Université Nice Sophia Antipolis

4. LE LANGAGE SQL DML

Hend AFFES
Jean-Yves Tigli

BAT3 – Polytech Nice – Sophia

2013-2014

Université Nice Sophia Antipolis

2. LE LANGAGE SQL DML

Contenu

2.1 Introduction

- Le sous-langage DML de SQL permet de consulter le contenu des tables et de les modifier. Il comporte 4 verbes.
- La requête select extrait des données des tables
- La requête insert insère de nouvelles lignes dans une table
- La requête delete supprime des lignes d'une table
- La requête update modifie les valeurs de colonnes de lignes existantes

2.1 Introduction

- SQL DML est la partie la plus spectaculaire du langage SQL.
- Il comporte deux grande classes de fonctions: l'extraction de données et la modification de données.
- L'extraction fait l'objet d'une seule commande : la requête select.
- SELECT: nous permet d'extraire d'une base de données les données répondant à des questions que nous pourrions nous poser.

Université Nice Sophia Antipolis

D’une manière générale, une requête simple contient trois parties principales:

- La clause select précise les valeurs (nom des colonnes, valeurs dérivées) qui constituent chaque ligne du résultat.
- La clause from indique la table ou les tables desquelles le résultats tire ses valeurs.
- La clause where spécifie la condition de sélection que doivent satisfaire les lignes qui fournissent le résultat.

-+ Toute requête select renvoie un résultat sous la forme d'une table

2.2 Extraction simple

Affichage de valeurs de certaines colonnes des lignes d'une table

```
select NCLI, NOM, LOCALITE
from   CLIENT;
```

NCLI	NOM	LOCALITE
B062	GOFFIN	Namur
B112	HANSENNE	Poitiers
B332	MONTI	Genève
B512	GILLET	Toulouse
C003	AVRON	Toulouse
C123	MERCIER	Namur
C400	FERARD	Poitiers
D063	MERCIER	Toulouse
F010	TOUSSAINT	Poitiers
F011	PONCELET	Toulouse
F400	JACOB	Bruxelles
K111	VANBIST	Lille
K729	NEUMAN	Toulouse
L422	FRANCK	Namur
S127	VANDERKA	Namur
S712	GUILLAUME	Paris

Affichage des valeurs de toutes les colonnes d'une table

```
select *
from   CLIENT;
```

*** = liste des colonnes**

2.2 Extraction simple

Affichage de NCLI et NOM des lignes de la table CLIENT relatives aux clients de Toulouse.

```
select NCLI, NOM
from   CLIENT
where  LOCALITE = 'Toulouse';
```

NCLI	NOM
B512	GILLET
C003	AVRON
D063	MERCIER
F011	PONCELET
K729	NEUMAN

D'autres comparateurs:

- Égal à : =
- Plus grand que: >
- Plus petit que : <
- Différent de: <>
- Plus grand ou égal: >=
- Plus petit ou égal: <=

- valeurs numériques et temporelles
- les chaînes de caractères ch1<ch2: ch1 est avant ch2 selon l'ordre lexicographique (exemple: a<b)

2.2 Extraction simple

Afficher les localités où habitent des clients de catégorie C1.

```
select LOCALITE
from   CLIENT
where  CAT = 'C1';
```

LOCALITE
Poitiers
Namur
Poitiers
Namur
Namur

lignes en double

Lignes dupliquées dans le résultat: clause distinct

```
select distinct LOCALITE
from   CLIENT
where  CAT = 'C1';
```

LOCALITE
Namur
Poitiers

Université Nice Sophia Antipolis

2.3 Conditions plus complexes

2.3 Conditions plus complexes - les valeurs *null*

Une condition élémentaire peut porter sur la présence de la valeur null

```
select NCLI
from   CLIENT
where  CAT is null;
```

NCLI
D063
K729

```
select NCLI
from   CLIENT
where  CAT is not null;
```

2.3 Conditions plus complexes - *in* et *between*

Une condition peut porter sur l'appartenance à une liste

```
select NCLI
from   CLIENT
where  CAT in ('C1','C2','C3');
```

```
select NCLI
from   CLIENT
where  LOCALITE not in ('Toulouse','Breda');
```

ou à un intervalle

```
select NCLI
from   CLIENT
where  COMPTE between 1000 and 4000;
```

