

SQL SERVER - LA SÉCURITÉ

P/2	Introduction
P/5	Vues système
P/8	Accès au serveur
P/10	Gestion des connexions
P/13	Gestion des utilisateurs de BDD
P/15	Gestion des Schémas
P/17	Gestion des droits
	Droit d'instruction
	Droit d'utilisation
	Droits au niveau base de données
	Droits au niveau serveur
P/22	Les rôles
	Rôles prédéfinis serveur
	Rôles prédéfinis base de données
	Rôles définis par l'utilisateur
P/27	Crédits

Introduction

La sécurité est un aspect primordial de la gestion des bases de données. Sans sécurité, les données sont exposées à des risques de tous types.

Il y a dans SQL Server, trois types d'objets qui vont nous permettre de sécuriser nos bases de données, et donc garantir une plus grande chance d'intégrités

pour nos données :



Les entités de sécurité

Ce sont les comptes de sécurité qui disposent d'un accès au serveur.



Les sécurisables

Les éléments sécurisables sont les ressources auxquelles le système d'autorisation du moteur de base de données régule l'accès. Par exemple, une table est un élément sécurisable. Certains éléments sécurisables peuvent être contenus dans d'autres, de façon à créer des hiérarchies imbriquées appelées « étendues », pouvant elles-mêmes être sécurisées.

Etendue sécurisable	Éléments sécurisables
Serveur	Groupe de disponibilité Point de terminaison Connexion Rôle du serveur Base de données
Base de données	Rôle d'application Assembly Clé asymétrique Certificat Contrat Catalogue FullText Liste de mots vides type de message Liaisons de service distant Rôle (de base de données) Routage schéma Liste des propriétés de recherche Service Clé symétrique Utilisateur
Schéma	Type Collection de schémas XML Objet - comprend les membres suivants : <ul style="list-style-type: none">• Agrégat• Fonction• Procédure• File d'attente• Synonyme• Table de charge de travail• Affichage• Table externe



Les autorisations

celles-ci sont accordées aux entités de sécurité afin de pouvoir travailler avec les sécurisables.



The diagram illustrates the architecture of Microsoft SQL Server, divided into two main levels: SQL Server Level and Database Level.

SQL Server Level:

- Fixed server role
- SQL Server Login
- User-defined fixed server role

Database Level:

- Fixed database role
- Database user
- Application role
- User-defined database role

Microsoft SQL Server:

- SQL Server Login
- Endpoint
- Database

Database:

- Application role
- Assembly
- Asymmetric key
- Certificate
- Contract
- Full-text catalog
- Message type
- Remote service binding
- Role
- Route
- Service
- Symmetric key
- User
- Schema

Schema:

- Table
- View
- Function
- Procedure
- Queue
- Synonym
- Type
- XML schema collection

Vues système

Il est possible d'obtenir un catalogue complet des **utilisateurs** et de leurs **privilèges**, grâce aux **vues système**.

Voici quelques unes de ces vues :



Sys.server_permissions

Permissions au niveau serveur.

```
select * from Sys.server_permissions
```

	class	class_desc	major_id	minor_id	grantee_principal_id	grantor_principal_id	type	permission_name	state	state_desc
1	100	SERVER	0	0	1	1	COSQ	CONNECT SQL	G	GRANT
2	100	SERVER	0	0	2	1	VWDB	VIEW ANY DATABASE	G	GRANT
3	100	SERVER	0	0	101	1	VWAD	VIEW ANY DEFINITION	G	GRANT
4	100	SERVER	0	0	102	1	AUTH	AUTHENTICATE SERVER	G	GRANT
5	100	SERVER	0	0	102	1	VWAD	VIEW ANY DEFINITION	G	GRANT
6	100	SERVER	0	0	102	1	VWSS	VIEW SERVER STATE	G	GRANT
7	100	SERVER	0	0	103	1	AUTH	AUTHENTICATE SERVER	G	GRANT
8	100	SERVER	0	0	105	1	CL	CONTROL SERVER	G	GRANT



Sys.server_principals

Entités de sécurité au niveau serveur.

```
select * from Sys.server_principals
```

	name	principal_id	sid	type	type_desc	is_disabled	create_date	modify_date	default_database_name	default_language_name	credential_id	orphan
1	sa	1	0x01	S	SQL_LOGIN	0	2003-04-08 09:10:35.460	2020-06-17 15:33:15.507	master	Français	NULL	0
2	public	2	0x02	R	SERVER_ROLE	0	2009-04-13 12:59:06.030	2009-04-13 12:59:06.030	NULL	NULL	NULL	1
3	sysadmin	3	0x03	R	SERVER_ROLE	0	2009-04-13 12:59:06.030	2009-04-13 12:59:06.030	NULL	NULL	NULL	1
4	securityadmin	4	0x04	R	SERVER_ROLE	0	2009-04-13 12:59:06.030	2009-04-13 12:59:06.030	NULL	NULL	NULL	1
5	serveradmin	5	0x05	R	SERVER_ROLE	0	2009-04-13 12:59:06.030	2009-04-13 12:59:06.030	NULL	NULL	NULL	1
6	setupadmin	6	0x06	R	SERVER_ROLE	0	2009-04-13 12:59:06.030	2009-04-13 12:59:06.030	NULL	NULL	NULL	1



