

MISE EN PLACE DE LA BASE

1. Connectez vous avec un shell distant sur le serveur faisant tourner la base oracle :

```
ssh usernameagalan@im2ag-oracle.e.ujf-grenoble.fr
```

2. Saisissez votre mot de passe suivi d'un retour de chariot.

3. Connectez vous à la base oracle avec le client SQL Oracle :

```
sqlplus usernameagalan/bd2013
```

4. Chargez la base de données du préambule avec la commande suivante :

```
@/tmp/voisins.sql
```

5. Chargez la base de données du TP avec la commande suivante :

```
@/tmp/empl.sql
```

Préambule au TP : Manipulations de nouvelles fonctionnalités de SQL
--

Nous allons commencer par introduire quelques concepts et éléments de syntaxe SQL nouveaux, qui pourraient vous être utiles dans la réalisation des requêtes du TP même. Nous allons donc introduire un schéma de base de données simple sur lequel nous donnerons quelques requêtes qui illustrent ces nouveaux éléments de SQL.

Nous nous allons prendre l'exemple d'une base des familles habitant Rue des Espoirs, leur ancienneté dans la Rue, ainsi que leur relations de voisinage.

On considère la relation Famille(NumR, NomF, DateE) ; avec NumR le numéro de la maison dans la rue, NomF le nom de la famille, DateE la date d'emménagement de la famille.

Contenu de la relation :

Famille('DUPONT', 1, '10-OCT-2003') ;

Famille('Durand', 2, '16-JAN-1990') ;

Famille('MARTIN', 3, '2-MAR-2004') ;

Famille('DOE', 4, '5-JUL-1993') ;

Famille('SMITH', 5, '11-NOV-2013') ;

La relation Voisin(NumR, NumRV) ; la famille habitant au numéro NumR de la rue a pour voisin la famille habitant au numéro NumRV.

Contenu de la relation :

Voisin(null,1)

Voisin (1,2)

Voisin (2,3)

Voisin (3,4)

Voisin (4,5)

Voisin(5,null)

La première maison de la rue n'a pas de voisin à gauche (null) et la dernière maison de la rue n'a pas de voisin à droite (null).

Il en découle que la direction 'droite' va du numéro le plus petit au numéro le plus grand dans la rue et vice-versa pour la 'gauche'.

On considérera qu'une hiérarchie de voisinage à droite et à gauche pour un numéro X correspond à la suite successive des voisins dans cette direction jusqu'à ce qu'on atteigne une valeur null.

Illustrations des éléments de syntaxe SQL :

- Imaginons que l'on veuille afficher les dates d'emménagement de 2 familles, mais qu'on ne se sait pas de comment les noms sont enregistrés dans la base. Sont-ils tout en majuscule, tout en minuscules, etc ?*

On peut alors utiliser les fonctions upper() et lower(), qui permettent respectivement de convertir une chaîne en majuscules ou en minuscules. Que font les requêtes suivantes ?

```
SELECT DateE  
From Famille  
Where upper(NomF)='DURAND' or upper(NomF)='SMITH' ;
```

```
SELECT DateE  
  
From Famille  
  
Where lower(NomF)='durand' or lower(NomF) = 'smith' ;
```

```
SELECT DateE, lower(NomF)  
  
From Famille  
  
Where lower(NomF)='durand' or lower(NomF) = 'smith' ;
```

- Si jamais nous avons besoin de n'afficher qu'une partie de la date dans un format particulier, nous pourrions utiliser la fonction `to_char(date, format)`. Que font les requêtes suivantes ?

```
SELECT to_char(DateE, 'MON') From Famille ;
```

```
SELECT to_char(DateE, 'YYYY') From Famille ;
```

```
SELECT to_char(DateE, 'Year') From Famille ;
```

```
SELECT NomF From Famille Where to_char(DateE, 'Mon') = 'Nov' ;
```

- On veut maintenant afficher la hiérarchie de voisinage à gauche ou à droite d'une maison donnée. SQL fournit pour cela les clauses `CONNECT BY [...] START WITH [...]`, et l'attribut `level`
- `CONNECT BY/START WITH` indique l'ordre dans lequel on veut parcourir l'information, et par quel nuplet on commence

```
SELECT NUMR
```

```
From Voisin
```

```
CONNECT BY PRIOR NumR=NumRV
```

```
START WITH NumR=3 ;
```

