següents: *EDIFICIS*(nomedifici, adreça), i *DESPATXOS*(edifici, número, superfície), on {*edifici*} és una clau forana que referencia *EDIFICIS*.

En aquest exemple, l'atribut *edifici* forma part tant de la clau primària com de la clau forana de la relació *DESPATXOS*.

## 2.6. Creació de les relacions d'una base de dades

Hem vist que una base de dades relacional consta de diverses relacions. Cada relació té diversos atributs que prenen valors d'uns certs dominis, també té una clau primària i pot tenir una o més claus foranes. Els **llenguatges dels SGBD relacionals** han de proporcionar la manera de definir tots aquests elements per a crear una base de dades.

Més endavant es veurà amb detall la sintaxi i el significat de les sentències de definició de la base de dades per al cas concret del llenguatge SQL. 

El llenguatge SQL s'explica al mòdul didàctic “El llenguatge SQL” d'aquesta assignatura. 

### 3. Operacions del model relacional

Les operacions del model relacional han de permetre manipular dades emmagatzemades en una base de dades relacional i, per tant, estructurades en forma de relacions. La manipulació de les dades inclou bàsicament dos aspectes: l'actualització i la consulta.

La sintaxi i el funcionament de les operacions d'actualització i consulta, en el cas concret del llenguatge relacional SQL, s'estudien amb detall al mòdul "El llenguatge SQL" d'aquesta assignatura.

**L'actualització de les dades** consisteix a fer que els canvis que es produeixen a la realitat quedin reflectits a les relacions de la base de dades.

#### Exemple d'actualització

Si una base de dades conté, per exemple, informació dels empleats d'una empresa i l'empresa contracta un empleat, caldrà reflectir aquest canvi afegint les dades del nou empleat a la base de dades.

Hi ha tres operacions bàsiques d'actualització: !

- a) **Inserció**, que serveix per a afegir una o més tuples a una relació.
- b) **Esborrat**, que serveix per a eliminar una o més tuples d'una relació.
- c) **Modificació**, que serveix per a alterar els valors que tenen una o més tuples d'una relació per a un o més dels seus atributs.

La **consulta de les dades** consisteix en l'obtenció de dades deduïbles a partir de les relacions que conté la base de dades.

#### Exemple de consulta

Si una base de dades conté, per exemple, informació dels empleats d'una empresa, pot interessar consultar el nom i cognom de tots els empleats que treballen en un despatx situat en un edifici que té per nom *Marina*.

L'obtenció de les dades que responen a una consulta pot requerir l'anàlisi i l'extracció de dades d'una o més de les relacions que té la base de dades.


Segons la manera com especifiquen les consultes, podem classificar els llenguatges relacionals en dos tipus: !


**1) Llenguatges basats en l'àlgebra relacional.** L'àlgebra relacional s'inspira en la teoria de conjunts. Per a especificar una consulta cal seguir un o més passos

que serveixen per a anar construint, mitjançant operacions de l'àlgebra relacional, una nova relació que contingui les dades que responen a la consulta a partir de les relacions emmagatzemades. Els llenguatges basats en l'àlgebra relacional són **llenguatges procedimentals**, atès que els passos que formen la consulta descriuen un procediment.

**2) Llenguatges basats en el càlcul relacional.** El càlcul relacional té el seu fonament teòric en el càlcul de predicats de la lògica matemàtica. Proporciona una notació que permet formular la definició de la relació que conté les dades que responen la consulta en termes de les relacions emmagatzemades. Aquesta definició no descriu un procediment i, per tant, es diu que els llenguatges basats en el càlcul relacional són **llenguatges declaratius** (no procedimentals).

El **llenguatge SQL**, en les sentències de consulta, combina construccions de l'àlgebra relacional i del càlcul relacional amb un predomini de les construccions del càlcul. Aquest predomini determina que l'SQL sigui un llenguatge declaratiu.

L'**estudi de l'àlgebra relacional** presenta un interès especial perquè ajuda a entendre quins serveis de consulta ha de proporcionar un llenguatge relacional, facilita la comprensió d'algunes de les construccions del llenguatge SQL i també serveix de base per al tractament de les consultes que efectuen els SGBD internament. Aquest darrer tema queda fora de l'àmbit de la present assignatura, però és necessari per a estudis més avançats sobre bases de dades. 

L'àlgebra relacional s'explica a l'apartat 5 d'aquest mòdul didàctic. 

## 4. Regles d'integritat

Una base de dades conté unes dades que, en cada moment, han de reflectir la realitat o, més concretament, la situació d'una porció del món real. En el cas de les bases de dades relacionals, això significa que l'extensió de les relacions, és a dir, les tuples que les relacions contenen, han de tenir valors que reflecteixin la realitat correctament. !

Sol ser força freqüent que determinades configuracions de valors per a les tuples de les relacions no tenen sentit, perquè no representen cap situació possible del món real.

Un sou negatiu

A la relació d'esquema *EMPLEATS(DNI, nom, cognom, sou)*, una tupla que té un valor de 21.000 per al sou, probablement no té sentit perquè els sous no poden ser negatius.

Anomenem **integritat** la propietat de les dades de correspondre a representacions plausibles del món real.

Evidentment, perquè les dades siguin íntegres, cal que compleixin diverses condicions.

El fet que els sous no puguin ser negatius, és una condició que s'hauria de complir a la relació *EMPLEATS*.

En general, les condicions que garanteixen la integritat de les dades poden ser de dos tipus: !

1) Les **restriccions d'integritat d'usuari** són condicions específiques d'una base de dades concreta, és a dir, són les que s'han de complir en una base de dades particular amb uns usuaris concrets, però que no són necessàriament rellevants en una altra base de dades.

Restricció d'integritat d'usuari a *EMPLEATS*

Aquest seria el cas de la condició anterior, segons la qual els sous no podien ser negatius. Observeu que aquesta condició era necessària en la base de dades concreta d'aquest exemple perquè hi apareixia l'atribut *sou* i pel significat que es volia donar a aquest atribut, però podria no ser necessària en una altra base de dades diferent on, per exemple, no hi hagués sous.

2) Les **regles d'integritat de model**, en canvi, són condicions més generals, pròpies d'un model de dades i s'han de complir en tota base de dades que segueixi aquell model.

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

En el cas del model de dades relacional, hi haurà una regla d'integritat per a garantir que els valors d'una clau primària d'una relació no es repeteixen en tuples diferents de la relació. Tota base de dades relacional ha de complir aquesta regla, que, per tant, és una regla d'integritat del model.

Els SGBD han de proporcionar la manera de definir les restriccions d'integritat d'usuari d'una base de dades i, un cop definides, han de vetllar perquè se satisfacin.

La manera de definir aquestes restriccions amb el llenguatge SQL s'explica al mòdul "El llenguatge SQL" d'aquesta assignatura.

Les regles d'integritat del model, en canvi, no s'han de definir per a cada base de dades concreta, perquè es consideren preestablertes per a totes les bases de dades d'un model. Un SGBD d'un model determinat ha de vetllar perquè es compleixin les regles d'integritat preestablertes pel seu model.

A continuació estudiarem amb detall les **regles d'integritat del model relacional**, regles que tot SGBD relacional ha de fer complir. !

#### 4.1. Regla d'integritat d'unicitat de la clau primària

La regla d'integritat d'unicitat està relacionada amb la definició de clau primària. Concretament, estableix que tota clau primària que s'esculli per a una relació no ha de tenir valors repetits.

Cal destacar que el mateix concepte de *clau primària* implica aquesta condició. El concepte de *clau primària* s'ha explicat al subapartat 2.4 d'aquest mòdul didàctic.

Exemple

Tenim la relació següent:

DESPATXOS		
<i>edifici</i>	<i>número</i>	<i>superfície</i>
Marina	120	10
Marina	122	15
Marina	230	20
Diagonal	120	10

En aquesta relació, com que la clau primària és formada per *edifici* i *número*, no hi ha cap despatx que repeteixi tant *edifici* com *número* d'un altre despatx. Però sí que es repeteixen valors d'*edifici*, per exemple Marina; i també es repeteixen valors de *número*, per exemple 120. Tot i això, l'*edifici* i el *número* no es repeteixen mai alhora.

Tot seguit expliquem aquesta regla de manera més precisa.

La **regla d'integritat d'unicitat de la clau primària** estableix que si el conjunt d'atributs *CP* és la clau primària d'una relació *R*, aleshores l'extensió de *R* no pot tenir en cap moment dues tuples amb la mateixa combinació de valors per als atributs de *CP*.

Un SGBD relacional haurà de garantir el compliment d'aquesta regla d'integritat per a totes les insercions i, també, per a totes les modificacions que afectin atributs que pertanyen a la clau primària de la relació.

#### Exemple

Tenim la relació següent:

DESPATXOS		
<i>edifici</i>	<i>número</i>	<i>superfície</i>
Marina	120	10
Marina	122	15
Marina	230	20
Diagonal	120	10

En aquesta relació no s'hauria de poder inserir la tupla <Diagonal, 120, 30>, ni modificar la tupla <Marina, 122, 15> de manera que passés a ser <Marina, 120, 15>.

## 4.2. Regla d'integritat d'entitat de la clau primària

La regla d'integritat d'entitat de la clau primària disposa que els atributs de la clau primària d'una relació no poden tenir valors nuls.

#### Exemple

Tenim la relació següent:

DESPATXOS		
<i>edifici</i>	<i>número</i>	<i>superfície</i>
Marina	120	10
Marina	122	15
Marina	230	20
Diagonal	120	10

En aquesta relació, com que la clau primària és formada per *edifici* i *número*, no hi ha cap despatx que tingui un valor nul per a *edifici*, ni tampoc per a *número*.

Aquesta regla és necessària perquè els valors de les claus primàries serveixen per a identificar les tuples individuals de les relacions. Si les claus primàries tinguessin valors nuls, podria passar que algunes tuples no es poguessin distingir.

#### Exemple de clau primària incorrecta amb valors nuls

A l'exemple anterior, si un despatx tingués un valor nul per a *edifici* perquè en un moment donat el nom de l'edifici no es coneix, per exemple <NULL, 120, 30>, la clau primària no ens permetria distingir-lo del despatx <Marina, 120, 10> ni del despatx <Diagonal, 120, 10>. No podríem estar segurs que el valor desconegut d'*edifici* no és ni Marina ni Diagonal.

A continuació definim aquesta regla de manera més precisa.

**La regla d'integritat d'entitat de la clau primària** estableix que si el conjunt d'atributs *CP* és la clau primària d'una relació *R*, l'extensió de *R* no pot tenir en cap moment cap tupla que tingui un valor nul per a algun dels atributs de *CP*.

Un SGBD relacional haurà de garantir l'acompliment d'aquesta regla d'integritat en totes les insercions i, també, en totes les modificacions que afectin atributs que pertanyen a la clau primària de la relació.

Exemple

A la relació *DESPATXOS* anterior, no s'hauria de poder inserir la tupla <Diagonal, NULL, 15>. Tampoc no hauria de ser possible modificar la tupla <Marina, 120, 10> de manera que passés a ser <NULL, 120, 10>.

#### 4.3. Regla d'integritat referencial

La regla d'integritat referencial està relacionada amb el concepte de *clau forana*. Concretament, determina que tots els valors que pren una clau forana han de ser valors nuls o valors que existeixen a la clau primària que referencia.

