Aplicaciones

Actualització de vistes en bases de dades relacionals: estat de la qüestió

Ernest Teniente i López
Prof. del Departament de Llenguatges i Sistemes Informàtics
Universitat Politècnica de Catalunya

0. Introducció

Sovint és desitjable integrar en la base de dades una definició del coneixement comú compartit per diferents usuaris per a facilitar el desenvolupament d'aplicacions, per a evitar la múltiple implementació de programes amb funcionalitats similars o bé per a proporcionar capacitat de raonament.

En bases de dades relacionals aquest coneixement es pot expressar mitjançant les *vistes* o *esquemes externs* .Quan l'usuari vol actualitzar una vista, la seva petició s'ha de traduir en un conjunt de modificacions de les relacions bàsiques que estan físicament emmagatzemades a la base de dades. *L'actualització de vistes* consisteix a determinar com s'ha d'efectuar aquesta traducció. Durant els darrers anys, s'ha realitzat mol-ta recerca en el camp de les bases de dades relacionals i en el de les deductives per a intentar resoldre aquest problema. Aquest article és una síntesi dels diferents mètodes que s'han proposat fins el moment actual en el camp relacional.

Es distingeixen bàsicament dos enfocaments. El primer suggereix que les vistes es tractin com a *tipus abstractes de dades*. D'aquesta manera, la definició d'una vista inclou la traducció que s'ha d'aplicar quan aquesta ha de ser actualitzada. L'altre enfocament consisteix en definir un procediment general de traducció (un *traductor*) que tradueix automàticament qualsevol actualització de vista en modificacions de les relacions bàsiques corresponents.

Aquest article s'estructura de la manera següent. A la secció 1 es repassen alguns conceptes importants i es descriuen detalladament tots els aspectes referents a l'actualització de vistes. A les seccions 2 i 3 es descriuen l'enfocament dels tipus abstractes de dades i el dels traductors automàtics, respectivament. Finalment, a la secció 4 es presenten dues propostes que tracten amb bases de dades o amb vistes no convencionals. Concretament s'estudien els casos de bases de dades semàntiques i de vistes basades en objectes.

1. Descripció del problema

En bases de dades es fa servir el concepte de vista o *esquema extern* per a delimitar aquells continguts de la base de dades rellevants per a cada grup d'usuaris. Una vista és una estructura de dades virtual, derivada mitjançant una funció de definició a partir dels fets bàsics o d'altres vistes. Per tant, l'extensió d'una vista no té una existència independent, sinó que està completament definida per l'aplicació de la funció de definició a l'extensió actual de la base de dades.

Exemple: considerem una base de dades relacional que conté una relació bàsica d'empleats (EMP) amb els atributs empleat, sou i departament, i una altra de departaments (DEPT) amb els atributs departament i director del departament. Suposem que el contingut de la base de dades és el següent:

EMP	(Emp,	Sou,	Dept)
	Joan	12.000	Vendes
	Toni	15.000	Vendes
	Enric	10.000	Finances

DEPT	(Dept,	Dir)
	Vendes	Maria
	Finances	Dolors

Sobre aquesta base de dades definim mitjançant l'àlgebra relacional una vista V1 que ens diu qui és el director de cada empleat:

V1 (Emp, Dir)
f1 = (EMP * DEPT) [Emp, Dir]

V1	(Emp,	Dir)
	Joan	Maria
	Toni	Maria
	Enric	Dolors

Avantatges principals de les vistes

- **simplifiquen la interfície amb l'usuari:** permeten que aquest ignori les dades que no li interessen. En l'exemple anterior, l'usuari que interaccioni amb la vista V1 no s'ha de preocupar dels sous ni dels departaments dels diferents empleats.

- **afavoreixen la *i*ndependència lògica de les dades:** fan possible que es pugui variar l'estructura lògica de la base de dades sense necessitat d'efectuar els canvis corresponents en els programes. Per exemple, suposem que, per algun motiu determinat, la relació bàsica d'empleats ha de ser substituïda per dues de les seves projeccions, per exemple R1 (Emp,Dept) i R2 (Emp,Sou), de tal manera que la relació bàsica EMP es pot reconstruir a partir d'aquestes. En aquest cas, definint la vista EMP com una join de les relacions bàsiques R1 i R2, no caldria modificar les aplicacions que accedeixen a l'antiga relació bàsica EMP. Aquest avantatge no es pot assolir si la base de dades no disposa de la capacitat per a definir i manipular vistes.

- **proporcionen una mesura de protecció:** impedeixen que l'usuari accedeixi a dades externes a la seva vista. En l'exemple anterior, l'usuari que treballi amb la vista V1 no té accés al sou dels diferents empleats de l'empresa.

Les operacions que s'efectuen sobre una vista no es poden realitzar directament, sinó que s'han de traduir a operacions equivalents sobre l'estructura de la base de dades. En un entorn ideal, l'usuari no hauria de distingir entre les relacions bàsiques i les vistes i, per tant, els llenguatges de consulta i els

de manipulació s'haurien d'aplicar de la mateixa manera en ambdós casos.

Els usuaris interaccionen amb les vistes mitjançant consultes i actualitzacions. Les **consultes** de vistes no recursives no presenten cap problema conceptual especial. Una consulta sobre la vista es processa component-la amb la definició de vista i s'obté així una consulta equivalent expressada en termes de les relacions bàsiques [Sto75]. En canvi, el cas de les vistes recursives encara no està completament resolt. Vegeu [BR86] per a una síntesi dels diferents mètodes proposats en aquest sentit. De manera similar a les consultes, qualsevol petició d'**actualització** de vista haurà de ser traduïda a modificacions dels fets bàsics corresponents. Un cop aplicades aquestes modificacions, el nou estat de la base de dades induirà un nou estat de la vista. Es pretén aconseguir que el nou estat es correspongui al màxim amb la realització de la petició directament sobre la vista original. Aquest procés es pot descriure mitjançant la **figura 1** [Kel85]. En aquest diagrama, BD és una base de dades qualsevol i V una funció de definició de vista. Quan l'usuari especifica una actualització U sobre la vista V(BD), aquesta es tradueix en un conjunt de modificacions dels fets bàsics T(U). Aquestes modificacions generen BD' quan s'apliquen a la base de dades actual. Aleshores, el nou estat de la vista és V(BD'). Vegeu que, en alguns casos, la vista no haurà canviat d'acord amb la petició de l'usuari. Aleshores, V(BD') serà diferent d'U(V(BD)). El processament de les actualitzacions de vista planteja alguns problemes importants:

- en general, la modificació de la base de dades que satisfà una petició d'actualització de vista no és única. Consideri's una altra vegada l'exemple anterior. Per a aconseguir que la Dolors deixi de ser la directora de l'Enric, es pot esborrar el fet que l'Enric és un empleat del departament de Finances (EMP(Enric,Finances)) o bé que la Dolors és la directora d'aquest departament (DEPT (Finances,Dolors)).

- una modificació de la base de dades pot produir *efectes secundaris* en la vista. No es produiran efectes secundaris si U(V(BD)) = V(BD'), és a dir, si la vista ha canviat exactament d'acord amb la petició de l'usuari. Els efectes secundaris són indesitjables i s'haurien d'evitar sempre que es pugui. En l'exemple anterior, per a aconseguir que la Maria deixi de ser la directora d'en Joan (és a dir, per a esborrar el fet vista V1(Joan,Maria)) es pot esborrar el tuple DEPT (Vendes,Maria). Malauradament, aquest esborrat també provocaria que la Maria deixés de ser la directora d'en Toni, i aquesta podria no ser la nostra intenció original.

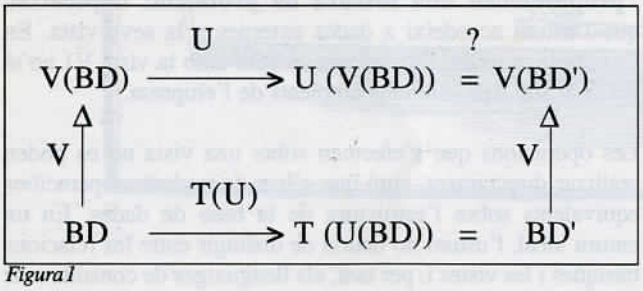

Figura 1

- quan es modifica una base de dades que ha de satisfer restriccions d'integritat es necessita algun mecanisme addicional per a comprovar que la base de dades actualitzada també satisfà aquestes restriccions.

