Module M4
Base de données

Chapitre 5
SQL
Requêtes Imbriquées

Auteurs : Laura Monceaux / Véronique Laime

Effectuer une sous-requête consiste à effectuer une requête à l'intérieur d'une autre, on parle également de requêtes en cascade ou requêtes imbriquées. Une requête imbriquée doit être placée à la suite d'une clause WHERE ou HAVING, et doit remplacer une constante ou un groupe de constantes qui permettraient en temps normal d'exprimer la qualification. Lors de l'execution des requêtes, les requêtes imbriquées sont évaluées en première.

Lorsque l'on utilise une Requête Imbriquée, il faut faire attention :

- au type de résultat retourné
- l'opérateur que l'on peut ainsi utiliser par rapport au type de résultat retourné

en effet nous allons voir que dans la plupart des cas, le résultat de la Requête Imbriquée est utilisé dans une condition de la clause WHERE ou HAVING.

La Requête Imbriquée peut retourner 5 types de résultats :

- une valeur : un tuple d'un attribut

  Si la Requête Imbriquée retourne une valeur, elle ne peut être utilisée que dans des conditions de type

  att op (select ..... from .....)
  où op est une opération arithmétique (=, < >, <, >, <=, >=)

  exemple : si nous reprenons l'exemple de la base de données des étudiants, si on désire connaître les étudiants qui ont eu la même note d'examen que 123456 en M1

```
select NumEtu
from Notes
where codemat = 'M1'
      and noteEX = (    select noteEX
                        from Notes
                        where NumEtu = 123456
                        and codemat = 'M1')
      and NumEtu <> 123456 ;
```

  Remarque : la RI retourne la note d'examen de l'étudiant 123456 en M1 (valeur)

- un ensemble de valeurs : n tuples d'un attribut

  Si la Requête Imbriquée retourne un ensemble de valeurs, elle ne peut être utilisée que dans des conditions de type

  att op' (select ..... from .....)
  où op' = {in | op all | op any}
  in : permet de vérifier qu'une valeur fait partie d'un ensemble de valeurs

op all : permet de vérifier que la valeur d'un attribut est «op» à toutes les valeurs retournées par la RI
op any : permet de vérifier que la valeur d'un attribut est «op» à au moins une valeur des valeurs retournées par la RI

exemple : si on désire les étudiants inscrits en M1 et M2

```
select NumEtu
from Notes
where codemat = 'M1'
        and NumEtu in  (      select NumEtu
                              from Notes
                              where codemat='M2');
```

Remarque : la RI retourne les étudiants inscrits au module M2 (liste)

exemple : si on désire les étudiants inscrits en M1 ayant eu la meilleur note à l'examen

```
select NumEtu
from Notes
where codemat = 'M1'
        and NoteEx >= all  (  select NoteEx
                              from Notes
                              where codemat='M1');
```

Remarque : la RI retourne les notes d'examen du module M1 (liste)

exemple : si on désire connaître les matières ayant la meilleure moyenne d'examen

```
select codemat
from Notes
group by codemat
having AVG((NoteCC+NoteEx)/2) >= all  (  select AVG((NoteCC+NoteEx)/2)
                                         from Notes
                                         group by codemat);
```

Remarque : la RI retourne les moyennes dans chaque module – on ne conserve dans le Requête Principale que les groupes où la moyenne est supérieure ou égale à toutes les autres.

- m valeurs : un tuple de m attributs

Si la Requête Imbriquée retourne m valeurs, elle ne peut être utilisée que dans des conditions de type

(att1,...,attn) op (select att1,...,attn from .....)
où op est une opération arithmétique (=, < >, <, >, <=, >=)

exemple : Si on désire les étudiants de M1 ayant eu les mêmes notes de CC et d'examen que l'étudiant 123456 dans cette matière

```
select NumEtu
from Notes
where codemat = 'M1'
      and noteEX,noteCC = (     select noteEX,noteCC
                                from Notes
                                where NumEtu = 123456
                                and codemat = 'M1')
      and NumEtu <> 123456 ;
```

Remarque : la RI retourne la note d'examen et de CC de l'étudiant 123456 en M1

- n tuples de m attributs

Si la Requête Imbriquée retourne n tuples de m attributs, elle ne peut être utilisée que dans des conditions de type

(att1,...,attn) op' (select att1,...,attn from .....)
où op' = {in | op all | op any}

Parfois, la Requête Imbriquée utilise un attribut d'une table de la requête principale, on appelle cette RI **une requête Corrélée :**

```
select ...
from Table T1 ...
.....          (select ....
                 from Table T2
                 where ..... T1.att = T2.att ....)
```

Pour chaque tuple de la Table T1, on executera la requête corrélée : l'utilisation d'une requête corrélée est très couteuse.

exemple : Si on désire les étudiants inscrits en M1 et M2 ayant eu la même note en CC dans ces 2 matières

```
select NumEtu
from Notes N1
where codemat = 'M1'
      and noteCC  = (select noteCC
                     from Notes N2
                     where N1.NumEtu = N2.NumEtu
                           and codemat = 'M2');
```

Remarque : on doit ici comparer les notes de CC en M1 et M2 du même étudiant. L'utilisation d'une Requête Corrélée est très couteuse, ici pour chaque étudiant du module M1 on cherchera s'il a une note en M2 identique.

- 0 ou n tuples

Si la Requête Imbriquée retourne 0 ou n tuples, elle ne peut être utilisée qu'avec l'opérateur EXISTS ou NOT EXISTS. Cette opérateur nécessite l'utilisation d'une requête corrélée.

```
select .....
from Table T1 .....
where EXISTS (select .... from Table T2 ... where ..... T1.att=T2.att ..);
```

Pour chaque tuple de la requête principale
- évaluation de la RI
- si la RI retourne au moins un tuple, le tuple de la RP est conservé
- si la RI ne retourne aucun tuple, le tuple de la RP n'est pas conservé

```
select .....
from Table T1 .....
where NOT EXISTS (select .... from Table T2 ... where ..... T1.att=T2.att ...);
```

Pour chaque tuple de la requête principale
- évaluation de la RI
- si la RI retourne au moins un tuple, le tuple de la RP n'est pas conservé
- si la RI ne retourne aucun tuple, le tuple de la RP est conservé

exemple : Si l'on désire les étudiants ne suivant pas les cours de M1

```
select NumEtu,NomEtu,PrenEtu
from Etudiant
where  NumEtu not in (select NumEtu from Notes where codemat ='M1'),

select NumEtu, NomEtu,PrenEtu
from Etudiant E
where not exists (select * from Notes N where E.NumEtu = N.NumEtu
                                              and codemat='M1');
```

Parfois, pour répondre à une requête en langage naturel, plusieurs requêtes SQL sont possibles avec ou sans requête imbriquée. On évitera dans ce cas d'utiliser les requêtes corrélées très couteuses. Il est tout à fait possible d'évaluer le coût des requêtes mais cela ne fait pas l'objet de ce cours.