

Bases de données

Chapitre 3

Structured Query Language

Manipulation des données

Introduction

Une fois la base créée comment l'interroger ?

Langage SQL

Structured Query Language

- ❑ L4G: Langage de 4e génération
- ❑ Origine : IBM, milieu des années 70
- ❑ plusieurs versions
- ❑ implémenté dans de nombreux SGBD
 - MySQL, Access, Postgre, Oracle, etc.

Caractéristiques de SQL

□ Fonctionnalités :

- Définition des objets de la base de données (LDD)
- **Manipulation de données (LMD)**
- Contrôle des accès aux données

Manipulation des données

Manipuler des données ??

Manipulation des données

□ Travailler sur les données

- Sélectionner
- Insérer
- Mettre à jour
- Supprimer

Requêtes sur une table

Running exemple

Employe(num, nom, age, fonction, sup, datent, salaire, comm, #numDept)

Departement(numDept, nom, noresp, datecre, loc)

Employe

num	nom	age	fonc	sup	datent	salaire	comm	numDept

Departement

numDept	nom	noresp	datecre	loc

Projection

Projection

Structure de base d'une projection

SELECT ... FROM ...

***SELECT <liste_d_attributs_recherchés> ou *
FROM <liste_de_tables>***

- clause **SELECT** pour spécifier le résultat attendu (liste attributs sur lequel on peut effectuer des opérations)
- clause **FROM** pour spécifier les relations à utiliser

Projection

- 1- Afficher le nom des départements et leur lieu d'implantation

Projection

1- Afficher le nom des départements et leur lieu d'implantation

```
SELECT nom , loc  
FROM Departement ;
```

Projection

2- Afficher toutes les informations sur les départements

Projection

2- Afficher toutes les informations sur les départements

```
SELECT *  
FROM Departement ;
```

Elimination des doublons

3- Afficher la fonction de tous les employés

Elimination des doublons

3- Afficher la fonction de tous les employés

```
SELECT fonc  
FROM Employe ;
```

4- Afficher toutes les fonctions assurées
(sans doublons)

Elimination des doublons

3- Afficher la fonction de tous les employés

```
SELECT fonc  
FROM Employe ;
```

4- Afficher toutes les fonctions assurées
(sans doublons)

```
SELECT DISTINCT fonc  
FROM Employe ;
```

Selection

Selection

□ Selection:

- Sélectionner uniquement des enregistrements qui vérifient certaines conditions
- Définir le critère de sélection dans la clause **where**

```
SELECT liste_d_attributs_recherchés ou *  
FROM liste_des_relations  
WHERE condition ;
```

Sélection

5 – Quels sont les employés embauchés dans le département 12

Sélection

5 – Quels sont les employés embauchés dans le département 12

```
SELECT nom  
FROM Employe  
WHERE numDept = 12
```

Opérations

Arithmétiques

$+$, $-$, $*$, $/$
 $+$, $-$

opérateurs unaires

Fonctions numériques

abs , log , cos , sin , mod , power ...

Arithmétiques sur date

$+$, $-$

Fonctions sur chaînes

length , concat , ...

Sélection

6 - Quels sont les employés dont le revenu est supérieur à 2500 euros

(dans la table employé l'attribut comm représente un pourcentage sur le salaire)

Sélection

6 - Quels sont les nom des employés dont le revenu est supérieur à 2500 euros

```
SELECT nom  
FROM Employe  
WHERE salaire + comm*salaire >= 2500;
```


Condition de sélection

Expression logique sur les attributs

On peut également utiliser des connecteurs logiques pour obtenir des expressions booléennes complexes

or and not

Opérateurs spécifiques

7- Quels sont les employés qui gagnent entre 2000 et 3000 euros

Opérateurs spécifiques

7- Quels sont les employés qui gagnent entre 2000 et 3000 euros

SELECT nom

FROM Employe

WHERE (salaire >= 2000 AND salaire <= 3000)

Opérateurs spécifiques

7- Quels sont les employés qui gagnent entre 2000 et 3000 euros

SELECT *

FROM Employe

WHERE (salaire **between 2000 and 3000)**

Opérateurs spécifiques

10- Quels sont les employés dont le nom est soit Dupont, soit Durand, soit Lemaitre ?