Sys.sql_logins

Connexions au niveau serveur.

```
select * from Sys.sql_logins
```

	name	principal_id	sid	type	type_desc	is_disabled	create_date	modify_date	default_database
1	sa	1	0x01	S	SQL_LOGIN	0	2003-04-08 09:10:35.460	2020-06-17 15:33:15.507	master
2	##MS_PolicyTsqlExecutionLogin##	257	0xCE62FF92465C5E4E90E9AC73CD01CA8D	S	SQL_LOGIN	1	2017-08-22 19:39:30.460	2019-08-01 12:29:44.717	master
3	javaSDBM	266	0x8E9276B1F1207A4985ADD8176669EC28	S	SQL_LOGIN	0	2019-07-31 11:38:58.393	2019-10-14 15:15:21.543	master
4	##MS_PolicyEventProcessingLogin##	267	0xCC15B5BB00FB3E4DAEF85CD4155DB61B	S	SQL_LOGIN	1	2019-08-01 12:29:44.697	2019-08-01 12:29:44.710	master
5	C_JAVA	270	0xB7AE51599534B449AA2EB56315ECCF8B	S	SQL_LOGIN	0	2020-03-12 08:20:31.077	2020-03-12 08:20:31.087	SDBM



Sys.Server_role_members

Bénéficiaires d'un rôle au niveau serveur.

```
select * from Sys.Server_role_members
```

	role_principal_id	member_principal_id
1	3	1
2	3	259
3	3	260
4	3	261
5	3	262
6	3	264



Sys.database_permissions

Permissions au niveau base de données.

```
select * from Sys.database_permissions
```

	class	class_desc	major_id	minor_id	grantee_principal_id	grantor_principal_id	type	permission_name	state	state_desc
1	0	DATABASE	0	0	0	1	VWCK	VIEW ANY COLUMN ENCRYPTION KEY DEFINITION	G	GRANT
2	0	DATABASE	0	0	0	1	VWCM	VIEW ANY COLUMN MASTER KEY DEFINITION	G	GRANT
3	0	DATABASE	0	0	1	1	CO	CONNECT	G	GRANT
4	1	OBJECT_OR_COLUMN	-599	0	0	1	SL	SELECT	G	GRANT
5	1	OBJECT_OR_COLUMN	-598	0	0	1	SL	SELECT	G	GRANT
6	1	OBJECT_OR_COLUMN	-596	0	0	1	SL	SELECT	G	GRANT
7	1	OBJECT_OR_COLUMN	-592	0	0	1	SL	SELECT	G	GRANT



Sys.database_principals

Entités de sécurité au niveau base de données.

```
select * from Sys.database_principals
```

	name	principal_id	type	type_desc	default_schema_name	create_date	modify_date	owning_principal_id	sid
1	public	0	R	DATABASE_ROLE	NULL	2003-04-08 09:10:42.317	2009-04-13 12:59:14.467	1	0x010500000000000009040000
2	dbo	1	U	WINDOWS_USER	dbo	2003-04-08 09:10:42.287	2020-09-18 10:35:53.200	NULL	0x010500000000000005150000
3	guest	2	S	SQL_USER	guest	2003-04-08 09:10:42.317	2003-04-08 09:10:42.317	NULL	0x00
4	INFORMATION_SCHEMA	3	S	SQL_USER	NULL	2009-04-13 12:59:11.717	2009-04-13 12:59:11.717	NULL	NULL
5	sys	4	S	SQL_USER	NULL	2009-04-13 12:59:11.717	2009-04-13 12:59:11.717	NULL	NULL
6	db_owner	16384	R	DATABASE_ROLE	NULL	2003-04-08 09:10:42.333	2009-04-13 12:59:14.467	1	0x010500000000000009040000



Sys.database_role_members

Bénéficiaires d'un rôle au niveau base de données.

```
select * from Sys.database_role_members
```

165 %

Résultats

Messages

	role_principal_id	member_principal_id
1	16384	1

Accès au serveur

Avant de pouvoir commencer à travailler avec les données de nos bases, il est impératif de se logger sur le serveur SQL. Cette étape permet de s'identifier au niveau du serveur SQL, afin de pouvoir exploiter les droits qui ont été attribués à notre connexion.

Dans SQL Server, il existe 2 modes d'authentification :



Mode de sécurité Windows

Les comptes de connexion sont définis à partir des utilisateurs et groupes Windows : SQL Server autorise les utilisateurs Windows à se connecter au serveur SQL, et ne gère pas les mots de passe



Mode de sécurité Mixte

C'est SQL Server qui se charge de vérifier que l'utilisateur existe et qu'il possède le bon mot de passe. Cela signifie que les utilisateurs sont entièrement gérés par SQL Server, autant les login que les mots de passe. Ce type d'identification est bien adapté pour une gestion des utilisateurs qui ne passent pas par une authentification Windows.