2.3 Conditions plus complexes - Les masques

Une condition peut porter sur la présence de certains caractères dans une valeur

```
select NCLI
from   CLIENT
where  CAT like 'B_';
```

'_' = un caractère quelconque

masques

```
select NPRO
from   PRODUIT
where  LIBELLE like '%SAPIN%';
```

'%' = une chaîne quelconque

Un masque définit une famille de chaînes de caractères :

'B_'	→	'B1' 'Bd' 'B '
'B_'	↛	'xB' 'B' 'B12'

'%SAPIN%'	→	'PL. SAPIN 200x20x2' 'Boite en SAPIN' 'SAPIN VERNI'
'%SAPIN%'	↛	'Boite en Sapin' 'Achetez S A P I N !'

2.3 Conditions plus complexes - Combinaisons logiques

La condition de sélection introduite par la clause where peut être constituée d'une expression booléenne de conditions élémentaires (opérateurs and, or, not et parenthèses).

Afficher les lignes pour lesquelles, simultanément LOCALITE a pour valeur 'Toulouse' et COMPTE a une valeur négative.

```
select NOM, ADRESSE, COMPTE
from   CLIENT
where  LOCALITE = 'Toulouse' and COMPTE < 0;
```

```
select NOM, ADRESSE, COMPTE
where  COMPTE > 0
and    (CAT = 'C1' or LOCALITE = 'Paris')
```

Etant donné les conditions P et Q relatives aux lignes de la table T, la clause:

Where P and Q : sélectionne les lignes de T qui vérifient simultanément P et Q

Where P or Q : sélectionne les lignes de T qui vérifient P ou Q ou les deux

Where not P : sélectionne les lignes de T qui ne vérifient pas P

Université Nice Sophia Antipolis

2.4 Données extraites et données dérivées

2.4 Données extraites et données dérivées - expressions de calcul

```
select 'TVA de ', NPRO, ' = ',0.21*PRIX*QSTOCK
from   PRODUIT
where  QSTOCK > 500;
```

TVA de	NPRO	=	0,21*PRIX*QSTOCK
TVA de	CS264	=	67788
TVA de	PA45	=	12789
TVA de	PH222	=	37770.6
TVA de	PS222	=	47397

```
select NPRO as Produit, 0.21*PRIX*QSTOCK as Valeur_TVA
from   PRODUIT
where  QSTOCK > 500;
```

'Produit' est un alias de colonne

Produit	Valeur_TVA
CS264	67788
PA45	12789
PH222	37770.6
PS222	47397

Les fonctions SQL

Outre les 4 opérations arithmétiques (+,-,*,/,…), SQL offre une large gamme de fonctions

Fonctions de chaîne de caractères: ces fonctions renvoient soit une chaîne, soit un nombre entier

• char_length(ch): donne le nombre de caractère de la chaîne ch.

•Position(ch1 in ch2): donne la position de chaîne ch1 dans la chaîne ch2; 1 si ch1 est vide et 0 si ch1 n'apparaît pas dans ch2.

•Ch1||ch2: construit une chaîne composée de la concaténation (mise bout à bout) des chaines ch1 et ch2.

•Lower(ch): construit une chaîne formée des caractères de ch transformés en miniscules.

•substring(ch from I for L): construit une chaîne formée des L caractères de la chaîne ch partant de la position I.

Université Nice Sophia Antipolis

2.5 Les fonctions agrégatives (statistiques)

2.5 Les fonctions agrégatives (statistiques)

Il existe également des fonctions prédéfinies qui donnent une valeur agrégée calculée pour les lignes sélectionnées

- count(*) : donne le nombre de lignes trouvées
- count(nom-colonne) :donne le nombre de valeurs de la colonne
- avg(nom-colonne) :donne la moyenne des valeurs de la colonne
- min(nom-colonne) :donne le minimum des valeurs de la colonne
- max(nom-colonne) :donne le maximum des valeurs de la colonne

2.5 Les fonctions agrégatives (statistiques)

```
select 'Namur',avg(COMPTE) as Moyenne,
       max(COMPTE)-min(COMPTE) as Ecart_max,
       count(*) as Nombre
from   CLIENT
where  LOCALITE = 'Namur';
```

Namur	Moyenne	Ecart_max	Nombre
Namur	-2520	4580	4

le résultat ne comprend qu'une seule ligne

```
select sum(QSTOCK*PRIX)
from   PRODUIT
where  LIBELLE like '%SAPIN%';
```

