



INSTITUT SUPÉRIEUR
INDUSTRIEL DE BRUXELLES
I.S.I.B.

ORIENTATION GÉNIE INDUSTRIEL
EN INFORMATIQUE

MA1

Rapport de laboratoire
4IN0402 Systèmes de gestion de bases de données

Auteur :
HYPOLITE IRANKUNDA

Professeur :
MR S. RODRIGUES

Année académique 2020 — 2021

Table des figures

1	Téléchargement de SQL server 2019	1
2	Centre d'installation SQL Server	1
3	Installation de SQL Server Manager	2
4	Interface Microsoft SSMS	2
5	MLD	3
6	Création d'une base de donnée à l'aide de l'interface	4
7	Création de la base de donnée depuis un script	5
8	Diagramme base de donnée	5
9	TDU interface	6
10	Choix du TDU	6
11	Procédure	8
12	Trigger	8
13	Résultat trigger	9
14	Fonction	9
15	Sauvegarde manuel	10
16	Sauvegarde manuel successfully	11
17	Backup restored	11

Liste des tableaux

Table des matières

1	Introduction	1
2	Installation de SQL server	1
3	Modèle logique de la base de donnée	3
4	Création de types de données définis par l'utilisateur (TDU)	6
5	Définition de contraintes	7
6	Procédures stockées (traitement sur les données) / déclencheurs (Triggers)	7
6.1	Procédures stockées	7
6.2	Triggers	8

7 Définition fonction utilisateur **9**

8 Sauvegarde (manuel / automatique) / restauration **10**

8.1 Manuel 10

8.2 Manuel 10

8.3 Automatique 10

8.4 Restauration 10

Bibliographie **11**

1 Introduction

2 Installation de SQL server

La première étapes consiste à installer SQL serveur en téléchargeant le fichier ".exe" sut le site de Microsoft (<https://www.microsoft.com/en-us/sql-server/sql-server-downloads>) comme on peut le voir sur la figure 1. Le fichier ".exe" va installer sur l'ordinateur le Centre d'installation SQL Server comme on peut le voir sur la figure 2. Il suffit ensuite d'aller dans la partie installation et sélectionner une nouvelle installation autonome de SQL Server et de suivre et sélectionner ses préférence pendant l'installation automatique.

Or, download a free specialized edition



FIGURE 1 – Téléchargement de SQL server 2019

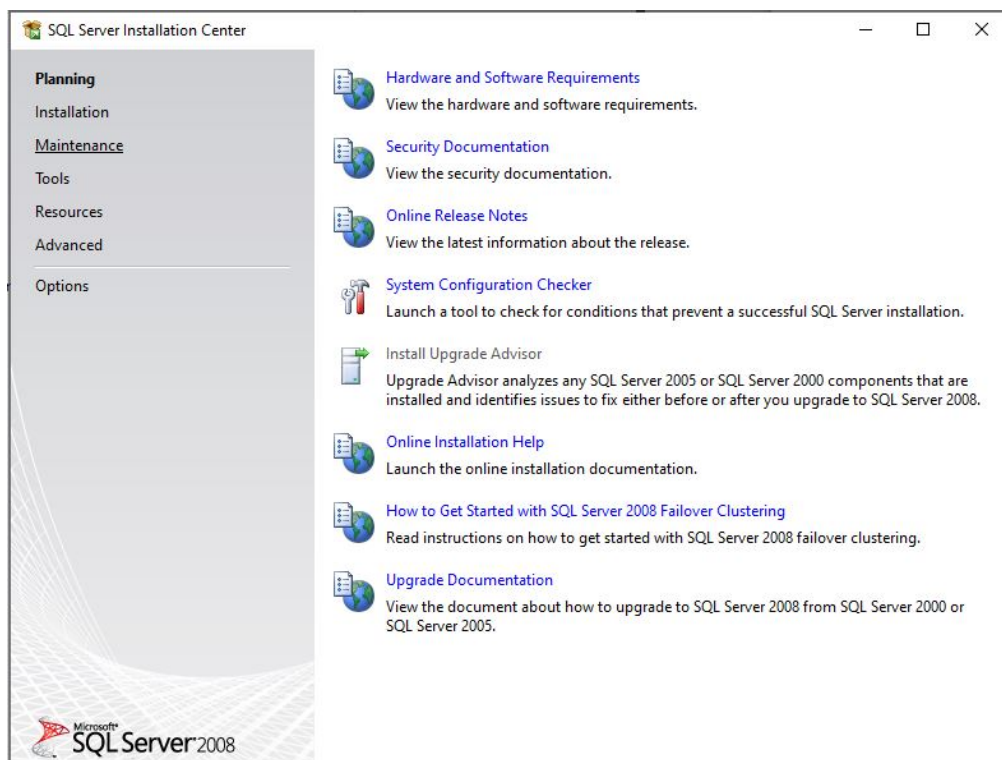


FIGURE 2 – Centre d'installation SQL Server

Pour Faciliter l'utilisation de SQL serveur et l'administration de la base de donnée il va falloir télécharger SQL Server Management Studio (SSMS) comme indiqué sur la figure 3. La figure 4 présente l'interface du manager studio avec une interface Object Explorer (explorateur d'objet) dans lequel on peut remarquer le nom du serveur de base de donnée **DESKTOP-A79...** la présence de certaine base de donnée préalablement créés.