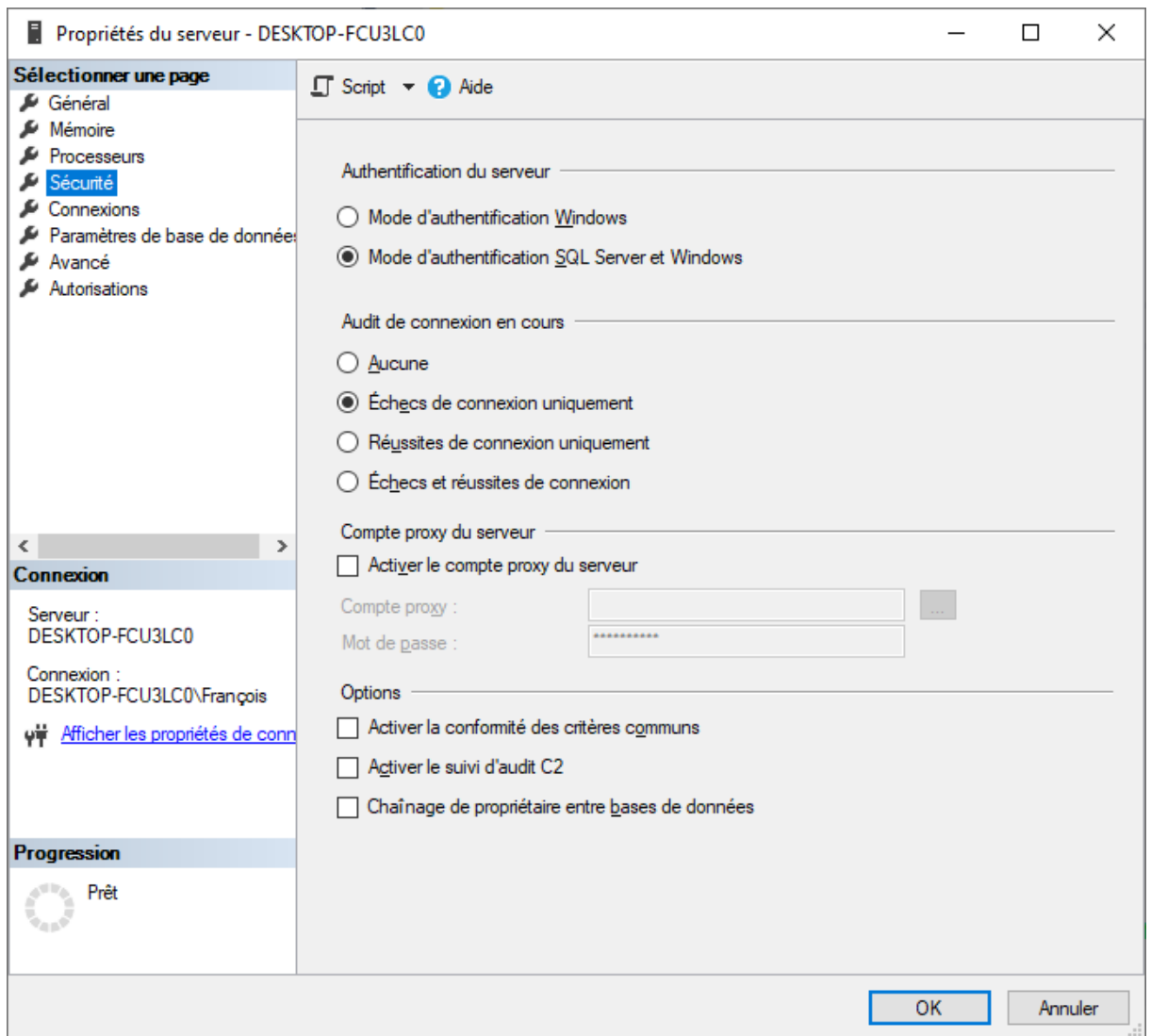


Changer de mode d'authentification

En principe, vous avez installé SQL Server avec l'authentification mixte. Si ça n'est pas le cas, vous pouvez changer ce paramètre grâce à SSMS :

- Dans le menu contextuel de votre instance de serveur, choisissez : **Propriétés**.
- Choisissez ensuite la page **Sécurité**.

Il ne vous reste plus maintenant qu'à changer le mode d'authentification.



Gestion des connexions

En authentification Windows, les noms de groupes ou les utilisateurs **doivent être les mêmes que sous Windows**.



Windows vs SQL

Les procédures de création, modification, suppression sont identiques pour les deux modes.

La seule différence est que :

- En authentification Windows, la gestion et la stratégie de mot de passe est déléguée à Windows.
- En authentification SQL, c'est à vous de mettre en place cette stratégie.



Avec SSMS

Dans le menu contextuel de votre instance de serveur, déployez le nœud **Sécurité**, puis choisissez **Connexions**.

Vous pouvez alors éditer ou supprimer une connexion existante, ou encore en créer une nouvelle, grâce à l'option **Nouvelle connexion** du menu contextuel.

