

Introduction à SQL
IUT Lumière, DUT QLIO
2006-2007
Pierre Parrend, Régis Martinez, Soumaya El-Kadiri

SQL: Introduction

I. Le Language SQL

Pour interroger une Base de Données (BD), il faut dialoguer avec elle:

- macros Excel
- requêtes Access
- SQL – Structured Query Language

SQL est le langage le plus complet pour interagir une BD.

Il permet:

- l'interrogation et la manipulation d'une BD
- la création, modification et destruction de tables d'une BD
- la création de Bases de Données

Exemple:

Table Animaux				
ID	Espèce	Race	Naissance	Nom
1	Chien	Yorkshire	21.01.2007	Bidi
2	Chat	Siamois	15.08.2005	Tim
3	Gorille	NULL	30.10.2000	Georges
4	Elephant	D'Asie	23.04.1995	Dumbo
5	Giraphe	NULL	02.07.2003	Helena

II. Tables de Données

A. Ma première requête SQL

```
Select *  
  from Animaux  
  where Espèce = Gorille
```

Explications:

- **Select**: nom des colonnes sélectionnées
exemples: 'Select *'; ou 'Select ID, Espèce, Nom'
- **from**: nom de la table considérée
exemple: 'from Animaux'; ou 'from Animaux, Vaccination'
- **where**: conditions sur la table
exemple: 'where Espèce = Gorille'; ou 'where Naissance > 10.01.2001'

B. Les 4 types de requêtes

- Requêtes d'Interrogation de Données: Select
- Requêtes de Manipulation de Données: INSERT, UPDATE, DELETE
- Requêtes de Contrôle d'Accès aux Données: GRANT, DENY, REVOKE
- Requête de Définition de Données: CREATE, ALTER, DROP, RENAME, TRUNCATE

C. La Sélection

- Sélectionner des colonnes

```
Select ID, Espèce, Nom  
  from Animaux;
```

ID	Espèce	Nom
--	-----	---
1	Chien	Bidi
2	Chat	Tim
3	Gorille	Georges
4	Elephant	Dumbo
5	Giraphe	Helena

- Sélectionner toute une table

```
Select *  
  from Animaux;
```

- Expressions Arithmétiques

Expression sur des données avec + - * /

```
Select Anom, Salaire*12
from Agent;
```

Remarque: la colonne 'Salaire*12' existe uniquement à l'affichage

Renommer les colonnes

```
Select Anom, Salaire*12 AS 'Salaire Annuel'
from Agent;
```

Ordonner les réponses avec ORDER BY

```
Select Anom, Fonction, Salaire
from Agent
ORDER BY Salaire DESC;
```

ASC: classement ascendant

DESC: classement descendant

D. La Comparaison

- Sélectionner des lignes

Par expression de condition

```
Select Anom, Fonction, Salaire
from Agent
where Service_ID = 10;
```

Par comparaison sur les valeurs de champs

```
Select Anom, Fonction, Salaire
from Agent
where Anom = 'DUPONT'
AND Aprenom LIKE '%ea%'
```

- Opérateurs de comparaison

Les opérateurs arithmétique sont:

Les opérateurs de comparaison sont:

Les opérateurs logiques sont:

= > >= < <= != <>

BETWEEN, IN, LIKE, IS NULL

AND, OR, NOT

On utilise l'opérateur BETWEEN pour tester l'appartenance à un intervalle.

```
SELECT   ANom, APrenom, Fonction, Salaire
FROM     Agent
WHERE    DateE   BETWEEN   09.08.2001
                                AND   09.08.2006;
```

On utilise l'opérateur IN pour tester l'appartenance à une liste de valeurs.

```
SELECT   IdService, SNom, Collect_Id
FROM     Service
WHERE    Collect_Id IN  (38,42,69,01);
```

