

BASES DE DADES CARTESIANES

Carles Franquesa i Niubò

Departament de Matemàtica Aplicada
i Anàlisi

BASES DE DADES CARTESIANES

Carles Franquesa i Niubò

Departament de Matemàtica Aplicada
i Anàlisi

Índex

Pròleg , per Josep Maria Bañeres	11
Preàmbul	13
1. Teoria de conjunts	17
1.1. Definicions i nomenclatura	17
1.1.1. Abstracció	19
1.1.2. Descripció de conjunts	20
Per enumeració dels seus elements	20
A partir de propietats dels seus elements	21
1.2. Operacions entre conjunts	22
1.2.1. Unió	23
1.2.2. Intersecció	24
1.2.3. Diferència	26
1.2.4. Producte cartesià	27
1.3. Lògica de predicats	29
1.3.1. Càlcul de predicats	29
Operació de negació lògica	30
Operació lògica disjuntiva	30
Operació lògica conjuntiva	31
Lleis d'Augustus De Morgan	32
1.3.2. Relació amb la teoria de conjunts	33
2. Anàlisi del projecte	35
2.1. Sistemes gestors de bases de dades	35
2.1.1. Arquitectura client-servidor	36
2.2. Principi d'independència de les dades	37
2.2.1. Arquitectura a tres nivells	38
2.3. Immersió en el domini	40
2.4. Definició de requeriments	41
2.4.1. Informació estadística	41
2.5. Trajectòria de l'error	42
2.6. Exemple per a un club esportiu	43
2.6.1. Requeriments per al club esportiu	43
3. Disseny de bases de dades	47
3.1. Models de dades	50
3.2. Domini del problema	51
3.3. Màximes de qualitat d'un disseny	51
3.3.1. Prohibir la redundància	52
3.3.2. Prohibir nuls estructurals	53
3.3.3. Unificar conceptes	53

4. Model entitat relació	55
4.1. Entitats	56
4.2. Atributs	57
4.3. Atributs clau	60
4.4. Atributs multivalorats	61
4.5. Atributs compostos	63
4.6. Atributs calculats	64
4.7. Relacions	66
4.7.1. Dependències d'existència	67
4.7.2. Cardinalitat d'una relació binària	68
4.8. Relacions 1:1	69
4.9. Relacions 1:N	70
4.9.1. Entitats de suport a la interfície	73
4.9.2. Entitats febles	75
4.9.3. Classificació de les relacions 1:N	77
4.10. Relacions M:N	78
4.10.1. Atributs en relacions M:N	82
4.11. Autorelacions	83
4.11.1. Autorelacions 1:1	83
4.11.2. Autorelacions 1:N	85
4.11.3. Autorelacions M:N	86
4.12. Càlcul automàtic de les cardinalitats	88
4.13. Relacions ternàries	89
4.13.1. Cardinalitats en les relacions ternàries	91
4.13.2. Relació ternària amb tres relacions binàries	92
4.14. Model entitat relació estès	93
4.14.1. Especialització i generalització d'entitats	94
Especialitzacions o generalitzacions completes	96
Especialitzacions o generalitzacions disjunctes	96
4.14.2. Agregació d'entitats	97
5. Àlgebra relacional	99
5.1. Àlgebres	100
5.2. Relacions	100
5.2.1. Relació	101
Tuples	102
Càlcul relacional per tuples	104
Dominis	104
Definició formal de relació	105
Compatibilitat entre relacions	107
5.2.2. Visió cartesiana d'una relació	108
5.3. Model relacional d'una base de dades	111
5.3.1. Atributs estructurals i atributs descriptius	112
5.3.2. Claus primàries	112
5.3.3. Claus foranes	114
5.3.4. Transformació del model ER al model relacional	116
5.3.5. Diagrama d'esquemes d'una base de dades	123
5.4. Operacions bàsiques amb relacions	126
5.4.1. Operació de selecció	127
5.4.2. Operació de projecció	129
Composició d'operacions	131
5.4.3. Producte cartesià de relacions	132
5.4.4. Unió de relacions	135
5.4.5. Diferència de relacions	136
5.4.6. Renomenament de relacions	137

