# SQL Server 2022 Administration

Serveur SGBDR

Olivier D.

# Table des matières

1	Prés	sentation	3
	1.1	Introduction	3
	1.2	Fonctionnement	3
	1.3	BD SQL Server	4
2	Inst	allation et configuration	5
	2.1	Installer SQL Server	5
	2.2	Déroulement de l'installation	5
3	Ges	tion de la base de données	7
	3.1	Organisation logique	7
	3.2	Organisation physique	7
	3.3	Création d'une base	<u>c</u>
	3.4	Gérer une base de données	
	3.5	Supprimer une base	10
	3.6	Structure des index	11
	3.7	Compression des données	11
4	Ges	tion de la sécurité d'accès	
	4.1	Introduction	12
	4.2	Gestion des accès	12
	4.3	Création de la connexion (à l'instance / au serveur)	13
	4.4	Création du mappage	13
	4.5	Les autorisations (utilisateurs de BD) au niveau de la base de données	13
	4.6	Au niveau du serveur	
	4.7	Gestion des schémas	15
	4.8	Les rôles	
	4.9	Conclusion	16
5	Les	tâches planifiées	17
	5.1	Introduction	17
	5.2	Configuration de la messagerie (se fait une fois)	17
	5.3	Les opérateurs	17
	5.4	Les travaux	18
	5.5	Les alertes	18
	5.6	Bilan	19
6	Tran	nsfert de données	20
	6.1	Import / Export de données	20
	6.2	L'utilitaire BCP	20
	6.2	ccic	20

# 1 Présentation

## 1.1 Introduction

**OLTP**: Bases de données contenant les données de travail (type le plus fréquent)

**OLAP**: Bases de données contenant les statistiques (BI: Business Intelligent)

Outils : le plus utilisé = Management Studio ; il y en a de nombreux autres +/- utiles

On peut réaliser les actions d'administration en ligne de commande ou en graphique

**But d'un SGBDR** : stocker les données, vérifier les contraintes d'intégrité, garantir la cohérence des données, assurer les relations.

Modes d'authentification : mixte (SQL server + Windows) ou juste Windows :

> Sécurité > Connexion > sa (compte SQL Serveur)

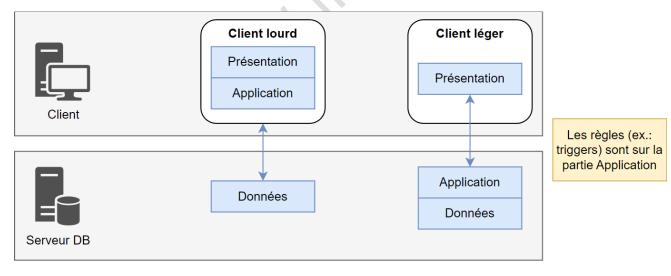
Gérer les services (MSSQLSERVER = instance SQL Server ; plusieurs instances possibles) :

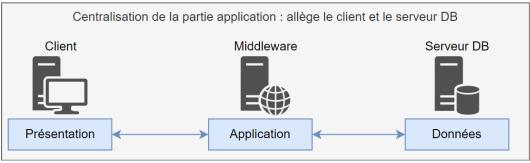
> SQL Server > Outils de configuration > Gestionnaire de configuration SQL Server

#### Attention: un seul Service Agent par instance.

- Système d'exploitation des clients : Windows, Linux (via Java), etc.
- Système d'exploitation pour le serveur : <u>dépend de la version de SQL Server</u>
- Moteur : un par instance
- Service : MYSQLSERVER est le nom obligatoire de la première instance
- Agent : pour les tâches planifiées

# 1.2 Fonctionnement





Fonctionnement global de SQL Server

Langage des requêtes : Transact-SQL

#### Les composants et l'outil de gestion

- Connectivité client (TCP/IP par défaut), le client doit avoir le driver installé
- Outils de gestion (Management Studio, etc.)
- Possibilité de passer une DB de SQLServer à Oracle via Export/Import

Nom des objets : <nom\_d'instance>. <base\_de\_donnees>. <schema>.

Exemple: instance.TP.dbo.table1

#### Bases de données systèmes

- Master (fonctionnement de SQL Server)
- Model (contient les composants de toutes les futures BD)
- Tempdb (tables temporaires)
- Msdb (informations liées à l'agent)

On travaille sur des vues des BD systèmes : SYS et INFORMATION\_SCHEMA

## 1.3 BD SQL Server

#### Composants

## Tables systèmes :

Éviter de travailler avec les tables systèmes. A la place, travailler avec le schéma SYS.

SELECT \* FROM sys.sysdatabases : liste des BD (systèmes + BD utilisateurs)

#### Procédures stockées (sp\_xxx):

Afficher des éléments, modifier des tables systèmes exec NomProcedure param1, param2 exec sp\_helpdb [nomBD]

# Le catalogue :

Vues systèmes permettant d'obtenir des informations système

## Fonctions systèmes :

Permettent de récupérer des valeurs concernant la BD sur laquelle on travaille

SELECT DB\_ID; : Identifiant de la base de données actuellement sélectionnée

#### Schéma d'information:

Visualisation de paramètres, indépendamment des tables systèmes SELECT \* FROM INFORMATION\_SCHEMA.xxx

#### Les taches de l'administrateur

- Gérer les Services SQL Server, les instances SQL Server, la configuration du réseau
- Mise en place de sauvegardes / restaurations
- Disponibilité des données (par rapport à la politique de l'entreprise)
- Import / export des données

# 2 Installation et configuration

## 2.1 Installer SQL Server

Nota: SQL Server 2022 est aussi disponible pour linux ainsi que pour les containers Docker.