Propriétés de la connexion - C_JAVA

Sélectionner une page

- Général
- Rôles du serveur
- Mappage d'utilisateur
- Éléments sécurisables
- État

Connexion

Script ? Aide

Nom d'accès :

☐ Authentification Windows
☒ Authentification SQL Server

Mot de passe :

Confirmer le mot de passe :

☐ Spécifier l'ancien mot de passe

Ancien mot de passe :

☐ Appliquer la stratégie de mot de passe
☐ Appliquer l'expiration du mot de passe
☐ L'utilisateur doit changer de mot de passe à la prochaine connexion

☐ Mappé au certificat
☐ Mappé à la clé asymétrique
☐ Mapper aux informations d'identification

Informations d'identification mappées

Informations ...	Fournisseur

Base de données par défaut :

Langue par défaut :

Progression

Prêt



Avec Transact SQL

Authentification Windows :

```
CREATE LOGIN login
FROM WINDOWS WITH DEFAULT_DATABASE=master,
DEFAULT_LANGUAGE=Français
```

Authentification SQL :

```
CREATE LOGIN login
WITH PASSWORD='password' MUST_CHANGE,
DEFAULT_DATABASE=Master,
DEFAULT_LANGUAGE=Français,
CHECK_EXPIRATION=OFF,
CHECK_POLICY=OFF
GO
```

Expression	Signification
FROM WINDOWS	Indique que la connexion est mappée sur une connexion Windows.
PASSWORD	Utilisez un mot de passe fort. Les informations de mot de passe stockées sont calculées à l'aide de l'algorithme SHA-512.
MUST_CHANGE	SQL Server demandera un nouveau mot de passe lors de la première utilisation de la connexion.
DEFAULT_LANGUAGE	Langue par défaut attribuée à la connexion. Si ce paramètre est omis, il est initialisé avec la langue par défaut du serveur.
CHECK_EXPIRATION	Si OFF, le mot de passe n'expire jamais.
CHECK_POLICY	Indique que les stratégies de mot de passe Windows de l'ordinateur sur lequel SQL Server s'exécute doivent s'appliquer à cette connexion

Gestion des utilisateurs de BDD

Après avoir créé un compte (identité de serveur) pour la connexion au serveur SQL, il convient de lui accorder des droits d'accès à une ou plusieurs bases de données.

- Les droits d'accès, d'écriture, de lecture sur les bases de données ne sont pas attribués aux connexions, mais aux utilisateurs de bases de données.
- Une connexion ne pourra exécuter des opérations sur une base que s'il existe un compte utilisateur défini sur cette base et qui lui est associé.

Il faut donc créer l'utilisateur de la base de données, puis l'associer à une connexion.



Avec SSMS

Dans le menu contextuel de votre base de données, déployez le nœud **Sécurité**, puis choisissez **Utilisateur**.

Vous pouvez alors éditer ou supprimer un utilisateur existant, ou encore en créer un nouveau, grâce à l'option **Nouvel utilisateur** du menu contextuel.

Utilisateur de la base de données - Nouveau

Sélectionner une page

- Général
- Schémas appartenant à un rôle
- Appartenance
- Éléments sécurisables
- Propriétés étendues

Script ? Aide

Type d'utilisateur :

Utilisateur SQL avec connexion

Nom d'utilisateur :

Nom de connexion :

Schéma par défaut :

Connexion

Serveur :
DESKTOP-FCU3LC0

Connexion :
JavaSDBM

[Afficher les propriétés de connexion](#)

Progression

Prêt

OK Annuler

Précisez dans un premier temps le nom de l'utilisateur de base de données et dans un second temps, la connexion serveur à lui associer.

Votre utilisateur est maintenant créé et mappé sur une connexion existante.



Avec Transact SQL

```
CREATE USER utilisateur
FOR LOGIN connexion
WITH DEFAULT_SCHEMA=nomSchema
```



Comptes utilisateurs par défaut

Chaque base de données de SQL Server dispose de 2 comptes utilisateurs par défaut : **dbo** et **guest**.



L'utilisateur dbo

Dans toutes les bases de données :

- le compte de connexion **sa** (System Administrator)
- les connexions disposant du rôle **sysadmin**,

sont mappés à un compte d'utilisateur spécial appelé **dbo**.

- tous les objets créés par ces comptes appartiendront à l'utilisateur **dbo**,
- tous les objets créés par d'autres utilisateurs appartiennent à l'utilisateur qui les créent.



L'utilisateur guest

Une fois qu'une connexion à une instance de SQL Server est établie, un compte d'utilisateur distinct doit exister dans chaque base de données à laquelle l'utilisateur doit accéder.

Exiger un compte d'utilisateur dans chaque base de données empêche les utilisateurs de se connecter à une instance de SQL Server et d'accéder à toutes les bases de données de ce serveur.

L'existence d'un compte d'utilisateur **guest** (invité) dans une base de données permet de contourner cette exigence en permettant à une connexion sans compte d'utilisateur de base de données d'y accéder.