El problema de l'actualització de vistes consisteix a determinar com s'ha de traduir apropiadament una petició d'actualització de vista en un conjunt de modificacions dels fets bàsics corresponents. La definició de traducció correcta pot variar amb la intenció de l'usuari i inclús dependre de l'estat de la base de dades en el moment de la petició. Per aquest motiu, la majoria d'autors defineixen el seu propi criteri de correctesa. S'han proposat dos enfocaments principals per a resoldre aquest problema. El primer suggereix que les vistes es tractin com a *tipus abstractes de dades*, de tal manera que la definició de la vista inclogui una descripció del conjunt d'actualitzacions de vista permeses, conjuntament amb la seva traducció. El segon enfocament consisteix a definir un procediment general de traducció (un *traductor*). Les entrades del traductor són la definició de la vista, l'actualització de vista demanada i l'estat actual de la base de dades. La sortida del traductor són les modificacions de la base de dades que satisfan la petició.

2. Enfocament dels tipus abstractes de dades

Els tipus abstractes de dades caracteritzen els objectes d'un tipus per les operacions que en ells es realitzen i amaguen a l'usuari els detalls de la seva representació interna. Aquesta idea es pot aplicar per a resoldre el problema de l'actualització de vistes. En aquest cas, la definició de la vista especifica explícitament, per cada tipus d'actualització permesa, quina és la seva traducció. L'usuari que defineix la vista (p.ex. l'administrador de la base de dades) serà l'encarregat de prendre aquesta decisió. L'enfocament dels *tipus abstractes de dades* té alguns avantatges importants. En primer lloc, la definició explícita de la traducció escollida resol el problema de l'existència de múltiples solucions. A més, peticions del tipus "incrementa en 1000 pts el promig del sou dels empleats de determinat departament" es poden tractar més fàcilment que en l'enfocament dels traductors automàtics. Finalment, la traducció seguint aquest enfocament es pot implementar en els sistemes de gestió de bases de dades actuals (vegeu [TFC83] per a un exemple concret d'implementació en el cas de SQL/DS). Malgrat aquests avantatges, convé remarcar que s'aconsella introduir característiques addicionals en el llenguatge de definició de vistes per a seguir aquest enfocament.

Exemple [FC85]: Considerem una base de dades que conté una relació bàsica d'empleats (Emp) amb els atributs empleat i departament al qual està assignat i una altra de departaments (Dpt) amb els atributs departament i director del departament. Aquestes relacions estan subjecte a les dependències funcionals següents: Emp: EMP -> DEPT i Dpt: DEPT -> DIR. En aquest exemple, un mòdul s'ha d'entendre com un conjunt d'estructures de la base de dades més un conjunt d'operacions sobre aquestes estructures. El mòdul SUBORDINACIÓ permet que un usuari assigni un empleat a un director de

departament. La definició d'aquest mòdul és:

module SUBORDINACIÓ
  schemes
    EM [EMP, DIR]
  constraints
    (EM(e,m) ∧ EM(e,n) => m=n)
  operations
    ASSIGNAR(e,m): afegir (e,m) a EM;
using
  view definition
    EM [EMP, DIR]: {(e,m) / (Emp(e,d) i Dpt(d,m)) }
  operations translation
    ASSIGNAR(e,m) si ∃d Dpt(d,m) ∧ ¬∃d' Emp(e,d')
    llavors afegir (e,d) a Emp;
end-module

El mòdul SUBORDINACIÓ defineix una vista EM(Emp,Dir), que diu qui és el director de cada empleat. Aquest vista ha de satisfer totes les restriccions d'integritat especificades a partir de la paraula *constraints* . En l'exemple, només hi ha una restricció que imposa que un empleat no pot tenir dos directors diferents. Finalment, trobem la definició de les operacions permeses sobre la vista EM. En aquest cas, l'única operació que es pot efectuar és l'assignació d'empleats a directors de departament (ASSIGNAR(e,m)). Els usuaris que treballen amb aquest mòdul només coneixen aquestes tres parts: el nom i atributs de la vista, les restriccions d'integritat i les operacions permeses. Les definicions que apareixen sota la clàusula *using* no són visibles a l'usuari, ja que es corresponen a la definició de la vista i a la traducció de les seves actualitzacions. En aquest cas, l'operació ASSIGNAR(X,Y) es tradueix en una assignació de l'empleat X al departament dirigit per Y sempre i quan X no sigui ja un empleat i Y ja sigui un director de departament. A [TFC83] s'explica com es podria implementar aquest mòdul en el sistema de gestió de base de dades SQL/DS. Com a exemple dels efectes de l'operació ASSIGNAR, suposem que el contingut de la base de dades i de la vista és el següent:

Emp	EMP	DEPT	Dpt	DEPT	DIR	EM	EMP	DIR
	Joan	Vendes		Vendes	Maria		Joan	Maria
	Pere	Finances		Finances	Anna		Pere	Anna
	Alícia	Vendes					Alícia	Maria

Donada la traducció de l'operació ASSIGNAR especificada en el mòdul anterior i l'estat actual de la base de dades tindrem:

- ASSIGNAR (Joan, Anna) falla perquè en Joan ja és un empleat.
- ASSIGNAR (Robert, Laura) falla perquè la Laura no dirigeix cap departament.
- ASSIGNAR (Robert, Anna) té èxit perquè en Robert no és un empleat i l'Anna dirigeix el departament de Finances. En aquest cas, l'actualització de vista demanada es traduirà en la inserció del tuple Emp(Robert, Finances).

Una altra proposta que segueix l'enfocament dels tipus abstractes de dades [SM89] utilitza el llenguatge de programació i consultes NETUL per a definir vistes. Una vista és considerada com una funció d'aquest llenguatge i l'usuari pot definir procediments específics de traducció. Aleshores, una petició d'actualització de vista, combinada amb la definició de la vista, es passa com a paràmetre al procediment definit per l'usuari, que s'encarrega d'efectuar la traducció.

La contribució principal d'aquest treball és que, en l'entorn considerat, el poder de les definicions de vista és igual al poder de les funcions en els llenguatges de programació. És a dir, il.limitat. Per tant aquest mètode no presenta cap problema a l'hora de definir vistes recursives. Manchanda i Warren [MW 88] suggereixen una manera per a aplicar aquest enfoca-ment en el camp de les bases de dades lògiques. Per a fer-ho, defineixen el llenguatge DLP (*Dynamic Logic Programming*) que permet definir la inserció i l'esborrat d'un tuple de la base de dades. En aquest llenguatge, la inserció del tuple ED (Joan, Vendes) a la relació bàsica ED (Empleat,Dept) s'indica com ← <+ED (Joan,Vendes)> i l'esborrat del mateix tuple s'indica mitjançant ← <-ED(Joan, Vendes)>. Aquests operadors són usats per a definir la traducció d'una actualització de vista.

Com a exemple, considerem la la vista EDP (Emp, Dept, Proj) definida de la manera següent:

EDP (Emp, Dept, Proj) ← ED(Emp, Dept) ∧ EP (Emp, Proj)
on ED i EP es corresponen a les relacions bàsiques Empleat-Departament i Empleat-Projecte, respectivament. En aquest llenguatge, l'esborrat del tuple (Joan, Vendes, Proj1) de la vista EDP s'especificarà com ← <-EDP(Joan, Vendes, Proj1)>.

En general, l'esborrat d'aquest tuple vista es pot satisfer de dues maneres diferents: esborrant ED(Joan,Vendes) o esborrant EP (Joan,Proj1). L'administrador de la base de dades pot decidir que l'usuari que treballa amb la vista EDP té autoritat per a donar de baixa empleats d'un projecte, però no d'un departament. Aleshores, pot indicar la traducció desitjada especificant el següent traductor d'esborrats de la vista:

<-EDP(Emp,Dept,Proj)> ← ED(Emp,Dept) ∧<-EP(Emp, Proj)>

La semàntica d'aquest traductor és que l'esborrat d'un tuple de la vista es tradueix sempre en l'esborrat d'un tuple de la relació bàsica EP. De manera similar, es podria especificar la traducció de les peticions d'inserció.

3. Enfocament dels traductors automàtics

L'enfocament dels tipus abstractes de dades, descrit en el capítol anterior, presenta dos inconvenients importants. Primerament, cal destacar el treball necessari per a definir les traduccions i la possibilitat de què aquestes s'hagin de redefinir durant la vida de la base de dades. A més, en alguns casos, l'especificació d'una única traducció que satisfà l'actualització d'una vista pot no ser desitjable. Considerem una altra vegada el darrer exemple de la secció anterior. Si l'usuari de la vista EDP estigués autoritzat per a donar de baixa empleats d'un projecte així com d'un departament, seria impossible especi-

ficar una única traducció en temps de definició de la vista. Aleshores, hauria de ser l'usuari qui, en temps d'execució, decidís quina és la traducció més adient en cada cas.

