SQL Server management Studio

Table des matières

[Installation des outils de gestion de SQL Server Express 1](#_Toc534550194)

[Découverte de SSMS 2](#_Toc534550195)

[Création d'une première base de données 5](#_Toc534550196)

[Création de table 6](#_Toc534550197)

[Par script SQL 6](#_Toc534550198)

[Par l'outil de création de table 8](#_Toc534550199)

[Ajout et modifications de données 12](#_Toc534550200)

[Par requête SQL 12](#_Toc534550201)

[Via l'outil de création/modification de données 13](#_Toc534550202)

[Modification de table 13](#_Toc534550203)

[Opération de sauvegarde et restauration 15](#_Toc534550204)

[Sauvegarde sous forme de fichier .bak 15](#_Toc534550205)

[Restauration d'une base de données à partir d'un .bak 16](#_Toc534550206)

[Génération de scripts 19](#_Toc534550207)

# Installation des outils de gestion de SQL Server Express

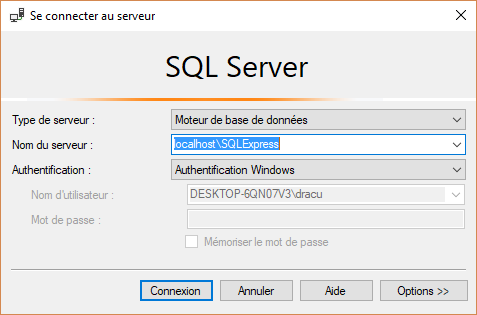
L'installation de SQL Server est maintenant fonctionnelle, néanmoins aucun outil n'est installé par défaut. Visual Studio permet des manipulations de base sur SQL Server, mais dans l'industrie, on préfèrera installer un outil bien plus puissant appelé "SQL Server Management Studio", abrégé en "SSMS".

Pour ce faire, installez SSMS via SSMS-Setup-FRA.exe présent dans le répertoire "commun" (Copiez le en local d'abord, ne le lancez pas depuis le répertoire commun). Les options par défaut de l'installateur sont bonnes.

*Pour information, SQL Server Management Studio se base sur une petite partie de Visual Studio (appelée "Visual Studio Shell") qui sera installée automatiquement. Une fois SSMS installé, il y aura donc potentiellement deux Visual Studio sur la machine, dans des versions différentes, ce qui n'est pas problématique en soi, mais si vous vous demandez d'où vient cette autre version de Visual Studio qui pourrait apparaitre plus tard, l'explication vient de là.*

Puis lancez SQL Server management studio en tapant la touche "Windows", puis "SSMS" puis "Entrée"

Le premier écran vous demander de saisir les informations permettant de se connecter à l'instance SQL, à savoir :



* Le nom du serveur
  + Dans notre cas, comme il s'agit d'une instance nommée, elle est de la forme :

NomDeVotrePC\NomDeLinstanceNommee

Donc par exemple

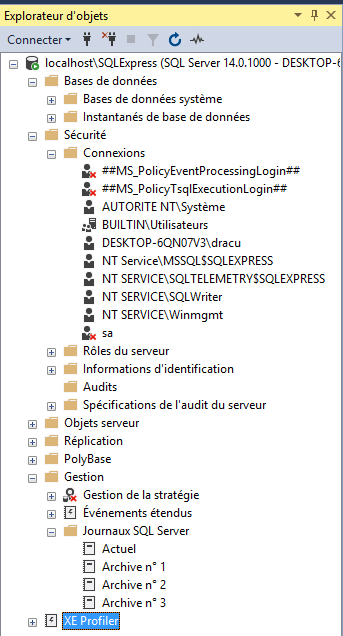
localhost\SQLExpress

* + Si l'instance était une instance par défaut, il suffirait ici de saisir "NomDeVotrePC" ou même plus simplement "."
* Les informations de connexion
  + Par défaut, SSMS tente de se connecter avec l'utilisateur Windows courant
  + Il est possible de se connecter avec un autre utilisateur Windows
  + Ou de se connecter avec un compte de connexion SQL.
  + Dans notre cas, on se connectera toujours avec l'utilisateur Windows courant.

SSMS se connecte alors au serveur, une nouvelle entrée apparait dans le panneau "Explorateur d'objet".

# Découverte de SSMS

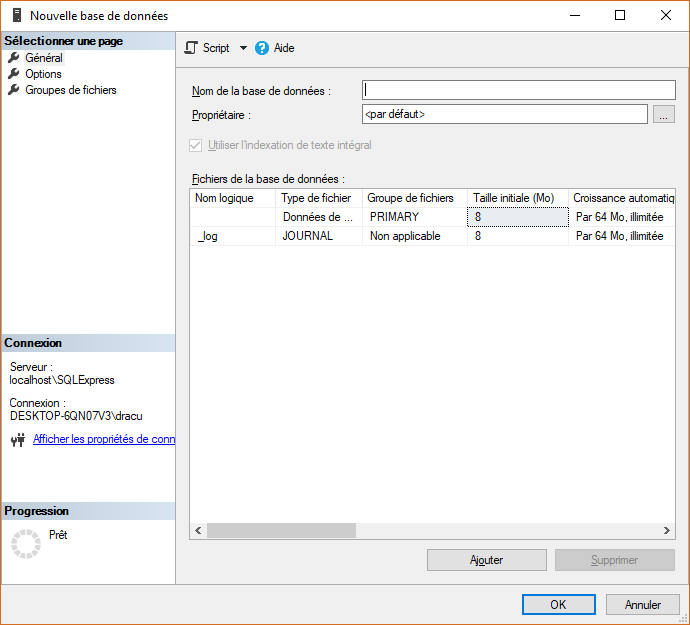
Une fois connecté à une instance SQL Serveur, on peut remarquer un certain nombre de choses (j'ai déplié certains nœuds pour plus d'explication)



