



COURS .NET SQL SERVER



Lemettre Arnaud
Arnaud.lemettre@gmail.com

SOMMAIRE



- ➞ Introduction
- ➞ Présentation
- ➞ Connexion avec Sql Management
- ➞ Création de fonctions SQL
- ➞ Utilisation de l'interface
- ➞ Les autres services

INTRODUCTION



- ➡ SQL Server est un système de gestion de base de données (SGBD) développé et commercialisé par Microsoft.
- ➡ Bien qu'il ait été initialement co-développé par Sybase et Microsoft, Ashton-Tate a également été associé à sa première version, sortie en 1989. Cette version est sortie sur les plateformes Unix et OS/2. Depuis, Microsoft a porté ce système de base de données sous Windows et il est maintenant uniquement pris en charge par ce système.



PRÉSENTATION

PRÉSENTATION



- ➡ Sql Server se décline en plusieurs versions :
 - ▶ Les plus utilisées version 2000, 2005, 2008, 2012
- ➡ Ainsi qu'en plusieurs éditions :
 - ▶ Compact
 - ▶ Express
 - ▶ Standard
 - ▶ Enterprise



PRÉSENTATION



➞ Compact

- ▶ Pour le compact framework (windows mobile ...)
- ▶ Ne contient pas toutes les fonctionnalités

➞ Express

- ▶ Contient la majorité des fonctionnalités, mais nombre d'utilisateurs (50 users simultanés), ainsi que taille des bases limitées (4Go), pour la version 2008R2 (10Go)

➞ Standard

- ▶ Contient l'ensemble des fonctionnalités

➞ Enterprise

- ▶ Permet la réalisation de cluster

PRÉSENTATION



- ➡ Une base d'exemple peut-être installée, elle permet de travailler sur des cas concrets d'utilisations. Elle s'appelle :
 - ▶ AdventureWorks
- ➡ Vous retrouverez souvent ce nom dans la littérature traitant de SQL Server. Les scripts d'installations sont disponibles sur le net.

PRÉSENTATION

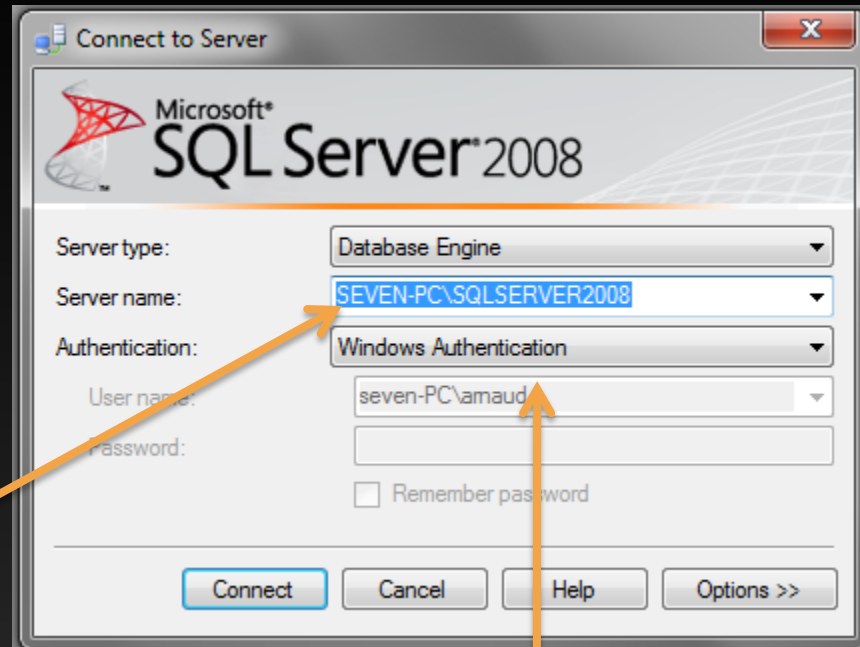


- ➡ Le programme d'installation est disponible à cette adresse :
 - ▶ <http://www.microsoft.com/express/Database/>
- ➡ Les différents types d'installation
 - ▶ With advanced services (full text, outil de BI ...)
 - ▶ Management tools : SQL Server Management Tools 2008



UTILISATION DE SQL MANAGEMENT

UTILISATION DE SQL MANAGEMENT



Server\InstanceSQL

Mode de connexion :
soit par compte windows
soit par connexion SQL

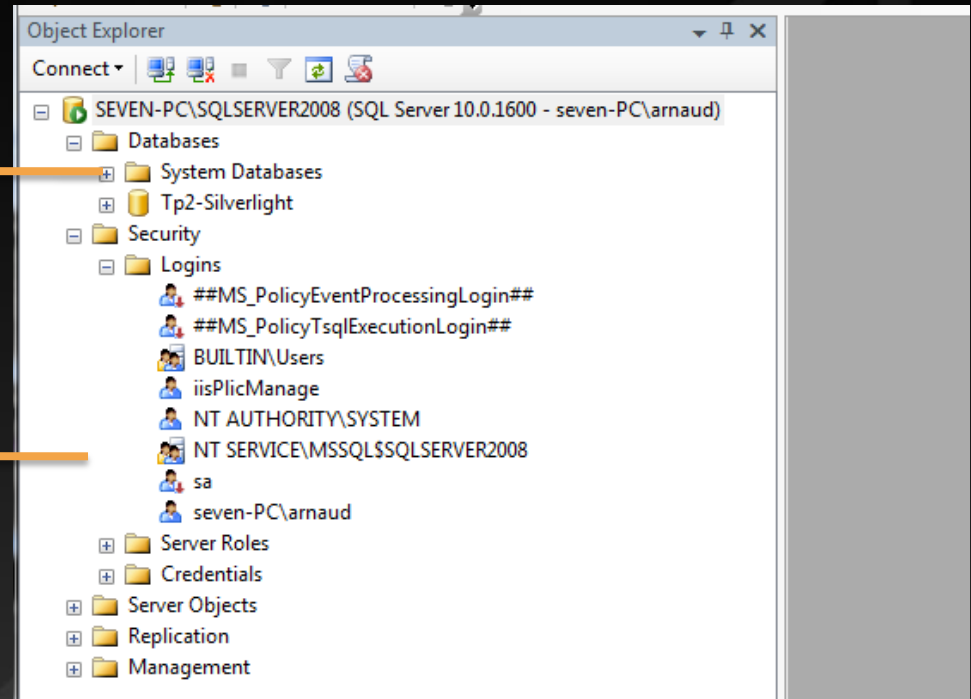
UTILISATION DE SQL MANAGEMENT



Liste des bases de données
présentes sur le serveur



Liste des utilisateurs, ayant
la possibilité d'effectuer une
action sur une base de données

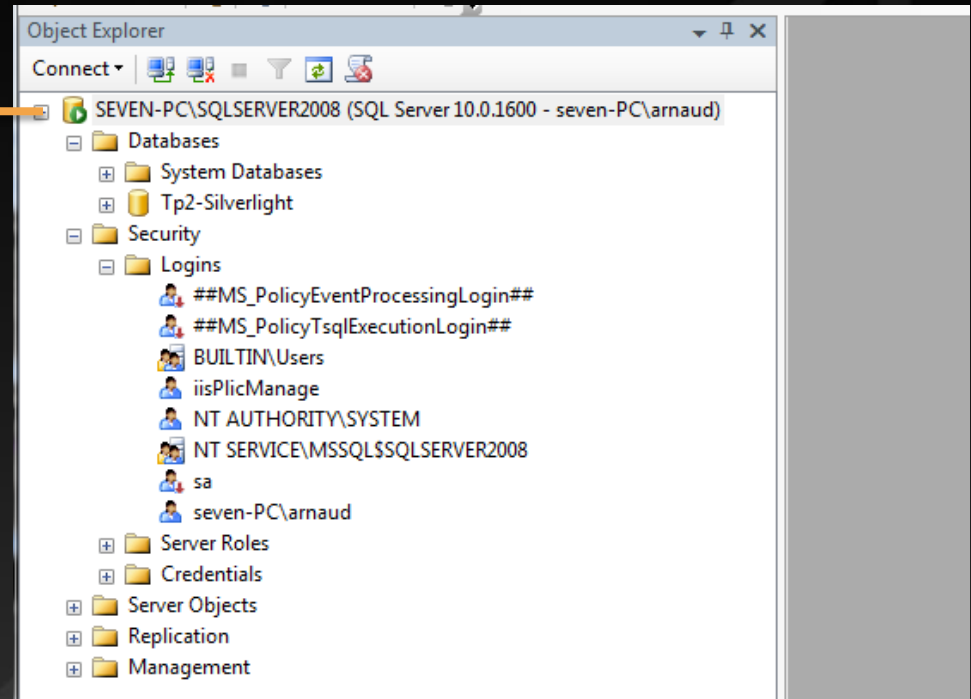
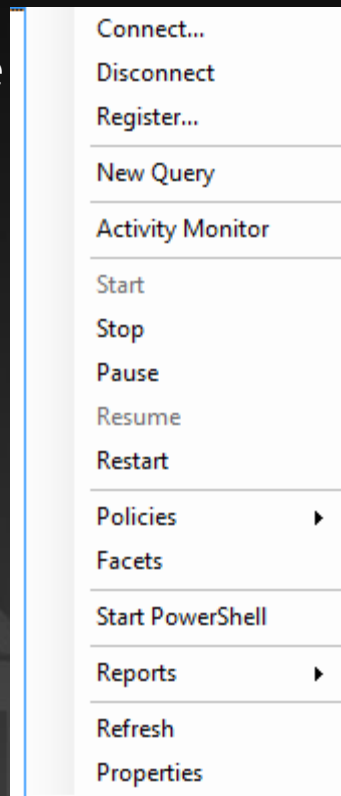