5.5.	Operacions addicionals	141
5.5.1.	Intersecció de relacions	141
5.5.2.	Reunió natural	143
	Reunió interna	144
	Reunions externes	146
5.5.3.	Divisió de relacions	148
5.5.4.	Assignació de relacions	149
5.6.	Àlgebra relacional estesa	151
5.6.1.	Projecció generalitzada	151
5.6.2.	Valors nuls	152
	Lògica de tres valors	153
5.6.3.	Funcions d'agregació	153
6.	Llenguatge estructurat de consultes	159
6.1.	Entorn de treball	160
6.1.1.	Familiarització amb el PSQL	162
	Familiarització amb el <code>psql</code>	162
	Familiarització amb l'SQL	163
6.2.	Llenguatge de definició de dades	164
6.2.1.	Multiconjunts	164
6.2.2.	Taules	164
6.2.3.	Metadades	165
6.2.4.	Dominis dels atributs	166
	Tipus primitius	166
	Tipus alfanumèrics	167
	Tipus temporals	168
	Tipus autoincrementals	169
6.3.	Construcció de la base de dades	170
6.3.1.	Espai de treball	171
6.3.2.	Esript principal	171
6.3.3.	Creació dels dominis	173
6.3.4.	Creació de les taules	175
	Integritat referencial	178
	Claus candidates	183
6.4.	Llenguatge de manipulació de dades	192
	Esripts d'inserció	192
6.4.1.	Consultes de lectura	193
	Estructura fonamental	193
	Dades	194
	Clàusula <code>SELECT</code>	196
	Clàusula <code>FROM</code>	198
	Clàusula <code>WHERE</code>	204
6.4.2.	Funcions d'agregació	207
	De tots els tipus d'atributs	207
	D'atributs numèrics	209
	Clàusula <code>GROUP BY</code>	210
	Clàusula <code>HAVING</code>	212
6.4.3.	Clàusula d'ordenació	213
	Clàusula <code>ORDER BY</code>	213
	Consulta de lectura amb totes les clàusules	215
6.4.4.	Vistes	216
6.4.5.	Consultes d'actualització	218
	Clàusula <code>INSERT INTO</code>	218
	Clàusula <code>UPDATE SET</code>	221
	Clàusula <code>DELETE FROM</code>	222

6.4.6.	Operacions amb conjunts	224
	Clàusula UNION	224
	Clàusula EXCEPT	228
	Clàusula INTERSECT	230
6.4.7.	Operacions de reunió de la clàusula FROM	231
	Clàusula NATURAL INNER JOIN	232
	Clàusula NATURAL LEFT OUTER JOIN	234
	Clàusula NATURAL RIGHT OUTER JOIN	235
	Clàusula NATURAL FULL OUTER JOIN	236
	Clàusula INNER JOIN USING	237
	Clàusula INNER JOIN ON	238
6.4.8.	Valors nuls	239
	Predicat IS NULL	239
	Predicat IS UNKNOWN	240
6.4.9.	Consultes internes	241
	En l'argument de la clàusula SELECT	242
	En l'argument de la clàusula FROM	243
6.4.10.	Consultes internes en la clàusula WHERE	244
	Clàusula IN	246
	Clàusules SOME i ALL	247
	Clàusula EXISTS	250
7.	El Sistema gestor de bases de dades PostgreSQL	255
7.1.	Documentació de PostgreSQL	256
7.2.	Estructura del projecte	258
7.3.	Usuaris i seguretat en la base de dades	260
7.3.1.	Permisos	261
	Aportacions al projecte del club esportiu	265
7.3.2.	Privilegis	266
7.3.3.	Concurrència i transaccions	271
7.4.	Components lògics d'una base de dades	276
7.4.1.	Components de dades	276
	Atributs indexats	276
7.5.	Components de control	279
7.5.1.	Implementació de l'aplicació per al club esportiu	279
	Implementació de funcions genèriques	280
	Implementació de funcions específiques	281
7.6.	Funcions	281
7.6.1.	Capçalera	282
7.6.2.	Cos	283
7.6.3.	Entrada i sortida de dades amb PL/pgSQL	285
7.6.4.	Consultes d'actualització en PL/pgSQL	287
7.6.5.	Consultes de lectura en PL/pgSQL	290
	Consultes que retornen un sol valor	291
	Clàusula PERFORM	295
	Consultes que retornen registres	296
7.7.	Disparadors	299
7.7.1.	Funcions de tipus TRIGGER	300
	Capçalera de les funcions TRIGGER	300
	Cos de les funcions TRIGGER	300
	Valors de retorn dels disparadors	301
7.7.2.	Sintaxi de declaració d'un disparador	301
7.8.	Aplicacions client	303
7.8.1.	Llenguatge amfitrió i SQL incrustat	303
	Connexió	304