L'enfocament dels *traductors automàtics* està pensat per a superar aquestes limitacions. Aquest enfocament consisteix a definir un *traductor* que, a partir de la definició de la vista, l'actualització de vista demanada i l'estat actual de la base de dades, obté automàticament el conjunt de modificacions de les relacions bàsiques que satisfan la petició. Cal tenir en compte que, en general, la traducció que satisfà una actualització de vista no és única i, per tant, es necessita algun mecanisme per a escollir la traducció que s'aplicarà. Els mètodes que segueixen aquest enfocament es poden classificar de la manera següent: **complement d'una vista** [Ban79, Spy80, BS81, KU84, CP84, Kel87, Heg90]; **traducció segons el tipus de vista** [FSS79, CA79, Mas84, Kel85, Kel86a, Kel86b, Dat86, SLW88, LS91]; **hipòtesi de la relació universal** [BV88, Lan90]. Cadascuna d'aquestes classes es descriu en detall a les seccions 3.1, 3.2 i 3.3.

Abans, però, de descriure tots aquests mètodes anem a presentar un treball [DB78, DB82] que es pot aplicar en tots els casos anteriors. En aquest, es presenten criteris de correctesa per a traduir actualitzacions de vista i es dedueixen condicions per a conèixer si un procediment de traducció determinat produeix traduccions correctes segons els criteris anteriors. Es consideren les 4 hipòtesis següents:

1. les úniques restriccions de l'esquema conceptual són les claus de les relacions bàsiques i les dependències funcionals del tipus R:A-->B, on A i B són atributs de la relació R. Cada relació bàsica ha de tenir una clau primària.

2. les vistes es poden definir només mitjançant projeccions, seleccions i equi-joins.

3. les modificacions de vista poden ser esborrats i modificacions d'un o més tuples o bé insercions d'un únic tuple.

4. només es consideren procediments que tradueixen actualitzacions de vista en seqüències de modificacions del mateix tipus. Per exemple, esborrats d'una vista en esborrats de les relacions bàsiques.

A [DB82] es defineix el concepte de *traducció* com un conjunt de modificacions de la base de dades que satisfà l'actualització de la vista i que preserva la consistència semàntica. Una *traducció exacta* és aquella que no produeix efectes secundaris. Les condicions per a conèixer si un traductor genera traduccions correctes s'obtenen a partir de la definició de vista i de les dependències funcionals de l'esquema conceptual.

Per exemple, considerem un esquema de base de dades que conté les relacion bàsiques EMP(Emp,Dept) i DEPT(Dept,Dir) i les dependències funcionals EMP:Emp-->Dept i DEPT: Dept-->Dir. Mitjançant SQL es defineix la vista EDM (Emp, Dept, Dir) de la manera següent:

```
DEFINE VIEW EDM (Emp, Dept, Dir)
    AS SELECT EMP.Emp, EMP.Dept, DEPT.Dir
        FROM EMP, DEPT
        WHERE EMP.Dept = DEPT.Dept
```

A partir de la definició de la vista i de les dependències funcionals s'obté un graf (anomenat *graf de dependències de la vista*) que s'utilitzarà per a determinar si un traductor genera traduccions correctes. El graf de dependències de la vista EDM és a la **figura 2.** En aquest exemple concret, els nodes es corresponen a atributs de la vista o de les relacions bàsiques. Els arcs bidireccionals se situen entre els atributs de la vista i els atributs de les relacions bàsiques que els defineixien (per exemple, entre EDM.Emp i EMP.Emp) i entre els atributs que apareixen en la igualtat de la qualificació (per exemple, entre EMP.Dept = DEPT.Dept). Els arcs unidireccionals representen la informació subministrada per les dependències funcionals. Vegem com es pot usar aquest graf per a derivar condicions de traduïbilitat pels esborrats. Dayal i Bernstein demostren que una traducció d'esborrat és exacta quan per cada atribut B de la vista existeix un atribut A de la relació bàsica a esborrar tal que hi ha un camí entre A i B en el graf de dependències de la vista. A [DB82] es poden trobar condicions similars pel cas d'insercions i de modificacions.

Per exemple, un procediment que tradueixi peticions d'esborrat de tuples de la vista EDM en esborrats de tuples de la relació bàsica EMP és un traductor exacte ja que hi ha un camí d'EMP.Dept a tots els atributs de la vista EDM. En canvi, com que no existeix cap camí que partint d'un atribut de DEPT ens porti a EDM.Emp el procediment que tradueix peticions d'esborrat de tuples de la vista EDM en esborrats de tuples de la relació bàsica DEPT no és un traductor exacte. Observi's que en aquest darrer cas, es podrien produir efectes secundaris. En l'article [DB82] també es presenten les condicions necessàries per a conèixer si un determinat procediment és un traductor i es demostra que les traduccions úniques i exactes només es poden aconseguir sota condicions bastant rigoroses.

3.1. Complement d'una vista

Bancilhon i Spyratos [Ban79, Spy80, BS81] proposen un mètode molt elegant per a traduir actualitzacions de vista. A cada vista se li associa un *complement* que descriu la informació no visible en la vista. Intuïtivament, un complement d'una vista és la informació addicional que cal donar conjuntament amb la vista per a poder calcular tota la base de dades. Aleshores, una actualització de vista es tradueix de tal manera

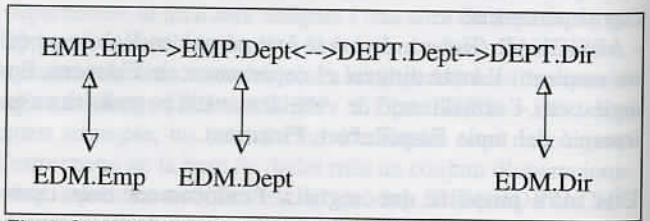

Figura 2: graf de dependències de la vista EDM

que només es canvia la vista, mentre que el seu complement es manté invariant.

Què s'entén per *la informació no visible en una vista* ? Consideri's per exemple una base de dades que conté una relació bàsica d'empleats EMP amb els atributs nom de l'empleat i departament al qual està assignat i una altra de departaments DEPT amb els atributs nom i director del departament. Suposem que el contingut d'aquestes relacions és el següent:

EMP	(Emp,	Dept)	DEPT	(Dept,	Dir)
	Joan	Vendes		Vendes	Maria
	Toni	Vendes		Finances	Dolors
	Enric	Finances			

Sobre aquesta base de dades es defineixen dues vistes: V1 i V2. Un fet vista V1(e,d) diu que l'empleat e està assignat al departament d i la vista V2 ens dóna informació de qui és el director de cada departament.

V1 (Emp, Dept)
f1 = EMP [Emp, Dept]

V1	(Emp,	Dept)
	Joan	Vendes
	Toni	Vendes
	Enric	Finances

V2 (Emp, Dir)
f2 = (EMP * DEPT) [Emp, Dir]

V2	(Emp,	Dir)
	Joan	Maria
	Toni	Maria
	Enric	Dolors

La vista V1 s'ha definit restringint la nostra atenció a la relació bàsica d'empleats. Aleshores, un tuple d'aquesta vista es pot actualitzar fàcilment mantenint el contingut de la relació DEPT invariant. En aquest cas, la noció d'informació no visible en V1 és bastant simple ja que la base de dades es pot descompondre en dues parts independents. Malauradament, en el cas general serà molt més complicat definir el complement d'una vista. Per exemple, considerem la vista V2. És difícil definir exactament quina és la informació no visible en aquesta vista. La relació bàsica EMP podria ser el complement, però també ho podria ser DEPT. A més, interessa definir complements *mínims* d'una vista. Diem que un complement és mínim si no existeix cap altra vista que proporciona menys informació i que també es un complement. A [Ban79] es demostra que una vista pot tenir diversos complements mínims. En l'exemple anterior, tant EMP com DEPT són complements mínims de V2. Bancilhon i Spyratos argumenten que l'existència de múltiples complements es correspon a l'existència de diferents polítiques d'actualització de vistes. Lògicament, la tria d'una política dependrà de cada aplicació concreta. L'exemple següent serveix per a constatar aquesta informació.

