CY Cergy Paris Université
Master M2 SIC

Intégration de données

Exercices dirigés - corrigé

Pour chacun des cas suivants, donnez la description du médiateur (mapping entre le modèle global et les sources) et montrez comment les requêtes précisées sont traitées par le médiateur.

1. Global as view

Sources de données :

S1: *FilmTitre* (fid, titre)
FilmDetails (fid, année, genre, réalisateur)
Acteur (aid, nom, pays)
Joue (aid, fid)
S2: *FilmInfo* (titre, année, genre)
S3: *Realisateur* (nom, titre)

Modèle global du médiateur GAV:

M: ***Film*** (titre, année, genre, réalisateur)
Acteur (nom, titre)

Requêtes :

a) Les titres et années des films de James Cameron.
b) Les titres des films de James Cameron.
c) Les acteurs du film Avatar.
d) Les acteurs dirigés par James Cameron.

Solution

Mapping: M = V(S1, S2, …)

M: ***Film*** (t, a, g, r) :- **S1**: *FilmTitre* (fid, t) **S1**: *FilmDetails* (fid, a, g, r) U
S2: *FilmInfo* (t, a, g) **S3**: *Realisateur* (r, t)
M: ***Acteur*** (n, t) :- **S1**: *Acteur* (aid, n, p?) **S1**: *Joue* (aid, fid) **S1**: *FilmTitre* (fid, t)

Remarque: les mappings partiels, où pour un élément de M on ne peut retrouver qu'une partie des composantes, sont aussi possibles si on n'a pas de mapping complet pour l'élément. Ces mappings peuvent apporter des réponses aux requêtes qui ne concernent pas les composantes manquantes.

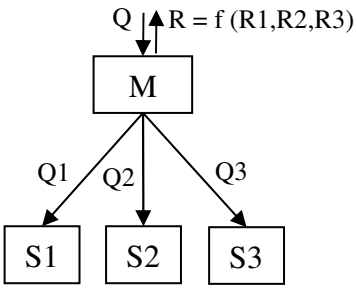
Requêtes

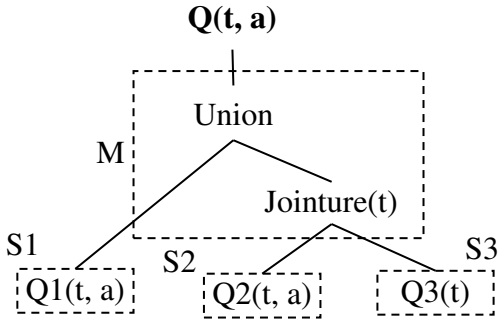
a) Les titres et années des films de James Cameron.
Q(t, a) = M: Film(t, a, g', "James Cameron") =
S1: *FilmTitre* (fid, t) **S1**: *FilmDetails* (fid, a, g', "James Cameron") U
S2: *FilmInfo* (t, a, g') **S3**: *Realisateur* ("James Cameron", t)

Q1(t, a) = **S1**: *FilmTitre* (fid, t) **S1**: *FilmDetails* (fid, a, g', "James Cameron")
Q2(t, a) = **S2**: *FilmInfo* (t, a, g')
Q3(t) = **S3**: *Realisateur* ("James Cameron", t)

En SQL :
Q(t, a) = **select** f.titre, f.année **from** M:Film f **where** f.réalisateur = "James Cameron"
Q1(t, a) = **select** ft.titre, fd.année **from** S1:FilmTitre ft, S1:FilmDetails fd **where** fd.réalisateur = 'James Cameron' and ft.fid=fd.fid
Q2(t, a) = **select** fi.titre, fi.année **from** S2:FilmInfo fi
Q3(t) = **select** r.titre **from** S3:Realisateur r **where** r.nom = 'James Cameron'

Plan d'exécution :





Optimisation : Q2(t, a) retourne tout S2, quand seulement les films de James Cameron sont utiles. Cela peut faire transiter sur le réseau une grande masse de données inutilement.
Amélioration : réaliser la jointure entre S2 et S3 directement dans S2. On récupère d'abord les résultats de Q3(t) sous la forme d'une liste de titres de films LT et on envoie à S2 la requête modifiée suivante (en SQL) :
Q2'(t, a) = **select** fi.titre, fi.année **from** S2:FilmInfo fi **where** fi.titre in (LT)

