

Disseny multidimensional

Alberto Abelló Gamazo

PID_00189746

Índex

Introducció.....	5
Objectius.....	6
1. Les necessitats dels analistes i les eines OLAP.....	7
1.1. Bases de dades estadístiques	8
1.2. Fulls de càlcul	9
1.3. Eines OLAP i multidimensionalitat	10
2. Components del model multidimensional.....	19
2.1. Estructures de dades	19
2.1.1. Dimensions	20
2.1.2. Fets	23
2.2. Operacions sobre les dades	28
2.3. Restriccions d'integritat inherents al model	33
2.3.1. Unicitat i entitat de la Base	33
2.3.2. Acumulació o agregació	34
2.3.3. Transitivitat	38
3. Disseny conceptual.....	39
3.1. Una metodologia per a dissenyar una Estrella	39
3.1.1. Triar el Fet	40
3.1.2. Trobar el grànul escaient	40
3.1.3. Escollir les dimensions que s'utilitzaran en l'anàlisi	41
3.1.4. Trobar els atributs de cada dimensió	42
3.1.5. Distingir entre descriptors i jerarquies d'agregació	43
3.1.6. Decidir quines són les mesures que interessin	45
3.1.7. Definir Cel·les	46
3.1.8. Explicitar les restriccions d'integritat	47
3.1.9. Estudiar la viabilitat	48
3.2. Reconsideracions en el disseny conceptual	49
3.2.1. Dimensions amb múltiples rols	49
3.2.2. Dependències entre dimensions	50
3.2.3. Minidimensions	52
3.2.4. Disseny de dades heterogènies	52
3.2.5. Fets amb una única mesura	54
4. Disseny lògic.....	56
4.1. L'Estrella (el cas bàsic)	56
4.2. El flocc de neu	58
4.3. Conformació: compartició de dimensions	60

4.4.	Dimensions degenerades, de deixalles i ombres	62
4.5.	Generalitzacions/especialitzacions	63
4.6.	Estructures temporals	65
5.	Consultes amb SQL'99	69
5.1.	Estructura bàsica de la consulta	69
5.2.	GROUPING SETS	71
5.2.1.	ROLLUP	76
5.2.2.	CUBE	77
6.	Disseny físic	79
6.1.	Pla i tècniques bàsiques d'accés	79
6.2.	Índexs de mapes de bits	81
6.3.	Particions horitzontals	83
6.4.	Particions verticals, eines VOLAP	84
6.5.	Matrius <i>n</i> -dimensionals, eines MOLAP i HOLAP	85
6.6.	Tècniques de preagregació	88
Resum		92
Activitats		93
Exercicis d'autoavaluació		93
Solucionari		95
Glossari		98
Bibliografia		100

Introducció

No n'hi ha prou de tenir un magatzem de dades per a poder tenir tota la informació que ens cal i ser capaços d'utilitzar-la per a prendre decisions. La gran quantitat de dades i la manca d'expertesa en informàtica per part dels analistes fan que sigui imprescindible disposar d'algun tipus d'eina que en faciliti la consulta.

D'aquesta manera, a partir del magatzem de dades corporatiu, s'acostumen a dissenyar petits magatzems de dades departamentals que apropen les dades als usuaris. Aquests magatzems de dades departamentals es dissenyen utilitzant el model multidimensional, de manera que es puguin utilitzar **eines OLAP** per a consultar-los. Les eines OLAP i els magatzems de dades no s'exclouen, es complementen.

A grans trets, el model multidimensional distingeix dos tipus de dades: els fets que volem analitzar i les dimensions que utilitzem per a analitzar-los. Aquesta divisió produeix dos beneficis: d'una banda, podem utilitzar tècniques específiques d'emmagatzematge i accés de dades; i de l'altra, facilitem la comprensió de les dades de manera que els analistes són capaços de formular les seves consultes sense gairebé cap coneixement informàtic i mitjançant eines gràfiques molt intuïtives.

En aquest mòdul trobareu una definició formal del model multidimensional, i també una guia per a fer un bon disseny conceptual, lògic i físic. També veureu les noves paraules reservades que inclou l'estàndard SQL'99 per a facilitar les consultes multidimensionals.

OLAP

De l'anglès *on-line analytical processing* ('processament analític en línia').

Objectius

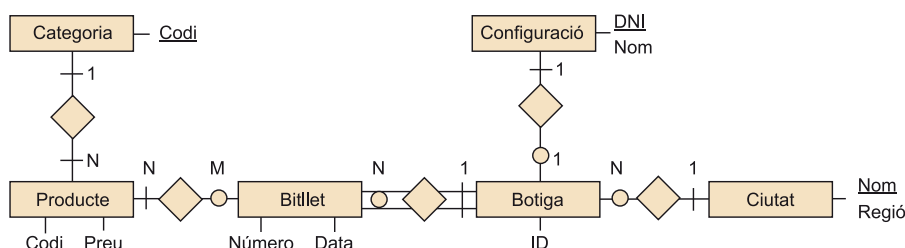
Aquest mòdul didàctic presenta el model multidimensional i alguns conceptes que hi estan associats. Amb aquest mòdul assolireu els objectius següents:

- 1.** Conèixer els components del model multidimensional (estructures de dades, operacions i restriccions d'integritat).
- 2.** Entendre què és el disseny multidimensional i els problemes de disseny que presenta (en l'àmbit conceptual, com en el lògic i físic).
- 3.** Ser capaç de dissenyar una base de dades multidimensional.
- 4.** Saber què ens ofereix l'estàndard SQL per a facilitar les consultes multidimensionals.
- 5.** Comprendre els mecanismes d'emmagatzematge i indexació associats a les eines multidimensionals (MOLAP, ROLAP, etc.).

1. Les necessitats dels analistes i les eines OLAP

Fins ara esteu acostumats a veure eines OLTP, que estan basades en la gestió de transaccions (conjunts d'operacions d'alta, baixa, modificació i consulta de dades) que ajuden en el funcionament diari del negoci. La pregunta que us hauríeu de fer en aquest moment és si aquestes eines són apropiades per a analitzar el funcionament del negoci i prendre decisions. Creieu que un directiu d'una companyia serà capaç de fer una consulta amb SQL sobre un esquema ER o simplement estarà disposat a dedicar el seu valuós temps intentant fer-la?

Figura 1



Les dificultats de consulta per als usuaris no informàtics

Un usuari no informàtic ni tan sols entendria un esquema ER tan senzill com el que teniu en la figura 1. A més, no es tracta simplement de consultar el preu d'un cert producte, sinó, donada una cadena de botigues repartides per tot l'Estat, saber com han evolucionat les vendes dels diferents productes durant el mes passat respecte al mateix mes de l'any anterior.

Com ja sabem, el model ER va ser concebut per a reduir la quantitat de dades redundants i evitar d'haver de modificar molts registres alhora en fer un únic canvi. Això s'aconsegueix a costa d'empitjorar el temps de resposta a les consultes (assumint que hi haurà un nombre elevat d'actualitzacions respecte al nombre de consultes). La pregunta que cal que ens fem ara és la següent: això serveix per als analistes? Què voldran actualitzar aquests usuaris? No res, ja que ells només volen saber com va l'empresa, no introduir-hi dades noves.

El model ER no facilita precisament la consulta de les dades. De fet, probablement a causa que barreja diferents processos de negoci, als usuaris se'ls faria difícil ni tan sols recordar l'esquema. A més, tampoc no és fàcil construir un programari que faciliti aquesta consulta.

Nota

El sistema que utilitzarem serà "cometes" per a les instàncies, *cursiva* per a termes anglesos i operacions del model multidimensional, tipus de lletra CourierNew per als elements dels esquemes, i majúscules per als noms dels elements del model multidimensional i les operacions. A més, en negreta hi haurà la primera aparició dels termes principals que introduïm.

El model ER, que resulta molt convenient en entorns en què es produeixen canvis freqüents, no és gaire adient per a entorns d'anàlisi en què el que preval és tenir respostes ràpides a les consultes i que l'esquema sigui fàcil d'entendre. El sistema ja no ha d'ajudar a vendre, comprar, produir o transportar, sinó a avaluar, comparar, pressupostar, planificar, projectar, etc.

1.1. Bases de dades estadístiques

El primer que se'ns podria acudir per a solucionar aquest problema és pensar en les bases de dades estadístiques. Aquests sistemes es fan servir per a grans estudis socioeconòmics com l'estudi del cens, la producció nacional o els patrons de consum. Permeten d'analitzar les dades des de diferents punts de vista i mostrar tot tipus de mesures estadístiques. En la taula següent podeu veure un exemple del que es podria obtenir amb una d'aquestes eines.

Figura 2

Comptador d'articles venuts		Barcelona	Tarragona	Lleida	Girona
5-1-2000	Bolígrafs	15	3	7	1
	Gomes	3	1	0	5
	Portamines	4	0	6	2

Suma d'ingressos per article		Barcelona	Tarragona	Lleida	Girona
5-1-2000	Bolígrafs	39,5	5,1	15,9	1,2
	Gomes	1,4	0,3	0	1,4
	Portamines	8,7	0	11	5,1

Cal distingir clarament les taules estadístiques de les taules relacionals. En les primeres podem canviar les files per columnes i viceversa, mentre que en les segones, no. És equivalent tenir una taula estadística d'articles per població que una de població per articles. Però cada valor numèric que hi ha en la taula estadística seria en una fila diferent d'una taula relacional. A més, aquests valors corresponen a atributs acumulats.

A les bases de dades estadístiques se'ls demana que garanteixin la confidencialitat de les dades dels individus. Conseqüentment, el seu punt fort és la seguretat (més concretament, els mecanismes de protecció d'inferència de dades) i no tant la facilitat de consulta o la presentació dels resultats, com seria

Dades estadístiques

Per a acostar-nos al concepte amb més propietat, en comptes de bases de dades estadístiques hauríem de parlar simplement d'eines que faciliten les consultes estadístiques a bases de dades.

Lectura complementària

Podeu veure exemples de taules estadístiques en *l'Anuari estadístic de Catalunya 1992-2001*. Col·lecció Estadística de Síntesi. Departament d'Economia i Finances. Institut d'Estadística de Catalunya.

Atributs acumulats

Els atributs acumulats s'obtenen com a resultat d'aplicar una funció d'agregació a atributs detallats.

d'esperar si les volguéssim utilitzar per a prendre decisions en una empresa. Acostuma a tractar-se simplement de sistemes relacionals amb un programari afegit per a millorar la seguretat i les consultes.

Un sistema relacional ja ofereix una certa flexibilitat en l'estructuració i consulta de les dades. No obstant això, obtenir la suma acumulada de vendes combinant totals i subtotals, o determinar un cert rànquing (donar els deu països amb un total de vendes més gran), és molt difícil, si no impossible, amb SQL. En el millor dels casos, caldria la intervenció d'un informàtic per a fer la consulta. L'usuari no la podria fer. A més, el model relacional no considera les **jerarquies d'agregació** (les ciutats s'agreguen en regions, les regions en estats, etc.).

El que necessitem és alguna cosa més flexible que una base de dades estadística. Després de demanar la suma de vendes per estats, ens hauria de permetre d'aïllar-ne un (per exemple, el que té la proporció de vendes més gran en relació amb la seva població) i veure'n les vendes desglossades per regions. L'usuari final, sense cap coneixement específic d'informàtica, hauria de poder navegar fàcilment per les dades.

Una base de dades relacional (estadística o no) no dona la flexibilitat i facilitat d'ús que requereix l'anàlisi en línia.

Mecanismes de protecció d'inferència

Entenem per mecanismes de protecció d'inferència tot allò que ajudi a evitar que a partir de les dades d'un cert conjunt d'individus es puguin inferir o deduir les dades corresponents a aquests; per exemple, aconseguir l'edat d'en Jordi a partir de la mitjana d'edat dels habitants de Lleida.

Navegar

En aquest context, el terme *navegar* vol dir fer un conjunt de consultes de manera que quan es vegi el resultat d'una es decideix quina serà la següent.

1.2. Fulls de càlcul

Realment, els analistes no utilitzen les bases de dades estadístiques, principalment a causa del fet que són difícils de fer servir. El problema no és que no es puguin fer servir per a tasques d'anàlisi, sinó com n'és de difícil i pesat fer-ho. El que sí que utilitzen en el seu dia a dia són els fulls de càlcul, sens cap mena de dubte molt més fàcils d'utilitzar. Cada cel·la conté una dada que podem referenciar fàcilment mitjançant les seves coordenades bidimensionals (files × columnes). Podem operar amb les dades simplement referenciant-les amb una lletra i un nombre. Si una dada canvia, no cal modificar totes les fórmules en què apareix, sinó solament canviar el valor d'una certa cel·la. Aquest funcionament facilita el que s'anomena *anàlisi what-if* ('què passa si'). És a dir, què passarà si canvio aquest valor (per exemple, un preu de venda, el nombre d'unitats produïdes, el temps de producció, etc.)? Com variarà el meu negoci?

Cel·les tridimensional

Si a més de files i columnes el full de càlcul permet de treballar amb pàgines, podem parlar d'un espai de cel·les tridimensional.

Tot i aquesta facilitat d'ús, els fulls de càlcul encara tenen algunes mancances:

- No són apropiats per a grans quantitats de dades.
- No aporten cap significat a les dades (les cel·les s'identifiquen simplement per les seves coordenades).

- La creació d'informes no és prou senzilla.
- De la mateixa manera que les bases de dades estadístiques, no faciliten l'ús de jerarquies d'agregació.

A més d'això, la posició de les dades pot determinar les operacions que es puguin fer. Si no es vol haver d'explicitar una per una tota la llista de cel·les que intervenen en una operació, aquestes han de ser consecutives, de manera que es pugui donar un rang (per exemple, `SUMA (D7 : D123)`). Per tant, hem de conèixer *a priori* amb quines dades operarem, per a poder-les col·locar en les cel·les adients. La pregunta que sorgeix en aquest punt és si sempre hi haurà alguna manera de col·locar les dades, de manera que puguem fer fàcilment totes les operacions que volem. En els casos que això no sigui possible, s'hauria de tenir un full de càlcul diferent per a cada operació o conjunt d'operacions que es pugui definir donada una posició de les dades.

Tot i que podria solucionar el problema anterior, tenir molts fulls de càlcul amb les mateixes dades generaria molta redundància, amb tot els inconvenients que, com ja sabeu, això comporta. El que ens cal és, sense perdre la facilitat d'ús, poder consultar i reestructurar les dades de manera fàcil i flexible.

Necessitem un sistema híbrid que ens proporcioni la flexibilitat i potència d'un full de càlcul, i l'estructura i facilitat de consulta d'una base de dades.

1.3. Eines OLAP i multidimensionalitat

La solució actual a les necessitats d'anàlisi de les empreses són les eines OLAP. Aquest terme va ser introduït per E.F. Codd l'any 1993, encara que ja hi havia eines informàtiques específiques per a l'anàlisi molt abans. Literalment, es refereix a la possibilitat de processar consultes en línia (en contraposició amb el processament per lots *-batch-*) amb l'objectiu d'analitzar dades. Per a entendre més bé el que són aquestes eines, podem emprar l'anomenat *test FASMI*¹. Segons aquesta definició, un sistema OLAP ha de fer el següent:

1) **FAST**: respondre la majoria de les consultes en aproximadament cinc segons (excepcionalment, podria arribar a trigar-ne vint). Això fa que s'hagin d'implementar tècniques específiques d'indexació i cerca, amb mecanismes especials d'emmagatzematge.

El temps de resposta ha de ser petit

Estudis recents demostren que, si un usuari triga més de trenta segons a obtenir el resultat de la seva petició, tendeix a pensar que el procés falla, si no és que se l'avis de la durada. En qualsevol cas, si la resposta de l'ordinador triga massa a arribar, els usuaris es distreuen i perden el fil dels seus raonaments.

Lectura recomanada

E. F. Codd; S. B. Codd; C. T. Salley (1993). *Providing OLAP (On-Line Analytical Processing) to User-Analysts: An IT Mandate*. Arbor Software, Technical Report.

⁽¹⁾De l'anglès *fast analysis of shared multidimensional information*.

2) ANALYSIS: oferir eines d'anàlisi estadística i generació d'informes sense que calgui programar res (ni tan sols mitjançant un llenguatge de quarta generació –4GL– com l'SQL). Una eina OLAP podria ajudar en l'estudi de sèries temporals, l'adjudicació de costos, el canvi de monedes, la prospecció de dades, la definició de ràtios, etc. Només amb un magatzem de dades ja podríem contestar les preguntes "què?" i "qui?". Amb una eina OLAP, a més d'això, també hauríem de poder respondre "per què?" i "què passa si...?". Les eines OLAP són un complement imprescindible dels magatzems de dades.

3) SHARED: implementar els mecanismes de seguretat (control d'accés i confidencialitat de les dades) i concurrència (encara que els analistes no vulguin escriure noves dades, sí que voldran escriure i possiblement compartir els resultats de l'anàlisi) necessaris per a compartir informació.

4) INFORMATION: ser capaç de guardar tota la informació necessària. Aquest punt té dos vessants. Primerament, el volum de dades pot arribar a ser molt gran. D'altra banda, el sistema no s'ha de limitar a contenir dades, sinó que també ha d'enregistrar quin és el seu significat: les metadades (informació = dades + metadades).

Aquestes quatre característiques són molt importants, però encara queda la cinquena, la més important de totes, la que realment distingeix les eines OLAP i a la qual està dedicada aquest mòdul didàctic: la **multidimensionalitat**.

Tota eina OLAP ha de ser multidimensional.

Com ja hem dit abans, els analistes no són informàtics. És per això que han de ser les eines les que s'hi adaptin i no a l'inrevés. Volem que els usuaris no depenguin del departament d'informàtica per a fer una simple consulta. Ara no parlem d'un entorn transaccional amb consultes predefinides que gairebé mai no canvien, sinó d'un entorn d'anàlisi amb consultes *ad hoc* que l'usuari ha de formular a mesura que té la necessitat de veure unes quantes dades. Per a facilitar-los la feina, una eina OLAP ha de presentar les dades com els analistes estan acostumats a veure-les, és a dir, en termes de **fets i dimensions**, en comptes de taules, atributs i claus foranes.

Exemple de fets i dimensions d'anàlisi

Imaginem-nos que es vol analitzar la distribució i els valors de les vendes d'una cadena de supermercats. Vendes seria el nostre fet objecte d'anàlisi. Un possible espai tetradimensional per a analitzar aquest fet estaria definit en aquest cas per les dimensions *Producte*, *Client*, *Temps* i *Població* en què s'ha produït la venda.

La multidimensionalitat

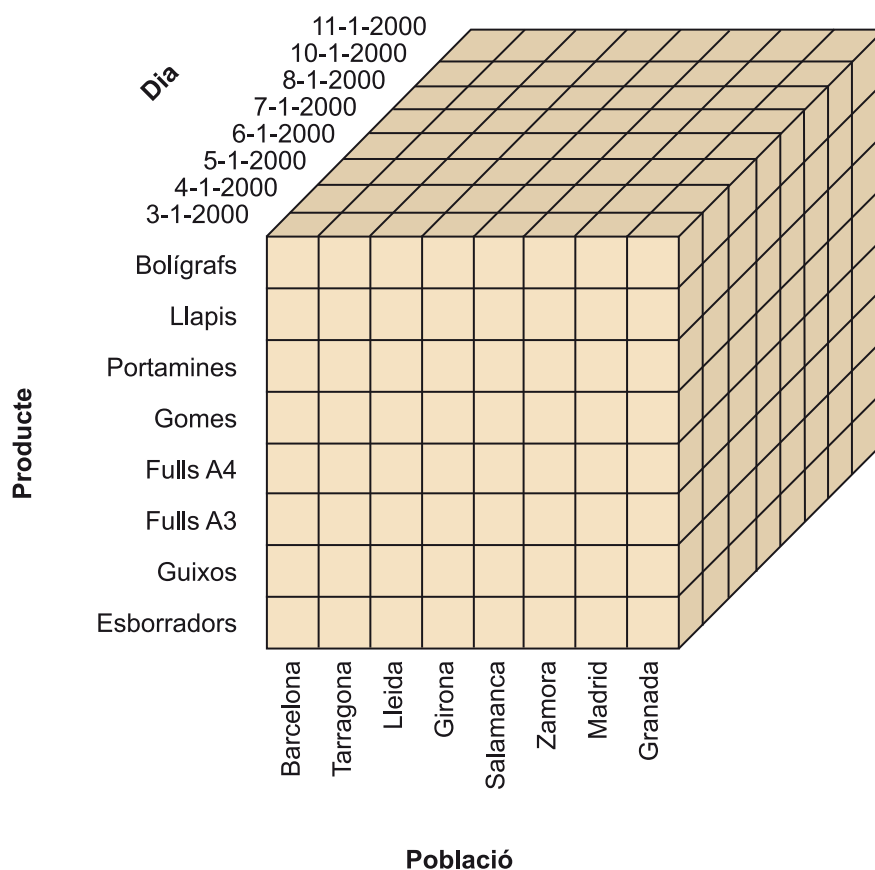
La multidimensionalitat no té els seus orígens en les bases de dades, sinó que es basa en l'àlgebra de matrius, que ha estat utilitzada per a l'anàlisi manual de dades des del segle XIX.

La multidimensionalitat es basa en la dicotomia entre dades mètriques (què volem analitzar) i dades descriptives (què, a qui, on, quan, com, etc.). Les dimensions (dades descriptives) defineixen un espai n -dimensional, conegut com a **cub**, en què col·loquem els fets (dades mètriques) que volem analitzar (en certa manera, això generalitza els fulls de càlcul, de manera que podem tenir qualsevol nombre de dimensions). De cada posició en aquest espai en direm **cel·la**. Cada cel·la correspon a un fet en concret que queda determinat per les dimensions d'anàlisi que utilitzem. Adoneu-vos de la diferència amb els fulls de càlcul: les cel·les no s'identifiquen per simples caràcters muts, sinó pels valors de les dimensions d'anàlisi, que sí que tenen associat un significat.

Hipercub

Dir-ne *cub* és un abús de llenguatge, ja que les dimensions no necessàriament tindran la mateixa longitud. A més, generalment tindrà més de tres dimensions. Per tant, hauríem de parlar en tot cas d'*hipercub*.

Figura 3



Cub tridimensional

Si per simplificar el dibuix oblidem momentàniament la dimensió *Client*, en el cas anterior tindríem un cub com el de la figura 3. Cadascuna de les cel·les d'aquest cub representa vendes, que és el tipus de fets que volíem analitzar. Concretament, la cel·la que hi ha en la intersecció entre "Lleida", "Gomes" i "7-1-2000" contindrà totes les dades que tinguem sobre les vendes d'aquest article, en aquesta població i en la data donada.

La multidimensionalitat consisteix simplement a concebre les dades que volem analitzar en termes de fets i dimensions d'anàlisi, de manera que les podem situar en un espai n -dimensional.

El cub

El cub no és res més que una metàfora de com s'han d'entendre les dades. No vol dir que les eines hagin de dibuixar cubs n -dimensionals a la pantalla, ni que forçosament hagin d'emmagatzemar les dades en matrius de n dimensions. Aquest concepte serveix simplement per a ajudar els usuaris a entendre el que poden fer amb les dades.

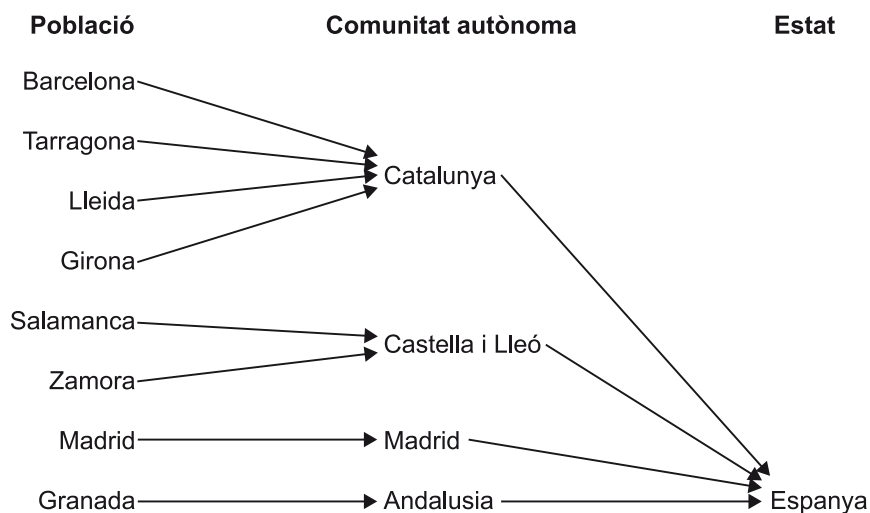
Tenir una gran base de dades que abasti tota l'empresa no té gaire valor per als analistes, que molt probablement es veuran sobrepassats pel seu volum i complexitat. És molt més apropiat focalitzar en un únic tema (fet). La primera contribució de la multidimensionalitat és que permet de donar a cada analista només el cub o conjunt de cubs que correspongui al fet o fets en què estigui interessat. Així reduïm la complexitat del problema i simplifiquem la seva feina.

Tot i que, d'una banda, simplifiquem la visió de les dades perquè els analistes les puguin entendre; de l'altra, hem d'afegir les funcionalitats que en demanen. En aquest sentit, hem de parlar de **nivells de detall i jerarquies d'agregació**. Els punts que hi ha en cada dimensió es poden agrupar per nivells segons una certa jerarquia. Un conjunt de punts d'un cert nivell formen un altre punt en el nivell immediatament superior. Això és especialment important perquè, per a prendre decisions, el que és més habitual és mirar les dades resumides o agregades per grups (per exemple, gammes de productes, regions geogràfiques, etc.).

Jerarquia d'agregació de la dimensió geogràfica

Podem veure un exemple molt clar de jerarquia d'agregació de la dimensió geogràfica dibuixada en la figura 4. Un conjunt de ciutats formen part d'una comunitat autònoma i un conjunt de comunitats formen un estat. D'aquesta manera, la nostra jerarquia d'agregació de la dimensió geogràfica té tres nivells d'agregació diferents: Població, Comunitat i Estat.

Figura 4



La multidimensionalitat dóna molt més que la simple possibilitat de visualitzar les dades en forma de cub. També ens proporciona els fonaments per a poder-lo manipular de manera fàcil i flexible, sense perdre capacitat de càlcul. El canvi del nivell de detall, amb la possibilitat de seleccionar els elements de les dimensions i canviar l'objecte d'anàlisi, és el que anomenarem navegabilitat. Podem continuar veient aquesta navegabilitat en termes de cubs. Així, les eines OLAP ofereixen, amb petites variacions, les operacions que es detallen a continuació.

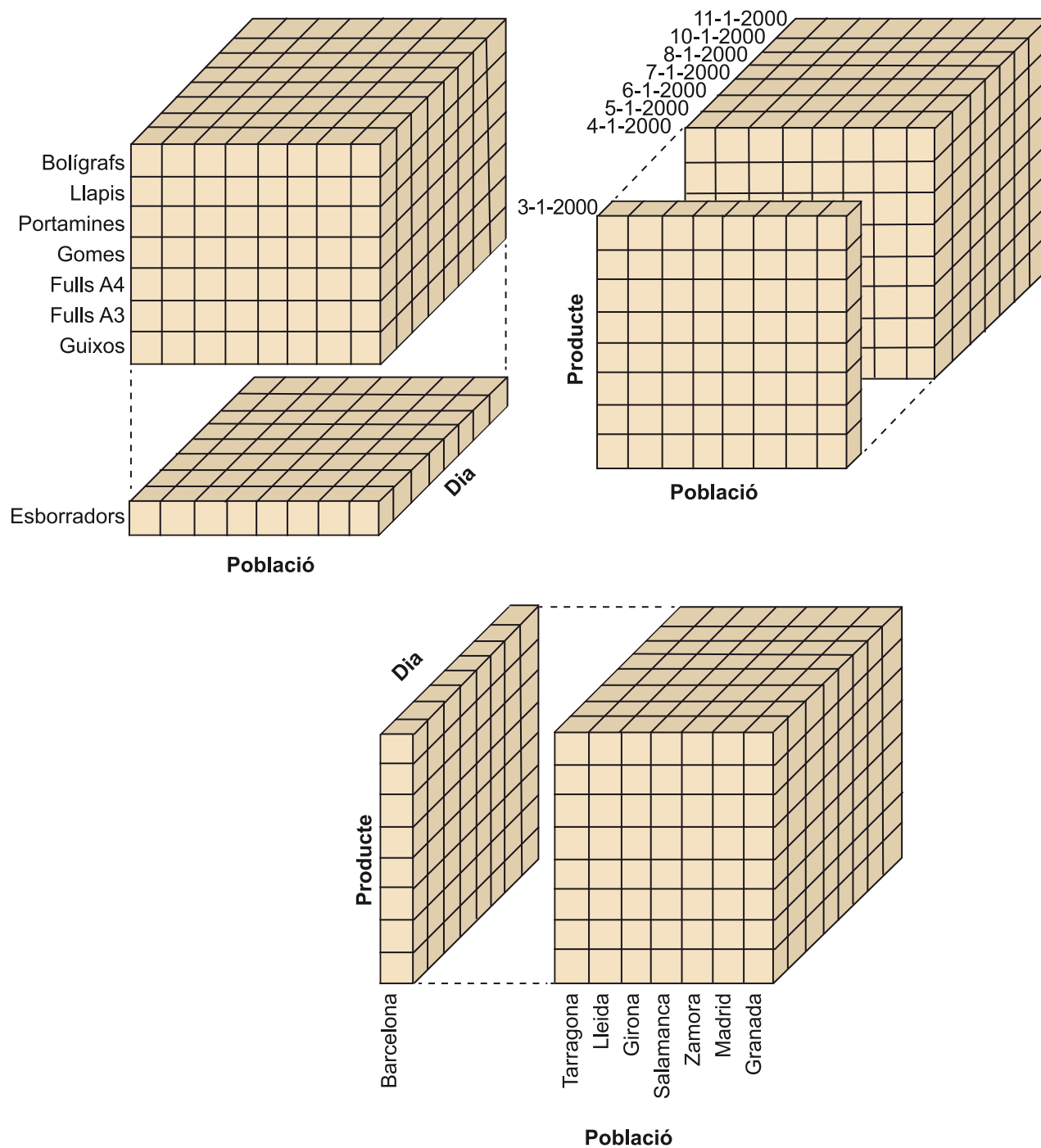
APL

La primera eina multidimensional que corria sobre un ordinador va ser el llenguatge de programació APL, desenvolupat per IBM al final de la dècada dels seixanta. Oferia la possibilitat de definir variables multidimensionals i operar-les mitjançant un conjunt d'operadors específics. Tot i no ser gaire amigable, es va utilitzar molt durant tota la dècada dels setanta.

Slice²: fa un tall al cub de manera que es redueix el nombre de dimensions. Es tracta simplement de fixar un valor a una de les dimensions del cub, de manera que passem a tenir un cub amb $n-1$ dimensions en què totes les cel·les fan referència al valor que hem triat.

(2) *Slice* literalment vol dir 'llesca'.

Figura 5



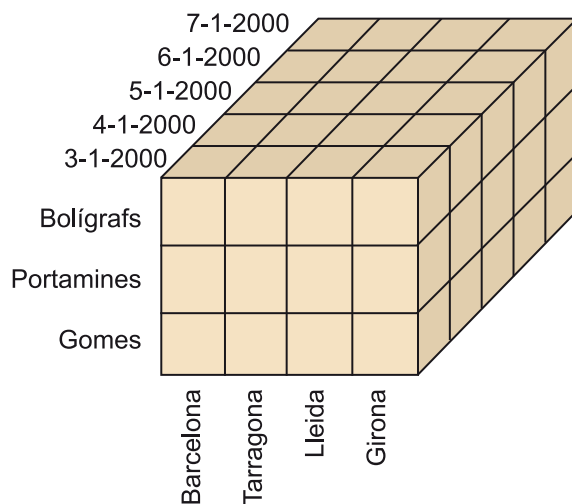
Exemple de *Slice*

La figura 5 exemplifica les tres possibles maneres de fer *Slice* en un cub tridimensional. Fixem-nos en la de dalt a l'esquerra. El que fem en aquest cas és, de tot el cub tridimensional que teníem, quedar-nos només amb un cub bidimensional (una *Slice*), en què totes les cel·les contenen dades referents a "Esborradors". Si a aquest cub bidimensional resultant li tornéssim a fer una *Slice*, ara per al valor "Salamanca" en la dimensió geogràfica, ens quedaria un cub unidimensional (una línia de cel·les), que mostraria les vendes d'esborradors que hi ha hagut a Salamanca cadascun dels dies.

*Dice*³: selecciona un subespai del cub original, sense reduir el nombre de dimensions. Això s'aconsegueix seleccionant un subconjunt de valors en cada una de les dimensions.

⁽³⁾Una traducció literal al català és 'dau'.