Cette requête se lit comme : commencer par `NUMR=3` puis chercher les nuplets pour lesquels ce `numR` est `numRV`, puis recommencer. Ainsi

`numR=3` -> le(s) nuplet(s) pour lequel `NUMRV=3` est (2,3)

`numR=2` -> le nuplet pour lequel `NUMRV=2` est (1,2)

`numR=1` -> ne nuplet pour lequel `NUMRV=1` est (null,1), étant donné la valeur null, l'affichage ne se fait pas pour ce nuplet.

```
SELECT nvl(NUMR, 0)
```

```
From Voisin
```

```
CONNECT BY PRIOR NumR=NumRV
```

```
START WITH NumR=3 ;
```

C'est la même chose mais le dernier n-uplet sera affiché grâce à la fonction `nvl`.

que font les requêtes suivantes ?

```
SELECT NUMR
```

```
From Voisin
```

```
CONNECT BY PRIOR NumRV=NumR
```

```
START WITH NumRV=1 ;
```

```
SELECT NOMF
```

```
From Famille Natural Join Voisin
```

```
CONNECT BY PRIOR NumRV=NumR
```

```
START WITH NumR=1 ;
```

On peut demander le niveau dans la hiérarchie de chaque information grâce à level :

```
SELECT NOMF , level
```

```
From Famille Natural Join Voisin
```

```
CONNECT BY PRIOR NumRV=NumR
```

```
START WITH NumR=1 ;
```

- *On peut utiliser la fonction lpad afin de rendre l'affichage plus plaisant en indentant chaque niveau de la hiérarchie.*

```
SELECT lpad(NomF, length(NomF) + level*2)
```

```
From Famille Natural Join Voisin
```

```
CONNECT BY PRIOR NumRV=NumR
```

```
START WITH NumR=1 ;
```

- *Il se peut que le nom des colonnes soit long ou peu pratique et que l'on veuille changer son nom. SQL permet de faire cela avec l'instruction AS. On peut également omettre le AS et écrire juste Famille F. Que fait la requête suivante ?*

```
SELECT NomF AS NF, NumR AS NR
```

```
From Famille natural join Voisin ;
```

- Si jamais on cherche une famille mais qu'on ne connaît qu'une partie du nom, comment faire en SQL pour la trouver ? L'opérateur LIKE suivi d'une chaîne format permet de capturer une correspondance partielle. Par exemple pour les noms finissant par T la condition correspondante en SQL serait `NomF LIKE '%T'`. Le caractère % s'appelle un caractère générique qui peut correspondre à n'importe quelle chaîne de longueur arbitraire, mais qui est cependant contrainte par les caractères avant et après le %. Que pensez-vous que fait cette requête ?

```
SELECT NomF NF From Famille Where lower(NomF) LIKE '%a%' ;
```

```
SELECT NomF NF From Famille Where lower(NomF) LIKE 'd%' ;
```

- SQL fournit également de nombreuses fonctions de manipulation de dates. Par exemple `sysdate` correspond toujours à la date et heure actuelle (tels que sur le serveur de bases de données) et `add_months` permet de soustraire ou d'ajouter un nombre de mois à une date donnée. Que fait cette requête ?

```
SELECT NomF
```

```
From Famille
```

```
Where DateE < sysdate AND DateE > add_months(sysdate, -12) ;
```

TP

Soit les relations suivantes :

EMP(EMPNO, ENAME, JOB, MGR, HIREDATE, SAL, COMM, DEPTNO)

Avec

<code>EMPNO NUMBER(4)</code>	<code>: numéro d'employé</code>
<code>ENAME CHAR(10)</code>	<code>: nom de l'employé</code>
<code>JOB CHAR(9)</code>	<code>: emploi</code>
<code>MGR NUMBER(4)</code>	<code>: numéro du chef hiérarchique</code>
<code>HIREDATE DATE</code>	<code>: date d'embauche</code>
<code>SAL NUMBER(7,2)</code>	<code>: salaire mensuel</code>
<code>COMM NUMBER(7,2)</code>	<code>: commission mensuelle</code>
<code>DEPTNO NUMBER(2)</code>	<code>: numéro de département</code>