* Le nœud "Bases de données" ne contient aucun élément autre que "Bases de données système" et "Instantanées de base de données" car aucune base de données n'est créée par défaut. Lorsque l'on créera des bases de données dans notre instance, elles apparaitront ici.
* Dans le nœud "Sécurité" puis "Connexions" apparaissent un certain nombre de comptes déjà existants, notamment l'utilisateur courant, ainsi que "sa" qui est probablement désactivé. Si vous double-cliquez sur votre login, puis que vous allez dans "Rôles du serveur", vous pourrez voir les rôles de sécurité que vous avez, notamment "sysadmin" (le plus important, il vous donne tous les droits).
  + Profitez-en pour ajouter un nouveau compte de connexion pour votre voisin de table en faisant un clic-droit sur "Connexions" puis "Nouvelle connexion…"
  + Dans l'onglet "Général", renseignez son nom d'accès sous la forme "DomainActiveDirectory\LoginUtilisateur" (celui utilisé pour se connecter à sa machine), puis validez.
  + Demandez à votre collègue de se connecter à votre instance SQL Server. Normalement, il devrait pouvoir s'y connecter, mais il n'y verra rien car son rôle par défaut ne lui permet rien (c'est "voulu").
* Le nœud "Gestion" puis "Journaux SQL Server" contient les journaux ("logs") de SQL Server et peuvent être très pratiques pour rechercher la cause de certains soucis, par exemple ceux de connexion.

# Création d'une première base de données

Nous allons créer notre première base de données. Pour ce faire, effectuez un clic-droit sur le nœud "Base de données", puis cliquez sur "Nouvelle base de données". L'écran suivant vous permettra de configurer la nouvelle base de données à créer.



Renseignez donc le nom puis notez que cet écran vous permet aussi de spécifier où seront disposés les fichiers qui contiendront plus tard :

* Les données à proprement parler (un fichier avec une extension ".mdf")
* Le journal de la base de données : concrètement la liste de toutes les opérations effectuées depuis la création de la base de données, permettant de revenir dans l'état d'une base de données dans le passé, de faire des sauvegardes incrémentales etc. (un fichier avec l'extension ".ldf")

Avant de cliquer sur OK, cliquez plutôt sur le bouton "Script" en haut de la fenêtre. Cela vous ouvrira un fichier texte qui comprendra une requête SQL vous permettant de créer la base de données telle que vous l'avez configuré dans la fenêtre. En fait, valider la fenêtre avec le bouton "OK" ne ferait qu'exécuter l'équivalent de la requête que vous avez généré. L'intérêt d'avoir ce script plutôt que de juste cliquer sur "OK" est que vous pouvez conserver ce script quelque part pour le relancer à l'envie plus tard, sans avoir à réeffectuer toutes les opérations de configuration manuellement.

*Il en sera de même plus tard, les différentes manipulations que SSMS vous proposera d'effectuer (création de table, de lignes, de vues, manipulation de droits, suppressions, sauvegarde et restauration etc.) ne sont en général que des interfaces graphiques qui vous permettent de générer du SQL en "sous-main". Il est toujours possible de récupérer ce code SQL, tout comme il est possible d'effectuer toutes ces opérations directement en SQL.*

Maintenant que vous avez à votre disposition le script SQL à exécuter pour générer votre première base de données, cliquez sur le bouton "Exécuter" dans la barre d'outils. L'onglet "Résultats" (ou "Messages" selon la version installée) devrait alors afficher "Commandes réussies.". La nouvelle base de données ne s'affichera pas immédiatement dans l'explorateur d'objets car il faut rafraichir la liste pour qu'elle apparaisse. Vous pouvez fermer puis rouvrir le nœud "Base de données" ou effectuer un clic droit sur ce nœud et cliquer sur "Rafraichir". Il en sera de même pour tous les objets que l'on va créer par la suite (tables, utilisateurs etc.)

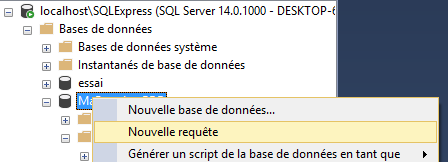
# Création de table

La création d'une table peut bien sûr s'effectuer par script SQL (via l'instruction CREATE TABLE…) ou en utilisant un outil plus visuel facilitant quelque peu cette écriture.

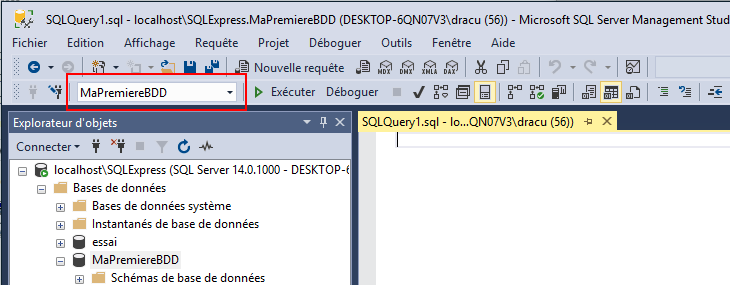
## Par script SQL

Pour ce faire, il faut :

* Faire un clic-droit sur le nœud de la base de données dans laquelle vous souhaitez créer votre table
* Sélectionner l'option "Nouvelle requête"



Une nouvelle fenêtre de requêtage apparait alors, et l'on observe en haut à gauche que la bonne base de données est déjà sélectionnée.