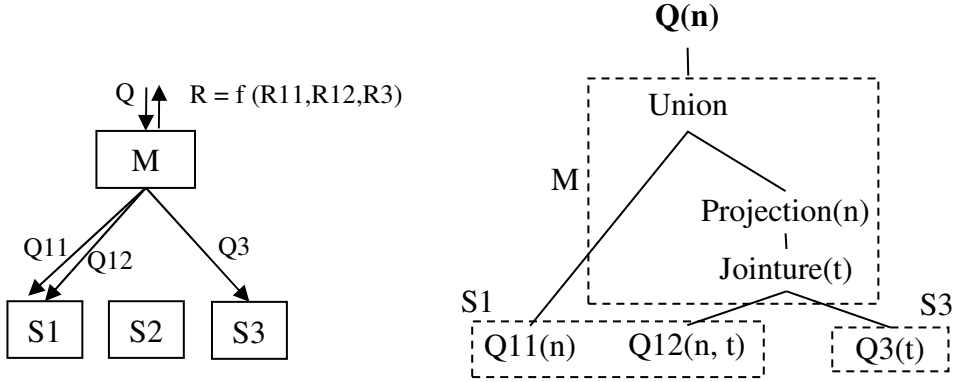
b) Les titres des films de James Cameron.
Q(t) = M: Film(t, a', g', "James Cameron") =
S1: *FilmTitre* (fid, t) **S1**: *FilmDetails* (fid, a', g', "James Cameron") U
~~**S2**: *FilmInfo* (t, a', g')~~ **S3**: *Realisateur* ("James Cameron", t)
S2: *FilmInfo* (t, a', g') est ici inutile
Q(t) = Q1(t) U Q3(t)

c) Les acteurs du film Avatar.
Q(n) = M:Acteur(n, "Avatar") =
S1: *Acteur* (aid, n, p?) **S1**: *Joue* (aid, fid) **S1**: *FilmTitre* (fid, "Avatar")
Q(n) = Q1(n)

d) Les acteurs dirigés par James Cameron.
Q(n) = M:Acteur(n, t) M:Film(t, a', g', "James Cameron") =
S1: *Acteur* (aid, n, p?) **S1**: *Joue* (aid, fid) **S1**: *FilmTitre* (fid, t) **S1**: *FilmTitre* (fid1, t)
S1: *FilmDetails* (fid1, a', g', "James Cameron") U
S1: *Acteur* (aid, n, p?) **S1**: *Joue* (aid, fid) **S1**: *FilmTitre* (fid, t) ~~**S2**: *FilmInfo* (t, a', g')~~
S3: *Realisateur* ("James Cameron", t)

Pour la première combinaison, on veut à travers la jointure sur le titre, avoir le même film. Mais ici, on dispose d'un moyen plus fort de s'en assurer (le titre n'étant pas une vraie clé pour les films) : le fid. Donc on peut mettre fid=fid1, pour avoir le même film.
La première combinaison devient :
S1: *Acteur* (aid, n, p?) **S1**: *Joue* (aid, fid) ~~**S1**: *FilmTitre* (fid, t)~~ **S1**: *FilmDetails* (fid, a', g', "James Cameron")
La présence de FilmTitre devient inutile, car le titre ne sert plus à faire le lien avec Acteur, le fid s'en charge.

Résultat :
Q(n) = M:Acteur(n, t) M:Film(t, a', g', "James Cameron") =
S1: *Acteur* (aid, n, p?) **S1**: *Joue* (aid, fid) **S1**: *FilmDetails* (fid, a', g', "James Cameron") U
S1: *Acteur* (aid, n, p?) **S1**: *Joue* (aid, fid) **S1**: *FilmTitre* (fid, t) **S3**: *Realisateur* ("James Cameron", t)
Q(n) = Q11(n) U Proj(n)(Q12(n, t) Q3(t))



Remarque : le résultat est une union de combinaisons de sources (jointures entre sources).
Question : est-il possible d'avoir certaines de ces combinaisons redondantes, çàd leurs résultats sont déjà fournis par d'autres combinaisons ?
Réponse : sur le principe oui, une situation type est une jointure de type SiSj, comparée avec Si tout seul. La jointure de Si avec Sj apporte une contrainte supplémentaire aux données de Si, donc la source Si toute seule peut en principe fournir toute seule les résultats produits par la jointure SiSj.