Consultes de lectura	304
Consultes d'actualització	304
7.9. Administració de la base de dades	305
7.9.1. Parada i iniciació del sistema	305
MS-DOS	305
Linux	306
El programa <code>pg_ctl</code>	308
7.9.2. Manteniment i còpies de seguretat	308
7.9.3. Configuració d'un servidor PostgreSQL	310
Arxiu de paràmetres de configuració	311
Arxiu d'autenticació basada en el host	315
Arxiu de mapeig d'usuaris	317
A. Estil	319
A.1. Model entitat relació	319
A.2. Model relacional	320
A.3. Llenguatge procedimental	320
A.4. Versions dels sistemes utilitzats	320
B. Codis ASCII	321
C. Dades per a l'exemple del club esportiu	323
C.1. Taula comarca	323
C.2. Taula ciutat	324
C.3. Taula persona	325
C.4. Taula telèfons	326
C.5. Taula coneix	326
C.6. Taula soci	327
C.7. Taula treballador	327
C.8. Taula esport	328
C.9. Taula fa	329
C.10. Taula nòmines	330
Bibliografia	331
Índex de caixes	331
Índex de diagrames	334
Índex de figures	334
Índex de models	336

Pròleg

Quan el Dr. Franquesa em va demanar que escrivís el pròleg del seu llibre Bases de dades cartesianes em vaig sentir honorat però també alhora preocupat per saber si podria estar a l'alçada del que se'm requeria. Suposo que va pensar en mi tot recordant els anys que vàrem compartir a l'assignatura d'Introducció a les Bases de Dades. Varen ser els darrers anys abans de la meua jubilació i en guardo un gran record que crec que els que alumnes de llavors compartiran amb nosaltres.

Fins a data d'avui, no he tingut coneixement d'altres llibres que versin sobre aquests continguts en la nostra llengua catalana, i tan sols per això ja ens hauríem de felicitar els qui estimem les bases de dades i al mateix temps aquesta llengua. Però no és només això.

A mesura que he anat cosint un camí de més de quaranta anys de docència en estudis superiors, he anat adquirint la convicció que allò més motivador i estimulant ha estat la vocació. Per als professors vocacionals no hi ha gratificació més gran que observar la magnitud dels projectes en què ara treballen alumnes que un dia van arribar amb la incertesa als ulls, i al mateix temps amb una ingenuïtat voraç, disposats a alimentar-se d'allò se'ls indica, allò que cal preveure que és el que transcendirà en les seves activitats professionals en un futur més proper del que sembla. Per això cal entusiasme, il·lusió, loquacitat i coherència en els continguts. Aquest llibre és un exponent de tots aquests factors. Més que un llibre de teoria, sembla un discurs. Una conversa.

Les bases de dades han evolucionat força des del seu començament allà als anys 60 del segle passat. Llavors, el mateix hardware limitava el que es podia fer i les primeres implementacions van ser per resoldre problemes com l'escandall de peces en la fabricació amb una estructura, fictícia per poc realista, de tipus jeràrquic: una peça, o bé un conjunt de peces, mai depenia de més d'una peça en un nivell superior. Al cap de poc de començar a utilitzar aquestes bases de dades jeràrquiques, el fonament teòric d'altres solucions s'estava cuinant per part de veritables pioners com Edgar F. Codd i, més tard, Christopher J. Date. El conjunt de solucions trobades va suposar l'aparició teòrica de les bases de dades relacionals. Les limitacions del hardware van impedir-ne el desenvolupament i en van posposar la implementació gairebé quinze anys.

