Institut Supérieur industriel de Bruxelles

Rue Royal 150, 1000 Bruxelles



Rapport

Système de gestion de bases

De données

Auteur : Professeur :

Wail KITTA Rodrigues Abilio Sergio E Sousa

Introduction

Dans ce cadre de ce cours, il faudra mettre en place un système de gestion de base de données sur SQL Server Management. L’objectif de ce cours est une meilleure compréhension des différents éléments pour pouvoir gérer une base de données (table, contrainte, procédure, etc.…) et leurs prises en main.

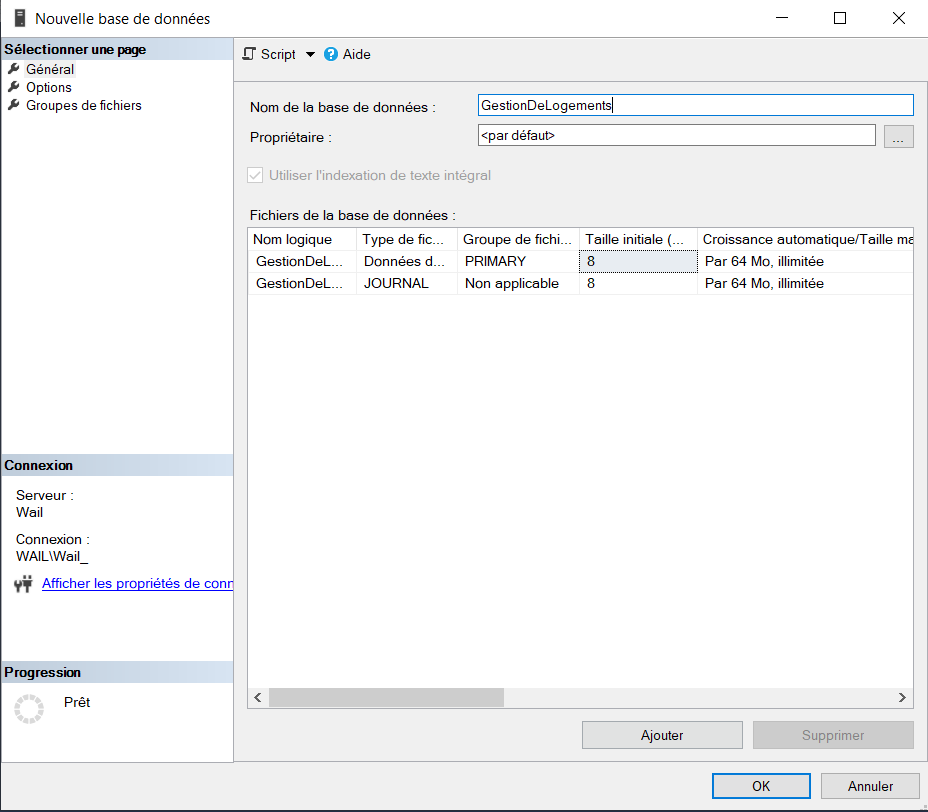
Le choix de système est la gestion de logements.

Création de tables

Donc le choix de système est la gestion de logements. De ce fait les tables qui répresente les informations de notre système seront :

* La table “**Logement**”
* La table “**Locataire**”
* La table “**Quartier**”
* La table “**Type de logement**”
* La table “**Commune**”

Avant de créer ces tables, il faut d’abord créer la base de données. Tout d’abord, il suffit qu’après la connexion à SQL Server Management d’aller dans la section “Bases de données” dans l’explorateur de fichier, clic-droit sur le dossier Bases de données -> Nouvelle base de données. Une fois arrivé au wizard de création de la base de données, il suffit juste d’entrer le nom de notre BD. (Voir figure 1)



[Figure 1 : Wizard de création de base de donnée]

Mise en place des tables, il existe 2 manières d’élaborer des tables dans SQL Server Management, la première est la création d’une table via une requête de création de table, et la seconde via l’interface de SQL Server Management.

C’est par la deuxième option que nous allons élaborer nos tables.

**Table Logement :**

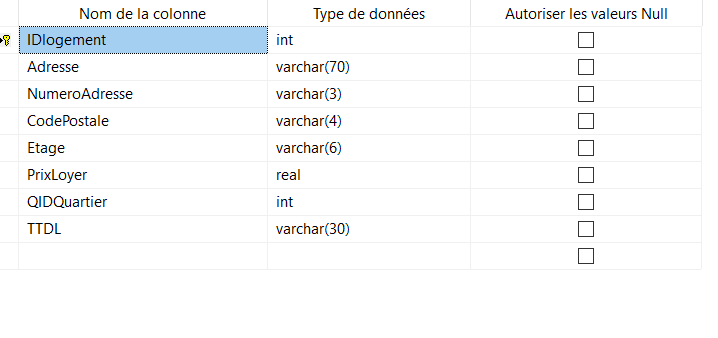
Pour créer une table, il faut aller dans une base de données et faire un clique droit sur le dossier Tables -> Nouveau -> Table.



[Figure 2 : Ecran de création d’une table]

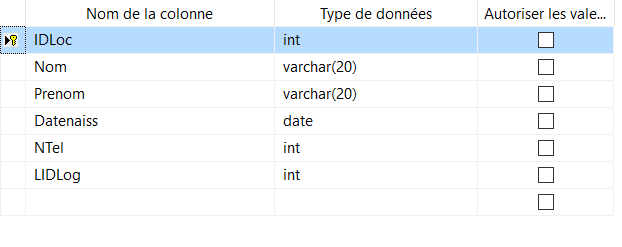
Ensuite, il faut remplir la table donc ici pour la table “**Logement”**  il y aura comme attributs:

* IDLogement : Identifiant de logement(clé primaire).
* Adresse
* NumeroAdresse
* CodePostale
* Etage
* PrixLoyer
* QIDQuartier : Identifiant de Quartier (clé étrangère)
* TTDL : Type de Logement (clé étrangère)



[Figure 3 : Création des attributs dans la table “Logement”]

**Table Locataire :**

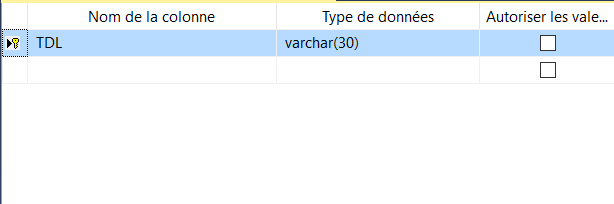


[Figure 4 : Création des attributs dans la table “Locataire”]

Donc voici les attributs correspondants pour la table “**Locataire”** :

* IDLoc : Identifiant Locataire(clé primaire).
* Nom
* Prenom
* Datenaiss : Date de naissance
* NTel : Numéro de téléphone
* LIDLog : c’est le *ID de logement* de la table “**Logement”** donc ceci c’est une clé étrangère dans cette table ci-dessus.

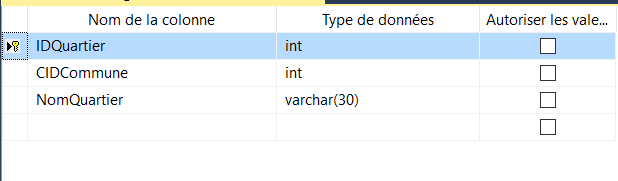
**Table Type de logement :**



[Figure 5 : Création des attributs dans la table “Type de Logement”]

Ici il y a qu’un attribut *TDL* qui est le Type de Logement (clé primaire).

**Table Quartier :**

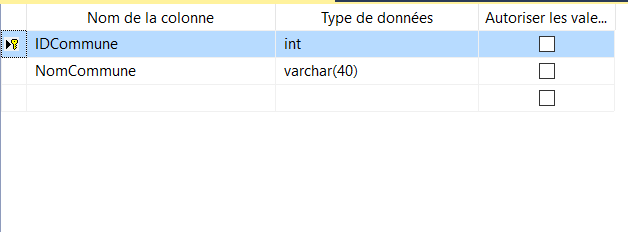


[Figure 6 : Création des attributs dans la table “Quartier”]

Donc voici les attributs correspondants pour la table “**Quartier”** :

* IDQuartier : Identifiant Quartier(clé primaire).
* CIDCommune : c’est le ID Commune de la table “Commune” (clé étrangère).
* NomQuartier

**Table Commune :**

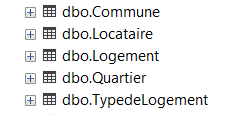


[Figure 7 : Création des attributs dans la table “Commune”]

Donc voici les attributs correspondants pour la table “**Commune”** :

* IDCommune : Identifiant Commune (clé primaire).
* NomCommune

Pour finir tous les tables ont été créés:

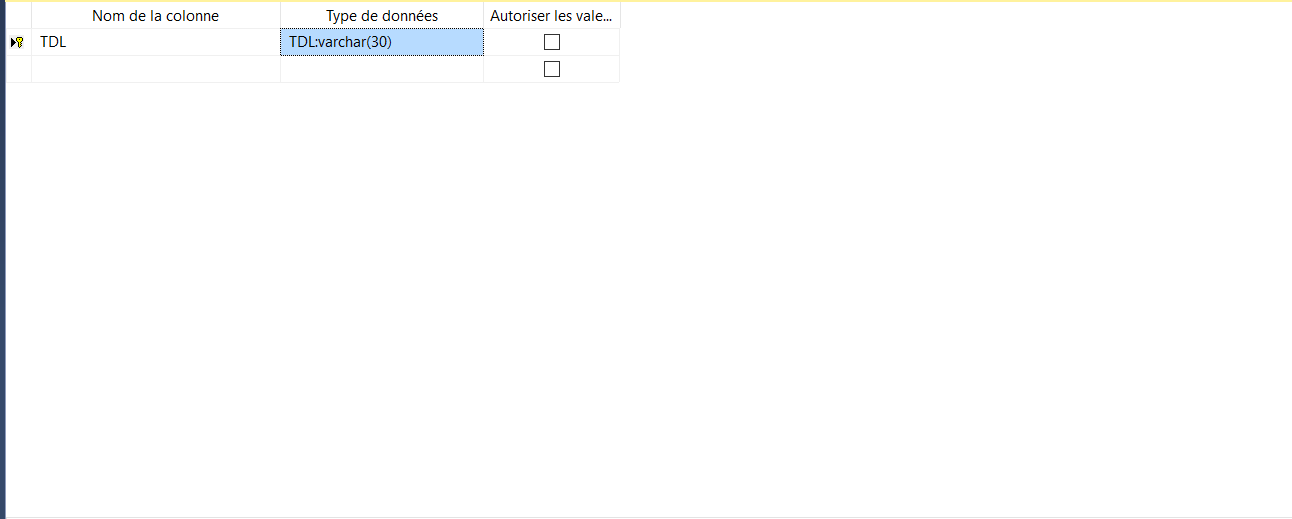


[Figure 8 : tous les tables créés]

Création TDU:

Les Types de données définis par l’utilisateur (TDU) va permettre de créer des types de données personnalisés avec des règles et des contraintes spécifiques.

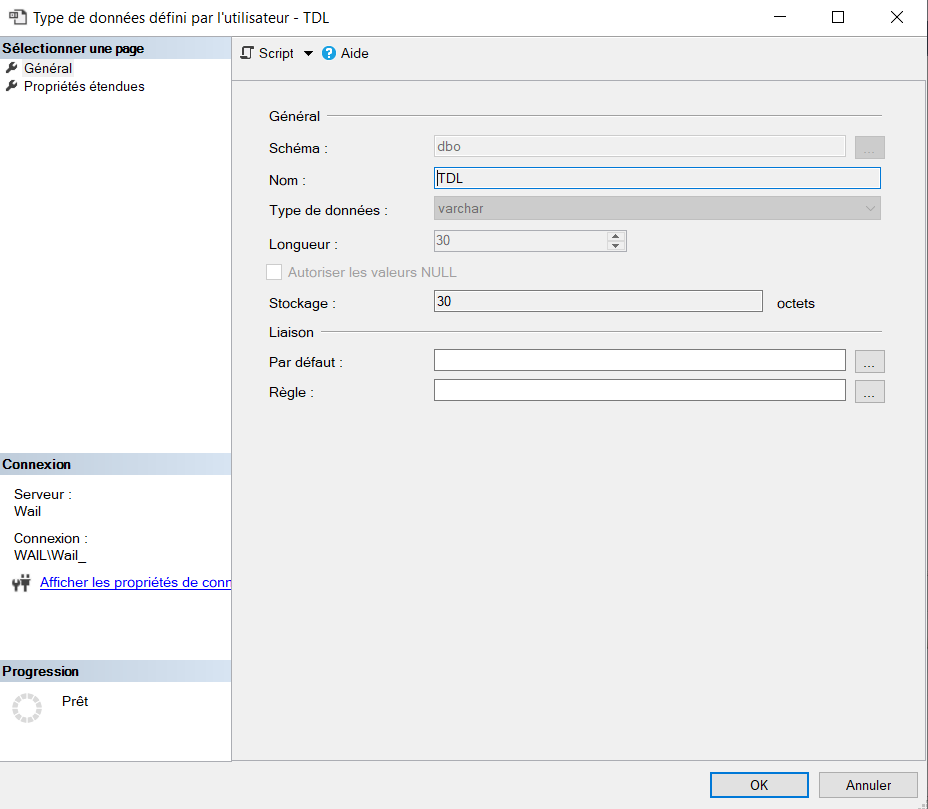
Par exemple avec l’attribut TDL de la table “TypedeLogement” :



[Figure 9 : Type Adresse ]

Donc on a créé le type de donnée personnalisé qui est “TDL” où c’est varchar (30).