Dans notre cas, on a une combinaison avec S1 toute seule et une autre avec S1 jointe à S3. Dans S1 tout seul, les films de James Cameron sont donnés par FilmDetails, dont on a dit qu'elle ne fournit d'information que pour une partie des films. Il est donc possible que certains des films de FilmTitre soient de James Cameron, sans qu'on l'indique dans FilmDetails. Donc la combinaison S1S3 peut apporter de nouveaux résultats dans ce cas de figure, ce qui signifie qu'elle n'est pas redondante.

2. Local as view

Modèle global du médiateur LAV:

M: ***Film*** (titre, année, genre, réalisateur)
Acteur (nom, titre)

Sources de données :

S1: titre, années et réalisateurs de comédies
S2: acteurs avec le titre et année des films où ils ont joué
S3: réalisateurs, titres des films et acteurs qui y jouent
S4: titre et genre de films français d'avant 1970

Requêtes :

a) Les titres et années des films de James Cameron.
b) Les titres des films de James Cameron.
c) Les acteurs du film Avatar.
d) Les acteurs dirigés par James Cameron après 1980.

Solution

Mapping: Si ⊆ Vi(M)

S1(t, a, r) ⊆ M:Film(t, a, "comédie", r)
S2(n, t, a) ⊆ M:Acteur(n, t) M:Film(t, a, g', r')
S3(r, t, n) ⊆ M:Film(t, a', g', r) M:Acteur(n, t)
S4(t, g) ⊆ M:Film(t, a, g, r'), a<1970

Requêtes

a) Les titres et années des films de James Cameron.

Q(t, a) = M: Film(t, a, g', "James Cameron")
S1 est la seule source qui permet de retourner le titre et l'année du film, tout en filtrant par le nom du réalisateur.
Q(t, a) = S1(t, a, "James Cameron") = Q1(t,a)
En SQL : Q(t, a) = **select** S1.titre, S1.annee **from** S1 **where** S1.realisateur = 'James Cameron'

b) Les titres des films de James Cameron.

Q(t) = M: Film(t, a', g', "James Cameron")
S1 et S3 peuvent répondre à la requête.
Q(t) = S1(t, a', "James Cameron") U S3("James Cameron", t, n') = Q1(t) U Q3(t)

c) Les acteurs du film Avatar.

Q(n) = M:Acteur(n, "Avatar") = S2(n, "Avatar", a') U S3(r', "Avatar", n) = Q2(n) U Q3(n)

d) Les acteurs dirigés par James Cameron.

Q(n) = M:Acteur(n, t) M:Film(t, a', g', "James Cameron")
On applique l'algorithme Bucket et on regarde pour chaque part de la requête quelles sources peuvent la fournir :

- M:Acteur(n, t) : S2, S3
- M:Film(t, a', g', "James Cameron") : S1, S3

On génère toutes les combinaisons de sources possibles pour composer la requête :

- S2(n, t, a') S1(t, a'', "James Cameron")
- S2(n, t, a') S3("James Cameron", t, n')
- S3(r', t, n) S1(t, a'', "James Cameron")
- S3(r', t, n) S3("James Cameron", t, n') = S3("James Cameron", t', n)

La source S3 peut fournit des réponses toute seule, il faut voir si S2S3 ou S3S1 ne sont pas redondantes :

- S2(n, t, a') S3("James Cameron", t, n') peut apporter des noms d'acteurs supplémentaires par rapport à ceux de S3 pour les films de James Cameron.
- S3(r', t, n) S1(t, a'', "James Cameron"): S1 apporte des titres de films de James Cameron, mais qui sont déjà dans S3 (jointure sur t), donc cette combinaison est redondante.

Résultat : Q(n) = S3("James Cameron", t', n) U
S2(n, t, a') S1(t, a'', "James Cameron") U
S2(n, t, a') S3("James Cameron", t, n')

Pour effectuer la requête sur S2 une seule fois, on peut réécrire l'union de jointures en jointure d'unions :
Q(n) = Q3(n) U Projection(n) (Q2(n, t) (Q1(t) U Q31(t)))