

420-KHG-LG, Semaine 03- Séance2

Retour sur les requêtes avec jointures

Quelques fonctions SQL :

Les fonctions agissant sur les groupes

Ces fonctions sont utilisées pour traiter des groupes de rangées et d'afficher un seul résultat. Même si ce sont des fonctions de groupement, elles ne s'utilisent pas tout le temps avec la clause GROUP BY.

Les fonctions MIN et MAX: ce sont des fonctions qui s'utilisent pour afficher la valeur MIN (ou MAX) parmi l'ensemble des valeurs de la colonne indiquée.

Exemple

```
SELECT MAX (NOTE) FROM RESULTATS WHERE CODE_COURS ='KED';
```

Les fonctions AVG et SUM

AVG s'utilise pour obtenir une valeur moyenne des valeurs de la colonne indiquée

SUM s'utilise pour obtenir une valeur totale des valeurs de la colonne indiquée

Exemple

```
SELECT AVG (NOTE) FROM RESULTATS WHERE CODE_COURS ='KED';
```

Les fonctions VARIANCE et STDDEV: Pour calculer la variance et l'écart type sur les valeurs d'une colonne

La fonction COUNT: cette fonction permet de compter le nombre de lignes (rangées) qui répondent à un critère. La clause GROUP BY peut être utilisée ou non

Si une colonne est présente dans la clause SELECT alors elle doit être présente dans la clause GROUP BY

La clause GROUP BY: cette clause permet d'indiquer au système de regrouper des enregistrements selon des valeurs distincts qui existent pour les colonnes spécifiées. La clause HAVING permet de mieux cibler les enregistrements spécifiés.

Exemples

```
SELECT CODEPRG, COUNT(CODEPRG)
FROM ETUDIANTS
GROUP BY CODEPRG;
```

CODEPRG	COUNT(CODEPRG)
430	2
420	4
410	3

```
SELECT CODEPRG, COUNT(CODEPRG)
FROM ETUDIANTS
GROUP BY CODEPRG
HAVING CODEPRG = '420';
```

Cette requête calcule le nombre total d'étudiants.

```
SELECT COUNT(*)
FROM ETUDIANTS;
```

Modification de la définition d'une table : La commande ALTER TABLE

Il est parfois nécessaire de modifier la structure d'une table, la commande ALTER TABLE sert à cela. Cette commande change la structure de la table mais pas son contenu.

Les types de modifications acceptées sont les suivants:

- ✓ Ajout d'une nouvelle colonne à la table avec ses contraintes
- ✓ Augmente ou diminue la largeur d'une colonne existante
- ✓ Changer la catégorie d'une colonne, d'obligation à optionnelle ou vice versa (NOT NULL à NULL ou vice versa)
- ✓ Spécification d'une valeur par défaut pour une colonne existante
- ✓ Changer le type de données d'une colonne existante
- ✓ Spécification d'autres contraintes pour une colonne existante
- ✓ Activer ou désactiver une contrainte
- ✓ Détruire une contrainte.

L'option ADD

Cette option permet d'ajouter une colonne ou une contrainte à une table existante.

Attention!!

Si la table contient des valeurs, alors la colonne ajoutée doit être mise à jour.

Si une contrainte est ajoutée à une colonne alors que celle-ci contient déjà des données qui ne correspondent pas à la contrainte, alors la modification de la structure de la table sera refusée.

Exemples :

Voici la commande CREATE initiale pour la table Employes

```
CREATE TABLE Employes(NumEmp number, nom varchar2(15), prenom  
varchar2(20));
```

```
ALTER TABLE Employes ADD (Salaire NUMBER (8,2));
```

Permet d'ajouter la colonne Salaire à la table Employes

```
ALTER TABLE Employes ADD CONSTRAINT emppk PRIMARY KEY  
(NumEmp);
```

Permet d'ajouter une contrainte de clé primaire sur la colonne Numemp de la table Employes

L'option MODIFY:

Cette option permet de modifier le type de données, la valeur par défaut et la contrainte de NOT NULL sur une table déjà existante. Il est impossible de raccourcir la taille d'une colonne (la longueur des données) si celle-ci contient des données.

```
ALTER TABLE Employes MODIFY (nom NOT NULL);
```

L'option ENABLE /DISABLE

Cette option sert à activer ou désactiver une contrainte.

```
ALTER TABLE Employes DISABLE Primary Key;
```

L'option DROP

Cette option sert à supprimer une contrainte sur une table déjà existante. Lorsque

```
ALTER TABLE Employes DROP Primary Key;
```

Ou

```
ALTER TABLE Employes DROP CONSTRAINT emppk;
```

```
ALTER TABLE Employes DROP COLUMN nom;
```

```
ALTER TABLE Employes RENAME COLUMN Salaire TO SalaireEmp;
```

Supprimer une table : la commande <DROP TABLE>

La commande DROP permet de supprimer un objet de la base de données. (Table, indice, synonyme..)

```
DROP TABLE personne;
```

Renommer une table : la commande <RENAME>

Permet de renommer une table ou un objet de la base de données

Syntaxe

```
RENAME <Ancien_nom> TO <Nouveau_nom>;
```

```
RENAME Employes TO EmployesInfo;
```

Exercice 1 : Travailler avec la commande ALTER TABLE

1. Ajouter la colonne CODEEQUIPE de type CHAR(3) à votre table JOUEURS ;
2. Modifier la table pour que le CODEEQUIPE prenne ses valeurs dans (MTL, OTT, TOR, VAN)
3. Faire en sorte que le CODEEQUIPE soit par défaut à MTL.
4. Effectuer une insertion pour vérifier.
5. Mettre à jour les valeurs de cet attribut pour toute la table.
6. Changer le nom de la colonne CODEEQUIPE par CODEVILLE.
7. Écrire la requête qui permet de supprimer la colonne CODEVILLE.
8. Quelle la requête qui vous permet de détruire la contrainte sur le salaire.
9. Quelle est la requête qui supprime la table JOUEURS. Supprimer la table JOUEURS.