Rapport S1.04 : Partie Manipulation

Fait par : - PIGNEAUX Loris (TD2/TP3)

- DUMAI Etienne (TD2/TP3)

- MARQUES DA SILVA Thomas(TD2/TP3)

BUT Informatique

Semestre 1

Table des matières

[JOIN avec multi jointures : 4](#_Toc156230361)

[**Son utilité :** 4](#_Toc156230362)

[**Exemple sur sql Oracle :** 4](#_Toc156230363)

[JOIN sans multi jointures : 4](#_Toc156230364)

[**Son utilité :** 4](#_Toc156230365)

[**Exemple sur sql Oracle :** 4](#_Toc156230366)

[ORDER BY avec multi tri : 5](#_Toc156230367)

[**Son utilité :** 5](#_Toc156230368)

[**Exemple sur sql Oracle :** 5](#_Toc156230369)

[ORDER BY sans multi tri : 5](#_Toc156230370)

[**Son utilité :** 5](#_Toc156230371)

[**Exemple sur sql Oracle :** 5](#_Toc156230372)

[GROUP BY (1) : 6](#_Toc156230373)

[**Son utilité :** 6](#_Toc156230374)

[**Exemple sur sql Oracle :** 6](#_Toc156230375)

[GROUP BY (2) : 6](#_Toc156230376)

[**Son utilité :** 6](#_Toc156230377)

[**Exemple sur sql Oracle :** 6](#_Toc156230378)

[GROUP BY (3) : 7](#_Toc156230379)

[**Son utilité :** 7](#_Toc156230380)

[**Exemple sur sql Oracle :** 7](#_Toc156230381)

[GROUPE BY HAVING (1) : 7](#_Toc156230382)

[**Son utilité :** 7](#_Toc156230383)

[**Exemple sur sql Oracle :** 7](#_Toc156230384)

[GROUPE BY HAVING (2) : 8](#_Toc156230385)

[**Son utilité :** 8](#_Toc156230386)

[**Exemple sur sql Oracle :** 8](#_Toc156230387)

[GROUPE BY HAVING avec sous requêtes : 8](#_Toc156230388)

[**Son utilité :** 8](#_Toc156230389)

[**Exemple sur sql Oracle :** 8](#_Toc156230390)

[Fonction d’agrégation (1) : 9](#_Toc156230391)

[**Son utilité :** 9](#_Toc156230392)

[**Exemple sur sql Oracle :** 9](#_Toc156230393)

[Fonction d’agrégation (2) : 9](#_Toc156230394)

[**Son utilité :** 9](#_Toc156230395)

[**Exemple sur sql Oracle :** 9](#_Toc156230396)

[Sous requêtes (1) : 10](#_Toc156230397)

[**Son utilité :** 10](#_Toc156230398)

[**Exemple sur sql Oracle :** 10](#_Toc156230399)

[Sous requêtes (2) : 10](#_Toc156230400)

[**Son utilité :** 10](#_Toc156230401)

[**Exemple sur sql Oracle :** 10](#_Toc156230402)

[Sous requêtes (3) : 11](#_Toc156230403)

[**Son utilité :** 11](#_Toc156230404)

[**Exemple sur sql Oracle :** 11](#_Toc156230405)

[Sous requêtes (4) : 11](#_Toc156230406)

[**Son utilité :** 11](#_Toc156230407)

[**Exemple sur sql Oracle :** 11](#_Toc156230408)

JOIN avec multi jointures :

SELECT CAT.libelle AS categorie, F.libelle AS famille, A.code, A.etat

FROM ARTICLE A

JOIN CATEGORIE CAT ON A.codeCat = CAT.code

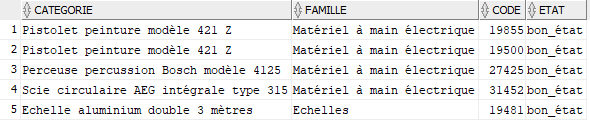
JOIN FAMILLE F ON CAT.codeFam = F.code;

# 

## **Son utilité :**

Connaître les détails des articles, leurs catégories, et la famille à laquelle chaque catégorie appartient.

## **Exemple sur sql Oracle :**



JOIN sans multi jointures :

SELECT E.nom, E.prenom, M.nom AS nom\_magasin

FROM EMPLOYE E

JOIN MAGASIN M ON E.codeMag = M.code;

## **Son utilité :**

Récupérer les noms des employés ainsi que les noms des magasins dans lesquels ils travaillent.

## **Exemple sur sql Oracle :**



ORDER BY avec multi tri :

SELECT E.codeMag, M.nom AS nom\_mag, E.code AS code\_employ, E.nom AS nom\_employee, COUNT(CON.numero) AS num\_contrat

FROM EMPLOYE E

JOIN MAGASIN M ON E.codeMag = M.code

JOIN CONTRAT CON ON E.code = CON.codeEmp

GROUP BY E.codeMag, M.nom, E.code, E.nom

ORDER BY E.codeMag, COUNT(CON.numero) DESC;

## **Son utilité :**

Connaitre le nombre de contrats réalisé par chaque employé de chaque magasin.

## **Exemple sur sql Oracle :**



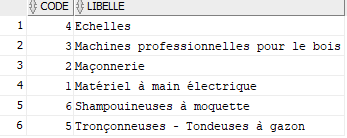
ORDER BY sans multi tri :

SELECT \* FROM FAMILLE ORDER BY libelle;

## **Son utilité :**

Classer les Famille par leur libellé.

