

# Introduction aux bases de données

Gestion de l'information hiérarchisée

# Gestion de l'information hiérarchisée

## Plan de la séance

- Retour sur la dernière séance:
  - Point de vue de l'étudiant
  - Point de vue de l'enseignant.
- Rappels:
  - La commande SELECT
- Données hiérarchisées:
  - Définition

# Information hiérarchisée: Définition

- Définition:

Des données hiérarchiques sont des données stockées dans une table avec une relation **réursive** (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).

- Exemple

Dans la table ci-après, vous remarquerez que le MGR est un numéro d'employé. L'employé FORD a comme MGR 7566, et quand on regarde dans la colonne EMPNO, on trouve le numéro 7566 qui correspond à JONES. On peut lire donc que le supérieur hiérarchique de FORD est JONES.

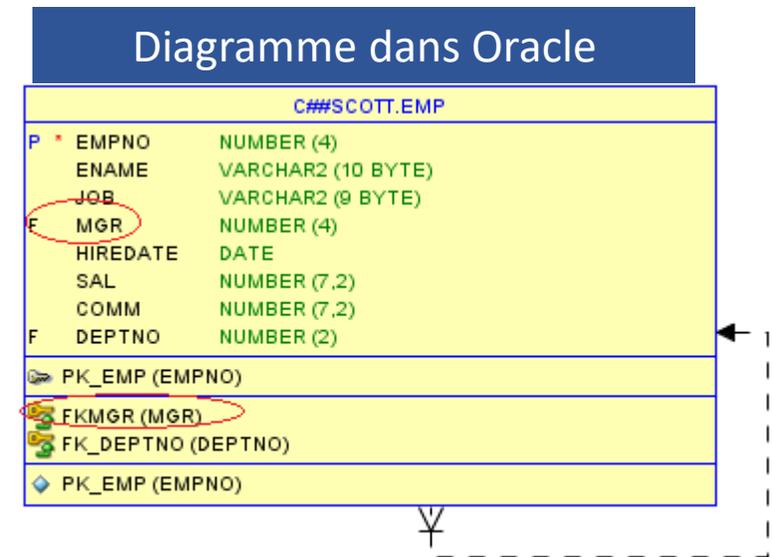
Est-ce que JONES a un supérieur hiérarchique ? La réponse est oui. Il s'agit du MGR 7839 qui correspond à KING

EMPNO	ENAME	JOB	MGR
7839	KING	PRESIDENT	(null)
7698	BLAKE	MANAGER	7839
7782	CLARK	MANAGER	7839
7566	JONES	MANAGER	7839
7902	FORD	ANALYST	7566
7369	SMITH	CLERK	7902
7499	ALLEN	SALESMAN	7698
7521	WARD	SALESMAN	7698
7654	MARTIN	SALESMAN	7698
7844	TURNER	SALESMAN	7698
7900	JAMES	CLERK	7698
7934	MILLER	CLERK	7782
7876	ADAMS	CLERK	7788
7788	SCOTT	ANALYST	7566

# Information hiérarchisée: Diagramme référentiel

Le diagramme référentiel d'une tel table, va montrer le lien entre deux colonnes de la même table

