Chapitre 3

Requêtes : Comptage et découpage

Chapitre 3 : Plan

Comptage et calculs

Fonctions générales

COUNT (*)

COUNT (DISTINCT colonne)

MIN (colonne)

MAX (colonne)

Remarque

L'utilisation des fonctions comme COUNT ne se traduit pas habituellement en algèbre relationnelle, non parce que c'est intraduisible (par exemple COUNT correspond précisément à Card, noté aussi # en mathématique), mais parce que le SQL en est trop éloigné.

Comptage simple

COUNT (*)

renvoie le dernier RowNum (le numéro du dernier t-uple) du résultat du mapping

Exemple :

Quel est le nombre d'acteurs enregistrés dans la table Acteurs ?

SELECT COUNT (*)
FROM Acteurs
;

Remarque

Si la table est mal construite, si elle comporte des **lignes de valeurs manquantes**, alors ces lignes sont comptées par **COUNT (*)**

Si la table est correcte, alors elle comporte une **clé primaire**, et **COUNT (*)** donne le nombre de valeurs de la clé.

Comptage de valeurs différentes

COUNT (DISTINCT <colonne>)

renvoie le nombre de valeurs différentes de la colonne.

Exemple :

De combien de pays viennent les acteurs enregistrés dans la table Acteurs ?

```
SELECT COUNT ( DISTINCT Pays )
FROM Acteurs
;
```

Remarque

Il est possible de mettre plusieurs colonnes dans les COUNT

COUNT (DISTINCT Nom, Pays **)**

Division normale par comptage

Table-entité Commerces :

COMMERCES([RCS](1),ville, type)

Requête Q : Quels sont les villes où tous les types de commerces possibles sont implantés ?

En algèbre relationnelle :

COMMERCES[ville, type]
/
COMMERCES[type]

Division normale par comptage

Comme pour la traduction par différence, on découpe en deux requêtes :

Requête Qa : Quel est le nombre total de type de commerces ?

Requête Qb = Q : Quelles sont les villes ayant Qa types de commerces différents ?

Division normale par comptage

Comme pour la traduction par différence, on découpe en deux requêtes :

Requête Qa : Quel est le nombre total de type de commerces ?

Requête Qb = Q : Quelles sont les villes ayant Qa types de commerces différents ?

Division normale par comptage

```
SELECT      Ville    - - clé de groupe
FROM        Commerces
GROUP BY    Ville
HAVING COUNT (DISTINCT Type) = (
            SELECT COUNT (DISTINCT Type)
            FROM Commerces
                                )
;
```

Même exemple par différence

Requête Qc : Quels sont les types de commerces ?

Requête Qd : Quels sont les types de commerces de la ville **V** ?

Requête Qe = Q : Quelles sont les villes **V** telles que Qc - Qd est vide ?

Même exemple par différence

```
SELECT DISTINCT  Ville
FROM Commerces C1
WHERE NOT EXISTS (
                        SELECT  DISTINCT  Type
                        FROM Commerces C2
                        MINUS
                        SELECT  DISTINCT  Type
                        FROM Commerces C3
                        WHERE C3.Ville = C1.Ville
                        )
;
```

Division normale par comptage : exercice

On reprend l'exemple du chapitre précédent :

imatriculation	marque	puissance	Dept d'immat
24ET7898	RENAULT	8	22
76YU9087	PEUGEOT	8	56
75GY6435	AUDI	8	35
67HR4321	PEUGEOT	7	35
46FC5687	RENAULT	7	56
55YT9462	PEUGEOT	9	22

Immat	marque	puiss	Date 1ere imat
34E87	Yamaha	8	17/06/2004
87Y54	Yamaha	7	08/05/2010
98I09	Honda	8	24/07/2009

Quelles sont les marques de voitures représentées par des voitures dont les puissances sont celles de toutes les motos ?

Voitures[marque,puissance]/Motos[puissance]
résultat : Peugeot et Renault

Division normale par comptage : exercice

À VOUS !!

Recherche d'extremum

MAX (<colonne> **)**
renvoie la dernière valeur de la colonne triée

MIN (<colonne> **)**
renvoie la première valeur de la colonne triée

Exemple :

Quel est le salaire minimum des acteurs ?

SELECT MIN (Salaire **)**
FROM Acteurs
;

Erreur classique 1

Quel est l'employé (ou les employés) qui gagne le plus ?

```
SELECT  Nom
FROM Employes
WHERE Salaire = MAX (Salaire)
;
```

Le SGBD n'apprécie pas ...

La bonne réponse

```
SELECT  Nom
FROM Employes
WHERE salaire = (SELECT MAX (Salaire)
                  FROM Employes
                 )
;

-- C'est moi ...
```

Fonctions numériques : somme

SUM (<colonne> **)**
renvoie le total des valeurs d'une colonne numérique

Exemple :

Quel est le total des salaires des employés du Morbihan ?

```
SELECT  SUM (Salaire)
FROM      Employes
WHERE    NoDept = 56
;
```

Fonctions numériques : moyenne

AVG (<colonne> **)**
renvoie la moyenne des valeurs d'une colonne numérique.