## **Exemple sur sql Oracle :**



GROUP BY (1) :

SELECT codeFam, COUNT(\*) as num\_categorie

FROM CATEGORIE

GROUP BY codeFam;

## **Son utilité :**

Compter le nombre de catégorie de chaque famille.

## **Exemple sur sql Oracle :**



GROUP BY (2) :

SELECT codeCli, SUM(montantCheqCaut) as montant\_total\_cheque

FROM CONTRAT

GROUP BY codeCli;

## **Son utilité :**

Calculer la somme totale des chèques de caution pour chaque client dans la table des contrats.

## **Exemple sur sql Oracle :**



## 

GROUP BY (3) :

SELECT codeEmp, COUNT as num\_contrat

FROM CONTRAT

GROUP BY codeEmp;

## **Son utilité :**

Compter le nombre de contrats créés par chaque employé.

## **Exemple sur sql Oracle :**



GROUPE BY HAVING (1) :

SELECT codeCat, COUNT(\*) AS nombre\_articles

FROM ARTICLE

GROUP BY codeCat

HAVING COUNT(\*) > 1;

## **Son utilité :**

Trouver les catégories ayant un nombre d’articles supérieur à une certaine valeur.

## **Exemple sur sql Oracle :**



GROUPE BY HAVING (2) :

SELECT codeFam, AVG(delaiRetour) AS retour\_delai

FROM CATEGORIE

GROUP BY codeFam

HAVING AVG(delaiRetour) > 10;

# **Son utilité :**

Lister les CATEGORIES ayant un délai moyen de retour supérieur à un seuil spécifique.

## **Exemple sur sql Oracle :**



GROUPE BY HAVING avec sous requêtes :

SELECT CAT.code, CAT.libelle, COUNT(ART.code) AS num\_article

FROM CATEGORIE CAT

JOIN ARTICLE ART ON CAT.code = ART.codeCat

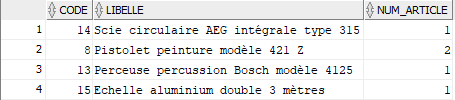
GROUP BY CAT.code, CAT.libelle

HAVING COUNT(ART.code) IN (SELECT nombre\_article FROM (SELECT COUNT(\*) AS nombre\_article FROM ARTICLE GROUP BY codeCat) T);

## **Son utilité :**

Trouver les catégories qui ont un nombre d’articles supérieur à la moyenne.

## **Exemple sur sql Oracle :**



Fonction d’agrégation (1) :

SELECT A.codeCat, COUNT(\*) AS nombre\_articles

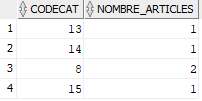
FROM ARTICLE A

GROUP BY A.codeCat;

## **Son utilité :**

Trouver les clients ayant un nombre de contrats qui ne correspond pas au nombre de contrats le moins courant parmi tous les clients:

## **Exemple sur sql Oracle :**



Fonction d’agrégation (2) :

SELECT codeTypeCli, SUM(montantCheqCaut) AS total\_caution, AVG(montantCheqCaut) AS moyenne\_caution

FROM CONTRAT CO

JOIN CLIENT CL

ON CL.CODETYPECLI = CO.CODECLI

GROUP BY codeTypeCli;

## **Son utilité :**

Liste le nombre d’employé(s) qui a fait plus d’un contrat.

## **Exemple sur sql Oracle :**



Sous requêtes (1) :

SELECT C.code, C.nom, C.prenom, COUNT(CON.numero) AS num\_contrat

FROM CLIENT C

JOIN CONTRAT CON ON C.code = CON.codeCli

GROUP BY C.code, C.nom, C.prenom

HAVING COUNT(CON.numero) NOT IN (SELECT COUNT(\*) FROM CONTRAT GROUP BY codeCli HAVING COUNT(\*) < 1);

## **Son utilité :**

Identifier les clients ayant un nombre de contrats qui ne correspond pas au nombre de contrats le moins courant parmi tous les clients.

## **Exemple sur sql Oracle :**



Sous requêtes (2) :

SELECT E.code, E.nom, E.prenom

FROM EMPLOYE E

WHERE E.code IN (

SELECT CON.codeEmp

FROM CONTRAT CON

GROUP BY CON.codeEmp

HAVING COUNT(\*) > 1 );

## **Son utilité :**

Liste le nombre d’employé(s) qui a fait plus d’un contrat.

## **Exemple sur sql Oracle :**



Sous requêtes (3) :

SELECT CL.code, CL.nom, CL.prenom

FROM CLIENT CL

WHERE CL.code NOT IN (

SELECT CON.codeCli

FROM CONTRAT CON

JOIN LIGNECONTRAT LC ON CON.numero = LC.numeroCont

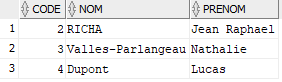
JOIN ARTICLE A ON LC.numeroArticle = A.code

WHERE A.codeCat IN ('13', '15'));

## **Son utilité :**

Lister les clients qui n'ont pas loué d’articles de certaines catégories:

## **Exemple sur sql Oracle :**



Sous requêtes (4) :

SELECT CL.code, CL.nom, CL.prenom

FROM CLIENT CL

WHERE CL.code IN (

SELECT A.codeClient

FROM ABONNEMENT A

WHERE A.codeTypeAbont = 2);

## **Son utilité :**

Lister les clients qui ont un type d’abonnement spécifique :

## **Exemple sur sql Oracle :**

