Gestion de l'information hiérarchisée

Des données hiérarchiques sont des données stockées dans une table avec une relation récursive (relation sur la même table ou entité) dans une cardinalité maximale est 1. La table contient au moins deux colonnes : une colonne de clé primaire et une colonne définissant la clé étrangère (clé enfant) et qui réfère à la clé primaire (clé parent). Un exemple de la table EMPLOYES dans laquelle on souhaite mettre en évidence la hiérarchisation entre les employés (employés avec les responsables hiérarchiques).

Exemple : Dans la table EMPLOYES, on voit bien les employés avec leur responsable direct. Comme par exemple DUBOIS a son responsable qui est ROY , et ROY a son responsable Cristophe ….

Pour mettre en évidence la hiérarchie dans une table lors d'une sélection, la commande SELECT est accompagnée de deux nouvelles clauses. CONNECT BY et START WITH

NUM	NOM	NUMRES
17	DUBOIS	16
10	alibaba	(null)
12	Martin	10
11	Patoche	10
13	Cristophe	10
14	FAFAR	11
16	ROY	13

```
SELECT NUM, NOM,NUMRES,LEVEL
 FROM EMPLOYES
START WITH NUM =10
 CONNECT BY PRIOR NUM = NUMRES;
```

Cette requête nous donne tous les employés sous l'employé dont le numéro est 10.

NUM	NOM	NUMRES	LEVEL
10	alibaba	(null)	1
11	Patoche	10	2
14	FAFAR	11	3
12	Martin	10	2
13	Cristophe	10	2
16	ROY	13	3
17	DUBOIS	16	4

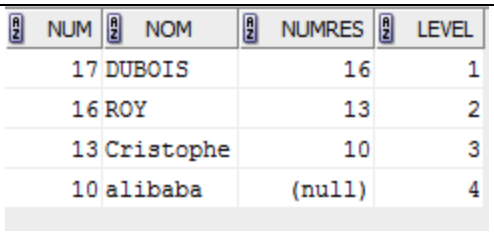
Cette requête va nous donner tous les employés sous l'employé Patoche

```
SELECT num, NOM,NUMRES,LEVEL
 FROM EMPLOYES
START WITH NOM ='Patoche'
 CONNECT BY PRIOR NUM = NUMRES;
```

NUM	NOM	NUMRES	LEVEL
11	Patoche	10	1
14	FAFAR	11	2

CONNECT BY: cette clause est obligatoire, elle permet de connecter deux colonnes (clé primaire et clé enfant) dans une même table et indique au système comment présenter l'information (dans notre cas, si on souhaite avoir les subordonnées ou les responsables)
PRIOR : indique le sens du parcours de la hiérarchie (ou de l'arbre).
Selon qu'il soit placé à gauche, on parcourt l'arbre vers le bas (on extrait les subordonnés), ou à droite, on parcourt l'arbre vers le haut (on extrait les supérieurs)

```
SELECT num, NOM,NUMRES,LEVEL
 FROM PERSONNE
START WITH NOM='DUBOIS'
 CONNECT BY PRIOR NUM =
NUMRES;
```

SELECT num, nom,numres,level FROM PERSONNE START WITH NOM='DUBOIS' CONNECT BY NUM = PRIOR NUMRES;	

NUM	NOM	NUMRES	LEVEL
17	DUBOIS	16	1
16	ROY	13	2
13	Cristophe	10	3
10	alibaba	(null)	4

Dans le premier cas, nous avons ramené les subordonnées de DUBOIS
Dans le deuxième cas, nous avons ramené tous les responsables de DUBOIS.

START WITH : cette clause est optionnelle, elle indique, pour quelle occurrence (dans notre cas pour quel employé) on doit sélectionner les subordonnées
La pseudo colonne LEVEL est ajoutée pour montrer le niveau de hiérarchie entre les enregistrements.

Création de séquence pour Insertion :

```
CREATE SEQUENCE <Nom_de_sequence> INCREMENT BY
<intervalle>
START WITH <value_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 WITN 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 nombre 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.

ATTETION !!!:

- ✓ 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.