Pour créer un TDU sur SQL Server Management, il faut aller dans la base de données utilisé et aller sur le dossier Programmabilité -> Types -> Types de Données définis par l’utilisateur. Puis , il faut faire clic-droit sur le dossier Type de données définis par l’utilisateur -> Nouveau type de données définis par l’utilisateur où un wizard s’affichera (voir figure 10 ci-dessous) :



[Figure 10 : Wizard de création de type]

Donc le TDU c’est une type de donnée personnalisé qui va servir à remettre dans un attribut qui se trouve dans 2 ou plusieurs tables.

Création des Contraintes:

Les contraintes ce sont des règles et des conditions appliquées aux données d’une table pour garantir l’intégrité et la cohérence des données. Ici on a utilisé que 4 contraintes :

* Contrainte de clé primaire
* Contrainte d’intégrité référentielle (Contrainte clé étrangère)
* Contrainte de vérification (Contrainte Check)
* Contrainte de non-nullité

Ces contraintes sont définies lors de la création d’une table ou peuvent être ajoutées à l’aide des commandes.

* **Contrainte de clé primaire**

Les contraintes primaires sont utilisées pour les colonnes servant d’identifiant unique. En effectuant un clic-droit sur une ou plusieurs colonnes d’une table il est possible de la ou les “*définir comme clé primaire*” , une fois fait, il est possible de voir l’apparition d’une icône en forme de clé.(Figure 12,13,14,15,16)



[Figure 12 : clé primaire IDlogement de la table Logement]



[Figure 13 : clé primaire IDLoc de la table Locataire]



[Figure 14 : clé primaire TDL de la table TypedeLogement]



[Figure 15: clé primaire IDQuartier de la table Quartier]

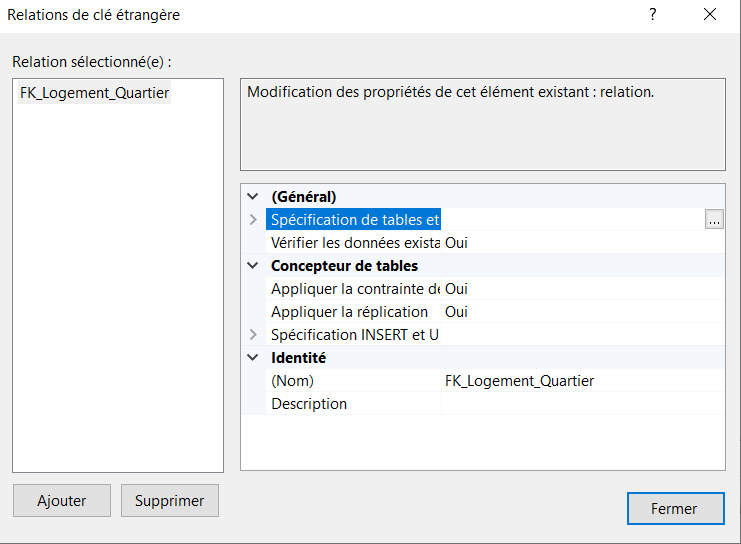


[Figure 16 : clé primaire IDCommune de la table Commune]

* **Contrainte de la clé étrangère**

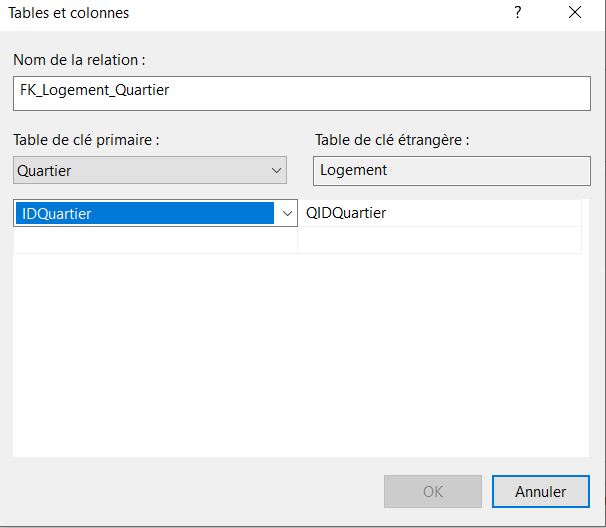
Les contraintes de clé étrangère sont utilisées pour établir une relation entre deux tables en liant une colonne de la table enfant à la clé primaire de la table parente.

Pour élaborer des contraintes de clé étrangère via l’interface, il faut la faire dans une colonne d’une table qui sera bien sûr une clé étrangère donc clic-droit sur la colonne puis cliquez sur *Relations* puis cette fenêtre apparaitra voir ci-dessous (Figure 17).



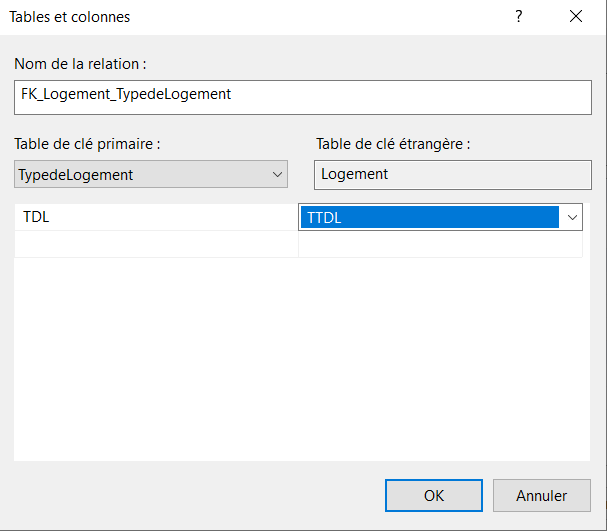
[Figure 17 : Wizard de création de relation]

Donc ici il y a déjà une contrainte, pour le moment cette contrainte ne fait rien afin qu’elle puisse réaliser la tâche clé étrangère, nous allons la configurer. En allant sur *Général -> Spécifivation de tables et …* , cette fenêtre apparaitra (voir figure 18 ci-dessous).



[Figure 18 : Configuration de la contrainte étrangère]

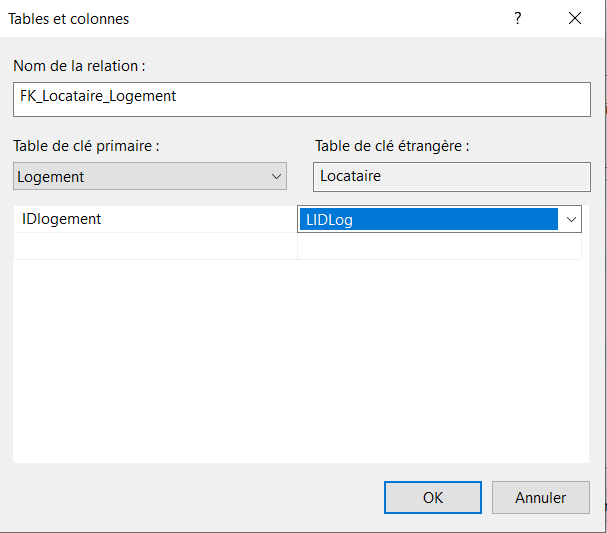
Ici c’est dans la table Logement, la clé étrangère c’est QIDQuartier qui est la clé primaire IDQuartier de la table Quartier, donc ici on les a liés.



[Figure 19 : Configuration de la contrainte étrangère TTDL]

Ici la clé étrangère TTDL de la table Logement qui est la clé primaire TDL de la table TypedeLogement.

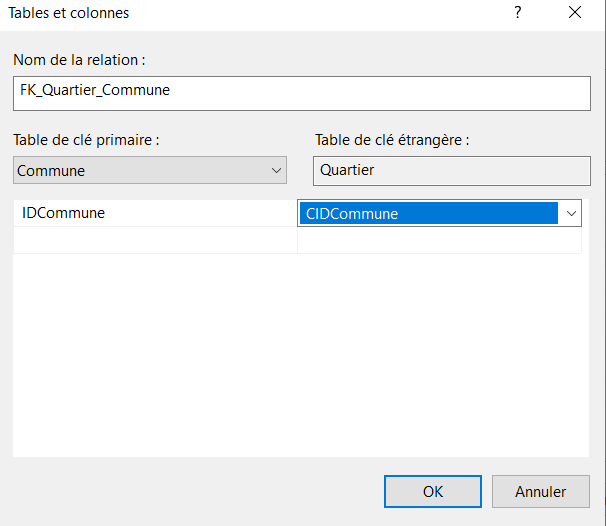
**Table Locataire :**



[Figure 20 : Configuration de la contrainte étrangère LIDLog]

Ici la clé étrangère LIDLog de la table Locataire qui est la clé primaire IDlogement de la table Logement.

**Table Quartier :**



[Figure 21 : Configuration de la contrainte étrangère CIDCommune]

Ici la clé étrangère CIDCommune de la table Quartier qui est la clé primaire IDCommune de la table Commune.

**Table TypedeLogement :**

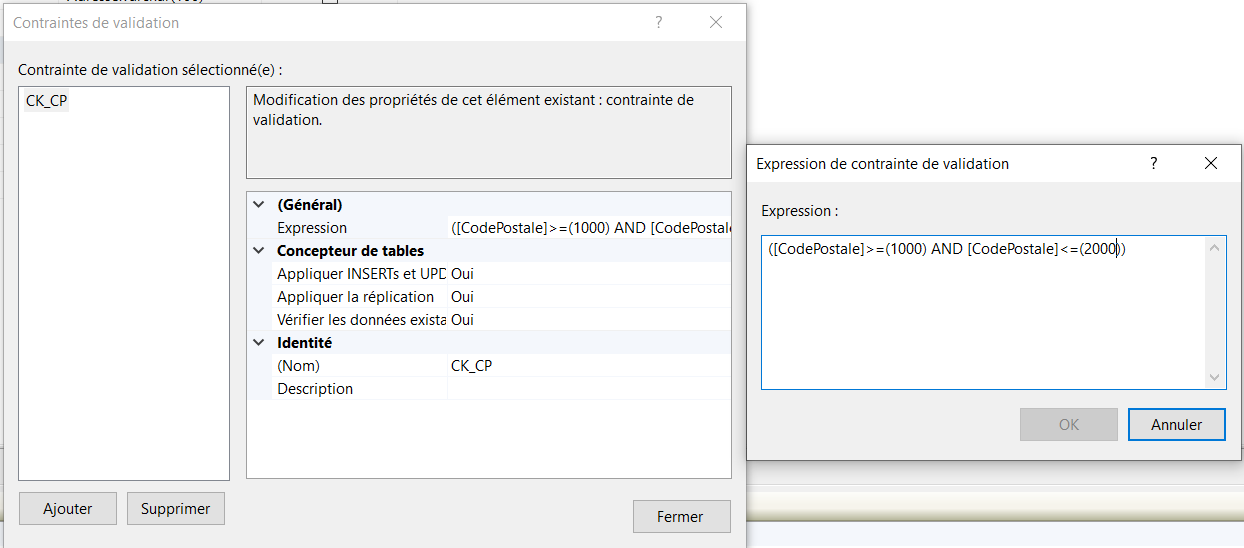
Ici il n’y a pas de clé étrangère.

**Table Commune :**

Ici il n’y a pas de clé étrangère.

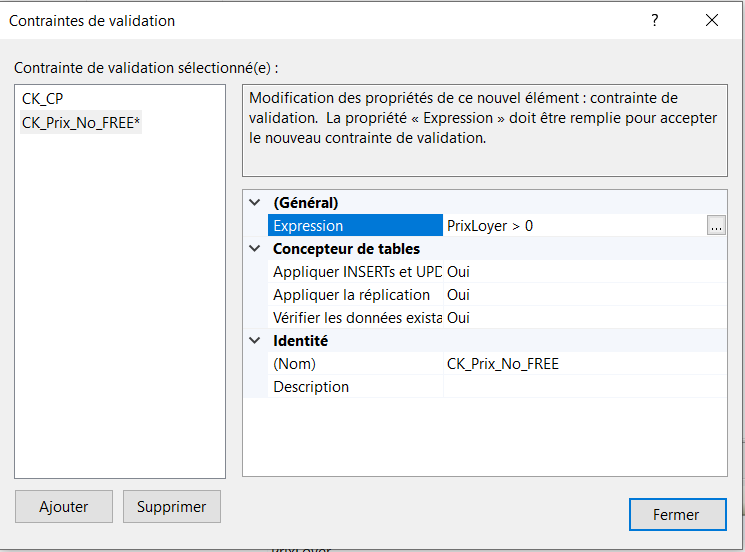
* **Contrainte Check**

Une contrainte Check permet de spécifier une condition ou une expression logique pour les valeurs dans une colonne.



[Figure 22 : Contrainte de validation du CodePostale]

Ici la contrainte CK\_CP donc la contrainte de vérification du CodePostale donc d’après l’expression il accepte les valeurs de 1000 à 2000. La colonne CodePostale se trouve dans la table Logement.