UTILISATION DE SQL MANAGEMENT



Permet d'interagir avec le serveur :

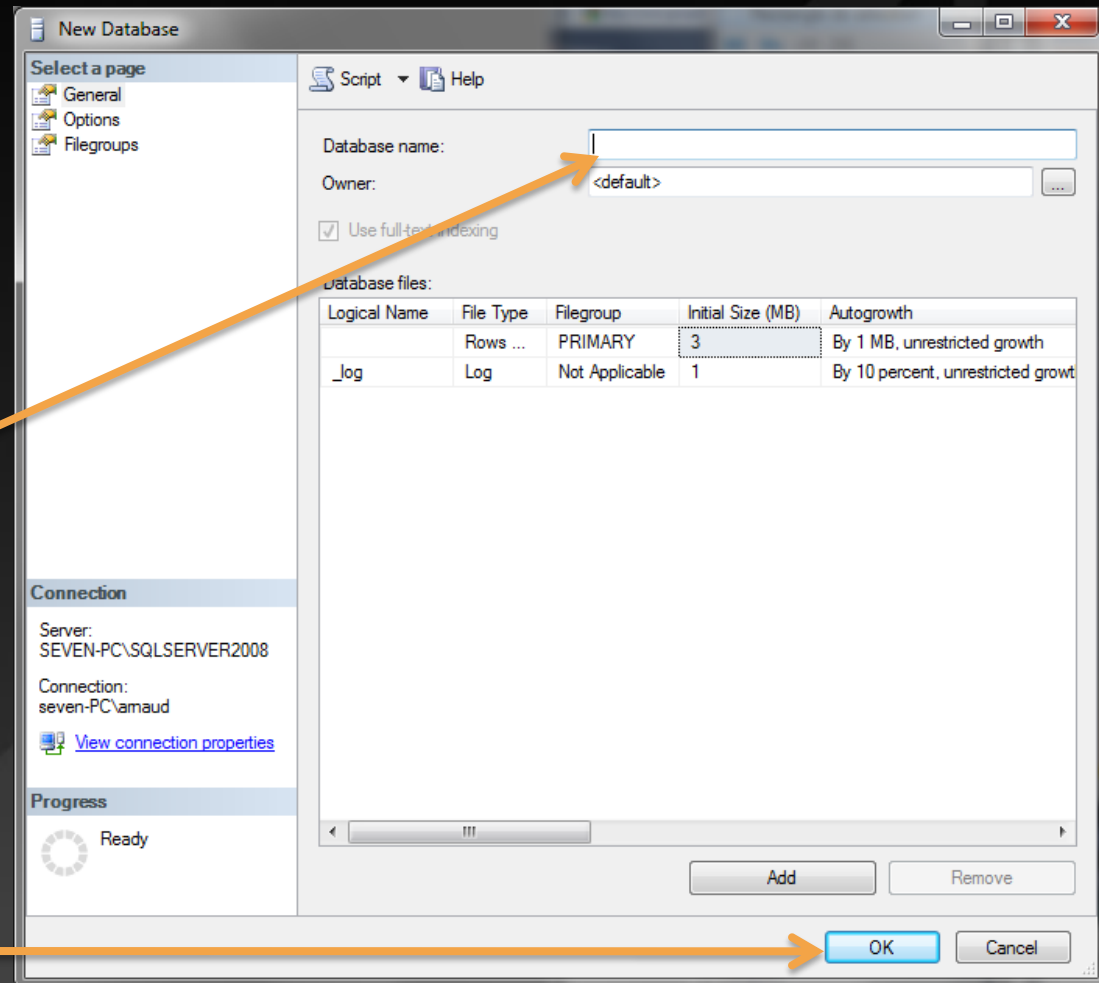
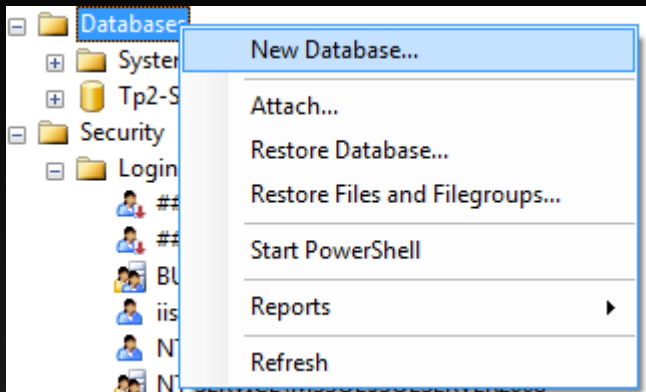
- Arrêt
- Redémarrage
- ...



UTILISATION DE SQL MANAGEMENT



➡ Création d'une base de données.

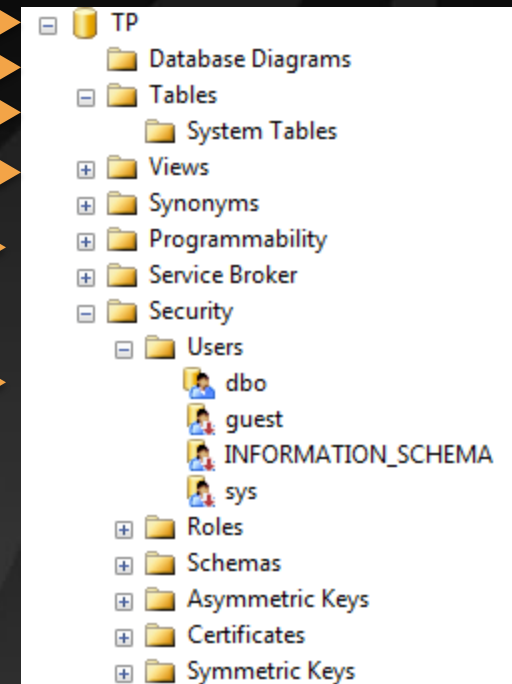


UTILISATION DE SQL MANAGEMENT



➡ Une base de données :

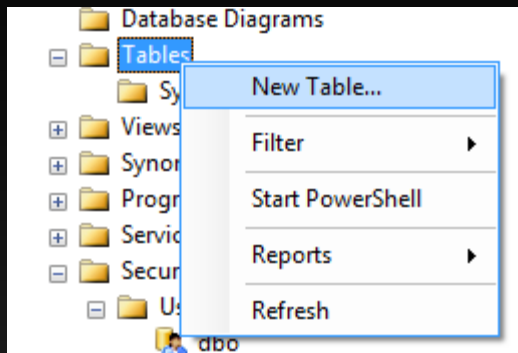
Nom de la base →
Schéma visuel de la base →
Les tables →
Les vues →
Les procédures stockées →
Les utilisateurs ayant accès à la base →



UTILISATION DE SQL MANAGEMENT



➡ Création d'une table :



