

# APLICACIÓ DE L'ANÀLISI DE PROCEDÈNCIES AL DISSENY DEL CONTINGUT DE BASES DE DADES

A. OLIVÉ I RAMON

*A la primera part de l'article, es fa una exposició de l'anàlisi de precedències entre conjunts d'informació i, en particular, de l'anàlisi de derivabilitat, en base al model relacional de dades.*

*A la segona part, s'aplica l'anàlisi de precedències a alguns problemes que es presenten durant el disseny de bases de dades. Aquests problemes són: definició i validació dels requeriments d'una base de dades; disseny del contingut de la base de dades; i disseny de la dinàmica dels processos d'afegir, derivar i esborrar informacions de la base de dades.*

*S'explica un exemple concret al qual s'apliquen els conceptes i els mètodes proposats.*

## 1. INTRODUCCIÓ

En aquest article presentem una tècnica i comentem la seva aplicació a alguns aspectes del disseny de bases de dades (BD). La tècnica és l'"anàlisi de precedències entre conjunts d'informació" que, aplicada al model informacional, és explicada en /1/.

L'anàlisi de precedències es basa en el fet que les informacions d'un cert univers de discurs /2/ mantenen relacions de precedència entre elles. Es diu que un conjunt d'informació A és precedent d'un altre B si existeix un procés d'informació que deriva B a partir de A. L'existència d'aquestes relacions de precedència permet de classificar les informacions d'un univers de discurs en dues classes: les bàsiques i les derivades. Les primeres són aquelles que no es poden derivar de les altres, mentre que les segones són les que es poden derivar de les primeres.

Si les relacions de precedència es defineixen formalment, és possible de donar un procediment formalitzat per a resoldre alguns problemes que es plantegen en l'àrea dels sistemes d'informació i, en particular, en l'àrea de les bases de dades. Un d'ells és el de determinar quines informacions bàsiques són necessàries per a derivar un cert conjunt d'informació. Un altre és el de determinar quines informacions d'un conjunt A són necessà-

ries, i quines no, per a derivar un conjunt B. Un tercer problema, potser més important, és el de verificar si a partir d'un conjunt d'informació A és o no és possible de derivar un conjunt B. Aquest darrer problema l'anomenem "anàlisi de derivabilitat".

L'anàlisi de precedències es pot utilitzar per a resoldre alguns problemes que es presenten en l'àrea de les BD. En aquest treball ens centrarem en l'aspecte del disseny del contingut d'una BD, contemplat a dos nivells: al nivell de la validació lògica dels requeriments d'entrades i sortides i al nivell del disseny pròpiament dit.

La definició dels requeriments d'una BD, /4, 5/, consisteix bàsicament en la definició de les seves entrades (tipus, freqüència, volum etc.), de les seves sortides (tipus, freqüència, volum, temps de resposta, etc.) i dels tipus d'informació bàsiques o derivades existents a l'univers de discurs. Aquests requeriments proporcionen la informació necessària per a dissenyar la BD.

La validació lògica dels requeriments d'entrades i sortides és verificar que -des d'un punt de vista lògic- les sortides es poden obtenir a partir de les entrades. Si en algun cas no és així, els requeriments no són consistents, ja que es demana alguna sortida que no és derivable de cap manera a partir de

- A. Olivé i Ramon. Facultat d'Informàtica de la U.P.B. Pça. Eusebi Güell, 1. Barcelona, 34.  
- Article rebut el Setembre del 1980.

les entrades.

El disseny del contingut de la BD és un problema posterior que consisteix a determinar quins tipus d'informació i quines informacions de cada tipus s'han d'emmagatzemar a la BD. Aquest contingut ha de ser tal que garanteixi que les sortides, definides en els requeriments, es podran derivar en tots els casos.

La solució trivial d'aquest problema consisteix a emmagatzemar, sense cap mena d'agregació, totes les entrades rebudes. Aquesta solució, però, pot consumir molt d'espai de memòria auxiliar i pot implicar un temps de resposta a les consultes inadmissible. Per aquesta raó, s'acostuma a fer una certa agregació de les entrades, compatible amb les consultes que es faran, malgrat que això impliqui un procés d'actualització quan entra una informació que afecta les informacions agregades.

Per tant, decidir el contingut de la BD porta aparellat el decidir els processos que es faran amb les entrades. A més, implica decidir quan s'esborraran les informacions innecessàries, i quines són.

Al nostre parer, ambdós problemes no han rebut, llevat d'algunes excepcions com /4/, -- l'atenció que es mereixen. En aquest article tractarem de fer-hi una contribució, sense pretendre de fer un nou mètode de disseny.

L'article està estructurat en 6 seccions. La secció 2 presenta els elements indispensables del model de dades que utilitzarem, que és el model relacional. Presenta també un petit exemple, que farem servir a les seccions posteriors. La tercera secció explica l'anàlisi de precedències, aplicat al model relacional i en particular l'anàlisi de la derivabilitat. A la quarta secció fem l'aplicació al problema de la definició i validació dels requeriments d'una BD. Continuant l'exemple iniciat a la secció 2, definim les entrades i sortides d'una BD i en fem la validació. La secció 5 és dedicada al disseny del contingut d'una BD. Presenta el problema i un mètode de solució, aplicant-lo a la BD exemple. Finalment, la secció 6 fa les conclusions.

## 2. DICIONARIS I CONJUNTS D'INFORMACIÓ

### 2.1 El model relacional /6, 7, 8/

Un model és una eina que permet de modelar les informacions d'un cert univers de discurs. Per a nosaltres, l'univers de discurs d'una BD comprèn totes les informacions que la BD rep, processa, emmagatzema o deriva durant el seu període de vida. Per tant, l'univers de discurs no està limitat a només un instant sinó que inclou un període, normalment llarg, de temps. Per aquesta raó, les informacions sobre, per exemple, estocs de productes, saldos de comptes, màquines avariades, etc. porten sempre una referència de temps (dia, mes, any, etc.) ja que sense ella la informació no té sentit.

De l'aplicació del model en surt una llista de tipus d'informació, que en el model relacional són "tipus de relació" ("relational scheme" en la terminologia anglesa usual). - Un tipus de relació  $R$  és un conjunt ordenat d'atributs  $\{A_1, \dots, A_n\}$ . Cada atribut té un nom i té associat un domini. Un domini  $D_i$  és un conjunt de valors d'un mateix tipus. Designarem per  $D(A_j)$  el domini associat amb l'atribut  $A_j$ .

Una relació  $r$  en  $R$  és un subconjunt del producte cartesià  $D(A_1) \times \dots \times D(A_n)$ . Els elements d'una relació s'anomenen ènuples  $t = \langle a_1, \dots, a_n \rangle$  on  $a_j \in D(A_j)$  per  $j = 1 \div n$ , essent  $a_j$  un component de l'ènuple.

El conjunt  $R_j$  de tots els ènuples d'un cert  $R_j$  que existeixen a l'univers de discurs és la seva relació total.

Cada  $R$  té una clau. Una clau és una col·lecció d'atributs de  $R$ ,  $K(R)$ , tal que no hi ha dos ènuples a  $R$  amb el mateix component  $K$ . - Com és usual, subratllarem els atributs de  $K$  en  $R$ .

Estem interessats, només, en tipus de relació que estan en tercera forma normal, 3NF, /7/ i que són elementals. Un tipus de relació és elemental si té, com a màxim, un atribut no clau. Això no és cap limitació, ja que qualsevol tipus de relació en 3NF pot ésser descomposat en d'altres d'elementals i, en canvi, és molt convenient de cara a l'anàlisi de precedències.