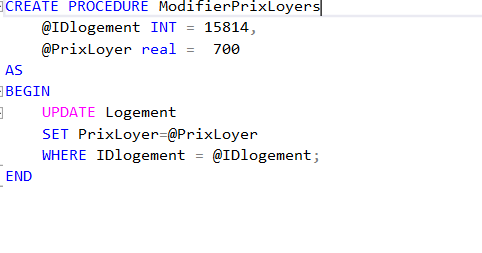
[Figure 23 : Contrainte de validation du PrixLoyer]

Ici la contrainte CK\_Prix\_NOT\_FREE donc la contrainte de vérification du PrixLoyer donc d’après l’expression il accepte les valeurs plus grand que 0. La colonne PrixLoyer se trouve dans la table Logement.

Création des procédures :

Une Procédure stockée est un objet de base de données qui contient une ou plusieurs instructions SQL précompilées et stockées dans le serveur de base de données.

Voici ici il y a qu’une procédure dans laquelle on modifie le PrixLoyer.



[Figure 24 : Procédure stockée modification de PrixLoyer]

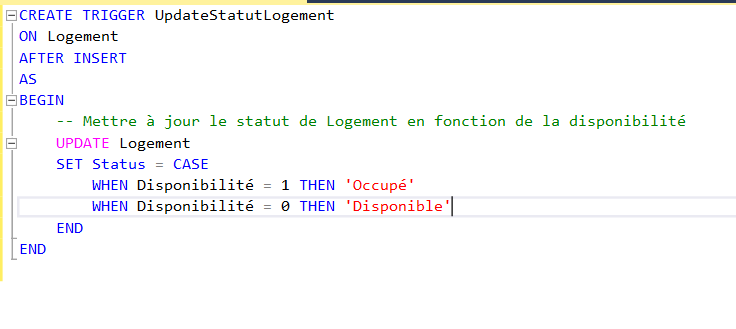
Pour y parvenir, il suffit de choisir une base de données et puis aller dans le dossier *Programmabilité -> Procédures stockées* clic-droit sur le dossier *Procédures stockées* donc *Nouveau -> Procédure stockée...* , Après il y aura apparition de la figure 24 ci-dessus.

A quoi sert une procédure stockée? Une procédure stockée est créée et enregistrée dans la base de données, ce qui permet de l'exécuter à plusieurs reprises sans avoir à réécrire le code SQL à chaque fois.

Les procédures stockées sont des routines qui peuvent être appelées explicitement pour effectuer des opérations spécifiques.

Création de Trigger :

Un trigger est un type d’objet de base de données qui est associé à une table et est automatiquement exécuté en réponse à certains événements, tels que l'insertion, la mise à jour ou la suppression de données dans cette table. Un trigger permet d'effectuer des actions supplémentaires avant ou après ces événements, comme exécuter une série d'instructions SQL, appeler une procédure stockée, mettre à jour d'autres tables, etc.



[Figure 25 :Trigger pour vérification disponibilité Logement]

Donc ici ce Trigger va *vérifier la disponibilité de logement.* Donc les triggers sont souvent utilisés pour implémenter des règles métier, appliquer des contraintes personnalisées, effectuer des audits ou enregistrer des informations supplémentaires lors de la modification des données.

Les déclencheurs sont des objets de base de données qui sont automatiquement activés lorsqu'une action spécifique se produit sur une table.

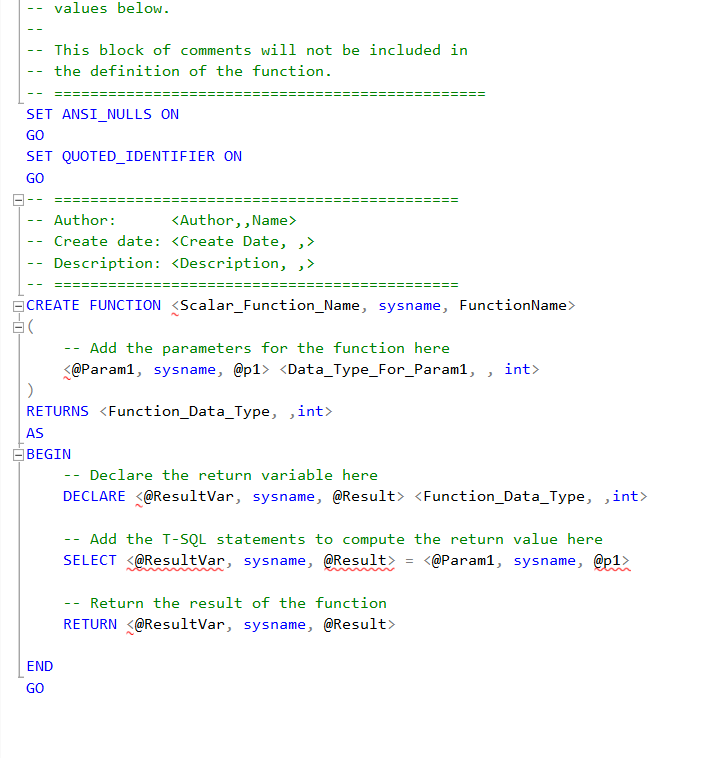
Création Fonction utilisateur :

Une fonction d’utilisateur en SQL est une fonction définie par l’utilisateur qui peut être utilisée pour effectuer des opérations spécifiques sur les données dans une base de données.

Il existe deux types de fonctions d’utilisateur en SQL :

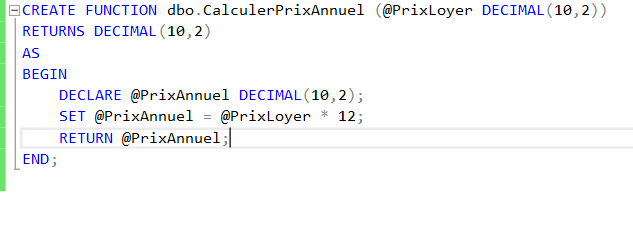
* Fonction Scalaire
* Fonction de table

Ici Faisons une fonction scalaire donc , d’abord il faut aller dans le dossier *Programmabilité -> Fonctions* clic-droit *Nouveau -> Fonctions scalaire :*



[Figure 26 : Fonction scalaire fenêtre]

Donc il ne reste qu’à écrire l’équation.

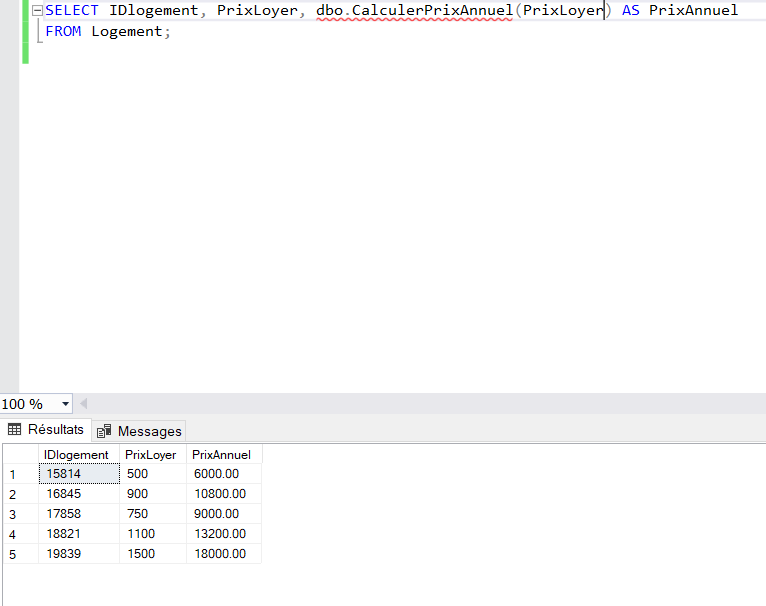


[Figure 27 : Fonction scalaire Calculer Prix Annuel]

Donc cette Fonction scalaire va permettre de *Calculer le Prix Annuel* pour chaque logement.

Ce type de fonction prend un ou plusieurs paramètres en entrée et retourne une valeur unique.

On va tester ça avec une requête :



[Figure 28 : Requête Calculer Prix Annuel pour le logement]

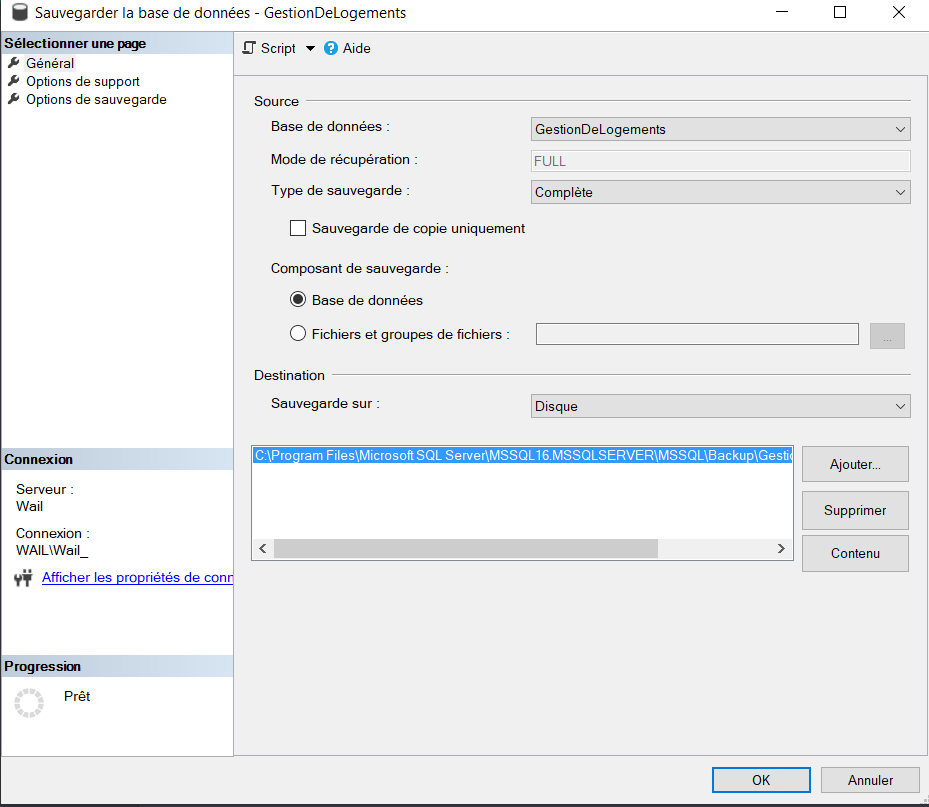
Ici, on a les valeurs du PrixAnnuel.

Création des systèmes de sauvegarde (Backup) et restauration :

Une sauvegarde (Backup) est une copie des données dans une base de données créé dans le but de protéger les données en cas de perte, de panne matérielle, de corruption des données, etc...

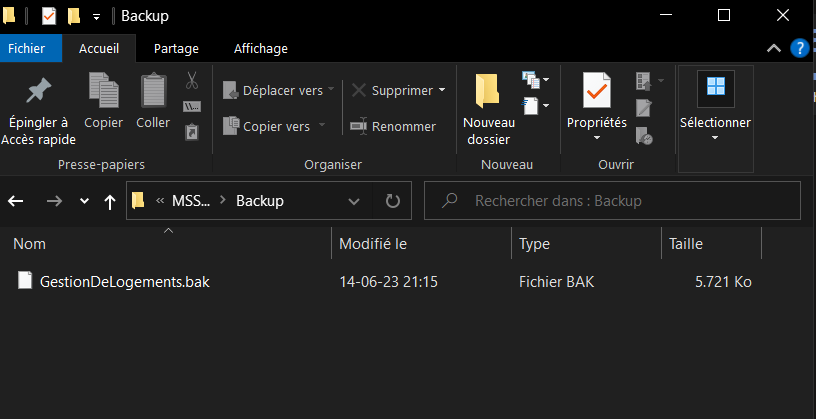
Elaborons une ***sauvegarde (Backup) manuelle***:

Clic-droit sur la base de données *Tâches -> Sauvegarder...* Une page va être affiché voir figure ci-dessous:



[Figure 29 : Fenêtre Sauvegarde de la base de données Manuel]

Donc, il faut choisir la base de données ici c’est *GestionDeLogements. L*e type de sauvegarde ici on va faire *Complète puis* bien sûr le composant de sauvegarde c’est *Base de données,* pour finir la destination donc vers où on sauvegarde, ici c’est dans le disque dur.

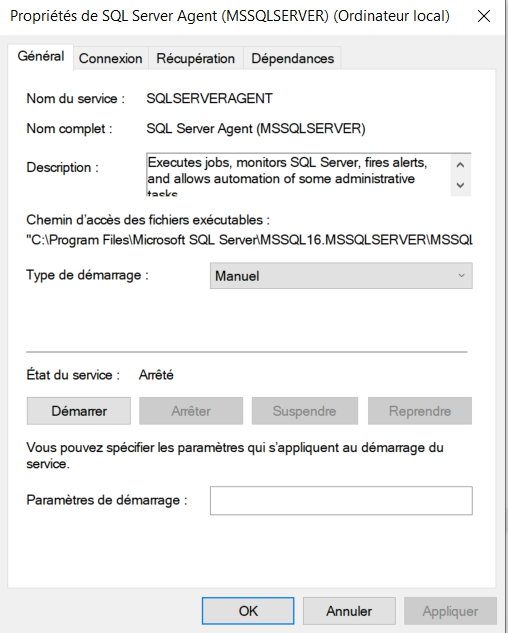


[Figure 30 : Résultat de la sauvegarde manuelle]

Donc la sauvegarde manuelle a été réussie.

Passons à la ***sauvegarde (Backup) automatique*** :

D’abord le SQL Server Agent est désactivé donc pour l’activer faut aller dans les services de la machine.