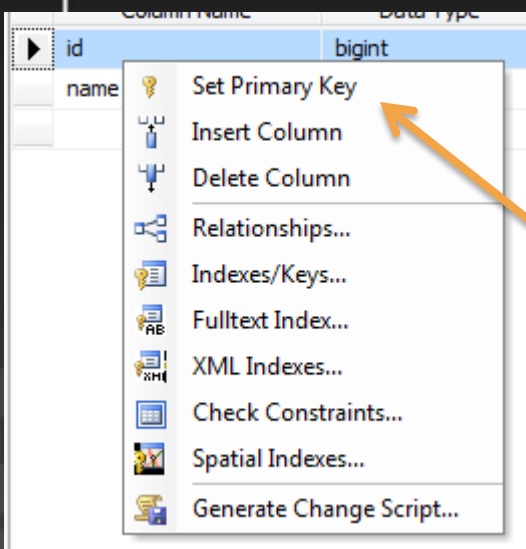
Column Name	Data Type	Allow Nulls
id	bigint	<input type="checkbox"/>
name	nvarchar(50)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Nom de la
colonne

Type de
la
colonne

On autorise ou non
les valeurs null

Permet de mettre une clé primaire sur la colonne

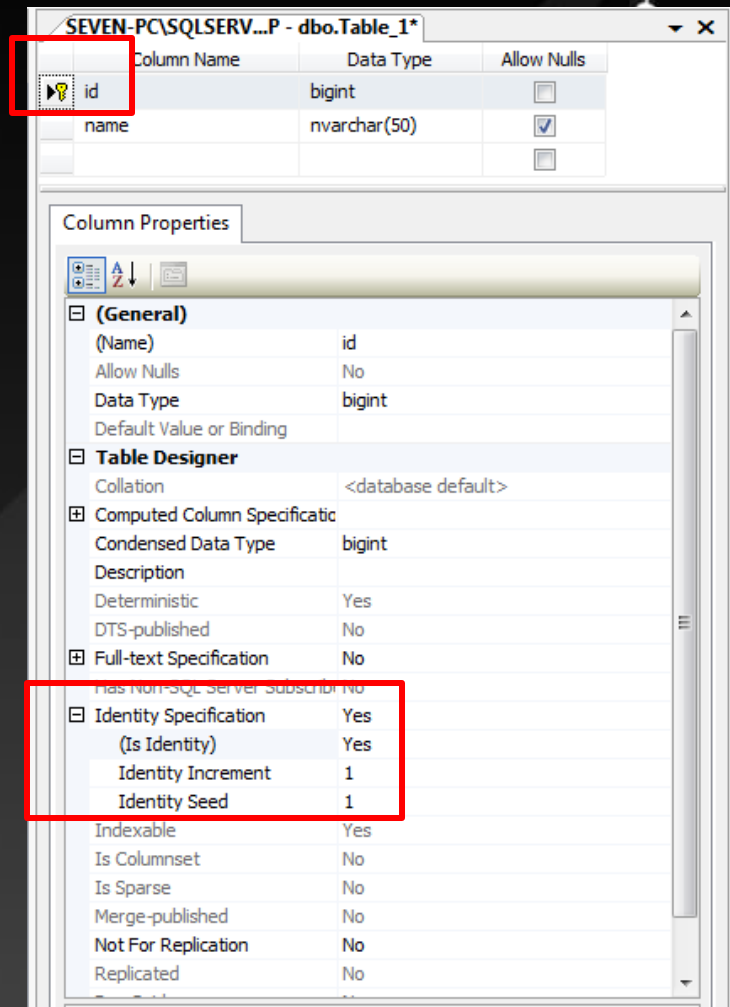


UTILISATION DE SQL MANAGEMENT



- ➡ Mettre une contrainte de clé primaire, signifie que celle-ci doit être unique sous peine de déclencher une contrainte d'intégrité. Pour cela, nous pouvons fixer des valeurs d'auto incrémentation.

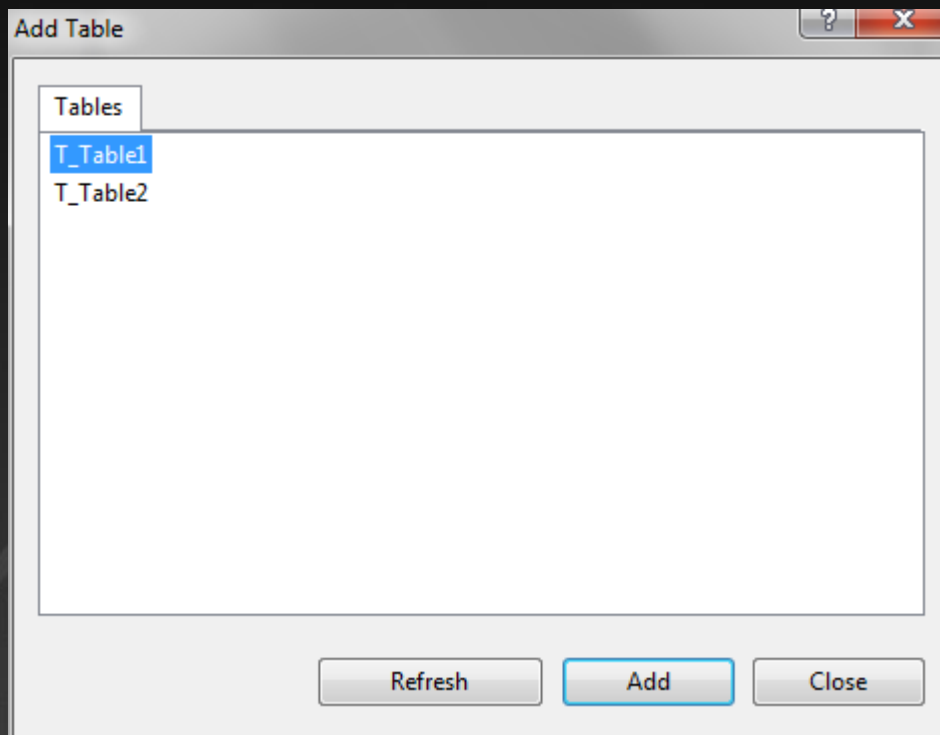
L' enregistrement de la table (ctrl + S), permettra de spécifier le nom de la table



UTILISATION DE SQL MANAGEMENT

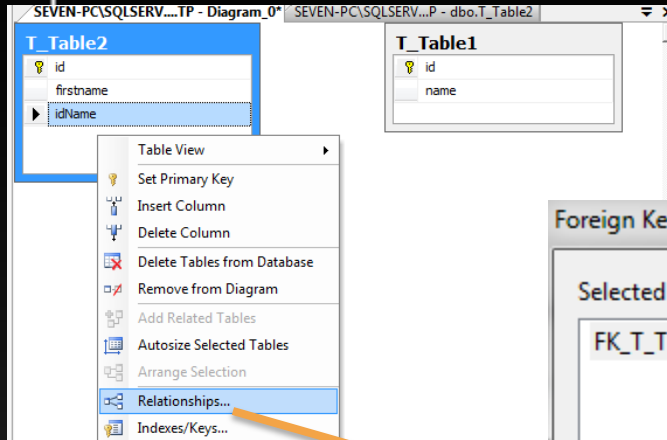


- ➡ De la même manière, nous pouvons spécifier les relations avec les clés étrangères. Pour y parvenir, soit on peut faire comme pour les clés primaires ou passer par le schéma de la base.