Dans cette fenêtre de requêtage, nous allons maintenant créer une table Utilisateurs contenant quelques colonnes. Saisissez le code suivant :

CREATE TABLE Utilisateurs (

UtilisateurId int IDENTITY(1,1),

Prenom nvarchar(255),

Nom nvarchar(255),

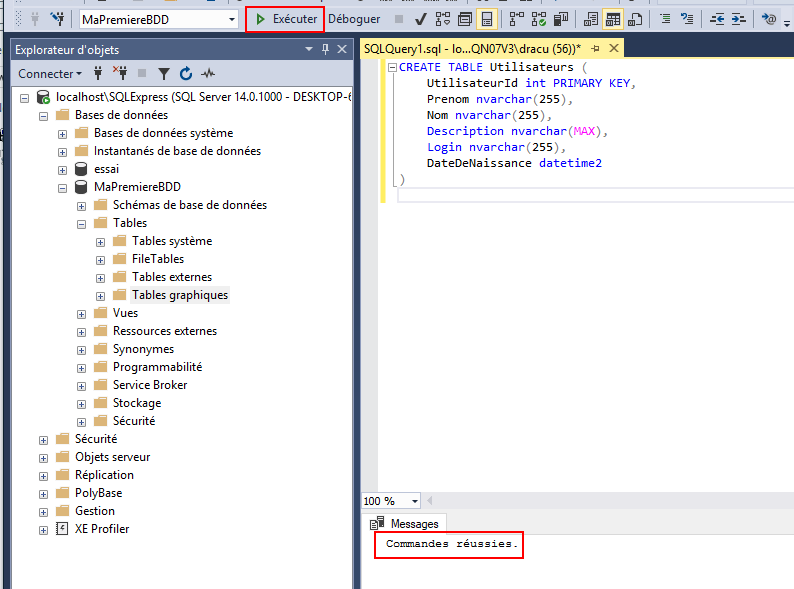
Description nvarchar(MAX),

Login nvarchar(255),

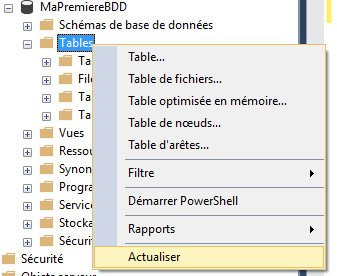
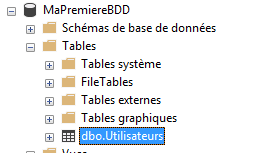
DateDeNaissance datetime2 NOT NULL

)

Puis cliquez sur le bouton "Exécuter" (raccourci F5) et vérifiez que l'onglet "Message" affiche bien "Commandes réussies"



En rafraichissant le nœud "Tables", la nouvelle table apparait.

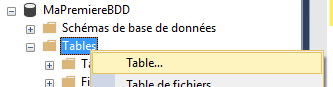


## Par l'outil de création de table

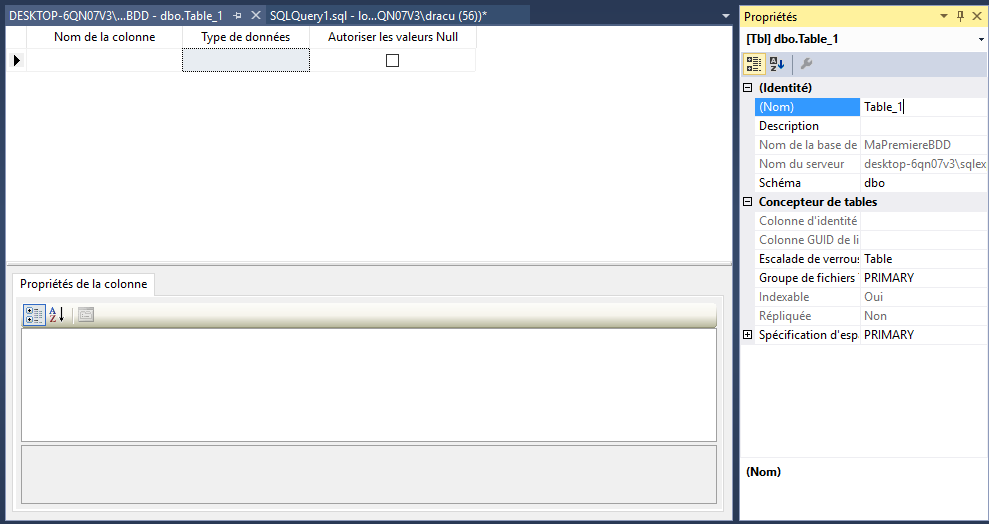
Nous allons maintenant créer la table "HistoriqueDeConnexion" qui contiendra les colonnes suivantes :

* HistoriqueDeConnexionId de type int, clé primaire de la table, et auto-incrémentée
* UtilisateurConnecteId de type int, clé étrangère vers la table Utilisateurs
* DateDeConnection de type datetime2 qui doit être renseignée ("NOT NULL ")
* DateDeDeconnection de type dateime2 qui cette-fois ci peut être vide
* AdresseIP de type nvarchar(39)

Pour ce faire, il faut tout d'abord effectuer un clic-droit sur le nœud "Table" puis sélectionner l'option "Table…"



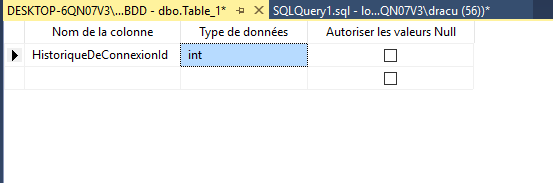
La fenêtre qui s'ouvre se décompose en trois blocs :