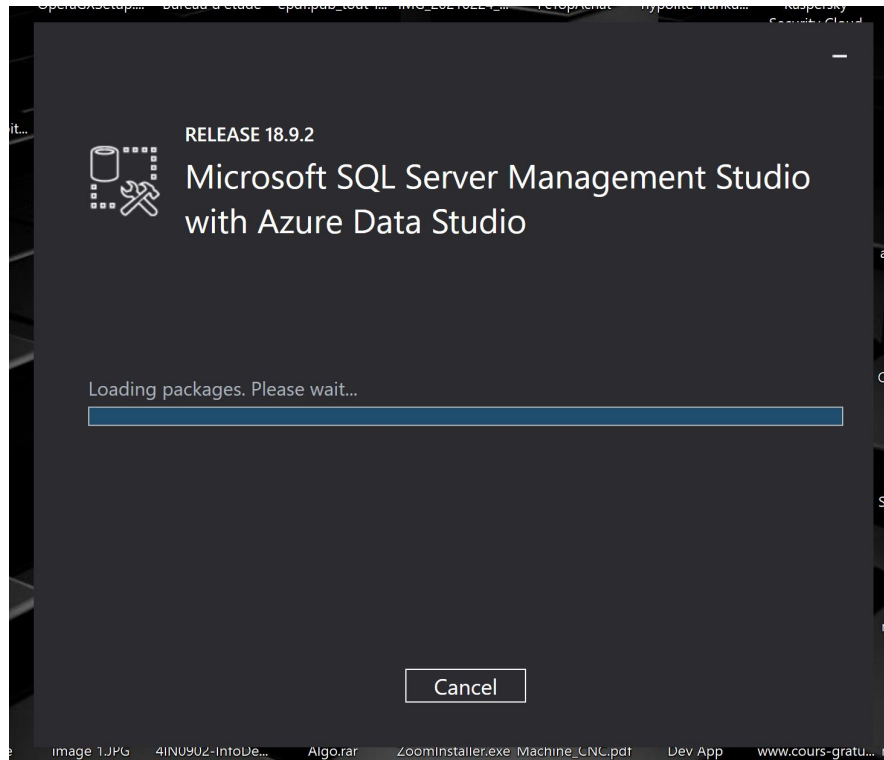


FIGURE 3 – Installation de SQL Server Manager

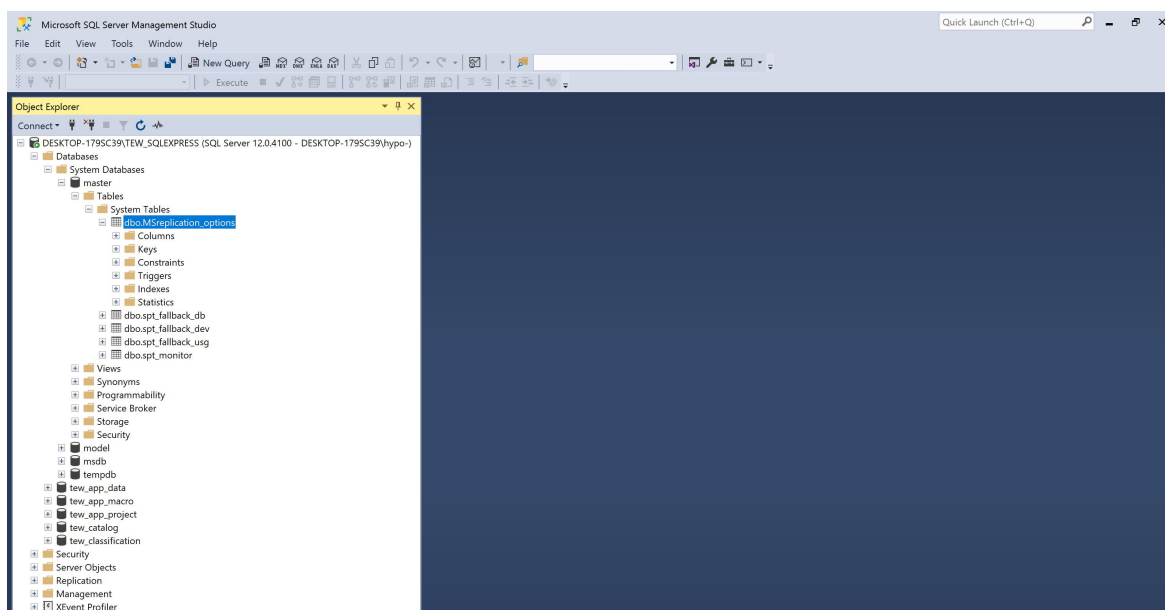


FIGURE 4 – Interface Microsoft SSMS

3 Modèle logique de la base de donnée

Le système choisi pour ce projet est un regroupement de magasins d'article d'occasion, jeuxvidéos et livres, qui utilise un système similaire à celui du magasin Pêlé-Mêle. La figure 5 présente le modèle logique (MLD) de la base de donnée construit directement sur Win Design.