Avui en dia moltes de les bases de dades ja són del tipus relacional, tot i que ja s'apunten noves solucions agosarades que encara han de demostrar la seva utilitat real i que tenen uns fonaments matemàtics no tan robustos i per descomptat molt menys rigorosos que els del model relacional. En molts casos es tracta de permetre nous tipus de dades elementals, com bases de dades d'objectes, o de bases de dades de molta grandària, normalment de tipus multimèdia, que se suporten més en la quantitat de recursos materials de què es disposa que en la integritat de les dades.

En aquest llibre, el Dr. Franquesa fa un estudi molt complet de les bases de dades relacionals, i en particular proposa la solució real amb la implementació d'una d'elles.

El llibre es proposa assentar primerament una base teòrica inicial de tipus matemàtic seguida del que en diu "Anàlisi del projecte", on es recull l'ambient en el qual la solució informàtica es trobarà finalment circumscrita.

Segueix un capítol de disseny previ, ja a l'estudi nucleic del model entitat-relació, per entrar al capítol següent a l'àlgebra relacional, on s'estudia quins mecanismes es poden utilitzar, i com s'utilitzen, per obtenir resultats de la base de dades preparada, dissenyada, com a solució al problema abordat pel projecte. En aquest capítol, el cinquè, l'autor proposa uns gràfics magnífics i molt aclaridors per a representar múltiples tuples.

Entra després a la descripció, central en diversos aspectes, del llenguatge estructurat de consultes, que ens ha de servir primer per a descriure la pròpia base de dades i després per arribar a utilitzar les dades que contindrà. Naturalment, el fet de tenir les dades servirà de poc si no podem consultar-les, esborrar-les o afegir-ne quan ens convingui. Per fi, caldrà que tot el que s'ha vist i estudiat es posi en context en una implementació en particular, escollint per a fer-ho el SGBD PostgreSQL, que és no solament una de les millors implementacions, sinó també una de les més fàcils de trobar.

Em cal dir que, quan impartíem l'assignatura d'Introducció a les bases de dades, m'hauria agradat tenir aquest llibre com a base per a les explicacions als alumnes. Al mateix temps, els serviria a ells com a referència actual i futura sobre les bases de dades.

He dit que m'agradaria haver tingut aquest llibre i no he estat exacte: el que m'hauria agradat hauria estat escriure'!

Josep Maria Bañeres

*Professor responsable de Bases de Dades,
ja retirat, de la Universitat de Barcelona.*

Cabrils, octubre de 2014

Preàmbul

A l'hora de formular continguts teòrics, els humans mostrem predilecció per suportar-los sobre la lògica matemàtica. La mateixa rel etimològica del terme *lògica* s'utilitza llavors com a sufix per indicar estudi, manera d'entendre. Això és així perquè un contingut teòric exposa un model d'alguna faceta de la realitat, i per poder-se desenvolupar necessita una coherència que troba prou ben fonamentada en la teoria de conjunts.

Les bases de dades relacionals, que ens disposem a explicar, no difereixen d'aquesta aproximació, i així, estructuren unes normes que finalment troben connexió amb la praxi en la implementació que se'n fa en màquines computadores.

