***Date de dernière modification** : 30/6/2015*

Annexe **7**

Le langage SQL-DML (1)

Ce chapitre étudie un type de requête basée sur le quantificateur universel (*pour tout*) et propose un jeu d'exercices le plus souvent accompagnés de leur solution.

A7.1 LE QUANTIFICATEUR UNIVERSEL EN SQL

*Cette section, reprise de l'annexe A24 (**SQL, les ensembles et la logique**), développe le paragraphe 7.7, **c) Condition de totalité (pour tout)**, du chapitre 7 de l'ouvrage.*

A7.1.1 La quantificateur *pour tout* en logique

Le quantificateur universel (∀) est utilisé comme suit :

$\forall a \in A, p(a)$

et se lit : *pour tout élément a de A, le prédicat p où x est remplacé par a, est vérifié*, ou, plus simplement, *tous les éléments de A vérifient la condition (prédicat) p*. Cette expression, si p ne comporte pas d'autres variables que a, a manifestement une valeur de vérité : c'est donc une proposition et non plus un prédicat.

À titre d'exemple, on pourrait exprimer le fait que la colonne DATECOM de la table COMMANDE est obligatoire par la proposition :

$\forall c \in$ COMMANDE, c.DATECOM is not null

L'expression est évidemment moins concise que la déclaration SQL **`not null`** !

Il est possible de transformer un quantificateur dans l'autre :

$\exists x, p(x) \equiv \neg(\forall x, \neg p(x))$

$\forall x, p(x) \equiv \neg(\exists x, \neg p(x))$

La seconde règle est particulièrement importante puisque, s'il existe bien en SQL une forme correspondant au quantificateur existentiel (prédicat exists), il n'existe en revanche rien qui corresponde au quantificateur universel. C'est ce que nous allons étudier dans la section suivante.

A7.1.2 Les quantificateurs en SQL

SQL propose la fonction logique exists (et son inverse not exists), dont l'argument est une expression renvoyant une table, et qui indique si cette table est non vide (true) ou vide (false). Cette fonction permet de simuler le quantificateur existentiel. En effet, la forme

```
select NPRO, LIBELLE
from   PRODUIT PRO
where  exists ( select *
                from   DETAIL DET
                where  NPRO = PRO.NPRO and QCOM > 10);
```

peut être interprétée comme la requête abstraite (le prédicat P encapsule la condition de la seconde clause `where`) :

```
select NPRO, LIBELLE
from   PRODUIT pro
where  ∃det ∈ DETAIL, P(det,pro);
```

SQL ne proposant pas de forme exprimant le quantificateur universel, nous appliquerons l'équivalence définie ci-dessus.

Recherchons par exemple *les commandes qui spécifient tous les produits*. Considérons une (ligne de) COMMANDE M. Celle-ci est sélectionnée si les PRODUITs commandés par M et les PRODUITs forment deux ensembles identiques, c'est-à-dire si le second ensemble est inclus dans le premier[1]. Ou encore, M est retenue si, *pour tout PRODUIT P, P est dans l'ensemble des PRODUITs commandés par M*. Appliquons l'équivalence

$(\forall x, p(x)) \equiv \neg(\exists x, \neg p(x))$

Il vient :

> la COMMANDE M est retenue s'***il n'existe pas*** *de PRODUIT P, tel que P* ***n'est pas*** *dans l'ensemble des PRODUITs commandés par M.*

La traduction mot-à-mot de cette formule en SQL ne pose pas de problèmes insurmontables :

1. Sachant que le premier est forcément inclus dans le second.

```
la COMMANDE M est retenue      →  select NCOM from COMMANDE M
si,                            →  where
  il n'existe pas              →     not exists
    de PRODUIT P,              →       (select * from PRODUIT P
    tel que                    →         where
      P n'est pas dans         →             P.NPRO not in
        l'ensemble des PRODUITs →         (select NPRO from DETAIL
        commandés par M.       →          where NCOM = M.NCOM));
```

En rassemblant ces fragments, on obtient (l'alias P, désormais inutile, pourrait être ignoré) :

```
select NCOM
from   COMMANDE M
where  not exists (select *
                   from   PRODUIT P
                   where  P.NPRO not in (select NPRO
                                         from   DETAIL
                                         where  NCOM = M.NCOM));
```

La sous-requête interne (`select NPRO from DETAIL` ...) extrait les produits référencés par la commande courante M. La sous-requête intermédiaire (`select * from PRODUIT P` ...) définit l'ensemble des produits que M ne référence pas. La requête principale ne retient que les commandes pour lesquelles cet ensemble est vide.