2.5 Les fonctions agrégatives (statistiques)

Attention aux valeurs dupliquées

```
select count(NCLI)
from   COMMANDE;
```

count(NCLI)
7

```
select distinct count(NCLI)
from   COMMANDE;
```

count(NCLI)
7

```
select count(distinct NCLI)
from   COMMANDE;
```

count(NCLI)
5

2.5 Les fonctions agrégatives (statistiques)

```
select count(NCLI) as Nombre,
       count(NOM) as Noms,
       count(LOCALITE) as Localités,
       count(CAT) as Catégories
from   CLIENT;
```

Nombre	Noms	Localités	Catégories
16	16	16	14

```
select count(distinct NCLI) as Nombre,
       count(distinct NOM) as Noms,
       count(distinct LOCALITE) as Localités,
       count(distinct CAT) as Catégories
from   CLIENT;
```

Nombre	Noms	Localités	Catégories
16	15	7	4

Université Nice Sophia Antipolis

2.6 Les sous-requêtes

2.6 Les sous-requêtes - Principe

Les numéros des clients de Namur :

```
select NCLI
from   CLIENT
where  LOCALITE = 'Namur';
```

NCLI
B062
C123
L422
S127

2.6 Les sous-requêtes - Principe

Les numéros des clients de Namur :

NCLI
B062
C123
L422
S127

Les numéros des commandes des clients de Namur :

```
select NCOM, DATECOM
from   COMMANDE
where  NCLI in ('C123','S127','B062','L422');
```

ne marche qu'une fois

mieux :

```
select NCOM, DATECOM
from   COMMANDE
where  NCLI in (select NCLI
                from   CLIENT
                where  LOCALITE = 'Namur');
```