Configuration logicielle et matérielle requises : SQL Server 2022 : Configurations logicielle et matérielle requises

Comparaison des versions de SQL Server 2017 à 2022 : <u>SQL Server 2022—Comparison</u>

Comparaison des fonctionnalités : Editions and supported features of SQL Server 2022 - SQL Server

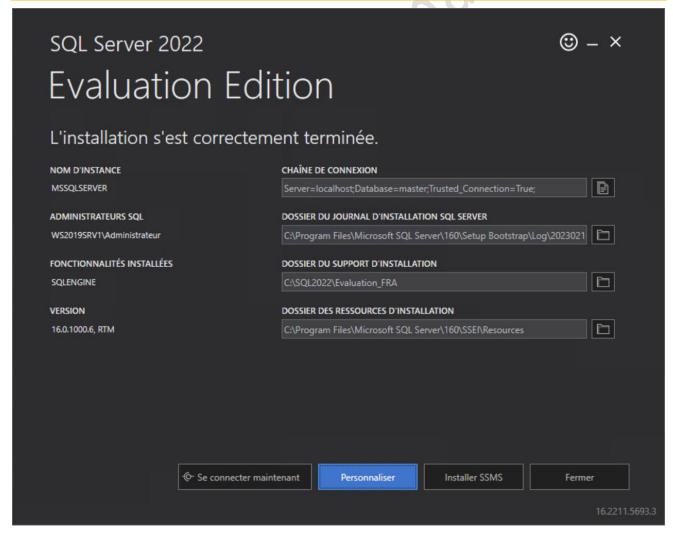
Fonctionnalité	Enterprise	Standard	Web	Express
Nombre de CPU / instance	Maximum de l'OS	4 sockets ou 24 cœurs	4 sockets ou 16 cœurs	1 socket ou 4 cœurs
Mémoire Max / instance	Maximum de l'OS	128 Go	64 Go	1410 Mo

Limites de fonctionnalités pour SQL Server 2022

## 2.2 Déroulement de l'installation

J'installe SQL Server 2022 (version 16.2211.5693.3 - installation de base) sur Windows Server 2019.

Il est nécessaire d'effectuer toutes les mises à jour du système d'exploitation (et de redémarrer si besoin après) avant d'installer le logiciel.



Une fois l'installation terminée, cliquez sur « Installer SSMS » ou « Fermer »

#### Règles de pare-feu :

Source : Configurer le Pare-feu Windows pour autoriser l'accès à SQL Server

Par défaut, SQL Server utilise les port 1433 et 1434

Script PowerShell pour ajouter les règles de pare-feu :

New-NetFirewallRule -DisplayName "SQLServer default instance" -Direction Inbound -LocalPort 1433 - Protocol TCP -Action Allow

New-NetFirewallRule -DisplayName "SQLServer Browser service" -Direction Inbound -LocalPort 1434 - Protocol UDP -Action Allow

Type d'installation : la première instance s'appelle toujours MSSQLSERVER

Choix de l'édition : vous pouvez sélectionner la version que vous souhaitez

#### **Installer SSMS**

Télécharger et installer **SSMS** : <u>SQL Server Management Studio</u>

#### Ajout / suppression de composants à SQL Server :

Dans le Centre d'installation SQL Server, sélectionner Nouvelle installation autonome... puis sélectionner l'emplacement de votre média d'installation (C:\SQL2022\Evaluation\_FRA dans mon cas) et faites Suivant jusqu'à Sélection de fonctionnalités.

#### Vérifier l'installation :

C:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSERVER\MSSQL\DATA: emplacement des données et fichiers journaux par défaut, outils, etc.

#### Modifier la sécurité

Dans SSMS, se connecter au serveur puis

Clic droit sur ☐ ₩S2019SRV1 (SQL Server 16.0.1000.6 - \ > Properties > Sécurity:

- Windows authentication mode: login/password utilisateur Windows (authentification conseillée)
- SQL Server and Windows auth. mode: Permet l'authentification mixte

Il faut ensuite redémarrer l'instance pour appliquer les modifications.

## Services Windows liés à SQL Server

Nom	Description	État	Type de démarrage	Ouvrir une session en tant que
SQL Server (INSTANCE_TEST)	Permet de stocker et de traiter des données, de c	En cours	Automatique (début différé)	NT Service\MSSQL\$INSTANCE_TE
SQL Server (MSSQLSERVER)	Permet de stocker et de traiter des données, de c	En cours	Automatique (début différé)	NT Service\MSSQLSERVER
SQL Server Agent (INSTANCE_TEST)	Exécute les travaux, analyse SQL Server, déclench		Manuel	NT Service\SQLAgent\$INSTANCE
SQL Server Agent (MSSQLSERVER)	Exécute les travaux, analyse SQL Server, déclench		Manuel	NT Service\SQLSERVERAGENT
SQL Server Browser	Fournit des informations de connexion SQL Serv	En cours	Automatique	Service local
SQL Server CEIP service (MSSQLSERVER)	CEIP service for Sql server	En cours	Automatique (début différé)	NT Service\SQLTELEMETRY