Examinons deux variantes équivalentes. La première se déduit de l'observation que les valeurs de NCOM qui nous intéressent sont toutes présentes dans la table DETAIL. On peut donc réécrire la requête sous la forme :

```
select distinct NCOM
from   DETAIL M
where  not exists (select *
                   from   PRODUIT P
                   where  P.NPRO not in (select NPRO
                                         from   DETAIL
                                         where  NCOM = M.NCOM));
```

La seconde exploite la propriété qui veut que si l'ensemble A est inclus dans B, et que A et B sont de même taille, alors A = B. Considérons les lignes de DETAIL relatives à une commande. L'ensemble des valeurs de NPRO de ces lignes représente les produits commandés. S'il contient autant de valeurs qu'il y a de lignes dans la table PRODUIT, alors cette commande spécifie tous les produits. On peut alors écrire[2] :

```
select NCOM
from   DETAIL
group by NCOM
having count(distinct NPRO) = (select count(*) from PRODUIT);
```

2. Dans cet exemple, le modifieur `distinct` n'est pas nécessaire. Pourquoi ?

Le script **Le quantificateur universel en SQL.sql** reprend ces requêtes après avoir complété temporairement les commandes 30179 et 30185 de sorte qu'elles référencent tous les produits.

A7.2 LES SCRIPTS SQLfast DU CHAPITRE 7

Le script **DML1-requetes.sql** reprend en un unique fichier les requêtes du chapitre 7, adaptées à SQLite 3. La rédaction des scripts relatifs aux autres exercices de cette annexe est laissée au bons soins du lecteur. La bonne exécution de ces scripts réclame l'existence préalable de la base de données CLICOM.db. Si tel n'est pas la cas, l'interpréteur SQLfast le signalera en termes choisis.

A7.3 EXERCICES DU CHAPITRE 7

a) Énoncés de type 1

A7.1 Afficher les caractéristiques des produits (c'est-à-dire, pour chaque produit, afficher ses caractéristiques).

```
select *
from   PRODUIT
```

A7.2 Afficher la liste des localités dans lesquelles il existe au moins un client.

```
select distinct LOCALITE
from   CLIENT
```

A7.3 Afficher le numéro, le nom et la localité des clients de catégorie C1 n'habitant pas à Toulouse.

```
select NCLI, NOM, LOCALITE
from   CLIENT
where  CAT = 'C1'
and    LOCALITE <> 'Toulouse'
```

A7.4 Afficher les caractéristiques des produits en acier.

```
select *
from   PRODUIT
where  LIBELLE like '%ACIER%'
```

A7.5 Donner le numéro, le nom et le compte des clients de Poitiers et de Bruxelles dont le compte est positif.

```
select NCLI, NOM, COMPTE
from   CLIENT
where  LOCALITE in ('Poitiers','Bruxelles')
and    COMPTE > 0
```

b) Énoncés de type 2

A7.6 Quelles catégories de clients trouve-t-on à Toulouse ?

```
select distinct CAT
from   CLIENT
where  LOCALITE = 'Toulouse'
and    CAT is not null
```

A7.7 Afficher le numéro, le nom et la localité des clients dont le nom précède alphabétiquement la localité où ils résident.

```
select NCLI, NOM, LOCALITE
from   CLIENT
where  NOM < LOCALITE
```

A7.8 Afficher le total, le minimum, la moyenne et le maximum des comptes des clients (compte non tenu des commandes actuelles).

```
select sum(COMPTE) as somme,
       avg(COMPTE) as moyenne,
       min(COMPTE) as minimum,
       max(COMPTE) as maximum
from   CLIENT
```

A7.9 Afficher les numéros des clients qui commandent le produit de numéro `'CS464'`.

```
select distinct NCLI
from   COMMANDE
where  NCOM in (select NCOM
                from   DETAIL
                where  NPRO = 'CS464')
```