Considerem una altra vegada l'exemple anterior i suposem que l'usuari vol esborrar de la vista V2 que la Dolors és la directora de l'Enric. Si definim EMP com a complement de V2, aquesta petició es traduirà en l'esborrat del tuple DEPT(Finances,Dolors). En canvi, si DEPT fos el complement de V2 escollit, la traducció es reduiria a suprimir el tuple EMP(Enric,Finances). És a dir, en el primer cas la política d'actualització de vistes rau a destituir directors de departaments, mentre que en el segon consisteix a acomiadar els empleats. Es poden destacar tres **resultats** importants d'aquesta teoria:

- donada una vista, sempre n'existeix un complement.
- donada una vista i un complement, la traducció que satisfà la petició de modificació i deixa invariant el complement (si existeix) és única.
- a [BS81] es presenta una fórmula que donada una petició d'actualització traduïble permet calcular la seva traducció.

Com els mateixos autors reconeixen, l'inconvenient principal d'aquesta teoria rau en la seva implementació pràctica. En aquest sentit, a [CP84] s'estudien els problemes que sorgeixen quan s'intenta aplicar aquest enfocament en el context del model relacional. En concret s'estudia un àmbit reduït d'aplicació en què l'esquema de la base de dades conté una única relació R (hipòtesi de la relació universal) i un conjunt de dependències funcionals i on totes les vistes es defineixen mitjançant projeccions d'aquesta relació.

En aquest entorn es dedueixen les condicions per a conèixer quan dues vistes són complementàries i es presenten les condicions necessàries i suficients per a implementar la inserció, l'esborrat i la modificació d'un tuple en una vista, mantenint el complement invariant (considerant que quan l'usuari especifica una vista, també especifica el seu complement). El resultat més important d'aquest treball és la demostració que trobar el complement més petit (p.ex. el complement amb el menor nombre d'atributs) és un problema NP-complet.

Una altra crítica que s'ha fet a aquesta teoria [KU84, Kel87] és que es poden refusar actualitzacions de vista que tenen traducció pel fet que aquesta no manté el complement invariant. Per tant, en alguns casos pot ser massa restrictiu imposar aquesta condició.

En l'exemple següent [Kel87], es presenta un traductor que és força raonable però que no preserva cap complement. Suposem que tenim la relació AB, amb dos atributs A i B, i la dependència funcional A --> B. Suposem que el domini d'A conté com a mínim un element a1, i el domini de B en contingui com a mínim dos, b1 i b2. Definim la vista V com la selecció de tots els tuples d'AB on B=b1. El traductor següent no preserva cap dels possibles complements d'aquesta vista: Insert tuple <a,b>: Si existeix un tuple <a,y>, aleshores reemplaçar <a,y> per <a,b>, altrament, inserir <a,b>.

Per a resoldre aquest problema, a [KU84] es proposa restringir el tipus dels complements seleccionats quan es desitgi acceptar totes les actualitzacions traduïbles. Per a fer-ho es defineix el concepte de vistes *independents* . Si dues vistes són independents i complementàries es pot mantenir l'estat d'una vista invariant mentre es genera qualsevol estat possible de l'altra vista. Per tant, la tria d'un complement independent permet que totes les actualitzacions que s'expressen contra la vista es puguin traduir a modificacions de la base de dades.

3.2. Traducció segons el tipus de vista

Dins l'enfocament dels **Traductors Automàtics**, una segona alternativa per a resoldre el problema de l'actualització de vistes consisteix a definir per cada operador de definició d'una vista quina és la seva traducció. Per cada operador (selecció, projecció, join, etc.) i tipus d'actualització (inserció, esborrat, modificació) es donarà la seva traducció expressada en modificacions de les relacions bàsiques corresponents. Així, per exemple, es pot definir una regla que ens diu quines són les modificacions de les relacions bàsiques que cal efectuar per a satisfer una petició d'esborrat de qualsevol vista definida mitjançant una projecció.

Malgrat diferir en el tipus d'operadors tractats, les diferents propostes realitzades en aquest sentit [FSS79, Mas84, Kel85, Kel86a, Dat86, LS91] es poden resumir en la **figura 3**. Aquesta taula és aplicable per a actualitzar un únic tuple d'una vista, però exceptuant el cas de la join, també serveix per a actualitzacions múltiples, tenint en compte que en realitza una cada vegada. Per simplificar la presentació, no hem considerat el cas de peticions de modificació [FSS79, Kel85, Dat86] ni els operadors producte cartesià [FSS79, CA79] i intersecció [LS91]. La traducció en aquests casos es pot trobar a les referències esmentades.

No tots els autors proposen totes les traduccions expressades en la **figura 3,** sinó que en algunes referències es consideren casos particulars d'aquestes. Aquest fet serà estudiat detalladament quan es comenti la traducció de cada operador concret. L'operació Inserir en R (resp. Esborrar de R) es defineix de tal manera que no produeixi cap efecte sobre R en el cas que el tuple que s'ha d'inserir (resp. esborrar) pertanyi (resp. no pertanyi) a la base de dades. Com es pot comprovar, en alguns casos existeixen diverses traduccions que satisfan una actualització de vista. Aquest fet denota una ambigüitat semàntica en el sentit que manca coneixement per a escollir quina és la traducció més adient. Més endavant explicarem les solucions proposades per a resoldre aquest problema. De la mateixa manera que [Mas84, LS91] anomenarem aquests casos ***Problemes d'Ambigüitat Semàntica (SAP).***

Projecció : Per a simplificar, només hem considerat el cas en què els atributs de la vista V inclouen tots els atributs clau de la relació bàsica R. Aleshores, l'esborrat d'un tuple de la vista es tradueix en un esborrat d'un tuple de la relació bàsica subja-cent. Una inserció d'un tuple en la vista es tradueix també en la inserció d'un tuple bàsic. En aquest darrer cas, caldrà deter-minar els valors dels atributs de R que no apareixen a V. La majoria de les propostes realitzades [FSS79, Mas84, Kel85, Dat86] recomanen l'ús del valor NUL per aquests atributs, però també es podrien usar valors per defecte o bé podria ser que el valor d'aquests atributs que no apareixen a la vista depengués dels atributs que són visibles en ella [LS91]. Per tant, en peticions d'inserció ens trobem amb un problema d'ambigüitat semàntica *(SAP-PROJ)* ja que necessitem coneixement addicional per a determinar exactament el valor dels atributs invisibles en la vista.

Selecció: Alguns autors [FSS79, Mas84, Dat86] proposen que les insercions i esborrats d'una vista es tradueixin sempre

	INSERCIONS	ESBORRATS
PROJECCIÓ: V = R [X] [FSS79], [Mas84], [Kel85], [Dat86], [LS91]	Inserir en R *SAP-PROJ*	Esborrar de R
SELECCIÓ: V = R(XΘY) [FSS79], [Mas84], [Kel85], [Dat86], [LS91]	Inserir en R o Modificar atribut selecció	Esborrar de R o Modificar atribut selecció *SAP-SEL*
UNIÓ: V = R ∪ S [FSS79], [Mas84], [Kel86a], [LS91]	Inserir en R o Inserir en S o ambdues *SAP-UNIÓ*	Esborrar de R i Esborrar de S
DIFERÈNCIA: V = R - S [Mas84], [LS91]	Inserir en R i Esborrar de S	Esborrar de R o Inserir en S o ambdues *SAP-DIF*
JOIN: V = R[X=Y]S [Kel85], [Dat86], [LS91]	Inserir en R i Inserir en S	Esborrar de R o Esborrar de S o ambdues *SAP-JOIN*

Figura 3: Traducció. per cada operador.

en insercions i esborrats de les relacions bàsiques que apareixen en la definició de la vista. En canvi, [Kel85] i [LS91] consideren que en alguns casos és millor traduir una actualització d'una vista en una inserció o supressió de la relació bàsica corresponent i que en altres és millor traduir-la en una modificació dels valors d'alguns atributs d'aquesta. L'exemple següent [Kel85] serveix per a il.lustrar aquest segon raonament. Suposem que tenim una relació bàsica d'empleats EMP amb els atributs codi d'empleat, nom, localització i un atribut que indica si l'empleat pertany a l'equip de futbol de l'empresa o no. L'empresa té dues localitzacions: Barcelona i Mataró. Sobre aquesta relació es defineix una vista que ens diu tots els empleats que treballen a Mataró: V1 = EMP (Localització = "Mataró"). Si volem esborrar l'empleat amb codi = 17 d'aquesta vista, una traducció raonable consisteix a esborrar l'empleat de la base de dades. Així, haurem traduït un esborrat d'una vista en un esborrat del tuple corresponent.

