

RÉSURGENCES INSERTION



ACCATONE



SQL

Structured Query Language

Introduction



Objectifs de ce cours

Ce cours va vous permettre de comprendre le fonctionnement d'une base de donnée et les principales requêtes de base du langage SQL.

Objectif(s) pédagogique(s) :

- S'initier à la pratique de SQL

Version 22 novembre 2021, Alexis Brou.

Plan du cours

- ❖ Introduction à SQL
- ❖ Rappel et Règle
- ❖ Requête de création de base et de table
- ❖ Modification de table



Introduction à SQL

Le SQL (Structured Query Language) est un langage permettant de communiquer avec une base de données. Ce langage informatique est notamment très utilisé par les développeurs web pour communiquer avec les données d'un site web. Le cours d'aujourd'hui contient les principales commandes pour lire, insérer, modifier et supprimer des données dans une base.

Introduction à SQL

SQL se subdivise en 3 sous-langages :

- LDD (Langage de Définition de Données) : création, modification et suppression des objets que peut manipuler une BD (tables, vues et index, etc).
- LMD (Langage de Manipulation de Données) : ajout, suppression, modification et extraction des données.



Introduction à SQL

- LCD (Langage de Contrôle de Données) : sécurisation et validation des données.

Chacun de ces sous-langages propose ses mots-clés propres.
Voici les principales primitives que nous
verrons au cours de ce chapitre :

Introduction à SQL

| LDD | LMD | LCD |
|-------------------------|--------------------------------------|---------------------------------------|
| CREATE ALTER DROP | SELECT INSERT UPDATE DELETE | GRANT REVOKE COMMIT ROLLBACK |

Rappel et règle

Une table est composée de plusieurs colonnes, un nom de colonne est appelé champ.

Chaque champ doit avoir un type que l'on choisit lors de la création (int, varchar, date ,float ...)

Normalement chaque table doit contenir un champ qui sera défini comme clé primaire(id)



Rappel et règle

Les mots clés doivent être en MAJUSCULES

Les noms de tables/champs en minuscule



Requête de création de base et de table

Commande pour créer une base :

```
CREATE DATABASE mydatabase CHARACTER SET utf8 COLLATE  
utf8_general_ci;
```

Requête de création de base et de table

Commande pour créer une table :

En général :

```
CREATE TABLE <nom_table>  
( <nom_colonne> <type_données> <specifications> [ ,... ],  
PRIMARY KEY(id)  
| CONSTRAINT <nom de la contrainte> FOREIGN KEY(clé étranger de la table)  
REFERENCES <nom table_référencé>(nom colonne référencé)  
,[ , ...])ENGINE=InnoDB|MyISAM DEFAULT CHARSET=utf8
```

1 – Définition des attributs + contraintes sur ces colonnes
(specifications)

2 – Contraintes nommées sur la table

Requête de création de base et de table

Pour les colonnes :

- le nom suit les règles des identificateurs
- type_données : un type du SGBD
- Spécifications possible :

- ❖ NOT NULL
- ❖ DEFAULT(<EXP>)
- ❖ UNSIGNED
- ❖ AUTO_INCREMENT POUR LES CLÉ PRIMAIRE QUI SONT DES ENTIERS



Requête de création de base et de table

Contrainte clé primaire

– PRIMARY KEY : spécifie que cette colonne est la clé primaire. Si plusieurs colonnes forment la clé primaire : contrainte de table.

Requête de création de base et de table

Le type des attributs est à choisir parmi un ensemble fini de types. Les principaux types possibles sont

selon les standards:

- VARCHAR(n) : chaîne de caractères de longueur variable (maximum n)
- CHAR(n) : chaîne de caractères de longueur fixe (n caractères)
- INT : entier
- FLOAT : réel
- DATE : date
- DECIMAL : Nombre décimal
- ...

Requête de création de base et de table

Contraintes de table : la contrainte de clé étrangère :

```
CONSTRAINT <nom_contrainte> FOREIGN KEY ( <clé étrangère de la table> [,  
...]
```

```
) REFERENCES <nom_table_référencée> ( <nom_colonne_référencée> [,...] )
```

```
[ ON DELETE { CASCADE | SET NULL | SET DEFAULT | NO ACTION |  
RESTRICT} ]
```

```
[ ON UPDATE { CASCADE | SET NULL | NO ACTION | RESTRICT} ]
```

Requête de création de base et de table

Lors de la définition d'une contrainte de clé étrangère on peut spécifier ce que l'on fait quand les colonnes de la table référencée sont supprimées (ON DELETE)

- Supposons que la table T a une clé étrangère portant sur la table T' :
 - CASCADE : mise à jour et suppression de T' répercutés sur T
 - NO ACTION : la table n'est pas modifiée (action par défaut).

