



TP 2: Fonctions Analytiques

Préambule

Les fonctions analytiques ont été introduites dans Oracle 8i. Elles permettent d'extraire des informations de manière beaucoup plus simple et performante qu'auparavant pour une classe assez large de problèmes. Les fonctions analytiques opèrent donc sur un ensemble de lignes définies de manière relative; elles se distinguent des fonctions agrégats qui opèrent sur l'ensemble d'un groupe défini par le group by. En ce sens, elles sont d'une granularité plus fine.



EXERCICE 1

On créera dans son compte les tables Emp et Dept du compte Scott de la base de données Ora10g. On s'intéresse au nombre d'employés et au salaire moyen annuel des employés selon différents critères (département, emploi/métier, année d'embauche).

- Donner le nombre d'employés et le salaire moyen par département et par emploi.
- Même question mais on veut également le nombre d'employés et le salaire moyen pour chaque département. Proposer une solution en SQL de base, une solution utilisant GROUPING SET, une solution voisine utilisant la clause ROLLUP, enfin la solution avec GROUP BY dName, ROLLUP (Job).
- Mêmes questions qu'en 2, mais utiliser les fonctions CASE WHEN...THEN... ELSE... END et GROUPING pour améliorer l'affichage.
- Mêmes questions qu'en 3 en ajoutant l'année d'embauche comme dimension.
- Même question que 3 mais en ajoutant tous les sous-totaux. Proposer une solution en SQL de base et une solution utilisant la clause CUBE.
- Même question que 5, mais en ajoutant l'année d'embauche comme dimension.

```
Sql> CREATE TABLE dept(deptno NUMBER(2) NOT NULL,
dname VARCHAR2(14), loc VARCHAR2(13));
```

```
- INSERT INTO dept VALUES(10, 'ACCOUNTING', 'NEW YORK');
- INSERT INTO dept VALUES(20, 'RESEARCH', 'DALLAS');
- INSERT INTO dept VALUES(30, 'SALES', 'CHICAGO');
- INSERT INTO dept VALUES(40, 'OPERATIONS', 'BOSTON');
```

```
CREATE TABLE emp (empno NUMBER(4) NOT NULL, ename
VARCHAR2(10), job VARCHAR2(9), mgr NUMBER(4), hiredate DATE,
sal NUMBER(7,2), comm NUMBER(7,2), deptno NUMBER(2));
```

```
INSERT INTO emp VALUES(7369, 'SMITH', 'CLERK', 7902,
'17/12/80', 800, NULL, 20);
INSERT INTO emp VALUES(7499, 'ALLEN', 'SALESMAN', 7698,
'20/02/81', 1600, 300, 30);
INSERT INTO emp VALUES(7521, 'WARD', 'SALESMAN', 7698,
'22/02/81', 1250, 500, 30);
INSERT INTO emp VALUES(7566, 'JONES', 'MANAGER', 7839,
'02/04/81', 2975, NULL, 20);
INSERT INTO emp VALUES(7654, 'MARTIN', 'SALESMAN', 7698,
'28/09/81', 1250, 1400, 30);
INSERT INTO emp VALUES(7698, 'BLAKE', 'MANAGER', 7839,
'01/05/81', 2850, NULL, 30);
INSERT INTO emp VALUES(7782, 'CLARK', 'MANAGER', 7839,
'09/06/81', 2450, NULL, 10);
INSERT INTO emp VALUES(7788, 'SCOTT', 'ANALYST', 7566,
'19/04/87', 3000, NULL, 20);
INSERT INTO emp VALUES(7839, 'KING', 'PRESIDENT', NULL,
'17/11/81', 5000, NULL, 10);
INSERT INTO emp VALUES(7844, 'TURNER', 'SALESMAN', 7698,
'08/09/81', 1500, NULL, 30);
INSERT INTO emp VALUES(7876, 'ADAMS', 'CLERK', 7788,
'23/05/87', 1100, NULL, 20);
INSERT INTO emp VALUES(7900, 'JAMES', 'CLERK', 7698,
'03/12/81', 950, NULL, 30);
INSERT INTO emp VALUES(7902, 'FORD', 'ANALYST', 7566,
'03/12/81', 3000, NULL, 20);
INSERT INTO emp VALUES(7934, 'MILLER', 'CLERK', 7782,
'23/01/82', 1300, NULL, 10);
```

EXERCICE 2

On souhaite concevoir un entrepôt de données permettant d'étudier les ventes de 3 succursales d'une entreprise.

Les succursales se trouvent dans trois villes: Chicago, Toronto et Vancouver.

- Faire une requête pour obtenir une table Ventes ayant pour attribut Ville, Période, Item, Qte, regroupant les informations des trois villes.
- Construire le cube de données du total des ventes selon les dimensions ville, année et item.
- On souhaite maintenant regrouper les objets par département (Téléphonie, Informatique et Ménager). Créer la table Items ayant pour attributs IdItem, NomItem, Dept et la peupler à partir de Ventes.
- Construire le cube de données du total des ventes selon les dimensions ville, année et département.
- Donner le total des ventes par produit et par département selon les dimensions ville et année.

CHICAGO

ITEM	PERIODE	QTE
27	01/01/99	25
28	01/01/99	15
25	01/02/99	8
26	01/02/99	12
23	01/03/99	3
24	01/02/00	4
22	01/03/00	3

TORONTO

ITEM	PERIODE	QTE
27	01/02/99	11
28	01/02/99	9
25	01/05/99	13
26	01/06/99	7
23	01/01/99	5
24	01/01/00	7

VANCOUVER

ITEM	PERIODE	QTE
27	01/10/99	15
28	01/12/99	5
25	01/06/99	2
26	01/07/99	8
23	01/02/99	10
24	01/02/00	5
22	01/01/00	5
22	01/03/00	8

Questions fréquentes

- Comment obtenir une valeur de type **date** à partir d'une chaîne de caractères ?
 - préciser le format par défaut: `alter session set nls_date_format = 'DD/MM/YY';`
 - ou appeler la fonction de conversion `to_date('01/01/93', 'DD/MM/YY')`. Exemple de formats : 'YYYY-MM-DD' ou 'YYYY-MM-DD HH24:MI:SS' (en précisant l'heure)
- Comment obtenir le mois et l'année d'une **date** ?
 - `to_char(o_orderdate, 'MM')` et `to_char(o_orderdate, 'YY')`
- L'ordre des attributs du **rollup** importe-t-il ?
 - oui : `group by rollup(a, b, c)` est différent de `group by rollup(c, b, a)`
- Comment compléter le résultat d'une requête R avec un nouvel attribut A ayant une valeur constante ?
 - réponse générale: Effectuer un produit cartésien entre R et une relation T(A) contenant un seul n-uplet.
 - réponse spécifique: ajouter directement le nouvel attribut dans la clause select de R: `select v as A, ... from` La valeur v peut être *null*.
- Peut-on faire l'**union** de 2 relations R1 et R2 qui n'ont pas le même schéma ?
 - Non, les schémas de R1 et R2 doivent être de type identiques (i.e., type des attributs identiques 2 à 2).
- Peut-on faire l'union de sous requêtes qui sont chacune **triées** avec un **order by** ?
 - Non, il faut effectuer le tri après l'union, donc pas d' **ORDER BY** dans des sous-requêtes combinées par une union.
- Peut-on ajouter un nouvel attribut avec une valeur constante dans le résultat d'une requête ?
 - Oui, il est possible d'ajouter un attribut avec une valeur constante (par exemple la valeur null) dans la clause **SELECT**.