

## 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 subordonnés 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)

<pre>SELECT num, NOM, NUMRES, LEVEL FROM PERSONNE START WITH NOM = 'DUBOIS' CONNECT BY <b>PRIOR</b> NUM = NUMRES;</pre>	
---	--

<pre> SELECT num, nom,numres,level FROM PERSONNE START WITH NOM='DUBOIS' CONNECT BY NUM = PRIOR NUMRES; </pre>	<table border="1"> <thead> <tr> <th>NUM</th> <th>NOM</th> <th>NUMRES</th> <th>LEVEL</th> </tr> </thead> <tbody> <tr> <td>17</td> <td>DUBOIS</td> <td>16</td> <td>1</td> </tr> <tr> <td>16</td> <td>ROY</td> <td>13</td> <td>2</td> </tr> <tr> <td>13</td> <td>Cristophe</td> <td>10</td> <td>3</td> </tr> <tr> <td>10</td> <td>alibaba</td> <td>(null)</td> <td>4</td> </tr> </tbody> </table>	NUM	NOM	NUMRES	LEVEL	17	DUBOIS	16	1	16	ROY	13	2	13	Cristophe	10	3	10	alibaba	(null)	4
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.