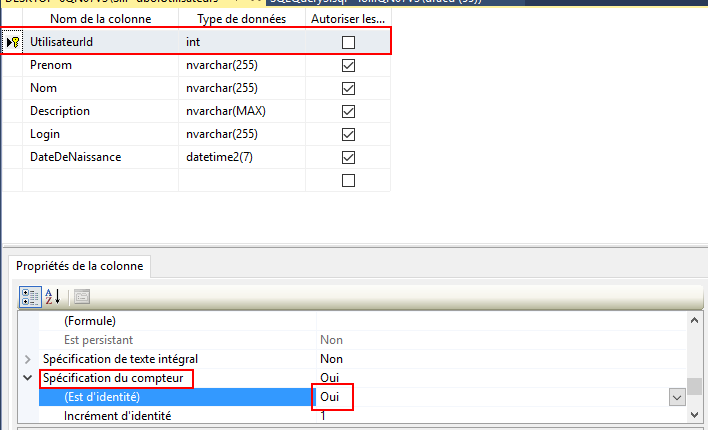
* Le bloc en haut à gauche contiendra la liste des colonnes de la table que l'on souhaite créer et les options principales
* Le bloc en bas à gauche contiendra les options "avancées" de la colonne actuellement sélectionnée
* Le bloc à droite contient les options de la table que l'on souhaite créer.

Plusieurs étapes sont à effectuer :

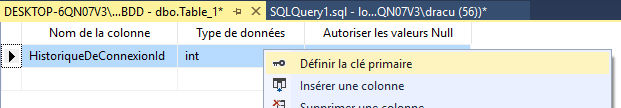
* Il faut tout d'abord donner le bon nom à la table que l'on souhaite créer en modification le champs "(Nom)" du bloc "Propriétés" de droite en renommant "Table\_1" en "HistoriqueDeConnexion"
* Puis pour chaque colonne à créer :
  + Ajouter une ligne correspondant dans la liste des colonnes en lui donnant un nom, et un type de données



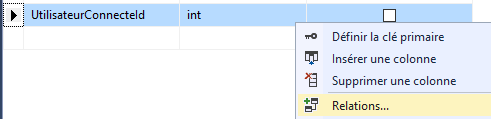
* + Eventuellement modifier ses propriétés avancées en fonction des besoins:
    - Par exemple, la colonne HistoriqueDeConnexionId est auto-incrémentée

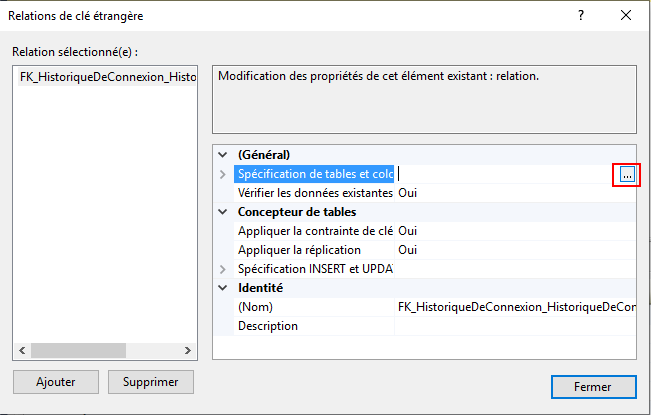


* + Lui associer des comportements avancés (clé primaire, clé secondaire par exemple) en effectuant un clic droit dessus
    - Par exemple, la colonne "HistoriqueDeConnexionId" est notre clé primaire

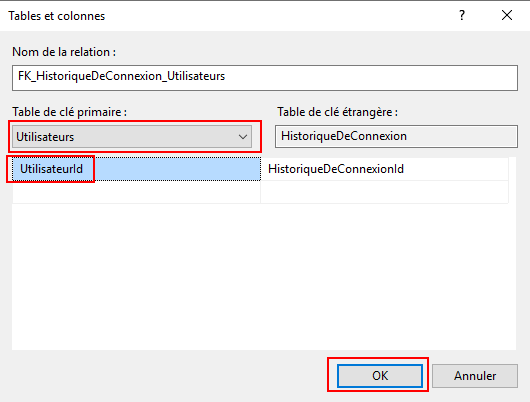


* + - La colonne " UtilisateurConnecteId" doit avoir une relation de type "clé étrangère" sur la colonne "UtilisateurId" de la table Utilisateur

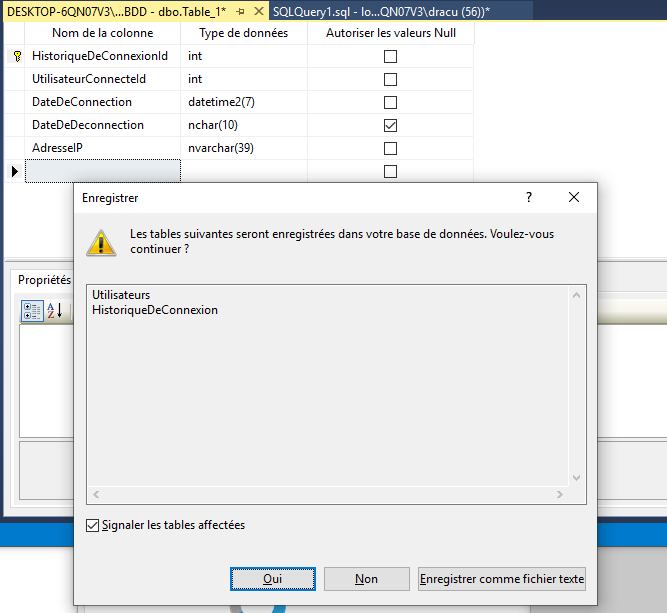




*(Nb. Le bouton est caché par défaut, il faut cliquer dans le champs vide "Spécification de tables et colonnes pour qu'il apparaisse)*



Puis, une fois toutes les colonnes bien définies, il suffit alors d'enregistrer (raccourci "CTRL+S") pour que la table soit créée. SSMS informe alors des tables qui seront modifiées. En cliquant sur "Oui", le script de création de la table sera immédiatement exécuté, mais il est aussi possible de sauvegarder le script dans un fichier texte à part pour l'exécuter manuellement plus tard.