L'algebra relacional permet la derivació de noves relacions a partir d'altres prèviament definides, utilitzant un conjunt d'operacions /6, 8/. En aquest treball utilitzarem la unió, la intersecció, la resta, la projecció i la restricció. Només definirem aquesta última, ja que les demés són clàssiques a la literatura de BD.

La restricció de  $r$  per  $s$ ,  $r|s$ , amb  $S \subseteq R$  es defineix com la relació  $\{t | t \in r \wedge \exists t_1 (t_1 \in s \wedge t[S] = t_1)\}$ . Aquesta operació és idèntica al "natural join", però ens interessa de considerar-la separatament perquè té la propietat  $r|s \subseteq r$  que no té el "natural join". Aquesta propietat, com veurem més endavant, és útil de cara a la definició de conjunts d'informació.

## 2.2 Diccionari

Entenem per diccionari d'un univers de discurs d'una BD la definició de:

- Els tipus de relació presents a l'univers de discurs
- Els conjunts de valors que poden prendre - els dominis dels atributs
- El grup (bàsic, derivat amb operació de model, derivat amb operació d'usuari) al qual pertany cada tipus de relació
- Les precedències dels tipus de relació derivats (Secció 3)
- Les equivalències entre conjunts d'informació (Apt. 2.3)
- Altres elements no considerats aquí, com ara descripcions, volums, autoritzacions d'accés, "integrity constraints", etc.

En un cert diccionari, un tipus de relació és bàsic o derivat. És bàsic si els seus ènuples no es poden derivar dels ènuples d'altres tipus de relació; en cas contrari, és derivat. En aquest darrer cas, s'ha de distingir entre els que es poden derivar amb les operacions del model i els altres, que s'han de derivar amb operacions definides pels usuaris. Els tipus de relació bàsics i els derivats amb operació d'usuari s'han de definir explícitament al diccionari. Els altres no és necessari de fer-ho, ja que són implícits.

A la figura 1 s'ha definit un exemple de llista de tipus de relacions, que farem ser-

vir en aquest article. L'exemple correspon a una aplicació bancària, simplificada, en la qual hi ha transaccions que afecten a comptes.  $R_1$  a  $R_5$  donen els atributs de cada transacció: l'oficina en la qual es fa, la data, el tipus (dipòsit, extracció, etc.), el compte al qual afecta i el seu import. Se suposa que cada compte té assignat un cert crèdit màxim, que pot variar d'un dia a l'altre.  $R_6$  és el tipus de relació corresponent a aquestes variacions de límit de crèdit. La clau de  $R_6$  és COMPTE i DATA perquè un mateix compte pot tenir més d'una variació durant el període de vida.

$R_7$  és el saldo dels comptes.  $R_8$  indica el límit de crèdit vigent per a cada compte i data, que és igual al de l'última variació que s'hi ha fet, o a l'inicial si no hi ha hagut cap canvi.  $R_9$  conté els ènuples  $\langle c, d, e \rangle$  -- tals que el saldo del compte  $c$  en  $d$  és negatiu i supera en  $e$  el límit de crèdit vigent en  $d$ . Finalment,  $R_{10}$  dona el nombre de transaccions fetes per una oficina durant un cert mes.

Observi's que  $R_6$  a  $R_{10}$  inclouen un atribut de temps, perquè l'univers de discurs no es limita a un sol instant de temps, sinó que inclou tot el període de vida de la BD.

### Bàsics

- $R_1$  (TRANSACCIÓ, OFICINA)
- $R_2$  (TRANSACCIÓ, DATA)
- $R_3$  (TRANSACCIÓ, TIPUS)
- $R_4$  (TRANSACCIÓ, COMPTE)
- $R_5$  (TRANSACCIÓ, IMPORT:PTS.)
- $R_6$  (COMPTE, DATA, NOU-LÍMIT-CRÈDIT:PTS.)

### Derivats amb operació d'usuari

- $R_7$  (COMPTE, DATA, SALDO:PTS.)
- $R_8$  (COMPTE, DATA, LÍMIT-CRÈDIT:PTS.)
- $R_9$  (COMPTE, DATA, EXCEDEIX-CRÈDIT:PTS.)
- $R_{10}$  (OFICINA, MES, NOMBRE-TRANSACCIONS)

Figura 1. Tipus de relacions de l'exemple

Designarem per  $D(R_i, A_j)$  el conjunt de valors que pot pendre l'atribut  $A_j$  en  $R_i$ . Aquests conjunts de valors s'han de definir al diccionari. Entre els que utilitzarem en aquest article hi ha:

$D(R_1, \text{TRANSCACCIÓ}) = \{1 \div 99.999\}$   $i=1 \div 5$   
 $D(R_i, \text{DATA}) = \{d_1 \div d_n\}$   $i=2, 6, 9$   
 $D(R_i, \text{DATA}) = \{d_0, d_1 \div d_n\}$   $i=7, 8$   
 $D(R_{10}, \text{MES}) = \{m_1 \div m_2\}$

Se suposa que les dates tenen el format dia-mes-any i els mesos el format mes-any. Les dates de  $R_2, R_6$  i  $R_9$  van de  $d_1$  a  $d_n$ . En canvi, les de  $R_7$  i  $R_8$  tenen un dia més, el  $d_0$ , que és la data inicial de l'univers de discurs.

### 2.3 Conjunts d'informació

Qualsevol relació  $r$  és un conjunt d'informació (ci). Una relació  $r$  és sempre un subconjunt de la relació total  $R$ . Això suggereix un mètode de definició dels cis consistent en expressar-los com una restricció de la relació total corresponent:  $I = R_i | r_k$  essent  $r_k$  la relació que ens permet de seleccionar els ènuples de  $R_i$  que volem. Si s'omet  $r_k$  llavors  $I = R_i$ . Si  $r_k = \phi$  llavors  $I = \phi$ . En  $R_i | r_k$  anomenarem  $R_i$  el conjunt base i  $r_k$  el conjunt de res tricció.

Així, per exemple,  $R_6 | \{\text{DATA} = d\}$  és el ci que conté totes les variacions de límits de crèdits que s'han fet en el dia  $d$ . El cas en el qual  $r_k$  és una relació unària explícita del tipus  $\{\text{Atribut} = \text{valor}\}$  és molt freqüent i és convenient de simplificar la notació  $R_i | \{\text{Atribut} = \text{valor}\}$  per l'equivalent  $R_i | \text{valor}$  sempre i quan no hi pugui haver confusió sobre l'atribut al qual pertany el valor.

$R_i$  o  $r_k$  poden ésser el resultat d'operacions sobre relacions. Per exemple,  $R_5 | ((R_4 | c) [1] \cap (R_2 | d) [1])^1$  és el ci que conté els imports de les transaccions del compte  $c$  fetes en el dia  $d$ .

Finalment, en l'expressió de  $r$  es poden utilitzar funcions. Això resulta molt convenient especialment per a tractar els aspectes de temps. Així, per exemple, si  $\text{MES}(D)$  és una funció que aplicada a un conjunt  $D$  de dates ens dóna el conjunt de llurs mesos, tenim -- que  $R_{10} | \text{MES}(d)$  és<sup>2</sup> el conjunt d'ènuples de

$R_{10}$  corresponents al mes de la data  $d$ . Semblantment, si  $\text{DIES}(M)$  és una funció que aplicada a un conjunt  $M$  de mesos en dóna el conjunt de totes les dates dels mesos de  $M$ , l'expressió  $R_9 | \text{DIES}(\text{MES}(d))$  correspon al conjunt d'ènuples de  $R_9$  tals que llur data és una de les del mes de  $d$ . Si, per exemple,  $d = 10-05-1980$  llavors  $\text{MES}(d) = 05-1980$  i  $\text{DIES}(\text{MES}(d)) = \{01-05-1980 \div 31-05-1980\}$ .

