99 Liste des commandes LDD (Terminée)

vendredi 26 novembre 2021

14:20

|  |  |  |
| --- | --- | --- |
| **Besoin** | **SQL Server** | **MySql** |
| Création d'un BDD | CREATE DATABASE testDb; | CREATE DATABASE testDB;  Ou  CREATE SCHEMA `testdb` ; |
| Création d'une BDD si elle n'existe pas | IF DB\_ID('testDB') IS NULL  BEGIN  CREATE DATABASE testDB;  END | CREATE DATABASE IF NOT EXISTS testDB;  Ou  CREATE SCHEMA IF NOT EXISTS testDB; |
| Suppression d'une BDD | DROP DATABASE testDb; | DROP DATABASE testDb;  Ou  DROP SCHEMA testDb; |
| Suppression d'une BDD si elle existe | IF DB\_ID('testDB') IS NOT NULL  BEGIN  DROP DATABASE testDB;  END | DROP DATABASE IF EXISTS testDB;  Ou  DROP SCHEMA IF EXISTS `testdb` ; |
| Sélectionner une BDD en particulier pour travailler dessus | Use testDb;  Ou  Use [testDb]; | Use testDb;  Ou  Use **`**testDB**`**; |
| Création d'une table "simple" (sans clé primaire) | Use testDb;    CREATE TABLE Personnes (  PersonID int,  Nom varchar(255),  Prenom varchar(255),  Adresse varchar(255),  Ville varchar(255)  ); | Use testDb;    CREATE TABLE Personnes (  PersonID int,  Nom varchar(255),  Prenom varchar(255),  Adresse varchar(255),  Ville varchar(255)  ); |
| Supprimer une table si elle existe | Use testDb;  DROP TABLE IF EXISTS Personnes ; | Use testDb;  DROP TABLE IF EXISTS Personnes ; |
|  |  |  |
| Création d'une table avec une clé primaire "simple" (1 seule colonne) | Use testDb;  DROP TABLE IF EXISTS Personnes ;    CREATE TABLE Personnes (  PersonID int,  Nom varchar(255),  Prenom varchar(255),  Adresse varchar(255),  Ville varchar(255),  CONSTRAINT PK\_Person PRIMARY KEY (PersonID)  );    Quand on essaie d'ajouter une nouvelle personne avec un ID existant    、 i' 「 OSC コ 5 ロ L ~ Ma 9 ル ョ 当 【 ud を  ( 01 蓄 ー ! 「 き ョ ・ : ー - ー ) 第 2 ↓ ・ : を E 「 0 第 当 コ ー  ・ 降 0 著 一 言 ミ ま ・ . 《 第 き 「 象 dupbqu& ニ ド  ・ ー P ・ 0 コ ・ - - ョ 物 0 ー を 無 、 窰 ま を 、 雪 00- を - ・ d 鴛 「 ( 0 ・ 1  ′ - ー ・ 、 、 0 「 き 二 、 6 新 第 当 2 。 1 ~ 当 - ・ PRIMARY K 「 、  を 2 第 。 第 ・ ヨ ー 新 す ト | Use testDb;  DROP TABLE IF EXISTS Personnes ;    CREATE TABLE Personnes (  PersonID int,  Nom varchar(255),  Prenom varchar(255),  Adresse varchar(255),  Ville varchar(255),  PRIMARY KEY (PersonID)  );    Quand on essaie d'ajouter une nouvelle personne avec un ID existant    Saet  SQL «rg»t to the  Show the  to  *hie  [WERT INTO • t%tdb , ,  re, FOG ,  ERROR 1%2: f fu  [WERT INTO t%tdb , ,  FOG , |
| Création d'une table avec une clé primaire "composée" (plusieurs colonnes) | Use testDb;  DROP TABLE IF EXISTS Personnes ;    CREATE TABLE Personnes (  PersonID int,  Nom varchar(255),  Prenom varchar(255),  Adresse varchar(255),  Ville varchar(255),  CONSTRAINT PK\_Person PRIMARY KEY (Nom , Prenom )  );      Quand on essaie d'ajouter une nouvelle personne avec un NOM et PRENOM déjà existant    0 | Use testDb;  DROP TABLE IF EXISTS Personnes ;    CREATE TABLE Personnes (  PersonID int,  Nom varchar(255),  Prenom varchar(255),  Adresse varchar(255),  Ville varchar(255),  PRIMARY KEY (Nom ,Prenom )  );      Quand on essaie d'ajouter une nouvelle personne avec un NOM et PRENOM déjà existant    A - g SQL t to thedatabase  囟 「 2 5 ter , |