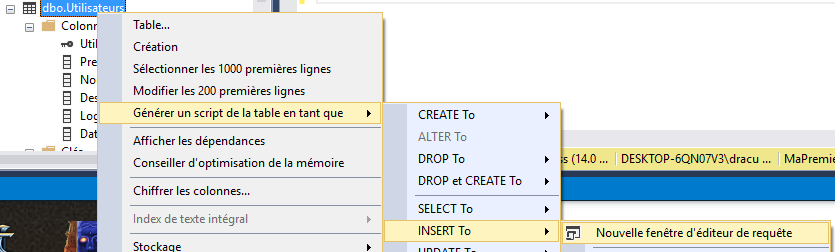
Profitez-en maintenant pour créer les quelques tables proposées en fonction du scénario choisi tous ensemble.

# Ajout et modifications de données

Maintenant que les tables sont créées, on peut s'occuper des données. Pour ajouter et modifier des données, il y a là aussi plusieurs manières de faire.

## Par requête SQL

Tout d'abord, on peut le faire par requête SQL via l'instruction INSERT INTO. On peut saisir cette requête à la main, ou utiliser l'outil de génération de script :



Puis en remplaçant les valeurs dans le script

INSERT INTO [dbo].[Utilisateurs]

([Prenom]

,[Nom]

,[Description]

,[Login]

,[DateDeNaissance])

VALUES

('Mike'

,'Brant'

,'RIP'

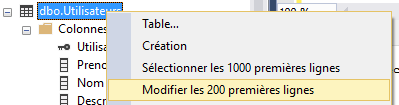
,'LMTA'

,'1947-02-01')

Notez qu'il existe tout un tas de script que vous pouvez générer facilement à partir d'une table déjà existante, notamment les SELECT, UPDATE, DELETE, ou CREATE TABLE.

## Via l'outil de création/modification de données

En effectuant un clic droit sur la table, puis en sélectionnant l'option "Modifier les 200 premières lignes"



Un tableur s'ouvre, vous permettant :

* De modifier les 200 premières lignes (…) directement, la modification s'effectuant lors de l'appui de la touche "Enter" ou lors du passage d'une ligne à l'autre
* De rajouter des nouvelles lignes directement en saisissant des valeurs sur la ligne de saisie du tableur (la toute dernière)

# Modification de table

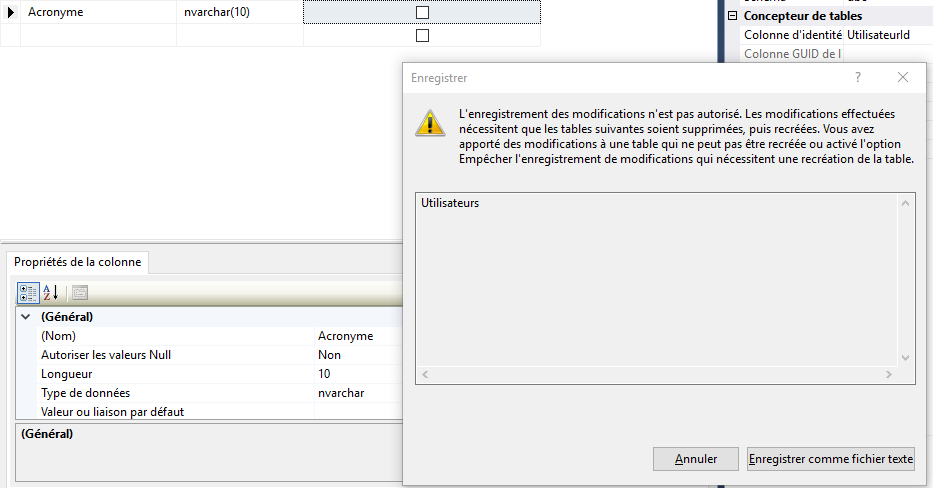
Il est possible de modifier une table soit via l'instruction ALTER TABLE, soit via le même outil que celui utilisé lors de la création. Pour ce faire, il faut effectuer un clic droit sur la table que l'on souhaite modifier, et sélectionner l'option "Création" (…)



Attention, l'outil de modification de table ne permet pas de tout faire. Imaginons le scénario suivant :

* On souhaite rajouter une nouvelle colonne "Acronyme" dans la table "utilisateur". Cette colonne de type nvarchar(10) doit toujours être renseignée (donc la valeur null ne peut être acceptée).

Essayez de rajouter la colonne puis de sauvegarder (attention à bien décocher le fait que la valeur null soit autorisée). Un message vous informera que la sauvegarde n'est pas possible car rajouter la colonne non null n'est pas possible sur une table ayant déjà des données.



Deux possibilités s'offrent à vous :

* La première consiste à désactiver ce message d'avertissement, en laissant SSMS détruire puis recréer totalement la table de zéro, mais vous perdrez alors toutes vos données.
  + Pour ce faire, il faut aller dans le menu "Outils", puis "Options", puis "Concepteurs", puis décocher l'options "Empêcher l'enregistrement de modifications qui nécessitent une recréation de la table"
  + Vous pourrez alors sauvegarder vos modifications
  + Je vous conseille TRÈS FORTEMENT de réactiver cette option une fois vos modifications apportées, car sa désactivation peut être assez dévastatrice, surtout pour une modification faite trop vite le matin même du jour de votre présentation par exemple 😉
* La seconde (la plus propre) consiste à écrire le code correspondant vous-même, en gérant le cas particulier des lignes existantes. Par exemple, vous pourriez écrire un code SQL qui :
  + Créerait la colonne avec la possibilité d'avoir null temporairement
  + Assignerait à chaque ligne la première lettre du nom et du prénom comme acronyme
  + Modifierait la colonne pour supprimer la possibilité d'avoir la valeur null

Effectuez ces deux approches en débutant par la seconde pour vous y familiariser.