Figura 6



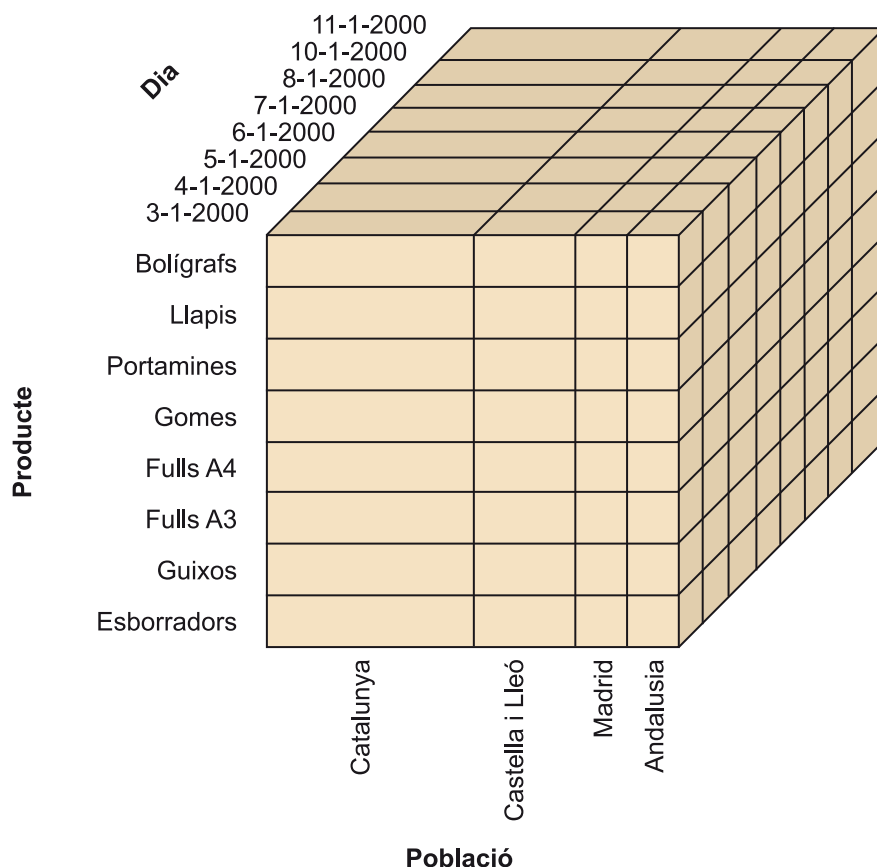
Exemple de *Dice*

En la figura 6 podem veure el resultat d'aplicar una operació de *Dice* al cub tridimensional que tenim a la figura 3. De tot l'espai de mida $8 \times 8 \times 8$ que teníem originàriament, ens quedem només amb un subespai de $4 \times 3 \times 5$, que té moltes menys cel·les, però que continua essent tridimensional.

*Roll-up*⁴: redueix el detall amb què veiem les dades. Agrupa les cel·les del cub seguint una certa jerarquia d'agregació. Per a obtenir les dades de cada cel·la aplica una certa operació matemàtica (com ara la suma o la mitjana) a les dades de les cel·les que formen part de cadascun dels grups.

⁽⁴⁾'Enrotllar', en català.

Figura 7



Exemple de Roll-up

La figura 7 mostra el resultat de fer un *Roll-up* fins al nivell *Regio* en el cub que tenim a la figura 3. Ara, en comptes de tenir una cel·la per a cadascuna de les ciutats catalanes, tenim una única cel·la que conté les dades que fan referència a tot el grup de ciutats de Catalunya. El que hem fet és moure'ns des del nivell *Població* fins al nivell *Regio* dins la jerarquia d'agregació de la dimensió geogràfica que hem explicat en l'exemple corresponent. Passa el mateix per a les ciutats de Castella i Lleó, Andalusia i Madrid. Aquests dos darrers casos, són molt senzills, perquè cada regió conté únicament una ciutat (en el nostre cas) i, consegüentment, les dades de la regió són les mateixes que les de les ciutats corresponents. Però què passa amb "Catalunya" i "Castella i Lleó"? Com n'obtenim les dades? Hem d'aplicar una certa funció d'agregació. Si parlem de quantitats venudes, la de tot Catalunya és la suma de les quantitats venudes a Barcelona, Tarragona, Lleida i Girona (considerant que només tenim botigues en aquestes ciutats, o que només ens interessin aquestes).

*Drill-down*⁵: augmenta el detall amb què veiem les dades. És l'operació inversa al *Roll-up*. En comptes de pujar en una jerarquia d'agregació, baixem, desfem els grups. Adoneu-vos que les funcions d'agregació que hem utilitzat en fer *Roll-up* no es poden desfer si no disposem de les dades originals. Heu d'entendre aquesta operació com un desfer (*undo*) de l'operació *Roll-up*.

Exemple de Drill-down

Aquesta operació correspondria al pas del cub de la figura 7 al de la figura 3. Adoneu-vos, que per a poder-ho fer, hem de disposar de les dades originals. Com aconseguiríem les vendes a cadascuna de les poblacions catalanes si només tinguéssim la suma de quantitats venudes a tot Catalunya?

⁽⁵⁾La traducció literal al català és 'desguassar'.

*Drill-across*⁶: canvia el tema d'anàlisi. Després d'aplicar aquesta operació, continuem tenint el mateix espai n -dimensional que teníem, però ara les cel·les contindran dades que corresponen a un tipus de fet diferent. En termes d'àlgebra relacional, el *Drill-across* s'assembla a una combinació (*join*), en el sentit que associa cada element d'un cub amb un element d'un altre de la mateixa manera que fa la combinació entre taules.

⁽⁶⁾ *Drill-across* literalment vol dir 'foradar a través'. Realment s'utilitza per similitud amb *drill-down*.

Exemple de *Drill-across*

Amb aquesta operació, si partim del cub de la figura 6, obtindrem un cub igual que aquell, però que, en lloc de contenir dades de vendes, conté dades de producció, per exemple. La cel·la que hi ha en la intersecció entre "Lleida", "Gomes" i "7-1-2000" contindrà totes les dades que tinguem sobre la producció d'aquest article, en aquesta població i en la data donada.

Les dimensions s'utilitzen per a seleccionar i agregar les dades al nivell de detall volgut.

Aquestes operacions, juntament amb el canvi d'ordre dels elements que formen les dimensions i el canvi de posicions d'aquestes (altrament dit *fer una rotació* o *pivotar*), són millores importants respecte a la rigidesa en les files i columnes d'un full de càlcul. Però deixant de banda la navegabilitat i aquesta flexibilitat de presentació, les eines OLAP també han d'oferir facilitats per a fer informes. Encara que en pantalla fossin capaces de representar tres o més dimensions, en paper això resulta clarament complicat. Podeu trobar representacions gràfiques més o menys sofisticades, però podríem dir que les dues possibilitats de representació més habituals són les taules estadístiques (com, per exemple, les de la figura 2) i les taules relacionals (com, per exemple, les de les figures 8 i 9).

Figura 8

Dia	Producte	Població	Nombre d'articles	Ingressos
5-1-2000	Bolígrafs	Barcelona	15	39,5
5-1-2000	Gomes	Barcelona	3	1,4
5-1-2000	Portamines	Barcelona	4	8,7
5-1-2000	Bolígrafs	Tarragona	3	5,1
5-1-2000	Gomes	Tarragona	1	0,3
5-1-2000	Portamines	Tarragona	0	0
5-1-2000	Bolígrafs	Lleida	7	15,9
5-1-2000	Gomes	Lleida	0	0
5-1-2000	Portamines	Lleida	6	11,0
5-1-2000	Bolígrafs	Girona	1	1,2
5-1-2000	Gomes	Girona	5	1,4

Dia	Producte	Població	Nombre d'articles	Ingressos
5-1-2000	Portamines	Girona	2	5,1

Figura 9

Dia	Producte	Població	Nombre d'articles	Ingressos
5-1-2000	Bolígrafs	Catalunya	26	61,7
5-1-2000	Gomes	Catalunya	9	3,1
5-1-2000	Portamines	Catalunya	12	24,8

Exemple d'informe

Pensem que, del cub de la figura 6, només ens quedem amb la llesca que conté les dades del dia "5-1-2000". A més, imaginem-nos que cada cel·la conté tant el nombre d'unitats venudes com la quantitat d'euros ingressada. En la taula relacional de la figura 8 podeu veure l'informe que resultaria d'aquesta consulta. Si ara féssim un *Roll-up* fins al nivell *Regio*, obtindríem l'informe de la taula relacional de la figura 9, en què tenim resumides les dades per a tot Catalunya (hem sumat les dades de les ciutats).

La simplicitat de la concepció multidimensional de les dades comporta dos beneficis. D'una banda, ajuda els analistes a entendre les dades. De l'altra, ajuda els informàtics a preveure les consultes que faran els analistes, de manera que pot optimitzar el temps de resposta amb molta més facilitat.

2. Components del model multidimensional

Ara que ja heu vist què és una eina OLAP i els conceptes bàsics de la multidimensionalitat, ho formalitzarem estudiant per separat les estructures, operacions i restriccions d'integritat pròpies d'un model multidimensional. Aquest model és independent de qualsevol eina i us servirà per a aclarir els conceptes generals. Podeu considerar que el que veureu en aquest apartat és a les eines OLAP el mateix que el model relacional és a les bases de dades relacionals.

Els tres components d'un model de dades

Tot model de dades està format per tres components: estructures de dades, operacions sobre les dades i restriccions d'integritat inherents al mateix model. En el cas del model relacional, les estructures són les relacions; els conjunts d'operacions, per exemple, l'àlgebra relacional o el llenguatge SQL; i les restriccions d'integritat, la unicitat i entitat de la clau primària, la integritat referencial i la integritat de dominis.

2.1. Estructures de dades

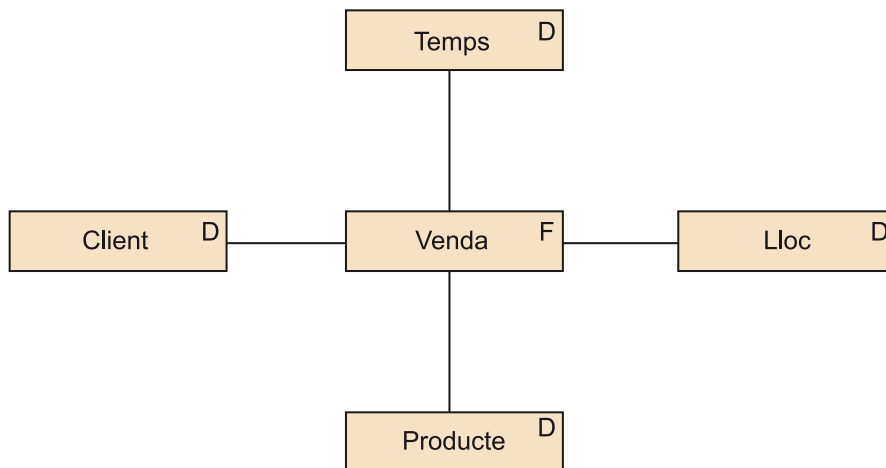
El model multidimensional està marcat per la dicotomia fet-dimensió. Tots els elements han de ser a un costat o a l'altre d'aquesta línia divisòria. Conseqüentment, el primer que hi ha són Fets i Dimensions.

Un Fet representa un tema objecte d'anàlisi.

Una Dimensió representa un punt de vista que utilitzarem en l'anàlisi de les dades.

Per a dibuixar els elements i les relacions entre aquests utilitzarem la notació d'UML, amb què ja hauríeu d'estar familiaritzats. Dimensions i Fets són classificadors. Per tant, els dibuixarem amb rectangles amb el nom a dins, però hi posarem una *D* o una *F* a la cantonada superior dreta per a distingir els uns dels altres. Lligarem cada Fet a les seves Dimensions mitjançant associacions.

Figura 10



Exemple de Fets i Dimensions

En la figura 10 podeu veure un esquema amb un Fet i quatre Dimensions d'anàlisi. Volem analitzar les vendes segons quan i on es van produir, i què i a qui es va vendre.

Els classificadors (Fets i Dimensions) contenen altres elements que donen més detall sobre les dades. Comencem primer per analitzar el contingut de les Dimensions.

2.1.1. Dimensions

Sabem que una dimensió representa un punt de vista des del qual es poden analitzar les dades. Considerarem una dimensió específica i a partir d'aquí definirem els conceptes.

Dins una dimensió podem distingir grups d'instàncies segons la seva grandària (granularitat).

Un Nivell representa un conjunt d'instàncies d'una Dimensió que tenen la mateixa granularitat i el dibuixarem com una classe (un rectangle amb els noms a dins) amb una *N* a la cantonada superior dreta.

Dimensió

Dimensió ve del llatí *dime-tiri*, que es tradueix per 'mesurar' (per exemple, diem que les dimensions d'una nevera són $60 \times 60 \times 180$). Realment, els romans van manllevar la paraula dels grecs, que la utilitzaven en un sentit molt més filosòfic, equivalent a l'accepció que trobem actualment en la geometria (per exemple, parlem d'espai tridimensional).

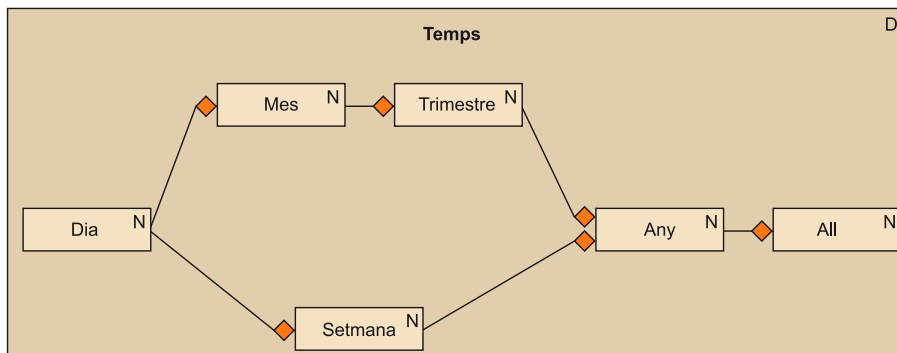
Exemple de granularitats

Dins la Dimensió *Lloc*, hi hauria que les poblacions tenen una granularitat més petita que les regions i aquestes tenen una granularitat més petita que els estats. Per tant, representarem amb una mateixa classe (Nivell) totes les poblacions, però tindrem una classe diferent per a les regions i una altra més per als estats. Totes tres classes representen llocs, per tant, estaran dins la mateixa Dimensió.

Les instàncies d'un cert Nivell s'agrupen per a donar lloc a instàncies d'un altre Nivell de granularitat més gran. Podem dir que les instàncies d'un Nivell formen instàncies d'un altre Nivell, o que hi ha relacions part-tot entre Nivells.

Representarem aquestes relacions amb agregacions entre els Nivells i distingirem un Nivell especial, que anomenarem **All**, que representa l'agrupació de totes les instàncies de la Dimensió alhora.

Figura 11



Exemple de jerarquia d'agregació no lineal

En la Dimensió temporal dibuixada en la figura 11, podem veure que un conjunt de dies dóna lloc a un mes, però un conjunt de dies també pot donar lloc a una setmana. També podem agrupar mesos per a obtenir trimestres i podem obtenir anys per agrupació de setmanes o trimestres. Finalment, veiem un Nivell especial **All** que representa el grup format per tots els anys que estem interessats a analitzar.

Generalment, els Nivells dins una Dimensió i les agregacions que els uneixen formen un graf dirigit, conegut com a *jerarquia d'agregació*.

Dels axiomes de la mereologia es poden deduir les propietats d'aquest graf següents:

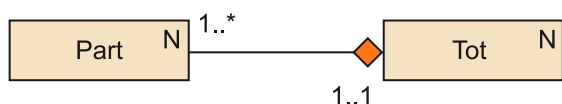
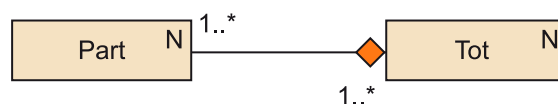
- No pot contenir cicles.
- Conté un únic Nivell que no té parts (l'anomenen *atòmic*).
- Hi pot haver o no un Nivell **All**, però si hi és:
 - Només n'hi ha un.
 - Conté exactament una instància.
 - No és part de cap altre Nivell.
- Tots els Nivells que no són part de cap altre Nivell, es poden connectar directament al Nivell **All**.
- Totes les instàncies (menys les del Nivell atòmic) han de tenir almenys una part.
- Totes les instàncies (menys les del Nivell atòmic) poden tenir més d'una part.

Mereologia

La mereologia és la ciència que estudia les relacions part-tot.

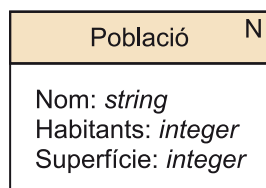
- Els conjunts de parts de dues instàncies d'un mateix nivell no han de ser disjunts necessàriament.
- Sempre podem construir el graf de manera que totes les instàncies participin en un tot, menys la del Nivell A11.

Figura 12

Opció b)**Opció a)**

D'aquestes propietats deduïm que les possibles multiplicitats de les agregacions que hi ha dins una Dimensió són únicament les que teniu dibuixades en la figura 12 (en alguns casos particulars, podríem trobar que en comptes de l'asterisc hi ha un valor concret per a les fites superiors). L'única tria que hi ha realment és si una part pot participar en un únic tot (opció a) o en més d'un (opció b). El que és més habitual és l'opció a, per tant, si no s'indica res, pensarem que la multiplicitat que tenim és aquesta.

Figura 13



Com podeu veure en la figura, un Nivell té associats un conjunt d'atributs, els anomenats *descriptors*.

Els atributs que podem trobar en un Nivell contenen la informació no jeràrquica i estan definits sobre un domini discret. S'anomenen *Descriptors*.

Els Descriptors no s'utilitzen per a formar grups, sinó simplement per a seleccionar instàncies o mostrar-los en els informes.

Relacionant els nous conceptes veiem el següent:

Una Dimensió conté un conjunt de Nivells relacionats per agregacions. Cadascun dels Nivells té atributs que anomenem *Descriptors*.

Finalment hem de comentar que, tot i que l'anàlisi multidimensional es basa en l'àlgebra de matrius, cal dir que, al contrari que en el cas dels espais lineals, el model multidimensional, per si mateix, no inclou cap tipus d'ordre o dis-

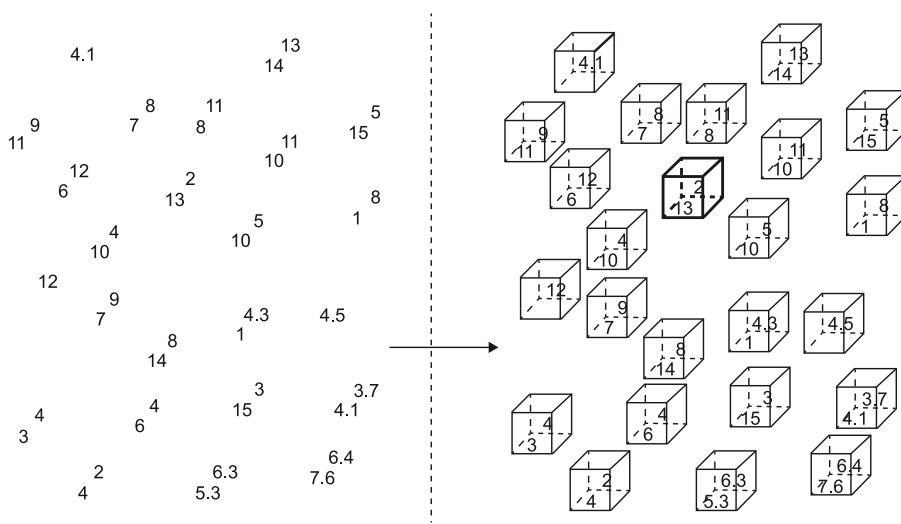
tància entre les instàncies d'una Dimensió. Si parléssim de la Dimensió temporal, podríem definir fàcilment aquest ordre (les deu del matí van abans que les tres de la tarda i estan separades per cinc hores), però com ho faríem amb la Dimensió de productes o clients?

2.1.2. Fets

Oblidem-nos per un moment de les Dimensions i pensem que volem analitzar el Fet. Tenim un conjunt immens de dades (que anomenarem **mesuraments**, en un sentit ampli de la paraula) dins el nostre magatzem de dades. El que volem és posar una mica d'ordre en aquest conjunt inabastable i apropar les dades als analistes. El primer que cal fer és representar els mesuraments que fan referència al mateix esdeveniment⁷ dins una mateixa estructura. D'això en direm *cel·la* (escrit amb minúscules).

(7) Qualsevol succés o concepte susceptible de ser analitzat.

Figura 14



Exemple de cel·les

A la part esquerra de la figura 14 tenim un conjunt de mesuraments: quantitats produïdes, quantitats venudes, costos, ingressos, etc. A la part dreta de la mateixa figura hem agrupat els mesuraments que fan referència al mateix esdeveniment. Per exemple, la cel·la vermella podria correspondre a la venda que es va fer a en Jordi fa un parell de dies. Li vam vendre dos objectes i li vam cobrar tretze euros.

Tot i aquesta primera agrupació de mesuraments en cel·les, encara no complim els requeriments dels usuaris. El pas següent és poder unir cel·les per a obtenir-ne d'altres més grans, que no solament representin un esdeveniment, sinó molts altres (per exemple, totes les vendes que es van fer abans d'ahir: la d'en Jordi, la d'en Joan, la d'en Pere, etc.). El conjunt de totes les cel·les C amb la unió $\cup$ forma un semigrup commutatiu. És a dir, compleix les propietats següents:

- Tancat (la unió de dues cel·les sempre és una altra cel·la).

$$\forall x, y \in C \quad x \cup y \in C$$

- Commutatiu (no importa l'ordre en què fem una unió, obtindrem la mateixa cel·la).

$$\forall x, y \in C \quad x \cup y = y \cup x$$

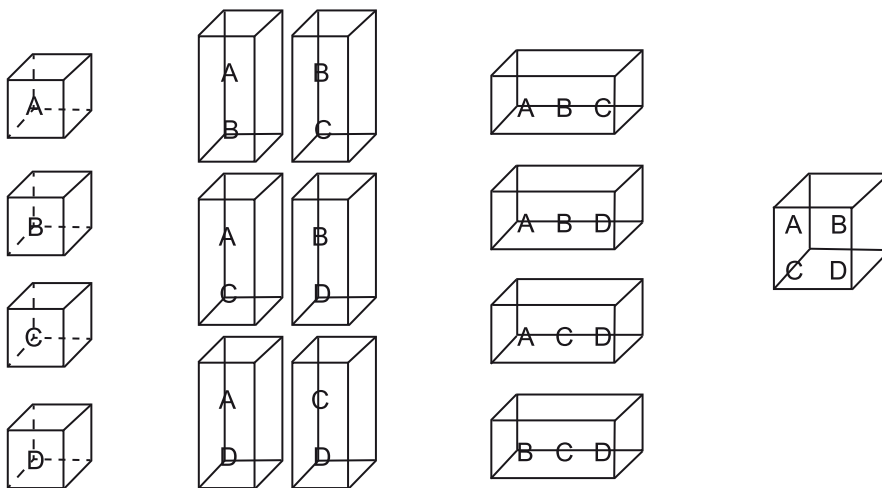
- Associatiu (podem prioritzar una seqüència d'unions com vulguem, sense alterar la cel·la resultat).

$$\forall x, y, z \in C \quad x \cup (y \cup z) = (x \cup y) \cup z$$

- Element neutre (hi ha un element que operat amb qualsevol altre no el modifica).

$$\forall x \in C \quad x \cup \emptyset = x$$

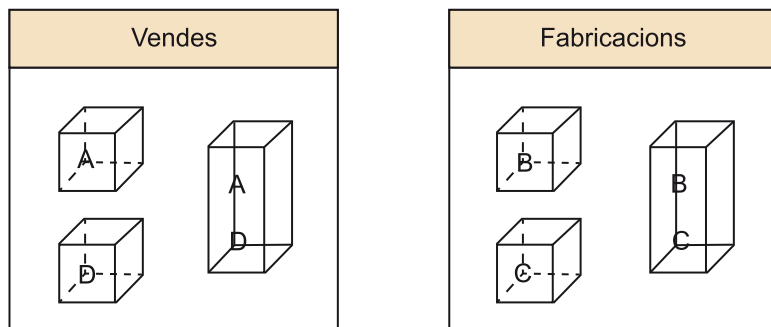
Figura 15



Si li diem C_A al conjunt de totes les cel·les atòmiques⁽⁸⁾ i permetem totes les unions possibles, obtenim que C conté $2^{\text{Card}(C_A)} - 1$ cel·les (que és la cardinalitat del conjunt de parts de C_A , $\text{Card}(P(C_A))$). En la figura 15, podeu veure totes les cel·les que es poden obtenir a partir de quatre cel·les atòmiques. La cardinalitat de C creix exponencialment respecte al nombre de cel·les atòmiques. Conseqüentment, no cal tenir gaires cel·les atòmiques perquè el problema de guardar o consultar C esdevingui intractable.

⁽⁸⁾ Les cel·les atòmiques són aquelles que no podem obtenir com a resultat de la unió d'altres cel·les.

Figura 16



Pensem ara en el significat d'aquesta unió de cel·les. Quin sentit té fer la unió d'una cel·la que representa vendes amb una cel·la que representa la fabricació d'un cert producte? Quin tipus de cel·la obtindrem com a resultat? Una venda? Una fabricació? No té gaire sentit unir cel·les de tipus diferents. Si restringim la unió a cel·les del mateix tipus (el mateix Fet), ja només tenim $\sum_i \text{Card}(P(C_{Ai}))$ cel·les, essent C_{Ai} el conjunt de cel·les atòmiques instància del Fet i . En la figura 16 podeu veure quines cel·les podem obtenir si considerem que A i D són d'un tipus de Fet, i B i C d'un altre tipus.

El conjunt de cel·les s'ha reduït dràsticament. No obstant això, encara n'hi ha moltes que no interessin als analistes. Les cel·les guanyen significat només quan estan associades a instàncies de les Dimensions. Només quan sabem quin producte es va vendre, a qui li vam vendre, on li vam vendre, etc., la cel·la aconsegueix tot el seu significat. Conseqüentment, només interessin les cel·les que estan lligades a una instància de cadascuna de les Dimensions. No agrupem qualsevol conjunt de cel·les sinó que utilitzem com a criteri d'agrupació les Dimensions. No agruparem una cel·la que contingui dades mensuals amb una altra que contingui dades trimestrals. A quin Nivell de la Dimensió temporal estaria el resultat d'aquesta unió? El que sí que farem és agrupar tres cel·les mensuals per a obtenir una cel·la trimestral.

Les instàncies de les Dimensions

Recordeu que les instàncies de les Dimensions estan classificades per Nivells. Per tant, cada cel·la està associada a un Nivell en cadascuna de les Dimensions.

Del conjunt de cel·les del mateix Fet que estan associades a instàncies del mateix Nivell per a cadascuna de les Dimensions en direm *Cel·la*⁹.

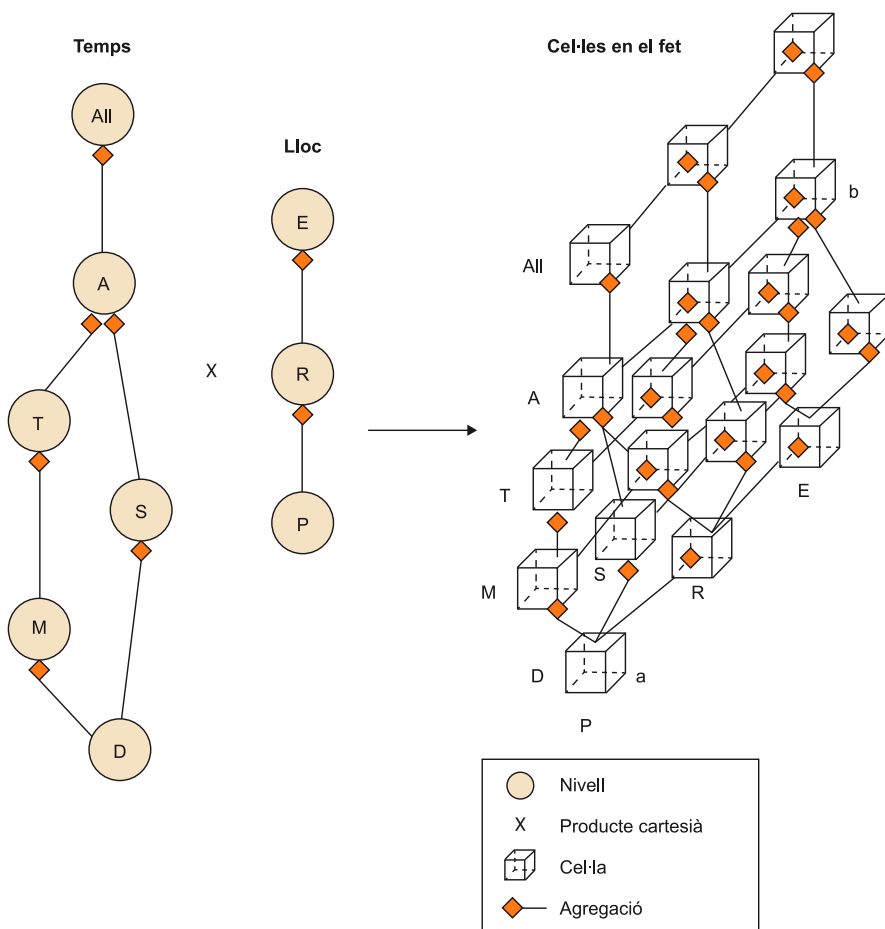
⁽⁹⁾Utilitzarem la majúscula per a distingir els conjunts de cel·les de les cel·les individuals.

Una Cel·la (que dibuixarem com una classe amb una C a la cantonada superior dreta) representa un conjunt d'instàncies d'un Fet que tenen la mateixa granularitat.

Dins un Fet tindrem tantes Cel·les com elements hi ha en el producte cartesià dels Nivells de les Dimensions. Igual que els Nivells, aquestes Cel·les estaran relacionades per agregacions. Això indica que les instàncies d'una Cel·la componen les instàncies de la Cel·la immediatament superior.

Les instàncies de les cel·les s'agrupen en **jerarquies d'agregació**.

Figura 17



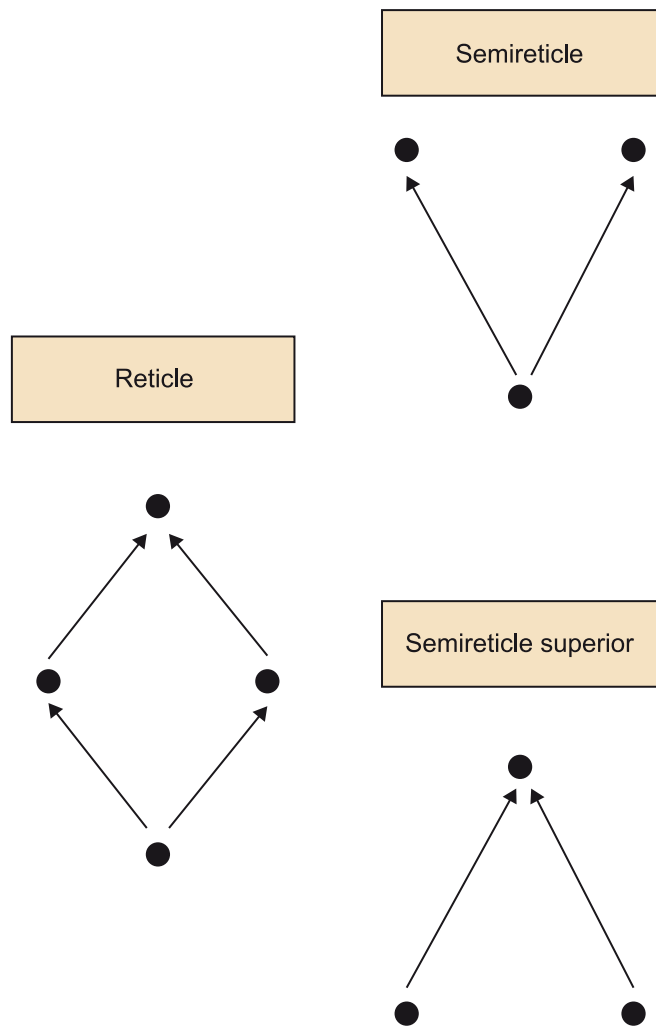
Exemple de Cel·les en un Fet

En la figura 17 teniu dibuixat un esquema amb les dues Dimensions que hem vist fins ara: Temps i Lloc. La primera té sis Nivells i la segona només en té tres. Per a cada combinació possible de Nivells d'aquestes Dimensions, tenim una Cel·la diferent al nostre Fet. Per exemple, la Cel·la "a" correspon a les granularitats Dia-Població, la Cel·la "b" correspon al Nivell Any-Estat, així fins a arribar a les divuit (sis per tres) combinacions que hi ha. Totes les instàncies de la Cel·la "b" estan associades amb una instància d'Any i amb una altra d'Estat.