| Ajout d'une clé primaire sur une table sans clé primaire | Use testDb;  DROP TABLE IF EXISTS Personnes ;    CREATE TABLE Personnes (  PersonID int NOT NULL,  Nom varchar(255),  Prenom varchar(255),  Adresse varchar(255),  Ville varchar(255)  );    ALTER TABLE Personnes  ADD CONSTRAINT PK\_Person PRIMARY KEY(PersonID);    **Remarque : La colonne utilisée pour la clé primaire ne doit pas accepter les valeurs NULL**    Quand on essaie d'ajouter une nouvelle personne avec un ID existant  0 | Use testDb;  DROP TABLE IF EXISTS Personnes ;    CREATE TABLE Personnes (  PersonID int,  Nom varchar(255),  Prenom varchar(255),  Adresse varchar(255),  Ville varchar(255)  );    ALTER TABLE Personnes  ADD PRIMARY KEY(PersonID);    Quand on essaie d'ajouter une nouvelle personne avec un ID existant  Texte de remplacement généré par une machine : |
|  |  |  |
| Ne pas autoriser les valeurs NULL sur une colonne d'une table | Use testDb;  DROP TABLE IF EXISTS Personnes ;    CREATE TABLE Personnes (  PersonID int NOT NULL,  Nom varchar(255),  Prenom varchar(255),  Adresse varchar(255),  Ville varchar(255)  );      Quand on essaie d'ajouter une nouvelle personne avec un ID non renseigné  Texte de remplacement généré par une machine : | Use testDb;  DROP TABLE IF EXISTS Personnes ;    CREATE TABLE Personnes (  PersonID int NOT NULL,  Nom varchar(255),  Prenom varchar(255),  Adresse varchar(255),  Ville varchar(255)  );      Remarque : MySql ne fera pas d'erreur mais mettra une valeur par défaut (0 pour un entier) |
| Faire en sorte que toutes les valeurs d'une colonne soient différentes (sur une colonne non clé primaire) | Use testDb;  DROP TABLE IF EXISTS Personnes ;    CREATE TABLE Personnes (  PersonID int NOT NULL,  Nom varchar(255),  Prenom varchar(255),  Adresse varchar(255),  Ville varchar(255),  CONSTRAINT UC\_Person UNIQUE (Nom)  );      Quand on essaie d'ajouter une nouvelle personne avec un Nom existant  Texte de remplacement généré par une machine : | Use testDb;  DROP TABLE IF EXISTS Personnes ;    CREATE TABLE Personnes (  PersonID int NOT NULL,  Nom varchar(255),  Prenom varchar(255),  Adresse varchar(255),  Ville varchar(255),  UNIQUE (Nom)  );      Quand on essaie d'ajouter une nouvelle personne avec un Nom existant  Texte de remplacement généré par une machine : |
| Faire en sorte que toutes les valeurs d'une colonne respectent une condition de validité | Use testDb;  DROP TABLE IF EXISTS Personnes ;    CREATE TABLE Personnes (  PersonID int NOT NULL,  Nom varchar(255),  Prenom varchar(255),  Adresse varchar(255),  Ville varchar(255),  CONSTRAINT CHK\_Person CHECK (PersonID > 0)  );    Quand on essaie d'ajouter une nouvelle personne avec un ID inférieur à 0  0 | Use testDb;  DROP TABLE IF EXISTS Personnes ;    CREATE TABLE Personnes (  PersonID int NOT NULL,  Nom varchar(255),  Prenom varchar(255),  Adresse varchar(255),  Ville varchar(255),  CHECK (PersonID > 0)  );    Quand on essaie d'ajouter une nouvelle personne avec un ID inférieur à 0  Texte de remplacement généré par une machine : |
| Faire en sorte qu'une valeur soit automatiquement affectée à une colonne lors d'un ajout si la valeur n'a pas été précisée par l'utilisateur | Use testDb;  DROP TABLE IF EXISTS Personnes ;    CREATE TABLE Personnes (  PersonID int NOT NULL,  Nom varchar(255),  Prenom varchar(255),  Adresse varchar(255),  Ville varchar(255) DEFAULT 'Ville inconnue'  );    Avant l'insertion  30 0  Après l'insertion | Use testDb;  DROP TABLE IF EXISTS Personnes ;    CREATE TABLE Personnes (  PersonID int NOT NULL,  Nom varchar(255),  Prenom varchar(255),  Adresse varchar(255),  Ville varchar(255) DEFAULT 'Ville inconnue'  );    Avant l'insertion  01u0  Après l'insertion  麡 8u2 01u8 |