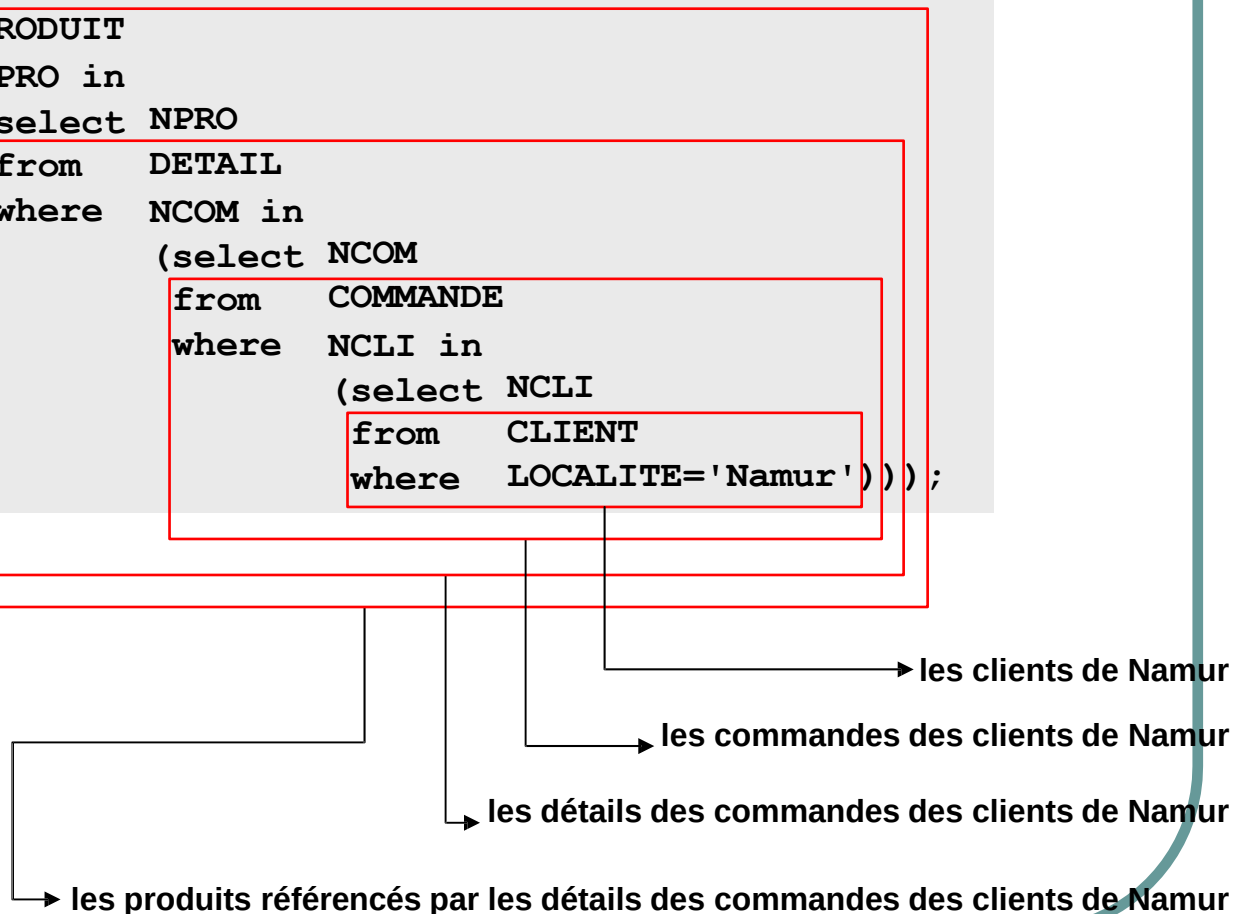
marchera toujours

2.6 Les sous-requêtes - Principe

```
select *
from   PRODUIT
where  NPRO in
       (select NPRO
        from   DETAIL
        where  NCOM in
               (select NCOM
               from    COMMANDE
               where   NCLI in
                       (select NCLI
                        from   CLIENT
                        where  LOCALITE='Namur')));
```

2.6 Les sous-requêtes - Principe

```
select *
from   PRODUIT
where  NPRO in
       (select NPRO
        from   DETAIL
        where  NCOM in
               (select NCOM
                from   COMMANDE
                where  NCLI in
                       (select NCLI
                        from   CLIENT
                        where  LOCALITE='Namur')));
```



2.6 Les sous-requêtes - Condition d'association

Une condition ***in (sous-requête)*** **correspond le plus souvent à une condition d'association = *qui sont associés à ...***

```
select *
from   T
where  CT in (select CS
              from   S
              where  <condition>);
```

"on recherche les T qui sont associés à des S qui ..."

```
select *
from   COMMANDE
where  NCLI in (select NCLI
                from   CLIENT
                where  LOCALITE = 'Namur');
```

```
select *
from   CLIENT
where  NCLI in (select NCLI
                from   COMMANDE
                where  DATECOM = '12-09-2009');
```

2.6 Les sous-requêtes - Références multiples

Condition d'association quantifiée :
***recherche des commandes** d'au moins 3 détails*

```
select NCOM, DATECOM, NCLI
from   COMMANDE C
where  (select count(*)
        from DETAIL
        where NCOM = C.NCOM) >= 3;
```

Université Nice Sophia Antipolis

2.7 Les quantificateurs ensemblistes

2.7 Les quantificateurs ensemblistes - exists, not exists

exists et not exists

le prédicat exists(E), où E est une sous-requête,
est *vrai* si l'ensemble désigné par E est *non vide*

quels sont les produits pour lesquels il existe au moins un détail ?

```
select NPRO, LIBELLE
from   PRODUIT as P
where  exists (select *
               from   DETAIL
               where  NPRO = P.NPRO);
```

le prédicat not exists(E),
est vrai si l'ensemble désigné par E est vide

2.7 Les quantificateurs ensemblistes - all, any

all et any

quels sont les commandes qui spécifient la plus petite quantité de PA60 ?

```
select distinct NCOM
from   DETAIL
where  QCOM <= all (select QCOM
                    from   DETAIL
                    where  NPRO = 'PA60')
and    NPRO = 'PA60';
```

```
select distinct NCOM
from   DETAIL
where  QCOM = (select min(QCOM)
               from   DETAIL
               where  NPRO = 'PA60')
and    NPRO = 'PA60';
```

variante

2.7 Les quantificateurs ensemblistes - all, any

all et any

quels sont les commandes qui ne spécifient pas la plus petite quantité de PA60 ?

```
select *
from   DETAIL
where  QCOM > any (select QCOM
                   from   DETAIL
                   where  NPRO = 'PA60')
and    NPRO = 'PA60';
```

```
select distinct NCOM
from   DETAIL
where  QCOM > (select min(QCOM)
               from   DETAIL
               where  NPRO = 'PA60')
and    NPRO = 'PA60';
```

Variante