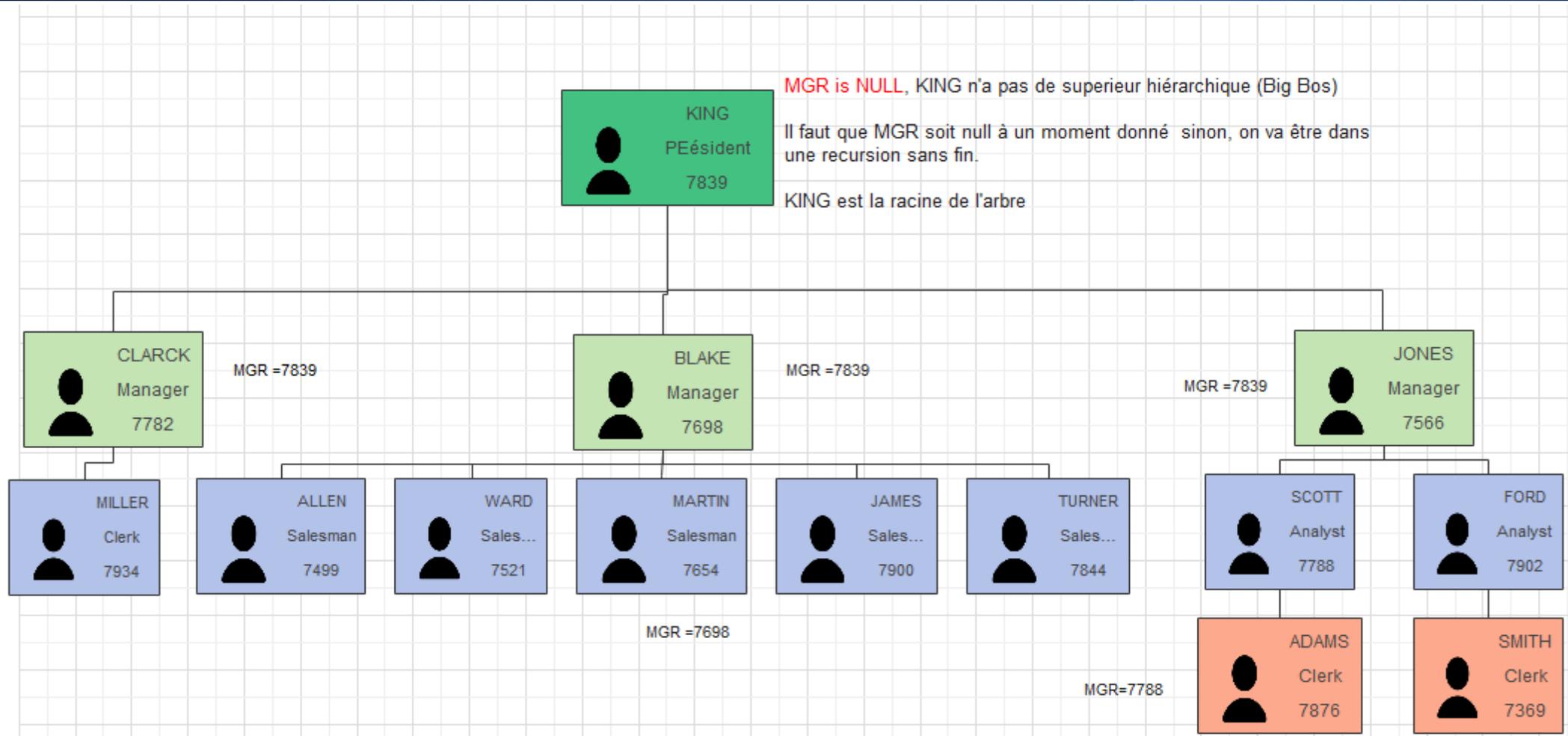
Dans les diagrammes suivants, on voit bien qu'il y a un lien entre empno (pk) et MGR (fk)



Le fait que le trait est en pointillé cela veut dire que MGR peut être null.

Plus de détails concernant le type de trait (plein ou pointillé) seront donnés dans un autre cours: Le modèle relationnel

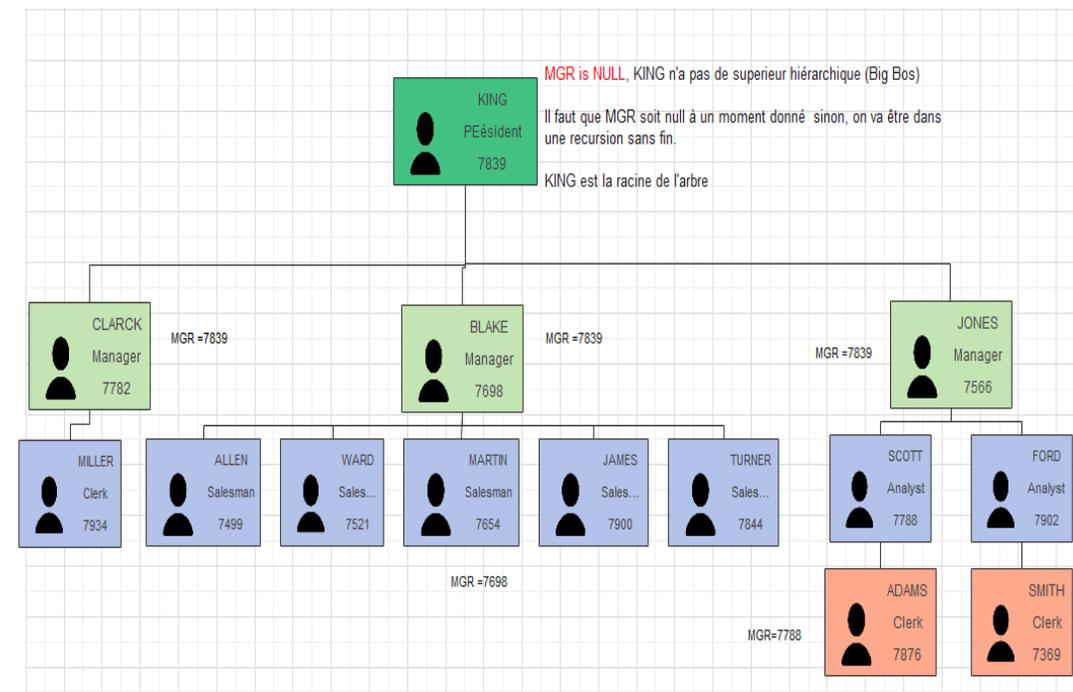
# Information hiérarchisée: Exemple



# Information hiérarchisée: L'arbre à l'envers

## Contraintes:

- Chaque employé a un 0 ou **un** responsable hiérarchique.
- Chaque employés pourrait avoir 0 ou plusieurs subordonnés
- Il y a toujours une fin(un chef) pour l'arbre. Éviter d'avoir une récursion sans fin .
- Le parcours de l'arbre peut se faire vers le bas, dans ce cas on cherche les subordonnés.
- Le parcours de l'arbre peut se faire ver le haut, dans ce cas on cherche les responsable hiérarchique.