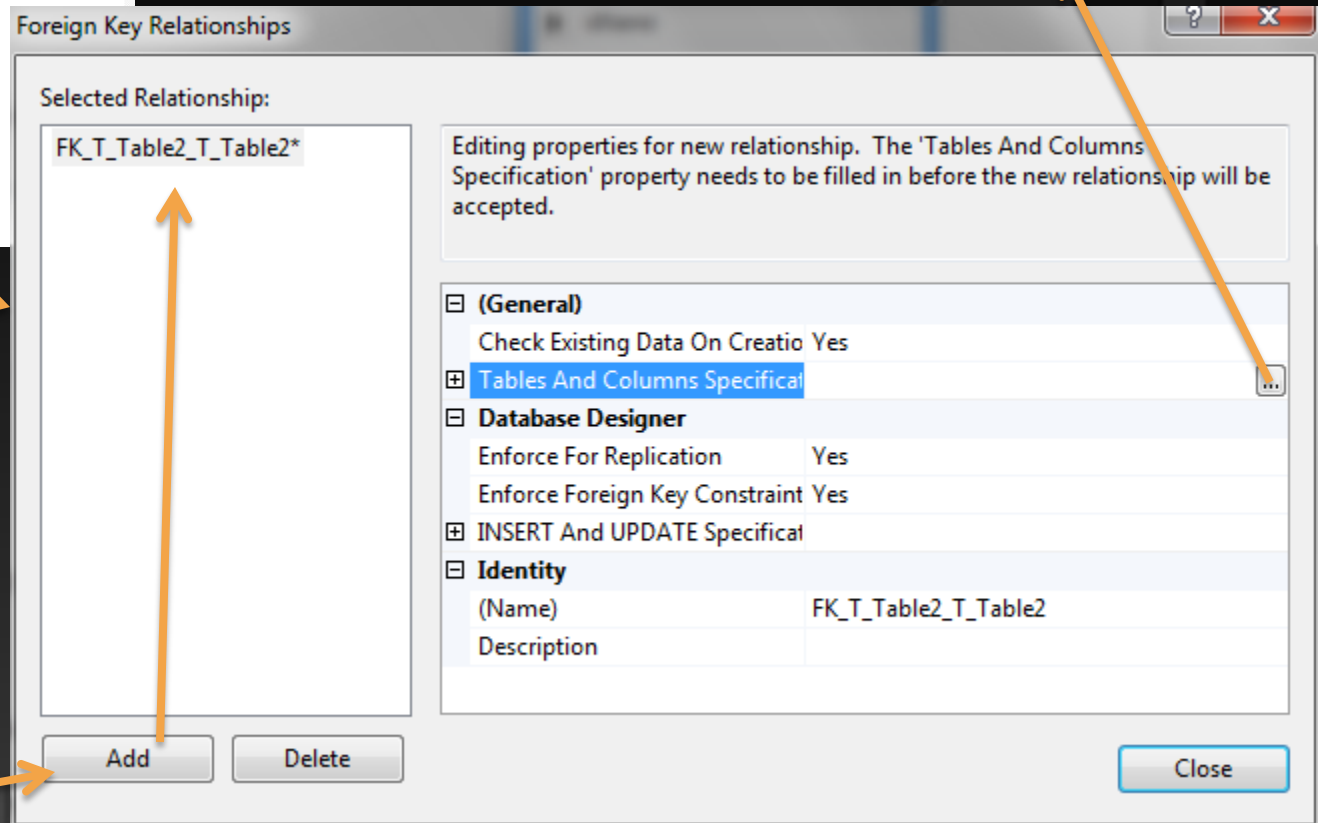


Sélectionner l'ensemble des tables à ajouter au schéma

UTILISATION DE SQL MANAGEMENT



Permet de spécifier la relation



Ajout d'une clé étrangère

UTILISATION DE SQL MANAGEMENT



Tables and Columns

Relationship name:
FK_T_Table2_T_Table1

Primary key table:
T_Table1

Foreign key table:
T_Table2

id	idName
----	--------

OK Cancel

Sélection de
l'autre table

Sélection de
la clé primaire

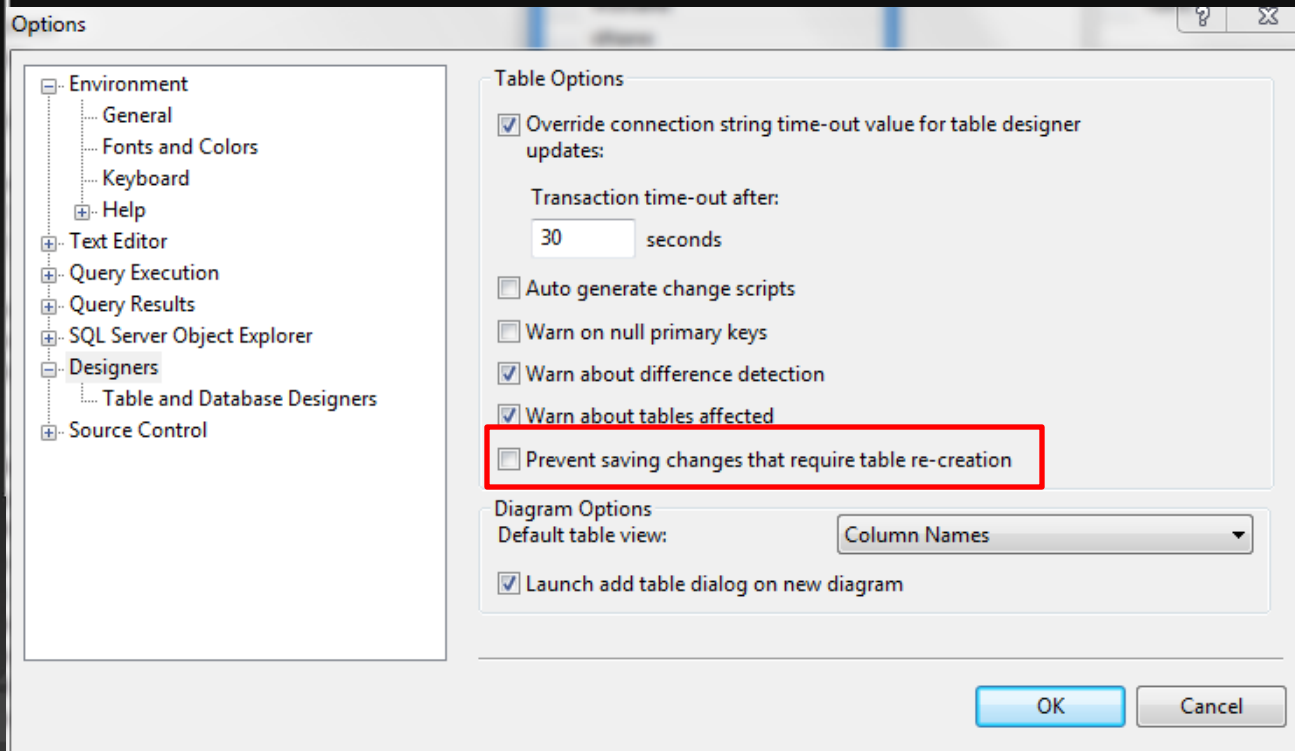
Sélection de
l'id

Une fois les fenêtres fermées, faire un ctrl + S

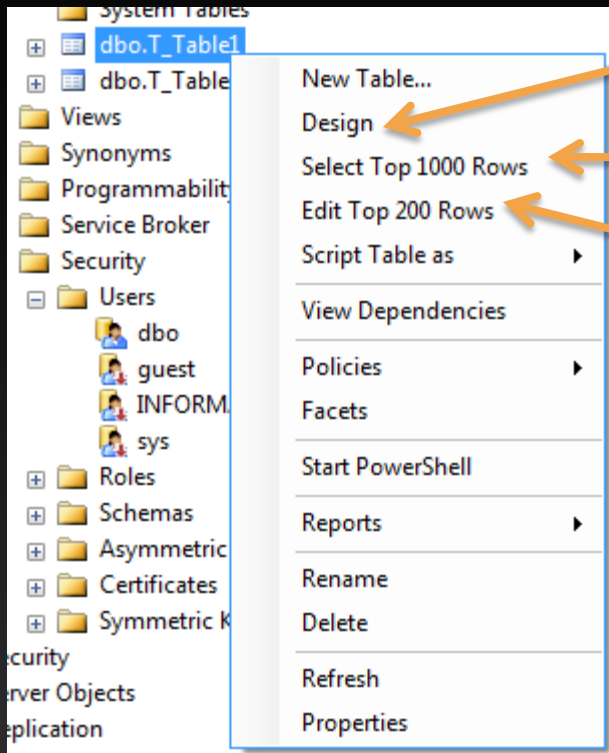
UTILISATION DE SQL MANAGEMENT



Certaines fois, l'éditeur ne permet pas d'enregistrer ce type de modification
il faut décocher la case suivante :



UTILISATION DE SQL MANAGEMENT

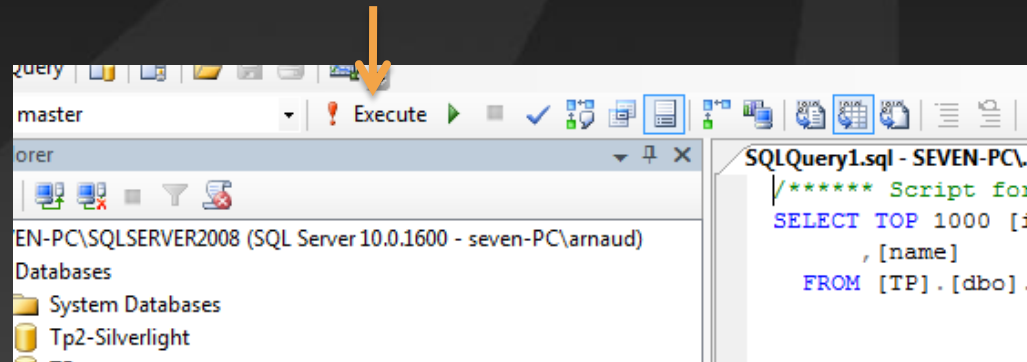


Accède à la création des champs sur une table

Affiche les 1000er éléments.