Però bé, no avancem històries. Observem què tenim al davant, d'entrada. Assumint la força brutal que tenen dos conceptes tan importants com l'espai i el temps, quedí clar que el que aquí tractarem és l'espai. L'espai són els objectes, el temps són els accions. Orientar el discurs cap a les accions o els procediments convertiria aquest text en un llibre d'algorísmia. I aquí no es tracta d'això. Aquí abordem els objectes, o sigui els espais. De fet, és la part senzilla, ja que sembla clar que els substantius són més fàcils de comprendre que els verbs. En la llengua catalana en tenim un bon grapat de proves. El diccionari, [2], [6], permet dir píxel, però no pixelar. Una cosa fa olor, no olora. I encara que sigui permès besar o petonejar, en català acostumem a fer-nos petons. En definitiva, és ben clar que som proclius a suportar la semàntica en els substantius. La prova més clara que l'univers dels substantius és més senzill que el dels verbs és que els substantius tan sols els classifiquem en femenins o masculins, singulars i plurals, o sigui gènere i nombre. En canvi, per als verbs tenim les conjugacions verbals amb els seus passats, presents i futurs, a part de les tres formes impersonals. Així doncs, tenim sort, les dades viuen al món dels substantius.

De fet, per estructurar la realitat, cal començar per les dades. I més filosòficament encara, el temps sorgeix a partir de la repetició en les referències als objectes. El temps estableix diferències entre una cosa i ella mateixa. Els instants de creació poden servir per identificar gairebé qualsevol objecte. O sigui, que cal anar amb compte a l'hora d'incorporar el temps en les dades que tenen referències temporals.

Per comprendre amb profunditat on cal situar la feina que es proposa, veiem primer una partició de les teories tal com les entenem. Convinguem que una teoria pot entendre's com un conjunt de veritats inqüestionables més un conjunt de regles que treballen aquestes veritats per construir-ne de noves. Mirat des d'aquest aspecte, qualsevol cosa és una teoria. Una bicicleta, i les aplicacions que se li puguin donar, per exemple, constitueixen una teoria. Tot això entra en escena per fer saber que respecte a les teories hi ha dues classificacions en tres grups. Per un costat, es pot classificar una teoria com a incompleta, categòrica o contradictòria. I per altra banda com a teoria oberta, sòlida o inconsistent. I segons es miri des del punt de vista deductiu o bé empíric, s'utilitza l'una o l'altra classificació. Però bé, mirat des de la distància, sense cap rigor, tot plegat ens resulta útil per disposar d'una terminologia que ubica els continguts que s'exposen en aquest text.

Una teoria oberta, o incompleta, és aquella que sense entrar en cap contradicció postula un conjunt de veritats que després es poden enriquir amb aplicacions concretes donant pas a moltes altres teories. Qualsevol àlgebra és una teoria oberta. I de manera molt més prosaica també podem considerar qualsevol cosa: una bicicleta, un telèfon o una sabata. Tot això són teories obertes, ja que són veritats que no es contradiuen, cosa que les fa teories, i al mateix temps deixen marge per a la interpretació, cosa que les fa obertes. Podríem parlar de bicicletes vermelles o verdes, establint dues teories diferents.

Teoria categòrica és aquella que pretén classificar el conjunt de totes les coses dites. És molt pretensiosa, ja que per qualsevol predicat formulable, o sigui, per qualsevol pregunta que es pugui respondre sí o no, l'anàlisi matemàtica ens pretén respondre. L'anàlisi matemàtica és l'única teoria categòrica que coneixem. Per això és una eina tan estimada en totes les disciplines.

Finalment, en les teories inconsistents es produeixen espais de confusió en els quals un mateix predicat resulta cert i fals alhora. Són teories que moren aviat. No tenen gaire utilitat.

De tot plegat és fàcil desprendre'n una analogia amb la vida d'una persona al llarg de les tres edats.

Doncs bé, el desenvolupament d'una base de dades per satisfer un conjunt de requeriments és una teoria oberta. O sigui, incompleta. Això vol dir que qualsevol objectiu té més d'una aproximació correcta. Ningú ens podrà assegurar que la solució implementada sigui òptima.

De fet, a partir del seu disseny, ningú pot demostrar que una base de dades donarà resposta a totes les sol·licituds que se li puguin formular. Es tracta d'un territori relliscós. Per això, afloxem els nostres objectius. Una pretensió més assequible és la de detectar que alguna part de l'estructura de la base grinyola. Si hi ha errors en el disseny, és possible que els detectem. I quan siguem incapços de detectar-ne, llavors podrem suposar que el disseny és correcte.