- SQL Server CEIP : service de collecte d'utilisation et de diagnostic
- SQL Server (INSTANCE) : Service de l'instance
- SQL Server Agent (INSTANCE) : Scheduler de l'instance
- SQL Server Browser : listener des connexions entrantes

# 3 Gestion de la base de données

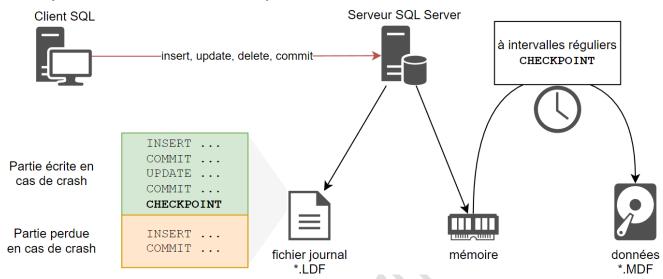
# 3.1 Organisation logique

## **Transactions**

Une transaction est un ensemble d'ordre (tout ou rien) : BEGIN TRAN, COMMIT TRAN, ROLLBACK TRAN, SAVE TRAN

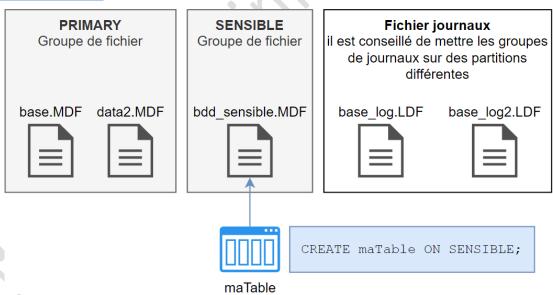
Tant que la transaction n'est pas validée, il y a un verrou sur la table : les autres utilisateurs n'y ont pas accès.

#### Fichiers journaux et fonctionnement des checkpoints



# 3.2 Organisation physique

#### Partitionnement des tables



On peut sauvegarder un groupe de fichier (périodicité à définir).

L'augmentation du nombre de groupe de fichiers améliore la performance

#### La base principale est dans PRIMARY

Pas de groupe de fichiers pour les journaux

Groupe de fichiers : même fonctionnement que les FS

Une table affectée à un groupe de fichier ne peut plus être déplacée

Pour déplacer les données d'une table dans une autre (ici de bdd\_sensible.client vers clients) :

INSERT INTO clients (nom, prenom)

SELECT nom, prenom

FROM dbb\_sensible.dbo.clients;

#### Intérêt du partitionnement

Selon la valeur d'une colonne (ex. : date), il est possible d'insérer les données dans un groupe de fichiers différent (archivage). L'intérêt est de pouvoir faire une optimisation des accès ou bien des sauvegardes plus nombreuses.

On peut définir la périodicité des sauvegardes selon les groupes de fichiers.

L'augmentation du nombre de groupe de fichiers améliore la performance.

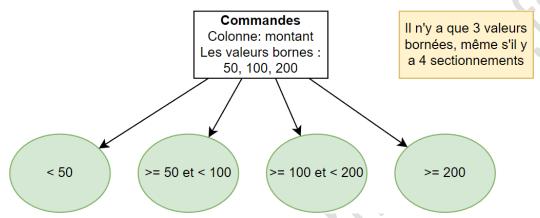
Un groupe de fichiers peut être en lecture seule.

#### Méthode de partitionnement

Il faut donc définir les colonnes ainsi que les valeurs à partitionner

Exemple pour une table « commandes »

Colonne = montant



Chaque groupe de fichiers est paramétrable différemment des autres

#### Pour l'utilisateur (logiquement) il n'y a qu'une table, physiquement il y a plusieurs tables.

#### **Etapes**

## 1. Définir les bornes (ici 3 bornes)

Découpe les données puis les envoie au schéma.

Si la première borne est inférieur strict : égalité à droite (RIGHT), sinon égalité à gauche (LEFT) CREATE PARTITION FUNCTION pfCommandes(int) AS RANGE RIGHT FOR VALUES (50, 100, 200);

La fonction peut ensuite être utilisée dans plusieurs schemas de partition.

## 2. Créer un schéma de partitionnement : définir les groupes (ici 4 groupes) :

## Nota: les filegroups doivent exister (avoir été créés).

Distribue les données aux filegroup (4 groupes pour cet exemple)

CREATE PARTITION SCHEME schemaCommande

AS PARTITION pfCommandes —— nom de la partition (groupe de fichiers)
TO (groupe1,groupe2,groupe3,groupe4); —— liste des filegroups qui seront utilisés

## 3. Créer la table :

Y définir le nom de la colonne sur laquelle le partitionnement s'effectue.