Exemple :

Quel est le prix moyen des produits ?

```
SELECT AVG (Prix)
FROM       Produits
;
```

Fonctions numériques : erreur classique 2

Encore une erreur classique : Confondre

COUNT ( ) [nombre de tuples]
On peut compter des cochons, ou n'importe quoi d'autre

et

SUM ( ) [somme des valeurs des tuples]
On ne peut pas faire la somme de deux **cochons**, mais seulement de leurs **poids**

Un piège !

Quels sont les libellés des produits dont le prix est supérieur à la moyenne ?

```
SELECT  DISTINCT LibProduit
FROM      Produits
WHERE  Prix > AVG (Prix)
;
```

C'est faux !!! (voir erreur classique 1)

La bonne réponse

Quels sont les libellés des produits dont le prix est supérieur à la moyenne ?

```
SELECT  DISTINCT LibProduit
FROM     Produits
WHERE  Prix > (SELECT   AVG (Prix)
                FROM  Produits
               )
;
```

REGROUPEMENT

Remarque

L'utilisation des fonctions de regroupement ne se traduit pas non plus habituellement en algèbre relationnelle, sans doute parce que cela n'a pas été jugé utile.

Le regroupement

Problème typique :

Une table d'employés contient les employés de plusieurs services ; comment donner des résultats **par service** ?

Le regroupement

La solution :

Partage de table en sous-tables suivant une colonne :
R \ A

La clause **GROUP BY** placée en **fin de mapping** permet de séparer (virtuellement) la table étudiée en sous-tables suivant les différentes valeurs d'une colonne

Le regroupement

Exemple :

Si la table possède une colonne **NoDept**, alors avec :

GROUP BY NoDept

on sépare la table en sous-tables de telle sorte que dans chaque sous-table, tous les tuples aient le même numéro de département

Le regroupement

Ainsi :

```
SELECT      NoDept
FROM        Employes
GROUP BY    NoDept
;
```

produit exactement le même résultat que :

```
SELECT      DISTINCT    NoDept
FROM        Employes
;
```

Remarque

La colonne de projection est **obligatoirement** la colonne de regroupement.

```
SELECT      Salaire
FROM        Employes
GROUP BY  NoDept
;
```

ERREUR à la ligne 1 :
ORA-00979: N'est pas une expression GROUP BY

Remarque

Même RowNum est interdit !

```
SELECT      Rownum, NoDept
FROM        Employes
GROUP BY  NoDept
;
```

ERREUR à la ligne 1 :
ORA-00979: N'est pas une expression GROUP BY

Remarque

Cependant, et **c'est là l'intérêt du regroupement**, on peut ajouter à la colonne de projection une fonction comme :

COUNT()
SUM()
...

Exemple

Quels sont les salaires cumulés par département ?

```
SELECT      NoDept , SUM ( Salaire )
FROM        Employes
GROUP BY  NoDept
;
```

Exemple

Quels sont les quantités totales commandées par produit et par client ?

```
SELECT   LibProduit , NomClient , SUM (Qte)
FROM     Commandes
GROUP BY  LibProduit , NomClient
;
```

Restriction interne

On peut restreindre les tuples à l'intérieur d'un groupe, c'est-à-dire d'une sous-table par la clause **HAVING.**

à ne pas confondre avec WHERE !

Exemple :

Quels sont les salaires moyens par fonction pour les groupes de plus de deux employés ?

```
SELECT  Fonction , AVG(Salaire)
FROM       Employes
GROUP BY  Fonction
HAVING  COUNT ( * ) > 2
;
```

Remarque

Attention : ne pas confondre
WHERE
qui s'applique **à chaque tuple**, avec

HAVING
qui s'applique **au groupe** de tuples

Exemple

```
SELECT    NoDept ,  COUNT(*)
FROM          Employes
WHERE Fonction IN ('Ingénieur', 'Commercial')
GROUP BY  NoDept
HAVING  COUNT(*)   > = 2
;
```

Remarque

Le test de **HAVING** porte **obligatoirement sur** la colonne de regroupement.

```
SELECT      NoDept
FROM        Employes
GROUP BY    NoDept
HAVING      Salaire > 1200
;
```

ERREUR à la ligne 4 :
ORA-00979: N'est pas une expression GROUP BY

Exemple de mapping correct

Quels sont les depts contenant au moins deux ingénieurs commerciaux ?

```
SELECT     NoDept ,  COUNT(*)
FROM          Employes
WHERE Fonction IN ('Ingénieur', 'Commercial')
GROUP BY  NoDept
HAVING  COUNT(*)   > = 2
;
```

Composition de fonctions

Quels est le nb max d'employés dans un département ?

```
SELECT    MAX ( COUNT(*)  )
FROM          Employes
GROUP BY  NoDept
;
```

et ce résultat peut être réintroduit dans une clause **HAVING**

Le renommage

Quelle est la quantité de produits commandée par produit et par client ?

```
SELECT   SUM ( Qte  ) AS Somme
FROM          Commandes
GROUP BY  LibProduit, NomClient
;
```

Pas d'erreur, mais c'est **nul** !!!
Il faut toujours projeter les colonnes du **GROUP BY** pour que le résultat soit lisible
(ce qui n'est pas fait ici !)

Le renommage

Quelle est la quantité de produits commandée par produit et par client ?

Voici le bon mapping :

```
SELECT   LibProduit, NomClient, SUM ( Qte  ) AS Somme
FROM          Commandes
GROUP BY  LibProduit, NomClient
;
```