Mustapha NEZZARI

Dans ce compte rendu, vous trouverez les réponses aux exercices 1 et 2 du TP1. Chaque réponse est composée de la requête SQL, d’une capture d’écran du résultat et d’une explication de la requête.

Compte rendu du TP1

SQL OLAP

Sommaire

[Exercice 1 4](#_Toc466813385)

[Question a 4](#_Toc466813386)

[Requête 4](#_Toc466813387)

[Résultat 4](#_Toc466813388)

[Explications 4](#_Toc466813389)

[Question b 5](#_Toc466813390)

[Requête 5](#_Toc466813391)

[Résultat 5](#_Toc466813392)

[Explications 5](#_Toc466813393)

[Question c 6](#_Toc466813394)

[Requête 6](#_Toc466813395)

[Résultat 6](#_Toc466813396)

[Explications 6](#_Toc466813397)

[Question d 7](#_Toc466813398)

[Requête 7](#_Toc466813399)

[Résultat 7](#_Toc466813400)

[Explications 7](#_Toc466813401)

[Question e 8](#_Toc466813402)

[Question f 8](#_Toc466813403)

[Requête 8](#_Toc466813404)

[Résultat 8](#_Toc466813405)

[Explications 8](#_Toc466813406)

[Question g 9](#_Toc466813407)

[Requête 9](#_Toc466813408)

[Résultat 9](#_Toc466813409)

[Explications 9](#_Toc466813410)

[Exercice 2 11](#_Toc466813411)

[Question 1 11](#_Toc466813412)

[Requête 11](#_Toc466813413)

[Résultat 11](#_Toc466813414)

[Explications 11](#_Toc466813415)

[Question 2 12](#_Toc466813416)

[Requête 12](#_Toc466813417)

[Résultat 12](#_Toc466813418)

[Explications 12](#_Toc466813419)

[Question 3 13](#_Toc466813420)

[Requête 13](#_Toc466813421)

[Résultat 13](#_Toc466813422)

[Explications 14](#_Toc466813423)

[Question 4 15](#_Toc466813424)

[Requête 15](#_Toc466813425)

[Résultat 15](#_Toc466813426)

[Explications 16](#_Toc466813427)

[Question 5 17](#_Toc466813428)

[Requête 17](#_Toc466813429)

[Résultat 17](#_Toc466813430)

[Explications 17](#_Toc466813431)

[Question 6 18](#_Toc466813432)

[Requête 18](#_Toc466813433)

[Résultat 18](#_Toc466813434)

[Explications 19](#_Toc466813435)

[Question 7 20](#_Toc466813436)

[Requête 20](#_Toc466813437)

[Résultat 20](#_Toc466813438)

[Explications 20](#_Toc466813439)

[Question 8 21](#_Toc466813440)

[Requête 21](#_Toc466813441)

[Résultat 21](#_Toc466813442)

[Explications 22](#_Toc466813443)

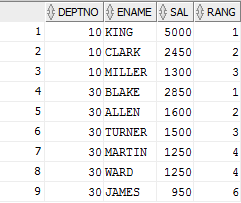
# Exercice 1

## Question a

### Requête



### Résultat



### Explications

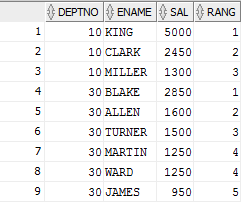
Cette requête classe les salaires des employés pour les départements 10 et 30. Ce classement est réalisé grâce à la fonction fenêtrée Rank () qui utilise la clause partition **BY** pour faire des groupes de classements par département. La clause **ORDER** **BY** sal permet de définir sur quelle colonne le tri est fait dans chaque groupe, ici le salaire par ordre décroissant. On filtre enfin sur les départements 10 et 30 grâce à la clause **WHERE** deptno = 10 **OR** deptno = 30.

## Question b

### Requête



### Résultat



### Explications

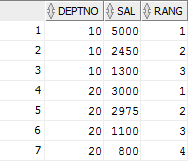
Cette requête effectue la même chose que la requête précédente à la seule différence que les trous sont comblés dans le rang grâce à la fonction fenêtrée Dense\_rank ().

## Question c

### Requête



### Résultat



### Explications

Cette requête classe le salaire des employés par ordre décroissant pour les départements 10 et 20. On utilise la même requête qu’à la question précédente sauf que l’on filtre cette fois sur les départements 10 et 30 et que l’on a une clause **DISTINCT** pour ne pas avoir de doublons au niveau des salaires si par exemple deux employés gagnent le même salaire. Enfin la clause **ORDER** **BY** deptno permet d’ordonner le tout par département.

## Question d

### Requête

#### Avec le Group By



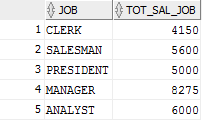
#### Avec le Partition by



#### Avec une sous requête



### Résultat



### Explications

A COMPLETER

## Question e

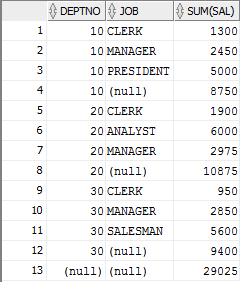
A COMPLETER

## Question f

### Requête



### Résultat



### Explications

Cette requête permet de connaître le montant total des salaires versés :

* Tout département et job confondu (en orange)
* Par département (en vert)
* Par département et job : (en bleu)

Ceci est possible grâce à la clause **GROUP** **BY** rollup ( deptno, job ) qui va permettre à la fonction SUM () de faire la somme sur des granularité de plus en plus petite, c’est-à-dire d’abord sur l’ensemble orange, puis vert puis bleu.