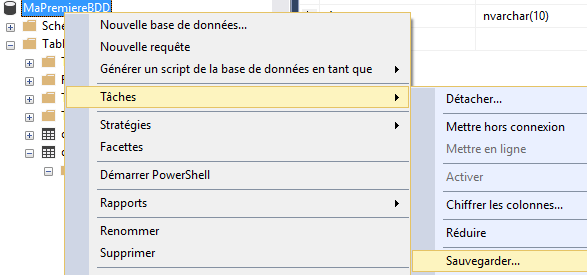
# Opération de sauvegarde et restauration

La sauvegarde d'une base de données et sa restauration sont des opérations qui sont effectuées sur quasiment toutes les bases existantes pour des raisons évidentes, que ce soit pour ne pas risquer de perdre ses données, pour migrer une base de données d'un serveur à un autre, pour se partager une base de données entre différents développeurs etc. Il existe plusieurs manières de les effectuer, chaque méthode ayant différents usages et intérêts.

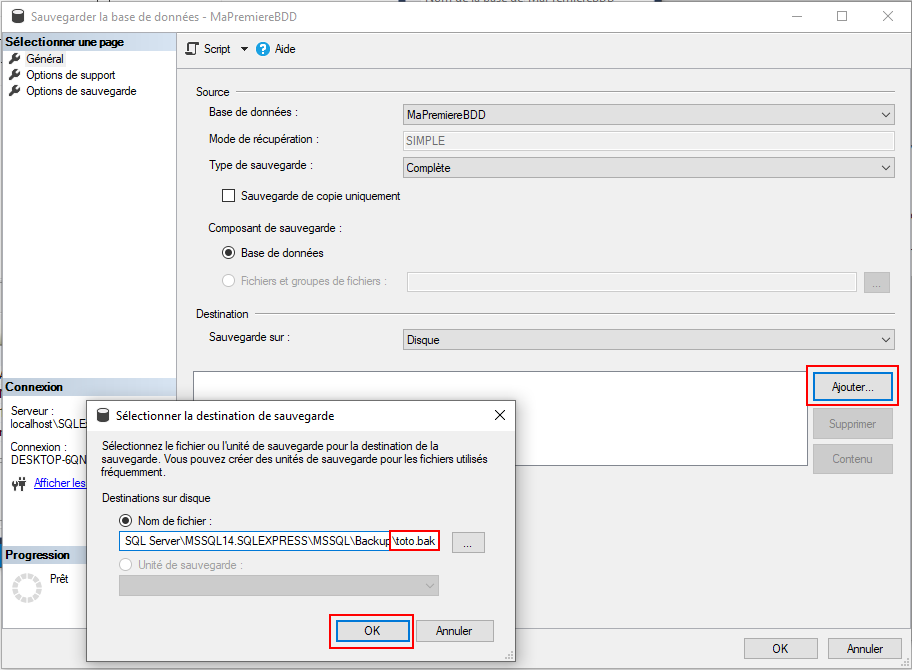
## Sauvegarde sous forme de fichier .bak

SQL propose un procédé assez puissant (et complexe) pour effectuer des sauvegardes de base. Ce procédé est pensé pour gérer beaucoup de cas, notamment des bases de données très volumineuses, des sauvegardes sur cassettes, des sauvegardes partielles etc. L'interface peut donc sembler étrange mais les choix par défaut sont bien positionnés. Attention, cette façon de faire a un gros désavantage dont il sera question plus loin dans le document.

La principale manière d'effectuer une sauvegarde consiste à effectuer un clic-droit sur la BDD que l'on souhaite sauvegarder, puis à sélectionner l'option "Tâches" puis "Sauvegarder…"



La fenêtre qui s'ouvre contient un certain nombre d'options avancées (type de sauvegarde, composant de sauvegarde etc.) sur lesquels nous ne reviendront pas (je peux y répondre à l'oral si ça vous intéresse cependant), hormis la dernière, la "Destination". C'est dans ce champs là que vous pourrez spécifier où le fichier .bak sera généré, en cliquant sur "Ajouter…" puis en sélectionnant un nom de fichier

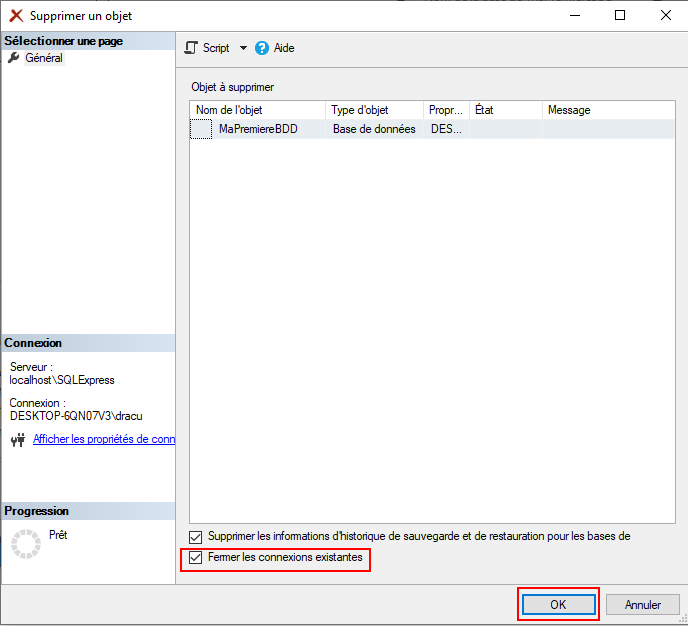


En validant la fenêtre de sauvegarde de la base (ou en générant puis exécutant le script SQL correspondant, cf. document "Installation SQL server.docx"), la base est alors sauvegardée à l'endroit spécifié.