A l'apèndix 1 definim les funcions que utilitzarem en aquest treball.

### 2.4 Equivalències entre conjunts d'informació

A la secció anterior hem definit els conjunts d'informació mitjançant expressions de l'àlgebra relacional. Convé, però, de distingir entre una expressió  $E$  i el seu valor, que denotem per  $\alpha(E)$  o, simplement, per  $\alpha E$ . El valor d'una expressió és el conjunt d'informacions que designa.

Aquesta distinció ens permet de definir l'equivalència entre conjunts d'informació. Diem que dues expressions  $E_1$  i  $E_2$  són equivalents si  $\alpha E_1 = \alpha E_2$ . Hi ha dues menes d'equivalències: les de model (EM) i les d'usuari (EU). Les primeres són les que es poden definir a nivell de model, mentre que les segones només són aplicables a un cert univers de discurs.

A continuació s'indiquen, sense demostració, les equivalències de model que es faran servir més endavant.  $X$  és un conjunt d'atributs i  $r, s, i, t$  són relacions.

- EM.1  $r | s \cup r | t = r | (s \cup t)$   
 EM.2  $r | s \cap r | t = r | (s \cap t)$   
 EM.3  $(r | s) [X] \cup (r | t) [X] = (r | (s \cup t)) [X]$  amb  $\underline{S} = \underline{T}$   
 EM.4  $(r | s) [X] \cap (r | t) [X] = (r | (s \cap t)) [X]$  amb  $\underline{S} = \underline{T}$   
 EM.5  $r | r [X] = r$   
 EM.6 Si  $r [X] = t [X]$  llavors  $(r | s) [X] \cap t [X] = (r | s) [X]$   
 EM.7 Si  $\underline{S} \subseteq X \subseteq \underline{R}$  llavors  $(r | s) [X] = r [X] | s$   
 EM.8  $r | D(r, A_j) = r$  essent  $A_j \in \underline{R}$

En l'exemple farem servir les equivalències d'usuari següents:

- EU.1  $R_1 [1] = R_2 [1]$   
 EU.2  $R_2 [1] = R_3 [1]$

$$\text{EU.3 } R_2 [1] = R_4 [1]$$

$$\text{EU.4 } R_2 [1] = R_5 [1]$$

### 2.5 Conjunts de conjunts d'informació

Anomenem conjunt de conjunts d'informació -- (cci) a un conjunt els elements del qual són conjunts d'informació. Per exemple,  $A = \{R_1 | (R_2 | d) [1], R_6 | d\}$  és un ccci de dos elements. Essent conjunts, als cci's se'ls hi pot aplicar les operacions clàssiques d'unió intersecció i resta de conjunts.

Si  $A = \{I_1, \dots, I_n\}$  és un cci, el valor de  $\alpha A$  és  $\alpha A = \bigcup_{I_j \in A} \alpha I_j$ . Dos ccis  $A_1$  i  $A_2$  són equivalents si  $\alpha A_1 = \alpha A_2$ .

Un cci és normalitzat quan els conjunts base dels seus cis són tots diferents. En cas contrari, en direm no normalitzat. El cci normalitzat equivalent a un cci no normalitzat  $A$  el designarem per  $\delta A$ . Per exemple, si  $A = \{R_1 | r, R_1 | s, R_2 | t\}$  tindrem  $\delta A = \{R_1 | (r \cup s), R_2 | t\}$ .

A continuació definim les operacions normalitzades d'unió ( $\cup$ ), intersecció ( $\cap$ ) i resta ( $\dot{-}$ ) de ccis. Siguin  $A_1$  i  $A_2$  dos ccis normalitzats. El cci  $A = A_1 \cup A_2$  és el cci normalitzat tal que  $\alpha A_1 \cup \alpha A_2 = \alpha A$ . Les altres -- dues operacions es defineixen semblantment.

Per exemple, si  $A_1 = \{R_1 | r_1, R_2 | r_2\}$  i  $A_2 = \{R_1 | r_3, R_3 | r_4\}$ , tindrem  $A_1 \cup A_2 = \{R_1 | (r_1 \cup r_3), R_2 | r_2, R_3 | r_4\}$ ,  $A_1 \cap A_2 = \{R_1 | (r_1 \cap r_3)\}$  i  $A_1 \dot{-} A_2 = \{R_1 | (r_1 - r_3)\}$ .

En el cas particular de la unió es demostra fàcilment que  $\delta A \cup \delta B = \delta(A \cup B)$ . Aquesta -- propietat, però, no es satisfeta per les altres dues operacions.

### 3. ANÀLISI DE PRECEDÈNCIES

El concepte "precedència entre conjunts d'informació" fou originat per Langefors /9/ i ha estat utilitzat en diferents treballs fets en els àrees del sistema d'informació i de les BD. Així, per exemple, Nunamaker /10/ l'emprà en el seu SODA, un sistema per a dissenyar i construir automàticament SI; Waters /11/ ha construït un programa "analtzador de precedències" capaç d'analtzar xarxes de precedències entre informacions; i Bubenko -

/4/ utilitza les precedències per a definir les informacions derivades d'una BD.

### 3.1 Relació de precedència

Diem que els conjunts d'informació  $I, \dots, I_n$  són precedents del conjunt  $I'$  si existeix un procés d'informació tal que les seves entrades són exactament  $I, \dots, I_n$  i la seva sortida és  $I'$ . Els conjunts  $I, \dots, I_n$  són suficients per a obtenir  $I'$ .

Això es pot estendre fàcilment a ccis. Siguin  $X, Y, Z, W$  ccis. Diem que  $Y$  és un precedent de  $X$  si tots els cis de  $X$  poden ésser derivats a partir dels cis de  $Y$ .

La relació de precedència  $P$  comprèn totes les precedències existents entre els ccis d'un cert univers de discurs. Escrivim  $P[X, Y]$  si el cci  $Y$  és un precedent del cci  $X$ . Per -- definició,  $P$  té les següents propietats:

- (a)  $P[X, X]$
- (b) Si  $P[X, Y]$  llavors, per a qualsevol  $Z$ ,  $P[X, Y \cup Z]$ . Per tant, si  $Y \subseteq W$  i  $P[X, Y]$  tindrem també  $P[X, W]$
- (c) Si  $P[X, Y]$  i  $P[Z, W]$  llavors  $P[X \cup Z, Y \cup W]$ . En particular, si ...  $P[X, Y]$  i  $P[Z, Y]$  llavors  $P[X \cup Z, Y]$
- (d) Si  $P[X, Y]$  i  $Z \subseteq Y$  i  $P[Z, W]$  llavors,  $P[X, (Y-Z) \cup W]$ . En particular, si ..  $P[X, Y \cup Z]$  i  $P[Z, W]$ , llavors ...  $P[X, Y \cup W]$ . També, si  $P[X, Y]$  i  $P[Y, Z]$ , llavors  $P[X, Z]$
- (e) Si  $P[X, Y]$  i  $Z \subseteq X$ , llavors  $P[Z, Y]$

Per tant,  $P$  és una relació binària, antisimètrica, reflexiva i transitiva.

Un element de  $P$  és una precedència. Hi ha -- dues classes de precedències en  $P$ : les bàsiques i les derivades. Les precedències derivades s'obtenen a partir de les bàsiques mitjançant les propietats (a) - (e) anteriors. Les bàsiques, en canvi, s'obtenen aplicant -- unes regles de precedència.

