

Requêtes SELECT avec les opérateurs d'ensembles

L'opérateur INTERSECT : cet opérateur permet de ramener l'intersection des données entre deux tables. Les deux commandes SELECT doivent avoir le même nombre d'attributs, et des attributs de même type et dans le même ordre.

Syntaxe:

```
Instruction SELECT1  
INTERSECT  
Instruction SELECT 2  
[ORDER BY].
```

- Le nombre de colonnes renvoyées par SELECT 1 doit être le même que celui renvoyé par SELECT 2
- Le type de données SELECT 1 doit être le même que celui de SELECT 2
- La clause optionnelle ORDER BY doit se faire selon un numéro de colonne et non selon le nom.
- SELECT 1 et SELECT 2 ne peuvent contenir des clauses ORDER BY.

Exemple

```
SELECT NOM, PRENOM FROM ETUDIANTS  
INTERSECT  
SELECT NOM, PRENOM FROM ENSEIGNANTS  
ORDER BY 1
```

Permet de ramener tous les étudiants qui sont en même temps des enseignants.

L'opérateur UNION

Cet opérateur renvoi l'ensemble des lignes des deux tables. Si des lignes sont redondantes elles sont renvoyées une seule fois. Pour renvoyer toutes les lignes, utiliser l'option ALL

Les mêmes contraintes qui s'appliquent pour INTERSECT s'appliquent pour UNION

Syntaxe

```
Instruction SELECT1  
UNION [ALL]  
Instruction SELECT 2  
[ORDER BY].
```

```
SELECT NOM, PRENOM FROM ETUDIANTS  
UNION  
SELECT NOM, PRENOM FROM ENSEIGNANTS  
ORDER BY 1
```

Permet de ramener tous les étudiants et tous les enseignants. Les enseignants qui sont en même temps des étudiants sont ramenés une seule fois.

L'opérateur MINUS

Cet opérateur renvoi l'ensemble des lignes de la première table MOINS les lignes de la deuxième table.

Les mêmes contraintes qui s'appliquent pour INTERSECT s'appliquent pour MINUS

Syntaxe

```
Instruction SELECT1  
MINUS  
Instruction SELECT 2  
[ORDER BY].
```

Création de séquences pour insertion

```
CREATE SEQUENCE <Nom_de_séquence> INCREMENT BY  
    <intervalle>  
    START WITH <valeur_de_depart>  
    MAXVALUE <valeur_maximale>  
    MINVALUE <valeur_minimale>  
    CYCLE
```

Exemple 1

```
CREATE SEQUENCE seq1  
START WITH 10  
MAXVALUE 100  
INCREMENT BY 10;
```

Et on utilise le INSERT comme suit:

```
INSERT INTO EmployesInfo (numemp, nom) VALUES  
(seq1.nextval, 'Simpson');
```

L'employé (10, Simpson) est inséré (si c'est la première fois qu'on utilise la séquence seq1)

```
INSERT INTO EmployesInfo (numemp, nom) VALUES  
(seq1.nextval, 'Blues');
```

L'employé (20, Blues) est inséré (si c'est la deuxième fois qu'on utilise la séquence seq1)

Exemple 2

```
CREATE SEQUENCE seq2  
INCREMENT BY 2  
START WITH 20;
```

START WITH n: n indique la valeur de départ de la séquence. Dans une séquence croissante, la valeur par défaut est la valeur minimale, et dans une séquence décroissante la valeur par défaut est la valeur maximale.

INCREMENT BY n: n est le PAS. Pour préciser l'intervalle entre les nombres générés. Par défaut cette valeur est 1. Le nombre n peut être positif pour générer une séquence croissante ou négatif pour générer une séquence décroissante.

MAXVALUE n: n indique la valeur maximale de la séquence. Par défaut $n=10^{**}27$ pour une séquence négative et $n=-1$ pour une séquence négative.

MINVALUE n : n indique la valeur maximale de la séquence. Par défaut $n=-10^{**}27$ pour une séquence décroissante et $n=1$ pour une séquence croissante.

CYCLE : indique que la séquence continue à générer des valeurs à partir de la valeur minimale une fois atteinte la valeur maximale pour une séquence croissante et contrairement pour une séquence décroissante.

NEXTVAL : pour augmenter la séquence du PAS et obtenir une valeur.

CURRVAL : pour obtenir la valeur courante de la séquence.

ATTENTION !!!:

- ✓ Ne jamais utiliser une séquence avec CYCLE pour générer des valeurs de clé primaire
- ✓ Lorsqu'un enregistrement est supprimé de la table, le numéro de séquence correspondant n'est pas récupéré.
- ✓ Lors de l'insertion d'un enregistrement, s'il y a violation d'une contrainte d'intégrité (l'enregistrement n'a pas été inséré) le numéro de séquence est perdu.