[Figure 31 : Propriété de SQL Server Agent ]

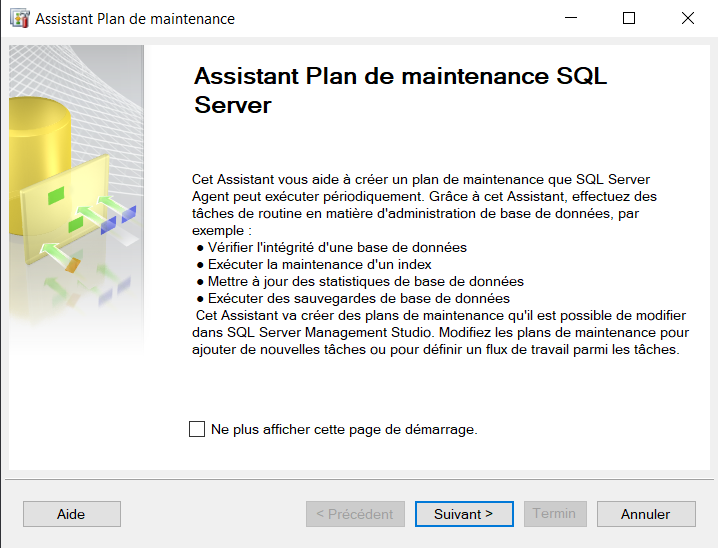
Donc on va le démarrer mais avec du type de démarrage Automatique.



[Figure 32 : Activation de SQL server Agent dans SQL Server Management]

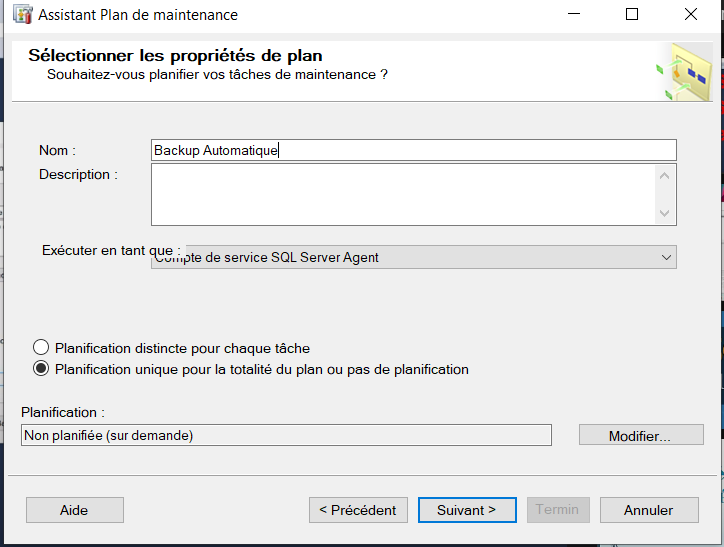
Donc ici on l’a activé on est prêt pour faire une sauvegarde automatique.

Puis, il faut se rédiriger sur le dossier *Gestion ->Plans de maintenance ,* ensuite clic-droit sur ce dossier Assistant Plan de maintenance, il y aura une fenêtre qui s’affiche :



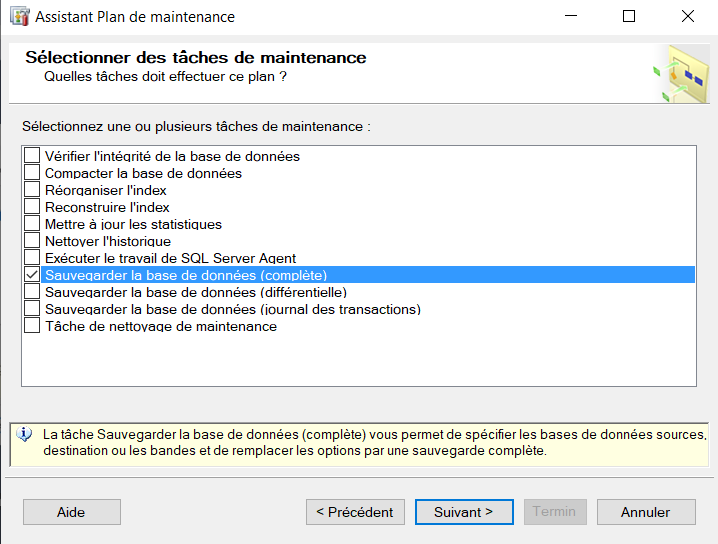
[Figure 33 : Assistant Plan de maintenance SQL Server]

Donc on fait Suivant puis on aura :



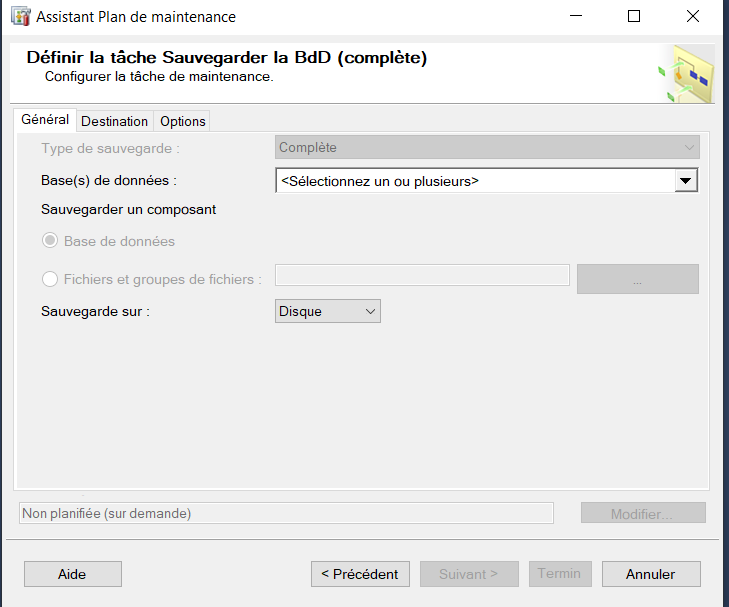
[Figure 34 : Sélectionner les propriétés de plan]

Il faut mettre le Nom, on peut aussi décrire puis on fait suivant :



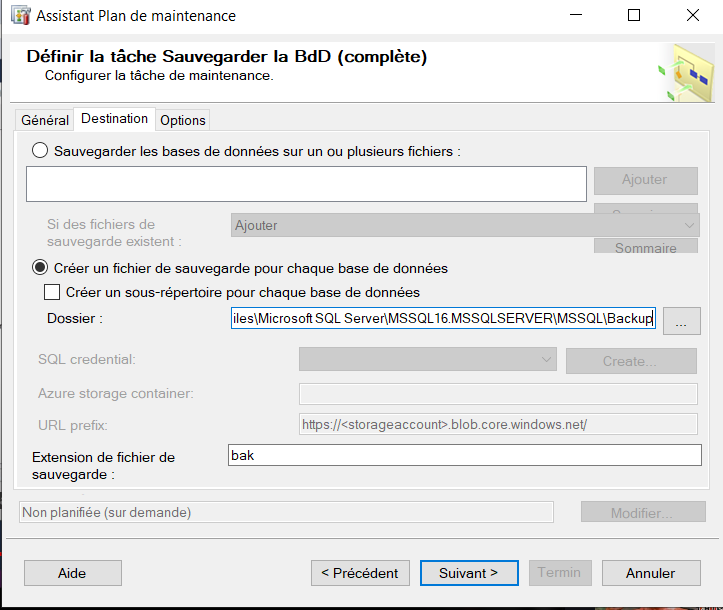
[Figure 35 : Sélectionner des tâches de maintenance]

Ici on sélectionne *Sauvegarder la base de données (complète)* donc avec toute la base de données



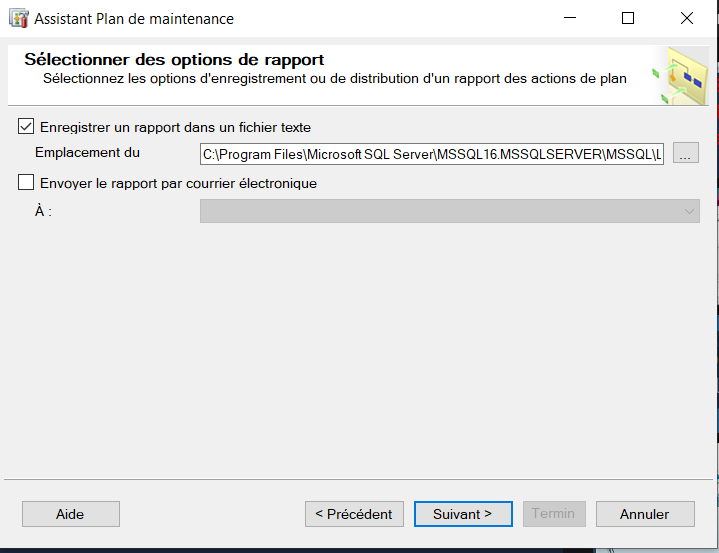
[Figure 36 : Définir la tâche Sauvegarder le BdD (complète)]

Ici on sélectionne la base de données et vers dans quel destination ic c’est dans Disque.



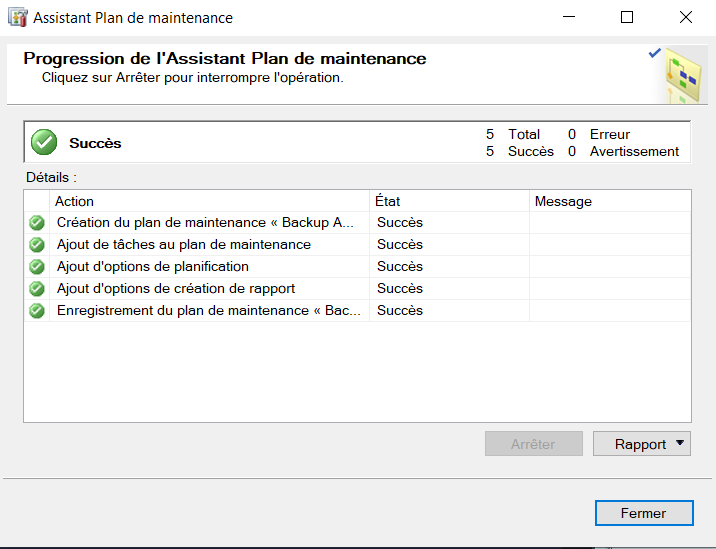
[Figure 37 : Ici c’est dans l’option Destination]

Donc on voit le dossier et aussi son type de dossier “bak”. Puis suivant.



[Figure 38 : Sélectionner des options de rapport]

Ici il y a deux options soit enregistrer un rapport dans un fichier texte ou soit on envoie le rapport par courrier électronique.



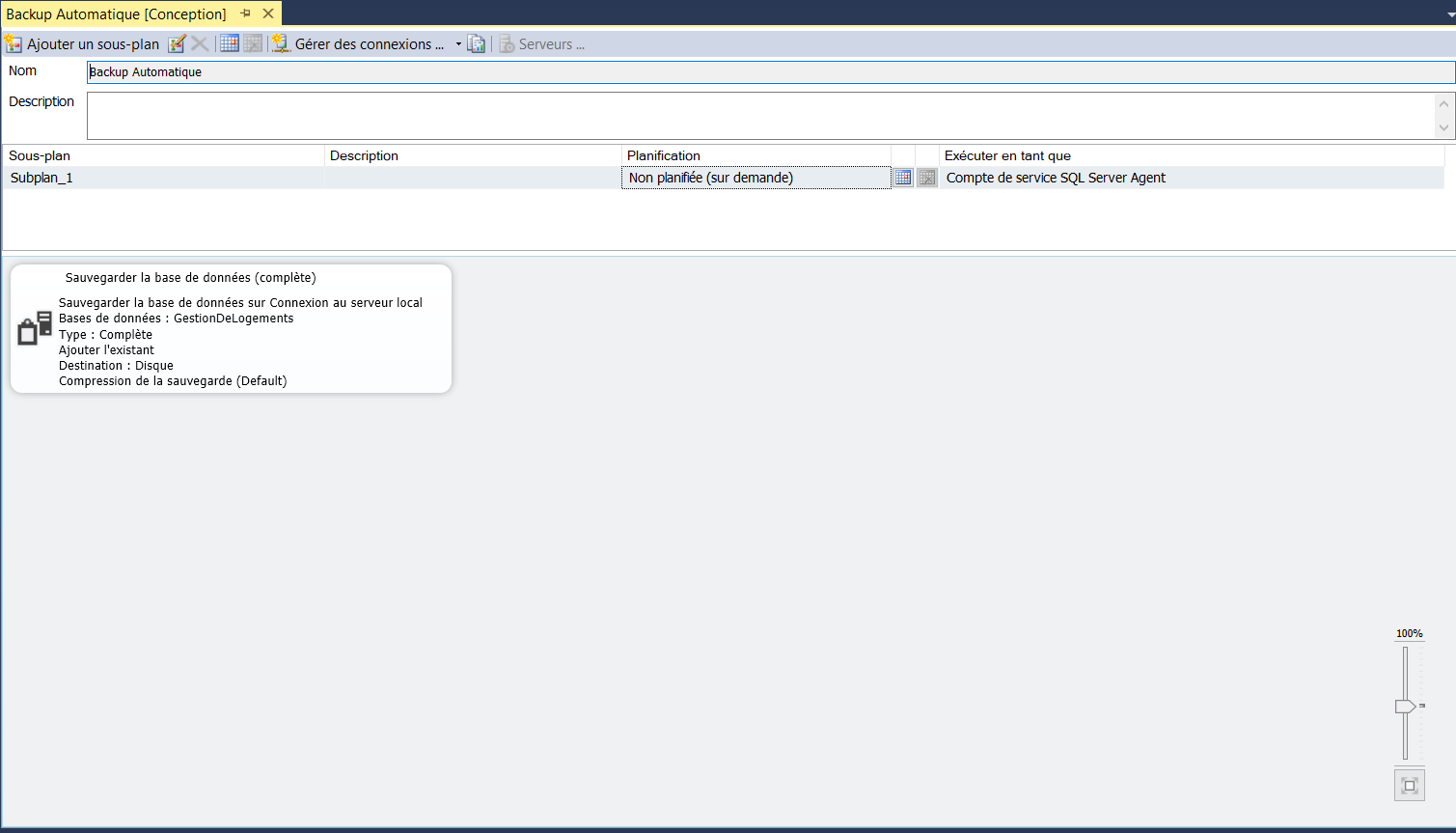
[Figure 39 : Progression de l’Assistant Plan de maintenance]

Ici on enregistre tout ça.



