Afpa

SQL SERVER - PRÉSENTATION DU DDL

Appui technique

Introduction

DDL

Le **DDL** (Data Definition Language) est un sous-ensemble de SQL qui va nous permettre comme son nom l'indique de définir la structure de la base de données.

Nous allons ici découvrir les principales caractéristiques de ce langage.

CREATE

L'instruction CREATE permet comme son nom l'indique de créer des éléments dans la base de données.

Database

CREATE DATABASE

La création d'une base de données en SQL est possible en ligne de commande.

Même si les systèmes de gestion de base de données (SGBD) sont souvent utilisés pour créer une base, il convient de connaître la commande à utiliser, qui est très simple.

Voici un exemple :

```
CREATE DATABASE MaDB
```

Structure :

```
CREATE DATABASE nomBaseDeDonnees
```

GO

GO n'est pas une commande SQL. C'est une commande de l'interpréteur qu'est SSMS ou SQLcmd.

Cette commande permet de forcer l'envoi du lot de commande SQL et d'attendre le retour du serveur avant de poursuivre.

Elle est indispensable dans certains cas de figure, car le serveur ne peut pas toujours comprendre ce que l'on veut faire d'un point de vue analyse syntaxique ou objet.

On ajoutera systématique GO après la création de la base de données pour que le reste du script puisse s'exécuter correctement

```
CREATE DATABASE MaDB
Go
USE MaDB
```

USE

L'instruction USE permet de préciser que la suite du script s'exécutera sur cette base de données.

Table

CREATE TABLE

La commande CREATE TABLE permet de créer une table en SQL.

Une table est une entité qui est contenu dans une base de données pour stocker des données ordonnées dans des colonnes.

La création d'une table sert à définir les colonnes et le type de données qui seront contenus dans chacun des colonne (entier, chaîne de caractères, date, valeur binaire ...).

Voici un exemple :

```
CREATE TABLE Voitures (
IdVoiture INT PRIMARY KEY IDENTITY(1,1) ,
Immatriculation VARCHAR(20) NOT NULL,
Couleur VARCHAR(20),
Modele VARCHAR(20),
)
```

Structure :

```
CREATE TABLE nomDeLaTable (

nomColonne typeDeDonnées ATTRIBUTS ,

nomColonne typeDeDonnées ATTRIBUTS ,

nomColonne typeDeDonnées ATTRIBUTS ,

...

)
```

nomDeLaTable

le nom de la table peut être un mot simple ou composé (de préférence éviter les espaces en utilisant le snake_case ou le CamelCase)

Il peut être préfixé du nom du schéma, lui-même peut être préfixé du nom de l abse

Colonnes

le nom de la colonne peut être un mot simple ou composé (de préférence éviter les espaces en utilisant le snake_case ou le CamelCase)

Le type de données : voici quelques exemples des types les plus utilisés : int, decimal, float, money, varchar(longueur), text(longueur), date, datetime, ...

les attributs :

- PRIMARY KEY : pour définir la clé primaire

  IDENTITY(départ,pas) : la base de données incrémente automatique le curseur du "pas" à chaque nouvel enregistrement

  NULL | NOT NULL : Autorise ou pas la valeur null dans la colonne. Par défaut l'attribut null est attribué)

UNIQUE : toutes les valeurs de la colonne doivent être différente (exemple l'adresse mail si elle sert d'identifiant)

...

Pour aller plus loin

Documentation Officielle sur le CREATE TABLE

ALTER

La commande ALTER TABLE en SQL permet de modifier une table existante. Idéal pour ajouter une colonne, supprimer une colonne ou modifier une colonne existante, par exemple pour changer le type.