A7.10 Afficher les localités des clients qui commandent le produit de numéro `'CS464'`.

```
select distinct LOCALITE
from   CLIENT
where  NCLI in (select NCLI
                from   COMMANDE
```

```
                  where  NCOM in (select NCOM
                                  from   DETAIL
                                  where  NPRO = 'CS464'))
```

A7.11 Donner le numéro et le nom des clients de Namur qui n'ont pas passé de commandes.

```
select NCLI, NOM
from   CLIENT
where  NCLI not in (select NCLI from COMMANDE)
and    LOCALITE = 'Namur'
      ou
select NCLI, NOM
from   CLIENT
where  not exists (select * from COMMANDE where NCLI = CLIENT.NCLI)
and    LOCALITE = 'Namur'
```

A7.12 Quels sont les produits en sapin qui font l'objet d'une commande ?

```
select NPRO
from PRODUIT
where LIBELLE like '%SAPIN%'
and exists (select * from DETAIL where NPRO = PRODUIT.NPRO)
      ou
select NPRO
from PRODUIT
where LIBELLE like '%SAPIN%'
and NPRO in (select NPRO from DETAIL)
      ou
select distinct NPRO from PRODUIT P, DETAIL D (! jointure)
where LIBELLE like '%SAPIN%' and P.NPRO = D.NPRO;
```

A7.13 Ecrire les requêtes SQL qui recherchent les clients (on simplifiera si nécessaire) :

- habitant à Lille ou à Namur.

```
select NCLI, LOCALITE
from   CLIENT
where LOCALITE in ('Lille','Namur')
```

- qui à la fois habitent à Lille et n'habitent pas à Namur.

= "qui habitent à Lille".

- qui habitent à Lille ou n'habitent pas à Namur.

= "qui n'habitent pas à Namur"

- qui n'habitent ni à Lille ni à Namur.

```
select NCLI, LOCALITE
from   CLIENT
where  LOCALITE not in ('Lille','Namur')
```

- qui n'habitent pas à Lille ou qui n'habitent pas à Namur.
 = "tous les clients"
- de catégorie C1 habitant à Namur.

```
select NCLI, LOCALITE, CAT
from   CLIENT
where  CAT = 'C1'
and    LOCALITE = 'Namur'
```

- de catégorie C1 ou habitant à Namur.

```
select NCLI, LOCALITE, CAT
from   CLIENT
where  CAT = 'C1'
or     LOCALITE = 'Namur'
```

- de catégorie C1 n'habitant pas à Namur.

```
select NCLI, LOCALITE, CAT
from   CLIENT
where  CAT = 'C1'
and    LOCALITE <> 'Namur'
```

- qui n'ont pas été sélectionnés dans la question précédente.

```
select NCLI, LOCALITE, CAT
from   CLIENT
where  CAT <> 'C1'
or     CAT is null
or     LOCALITE = 'Namur'
```

- qui soit sont de catégorie B1 ou C1, soit habitent à Lille ou à Namur (ou les deux conditions).

```
select NCLI, LOCALITE, CAT
from   CLIENT
where  CAT in ('B1','C1')
or     LOCALITE in ('Lille','Namur')
```

- qui soit sont de catégorie B1 ou C1, soit habitent à Lille ou à Namur (mais pas les deux conditions).

```
select NCLI, LOCALITE, CAT
from   CLIENT
where  (CAT in ('B1','C1') and LOCALITE not in ('Lille','Namur'))
or     (CAT not in ('B1','C1') and LOCALITE in ('Lille','Namur'))
```

- qui sont de catégorie B1 ou C1, et qui habitent à Lille ou à Namur.

```
select NCLI, LOCALITE, CAT
from   CLIENT
where  CAT in ('B1','C1')
and    LOCALITE in ('Lille','Namur')
```

- qui n'ont pas été sélectionnés dans la question précédente.

```
select NCLI, LOCALITE, CAT
from   CLIENT
where  CAT not in ('B1','C1')
or     LOCALITE not in ('Lille','Namur')
```

c) Énoncés de type 3

A7.14 Afficher la valeur totale des stocks (compte non tenu des commandes actuelles).

```
select sum(QSTOCK*PRIX) as TOTAL
from PRODUIT
```

A7.15 Afficher le numéro et le libellé des produits en sapin :

- qui ne sont pas commandés,
- qui sont commandés à Toulouse,
- qui ne sont pas commandés à Toulouse
- qui ne sont commandés qu'à Toulouse,
- qui ne sont pas commandés qu'à Toulouse,
- qui sont commandés à Toulouse, mais aussi ailleurs.