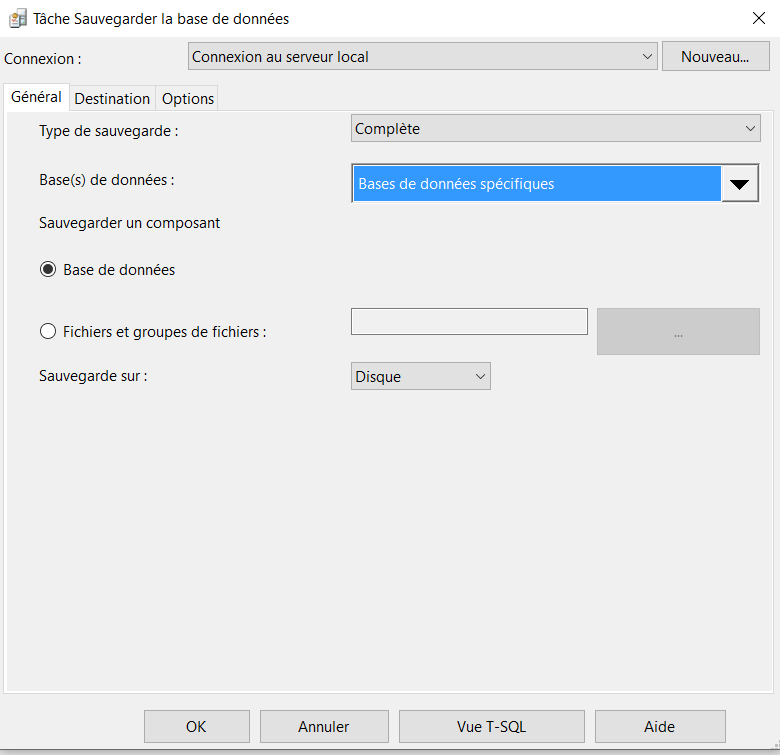
[Figure 40 : Backup Automatique dans le plans de maintenance qui a été ajouté]

On clic dessus, il va s’afficher cette fenêtre ci-dessous.



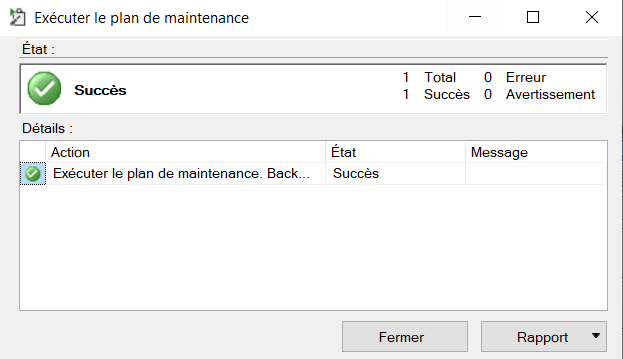
[Figure 41 : Fenêtre de Backup Automatique]

On clique sur la sauvegarder de la base de données sur Connexion au serveur local



[Figure 42 : Tâche Sauvegarder la base de données]

Ici c’est les options enregistrées du tantôt, du coup on clique Ok.



[Figure 43 : Exécution de plan de maintenance]

Le dossier du Backup Automatique a été créé dans le disque.



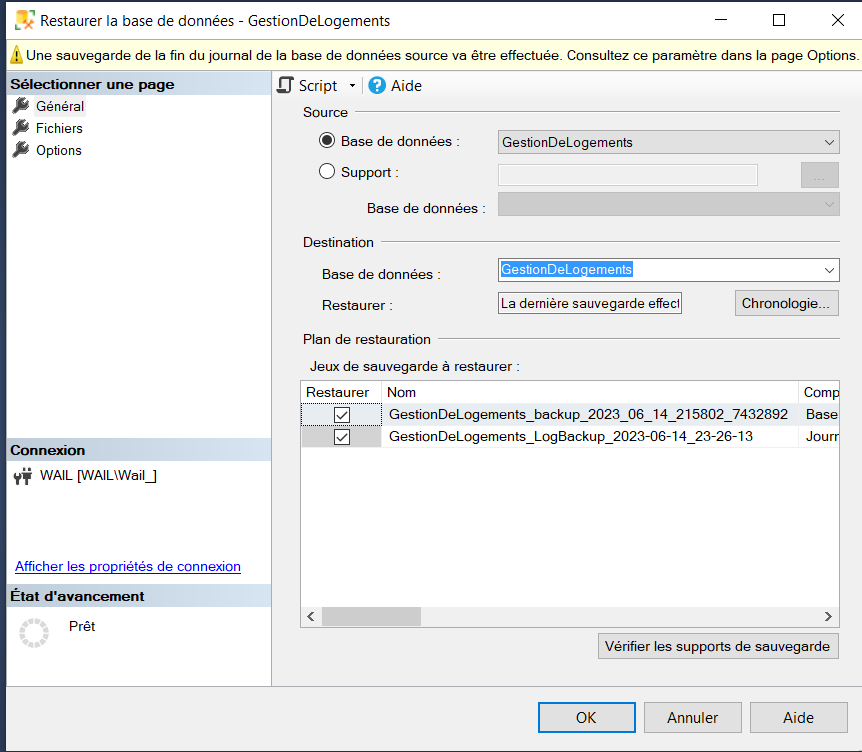
[Figure 44 : Dossier créé du Backup Automatique]

Donc le dossier a été créé de type bak.

Restauration :

Pour restaurer une base de données, il faut aller dans l’explorateur de fichier puis aller dans une base de données clic droit sur la base de données que l’on désire restaurer ->

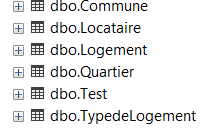
Tâche -> Restaurer -> Base de données puis il y aura apparition d’une fenêtre (voir figure ci-dessous):



[Figure 45 : Restaurer la base de données ]

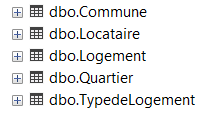
Ici le premier dossier c’est le dernier dossier qui a été créé pour le backup automatique, le deuxième c’est parce que j’ai restauré avec le premier dossier.

On teste la restauration donc on va rajoute une table dans la base de données:



[Figure 46 : Création de table Test]

On va tester la restauration

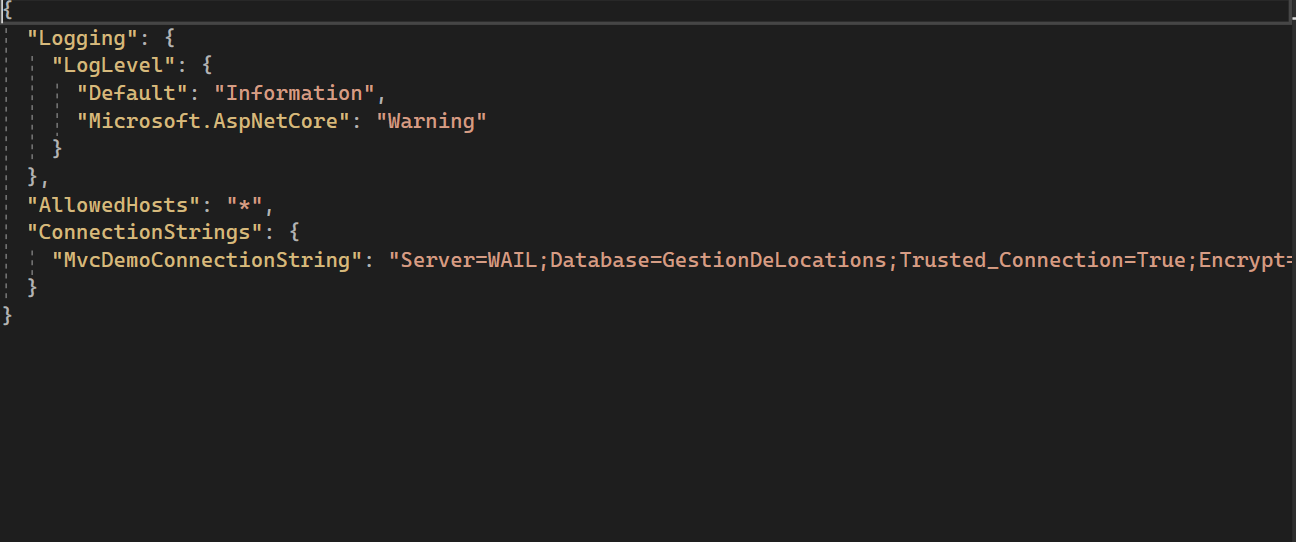


[Figure 47 : Restauration réussie]

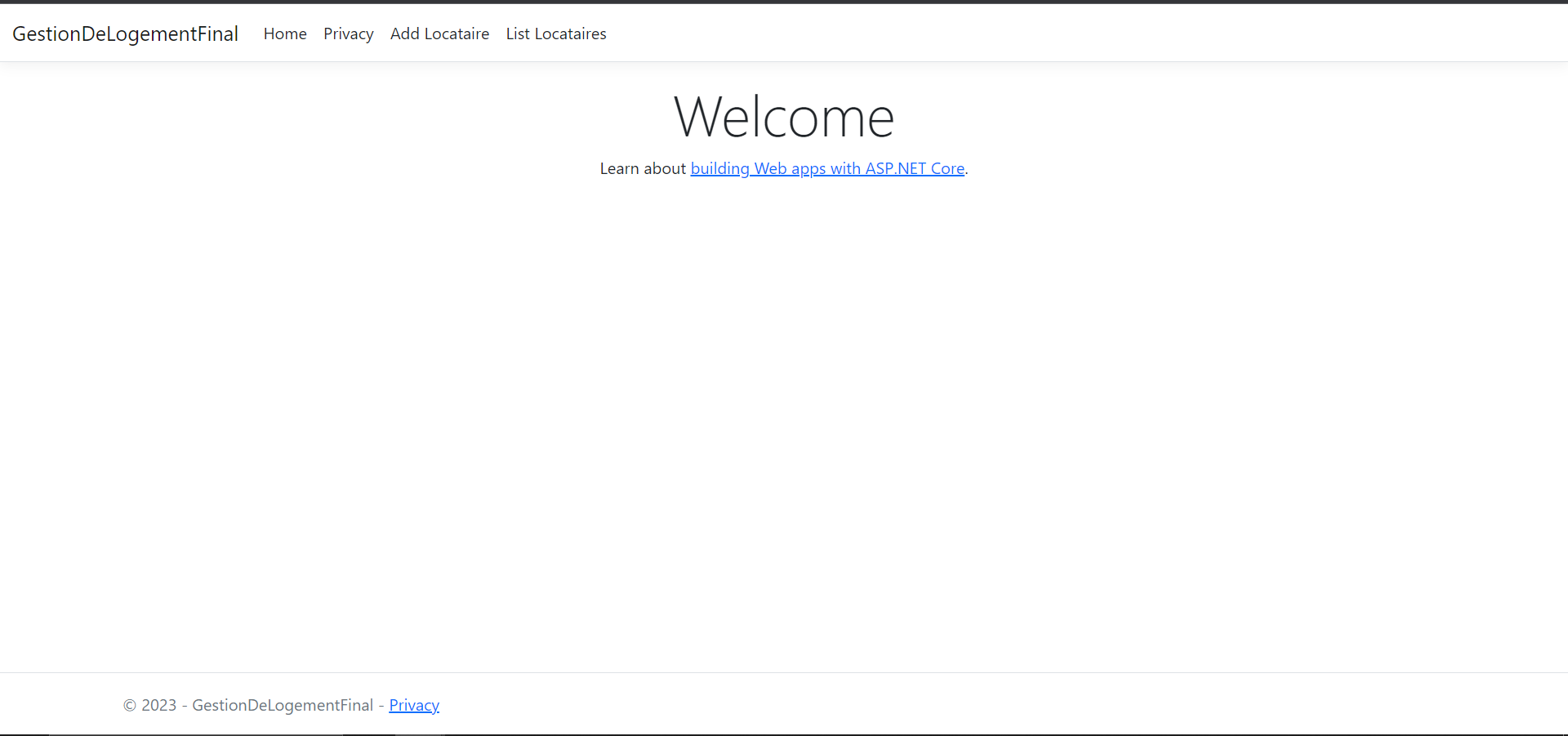
La restauration a été bien effectué.