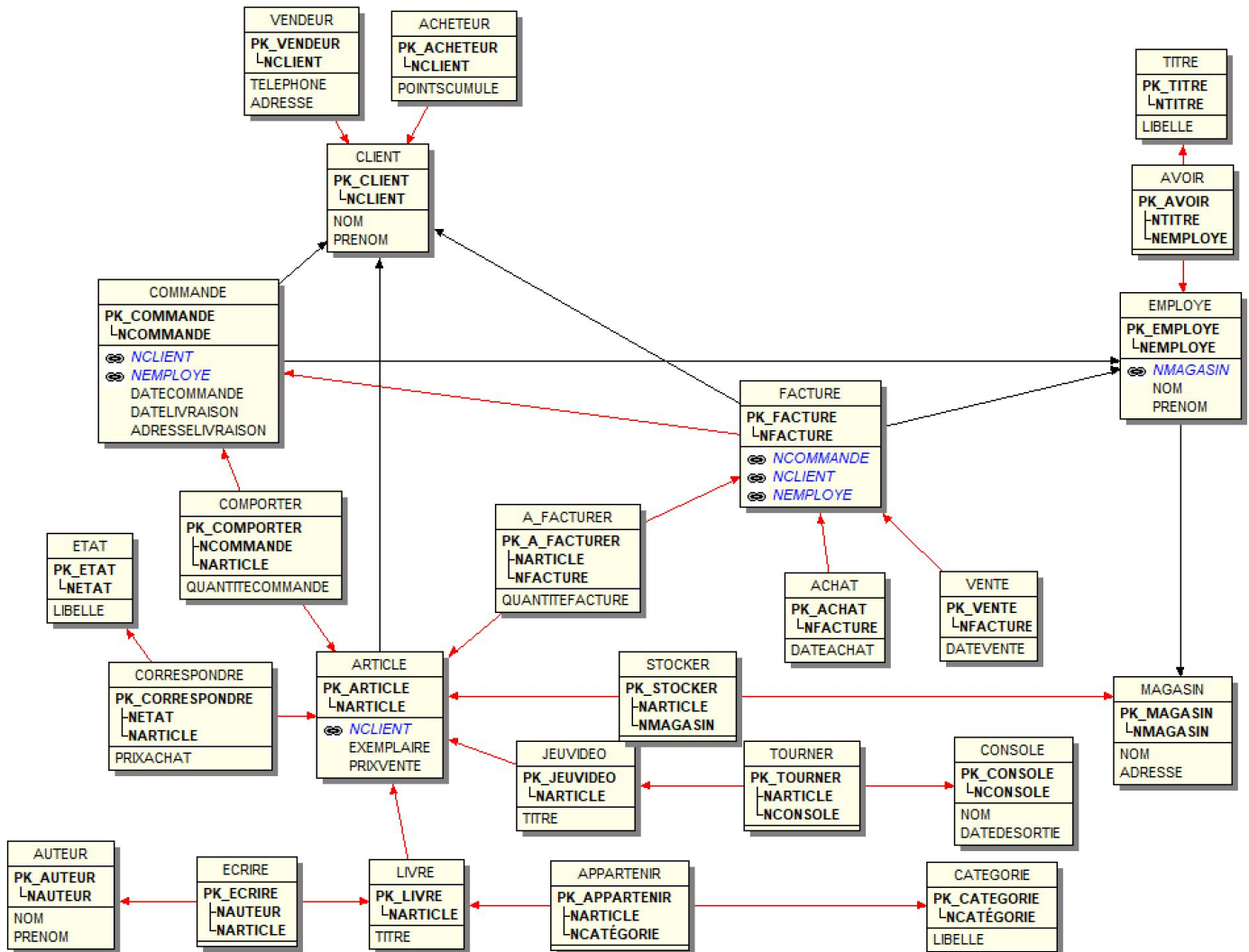


FIGURE 5 – MLD

Il est possible de créer une base de donnée directement depuis l'interface en effectuant un clic droit sur Databases et en sélectionnant New Database ce qui fait apparaître la fenêtre New Data Base qu'on peut voir sur la figure 6. Depuis cette fenêtre on peut spécifier le nom de la base de donnée le propriétaire et aussi la taille en mémoire de la base de donnée.