A7.16 Combien y a-t-il de commandes spécifiant un (ou plusieurs) produit(s) en acier ? *(! jointures)*

```
select count(*)
from COMMANDE
where  NCOM in (select NCOM
                from DETAIL
                where  NPRO in (select NPRO
                                from PRODUIT
                                where LIBELLE like '%ACIER%'));
```

ou

```
select count(*)  (! jointures)
from   COMMANDE M
where  NCOM in (select NCOM
                from DETAIL D, PRODUIT P
                where  D.NPRO = P.NPRO
                and    LIBELLE like '%ACIER%')
```

ou

```
select count(distinct M.NCOM)  (! jointures)
from   COMMANDE M, DETAIL D, PRODUIT P
```

```
where  M.NCOM = D.NCOM
and    D.NPRO = P.NPRO
and    LIBELLE like '%ACIER%'
      ou
```

A7.17 Dans combien de localités trouve-t-on des clients de catégorie C1 ?

```
select count(distinct LOCALITE)
from   CLIENT
where  CAT = 'C1'
```

d) Énoncés de type 4

A7.18 Exprimer de trois manières différentes la requête : *quels sont les produits qui ne sont pas commandés ?*

A7.19 Afficher le numéro et le nom des clients qui n'ont pas commandé de produits en sapin.

```
select NCLI, NOM
from CLIENT
where NCLI not in (select NCLI from COMMANDE
                   where NCOM in (select NCOM from DETAIL
                                  where NPRO in (select NPRO
                                                 from PRODUIT
                                       where LIBELLE like '%SAPIN%')));
```

A7.20 A la question : "*rechercher les localités dans lesquelles on n'a pas commandé de produit PA60*", quatre utilisateurs proposent les requêtes suivantes. Indiquer la (ou les) requêtes correctes, et interprétez les autres.

```
  select distinct LOCALITE
  from   CLIENT
  where  NCLI in
              (select NCLI
               from   COMMANDE
               where  NCOM in
                          (select NCOM
                           from   DETAIL
                           where  NPRO <> 'PA60'))
```

```
LOCALITE
====================
Lille
Poitiers
Toulouse
```

```
  select distinct LOCALITE
  from   CLIENT
```

```
  where  NCLI in
              (select NCLI
               from   COMMANDE
               where  NCOM not in
                          (select NCOM
                           from   DETAIL
                           where  NPRO = 'PA60'))
```

```
LOCALITE
====================
Lille
Poitiers
```

```
  select distinct LOCALITE
  from   CLIENT
  where  NCLI not in
              (select NCLI
               from   COMMANDE
               where  NCOM in
                          (select NCOM
                           from   DETAIL
                           where  NPRO = 'PA60'))
```

```
LOCALITE
====================
Bruxelles
Geneve
Lille
Namur
Paris
Poitiers
Toulouse
```

```
  select distinct LOCALITE
  from   CLIENT
  where  LOCALITE not in
              (select LOCALITE
               from   CLIENT
               where  NCLI in
                       (select NCLI
                        from   COMMANDE
                        where  NCOM in
                               (select NCOM
                                from   DETAIL
                                where  NPRO = 'PA60')))
```

seule expression correcte

```
LOCALITE
====================
Bruxelles
Geneve
Lille
Paris
```

A7.21 Que signifie la requête suivante ?

```
select *
from   COMMANDE
where  NCOM not in (select NCOM
                    from   DETAIL
                    where  NPRO <> 'PA60')
```

A7.22 Dans quelles localités a-t-on commandé en décembre 2008 ?

```
select distinct LOCALITE
from   CLIENT
where  NCLI in (select NCLI
                from   COMMANDE
                where  DATECOM like '%DEC-2008')
```

A7.23 On suppose qu'on n'a pas trouvé utile de déclarer NCOM clé étrangère dans la table DETAIL. Il est donc possible que certaines lignes de DETAIL violent la contrainte d'intégrité référentielle portant sur cette colonne. Ecrire une requête qui recherche les anomalies éventuelles.

```
select NCOM, NPRO
from   DETAIL D
where  not exists (select * from COMMANDE where NCOM = D.NCOM);
```