Hi ha dues classes de regles de precedència: les de model (RPM) i les d'usuari (RPU). Les primeres es poden utilitzar en qualsevol uni

vers de discurs d'un cert model, mentre que les segones només es poden aplicar en un cert univers de discurs.

### 3.2 Regles de precedència de model

A continuació s'indiquen les RPM, relatives al model relacional, que farem servir als -- exemples. En el que segueix,  $r$ ,  $s$  i  $t$  són relacions i  $X$  un conjunt d'atributs.

RPM.1 Si  $\underline{S} \subseteq X \subseteq \underline{R}$  llavors  $P[r[X]|s, r|s]$

En efecte, projectant  $r|s$  tindriem  $(r|s)[X]$  que, per EM.7, és  $r[X]|S$ .

RPM.2 Si  $\underline{S} = \underline{T}$  llavors  $P[r|(s \cup t)$   
 $\{r|s, r|t\}]$

Si fem la unió de  $r|s$  i  $r|t$  tenim  $(r|s) \cup (r|t)$  que per EM.1 és  $r|(s \cup t)$ .

### 3.3 Regles de precedència d'usuari

Com hem dit abans, les RPU només es poden -- aplicar en un cert univers de discurs. El mètode de definició de les RPU que proposem -- consisteix a definir, per cada tipus de relació  $\underline{R}$  derivat amb operació d'usuari,  $P[R|s, \underline{X}]$  essent  $R$  la relació total corresponent a  $\underline{R}$ ,  $s$  qualsevol relació tal que  $\underline{S} \subseteq \underline{R}$  i  $X$  els -- precedents de  $R|S$ .

En  $R|s$ ,  $s$  actua com a paràmetre de la restricció, ja que es refereix a qualsevol relació, amb l'única condició que  $\underline{S} \subseteq \underline{R}$ . Per aquesta raó,  $X$  tindrà també  $s$  com a paràmetre. - D'aquesta manera, podem aplicar les RPU a - qualsevol situació concreta. N'hi ha prou -- amb substituir en  $X$   $s$  per la relació realment present.

A continuació s'indiquen les RPU corresponents als quatre tipus de relació derivats - amb operació d'usuari, de la figura 1. A - efectes de simplificació de notació se suposarà que en  $R_1|r$ ,  $\underline{R} \subseteq K(R_1)$ .

RPU.1  $P[R_7|r, \{R_7|ANT(r),$

$R_{2,3,4,5} | ((R_2|r[2])[1]) \cap$

$\cap (R_4|r[1])[1])\}^3$

$R_7|r$  és els saldos corresponents als <compte

data> indicats en  $r$ . Els seus precedents són  $R_7|ANT(r)$  que és els saldos en el dia anterior dels mateixos comptes i el conjunt  $R_{2,3,4,5}$  corresponents als dies i comptes indicats en  $r$ .

RPU.2  $P[R_8|r, \{R_8|ANT(r), R_6|r\}]$

Per a obtenir els límits de crèdit vigents per als <compte, data> indicats en  $r$  cal disposar dels vigents en la data anterior i dels canvis produïts.

RPU.3  $P[R_9|r, \{R_7|r, R_8|r\}]$

Els excessos de crèdit s'obtenen a partir del saldo i dels límits de crèdit vigents.

RPU.4  $P[R_{10}|r, \{R_{12} | ((R_1|r[1])[1]) \cap$   
 $\cap (R_2|DIES(r[2]))[1])\}^3$

Per a derivar el nombre de transaccions corresponent a les <oficina, mes> indicats en  $r$  cal el conjunt de  $R$ , corresponent a les oficines presents en  $r$  i amb data pertanyent als mesos de  $r$ .

### 3.4 Anàlisi de derivabilitat

Siguin  $X, Y$  ccis. Diem que el cci  $X$  és derivable del cci  $Y$  en un cert univers de discurs si, i només si,  $P[X, Y]$ . Per tant, analitzar la derivabilitat de  $X$  consisteix a verificar si existeix en  $P$  la precedència  $P[X, Y]$ .

Si tenim en compte la propietat (e) de  $P$ , resulta que si el cci  $X$  és derivable del cci  $Y$  implica que, per cada  $I_j \in X$ ,  $P[I_j, Y]$ . Segons això, l'anàlisi de la derivabilitat del cci  $X$  a partir del cci  $Y$  es pot fer a passos. A cada pas, tenim un cci  $I_j$  de  $X$  pel qual hem de verificar que  $P[I_j, Y]$ .

Normalment, no es dedueix directament --  $P[I_j, Y]$  sinó que es dedueix una precedència  $P[I_j, Z]$  essent  $Z$  un cci. En aquest cas, el pas es divideix en tants subpassos com cis - hi ha a  $Z$ , i  $I_j$  serà derivable si tots els - cis de  $Z$  ho són. Si algun d'aquests no ho és,  $I_j$  tampoc. Abans, però, d'arribar a aquesta conclusió cal explorar totes les alternatives possibles.

Sigui  $I_j = R|s$  el ci a analitzar en algun pas. Les alternatives possibles són:

1. Substituir R o s pel seu equivalent de model (EM) o d'usuari (EU), si n'hi ha.
2. Si R és derivat per una operació de model, podem deduir els precedents de  $I_j$  de les regles de precedència de model (RPM).
3. Si R és derivat per una operació d'usuari, podem deduir els precedents de  $I_j$  de les regles de precedència d'usuari (RPU).

En les seccions següents aplicarem aquesta - mecànica en alguns casos basats en l'exemple.

#### 4. VALIDACIÓ DELS REQUERIMENTS D'UNA BASE DE DADES

##### 4.1 Requeriments d'una base de dades

No hi ha acord unànime sobre el que són exactament els requeriments d'una BD. Això no obstant, diversos autors, per exemple /4, 5, 12/, coincideixen que aquests requeriments - han de contenir dues menes d'informacions: - les relatives a l'estructura de les informacions (de l'univers de discurs) de la BD i - les relatives a l'ús previst de la BD.

Kahn /12/ les anomena "information structure perspective" (ISP) i "usage perspective" (UP), respectivament, i les considera necessàries per a fer el disseny lògic d'una BD. En la - ISP ha d'haver-hi definides les entitats, -- llurs propietats i les relacions existents - entre elles, així com informació sobre volum, privacitat, etc. En canvi, la UP defineix -- els processos que es fan, llur tipus, les in - formacions que utilitzen i els aspectes rela - tius a volum, freqüència, prioritats, etc.

Semblantment, per a Bubenko et al. /4/ els - requeriments consisteixen en (a) Una descripció de les transaccions, amb llur contingut, freqüència, ordre de sorteig i tipus (actualització, addició o supressió); (b) Una descripció de les consultes, amb llur expressió, freqüència, ordre de sorteig, temps de res - posta i "actualitat" exigida de la informa - ció; (c) Una descripció de l'estructura de - la informació; i (d) Els requeriments i de - sitjos dels usuaris (flexibilitat, espai de

memòria auxiliar, etc.)

Per a calcular el contingut d'una BD, és im - prescindible de disposar d'aquestes dues clas - ses de requeriments, ja que el contingut ne - cessari d'una BD depèn de les entrades, sor - tides i processos previstos. En concret, els elements que ens interessin són: (1) El dic - cionari; (2) Les transaccions d'entrada d'in - formació a la BD; i (3) Les consultes a la - BD. En les seccions anteriors hem indicat -- com s'ha de definir el diccionari. A continua - ció indiquem com s'han de definir les transac - cions i les consultes.