## La colonne doit faire partie de la clé primaire

CREATE TABLE commandes (

numero int identity,

montant money,

libelle nvarchar(100) NOT NULL,

CONSTRAINT pk\_commandes PRIMARY KEY(numero, montant) ) ON schemaCommande(montant);

#### Exemple : création d'un partitionnement sur plusieurs filegroups :

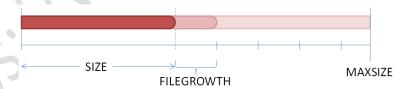
Dans SSMS, Créer la base SSMS en faisant clic-droit sur le dossier Databases > New database > SSMS

Source: CREATE PARTITION SCHEME (Transact-SQL)

```
/* Ajouter les dossiers de fichiers */
ALTER DATABASE SSMS ADD FILEGROUP CentreNord;
ALTER DATABASE SSMS ADD FILEGROUP IleDeFrance;
ALTER DATABASE SSMS ADD FILEGROUP NordEst;
ALTER DATABASE SSMS ADD FILEGROUP Ouest;
ALTER DATABASE SSMS ADD FILEGROUP Sud;
ALTER DATABASE SSMS ADD FILEGROUP SudEst;
ALTER DATABASE SSMS ADD FILEGROUP SudOuest;
/* Définir les bornes */
CREATE PARTITION FUNCTION pfCLIENTS(int)
AS RANGE RIGHT FOR VALUES (2,3,4,5,6,7);
/* Ajouter le schéma de partitionnement */
CREATE PARTITION SCHEME schemaCLIENTS
AS PARTITION pfCLIENTS
TO (IleDeFrance, CentreNord, NordEst, Ouest, SudOuest, SudEst, Sud);
/* Definir la colonne utilisée pour le partitionnement */
DROP TABLE CLIENTS;
CREATE TABLE CLIENTS (
               int
                              NOT NULL,
numero
               nvarchar(50)
                              NOT NULL,
nom
              nvarchar(50)
                             NOT NULL,
prenom
adresse
               nvarchar(100),
               char(5),
codepostal
ville
               nvarchar(50),
region
               int
                              NOT NULL,
CONSTRAINT pk_clients PRIMARY KEY(numero, region))
ON schemaCLIENTS (region);
```

#### 3.3 Création d'une base

- Nom unique (au sein de l'instance : NAME
- Groupe de fichiers : 0N
- Emplacement de la base : FILENAME
- Taille initiale (incompressible): SIZE
- Taille max et pas d'augmentation (filegrowth) : MAX SIZE, FILEGROWTH



Le FILEGROWTH représente le pas d'incrément de la taille, le MAXSIZE représente la taille maximale

#### Le script de création de base

Si on clique sur script:

- [ ] autour des noms d'objets, permet les espaces
- N devant les chaînes, met au format Unicode
- Création d'une BD dans un groupe de fichier défini :

```
CREATE DATABASE TSOL
ON PRIMARY (
NAME = 'TSQL',
FILENAME='C:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSERVER\MSSQL\DATA\TSQL.mdf',
SIZE = 5 MB)
LOG ON (
NAME = 'TSQL_log',
FILENAME = 'C:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSERVER\MSSQL\DATA\TSQL_log.ldf',
SIZE = 2 MB);
```

#### 2. Ajouter un groupe de fichiers :

```
USE TSQL;
GO
ALTER DATABASE TSQL ADD FILEGROUP TPTSQL;
GO
```

## 3. Ajouter un fichier à un groupe de fichiers existant :

```
ALTER DATABASE TSQL

ADD FILE (

NAME = 'TSQL_2',

FILENAME = 'C:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSERVER\MSSQL\DATA\TSQL_2.ndf',

SIZE = 5 MB ,

FILEGROWTH = 2 MB,

MAXSIZE = 10 MB )

TO FILEGROUP TPTSQL;

GO
```

## 4. Ajouter un fichier de log:

```
ALTER DATABASE TSQL

ADD LOG FILE (

NAME = 'TSQL_LOG2',

FILENAME = 'C:\Program Files\Microsoft SQL Server\MSSQL16.MSSQLSERVER\MSSQL\DATA\TSQL_LOG2.ldf',

SIZE = 10 MB ,

MAXSIZE = 10 MB );

GO
```

#### 5. Modifier un fichier:

```
ALTER DATABASE TSQL MODIFY FILE ( NAME = 'TSQL', MAXSIZE = UNLIMITED );
GO
```

## 6. Créer une table dans un système de fichiers défini :

```
CREATE TABLE CLIENTS (
             INT
                            CONSTRAINT pk_clients PRIMARY KEY,
numero
nom
              NVARCHAR(50)
                            NOT NULL,
              NVARCHAR(50)
                            NOT NULL,
prenom
             NVARCHAR(100),
adresse
codepostal
             CHAR(5),
ville
              NVARCHAR(50)
ON TPTSQL;
```

## 3.4 Gérer une base de données

Modifier la taille d'un fichier : SIZEaprès ≥ SIZEavant

Libérer de l'espace : utilitaire DBCC

Rapports: clic droit sur la BDD¹ > rapports > rapports standard > ...

Création d'un groupe de fichiers et ajout de fichiers

# 3.5 Supprimer une base

DROP DATABASE NomBase;

Ne pas être connecté à la base :

- Propriétés de la base > Options > Restriction d'accès > Single user
- Changer de base (en haut à gauche ou USE DATABASE master)

Opération irréversible, supprime physiquement tous les fichiers

<sup>&</sup>lt;sup>1</sup> BDD = base de données = DB = database

## 3.6 Structure des index

Source: CREATE INDEX (Transact-SQL)

- Augmente la vitesse et la taille de la base
- La clé primaire est toujours indexée (en clustered par défaut, mais peut être changé).
- Il est conseillé d'indexer les jointures externes (en nonclustered donc).
- Un seul index ordonnés (CLUSTERED) par table
- Index non-ordonnés (NONCLUSTERED)

#### Ajout d'un index à une colonne :

CREATE NONCLUSTERED INDEX ix\_marque\_produits
ON produits (marque ASC);

## 3.7 Compression des données

Clic droit sur tables > stockage > gérer la compression

Libérer de l'espace : utilitaire DBCC (RI p86-87)