E. Manipulation des Données

- LOWER convertit en minuscule
- UPPER convertit en majuscule
- INITCAP convertit l'initial en capita
- CONCAT concatène des valeurs
- SUBSTR retourne une sous-chaîne
- LENGTH retourne un nombre de caractères
- ROUND(col,n) arrondit une valeur à la décimale spécifiée
- TRUNC (col,n) tronque une valeur à la décimale spécifiée
- MOD(m,n) retourne le reste de la division de m par n

Exemple:

```
SELECT
  UPPER (ANom),  CONCAT (ANom, APrenom), Salaire
FROM
  Agent
WHERE
  LOWER (Fonction) = 'directeur';
```

F. Les Jointures

Objectifs:

- rechercher des données dans plusieurs tables

Moyens:

- les occurrences des différentes tables sont liées entre elles par leurs clés (primaires et étrangères)
- La condition de jointure s'exprime dans la clause 'Where'

```
SELECT   Ag.ANom , Ag.Service_Id , Se.IdService
FROM     Agent Ag, Service Se
WHERE    Ag.Service_Id = Se.IdService ;
```

G. Les Requêtes Complexes

Objectif:

- Utiliser le résultat d'une requête pour en réaliser une deuxième

Moyen:

- On imbrique plusieurs requêtes les unes dans les autres
 - La requête imbriquée (sous-requête, ou sous-interrogation) est réalisée avant la requête principale
 - On peut donc chercher des données à partir de critères inconnus
- Requête imbriquée renvoyant une valeur

```
SELECT  ANom, Fonction
FROM    Agent
WHERE   Service_Id = ( SELECT      Service_Id
                       FROM        Agent
                       WHERE        ANom = 'Dupont' ) ;
```

- Requête imbriquée renvoyant plusieurs valeurs

```
SELECT  ANom, Salaire, Service_Id
FROM    Agent
WHERE   Salaire_Id IN ( SELECT  MIN (Salaire)
                       FROM      Agent
                       GROUP BY   Service_Id );
```

- Requête imbriquée avec comparaison

```
SELECT  Service_Id, AVG(Salaire)
FROM    Agent
GROUP BY Service_Id
HAVING  AVG(Salaire) > ( SELECT  AVG(Salaire)
                          FROM    Salarie
                          WHERE   Service_Id = A10 );
```

H. Mises à Jour

- Insérer des données

Pour insérer une ligne dans la table on utilise la commande SQL INSERT

```
INSERT INTO  Agent (IdAgent, ANom, Aprenom) VALUES (25236,
'Dalton', 'Joe');
```

- Effacer des données

Pour supprimer des données dans une table on utilise la commande SQL DELETE

```
DELETE FROM Agent WHERE IdAgent = 25236 ;
```

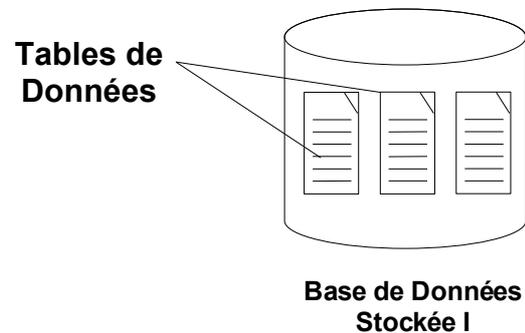
- Modifier des données

Pour modifier des données dans une table on utilise la commande SQL UPDATE

```
UPDATE Agent WHERE Salaire = 2500 ;
```

III. Bases de Données

A. Structure d'une Base de Données



B. Création de base de données

La requête SQL permettant de créer une Base de Données 'business' est la suivante:

```
CREATE DATABASE business;
```

La requête suivante permet de détruire cette Base de Données:

```
DROP DATABASE business;
```

C. Création et destruction de tables de données

La Base de Données 'business' contient une table clients, qui est créée par la requête:

```
CREATE TABLE clients
(
  identifiant int PRIMARY KEY,
  nom varchar[50],
  prenom varchar[50],
  adresse varchar[50]
);
```

La table est détruite comme suit:

```
DROP TABLE clients;
```