Observeu que tot el que imposa la regla d'integritat referencial és implicat per la mateixa noció de *clau forana*, que s'ha explicat al subapartat 2.5 d'aquest mòdul.

Exemple

Si tenim les relacions següents:

- Relació *DESPATXOS*:

DESPATXOS		
<i>edifici</i>	<i>número</i>	<i>superficie</i>
Marina	120	10
Marina	122	15
Marina	230	20
Diagonal	120	10

- Relació *EMPLEATS*:

EMPLEATS				
<i>DNI</i>	<i>nom</i>	<i>cognom</i>	<i>edificidesp</i>	<i>númerodesp</i>
40.444.255	Joan	Garcia	Marina	120
33.567.711	Marta	Roca	Marina	120
55.898.425	Carles	Bonmatí	Diagonal	120
77.232.144	Elena	Pla	NULL	NULL



on *edificidesp* i *númerodesp* de la relació *EMPLEATS* formen una clau forana que referencia la relació *DESPATXOS*, llavors ha de passar que els valors no nuls d'*edificidesp* i *númerodesp* de la relació *EMPLEATS* siguin a la relació *DESPATXOS* com a valors d'*edifici* i *número*. Per exemple, l'empleat <40.444.255, Joan, Garcia, Marina, 120> té el valor Marina per a *edificidesp* i el valor 120 per a *númerodesp*, de manera que a la relació *DESPATXOS* hi ha un despatx amb valor Marina per a *edifici* i amb valor 120 per a *número*.

La necessitat de la regla d'integritat referencial prové del fet que les claus foranes tenen per objectiu establir una connexió amb la clau primària que referencien. Si un valor d'una clau forana no fos present a la clau primària corresponent, representaria una referència o una connexió incorrecta.

#### Referència incorrecta

Suposem que a l'exemple anterior hi hagués un empleat amb els valors <56.666.789, Pere, López, València, 325>. Com que no hi ha cap despatx amb els valors València i 325 per a *edifici* i *número*, la tupla d'aquest empleat fa una referència incorrecta, és a dir, indica un despatx per a l'empleat que, de fet, no existeix.

Tot seguit expliquem la regla de manera més precisa.

**La regla d'integritat referencial** estableix que si el conjunt d'atributs *CF* és una clau forana d'una relació *R* que referencia una relació *S* (no necessàriament diferent de *R*) que té per clau primària *CP*, aleshores, per a tota tupla *t* de l'extensió de *R*, els valors per al conjunt d'atributs *CF* de *t* són o bé valors nuls, o bé valors que coincideixen amb els valors per a *CP* d'alguna tupla *s* de *S*.

En el cas en què una tupla *t* de l'extensió de *R* tingui valors per a *CF* que coincideixen amb els valors per a *CP* d'una tupla *s* de *S*, diem que *t* és una tupla que referencia *s* i que *s* és una tupla que té una clau primària referenciada per *t*.

Un SGBD relacional haurà de fer complir aquesta regla d'integritat. Haurà d'efectuar comprovacions quan es produeixin les operacions següents:

- a) Insercions en una relació que tingui una clau forana.
- b) Modificacions que afectin atributs que pertanyen a la clau forana d'una relació.
- c) Esborrats en relacions referenciades per altres relacions.
- d) Modificacions que afectin atributs que pertanyen a la clau primària d'una relació referenciada per alguna altra relació.

#### Exemple

Reprenem l'exemple anterior, on *edificidesp* i *númerodesp* de la relació *EMPLEATS* formen una clau forana que referencia la relació *DESPATXOS*:

- Relació *DESPATXOS*:

DESPATXOS		
<i>edifici</i>	<i>número</i>	<i>superfície</i>
Marina	120	10
Marina	122	15
Marina	230	20
Diagonal	120	10

- Relació *EMPLEATS*:

EMPLEATS				
<i>DNI</i>	<i>nom</i>	<i>cognom</i>	<i>edificidesp</i>	<i>númerodesp</i>
40.444.255	Joan	Garcia	Marina	120
33.567.711	Marta	Roca	Marina	120
55.898.425	Carles	Bonmatí	Diagonal	120
77.232.144	Elena	Pla	NULL	NULL


Les operacions següents provocarien l'incompliment de la regla d'integritat referencial:

- Inserció de <12.764.411, Jordi, Puig, Diagonal, 220> a *EMPLEATS*.
- Modificació de <40.444.255, Joan, Garcia, Marina, 120> d'*EMPLEATS* per <40.444.255, Joan, Garcia, Marina, 400>.
- Esborrat de <Marina, 120, 10> de *DESPATXOS*.
- Modificació de <Diagonal, 120, 10> de *DESPATXOS* per <París, 120, 10>.

Un SGBD relacional ha de vetllar perquè es compleixin les regles d'integritat del model. Una manera habitual de mantenir aquestes regles és rebutjar tota operació d'actualització que deixi la base de dades en un estat en què alguna regla no es compleixi. En alguns casos, però, el SGBD té la possibilitat d'acceptar l'operació i efectuar accions addicionals compensatòries, de manera que l'estat que s'obtingui satisfaci les regles d'integritat, tot i haver executat l'operació.

Aquesta darrera política pot ser aplicada en les operacions d'actualització que violarien la regla d'integritat referencial següents:

- a) Esborrat d'una tupla que té una clau primària referenciada.
- b) Modificació dels valors d'atributs de la clau primària d'una tupla que té una clau primària referenciada.

En els casos anteriors, algunes de les polítiques que es podran aplicar seran les següents: **restricció**, **actualització en cascada** i **anul·lació**. A continuació expliquem el significat de les tres possibilitats esmentades. 

### 4.3.1. Restricció

La política de restricció consisteix a no acceptar l'operació d'actualització.

Més concretament, la **restricció en cas d'esborrat** consisteix a no permetre esborrar una tupla si té una clau primària referenciada per alguna clau forana.

Similarment, la **restricció en cas de modificació** consisteix a no permetre modificar cap atribut de la clau primària d'una tupla si té una clau primària referenciada per alguna clau forana.

Exemple d'aplicació de la restricció

Suposem que tenim les relacions següents:

- Relació *CLIENTS*:

CLIENTS	
<u>numclient</u>	...
10	—
15	—
18	—

- Relació *COMANDES\_PENDENTS*:

COMANDES_PENDENTS		
<u>numcom</u>	...	numclient*
1.234	—	10
1.235	—	10
1.236	—	15

\* {numclient} referencia *CLIENTS*.

**a)** Si apliquem la restricció en cas d'esborrat i, per exemple, volem esborrar el client número 10, no serà possible fer-ne l'esborrat perquè té comandes pendents que el referencien.

**b)** Si apliquem la restricció en cas de modificació i volem modificar el número del client 15, no serà possible fer-ne la modificació perquè també té comandes pendents que el referencien.

### 4.3.2. Actualització en cascada

La política d'actualització en cascada consisteix a permetre l'operació d'actualització de la tupla i a efectuar operacions compensatòries que propaguin en cascada l'actualització a les tuples que la referenciaven, per a mantenir la integritat referencial.

Més concretament, l'**actualització en cascada en cas d'esborrat** consisteix a permetre l'esborrat d'una tupla  $t$  que té una clau primària referenciada i esborrar també totes les tuples que referencien  $t$ .

Similarment, l'**actualització en cascada en cas de modificació** consisteix a permetre la modificació d'atributs de la clau primària d'una tupla  $t$  que té una clau primària referenciada i modificar de la mateixa manera totes les tuples que referencien  $t$ .

Exemple d'aplicació de l'actualització en cascada

Suposem que tenim les relacions següents:

- Relació *EDIFICIS*:

EDIFICIS	
<i>nomedifici</i>	...
Marina	—
Diagonal	—

- Relació *DESPATXOS*:

DESPATXOS		
<i>edifici*</i>	<i>número</i>	<i>superfície</i>
Marina	120	10
Marina	122	15
Marina	230	20
Diagonal	120	10

\* {edifici} referencia *EDIFICIS*.

a) Si apliquem l'actualització en cascada en cas d'esborrat i, per exemple, volem esborrar l'edifici Diagonal, s'esborrarà també el despatx Diagonal 120 que hi ha a l'edifici i ens quedarà:

- Relació *EDIFICIS*:

EDIFICIS	
<i>nomedifici</i>	...
Marina	—

- Relació *DESPATXOS*:

DESPATXOS		
<i>edifici*</i>	<i>número</i>	<i>superfície</i>
Marina	120	10
Marina	122	15
Marina	230	20

\* {edifici} referencia *EDIFICIS*.

**b)** Si apliquem l'actualització en cascada en cas de modificació i volem modificar el nom de l'edifici Marina per Mar, també es canviarà Marina per Mar als despatxos Marina 120, Marina 122 i Marina 230, i ens quedarà:

- Relació *EDIFICIS*:

EDIFICIS	
<i>nomedifici</i>	...
Mar	—

- Relació *DESPATXOS*:

DESPATXOS		
<i>edifici*</i>	<i>número</i>	<i>superfície</i>
Mar	120	10
Mar	122	15
Mar	230	20

\* {edifici} referencia *EDIFICIS*.

#### 4.3.3. Anul·lació

Aquesta política consisteix a permetre l'operació d'actualització de la tupla i a efectuar operacions compensatòries que posen valors nuls als atributs de la clau forana de les tuples que la referencien, amb l'objectiu de mantenir la integritat referencial.

Com que generalment els SGBD relacionals permeten establir que un determinat atribut d'una relació no admet valors nuls, només s'hi pot aplicar la política d'anul·lació si els atributs de la clau forana admeten valors nuls.

Més concretament, l'**anul·lació en cas d'esborrat** consisteix a permetre l'esborrat d'una tupla  $t$  que té una clau referenciada  $i$ , a més, modificar totes les tuples que referencien  $t$ , de manera que els atributs de la clau forana corresponent prenguin valors nuls.

Similarment, l'**anul·lació en cas de modificació** consisteix a permetre la modificació d'atributs de la clau primària d'una tupla  $t$  que té una clau referenciada  $i$ , a més, modificar totes les tuples que referencien  $t$ , de manera que els atributs de la clau forana corresponent prenguin valors nuls.

Exemple d'aplicació de l'anul·lació

La millor manera d'entendre en què consisteix l'anul·lació és mitjançant un exemple. Tenim les relacions següents:

- Relació *VENEDORS*:

VENEDORS	
<i>numvenedor</i>	...
1	—
2	—
3	—

- Relació *CLIENTS*:

CLIENTS		
<i>numclient</i>	...	<i>venedorassig*</i>
23	—	1
35	—	1
38	—	2
42	—	2
50	—	3

\* {*venedorassig*} referencia *VENEDORS*.

a) Si apliquem l'anul·lació en cas d'esborrat i, per exemple, volem esborrar el venedor número 1, es modificaran tots els clients que el tenien assignat i passaran a tenir un valor nul a *venedorassig*. Ens quedarà:

- Relació *VENEDORS*:

VENEDORS	
<i>numvenedor</i>	...
2	—
3	—

- Relació *CLIENTS*:

CLIENTS		
<i>numclient</i>	...	<i>venedorassig*</i>
23	—	NULL
35	—	NULL
38	—	2
42	—	2
50	—	3

\* {*venedorassig*} referencia *VENEDORS*.

b) Si apliquem l'anul·lació en cas de modificació, i ara volem canviar el número del venedor 2 per 5, es modificaran tots els clients que el tenien assignat i passaran a tenir un valor nul a *venedorassig*. Ens quedarà:

- Relació *VENEDORS*:

VENEDORS	
<i>numvenedor</i>	...
5	—
3	—

- Relació *CLIENTS*:

CLIENTS		
<i>numclient</i>	...	<i>venedorassig*</i>
23	—	NULL
35	—	NULL
38	—	NULL
42	—	NULL
50	—	3

\* {*venedorassig*} referencia *VENEDORS*.