Amb això obtenim que cada Fet conté un graf amb estructura de semireticle inferior, que serà un reticle només si totes les Dimensions també ho són. Aquest graf ens mostra com podem agrupar les cel·les per a obtenir-ne de més complexes.

Reticle

És un conjunt ordenat en què dos elements qualssevol en tenen un de suprem (la més petita de les fites superiors o elements majorants) i un d'ínfim (la més gran de les fites inferiors o elements minorants). Si només hi ha una de les dues fites s'anomena *semireticle* (superior o inferior segons quina tinguem).



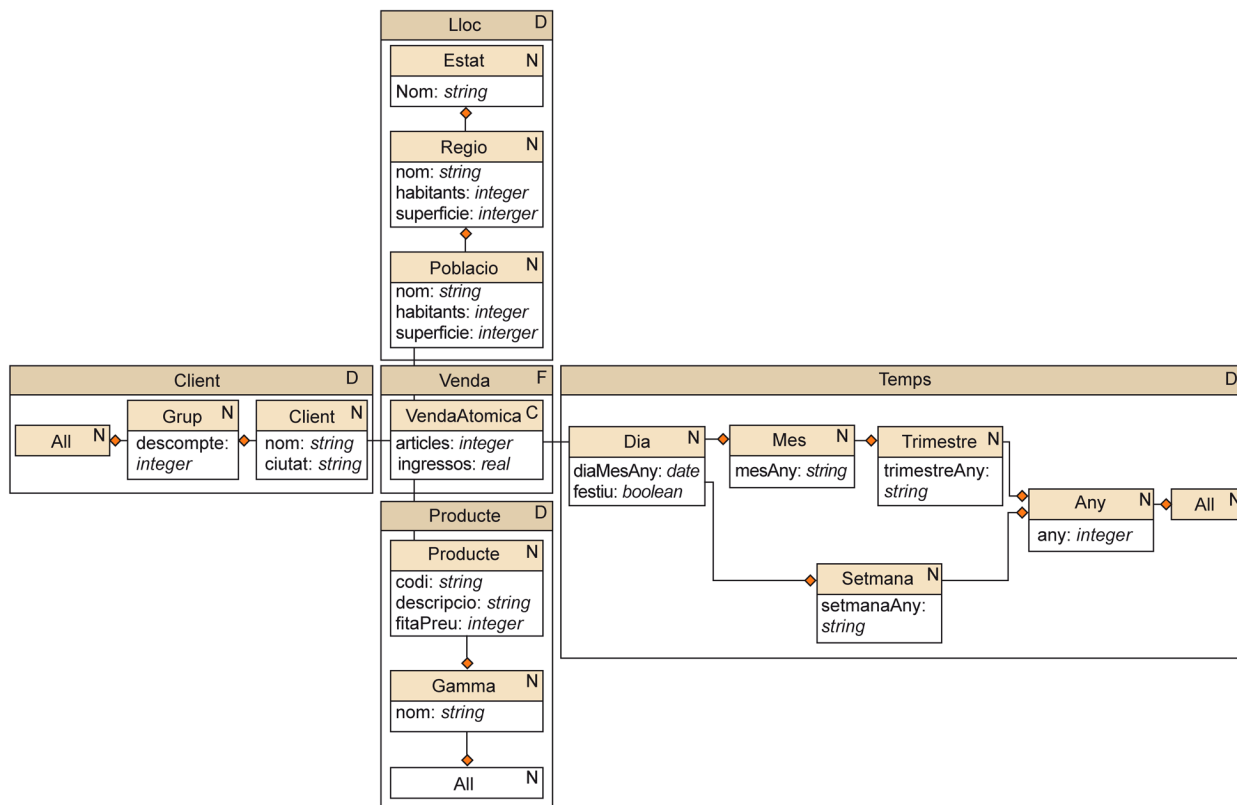
Una Mesura és un atribut d'una Cel·la.

Els valors de les Mesures (que abans hem anomenat *mesuraments*) d'una cel·la s'obtenen com a funció dels mesuraments de les cel·les que la componen. Al contrari del que ocorre amb els Descriptors, la majoria de Mesures són de tipus numèric, tot i que n'hi ha algunes que són booleanes o, fins i tot, textuales.

Arribat a aquest punt, podem relacionar els conceptes vistos en aquest apartat:

Un Fet conté un conjunt de Cel·les (amb *C* majúscula) relacionades per agregacions. Cadascuna de les Cel·les té atributs que anomenem *Mesures*.

Figura 18



Exemple d'esquema multidimensional

La figura 18 mostra un esquema multidimensional sencer. A causa de la forma que tenen, s'acostumen a anomenar *esquemes en estrella*. Per a simplificar-ho, només es dibuixen les Cel·les que tenen especial interès (en aquest cas, únicament la Cel·la associada al Nivell atòmic de cada Dimensió).

Com podeu veure en la figura 18, associem les Cel·les amb els Nivells corresponents. La multiplicitat d'aquestes associacions sempre és $*-1$. Cada cel·la s'associa amb una única instància d'un Nivell. Consegüentment, no cal explicitar aquesta multiplicitat.

Els principals elements d'un esquema multidimensional són, d'una banda, les Dimensions, els Nivells i els Descriptors; i de l'altra, simètricament els Fets, les Cel·les i les Mesures.

2.2. Operacions sobre les dades

En aquest apartat veurem un conjunt d'operacions algebraiques sobre cubs. Malgrat que hem dit que la multidimensionalitat es fonamenta en la representació de les dades en forma de cubs, encara no hem vist què és un cub. Un cub és una funció injectiva que va d'un espai n -dimensional finit (definit pel producte cartesià de n Nivells $\{N_1, \dots, N_n\}$) al conjunt d'instàncies d'una Cel·la (C_c).

$$c(x): N_1 \times \dots \times N_n \rightarrow C_c$$

Un cub simplement és una funció que diu a cada punt de l'espai quina cel·la hi va. Per tant, podem fer amb aquest qualsevol cosa que fariem amb una funció (per exemple, compondre'n o unir-ne dos).

Figura 19

Fet $\Rightarrow$ <i>Drill-across</i>	Dimensió $\Rightarrow$ <i>CanviBase</i>
Cel·la $\Rightarrow$ –	Nivell $\Rightarrow$ <i>Roll-up</i>
Mesura $\Rightarrow$ <i>Projecció</i>	Descriptor $\Rightarrow$ <i>Selecció</i>

Cada operació afecta directament un únic tipus d'element del model (podeu veure les correspondències en la taula de la figura 19): *Drill-across* permet de seleccionar un Fet; *CanviBase* permet de seleccionar el conjunt de Dimensions que utilitzarem; *Roll-up* permet de seleccionar el Nivell de detall en cada Dimensió; *Projecció* permet de triar les Mesures que volem veure, i *Selecció* permet d'utilitzar els Descriptors per a triar les instàncies concretes de cada Nivell que volem veure. Per què no hi ha cap operació associada a Cel·la? Com podem triar la Cel·la que volem veure? No hi pot haver una operació associada a Cel·la perquè entraria en conflicte amb el *Roll-up*. Com que la Cel·la queda determinada pels Nivells que triem, no podem triar Nivells i Cel·les alhora.

Veiem les definicions d'aquestes cinc operacions:

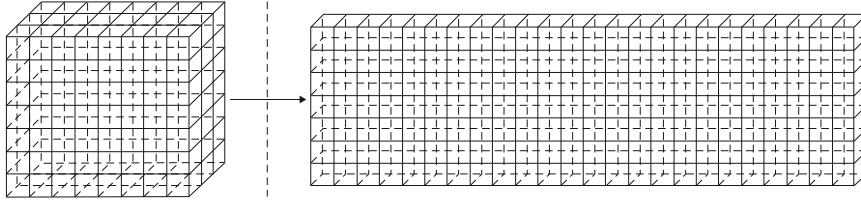
Drill-across: canvia l'objecte d'anàlisi, és a dir, el Fet. Per poder-ho fer hi ha d'haver alguna relació entre el Fet que tenim i el Fet que volem tenir. Necessitem una funció injectiva entre els dos, que determini per quina cel·la del cub destinació hem de substituir cada cel·la del cub origen. El cas més comú és que els dos Fets comparteixin Dimensions de manera que els mateixos Nivells identifiquin les cel·les dels dos cubs.

$$c_{\text{destinacio}}(x) = \text{Drill-across}_{f(c_{\text{origen}})}(c_{\text{origen}}) = f(c_{\text{origen}}(x))$$

Projecció: simplement selecciona el conjunt de Mesures que volem veure d'entre les que hi ha disponibles en la Cel·la del cub origen. És equivalent a l'operació homònima de l'àlgebra relacional.

$$c_{\text{destinacio}}(x) = \text{Projeccio}_{m_1, \dots, m_k}(c_{\text{origen}}) = c_{\text{origen}}(x)[m_1, \dots, m_k]$$

Figura 20



CanviBase: redistribueix el mateix conjunt de cel·les dins un altre espai. Per a aplicar-la necessitem disposar d'una funció injectiva entre els dos espais (és a dir, entre Dimensions), de manera que cada punt de l'espai del cub destinació determini un punt del cub origen. Això pot servir simplement per a reordenar els punts de l'espai o també pot canviar el nombre de Dimensions (com teniu esquematitzat en la figura 20), però no canvia el nombre de cel·les, ni el seu contingut.

$$c_{destinacio}(x) = CanviBase_{f(c_{origen})} = c_{origen}(f(x))$$

Roll-up: agrupa les cel·les d'un cub basant-se en una jerarquia d'agregació. Augmenta la granularitat fins a un cert Nivell. Redueix el nombre de cel·les, no el de Dimensions.

$$c_{destinacio}(x) = Roll-up_{Nivell}(c_{origen}) = \bigcup_{r(y)=x} c_{origen}(y)$$

Selecció: mitjançant un predicat lògic sobre els Descriptors del Nivells, selecciona un conjunt de punts dins l'espai n -dimensional. És absolutament equivalent a l'operació homònima de l'àlgebra relacional.

$$c_{destinacio}(x) = Seleccio_{predicat}(c_{origen}) = \begin{cases} c_{origen}(x), & \text{si } predicat(x) = \text{cert} \\ indefinit, & \text{si } predicat(x) = \text{fals} \end{cases}$$

Aquest conjunt d'operacions és tancat, és a dir, els operands són cubs i el resultat també. Això implica que podem concatenar les operacions. Per exemple, l'operació *Slice* que hem vist en l'apartat "Eines OLAP i multidimensionalitat" seria equivalent a triar un punt mitjançant l'operació *Selecció* i fer un canvi de base per a reduir en un el nombre de Dimensions. Noteu que la Dimensió que eliminem amb el canvi de base només conté la instància "k" i, per tant, tot i que perdem una Dimensió, no perdem cap cel·la (l'espai $a \times b \times 1$ té el mateix nombre de punts que $a \times b$).

$$Slice_{Ni=k}(c_{origen}) = CanviBase_{f_{N1 \times \dots \times Nn \rightarrow N1 \times \dots \times Ni-1 \times Ni+1 \times \dots \times Nn}}(Seleccio_{Ni=k}(c_{origen}))$$

Una altra operació que podeu trobar a faltar és el *Drill-down*. Com ja hem dit, és la inversa del *Roll-up* i no es pot fer si no disposeu de les dades detallades.

Exemple de seqüència d'operacions

Tenim el cub $2 \times 2 \times 2$ següent:

Unitats produïdes per producte, fàbrica i mes	Fàbrica del Vallès		Fàbrica del Prat	
	Gener 2002	Febrer 2002	Gener 2002	Febrer 2002
Bolígrafs	100.000	110.000	450.000	420.000
Gomes	337.000	473.000	904.000	995.000

Volem veure per als mateixos mesos i productes, els articles venuts a Catalunya. Hauríem de fer un *Drill-across* cap a *Vendes*, però no és possible fer-ho directament perquè les Dimensions no coincideixen. Primer ens hem de desfer de la Dimensió *Fàbriques*.

$A := \text{Roll-up}_{\text{Fàbrica}::\text{All}}(\text{"Unitats produïdes per Producte, Fàbrica i Mes"})$

A	All	
	Gener 2002	Febrer 2002
Bolígrafs	550.000	530.000
Gomes	1.241.000	1.468.000

Ara només tenim un valor en la Dimensió de fàbriques, per tant, podem fer un canvi de base per a quedar-nos amb les mateixes quatre cel·les però una Dimensió menys.

$B := \text{CanviBase}_{\text{Producte} \times \text{Temps}}(A)$

B	Gener 2002	Febrer 2002
Bolígrafs	500.000	530.000
Gomes	1.241.000	1.468.000

Com que B només té dues Dimensions i totes dues les tenim a *Vendes*, ja podem fer el *Drill-across*. Atès que la funció que utilitzem és la identitat entre Dimensions, no cal explicitar-la, sinó simplement dir quin és el Fet destinatí (Vendes en aquest cas).

$C := \text{Drill-across}_{\text{Vendes}}(B)$

C	Gener 2002	Febrer 2002
Bolígrafs	articles: 465.837	articles: 513.284
	ingressos: 973.427'30	ingressos: 1.075.143'80
Gomes	articles: 1.348.378	articles: 1.490.281
	ingressos: 498.462'20	ingressos: 523.093'90

Hem obtingut ara cel·les de *Vendes* amb totes les seves Mesures. Només estàvem interessats a veure articles, així que podem projectar només aquesta Mesura.

$$D := \text{Projeccio}_{\text{articles}}(C)$$

D	Gener 2002	Febrer 2002
Bolígrafs	465.837	513.284
Gomes	1.348.378	1.490.281

Ara ja tenim les dades que volíem, però no volem que facin referència a tots els supermercats, sinó solament als que hi ha situats a Catalunya. Per tant, cal introduir-hi la Dimensió geogràfica.

$$E := \text{CanviBase}_{\text{Articles} \times \text{Lloc} \times \text{Temps}}(D)$$

E	Espanya	
	Gener 2002	Febrer 2002
Bolígrafs	465.837	513.284
Gomes	1.348.378	1.490.281

Si ara volem seleccionar les dades de Catalunya, abans hem d'aconseguir les dades d'Espanya detallades per regió. En aquest cas ho podem fer, perquè tenim disponibles les dades per població (tal com indica l'esquema de la figura 18). Adoneu-vos que si les tinguéssim només emmagatzemades per estat, no ho podríem fer.

$$F := \text{Drill-down}_{\text{Lloc::Regió}}(E)$$

F	Catalunya		Castella i Lleó		Madrid		Andalusia	
	Gener 2002	Febrer 2002	Gener 2002	Febrer 2002	Gener 2002	Febrer 2002	Gener 2002	Febrer 2002
Bolígrafs	275.827	290.918	85.472	111.291	58.172	59.723	46.366	51.352
Gomes	784.172	918.012	293.829	288.409	141.003	140.298	129.374	143.562

Finalment, només hem de seleccionar les dades de Catalunya per a tenir el que ens interessa.

$$R := \text{Seleccio}_{\text{Regio.nom}=\text{"Catalunya"}}(F)$$

R	Catalunya	
	Gener 2002	Febrer 2002
Bolígrafs	275.827	290.918
Gomes	784.172	918.012

2.3. Restriccions d'integritat inherents al model

Ja hem vist quins són els elements del model i les operacions que es poden fer amb les dades. Ara hem de veure quines dades no es poden inserir i quines operacions no estan permeses.

2.3.1. Unicitat i entitat de la Base

Recordeu que un cub és una funció que va d'un espai n -dimensional a una Cel·la.

Dels diferents conjunts de Nivells que defineixin espais en què puguem col·locar les instàncies d'una Cel·la en direm *Bases*.

No és necessari que totes les Dimensions d'una Cel·la participin en una Base. La Base simplement indica quines Dimensions identifiquen les cel·les (és el mateix concepte que la clau candidata del model relacional).

Figura 21

VendaAtomica	C
articles: <i>integer</i> ingressos: <i>real</i>	
<<Base>> [Producte, Dia, Poblacio, Client]	

Exemple de Base

En el nostre exemple de la cadena de supermercats, una certa venda estaria identificada per un dia, un producte, una població i un client. Per tant, *Producte*, *Dia*, *Poblacio* i *Client* defineixen un espai en què podem col·locar les instàncies de *VendesAtomiques*. En la figura 21 podeu veure com representariem això.

Les Bases han de complir les restriccions següents:

- No podem col·locar dues cel·les al mateix punt de l'espai. Consegüentment, els valors que tinguin les cel·les per a les associacions amb els Nivells d'una Base han de ser diferents per a cada cel·la. Per exemple, no hi pot haver dues instàncies de *VendaAtomica* associades al mateix *Producte*, *Dia*, *Poblacio* i *Client*.
- Hem de saber a quin punt de l'espai col·loquem cada cel·la. Consegüentment, les associacions amb els Nivells d'una Base no admetran el valor nul. Cada cel·la ha d'estar relacionada amb una instància de cada Nivell que forma una Base. Les associacions amb aquests Nivells tenen com a multiplicitat mínima un 1 del costat del Nivell.
- Els Nivells que formen una Base han de ser funcionalment independents. Si hi ha un Nivell que depèn dels altres, el traurem del conjunt que forma

la Base. En el nostre exemple, no podria ser que un client només pogués comprar a una certa població. Si fos així, la base estaria formada només pels Nivells `Client`, `Dia`, i `Producte`. La població ja quedaria determinada pel client.

El conjunt de Nivells que formen una Base han de ser funcionalment independents. A més, les associacions d'aquests Nivells amb la Cel·la han de tenir multiplicitat mínima 1 del costat del Nivell i no pot ser que dues cel·les estiguin associades amb les mateixes instàncies per tots els Nivells de la Base.

2.3.2. Acumulació o agregació

Sens dubte, l'operació més característica de l'anàlisi multidimensional és el *Roll-up*. Dissortadament, també és la més problemàtica. Vegem quines són les tres condicions perquè el resultat d'un agregació sigui correcte:

Compatibilitat

Habitualment, quan s'acumulen un conjunt de dades, el que es fa és sumar-les, però no sempre és així. Realment es poden aplicar altres operacions com, per exemple, la mitjana, el mínim, el màxim o, fins i tot, el producte. L'operació que s'apliqui depèn del tipus de Mesura que agreguem i de la Dimensió al llarg de la qual ho fem. Certes operacions són incompatibles amb alguns tipus de Mesures i Dimensions.

Exemples d'operacions

La quantitat venuda durant un mes és la suma de les quantitats venudes cadascun dels dies.

O per a obtenir l'interès anual multipliquem els mensuals.

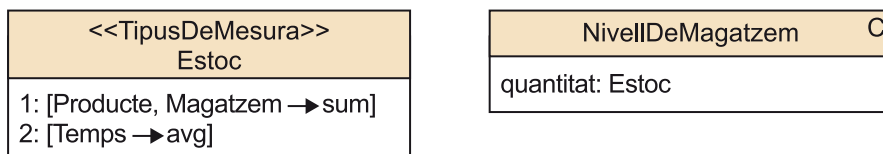
L'operació d'agregació, el tipus de Mesura que agreguem i la Dimensió al llarg de la qual ho fem han de ser compatibles.

Agregació d'estocs

Un exemple clar d'agregació diferent segons la Dimensió són els estocs de productes als magatzems. Si registrem l'estoc diàriament a cada magatzem que tinguem, l'estoc mensual no serà la suma dels estocs diaris (la suma és incompatible amb els estocs i la Dimensió `Temps`). Si avui tinc un cotxe al magatzem i demà també en tinc un, en total no en tinc dos, perquè realment són el mateix cotxe. El que és més habitual en aquest cas és fer la mitjana. En canvi, si tinc dos magatzems i tinc un cotxe al mateix temps a cadascun d'aquests, sí que tinc dos cotxes.

L'operació d'agregació més comuna és la suma. Per això, en comptes de parlar de compatibilitat entre Mesura, Dimensió i operació, de vegades es parla simplement de Mesures additives (si es poden sumar en qualsevol Dimensió), semiadditives (si hi ha Dimensions en què no es poden sumar) i no additives (si no es poden sumar al llarg de cap Dimensió).

Figura 22



Les Mesures

Les Mesures que fan referència a ingressos acostumen a ser perfectament additives al llarg de qualsevol Dimensió. Per contra, les que fan referència a estats, com ara els estocs o els saldos, acostumen a ser semi-additives (no es poden sumar al llarg del temps). Finalment, algunes que fan referència a intensitat (com, per exemple, la temperatura) o es basen en fórmules matemàtiques són no additives.

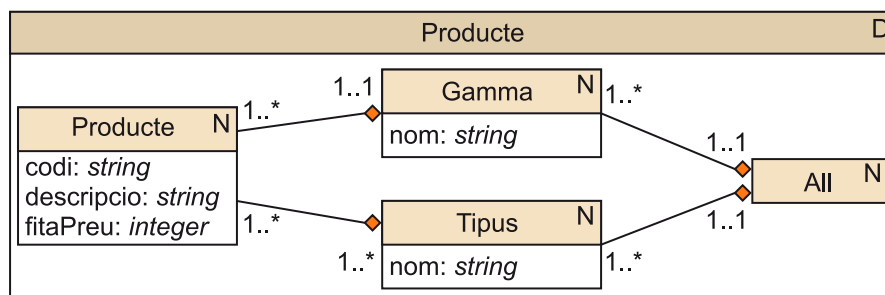
Si no es diu el contrari, assumirem la utilització de la suma per a fer *Roll-up*. Si volguéssim utilitzar una operació matemàtica diferent, caldria explicitar-ho com un paràmetre més de la mateixa operació ($\text{Roll-up}_{\text{Nivell}}(\text{cub}, \text{operacio})$) o en l'esquema (definint un tipus de Mesura com mostra la figura 22). En aquest cas, els estocs s'agregarien mitjançant la suma al llarg de *Producte* i *Magatzem*, i mitjançant la mitjana al llarg de *Temps*. Com que certes operacions no són commutables (no és el mateix la suma de mínims que el mínim de les sumes), també podeu indicar l'ordre en què s'han de fer les agregacions (en l'exemple, primer es farien les sumes i després les mitjanes).

Disjuntivitat

Segons l'operació d'agregació que apliquem, correm el risc de considerar més d'una vegada la mateixa dada sense que ens n'adonem. Això ocorre quan hi ha un Nivell que conté instàncies amb conjunts de parts no disjunts. En aquests casos és obligatori indicar les multiplicitats en la jerarquia d'agregació per a saber quines cel·les podem utilitzar en l'agregació i quines no. En el pitjor cas, haurem d'utilitzar les instàncies de la Cel·la atòmica, que per definició han de ser disjunts.

Certes operacions d'agregació demanen que els conjunts de parts de les instàncies que utilitzem com a operands siguin disjunts.

Figura 23



Exemple d'instàncies no disjunts

Considereu ara la Dimensió *Producte* com la teniu dibuixada en la figura 23. S'ha afegit un nou Nivell: *Tipus*. Ara és essencial explicitar les multiplicitats de les agregacions. Fixeu-vos que l'agregació entre *Producte* i *Tipus* té multiplicitat 1..*-1..* en comptes d'1..*-1..1 com la resta d'agregacions. Això vol dir que hi ha productes que són de més d'un tipus i els conjunts de parts de les instàncies de *Tipus* no són disjunts. El producte "Kinder sorpresa" és una joguina i una xocolata. Conseqüentment, pertany als tipus "Joguina" i "Xocolata" alhora. Si ara intentem calcular els ingressos totals sumant els ingressos de joguines i xocolates, estarem comptant els ingressos per vendes de "Kinder sorpresa" dues vegades. Segons el tipus de Mesura, això pot ser correcte o no. Podeu indicar els casos en què no sigui vàlid com es mostra en la figura 24. Així, sabem que no hem de calcular totes les Mesures de tipus *Ingres* a partir de dades de granularitat *Tipus*, sinó que ho haurem de fer des d'una granularitat més petita, com ara *Producte*.

Figura 24

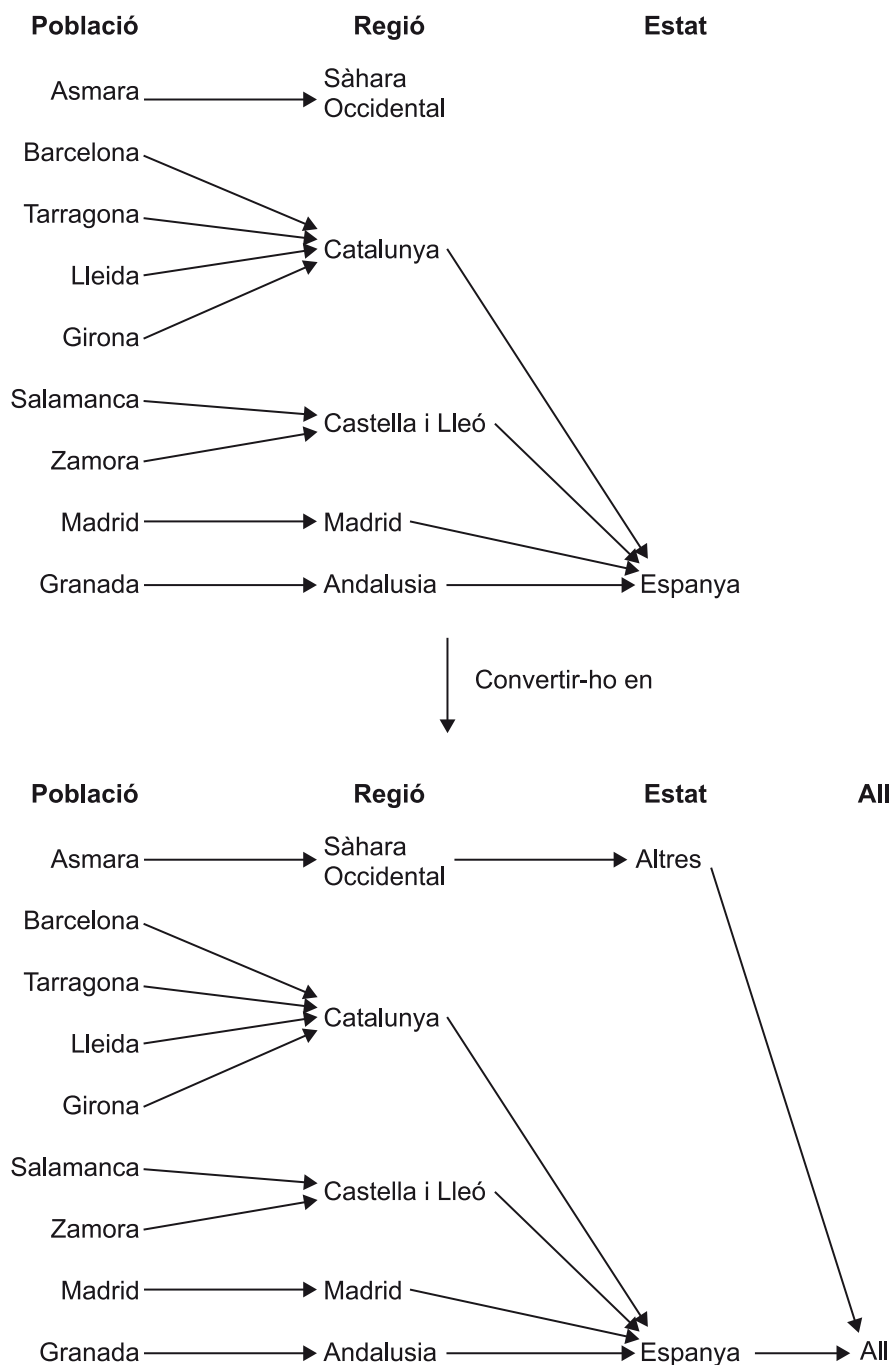
Tipus N	VendaAtomica C
nom: <i>string</i>	articles: <i>integer</i>
Font invàlida per: Ingres	ingressos: Ingres

Completesa

Podria ser que alguns Nivells continguessin instàncies que no participessin en la composició de cap instància dels Nivells superiors. Això ho hem d'evitar afegint instàncies fictícies al Nivell superior, per no oblidar-nos d'operar les dades associades a aquestes instàncies. La multiplicitat mínima de l'agregació al costat del compost ha de ser sempre 1. Un Nivell ha de cobrir completament els Nivells que té per sota en la jerarquia d'agregació.

Hem de garantir que totes les instàncies participen com a mínim en una instància dels Nivells immediatament superiors de la jerarquia d'agregació.

Figura 25



Exemple de no-completesa en la jerarquia d'agregació

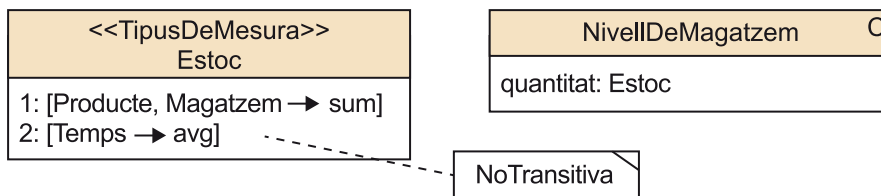
Imaginem que ampliem el negoci de supermercats i n'obrim un de nou a Asmara. A més d'afegir aquesta població a la Dimensió **Lloc**, també l'afegim a la regió "Sàhara Occidental" de la qual forma part. Però com que el referèndum d'autodeterminació encara no s'ha fet, decidim no registrar-hi a quin estat pertany (podeu veure com queda la Dimensió **Lloc** en la part superior de la figura 25). Com podem calcular ara el total d'ingressos? Ja no serveix consultar els ingressos obtinguts a Espanya, perquè hi ha una regió que no és part d'"Espanya". Tot i que **Regio** cobreix completament **Poblacio**, **Estat** ja no cobreix **Regio**. La solució és crear una instància artificialment d'**Estat**, que podem dir "Altres", que tingui com a parts "Sàhara Occidental" i totes les altres regions que no formin part de cap estat. Amb això, **Estat** ja cobreix **Regio**, però encara cal crear el Nivell **All** amb una instància que agrupi "Espanya" i "Altres" (la Dimensió **Lloc** queda com està dibuixada en la part inferior de la figura 25). Adoneu-vos que abans d'afegir "Asmara" no calia el Nivell **All**, perquè el Nivell més alt de la jerarquia ja tenia només una instància.

2.3.3. Transitivitat

La darrera restricció que hem de tenir en compte, també relacionada amb l'agregació, és la transitivitat de les operacions d'agregació. De vegades, no obtenim el mateix resultat si operem amb resultats parcials que si operem directament amb les dades bàsiques. Hi ha operacions que no són transitives (per exemple, la mitjana).

Ens hem d'assegurar que les operacions d'agregació són transitives. En cas que no ho siguin, el resultat correcte sempre és el que s'obté d'operar amb les dades atòmiques.

Figura 26



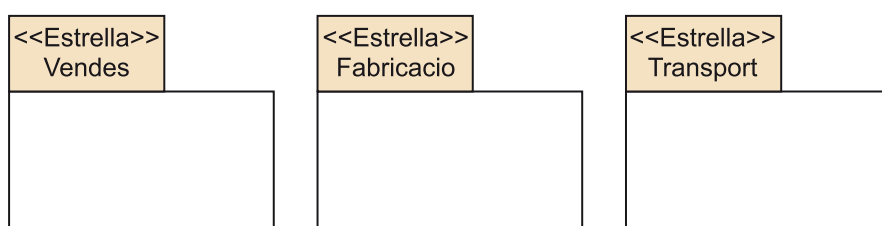
Exemple de no-transitivitat

Per simplicitat, imaginem que obrim el supermercat cada dia de l'any. Cada dia del mes de gener tenim al magatzem 100 unitats, cada dia del mes de febrer 300 i cada dia del mes de març 200. Sembla clar que la mitjana d'unitats que tenim al gener és 100, al febrer, 300 i al març, 200. Quin és l'estoc durant el primer trimestre? 200 unitats? No podem fer la mitjana de les mitjanes mensuals. Hem de fer la mitjana dels estocs diaris. Realment, la mitjana és 196,6 ($100 \times 31 + 300 \times 28 + 200 \times 31 = 196,6$). Si interessa reflectir això, ho podem fer com podeu veure en la figura 26.

3. Disseny conceptual

Tot i que les eines OLAP es consideren relativament fàcils d'utilitzar, construir-les i mantenir-les requereix coneixements especialitzats. És per això que dedicarem bona part d'aquest mòdul a estudiar-ne el disseny. De la mateixa manera que per a les bases de dades operacionals, ho farem en tres passos: Disseny conceptual, Disseny lògic i Disseny físic. En aquest apartat veurem com es pot fer un disseny conceptual multidimensional.

Figura 27



El primer que farem és aconseguir un esquema expressat amb UML dels cubs de dades que ens interessin.

Un Fet i el seu corresponent conjunt de Dimensions formen una Estrella.

Un conjunt d'estrelles formen una Constel·lació.