SHRINKDATABASE: Compacte tous les fichiers de la base

SHRINKFILE: Compacte une partie des fichiers de la base

AUTO\_SHRINK: propriétés de la base > options > réduction automatique

SHRINKFILE: Compacte une partie des fichiers de la base

DBCC SHRINKFILE('TSQL', 2000000)

G0

AUTO\_SHRINK: Compactage automatique

Réduit automatiquement le fichier dès qu'il atteint la taille critique de 25% d'espace libre

ALTER DATABASE TSQL SET AUTO\_SHRINK ON

G0

# 4 Gestion de la sécurité d'accès

## 4.1 Introduction

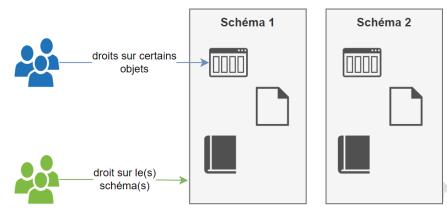


Illustration des droits

Le paramétrage de la sécurité se fait instance par instance :

Instance (clic-droit) > propriétés > sécurité: Connexion: Windows | Mixte (Windows + SQL Server)

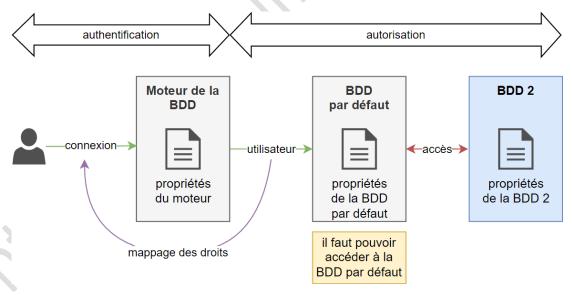
Instance (clic-droit) > propriétés > Permissions

- Grant : autorisation d'accéder
- With grant : peut attribuer les droits à d'autres utilisateurs
- Deny : interdiction d'accéder

Base de données (clic-droit) > propriétés > Permissions

Droits des utilisateurs de la base de données uniquement

## 4.2 Gestion des accès



Deux niveaux de sécurité : au niveau de l'instance (du serveur) + au niveau de chaque base

Une connexion est mappée à un utilisateur.

Un utilisateur est associé à une base de données par défaut (BDD utilisée).

## L'utilisateur peut avoir des droits :

- Sur la Base de données
- Sur certains objets de la Base de données (schéma, tables, vues, colonnes)
- Sur le serveur
- Via les rôles (offrant des privilèges)

Les propriétés de la BD donnent des droits à un utilisateur sur cette base de données.

#### Les rôles :

- Rôles serveurs (déjà définis)
- Rôles de bases de données (personnalisables)

#### Etapes de la création d'une connexion SQL Server

- Créer une connexion : soit Windows, soit SQL Server (si mixte)
- Créer un mappage
- Donner des autorisations

# 4.3 Création de la connexion (à l'instance / au serveur)

Il faut être membre de sysadmin ou de securityadmin pour pouvoir ajouter des connexions

#### Ajouter une nouvelle connexion, et lier cette connexion à une base :

```
Instance > Sécurité> Logins > New login
```

Il faut définir une base par défaut. Eviter master (car guest y a un compte activé par défaut)

On ne peut pas encore se connecter à la base qui a été définie par défaut

Désactiver une connexion : ALTER LOGIN xxx ENABLE | DISABLE;

## Mappage « automatique »:

- Pour le rôle sysadmin, c'est le compte utilisateur db\_owner qui est mappé
- Si on utilise « master » comme base par défaut, le mappage se fait car l'utilisateur « guest » est actif.
- Si on active « guest » sur la BD, le mappage est automatique

#### Si on active l'utilisateur guest sur une BD, n'importe qui peut s'y connecter

Pour activer 'guest', il faut ajouter l'utilisateur 'guest' et octroyer 'connecter'

```
Source : <u>CREATE LOGIN (Transact-SQL)</u>
```

```
/* definir une nouvelle connexion */
CREATE LOGIN odehecq
    WITH PASSWORD='qceheDo123',
    DEFAULT_DATABASE=TSQL;
G0
/* intégrer des comptes d'utilisateurs Windows */
CREATE LOGIN [WS2019SRV1\jdupont]
    FROM WINDOWS
    WITH DEFAULT_DATABASE=TSQL;
G0
```

# 4.4 Création du mappage

Source: CREATE USER (Transact-SQL)

Il faut lier un utilisateur de BD à un login de connexion, et lui affecter des rôles :

```
Instance > Base de données > Permissions > Logins > New login
```

CREATE USER nomDUtilisateur FOR LOGIN nomDeConnexion;

Une connexion doit être mappée à un utilisateur pour pouvoir se connecter à une base de données

# 4.5 Les autorisations (utilisateurs de BD) au niveau de la base de données

Pour pouvoir gérer les accès à la base, il faut disposer du rôle db\_owner (propriétaire de la base) ou db\_accessadmin.

Pour voir le résultat obtenu / modifier les autorisations :

```
Base de données > Propriétés > Autorisations > Utilisateur
```

Par défaut il n'y a que 'se connecter'

Utilisation des mots clé GRANT, REVOKE, DENY pour modifier des droits sur des objets sécurisables, sur des objets, ou sur toute la base de données

Autorisations possibles : créer une table, créer un schéma, exécuter (une procédure stockée), insérer ...