#### 4.3.4. Selecció de la política de manteniment de la integritat referencial

Hem vist que en cas d'esborrat o modificació d'una clau primària referenciada per alguna clau forana hi ha diverses polítiques de manteniment de la regla d'integritat referencial.

La manera de definir aquestes polítiques de manteniment de la integritat amb el llenguatge SQL s'explica al mòdul "El llenguatge SQL" d'aquesta assignatura.

El dissenyador pot escollir per a cada clau forana quina política s'aplicarà en cas d'esborrat de la clau primària referenciada i quina en cas de modificació de la clau primària referenciada. El dissenyador haurà de tenir en compte el significat de cada clau forana concreta per a poder escollir adequadament. !

Aplicació de polítiques diferents

Pot passar que, per a una determinada clau forana, la política adequada en cas d'esborrat sigui diferent de la política adequada en cas de modificació. Per exemple, pot ser necessari aplicar la restricció en cas d'esborrat i l'actualització en cascada en cas de modificació.

#### 4.4. Regla d'integritat de domini

La regla d'integritat de domini està relacionada, com el nom ho indica, amb la noció de *domini*. Aquesta regla estableix dues condicions.

La **primera condició** és que un valor no nul d'un atribut  $A_i$  ha de pertànyer al domini de l'atribut  $A_j$ , és a dir, ha de pertànyer a  $\text{domini}(A_j)$ .

Aquesta condició implica que tots els valors no nuls que conté la base de dades per a un determinat atribut han de ser del domini declarat per a aquell atribut.

### Exemple

Si a la relació EMPLEATS(*DNI*, *nom*, *cognom*, *edatemp*) hem declarat que domini(*DNI*) és el domini predefinit dels enters, llavors no podrem inserir, per exemple, cap empleat que tingui per *DNI* el valor 'Lluís', que no és un enter.

Recordem que els dominis poden ser de dos tipus: predefinitos o definits per l'usuari. Observeu que els dominis definits per l'usuari resulten molt útils perquè ens permeten determinar de manera més específica quins seran els valors admesos pels atributs.

### Exemple

Suposem ara que a la relació EMPLEATS(*DNI*, *nom*, *cognom*, *edatemp*) hem declarat que domini(*edatemp*) és el domini definit per l'usuari *edat*. Suposem també que el domini *edat* s'ha definit com el conjunt dels enters que són entre 16 i 65. En aquest cas, per exemple, no serà possible inserir-hi un empleat amb un valor de 90 per a *edatemp*.

La segona condició de la regla d'integritat de domini és més complexa, especialment en el cas de dominis definits per l'usuari, i els SGBD actuals no la suporten per aquests darrers dominis. Per aquests motius només la presentarem superficialment. !

Aquesta **segona condició** serveix per a establir que els operadors que és possible aplicar sobre els valors depenen dels dominis d'aquests valors. És a dir, un operador determinat només es pot aplicar sobre valors que tinguin dominis adequats per a aquell operador.

### Exemple

Analitzarem aquesta segona condició de la regla d'integritat de domini amb un exemple concret. Si a la relació EMPLEATS(*DNI*, *nom*, *cognom*, *edatemp*) s'ha declarat que domini(*DNI*) és el domini predefinit dels enters, aleshores no es permetrà consultar tots els empleats el *DNI* dels quals sigui igual a 'Elena' ( $DNI \subseteq \text{'Elena'}$ ). El motiu és que no té sentit que l'operador de comparació  $\subseteq$  s'apliqui entre un *DNI* que té per domini els enters i el valor 'Elena', que és una tira de caràcters.

Així, doncs, el fet que els operadors que es poden aplicar sobre els valors depenguin del domini d'aquests valors permet detectar errors que es podrien cometre quan es consulta o s'actualitza la base de dades. Els dominis definits per l'usuari són molt útils, perquè ens permeten determinar de manera més específica quins seran els operadors que es podran aplicar sobre els valors.

### Exemple

Vegem un altre exemple amb dominis definits per l'usuari. Suposem que a la coneguda relació EMPLEATS(*DNI*, *nom*, *cognom*, *edatemp*) s'ha declarat que domini(*DNI*) és el domini definit per l'usuari *nombresDNI* i que domini(*edatemp*) és el domini definit per l'usuari *edat*. Suposem que *nombresDNI* correspon als enters positius i que *edat* correspon als enters que són entre 16 i 65. En aquest cas serà incorrecte, per exemple, consultar els empleats que tenen el valor de *DNI* igual al valor d'*edatemp*. El motiu és que, encara que tant els valors de *DNI* com els d'*edatemp* siguin enters, els seus dominis són diferents i, segons el significat que l'usuari els dona, no té sentit comparar-los.


!  
Recordeu que els conceptes de *domini predefinit* i *domini definit per l'usuari* s'han explicat al subapartat 2.2 d'aquest mòdul didàctic.

### Lectura complementària

Per a estudiar amb més profunditat la segona condició de la regla d'integritat de domini, podeu consultar l'obra següent:

**C.J. Date** (2001). *Introducción a los sistemas de bases de datos* (7a ed., cap. 19). Prentice-Hall.



Els SGBD relacionals actuals, però, no donen suport a la segona condició de la regla d'integritat de domini per dominis definits per l'usuari. Per a fer-ho, caldria que el dissenyador tingués alguna manera d'especificar, per a cada operador que es volgués utilitzar, en quines combinacions de dominis definits per l'usuari té sentit que s'apliqui. El llenguatge estàndard SQL no inclou actualment aquesta possibilitat. 

## 5. L'àlgebra relacional

Com ja hem comentat a l'apartat dedicat a les operacions del model relacional, l'àlgebra relacional s'inspira en la teoria de conjunts per a especificar consultes a una base de dades relacional.

Vegeu l'apartat 3 d'aquest mòdul didàctic.



Per a **especificar una consulta** en àlgebra relacional, cal seguir un o més passos que serveixen per a anar construint, mitjançant operacions de l'àlgebra relacional, una nova relació que contingui les dades que responen a la consulta a partir de les relacions emmagatzemades. Els llenguatges basats en l'àlgebra relacional són procedimentals, atès que els passos que formen la consulta descriuen un procediment.

La visió que presentarem és la d'un llenguatge teòric i, per tant, hi inclourem només les seves operacions fonamentals i no les construccions que es podrien afegir a un llenguatge comercial per a facilitar qüestions com ara l'ordre de presentació del resultat, el càlcul de dades agregades, etc.

Una característica destacable de totes les operacions de l'àlgebra relacional és que tant els operands com el resultat són relacions. Aquesta propietat s'anomena **tancament relacional**.

### Implicacions del tancament relacional

El fet que el resultat d'una operació de l'àlgebra relacional sigui una nova relació té implicacions importants:

1. El resultat d'una operació pot fer d'operand d'una altra operació.
2. El resultat d'una operació complirà totes les característiques que ja coneixem de les relacions: no-ordenació de les tuples, absència de tuples repetides, etc.

Les operacions de l'àlgebra relacional han estat classificades de diverses maneres, d'entre les quals assenyalarem les tres següents:

- 1) Si es poden expressar o no en termes d'altres operacions:
  - a) **Operacions primitives:** són aquelles a partir de les quals podem definir tota la resta. Aquestes operacions són la unió, la diferència, el producte cartesià, la selecció i la projecció.
  - b) **Operacions no primitives:** la resta d'operacions de l'àlgebra relacional que no són estrictament necessàries, perquè es poden expressar en termes de les primitives, però permeten formular algunes consultes de manera més còmoda. Hi ha diverses versions de l'àlgebra relacional segons les operacions no primitives que s'hi inclouen. Nosaltres estudiarem les operacions no primitives que s'utilitzen amb més freqüència: la intersecció i la combinació.
- 2) Segons el nombre de relacions que tenen com a operands:
  - a) **Operacions binàries:** són les que tenen dues relacions com a operands. Són binàries totes les operacions excepte la selecció i la projecció.

**b) Operacions unàries:** són aquelles que tenen una única relació com a operand. La selecció i la projecció són unàries.

**3) Si s'assemblen o no a les de la teoria de conjunts:**

**a) Operacions conjuntistes:** són les operacions que s'assemblen a les de la teoria de conjunts. Es tracta de la unió, la intersecció, la diferència i el producte cartesià.

**b) Operacions específicament relacionals:** són tota la resta d'operacions, és a dir, la selecció, la projecció i la combinació.

Les operacions de l'àlgebra relacional classificades segons si són conjuntistes o específicament relacionals s'estudien als subapartats 5.1 i 5.2 d'aquest mòdul.

Com ja hem dit anteriorment, les operacions de l'àlgebra relacional obtenen com a resultat una nova relació. Així, veurem que si fem una operació de l'àlgebra com ara  $EMPLEATS\_ADM < EMPLEATS\_PROD$  per a obtenir la unió de les relacions  $EMPLEATS\_ADM$  i  $EMPLEATS\_PROD$ , el resultat de l'operació és una nova relació que té la unió de les tuples de les relacions de partida.

Aquesta nova relació ha de tenir un nom. En principi, considerem que el seu nom és la mateixa expressió de l'àlgebra relacional que l'obté, és a dir, la mateixa expressió  $EMPLEATS\_ADM < EMPLEATS\_PROD$ . Com que aquest nom és llarg, de vegades pot ser interessant canviar-lo per un de més simple. Això ens facilitarà les referències a la nova relació i serà especialment útil en els casos en què vulguem utilitzar-la com a operand d'una altra operació. Farem servir l'operació auxiliar *reanomenar* amb aquest objectiu.

L'operació **reanomenar**, que denotarem amb el símbol  $\text{:}$ , permet assignar un nom  $R$  a la relació resultant d'una operació de l'àlgebra relacional de la manera següent:

$$R \text{:} E,$$

essent  $E$  l'expressió d'una operació de l'àlgebra relacional.

A l'exemple, per a donar el nom  $EMPLEATS$  a la relació resultant de l'operació  $EMPLEATS\_ADM < EMPLEATS\_PROD$ , faríem:


$$EMPLEATS \text{:} EMPLEATS\_ADM < EMPLEATS\_PROD.$$

Cada operació de l'àlgebra relacional dona uns noms per defecte als atributs de l'esquema de la relació resultant, tal com veurem més endavant.

En alguns casos, pot ser necessari canviar aquests noms per defecte per altres noms. Per aquest motiu, també permetrem canviar el nom de la relació i dels seus atributs mitjançant l'operació *reanomenar*.