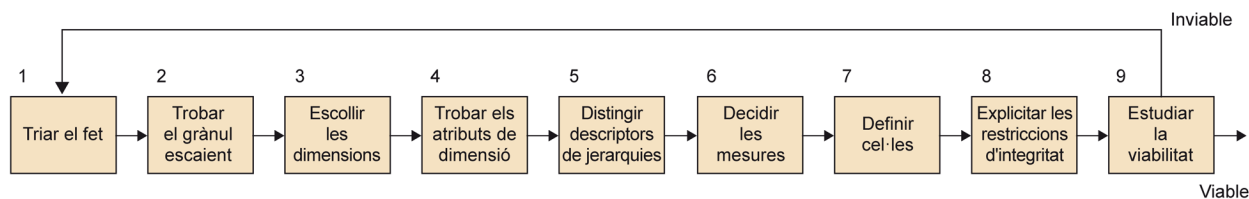
Podem representar una Estrella com un paquet (com podeu veure en la figura 27). Vegem ara com podem dissenyar el que hi ha dins de cadascun d'aquests paquets.

3.1. Una metodologia per a dissenyar una Estrella

Un dels avantatges del disseny multidimensional és que utilitza la tècnica de dividir i vèncer. No intentem de dissenyar de cop tots els magatzems departamentals de la nostra empresa, ni satisfer tot d'una les necessitats dels analistes, sinó que fem petits dissenys força independents els uns dels altres.

Dissenyem per separat cada Estrella i ho fem en un procés iteratiu que consta de nou passos.

Figura 28



3.1.1. Triar el Fet

Primer de tot hem de saber quin tema estudiarem amb la nostra Estrella, quin serà el Fet. Un Fet no és res més que un conjunt d'esdeveniments amb dades numèriques associades. Els candidats habituals són els processos de negoci. Per **processos de negoci** entenem tots els processos operacionals de l'organització que estan suportats per algun sistema informàtic del qual se'n poden extreure dades. Probablement, els mateixos analistes us suggeriran el procés de negoci que heu de modelar. Però adoneu-vos del condicionament tan fort que tenim per a la tria d'un Fet: hem de poder disposar de les dades, preferiblement en suport informàtic.

El primer que hem de fer per a dissenyar una Estrella és triar el Fet objecte d'anàlisi.

Primerament, intenteu definir Fets "petits". Definir de cop una única Estrella que inclogui tots els àmbits de l'empresa és pràcticament impossible. El que és més fàcil és començar desenvolupant Estrelles per a analitzar processos de negoci que estiguin suportats per un únic sistema operacional. Per exemple, si tenim un sistema de gestió de vendes, ens resultarà relativament fàcil dissenyar i posar en funcionament una Estrella per a l'anàlisi de vendes.

3.1.2. Trobar el grànul escaient

El grànul és l'individu últim que volem analitzar, la Cel·la més petita que volem tenir disponible.

Ja no parlem d'un concepte gran i abstracte com un procés de negoci, sinó del tipus d'objecte concret que volem analitzar. Els grànuls més típics són les transaccions individuals (per exemple, les vendes) o les instantànies diàries d'un estat (per exemple, els estocs). Generalment, acostuma a ser qualsevol tipus d'esdeveniment que es doni amb una freqüència relativament alta.

Exemple de grànuls possibles

Els grànuls candidats a la nostra Estrella de vendes podrien ser "vendes mensuals de producte per botiga", "vendes diàries d'un cert producte a un client en una població", "compres que fa un client en un moment donat en una certa botiga" o, fins i tot, "línies dels tiquets de compra", que representen la venda d'un producte en un moment donat en una certa botiga.

Triar el grànul del nostre Fet és molt important, perquè determina la dimensionalitat de la base de dades i, com veurem quan estudiem la viabilitat de la implementació, té un impacte directe en la grandària del conjunt de dades. Un grànul molt petit resultarà en una base de dades molt gran. Per contra, un grànul més gran implicarà renunciar a alguna Dimensió o més possiblement a la possibilitat de calcular certes Mesures, a causa, per exemple, de problemes amb la transitivitat de les operacions o la no-disjuntivitat dels compostos.

Exemple de tria del grànul

Què ocorre si triem com a grànul "vendes mensuals de producte per botiga"? Com sabrem quin dia del mes hem venut més? Aquest seria un grànul massa gran. Ara bé, penseu els problemes que pot generar triar com a grànul "línies de tiquets de compra". Cada cop que una caixera passa un article pel lector de codis de barres, generem una cel·la en el nostre sistema. Quan triga una caixera a fer això? Quantes caixeres tenim? Quant de temps treballen diàriament? Probablement, el sistema acabarà contenint més dades de les que pot manejar. Si realment no ens cal tant de detall, hauríem d'intentar reduir el volum de dades d'alguna manera. En el nostre exemple pensem que els sistemes operacionals identifiquen cada client (per exemple, amb el número de targeta de crèdit) i que un mateix client compra el mateix article moltes vegades, de manera que podem generar una única cel·la per totes les seves compres d'un mateix producte, en tot un dia, dins la mateixa població.

Un bon disseny sempre demana triar el grànul més petit possible, no perquè els usuaris vulguin veure sempre les dades més detallades, sinó per no perdre la possibilitat de calcular cap dada derivada amb el mínim error. El límit sempre és la disponibilitat de dades en els sistemes operacionals. No podem triar "línies dels tiquets de compra" si el sistema operacional corresponent només registra la quantitat total de la compra.

Triar un grànul massa gran representa perdre informació. Però triar-lo **massa** petit pot representar malbaratar espai o arribar a fer inviable el projecte per excés de dades.

Grandària d'un grànul

Entenem que un grànul és més gran com més esdeveniments representa. Així és més gran la granularitat Mes que Dia i Dia que Hora.

Vegeu també

Ja hem vist aquests conceptes en l'apartat "Restriccions d'integritat inherents al model".

3.1.3. Escollir les dimensions que s'utilitzaran en l'anàlisi

Unes Dimensions típiques són Temps, Producte, Client, Promocio, Magatzem, Tipus o Estat. Si el grànul està clarament definit, trobar un primer conjunt de Dimensions d'anàlisi és immediat a partir de la mateixa definició. A aquest primer conjunt inicial, hi podem afegir altres Dimensions, però només si cada combinació de les instàncies de les Dimensions inicials determina una instància de la Dimensió que volem afegir. Si en determinés més d'una, voldria dir que hem de reconsiderar la Dimensió o el grànul mateix per a poder-la afegir.

La multiplicitat de les associacions entre Fets i Dimensions sempre és molts a un. Moltes instàncies del Fet s'associen amb la mateixa instància de Dimensió, però només una instància de Dimensió s'associa amb cada instància del Fet.

Exemple de tria de dimensions

Si el nostre grànul és "vendes diàries d'un cert producte a un client en una població", llavors tindrem almenys les Dimensions Temps, Producte, Client i Lloc. A més d'aquestes quatre Dimensions, n'hi ha d'altres, com ara Promocio, que queda determinada per les altres. Un cert producte en un moment donat i en una població està essent promocionat d'alguna manera (per exemple, amb un 10% de descompte) i volem estudiar com afecta això les vendes. Per contra, no podríem afegir la Dimensió Publicitat perquè no podem associar cap tipus de publicitat en concret a una cel·la. Imaginem que hi ha cartells publicitaris a les carreteres, que coincideixen amb diferents campanyes radiofòniques locals i altres campanyes televisives estatals. Com podem associar això a una venda a una població? Si el que s'anuncia és un establiment, com ho associarem a la venda d'un producte? Les compres poden no coincidir exactament en el temps amb les campanyes publicitàries que les provoquen. Com associarem una campanya a un dia en concret? Publicitat no queda determinada per Temps, Producte, Client i Lloc.

El grànul mateix ja determina un primer conjunt de Dimensions. Hem d'afegir a aquest els altres punts de vista que vulguem utilitzar en l'anàlisi i que quedin determinats pel conjunt inicial de Dimensions.

El que és més habitual és trobar entre quatre i quinze Dimensions. Trobar-ne dues o tres és molt estrany i us hauria de fer sospitar que hi ha alguna Dimensió més que es podria afegir al disseny (per exemple, la Dimensió temporal és gairebé omnipresent). En l'extrem oposat, que n'hi hagi vint tampoc no sembla fàcilment justificable. Moltes d'aquestes Dimensions resultaran prescindibles o en trobarem algunes parelles o grups que estaran fortament correlacionats de manera que es podrien representar com una única Dimensió, com veurem en l'apartat "Dependències entre Dimensions".

Les dimensions

Heu d'entendre les dimensions com les variables independents que afecten cada observació. Penseu que és difícil trobar un fenomen amb moltes variables i un amb poques no resulta gaire interessant d'analitzar.

3.1.4. Trobar els atributs de cada dimensió

D'entre els atributs que podem trobar en els sistemes operacionals, hem d'escollir els que pertanyen a les Dimensions i ens seran útils per a triar i descriure l'espai d'anàlisi. Hem d'agafar qualsevol atribut que creiem que pugui ser útil per a seleccionar, agrupar o simplement posar com a capçalera d'un informe. Una Dimensió pot tenir amb relativa facilitat més de cinquanta atributs. Cal documentar tant l'origen com la interpretació de cadascun dels atributs.

Com que els atributs de Dimensió han de servir per a fer seleccions i agrupacions, han de tenir sempre un domini discret, mai continu. Com seleccionareu un valor si té infinits decimals? A més, un atribut mai no ha d'estar codificat ni abreujat i ha de ser fàcilment intel·ligible per a l'usuari (en principi, un codi de barres no és un bon atribut per a la Dimensió *Producte*). Pot interessar tenir alguns codis no directament interpretables (com ara el codi de barres) per a identificar les instàncies dins el mateix sistema. Però, atès que mai no seran utilitzats per l'usuari, hem de calcular molt bé l'espai que realment ens estalviem amb aquests. Finalment, també cal dir que la qualitat d'un atribut ha d'estar garantida, no hi pot haver errors en la seva escriptura.

Exemple de Dimensió temporal

Per exemple, en una Dimensió temporal, a més de saber el dia de la setmana, el mes o l'any, hauríem de poder seleccionar dies festius o laborables, setmanes de vacances o de classe, de matrícula o d'exàmens, anys normals o de traspàs, etc.

Quan es van definir les Dimensions dins l'apartat "Estructures de dades" dins de "Components del model multidimensional" ja es va parlar d'Atributs de Dimensió i de les característiques més importants. Però a tall de recordatori tenim la definició següent:

Un atribut de Dimensió ha d'estar definit sobre un domini discret, ha de ser descriptiu, fàcil de recordar i entenedor a primer cop d'ull. Els atributs textuais acostumen a pertànyer a les Dimensions, mentre que els atributs numèrics són habitualment Mesures.

Exemple d'atribut de dimensió

Podrem considerar que el preu d'un producte (un atribut numèric) ens ve donat i que mai no canviarà. Si creiem que ens pot ser útil per a seleccionar vendes, estariem parlant d'un atribut de la Dimensió *Producte* en l'*Estrella Vendes*.

Si un atribut ens sembla interessant i no compleix les característiques desitjables, cal que el modifiquem fins que prengui la forma adequada: fent-lo més explicatiu (per exemple, canviant codis per frases), definint rangs de valors (per exemple, preus de menys d'un euro, d'entre un i cinc euros, d'entre cinc i cinquanta o de més de cinquanta), netejant-lo (per exemple, eliminant valors nuls, o arreglant errors d'ortografia), etc.

S'ha de convertir qualsevol tipus de codi que hi hagi en els sistemes operacionals en un text explicatiu i discretitzar els dominis continus per a facilitar la selecció i comprensió dels atributs de Dimensió.

3.1.5. Distingir entre descriptors i jerarquies d'agregació

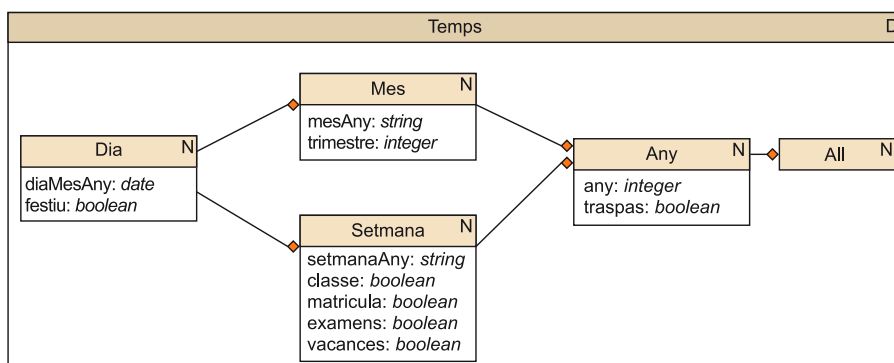
D'entre els atributs que hi ha en una Dimensió, n'hem de distingir dos tipus: els que utilitzarem per a agrupar i els que serviran simplement per a seleccionar. Per exemple, utilitzarem els mesos i els anys per a agrupar dies, mentre que

utilitzarem la festivitat o no festivitat d'un dia simplement per a seleccionar. Els primers són els que defineixen les jerarquies d'agregació, els segons són els descriptors.

Un atribut per si mateix no és d'un tipus o d'un altre. Que un atribut defineixi un Nivell dins una jerarquia d'agregació o no depèn molt més del que els usuaris volen, que de les dades en si mateixes. Les jerarquies s'haurien de consensuar amb els usuaris. Amb jerarquies massa grans augmentem la complexitat del sistema, però amb jerarquies massa simples podem perdre possibilitats d'anàlisi o alentir el sistema.

El Nivell atòmic de les jerarquies queda definit pel grànul que hàgim triat. Si tenim vendes diàries, el Nivell atòmic de la Dimensió Temps serà Dia i no Hora ni Mes. A partir d'aquest Nivell, la resta del graf està definit per les dependències funcionals que hi hagi entre els atributs d'agrupació que triem. Col·locarem la resta d'atributs com a Descriptors en el Nivell que els pertorqui.

Figura 29



Una altra interpretació de la jerarquia d'agregació

Si tornem a veure ara la Dimensió de la figura 11, podem veure que Dia determina Setmana i Mes, però ni Setmana determina Mes ni Mes determina Setmana, tot i que tots dos determinen Any. Recordeu que les dependències funcionals compleixen la propietat transitiva. Conseqüentment, com que Mes determina funcionalment Trimestre i aquest determina Any, podem dir que Mes determina Any. No obstant això, recordeu també que les agregacions també compleixen la propietat transitiva i que no cal explicitar la que hi ha entre Mes i Any. Si els usuaris no utilitzessin els trimestres per a agrupar mesos, aquest Nivell desapareixeria de la jerarquia (seria simplement un atribut associat a Mes, com podeu veure en la figura 29) i llavors sí que dibuixaríem l'agregació entre Mes i Any. En aquesta figura, hem dibuixat a cada Nivell només els Descriptors que en depenen directament.

Podeu entendre les agregacions que formen una jerarquia d'agregació com a dependències funcionals entre atributs de Dimensió que utilitzem per a agrupar.

3.1.6. Decidir quines són les mesures que interessin

Típicament, les Mesures són atributs numèrics que es poden sumar com, per exemple, les quantitats venudes o els ingressos que genera una venda. Podeu triar qualsevol atribut que tingueu directament disponible en les bases de dades operacionals o que pugueu derivar a partir dels que hi tingueu. Penseu només que l'espai que ocupin les Mesures (és a dir, la grandària de les cel·les) afectarà sensiblement el volum total de la base de dades (si ocupen molt, la base de dades serà massa gran per a ser funcional). El que és millor és trobar totes les Mesures interessants per als analistes i després escollir-ne només les que pensem que seran més útils.

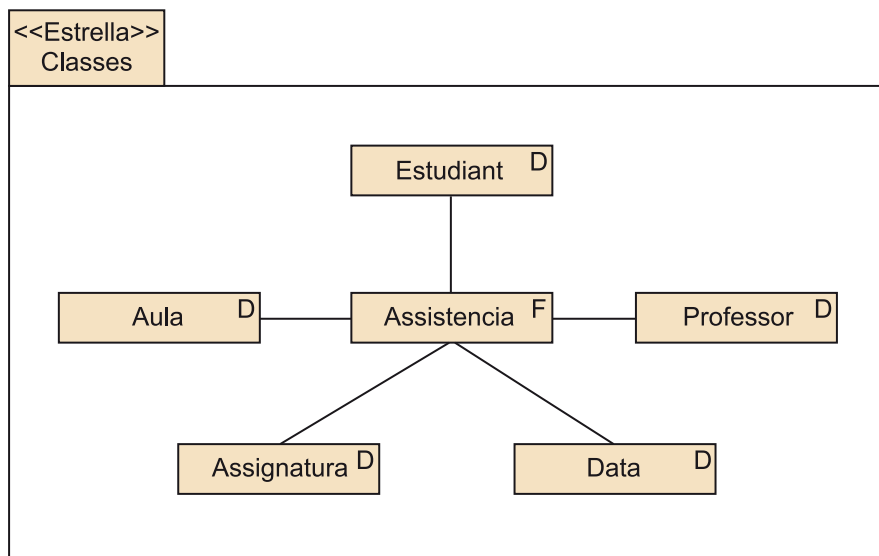
Nota

Si una Mesura és derivada, cal que ho indiqueu com faríeu amb qualsevol atribut amb UML (posant una barra davant el nom i adjuntant un comentari amb la fórmula utilitzada en la derivació).

Les Mesures són atributs numèrics normalment additius.

Generalment, com més Mesures contingui un Fet, més útil serà l'Estrella. Però de vegades trobareu alguns Fets molt útils que no tenen cap Mesura. Com ja hem dit abans, les instàncies dels Fets poden representar esdeveniments. Normalment, en donar-se els esdeveniments, mesurem alguna cosa. No obstant això, hi ha casos en què únicament volem tenir constància que l'esdeveniment s'ha donat. En aquest cas, el Fet no té cap Mesura.

Figura 30



Exemple de Fet sense Mesures

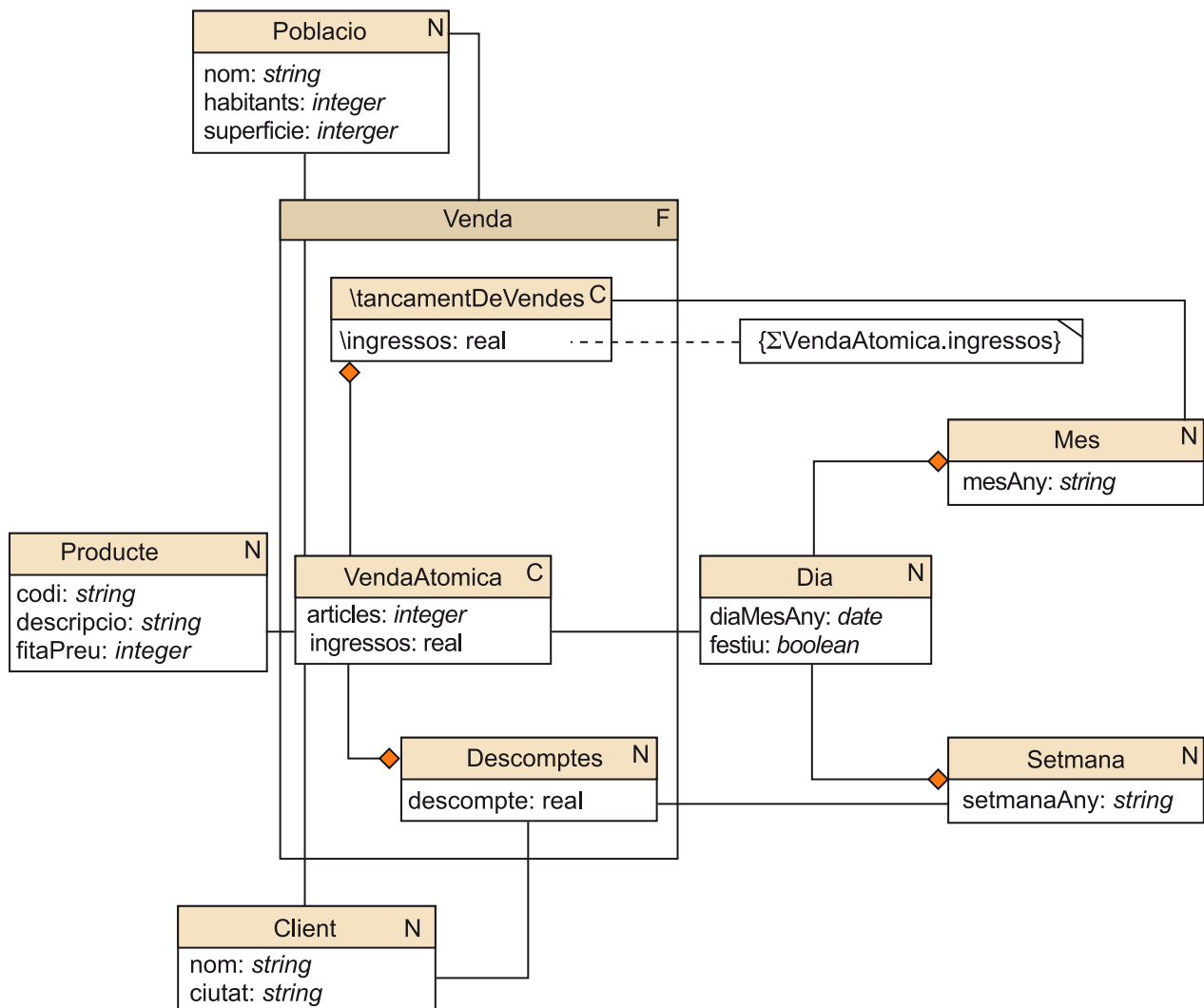
Penseu que som en una universitat presencial i volem analitzar l'assistència a classe de cada alumne, segons l'assignatura, el professor que imparteix la classe i l'aula i la data en què es fa aquesta. Tindríem l'Estrella de la figura 30. *Assistencia* no tindria cap Mesura, perquè l'únic que ens interessa és tenir constància de si un alumne ha assistit a classe o no. Amb aquesta Estrella sense Mesures ja podem contestar consultes com ara quin professor té més alumnes a classe o quants professors han fet alguna classe en una certa aula.

Hi ha alguns Fets que no tenen cap Mesura.

3.1.7. Definir Cel·les

Veureu que en el conjunt de Mesures que heu triat n'hi ha de diferents granularitats. Cada una pertany a una de les Cel·les del Fet i heu d'anar amb molt de compte de posar-les allà on els pertoca. Heu de dibuixar Cel·les de diferent granularitat dins el mateix Fet, per a contenir les Mesures de la granularitat corresponent. De tot el reticle de Cel·les que teniu exemplificat en la figura 17, només cal que dibuixeu les Cel·les que contenen Mesures o que considereu especialment rellevants.

Figura 31



Exemple de Mesures de diferent granularitat

Pensem que ens interessen quatre Mesures: el nombre d'articles venuts, els ingressos que genera una venda, els ingressos que tenim mensualment a cada població (quan tanquem els comptes) i el descompte que fem a cada client. Considerem que el descompte s'aplica setmanalment i depèn de la despesa que ha fet aquest client durant el període de temps corresponent i del tracte que hi tinguem. En la figura 31 podeu veure com quedaria l'esquema. Apareixen tres Cel·les diferents que representen les tres granularitats que ens interessen. Cada Cel·la conté les Mesures que tenim en aquesta granularitat. Fixeu-vos que els ingressos mensuals per població són una Mesura derivada que s'obté sumant els ingressos que tenim en Cel·les de granularitat més petita. Com que és l'única Mesura que conté aquesta Cel·la, també podem marcar la Cel·la com a derivada (amb una barra davant el nom). Per contra, no podem calcular el descompte directament de les altres Mesures, perquè depèn de l'oferta que fem a cada client en cada moment. Per a simplificar la representació, podem considerar que, si una Cel·la no està associada a cap Nivell d'una Dimensió, ho està al Nivell A11. D'aquesta manera, tot i que no quedi reflectit en el dibuix, Descomptes està associat al Nivell Poblacio::A11 i al Nivell Producte::A11.

De totes les Cel·les que expliciteu, caldrà emmagatzemar-ne algunes físicament i unes altres només contindran Mesures que es podran obtenir simplement fent *Roll-up* i aplicant l'operació que correspongui (i que marcarem com a derivades). Però en el disseny conceptual no hem de considerar si unes dades derivades s'emmagatzemen físicament o simplement es calculen quan cal. Això ja serà una decisió de disseny físic. En aquest moment només cal deixar constància de tot el que es consideri especialment rellevant.

Cal explicitar les Cel·les que contenen Mesures interessants per als analistes, encara que siguin derivades. Podríem no posar tots els elements que siguin derivats en l'esquema. Els explicitem simplement per a remarcar com són d'importants.

3.1.8. Explicitar les restriccions d'integritat

Un cop ja tenim totes les Mesures, Cel·les i Nivells, tan sols queda expressar les restriccions d'integritat corresponents (Bases, restriccions d'agregació i restriccions de transitivitat), com ja hem vist en l'apartat "Restriccions d'integritat inherents al model".

Com a últim pas del disseny conceptual, cal expressar les restriccions d'integritat.

Com a especialment important, en aquest moment cal destacar les Bases. El conjunt inicial de Dimensions que trobem després de definir el grànul ha de donar lloc a una Base. Substituint Dimensions en aquesta segons les dependències funcionals que hi ha amb la resta de Dimensions, podem obtenir les altres Bases de l'espai.

Exemple de Bases d'un espai

Ja hem vist abans que una Base de VendaAtòmica és {Producte, Dia, Client, Població}. En cas d'afegir la Dimensió Promoció, aquesta no formaria part de la Base, perquè queda completament determinada per les Dimensions Temps, Lloc i Producte (en un moment donat, en una certa població, un producte només es promociona d'una manera). Si pensem que una promoció determina un producte i que en tot moment tots els productes tenen una promoció o altra (podem considerar la no-promoció també com un tipus de promoció), llavors podem substituir Producte per Promoció i obtenir una segona Base de l'espai {Promoció, Dia, Client, Població}. La Base de Descomptes seria {Client, Setmana} i la de TancamentDeVendes seria {Població, Mes}.

3.1.9. Estudiar la viabilitat

Un cop ja hem acabat el disseny conceptual, cal veure si l'Estrella és realment implementable o no. S'ha d'estimar l'espai que ocuparan totes les dades. La manera de saber quant d'espai ocuparan és mirar el contingut dels sistemes operacionals. Si això és massa complicat, podem fer una estimació. Per a fer-ho, podem considerar només el que ocuparà emmagatzemar les instàncies del Fet. El volum de dades que puguin ocupar les Dimensions resulta en general insignificant respecte al que ocuparà el Fet.

Per a fer l'estimació del que caldrà per a emmagatzemar totes les instàncies d'una Cel·la, calculem la mida de l'espai que defineix cadascuna de les seves Bases, multiplicant el nombre d'instàncies de cada Nivell que les formen. L'espai més petit ens donarà el nombre màxim d'instàncies que pot tenir la Cel·la.

Exemple d'estimació d'instàncies

Pensem que tenim 1.000 clients i 1.000 productes diferents, venem a 10 poblacions i volem emmagatzemar dades durant 1.000 dies. A més, tenim una mitjana de tres promocions diferents per a cada producte (31.000 promocions en total). Si agafem la Base {Producte, Dia, Client, Població}, obtenim un espai de $1.000 \times 1.000 \times 1.000 \times 10$ (10^{10} possibles cel·les). Per contra, si considerem la Base {Promoció, Dia, Client, Població}, obtenim un espai de $3.000 \times 1.000 \times 1.000 \times 10$ ($3 \cdot 10^{10}$ possibles cel·les). Consegüentment, el nombre màxim de cel·les que podem tenir en el nostre espai és 10^{10} .

Aquesta fita pot resultar una mica excessiva. S'ha d'intentar d'afinar més, coneixent com de disperses queden en aquest espai les cel·les que hi volem analitzar.

Exemple de millora en la fita

Un client comprarà a totes les nostres botigues? La resposta, probablement, és que no. Comprarà només a les que tingui més a prop de casa. No es donaran totes les combinacions client-població i, si es donessin, no serien tots els dies. Podem considerar que un client només comprarà en una població. Així, tot i que l'espai sigui de $1.000 \times 1.000 \times 1.000 \times 10$, sabem que tindrem $1.000 \times 1.000 \times 1.000$ cel·les com a molt. Si a més estímul que la mitjana d'articles diferents que compra un client al dia és 10, ens queden aproximadament $10 \times 1000 \times 1000$ cel·les.

Un cop sabem quantes cel·les tindrem, multipliquem aquesta xifra pel nombre de bytes que ocuparà cada cel·la (6 bytes de les Mesures articles i ingressos més 20 bytes dels identificadors de les 5 Dimensions, en el nostre cas) i obtindrem una estimació del que ocuparà la nostra Cel·la (26×10^7 bytes = 26×10^4 kbytes = 260×10^1 Mbytes = 2,6 Gbytes). Generalment, el que ocupa la Cel·la

atòmica són ordres de magnitud més gran que el que ocupen les altres Cel·les (Descomptes ocuparia aproximadament 1,7 Mbytes i TancamentDeVendes només 4,3 kbytes). Només si tenim Dimensions molt petites o moltes Cel·les en el nostre esquema, realment cal sumar aquestes quantitats.

Per a saber quant d'espai ocuparà la nostra Estrella, el que és més realista és mirar les dades que contenen els sistemes operacionals (les nostres fonts de dades) per a saber quantes contindrà en el nostre sistema d'anàlisi. Si no és possible, també ho podem estimar a partir del nombre d'instàncies del Nivell atòmic de cada Dimensió i la grandària de la cel·la.

Serà capaç de manejar aquest volum de dades el nostre sistema? Serà prou bona la velocitat de resposta? Podem carregar dins la finestra d'actualització totes les dades necessàries per a mantenir-lo actualitzat? Si la resposta a qualsevol d'aquestes preguntes és que no, solament tenim dues opcions: 1) millorar el programari i maquinari que tenim o 2) replantejar el disseny triant un grànul no tan fi o descartant algunes Mesures que no siguin essencials.

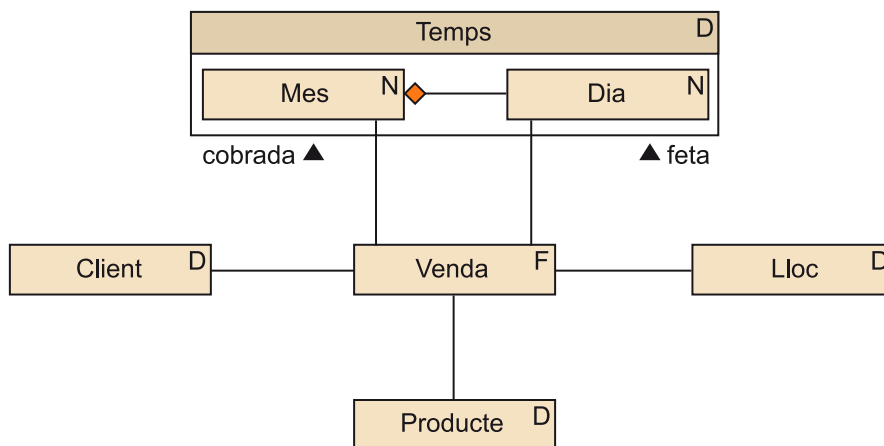
3.2. Reconsideracions en el disseny conceptual

En aquest apartat veurem algunes millores que, en alguns casos, es poden fer en el disseny d'una Estrella.

3.2.1. Dimensions amb múltiples rols

Fins ara, totes les Dimensions que hem utilitzat per a analitzar un Fet eren ben diferents. No obstant això, és força habitual que hi hagi Dimensions tan semblants que, de fet, hauríem de parlar de la mateixa Dimensió repetida, que té rols diferents. De vegades, d'una Dimensió a una altra, només canvia la jerarquia d'agregació que té definida o els Descriptors que ens són interessants. Normalment, ens quedarem amb una única Dimensió que associarem amb el Fet tants cops com calgui. No obstant això, en alguns casos, per exemple, per motius de confidencialitat o limitacions en l'eina OLAP, podria interessar més mantenir les Dimensions diferenciades.

Figura 32



Exemple de Dimensió amb rols múltiples

Si en el nostre sistema de vendes admetem el pagament endarrerit (a trenta, seixanta i noranta dies, per exemple), tindrem dues dates associades a cada venda, primer aquella en què s'ha produït la venda realment i després aquella en què hem cobrat. Podem representar això com veieu en la figura 32. Tant podria ser que totes dues associacions anessin a parar al mateix Nivell com a Nivells diferents. Si l'únic que ens interessés fos el mes en què hem cobrat, tindrem que el final de l'associació *cobrada* lliga amb el Nivell Mes en comptes de Dia, però la Dimensió encara és la mateixa.

Una Dimensió pot estar associada més d'un cop amb un mateix Fet i tenir rols diferents.

3.2.2. Dependències entre dimensions

El quocient entre el nombre màxim teòric de cel·les que hi pot arribar a haver i les cel·les que hi ha realment és la Dispersió del cub.

