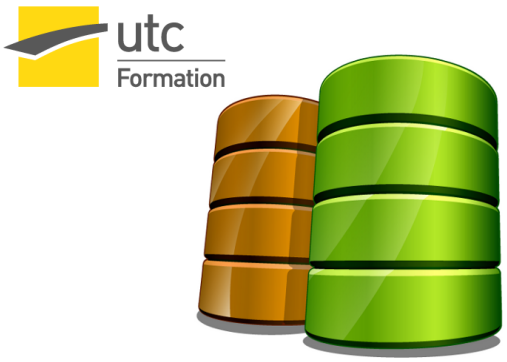
Exploitation multi-hiérarchique et multi-dimensionnelle d'un data warehouse

Data warehouse
http://dwh.crzt.fr



Stéphane Crozat

1 septembre 2016

Table des matières

Objectifs

- savoir interroger en SQL un data warehouse en vue d'applications décisionnelles

Introduction

- Volume de cours : 1h
- Volume d'exercice : 3h

Extensions SQL pour l'exploration de données

I

Certains SGBDR dont Oracle propose les extensions ROLLUP et CUBE à l'instruction SQL GROUP BY pour faciliter l'exploration de données.

1. Exploration multi-niveaux avec GROUP BY ROLLUP

Syntaxe : GROUP BY ROLLUP

```
1 SELECT ...
2 GROUP BY ROLLUP (a, b, c, ...)
```

GROUP BY ROLLUP (a, b, c, ...) permet de créer tous les sous totaux selon les *attributs ordonnés de groupement* (ici a, b et c) en allant du plus général (a) au plus détaillé (c).

Méthode

Cette clause est typiquement utilisée pour parcourir une hiérarchie.

Exemple

```
1 SELECT d.tri, d.mon, count(*)
2 FROM ventes v, date d
3 WHERE v.dat=d.dat
4 GROUP BY ROLLUP (d.tri, d.mon);
```

1	1	7159
1	2	7630
1	3	9300
1		24089
2	4	7317
2	5	8017
2	6	8977
2		24311
3	7	7397
3	8	8328
3	9	9042
3		24767
4	10	7448
4	11	7764
4	12	12845
4		28057
		101224

Résultat d'exécution GROUP BY ROLLUP (d.tri, d.mon)

⚠ *Attention : Ordre*

L'ordre (a, b, c) est important et doit être du plus gros grain au plus fin.

2. Exploration multi-dimensions avec GROUP BY CUBE

Syntaxe : GROUP BY CUBE

```
1 SELECT ...
2 GROUP BY CUBE (a, b, c, ...)
```

GROUP BY CUBE permet de créer tous les sous totaux possibles pour toutes les combinaisons des attributs de groupement.

Méthode

Cette clause est typiquement utilisée pour faire des *analyses croisées*.

Exemple

```
1 SELECT p.bs, m.bs, count(*)
2 FROM ventes v, f_dw_produit p, f_dw_mag m
3 WHERE v.pro=p.isbn AND v.mag=m.mag
4 GROUP BY CUBE (p.bs, m.bs);
```

Livre BS	Mag BS	Nb Ventes		
		476394		
	0	137368		
	1	339026	71.17%	*Mag BS*
0		324027	68.02%	*Ventes des Livres Non BS en général*
0	0	108736	79.16%	*Ventes des Livres Non BS dans les Mag Non BS*
0	1	215291	63.50%	*Ventes des Livres Non BS dans les Mag BS*
1		152367	31.98%	*Ventes des Livres BS en général*
1	0	28632	20.84%	*Ventes des Livres BS dans les Mag non BS*
1	1	123735	36.50%	*Ventes des Livres BS dans les Mag BS*

Exemple de résultat CUBE analysé dans un tableur

Conseil

Il est souvent nécessaire de récupérer le résultat dans un tableur pour le manipuler et l'interpréter.

Attention

Si l'on projette trop de dimensions dans le CUBE la combinatoire devient grande et les résultats difficiles à interpréter.