Permet de modifier les 200er éléments

Permet de rafraîchir les données lors d'une sélection





REQUÊTES SQL

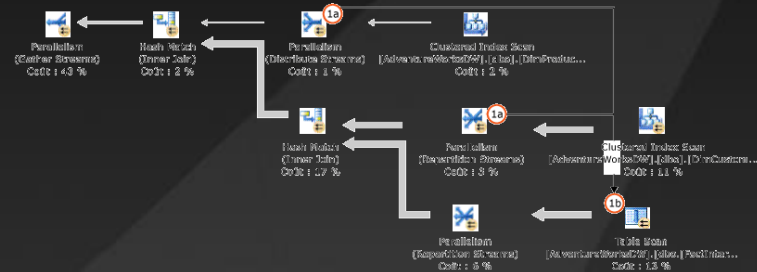


Table Scan	
Analyse des lignes d'une table.	
Physical Operation	TableScan
Logical Operation	Table Scan
Actual Number of Rows	26 635
Estimated I/O Cost	0,846907
Estimated CPU Cost	0,0166291
Estimated Operator Cost	0,863536 (13 %)
Estimated Subtree Cost	0,863536
Estimated Number of Rows	60 398
Estimated Row Size	195 B
Actual Rebinds	0
Actual Rewinds	0
Ordered	False
Node ID	9
Objet	
[AdventureWorksDW].[dbo].[FactInternetSales][F]	

REQUÊTES SQL



Faire un select :

```
SELECT <comma-delimited list of expressions or column names>  
[INTO <table name>]  
FROM <tables, views, or linked servers>  
[WHERE <search condition(s)>]  
[GROUP BY <comma-delimited list of columns>]  
[HAVING <search condition(s)>]  
[ORDER BY <comma-delimited list of columns> <ASC or DESC>]
```

REQUÊTES SQL



Concrètement, faire new Query :

```
SELECT *
```

```
FROM T_Table1
```

En haut de l'éditeur :



Nom de la base de données

Lance
l' exécution

Vérifie la syntaxe
Permet de debugger la requête

Résultat :

Results			Messages	
	id	name		
1	1	LG		
2	2	philips		
3	3	Sony		

REQUÊTES SQL



Avec une clause where :

```
SELECT [T_Table1].name  
FROM T_Table1  
where [T_Table1].id = 1
```

Résultat :

A screenshot of a SQL Server 'Results' window. The window has two tabs: 'Results' and 'Messages'. The 'Results' tab is active, showing a table with one column named 'name' and one row containing the value 'LG'. The row is highlighted with a dashed border.

	name
1	LG

REQUÊTES SQL



Avec une clause order by:

```
SELECT [T_Table1].name  
FROM T_Table1  
ORDER BY [T_Table1].name DESC
```

Résultat :

	name
1	Sony
2	philips
3	LG

REQUÊTES SQL



Avec une clause group by & Having :

```
SELECT  store_name, SUM(Sales) FROM [T_Sale]
GROUP BY store_name
```

	store_name	(No column name)
1	logitech	15
2	microsoft	1

```
SELECT  store_name, SUM(Sales) FROM [T_Sale]
GROUP BY store_name
HAVING SUM(sales) > 10
```

	store_name	(No column name)
1	logitech	15

Un opérateur arithmétique doit au moins apparaître dans l'instruction select.

REQUÊTES SQL



Réaliser un insert :

```
INSERT INTO [T_Table1]  
VALUES ('logitech')
```

Insert sans toutes les valeurs :

```
INSERT INTO [T_Table2] (idName,  
    firstname)  
VALUES (1, 'mouse')
```



Attention cependant à ce que toutes les colonnes n'ayant pas de null autorisé / ou de valeur définie à l'avance soit bien passées en paramètre.

REQUÊTES SQL



La suppression :

```
DELETE FROM [T_Table2]  
WHERE [T_Table2].id = 2
```

Suppression de toutes les lignes d'une table :

```
DELETE FROM [T_Table2]
```

REQUÊTES SQL



La mise à jour :

```
UPDATE [T_Table2]
SET idName = 1
WHERE [T_Table2].firstname = 'keyboard'
```

REQUÊTES SQL



Quelques fonctions utiles :

- ▶ AVG
- ▶ Count
- ▶ MAX
- ▶ Min
- ▶ Sum
- ▶ GetDate
- ▶ IfNull
- ▶ DataLength
- ▶ Lower
- ▶ Upper
- ▶ Replace
- ▶ Reverse
- ▶ PatIndex
- ▶ Fonction mathématiques (cos, sinus , ...)
- ▶ DateAdd
- ▶ Truncate
- ▶



PROCÉDURES STOCKÉES

PROCÉDURES STOCKÉES



- ➡ Une **procédure stockée** (ou *stored procedure* en anglais) est un ensemble d'instructions SQL pré-compilées, stockées sur le serveur, directement dans la base de données. Elles peuvent être exécutées sur demande : lancées par un utilisateur, un administrateur DBA ou encore de façon automatisée par un événement déclencheur (de l'anglais "trigger").

Wikipedia

PROCÉDURES STOCKÉES

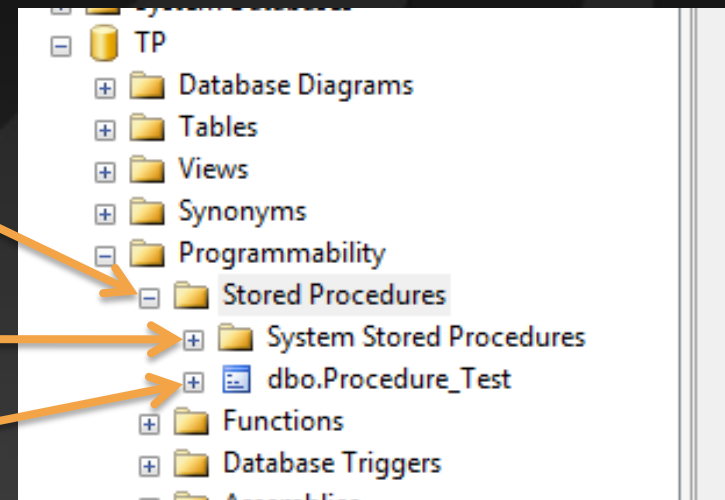


- ➡ Certaines procédures sont déjà présentes dans le SGBD, par exemple le fait d'itérer sur des objets de la database.

Dossier contenant l'ensemble
des procédures stockées

Procédure du système

Procédure personnalisable



PROCÉDURES STOCKÉES



Création d'une procédure

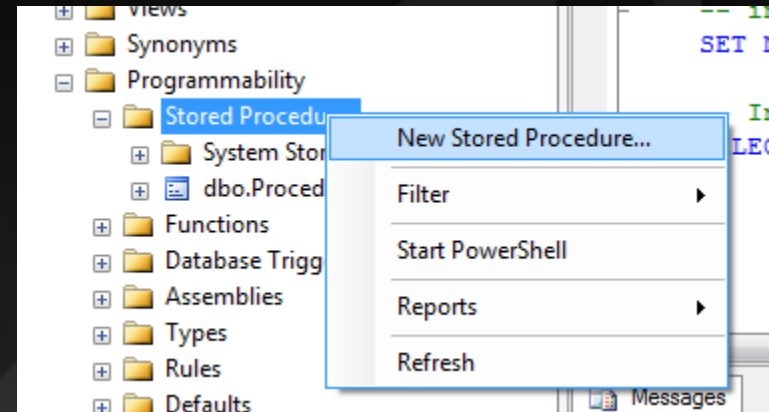
```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
-- =====
-- Author:          <Author,,Name>
-- Create date: <Create Date,,>
-- Description:      <Description,,>
-- =====

CREATE PROCEDURE Procedure_Test
    -- Ajout des paramètres
    -- ici valeur par défaut : test
    @Param varchar(50) = 'test'

AS
BEGIN
    -- SET NOCOUNT ON added to prevent extra
    result sets from
    -- interfering with SELECT statements.
    SET NOCOUNT ON;

    -- Mettre votre code ici
    SELECT * from [T_Table1]

END
GO
```