# Information hiérarchisée: Syntaxe et mise en œuvre

## Syntaxe

```
SELECT <liste_colonnes>, [ LEVEL ]  
FROM <nom_table>  
[ START WITH <condition(s)> ]  
CONNECT BY [PRIOR] <condition(s)>;
```

**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 :** (optionnelle mais importante) indique le sens du parcours de la hiérarchie (ou de l'arbre).

Selon qu'il soit placé à gauche de la clé parent, on parcourt l'arbre vers le bas (on extrait les subordonnés), ou à droite de la clé parent, on parcourt l'arbre vers le haut (on extrait les supérieurs)

La condition est toujours cle parent = cle enfant

# Information hiérarchisée: Syntaxe et mise en œuvre

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

# Information hiérarchisée: Syntaxe et mise en œuvre

## Exemple 1:

```
SELECT ename, job, level
FROM syemp
START WITH ename='JONES'
CONNECT BY PRIOR empno = mgr
ORDER BY level;
```

On ramène le nom, le job et le degré de hiérarchie

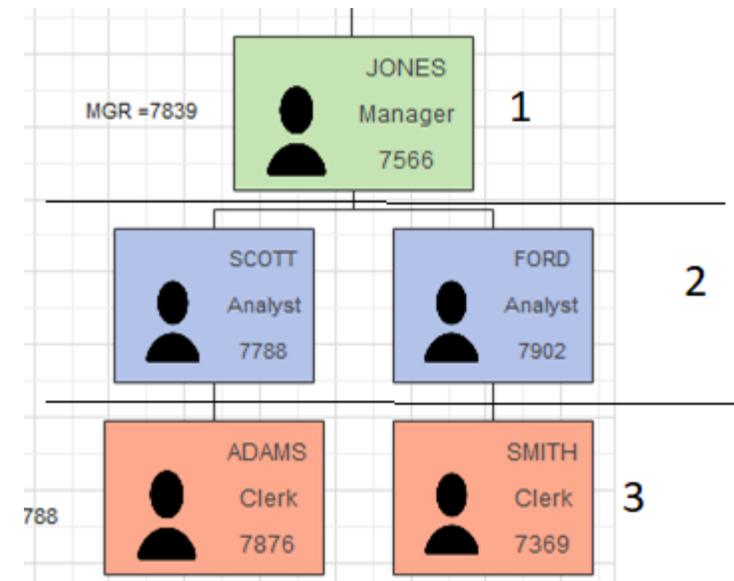
On commence par l'employé JONES.

Ramène-t-on les supérieurs de JONES ? Ou les subordonnés de JONES?

Selon la requête, PRIOR est placé à gauche de empno donc à gauche de la clé parent ce qui veut dire que l'arbre sera parcouru vers le bas à partir de JONES

Remarquez que JONES a le level 1

ENAME	JOB	LEVEL
JONES	MANAGER	1
FORD	ANALYST	2
SCOTT	ANALYST	2
SMITH	CLERK	3
ADAMS	CLERK	3



# Information hiérarchisée: Syntaxe et mise en œuvre

## Exemple 1:

```
SELECT ename, job, level
FROM syemp
START WITH ename='JONES'
CONNECT BY empno = PRIOR mgr
order by level;
```

On ramène le nom, le job et le degré de hiérarchie

On commence par l'employé JONES.

Ramène-t-on les supérieurs de JONES ? Ou les subordonnés de JONES?

Selon la requête, PRIOR est placé à droite de empno donc à droite de la clé parent ce qui veut dire que l'arbre sera parcouru vers le haut à partir de JONES

Remarquez que JONES a le level 1

	ENAME	JOB	LEVEL
1	JONES	MANAGER	1
2	KING	PRESIDENT	2



# Information hiérarchisée: Conclusion

- Les requêtes hiérarchiques permettent d'afficher une relation hiérarchique existant entre des lignes d'une table.
- CONNECT BY est une clause obligatoire qui se compare à une jointure puisqu'elle lie la clé primaire et la clé étrangère.
- Il est possible de déterminer la direction (PRIOR) et le point de départ du parcours (START WITH).
- PRIOR, qu'il soit à gauche ou à droite de l'égalité, il est important de voir si c'est l'attribut PARENT ou l'attribut ENFANT qui est juste après le PRIOR.
- Les données hiérarchiques peuvent être vues comme un arbre à l'envers (voir page 5)
- L'arbre a toujours une racine, une valeur NULL au sommet(voir page 4)
- La ligne du START WITH a toujours le Level 1

# Gestion de l'information hiérarchisée



Conclusion



Questions