## Question g

### Requête

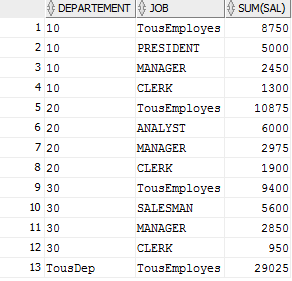
#### Avec NVL ()



#### Avec DECODE ()



### Résultat



### Explications

Les deux requêtes font exactement la même chose qu’à la question précédente si ce n’est que les valeurs (null) sont remplacées par du texte grâce aux fonctions NVL () et DECODE ().

La fonction NVL (texte, defaut) prend en argument :

* texte : Le texte à afficher
* defaut : La valeur par défaut à afficher si le premier argument est (null)

La fonction DECODE (expression, [recherche, resultat]+, defaut) prend en argument :

* expression : L’expression dans laquelle on va rechercher des valeurs
* [recherche, resultat] : Un bloc qui peut se répéter, qui comprend deux arguments :
  + recherche : La valeur à rechercher (et à remplacer par l’argument suivant) dans l’expression
  + resultat : La valeur qui remplacera l’argument précédent s’il est trouvé
* defaut : La valeur par défaut si aucune des expressions du bloc précédent n’est trouvée.

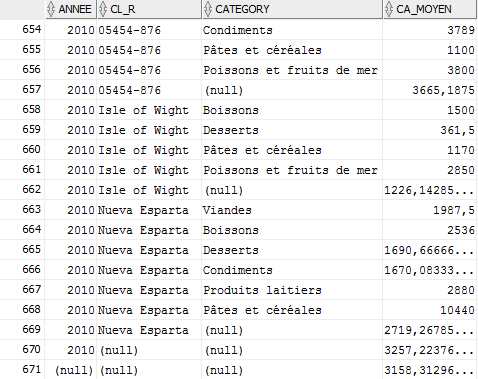
# Exercice 2

## Question 1

### Requête



### Résultat



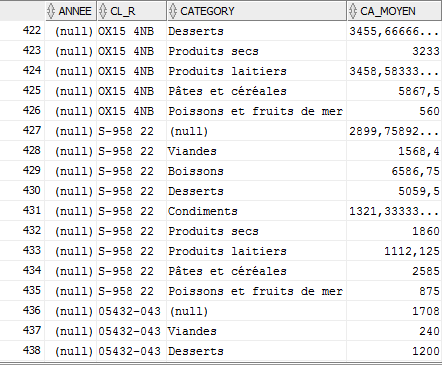
### Explications

## Question 2

### Requête



### Résultat



### Explications

## Question 3

### Requête



### Résultat



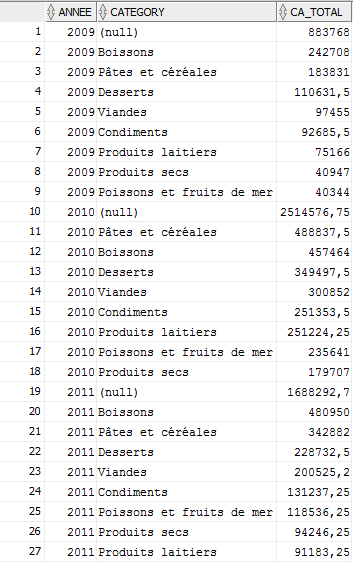
### Explications

## Question 4

### Requête



### Résultat



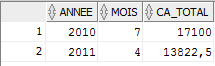
### Explications

## Question 5

### Requête



### Résultat



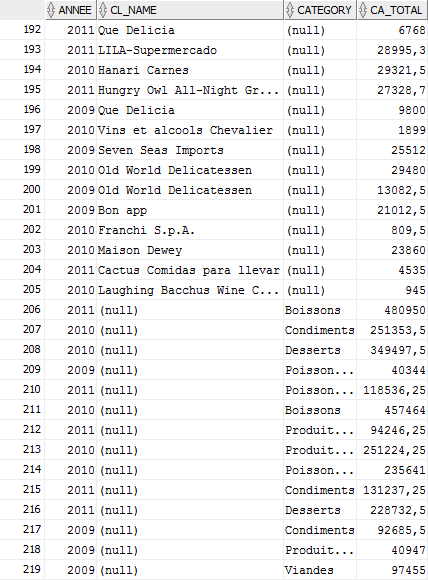
### Explications

## Question 6

### Requête



### Résultat



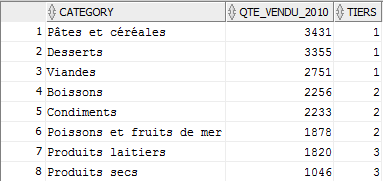
### Explications

## Question 7

### Requête



### Résultat



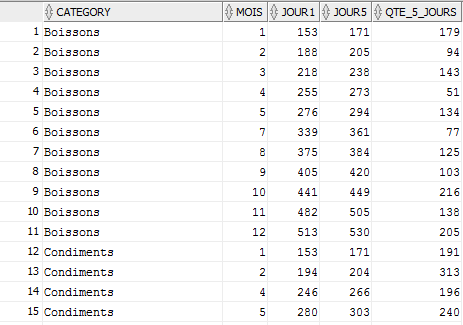
### Explications

## Question 8

### Requête



### Résultat



### Explications