Gestion des données à partir d’une page web (asp) :

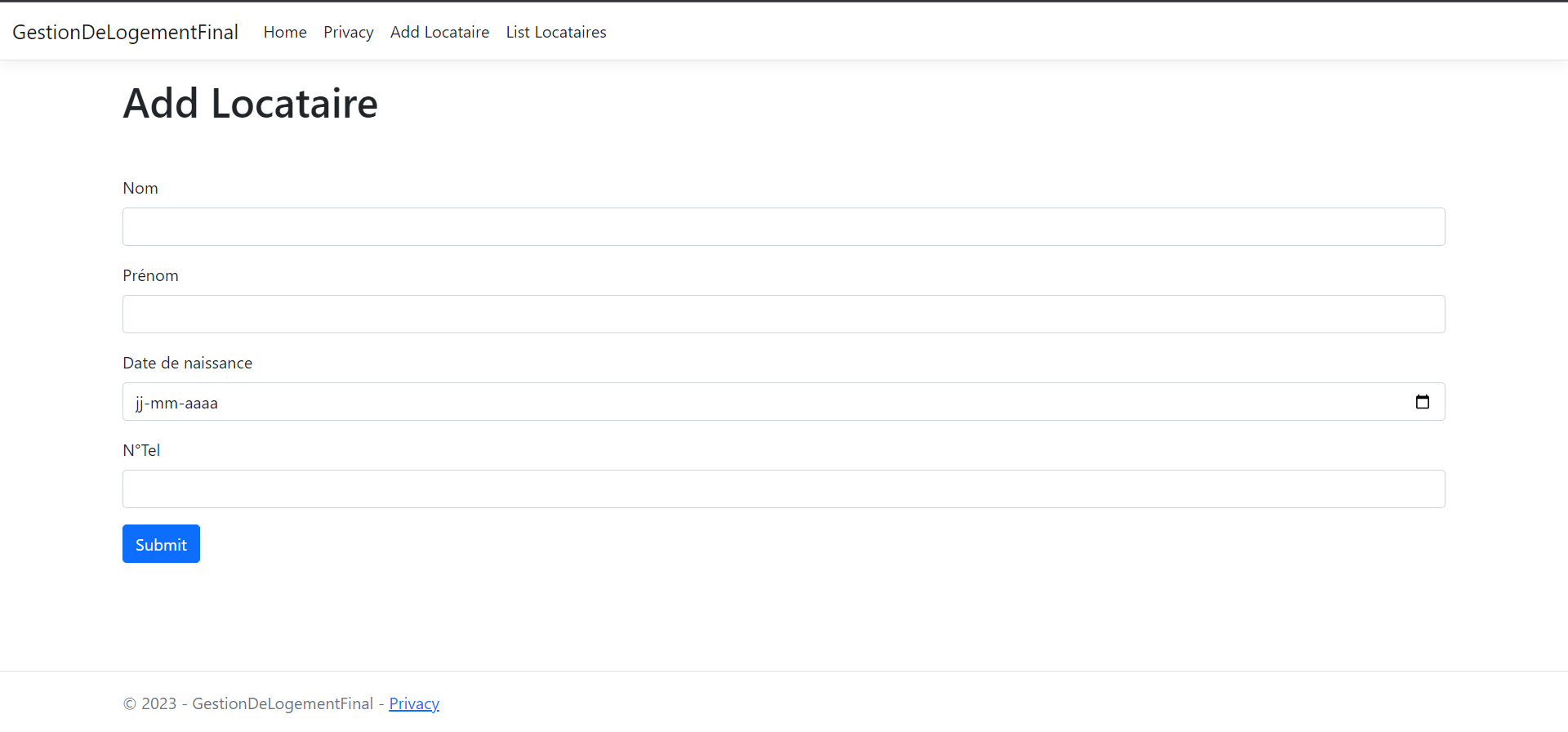
Ici j’ai utilisé visual studio 2022 j’ai utilisé une topologie MVC pour faire le programme j’ai dû faire une connexion avec sql server:



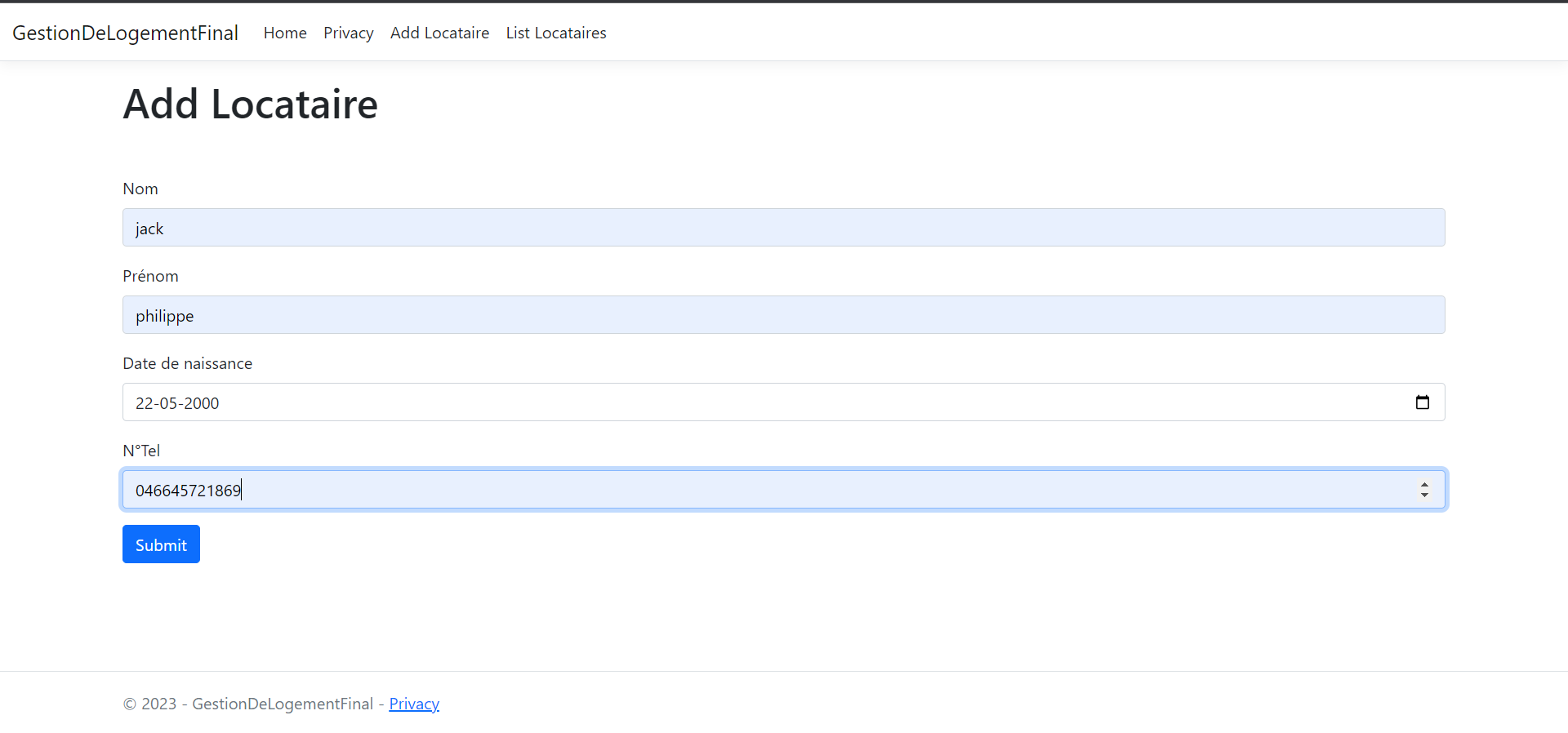
Puis le reste c’est MVC.



Ici on est dans la page d’accueil et du coup on va essayer d’ajouter un locataire du coup on va cliquer dans Add Locataire

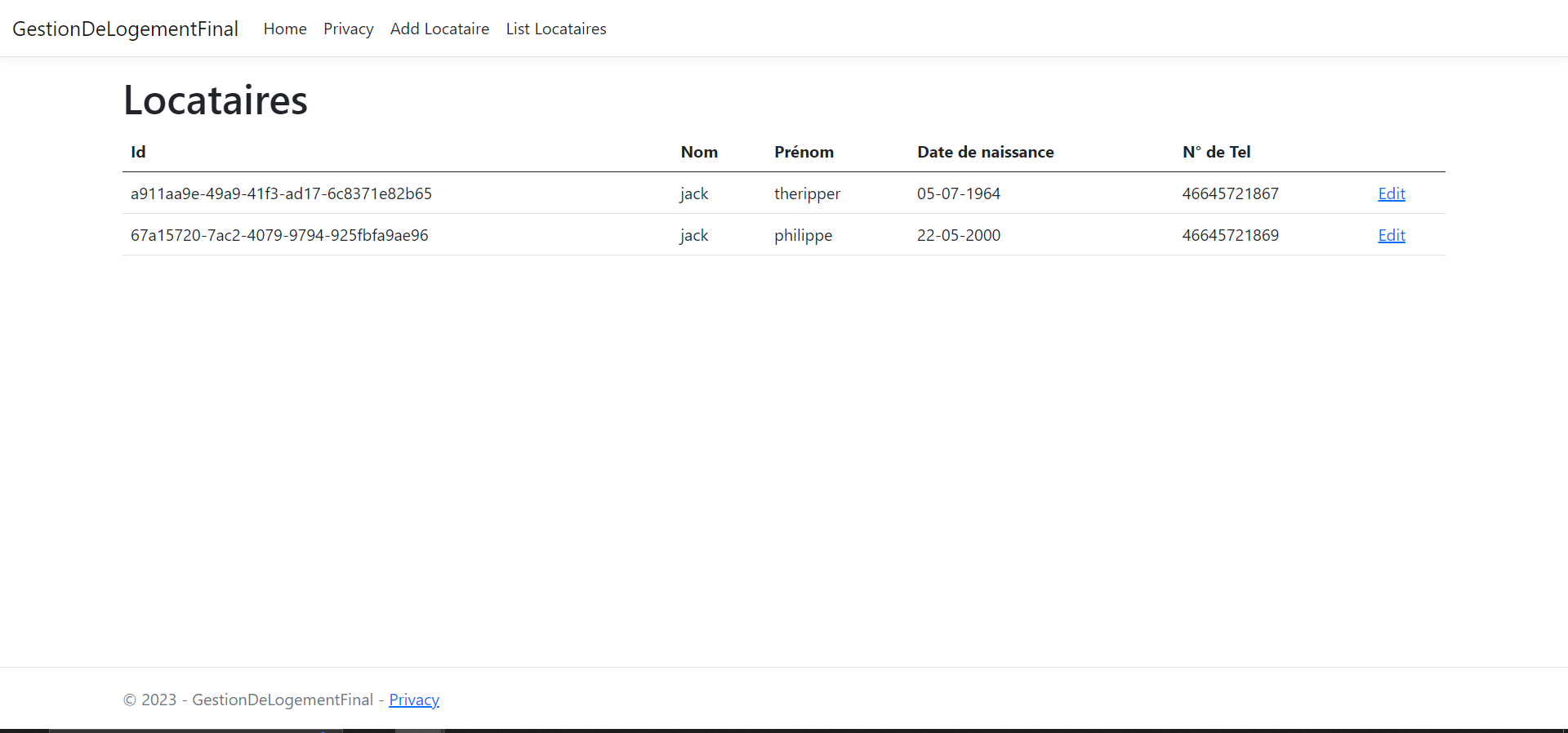


On aura ceci on va ajouter du coup un locataire (create) :