## Octroyer des droits :

Source: GRANT (Transact-SQL)

Octroyer (donne le droit de faire des actions)

GRAND CREATE TABLE TO AR12; : donne à AR12 le droit de 'create table'

#### Octroyer le transfert de droits

Octroyer doit déjà être coché (prérequis pour l'attribution de ce droit).

Permet à l'utilisateur d'attribuer l'autorisation sélectionnée à d'autres utilisateurs de cetteBD.

GRAND...TO AR12 WITH GRANT OPTION; : donne à AR12 le droit d'attribuer le droit

## Refuser (sur certains objets « éléments sécurisables »)

```
Instance > Base de données > Sécurité > Utilisateurs > éléments sécurisable
```

Sélectionner l'objet et donner (ou enlever) des droits sur certains objets de la table

Source : <u>DENY (Transact-SQL)</u>

« Refuser » (DENY) est prioritaire sur « octroyer » (GRANT) et sur « avec autorisations » (WITH GRANT OPTION)

DENY SELECT ON table1 TO AR12;

#### Modifier des autorisations

Annuler une allocation de droits : REVOKE [GRANT OPTION | GRANT] TO AR12

Supprimer les droits qui ont été octroyés par un utilisateur : option CASCADE

Pour donner à un utilisateur le droit de créer des tables, il faut lui donner le droit de modifier le schéma

#### Transact SQL:

```
/* Définir les utilisateurs de bases de données */
USE [TSQL]
CREATE USER [jdupont] FOR LOGIN [WS2019SRV1\jdupont];
/* Travailler avec le compte Guest */
GRANT CONNECT TO [guest];
/* Accorder des droits spécifiques */
use [TSQL]
GO
GRANT CREATE TABLE TO [jdupont]
GRANT CREATE VIEW TO [jdupont]
GRANT CREATE PROCEDURE TO [jdupont]
/* Interdire l'utilisation de certains privilèges */
use [TSQL]
GO
DENY CREATE TABLE TO [Léon]
DENY CREATE FUNCTION TO [Léon]
```

## 4.6 Au niveau du serveur

Il vaut mieux passer par les rôles, plus simple.

## 4.7 Gestion des schémas

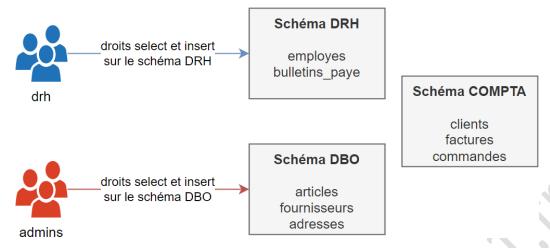


Illustration de l'utilisation des droits sur les schémas

Un schéma est un rassemblement logique d'objets. Le but est de permettre une administration plus facile des objets contenus dans ce schéma.

#### Ajouter de nouveaux schémas (ne pas toucher aux pré-existants) :

Source: CREATE SCHEMA (Transact-SQL)

Instance > Base de données > Sécurité > Schéma

CREATE SCHEMA nomSCHEMA

AUTHORIZATION nomProprietaire

Vérifier: SELECT nomSchema.monObjet

Déplacer un objet d'un schéma à un autre : ALTER SCHEMA nomSchema TRANSFER nomObjet

#### 4.8 Les rôles

Un rôle est un ensemble de permissions.

Il est plus simple de passer par des rôles quand plusieurs utilisateurs ont les mêmes autorisations.

## Rôle serveur

```
Instance > Security > Logins > utilisateur > (clic droit) > properties > Server Roles
```

Permet d'ajouter ou de supprimer des membres à un rôle

• Droit sysadmin = donner le droit admin de server

Il n'est pas possible d'ajouter d'autres rôles serveurs (à part pour les mettre membre de ceux existants).

#### Rôle de base de données

Rôle db\_owner: administrateur de la base de données

Database > Security > Roles > Database Roles

- Le rôle db\_datareader permet de faire des SELECT de toute les tables de la BD
- Le rôle db\_datawriter permet de faire INSERT/UPDATE/DELETE des données
- Le rôle db\_ddladmin permet de faire CREATE/ALTER/DROP des objets de la BD

Les rôles personnalisés sont utilisés uniquement au niveau de la base de données, et jamais au niveau serveur (instance)

## Rôle d'application

**Sont définis au niveau de la base de données**. A créer soi-même. Pour qu'une application puisse avoir le droit de faire certaines choses, quel que soit l'utilisateur qui exécute l'application.

Un rôle d'application ne possède aucun membre, et est protégé par un mot de passe.