Anomenem T el període de vida previst de la BD, que definim com un conjunt d'instantis --  $T = \{t_i \div t_j\}$ , expressats en una única unitat de temps (segon, hora, mes, etc.). Aquesta uni - tat ha d'ésser suficientment "fina" per a po - der definir sense ambigüitat els esdeveni - ments de la BD.

Siguin  $I_1, \dots, I_m$  les classes de transac - cions previstes. Una transacció d'una classe  $I_j$  es pot produir en un instant qualsevol -- d'un conjunt  $T(I_j) \subseteq T$ , prèviament definit. -  $I_j(t)$  és la transacció de la classe  $I_j$  feta en t. Si  $t \notin T(I_j)$  llavors, per definició, -  $I_j(t) = \phi$ . Per altra banda, encara que  $t \in T(I_j)$  si en t no hi ha cap entrada de la classe  $I_j$ ,  $I_j(t) = \phi$ .

Si la transacció és periòdica, els instantis de  $T(I_j)$  es generen a partir de  $\alpha + \beta n$ , -- amb  $n \geq 0$ , essent  $\alpha$  l'instant corresponent a la primera entrada i  $\beta$  el període. En el cas general, però, no cal que les transaccions - siguin periòdiques.

La definició de les transaccions consisteix a definir, per a cada classe  $I_j$ , el conjunt d'instantis  $T(I_j)$  i el cci  $I_j(t)$  essent t un instant qualsevol de  $T(I_j)$ .  $I_j(t)$  serà un con - junt de cis, cada un dels quals es defineix mitjançant una expressió en àlgebra relacio - nal, que tindrà com a paràmetre t. Per exem - ple,  $T(I_1) = \{d_1 \div d_n\}$ ,  $I_1(d) = \{R_{1,2,3,4,5} | (R_2 | d) [1]\}$  que indica que cada dia  $d \in T(I_1)$  entra tota la informació relativa a les transaccions -- originades en d.

Observi's que amb el nostre enfocament no té sentit i no cal definir transaccions d'"ac - tualitzar" o d'"esborrar", ja que associem a

cada informació una referència al temps. Actualitzar o esborrar una informació és afegir una informació i no modificar-ne una d'existents. Per exemple, per a nosaltres el saldo d'un compte en el dia  $d$  no és una actualització del saldo del dia  $d-1$ , sinó una nova informació. Senko, /3, pag. 239/, propugna, a un altre nivell, aquest mateix enfocament.

Designarem per  $I(t) = \bigcup_{j=1, m} I_j(t)$  el cci unió de tots els ccis que entren en  $t$ , per  $I_j(t_1, t_2) = \bigcup_{t_i \in A} I_j(t_i)$  amb  $A = \{t_1 \div t_2\}$  el cci que conté les entrades de la classe  $I_j$  fetes entre  $t_1$  i  $t_2$ , i per  $I(t_1, t_2) = \bigcup_{j=1, m} I_j(t_1, t_2)$  la unió de tots els ccis que entren entre  $t_1$  i  $t_2$ .

D'una manera semblant s'ha de fer la definició de les consultes. Si hi ha  $n$  classes de consultes  $Q_1, \dots, Q_n$  per a cada classe  $Q_i$  s'ha de definir el cci  $Q_i(t)$ , essent  $t$  un instant qualsevol de  $T(Q_i)$ .  $Q(t)$ ,  $Q_j(t_1, t_2)$  i  $Q(t_1, t_2)$  es defineixen semblantment al cas anterior.

A l'apartat següent definim els requeriments d'una BD exemple.

#### 4.2 Exemple

Partirem del diccionari que s'ha definit a les dues seccions anteriors. El període de vida previst és  $T = \{d_0, d_1, \dots, d_n\}$  essent la unitat de temps un dia,  $d_0$  el primer dia i indeterminat l'últim dia.

Hi ha dues classes de transaccions d'entrada:  
 $I_1(d) = \{R_{7,8} | d\}$   $T(I_1) = \{d_0\}$   
 $I_2(d) = \{R_{1,2,3,4,5} | (R_2 | d) [1], R_6 | d\}$   
 $T(I_2) = \{d_1 \div d_n\}$

$I_1$  correspon a la creació de la BD, que es fa en data  $d_0$ . S'entren els saldos i els límits de crèdit vigents a l'inici del període de vida.  $I_2$  entra diàriament les transaccions i els canvis de crèdit que s'hagin produït.

$I(d)$  i  $I(d_0, d)$  seran:

$I(d) = \{R_{7,8} | d, R_{1,2,3,4,5} | (R_2 | d) [1], R_6 | d\}$   
 $I(d_0, d) = \{R_{7,8} | d_0\} \cup \bigcup_{d_i \in B} \{R_6 | d_i, R_{1,2,3,4,5} | (R_2 | d_i) \cdot [1]\}$

amb  $d = \{d\} \cap \{d_0\}$  i  $B = \{d_1 \div d\}$

Observi's que quan  $d = d_0$   $I(d_0) = \{R_{7,8} | d_0\}$  ja que  $R_2 | d_0 = R_6 | d_0 = \phi$  degut als valors  $D(R_2, DATA) = D(R_6, DATA) = \{d_1 \div d_n\}$ .

Hi ha també dues classes de consultes:

$Q_1(d) = \{R_{7,8} | d\}$   $T(Q_1) = \{d_1 \div d_n\}$   
 $Q_2(d) = \{R_9 [1,2] \text{DIES}(\text{MES}(d)), R_{10} [\text{MES}(d)]\}$   
 $T(Q_2) = \text{ÚLTIM-DIA}(\text{MES}(T(Q_1)))$

$Q_1$  és els saldos dels comptes i els excessos de crèdit en el dia  $d$ .  $Q_1$  s'ha d'obtenir diàriament, a partir del dia  $d_1$ .  $Q_2$ , en canvi, s'ha d'obtenir l'últim dia de cada mes. Consisteix en els ènuples, obtinguts per projecció de  $R_9$  sobre COMPTE i DATA, tals que llur DATA pertanyi al mes en qüestió, i en els ènuples de  $R_{10}$  corresponents al mateix mes.

#### 4.3 Validació lògica dels requeriments d'entrades i sortides

La validació dels requeriments d'una BD és una tasca complexa, difícil i mal definida. Aquí no pretenem de fer-ne un tractament complet sinó que volem limitar-nos a una part concreta d'aquesta validació i aplicar l'anàlisi de precedències a la seva solució.

El problema és la validació lògica dels requeriments d'entrades i sortides de la BD. Es tracta de verificar que des d'un punt de vista lògic les consultes definides es poden derivar de les entrades. Si no és així, els requeriments són incorrectes ja que hi ha alguna consulta, feta en algun instant  $t$ , que no es pot respondre amb la informació que ha entrat fins  $t$ . Caldrà, doncs, afegir alguna entrada a  $I(t)$  o suprimir alguna consulta de  $Q(t)$ .

Més formalment, es tracta de verificar que per a cada  $Q_j$ ,  $j=1$  a  $n$ , per a  $\forall t \in T(Q_j)$ ,  $Q_j(t)$  es pot derivar a partir de  $I(t_i, t)$ . O sigui, verificar que existeix  $P [Q_j(t), I(t_i, t)]$ .

A l'apartat següent fem aquesta verificació per a la BD exemple.