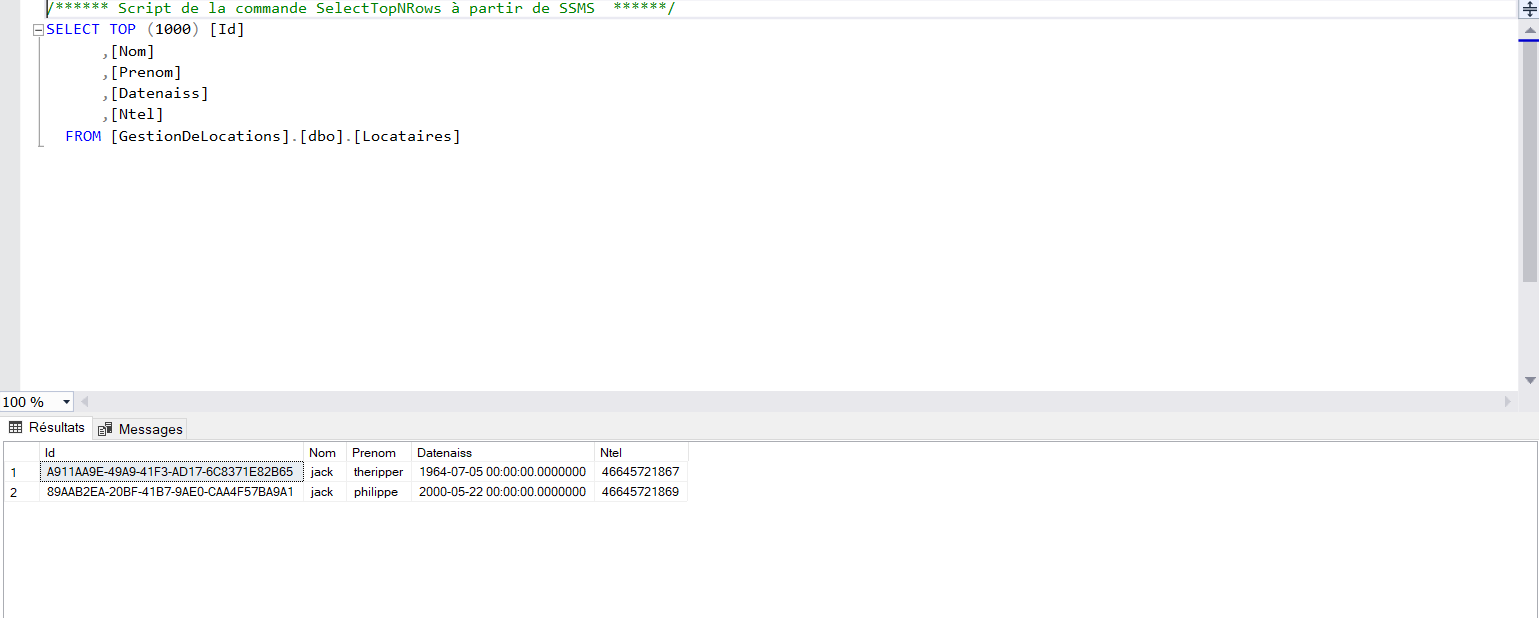


Exemple comme celui-ci et puis on clique Submit

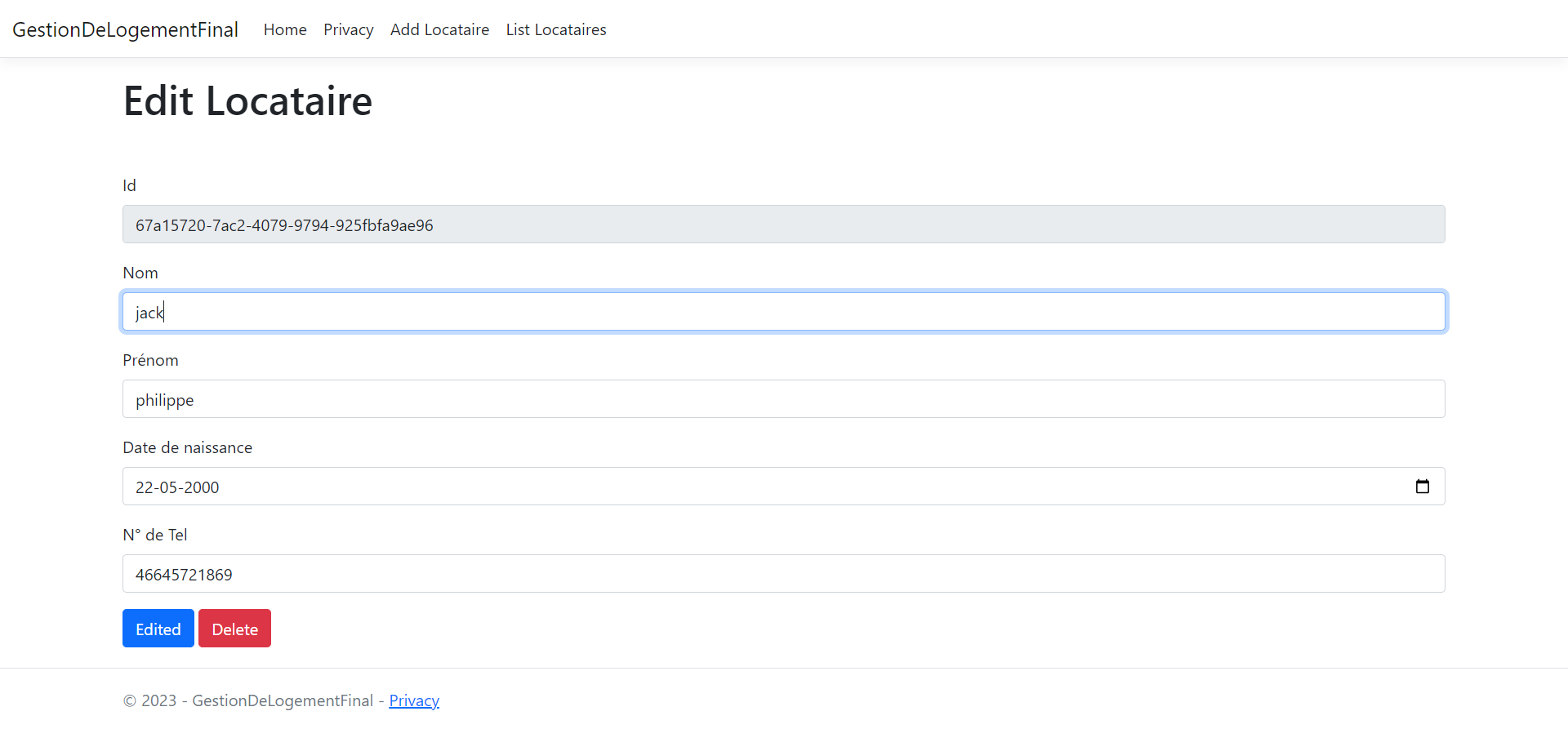
Et on se rend dans la List Locataire



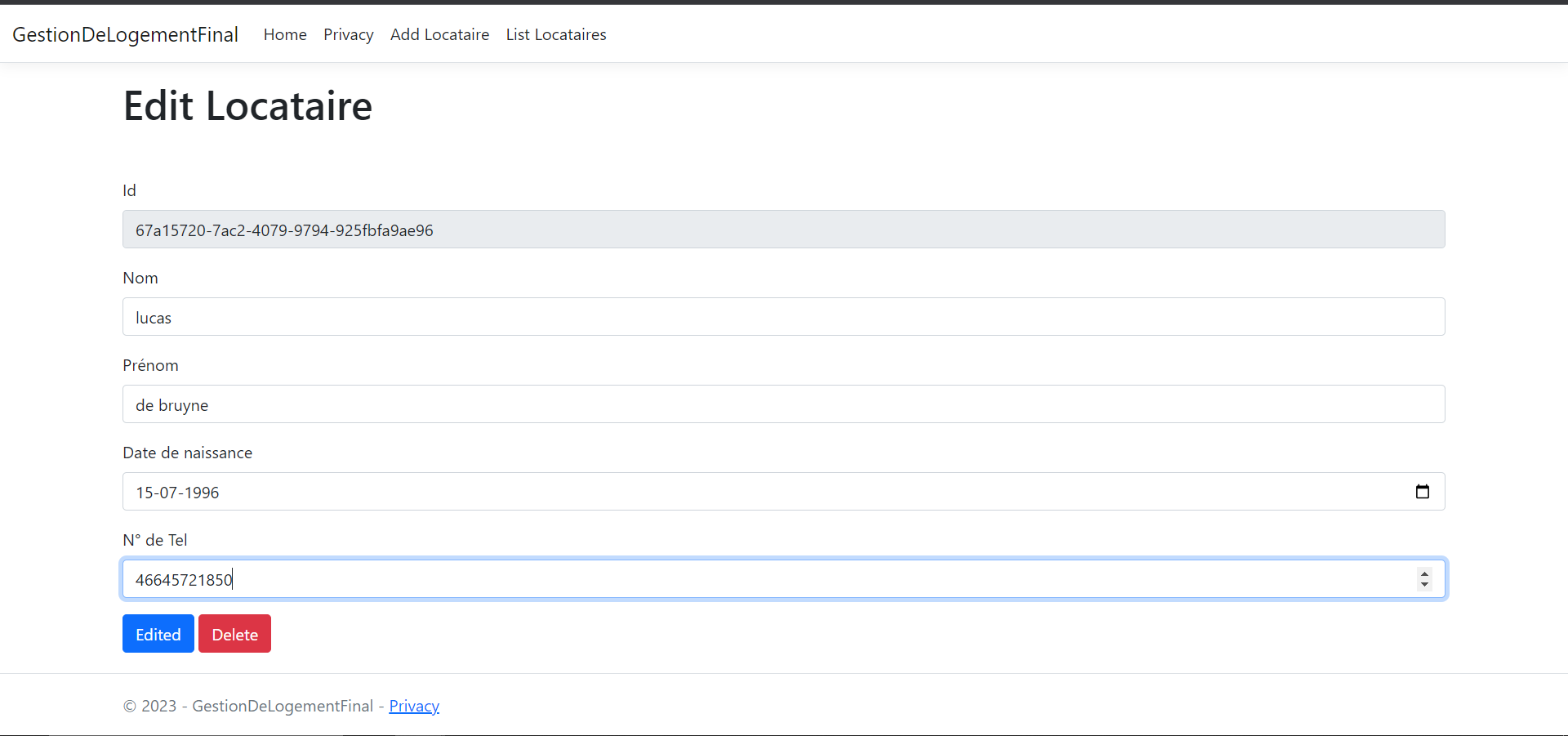
Ca a été bien ajouté mais on doit vérifier si ca s’est ajouté dans sql server



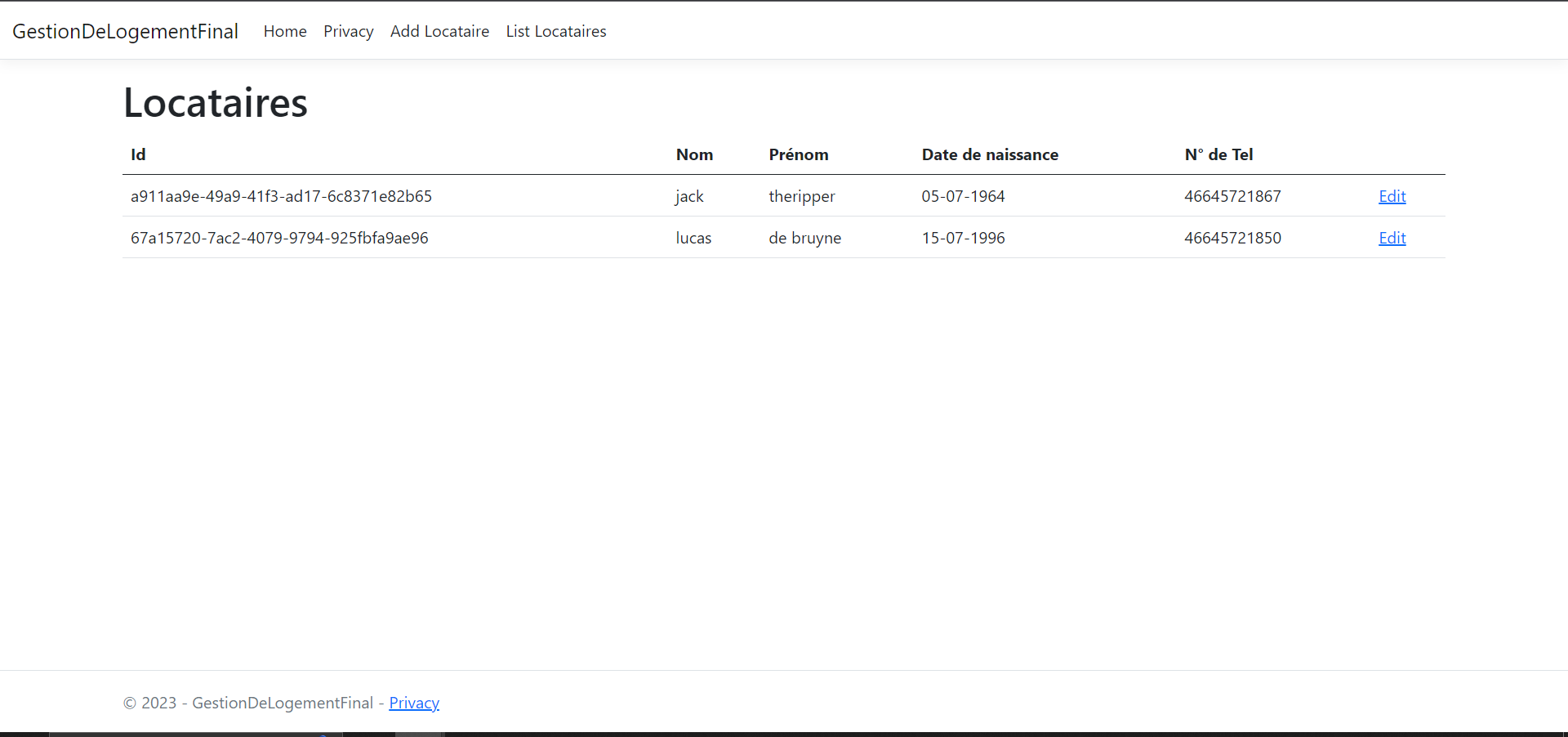
Oui ça s’est bien ajouté du coup on va passez à la suite on va essayer de l’edit