Utilitzarem també l'operació *reanomenar* per a canviar l'esquema d'una relació. Si una relació té l'esquema  $S(B_1, B_2, \dots, B_n)$  i volem canviar-lo per  $R(A_1, A_2, \dots, A_n)$ , ho farem de la manera següent:

$$R(A_1, A_2, \dots, A_n) \text{ : } \bowtie S(B_1, B_2, \dots, B_n).$$

A continuació presentarem un exemple que farem servir per a il·lustrar les operacions de l'àlgebra relacional. Després veurem amb detall les operacions. 

Suposem que tenim una base de dades relacional amb les quatre relacions següents:

- 1) La relació *EDIFICIS\_EMP*, que conté dades de diversos edificis que una empresa té per a desenvolupar les seves activitats.
- 2) La relació *DESPATXOS*, que conté dades de cadascun dels despatxos que hi ha als edificis anteriors.
- 3) La relació *EMPLEATS\_ADM*, que conté les dades dels empleats de l'empresa que porten a terme tasques administratives.
- 4) La relació *EMPLEATS\_PROD*, que emmagatzema les dades dels empleats de l'empresa que s'ocupen de tasques de producció.

Tot seguit descrivim els esquemes de les relacions anteriors i les seves extensions en un moment determinat:

- Esquema i extensió d'*EDIFICIS\_EMP*:

EDIFICIS_EMP	
<i>edifici</i>	<i>superfície</i>
Marina	15
Diagonal	10

- Esquema i extensió de *DESPATXOS*:

DESPATXOS		
<i>edifici</i>	<i>número</i>	<i>superfície</i>
Marina	120	10
Marina	230	20
Diagonal	120	10
Diagonal	440	10

- Esquema i extensió d'*EMPLEATS\_ADM*:

EMPLEATS_ADM				
<i>DNI</i>	<i>nom</i>	<i>cognom</i>	<i>edificidesp</i>	<i>númerodesp</i>
40.444.255	Joan	Garcia	Marina	120
33.567.711	Marta	Roca	Marina	120

- Esquema i extensió d'*EMPLEATS\_PROD*:

EMPLEATS_PROD				
<i>DNIemp</i>	<i>nomemp</i>	<i>cognomemp</i>	<i>edificidesp</i>	<i>númerodesp</i>
33.567.711	Marta	Roca	Marina	120
55.898.425	Carles	Bonmatí	Diagonal	120
77.232.144	Elena	Pla	Marina	230
21.335.245	Jordi	Soler	NULL	NULL
88.999.210	Pere	González	NULL	NULL

Es considera que els valors nuls dels atributs *edificidesp* i *númerodesp* de les relacions *EMPLEATS\_PROD* i *EMPLEATS\_ADM* indiquen que l'empleat corresponent no té despatx.

### 5.1. Operacions conjuntistes

Les operacions conjuntistes de l'àlgebra relacional són la **unió**, la **intersecció**, la **diferència** i el **producte cartesià**.

#### 5.1.1. Unió


La unió és una operació que, a partir de dues relacions, obté una nova relació formada per totes les tuples que són en alguna de les relacions de partida.

La unió és una operació binària i la unió de dues relacions *T* i *S* s'indica  $T \cup S$ .

La unió de les relacions *EMPLEATS\_ADM* i *EMPLEATS\_PROD* dona una nova relació que conté tant els empleats d'administració com els empleats de producció i s'indicaria així:  $EMPLEATS_ADM \cup EMPLEATS_PROD$ .

Només té sentit aplicar la unió a relacions que tinguin tuples similars.

Per exemple, es pot fer la unió de les relacions *EMPLEATS\_ADM* i *EMPLEATS\_PROD* perquè les seves tuples s'assemblen. En canvi, no es podrà fer la unió de les relacions *EMPLEATS\_ADM* i *DESPATXOS* perquè, com heu pogut observar a les taules, les tuples respectives són de tipus diferent.

Més concretament, per a poder aplicar la unió a dues relacions, cal que les dues relacions siguin compatibles. Diem que dues relacions *T* i *S* són **relacions compatibles** si: 

- Tenen el mateix grau.
- Es pot establir una bijecció entre els atributs de *T* i els atributs de *S* que fa correspondre a cada atribut  $A_i$  de *T* un atribut  $A_j$  de *S*, de manera que es compleix que  $\text{domini}(A_i) \subseteq \text{domini}(A_j)$ .

Exemple de relacions compatibles

Les relacions *EMPLEATS\_ADM* i *EMPLEATS\_PROD* tenen totes dues grau 5. Podem establir la bijecció següent entre els seus atributs:

- A *DNI* d'*EMPLEATS\_ADM* li correspon *DNIemp* d'*EMPLEATS\_PROD*.
- A *nom* d'*EMPLEATS\_ADM* li correspon *nomemp* d'*EMPLEATS\_PROD*.
- A *cognom* d'*EMPLEATS\_ADM* li correspon *cognomemp* d'*EMPLEATS\_PROD*.
- A *edificidesp* d'*EMPLEATS\_ADM* li correspon *edificidesp* d'*EMPLEATS\_PROD*.
- A *númerodesp* d'*EMPLEATS\_ADM* li correspon *númerodesp* d'*EMPLEATS\_PROD*.

A més, suposarem que els dominis dels seus atributs s'han declarat de manera que es compleixi que el domini de cada atribut d'*EMPLEATS\_ADM* sigui el mateix que el domini del seu atribut corresponent a *EMPLEATS\_PROD*.

De tot això, podem arribar a la conclusió que *EMPLEATS\_ADM* i *EMPLEATS\_PROD* són relacions compatibles.

A continuació passarem a definir els atributs i l'extensió de la relació resultant d'una unió.

**Els atributs de l'esquema de la relació resultant de  $T \cup S$  coincideixen amb els atributs de l'esquema de la relació *T*.**

**L'extensió de la relació resultant de  $T \cup S$  és el conjunt de tuples que pertanyen a l'extensió de *T*, que pertanyen a l'extensió de *S* o que pertanyen a l'extensió d'ambdues relacions.**

#### No-repetició de tuples

Fixeu-vos que en cas que una mateixa tupla estigui a les dues relacions que s'uneixen, el resultat de la unió no la tindrà repetida. El resultat de la unió és una nova relació i no pot tenir repeticions de tuples.


Exemple d'unió

Si volem obtenir una relació *R* que tingui tots els empleats de l'empresa de l'exemple anterior, farem la unió de les relacions *EMPLEATS\_ADM* i *EMPLEATS\_PROD* de la manera següent:

$$R := \text{EMPLEATS\_ADM} \cup \text{EMPLEATS\_PROD}.$$

Aleshores la relació *R* resultant serà la reflectida en la taula de la pàgina següent:

R				
<i>DNI</i>	<i>nom</i>	<i>cognom</i>	<i>edificidesp</i>	<i>númerodesp</i>
40.444.255	Joan	Garcia	Marina	120
33.567.711	Marta	Roca	Marina	120
55.898.425	Carles	Bonmatí	Diagonal	120
77.232.144	Elena	Pla	Marina	230
21.335.245	Jordi	Soler	NULL	NULL
88.999.210	Pere	González	NULL	NULL

El fet que els atributs de la relació resultant coincideixin amb els atributs de la relació que figura en primer lloc a la unió és una convenció; teòricament, també hauria estat possible convenir que coincidissin amb els de la relació que figura en segon lloc. 

### 5.1.2. Intersecció

La intersecció és una operació que, a partir de dues relacions, obté una nova relació formada per les tuples que pertanyen a totes dues relacions de partida.

La intersecció és una operació binària i la intersecció de dues relacions  $T$  i  $S$  s'indica  $T \cap S$ .

La intersecció de les relacions *EMPLEATS\_ADM* i *EMPLEATS\_PROD* obté una nova relació que inclou els empleats que són alhora d'administració i de producció; s'indicaria com  $EMPLEATS\_ADM \cap EMPLEATS\_PROD$ .

La intersecció, com la unió, només es pot aplicar a relacions que tinguin tuples similars. Per a poder fer la intersecció de dues relacions cal, doncs, que les relacions siguin compatibles.

A continuació definirem els atributs i l'extensió de la relació resultant d'una intersecció.

**Els atributs de l'esquema de la relació resultant de  $T \cap S$  coincideixen amb els atributs de l'esquema de la relació  $T$ .**

**L'extensió de la relació resultant de  $T \cap S$  és el conjunt de tuples que pertanyen a l'extensió d'ambdues relacions.**

### Exemple d'intersecció

Si volem obtenir una relació  $R$  que tingui tots els empleats de l'empresa de l'exemple que treballen tant a administració com a producció, farem la intersecció de les relacions  $EMPLEATS\_ADM$  i  $EMPLEATS\_PROD$  de la manera següent:

$$R := EMPLEATS\_ADM \cap EMPLEATS\_PROD.$$

Aleshores la relació  $R$  resultant serà:

R				
DNI	nom	cognom	edificidesp	númerodesp
33.567.711	Marta	Roca	Marina	120

Observeu que s'ha pres la convenció que els atributs de la relació resultant coincideixin amb els atributs de la relació que figura en primer lloc.

### 5.1.3. Diferència

La diferència és una operació que, a partir de dues relacions, obté una nova relació formada per totes les tuples que són a la primera relació i, en canvi, no són a la segona. La diferència és una operació binària i la diferència entre les relacions  $T$  i  $S$  s'indica com  $T \setminus S$ .

La diferència  $EMPLEATS\_ADM$  menys  $EMPLEATS\_PROD$  obté una nova relació que conté els empleats d'administració que no són alhora empleats de producció i s'indicaria així:  $EMPLEATS\_ADM \setminus EMPLEATS\_PROD$ .

La diferència, com passava en la unió i la intersecció, només té sentit si s'aplica a relacions que tinguin tuples similars. Per a poder fer la diferència de dues relacions cal que les relacions siguin compatibles.

Tot seguit definim els atributs i l'extensió de la relació resultant d'una diferència.

**Els atributs de l'esquema de la relació resultant de  $T \setminus S$  coincideixen amb els atributs de l'esquema de la relació  $T$ .**

**L'extensió de la relació resultant de  $T \setminus S$  és el conjunt de tuples que pertanyen a l'extensió de  $T$ , però no a la de  $S$ .**

### Exemple de diferència

Si volem obtenir una relació  $R$  amb tots els empleats de l'empresa de l'exemple que treballen a administració, però no a producció, farem la diferència de les relacions  $EMPLEATS\_ADM$  i  $EMPLEATS\_PROD$  de la manera següent:

$$R := EMPLEATS\_ADM \setminus EMPLEATS\_PROD.$$



Aleshores la relació  $R$  resultant serà:

R				
DNI	nom	cognom	edificidesp	númerodesp
40.444.255	Joan	Garcia	Marina	120


S'ha pres la convenció que els atributs de la relació resultant coincideixin amb els atributs de la relació que figura en primer lloc.

#### 5.1.4. Producte cartesià

El producte cartesià és una operació que, a partir de dues relacions, obté una nova relació formada per totes les tuples que resulten de concatenar tuples de la primera relació amb tuples de la segona relació.

El producte cartesià és una operació binària. Essent  $T$  i  $S$  dues relacions que compleixen que els seus esquemes no tenen cap nom d'atribut comú, el producte cartesià de  $T$  i  $S$  s'indica com  $T \times S$ .