Suposem que tenim una altra vista que ens dóna els empleats que juguen a l'equip de futbol de l'empresa, V2 = EMP (Futbol = *Sí*). Sembla poc raonable traduir l'esborrat de l'empleat amb codi = 14 d'aquesta vista en un esborrat del tuple de l'empleat (si no és que considerem que el futbol és molt important). Una possibilitat més raonable consisteix en modificar el valor de l'atribut Futbol amb un *No* . Per tant, en aquest segon cas un esborrat d'una vista s'ha traduït en una modificació del valor de l'atribut de selecció.

En peticions d'esborrat, caldrà usar algun tipus de coneixement per a decidir quina opció es més raonable en cada cas *(SAP-SEL).* Observi's que, en peticions d'inserció, la tria d'una de les dues alternatives depèn de l'estat de la base de dades i per tant no es produeix cap problema d'ambigüitat semàntica.

Unió i Diferència: El raonament que s'ha d'aplicar en les vistes definides mitjançant unió o diferència és similar als anteriors. En aquests casos ens apareixen dos problemes nous d'ambigüitat semàntica, *SAP-UNIÓ* i *SAP-DIF.*

Join: Hi ha quatre condicions possibles en la restricció d'una join del tipus V = R[X=Y]S:

a) X = Clau(R) i Y = Clau(S)
b) X = Clau(R) i Y ¬=Clau(S)
c) X ¬=Clau(R) i Y = Clau(S)
c) X ¬=Clau(R) i Y ¬=Clau(S)

El cas a) es correspon a una join sobre la clau primària de les dues relacions. Els casos b) i c) es corresponen a una join entre la clau primària d'una relació i la forana de l'altra. [Kel85] i [Dat86] tracten els casos a), b) i c). A [LS91] es tracten també tres subcasos de la quarta condició. En la taula precedent s'ha expressat només la traducció corresponent als tres primers casos. Una petició d'inserció es tradueix en la inserció d'un tuple de cada relació bàsica que apareix en la definició de la vista. En esborrats d'una vista cal distingir dos casos diferents: en b) i c) l'esborrat de la vista es tradueix en un esborrat d'un tuple de la relació bàsica que participa en la join amb clau forana. En el cas a) es pot escollir entre l'esborrat d'un tuple de qualsevol de les relacions bàsiques que apareix en la definició de la vista o bé en l'esborrat d'ambdós. Aquest cas es correspon al problema d'ambigüitat semàntica *SAP-JOIN.*

L'exemple següent serveix per a il.lustrar l'enfocament descrit en aquesta secció. Considerem una base de dades que conté una relació bàsica d'empleats i una de departaments: EMP (Emp, Dept, Sou) i DEPT (Dept, Dir), on els atributs subratllats es corresponen a les claus primàries de cada relació.

Sobre aquesta base de dades es defineixen dues vistes:

V1 = EMP[Emp,Dept]; V2 = EMP[Dept =Dept]DEPT.

El contingut de la base de dades és el següent:

EMP	Emp	Dept	Sou
	Joan	D1	1000
	Pere	D1	1000
	Marc	D2	1500

DEPT	Dept	Dir
	D1	Maria
	D2	Anna

V1	Emp	Dept
	Joan	D1
	Pere	D1
	Marc	D2

V2	Emp	Dept	Sou	Dir
	Joan	D1	1000	Maria
	Pere	D1	1000	Maria
	Marc	D2	1500	Anna

Traducció per cadascuna de les actualitzacions de vista següents:

esborrar (V1(Marc, D2))=> esborrar (EMP(Marc, D2, 1500))

inserir(V2(Quim,D2,1500,Anna))=>inserir(EMP(Quim,D2,1500)),
inserir(DEPT(D2,Anna)).

Observi's que aquest segon inserir no modifica el contingut de la base de dades ja que el tuple DEPT (D2, Anna) ja hi pertany.

esborrar(V2(Joan,D1,1000,Maria)=>esborrar(EMP(Joan,D1,1000))

És a dir, en aquest cas un esborrat d'una vista definida mitjançant una join es tradueix en un esborrat de la relació bàsica que participa en la join amb clau forana.

Una proposta diferent la podem trobar a [CA79]. En aquest article es descriuen procediments generals de traducció per a cada tipus d'actualització de vista, independentment de quin hagi estat l'operador usat en la seva definició. Aquests traductors es poden aplicar a vistes definides mitjançant seleccions, projeccions i joins. Malgrat aquesta diferència a l'hora de presentar els traductors, les solucions obtingudes es corresponen a les expressades anteriorment.

Vistes definides mitjançant diversos operadors: Les traduccions expressades en la taula anterior es corresponen a vistes simples. És a dir, a vistes definides mitjançant un únic operador. No obstant, en general una vista es pot definir usant diversos operadors. En aquest cas, considerant la seqüència d'avaluació de l'expressió que defineix la vista podem iden-

tificar una seqüència de vistes V_1, V2,..., V_{k-1} corresponent de manera que, per a cada i = 2 ... k, la vista V_i es deriva a partir de la vista V_{i-1} mitjançant una única operació de definició. En la **figura 4** [FSS79] es pot veure com l'aplicació iterativa de les regles de traducció és suficient per a actualitzar vistes definides mitjançant qualsevol nombre d'operacions. La modificació demanada U_k sobre la vista V_k es transforma en una modificació U_{k-1} sobre la vista V_{k-1} d'acord amb l'operació mitjançant la qual V_k es va derivar de V_{k-1}. La modificació en V_{k-1} també es transforma en una modificació sobre V_{k-2}, i així successivament, fins que s'obté una modificació U0 de les relacions bàsiques RB. L'aplicació d'U0 sobre RB condueix a les noves relacions bàsiques RB'. A partir d'aquestes s'apliquen les diferents operacions de definició fins a obtenir la vista modificada V_k'.

Problemes d'ambigüitat semàntica: Com hem esmentat anteriorment, un dels problemes més importants de l'actualització de vistes és el de l'existència de múltiples traduccions que satisfan una petició d'actualització. Aquest problema s'anomena Problema d'Ambigüitat Semàntica (SAP). La tria arbitrària d'una solució és inacceptable ja que podria conduir a un estat de la base de dades semànticament inconsistent. Per tant, cal algun mecanisme addicional per a obtenir el coneixement necessari per a escollir la traducció més adient.

S'han realitzat diferents propostes per a resoldre aquest problema. Keller [Kel86a,Kel86b] suggereix que tota la semàntica necessària s'obtingui en el temps de definició de la vista. Això es pot aconseguir mitjançant un diàleg estructurat amb l'administrador de la base de dades. Els interrogants que es plantegen durant el diàleg es basen en la definició de la vista, en la informació estructural de l'esquema de la base de dades i en les preguntes fetes prèviament durant el diàleg. Segons les respostes que es donin s'obtindrà un traductor específic o altre. Aleshores, usant aquest traductor, les actualitzacions de vista es podran traduir unívocament en modificacions de la base de dades. El diàleg següent [Kel86b], per a desambiguar SAP-SEL, ens serveix com a exemple:

Preguntar: Es poden esborrar tuples d'una vista definida mitjançant seleccions ? Si no, acabar.

Preguntar: L'esborrat d'un tuple de la vista s'hauria de traduir en un esborrat del tuple bàsic de la relació subjacent o en una modificació? Si esborrat, aleshores l'esborrat d'un tuple de la vista se satisfà esborrant el tuple bàsic corresponent; acabar.

Preguntar: Quins dels atributs de selecció han de ser modificats? Si aquest atribut té més d'un valor excloent, aleshores ***preguntar*** quin valor s'ha d'usar ? L'esborrat d'un tuple de la vista se satisfà modificant el valor de l'atribut de selecció especificat pel valor d'exclusió especificat.