#### 4.4 Exemple

Comencem verificant que  $Q_1(d) = \{R_{7,8} | d\}$  es pot



derivar de  $I(d_0, d)$

#### Anàlisi de $R_7|d$

Aplicant RPU.1 resulta  $P[R_7|d, X]$  amb  $X = \{R_7|ANT(d), R_{2,3,4,5} | ((R_2|d) [1] \cap R_4 [1])\}$  - d'on -tenint en compte EU.3 i EM.6- es pot -treure  $R_4 [1] : X = \{R_7|ANT(d), R_{2,3,4,5} | (R_2|d)[1]\}$

Aplicant novament RPU.1 a  $R_7|ANT(d)$  i així -successivament fins que  $ANT^1(d) = d_0$ , tenim --  $P[R_7|d, Y]$  amb:  $Y = \{R_7|d_0\} \cup \bigcup_{d_1 \in B} \{R_{2,3,4,5} | (R_2|d_1) [1]\}$  essent  $B = \{d_1 \div d\}$ . Com que --  $Y \subset I(d_0, d)$  resulta que  $R_7|d$  és derivable.

#### Anàlisi de $R_9|d$

Aplicant RPU.3 tenim  $P[R_9|d, \{Y, Z\}]$  amb  $Y$  -com abans i  $Z = R_8|d$ . Aplicant RPU.2 resulta -  $P[R_8|d, W]$  amb  $W = \{R_8|d_0\} \cup \bigcup_{d_1 \in B} \{R_8|d_1\} d'$  on  $W \subset I(d_0, d)$ .

Un cop verificada la derivabilitat de  $Q_1(d)$ , resta per verificar que  $Q_2(d)$  també es pot -derivar de  $I(d_0, d)$ .

#### Anàlisi de $R_9 [1, 2] | DIES(MES(d))$

Aplicant RPM.1 resulta:

$P[R_9 [1, 2] | DIES(MES(d)), R_9 | DIES(MES(d))] i$  per RPU.3:

$P[R_9 | DIES(MES(d)), Y \cup W]$  essent  $Y$  i  $W$  com abans.

#### Anàlisi de $R_{10} | MES(d)$

Aplicant RPU.4 dona  $P[R_{10} | MES(d), U]$  amb --  $U = \{R_{1,2} | ((R_1 [1]) \cap (R_2 | DIES(MES(d)) [1]))\}$  d'on -tenint en compte EU.1 i EM.6- es pot -suprimir  $R_1 [1] : U = \{R_{1,2} | (R_2 | DIES(MES(d)) [1])\}$ .

Aplicant a  $U$  la RPM.2 tenint en compte que -  $d = \text{ÚLTIM-DIA}(MES(d))$ , surt  $P[U, V]$  amb: --  $V = \bigcup_{d_1 \in D} \{R_{1,2} | (R_2 | d_1) [1]\}$  essent  $D = DIES(MES(d))$  essent  $D = DIES(MES(d))$ . D'on tenim  $V \subset I(d_0, d)$  i, per tant,  $R_{10} | MES(d)$  és derivable.

### 5. CÀLCUL DEL CONTINGUT D'UNA BASE DE DADES

Un problema important que es presenta en el

disseny d'una base de dades és la determinació de les informacions que hi han d'estar -explícitament emmagatzemades. Aquestes informacions, segons la terminologia emprada per Sundgren /13/, constitueix el nucli de la BD a partir del qual es pot derivar tot el contingut informacional de la BD.

Aquest és un problema de disseny perquè normalment existeixen moltes alternatives possibles i cal triar-ne una. Per exemple, en el nostre cas, pot triar-se entre emmagatzemar el saldo dels clients, actualitzant-lo a cada transacció, o emmagatzemar totes les transaccions que s'hagin produït i calcular el -saldo cada cop que es necessiti.

En la nostra opinió, aquest problema s'ha estudiat poc i és tractat superficialment, o ignorat, en bona part de la literatura sobre el disseny de BD. Una de les excepcions és l'article de Bubenko et al. /4/ on es planteja el disseny de l'esquema d'una BD com un -procés de generació d'alternatives, llur anàlisi i tria entre elles. El disseny final --normalment estarà en un punt intermedi entre dos extrems. Un extrem és la BD en la que no més s'emmagatzemen les transaccions, de manera que cada vegada que s'ha de derivar una -sortida s'han d'executar tots els processos necessaris, prenent les transaccions com a dades d'entrada. L'altre extrem és la BD que té explícitament emmagatzemades totes les informacions que seran demanades, de forma que cada cop que entra una transacció s'han d'actualitzar totes les informacions afectades.

La decisió del contingut de la BD consisteix a determinar no solament quins tipus d'informació han d'estar explícitament emmagatzemades a la BD, sinó també quines informacions de cada tipus han d'estar-hi, en un moment -determinat. Es a dir, per exemple, no solament s'ha de fixar que s'emmagatzemaran els saldos dels clients sinó també quins saldos hi haurà (els corresponents a un instant determinat, tots els del mes en curs, els tres últims diferents, etc.)

Per aquesta raó, decidir el contingut de la base de dades comporta també decidir la seva dinàmica, que inclou les decisions sobre: (1) en quins instants es processaran les transaccions; (2) quines informacions afegirà o actualitzarà cada transacció a la BD; i (3) -

cada quan s'esborraran informacions de la BD i quines. Totes aquestes qüestions s'han de resoldre en conjunt i formen una unitat.

Per a fer aquest disseny cal tenir en compte per a cada alternativa, una sèrie d'aspectes tècnics contraposats, com són el temps de -- procés de les consultes, el temps d'actualització, l'espai de memòria auxiliar ocupat, -- etc. L'elecció s'ha de fer buscant un equilibri entre elles, en un SGBD concret.

Cal, però, que les alternatives que s'estudien siguin lògicament factibles. És a dir, que partint del contingut de la BD que implica una alternativa es puguin derivar les consultes previstes i es pugui fer el tractament de les transaccions. Si no és així, -- l'alternativa no és vàlida i no cal tenir-la en compte.

L'anàlisi de precedències es pot utilitzar -- per a fer l'anàlisi de la factibilitat lògica. Aquesta aplicació és la que presentem a continuació.

### 5.1 Marc de referència

Durant el període de vida  $T$  d'una BD s'executen un cert nombre de processos de quatre -- classes: (1) Processar les transaccions d'entrada, a fi de derivar les informacions que s'han d'afegir a la BD; (2) Afegir aquestes informacions a la BD; (3) Derivar les sortides sol·licitades; i (4) Esborrar les informacions que ja no són necessàries.

Quan en un mateix instant  $t \in T$  s'executen -- processos de dues o més classes, suposarem -- que s'executen en l'ordre indicat a la figura 2. És a dir, primer s'obtenen les informacions a afegir a la BD,  $A(t)$ , a partir de les transaccions  $I(t)$  i del contingut inicial de la BD en  $t$ ,  $B(t)$ . A continuació, s'afegeix  $A(t)$  a  $B(t)$ , obtenint  $B'(t)$ . A partir de  $B'(t)$  s'obtenen les sortides  $Q(t)$ . Finalment, s'esborren de  $B'(t)$  les informacions que ja no es necessitaran més,  $D(t)$ , obtenint finalment  $B''(t)$ .  $B''(t)$  serà el contingut inicial de la BD en  $t+1$ :  $B''(t) = B(t+1)$ .