Si calculem el producte cartesià d'*EDIFICIS\_EMP* i *DESPATXOS*, obtindrem una nova relació que conté totes les concatenacions possibles de tuples d'*EDIFICIS\_EMP* amb tuples de *DESPATXOS*.

Si es vol calcular el producte cartesià de dues relacions que tenen algun nom d'atribut comú, només cal reanomenar prèviament els atributs adequats d'una de les dues relacions. 

Tot seguit definim els atributs i l'extensió de la relació resultant d'un producte cartesià.

**Els atributs de l'esquema de la relació resultant de  $T \times S$  són tots els atributs de  $T$  i tots els atributs de  $S^*$ .**

**L'extensió de la relació resultant de  $T \times S$  és el conjunt de totes les tuples de la forma  $\langle v_1, v_2, \dots, v_m, w_1, w_2, \dots, w_n \rangle$  on es compleix que  $\langle v_1, v_2, \dots, v_m \rangle$  pertany a l'extensió de  $T$  i que  $\langle w_1, w_2, \dots, w_n \rangle$  pertany a l'extensió de  $S$ .**

\* Recordeu que  $T$  i  $S$  no tenen cap nom d'atribut comú.

Exemple de producte cartesià

El producte cartesià de les relacions *DESPATXOS* i *EDIFICIS\_EMP* de l'exemple es pot fer com s'indica (cal reanomenar atributs prèviament):


$EDIFICIS(nomedifici, supmitjadesp) \bowtie EDIFICIS\_EMP(edifici, supmitjadesp).$

$R \bowtie EDIFICIS \times DESPATXOS.$

Aleshores, la relació  $R$  resultant serà:

R				
<i>nomedifici</i>	<i>supermitjadesp</i>	<i>edifici</i>	<i>número</i>	<i>superfície</i>
Marina	15	Marina	120	10
Marina	15	Marina	230	20
Marina	15	Diagonal	120	10
Marina	15	Diagonal	440	10
Diagonal	10	Marina	120	10
Diagonal	10	Marina	230	20
Diagonal	10	Diagonal	120	10
Diagonal	10	Diagonal	440	10

Convé assenyalar que el producte cartesià és una operació que rarament s'usa de manera explícita, perquè el resultat que dona no sol ser útil per a resoldre les consultes habituals.

Tot i això, el producte cartesià s'inclou en l'àlgebra relacional perquè és una operació primitiva, a partir de la qual es defineix una altra operació de l'àlgebra, la combinació, que es fa servir molt freqüentment. 

## 5.2. Operacions específicament relacionals

Les operacions específicament relacionals són la **selecció**, la **projecció** i la **combinació**.

### 5.2.1. Selecció

Podem veure la selecció com una operació que serveix per a escollir algunes tuples d'una relació i eliminar-ne la resta. Més concretament, la selecció és una operació que, a partir d'una relació, obté una nova relació formada per totes les tuples de la relació de partida que compleixen una condició de selecció especificada.

La selecció és una operació unària. Essent  $C$  una condició de selecció, la selecció de  $T$  amb la condició  $C$  s'indica com  $T(C)$ .

Per a obtenir una relació que tingui tots els despatxos de l'edifici Marina que tenen més de 12 metres quadrats, podem aplicar una selecció a la relació *DESPATXOS* amb una condició de selecció que sigui *edifici* = Marina i *superfície* > 12; s'indicaria *DESPATXOS*(*edifici* = Marina i *superfície* > 12).

En general, la condició de selecció  $C$  és formada per una o més clàusules de la forma:

$$A_i \theta v,$$

o bé:

$$A_i \theta A_j,$$

on  $A_i$  i  $A_j$  són atributs de la relació  $T$ ,  $\theta$  és un operador de comparació\* i  $v$  és un valor. A més, es compleix que:

\* És a dir,  $=$ ,  $>$ ,  $<$ ,  $\neq$ ,  $\#$ ,  $>=$ ,  $<=$ , o  $\$$ .

- A les clàusules de la forma  $A_i \theta v$ ,  $v$  és un valor del domini de  $A_i$ .
- A les clàusules de la forma  $A_i \theta A_j$ ,  $A_i$  i  $A_j$  tenen el mateix domini.

Les clàusules que formen una condició de selecció es connecten amb els operadors booleans següents: “i” ( $\wedge$ ) i “o” ( $\vee$ ).

Tot seguit definim els atributs i l'extensió de la relació resultant d'una selecció.

**Els atributs de l'esquema de la relació resultant de  $T(C)$  coincideixen amb els atributs de l'esquema de la relació  $T$ .**

**L'extensió de la relació resultant de  $T(C)$  és el conjunt de tuples que pertanyen a l'extensió de  $T$  i que satisfan la condició de selecció  $C$ . Una tupla  $t$  satisfà una condició de selecció  $C$  si, després de substituir cada atribut que hi ha a  $C$  pel seu valor a  $t$ , la condició  $C$  s'avalua al valor cert.**

Exemple de selecció

Si volem obtenir una relació  $R$  amb els despatxos de la base de dades de l'exemple que són a l'edifici Marina i que tenen una superfície de més de 12 metres quadrats, farem la selecció següent:

$$R := \text{DESPATXOS}(\text{edifici} = \text{Marina} \wedge \text{superfície} > 12).$$

La relació  $R$  resultant serà:

R		
<i>edifici</i>	<i>número</i>	<i>superfície</i>
Marina	230	20

### 5.2.2. Projectió

Podem veure la projectió com una operació que serveix per a escollir alguns atributs d'una relació i eliminar-ne la resta. Més concretament, la

projecció és una operació que, a partir d'una relació, obté una nova relació formada per totes les (sub)tuples de la relació de partida que resulten d'eliminar uns atributs especificats.

La projecció és una operació unària. Essent  $\{A_i, A_j, \dots, A_k\}$  un subconjunt dels atributs de l'esquema de la relació  $T$ , la projecció de  $T$  sobre  $\{A_i, A_j, \dots, A_k\}$  s'indica com  $T[A_i, A_j, \dots, A_k]$ .

Per a obtenir una relació que tingui només els atributs *nom* i *cognom* dels empleats d'administració, podem fer una projecció a la relació *EMPLEATS\_ADM* sobre els atributs *nom* i *cognom*. S'indicaria de la manera següent: *EMPLEATS\_ADM[nom, cognom]*.

A continuació definirem els atributs i l'extensió de la relació resultant d'una projecció.

Els atributs de l'esquema de la relació resultant de  $T[A_i, A_j, \dots, A_k]$  són els atributs  $\{A_i, A_j, \dots, A_k\}$ .

L'extensió de la relació resultant de  $T[A_i, A_j, \dots, A_k]$  és el conjunt de totes les tuples de la forma  $\langle t.A_i, t.A_j, \dots, t.A_k \rangle$  on es compleix que  $t$  és una tupla de l'extensió de  $T$  i on  $t.A_p$  denota el valor per a l'atribut  $A_p$  de la tupla  $t$ .

Eliminació de les tuples repetides

Fixeu-vos que la projecció elimina implícitament totes les tuples repetides. El resultat d'una projecció és una relació vàlida i no pot tenir repeticions de tuples.

Exemple de projecció

Si volem obtenir una relació  $R$  amb el nom i el cognom de tots els empleats d'administració de la base de dades de l'exemple, farem la projecció següent:

$R := \text{EMPLEATS\_ADM}[\text{nom}, \text{cognom}]$ .

Aleshores, la relació  $R$  resultant serà:

R	
<i>nom</i>	<i>cognom</i>
Joan	Garcia
Marta	Roca

### 5.2.3. Combinació

La combinació és una operació que, a partir de dues relacions, obté una nova relació formada per totes les tuples que resulten de concatenar tuples de la primera relació amb tuples de la segona relació i que compleixen una condició de combinació especificada.

La combinació és una operació binària. Essent  $T$  i  $S$  dues relacions els esquemes de les quals no tenen cap nom d'atribut comú, i essent  $B$  una condició de combinació, la combinació de  $T$  i  $S$  segons la condició  $B$  s'indica  $T[B]S$ .

Per a obtenir una relació que tingui les dades de cadascun dels empleats d'administració juntament amb les dades del despatx on treballa, podem fer una combinació de les relacions *EMPLEATS\_ADM* i *DESPATXOS*, on la condició de combinació indiqui el següent: *edificidesp*  $\leq$  *edifici* i *númerodesp*  $\leq$  *número*. La condició de combinació fa que el resultat només combini les dades d'un empleat amb les dades d'un despatx si l'*edificidesp* i el *númerodesp* de l'empleat són iguals que l'*edifici* i el *número* del despatx, respectivament. És a dir, la condició fa que les dades d'un empleat es combinin amb les dades del despatx on ell treballa, però no amb dades d'altres despatxos.

La combinació de l'exemple anterior s'indicaria de la manera següent:

*EMPLEATS\_ADM*[*edificidesp*  $\leq$  *edifici*, *númerodesp*  $\leq$  *número*]*DESPATXOS*.

Si es vol fer la combinació de dues relacions que tenen algun nom d'atribut comú, només cal reanomenar prèviament els atributs repetits d'una de les dues.

En general, la **condició  $B$**  d'una combinació  $T[B]S$  és formada per una o més comparacions de la forma

$$A_i \theta A_j,$$

on  $A_i$  és un atribut de la relació  $T$ ,  $A_j$  és un atribut de la relació  $S$ ,  $\theta$  és un operador de comparació ( $\leq$ ,  $\geq$ ,  $=$ ,  $<$ ,  $>$ ,  $\neq$ ,  $\in$ ,  $\notin$ ), i es compleix que  $A_i$  i  $A_j$  tenen el mateix domini. Les comparacions d'una condició de combinació se separen per comes.

Tot seguit definim els atributs i l'extensió de la relació resultant d'una combinació.

**Els atributs de l'esquema de la relació resultant de  $T[B]S$**  són tots els atributs de  $T$  i tots els atributs de  $S^*$ .

**L'extensió de la relació resultant de  $T[B]S$**  és el conjunt de tuples que pertanyen a l'extensió del producte cartesià  $T \times S$  i que satisfan totes les comparacions que formen la condició de combinació  $B$ . Una tupla  $t$  satisfà una comparació si, després de substituir cada atribut que figura a la comparació pel seu valor a  $t$ , la comparació s'avalua al valor cert.

\* Recordeu que  $T$  i  $S$  no tenen cap nom d'atribut comú.

#### Exemple de combinació

Suposem que es desitja trobar les dades dels despatxos que tenen una superfície més gran o igual que la superfície mitjana dels despatxos de l'edifici on estan situats. La combinació següent ens proporcionarà les dades d'aquests despatxos juntament amb les dades del seu edifici (observeu que cal reanomenar prèviament els atributs):

*EDIFICIS*(*nomedifici*, *supmitjadesp*)  $\leq$  *EDIFICIS\_EMP*(*edifici*, *supmitjadesp*),

*R*  $\leq$  *EDIFICIS*[*nomedifici*  $\leq$  *edifici*, *supmitjadesp*  $\#$  *superficie*]*DESPATXOS*.