Ajout d'une colonne

L'ajout d'une colonne dans une table est relativement simple et peut s'effectuer à l'aide d'une requête ressemblant à ceci:

Voici un exemple :

```
ALTER TABLE Voitures
ADD Energie VARCHAR(20)
```

Structure :

```
ALTER TABLE nomDeLaTable
ADD nomColonne typeDeDonnées ATTRIBUTS
```

Modification d'une colonne

La modification d'une colonne (type ou attribut) dans une table peut s'effectuer à l'aide d'une requête ressemblant à ceci:

Voici un exemple :

```
ALTER TABLE Voitures
ALTER COLUMN Immatriculation INT NULL
```

Structure :

```
ALTER TABLE nomDeLaTable
ALTER COLUMN nomColonne typeDeDonnées ATTRIBUTS
```

Suppression d'une colonne

La suppression de colonne dans une table peut s'effectuer à l'aide d'une requête ressemblant à ceci:

Voici un exemple :

```
ALTER TABLE Voitures
DROP Energie

ou

ALTER TABLE Voitures
DROP COLUMN Energie
```

Structure :

```
ALTER TABLE nomDeLaTable

DROP nomColonne

Ou

ALTER TABLE nomDeLaTable

DROP COLUMN nomColonne
```

Ajout d'une contrainte

L'ajout d'une contrainte sur une table peut être fait lors de la déclaration de la table ou après la création de toutes les tables.

Voici l'instruction pour créer une contrainte entre 2 tables à posteriori.

Voici un exemple :

```
CREATE TABLE Marques
(IdMarque INT PRIMARY KEY IDENTITY(1,1),
libelleMarque VARCHAR(20) NOT NULL
)

ALTER TABLE Voitures
ADD MarqueId int

ALTER TABLE Voitures
   ADD CONSTRAINT FK_Voitures_Marques FOREIGN KEY (MarqueId)
      REFERENCES Marques (IdMarque)
      ON DELETE CASCADE
      ON UPDATE CASCADE
```

Structure :

```
ALTER TABLE nomDeLaTable
  ADD CONSTRAINT nomDeLaContrainte FOREIGN KEY (nomCleSecondaire)
    REFERENCES Marques (nomClePrimaire)
    ON DELETE methode
    ON UPDATE methode
```

Le nom de la contrainte n'a pas d'impact sur le reste

La méthode détermine ce qui sera fait lors du delete ou lors de l'update.

Les valeurs peuvent être :

- Cascade : la modification est propagée
- No Action : rien n'est propagé. Un rollback peut intervenir s'il existe des enregistrements qui empêche cette instruction.
- Set Null : les clés étrangères sont remplacées par null si la valeur null est autorisée
- Set Default : les clés étrangères sont remplacées par la valeur par défaut si elle a été définie

DROP

L'instruction DROP permet de supprimer des éléments dans la base de données voire la base elle-même.

Database

DROP DATABASE

La suppression d'une base de données en SQL est possible en ligne de commande.

Voici un exemple :

```
DROP DATABASE MaDB
```

Structure :

```
DROP DATABASE nomBaseDeDonnees
```

Table

DROP TABLE

La commande DROP TABLE permet de supprimer une table.

Voici un exemple :

```
DROP TABLE Voitures
```

Structure :

```
DROP TABLE nomDeLaTable
```

TRUNCATE

L'instruction TRUNCATE permet de supprimer tous les enregistrements d'une table sans supprimer la table en elle-même

Table

TRUNCATE TABLE

La commande TRUNCATE TABLE permet de vider une table.

Voici un exemple :

```
TRUNCATE TABLE Voitures
```

Structure :

```
TRUNCATE TABLE nomDeLaTable
```

TRUNCATE vs DELETE

L'instruction TRUNCATE est semblable à l'instruction DELETE sans utilisation de WHERE.

Parmi les petite différences TRUNCATE est toutefois plus rapide et utilise moins de ressource.

La commande TRUNCATE va ré-initialiser la valeur de l'auto-incrément, s'il y en a un.

Ces gains en performance se justifie notamment parce que la requête n'indiquera pas le nombre d'enregistrement supprimés et qu'il n'y aura pas d'enregistrement des modifications dans le journal.

Essentiel à retenir

L'instruction TRUNCATE ne s'exécute que si les contraintes d'intégrité sont respectées

Crédits

OEUVRE COLLECTIVE DE L'AFPA

Sous le pilotage de la Direction de l'ingénierie

DATE DE MISE À JOUR

07/09/2022

© AFPA

Reproduction interdite

Article L 122-4 du code de la propriété intellectuelle.

« Toute représentation ou reproduction intégrale ou partielle faite sans le consentement de l'auteur ou de ses ayants droit ou ayants cause est illicite. Il en est de même pour la traduction, l'adaptation ou la reproduction par un art ou un procédé quelconques. »