Ce compte **guest** est créé par défaut dans toute nouvelle base de données, mais il est par défaut désactivé.

Gestion des Schémas

Un schéma est un ensemble logique d'objets à l'intérieur des bases de données sur le serveur.

Dès sa création, un utilisateur est obligatoirement mappé sur un schéma.

Si à la création de l'utilisateur, aucun nom de schéma n'est précisé, alors il sera mappé sur le schéma **dbo** par défaut.



Remarques

- Un même utilisateur de base de données peut-être le propriétaire de plusieurs schémas.
- Lorsqu'on attribue un schéma par défaut à un utilisateur, il n'est pas nécessaire que cet utilisateur fasse référence au nom du schéma lorsqu'il accède à un objet de ce schéma.
- En cas de conflit d'accès entre plusieurs objets de différents schémas, SQL Server utilisera en priorité l'objet qui existe dans le schéma **sys**, puis l'objet qui existe dans le **schéma par défaut** de l'utilisateur, et enfin l'objet qui existe dans le schéma **dbo**.



Avec SSMS

Dans le menu contextuel de votre base de données, déployez le nœud **Sécurité**, puis choisissez **Schéma**.

Vous pouvez alors éditer ou supprimer un schéma existant, ou encore en créer un nouveau, grâce à l'option **Nouvel schéma** du menu contextuel.

Schéma - Nouveau

Sélectionner une page

- Général
- Autorisations
- Propriétés étendues

Connexion

Serveur :
DESKTOP-FCU3LC0

Connexion :
DESKTOP-FCU3LC0\François

[Afficher les propriétés de connexion](#)

Progression

Prêt

Script ? Aide

Un schéma contient des objets de base de données tels que des tables, des vues et des procédures stockées. Le propriétaire d'un schéma peut être un utilisateur ou un rôle de base de données, ou un rôle d'application.

Nom du schéma :

Propriétaire du schéma :

Rechercher...

OK Annuler

Précisez dans un premier temps le nom du schéma, et dans un second temps, son propriétaire.



Avec Transact SQL

```
CREATE SCHEMA schema_name  
    AUTHORIZATION owner_name
```


Gestion des droits

Les droits sont les autorisations qui vont nous permettre de travailler avec notre base de données. Ils sont organisés de façon hiérarchique par rapport aux éléments sécurisables du serveur.



Remarque

L'attribution des droits peut être faite à tous les niveaux :

- Serveur,
- Base de données
- Schéma
- Objets.

Ils peuvent être attribués à :

- Un rôle
- Une connexion
- Un Utilisateur



GRANT, DENY, REVOKE

Il est possible de gérer ces permissions à l'aide de SSMS, ou encore grâce à 3 instructions de Transact SQL :

- **GRANT** : permet d'attribuer un privilège
- **REVOKE** : permet de retirer un privilège, si celui-ci a été attribué auparavant
- **DENY** : permet d'interdire un privilège, même si il a été attribué au travers d'un rôle

Droit d'instruction

Ces droits correspondent aux droits qui permettent de créer (mettre à jour, supprimer) de nouveau objets dans la base. Les utilisateurs qui possèdent de tels droits sont donc capable de créer leurs propres tables...

Voici les principaux droits disponibles :

- **CREATE DATABASE,**
- **CREATE TABLE,**
- **CREATE FUNCTION,**
- **CREATE PROCEDURE,**
- **CREATE VIEW,**
- **BACKUP DATABASE,**
- **BACKUP LOG**

- etc...

Droit d'utilisation

Les autorisations peuvent être gérées pour des objets spécifiques, d'un type particulier ou appartenant à un schéma spécifique, mais dépendant de la portée (voir schéma).

- Au niveau serveur, les autorisations portent sur les points de terminaison, les connexions et les rôles.
- Au niveau base de données, les autorisations portent sur des rôles, des schémas, des tables ...

L'accès à tous ces objets est contrôlé en octroyant, refusant ou annulant la possibilité d'utiliser certaines instructions ou procédures stockées ;

Les principaux droits d'utilisation :

- **INSERT,**
- **UPDATE,**
- **SELECT,**
- **DELETE,**
- **EXECUTE.**

Droits au niveau base de données

Les droits au niveau des bases de données vont donner des droits aux utilisateurs qui ne seront valables que sur une base de données précise. Au niveau base de données, il est possible de donner des droits à un utilisateur, à un schéma, à une assembly ou encore à un objet service broker.