Aleshores, la relació  $R$  resultant serà:

R				
<i>nomedifici</i>	<i>supmitjadesp</i>	<i>edifici</i>	<i>número</i>	<i>superfície</i>
Marina	15	Marina	230	20
Diagonal	10	Diagonal	120	10
Diagonal	10	Diagonal	440	10

Suposem ara que per a obtenir les dades de cadascun dels empleats d'administració, juntament amb les dades del despatx on treballen, utilitzem la combinació següent:

$R : \bowtie \text{EMPLEATS\_ADM}[\text{edificidesp} \bowtie \text{edifici}, \text{númerodesp} \bowtie \text{número}] \text{DESPATXOS}.$

La relació  $R$  resultant serà:

R							
<i>DNI</i>	<i>nom</i>	<i>cognom</i>	<i>edificidesp</i>	<i>númerodesp</i>	<i>edifici</i>	<i>número</i>	<i>superfície</i>
40.444.255	Joan	Garcia	Marina	120	Marina	120	10
33.567.711	Marta	Roca	Marina	120	Marina	120	10

La relació  $R$  combina les dades de cada empleat amb les dades del seu despatx.

De vegades, la combinació rep el nom de  **$\theta$ -combinació** i quan totes les comparacions de la condició de combinació tenen l'operador " $=$ " s'anomena **equi-combinació**.

Segons això, la combinació del darrer exemple és una equicombinació.

Observeu que el resultat d'una equicombinació sempre inclou un o més parells d'atributs que tenen valors idèntics a totes les tuples.

A l'exemple anterior, els valors d'*edificidesp* coincideixen amb els d'*edifici* i els valors de *númerodesp* coincideixen amb els de *número*.

Com que un de cada parell d'atributs és superflu, s'ha establert una variant de combinació anomenada *combinació natural* per tal d'eliminar-los.

La **combinació natural** de dues relacions  $T$  i  $S$  es denota com  $T * S$  i consisteix bàsicament en una equicombinació seguida de l'eliminació dels atributs superflus on, a més, es considera per defecte que la condició de combinació iguala tots els parells d'atributs que tenen el mateix nom a  $T$  i a  $S$ .

Observeu que, a diferència de l'equicombinació, la combinació natural s'aplica a relacions que tenen noms d'atributs comuns.

## Exemple de combinació natural

Si fem:

$$R : \bowtie EDIFICIS\_EMP * DESPATXOS,$$

es considera que la condició de combinació és *edifici*  $\bowtie$  *edifici* perquè *edifici* és l'únic nom d'atribut que figura tant a l'esquema d'*EDIFICIS\_EMP* com a l'esquema de *DESPATXOS*. El resultat d'aquesta combinació natural és:

R			
<i>edifici</i>	<i>supmitjadesp</i>	<i>número</i>	<i>superfície</i>
Marina	15	120	10
Marina	15	230	20
Diagonal	10	120	10
Diagonal	10	440	10

Fixeu-vos que s'ha eliminat un dels atributs de nom *edifici*.

De vegades, abans de la combinació natural, cal aplicar l'operació *reanomenar* per a fer coincidir els noms dels atributs que ens interessa que s'igualin.

## Exemple de combinació natural amb reanomenament

Per exemple, si volem obtenir les dades de cadascun dels empleats d'administració juntament amb les dades del despatx on treballen, però sense repetir valors d'atributs superflus, farem la combinació natural següent, que requereix un reanomenament previ:

$$D(\textit{edificidesp}, \textit{númerodesp}, \textit{superfície}) : \bowtie \textit{DESPATXOS}(\textit{edifici}, \textit{número}, \textit{superfície}),$$

$$R : \bowtie \textit{EMPLEATS\_ADM} * D.$$

Llavors, la relació *R* resultant serà:

R					
<i>DNI</i>	<i>nom</i>	<i>cognom</i>	<i>edificidesp</i>	<i>númerodesp</i>	<i>superfície</i>
40.444.255	Joan	Garcia	Marina	120	10
33.567.711	Marta	Roca	Marina	120	10

## 5.3. Seqüències d'operacions de l'àlgebra relacional

En molts casos, per a formular una consulta en àlgebra relacional cal utilitzar diverses operacions, que s'apliquen en un cert ordre. Per a fer-ho hi ha dues possibilitats:

- 1) Utilitzar una única expressió de l'àlgebra que inclogui totes les operacions amb els parèntesis necessaris per a indicar l'ordre d'aplicació.

2) Descompondre l'expressió en diversos passos on cada pas apliqui una sola operació i obtingui una relació intermèdia que es pugui fer servir en els passos subseqüents.

Exemple d'utilització de seqüències d'operacions

Per a obtenir el nom i el cognom dels empleats tant d'administració com de producció, és necessari fer una unió d'*EMPLEATS\_ADM* i *EMPLEATS\_PROD* i després fer una projecció sobre els atributs *nom* i *cognom*. L'operació es pot expressar de les maneres següents:

a) Es pot utilitzar una sola expressió:

$$R : \bowtie (EMPLEATS\_ADM \cup EMPLEATS\_PROD)[nom, cognom].$$

b) O bé podem expressar-ho en dos passos:

- $EMPS : \bowtie EMPLEATS\_ADM \cup EMPLEATS\_PROD;$
- $R : \pi EMP[nom, cognom].$

En els casos en què una consulta requereix efectuar moltes operacions, resulta més senzilla la segona alternativa perquè evita expressions complexes. !

Altres exemples de consultes formulades amb seqüències d'operacions

Vegem alguns exemples més de consultes a la base de dades formulades amb seqüències d'operacions de l'àlgebra relacional.

1) Per a obtenir el nom de l'edifici i el número dels despatxos situats en edificis on la superfície mitjana dels despatxos és més gran que 12, podem utilitzar la seqüència d'operacions següent:

- $A : \sigma EDIFICIS\_EMP(supmitjadep > 12);$
- $B : \sigma DESPATXOS * A;$
- $R : \pi B[edifici, número].$

2) Suposem ara que es desitja obtenir el nom i el cognom de tots els empleats (tant d'administració com de producció) que estan assignats al despatx 120 de l'edifici Marina. En aquest cas, podem fer servir la seqüència següent:

- $A : \bowtie EMPLEATS\_ADM \cup EMPLEATS\_PROD;$
- $B : \sigma A(edificidesp = Marina \wedge númerodesp = 120);$
- $R : \pi B[nom, cognom].$

3) Si volem consultar el nom de l'edifici i el número dels despatxos que no tenen assignat cap empleat d'administració, podem utilitzar aquesta seqüència:

- $A : \pi DESPATXOS[edifici, número];$
- $B : \pi EMPLEATS\_ADM[edificidesp, númerodesp];$
- $R : \sigma A \not\supset B.$

4) Per a obtenir el DNI, el nom i el cognom de tots els empleats d'administració que tenen despatx, juntament amb la superfície del seu despatx, podem fer el següent:

- $A[DNI, nom, cognom, edifici, número] : \bowtie EMPLEATS\_ADM[DNI, nom, cognom, edificidesp, númerodesp];$
- $B : \sigma A * DESPATXOS;$
- $R : \pi B[DNI, nom, cognom, superfície].$

!  
Recordeu que la base de dades que s'utilitza en els exemples s'ha descrit a la introducció de l'apartat 5 d'aquest mòdul didàctic.



## 5.4. Extensions: combinacions externes

Per a finalitzar el tema de l'àlgebra relacional, analitzarem algunes extensions útils de la combinació.

Les combinacions que s'han descrit obtenen les tuples del producte cartesià de dues relacions que satisfan una condició de combinació. Les tuples d'una de les dues relacions que no tenen a l'altra relació una tupla com a mínim, amb la qual, un cop concatenades, satisfacin la condició de combinació, no apareixen al resultat de la combinació, i podríem dir que les seves dades es perden.

Les combinacions s'han explicat al subapartat 5.3.3 d'aquest mòdul didàctic.

Per exemple, si fem la combinació natural següent (amb un reanomenament previ):

$$D(\text{edificidesp}, \text{númerodesp}, \text{superfície}) : \bowtie \text{DESPATXOS}(\text{edifici}, \text{número}, \text{superfície}),$$

$$R : \bowtie \text{EMPLEATS\_PROD} * D.$$

Com que és una combinació natural, es considera que la condició de combinació és *edifici*  $\bowtie$  *edifici* i *número*  $\bowtie$  *número*, i la relació *R* resultant serà:

R					
<i>DNltemp</i>	<i>nomemp</i>	<i>cognomemp</i>	<i>edificidesp</i>	<i>númerodesp</i>	<i>superfície</i>
33.567.711	Marta	Roca	Marina	120	10
55.898.425	Carles	Bonmatí	Diagonal	120	10
77.232.144	Elena	Pla	Marina	230	20

Noteu que en aquesta relació *R* no hi ha els empleats de producció que no tenen despatx assignat (amb valors nuls a *edificidesp* i *númerodesp*) i tampoc no hi ha els despatxos que no tenen cap empleat de producció perquè no compleixen la condició de combinació.

Convé destacar que les tuples que tenen un valor nul per a algun dels atributs que figuren a la condició de combinació sempre es perden, perquè en aquests casos la condició de combinació sempre s'avalua al valor fals.


En alguns casos, pot interessar fer combinacions de les dades de dues relacions sense que hi hagi pèrdua de dades de les relacions de partida. Aleshores, s'utilitzen les combinacions externes.

Les **combinacions externes** entre dues relacions *T* i *S* consisteixen en variants de combinació que conserven en el resultat totes les tuples de *T*, de *S* o d'ambdues relacions. Poden ser dels tipus següents:


**1) La combinació externa esquerra** entre dues relacions *T* i *S*, que denotem com  $T[C]_E S$ , conserva en el resultat totes les tuples de la relació *T*.

**2) La combinació externa dreta** entre dues relacions *T* i *S*, que denotem com  $T[C]_D S$ , conserva en el resultat totes les tuples de la relació *S*.

3) Finalment, la **combinació externa plena** entre dues relacions  $T$  i  $S$ , que denotem com  $T[C]_p S$ , conserva en el resultat totes les tuples de  $T$  i totes les tuples de  $S$ .

Aquestes extensions s'apliquen també al cas de la combinació natural entre dues relacions,  $T * S$ , concretament: 

- a) La combinació natural externa esquerra entre dues relacions  $T$  i  $S$ , que s'indica com  $T *_E S$ , conserva en el resultat totes les tuples de la relació  $T$ .
- b) La combinació natural externa dreta entre dues relacions  $T$  i  $S$ , que s'indica com  $T *_D S$ , conserva en el resultat totes les tuples de la relació  $S$ .
- c) Finalment, la combinació natural externa plena entre dues relacions  $T$  i  $S$ , que s'indica com  $T *_p S$ , conserva en el resultat totes les tuples de  $T$  i totes les tuples de  $S$ .

Les tuples d'una relació  $T$  que es conserven en el resultat  $R$  d'una combinació externa amb una altra relació  $S$ , tot i que no satisfan la condició de combinació, tenen valors nuls al resultat  $R$  per a tots els atributs que provenen de la relació  $S$ . 