Opérateurs spécifiques

10- Quels sont les employés dont le nom est soit Dupont, soit Durand, soit Lemaitre ?

SELECT *

FROM Employe

WHERE

**Nom = 'Dupont' OR nom = 'Durand' OR
nom = 'Lemaitre';**

Comparaison de chaînes

On peut utiliser des expressions régulières. C'est-à-dire chercher à savoir si une chaîne de caractères a une certaine forme.

11- Quels sont les employés dont le nom commence par L

```
SELECT nom  
FROM Employe  
WHERE nom like 'L%' ;
```

Comparaison de chaînes

13- Quels sont les employés pour lesquels l'âge n'a pas été renseigné

```
SELECT *  
FROM Employe  
WHERE age is null ;
```


Tri

Il est possible d'effectuer un tri sur les enregistrements sélectionnés.

order by nom_attribut [ASC] [DESC]

Tri

14 -Donner le nom, le salaire et la fonction des employés du département 12 selon l'ordre croissant des noms

Tri

14 - Donner le nom, le salaire et la fonction des employés du département 12 selon l'ordre croissant des noms

```
SELECT nom, salaire, fonc  
FROM Employe  
WHERE numDept = 12  
ORDER BY nom ASC
```

Requêtes sur plusieurs tables

Jointure

15 - Donner le nom des employés ainsi que la localisation de leur département

■ Difficultés:

- Les informations se trouvent sur deux tables différentes
- Associer les lignes de la première table à celle de la seconde

Jointure

□ Jointure

- Soient deux relations R et S
- Une jointure entre R et S permet d'associer l'ensemble des lignes de R et S satisfaisant une condition donnée
c.-à-d., portant sur un attribut de chacune des relations
- **Exemple: entre Employés et Département**
pour chaque employe on associe son **numDept** à la ligne correspondante dans la table **Departement**

Jointure

□ Deux syntaxes:

■ SQL1:

```
SELECT <attributs>  
FROM table1, table2  
WHERE <critere de jointure>
```

■ SQL2: (a privilégier)

```
SELECT <attributs>  
FROM table1 JOIN table2  
ON <critere de jointure>
```

Jointure

15 - Donner le nom des employés ainsi que leur lieu de travail

```
SELECT nom, loc  
FROM Employe, Departement  
WHERE dept = num;
```

Ou

```
SELECT nom, loc  
FROM Employe JOIN Departement  
ON dept = num;
```


Jointure

16 -Donner le nom des employés et le nom de leur département

Noms absolus - Renommage

16 -Donner le nom des employés et le nom de leur département

```
SELECT Employe.nom, Departement.nom  
FROM Employe JOIN Departement  
ON Employe.dept = Departement.num;
```

Jointure

- 17 - Dans quel département travaille l'employé Toto

Jointure

- ❑ 17 - Dans quel département travaille l'employé Toto

```
SELECT Employe.nom, Departement.nom  
FROM Employe JOIN Departement  
ON Employe.numDept =  
    Departement.numDept  
WHERE Employe.nom='TOTO'
```

Jointure externe

dans la jointure (interne **INNER JOIN**)
les tuples qui ne peuvent pas être joints
sont éliminés du résultat

dans la jointure externe (**OUTER JOIN**)
les tuples qui ne peuvent pas être joints
sont conservés du résultat

Union, Intersection, Différence

- Union -> **UNION**
- Intersection -> **INTERSECT**
- Différence -> **EXCEPT**

Requêtes de regroupement et d'agrégation

```
SELECT liste_d_attributs_recherchés et fonctions  
statistiques  
FROM liste_des_relations  
WHERE condition  
GROUP BY attributs de regroupement  
[HAVING condition sur le groupe ];
```