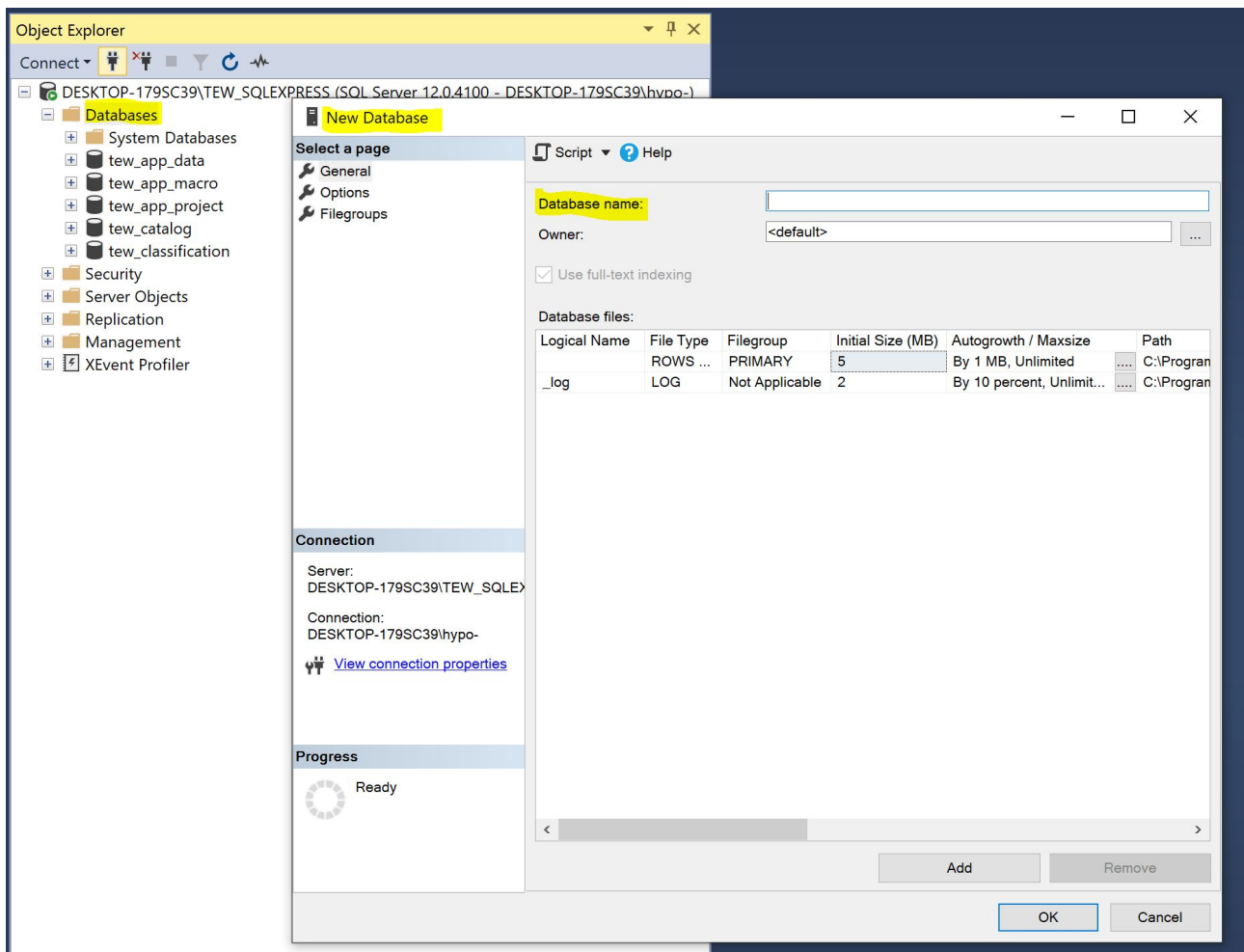


FIGURE 6 – Création d'une base de donnée à l'aide de l'interface

On peut aussi générer la création de la base de donnée directement depuis un script comme montré à la figure 7. Ensuite SQL serveur permet de créer une représentation graphique de la base de donnée par le biais d'un diagramme comme on peut le voir sur la figure 8.