Voici le contenu de la relation EMP :

empno	ename	job	mgr	hiredate	sal	comm	deptno
7369	smith	clerk	7902	17-DEC-2000	800	null	20
7499	allen	salesman	7698	20-FEB-1999	1600	300	30
7521	ward	salesman	7698	22-FEB-1991	1250	500	30
7566	jones	manager	7839	2-APR-2000	2975	null	20
7654	martin	salesman	7698	28-SEP-1998	1250	1400	30
7698	blake	manager	7839	1-MAY-1997	2850	null	30
7782	clark	manager	7839	9-JUN-1996	2450	null	10
7788	scott	analyst	7566	09-DEC-1995	3000	null	20
7839	king	president	null	17-11-1994	5000	null	10

7844	turner	salesman	7698	8-SEP-1998	1500	0	30
7876	adams	clerk	7788	12-JAN-1993	1100	null	20
7900	james	clerk	7698	3-DEC-2000	950	null	30
7902	ford	analyst	7566	3-DEC-1992	3000	null	20
7934	miller	clerk	7782	23-JAN-1999	1300	null	10

DEPT(DEPTNO,DNAME,LOC)

Avec

DEPTNO NUMBER(2)

: le numéro d'un département

DNAME CHAR(14)

: le nom du département

LOC LOC CHAR(13)

: la ville où se trouve le département

Voici le contenu de la relation DEPT :

deptno	dname	loc
10	accounting	new york
20	research	dallas
30	sales	chicago
40	operations	boston

SALGRADE(GRADE,LOSAL,HISAL)

avec

GRADE NUMBER

: grille de salaire

LOSAL NUMBER

: salaire minimum dans la grille

HISAL NUMBER

: salaire maximum dans la grille

Voici le contenu de la relation SALGRADE :

grade	losal	hisal
1	700	1200
2	1201	1400
3	1401	2000
4	2001	3000
5	3001	9999

Prenez le temps

- *de bien comprendre les relations, et les liens entre elles*
- *de vérifier que les réponses à vos requêtes sql sont correctes (en vérifiant vous même avec les données qui sont par ailleurs peu nombreuses)*

Nous avons essayé de donner un niveau de difficulté par question. Ainsi chaque question est écrite en vert, orange ou rouge :

- *vert : requête peu difficile*
- *orange : requête moyennement difficile*
- *rouge : requête avec des difficultés d'analyse ou d'écriture sql*

- 1. nom et date d'embauche des employés du département 20 par ordre chronologique inverse en indiquant le mois de l'embauche en toute lettre et l'année en chiffre. la colonne indiquant l'embauche s'intitulera EMBAUCHE, celle indiquant le nom s'intitulera NOM.*
- 2. indiquer les salaires dans l'entreprise (commission non comprise) en donnant le nom de chaque salarié (la colonne s'intitulera NOM) et son salaire, l'affichage se fera par ordre de salaire décroissant, puis par ordre alphabétique des employés.*
- 3. indiquer les revenus dans l'entreprise (salaire et commission comprise) en donnant le nom de chaque salarié (la colonne s'appellera NOM) et son revenu (la colonne s'intitulera REVENU) par ordre de revenu décroissant, puis par ordre alphabétique des employés*
- 4. Donner le salaire moyen dans l'entreprise. Le résultat s'intitulera MOYSAL.*
- 5. Donner le salaire moyen de chaque département, la colonne des départements s'intitulera DEPARTEMENT, celle des moyennes de salaire MOYSAL. Le résultat sera par ordre décroissant des moyennes de salaires.*
- 6. Donner la liste alphabétique des employés (la colonne s'intitulera NOM), en indiquant leur revenu (salaire + commission), la colonne s'intitulera REVENU. Le résultat sera affiché par ordre alphabétique des employés.*
- 7. Donner le revenu moyen dans l'entreprise, le résultat s'intitulera MOYREVENU*
- 8. Donner le revenu moyen de chaque département de l'entreprise. La colonne indiquant le département s'intitulera DEPARTEMENT, celle indiquant le revenu moyen MOYREVENU. Le résultat sera affiché par ordre décroissant de la moyenne des revenus.*
- 9. employés dont le nom commence par M et dont le salaire (hors commission) est supérieur ou égal à 1290\$ par ordre de salaire décroissant. La colonne indiquant le nom s'intitulera NOM.*
- 10. employés dont le nom commence par M et dont le revenu (salaire et commission) est supérieur ou égal à 1290\$ par ordre de revenu décroissant. La colonne indiquant le nom s'intitulera NOM, celle du revenu REVENU.*
- 11. départements ayant des CLERK, MANAGER **ET** des ANALYST. Le résultat affiché s'intitulera DEPARTEMENT*