# Transact SQL: /\* Définir un rôle de base de données \*/

```
USE [TSQL];
CREATE ROLE [roleTSQL];
ALTER AUTHORIZATION ON SCHEMA::[db_datareader] TO [roleTSQL]
/* Accorder des rôles de BD à des utilisateurs */
USE [TSQL];
G<sub>0</sub>
EXEC sp_addrolemember 'db_datareader', 'jdupont';
EXEC sp_addrolemember 'db_datawriter', 'Guest'
EXEC sp_addrolemember 'roleTSQL', 'Guest'
/* Définir un rôle d'application */
USE [SSMS];
G0
CREATE APPLICATION ROLE [roleApplication] WITH PASSWORD = 'SSMS';
ALTER AUTHORIZATION ON SCHEMA::[db_ddladmin] TO [roleApplication];
GRANT ALTER ANY SCHEMA TO [roleApplication];
/* Utiliser le rôle d'application*/
sp_setapprole 'roleApplication','SSMS'
CREATE TABLE TTEST(C int);
select USER;
```

#### Conclusion 4.9

Un utilisateur dispose d'un ensemble de privilèges accordés :

- Directement à la connexion utilisée
- Indirectement par l'intermédiaire d'un rôle fixe de serveur accordé à la connexion
- Indirectement par l'intermédiaire des rôles accordés à l'utilisateur de la base de données
- Directement à l'utilisateur de la base de données

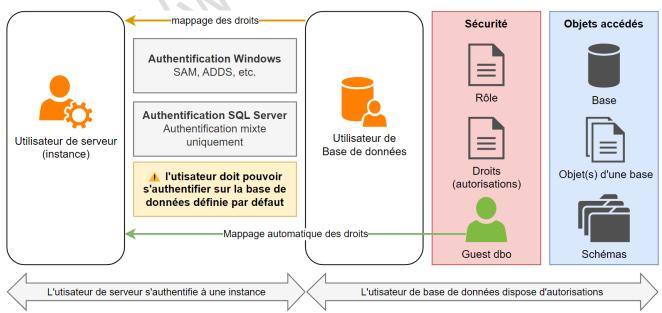


Schéma récapitulatif de la gestion des droits

# 5 Les tâches planifiées

#### 5.1 Introduction

Automatisation: simplification, diminution du travail; éviter les erreurs (oublis, etc.).

Anticiper: mise en place d'alertes (fichier plein par exemple).

L'agent SQL Server doit être démarré afin que les taches soient gérées.

## L'agent SQL Server travaille avec :

- L'Observateur des événements pour la gestion des erreurs SQL Server,
- L'Analyseur de performances pour la gestion des alertes sur des conditions de performances,
- La base MSDB afin de connaître la réponse à appliquer face à une alerte, ou bien les tâches planifiées à exécuter.

Msdb stocke les tâches planifiées et les opérateurs : si on la sauvegarde, on sauvegarde ça

# 5.2 Configuration de la messagerie (se fait une fois)

```
Instance > Mangement > Database Mail > Configure Database Mail
```

Pour établir la communication avec les opérateurs, l'agent SQL Server dispose de trois canaux de communication : le mail, la radiomessagerie (bipeur) et le message via le réseau (net send).

# 5.3 Les opérateurs

Sources: sp\_add\_operator (Transact-SQL) et sp\_help\_operator (Transact-SQL)

#### Ajout d'un opérateur :

```
USE msbd;
G0
EXEC dbo.sp_add_operator @name = N'toto', @email_address = N'toto@toto.fr';
G0
/* paramètres de l'opérateur toto */
EXEC sp_help_operator @operator_name = N'toto';
G0
```

# Modification de l'opérateur (source : sp\_update\_operator)

sp\_update\_operator

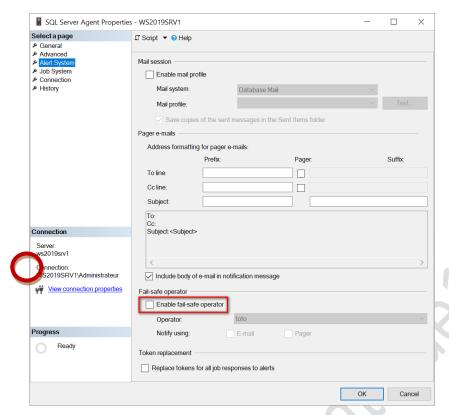
Suppression de l'opérateur (source : sp\_delete\_operator)

sp\_delete\_operator

#### Ajout d'un opérateur de prévention :

A faire lorsqu'on définit une signalisation des incidents par messagerie

- 1. Sous l'instance, Sélectionner l'objet SQL Server Agent > clic-droit > Start pour le démarrer
- 2. Sélectionner à nouveau l'objet SQL Server Agent > clic-droit > Properties
  - 3. Dans la page Alert System, activez Enable fail-safe operator et sélectionner la méthode pour le joindre



Définir un opérateur de prévention

## 5.4 Les travaux

C'est un ensemble de tâches.

La création d'un travail n'est possible que pour un administrateur du système (**sysdamin**) ou bien pour un utilisateur membre de l'un des trois rôles de base de données liés à SQL Server Agent.

Le propriétaire du travail et l'administrateur peuvent le modifier.

Tous les travaux sont stockés au sein de la table sysjobs dans la base msdb.

Les travaux ne peuvent s'exécuter que si le service SQLServerAgent est lancé.

## Création d'un travail

SQL Server Agent > Clic droit > New > Job

- Créer une nouvelle tâche, la nommer, l'activer
- Ajouter des étapes (Steps) de scripts (transact-SQL)
- Définir le comportement (Notifications) en cas de succès/échec/job terminé et sélectionner la méthode de notification (mail, écriture dans les journaux d'évènements)
- Définir la planification (Schedules) : une fois, périodique, etc.
- Définir les alertes (en cas d'échec)

## Les étapes peuvent être du type :

- SQL Server Integration Service Package : import/export de données
- Script PowerShell
- Script Transact-SQL
- Commande du système d'exploitation (.cmd)

## 5.5 Les alertes

Les alertes vont être définies afin de déclencher un traitement automatique pour corriger le problème et/ou avertir un opérateur.