Segons les respostes que doni l'administrador de la base de

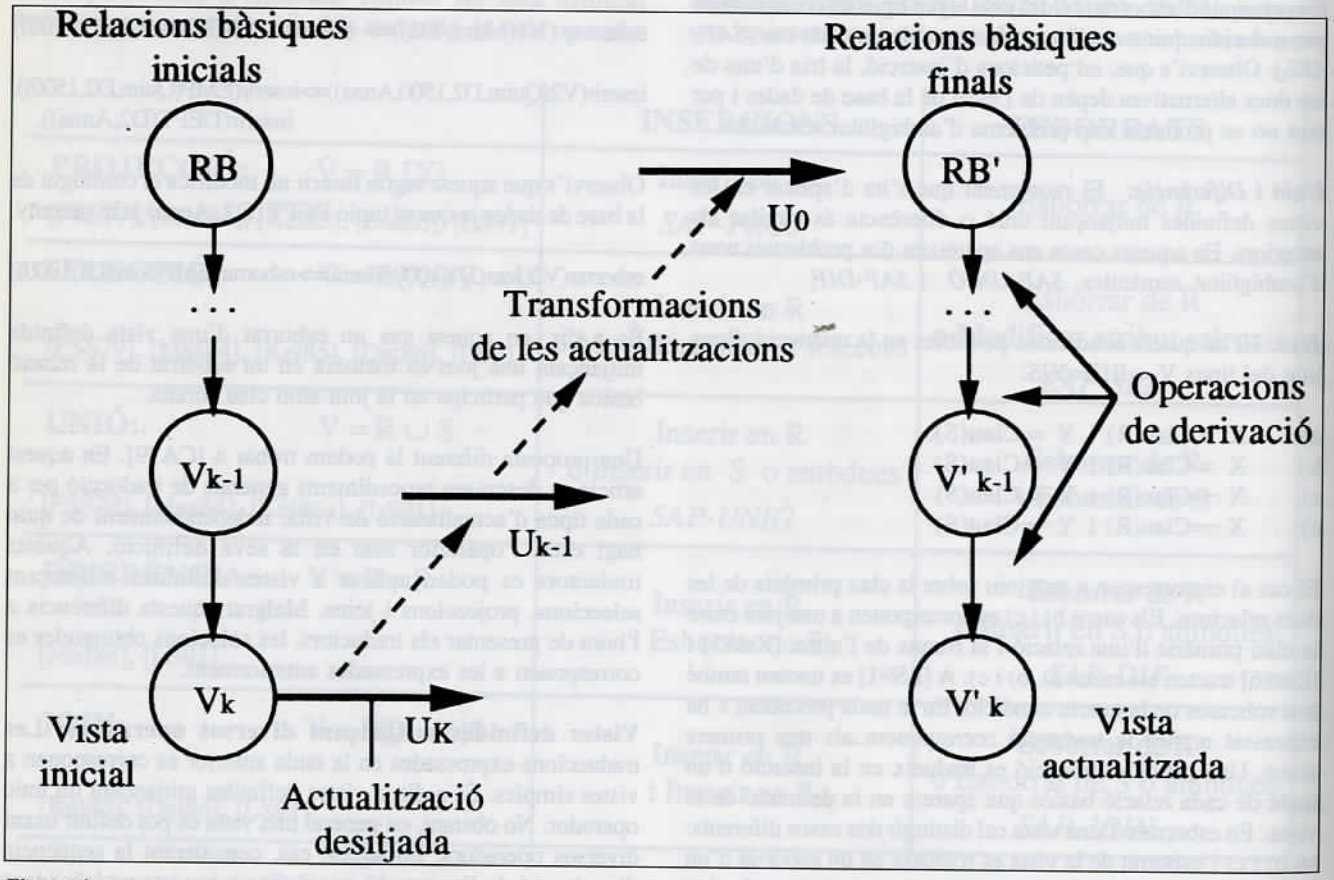

Figura 4

dades, l'esborrat d'un tuple d'una vista definida mitjançant selecció es traduirà en un esborrat o en una modificació del tuple bàsic corresponent. Per tant, en temps de definició s'especifica unívocament com s'ha de traduir una actualització de vista.

A diferència de Keller, Masunaga [Mas84] esmenta que, segons quina sigui la naturalesa de les ambigüitats semàntiques, caldrà la interacció amb l'usuari en temps d'execució. És a dir, en alguns casos els SAP es poden resoldre usant les restriccions d'integritat o dependències funcionals de la vista, però en altres es necessita la interacció humana per a escollir la millor traducció. La darrera proposta realitzada [SLW88,LS91] integra d'alguna manera les dues precedents. En aquesta, se suggereix que la informació semàntica necessària per a resoldre les ambigüitats s'obtingui tant en temps de definició com en temps d'execució. Segons aquesta proposta el coneixement semàntic es classifica en dos tipus diferents: *semàntica de la base de dades* i *semàntica de l'aplicació* . La semàntica de la base de dades està formada per les restriccions estructurals, com ara "CODI_EMP és la clau d'EMPLEATS", la informació extensional de la base de dades, com ara "hi ha un empleat amb CODI_EMP=323", i les restriccions d'integritat, com ara les dependències funcionals. El coneixement semàntic relatiu a l'aplicació s'anomena semàntica de l'aplicació.

Aquest coneixement el podrà especificar l'administrador de la base de dades en temps de definició o bé l'usuari en temps d'execució. La solució proposada consisteix en intentar resoldre les ambigüitats usant únicament el coneixement obtingut en temps de definició. Si aquest coneixement no és suficient, caldrà demanar a l'usuari el coneixement addicional necessari per a escollir la millor solució.

3.3. Hipòtesi de la relació universal

En aquest darrer apartat presentem els traductors automàtics basats en la *hipòtesi de la relació universal* [Sag83], segons la qual l'usuari veu la base de dades com si fos una única relació definida sobre l'univers d'atributs U. Els sistemes de bases de dades basats en aquesta hipòtesi intenten alliberar els usuaris d'haver d'especificar els camins d'accés tant en el nivell físic com en el lògic. Tota la informació sobre l'estructura lògica de la base de dades (p. ex. el seu esquema conceptual) és oculta per l'usuari. Quan aquest consulta o actualiza aquesta relació imaginària definida sobre U, ho fa només en termes dels atributs, sense cap coneixement de quin atribut apareix ni en quina relació.

A [Sag83] es proposa la *instància representativa* com una representació correcta de les dades emmagatzemades en la base de dades. Per a obtenir aquesta instància només es tenen en compte les dependències funcionals implicades per les claus de les relacions. L'exemple següent il.lustra com es pot obtenir aquesta instància. Vegeu la referència esmentada per a conèixer els detalls d'aquest procediment.

Exemple: Considerem una base de dades que conté les relacions R1 (ABCD), R2 (CGDEF), R3 (DEFB) i R4 (BCF), on els atributs subratllats es corresponen a les claus de cada relació i suposem que el seu contingut actual és R1(1,1,1,2), R2(1,1,1,1,1) i R3(1,1,1,1).

El primer que s'ha de fer per a calcular la instància representativa d'aquesta base de dades és augmentar cadascun dels tuples de les relacions a tuples definits sobre l'univers d'atributs. Això s'aconsegueix usant valors nuls distingibles (subíndex diferent equival a valor nul diferent) per aquells atributs que no apareixen en l'esquema de la relació. En l'exemple obtindríem:

A	B	C	D	E	F	G
1	1	1	2	δ_1	δ_2	δ_3
δ_4	δ_5	1	1	1	1	1
δ_6	1	δ_7	1	1	1	δ_8

Per a obtenir la instància representativa, cal calcular el *chase* d'aquesta relació. Informalment, el 'chase' consisteix a reemplaçar els valors nuls per valors no nuls o per altres valors nuls seguint les restriccions imposades per les dependències funcionals. En la relació anterior, podem reemplaçar el valor nul δ_5 per un 1 mitjançant l'aplicació de la dependència funcional DEF -> B al segon i tercer tuples. Un cop fet això, δ_2 és reemplaçat per un 1 aplicant la dependència funcional BC -> F al primer i segon tuples. No es pot aplicar cap altre dependència funcional, i per tant la instància representativa obtinguda és:

A	B	C	D	E	F	G
1	1	1	2	δ_1	δ_2	δ_3
δ_4	1	1	1	1	1	1
δ_6	1	δ_7	1	1	1	δ_8

La instància representativa satisfà les dependències funcionals incloses en els diferents esquemes de les relacions. Malauradament, no tots els esquemes de bases de dades tenen una única instància representativa. Sagiv imposa una *condició d'unicitat* per a determinar quan es produeix aquest fet. Quan l'usuari consulta o actualitza la base de dades, ho fa especificant un subconjunt dels atributs de la seva instància representativa. És a dir, les consultes i actualitzacions es corresponen a projeccions d'aquesta instància. Per tant, el problema de traduir una actualització especificada sobre la instància representativa en modificacions de les relacions bàsiques corresponents està relacionat amb el problema de l'actualització de vistes.

El concepte d'*extension join* (veure [Sag83]) s'utilitza per a calcular eficientment projeccions de la instància representativa en esquemes de bases de dades independents (on la consistència local implica la consistència global). Aquest concepte desenvolupa un paper molt important en els mètodes que descrivim a continuació.