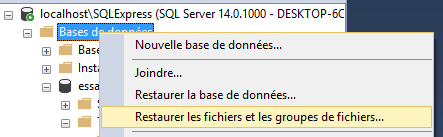
## Restauration d'une base de données à partir d'un .bak

Pour restaurer une base de données à partir d'un .bak, il faut :

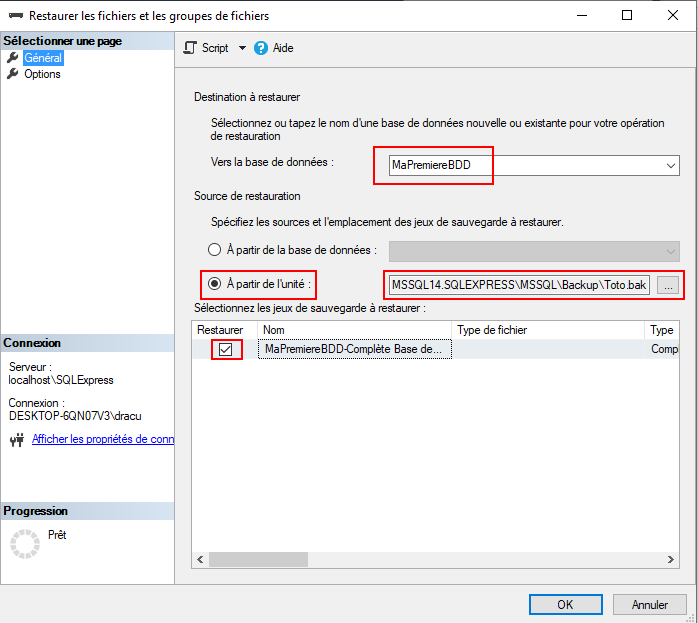
* Tout d'abord qu'aucune base de données avec le même nom que celle que l'on souhaite restaurer existe. Il faudra donc en général supprimer la base de données existante en effectuant un clic-droit sur celle-ci, puis en sélectionnant l'option "Supprimer". Dans la fenêtre qui s'ouvre, cochez l'option "**Fermer les connexions existantes**" car sinon, toute autre connexion en cours sur votre base (notamment par SSMS lui-même) pourrait empêcher la suppression de celle-ci.



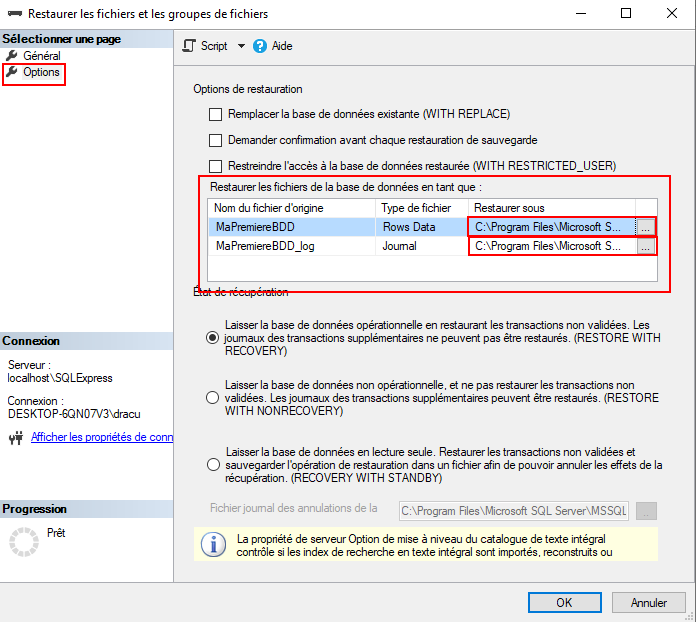
* Puis sur le nœud "Base de données", sélectionnez l'option "Restaurer les fichiers et les groupes de fichiers"



* Puis dans la fenêtre qui s'ouvre :
  + Entrez le nom de la base de données que vous souhaitez recréer
  + Sélectionnez "A partir de l'unité", puis sélectionnez votre fichier .bak, et cochez la case "Restaurer" dans la liste des jeux de sauvegarde à restaurer



Si vous restaurez un bak d'un PC à un autre, dans la page "Options", vérifiez que les chemins de fichier (colonne "Restaurer sous") des deux lignes qui apparaitront correspondent bien à des dossiers qui existent sur la machine sur laquelle vous souhaitez restaurer le fichier backup (par défaut, il tentera de réutiliser le même chemin de fichier que celui d'origine, qui n'a peut-être plus de sens sur le nouveau serveur).



Et enfin, Validez en cliquant sur "OK" en en générant puis exécutant le script correspondant.