Liste

- **ALTER**
- **ALTER ANY APPLICATION ROLE**
- **ALTER ANY ASSEMBLY**
- **ALTER ANY ASYMMETRIC KEY**
- **ALTER ANY CERTIFICATE**
- **ALTER ANY CONTRACT**
- **ALTER ANY DATABASE DDL TRIGGER**
- **ALTER ANY DATABASE EVENT NOTIFICATION**
- **ALTER ANY DATASPACE**
- **ALTER ANY FULLTEXT CATALOG**
- **ALTER ANY MESSAGE TYPE**
- **ALTER ANY REMOTE SERVICE BINDING**
- **ALTER ANY ROLE**
- **ALTER ANY ROUTE**

- ALTER ANY SCHEMA
- ALTER ANY SERVICE
- ALTER ANY SYMMETRIC KEY
- ALTER ANY USER
- AUTHENTICATE
- BACKUP DATABASE
- BACKUP LOG
- CHECKPOINT
- CONNECT
- CONNECT REPLICATION
- CONTROL
- CREATE AGGREGATE
- CREATE ASSEMBLY
- CREATE ASYMMETRIC KEY
- CREATE CERTIFICATE
- CREATE CONTRACT
- CREATE DATABASE
- CREATE DATABASE DDL EVENT NOTIFICATION
- CREATE DEFAULT
- CREATE FULLTEXT CATALOG
- CREATE FUNCTION
- CREATE MESSAGE TYPE
- CREATE PROCEDURE
- CREATE QUEUE
- CREATE REMOTE SERVICE BINDING
- CREATE ROLE
- CREATE ROUTE
- CREATE RULE
- CREATE SCHEMA
- CREATE SERVICE
- CREATE SYMMETRIC KEY
- CREATE SYNONYM
- CREATE TABLE
- CREATE TYPE
- CREATE VIEW
- CREATE XML SCHEMA COLLECTION
- DELETE

- EXECUTE
- INSERT
- REFERENCES
- SELECT
- SHOWPLAN
- SUBSCRIBE QUERY NOTIFICATIONS
- TAKE OWNERSHIP
- UPDATE
- VIEW DATABASE STATE
- VIEW DEFINITION

Droits au niveau serveur

Les privilèges au niveau serveur s'attribuent de la même manière que ceux de niveau base de données mais :

- Ce ne sont pas les mêmes.
- Ils ne sont pas attribués à un utilisateur, mais à une connexion.



Liste

- ADMINISTER BULK OPERATIONS
- ALTER ANY CONNECTION
- ALTER ANY CREDENTIAL
- ALTER ANY DATABASE
- ALTER ANY ENDPOINT
- ALTER ANY EVENT NOTIFICATION
- ALTER ANY LINKED SERVER
- ALTER ANY LOGIN
- ALTER RESOURCES
- ALTER SERVER STATE
- ALTER SETTINGS
- ALTER TRACE
- AUTHENTICATE SERVER
- CONNECT SQL
- CONTROL SERVER
- CREATE ANY DATABASE
- CREATE DDL EVENT NOTIFICATION
- CREATE ENDPOINT

- CREATE TRACE EVENT NOTIFICATION
- EXTERNAL ACCESS ASSEMBLY
- SHUTDOWN
- UNSAFE ASSEMBLY
- VIEW ANY DATABASE
- VIEW ANY DEFINITION
- VIEW SERVER STATE

Les rôles

Les rôles sont des sortes de groupements de droits.

On attribuera des des droits aux rôles, puis des rôles aux utilisateurs.

SQL Server fournit des rôles prédéfinis pour vous aider à gérer les autorisations

Rôles prédéfinis serveur

Les rôles de serveur fixes offrent des privilèges administratifs au niveau du serveur (administrateur système, créateur de base de données, responsable de la sécurité).

Rôles serveur	Description
sysadmin	Les membres du rôle serveur fixe sysadmin peuvent effectuer n'importe quelle activité sur le serveur.
serveradmin	Les membres du rôle serveur fixe serveradmin peuvent modifier les options de configuration à l'échelle du serveur et arrêter le serveur.
securityadmin	Les membres du rôle serveur fixe securityadmin gèrent les connexions et leurs propriétés. Ils peuvent attribuer des autorisations GRANT , DENY et REVOKE au niveau du serveur, et de la base de données, s'ils ont accès à une base de données. En outre, ils peuvent réinitialiser les mots de passe pour les connexions SQL Server. le securityadmin doit être traité comme équivalent au rôle sysadmin .
processadmin	Les membres du rôle serveur fixe processadmin peuvent mettre fin aux processus en cours d'exécution dans une instance de SQL Server.
setupadmin	Les membres du rôle serveur fixe setupadmin peuvent ajouter et supprimer des serveurs liés à l'aide d'instructions Transact-SQL. Cependant, l'appartenance au rôle sysadmin est nécessaire pour le faire au travers de Management Studio.
bulkadmin	Les membres du rôle serveur fixe bulkadmin peuvent exécuter l'instruction BULK INSERT .
diskadmin	Le rôle serveur fixe diskadmin permet de gérer les fichiers disque.
dbcreator	Les membres du rôle serveur fixe dbcreator peuvent créer, modifier, supprimer et restaurer n'importe quelle base de données.
public	Chaque connexion SQL Server appartient au rôle serveur public . Lorsqu'un principal de serveur ne s'est pas vu accorder ou refuser des autorisations spécifiques sur un objet sécurisable, l'utilisateur hérite des autorisations accordées à public sur cet objet. Vous ne devez affecter des autorisations publiques à un objet que lorsque vous souhaitez que ce dernier soit disponible pour tous les utilisateurs. Vous ne pouvez pas modifier l'appartenance au rôle public.