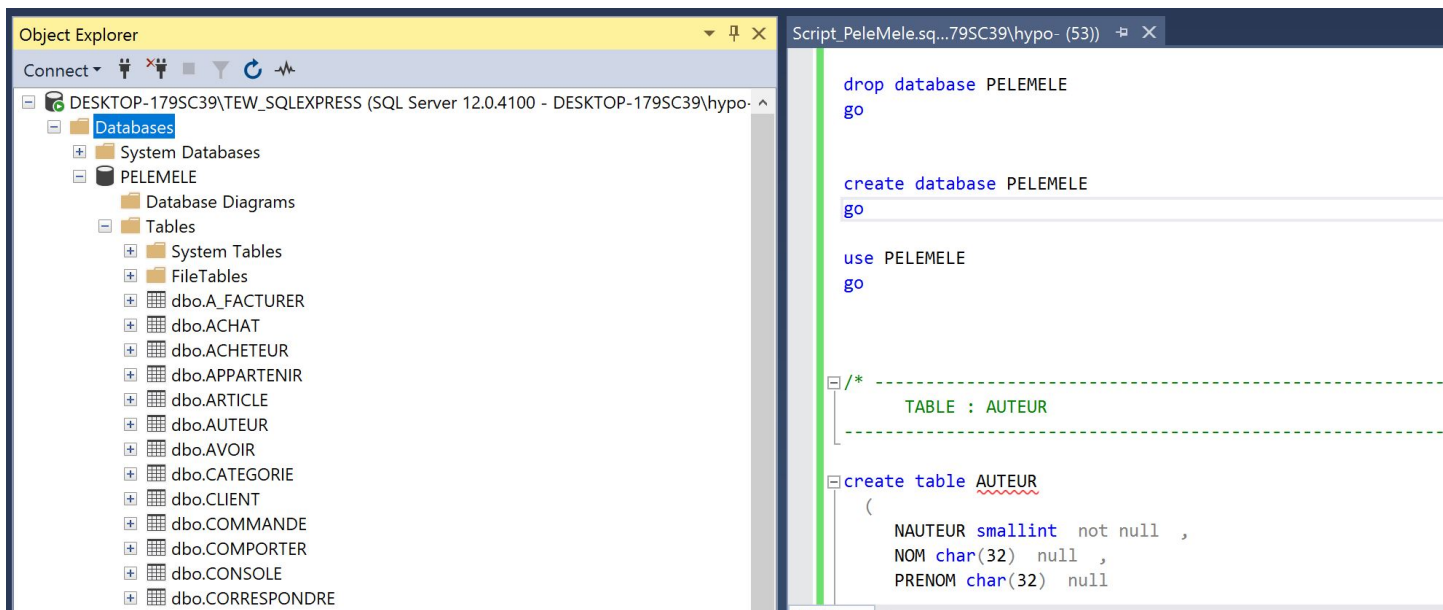


FIGURE 7 – Création de la base de donnée depuis un script

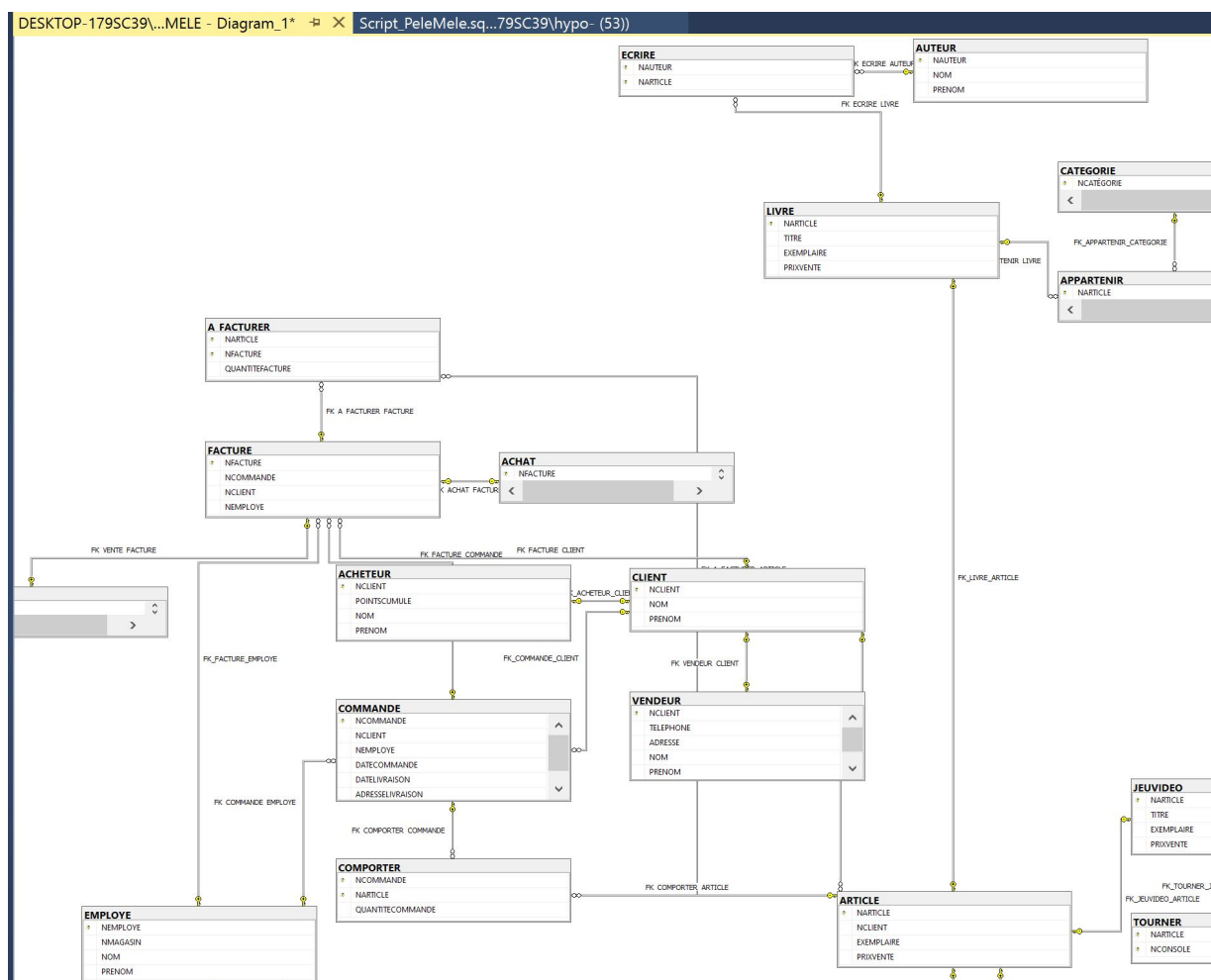


FIGURE 8 – Diagramme base de donnée

4 Création de types de données définis par l'utilisateur (TDU)

Un TDU est un type de données qui dérive d'un type de données existant. On les utilise pour étendre les types déjà disponible et créer nos propres types de données personnalisés. Dans notre exemple nous allons créer un TDU pour les numéros de téléphone car ceux ci ont toujours la même longueur, on permet ainsi un contrôle plus efficaces des numéros de téléphone.

La création d'un TDU peut se faire via l'interface de SQL Server Management. En déroulant une base de données et en allant ensuite dans Programmabilité -> Types -> Types de données définis par l'utilisateur comme on peut le voir sur la figure 9.