Exemples de combinacions naturals externes

- 1) Si fem la combinació natural dreta següent (amb un reanomenament previ):

$D(\text{edificidesp}, \text{númerodesp}, \text{superficie}) : \bowtie \text{DESPATXOS}(\text{edifici}, \text{número}, \text{superficie}),$

$R : \bowtie \text{EMPLEATS\_PROD} *_D D,$

la relació  $R$  resultant serà:

R					
<i>DNlemp</i>	<i>nomemp</i>	<i>cognomemp</i>	<i>edificidesp</i>	<i>númerodesp</i>	<i>superficie</i>
33.567.711	Marta	Roca	Marina	120	10
55.898.425	Carles	Bonmatí	Diagonal	120	10
77.232.144	Elena	Pla	Marina	230	20
NULL	NULL	NULL	Diagonal	440	10

Ara obtenim tots els despatxos a la relació resultant tant si tenen un empleat de producció assignat com si no. Fixeu-vos que els atributs *DNI*, *nom* i *cognom* per als despatxos que no tenen empleat reben valors nuls.

- 2) Si fem la combinació natural esquerra següent (amb un reanomenament previ):

$D(\text{edificidesp}, \text{númerodesp}, \text{superficie}) : \bowtie \text{DESPATXOS}(\text{edifici}, \text{número}, \text{superficie}),$

$R : \bowtie \text{EMPLEATS\_PROD} *_E D,$

aleshores la relació  $R$  resultant serà:

R					
<i>DNlemp</i>	<i>nomemp</i>	<i>cognomemp</i>	<i>edificidesp</i>	<i>númerodesp</i>	<i>superficie</i>
33.567.711	Marta	Roca	Marina	120	10
55.898.425	Carles	Bonmatí	Diagonal	120	10
77.232.144	Elena	Pla	Marina	230	20
21.335.245	Jordi	Soler	NULL	NULL	NULL
88.999.210	Pere	González	NULL	NULL	NULL

Aquesta combinació externa ens permet obtenir a la relació resultant tots els empleats de producció, tant si tenen despatx com si no. Observeu que l'atribut *superficie* per als empleats que no tenen despatx conté un valor nul.

3) Finalment, si fem la combinació natural plena següent (amb un reanomenament previ):

$$D(\textit{edificidesp}, \textit{númerodesp}, \textit{superficie}) : \bowtie \text{DESPATXOS}(\textit{edifici}, \textit{número}, \textit{superficie}),$$

$$R : \bowtie \text{EMPLEATS\_PROD} *_p D,$$

llavors la relació  $R$  resultant serà:

R					
<i>DNlemp</i>	<i>nomemp</i>	<i>cognomemp</i>	<i>edificidesp</i>	<i>númerodesp</i>	<i>superficie</i>
33.567.711	Marta	Roca	Marina	120	10
55.898.425	Carles	Bonmatí	Diagonal	120	10
77.232.144	Elena	Pla	Marina	230	20
21.335.245	Jordi	Soler	NULL	NULL	NULL
88.999.210	Pere	González	NULL	NULL	NULL
NULL	NULL	NULL	Diagonal	440	10

En aquest cas, a la relació resultant obtenim tots els empleats de producció i també tots els despatxos.

## Resum

En aquest mòdul didàctic hem presentat els **conceptes fonamentals del model relacional de dades** i, a continuació, hem explicat les **operacions de l'àlgebra relacional**:

1) Els aspectes més rellevants del model relacional que hem descrit són els següents:

a) Pel que fa a l'**estructura de les dades**:

- Consisteix en un conjunt de relacions.
- Una relació permet emmagatzemar dades relacionades entre si.
- La clau primària d'una relació permet identificar les seves dades.
- Les claus foranes de les relacions permeten referenciar claus primàries i, així, establir connexions entre les dades de les relacions.

b) Pel que fa a la **integritat de les dades**:

- La regla d'integritat d'unicitat i d'entitat de la clau primària: les claus primàries no poden contenir valors repetits ni valors nuls.
- La regla d'integritat referencial: els valors de les claus foranes han d'existir a la clau primària referenciada o bé han de ser valors nuls.
- La regla d'integritat de domini: els valors no nuls d'un atribut han de pertànyer al domini de l'atribut i els operadors que és possible aplicar sobre els valors depenen dels dominis d'aquests valors.

2) L'àlgebra relacional proporciona un conjunt d'operacions per a manipular relacions. Aquestes operacions es poden classificar de la manera següent:

a) Operacions conjuntistes: unió, intersecció, diferència i producte cartesià.

b) Operacions específicament relacionals: selecció, projecció i combinació.

Les operacions de l'àlgebra relacional poden formar seqüències que permeten resoldre consultes complexes.

## Exercicis d'autoavaluació

1. Donada la relació que correspon a la representació tabular següent:

Figura 4

Relació **DESPATXOS**

<i>edifici</i>	<i>números</i>	<i>superfícies</i>
<i>edifici</i>	<i>número</i>	<i>superfície</i>
Marina	140	10
Diagonal	250	15
Diagonal	110	NULL

- Digueu quin conjunt d'atributs té.
- Digueu quin domini té cadascun dels seus atributs.
- Escriviu totes les diverses maneres de denotar el seu esquema de relació.
- Escolliu una de les maneres de denotar el seu esquema de relació i utilitzeu-la per a dibuixar el conjunt de tuples corresponent a la seva extensió.

2. Digueu quines són totes les superclaus de les relacions següents:

- DESPATXOS(*edifici*, *número*, *superfície*), que té com a única clau candidata la següent: *edifici*, *número*.
- EMPLEATS(*DNI*, *NSS*, *nom*, *cognom*), que té les claus candidates següents: *DNI* i *NSS*.

3. Digueu, per a cadascuna de les operacions d'actualització següents, si se'n podria acceptar l'aplicació sobre la base de dades que s'ha utilitzat en aquest mòdul:

- Inserir a *EDIFICIS\_EMP* la tupla <Nexus, 30>.
- Inserir a *DESPATXOS* la tupla <Diagonal, NULL, 15>.
- Inserir a *EMPLEATS\_ADM* la tupla <55.555.555, Maria, Puig, Diagonal, 500>.
- Modificar a *DESPATXOS* la tupla <Marina, 230, 20> per <Marina, 120, 20>.
- Esborrar a *EMPLEATS\_PROD* la tupla <88.999.210, Pere, González, NULL, NULL>.
- Modificar a *EMPLEATS\_ADM* la tupla <40.444.255, Joan, Garcia, Marina, 120> per <33.567.711, Joan, Garcia, Marina, 120>.
- Esborrar a *EDIFICIS\_EMP* la tupla <Marina, 15> si per a la clau forana *edifici* de *DESPATXOS* s'ha seleccionat la política de restricció en cas d'esborrat.
- Esborrar a *EDIFICIS\_EMP* la tupla <Marina, 15> si per a la clau forana *edifici* de *DESPATXOS* s'ha seleccionat la política d'actualització en cascada en cas d'esborrat.

4. Escriu seqüències d'operacions de l'àlgebra relacional que resolguin les consultes següents a la base de dades que hem fet servir en aquest mòdul:

- Obtenir els despatxos amb una superfície més gran que 15. Concretament, es vol saber el nom de l'edifici, el número i la superfície d'aquests despatxos, juntament amb la superfície mitjana dels despatxos de l'edifici on estan situats.
- Obtenir el nom de l'edifici i el número dels despatxos que no tenen assignat cap empleat (ni de producció, ni d'administració).
- Obtenir el nom i el cognom dels empleats (tant d'administració com de producció) que no tenen despatx.
- Obtenir el nom i el cognom de tots els empleats (tant d'administració com de producció) que tenen despatx assignat, juntament amb la superfície del seu despatx i la superfície mitjana dels despatxos de l'edifici a què pertany el seu despatx.
- Obtenir els despatxos amb una superfície més gran que la superfície del despatx Diagonal, 120. Concretament, es vol saber el nom de l'edifici i el número d'aquests despatxos.
- Obtenir tots els despatxos de l'empresa (tant si tenen empleats com si no) juntament amb els empleats que tenen assignats (en cas que en tinguin). Concretament, es vol conèixer el nom de l'edifici, el número de despatx i el DNI de l'empleat.

5. Sigui  $R$  la relació que resulta de la intersecció de les relacions  $T$  i  $S$ , és a dir,  $R : 5 T > S$ . Escriu una seqüència d'operacions de l'àlgebra relacional que inclogui només operacions primitives i que obtingui com a resultat  $R$ .

La base de dades s'ha descrit a la introducció de l'apartat 5 d'aquest mòdul didàctic.

La base de dades s'ha descrit a la introducció de l'apartat 5 d'aquest mòdul didàctic.

6. Siguin les relacions d'esquema  $T(A, B, C)$  i  $S(D, E, F)$ , i sigui  $R$  la relació que resulta de la combinació següent:

$$R := T \bowtie B \bowtie D, C \bowtie E \bowtie S.$$

Escriviu una seqüència d'operacions de l'àlgebra relacional que inclogui només operacions primitives i que obtingui com a resultat  $R$ .

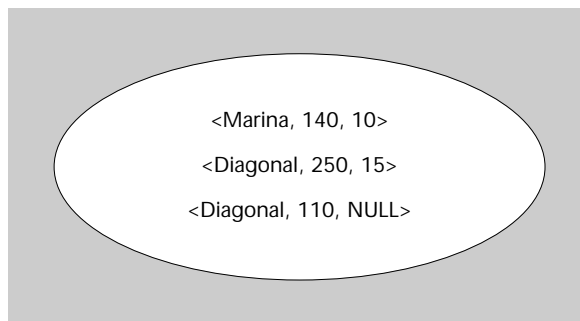
## Solucionari

### Exercicis d'autoavaluació

1.

- a) La relació representada té el conjunt d'atributs següent: *edifici*, *número*, *superfície*.  
b) Els dominis són domini(*edifici*)  $\ni$  *edificis*, domini(*número*)  $\ni$  *números* i domini(*superfície*)  $\ni$  *sup*.  
c) Les maneres de denotar l'esquema de relació són:  
• DESPATXOS(*edifici*, *número*, *superfície*),  
• DESPATXOS(*edifici*, *superfície*, *número*),  
• DESPATXOS(*número*, *edifici*, *superfície*),  
• DESPATXOS(*número*, *superfície*, *edifici*),  
• DESPATXOS(*superfície*, *edifici*, *número*),  
• DESPATXOS(*superfície*, *número*, *edifici*),  
que corresponen a les ordenacions possibles dels seus atributs.  
d) Escollirem la manera següent de denotar l'esquema de relació:  
DESPATXOS(*edifici*, *número*, *superfície*).  
Llavors el conjunt de tuples de la seva extensió serà:

Figura 5



2. Les superclaus de les relacions corresponents són:

- a) {*edifici*, *número*} i {*edifici*, *número*, *superfície*}.  
b) {*DNI*}, {*NSS*}, {*DNI*, *NSS*}, {*DNI*, *nom*}, {*DNI*, *cognom*}, {*NSS*, *nom*}, {*NSS*, *cognom*}, {*DNI*, *nom*, *cognom*}, {*NSS*, *nom*, *cognom*}, {*DNI*, *NSS*, *nom*}, {*DNI*, *NSS*, *cognom*} i {*DNI*, *NSS*, *nom*, *cognom*}.  
3.  
a) S'accepta.  
b) Es rebutja perquè viola la regla d'integritat d'entitat de la clau primària.  
c) Es rebutja perquè viola la regla d'integritat referencial.  
d) Es rebutja perquè viola la regla d'integritat d'unicitat de la clau primària.  
e) S'accepta.  
f) Es rebutja perquè viola la regla d'integritat d'unicitat de la clau primària.  
g) Es rebutja perquè viola la regla d'integritat referencial.  
h) S'accepta i s'esborra l'edifici Marina i tots els seus despatxos.