**Un point très important à savoir cependant : les fichiers .bak ne sont pas rétroactifs. Il n'est pas possible d'utiliser une sauvegarde effectuée sur un SQL server plus récent que là où l'on souhaite le restaurer. L'autre sens est possible cependant (restaurer un ancien .bak sur un SQL server récent). Le .bak n'est donc pas forcément la meilleure option pour faire passer une BDD d'un développeur à l'autre ! Attention notamment aux mises à jour automatiques de SQL Server par exemple.**

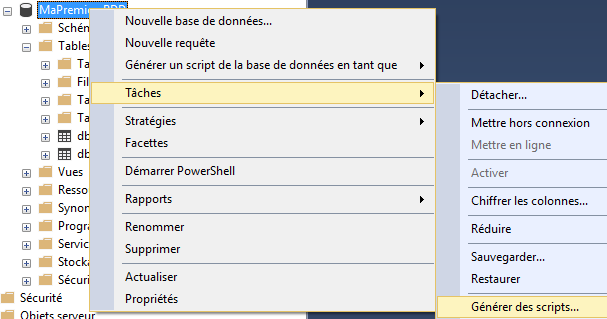
De même, si vous utilisez cela pour déplacer la base de données, par exemple pour le jour de la présentation d'un projet, assurez-vous au préalable que votre .bak et l'environnement destination soient compatibles.

## Génération de scripts

Il existe une dernière manière plus simple et très pratique pendant le développement (mais beaucoup moins performante et efficace, inutilisable sur des grosses BDD) pour sauvegarder et transférer une BDD qui consiste à utiliser un outil pour générer tous le code SQL à exécuter pour recréer une base de données exactement dans le même état, en recréant les tables une par une, en insérant les données à l'identique etc. Une fois ce script généré, il alors suffit de transférer ce fichier SQL sur la destination, puis à réexécuter par exemple via SSMS.

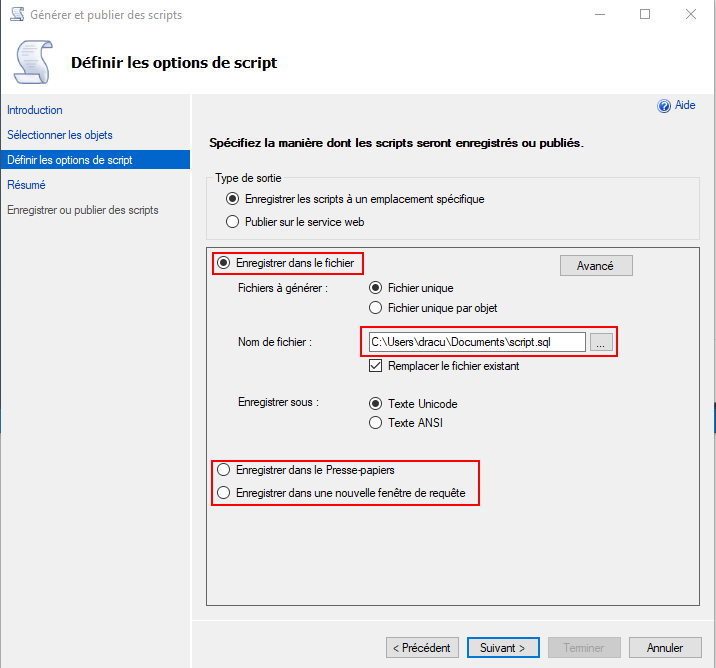
Pour ce faire, sur la BDD à sauvegarder :

* Effectuez un clic-droit puis "Tâches" puis "Générer des scripts…"

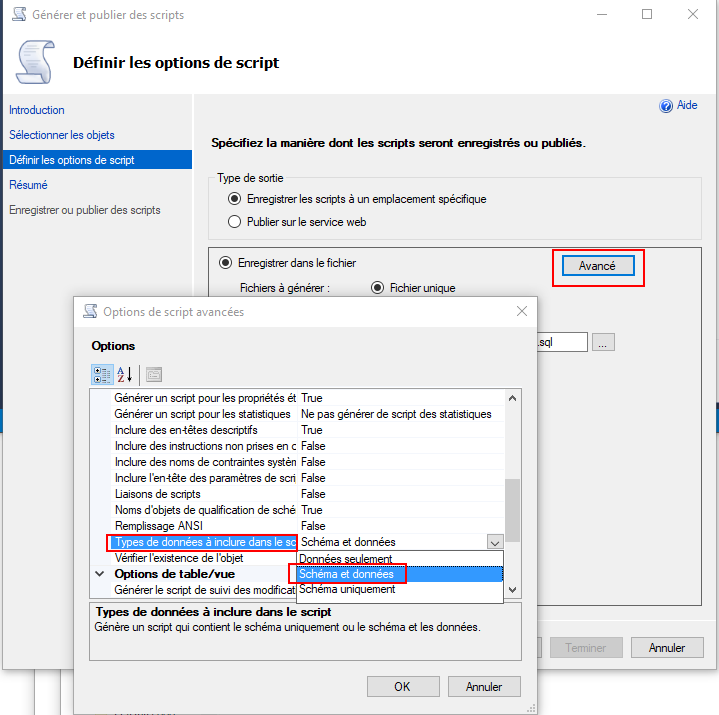


Puis dans la fenêtre qui s'ouvre, vous pouvez laisser les options par défaut, et valider chaque écran sauf dans la fenêtre "**Définir les options de script**" :

* Tout d'abord, vous noterez dans cet écran que c'est ici que vous spécifierez où sera créé le fichier script SQL correspondant. Vous pouvez d'ailleurs choisir de ne pas créer de fichier mais d'ouvrir directement une requête dans SSMS ou de l'enregistrer dans le presse-papier.



* Par défaut, le script généré ne contient que le schéma des données (tables, colonnes etc.) mais pas les données de chaque table. Pour inclure dans le fichier les instructions pour copier les données, il faut cliquer sur "Avancé" puis modifier l'option "Types des données à inclure dans le script" en y affectant la valeur "Schéma et données"



* Puis validez tout cela pour générer un fichier SQL directement exécutable par SSMS.