Una dispersió molt gran pot generar problemes de gestió dels cubs. La causa de la dispersió és que certes combinacions d'instàncies de les Dimensions no són vàlides. El possible valor d'una de les Dimensions depèn dels valors de les altres. Com més depenguin unes Dimensions de les altres, més alta serà la dispersió.

Si el valor d'una Dimensió depèn del valor d'una altra, diem que estan correlacionades.

Quan veiem que dues Dimensions estan correlacionades, les podem ajuntar per a obtenir una única Dimensió. Aquesta nova Dimensió tindrà una instància per a cada element del producte cartesià de les Dimensions originals que

realment tingui sentit. Tot i que som capaços de representar la mateixa informació, en fer això reduïm el nombre de Dimensions, però com a contrapartida les que queden són més grans.

La correlació entre dues Dimensions no ve d'aquestes, sinó del conjunt de dades que analitzem. En una Estrella podem veure que dues Dimensions estan molt correlacionades, mentre que en una altra no ho estan gens.

Figura 33

	Blau	Vermell	Groc		
Cotxe	X	X		Cotxe blau	X
Camió	X			Cotxe vermell	X
Tractor			X	Camió blau	X
				Tractor groc	X
Dues dimensions				Una dimensió	
Dispersió 9/4				Dispersió 4/4	

Exemple de Dimensions correlacionades

Imaginem-nos que parlem d'un concessionari de vehicles, que està interessat a analitzar les vendes segons el tipus de vehicle i els colors. Aquest concessionari és molt particular, ja que, com teniu esquematitzat en la figura 33, només ven cotxes vermells o blaus, camions blaus i tractors grocs. El color depèn molt del tipus de vehicle. Per tant, si les seves Dimensions d'anàlisi són Vehicle i Color, té un espai de mida nou que només contindrà quatre cel·les. Per contra, si tinguéssim una única Dimensió VehicleAcolorit, que només té instàncies per a les parelles vehicle-color que realment hi ha, la dispersió seria mínima (hi hauria una cel·la a cada punt de l'espai). Abans teníem dues Dimensions de mida tres i ara només en tenim una, però de mida quatre.

Hem de fusionar Dimensions només quan estiguin realment correlacionades i ens representi un guany en la dispersió del cub sense que el nombre d'instàncies de les Dimensions creixi excessivament. Penseu que els usuaris utilitzaran les Dimensions per a seleccionar les dades que volen veure. Com més instàncies tinguin les Dimensions i més complexos siguin els conceptes que representen, més difícil els resultarà fer seleccions.

Exemple de Dimensions que no interessa fusionar

En la nostra Estrella de vendes, hem vist que molt probablement un client sempre compraria a la població on resideix i que això generava certa dispersió en el cub. No obstant això, si ajuntem les Dimensions Client i Població no obtenim cap concepte concret. Cap analista no voldrà consultar parelles client-població. Voldrà consultar les vendes a un client i les vendes en una població majoritàriament per separat. A més, cap client no té prohibit comprar a una població. Consegüentment, si les poblacions són prou properes, és possible que l'espai Client × Població no sigui gaire dispers i tot client hagi comprat alguna vegada a cadascuna de les poblacions. La Dimensió ClientPoblació tindrà una instància per gairebé cada element del producte cartesià entre Client i Població.

Si volem reduir la dispersió, el que hem de fer és fusionar les Dimensions que estiguin molt correlacionades. Això augmentarà la mida de les Dimensions, però reduirà la dispersió del cub.

3.2.3. Minidimensions

En alguns casos, podem observar que una Dimensió és massa gran. D'una banda, això pot provocar dispersió en el cub, si no és que el Fet també té moltes instàncies; i de l'altra, dificulta les consultes. Per a solucionar-ho, podem crear una Dimensió més petita només amb els atributs més utilitzats. Sovint, amb això no n'hi ha prou i el que hem de fer és, a més, modificar el domini d'algun d'aquests atributs perquè l'atribut tingui menys valors possibles.

Tenir aquesta minidimensió no implica eliminar la Dimensió original. Poden coexistir totes dues dins la mateixa Estrella, però han d'aparèixer lligades per una associació.

Exemple de minidimensió demogràfica

Algunes empreses tenen milions de clients (penseu, per exemple, en telefonia). En aquests casos, la Dimensió Client és excessivament gran i dificulta clarament les consultes. Per a solucionar aquest problema, es defineixen perfils de clients. S'agafen només els atributs més utilitzats en les consultes (per exemple, edat, sexe, estat civil, nivell adquisitiu, etc.) i es crea una Dimensió demogràfica amb aquests. Si es manté la Dimensió Client al costat de la demogràfica, s'ha de definir una associació entre aquestes que lligui cada client amb el seu perfil. Si aquesta Dimensió demogràfica encara té massa instàncies, podem crear rangs de valors més petits en algun dels atributs. Per exemple, podem parlar només de tres valors de l'atribut edat: edats entre 0 i 30, 31 i 60 i més de 60.

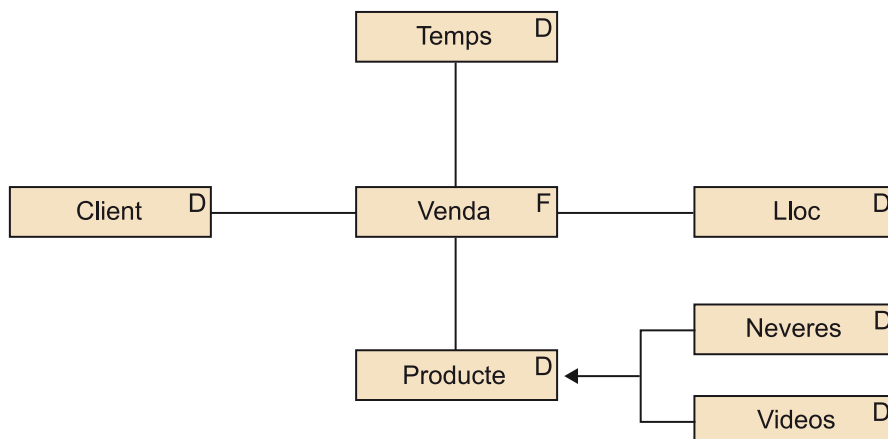
Quan tenim Dimensions massa grans podem crear minidimensions amb els atributs més utilitzats per a facilitar les consultes dels usuaris.

Adoneu-vos que el nombre d'instàncies d'una minidimensió (com ara la demogràfica) no depèn del nombre d'instàncies de la Dimensió (per exemple, Clients), sinó del nombre d'atributs que té i la grandària dels seus dominis. Si veiem que tenim molts atributs per a fer una minidimensió, en podem fer més d'una. Si portem això a l'extrem, podem fer una minidimensió per a cada atribut de la Dimensió original.

3.2.4. Disseny de dades heterogènies

Les instàncies d'algunes Dimensions (com ara Productes) no són gaire homogènies. Els atributs que valen per a un producte no tenen sentit per a un altre (per exemple, les neveres tenen litres de capacitat, mentre que els aparells de vídeo tenen sistema d'enregistrament). Això es tradueix en l'aparició de molts valors nuls (que dificulten les consultes) i la impossibilitat de definir certes jerarquies d'agregació perquè no són vàlides per a totes les instàncies.

Figura 34

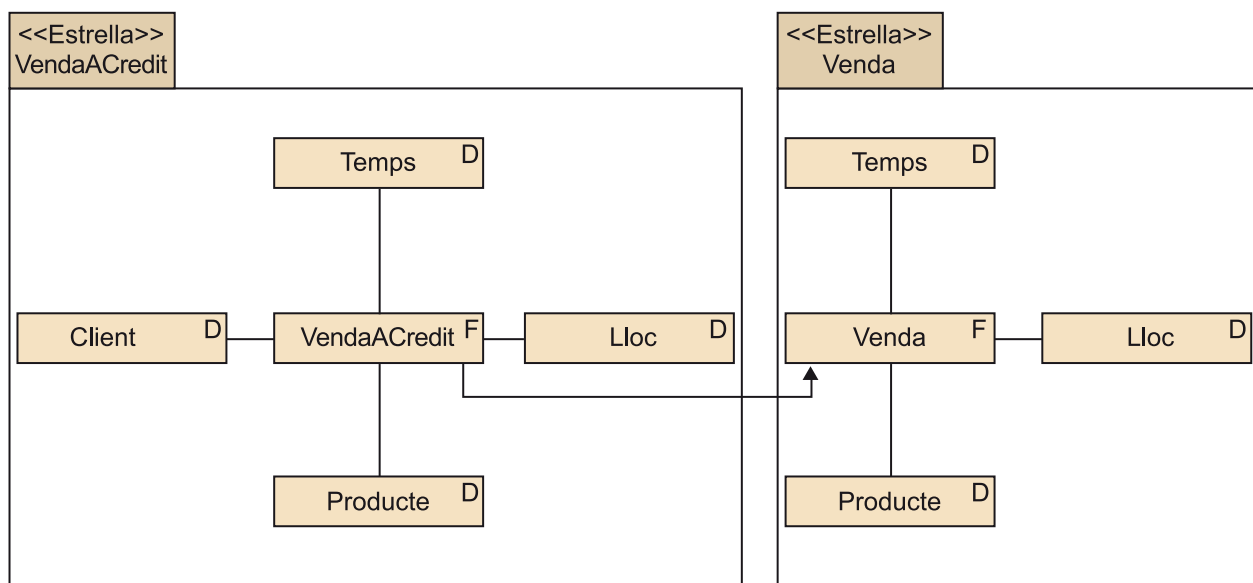


La manera de solucionar aquest problema és especialitzant les Dimensions (com podeu veure dibuixat en la figura 34). D'una banda, tenim la Dimensió general que té els Descriptors i jerarquia comuns a totes les instàncies. De l'altra, definim Dimensions més petites amb Descriptors i jerarquies més específiques. Els usuaris podran utilitzar la Dimensió que més els interressi en cada moment per a consultar el Fet.

El mateix pot ocórrer amb els Fets. Alguns esdeveniments tindran més atributs o, fins i tot, més Dimensions que uns altres. De la mateixa manera que en el cas de les Dimensions, el que hem de fer és especialitzar el Fet.

Si només canvia el nombre d'atributs, podem mantenir una única Estrella amb més d'un Fet. Per contra, si segons el tipus de Fet tenim més o menys Dimensions, val més definir una Estrella Diferent.

Figura 35



Exemple d'especialització d'un Fet

En la figura 35 podem veure dues Estrelles en què els Fets respectius estan relacionats per una especialització. Mentre que si tenim una venda a crèdit podem identificar el client, quan el pagament és en efectiu aquesta identificació no és possible. Consegüentment, la Dimensió *Client* no és present en l'Estrella per a estudiar les vendes en general.

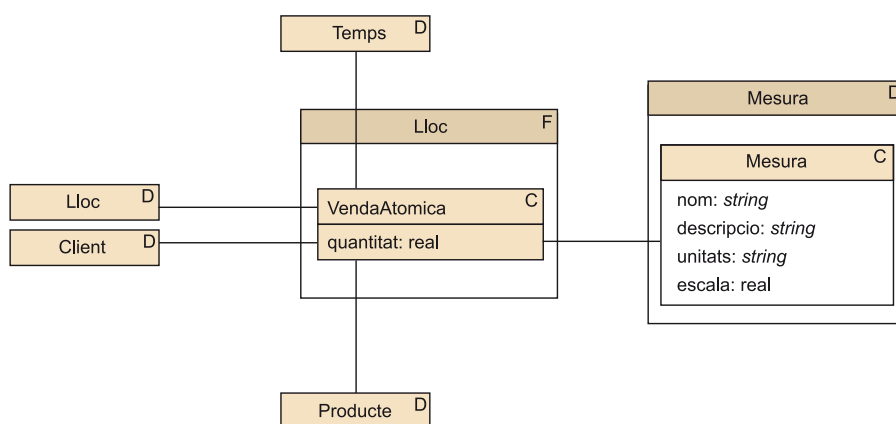
Quan les instàncies són heterogènies, podem especialitzar tant els Fets com les Dimensions per a evitar l'aparició de Descriptors i Mesures invàlides, o facilitar la definició de jerarquies d'agregació o Dimensions específiques.

3.2.5. Fets amb una única mesura

Un altre cas d'heterogeneïtat de les instàncies es dona quan en cada esdeveniment prenem diferents mesures. Per a poder especialitzar, cal poder distingir tipus diferents de Fets i que les Mesures siguin específiques de cada tipus concret. Però què podem fer quan les mesures que prenem depenen de l'esdeveniment que es dona i no podem definir tipus d'esdeveniments?

El que és millor en aquests casos és definir una Dimensió *Mesura*. El Fet tindrà una única Mesura, que podem anomenar *quantitat*. Segons a quina instància de *Mesura* associem una instància del Fet, *quantitat* tindrà un significat o un altre. A més d'ajudar a solucionar problemes de diferències en les Mesures de cada esdeveniment, aquest tipus de disseny significa que les Mesures ens interessin de manera individual i no totes alhora (amb un disseny com aquest, no hauria de ser freqüent que consultéssim tots els mesuraments que vam fer quan es donava un esdeveniment).

Figura 36



Exemple de Fet amb una Mesura

En la figura 36, podeu veure com hauríem de modificar la nostra Estrella de vendes perquè contingui una única Mesura. Aprofitant que generem una nova Dimensió, podem afegir Descriptors de les mesures com ara les unitats, l'escala, etc.

Com a opció de disseny, podem substituir totes les Mesures d'un Fet per una d'única, afegint una nova Dimensió Mesura.

4. Disseny lògic

Un cop ja hem vist com es fa un disseny conceptual multidimensional, ara veurem com es passa al model lògic. En el mercat hi ha diferents tipus d'implementacions multidimensionals: ROLAP, MOLAP o, fins i tot, O³LAP. A causa de la implantació actual dels sistemes relacionals, que cobreixen el mercat, l'opció més comuna és la implementació ROLAP.

ROLAP, MOLAP, O³LAP

Els tres conceptes provenen de l'anglès, ROLAP és *relational OLAP*, MOLAP és *multidimensional OLAP* i O³LAP *object-oriented OLAP*.

Una eina ROLAP no és res més que una capa de programari que rep consultes multidimensionals, les tradueix a SQL i les executa sobre un SGBD relacional. Ara veurem com s'implementa un esquema multidimensional sobre les taules d'un SGBD d'aquest tipus.

4.1. L'Estrella (el cas bàsic)

Igual que fèiem en el disseny conceptual, per passar al model lògic, ens fixem també en una Estrella cada cop. A més, en aquest apartat pensem que el Fet conté una única Cel·la. Per a implementar una Estrella necessitem una taula per al Fet (en què cada fila representa una cel·la de l'espai multidimensional) i una taula més per a cadascuna de les Dimensions. Les jerarquies d'agregació queden implícites en els valors dels atributs de les taules de Dimensió. No els explicitem amb taules diferents.

Exemple de taula de Dimensió amb jerarquia d'agregació implícita

La Dimensió Temps podria estar implementada amb la relació següent:

Temps(RowID, diaMesAny, mesAny, setmanaAny, any, festiu, laborable, traspas,...)

Adoneu-vos que aquesta relació no està normalitzada. L'atribut *diaMesAny* determina tant *mesAny* com *setmanaAny*. A més, qualsevol d'aquests dos atributs determina *any*. D'aquesta manera, sabem a quin mes, setmana i any pertany cadascun dels dies sense necessitat de definir la jerarquia d'agregació. Les agrupacions possibles de les files queden determinades pels valors dels mateixos atributs, en comptes de per una jerarquia explícita. Totes les files amb el mateix valor a l'atribut *any* s'agrupen per donar lloc a una instància a Nivell Any.

Taula de Dimensió

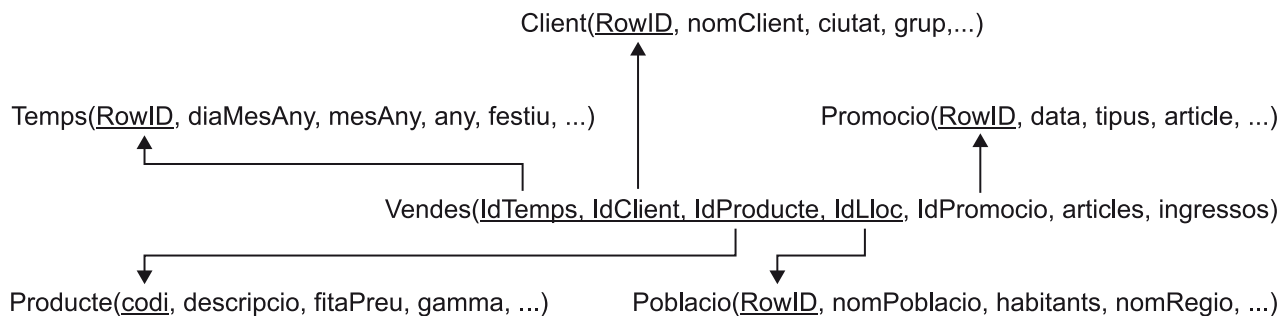
El terme anglès per a aquesta estructura de taules i claus foranes és *star join*.

La taula de Fet estarà lligada per claus foranes amb les taules de Dimensió.

Cada clau forana apunta de la taula del Fet cap a una de les taules de les Dimensions. Al costat de les claus foranes, la taula del Fet conté les Mesures, mentre que les taules de Dimensió contenen els Descriptors. Si el Fet no conté Mesures, la taula del Fet només contindrà les claus foranes cap a les taules de Dimensió.

Com a clau primària de la taula del Fet tindrem els atributs corresponents a una de les Bases de la Cel·la atòmica. La resta de Bases de la Cel·la donaran lloc a claus alternatives. En qualsevol cas, tant la clau primària com les alternatives seran subconjunts del conjunt de claus foranes que apunten cap a les taules de Dimensió.

Figura 37



Exemple d'esquema relacional amb forma d'estrella

En el cas de les vendes i les promocions, obtindríem l'esquema relacional de la figura 37. Podem veure que hi ha sis taules: una per a cada Dimensió (Temps, Producte, Poblacio, Promocio i Client) i una més per al Fet (Vendes). Aquestes taules estan lligades per cinc claus foranes, cadascuna de les quals va de la taula del Fet a la clau primària (el codi de producte per a la taula Producte o el RowID per a la resta de taules) d'una de les taules de les Dimensions. Quatre d'aquestes claus foranes formen la clau primària de la taula del Fet. Si considerem que {Promocio, Client, Dia, Poblacio} és una Base, llavors {IDPromocio, IDClient, IDTemps i IDLloc} seria clau alternativa de la taula Vendes. Els atributs articles i ingressos, les Mesures (que mai no són clau forana), no formen part de la clau primària de Vendes.

Nota

Recordeu que marquem la clau primària d'una relació subratllant els atributs que la formen.

Cada Estrella dóna lloc a una taula per al Fet i una més per a cadascuna de les Dimensions. Aquestes taules estan lligades per claus foranes que van de la taula del Fet a cadascuna de les taules de Dimensió. La clau primària de la taula del Fet és la concatenació de les claus foranes corresponents a una Base del Fet.

Adoneu-vos de la grandària de cadascuna de les taules. La taula del Fet serà d'ordres de magnitud més gran que qualsevol de les taules de Dimensió. Ocuparà més d'un 95% de l'espai utilitzat per l'Estrella. A primer cop d'ull pot semblar estrany, però una manera de reduir la grandària de la taula del Fet és definir substituïts de la clau primària en les taules de Dimensió. Com que els substituïts ocuparan menys espai que els atributs identificadors de la mateixa taula (per exemple, un DNI ocupa vuit caràcters, mentre que un RowID només quatre bytes), utilitzant-los reduïm la grandària de les columnes que formen la clau primària de les taules de Dimensió. Conseqüentment, també reduïm la grandària de la clau primària de la taula del Fet, perquè sabem que sempre està formada per claus foranes que apunten a les claus primàries de les taules de Dimensió. Un petit guany en cada fila de la taula del Fet, que n'acostuma a tenir milions, pot representar un gran estalvi d'espai.

Clau primària d'una taula

Recordeu que, com a clau primària d'una taula, no solament podem tenir atributs, sinó també substituïts (*surrogates*), coneguts també com a RowID. Vegeu el mòdul "Reconsideració dels models conceptual i lògic" de l'assignatura *Sistemes de gestió de bases de dades*.

L'efecte col·lateral de la utilització de substituïts de la clau primària és que prevenim possibles problemes davant de canvis en els atributs identificadors en els sistemes operacionals. Recordeu que volem analitzar llargs períodes de temps. Per tant, és altament probable que durant aquest període es produeixin recodificacions dels identificadors de les promocions, de les poblacions o dels clients, amb la qual cosa també hauríem de recodificar totes les dades que tinguem en la nostra Estrella. Si utilitzem substituïts per a la clau primària, ja no tenim aquest problema, perquè l'identificador és absolutament independent de l'origen de les dades i qualsevol canvi que es produeixi no l'afectarà.

Definim substituïts de la clau primària en les taules de Dimensió per a reduir la grandària de la taula del Fet. A més a més, també serveix per a evitar problemes si es modifiquen els identificadors en els sistemes operacionals.

A més de la grandària, també hi ha diferències en les operacions que fem en cada tipus de taula. Les taules de Dimensió es creen amb les dades a dins i molt rarament canvien. Només de tant en tant s'afegeix una nova fila o es canvia el valor d'una que ja hi havia (mai no s'esborren files). En la taula de Fets inserim files massivament de manera regular i només té modificacions si hem comès un error durant la inserció (tampoc no s'esborra mai res).

4.2. El floc de neu

Ja hem vist com es pot tractar el cas en què només tenim una Cel·la, però què cal fer quan en tenim més d'una dins el mateix Fet? De la mateixa manera que abans, definim una taula de Fet per a cada Cel·la, amb les corresponents claus foranes cap a les taules de Dimensió. Respecte a les taules de Dimensió, ara sí que cal explicitar els Nivells (normalitzar-les), tot i que només parcialment.

Adoneu-vos que, si normalitzem totalment una taula de Dimensió, obtindrem una taula diferent per a cadascun dels Nivells. Fer-ho per a totes les Dimensions genera un esquema amb forma de floc de neu. El guany d'espai respecte de l'espai total de l'Estrella no resultaria significatiu, perquè les taules de Dimensió són insignificants respecte del volum de dades que conté la taula del Fet. Per contra, empitjoraria el rendiment de les consultes, perquè cada cop que volguéssim operar amb una Dimensió hauríem de fer la combinació (*join*) de les taules de Dimensió corresponents als Nivells.

La normalització és una eina del món transaccional que evita les redundàncies i facilita la concurrència en presència d'actualitzacions, però augmenta el nombre de combinacions necessàries per a resoldre algunes consultes. Hem de pensar que l'objectiu d'un esquema multidimensional és respondre consultes

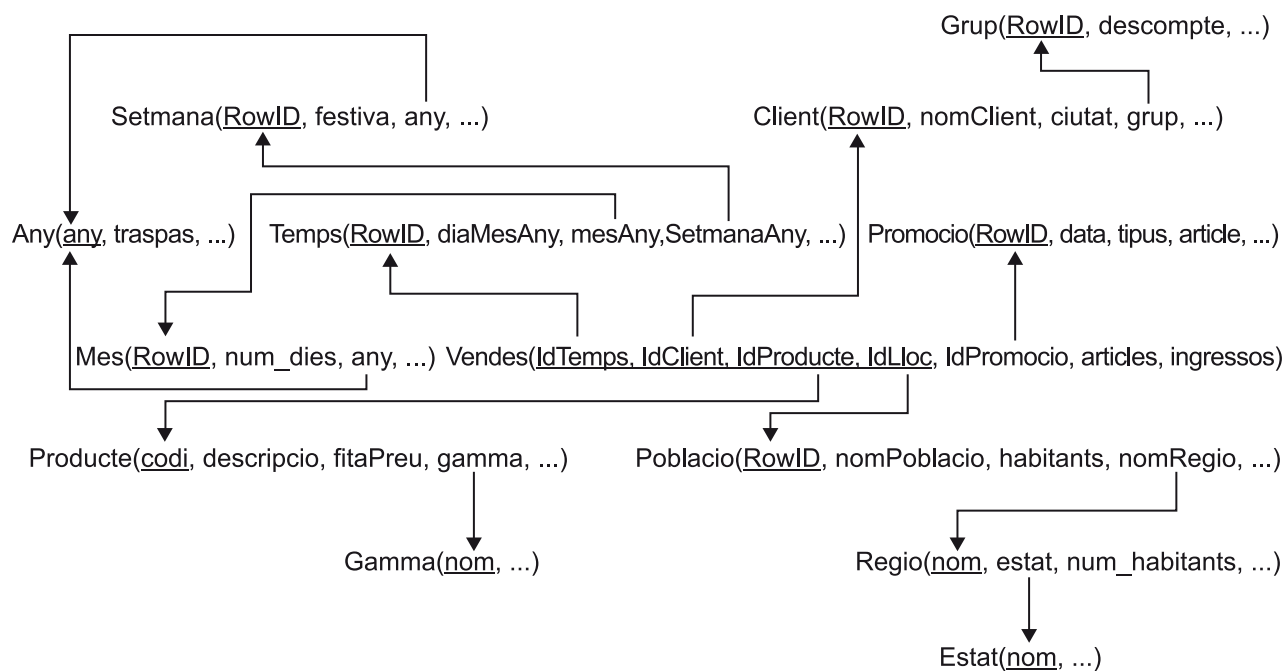
Estrella normalitzada

Una estrella completament normalitzada es coneix amb el nom de *floc de neu* (*snowflake*), per la seva forma d'estrella amb ramificacions, que recorda l'estructura d'un floc de neu.

de manera eficient (com ja hem dit, no hi ha actualitzacions) i resoldre combinacions no és gens ràpid. Amb les Dimensions desnormalitzades, podem millorar el rendiment d'algunes consultes fins a un 30%.

En la figura següent podeu veure com quedaria l'esquema un cop normalitzat. Podeu apreciar que, a més dels problemes ja mencionats, ara l'esquema és més difícil d'entendre (sobretot per a usuaris no experts). Dissortadament, a més de ser més difícil d'entendre per als usuaris, també ho és per a l'optimitzador de consultes. Alguns reconeixen una estrella i utilitzen tècniques específiques per a la resolució de consultes, però no n'hi ha cap que reconegui un floc de neu.

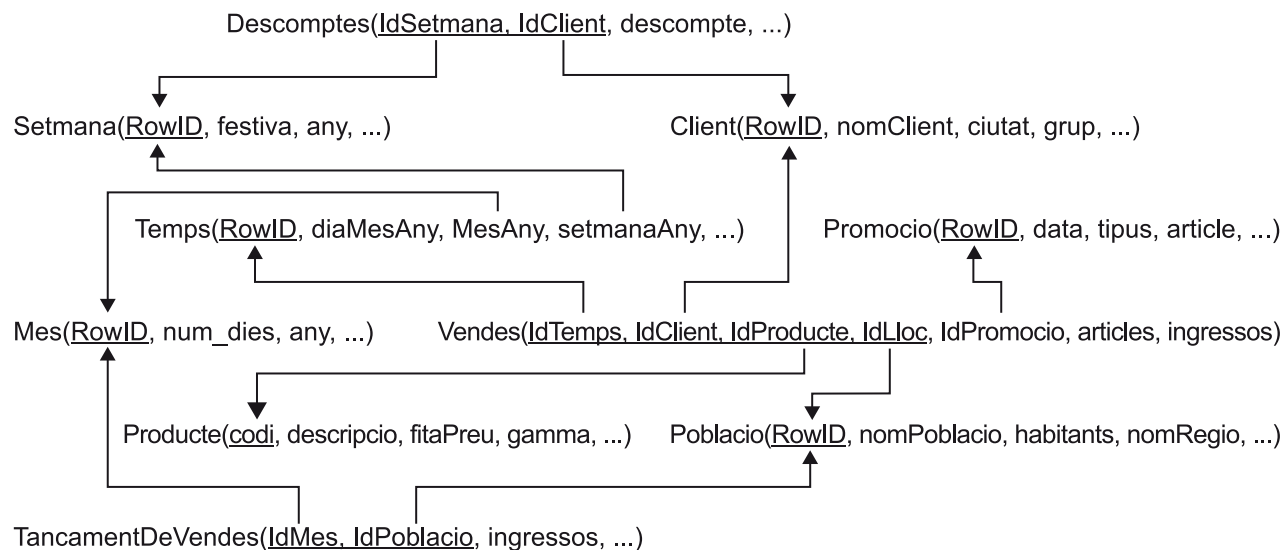
Figura 38



El floc de neu és un error de disseny que genera un guany inapreciable d'espai i una gran pèrdua de rendiment. Per a millorar el rendiment de les consultes, hem d'evitar normalitzar els esquemes, llevat de casos extrems en què la grandària de la Dimensió sigui comparable a la del Fet.

Només hem de definir taules de Dimensió per als Nivells que tenen alguna Cel·la associada. D'aquesta manera, les claus foranes de la taula del Fet apunten a la granularitat correcta. No pot ser que les Mesures corresponguin a dades mensuals i la clau forana de l'associació amb la Dimensió Temps apunti a una taula que conté informació de dies.

Figura 39



Exemple de pas a relacional d'un Fet amb múltiples Cel·les

En la figura 39 podeu veure com quedaria un esquema relacional quan tenim diferents Cel·les en un mateix Fet. En aquest cas, tenim tres taules de Fet (*Descomptes*, *Vendes* i *TancamentDeVendes*). Cadascuna d'aquestes apunta a les seves taules de Dimensió. Us heu de fixar especialment en la normalització parcial que ha tingut la taula *Temps*. Ara hi ha dues taules més (*Setmana* i *Mes*) relacionades amb aquesta mitjançant claus foranes. Adoneu-vos que la informació de l'any ja no és en la taula *Temps*, sinó en les taules *Setmana* i *Mes* (repetida en les dues). No s'ha acabat de normalitzar perquè no hi ha cap Cel·la a Nivell Any.

Només normalitzarem una Dimensió quan el Fet contingui Cel·les diferents i sigui necessari fer-ho per a relacionar la taula de Fet corresponent a cada Cel·la amb la granularitat adequada de la Dimensió.

4.3. Conformació: compartició de dimensions

Generalment, cada Estrella té les seves Dimensions. Però és habitual que certes Dimensions s'utilitzin en diferents Estrelles (per exemple, la Dimensió *Temps* o *Clients*). Cal consensuar entre els analistes implicats quin ha de ser el contingut i format d'aquestes Dimensions compartides. Això s'anomena *conformar les Dimensions*. D'aquesta manera, podrem utilitzar en diferents Estrelles taules de Dimensió que tinguin la mateixa forma, tot i que puguin estar implemen-tades sobre SGBD diferents o, fins i tot, sobre màquines diferents.

Conformar

Conformar vol dir 'fer alguna cosa conforme a una norma'.

Conformar les Dimensions és extremadament important en el disseny de les Estrelles, per permetre de navegar de l'una a l'altra, i canviar el tema objecte d'anàlisi. S'han de conformar les Dimensions que participin en Estrelles diferents per a fer possible el *Drill-across*.

Avantatges de tenir les Dimensions conformades:

- Possibilita el *Drill-across*.
- Estalvia feina de disseny i administració, perquè la mateixa taula s'utilitza més d'un cop. Si implementéssim totes les Estrelles dins un mateix SGBD, fins i tot podríem tenir únicament una taula per a cada Dimensió, que seria compartida per tantes Estrelles com calgués.
- Ajuda a donar la idea d'una factoria d'informació empresarial, en comptes de petits magatzems de dades departamentals aïllats. Tots els usuaris disposaran dels mateixos Descriptors i jerarquies d'agregació.

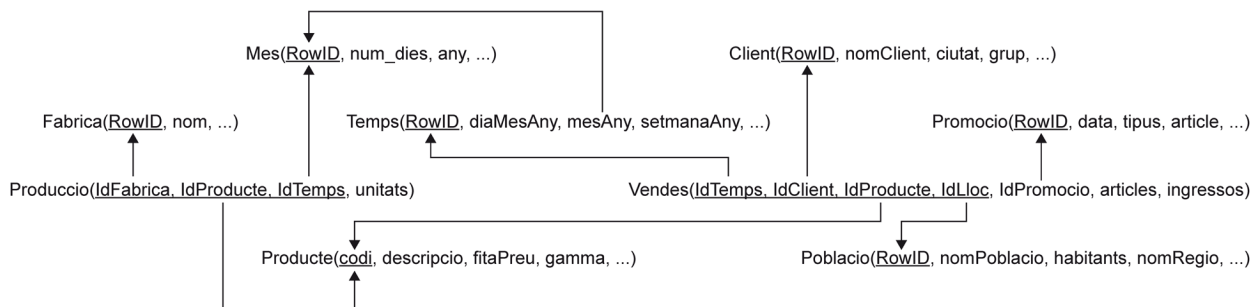
Conformació de Dimensions

Un bon indicador per a trobar dues Dimensions que s'han de conformar és que es diguin igual. Però aneu amb compte amb això, perquè, per exemple, la Dimensió temporal de vendes no es pot conformar amb la Dimensió temporal fiscal, la qual no ve marcada per les estacions o dies de festa, sinó pels tancaments de comptes i dates de declaració d'impostos.