Des que van començar a existir les bases de dades fins ara hi ha hagut un ús massiu d'aquestes tecnologies. La mateixa expressió, *base de dades*, és una de les primeres expressions que es va popularitzar de l'àmbit de la computació. Això ha provocat que una gran quantitat de desenvolupadors d'aplicacions procurassin incorporar les bases de dades a les seves feines. L'ús de les bases de dades ha estat tan massiu que el coneixement de per què serveixen s'ha diluït, de manera que a dia d'avui, una gran quantitat d'aplicacions que es vanten de connectar-se a bases de dades, no en treuen cap partit que no poguessin treure d'un sistema pla de fitxers.

Més greu encara que el malbaratament de recursos d'espai i temps que això suposa a les aplicacions, s'ha perdut la major part dels avantatges que un sistema gestor de bases de dades ofereix. És un problema greu, ja que els desenvolupadors de codi esmercen hores de feina en accions que el gestor resoldria molt millor del que poden fer ells. La responsabilitat d'aquest fet és sens dubte de tots plegats, de la societat. Queda malament parlar d'una aplicació que no utilitza una base de dades, i per tant, qualsevol aplicació s'assegura un prestigi usant-les.

Cal tenir això en compte, ja que tot plegat ens ve a explicar que ens trobem en un àmbit on hi ha un gran intrusisme, i per tant, cal desconfiar molt de les tendències.

Els arxius per realitzar els exercicis dels últims capítols contenen tant la implementació de la base de dades d'exemple com les dades que hi ha a l'Apèndix C. Es poden descarregar de:

<https://drive.google.com/folderview?id=0B1-YyXS4zdgrQWhGOGg2WUExd28&usp=sharing>

Capítol 1

Teoria de conjunts

La teoria de conjunts és profusament acceptada com la teoria més important de totes les teories. Dit això amb certa precaució per no fer enfadar els puristes, és segurament la primera, la teoria més abstracta.

Però bé, deixant-nos de superlatius, que no fan més que competir, i apropant-nos al tema de manera més pragmàtica, la teoria de conjunts és en l'essència del coneixement analític, i per tant científic, i per tant humà.

En aquest capítol es fa un repàs dels conceptes bàsics d'aquesta teoria, ja que són qüestions preliminars per als capítols posteriors. Es presenta la definició de les operacions que tracten amb conjunts. Tot i que d'alguna manera hi hagi passatges en què tot això pot semblar infantil, o fins i tot *naïf*, convé assentar sòlidament aquestes nocions bàsiques per no caure en malentesos. En la segona meitat del capítol es veu la lògica de predicats, cosina germana de la teoria de conjunts encara que més emparentada amb la deducció natural. Tot plegat és una mateixa cosa mirada des de diferents punts de vista.

1.1. Definicions i nomenclatura

Independentment del rigor amb què es podria descriure aquesta teoria, cal retenir tres conceptes.

En síntesi, anomenem *conjunt*, C , a una col·lecció d'*elements*, e_1, e_2 i e_3 per exemple, diferents entre ells. Diem que l'element e_1 *pertany* al conjunt C , i ho escrivim $e_1 \in C$, o sigui, dient “pertany” al símbol $\in$. I també podem dir que C *conté* l'element e_1 , encara que en aquest cas no introduïm cap símbol.

Per definició, un conjunt no té elements repetits. Si volem treballar o dir alguna cosa sobre un conjunt amb elements repetits, llavors haurem de dir que ens trobem davant d'un *multiconjunt*. Però bé, en aquestes alçades del discurs aquesta definició no té més transcendència.

Conjunt, element, i el tercer concepte essencial és el concepte de *subconjunt*. Resulta gairebé intuïtiu entendre que S és un subconjunt de C , cosa que anotem $S \subseteq C$, si S és un conjunt d'elements tots els quals pertanyen a C . També diem llavors que S està inclòs en C .

O sigui, un conjunt no és el conjunt de tots els seus subconjunts possibles. I per tant, un subconjunt no pertany al conjunt en el qual s'ha definit, hi és inclòs. Una altra cosa seria parlar del conjunt de subconjunts possibles. És una cosa diferent.