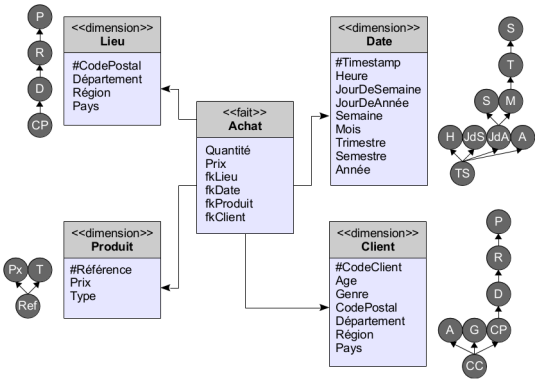
Application : Exploitation multi-dimensionnelle de données

II

1. Projet Fantastic : Rappel

Rappel

Cet exercice est la suite du projet Fantastic commencé dans le module *Introduction à la modélisation dimensionnelle*.



Exemple de modèle dimensionnel en étoile

Rappel

Objectifs du modèle dimensionnel (cf. p.)

2. Exercice : Projet Fantastic : Exploitation multi-dimensionnelle de données

Question 1

[solution n°1 p.15*]*

Effectuez une requête multi-niveaux permettant de calculer les ventes sur la hiérarchie date-mois-trimestre.

Question 2

[solution n°2 p.16*]*

Étudiez l'éventuelle influence des magasins (certains sont-ils plus performants ?) et de l'implantation dans les départements (certains sont-ils plus propices ?).

Indice :

Afin de ne pas être dépendant de l'organisation des magasins, effectuez la requête uniquement pour les magasins de type 'A' avec rayon BS (ce sont à la fois les magasins les plus nombreux et les plus performants).

Isolation de facteur (cf. p.)

Question 3

[solution n°3 p.17*]*

Effectuer une requête multi-dimensionnelle permettant de regarder si les magasins avec un rayon best-sellers vendent plus de livres best-sellers ou non.

On peut utiliser un tableur pour finaliser les analyses de ratio sur les valeurs renvoyées par la requête CUBE obtenue.

Indice :

Agrégation de faits (cf. p.)

Rappels

III

1. Rappels Oracle pour l'exploration des données

1.1. Sous-requêtes dans la clause FROM

Il est possible de raffiner progressivement une requête en enchaînant les questions dans la clause FROM.

Syntaxe

```
SELECT ... FROM
(SELECT ...
FROM ...
WHERE ...)
WHERE ...
```

Remarque : Sous-sous requête

Il est possible d'imbriquer successivement plusieurs sous-requêtes.

```
SELECT ... FROM
(SELECT ... FROM
  (SELECT...
  FROM ...
  WHERE)
WHERE ...)
WHERE ...
```

Syntaxe : Jointure de sous-requêtes

Il est possible de faire le produit ou la jointure de sous-requêtes.

```
SELECT ... FROM
(SELECT ... FROM
WHERE...)
(SELECT ... FROM
WHERE...)
WHERE ...
```

Méthode

Cette extension est particulièrement utile pour les calculs d'agrégat après filtrage ou pour enchaîner les calculs d'agrégat (par exemple pour faire la moyenne de comptage après regroupement).

Exemple : Enchaînement de calculs d'aggrégat

```
select avg(c) from (select count(b) as c from t group by a)
```

Exemple : Calcul de la moyenne des ventes par jour de la semaine

```
SELECT avg(q) FROM
```

```
(SELECT jds AS jds, COUNT(*) AS q
FROM ventes v
GROUP BY jds)
```

1.2. Fenêtrage des données

Syntaxe : Rownum

```
SELECT ... FROM ... WHERE rownum<=N;
-- avec N le nombre de lignes désirées.
```

Rownum

La restriction `ROWNUM <= N` dans la clause `WHERE` permet filtrer les N premières lignes de la table.

Remarque

`rownum` est une pseudo colonne qu'il est bien entendu possible de projeter : `SELECT rownum FROM ...`

Syntaxe : Utilisation avancée

```
SELECT a1, ..., aN FROM
(SELECT a1, ..., aN, rownum AS rnum FROM t)
WHERE rnum BETWEEN n1 AND n2
```

Cette syntaxe permet de sélectionner une fenêtre sur les données et pas seulement les N premières lignes.