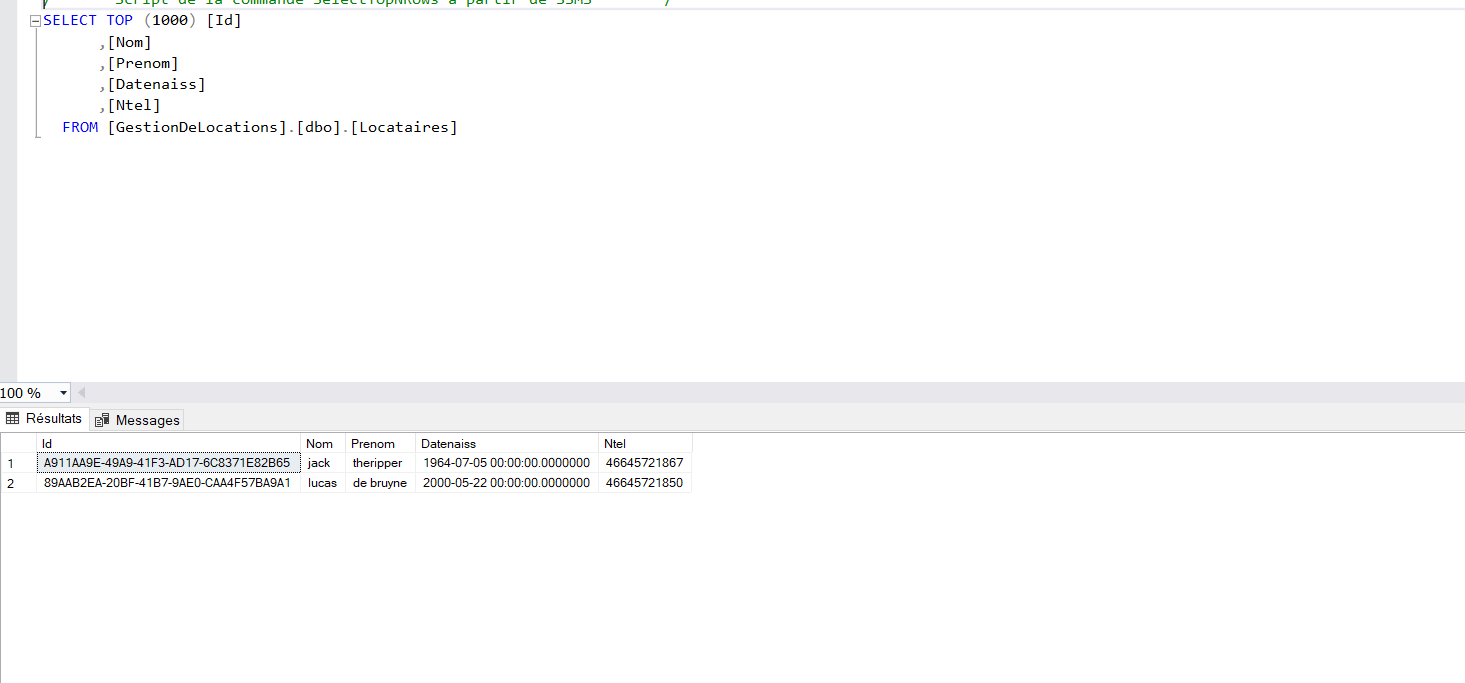
On va modifier tout pour voir si ca marche



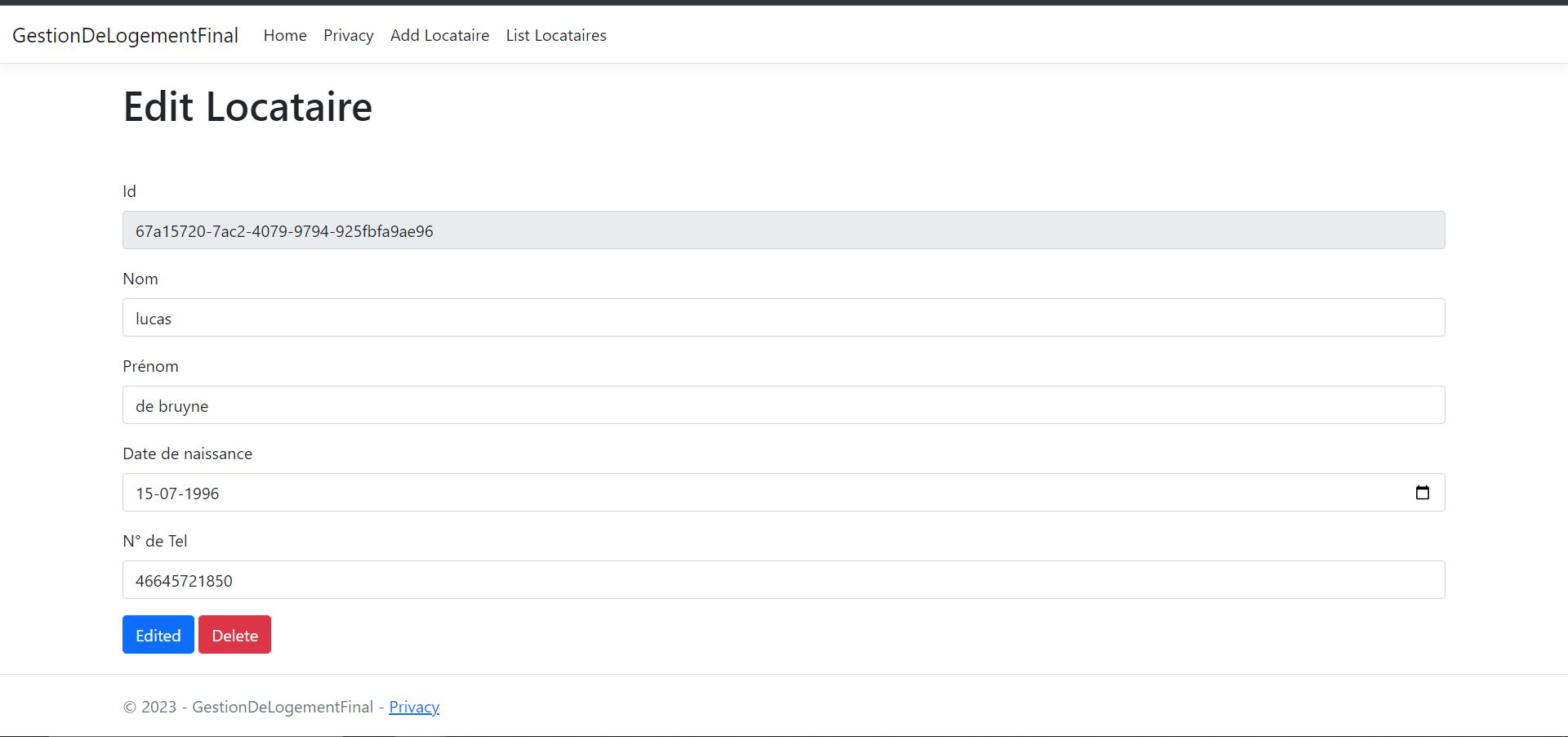
Là on a modifié on clique sur edited



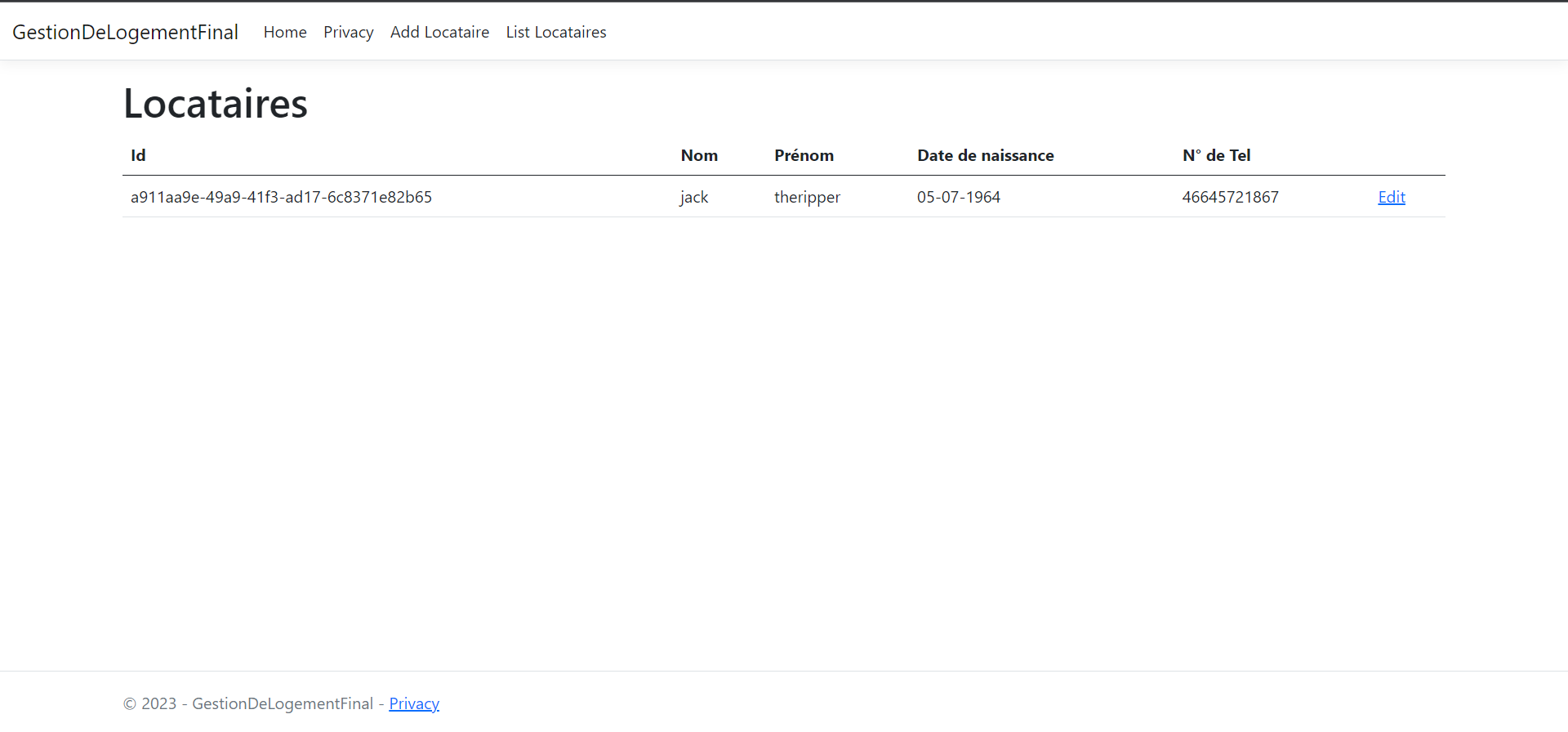
Ça a bien changé on va vérifier dans la base de données aussi

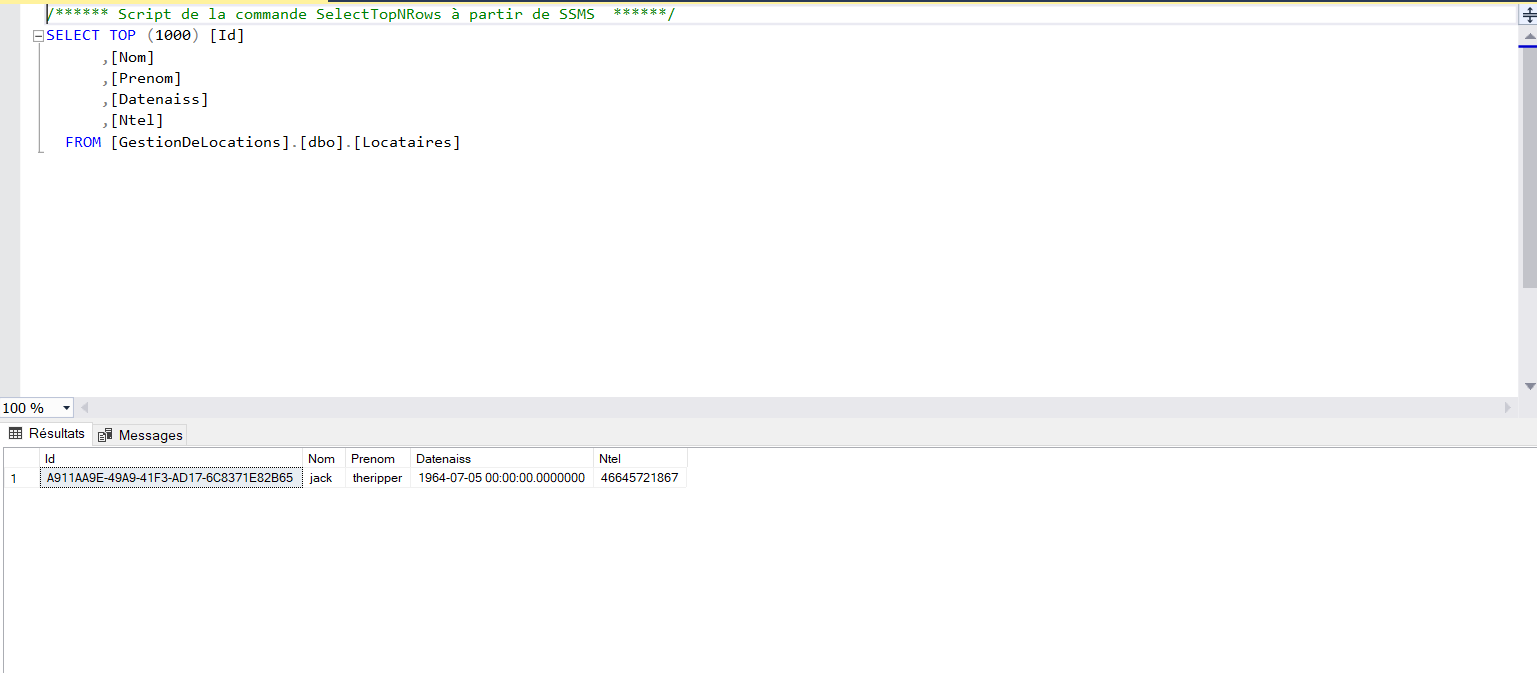


Ok ici c’est bien modifié du coup passons là au delete donc on va supprimer le locataire



Ici on clique dans delete





On l’aura plus ni dans l’interface et ni dans la base de données voila ça s’est bien effacé.