Ils sont gérés indépendamment des bases de données et sont stockés dans la table système **sys.server_principals**.

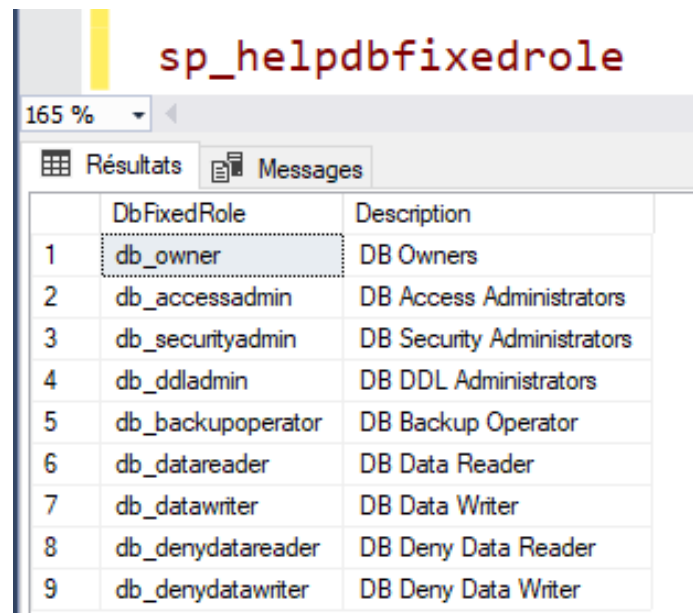
	name	principal_id	sid	type	type_desc	is_disabled	create_date
1	sa	1	0x01	S	SQL_LOGIN	0	2003-04-08
2	public	2	0x02	R	SERVER_ROLE	0	2009-04-13
3	sysadmin	3	0x03	R	SERVER_ROLE	0	2009-04-13
4	securityadmin	4	0x04	R	SERVER_ROLE	0	2009-04-13
5	serveradmin	5	0x05	R	SERVER_ROLE	0	2009-04-13
6	setupadmin	6	0x06	R	SERVER_ROLE	0	2009-04-13
7	processadmin	7	0x07	R	SERVER_ROLE	0	2009-04-13
8	diskadmin	8	0x08	R	SERVER_ROLE	0	2009-04-13
9	dbcreator	9	0x09	R	SERVER_ROLE	0	2009-04-13
10	bulkadmin	10	0x0A	R	SERVER_ROLE	0	2009-04-13

Rôles prédéfinis base de données

À l'exception du rôle de base de données **public**, les autorisations affectées aux rôles de base de données fixes (prédéfinis) ne peuvent pas être changées.

Rôle BDD	Description
db_owner	Les membres du rôle de base de données fixe db_owner peuvent effectuer toutes les activités de configuration et de maintenance sur la base de données et peuvent également supprimer la base de données dans SQL Server.
db_securityadmin	Les membres du rôle de base de données fixe db_securityadmin peuvent modifier l'appartenance au rôle, pour les rôles personnalisés uniquement, et gérer les autorisations. Les membres de ce rôle peuvent potentiellement élever leurs privilèges et leurs actions doivent être supervisées.
db_accessadmin	Les membres du rôle de base de données fixe db_accessadmin peuvent ajouter ou supprimer l'accès à la base de données des connexions Windows, des groupes Windows et des connexions SQL Server .
db_backupoperator	Les membres du rôle de base de données fixe db_backupoperator peuvent sauvegarder la base de données.
db_ddladmin	Les membres du rôle de base de données fixe db_ddladmin peuvent exécuter n'importe quelle commande DDL (Data Definition Language) dans une base de données.
db_datawriter	Les membres du rôle de base de données fixe db_datawriter peuvent ajouter, supprimer et modifier des données dans toutes les tables utilisateur.
db_datareader	Les membres du rôle de base de données fixe db_datareader peuvent lire toutes les données de toutes les tables utilisateur.
db_denydatawriter	Les membres du rôle de base de données fixe db_denydatawriter ne peuvent ajouter, modifier ou supprimer aucune donnée des tables utilisateur d'une base de données.
db_denydatareader	Les membres du rôle de base de données fixe db_denydatareader ne peuvent lire aucune donnée des tables utilisateur d'une base de données.

La procédure stockée **sp_helpdbfixedrole**, permet d'en obtenir la liste.