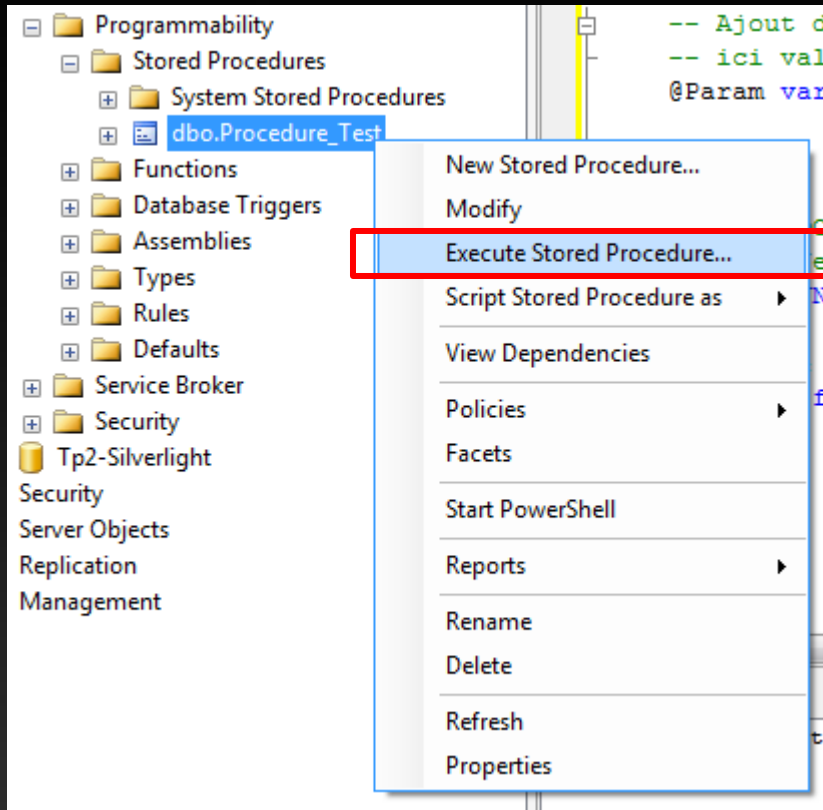
← Script permettant la génération d'une procédure stockée.

De base un template est directement proposé pour écrire vos procédures.

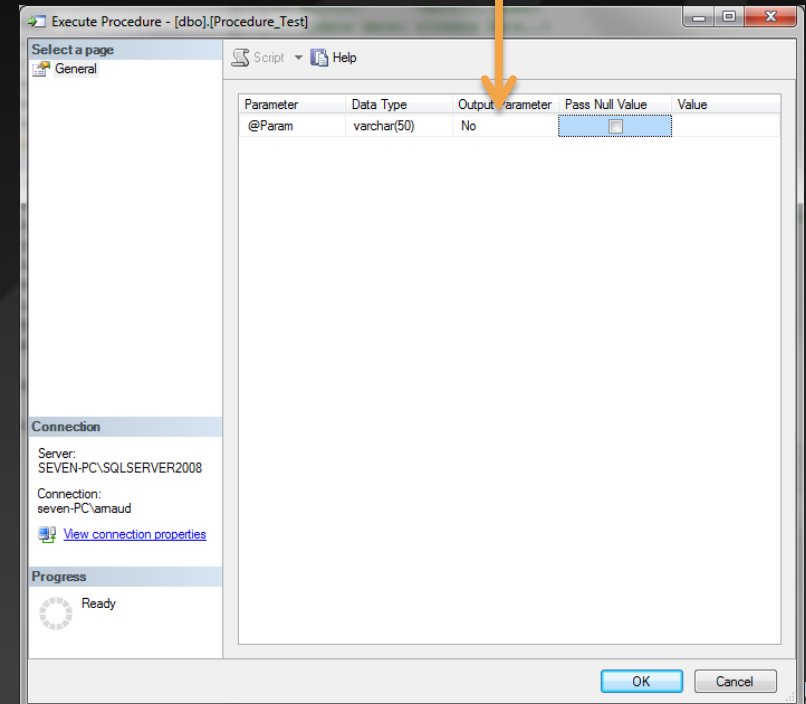
PROCÉDURES STOCKÉES



➡ Test d'une procédure stockée :



Liste des paramètres pour exécuter la procédure



PROCÉDURES STOCKÉES



➡ Test d'une procédure stockée :

```
SQLQuery2.sql - SEVEN-PC\...\... (54)  SQLQuery1.sql - SEVEN-PC\...\... (5)
USE [TP]
GO

DECLARE @return_value int

EXEC      @return_value = [dbo].[Procedure_Test]
          @Param = N'12'

SELECT    'Return Value' = @return_value

GO
```

Results Messages

	id	name
1	1	LG
2	2	philips
3	3	Sony
4	4	logitech

	Return Value
1	0

Utilisation d'une procédure stockée

Résultat de la procédure stockée

0 tout c'est bien passé sinon valeur d'erreur



LES TRIGGERS

LES TRIGGERS



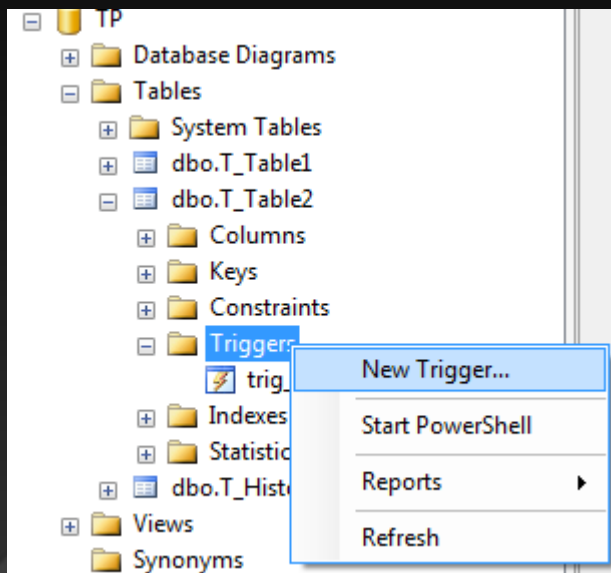
- ➡ Ceux-ci sont des réponses à des événements qui peuvent se produire dans une base de données.
- ➡ Il en existe 3 sortes :
 - ▶ Celles en rapport avec les données (insert, ...)
 - ▶ Celles en rapport avec les schémas (create, drop, ...)
 - ▶ Celles en rapport avec les connexions

Pour plus de détails voici un [lien](#) sur le site de Microsoft

LES TRIGGERS



- ➡ Pour illustrer l'utilisation de trigger, nous allons créer une table d'historique sur la suppression d'éléments dans une base.



LES TRIGGERS



Création de la table d'historique

```
USE [TP]
GO

/***** Object: Table [dbo].[T_History]    Script Date: 02/16/2010 23:02:43 *****/
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON
GO

CREATE TABLE [dbo].[T_History](
    [id] [bigint] IDENTITY(1,1) NOT NULL,
    [oldValue] [nvarchar](50) NOT NULL,
    [delDate] [datetime] NOT NULL,
    CONSTRAINT [PK_T_History] PRIMARY KEY CLUSTERED
(
    [id] ASC
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS
    = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
```

LES TRIGGERS



Code du trigger

```
CREATE TRIGGER trig_delTable2 ON  
    [T_Table2]  
  
AFTER DELETE  
  
AS  
  
DECLARE @isOnContract nvarchar(50)  
SET @isOnContract = (SELECT firstname FROM  
    Deleted)  
  
INSERT INTO [T_History] VALUES  
    (@isOnContract, GETDATE())
```

LES TRIGGERS



- ➡ Il existe plusieurs comportements pour les triggers. Par exemple AFTER (delete, insert ...) et INSTEAD OF qui permet d'exécuter le trigger à la place de l'opération déclenchée
- ➡ Des pseudos tables sont accessibles lors de l'exécution des triggers comme :
 - ▶ DELETED
 - ▶ INSERTED

LES TRIGGERS



- ➡ IF UPDATE (column) permet de savoir si une colonne a été renseignée ou non lors d'une insertion ou modification
- ➡ Pour désactiver les triggers :

```
ALTER TABLE [T_Table2] DISABLE TRIGGER trig_delTable2
```

- ➡ Pour activer les triggers :

```
ALTER TABLE [T_Table2] ENABLE TRIGGER trig_delTable2
```

On peut donc facilement imaginer un système de log, permettant de tracer les activités sur la base de données, avec un script activant / désactivant l'ensemble de ces fonctionnalités



LES VUES

LES VUES



- ➡ Une vue est une table virtuelle. C'est-à-dire dont les données ne sont pas stockées dans une table de la base de données, et dans laquelle il est possible de rassembler des informations provenant de plusieurs tables. On parle de "vue" car il s'agit simplement d'une représentation des données dans le but d'une exploitation visuelle.