12. *pour chaque employé, donner son supérieur hiérarchique. S'il n'a pas de supérieur hiérarchique, indiquez 0. La colonne indiquant l'employé sera intitulé EMPLOYE, celle indiquant son supérieur hiérarchique s'intitulera CHEF. Le résultat sera ordonné sur le numéro de l'employé.*
13. *Même question que la 12, mais ce sont les noms que l'on veut. Pour ceux qui n'auraient pas de supérieur hiérarchique, indiquez BIG BOSS*
14. *Pour chaque employé (indiquez son numéro), donner son ou ses subordonnés (indiquez leur numéro). S'il n'a pas de subordonné, indiquez 0. La colonne indiquant l'employé sera intitulé EMPLOYE, celle indiquant subordonnés s'intitulera SUBORDONNE. Le résultat sera ordonné sur le numéro de l'employé.*
15. *(question verte si vous avez fait la question 14, rouge sinon) Pour chaque employé (indiquez son numéro), donner le nombre de ses subordonnés. La colonne indiquant l'employé sera intitulé EMPLOYE, celle indiquant son supérieur hiérarchique s'intitulera NBSUBORDONNE. Le résultat sera ordonné sur le nombre décroissant de subordonnés.*
16. *Nom et numéro de département des supérieurs hiérarchiques dont les subordonnés perçoivent une commission. La colonne indiquant le nom s'intitulera NOM, celle indiquant le numéro du département DEPARTEMENT.*
17. *Nom des employés des départements de Chicago ou Dallas dont le revenu (salaire et commission) est supérieur à 1000\$, par ordre décroissant du revenu, puis par ordre alphabétique des employés. La colonne indiquant le revenu s'intitulera REVENU, celle indiquant le nom s'intitulera NOM.*
18. *Indiquez le nom, le revenu (salaire + commission), le numéro de département, le nom et le revenu du supérieur hiérarchique pour tous les employés qui ont un revenu (salaire + commission) supérieur à leur supérieur hiérarchique ? Le nom de l'employé sera intitulé NOM_EMP, son revenu REV_EMP, son département DEP_EMP, le nom de son supérieur NOM_CHEF, et le revenu du chef REV_CHEF*
19. *quels sont les métiers présents dans tous les départements ? Le résultat s'intitulera METIER et sera ordonné de façon croissante.*
20. *quelles sont les grilles de salaire pour chaque département ? Le résultat sera ordonné par département puis par grille de salaire.*
21. *afficher la hiérarchie de l'entreprise : afficher les noms des employés dans l'ordre hiérarchique, en indiquant pour chacun son niveau de hiérarchie*
22. *afficher la hiérarchie de l'entreprise : afficher les noms des employés dans l'ordre hiérarchique. Indenter la réponse (au moins 3 crans par niveau de hiérarchie). Le résultat s'intitulera HIERARCHIE.*
23. *nombre d'employés par niveau hiérarchique de l'entreprise. Le niveau de hiérarchie s'intitulera NIVEAU_HIERARCHIE, le nombre d'employés NB_EMP.*

24. *Moyenne des salaires par niveau hiérarchique. Le niveau de hiérarchie s'intitulera NIVEAU_HIERARCHIE, la moyenne des salaires MOY_SAL.*
25. *Quels sont les départements dont les revenus des employés sont au total les plus élevés ?*
26. *Liste des départements (numéro, nom et ville) qui ont plus de 3 employés, mais qui ne sont pas à Chicago. Le résultat s'intitulera DEPARTEMENT.*
27. *Noms des employés embauchés avant le 30 novembre 1999. Le résultat sera intitulé NOM et ordonnée alphabétiquement.*
28. *Noms des employés embauchés depuis 20 ans ou plus, ordonnés par ordre chronologique des embauches. La colonne des noms s'intitulera NOM, celle de la date d'embauche DATE_EMBAUCHE.*