A7.24 Normalement, à toute commande doit être associé au moins un détail. Écrire une requête qui vérifie qu'il en bien ainsi dans la base de données.

```
select NCOM
from   COMMANDE M
where  not exists (select * from DETAIL where NCOM = M.NCOM);
```

A7.25 Quels sont les produits (numéro et libellé) qui n'ont pas été commandés en 2008 ?

```
select NPRO, LIBELLE
from   PRODUIT
where  NPRO not in (select NPRO
                    from   DETAIL D
                    where  NCOM in (select NCOM
                                    from   COMMANDE M
                                    where  DATECOM like '%2008%');
```

ou

```
select NPRO, LIBELLE    (! jointure)
from   PRODUIT
where  NPRO not in (select NPRO
                    from   DETAIL D, COMMANDE M
                    where  D.NCOM = M.NCOM
```

```
                    and    M.DATECOM like '%2008%');
```

e) Énoncés de type 5

A7.26 Rechercher les clients qui ont commandé tous les produits.

Suggestion. Application du quantificateur *pour tout*. On recherche les clients tels qu'il n'existe pas de produits qui n'apparaissent pas dans les détails de leurs commandes.

```
select NCLI
from CLIENT C
where not exists (select *
                  from PRODUIT
                  where NPRO not in
                              (select NPRO
                               from DETAIL
                               where  NCOM in (select NCOM
                                               from COMMANDE
                                               where NCLI = C.NCLI)));

select NCLI    (! jointure)
from   CLIENT C
where not exists (select *
                  from PRODUIT
                  where NPRO not in (select NPRO
                                     from   DETAIL D, COMMANDE M
                                     where  D.NCOM = M.NCOM
                                     and    M.NCLI = C.NCLI))
```

A7.27 Dans quelles localités a-t-on commandé tous les produits en acier (tous clients confondus) ?

Exercice préparatoire : *quels sont les clients qui ont commandé tous les produits en acier*.

```
select NCLI
from   CLIENT C
where not exists (select *
                  from   PRODUIT
                  where  LIBELLE like '%ACIER%'
                  and    NPRO not in (select NPRO
                                      from   DETAIL D, COMMANDE M
                                      where  D.NCOM = M.NCOM
                                      and    M.NCLI = C.NCLI))
```

Question d'origine :

```
les localités L telles
     qu'il n'existe pas
           de produits en acier qui ne soit commandé
                          par un client de la localité L
```

```
select distinct LOCALITE
from   CLIENT C
where not exists (select *
                  from   PRODUIT
                  where  LIBELLE like '%ACIER%'
                  and    NPRO not in
                            (select NPRO
                             from   DETAIL D, COMMANDE M, CLIENT CC
                             where  D.NCOM = M.NCOM
                             and    M.NCLI = C.NCLI
                             and    CC.LOCALITE = C.LOCALITE))
```

A7.28 Rechercher les produits qui ont été commandés par tous les clients.

```
select NPRO
from   PRODUIT P
where  not exists (select *
                   from   CLIENT
                   where  NCLI not in (select NCLI
                                       from   COMMANDE M, DETAIL D
                                       where  M.NCOM = D.NCOM
                                       and    D.NPRO = P.NPRO))
```

A7.29 Rechercher les localités dont aucun client n'a passé de commande.

```
select distinct LOCALITE
from   CLIENT
where  LOCALITE not in (select LOCALITE
                        from   CLIENT C
                        where exists (select *
                                      from   COMMANDE
                                      where  NCLI = C.NCLI))
```

A7.30 Rechercher les localités dont tous les clients ont passé au moins une commande.

```
select distinct LOCALITE
from   CLIENT C
where  not exists (select * from CLIENT
                   where  LOCALITE = C.LOCALITE
                   and    NCLI not in (select NCLI from COMMANDE))
```

A7.31 Rechercher les produits qui sont commandés dans toutes les localités.

```
select NPRO
from   PRODUIT P
where not exists
           (select *
            from   CLIENT
```

```
where  LOCALITE not in
          (select LOCALITE
           from   CLIENT C, COMMANDE M, DETAIL D
           where  M.NCLI = C.NCLI
           and    D.NCOM = M.NCOM
           and    D.NPRO = P.NPRO))
```