4.

- a) Podem utilitzar la seqüència d'operacions següent:  
•  $A : \ni$  DESPATXOS(*superfície*  $\ni$  15),  
•  $R : \ni$   $A * EDIFICIS\_EMP$ .  
b) Podem utilitzar la seqüència d'operacions següent:  
•  $A : \ni$  DESPATXOS[*edifici*, *número*],  
•  $B : \ni$  EMPLATS\_ADM  $<$  EMPLATS\_PROD,  
•  $C : \ni$  B[*edificidesp*, *númerodesp*],  
•  $R : \ni$   $A \supseteq C$ .  
c) Podem utilitzar la seqüència d'operacions següent:  
•  $A : \ni$  EMPLATS\_ADM  $<$  EMPLATS\_PROD,  
•  $B : \ni$  A(*edificidesp*  $\ni$  NULL i *númerodesp*  $\ni$  NULL),  
•  $R : \ni$  B[*nom*, *cognom*].  
d) Podem utilitzar la seqüència d'operacions següent:  
•  $A : \ni$  EMPLATS\_ADM  $<$  EMPLATS\_PROD,  
• B(*DNI*, *nom*, *cognom*, *edifici*, *número*):  $\ni$  A(*DNI*, *nom*, *cognom*, *edificidesp*, *númerodesp*),  
•  $C : \ni$  B \* DESPATXOS,  
•  $D : \ni$  C \* EDIFICIS\_EMP,  
•  $R : \ni$  D[*nom*, *cognom*, *superfície*, *supmitjadesp*].  
e) Podem utilitzar la seqüència d'operacions següent:  
•  $A : \ni$  DESPATXOS(*edifici*  $\ni$  Diagonal i *número*  $\ni$  120),  
• B(*Ed*, *Num*, *Sup*) :  $\ni$  A(*edifici*, *número*, *superfície*),

- $C : \subseteq \text{DESPATXOS}[superficie \rightarrow Sup]B$ ,
  - $R : \subseteq C[edifici, número]$ .
- f) Podem utilitzar la seqüència d'operacions següent:
- $A : \subseteq \text{EMPLEATS\_ADM} < \text{EMPLEATS\_PROD}$ ,
  - $B(DNI, nom, cognom, edifici, número) : \subseteq A(DNI, nom, cognom, edificidesp, númeroesp)$ ,
  - $C : \subseteq \text{DESPATXOS} *_{\text{E}} B$ ,
  - $R : \subseteq C[edifici, número, DNI]$ .

5. La seqüència següent:

- $A : \subseteq T \supseteq S$ ,
- $R : \subseteq T \supseteq A$ ,

només inclou operacions primitives, atès que la diferència és primitiva, i obté el mateix resultat que  $R : \subseteq T \supseteq S$ .

6. La seqüència següent:

- $A : \subseteq T \times S$ ,
- $R : \subseteq A(B \subseteq D \wedge C \subseteq E)$ ,

que només inclou operacions primitives (un producte cartesià i una selecció), obté el mateix resultat que  $R : \subseteq T[B \subseteq D, C \subseteq E]S$ .

## Glossari

actualització

Fet de reflectir els canvis que es produeixen a la realitat a les relacions d'una base de dades.

actualització en cascada per al cas d'esborrat

Política de manteniment de la integritat referencial que consisteix en esborrar una tupla  $t$  que té una clau primària referenciada i esborrar també totes les tuples que referencien  $t$ .

actualització en cascada per al cas de modificació

Política de manteniment de la integritat referencial que consisteix en modificar atributs de la clau primària d'una tupla  $t$  que té una clau primària referenciada, i modificar de la mateixa manera totes les tuples que referencien la tupla  $t$ .

anul·lació per al cas d'esborrat

Política de manteniment de la integritat referencial que consisteix en esborrar una tupla  $t$  que té una clau referenciada i, a més, modificar totes les tuples que referencien  $t$  de manera que els atributs de la clau forana corresponent prenguin valors nuls.

anul·lació per al cas de modificació

Política de manteniment de la integritat referencial que consisteix en modificar atributs de la clau primària d'una tupla  $t$  que té una clau referenciada i, a més, modificar totes les tuples que referencien  $t$  de manera que els atributs de la clau forana corresponent prenguin valors nuls.

atribut (en el context del model relacional)

Nom del paper que exerceix un domini en un esquema de relació.

cardinalitat d'una relació

Nombre de tuples que pertanyen a la seva extensió.

clau alternativa d'una relació

Clau candidata de la relació que no s'ha escollit com a clau primària.

clau candidata d'una relació

Superclau  $C$  de la relació que compleix que cap subconjunt propi de  $C$  no és superclau.

clau forana d'una relació  $R$

Subconjunt dels atributs de l'esquema de la relació,  $CF$ , tal que existeix una relació  $S$  ( $S$  no ha de ser necessàriament diferent de  $R$ ) que té per clau primària  $CP$ , i es compleix que, per a tota tupla  $t$  de l'extensió de  $R$ , els valors per a  $CF$  de  $t$  són o bé valors nuls, o bé valors que coincideixen amb els valors per a  $CP$  d'alguna tupla  $s$  de  $S$ .

clau primària d'una relació

Clau candidata de la relació que s'ha escollit per a identificar les tuples de la relació.

combinació

Operació de l'àlgebra relacional que, a partir de dues relacions, obté una nova relació formada per totes les tuples que resulten de concatenar tuples de la primera relació amb tuples de la segona relació i que compleixen una condició de combinació especificada.



**combinació externa**

Extensió de combinació entre dues relacions,  $T$  i  $S$ , que conserva en el resultat totes les tuples de  $T$ , de  $S$  o d'ambdues relacions.

**combinació natural**

Variant de combinació que consisteix bàsicament en una equicombinació seguida de l'eliminació dels atributs superflus.

**consulta**

Obtenció de dades deduïbles a partir de les relacions que conté la base de dades.

**diferència**

Operació de l'àlgebra relacional que, a partir de dues relacions, obté una nova relació formada per totes les tuples que són a la primera relació i, en canvi, no són a la segona.

**domini (en el context del model relacional)**

Conjunt de valors atòmics.

**equicombinació**

Combinació en la qual totes les comparacions de la condició tenen l'operador " $=$ ".

**esborrat**

Fet d'esborrar una o més tuples d'una relació.

**esquema de relació**

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

**extensió d'una relació d'esquema  $R(A_1, A_2, \dots, A_n)$** 

Conjunt de tuples  $t_i$  ( $i \in 1, 2, \dots, m$ ) on cada tupla  $t_i$  és un conjunt de parells  $t_i \in \{ \langle A_1: v_{i1} \rangle, \langle A_2: v_{i2} \rangle, \dots, \langle A_n: v_{in} \rangle \}$  i, per a cada parell  $\langle A_j: v_{ij} \rangle$ , es compleix que  $v_{ij}$  és un valor de domini( $A_j$ ) o bé un valor nul.

**grau d'una relació**

Nombre d'atributs que pertanyen al seu esquema.

**inserció**

Fet d'afegir una o més tuples a una relació.

**integritat**

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

**intersecció**

Operació de l'àlgebra relacional que, a partir de dues relacions, obté una nova relació formada per les tuples que són a totes dues relacions de partida.

**llenguatge basat en el càlcul relacional:**

Llenguatge que proporciona un tipus de formulació de consultes fonamentat en el càlcul de predicats de la lògica matemàtica.

**llenguatge basat en l'àlgebra relacional**

Llenguatge que proporciona un tipus de formulació de consultes inspirat en la teoria de conjunts.

**modificació**

Fet d'alterar els valors que tenen una o més tuples d'una relació per a un o més dels seus atributs.

**producte cartesià**

Operació de l'àlgebra relacional que, a partir de dues relacions, obté una nova relació formada per totes les tuples que resulten de concatenar tuples de la primera relació amb tuples de la segona relació.

**projecció**

Operació de l'àlgebra relacional que, a partir d'una relació, obté una nova relació formada per totes les (sub)tuples de la relació de partida que resulten d'eliminar uns atributs especificats.

**reanomenar**

Operació auxiliar de l'àlgebra relacional que permet canviar els noms que figuren a l'esquema d'una relació.

regla d'integritat de domini

Regla que estableix que un valor no nul d'un atribut  $A_i$  ha de pertànyer al domini de l'atribut  $A_i$ , i que els operadors que és possible aplicar sobre els valors depenen dels dominis d'aquests valors.

regla d'integritat d'entitat de la clau primària

Regla que estableix que si el conjunt d'atributs  $CP$  és la clau primària d'una relació  $R$ , l'extensió de  $R$  no pot tenir en cap moment cap tupla que tingui un valor nul per a algun dels atributs de  $CP$ .

regla d'integritat de model

Condicions generals que han de complir-se a tota base de dades d'un model determinat.

regla d'integritat d'unicitat de la clau primària

Regla que estableix que si el conjunt d'atributs  $CP$  és la clau primària d'una relació  $R$ , l'extensió de  $R$  no pot tenir en cap moment dues tuples amb la mateixa combinació de valors per als atributs de  $CP$ .

regla d'integritat referencial

Regla que estableix que si el conjunt d'atributs  $CF$  és una clau forana d'una relació  $R$  que referencia una relació  $S$  (no necessàriament diferent de  $R$ ) que té per clau primària  $CP$ , aleshores, per a tota tupla  $t$  de l'extensió de  $R$ , els valors per a  $CF$  de  $t$  són o bé valors nuls o bé valors que coincideixen amb els valors per a  $CP$  d'alguna tupla  $s$  de  $S$ .

relació

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

restricció en cas de modificació

Política de manteniment de la integritat referencial que consisteix a no permetre modificar cap atribut de la clau primària d'una tupla si té una clau primària referenciada.

restricció en cas d'esborrat

Política de manteniment de la integritat referencial que consisteix a no permetre esborrar una tupla si té una clau primària referenciada.

restriccions d'integritat d'usuari

Condicions específiques que han de complir-se en una base de dades concreta.

selecció

Operació de l'àlgebra relacional que, a partir d'una relació, obté una nova relació formada per totes les tuples de la relació de partida que compleixen una condició de selecció especificada.

superclau d'una relació d'esquema  $R(A_1, A_2, \dots, A_n)$

Subconjunt dels atributs de l'esquema tal que no hi pot haver dues tuples a l'extensió de la relació que tinguin la mateixa combinació de valors per als atributs del subconjunt.

tancament relacional

Propietat de totes les operacions de l'àlgebra relacional segons la qual tant els seus operands com el seu resultat són relacions.

unió

Operació de l'àlgebra relacional que, a partir de dues relacions, obté una nova relació formada per totes les tuples que són en alguna de les relacions de partida.

## Bibliografia

### Bibliografia bàsica

Date, C.J. (2001). *Introducción a los sistemas de bases de datos* (7a ed.). Prentice-Hall.

Elmasri, R.; Navathe, S.B. (2000). *Sistemas de bases de datos. Conceptos fundamentales* (3a ed.). Madrid: Addison-Wesley Iberoamericana.