De fet, no hi hauria cap inconvenient en suposar un ordre diferent, com --per exemple-- l'indicat a la figura 3, en el qual primer es deriven les sortides  $Q(t)$  a partir de  $B(t)$

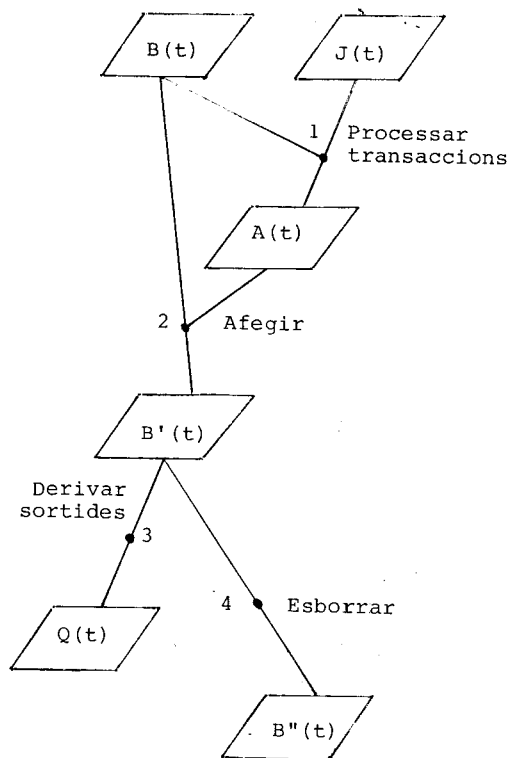


Fig. 2

i de  $I(t)$ , i després s'afegeixen i/o esborren informacions a/de la BD. En aquest cas, les relacions que s'indiquen més avall serien -- lleugerament diferents, però en essència -- idèntiques.

Del marc de referència que hem adoptar resulta que, per a  $\forall t \in T$ , es satisfan les relacions següents, on  $A(t)$ ,  $B(t)$ ,  $B'(t)$ ,  $B''(t)$  i  $D(t)$  són ccis normalitzats<sup>4</sup>:

$$B'(t) = B(t) \cup A(t) \quad (5.1)$$

$$D(t) = B'(t) - B''(t) \quad (5.2)$$

$$D(t) = \subseteq B'(t) \quad (5.3)$$

$$B''(t) = B(t+1) \quad (5.4)$$

$$B''(t) = B'(t) - D(t) \quad (5.5)$$

$$B''(t) = (B(t) \cup A(t)) - D(t) \quad (5.6)$$

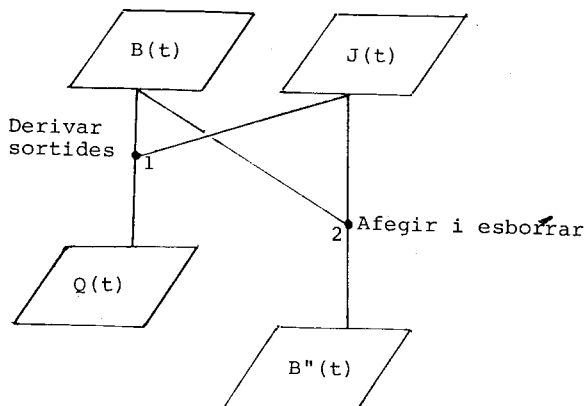


Fig. 3

Prendrem com a principi que una informació - només entra o s'afegeix a la BD una sola vegada durant el període de vida T. És a dir - que, per a qualsevols  $t_1, t_2 \in T$  diferents: -  $I(t_1) \cap I(t_2) = \emptyset$  i  $A(t_1) \cap A(t_2) = \emptyset$ . Observi's - que, lògicament, s'han de derivar tantes vegades com calgui.

A partir d'aquest principi es dedueix fàcilment:

$$A(t') \cap B(t) = \emptyset \quad \text{si } t' \geq t \quad (5.7)$$

$$B'(t') \cap D(t) = \emptyset \quad \text{si } t' > t \quad (5.8)$$

Tenint en compte (5.7), de (5.1) es treu:

$$A(t) = B'(t) \dot{-} B(t) \quad (5.9)$$

$$B(t) = B'(t) \dot{-} A(t) \quad (5.10)$$

Combinant (5.9) amb (5.4) i tenint en compte (5.8) resulta que:

$$A(t) = B'(t) \dot{-} B(t+1) \quad (5.11)$$

Als apartats següents utilitzarem aquestes - relacions per a determinar el contingut d'una BD.

## 5.2 Anàlisi lògica d'alternatives

És tasca de dissenyador de la BD el generar les alternatives possibles i triar la més -- convenient. Ara bé, per a que una alternativa sigui vàlida cal que satisfaci dues condicions.

En primer lloc, per a  $j=1+n, \forall t \in T(Q_j)$  s'ha de satisfer la precedència  $P[Q_j(t), B'(t)]$  ja que -en el marc de referència adoptat- -- les consultes es deriven a partir de  $B'(t)$ . En segon lloc,  $A(t)$  s'ha de poder derivar de  $B(t)$  i  $I(t)$ . Per tant, la segona condició és que es satisfaci, per a  $\forall t \in T, P[A(t), B(t) \cup I(t)]$ .

L'especificació d'una alternativa no cal que inclogui els valors de totes les variables -  $B(t), B'(t), A(t)$  i  $D(t)$  ja que unes es poden derivar d'altres. De fet, n'hi ha prou amb - definir  $B'(t)$ , ja que llavors es pot obtenir  $A(t)$  per (5.11),  $B(t)$  per (5.10) i  $D(t)$  per (5.2).

## 5.3 Exemple

Anem a analitzar l'alternativa consistent en:

$$B'(d) = \{R_{7,8} | \{d-1, d\}, R_{1,2,3,4,5} | (R_2 | DM(d)) [1], R_9 | DM(d)\}$$

és a dir,  $B'(d)$  conté: (1) els saldos i els límits de crèdits vigents en el dia  $d$  i en  $d-1$ , ja que aquests darrers encara no s'han esborrat; (2) totes les transaccions originades durant el mes actual, des del primer dia del mes fins el dia  $d$ ; i (3) els excessos de crèdits produïts en els mateixos dies. És fàcil de verificar que  $P[Q_j(d), B'(d)]$ , --  $j=1+n, \forall d \in T(Q_j)$ .

Aplicant (5.11) obtenim  $A(d)$ :

$$A(d) = B'(d) \dot{-} B'(d-1) = \{R_{7,8} | d, R_{1,2,3,4,5} | (R_2 | d) [1], R_9 | d\}$$

ja que segons (A1.3)  $DM(d) - DM(d-1) = d$

$B(d)$  s'obté mitjançant (5.10):

$$B(d) = B'(d) \dot{-} A(d) = \{R_{7,8} | d-1, R_{1,2,3,4,5} | (R_2 | DAM(d)) [1], R_9 | DAM(d)\}$$

ja que segons (A1.2)  $DM(d) - \{d\} = DAM(d)$ . Es verifica fàcilment que  $P[A(d), B(d) \cup I(d)]$ .

Finalment, aplicant (5.2) obtenim  $D(d)$ :

$$D(d) = B'(d) \dot{-} B''(d) = B'(d) \dot{-} B(d+1) = \{R_{7,8} | d-1, R_{1,2,3,4,5} | (R_2 | DIES(MES(d))) [1], R_9 | DIES(MES(d))\}$$

amb  $d' = d \cap \bar{ÚLTIM-DIA}(MES(d))$ . És a dir, cada dia s'ha d'esborrar  $R_{7,8} | d-1$ . A més, quan  $d$  és una data de final de mes, s'ha d'esborrar les transaccions i els excessos de crèdit del mes.