Les erreurs sont inscrites dans l'observateur d'évènements, journal 'application'.

## Liste des messages pouvant être entrés dans l'observateur d'évènements :

SELECT \* FROM sys.sysmessages [WHERE mslangid = 1036] -- 1036 = french

#### Créer un message d'alerte personnalisé

Source: sp addmessage

USE msdb;

G0

EXEC sp\_addmessage @msgnum = 50001, @severity = 10, @msgtext=N'Sample message', @lang='us\_english'; EXEC sp\_addmessage @msgnum=50001, @severity=10, @msgtext=N'Message d''exemple', @lang='French';

- Les alertes personalisées doivent avoir un numéro supérieur à 50000
- si severity > 19: inscrit dans le journal d'observateur des évènements
- with log = true: inscrit une ligne dans la table des logs
- <u>\( \lambda \)</u> il faut OBLIGATOIREMENT ajouter la version us\_english avant de mettre la version française

Lever une erreur (source : <u>RAISERROR</u>)

RAISERROR(50001, 10, 1);

- générer l'erreur 50001
- severity : sévérité de l'erreur vis à vis du script (titre indicatif)
- state : si plusieurs fois la même erreur dans le même script, permet de la localiser

Lister les messages : select \* from sys.messages;

#### Types d'événement qui sont inscrits dans l'observateur des événements

- Les messages dont le niveau de gravité et supérieur à 19. Ces messages sont également stockés dans la table **sys.messages** de la base **Master**. Pour forcer un message dont le niveau de gravité est inférieur à 19 à s'inscrire dans le journal, il faut exécuter la procédure **sp\_altermessage**.
- Toutes les instructions RAISERROR avec l'option WITH LOG.

#### **Gestion des alertes**

L'agent lit l'observateur des évènements en temps réel et définir une alerte quand les paramètres correspondent.

Seuls les messages inscrits dans l'observateur des événements peuvent déclencher une alerte.

#### Créer une alerte :

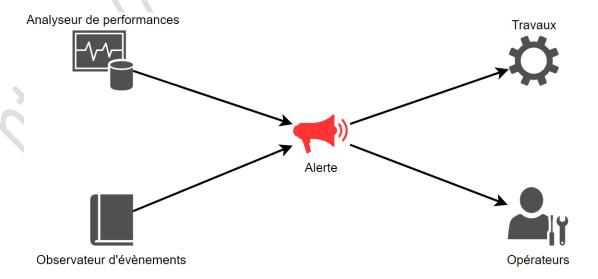
SQL Server Agent > Clic droit > New > Alert

Définir comment se déclenche l'alerte, définir l'opérateur à notifier, possibilité de définir un job à exécuter

Alerte > Type > SQL Server performance condition alert (condition de performances SQL server):

 Liste de tous les compteurs de SQL Serveur : exemple SQLServerDataBase, PercentLogUsed, instance : Test, compteur supérieur à 50M%

## 5.6 Bilan



# 6 Transfert de données

# 6.1 Import / Export de données

L'Assistant Importation et Exportation SQL Server constitue la méthode la plus simple pour copier des données entre des sources de données et pour construire des packages de base.

Source: http://technet.microsoft.com/fr-fr/library/ms140052.aspx

#### 6.2 L'utilitaire BCP

L'utilitaire **bcp** copie en bloc des données entre une instance de Microsoft SQL Server et un fichier de données dans un format spécifié par l'utilisateur. L'utilitaire **bcp** permet d'importer un grand nombre de nouvelles lignes dans des tables SQL Server ou d'exporter des données de tables dans des fichiers de données. Sauf lorsqu'il est utilisé avec l'option **queryout**, l'utilitaire ne nécessite aucune connaissance de Transact-SQL. Pour importer des données dans une table, vous devez utiliser un fichier de format créé pour cette table ou comprendre la structure de la table et les types de données valides pour ses colonnes.

Pour connaître les conventions de syntaxe utilisées pour la syntaxe **bcp**, consultez <u>Conventions de la syntaxe Transact-SQL (Transact-SQL)</u>.

Source: http://msdn.microsoft.com/fr-fr/library/ms162802.aspx

## 6.3 SSIS

SQL Server Integration Services (SSIS) est une plateforme qui permet de créer des solutions de transformation de données et d'intégration de données au niveau de l'entreprise. Integration Services vous permet de résoudre des problèmes professionnels complexes en copiant ou en téléchargeant des fichiers, en envoyant des messages électroniques en réponse à des événements, en mettant à jour des entrepôts de données, en nettoyant et en explorant des données et en gérant des données et des objets SQL Server. Les packages peuvent fonctionner en mode autonome ou de concert avec d'autres packages en réponse à des besoins professionnels complexes. Integration Services peut extraire et transformer des données à partir d'un éventail de sources, par exemple des fichiers de données XML, des fichiers plats et des sources de données relationnelles, puis charger les données dans une ou plusieurs destinations.

Integration Services inclut un ensemble riche de tâches et de transformations intégrées, des outils pour construire des packages, et le service Integration Services permettant d'exécuter et de gérer des packages. Vous pouvez faire appel aux outils graphiques Integration Services pour créer des solutions sans écrire une seule ligne de code. Vous pouvez également programmer le modèle objet Integration Services étendu pour créer des packages par programme et des tâches personnalisées de code et d'autres objets de package.

Source: http://technet.microsoft.com/fr-fr/library/ms141026.aspx