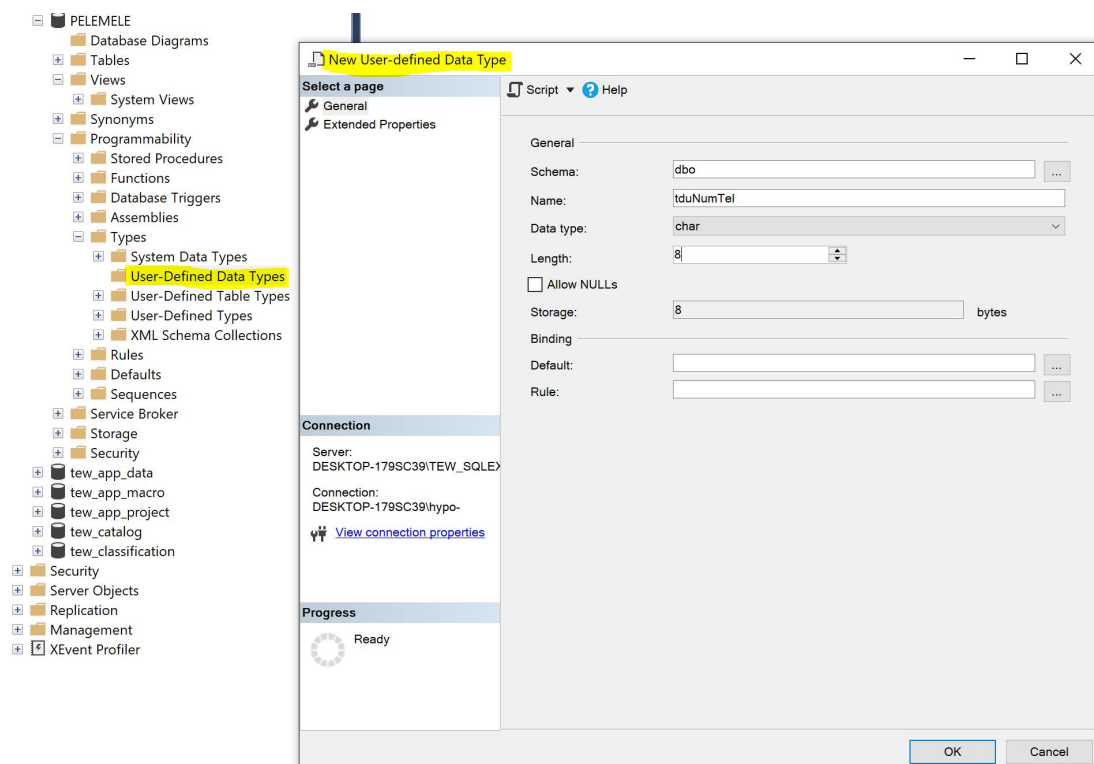


FIGURE 9 – TDU interface

Il est aussi possible d'effectuer cette action avec une requête SQL :

Create type tduNumTel2 from char(9)

La figure 10 montre la possibilité d'affecter notre nouveau type à la colonne d'une table.

DESKTOP-179SC39\...EMELE - dbo.Hypo			
	Column Name	Data Type	Allow Nulls
▼	NHypo	tduNumTel3:char(9)	<input type="checkbox"/>
			<input type="checkbox"/>

FIGURE 10 – Choix du TDU

5 Définition de contraintes

Les contraintes permettent d'implémenter des règles logiques et cohérentes aux colonnes de données d'une table pour empêcher l'utilisateur d'effectuer des enregistrements qui pourraient nuire à la logique de la base de données.

Il existe différents types de contraintes et ceux utilisés dans ce projet sont les suivants :

- Clés primaires : chaque table doit disposer d'au moins une clé primaire et les clés primaires dans une table sont uniques.
- Clés secondaires : les contraintes de clés secondaires (étrangères) permettent de référencer en majorité d'autres clés primaires ou d'autres colonnes d'une autre table. Les valeurs possibles de ces clés étrangères dépendent des valeurs existantes de la colonne qu'il référence.
- NOT NULL : cette contrainte oblige l'utilisateur à entrer une valeur pour un champ d'un attribut. Dans notre projet cette contrainte est appliquée au client-vendeur qui doit obligatoirement avoir un nom, prénom, numéro de téléphone et une adresse en cas de problème avec la vente effectuée.
- CHECK : La contrainte CHECK active une condition permettant de vérifier la valeur saisie dans un enregistrement. Si la condition est évaluée à false, l'enregistrement viole la contrainte et n'est pas entré dans la table [1]. Dans notre projet on utilise cette contrainte pour limiter le contrôle des points cumulés des clients-Acheteurs qui ne doivent pas dépasser les 100 points avec la requête :

```
PointCumule INT NOT NULL CHECK (PointCumule <= 100)
```
- Auto_incrémenter : Permet d'incrémenter de 1 un champ à chaque fois que sa ligne est remplie. Dans le projet il est utilisé pour les clés primaires numériques pour qu'à chaque fois qu'on crée une occurrence d'une table la valeur de sa clé primaire est automatiquement associée en étant l'incrément des clés primaires de l'occurrence de cette table précédemment créée.

6 Procédures stockées (traitement sur les données) / déclencheurs (Triggers)

6.1 Procédures stockées

Une procédure stockée est un bout de code qui permet d'effectuer un traitement sur la base de données et qui est enregistré sur le serveur. Les applications qui sont clientes de la base de données peuvent les appeler et s'en servir.

La procédure créée, à la figure 11, permet de donner le nombre d'employés en fonction du magasin

```

create proc cointEmployeFrom(@a varchar(20))
as
begin
Select count(*) as "Nombre d'employe" FROM EMPLOYE
INNER JOIN MAGASIN on EMPLOYE.NMAGASIN = MAGASIN.NMAGASIN
where MAGASIN.nom = @a

end

exec cointEmployeFrom "PeleMele Ixelles"

```

100 %

Results Messages

	Nombre d'employe
1	2

FIGURE 11 – Procédure

6.2 Triggers

Un trigger, également appelé déclencheur, permet d'exécuter un ensemble d'instruction SQL juste après un événement [2]. Le trigger de la figure 12 permet de créer un message stocké dans la table tabMessage à chaque création d'un employe, ce type de système peut être utilisé pour créer des messages d'erreur ou de update dans la table avec des variable correspondante.

```

alter TRIGGER Employe_Insertion
ON EMPLOYE
FOR INSERT
AS
BEGIN
    Declare @NEmploye varchar
    Select @NEmploye = NEmploye from inserted

    insert into TabMessage
    values (@NEmploye, 'Employe avec le numéro '+@NEmploye+' a été créé')
END

```

FIGURE 12 – Trigger

100 %

Results Messages

	id	message
1	1	Employe avec le numéro 5 a été créé
2	6	Employe avec le numéro 6 a été créé

	NEMPLOYE	NMAGASIN	NOM	PRENOM
1	1	1	Julius	Caesar
2	2	1	Momo	Dupuis
3	3	2	Rome	Capulet
4	4	2	Escobart	Dupuis
5	5	2	LILI	Zlpo
6	6	2	POPO	LOLO

FIGURE 13 – Résultat trigger

7 Définition fonction utilisateur

La fonction créée et représenté dans la figure 14 permet de faire une recherche sur un livre.

