

Morgado-Samagaio Jonathan

IBD / TP3 Oracle

Table des matières

Table des matières	2
1. Introduction.....	3
2. Requêtes.....	3
3. Conclusion	7

1. Introduction

Dans ce TP, nous allons à nouveau faire des requêtes mono-table. Nous allons voir les opérateurs d'agrégation qui sont donc GROUP BY, HAVING et ORDER BY.

2. Requêtes

- 1) Nous allons donc effectuer plusieurs requêtes de plus en plus compliquées pour comprendre comment les utiliser. Nous commençons par l'utilisation de COUNT pour compter le nombre d'éléments dans une colonne. On cherche le nombre de continent total. Pour cela, on utilise :

```
SELECT COUNT(nom) as  
FROM Continent;
```

Nous pouvons voir le COUNT qui compte donc le nombre d'éléments dans la colonne nom de

la table Continent et qui le met dans une colonne NDC. Nous obtenons donc :

COUNT (NOM)

- 2) Nous continuons avec COUNT cette fois ci pour utiliser deux colonnes d'une même table. On cherche le nombre de pays par continent. Pour cela, nous interrogeons la table Pays. On sélectionne la colonne nomc et on compte le nombre de pays avec une colonne nomc identique puis on regroupe par cette même colonne.

```
SELECT nomc, COUNT(nom) as  
FROM Pays  
GROUP BY nomc;
```

Nous obtenons donc le résultat suivant :

NOMC

Océanie
Asie
Amérique
Afrique

- 3) Nous allons introduire la commande LIKE qui permet de récupérer toute les cases dont les chaînes de caractères présentent une similitude. Nous allons, ici, récupérer le nombre de pays dans les continents commençant par la lettre A. On réutilise la commande de la question précédente en rajoutant une ligne :

```
SELECT nomc, COUNT(nom) as  
FROM Pays  
WHERE nomc LIKE 'A%'  
GROUP BY nomc;
```

On voit l'apparition du WHERE ... LIKE ... et qui nous renvoie :

NOMC
Asie
Amérique
Afrique

- 4) Nous avons encore une question se basant sur le principe de la question mais qui, cette fois-ci, nous demande d'utiliser MIN et MAX pour obtenir respectivement le minimum et le maximum des valeurs d'une colonne. Nous avons donc :

```
SELECT nomc, COUNT(nom) as NBP, MIN(population) as PMIN, MAX(population) as PMAX
FROM Pays
GROUP BY nomc;
```

Qui nous renvoie :

NOMC	NBP	PMIN
Océanie	16	161
Asie	48	39270
Amérique	35	5382
Afrique	55	9598

- 5) Nous allons maintenant appliquer le principe du COUNT à une autre table. Nous cherchons le nombre de montagne par pays mais nous voulons ajouter un classement par ordre décroissant. Pour cela nous allons utiliser ORDER BY. La table contenant les informations que l'on cherche est la table Localiser. Le résultat étant long, nous allons utiliser une commande vue au TP précédent : PAGESIZE. Nous exécutons donc :

```
SET PAGESIZE 1000;
SELECT nomp as Pays, COUNT(nomp) as NbMontagne
FROM Localiser
GROUP BY nomp
ORDER BY NbMontagne DESC;
```

Nous pouvons donc voir le ORDER BY qui prend comme attribut DESC pour trier par ordre décroissant. Nous obtenons donc :

PAYS
Indonésie
Chine
Suisse
Pakistan
Russie
Népal
Papouasie-Nouvelle-Guinée
Canada
États-Unis
Argentine
Éthiopie
Géorgie
Ouganda
Inde
Italie
Chili
Mexique
France
République démocratique du Congo
Kenya
Pérou
Tanzanie
Maroc
Azerbaïdjan
Rwanda

- 6) Nous appliquons à nouveau le même principe sur une autre table mais en rajoutant une contrainte. Nous cherchons les pays dont le nombre de fleuves qui les traversent est supérieur ou égal à 5. Nous allons donc utiliser HAVING qui nous permet de faire cette contrainte.

```
SELECT nomp, COUNT(nomp)
FROM Traverser
GROUP BY nomp
HAVING COUNT(nomp) >= 5
```

Et nous obtenons :

NOM
Russie
France
Chine
États-Unis
Inde
Espagne
Nouvelle-Zélande
Italie
Brésil
Canada
Argentine
Allemagne
Turquie
Maroc

- 7) Dans cette question nous allons exécuter une commande simple pour nous permettre de faire les prochaines. Nous cherchons donc la population max d'un pays :

```
SELECT MAX(population) as
FROM Pays
```

Et nous obtenons :

PMA
143900000

- 8) Cette question se décompose en deux parties. Nous cherchons en premier le pays le plus peuplé avec le résultat brut obtenu à la question précédente.

```
SELECT nom
FROM Pays
WHERE population = 143900000
```

Nous obtenons donc comme résultat :

NOM
Chine

Maintenant, nous allons faire la même requête mais en imbriquant celle de la question 7.

```
SELECT nom
FROM Pays
WHERE population = (SELECT MAX(population)
FROM Pays)
```

Nous pouvons donc voir qu'il est possible de mettre des commandes à l'intérieur d'autres car on obtient le même résultat qui est la Chine.

- 9) Cette question se déroule aussi en deux parties. Nous cherchons le continent ayant le plus de Pays. Pour cela, nous allons commencer par prendre la valeur brute obtenue à la question 2.

```
SELECT nomc
FROM Pays
GROUP BY nomc
```

Et nous obtenons :

```
NOMC
-----
Asie
```

Même principe pour la seconde partie. On cherche le continent ayant le plus de pays mais cette fois-ci en imbriquant et adaptant la requête.

```
SELECT nomc
FROM Pays
GROUP BY nomc
HAVING COUNT(nom) = (SELECT MAX(COUNT(nom))
                     FROM Pays)
```

Nous pouvons voir que l'imbrication est plus grande et que nous avons utilisé un COUNT dans un MAX car il faut que le résultat de la requête ne corresponde qu'à une seule valeur. Nous obtenons donc encore une fois l'Afrique.

- 10) Nous utilisons le même principe de requête imbriquer que dans les questions précédentes pour chercher le pays le montagneux.

```
SELECT nomp
FROM Localiser
GROUP BY nomp
HAVING COUNT(nomp) = (SELECT MAX(COUNT(nomp))
                     FROM Localiser)
```

Nous obtenons donc :

```
NOMP
-----
Indonésie
```

3. Conclusion

Dans ce TP, nous avons pu voir l'utilisation des nouveaux opérateurs d'agrégation : GROUP BY, HAVING, COUNT, ORDER BY. Nous avons vu leur utilité lors de la recherche d'une information spécifique. Nous avons aussi vu qu'il était possible d'imbriquer des requêtes dans d'autres afin d'éviter de rentrer des valeurs brutes.