La definició de subconjunt contempla els dos casos singulars en què S és el conjunt buit, que anotem $\emptyset$ i que per definició està inclòs en tots els conjunts, o l'altre extrem, en què S és igual a C . Ara bé, si volem indicar que S és un subconjunt de C tal com normalment ho imaginem, llavors diem que S és un subconjunt *propri* de C . De manera formal, utilitzant el símbol $\Leftrightarrow$ que vol dir *si i només si*,

$$S \subset C \Leftrightarrow S \subseteq C \text{ i } S \neq \emptyset \text{ i } S \neq C.$$

Caixa 1.1. *Definició de subconjunt propi.*

En aquest cas, podem dir també que S està inclòs en C *estrictament*. I ho representem sense la ratlleta, $S \subset C$.

De vegades interessa explicar que entre dos conjunts, un és subconjunt de l'altre sense la necessitat de reflectir la continència de quin és subconjunt de quin. Llavors diem que C i S són *compatibles*, i ho expressem amb el símbol $\sim$.

En la caixa 1.2 es formalitza la compatibilitat entre conjunts.

$$X \sim Y \Leftrightarrow X \subseteq Y \text{ o bé } Y \subseteq X$$

Caixa 1.2. *Definició de compatibilitat entre conjunts.*

Ergo, si són iguals, són compatibles. Altres exemples de compatibilitat entre

conjunts són els nombres enters i els nombres reals o les cadenes de caràcters i els números de telèfon.

El *cardinal* d'un conjunt és el nombre d'elements que conté. Un nombre enter no negatiu. En notació simbòlica, està clar que li podem dir n , minúscula perquè és un nombre i no un conjunt. I si volem expressar el cardinal amb referència al conjunt C , llavors utilitzem la notació de C entre barres, és a dir $|C|$. És fàcil veure que si S és un subconjunt propi de C , i si $n = |C|$ i $m = |S|$, llavors $n > m$.

Com s'ha dit més amunt, els elements d'un conjunt han de ser diferents entre ells, i això implica que han de ser identificables. Aquesta deducció resulta profundament constructiva. Els conjunts són conjunts d'elements identificables.

Des d'un punt de vista molt més filosòfic, es podria dir que hi ha alguna relació entre la identificació i el número u.

1.1.1. Abstracció

La capacitat de percebre característiques comunes en objectes o fenòmens i anomenar-ho amb una paraula és l'abstracció. Com és una cadira? Hi ha cadires d'una pota, o de tres o de quatre, amb respatller i sense. De fusta, de metall i de plàstic... en fi, hi ha tantes variacions de cadires que sembla mentida que ens entenguem quan utilitzem la paraula *cadira*. Ja ho veieu. Cada paraula ja és una abstracció.

Es diu que un concepte és més abstracte que un altre quan el primer designa un conjunt del qual el segon és subconjunt. Aquest procés mental pot arrossegar pèrdua d'informació. El títol d'un text és una abstracció, ja que sota un mateix títol poden existir diversos textos. L'abstracció sintetitza l'essència. I també una abstracció de gat és animal, i una abstracció de cadira és moble. De vegades, però, no és tan senzill. Ja es veu que com més concret és un concepte menys abstracte és. L'abstracció és la relació més vertical entre paraules. L'índex d'un llibre, que materialitza una aproximació vertical al contingut, també és una abstracció.

En la programació orientada a objectes, la idea d'abstracció hi és implementada. I no és per casualitat que tots els llenguatges que es desenvolupen sobre aquesta tecnologia concloguin en la idea d'*objecte* com a terme al màxim nivell d'abstracció. Però això, tot, en el fons, són objectes.

L'abstracció és una operació mental magníficament complicada, la resolució de la qual causa satisfacció. No és una operació calculable, ja que necessita una riquesa terminològica que és difícil estructurar. En qualsevol cas, diguem d'entrada que és una de les feines més boniques de les quals se n'ocupa una persona dissenyadora de bases de dades.