SQLQuery9.sql - D...79SC39\hypo- (55))* - X

```

alter function Test( @titre varchar(30) )
returns table
as
    return Select LIVRE.NARTICLE as "Numéro", LIVRE.TITRE, CATEGORIE.LIBELLE as "Catégorie",
    ETAT.LIBELLE as "ETAT", CORRESPONDRE.PRIXACHAT as "Prix d'achat en €", AUTEUR.NOM as "Auteur Fname",
    AUTEUR.PRENOM as "Auteur Sname" from ECRIRE
    INNER JOIN AUTEUR on AUTEUR.NAUTEUR = ECRIRE.NAUTEUR
    INNER JOIN LIVRE on LIVRE.NARTICLE = ECRIRE.NARTICLE
    INNER JOIN APPARTENIR on APPARTENIR.NARTICLE = ECRIRE.NARTICLE
    INNER JOIN CATEGORIE on APPARTENIR.NCATÉGORIE = CATEGORIE.NCATÉGORIE
    INNER JOIN CORRESPONDRE on LIVRE.NARTICLE = CORRESPONDRE.NARTICLE
    INNER JOIN ETAT on CORRESPONDRE.NETAT = ETAT.NETAT
    WHERE LIVRE.TITRE LIKE @titre;

select * from Test('L%');

```

100 %

Results Messages

	Numéro	TITRE	Catégorie	ETAT	Prix d'achat en €	Auteur Fname	Auteur Sname
1	1	Lala_Land	Horreur	Parfait	12	Hypolite	Irakund
2	2	La Belle	Horreur	Parfait	3	Hypolite	Irakund
3	2	La Belle	Humour	Parfait	3	Hypolite	Irakund
4	3	Le Bea	Horreur	Bien	43	Hypolite	Irakund
5	3	Le Bea	Humour	Bien	43	Hypolite	Irakund

