

Université A/Mira
Département Mathématiques
Module : BDDA, niveau : Master 1 PSA

Cours SQL (*Structured Query Language*)

Soit 3 tables : Eleves(#NomElv, AdrElv, VilleElv), Matieres(#NomMat, Coef),
 Notes(#NomElv, #NomMat, #Date, Note).

Légende : # : clé primaire

I. L'interrogation simple

1. Liste des élèves.

```
SELECT NomElv (Qu'est ce que je dois afficher ?)
FROM Eleves; (Où cela se trouve t-il ?)
```

2. Liste des matières avec leurs coefficients.

```
SELECT NomMat, Coef
FROM Matieres;
```

- **La close WHERE** : Elle permet de spécifier la ou les conditions que doivent remplir les lignes choisies.

Exemple : Liste des élèves habitant Béjaia.

```
SELECT NomElv
FROM Eleves
WHERE VilleElv = 'Béjaia';
```

Exemple : Liste des matières pour lesquelles l'élève "Abider" a eu au moins une note.

```
SELECT NomMat
FROM Notes
WHERE NomElv = 'Abider';
```

Remarque : Dans la close WHERE, on peut utiliser que des propriétés qui sont dans la table sélectionnée

- **La close GROUP BY** : Il est possible de subdiviser la table en groupes, chaque groupe étant l'ensemble de lignes ayant une valeur commune.

Exemple : Liste des élèves par ville.

```
SELECT NomElv, VilleElv
FROM Eleves
GROUP BY VilleElv;
```

- **La clause HAVING** : Elle ne s'utilise qu'avec le GROUP BY et permet de donner la ou les conditions que doivent remplir ces groupes.

Exemple : Liste des élèves regroupés par ville où habitent plus de 10 élèves.

```
SELECT NomElv, VilleElv
FROM Eleves
GROUP BY VilleElv
HAVING Count(NomElv) > 10;
```

Exemple : Liste des matières où plus de 35 notes ont été données.

```
SELECT NomMat
FROM Notes
GROUP BY NomMat
HAVING Count(Note) > 35;
```

- **La clause ORDER BY** : Elle permet de spécifier l'ordre dans lequel vont être affichées les lignes.

Exemple : Liste des matières dans l'ordre alphabétique.

```
SELECT NomMat
FROM Matieres
ORDER BY NomMat;
```

Liste des matières par ordre décroissant de coef., puis par ordre alphabétique de nom.

```
SELECT NomMat
FROM Matieres
ORDER BY Coef Desc, NomMat Asc;
```

II. Les requêtes multi-tables:

III. Soit 4 tables : Eleves(#RefElv, NomElv, PreElv, VilleElv, ClasseElv),
Classes(#NomCla, Niveau), Cours(#RefElv, #NomMat, NbHeure),
Matieres(#NomMat).

Légende:

: clé primaire

_____ : clé étrangère

Exemple : Liste des élèves (NomElv, PreElv) avec leur niveau.

```
SELECT NomElv, PreElv, Niveau
FROM Eleves, Classes
WHERE Eleves.ClasseElv = Classes.NomCla;
```

Exemple : Liste des élèves et nom des cours qu'ils suivent pendant plus de 3 heures.

```
SELECT NomElv, NomMat
FROM Eleves, Cours
WHERE (Eleves.RefElv = Cours.RefElv) AND (Cours.NbHeure > 3);
```

Remarque : d'abord il faut faire les jointures puis après les sélections

- *Requêtes où les données proviennent d'une table mais où la condition de sélection est faite sur une table différente*

Exemple : Liste des élèves de 1ère.

```
SELECT NomElv, PreElv
FROM Eleves
WHERE ClasseElv IN (SELECT NomCla FROM Classes WHERE Niveau =
                    '1ère');
```

Exemple : Liste des élèves qui font de la Peinture pendant plus de 2 heures.

```
SELECT NomElv, PreElv
FROM Eleves
WHERE RefElv IN (SELECT RefElv FROM Cours WHERE (NomMat =
                    'Peinture') AND (NbHeure > 2));
```

Exemple : Liste des élèves habitant 'Béjaia' et suivant des cours de BDD et de niveau BTS.

```
SELECT NomElv, PreElv
FROM Eleves
WHERE (VilleElv = 'Béjaia')
```

```
AND (RefElv IN (SELECT RefElv FROM Cours WHERE (NomMat = 'BDD'))))
AND (ClasseElv IN (SELECT NomCla FROM Classes WHERE (Niveau =
'BTS'))));
```

III. Insertion, Maj & Suppression dans une table.

Soit une table : Eleves(#NomElv, AdrElv, VilleElv).

Légende: # clé primaire.

- *Insertion de données* : INSERT INTO

Exemple: Insérer l'élève "Mahmoud" qui habite "Taskriout" à "Bejaia".

```
INSERT INTO Eleves (NomElv, AdrElv, VilleElv) VALUES ('mahmoud', 'Taskriout',
'Béjaia');
```

Remarque : Il existe une autre syntaxe qui permet d'insérer dans une table des lignes provenant d'une autre :

```
INSERT INTO nom de la table1 SELECT noms des colonnes à insérer FROM nom de la
table2;
```

- *Mise à jour de données* : UPDATE

Exemple : Modifier le nom de l'élève "Omar" en "Amar".

```
UPDATE Eleves SET NomElv = 'Amar' (Nouvelle valeur)
WHERE NomElv = 'Omar'; (Ancienne valeur)
```

- *Suppression de lignes de tables* : DELETE

Exemple: Supprimer l'élève "Amar".

```
DELETE FROM Eleves
WHERE NomElv = 'Amar';
```

VI. Les fonctions de groupes

Les fonctions de groupes sont :

- Avg (moyenne),
Count(*) (nombre de ligne),
Count(col) (nombre de valeurs non nulles de la colonnes),

Count(Distinct col) (nombre de valeurs non nulles différentes),
 Sum (somme),
 Min,
 Max,
 Variance (variance),
 Stddev (écart type : déviation standard).

Elles peuvent apparaître dans le Select ou le Having.

Une fonction (elle produit une valeur unique) s'applique à l'ensemble des valeurs d'une colonne d'une table (sauf pour la fonction Count).

- **L'AVG** : La commande AVG permet de calculer la moyenne d'un champ

Soit une table Commandes (#NumCmd, DateCmd, Désignation, FraisPort).

Exemple : La moyenne des frais de port pour les commandes dont les frais sont inférieurs 150DA.

```
SELECT AVG(FraisPort)
FROM Commandes
WHERE FraisPort < 150
```

- Le **COUNT**: La commande COUNT permet de compter les lignes.

Soit une table : Eleves(#NomElv, AdrElv, VilleElv).

Exemple: Le nombre d'élèves.

```
SELECT COUNT(NomElv)
FROM Eleves;
```

- Le **SUM**: La commande SUM fait la somme d'un champ.

Soit une table Acomptes (#DateAcpt, #NomOvr, Montant).

Exemple : La somme des acomptes de chaque ouvrier.

```
SELECT NomOvr, SUM(Montant)
FROM Acomptes
GROUP BY NomOvr;
```