Méthode : Exploration de données massives

Lorsque l'on est en présence de gros volumes de données, et que l'on veut se faire une idée du contenu de ces données, il n'est *pas souhaitable* de faire un simple SELECT *. En il serait trop long de rapatrier les dizaines de milliers de lignes et de plus cela serait inutile puisque seules quelques unes seraient effectivement lues.

L'usage de `rownum` permet de s'intéresser à des fenêtres de données représentatives, pour se faire une idée générale.

1.3. SQL*Plus

*Définition : SQL*Plus*

- SQL*Plus est un client Oracle basique en mode texte, qui n'est plus vraiment utilisé (on utilise Oracle SQL Developer à la place).
- SQL*Plus désigne aussi un langage interne à Oracle destiner à gérer la présentation des résultats de requêtes en mode texte (états textuels).

*Complément : SQL*Plus dans SQL Developer*

Oracle SQL Developer utilise également SQL*Plus mais ne supporte pas toutes les fonctions.

http://www.oracle.com/technetwork/developer-tools/sql-developer/sql-worksheet-commands-097146.html

Méthode : Usages

- Le paramétrage de la présentation des résultats de requête est utile au développeur pour avoir des retours lisibles dans son terminal d'exécution.
- Il peut aussi servir à des parties applicatives comme le formatage pour un export CSV.
- ...

Attention

SQL*PLus ne travaille ni sur le contenu ni sur la structure, uniquement sur la présentation.

1.3.1. Variables d'environnement

Syntaxe

SQL*Plus permet de fixer la valeur de variables d'environnement avec la commande :

```
1 SET param valeur
```

Ces paramètres peuvent être lus avec la commande :

```
1 SHOW param
```

Exemple

```
1 SET heading off
```

Permet de désactiver l'affichage des entêtes de colonne dans le résultat affiché.

Complément

http://docs.oracle.com/cd/B19306_01/server.102/b14357/ch12040.htm

1.3.2. Fichiers d'entrée et de sortie

Syntaxe : Exécuter un fichier

Pour exécuter un fichier contenant des commandes SQL ou SQL*Plus :

```
1 @path/filename
```

Syntaxe : Sortie dans un fichier

Pour enregistrer les résultats d'une exécution de requêtes dans un fichier :

```
1 SPOOL path/filename
2 -- requêtes dont on veut récupérer les résultats dans le fichier
3 SPOOL OFF
```

Complément

*Création d'un fichier CSV avec SQL*Plus* (cf. p.)

1.3.3. Formattage d'une colonne de requête

Syntaxe

```
1 COLUMN nom_colonne FORMAT format
```

- Largeur de la colonne : An
- Chiffre (avec ou sans zéro à gauche) : 9 / 0
- Symboles monétaires : $ / L
- Séparateurs de virgule et de milliers : . / ,
- ...

Exemple

```
1 COLUMN ename FORMAT A15
2 COLUMN sal FORMAT $99,990.00
3 COLUMN mgr FORMAT 999999999
```

Solutions des exercices

> **Solution** n°1

Exercice p. 8

Requête

```
1 SELECT d.dat, d.tri, d.mon, count(*)
2 FROM f_dw_ventes v, f_dw_date d
3 WHERE v.dat=d.dat
4 GROUP BY ROLLUP (d.tri, d.mon, d.dat);
```