FIGURE 14 – Fonction

8 Sauvegarde (manuel / automatique) / restauration

8.1 Manuel

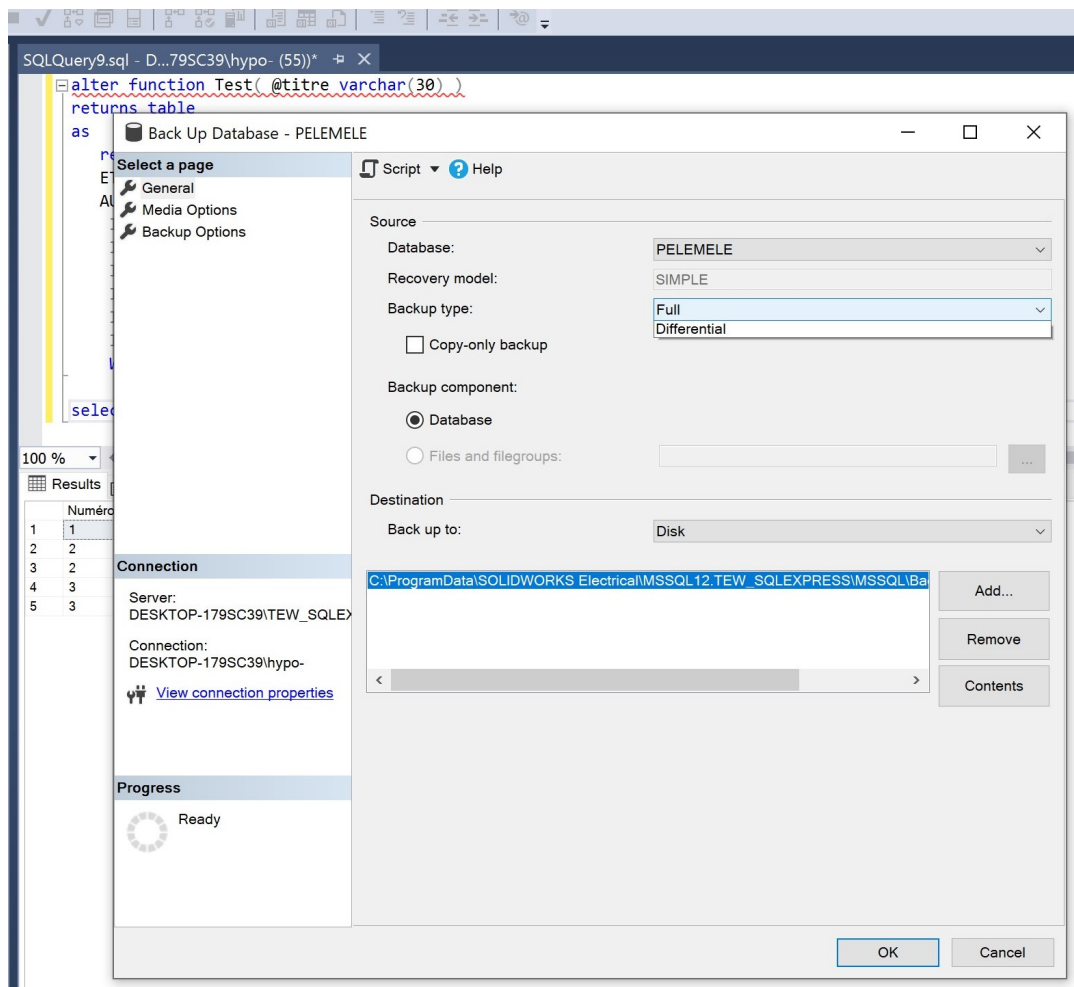


FIGURE 15 – Sauvegarde manuel

8.2 Manuel

8.3 Automatique

8.4 Restauration

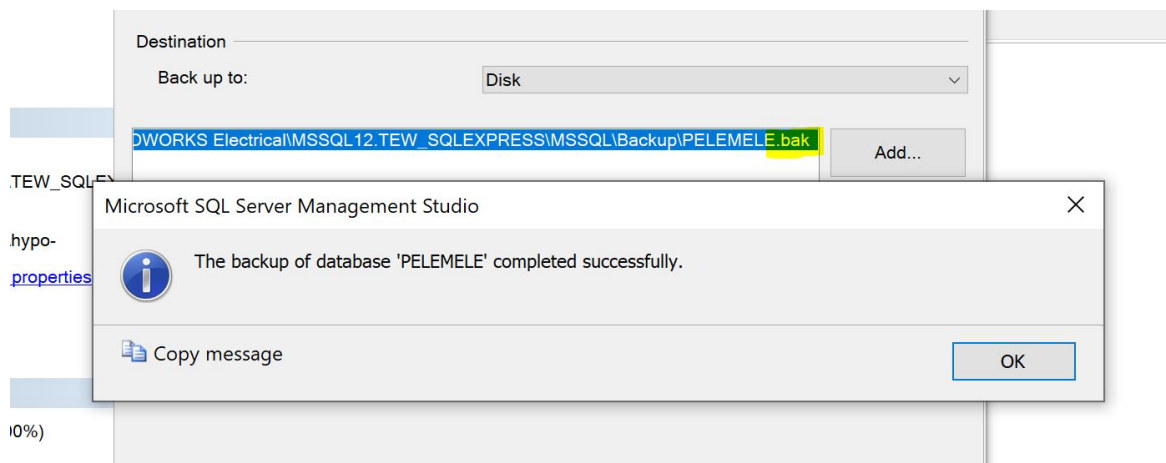


FIGURE 16 – Sauvegarde manuel successfully

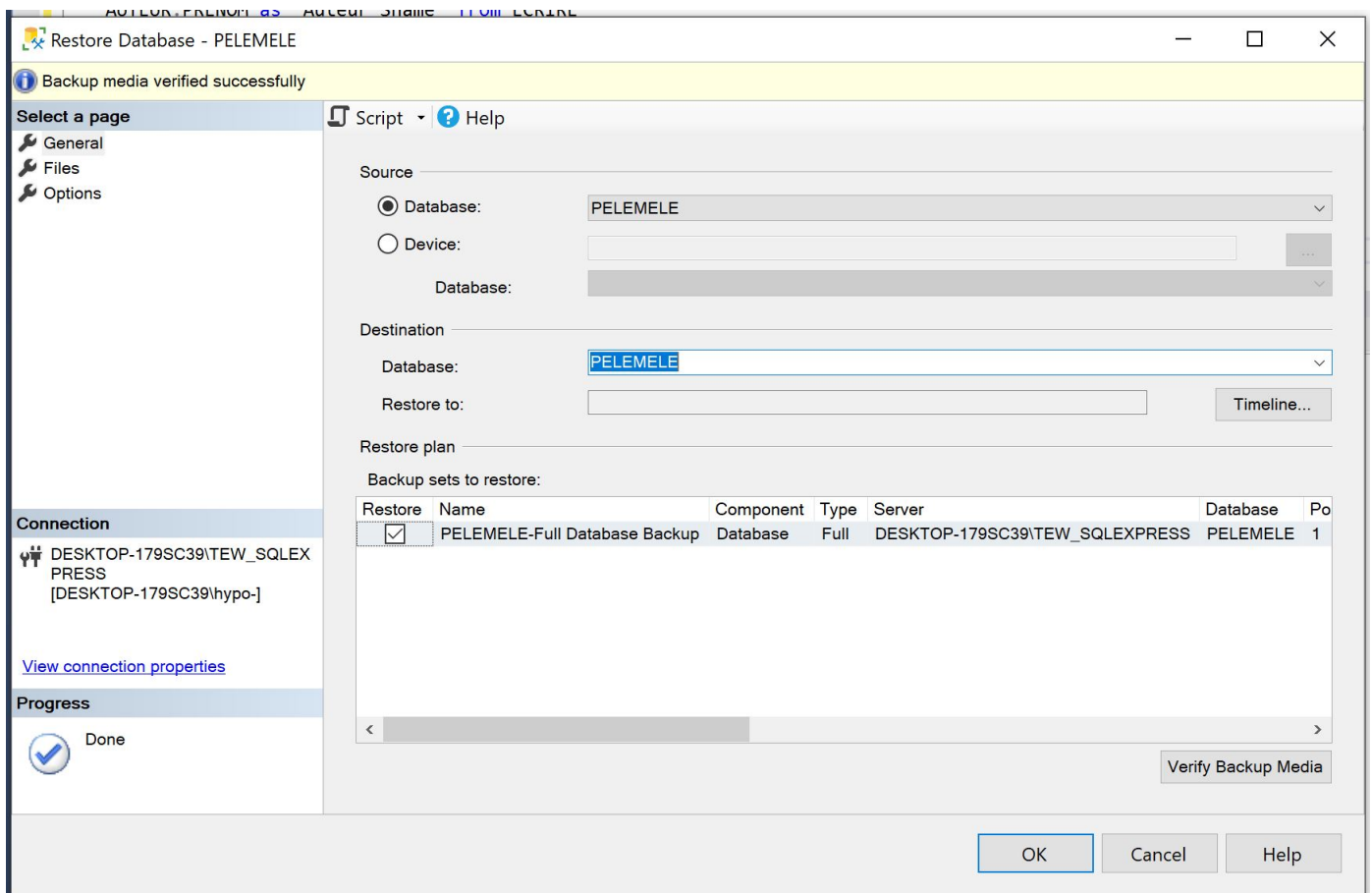


FIGURE 17 – Backup restored

Bibliographie

- [1] <https://developpement-informatique.com/article/284/les-contraintes-en-sql>
- [2] <https://sql.sh/cours/create-trigger#>