Una Dimensió conformada és la que s'utilitza en la implementació de més d'una Estrella. A l'hora de fer consultes, taules de Fet que comparteixen taules de Dimensió es poden substituir les unes per les altres (*Drill-across*) sense modificar la resta de la consulta.

Compartir una Dimensió no vol dir que dues Estrelles hagin d'utilitzar exactament la mateixa taula de Dimensió. També es pot fer si una Estrella utilitza una taula que representa les dades de la Dimensió a un cert grau d'agregació i una altra Estrella utilitza una altra taula de Dimensió que representa un grau d'agregació superior. Si aquest és el cas, totes dues taules formen part de la mateixa Dimensió.

Figura 40



Exemple de compartició de taules de Dimensió

En la figura 40 podem veure com *Vendes* i *Produccio* comparteixen Dimensions. Primerament, la taula de Dimensió *Producte* és apuntada per totes dues taules de Fet. Consegüentment, cal haver conformat la Dimensió *Producte* de les dues Estrelles. A més a més, però, també tenen en comú la Dimensió *Temps*. En aquest cas, cada Estrella la utilitza amb una granularitat diferent. *Produccio* la utilitza amb granularitat *Mes*, mentre que *Vendes* la utilitza amb granularitat *Dia*. Cal, doncs, haver conformat també aquesta Dimensió.

Conformar les Dimensions no implica que implementem totes les Estrelles en el mateix SGBD, ni tan sols ho hem de fer en la mateixa màquina. El que és més important és que les taules que implementin la mateixa Dimensió en Estrelles diferents tinguin la mateixa forma (els mateixos atributs i dominis semàntics) per a permetre de passar d'una taula de Fet a l'altra. Si aquestes taules es fusionen en una única taula o tenim una taula diferent en cada SGBD o en cada màquina és un tema que no hem de decidir quan es fa el disseny lògic.

4.4. Dimensions degenerades, de deixalles i ombres

En alguns casos podem estar davant de Dimensions que no tenen atributs ni jerarquies d'agregació, però que ens són útils simplement per a identificar les instàncies del Fet. Parlem en aquests casos de Dimensions degenerades, perquè no donen lloc a cap taula de Dimensió. Simplement tenim un atribut en la taula de Fet, que forma part de la clau, però que no és clau forana (no apunta a cap taula de Dimensió).

Exemple de Dimensió degenerada

L'exemple més habitual de Dimensió degenerada és el número que identifica la comanda. Tot i no tenir cap atribut, ens és útil per a identificar les vendes i agrupar totes les que s'han fet dins la mateixa comanda. Aquesta Dimensió *Comanda* no té cap atribut, perquè els hi hem tret tots (data, client, etc.) per a definir altres Dimensions d'anàlisi.

Les Dimensions degenerades acostumen a tenir el seu origen en atributs identificadors de les bases de dades operacionals que coincideixen amb el gràndul triat per l'Estrella en qüestió. És important conservar-los perquè permeten de saber exactament d'on provenen les dades.

Una Dimensió degenerada és aquella que no dóna lloc a una taula de Dimensió per manca d'atributs, però que sí que s'utilitza per a identificar les instàncies del Fet.

Un altre cas interessant es dóna a l'inrevés: quan hi ha atributs que descriuen el Fet, però que no corresponen a cap concepte en concret que ajudi a identificar-lo. Què hem de fer quan tenim un conjunt d'atributs (sovint booleans), que provenen dels sistemes operacionals, independents els uns dels altres? En

aquest cas cal crear una taula de Dimensió auxiliar que contingui tots aquests atributs, que no identificaran en cap cas les instàncies del Fet, però que utilitzarem per a seleccionar-les i agregar-les.

D'aquest tipus de taula de Dimensió, en direm **Dimensió de deixalles**. També lligarem aquest tipus de taules a la taula del Fet amb una clau forana, però en aquest cas no formarà part de la clau primària.

Per a contenir tots aquells atributs que provenen dels sistemes operacionals que no corresponguin a cap Dimensió en concret crearem una taula de Dimensió especial, la clau de la qual no formarà part de la clau primària del Fet.

En les Dimensions també hi ha atributs que molt probablement no ens servirán per a seleccionar, ni per a definir jerarquies d'agregació (com ara l'adreça, el número de telèfon o simplement un comentari a la Dimensió Client), però que potser voldrem afegir a la consulta per a generar l'informe final just abans d'imprimir-lo. Per no afectar el temps de resposta, podem posar tots aquests atributs poc utilitzats en una altra taula, que anomenarem **Dimensió ombra**.

Aquesta Dimensió ombra no estarà lligada directament a la taula de Fet, sinó que tindrem en la Dimensió de debò una clau forana que l'apuntarà. D'aquesta manera només caldrà accedir a aquests atributs (que habitualment ocupen molt d'espai) per a imprimir l'informe final i no afectaran el temps de resposta durant la navegació per les dades.

Una Dimensió ombra és aquella taula que conté atributs descriptius poc utilitzats i que no lliguem directament a la taula de Fet, sinó a una altra taula de Dimensió.

4.5. Generalitzacions/especialitzacions

Durant el disseny conceptual, quan hi ha instàncies heterogènies, ho representem amb una especialització del Fet o de la Dimensió, segons sigui el cas. Recordeu les tres opcions que tenim per a implementar aquest tipus de lligam entre classes:

- 1) Una taula per a la superclasse i una altra per a cada subclasse, totes amb la mateixa clau primària.
- 2) Una única taula que contingui tant els atributs de la superclasse com els de totes les subclasses.

3) Només una taula per a cadascuna de les subclasses, repetint els atributs de la superclasse en cada taula.

En el disseny de bases de dades operacionals, la millor opció és la primera (tot i que en algun cas molt especial podria interessar més alguna de les altres dues, com ja va estudiar en el seu moment). No obstant això, en el cas del disseny multidimensional no és així.

Si l'especialització és d'una Dimensió (com, per exemple, la de la figura 34), cal considerar si les taules estaran referenciades per alguna taula de Fet o no. De la mateixa manera que no normalitzem per evitar d'haver de fer combinacions, ara només separarem les subclasses i superclasse en taules diferents quan sigui estrictament necessari.

Si la taula corresponent a la superclasse no és apuntada per una clau forana des de cap Fet i les corresponents a les subclasses sí que ho són, el que és millor és la tercera opció, amb la qual ens estalviarem combinacions. De la mateixa manera, si les taules corresponents a les subclasses no són apuntades per cap taula de Fet, la millor opció serà la segona. Per exemple, en el cas de la figura 34, si les Dimensions *Neveres* i *Vídeos* no s'utilitzen en cap altra Estrella, seria millor tenir únicament la taula de Dimensió corresponent a *Producte*, que en aquesta taula també inclou els atributs específics de neveres i vídeos.

Si l'especialització és d'una Dimensió, la implementarem amb una taula de Dimensió per a cadascuna de les classes que hi ha associades a algun Fet.

Si el que hi ha especialitzat és un Fet (com teniu dibuixat en la figura 35), atesa la quantitat de files que té una taula de Fet, la segona opció queda descartada directament per la quantitat de valors nuls que genera (amb el consegüent malbaratament d'espai i temps de consulta).

En canvi, tot i que pugui malbaratar una mica d'espai, la tercera opció serà la millor, si cada analista està interessat en una única subclasse i vol tractar junts tant els atributs propis de la subclasse com els heretats de la superclasse. Amb aquesta opció, obtindríem diferents taules de Fet (relativament petites) per a cada analista, amb el consegüent guany de rendiment. Fer la unió de totes aquestes per a obtenir la superclasse sempre és possible. Però cal dir que si normalment no es vol accedir alhora als atributs de la superclasse i de les subclasses, la millor opció per a implementar una especialització d'un Fet és la primera (una taula de Fet per a cada subclasse i una altra per a la superclasse).

Si l'especialització és d'un Fet, hem de mirar si s'accedeix als atributs de superclasse i subclasses alhora o no. En cas que la majoria de vegades s'hi accedeixi alhora, és millor que només tinguem taules per a les subclasses. Quan és més habitual accedir-hi per separat, llavors és millor tenir els atributs de la superclasse en una taula diferent.

4.6. Estructures temporals

Ja hem dit que la Dimensió temporal és pràcticament en totes les Estrelles. Els Fets estan associats a la Dimensió temporal precisament perquè registra l'evolució del negoci, els seus canvis. Cada cop que es produeix un esdeveniment, afegim una nova instància del Fet. Aquestes instàncies corresponen a successos concrets (per exemple, vendes i moviments en el compte corrent) o a estats (per exemple, estocs i saldos).

Vist d'una altra manera, hi ha dues possibilitats de registrar l'evolució temporal: guardar les transaccions (diferencials) o guardar successions de fotografies. En el primer cas, l'esdeveniment que cal registrar seria donat per una certa acció en el món real de la qual guardem el valor o valors que porta associats. En el segon cas, cada cert temps registrem l'estat del nostre Fet. D'una banda, podem registrar cada venda concreta i, de l'altra, podem registrar en un moment donat quant s'ha venut.

Tots dos mecanismes ens permeten de mostrar l'evolució del negoci, tot i que el que es basa en transaccions és molt més detallat. A partir d'aquest podem construir les fotografies, però no a l'inrevés. Tot i així, els dos mecanismes són necessaris. Les transaccions ens donen el màxim grau de detall, però les fotografies ens permeten de saber ràpidament l'estat de l'empresa en un moment donat.

Hi ha dues maneres d'enregistrar els fets. Ho podem fer mitjançant transaccions entre estats o mitjançant fotografies de l'estat en un cert moment.

Els Fets sempre registren els canvis que es produeixen en el negoci. I les Dimensions, mai no registren cap canvi? No es modifiquen mai? Diu una màxima budista que l'única constant és que tot canvia. Per tant, no ens hem de preguntar si les Dimensions canvien, sinó amb quina freqüència ho fan.

Efectivament, les Dimensions canvien molt poc, però també tenen alguns canvis (obrirem noves botigues, canviarem el descompte que fem habitualment a un bon client, deixarem de vendre algun producte, etc.). Si els canvis són realment poc freqüents, una solució possible és definir una Estrella diferent cada cop que es produeixi un canvi en una de les seves Dimensions. Una al-

tra solució quan la Dimensió que canvia no és gaire gran és definir diferents versions de la taula de Dimensió i lligar-les totes al mateix cub. En aquest cas, l'usuari (o la interfície) ha de saber quina taula ha d'utilitzar per a resoldre la consulta en cada moment. La solució més bona i més general, però, és enregistrar els canvis dins la mateixa taula de Dimensió afectada.

En els sistemes operacionals, registrem un canvi simplement modificant el registre que toca. En un sistema d'anàlisi, això normalment no es pot fer així. Imaginem-nos que canvia el preu d'un producte. Si senzillament canviem el valor de l'atribut `preu` en la taula de Dimensió, semblarà que totes les vendes s'han fet amb aquest preu. No podrem veure la modificació que es produeix en el volum de vendes segons quin sigui el preu del producte.

Quan veiem que una Dimensió canvia, tenim tres opcions:

1) Sobreescriure el valor antic amb el nou (com es fa en els sistemes operacionals). Adoneu-vos que si fem això, renunciem a estudiar com afecta aquest canvi al Fet. D'altra banda, simplifica molt el tractament i no utilitza més espai del que és estrictament necessari. No obstant això, aquesta opció només és aconsellable per als casos en què descobrim que havíem comès una errada en la introducció de dades i la volem esmenar sense deixar-ne constància. També ho és quan l'atribut en concret no té cap incidència en l'anàlisi (per exemple, el nom d'una companyia). Si té incidència en l'anàlisi, perquè defineix una característica utilitzada en la selecció o agrupació d'instàncies (com ara els habitants d'una població), aquesta opció no és gens apropiada (podríem seleccionar fets en una població amb un cert nombre d'habitants en un moment que encara no els tenia).

2) Crear una nova fila en la taula de Dimensió (amb el seu *RowID*) que serà referenciada des d'ara per totes les instàncies de Fet que afegim. Aquesta és la solució més habitual. Adoneu-vos que implica la utilització d'un substitut de la clau primària en la taula de Dimensió. Com podríem identificar la segona fila dins la taula? Si utilitzéssim el DNI com a clau primària, no podríem registrar d'aquesta manera els canvis en les dades d'una persona, perquè totes dues files haurien de tenir el mateix DNI.

Una altra raó per a utilitzar substituïts de la clau primària en les taules de Dimensió és poder registrar els canvis sense tenir problemes d'identificació.

Fixeu-vos que amb aquesta solució el que fem realment és crear una nova instància de la Dimensió. En certa manera diem que hi ha dues persones diferents, una abans del canvi i una altra després. Per a fer consultes, no cal ser conscient del canvi. La mateixa condició ja indicarà quina de les dues files es tria i, per tant, en quins fets estem interessats.

Un problema d'aquesta opció és que no serà evident que dues files fan referència a la mateixa instància en moments diferents. Això es pot solucionar mantenint (a més del *RowID*) un atribut identificador (com ara el DNI) i afegir un atribut més que sigui el número de versió de la instància. Tindrem tantes versions d'una instància com canvis hi hagi en els seus atributs.

3) Tenir diferents atributs en la taula de Dimensió per a registrar el valor antic i el nou. És a dir, per a cada atribut que pugui tenir un canvi, ara utilitzem dos atributs. Aquesta solució, com ja us podeu imaginar, és molt limitada i útil només en casos molt concrets. Primerament, cal que el canvi sigui general en totes les files de la taula. Però, a més, només funciona si hi ha un únic canvi. Si n'hi hagués dos, caldrien tres atributs; si n'hi hagués tres, quatre atributs, etc. Una implementació com aquesta pot ser útil, per exemple, per a canviar els noms de les poblacions del castellà al català (o del castellà a l'euskera). Les poblacions són exactament les mateixes (no generem nous *RowID*). Simplement, volem utilitzar la nova denominació i no volem perdre la denominació castellana. Amb aquesta opció, en alguns casos, a més de l'atribut amb el valor antic, també cal un altre atribut amb la data del canvi.

Bàsicament hi ha tres possibilitats per a registrar els canvis en les taules de Dimensió:

- Modificar el valor directament.
- Crear una nova fila.
- Crear una nova columna.

Recordeu les minidimensions de les quals parlàvem en l'apartat de disseny conceptual. Fixeu-vos que definíem perfils d'usuari en comptes d'usuaris pròpiament. En aquest tipus de Dimensió, no reflectirem els canvis de cap d'aquestes tres maneres. Si canvia un dels atributs de l'usuari, simplement associarem els seus esdeveniments amb un perfil diferent. Però la taula de Dimensió no tindrà cap modificació si no és que aquest canvi genera un nou perfil d'usuari, que no hi havia abans. Si aquest és el cas, simplement l'afegirem ara a la taula.

Aquestes minidimensions també ajuden a gestionar els canvis en les Dimensions. Normalment, tenim atributs que canvien més sovint que uns altres. L'edat o els ingressos anuals tenen una freqüència de canvi molt més alta que la població de residència, el nom o el sexe. Per no haver d'afegir una nova fila

en la taula Client cada cop que canviïn aquests atributs, els podem ficar dins una minidimensió demogràfica, en què ja hem dit que no cal afegir noves files cada cop que es produeix un canvi, sinó simplement lligar cada instància del Fet amb el perfil corresponent.

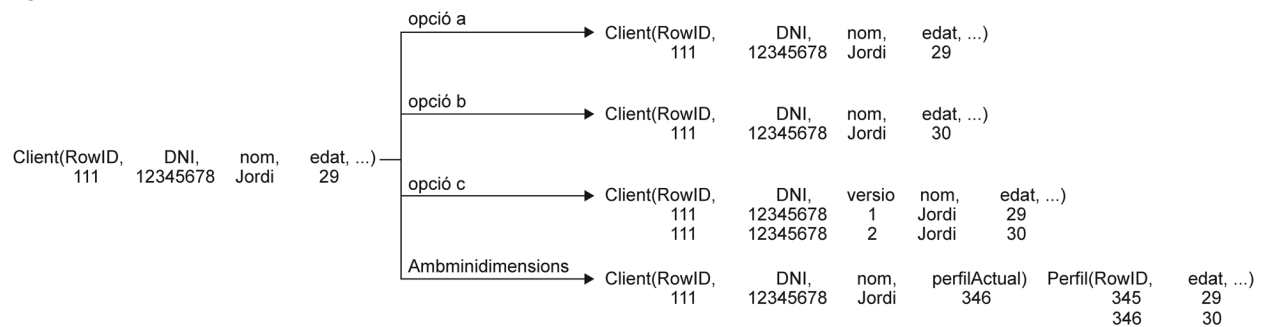
Els canvis en les minidimensions són més fàcils de gestionar que en les Dimensions normals, perquè no cal modificar-les. En alguns casos, crear-les pot ser una bona opció per a registrar els canvis.

Generalment, podeu distribuir els atributs de qualsevol Dimensió segons la freqüència de canvi, per a facilitar-ne la gestió. D'aquesta manera, tindríeu una taula de Dimensió que no canvia gairebé mai i una altra que canvia cada cert temps.

Exemple de gestió dels canvis en les Dimensions

En la figura següent podeu veure les quatre maneres de reflectir el canvi d'edat del Jordi dels vint-i-nou als trenta anys. En el cas d'utilitzar minidimensions, la taula de Fet tindria una clau forana cap a Perfil, per a deixar constància del perfil que tenia en Jordi en el moment en què es va produir l'esdeveniment.

Figura 41



5. Consultes amb SQL'99

Ara que ja hem vist com s'implementen les estrelles en un SGBD relacional, ve-gem com les podem consultar amb SQL estàndard. A més de veure l'estructura bàsica d'una consulta, també veurem les clàusules especials que incorpora l'especificació de l'estàndard de 1999 (SQL'99).

5.1. Estructura bàsica de la consulta

El que tenim en un esquema amb forma d'estrella és una taula de Fet i una taula per a cadascuna de les Dimensions. Per a fer una consulta, hem de posar totes aquestes taules en la clàusula FROM. A WHERE lligarem cada taula de Dimensió amb la taula de Fet utilitzant la clau forana corresponent (mai no lligarem Dimensions entre si) i afegirem les condicions que volem sobre els atributs de les taules de Dimensió. A SELECT posarem les Mesures de la taula de Fet que vulguem visualitzar amb la corresponent operació d'agregació. Finalment, posarem la clàusula GROUP BY amb els identificadors dels Nivells als quals volem veure les dades en cada Dimensió. Sempre afegirem els atributs que apareguin a GROUP BY a SELECT per a poder distingir les files. Habitualment, també s'afegeix la clàusula ORDER BY amb tots els atributs de les taules de Dimensió que tenim a SELECT, per a obtenir el resultat ordenat i facilitar-ne la visualització.

```
SELECT Nivell1, ..., Nivelln, Operacio(Mesura), ...  
FROM Fet, Dimensio1, ..., Dimension  
WHERE Fet=Dimensio1 AND ... AND Fet=Dimension  
AND Descriptor1=valor AND ... AND Descriptorm=valor  
GROUP BY Nivell1, ..., Nivelln  
ORDER BY Nivell1, ..., Nivelln
```

Cadascuna de les operacions del model multidimensional que hem vist en l'apartat corresponent té una relació directa amb aquesta estructura de consulta:

- Selecció: per a seleccionar uns punts o uns altres del nostre espai n -dimensional, el que hem de fer és afegir o treure condicions sobre els Descriptors en la clàusula WHERE.
- Projecció: traient-les de SELECT deixem de veure les Mesures que no ens interessin.

- *Roll-up*: per a augmentar o disminuir el nivell de detall amb què veiem les dades, només cal agrupar segons l'atribut que identifica Nivell o un altre de la Dimensió que correspon. Cal posar els atributs corresponents a GROUP BY. En cas de voler fer *Roll-up* fins al Nivell All, el que cal és treure tots els atributs de la Dimensió.
- *Drill-across*: si volem canviar el tema d'anàlisi, només cal substituir la taula de Fet en la clàusula FROM (recordeu que per a poder-ho fer cal que els dos Fets comparteixin Dimensions). Les Dimensions que no hi ha en el nou Fet desapareixeran.
- CanviBase: finalment, si el que volem és veure les mateixes dades ordenades de manera diferent, només cal modificar els atributs de SELECT i canviar l'ordre dels atributs a ORDER BY.

Exemple de seqüència d'operacions sobre una consulta SQL

Donada la consulta:

```
SELECT d1.nom_article, d2.nom_fabrica, d3.mesAny, SUM(fet.unitats)
FROM Produccio fet, Producte d1, Fabrica d2, Temps d3
WHERE fet.IDProducte=d1.ID
      AND fet.IDFabrica=d2.ID
      AND fet.IDTemps=d3.ID
      AND d1.nom_article IN ('Bolígrafs', 'Gomes')
      AND d2.num_treballadors>100
      AND d3.mesAny IN ('Gener2002','Febrer2002')
GROUP BY d1.nom_article, d2.nom_fabrica, d3.mesAny
ORDER BY d1.nom_article, d2.nom_fabrica, d3.mesAny;
```

$A := \text{roll-up}_{Fabrica:All}(\text{"Unitats produïdes per Producte, Fabrica i Mes"})$

```
SELECT d1.nom_article, 'All', d3.mesAny, SUM(fet.unitats)
FROM Produccio fet, Producte d1, Fabrica d2, Temps d3
WHERE fet.IDProducte=d1.ID
      AND fet.IDFabrica=d2.ID
      AND fet.IDTemps=d3.ID
      AND d1.nom_article IN ('Bolígrafs', 'Gomes')
      AND d2.num_treballadors>100
      AND d3.mesAny IN ('Gener2002','Febrer2002')
GROUP BY d1.nom_article, d3.mesAny
ORDER BY d1.nom_article, d3.mesAny;
```

$B := \text{canviBase}_{Producte \times Temps}(A)$

```
SELECT d1.nom_article, d3.mesAny, SUM(fet.unitats)
FROM Produccio fet, Producte d1, Fabrica d2, Temps d3
WHERE fet.IDProducte=d1.ID AND fet.IDTemps=d3.ID
      AND fet.IDFabrica=d2.ID
      AND d1.nom_article IN ('Bolígrafs', 'Gomes')
      AND d2.num_treballadors>100
      AND d3.mesAny IN ('Gener2002','Febrer2002')
GROUP BY d1.nom_article, d3.mesAny
ORDER BY d1.nom_article, d3.mesAny;
```

$C := \text{drill-across}_{Vendes}(B)$

```
SELECT d1.nom_article, d3.mesAny, SUM(fet.articles),
      SUM(fet.ingressos)
FROM Vendes fet, Producte d1, Temps d3
WHERE fet.IDProducte=d1.ID
      AND fet.IDTemps=d3.ID
      AND d1.nom_article IN ('Bolígrafs', 'Gomes')
      AND d3.mesAny IN ('Gener2002','Febrer2002')
```

```
GROUP BY d1.nom_article, d3.mesAny
ORDER BY d1.nom_article, d3.mesAny;
```

$D := \text{projecció}_{\text{articles}}(C)$

```
SELECT d1.nom_article, d3.mesAny, SUM(fet.articles)
FROM Vendes fet, Producte d1, Temps d3
WHERE fet.IDProducte=d1.ID
      AND fet.IDTemps=d3.ID
      AND d1.nom_article IN ('Bolígrafs', 'Gomes')
      AND d3.mesAny IN ('Gener2002', 'Febrer2002')
GROUP BY d1.nom_article, d3.mesAny
ORDER BY d1.nom_article, d3.mesAny;
```

$E := \text{canviBase}_{\text{Articles} \times \text{Lloc} \times \text{Temps}}(D)$

```
SELECT d1.nom_article, 'All', d3.mesAny, SUM(fet.articles)
FROM Vendes fet, Producte d1, Lloc d2, Temps d3
WHERE fet.IDProducte=d1.ID
      AND fet.IDLloc=d2.ID
      AND fet.IDTemps=d3.ID
      AND d1.nom_article IN ('Bolígrafs', 'Gomes')
      AND d3.mesAny IN ('Gener2002', 'Febrer2002')
GROUP BY d1.nom_article, d3.mesAny
ORDER BY d1.nom_article, d3.mesAny;
```

$F := \text{drill-down}_{\text{Lloc::Regió}}(E)$

```
SELECT d1.nom_article, d2.regio, d3.mesAny, SUM(fet.articles)
FROM Vendes fet, Producte d1, Lloc d2, Temps d3
WHERE fet.IDProducte=d1.ID
      AND fet.IDLloc=d2.ID
      AND fet.IDTemps=d3.ID
      AND d1.nom_article IN ('Bolígrafs', 'Gomes')
      AND d3.mesAny IN ('Gener2002', 'Febrer2002')
GROUP BY d1.nom_article, d2.regio, d3.mesAny
ORDER BY d1.nom_article, d2.regio, d3.mesAny;
```

$R := \text{selecció}_{\text{Regio.nom}=\text{'Catalunya'}}(F)$

```
SELECT d1.nom_article, d2.regio, d3.mesAny, SUM(fet.articles)
FROM Vendes fet, Producte d1, Lloc d2, Temps d3
WHERE fet.IDProducte=d1.ID
      AND fet.IDLloc=d2.ID AND fet.IDTemps=d3.ID
      AND d1.nom_article IN ('Bolígrafs', 'Gomes')
      AND d2.regio='Catalunya'
      AND d3.mesAny IN ('Gener2002', 'Febrer2002')
GROUP BY d1.nom_article, d2.regio, d3.mesAny
ORDER BY d1.nom_article, d2.regio, d3.mesAny;
```

5.2. GROUPING SETS

El que és més habitual és que els usuaris no solament vulguin veure unes certes dades, sinó també els totals. És a dir, el resultat d'una consulta seria una taula com aquesta:

Vendes	Catalunya		
	Gener 2002	Febrer 2002	Total
Bolígrafs	275.827 (a)	290.918 (a)	566.745 (c)
Gomes	784.172 (a)	918.012 (a)	1.702.184 (c)
Total	105.999 (b)	1.208.930 (b)	2.268.929 (d)

Aquesta taula no és pròpiament un cub, perquè barreja cel·les de quatre granularitats diferents (les (a), les (b), les (c) i la (d)). Tot i això la podem entendre com la unió de quatre cubs. De la mateixa manera que podem definir una funció per trossos, també podem definir la taula per trossos unint cubs.

Vendes (a) $\oplus$ *Roll-up*_{Temps::All}(*Vendes*) (b) $\oplus$ *Roll-up*_{Articles::All}(*Vendes*) (c) $\oplus$ *Roll-up*_{Articles::All, Temps::All}(*Vendes*) (d)

Adoneu-vos que la consulta SQL que hem vist abans només ens mostra les quatre cel·les (a). Per a aconseguir les altres cinc cel·les caldria unir quatre consultes, com es fa en la consulta següent:

(a)

```
SELECT d1.nom_article, d2.regio, d3.mesAny, SUM(fet.articles)
FROM Vendes fet, Producte d1, Lloc d2, Temps d3
WHERE fet.IDProducte=d1.ID
      AND fet.IDLloc=d2.ID
      AND fet.IDTemps=d3.ID
      AND d1.nom_article IN ('Bolígrafs','Gomes')
      AND d2.regio='Catalunya'
      AND d3.mesAny IN ('Gener2002','Febrer2002')
GROUP BY d1.nom_article, d2.regio, d3.mesAny
UNION
```

(b)

```
SELECT d1.nom_article, d2.regio, 'Total', SUM(fet.articles)
FROM Vendes fet, Producte d1, Lloc d2, Temps d3
WHERE fet.IDProducte=d1.ID
      AND fet.IDLloc=d2.ID
      AND fet.IDTemps=d3.ID
      AND d1.nom_article IN ('Bolígrafs','Gomes')
      AND d2.regio='Catalunya'
      AND d3.mesAny IN ('Gener2002','Febrer2002')
GROUP BY d1.nom_article, d2.regio
UNION
```

(c)

```
SELECT 'Total', d2.regio, d3.mesAny, SUM(fet.articles)
FROM Vendes fet, Producte d1, Lloc d2, Temps d3
WHERE fet.IDProducte=d1.ID
      AND fet.IDLloc=d2.ID
      AND fet.IDTemps=d3.ID
      AND d1.nom_article IN ('Bolígrafs','Gomes')
      AND d2.regio='Catalunya'
      AND d3.mesAny IN ('Gener2002','Febrer2002')
GROUP BY d2.regio, d3.mesAny
UNION
```

(d)

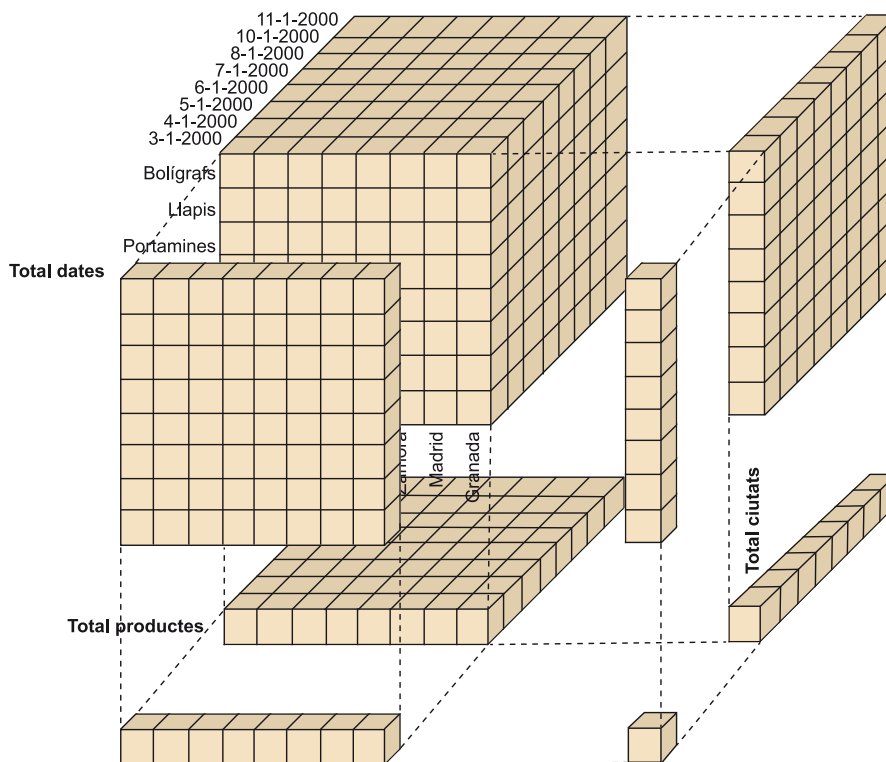
```

SELECT 'Total', d2.regio, 'Total', SUM(fet.articles)
FROM Vendes fet, Producte d1, Lloc d2, Temps d3
WHERE fet.IDProducte=d1.ID
      AND fet.IDLloc=d2.ID
      AND fet.IDTemps=d3.ID
      AND d1.nom_article IN ('Bolígrafs','Gomes')
      AND d2.regio='Catalunya'
      AND d3.mesAny IN ('Gener2002','Febrer2002')
GROUP BY d2.regio
ORDER BY d1.nom_article, d2.regio, d3.mesAny;

```

Si en comptes de calcular el total per a dues Dimensions ho volguéssim fer per a tres, ens caldrien set unions. La figura següent us mostra esquematitzats quins són els vuit cubs que caldria unir (vendes per dia, producte i ciutat; per dia i producte; per dia i ciutat; per producte i ciutat; per dia; per producte; per ciutat, i el total de vendes).

Figura 42



El nombre d'unions que s'han de fer per a calcular els totals creix exponencialment respecte al nombre de Dimensions que tinguem. Per sort, l'estàndard SQL'99 ja ens ofereix una altra sintaxi per a abreviar totes aquestes unions. Com podeu veure en aquesta consulta, només cal posar a GROUP BY les paraules clau 'GROUPING SETS' i entre parèntesis la llista d'agrupacions que es vol fer.

```
SELECT d1.nom_article, d2.regio, d3.mesAny, SUM(fet.articles)
FROM Vendes fet, Producte d1, Lloc d2, Temps d3
WHERE fet.IDProdcute=d1.ID
      AND fet.IDLloc=d2.ID
      AND fet.IDTemps=d3.ID
      AND d1.nom_article IN ('Bolígrafs', 'Gomes')
      AND d2.regio='Catalunya'
      AND d3.mesAny IN ('Gener2002','Febrer2002')
GROUP BY GROUPING SETS ((d1.nom_article, d2.regio, d3.mesAny),
                          (d1.nom_article, d2.regio),
                          (d2.regio, d3.mesAny),
                          (d2.regio))
ORDER BY d1.nom_article, d2.regio, d3.mesAny;
```

Per a calcular diferents agrupacions de la mateixa consulta sense haver d'explicitar les unions, l'estàndard SQL'99 ens ofereix les paraules reservades GROUPING SETS dins la clàusula GROUP BY.