LES VUES



➡ Création avec le designer :

SEVEN-PC\SQLSERVER2008 (SQL Server 10.0.1600 - s)

Databases

- System Databases
- TP
 - Database Diagrams
 - Tables
 - System Tables
 - dbo.T_Table1
 - dbo.T_Table2
 - dbo.T_History
 - Views
 - New View...
 - Filter
 - Start PowerShell
 - Reports
 - Refresh
- Security
- Server Objects
- Replication
- Management

SEVEN-PC\SQLSER...8.TP - dbo.View1

SEVEN-PC\SQLSERV...P - dbo.T_Table2

T_Table1

- * (All Columns)
- id
- name

T_Table2

- * (All Columns)
- id
- firstname
- idName

Column	Alias	Table	Output	Sort Type	Sort
name		T_Table1	<input checked="" type="checkbox"/>		
firstname		T_Table2	<input checked="" type="checkbox"/>		
			<input type="checkbox"/>		
			<input type="checkbox"/>		
			<input type="checkbox"/>		
			<input type="checkbox"/>		

```
SELECT dbo.T_Table1.name, dbo.T_Table2.firstname
FROM   dbo.T_Table1 INNER JOIN
       dbo.T_Table2 ON dbo.T_Table1.id = dbo.T_Table2.idName
```

LES VUES



Soit par code :

```
CREATE VIEW Vue  
(nom, prénom)  
AS SELECT name , firstname  
FROM [T_Table1] Alias1, [T_Table2]  
    AliasII  
WHERE Alias1.id = AliasII.idName
```


LES VUES



- ➞ L'avantage de travailler avec des vues :
 - ▶ Offrir juste les informations nécessaires à l'utilisateur (abstraction des tables)
 - ▶ Plus de sécurité
 - ▶ On peut regrouper un ensemble d'information au sein d'une même vue
 - ▶ Plus optimisée



OPÉRATIONS D'ADMINISTRATION

OPÉRATIONS D'ADMINISTRATION



- ➡ Savoir créer une base de données, ainsi que des requêtes SQL c'est déjà bien, mais en tant que développeur et ingénieur il faut aussi penser plus large :
 - ▶ Sauvegardes de base
 - ▶ Restaurations
 - ▶ Création de scripts
 - ▶ ...

OPÉRATIONS D'ADMINISTRATION



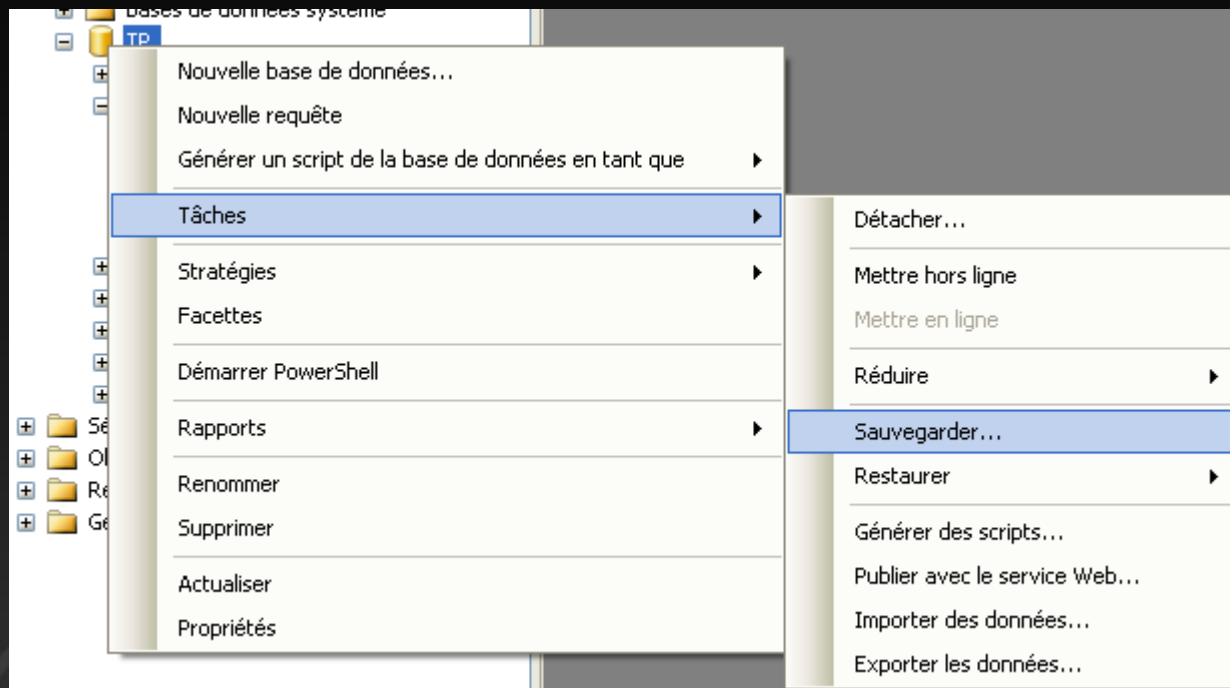
➡ Sauvegarde de base de données :

- ▶ Attention la création d'un backup de base de données a ses limites, à savoir :
 - ▶ Le backup ne pourra être restauré que sur une configuration identique, pas de saut de versions
- ▶ Avantage :
 - ▶ Le schéma, ainsi que les éléments et les enregistrements de base sont enregistrés dans le backup

OPÉRATIONS D'ADMINISTRATION



➡ Méthode :



OPÉRATIONS D'ADMINISTRATION



Sauvegarder la base de données - TP

Sélectionner une page

Général

Options

Script Aide

Source

Base de données : TP

Mode de récupération : SIMPLE

Type de sauvegarde : Complète

☐ Sauvegarde de copie uniquement

Composant de sauvegarde :

☒ Base de données

☐ Fichiers et groupes de fichiers :

Jeu de sauvegarde

Nom : TP-Complète Base de données Sauvegarde

Description :

Expiration du jeu de sauvegarde :

☒ Après : 0 jours

☐ Le : 17/02/2010

Destination

Sauvegarde sur : ☒ Disque ☐ Bande

C:\Program Files\Microsoft SQL Server\MSSQL10.SQLEXPRESS\MSSQL\Backup\TP.bak

Ajouter...

Supprimer

Sommaire

OK Annuler

Soit une sauvegarde
complète ou
différentielle

Permet de fixer une
date d'expiration

Chemin pour récupérer
le backup

OPÉRATIONS D'ADMINISTRATION



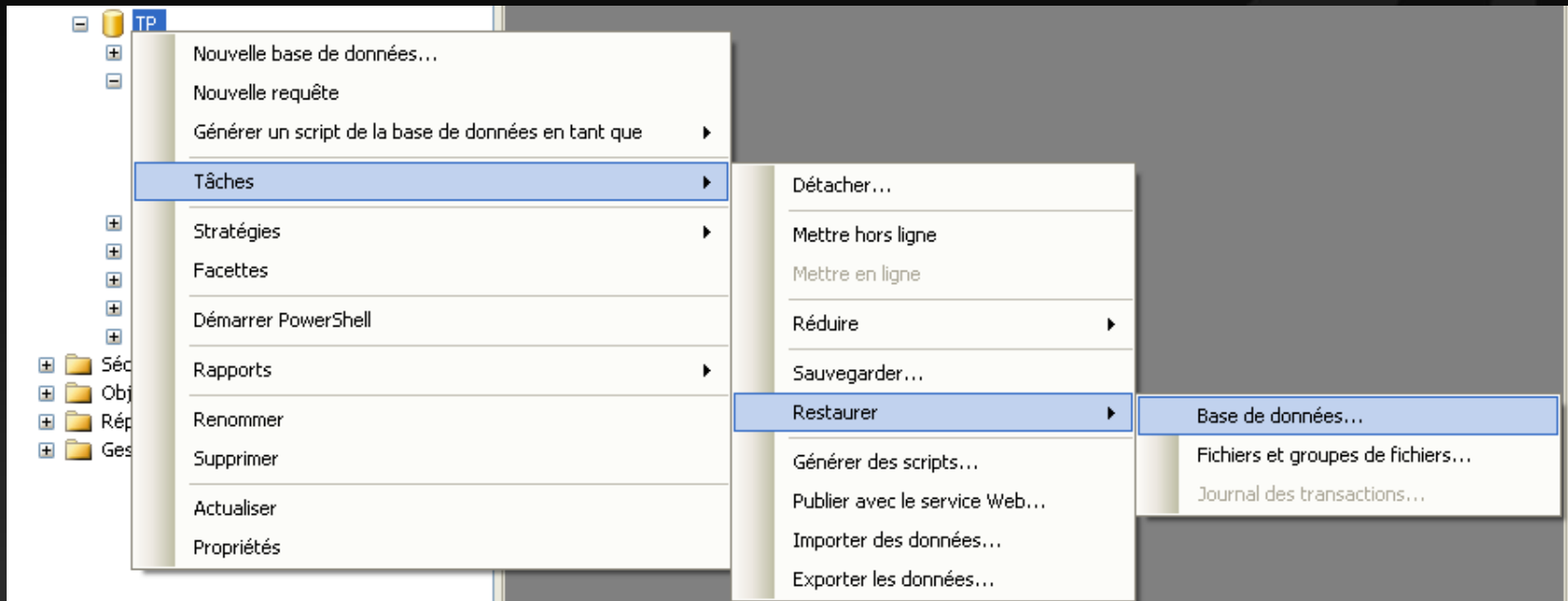
➡ Restauration d'un backup :