Observi's que aquesta alternativa emmagatzema les transaccions del mes de cara a poder derivar, a final de mes, la consulta ---  $R_{10} | MES(d)$ . Una altra alternativa podria consistir a acumular dia a dia el nombre de -- transaccions per oficina, de manera que a fi final de mes ja es tindria calculat el valor - demanat i no caldria emmagatzemar les transaccions. Per a fer això, caldria afegir als tipus de relació de la figura 1, el tipus:

$R_{11}$  (OFICINA, DATA, NOMBRE-TRANSACCIONS-EN-EL MES)

i la precedència:

$P[R_{11}|r, \{R_{11}|ANT(r), R_{12}|((R_1|r[1]) [1] \cap (R_2|r[2]) [1])\}]$

de manera que la consulta es transforma en  $R_{11}|d$ . No continuarem l'exemple amb aquesta variant, perquè la mecànica és idèntica al cas anterior.

## 6. CONCLUSIONS

En aquest treball hem presentat l'anàlisi de precedències i hem mostrar la seva aplicació a la validació lògica de requeriments i a l'anàlisi lògica d'alternatives de contingut d'una BD. Creiem que el mètode presentat és suficientment general i permet de resoldre ambdós problemes. Alhora, és formalitzat i, per tant, eventualment susceptible d'ésser automatitzat.

## 7. BIBLIOGRAFIA

- /1/ OLIVE, A.: "Anàlisi de precedències entre continguts d'informació. Aplicació al model informacional". QÜESTIÓ, Vol. 3 nº 3, 1979, pp. 129-143.
- /2/ NIJSSEN, G.M.: "A gross architecture for the next generation database management systems". En Modelling in DBMS, North Holland, Amsterdam, 1976, pp. 1-24.
- /3/ SENKO, M.E.: "Data structures and data accessing in data base systems. Past, present, future". IBM Systems Journal, nº 3, 1977, pp. 208-257.
- /4/ BUBENKO, J.A., BERILD, S., LINDENCROHN-OHLIN, E., NACHMENS, S.: "From information requirements to DBTG-data structures" Proc. of the ACM-SIGMOD/SIGPLAN Conference on Data, ACM, N.Y., 1976, pp. 73-85.
- /5/ TEOREY, T.J., FRY, J.P.: "The logical record access approach to database design" Computing Surveys, Vol. 12, 1980, pp.179-211.
- /6/ CODD, E.F.: "A relational model for large

shared data banks". Comm. ACM, nº 13, - 1970, pp. 377-387.

- /7/ CODD, E.F.: "Further normalization of the data base relational model". En Data Base Systems, Ristin R. (Ed.), Prentice-Hall, Englewood Cliffs, N.J., 1972, pp. 33-64.
- /8/ CODD, E.F.: "Relational completeness of data base sublanguages". En Data Base Systems", Ristin R. (Ed.), Prentice-Hall Englewood Cliffs, N.J., 1972, pp. 65-98.
- /9/ LANFEBORS, B.: "Theoretical analysis of information systems". Studentlitteratur, Lund, 1966.
- /10/ NUNAMAKER, J.: "A methodology for the design and optimization of information processing systems". Proc. SJCC Vol. 38, AFIPS Press, Montvale, N.J., 1971, pp. 283-294.
- /11/ WATERS, S.J.: "CAM 01: A precedence analyzer". The Computer Journal, Vol. 19, pp. 122-126.
- /12/ KAHN, B.K.: "A method for describing information required by the database design process". Proc. ACM SIGMOD, Washington D.C., 1976, pp. 53-64.
- /13/ SUNDGREN, Bo.: "Theory of data bases", - Mason/Charter Publishers, Inc., U.S.A. - 1975.

## APÈNDIX 1

En aquest apèndix definim les funcions utilitzades i llurs propietats. En el que segueix  $d$ ,  $d'$  i  $d''$  són dates en format dia-mes-any,  $m$  és un mes en format mes-any,  $D$  és un conjunt de dates i  $M$  un conjunt de mesos.

Les funcions MES( $d$ ), DIES( $m$ ) i ÚLTIM-DIA( $m$ ) donen, respectivament, el mes de la data  $d$ , el conjunt de totes les dates del mes  $m$  i la data més gran del mes  $m$ . Extenent-ho a  $D$  i  $M$  tenim:

$MES(D) = \{m \mid \exists d (d \in D \wedge m = MES(d))\}$

$DIES(M) = \{d \mid \exists m (m \in M \wedge d \in DIES(m))\}$

$ÚLTIM-DIA(M) = \{d \mid \exists m (m \in M \wedge d = ÚLTIM-DIA(m))\}$

La funció DM(d) ens dóna totes les dates que van des del primer dia del mes de  $\underline{d}$  fins el dia  $\underline{d}$ , ambdues incloses. DAM(d) és idèntica a DM(d) però no inclou  $\underline{d}$ . Formalment:

$$DM(d) = \{d'' \mid \text{MES}(d'') = \text{MES}(d) \wedge d'' \leq d\}$$

$$DAM(d) = \{d'' \mid \text{MES}(d'') = \text{MES}(d) \wedge d'' < d\}$$

D'aquestes definicions es dedueixen les següents propietats:

$$DAM(d) \cup \{d\} = DM(d) \quad (A1.1)$$

$$DM(d) - \{d\} = DAM(d) \quad (A1.2)$$

$$DM(d) - DM(d-1) = d \quad (A1.3)$$

Sigui  $D_1 = \{d_1 \div d-1\}$  i  $D_2 = \{d \div d_2\}$  amb  $-- D_1 \cap D_2 = \emptyset$ . La intersecció  $D_1 \cap \text{DIES}(\text{MES}(D_2))$  serà buida quan  $\underline{d}$  sigui un primer dia de mes; en cas contrari, serà igual a tots els dies del mes de  $\underline{d}$  que estan a  $D_1$ . O sigui:

$$D_1 \cap \text{DIES}(\text{MES}(D_2)) = DAM(d) \quad (A1.4)$$

Els conjunts DM(d) i DAM(d+1) són idèntics - quan  $\underline{d}$  no és un final de mes. En canvi, quan  $d = \text{ÚLTIM-DIA}(\text{MES}(d))$ , llavors  $DM(d) - DAM(d+1) = \text{DIES}(\text{MES}(d))$ . Per tant:

$$DM(d) - DAM(d+1) = \text{DIES}(\text{MES}(d')) \quad (A1.5)$$

amb  $d' = \{d\} \cap \text{ÚLTIM-DIA}(\text{MES}(d))$ .

La funció ANT(d) dóna la data anterior a la  $\underline{d}$ . Extesa a D és:

$$\text{ANT}(D) = \{d \mid \exists d'' (d'' \in D \wedge d = \text{ANT}(d''))\}$$

Extesa a una relació que té un atribut data i d'altres que designem per t, és:

$$\text{ANT}(r) = \{\langle t, d \rangle \mid \exists \langle t, d'' \rangle (\langle t, d'' \rangle \in r \wedge d = \text{ANT}(d''))\}$$

## NOTES

<sup>1</sup> A fi de simplificar la notació, en les projeccions utilitzarem posicions d'atribut - en comptes d'atribut.

<sup>2</sup> Com és usual, quan un conjunt està format per un sol element escrivim {d} o d indistintament.

<sup>3</sup> A fi d'alleugerir la notació, considerarem

equivalents les expressions  $R_i \mid X, R_j \mid X, R_k \mid X$  i  $R_{i,j,k} \mid X$ .

Fem la hipòtesi de ccis normalitzats perquè ens limitem al cas de BD centralitzades. -- Per a BD repartides, que no considerem en aquest article, la hipòtesi no és vàlida a nivell global però sí a nivell local.