Fonctions statistiques

`SUM(nom_d_attribut)`

`COUNT(*)`

`COUNT(DISTINCT nom_d_attribut)`

`MAX(nom_d_attribut)`

`MIN(nom_d_attribut)`

`AVG(nom_d_attribut)`

`AVG(DISTINCT nom_d_attribut)`

23 - Quel est le salaire maximal dans chaque département ?

```
SELECT dept, MAX(salaire)  
FROM employe  
GROUP BY dept ;
```

regroupement des employés selon leur département

pour chaque groupe, calcul du salaire maximal et affichage du nom du département et du salaire maximal

24 - Quel est le salaire maximal par département et par profession?

24 - Quel est le salaire maximal par département et par profession?

```
SELECT dept, fonc, MAX(salaire)  
FROM Employe  
GROUP BY dept, fonc ;
```

Attention!

```
SELECT dept, MAX(salaire)  
FROM Employe ;
```

ERREUR

Regroupement et jointure

```
SELECT D.nom, avg(E.salaire)
FROM Employe E, Departement D
WHERE E.dept = D.num
GROUP BY D.num, D.nom ;
```

calcul de la jointure, regroupement par numéro et nom de département

pour chaque groupe, calcul du salaire moyen et affichage du nom du département et du salaire moyen

Pourquoi regrouper sur D.num et D.nom ?

Clause Having

pour ajouter une condition sur les groupes à considérer

25 - Pour chaque département employant au moins 3 personnes, donner son numéro et la moyenne des salaires de ses employés

```
SELECT dept, AVG(salaire)  
FROM employe  
GROUP BY dept  
HAVING COUNT(num) >= 3;
```

26 - Pour chaque département employant au moins 3 personnes qui ne sont ni Directeur, ni Président, donner son numéro et la moyenne des salaires de ses employés, en ne comptabilisant ni les Directeurs, ni les Présidents

26 - Pour chaque département employant au moins 3 personnes qui ne sont ni Directeur, ni Président, donner son numéro et la moyenne des salaires de ses employés, en ne comptabilisant ni les Directeurs, ni les Présidents

```
SELECT dept, AVG(salaire)  
FROM employe  
WHERE fonc NOT IN ("Directeur", "Président")  
GROUP BY dept  
HAVING COUNT(num) >= 3;
```

Requêtes imbriquées

Requêtes imbriquées

SELECT liste_d_attributs_recherchés
FROM liste_des_relations
WHERE bloc dans la condition



**pour indiquer qu'un attribut doit prendre
ses valeurs dans une liste de valeurs définies
par un autre bloc**

Connecteurs entre blocs

= != > < IN EXISTS ANY ALL

EXISTS R retourne TRUE si R n'est pas vide,
FALSE sinon

t IN R retourne TRUE si t appartient à R,
FALSE sinon

Connecteurs entre blocs

val comp **ANY R**

retourne TRUE si la comparaison avec **au moins un** des tuples de **R** (relation unaire) renvoie TRUE

val comp **ALL R**

retourne TRUE si la comparaison avec **tous** les tuples de **R** (relation unaire) renvoie TRUE

Sous-requêtes indépendantes

Le bloc imbriqué peut être évalué séparément du bloc principal

27- Quels sont les noms des employés dont le salaire est supérieur au salaire moyen de l'entreprise ?

```
SELECT E.nom
```

```
FROM Employe E
```

```
WHERE E.salaire > (SELECT AVG(F.salaire)
```

```
FROM Employe F);
```

Sous-requêtes indépendantes

alternative à la jointure

28- Quels sont les noms des employés qui travaillent à Paris ?

```
SELECT E.nom  
FROM Employe E, Departement D  
WHERE E.dept=D.num  
AND D.loc='Paris';
```

Sous-requêtes – opérateur IN

```
SELECT E.nom  
FROM Employe E  
WHERE E.dept IN
```

```
(SELECT D.num  
FROM Departement D  
WHERE D.loc='Paris');
```

29 - Quels sont les départements qui ont un ingénieur ?