En aquest context, Langerak [Lan90] proposa algorismes per a traduir actualitzacions de vista en esquemes de bases de dades que satisfan la condició d'unicitat. Es consideren

insercions, esborrats i modificacions d'un únic tuple d'una vista definida mitjançant una projecció total de la instància representativa. Langerak demostra que els algorismes que ell proposa tradueixen l'actualització d'un tuple en la vista, i que l'algorisme és mínim i afecta només les relacions bàsiques subjacents. Una altra constatació important d'aquest treball és la demostració del fet que les restriccions que presenta l'enfocament del complement d'una vista (descrit a la secció 3.1) provoquen que aquest no es pugui aplicar en l'entorn considerat per Langerak.

En una altra proposta [BV88], s'estudia el problema de l'actualització de vistes en aquells esquemes de base de dades que no satisfan la condició d'unicitat. Brosda i Vossen suggereixen que en la fase de disseny de la base de dades s'especifiquin aquelles combinacions d'atributs (*objectes*) que poden participar en una operació d'actualització. Aquests objectes s'inclouen en l'esquema de la base de dades per a ser utilitzats posteriorment. Es permeten dos tipus d'actualització: "Insert Y" i "Delete Y", on Y és un subconjunt dels atributs de la instància representativa.

El procés de traducció es descompon en dues fases diferents: *comprovació* i *execució* . Durant la fase de comprovació, es verifica si Y es correspon a algun dels objectes especificats en la fase de disseny de la base de dades i en peticions d'inserció, caldrà verificar també que existeix la instància representativa de la base de dades actualitzada. Durant l'etapa d'execució, es realitzen les insercions i els esborrats dels tuples corresponents de tal manera que la instància representativa es pugui calcular per via d'extension joins. Els detalls dels procediments de traducció es poden trobar a [BV88].

4. Nous horitzons

En aquesta darrera secció presentem dues propostes que tracten amb bases de dades o vistes no convencionals. Concretament, estudiarem els casos de vistes basades en objectes [BSKW91] i de bases de dades semàntiques [CM89].

Mitjançant la combinació del concepte de vista i del concepte d'objecte, el model *objecte-vista* [BSKW91] admet simultàniament la definició d'unitats abstractes d'informació i la compartició d'aquestes unitats. A més, la informació bàsica està emmagatzemada a la base de dades, fet que facilita la compartició. En aquest entorn, els objectes-vista es defineixen com a vistes d'aquesta base de dades. Per tant, cada objecte-vista és un subconjunt de la base de dades que especifica una nova classe.

Per a poder tractar els objectes-vista, [BSKW91] defineixen el *model estructural* d'una base de dades. Es tracta d'un model semàntic de dades construït a partir de les relacions, que defineix classes d'entitats i les connexions (interrelacions) entre aquestes. Es distingeixen tres tipus de connexió:

- **de propietat:** inclou el concepte de dependència, on els tuples posseïts es relacionen amb un únic tuple propietari. Cardinalitat 1:n. Representació gràfica: -*

- **de referència:** relaciona una entitat amb una altra entitat més abstracta. Cardinalitat n:1. Representació gràfica: ->

- **de subconjunt:** Representació gràfica: O-

L'exemple següent [BSKW91], ens mostra l'esquema estructural d'una base de dades sobre la Universitat (**figura 5**). Un departament universitari es modela mitjançant vuit relacions: Department, Persones, Estudiants, Facultat, Personal, Curriculum, Cursos i Notes. Aquest model estructural indica que els cursos i les persones es relacionen amb un departament, que una persona pot ser un estudiant, una facultat o bé personal d'administració i serveis, que un curriculum descriu els cursos necessaris per a obtenir un títol determinat, i que les notes s'associen entre els estudiants i els cursos.

Per a definir un objecte-vista que representi informació detallada sobre un curs, cal seleccionar una relació*pivot* (en aquest cas Cursos). Aleshores, s'aplica un algorisme per a aïllar l'estructura jeràrquica de l'objecte-vista informació d'un curs, obtenint el resultat expressat en 'negritá en la **figura 5**.

El procediment per a actualitzar un objecte-vista es descomposa en **quatre** etapes:

-validació local respecte la definició de l'objecte-vista on es comprova que l'actualització demanada no violi restriccions estructurals ni autoritzacions de l'usuari.

- propagació dins de l'objecte-vist , fet que permet satisfer les dependències funcionals.

-traducció en un conjunt de modificacions de la base de dades. És a dir, es realitza la transformació de l'actualització de l'objecte-vista demanada en modificacions de la base de dades d'acord amb el traductor escollit.

- validació global envers el model estructural , per a mantenir la consistència global de les dades.

L'algorisme utilitzat en el pas 3 on s'efectua pròpiament la traducció és una extensió del procediment proposat a [Kel85] per a traduir actualitzacions de vista en bases de dades relacionals. En aquest sentit, a [BSKW91] també es proposa que el traductor escollit s'especifiqui completament en temps de definició mitjançant un diàleg estructurat amb l'administrador de la base de dades.

Per a acabar el capítol de nous horitzons, estudiarem una proposta d'actualització de vistes en bases de dades semàntiques definides mitjançant el model GSDM [CM89]. En aquest

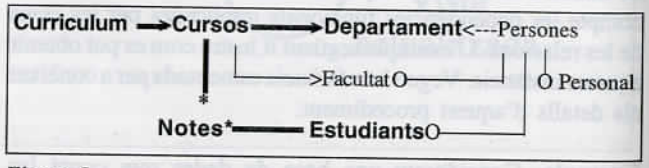

Figura 5

model semàntic, un esquema de base de dades està format per un conjunt de *classes* . L'exemple següent ens mostra l'especificació de la classe Persona mitjançant GSDM:

PERSONA
identifiers: Nom
member attributes:
Nom
value class: STRING
nulls not allowed
not changeable
Edat
value class: INTEGERS
Ingressos_anuals
value class: REALS
class attributes:
Total_persones
value class: INTEGER
derivation: nombre d'ocurrències d'aquesta classe

Cada classe té un nom i una col.lecció d'atributs associats. Hi ha dos tipus d'atributs: *member* i*class* . Els atributs 'member' descriuen les propietats de les instàncies d'una classe; per exemple, cada instància de la classe PERSONA té un nom, una edat i uns ingressos anuals. Els atributs 'class' especifiquen propietats de la classe com a un tot. Per exemple, l'atribut Total_persones ens diu el nombre total d'instàncies de la classe PERSONA. També es poden associar restriccions als atributs, com ara que l'atribut Nom de la classe PERSONA no pot tenir el valor nul.

En GSDM, una subclasse es pot derivar d'una altra classe o subclasse mitjançant diferents operadors. Una subclasse pot ser especificada per l'atribut d'un predicat, especificada explícitament (és a dir, controlada per l'usuari), la intersecció de dues classes, la diferència de dues classes, la unió de dues classes o el conjunt de valors actuals d'un atribut determinat.

Aleshores, es proposen algorismes per a traduir una actualització de vista en modificacions de les classes que estan emmagatzemades físicament a la base de dades. Concretament, es defineix un conjunt de regles d'actualització que s'usaran per a traduir l'actualització de classes derivades. Un exemple d'una d'aquestes regles pot ser *"la inserció en una classe A, definida com la intersecció de dues classes B i C, es tradueix en una inserció sobre B i en una sobre C"* . Chen i McLeod han dissenyat i implementat un prototipus experimental basat en aquest enfocament. La representació de les dades i els algorismes que utilitza, així com una breu descripció de la implementació es poden trobar a [CM89]. Aquest prototipus admet actualitzacions d'inserció i d'esborrat. El traductor rep les peticions dels usuaris i utilitza un algorisme basat en les regles d'actualització per a determinar la propagació de les modificacions.

5. Conclusions

Les vistes permeten delimitar els continguts de la base de dades rellevants per a cada grup d'usuaris. Aquest fet comporta una sèrie d'avantatges com ara afavorir la independència lògica de les dades, simplificar la interfície amb l'usuari i proporcionar una mesura de protecció. Les operacions que es realitzen sobre una vista han de ser traduïdes a operacions equivalents sobre l'estructura de la base de dades.

En un entorn ideal, l'usuari no hauria de distingir entre les relacions bàsiques i les vistes i, per tant, els llenguatges de consulta i de manipulació s'haurien d'aplicar de la mateixa manera en ambdós casos. Malauradament, els sistemes de gestió de bases de dades (SGBD) relacionals que podem trobar en el mercat actualment admeten únicament casos molt restringits d'actualitzacions de vista i, per aquest motiu, no poden assolir completament els avantatges esmentats anteriorment.

Durant els darrers anys, s'ha efectuat molta recerca per a intentar resoldre el problema de l'actualització de vistes tant en bases de dades relacionals com en bases de dades deductives. La integració dels resultats d'aquesta recerca en els SGBD actuals permetria augmentar-ne la seva potència i capacitat. Dissortadament, i malgrat els avenços realitzats hagin estat considerables, encara no es pot considerar completament resolt aquest problema.