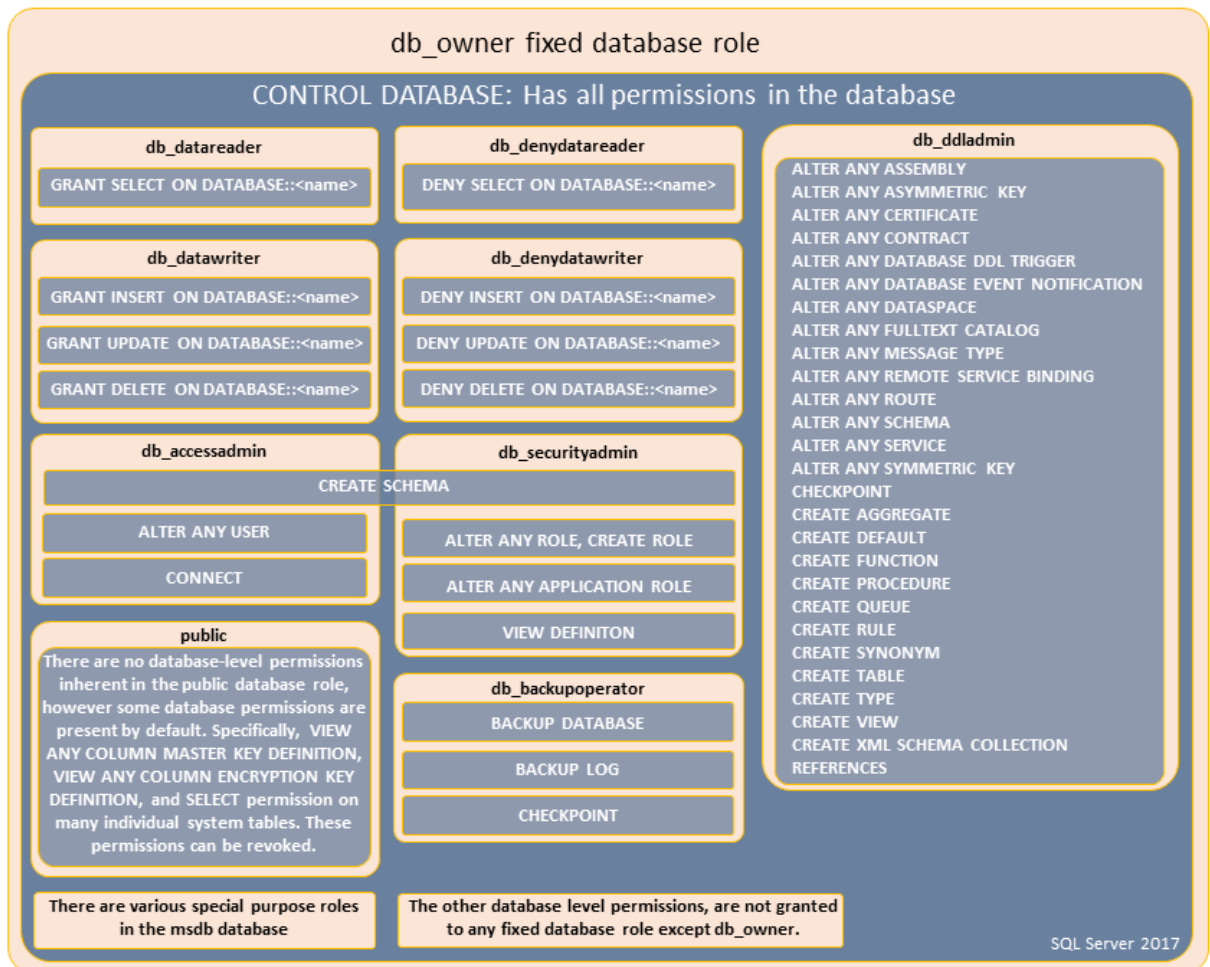


	DbFixedRole	Description
1	db_owner	DB Owners
2	db_accessadmin	DB Access Administrators
3	db_securityadmin	DB Security Administrators
4	db_ddladmin	DB DDL Administrators
5	db_backupoperator	DB Backup Operator
6	db_datareader	DB Data Reader
7	db_datawriter	DB Data Writer
8	db_denydatareader	DB Deny Data Reader
9	db_denydatawriter	DB Deny Data Writer



Synthèse des rôle BDD Fixes

DATABASE LEVEL ROLES AND PERMISSIONS: 11 fixed database roles, 77 database permissions



Rôles définis par l'utilisateur

Il est possible de définir ses propres rôles afin de faciliter l'administration des droits dans SQL Server.

Logiquement, on créera un rôle dit personnalisé lorsque plusieurs utilisateurs doivent avoir les mêmes droits et que ces droits n'existent pas dans les rôles prédéfinis.

Les rôles peuvent être accordés soit directement à un utilisateur, soit à un autre rôle.



Exemple

Créer un rôle nommé **Lecture** qui n'autorise que le **Select** sur la table **ARTICLE**, et ajouter l'utilisateur **camille** à ce rôle :

```
CREATE ROLE Lecture
GO
GRANT SELECT ON ARTICLE TO Lecture
GO
ALTER ROLE Lecture ADD MEMBER Camille
GO
```

Crédits

OEUVRE COLLECTIVE DE L'AFPA

Sous le pilotage de la Direction de l'ingénierie

DATE DE MISE À JOUR

15/10/2020

© AFPA

Reproduction interdite

Article L 122-4 du code de la propriété intellectuelle.

« Toute représentation ou reproduction intégrale ou partielle faite sans le consentement de l'auteur ou de ses ayants droit ou ayants cause est illicite. Il en est de même pour la traduction, l'adaptation ou la reproduction par un art ou un procédé quelconques. »