- ▶ Il faut déposer le fichier .bak, dans le répertoire
C:\Program Files\Microsoft SQL
Server\MSSQL10.SQLEXPRESS\MSSQL\Backup
- ▶ S'il n'est pas dans ce dossier alors le système ne verra pas
le fichier.

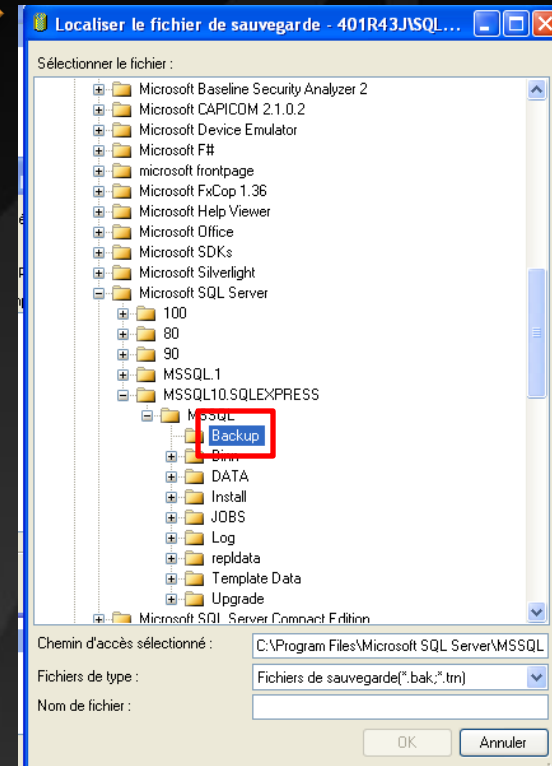
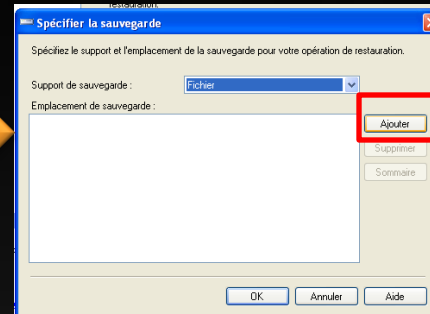
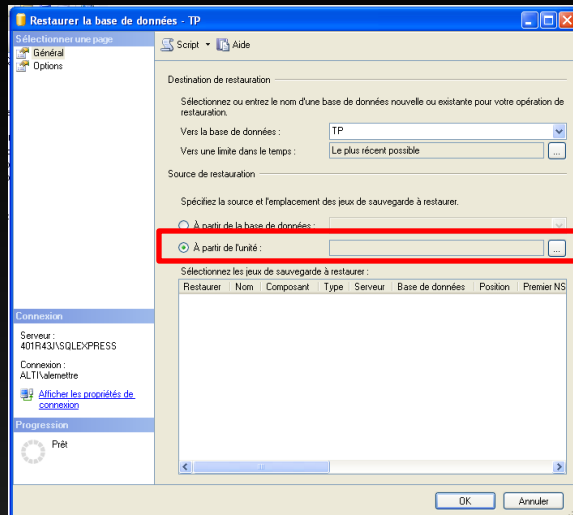
OPÉRATIONS D'ADMINISTRATION



Méthode :



OPÉRATIONS D'ADMINISTRATION



Il ne reste qu'à sélectionner votre bon backup
et valider l'ensemble des fenêtres.

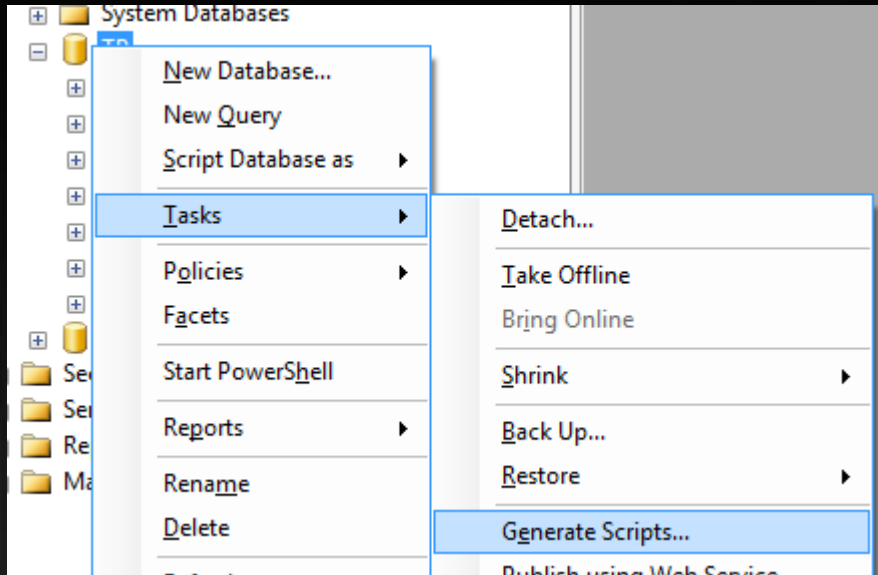
OPÉRATIONS D'ADMINISTRATION



➡ La génération de scripts :

- ▶ Permet de sauvegarder l'ensemble des éléments sauf les informations en base.
- ▶ Peut être compatible entre différentes versions de SQL server si des fonctions spécifiques ne sont pas utilisées.

OPÉRATIONS D'ADMINISTRATION



➡ Il ne reste plus qu'à suivre l'assistant pour générer le script

OPÉRATIONS D'ADMINISTRATION

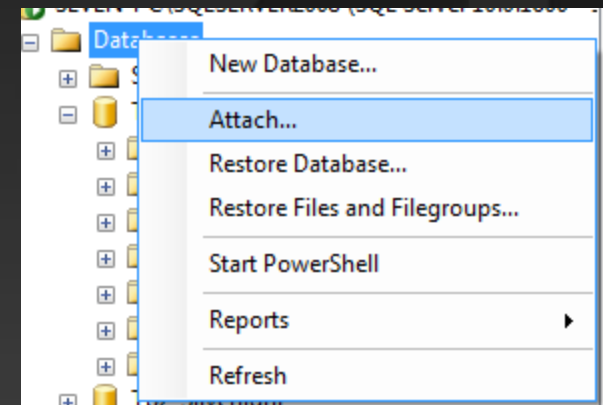


- ➡ Une autre alternative est de déplacer la base de données sur un autre serveur. Pour cela, on peut détacher la base et ensuite copier les fichiers :
 - ▶ C:\Program Files\Microsoft SQL Server\MSSQL10.SQLSERVER2008\MSSQL\DATA
 - ▶ .mdf (BDD)
 - ▶ .ldf (Journal de transactions)

OPÉRATIONS D'ADMINISTRATION



- ➡ Pour rattacher les fichiers, il faut les copier dans ce dossier :
 - ▶ C:\Program Files\Microsoft SQL Server\MSSQL10.SQLSERVER2008\MSSQL\DATA
 - ▶ .mdf (BDD)
 - ▶ .ldf (Journal de transactions)
- ➡ Puis faire Attach





LES AUTRES SERVICES

AVANCÉ



- ➡ Il existe d'autres utilisations beaucoup plus avancées de SQL Server.
- ➡ Cependant nous ne les verrons pas dans ce cours :
 - ▶ Full text
 - ▶ Gestion des fileStreams
 - ▶ Cube
 - ▶ Géolocalisation
 - ▶ Gestion XML
 - ▶ ...



QUESTIONS ?