El resultat de la consulta anterior seria aquest:

Nom_article	Regio	MesAny	Articles
Bolígrafs (a)	Catalunya (a)	Gener02 (a)	275827 (a)
Bolígrafs (a)	Catalunya (a)	Febrer02 (a)	290918 (a)
Bolígrafs (c)	Catalunya (c)	NULL (c)	566745 (c)
Gomes (a)	Catalunya (a)	Gener02 (a)	784172 (a)
Gomes (a)	Catalunya (a)	Febrer02 (a)	918012 (a)
Gomes (c)	Catalunya (c)	NULL (c)	1702184 (c)
NULL (b)	Catalunya (b)	Gener02 (b)	1059999 (b)
NULL (b)	Catalunya (b)	Febrer02 (b)	1208930 (b)
NULL (d)	Catalunya (d)	NULL (d)	2268929 (d)

A part del problema evident de la presentació, que ha d'arreglar la mateixa interfície gràfica, fixeu-vos en els valors nuls. Què volen dir, 'desconegut' o 'inaplicable'? Doncs, cap de les dues opcions. Aquest és un tercer significat dels valors nuls que s'ha definit en l'SQL'99. En aquesta taula volen dir 'total'.

El valor nul no solament vol dir 'desconegut' o 'inaplicable'. En el context de les GROUPING SETS, també pot voler dir 'total'.

Això planteja un nou problema. Com sabem si un valor nul vol dir que no coneixem el valor de l'atribut o que la fila és el resultat d'aplicar una operació d'agregació? Per a contestar aquesta pregunta, l'estàndard defineix la funció GROUPING.

La funció GROUPING té com a paràmetre un atribut. Retorna "1" si aquest atribut pren el valor 'total'. En qualsevol altre cas retorna "0". Amb aquesta funció, podem rescriure la consulta anterior de manera que aparegui la paraula *total* en comptes dels valors nuls sempre que convingui (com d2.regio apareix en totes quatre agrupacions, sabem que no pot prendre el valor 'total').

```
SELECT
    CASE WHEN GROUPING(d1.nom_article)=1 THEN 'TotalDeGenerIFebrer'
    ELSE d1.nom_article,
    d2.regio,
    CASE WHEN GROUPING(d3.mesAny)=1 THEN 'TotalDeBoligrafsIGomes'
    ELSE d3.mesAny,
    SUM(fet.articles)
FROM Vendes fet, Producte d1, Lloc d2, Temps d3
WHERE fet.IDProdcute=d1.ID
    AND fet.IDLloc=d2.ID
    AND fet.IDTemps=d3.ID
    AND d1.nom_article IN ('Boligrafs', 'Gomes')
    AND d2.regio='Catalunya'
    AND d3.mesAny IN ('Gener2002', 'Febrer2002')
GROUP BY GROUPING SETS ((d1.nom_article, d2.regio, d3.mesAny),
    (d1.nom_article, d2.regio),
    (d2.regio, d3.mesAny),
    (d2.regio))
ORDER BY d1.nom_article, d2.regio, d3.mesAny;
```

Ara, el resultat de la consulta seria aquest:

Nom_article	Regio	mesAny	Articles
Bolígrafs (a)	Catalunya (a)	Gener02 (a)	275827 (a)
Bolígrafs (a)	Catalunya (a)	Febrer02 (a)	290918 (a)
Bolígrafs (c)	Catalunya (c)	TotalDeGenerIFebrer (c)	566745 (c)
Gomes (a)	Catalunya (a)	Gener02 (a)	784172 (a)
Gomes (a)	Catalunya (a)	Febrer02 (a)	918012 (a)
Gomes (c)	Catalunya (c)	TotalDeGenerIFebrer (c)	1702184 (c)
TotalDeBolígrafsIGomes (b)	Catalunya (b)	Gener02 (b)	1059999 (b)
TotalDeBolígrafsIGomes (b)	Catalunya (b)	Febrer02 (b)	1208930 (b)
TotalDeBolígrafsIGomes (d)	Catalunya (d)	TotalDeGenerIFebrer (d)	2268929 (d)

Podeu distingir el significat d'un valor nul utilitzant la funció GROUPING.

5.2.1. ROLLUP

Com ja hem dit, el nombre de totals creix exponencialment respecte al nombre de Dimensions. Conseqüentment, encara que no hàgim d'escriure tota la consulta, només escriure totes les combinacions d'atributs a GROUPING SETS ja pot arribar a resultar massa complex. Per a facilitar encara més aquest tipus de consulta, l'estàndard també defineix la paraula reservada ROLLUP. Amb un cert conjunt d'atributs, calcula totes les agrupacions que resulten d'anar ignorant els atributs d'un en un, de dreta a esquerra. Vegem com s'incorporaria a la consulta anterior (pareu atenció a l'ordre dels atributs a GROUP BY i ORDER BY).

L'ordre dels atributs

L'ordre dels atributs d'agrupació dins de ROLLUP sí que afecta el resultat de la consulta. L'ordre dels atributs dins d'ORDER BY no afecta el resultat de la consulta.

```
SELECT d1.nom_article, d2.regio, d3.mesAny, SUM(fet.articles)<br/>
FROM Vendes fet, Producte d1, Lloc d2, Temps d3<br/>
WHERE fet.IDProdcute=d1.ID
      AND fet.IDLloc=d2.ID
      AND fet.IDTemps=d3.ID<br/>
      AND d1.nom_article IN ('Bolígrafs', 'Gomes')
      AND d2.regio='Catalunya'<br/>
      AND d3.mesAny IN ('Gener2002', 'Febrer2002')<br/>
GROUP BY ROLLUP (d2.regio, d1.nom_article, d3.mesAny);<br/>
ORDER BY d2.regio, d3.mesAny, d1.nom_article;
```

El resultat d'aquesta consulta seria el següent:

nom_article	regio	mesAny	articles
Bolígrafs (a)	Catalunya (a)	Gener02 (a)	275.827 (a)
Gomes (a)	Catalunya (a)	Gener02 (a)	784.172 (a)
Bolígrafs (a)	Catalunya (a)	Febrer02 (a)	290.918 (a)
Gomes (a)	Catalunya (a)	Febrer02 (a)	918.012 (a)
Bolígrafs (c)	Catalunya (c)	NULL (c)	566.745 (c)
Gomes (c)	Catalunya (c)	NULL (c)	1.702.184 (c)
NULL (d)	Catalunya (d)	NULL (d)	2.268.929 (d)
NULL (d)	NULL (d)	NULL (d)	2.268.929 (d)

Les primeres quatre files corresponen a "GROUP BY d2.regio, d1.nom_article, d3.mesAny"; les dues següents a "GROUP BY d2.regio, d1.nom_article"; la següent a "GROUP BY d2.regio"; i la darrera a "GROUP BY ()". Com només tenim un valor per a d2.regio, les dues últimes files tenen el mateix valor. Podem evitar que surti la darrera fixant aquest atribut i indicant que s'utilitzi en totes les agrupacions.

```
SELECT d1.nom_article, d2.regio, d3.mesAny, SUM(fet.articles)
FROM Vendes fet, Producte d1, Lloc d2, Temps d3
WHERE fet.IDProdcute=d1.ID
      AND fet.IDLloc=d2.ID
      AND fet.IDTemps=d3.ID
      AND d1.nom_article IN ('Bolígrafs', 'Gomes')
      AND d2.regio='Catalunya'
      AND d3.mesAny IN ('Gener2002', 'Febrer2002')
GROUP BY d2.regio, ROLL-UP (d1.nom_article, d3.mesAny);
ORDER BY d2.regio, d3.mesAny, d1.nom_article;
```

El resultat d'aquesta consulta és el mateix d'abans, menys la darrera fila. Amb això, podem rescriure la consulta original de quatre agrupacions de la manera següent:

```
SELECT d1.nom_article, d2.regio, d3.mesAny, SUM(fet.articles)
FROM Vendes fet, Producte d1, Lloc d2, Temps d3
WHERE fet.IDProdcute=d1.ID
      AND fet.IDLloc=d2.ID
      AND fet.IDTemps=d3.ID
      AND d1.nom_article IN ('Bolígrafs', 'Gomes')
      AND d2.regio='Catalunya'
      AND d3.mesAny IN ('Gener2002', 'Febrer2002')
GROUP BY GROUPING SETS
      ((d2.regio, ROLL-UP (d1.nom_article, d3.mesAny)),
      (d2.regio, d3.mesAny))
ORDER BY d1.nom_article, d2.regio, d3.mesAny;
```

```
GROUP BY ROLL-UP (a1,...,an
```

equivale a:

```
GROUP BY GROUPING SETS ((a1,...,an),
      (a1,...,an-1),
      ...
      (a1),
      ())
```

Nova sintaxi

```
SELECT COUNT(*)
FROM taula;
```

Amb SQL'99, ara també es pot escriure:

```
SELECT COUNT(*)
FROM taula
GROUP BY ();
```

5.2.2. CUBE

Amb ROLLUP ja no cal que escriguem totes les combinacions d'atributs que ens interessin, però encara n'hem d'escriure algunes. Les podem aconseguir totes directament si utilitzem la paraula reservada CUBE en comptes de ROLLUP. Amb la consulta següent obtenim les vuit cel·les que volíem originalment:

```
SELECT d1.nom_article, d2.regio, d3.mesAny, SUM(fet.articles)
FROM Vendes fet, Producte d1, Lloc d2, Temps d3
WHERE fet.IDProdcute=d1.ID
      AND fet.IDLloc=d2.ID
      AND fet.IDTemps=d3.ID
      AND d1.nom_article IN ('Bolígrafs', 'Gomes')
      AND d2.regio='Catalunya'
      AND d3.mesAny IN ('Gener2002', 'Febrer2002')
GROUP BY d2.regio, CUBE (d1.nom_article, d3.mesAny);
ORDER BY d1.nom_article, d2.regio, d3.mesAny;
```

```
GROUP BY CUBE (a,b,c)
```

equivale a:

```
GROUP BY GROUPING SETS ( (a,b,c),
                          (a,b),
                          (a,c),
                          (b,c),
                          (a),
                          (b),
                          (c),
                          ());
```

També podem combinar CUBE i ROLLUP per a obtenir els totals que ens interessin.

```
GROUP BY CUBE (a,b), ROLLUP (c,d)
```

és equivalent a:

```
GROUP BY GROUPING SETS ( (a,b,c,d),
                          (a,b,c),
                          (a,b),
                          (a,c,d),
                          (a,c),
                          (a),
                          (b,c,d),
                          (b,c),
                          (b),
                          (c,d),
                          (c),
                          ());
```

L'estàndard permet de combinar CUBE, ROLLUP i GROUPING SETS de qualsevol manera per a obtenir el resultat volgut.

Utilitzar CUBE, ROLLUP i GROUPING SETS, a més de facilitar l'escriptura de les consultes, també millora el rendiment del sistema perquè dona informació extra a l'optimitzador de consultes sobre el que es pretén fer.

6. Disseny físic

Un cop ja tenim les relacions que componen la nostra base de dades i també sabem quin tipus de consultes volem executar, només queda fer el disseny físic tenint al cap que cal obtenir un bon temps de resposta a les consultes.

6.1. Pla i tècniques bàsiques d'accés

Abans de veure quines eines específiques tenim des del punt de vista físic per a millorar el rendiment del sistema, penseu com seria el pla d'accés d'una consulta multidimensional (aquest pla pot tenir petites variacions segons l'SGBD):

- 1) Avaluar les condicions sobre cadascuna de les Dimensions per a obtenir un conjunt d'identificadors.
- 2) Combinar (fer el producte cartesià) els identificadors de totes les Dimensions per a obtenir els identificadors del Fet que ens interessin.
- 3) Cercar el Fet per a obtenir els valors (mesuraments) que volíem.
- 4) Ordenar els resultats.
- 5) Agrupar i operar mesuraments.

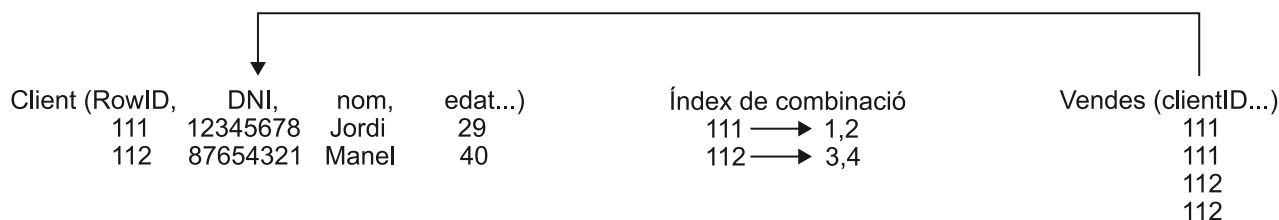
SGBD

Cal assegurar-se que l'SGBD reconeix l'esquema amb forma d'estrella i utilitza aquest pla d'accés.

En el segon pas del pla d'accés que acabem de veure, es fa el producte cartesià de tots els identificadors seleccionats en les Dimensions. Tot i que aquesta operació és molt costosa, no ho és tant com fer seqüencialment la combinació de la taula de Fet (sens dubte, la més gran de totes) amb cadascuna de les taules de Dimensió.

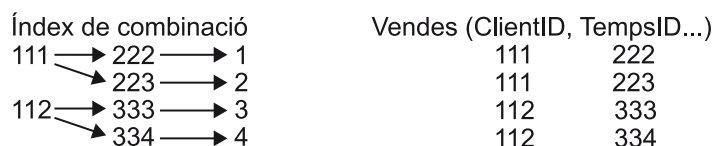
La pregunta que ens podem fer ara és com es pot abaratir encara més aquest segon pas. Podem evitar el producte cartesià? La resposta és que sí, mitjançant índexs de combinació (*join index*). Un índex de combinació és aquell definit sobre una clau forana, de manera que té els valors d'una taula i apunta a l'altra. Com podeu veure en la figura següent, en certa manera, l'índex conté el resultat de la combinació precalculat.

Figura 43



Quan definim una clau primària en una taula, l'SGBD defineix un índex B⁺-arbre sobre els atributs corresponents. Adoneu-vos que en una taula de Fet la clau primària està formada per claus foranes cap a les taules de Dimensió. Per tant, realment parlem d'un índex de combinació. Amb els identificadors d'una Dimensió, podem recórrer l'índex fins a tenir la branca o branques que ens interessin, no caldrà fer el producte cartesià de totes les Dimensions.

Figura 44



El problema que té aquest tipus d'índex és que cal que les condicions de la consulta afectin les Dimensions corresponents als primers atributs de la clau primària. Recordeu que en un índex B⁺-arbre l'ordre dels atributs és rellevant. No és el mateix construir un índex sobre la data i els clients, que sobre els clients i la data. Mentre que per a poder utilitzar el primer cal haver fixat la data, per a utilitzar el segon cal haver fixat el client. Sempre entrem en l'índex pel valor corresponent al primer atribut i comprovem els valors de la resta d'atributs seqüencialment. Consegüentment, per a poder utilitzar un índex B⁺-arbre, cal que l'usuari hagi fixat el valor o valors del primer atribut. Atès que moltes consultes multidimensionals es fan restringint la data, aquesta seria una bona tria per al primer atribut d'un índex de combinació.

En una taula, a més del de la clau primària, podem definir tants índexs com vulguem, però definir un per a cada combinació de Dimensions que pugui arribar a demanar l'usuari pot significar sobrecarregar el sistema amb la gestió d'índexs inútils.

Definició dels índexs

Definir els índexs malament pot voler dir que una consulta trigui hores a executar-se.

Els índexs B⁺-arbre són especialment útils per a fer consultes simples (sense agrupacions, ni agregacions, ni gaires combinacions) i funcionen més bé com més gran és la selectivitat de l'atribut (com menys valors repetits té). El principal problema en aquest cas és que els atributs de les consultes multidimensionals acostumen a no ser gaire selectius. A més, per a taules molt grans, l'índex B⁺-arbre pot ocupar massa espai. Si la taula de Fet conté menys de cinc Mesures, un índex B⁺-arbre pot ocupar un 80% de la grandària de la taula.

Els índexs B⁺-arbre poden ser útils per a resoldre algunes consultes multidimensionals, però no n'hi ha prou.

Adoneu-vos, també, que definir un índex com a *agrupat* (*cluster*) pot resultar molt profitós i no gaire costós si el primer atribut és la data. Com que les insercions es fan massivament i de data en data, sempre aniran a parar al final de la taula. Si la taula està ordenada per data, les dades que vam inserir ahir han d'estar abans que les que inserim avui, que han d'estar abans que les que inserim demà, etc. D'aquesta manera, la taula queda ordenada sense cap cost addicional.

Índex agrupat

Recordeu que un índex agrupat és aquell que no solament manté ordenat l'índex, sinó també les dades dins la taula.

6.2. Índexs de mapes de bits

La millor opció (encara que no disponible en tots els SGBD) per a indexar taules de Fet són els índexs de mapa de bits (*bitmap*). Un índex de mapa de bits és una matriu booleana de tantes columnes com valors tingui l'atribut utilitzat per a la indexació i tantes files com la taula indexada. Si la posició $[i,j]$ val cert, vol dir que en la fila i -èsima l'atribut pren el valor j -èsim.

Figura 45

Bolígrafs	Llapis	Portalàmines	Gomes	Fulls A4	Fulls A3	Guixos	Esborradors	Catalunya	Castella i Lleó	Madrid	Andalusia
1	0	0	0	0	0	0	0	1	0	0	0
0	0	1	0	0	0	0	0	1	0	0	0
0	1	0	0	0	0	0	0	0	0	0	1
0	0	0	0	0	0	0	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	0	0
1	0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	1	0	0	0	0	0	1
0	0	1	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	1	0	1	0	0	0
0	1	0	0	0	0	0	0	1	0	0	0

Exemple de mapes de bits

En la figura superior podeu veure dos exemples de mapes de bits per a la taula de Fet Vendes un per a l'atribut `Producte.nom` i l'altre per a `Regio.nom`. Podem veure que la primera i sisena files de la taula de vendes corresponen a vendes de bolígrafs (mirant la primera columna del mapa de bits esquerre) i que la quarta correspon a una venda a Madrid (mirant la tercera columna del mapa de bits dret).

Al contrari que els B^+ -arbre, els mapes de bits resulten especialment interessants per a atributs amb pocs valors possibles i una baixa selectivitat (el mateix valor repetit moltes vegades). És l'índex ideal per a utilitzar en la taula de clients amb un atribut com ara `sexe` (home o dona), però no té cap sentit fer-lo servir per a `DNI`, perquè tothom tindrà un DNI diferent (obtindríem una matriu molt gran en què cada columna únicament tindria un valor cert).

Aquest tipus d'índex ocupa molt poc espai (si la taula tingués un milió de files i l'atribut vuit valors possibles, l'índex ocuparia només 1 Mbyte) i permet de fer servir operadors lògics de baix nivell per a resoldre predicats sobre els atributs (només cal fer AND i OR sobre tires de bits, que són operacions molt ràpides). A més, s'obté un rendiment òptim per a resoldre consultes que no necessiten accedir a les dades. Només accedint a l'índex, podem comptar quantes files compleixen o deixen de complir una certa condició (comptant zeros o uns, respectivament). En els mapes de bits de la figura anterior podem veure que tenim una única venda d'esborradors i cinc vendes a Catalunya, sense conèixer el contingut de la taula de Fet.

Els mapes de bits són fàcils de construir, mantenir i utilitzar.

Una possible manera d'implementar-ho és definir un B^+ -arbre sobre una certa Dimensió i guardar matrius de bits en les fulles de l'arbre que indiquin quines files de la taula de Fet apunten al valor d'aquesta taula de Dimensió corresponent a la fulla en qüestió. Fixeu-vos en la diferència que hi ha amb l'apartat anterior en què definíem un únic arbre per a tota la taula de Fet. Ara tenim un arbre diferent per a cada taula de Dimensió cadascun dels quals conté mapes de bits de la taula de Fet. Amb això, per a fer una consulta, utilitzarem l'índex de cada Dimensió per a obtenir el mapa de bits corresponent, operarem amb tots i accedirem directament a les files indicades.

Figura 46

Bolígrafs		Llapis				Catalunya	
1		0		1		1	
0		0		0		1	
0		1		1		0	
0		0		0		0	
0	OR	0	=	0	AND	0	=
1		0		1		1	
0		0		0		0	
0		0		0		0	
0		0		0		1	
0		1		1		1	

Mapes de bits

Una variant dels mapes de bits és utilitzar un bit per a cada pàgina de disc, en comptes de cada fila. Per a millorar els mapes de bits, alguns sistemes també els apliquen tècniques de compressió i codificació.

Exemple d'ús dels mapes de bits

En la figura 46 podeu veure com utilitzariem els mapes de bits de les Dimensions Producte i Lloc per a saber quines vendes de bolígrafs i llapis s'han fet a Catalunya. Per a obtenir les dades, només cal accedir a les files primera, sisena i desena.

Els índexs de mapa de bits són especialment apropiats per a fer consultes amb moltes condicions sobre diferents atributs que tenen una selectivitat baixa. Si l'atribut sobre el qual definim l'índex té molts valors possibles, la matriu resulta molt dispersa i malbarata molt d'espai.

6.3. Particions horitzontals

Com ja hem dit força vegades, una taula de Fet realment és molt gran. Al mateix temps, accedir-hi és imprescindible per a resoldre qualsevol consulta. Conseqüentment, hem de facilitar l'accés a aquesta taula tant com puguem. Una manera de fer-ho és dividir-la en n subtaules o particions, posant a cadascuna l'enèsima part de les files (és millor que les particions no se superposin, que no tinguin files en comú). Sempre podem obtenir fàcilment la taula sencera fent la unió de les particions.

Cada partició pot estar en un disc diferent o, fins i tot, en una màquina diferent, amb el corresponent guany de rendiment (aplicant tècniques de paral·lelisme). Com a efecte col·lateral, també facilitem l'escalabilitat del sistema. Quan necessitem més espai, no cal un disc on puguem ficar totes les dades, sinó simplement un on puguem ficar com a mínim una de les particions. L'inconvenient en aquest cas és la disponibilitat. Només que una màquina o disc no funcioni, ja no podem respondre la consulta.

Podem definir les particions segons els valors de qualsevol Dimensió (o fins i tot més d'una Dimensió). Només assegureu-vos que aquesta Dimensió no canviarà mai. La reestructuració de totes les particions pot generar problemes greus. A més, tingueu en compte que fer particions és especialment útil (encara que no disposem de paral·lelisme, ni tan sols de discos diferents), si una consulta només ha d'accedir a una de les particions o com a mínim no ha d'accedir a totes. Per tant, hem de fer particions segons un atribut que s'acostumi a fixar sempre a les consultes. Un altre cop la data pot ser una bona tria. Podem tenir una partició per a cada mes, cada setmana o cada dia si cal.

La partició horitzontal d'una taula de Fet facilita el paral·lelisme i l'escalabilitat del sistema, al mateix temps que redueix el temps de resposta per a les consultes que utilitzin per a seleccionar els Descriptors de la Dimensió utilitzada com a criteri de partició.

6.4. Particions verticals, eines VOLAP

De la mateixa manera que podem fer una partició horitzontal de la taula del Fet, per a reduir-ne la grandària, també podem fer-la verticalment. En aquest cas, definim diferents taules amb subconjunts dels atributs, amb el benentès que cadascun d'aquests ha d'incloure la clau primària. Atès que habitualment cada consulta conté només una Mesura, ho podem portar a l'extrem i definir una taula diferent per a cadascuna, com podeu veure en la figura següent.

Figura 47

Vendes1(IDTemps, IdClient, IdProducte, IdLloc, IdPromocio, articles)

Vendes2(IDTemps, IdClient, IdProducte, IdLloc, IdPromocio, articles)

El guany de la partició vertical és que maximitzem la proporció de dades útils respecte de les dades a les quals s'ha accedit. Imaginem-nos que cada registre abans de la partició ocupa vint-i-vuit bytes i després de la partició només n'ocupa vint-i-quatre. Llavors, cada cop que accedim a una pàgina de disc obtenim aproximadament un 16% més de valors de la Mesura que ens interessa. Fixem-nos, però, que pràcticament necessitem el doble d'espai per a tenir Vendes1 i Vendes2 que el que calia abans per a tenir només Vendes, sense particions de cap tipus.

Per a millorar el rendiment del sistema, podem definir particions verticals de la taula de Fet. Cadascuna de les particions contindrà la clau primària i una o més Mesures.

La millora en aplicar aquesta tècnica seria molt més important si no calgués replicar la clau primària en cadascuna de les taules. En aquest cas, un registre ocuparia el mínim indispensable per a guardar un valor i tot el contingut de la pàgina de disc a la qual s'ha accedit ens seria útil. Doncs bé, hi ha eines OLAP que fan exactament això. Aquestes eines reben el nom de *VOLAP*⁽¹⁰⁾.

⁽¹⁰⁾Vertical OLAP.

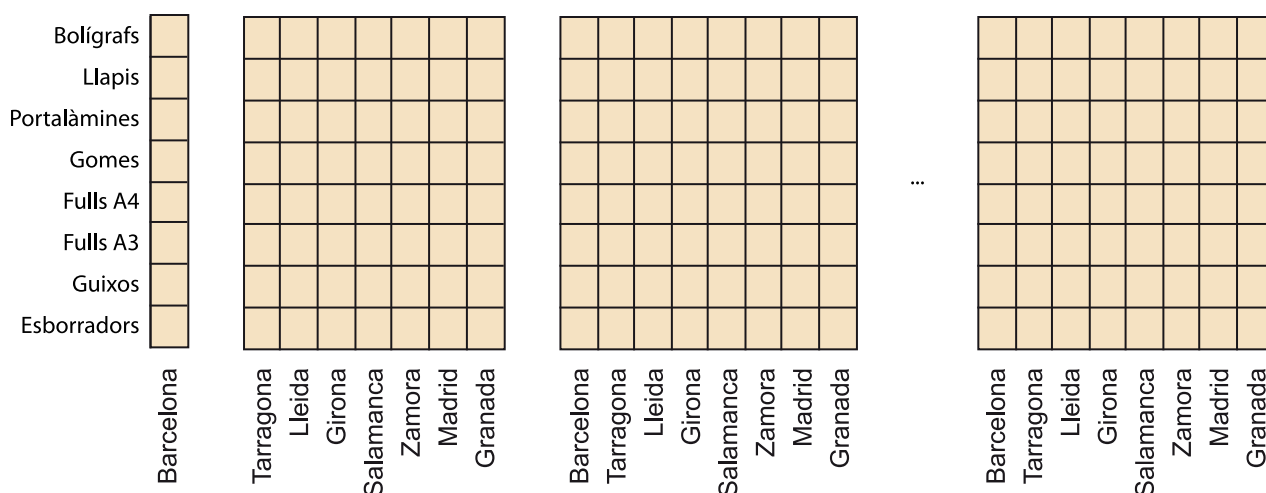
Les eines VOLAP ja no són relacionals. Es tracta d'SGBD específics per a anàlisi multidimensional que guarden els mesuraments d'una mateixa Mesura tots seguits sense els valors de la clau primària i utilitzen diferents tipus d'índex per a localitzar-los. Aquests sistemes poden obtenir un temps de resposta més de deu cops millor que amb un SGBD relacional i utilitzen fins a cent cops menys espai (apliquen mecanismes de compressió). El problema que tenen és que es tracta de solucions totalment propietàries, no estandarditzades, i que es basen en el supòsit que només volem veure les Mesures d'una en una.

Si les consultes només demanen valors d'una única Mesura, podem utilitzar eines VOLAP.

6.5. Matrius *n*-dimensionals, eines MOLAP i HOLAP

Hi ha gent que veu que les bases de dades relacionals no són apropiades per a l'anàlisi multidimensional i que utilitzar-les és artificios. De fet, van ser concebudes per a un món transaccional. Si el que volem consultar són cubs, per què emmagatzemem taules? La resposta són les eines MOLAP (*multidimensional OLAP*) o multidimensionals pures.

Figura 48

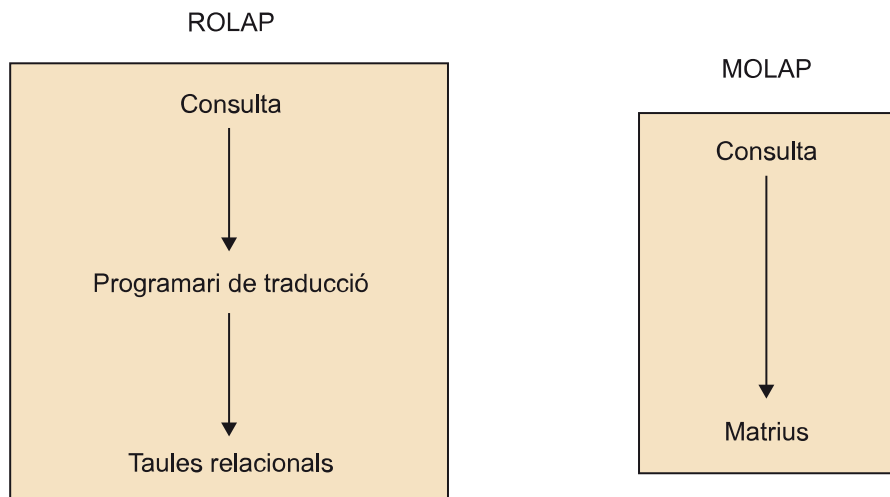


Les eines MOLAP emmagatzemen matrius, com la que teniu en la figura 48, i implementen sistemes d'indexació especials per a accedir-hi. Ara ja no parlem de claus primàries, ni de claus foranes. Cada element de la matriu conté únicament les Mesures. Aquestes matrius tenen exactament la mateixa estructura que els cubs que volem acabar visualitzant. És més, les matrius es defineixen

segons quins siguin els cubs consultats més freqüentment. És a dir, donat el conjunt de consultes crítiques (les executades més sovint i que demanen un temps de resposta més baix), l'eina MOLAP cerca la manera d'emmagatzemar les dades per a minimitzar el temps de resposta per a aquestes consultes.

Les eines MOLAP utilitzen com a sistema d'emmagatzematge matrius n -dimensionals, en comptes de taules relacionals.

Figura 49



En la figura 49 podeu veure la diferència de filosofies entre una implementació ROLAP i una MOLAP. Mentre que la ROLAP necessita una traducció de la consulta a SQL, en una implementació MOLAP la consulta es resol directament sobre les dades (no hi ha cap pas intermedi).

Com que no emmagatzemen claus primàries, us pot semblar que les eines MOLAP fan el mateix que les VOLAP. Això no és cert. La diferència entre una eina MOLAP i una VOLAP és que la MOLAP emmagatzema les dades segons les consultes que s'esperen, mentre que la VOLAP, no. En aquest sentit, una eina VOLAP és més a prop d'una ROLAP que d'una MOLAP.

Afavorir unes consultes més que unes altres és una cosa més natural del que sembla. Per molt que vulguem emmagatzemar matrius n -dimensionals, el disc únicament té dues dimensions (cilindres i sectors) i la memòria RAM simplement una. Conseqüentment, encara que una eina MOLAP intentés de gestionar totes les Dimensions de manera similar, no ho podria fer. N'hi hauria alguna a la qual s'accediria més eficientment que a les altres, perquè les dades estarien físicament agrupades. Per exemple, podríem tenir la informació dels dotze mesos de l'any dins el mateix sector i cilindre de manera que accediríem a tots sense necessitat d'esperar tota la rotació de disc, ni moure el braç per canviar de cilindre. Per contra, si tenim un mes a cada sector, haurem d'esperar que el disc giri completament per a tenir-los tots.

Aquest sistema d'emmagatzematge amb matrius dependents de les consultes és molt eficient, però massa rígid. Penseu què ocorre en la figura 48 si volem afegir una nova data. No hi ha cap problema, simplement afegim a la dreta una nova matriu de 8×8 . Però, què passa si volem afegir una nova ciutat? Cal reorganitzar tota la matriu per a tenir lloc per a posar els nous elements allà on els toca. Per exemple, hauríem d'obrir un forat entre les dades de Granada i Barcelona per a ficar els nous elements al mig. A més, afegir Dimensions a una eina MOLAP multiplica l'espai de disc utilitzat i és molt més complex que en una eina ROLAP, en què simplement cal afegir un nou atribut a la clau de la taula del Fet.

A més del problema de la rigidesa, hi ha el problema de les consultes no considerades crítiques o simplement imprevistes. Com que el sistema no les ha tingut en compte per a emmagatzemar les dades, és força probable que trigui molt a resoldre-les.