29 - Quels sont les départements qui ont un ingénieur ?

```
SELECT D.nom  
FROM Departement D  
WHERE D.num IN (SELECT E.dept  
FROM Employe E  
WHERE E.fonc='Ingenieur');
```

30 - Quels sont les départements qui n'ont pas d'ingénieur ?

```
SELECT D.nom  
FROM Departement D  
WHERE D.num NOT IN (SELECT E.dept  
FROM Employe E  
WHERE E.fonc='Ingenieur');
```

31- Quels sont les employés qui ont le salaire le plus élevé ?

31- Quels sont les employés qui ont le salaire le plus élevé ?

```
SELECT E.nom  
FROM Employe E  
WHERE E.salaire >= ALL (SELECT F.salaire  
FROM Employe F);
```

Sous-requêtes corrélées

Le bloc principal et le bloc imbriqué doivent être évalués simultanément

33 - Quels sont les noms des employés dont le salaire est supérieur au salaire moyen de leur département ?

```
SELECT X.nom  
FROM Employe X  
WHERE X.salaire > (SELECT AVG(Y.salaire)  
FROM Employe Y  
WHERE X.dept=Y.dept);
```

34- Quels sont les départements qui ont un secrétaire ?

34- Quels sont les départements qui ont un secrétaire ?

```
SELECT D.nom  
FROM Departement D  
WHERE EXISTS (SELECT *  
                FROM Employe E  
                WHERE E.fonc='Secrétaire'  
                AND E.dept=D.num);
```

35 - Quels sont départements qui n'ont pas d'ingénieur ?

SELECT D.nom

FROM Departement D

WHERE NOT EXISTS (SELECT *

FROM Employe E

WHERE E.fonc='Ingenieur'

AND E.dept=D.num);

Exercices

Exercices 1/3

Donner le nom des employés qui ont le même salaire que Dupont

Exercices 1/3

Donner le nom des employés qui ont le même salaire que Dupont

```
SELECT Y.nom  
FROM Employe X , Employe Y  
WHERE  
    X.nom='Dupont'  
    and Y.salaire = X.salaire  
    and X.nom != Y.nom;
```

Exercices 2/3

Quels sont les noms et salaires des employés qui gagnent plus que tout employé du département n° 5?

Exercices 2/3

Quels sont les noms et salaires des employés qui gagnent plus que tout employé du département n° 5?

```
SELECT E.nom, E.Salaire  
FROM Employe E  
WHERE E.salaire > ALL(SELECT F.salaire  
FROM Employe F  
WHERE F.dept=5);
```

Exercices 3/3

Quels sont le nom, le salaire, la fonction, le numéro de département des employés travaillant à Toulouse et dans un département ayant plus de 25 employés ?

Exercices 3/3

Quels sont le nom, le salaire, la fonction, le numéro de département des employés travaillant à Toulouse et dans un département ayant plus de 25 employés ?

```
SELECT E.nom, E.salaire, E.fonc, E.dept  
FROM Employe E, Departement D  
WHERE E.dept=D.num and D.loc='Toulouse'  
GROUP BY E.dept, E.nom, E.salaire, E.fonc  
HAVING count(E. num)>=25;
```


Exercices 3/3

```
SELECT E.nom, E.salaire, E.fonc, E.dept  
FROM Employe E  
WHERE E.Dept in (  
    SELECT D.num  
    FROM Departement D, Employe F  
    WHERE D.num=F.dept and  
        D.loc='Toulouse'  
    GROUP BY D.num  
    HAVING count(F. num)>=25);
```