Es distingeixen bàsicament dos enfocaments per a resoldre aquest problema: *el dels tipus abstractes de dades* i *el dels traductors automàtics* . Respecte el primer enfocament, convé destacar el fet que pot ser implementat amb els SGBD i els llenguatges de programació actuals. No podem oblidar, però, els inconvenients d'aquest enfocament. Per un costat, el treball necessari per a definir les traduccions i la possibilitat de què aquestes s'hagin de canviar durant la vida de la base de dades. Per l'altre, el fet que l'especificació d'una única traducció pot no ser desitjable en alguns casos.

En canvi, l'enfocament dels traductors automàtics no presenta aquests desavantatges. La seva aportació difereix segons les tres grans línes proposades. La idea del 'complement d'una vistá e s molt elegant, però es massa teòrica. Com els mateixos autors reconeixen, el seu inconvenient principal rau en la seva implementació pràctica. Tampoc podem oblidar la dificultat que comporta definir el complement d'una vista determinada. La 'hipòtesi de la relació universal' és aplicable en un àmbit molt reduït i bastant allunyat dels SGBD actuals. Per tant, tampoc sembla massa interessant en l'actualitat.

En canvi, des del meu punt de vista, els resultats obtinguts en el camp de la 'traducció segons el tipus de vistá s'on directament aplicables en els SGBD actuals. Malgrat aquest fet, caldria generalitzar el tipus de vista i d'actualització permesos i estudiar més profundament com resoldre el problema de l'ambiguïtat semàntica.

A part d'intentar superar aquestes limitacions, la investigació futura hauria d'adreçar aspectes com ara incorporar la satisfacció les restriccions d'integritat en el procés de traducció, fet que permetria obtenir solucions que satisfan

l'actualització demanada i les restriccions de la base de dades, la possibilitat de definir i actualitzar vistes recursives, amb el que s'augmentaria el tipus de vistes que es poden tractar, i l'estudi dels efectes secundaris que es produeixen en aplicar la traducció.

Agraïments

Vull donar les gràcies a l'Antoni Olivé per la seva inestimable ajuda durant la realització d'aquest treball. També vull agrair a D. Costal, E. Mayol, J. A. Pastor, C. Quer, M. R. Sancho, J. Sistac i T. Urpí els seus comentaris i suggeriments en les versions preliminars d'aquest article.

Referències

[Ban79] Bancilhon, F. *"Supporting View Updates in Relational Data Bases"*. In Data Base Architecture (Bracci and Nijssen Eds.). North Holland, Amsterdam, 1979.

[BR86] Bancilhon, F; Ramakrishnan, R. *"An Amateur's Introduction to Recursive Query Processing Strategies"*, Proc. ACM SIGMOD Int. Conf. on the Management of Data, Washington D.C., 1986, pp. 16-152.

[BS81] Bancilhon, F; Spyratos, N. *"Update Semantics of Relational Views"*, ACM Transactions on Database Systems, vol.6, núm.4, 1981, pp.557-575.

[BSKW91] Barsalou, T.; Siambela, N.; Keller, A.M.; Wiederhold, G. *"Updating Relational Databases through Object-Based Views"*, Proc. of the 1991 ACM SIGMOD Int. Conf. on Management of Data. SIGMOD Record, Volume 20, Issue 2, June 1991, pp. 248-257.

[BV88] Brosda, V.; Vossen, G. *"Update and Retrieval in a Relational Database through a Universal Schema Interface"*, ACM Transactions on Database Systems, vol.13, núm.4, 1988, pp.449-485.

[CA79] Carlson, C.R.; Arora, A.K. *"The Updatability of Relational Views Based on Functional Dependencies"*, Proc. 3rd Int. Computer Software and Applications Conf., IEEE Computer Society, Chicago, 1979, pp. 415-420.

[CM89] Chen, I.M.; Mc Leod, D. *"Derived Data Update in Semantic Databases"*, Proc. 15th VLDB Conf, Amsterdam, 1989, pp.225,235.

[CP84] Cosmadakis, S.; Papadimitriou, C. *"Updates of Relational Views"* , Journal of the Association for Computing Machinery, vol.31, núm.4, 1984, pp. 742-760.

[Dat86] Date, C.J. *"Updating views"* , a Relational Databases, Selected Writings, Addison - Wesley, 1986, pp.367-395.

[DB78] Dayal,U.; Bernstein,P.A. *"On the Updatability of Relational Views"* , Proc. 4th VLDB Conf., Berlin, Germany, 1978, pp. 262-278.

[DB82] Dayal, U.; Bernstein, P.A. *"On the correct translation of update operations on relational views"* , ACM Transactions on Database Systems, 8, 1982, pp. 382-416.

[FC85] Furtado, A.L.;Casanova, M.A. *"Updating Relational Views"* , in W.Kim et el. (eds), Query Processing in Database Systems, Springer-Verlag, 1985, pp. 127-142.

[FSS79]Furtado, A.L.; Sevcik, K.C, dos Santos, C.S. *"Permitting Updates through views of Databases"*, Information Systems, 1979.

[Heg90] Hegner, S.J. *"Foundations of Canonical Update Support for Closed Database Views"*. Int. Conf. on Database Theory, Paris, 1990, pp. 422-436.

[Kel85] Keller, A.M. *"Algorithms for Translating View Updates to Database Updates for Views Involving Selection, Projections and Joins "* .Proc. 4th. ACM SIGACT-SIGMOD Symp. on Principle of Database Systems, 1985, pp.154-163.

[Kel86a] Keller, A.M. *"The Role of Semantics in Translating View Updates"*, IEEE Computer, 19:1, January 1986, pp.63-73.

[Kel86b] Keller,A.M. *"Choosing Translator at View Definition Time"*. Proc. 12th VLDB, Kyoto, Japan, 1986, pp. 467-474.

[Kel87] Keller,A.M. *"Comments on Bancilhon and Spyratos: 'Update Semantics of Relational Views"* , Technical Note, ACM Trans. on Database Systems, Vol 12-3, 1987, pp. 521-523.

[KU84] Keller, A.; Ullman, J.D. *"On complementary and independent mappings"*. Proc. ACM SIGMOD Int. Conf. on Management of Data, 1984, Vol. 14-1, pp. 143-149.

[Lan90] Langerak, R. *"View Updates in Relational Databases with an Independent Schema Interface"* , ACM Transactions on Database Systems, vol.15, núm 1,1990, pp. 40-66.

[LS91] Larson, J; Sheth, A. *"Updating Relational Views Using Knowledge at View Definition and View Update Time"*, Information Systems, Vol. 16, nº 2, 1991, pp.145-168.

[MW88] Manchanda, S.; Warren, D.S. *"A logic-based language for database updates"*. In J.Minker (ed) "Foundations of Deductive Databases and Logic Programming". Morgan-Kaufmann Pub. 1988, pp. 363-394.

[Mas84] Masunaga, Y. *"A Relational Database View Update Translation Mechanism"*. Proc. 10th VLDB Conf., Singapore, 1984, pp. 309-320.

[MT85] Medeiros, C.B.; Tompa, F.W. *"Understanding the Implications of View Update Policies"*. Proc. 11th VLDB Conf., Stockholm, 1985, pp. 316-323.

[Sag83] Sagiv, Y. *"A Characterization of Globally Consistent Databases and Their Correct Access Paths"*. ACM Transactions on Database Systems, Vol. 8, núm. 2, 1983, pp. 266-286.

[Sto75] Stonebraker, M. *"Implementation of Integrity Constraints and Views by Query Modification"*. Proc. of the ACM SIGMOD Conference, 1975, pp. 65-78.

[SLW88] Sheth, A.P.; Larson,J.P.; Watkins,E. *"TAILOR, a tool for Updating Views"* . Proc. EDBT88, Venice, 1988. Published in Lecture Notes in Computer Science, nº 303, Springer-Verlag, New York, 1988.

[SM89] Subieta, K.; Missala, M. *"View Updating Through Predefined Procedures"*, Information Systems, vol.14, núm.4, 1989, pp.291-305.

[Spy80] Spyratos, N. *"Translation Structures of Relational Views"* , Proc. 6th VLDB Conf., Montreal, 1980, pp. 411-416.

[TFC83] Tucherman, L.; Furtado, A.L.; Casanova, M.A. *" A Pragmatic Approach to Structured Database Design"*, Proc. 9th VLDB Conf., Firenze, 1983, pp. 219-231.