Résultat (extrait)

```
1 01-JAN-13   1   1     1338
2 02-JAN-13   1   1     1340
3 03-JAN-13   1   1     1427
4 04-JAN-13   1   1     1325
5 05-JAN-13   1   1     3683
6 08-JAN-13   1   1     1262
7 09-JAN-13   1   1     1394
8 10-JAN-13   1   1     1507
9 11-JAN-13   1   1     1368
10 12-JAN-13   1   1     3513
11 15-JAN-13   1   1     1316
12 16-JAN-13   1   1     1322
13 17-JAN-13   1   1     1605
14 18-JAN-13   1   1     1247
15 19-JAN-13   1   1     3751
16 22-JAN-13   1   1     1418
17 23-JAN-13   1   1     1409
18 24-JAN-13   1   1     1346
19 25-JAN-13   1   1     1492
20 26-JAN-13   1   1     3361
21 29-JAN-13   1   1     1371
22 30-JAN-13   1   1     1322
23 31-JAN-13   1   1     1360
24             1   1    40477
25 01-FEB-13   1   2     1499
26 02-FEB-13   1   2     3347
27 05-FEB-13   1   2     1658
28 06-FEB-13   1   2     1318
29 07-FEB-13   1   2     1230
30 08-FEB-13   1   2     1461
31 09-FEB-13   1   2     3318
```

```
12-FEB-13   1   2     1515
13-FEB-13   1   2     1349
14-FEB-13   1   2     1307
15-FEB-13   1   2     1379
16-FEB-13   1   2     3531
19-FEB-13   1   2     1374
20-FEB-13   1   2     1566
21-FEB-13   1   2     1414
22-FEB-13   1   2     1311
23-FEB-13   1   2     3543
26-FEB-13   1   2     1357
27-FEB-13   1   2     1622
28-FEB-13   1   2     1270
            1   2    36369
01-MAR-13   1   3     1300
02-MAR-13   1   3     3455
05-MAR-13   1   3     1240
...
29-NOV-13   4  11     1372
30-NOV-13   4  11     3489
            4  11    41269
03-DEC-13   4  12     2311
04-DEC-13   4  12     2245
05-DEC-13   4  12     2278
06-DEC-13   4  12     2079
07-DEC-13   4  12     5091
10-DEC-13   4  12     2246
11-DEC-13   4  12     2287
12-DEC-13   4  12     2341
13-DEC-13   4  12     2217
14-DEC-13   4  12     5184
17-DEC-13   4  12     2086
18-DEC-13   4  12     2081
19-DEC-13   4  12     2251
20-DEC-13   4  12     2290
21-DEC-13   4  12     5073
24-DEC-13   4  12     4001
25-DEC-13   4  12     2128
26-DEC-13   4  12      844
27-DEC-13   4  12      782
28-DEC-13   4  12     2542
31-DEC-13   4  12      887
            4  12    53244
            4       135106
                    492113
```

> **Solution** n°2

Exercice p. 9

```
SELECT m.dpt, m.mag, count(*)
FROM f_dw_ventes v, f_dw_mag m
WHERE v.mag=m.mag AND m.ray = 'Author' AND m.bs = '1'
GROUP BY ROLLUP (m.dpt, m.mag);
```

```
13  M106             4680
13  M3               4815
13  M54              4634
```

```
 4 13  M68            4599
 5 13                18728
 6 18  M50            4592
 7 18                 4592
 8 22  M2             4797
 9 22                 4797
10 28  M144           4313
11 28  M21            4933
12 28  M92            4653
13 28                13899
14 30  M29            4596
15 ...
16 92  M138           4573
17 92  M80            4915
18 92                14247
19 95  M148           4759
20 95  M28            4771
21 95  M34            4824
22 95  M37            4867
23 95  M40            4795
24 95  M70            4707
25 95  M81            4767
26 95                33490
27                  281153
```

On observe que les magasins sont assez homogènes, et également les département si on les ramène au nombre de magasins qu'ils contiennent.

> **Solution** n°3

Exercice p. 9

```
1 SELECT b.bs || ';' || m.bs || ';' || count(*) || ';'
2 FROM f_dw_ventes v, f_dw_produit p, f_dw_bs b, f_dw_mag m
3 WHERE v.pro=p.isbn AND p.isbn=b.isbn AND v.mag=m.mag
4 GROUP BY CUBE (b.bs, m.bs);
```

Livre BS	Mag BS	Nb Ventes		
		476394		
	0	137368		
	1	339026	71.17%	*Mag BS*
0		324027	68.02%	*Ventes des Livres Non BS en général*
0	0	108736	79.16%	*Ventes des Livres Non BS dans les Mag Non BS*
0	1	215291	63.50%	*Ventes des Livres Non BS dans les Mag BS*
1		152367	31.98%	*Ventes des Livres BS en général*
1	0	28632	20.84%	*Ventes des Livres BS dans les Mag non BS*
1	1	123735	36.50%	*Ventes des Livres BS dans les Mag BS*

Exemple de résultat CUBE analysé dans un tableur