| Lors des ajouts faire en sorte que les valeurs d'une colonne soient automatiquement alimentées de façon séquentielle par la BDD | Use testDb;  DROP TABLE IF EXISTS Personnes ;    CREATE TABLE Personnes (  PersonID int IDENTITY(1,1) PRIMARY KEY,  Nom varchar(255),  Prenom varchar(255),  Adresse varchar(255),  Ville varchar(255)  );    Avant l'insertion    Après l'insertion | Use testDb;  DROP TABLE IF EXISTS Personnes ;    CREATE TABLE Personnes (  PersonID int AUTO\_INCREMENT,  Nom varchar(255),  Prenom varchar(255),  Adresse varchar(255),  Ville varchar(255),  PRIMARY KEY (PersonID)  );    Avant l'insertion  Adresse  RT  Après l'insertion |
|  |  |  |
| Faire en sorte que les valeurs d'une colonne soient obligatoirement connues comme clé primaire dans une autre table | Use testDb;  -- Attention à l'ordre des DROP  DROP TABLE IF EXISTS Personnes ;  DROP TABLE IF EXISTS Ville ;    CREATE TABLE Ville (  CityID int IDENTITY(1,1) PRIMARY KEY,  Nom\_ville varchar(255)  );    CREATE TABLE Personnes (  PersonID int IDENTITY(1,1) PRIMARY KEY,  Nom varchar(255),  Prenom varchar(255),  Adresse varchar(255),  Ville\_Id int FOREIGN KEY (Ville\_Id) REFERENCES Ville(CityID)  );    La table VILLE  Texte de remplacement généré par une machine : City1D  Nom_ville  Blois  Dijon  Tentative d'ajout d'une personne dans une ville inconnue  Texte de remplacement généré par une machine : RrsonlD  NUU  Ville_ld  CV  O  RG  Message d'erreur  Texte de remplacement généré par une machine : Microsoft SQL Server Management Studio  Aucune ligne n'a été mise à jour.  Les données de la ligne I n'ont pas été validées.  Source d'erreur : .Net *'Client Data Provider.  Message d'erreur : INSERT est en confit la contrainte  FOREIGN KEY • 061 • . Le confit s'est  produit dans la base de données •testDb", table "dbo.VlIIe•, column  CitylD'.  L'instruction a été arrêtée.  Corrigez les erreurs et réessayez Ou appuyez Sur ESC annuler les  modifications. | Use testDb;  -- Attention à l'ordre des DROP  DROP TABLE IF EXISTS Personnes ;  DROP TABLE IF EXISTS Ville;    CREATE TABLE Ville (  CityID int NOT NULL AUTO\_INCREMENT,  Nom\_ville varchar(255),  PRIMARY KEY (CityID)  );      CREATE TABLE Personnes (  PersonID int IDENTITY(1,1) PRIMARY KEY,  Nom varchar(255),  Prenom varchar(255),  Adresse varchar(255),  Ville\_Id int,  PRIMARY KEY (PersonID),  FOREIGN KEY (Ville\_Id) REFERENCES Ville(CityID)  );    La table VILLE  출曩昌춓目  Tentative d'ajout d'une personne dans une ville inconnue    Message d'erreur  eqeaep * 一 1011m1 b' 'IV |
|  |  |  |
| Faire en sorte que les informations d'une colonne (non clé primaire/étrangère) soient retrouvable  plus rapidement. | Use testDb;  DROP TABLE IF EXISTS Personnes ;    CREATE TABLE Personnes (  PersonID int IDENTITY(1,1) PRIMARY KEY,  Nom varchar(255),  Prenom varchar(255),  Adresse varchar(255),  Ville varchar(255)  );    CREATE INDEX IDX\_Nom  ON Personnes (Nom);      Ainsi tous les noms seront stockés en plus dans un Indexe. Cela permettra de faire des recherches par nom plus rapides. | Use testDb;  DROP TABLE IF EXISTS Personnes ;    CREATE TABLE Personnes (  PersonID int IDENTITY(1,1) PRIMARY KEY,  Nom varchar(255),  Prenom varchar(255),  Adresse varchar(255),  Ville varchar(255)  );    CREATE INDEX IDX\_Nom  ON Personnes (Nom);      Ainsi tous les noms seront stockés en plus dans un Indexe. Cela permettra de faire des recherches par nom plus rapides. |

**En complément : Certains stagiaires ont rencontré des problèmes avec la contrainte CHECK sur MySql**

Voici la syntaxe « Normale » : sur un exemple de vérification de l’Age qui doit être supérieur ou égal à 18

Une image contenant texte

Description générée automatiquement

Autre syntaxe pur contourner le problème : même exemple que dans le précédent mais en utilisant la notion de TRIGGER Une image contenant texte

Description générée automatiquement