Requête de création de base et de table

Exemple:

```
DROP TABLE IF EXISTS `employe`;
CREATE TABLE IF NOT EXISTS `employe` (
  `idEmploye` int(10) UNSIGNED NOT NULL AUTO_INCREMENT,
  `nom` varchar(50) NOT NULL,
  `prenom` varchar(25) NOT NULL,
  `sexe` char(1) NOT NULL,
  `service` varchar(25) NULL,
  `salaire` decimal(6,2) UNSIGNED NOT NULL,
  `dateContrat` date NOT NULL,
  PRIMARY KEY (`idEmploye`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

Modification de table

La modification d'une table se fait grâce au mot-clé ALTER TABLE.

Les principales modifications sur la structure d'une table sont :

- ❖ l'ajout d'un attribut
- ❖ la modification d'un attribut
- ❖ la suppression d'un attribut

Cependant, la modification de schéma est une opération délicate qui peut mettre en péril l'intégrité de la base de données. Aussi toutes les opérations de suppression ou de modification de relation, d'attribut ou de contrainte doivent être faites avec précaution.

Modification de table

Modification nom colonne

```
ALTER TABLE matable CHANGE colonne1 nouveaunomcolonne varchar(25) CHARACTER SET utf8  
NOT NULL
```

Ajout autre champ

```
ALTER TABLE matable ADD Column1 varchar(100) NOT NULL;
```

Supression colonne

```
ALTER TABLE employe DROP COLUMN test
```

Modification de table

Pour supprimer une table “nom_table” il suffit simplement d'utiliser la syntaxe suivante :

```
DROP TABLE nom_table
```

A savoir : s'il y a une dépendance avec une autre table, il est recommandé de les supprimer avant de supprimer la table. C'est le cas par exemple s'il y a des clés étrangères.

LMD

Le Langage de Manipulation de Données permet de :

- afficher ;
- insérer ;
- mettre à jour ;
- supprimer des données dans les tables.

Le LMD est donc composé de 4 primitives :

- SELECT pour l'affichage des données des tables.
- INSERT pour l'insertion de données dans les tables.
- UPDATE pour la mise à jour des données.
- DELETE pour la suppression des données.

Alimenter la base de données avec INSERT

L'opération d'insertion se fait grâce au mot-clé INSERT.

```
INSERT INTO <nom_table> [(<nom_col1>,  
<nom_col2>, ...)] VALUES (<val1>, <val2>, ...);
```

- Respect des types.
- Noms de colonnes à spécifier : dans l'ordre de définition de la table
- Si une colonne n'est pas énoncée, sa valeur par défaut lui est attribuée (NULL si non spécifiée).
- Les données qui violent des contraintes ne sont pas insérées.

Alimenter la base de données avec INSERT

Exemple :

```
.INSERT INTO `employe` (`nom`, `prenom`, `sexe`, `service`, `salaire`,  
`dateContrat`) VALUES  
( 'Dupont', 'Robert', 'M', 'Marketing', 20000, '2010-03-21');
```

```
INSERT INTO `employe` (`idEmploye`, `nom`, `prenom`, `sexe`, `service`,  
`salaire`, `dateContrat`) VALUES  
( 'Dupont', 'Aline', 'F', 'Communication', 8500, '2013-11-11'),  
( 'Durand', 'Laurence', 'F', 'Marketing', 14000, '1996-01-02'),  
( 'Lejeune', 'Sylvie', 'F', 'Marketing', 21500, '2018-09-02'),  
( 'Lefort', 'Max', 'M', NULL, 12000, '2005-09-11');
```



Liste Mots Clés

.SELECT .INSERT INTO

.WHERE .DISTINCT

.ORDER BY .UPDATE

.LIMIT .DELETE

.GROUP BY

.HAVING



UPDATE DELETE

UPDATE

La commande UPDATE permet de mettre à jour un ou plusieurs tuples d'une relation.

UPDATE DELETE

Exemple

-- Exemple avec une unique colonne:

```
UPDATE employe
```

```
SET salaire = salaire * 1.1
```

```
WHERE idEmploye = 6;
```

-- Exemple avec plusieurs colonnes:

```
UPDATE employe
```

```
SET salaire = 15000, idService = 3, sexe = 'M'
```

```
WHERE idEmploye = 7;
```

UPDATE DELETE

-- Exemple avec une suppression d'un employé

```
DELETE FROM employe
```

```
WHERE id = 6;
```

-- Suppression de toute les donnée de la table

```
DELETE FROM employe
```

```
TRUNCATE TABLE employe
```

TRUNCATE va réinitialiser l'auto-incrémente s'il y en a un.