Les eines MOLAP funcionen especialment bé quan tenim poques Dimensions i molt estables (que mai no canviïn). Donen molt bon resultat respecte al temps de resposta, però generalment un mal resultat pel que fa a l'emmagatzematge, especialment per a cubs amb una dispersió gran. Això és a causa del fet que es guarda el cub sencer, tant les cel·les plenes com les buides, per a facilitar la indexació i l'accés. Per a resoldre l'excés d'espai utilitzat, acostumen a utilitzar tècniques de compressió de dades.

El principal avantatge d'una eina MOLAP és la seva rapidesa per a consultar les dades. Els seus punts febles són la gestió de grans volums de dades i la rigidesa davant els canvis. Funcionen bé per a magatzems de dades departamentals relativament petits i estables.

Eines MOLAP

Algunes eines MOLAP intenten d'aprofitar més l'espai tot aplicant tècniques de compressió.

Malgrat que el temps de resposta sigui millor amb les eines MOLAP, les eines ROLAP s'imposen en el mercat més per la mateixa implantació i nivell de desenvolupament dels sistemes relacionals que perquè siguin realment adients per a tasques multidimensionals. L'estandardització del llenguatge SQL és un punt a favor molt important d'aquest tipus d'eines, ja que facilita la definició de traductors de consultes multidimensionals independents de l'SGBD. Però, a més de l'estandardització del llenguatge, també és cert que les eines ROLAP demostren una gran robustesa i flexibilitat per a tractar grans volums de dades com els de les taules de Fet. Adoneu-vos també que amb la implementació sobre un SGBD relacional no tenim problemes amb la dispersió dels cubs. Només hi ha files per a les cel·les que contenen dades. A més, les eines MOLAP baixen molt de rendiment quan es fan consultes imprevistes.

Per a aprofitar el que és millor de les eines MOLAP i les ROLAP, també hi ha eines HOLAP (*hybrid OLAP*). Les eines MOLAP són millors per a cubs densos. Les eines ROLAP són millors per a cubs dispersos. És així perquè les eines HO-

LAP identifiquen regions denses i disperses i les emmagatzemen segons una tècnica o l'altra. Quan arriba una consulta, la desfan segon la regió del cub involucrada i van a buscar les dades a un tipus de servidor o altre.

Les eines HOLAP barregen el millor de les eines ROLAP i MOLAP.

6.6. Tècniques de preagregació

Com hem vist fins ara, els diferents tipus d'eines multidimensionals aprofiten les diferents característiques de les consultes per a millorar el temps de resposta. Les eines ROLAP aprofiten especialment la divisió entre Fets i Dimensions; les VOLAP aprofiten que el més habitual és demanar una única Mesura; les MOLAP la repetició i previsió de consultes, etc. Doncs bé, hi ha una característica de l'anàlisi multidimensional que aprofiten totes aquestes: la freqüència d'ús de les operacions *Roll-up* i *Drill-down*. Sabem que l'usuari demanarà les dades a diferents granularitats. Una manera de millorar el temps de resposta és tenir calculat el resultat d'aquestes consultes abans que ho demani.

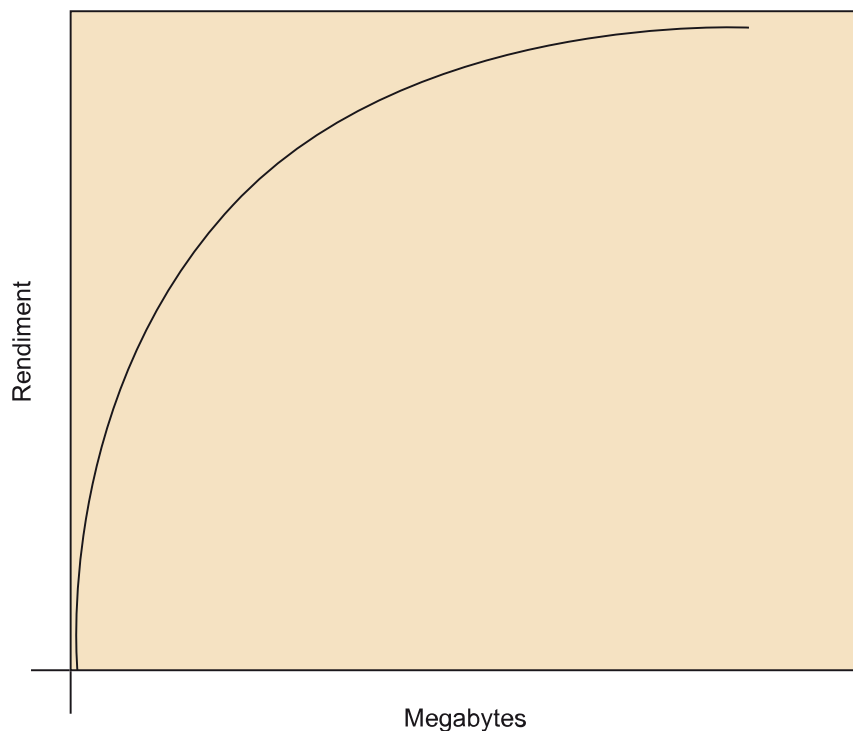
Independentment del tipus d'eina que utilitzem, sempre es poden aplicar tècniques de preagregació per a millorar el temps de resposta davant les operacions *Roll-up* i *Drill-down*. La preagregació és l'eina més potent per a millorar el temps de resposta d'una aplicació multidimensional.

Recordeu que en l'apartat "Components del model multidimensional" ja hem vist que cada Fet conté un reticle de Cel·les relacionades per agregacions. Malgrat que en el disseny conceptual només representem algunes (les més importants) d'aquestes Cel·les, l'usuari les voldrà veure totes en un moment o altre. Conseqüentment, el sistema ha de garantir que aquestes dades estan disponibles i es poden obtenir ràpidament.

El cost d'obtenir dades agregades no és donat pel càlcul que s'hagi de fer, sinó per la simple obtenció de les dades que s'han d'agregar.

La primera solució és guardar les instàncies de totes les Cel·les. Sens cap mena de dubte, és la solució més ràpida (pel que fa al temps de resposta) i la que malbarata més espai. Fixeu-vos que el nombre de Cel·les creix exponencialment respecte al nombre de Dimensions i Nivells. Simplement un Fet amb n Dimensions amb un Nivell per a cada una tindrà 2^n Cel·les. Desgraciadament, això fa que aquesta opció sigui pràcticament sempre inviable.

En l'extrem oposat, només podem emmagatzemar les instàncies de la Cel·la de granularitat més petita i calcular les instàncies de les altres Cel·les sota demanda. Aquesta és la solució que ocupa menys espai, però també la més lenta. El que cal fer realment és decidir quines Cel·les guardem físicament (altrament dit *materialitzades*) i quines calculem només quan la consulta ho demani.



Com podeu veure en la gràfica, dedicant molt poques *megues*, millorem molt el rendiment. Però arriba un moment que, per més espai que dediquem a emmagatzemar Cel·les, no millorem gens el rendiment del sistema.

Per a prendre aquesta decisió, us heu d'adonar que, com més alta sigui una Cel·la en el reticle, menys dispersa serà. Penseu que, només que un dia hàgim venut alguna cosa, quan agreguem els mesos, ja tindrem una cel·la per al mes corresponent. Si ens imaginem que només tenim dades d'aquest mes, a granularitat Dia, tenim una dispersió 31/1 (només un dels trenta-un punts de l'espai conté una cel·la), mentre que, a granularitat mes, tenim una dispersió 1/1 (l'únic punt de l'espai conté una cel·la). En un cas com aquest, necessitaríem el mateix nombre de tuples per a emmagatzemar les dades detallades que per a emmagatzemar les agregades, i no aconseguiríem cap guany en el temps de resposta. Com més disperses siguin les dades bàsiques, més espai ocuparan proporcionalment les dades agregades.

Abans d'emmagatzemar físicament una Cel·la, ens hem d'assegurar que cadascuna de les seves instàncies resulta de l'agregació de, com a mínim, deu instàncies de la Cel·la a partir de la qual la calculem. Si no, el petit guany de temps de resposta no compensarà la gran despesa d'espai.

L'espai que necessitaríem per a emmagatzemar totes les Cel·les és d'ordres de magnitud més gran que el que ocupen les dades bàsiques. L'experiència diu que hem de dedicar aproximadament el mateix espai per a dades preagregades que el que ocupen les dades bàsiques.

Ja hem pres una primera decisió en fer el disseny conceptual. Les Cel·les que hem explicitat (perquè contenen Mesures que no podem calcular a partir de les Mesures de les altres Cel·les o perquè l'usuari les considera especialment importants) són les primeres candidates per a emmagatzemar. Ara només cal veure quines de les altres volem emmagatzemar al costat d'aquestes. Per exemple, si emmagatzemem vendes mensuals (que denotarem `Vendes (Client, Producte, Poblacio, Mes)`), ja no cal que consultem les vendes diàries (`Vendes (Client, Producte, Poblacio, Dia)`) per a calcular les vendes anuals (`Vendes (Client, Producte, Poblacio, Any)`), perquè és molt més econòmic fer-ho a partir de les mensuals.

Decidir quines dades cal preagregar per a obtenir un temps de resposta proper al que obtindríem preagregant-les totes, però dedicant un espai de disc i temps d'actualització raonables, dependrà de molts factors (per exemple, el maquinari disponible, les característiques de la xarxa i del programari, el nombre d'usuaris, etc.). A més, mai no hi haurà el conjunt perfecte de Cel·les que cal preagregar, perquè les consultes dels usuaris evolucionen amb el pas del temps.

El problema de triar quines Cel·les materialitzem no és gens trivial (atenent a la seva complexitat computacional, es tracta d'un problema NP-complet). Però una bona solució és ordenar-les segons la seva utilitat. Les Cel·les més útils són les que serveixen per a resoldre les consultes més costoses i comunes. Una Cel·la és útil tant si la demana l'usuari molt sovint, com si serveix per a obtenir fàcilment la que demana sovint l'usuari. Una Cel·la no és gaire útil si ja hem decidit emmagatzemar les instàncies d'una altra Cel·la a partir de la qual ja la podem obtenir fàcilment. Un cop feta aquesta ordenació, sabent l'espai que ocupa cada Cel·la i el que tenim disponible, apliquem un algorisme voraç (*greedy*) i emmagatzemem tantes Cel·les com puguem, seguint l'ordre d'utilitat fixat anteriorment. Alguns sistemes ja fan la tria automàticament, un cop l'administrador indica l'espai de disc que hi vol dedicar.

Val a dir també que, a l'hora de decidir l'espai que volem dedicar a guardar dades preagregades, també hem de tenir en compte que caldrà actualitzar les dades cada cop que modifiquem la Cel·la atòmica. Adoneu-vos que això pot fer créixer molt la finestra d'actualització. Conseqüentment, per a decidir l'espai dedicat, no solament hem de mirar el preu del disc, sinó el temps que caldrà per a mantenir actualitzades aquestes dades preagregades.

La millor estratègia per a decidir quines dades tenim preagregades és l'assaig error.

Resum

De vegades, per a desenvolupar un cert tipus d'aplicacions, hem d'aplicar tècniques específiques de disseny. Aquest és el cas de les eines OLAP, utilitzades per a fer l'anàlisi multidimensional. El seu disseny es basa en la definició de Fets objecte d'anàlisi i les Dimensions utilitzades per a analitzar-los.

En aquest mòdul hem vist quins són els elements principals d'un model multidimensional: quines estructures de dades ofereix, quines operacions podem fer amb les dades i quines restriccions d'integritat podem definir. També hem estudiat com es fa un disseny conceptual multidimensional pas a pas i com es pot implementar en un SGBD relacional. Per acabar, heu conegut algunes tècniques d'accés per a millorar el temps de resposta i les noves paraules reservades definides en l'SQL'99 per a facilitar l'escriptura de les consultes.

Activitats

1. Llegiu les divuit regles d'avaluació d'eines multidimensionals del Dr. E.F. Codd. Les podeu trobar en el llibre d'E. Thomsen OLAP Solutions o també a <http://www.olapreport.com>.
2. Vegeu com un cert SGBD (per exemple, SQL Server de Microsoft) implementa les operacions ROLLUP i CUBE de l'estàndard SQL'99.

Exercicis d'autoavaluació

1. Feu el disseny multidimensional per al cas d'un conjunt de magatzems de productes d'un distribuïdor de supermercats. Aquest distribuïdor serveix com a intermediari entre els proveïdors i les botigues. Estem especialment interessats a analitzar el següent:

- a) La quantitat de cada producte (estoc) que tenim quan acaba el dia a cadascun dels magatzems, tenint en compte que podem haver comprat el mateix producte a diferents proveïdors.
- b) Cada moviment que hi ha als magatzems (independentment de si és una entrada –què ens serveix un proveïdor– o una sortida –cap a una botiga), segons el producte, l'hora en què té lloc i l'entitat (o bé proveïdor o bé botiga) que el genera.

Per a cada moviment tenim un albarà amb un codi que l'identifica unívocament, assignat pel sistema operacional de gestió dels magatzems comú a tota l'empresa. Un albarà conté les dades del magatzem corresponent, l'hora de lliurament i el nom del proveïdor o botiga, juntament amb la llista de productes que se serveixen.

Cada magatzem (de la mateixa manera que els proveïdors i botigues) és en una població i cada població pertany a una regió. Tant magatzems com botigues i proveïdors s'identifiquen per un nom. Per als magatzems, també disposem dels metres quadrats de superfície.

De la Dimensió temporal, ens n'interessen els mesos, els trimestres, els quadrimestres i els anys. A més d'una hora, dia, mes, trimestre, quadrimestre o any qualsevol, també volem poder seleccionar els dies festius i els mesos d'estiu.

Els productes estan identificats pel codi de barres, però també en volem veure la descripció. Interessa agrupar els productes en tipus de producte, que s'identifiquen pel seu nom. Pensem que un producte no pot ser de més d'un tipus.

2. Estimeu l'espai que ocupa l'esquema anterior, penseu que estem interessats en les dades corresponents en els darrers tres anys (mil dies aproximadament), el nostre catàleg conté 10.000 productes diferents (dividits en 1.000 tipus), tenim deu magatzems, servim 1.000 botigues i comprem a 1.000 fabricants. De mitjana, a cada fabricant li fem una comanda setmanal, mentre que cada botiga ens fa una comanda dia sí, dia no. En total, en acabar l'any tenim 235.000 albarans, amb una mitjana de 50 productes per albarà.

3. Donat l'esquema i els volums de dades anteriors, digueu quin tipus d'eina OLAP seria millor utilitzar (ROLAP, MOLAP, etc.).

4. Traduïu l'esquema anterior a relacional.

5. Penseu ara que per a millorar el temps de resposta de les consultes en la relació EstocDiari (Producte, Dia, Magatzem, quantitat, estoc) que conté l'estoc dels 10.000 productes (de 1.000 tipus), els 1.000 dies, als 10 magatzems (repartits en 3 regions), també decidim emmagatzemar les tres relacions següents, que contenen dades preagregades de les Cel·les corresponents:

Estoc1(Producte, Dia, Regio, quantitat, estoc)
 Estoc2(Producte, Any, Magatzem, quantitat, estoc)
 Estoc3(Tipus, Any, Magatzem, quantitat, estoc)

a) A quants tuples cal accedir per a resoldre una consulta que demana l'estoc anual d'un cert producte en una regió si ho calculem accedint a Estoc? A quants accedint a Estoc1? A quants accedint a Estoc2? A quants accedint a Estoc3?

b) Quina de les quatre relacions hauríem d'utilitzar per a saber el total d'articles (de tots els productes) que hem tingut en estoc en tots els anys en totes les regions?

c) Quines Cel·les podem calcular a partir d'Estoc3?

6. Trobeu una seqüència d'operacions multidimensionals que us permetin de passar del primer cub al segon. Tots dos pertanyen a l'Estrella *Inventari* dissenyada anteriorment.

Origen				
Quantitats de cada tipus de producte per albarà	1.111	2.222	3.333	4.444
Materials d'oficina	1	2	3	4
Aliments	4	3	2	1

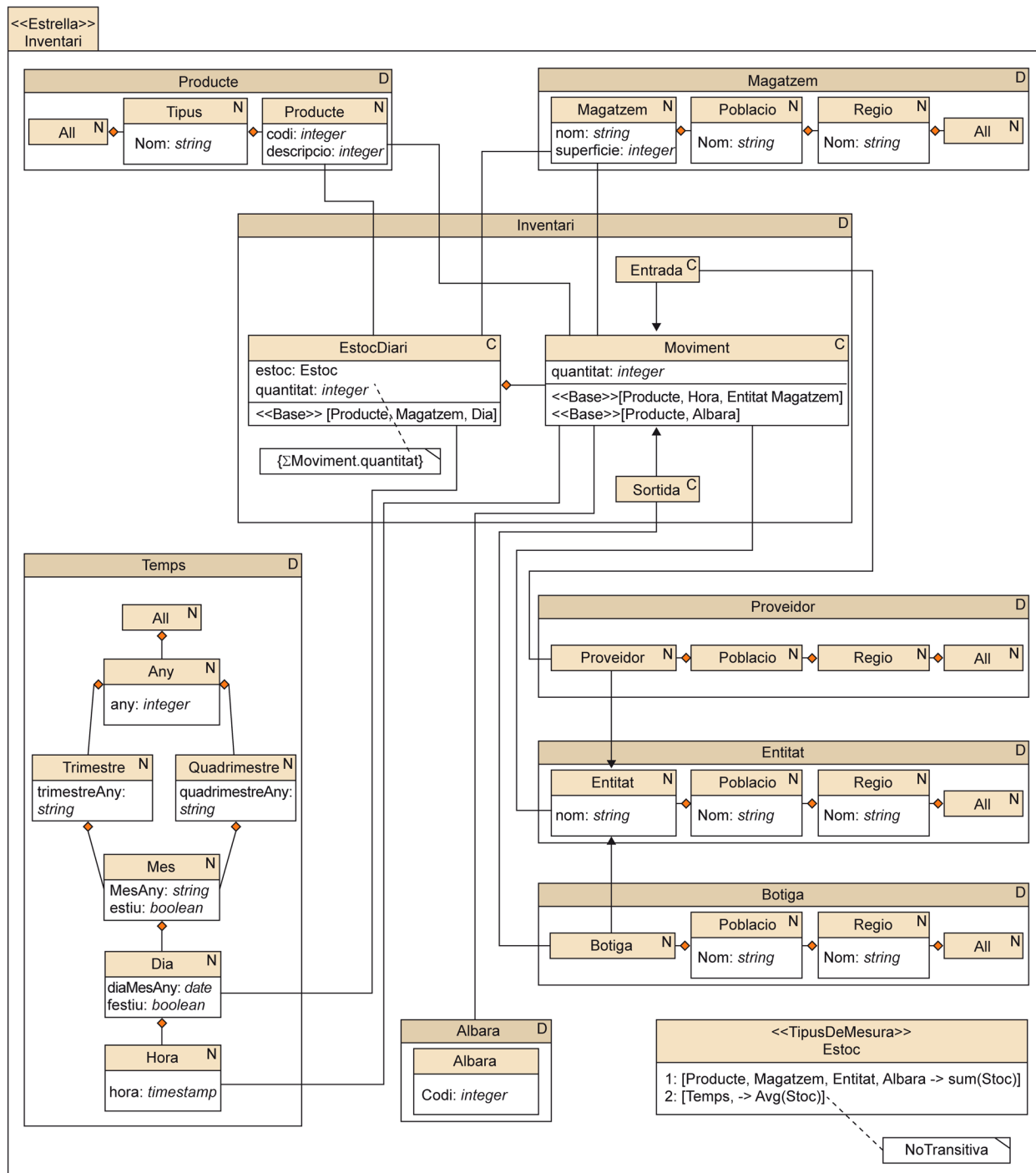
Destinació					
Quantitats de producte per regió de magatzem i dia		23-4-2002	24-4-2002	25-4-2002	26-4-2002
Catalunya	Gomes	1	0	0	2
	Bolígrafs	0	2	3	2

7. Feu una consulta de la taula Entrada definida en l'exercici 4, utilitzant les paraules reservades definides en l'estàndard de 1999, per a obtenir el resultat següent escrivint el mínim possible:

Regió de magatzem	Any	Tipus de producte	Quantitat
Catalunya	2001	Materials d'oficina	100
Catalunya	2001	Aliments	200
Catalunya	2001	Null	300
Catalunya	2002	Materials d'oficina	400
Catalunya	2002	Aliments	500
Catalunya	2002	Null	900
Catalunya	Null	Materials d'oficina	500
Catalunya	Null	Aliments	700
Catalunya	Null	Null	1.200

Solucionari

1. La quantitat que hi ha al magatzem pot provenir de diferents fabricants. Consegüentment, no podem associar la Dimensió Fabricant amb la Cel·la EstocMensual.



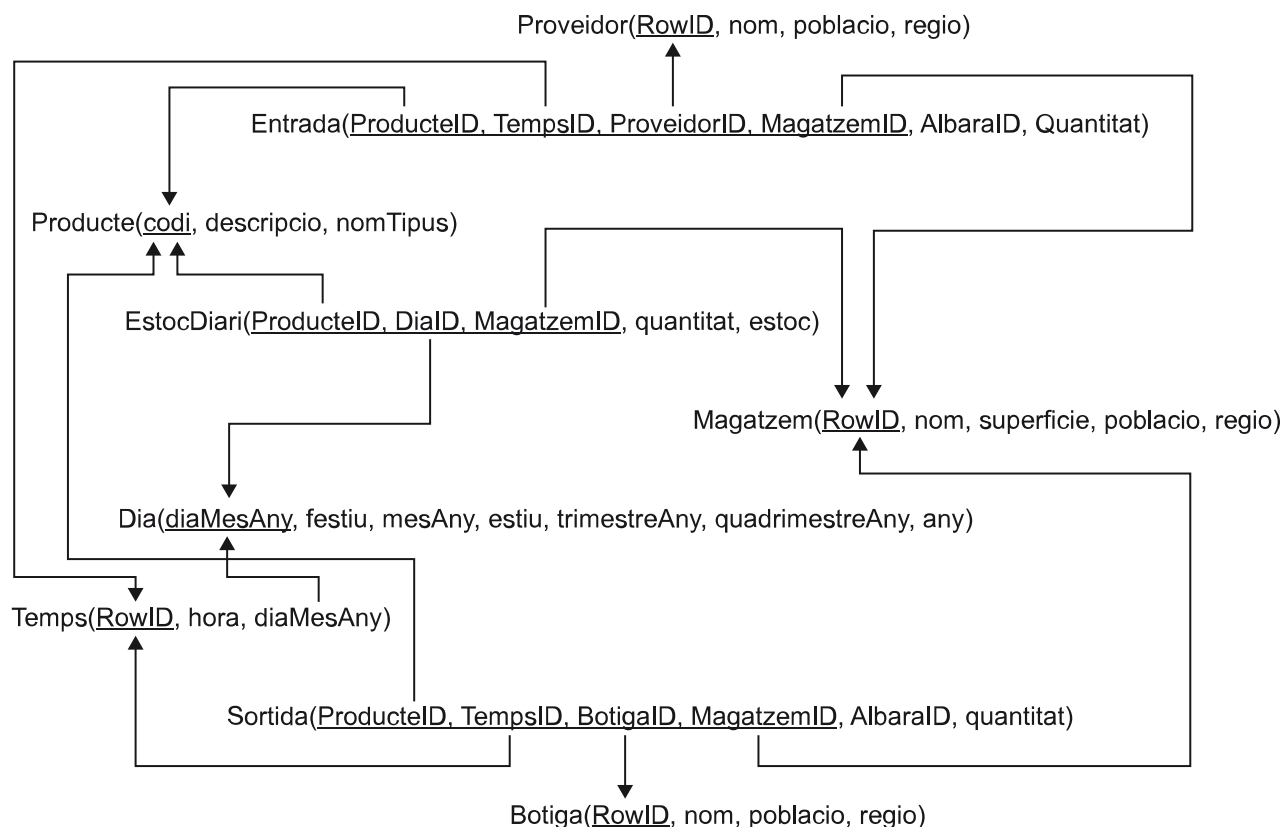
2. Una fita màxima de l'espai de la Cel·la Moviment és 4.800.000.000.000 cel·les (10.000 productes × 24.000 hores × 2.000 entitats × 10). Ara bé, podem obtenir una fita més petita si ho calculem a partir del nombre d'albarans. Així, obtenim que l'espai té com a màxim 7.050.000.000 cel·les (3 anys × 235.000 albarans/any × 10.000 productes). Ara bé, com l'enunciat diu que hi ha una mitjana de 50 productes en cada albarà, sabem que realment seran només 35.250.000 cel·les. Si pensem que cada quantitat ocupa 2 bytes i que necessitem cinc identificadors (un per a cada Dimensió) de 4 bytes cada un, la Cel·la ocuparia 775.500.000 bytes (775,5 Mb).

La Cel·la *EstocDiari* defineix un espai màxim de 100.000.000 cel·les (10.000 productes × 10 magatzems × 1.000 dies). Pensem que cada estoc ocupa 2 bytes i que necessitem tres identificadors de 4 bytes cada un, la Cel·la ocuparia 1.600.000.000 bytes (1.600 Mb).

En total tota l'Estrella ocuparà 2375,5 Mb (negligint l'espai que puguin ocupar les Dimensions).

3. Les dades corresponents als moviments són molt disperses. A cada hora no ens arriben comandes de tots els clients ni ens serveixen tots els proveïdors. Probablement, la millor tria seria una eina ROLAP, pel gran volum de dades que suporten i perquè no tenen problemes amb la dispersió. Per contra, la dispersió del cub corresponent a *EstocDiari* és pràcticament 1. Totes les cel·les que hi pot haver gairebé sempre hi seran. Si disposem d'una eina MOLAP que suporti aquest volum de dades, seria la millor opció, perquè probablement ens oferirà un temps de resposta més petit que una implementació ROLAP. Si volem tenir tots dos cubs dins el mateix sistema, hauríem d'escollir un HOLAP.

4. *Albara* és una Dimensió degenerada, que quedarà simplement com a part d'una clau alternativa. No desapareix completament, perquè ens pot interessar fer consultes per albarà. A més d'això, com les subclasses de *Moviment* no tenen cap atribut específic, pel que fa a l'espai que ocuparà, és equivalent tenir una superclasse o dues subclasses. Consegüentment, ens hem de fixar simplement en el temps de resposta. Serà més ràpid consultar una taula gran o dues de petites? En aquest cas implementarem l'especialització com si fos una partició horitzontal. Cada tipus de moviment està en una taula diferent (si els volem tots, només cal fer la unió).



5.

a) En el pitjor cas caldrà accedir a tots els tuples de cada relació. *EstocDiari* en conté 10^8 , *Estoc1* en conté 3×10^7 i *Estoc2* en conté 3×10^5 . *Estoc3* conté només 3×10 tuples, però no el podem utilitzar perquè les seves dades no són prou detallades (estan en granularitat tipus de productes i volem consultar productes).

b) Tot i que seria molt més eficient utilitzar *Estoc2* o *Estoc3*, hauríem d'utilitzar *Estoc1* si volem tenir el resultat exacte. Recordeu que al llarg del temps agreguem utilitzant la mitjana i que la mitjana no és una funció transitiva. Consegüentment, hem d'accedir a les dades al nivell més detallat, que en aquest cas és *Dia*.

c) Considerant que no podem continuar agregant al llarg de la Dimensió Temps per la raó esmentada anteriorment i que els tipus de producte són disjunts (com diu l'enunciat), podríem calcular les Cel·les següents:

Estoc4(Tipus,Any,Poblacio,quantitat)
 Estoc5(Tipus,Any,Regio,quantitat)
 Estoc6(Tipus,Any,quantitat)
 Estoc7(Any,Magatzem,quantitat)
 Estoc8(Any,Poblacio,quantitat)
 Estoc9(Any,Regio,quantitat)
 Estoc10(Any,quantitat)

6. Una possible solució podria ser la següent:

$A := Seleccio_{Tipus,nom="Materials\ d'oficina"}(origen)$
 $B := CanviBase_{Producte \times Magatzem \times Temps \times Entitat}(A)$
 $C := Drill-down_{Producte::Producte}(B)$
 $D := Roll-up_{Entitat::All}(C)$
 $E := Roll-up_{Magatzem::Regio}(D)$
 $F := Roll-up_{Temps::Dia}(E)$
 $destinacio := CanviBase_{Magatzem \times Producte \times Temps}(F)$

7.

```
SELECT m.regio, d.any, p.tipus, SUM(f.quantitat)
FROM entrada f, magatzem m, temps t, dia d, producte p
WHERE f.MagatzemID=m.RowId
AND f.TempsID=t.RowID
AND t.diaMesAny=d.diaMesAny
AND f.ProducteID=p.codi
AND d.any IN (2001,2002)
AND p.tipus IN ("Aliments", "Materials d'oficina")
AND m.regio="Catalunya"
GROUP BY m.regio, CUBE(d.any, p.tipus)
ORDER BY m.regio, d.any, p.tipus;
```

Glossari

dada *f* Mesurament, observació feta i emmagatzemada en algun sistema.

descriptor *m* Atribut d'una dimensió utilitzat per a seleccionar instàncies dels fets.

dice *f* Operació multidimensional que redueix la grandària de l'espai que selecciona valors en les dimensions.

dimensió *f* Punt de vista utilitzat en l'anàlisi d'un cert fet.

dispersió *f* Quocient entre el nombre màxim teòric d'instàncies que hi pot arribar a haver i les instàncies que hi ha realment en un cub.

drill-across *f* Operació multidimensional que, donat un espai, canvia les dades que s'hi mostren. Canvia d'un tema d'anàlisi a un altre.

drill-down *f* Operació multidimensional que mostra les dades més detallades.

entitat-interrelació

sigla: ER

en entity-relationship

factoria d'informació corporativa *f* Conjunt d'elements de programari i maquinari que ajuden en l'anàlisi de dades per a prendre decisions.

sigla: FIC

fet *m* Objecte d'anàlisi.

finestra d'actualització *f* Període de temps dedicat a l'actualització d'un magatzem de dades.

granularitat *f* Grandària d'un objecte respecte a un altre.

HOLAP *f* Eina OLAP que barreja característiques ROLAP i MOLAP.

en hybrid olap

informació *f* Dades rellevants per a algun decisor que afecten alguna de les seves decisions.

jerarquia d'agregació *f* Conjunt de relacions entre les instàncies d'una dimensió que indica com n'agrupem unes per a obtenir-ne d'altres.

magatzem de dades corporatiu *m* Conjunt de dades integrades i històriques de tota l'empresa.

magatzem de dades departamental *m* Conjunt de dades que resol les necessitats d'anàlisi d'un cert departament o conjunt d'usuaris.

mesura *f* Dada numèrica associada a un esdeveniment que volem analitzar.

metadada *f* Dada sobre les dades.

MOLAP *f* Eina OLAP que utilitza matrius n-dimensionals (en comptes de taules relacionals) per a emmagatzemar les dades.

en multidimensional OLAP

nivell d'agregació *m* Conjunt d'instàncies de la mateixa granularitat dins una dimensió.

O³LAP *f* Eina OLAP implementada sobre un SGBD orientat a l'objecte.

en object-oriented OLAP

OLAP *f* Sigles que fan referència a les eines d'anàlisi, normalment multidimensional. Categoria de tecnologia de programari que permet als analistes, gestors i executius de millorar el seu coneixement de les dades mitjançant l'accés ràpid, consistent i interactiu a una àmplia varietat de possibles vistes d'informació que ha estat transformada des de les dades operacionals per a reflectir la dimensionalitat real de l'empresa com l'entén l'usuari (The OLAP Council).

en on-line analytical processing

OLTP *f* Sigles que fan referència a sistemes operacionals, que ajuden en el dia a dia de la nostra empresa.

en on-line transaccional processing

ROLAP *f* Eina OLAP implementada sobre un SGBD relacional.

roll-up *f* Operació multidimensional que resumeix les dades augmentant la granularitat al llarg d'una Dimensió.

SGBD *m* Vegeu **sistema de gestió de bases de dades**.

sistema de gestió de bases de dades *m* Programari que gestiona i controla bases de dades. Les seves funcions principals són les de facilitar-ne l'ús simultani a molts usuaris de tipus diferents, independitzar l'usuari del món físic i mantenir la integritat de les dades.

sigla: SGBD

en database management system

sistema operacional *m* Sistema que ajuda en les operacions diàries de negoci d'una organització.

sistema transaccional *m* Sistema basat en transaccions de lectura/escriptura.

slice *f* Operació multidimensional que redueix la dimensionalitat de l'espai fixant un valor en una dimensió.

VOLAP *f* Eina OLAP que emmagatzema les dades per columnes en comptes de per files.
en vertical OLAP

Bibliografia

Gulutzan, P.; Pelzer, T. (1999). *SQL-99 Complete, Really*. Lawrence, Kansas: R&D Books.

Kimball, R. (1996). *The DataWarehouse Toolkit*. Nova York: John Wiley & Sons, Inc.

Thomsen, E. (1997). *OLAP Solutions*. Nova York: John Wiley & Sons, Inc.

<http://www.olapreport.com>

<http://www.olapcouncil.org>