

ADMINISTRATION ET SÉCURITÉ DES BASES DE DONNÉES

Support de Cours

LICENSE 3 - ASSRI

SÉCURITÉ

Des Bases de Données SQL Server

Année Scolaire 2022-2023

Par

Professeur: **Robert Yavo**

Email: ryavo@hotmail.com WhatsApp: +225-07-88-63-26-58

Table des matières

1. Introduction	3
1.1 Définition de la sécurité	3
1.2 Sécurisation de SQL Server	3
2. Meilleures pratiques de sécurité pour SQL Serveur.....	3
3. Mise en place de la sécurité d'un Serveur SQL Server : Exemple	4
3.1 Sécuriser le Serveur SQL.....	4
3.1.1 Configuration des protocoles et options de démarrage et connexion	4
3.1.2 Évaluation des vulnérabilités	5
3.1.3 Découvrir et classer les données SQL	6
3.2 Gérer les accès et les authentifications.....	7
3.2.1 Déterminer d'abord les Autorisations Effectives	7
3.2.2 Créer un compte de connexion	8
3.2.3 Créer un utilisateur de base de données.....	11
3.2.4 Créer un schéma de base de données.....	12
3.2.5 Joindre un rôle	12
3.2.6 Accorder une autorisation à une Procédure Stockées (principal)	14
3.2.7 Créer un rôle serveur	14
3.2.8 Créer un rôle d'application	15
3.2.9 Créer des informations d'identification	16
3.3 Chiffrement (Cryptage)	17
3.3.1 Configurer l'outil Always Encrypted à l'aide de PowerShell	17
3.3.2 Configurer le chiffrement de colonne à l'aide de l'Assistant Always Encrypted ..	17
3.3.3 Chiffrer une colonne de données sensible (Ex : Carte de Crédit)	18
3.4 Création des Audits.....	20
3.4.1 Créer un Audit du Serveur et une spécification	20
3.4.2 Créer une spécification de l'audit de la Base de données	21
3.4.3 Affichez les journaux d'audit	22
3.4.4 Écrire les événements d'audit dans le journal Windows (Events Viewer)	23
4. Références.....	25

1. Introduction

1.1 Définition de la sécurité

La sécurité de l'information (des données) de façon générale repose sur l'assurance des 3 principes fondamentaux qui sont la Disponibilité, l'Intégrité et la Confidentialité de l'information (des données).

Tout système d'information ou de gestion de base de données sécuritaire doit tenir compte de ces trois principes.

C'est le cas de Microsoft SQL Server que nous allons étudier.

1.2 Sécurisation de SQL Server

La sécurisation de SQL Server peut être vue comme une série d'étapes impliquant quatre domaines :

- Sécurité de la plateforme et du réseau (Physique, OS, Système de fichiers SQL)
- Sécurité des individus et des objets de SQL Serveur (Rôles, accès, permissions, authentification, encryptions et certificats)
- Sécurité des applications qui accèdent au système SQL Server.
- Sécurité des Outils, des utilitaires et des fonctions du moteur de la base de données

2. Meilleures pratiques de sécurité pour SQL Serveur

Une organisation doit avoir un plan ou une méthode de gestion de la sécurité de ses serveurs de données SQL Serveur qui respecte les règles suivantes :

- Protection au niveau des colonnes (Chiffrage, Masque et Autorisations)
- Protection au niveau des lignes d'enregistrements (limiter les enregistrements renvoyés aux usagers, les filtrer et limiter les modifications)
- Chiffrer les fichiers de données (Transparent Data Encryptions : TDE)
- Audit et Rapports sur les tables et colonnes dans les interactions avec SQL Serveur
- Établir les rôles serveurs et l'identification et l'authentification des usagers et des applications (mode Windows et/ou SQL Serveur)
- Établir la traçabilité et l'intégrité des données
- Avoir des outils et méthodes d'évaluation de la sécurité
- Connaître les menaces courantes liées à SQL Serveur
- Minimiser les Risques (menaces) de SQL Injection en revoyant les vulnérabilités.
- Gérer les Risques d'attaque par canal (Side-Channel risks)
- Gérer les menaces d'infrastructures (Brute force access, Password cracking/spray, Ransomware attacks, etc.)

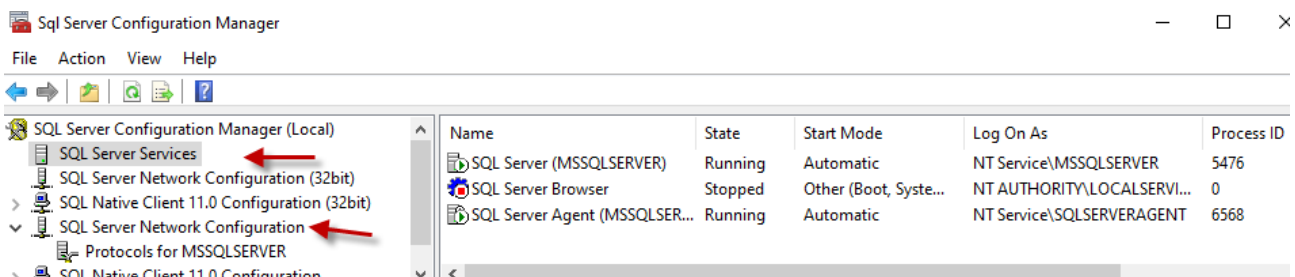
3. Mise en place de la sécurité d'un Serveur SQL Server : Exemple

3.1 Sécuriser le Serveur SQL

3.1.1 Configuration des protocoles et options de démarrage et connexion

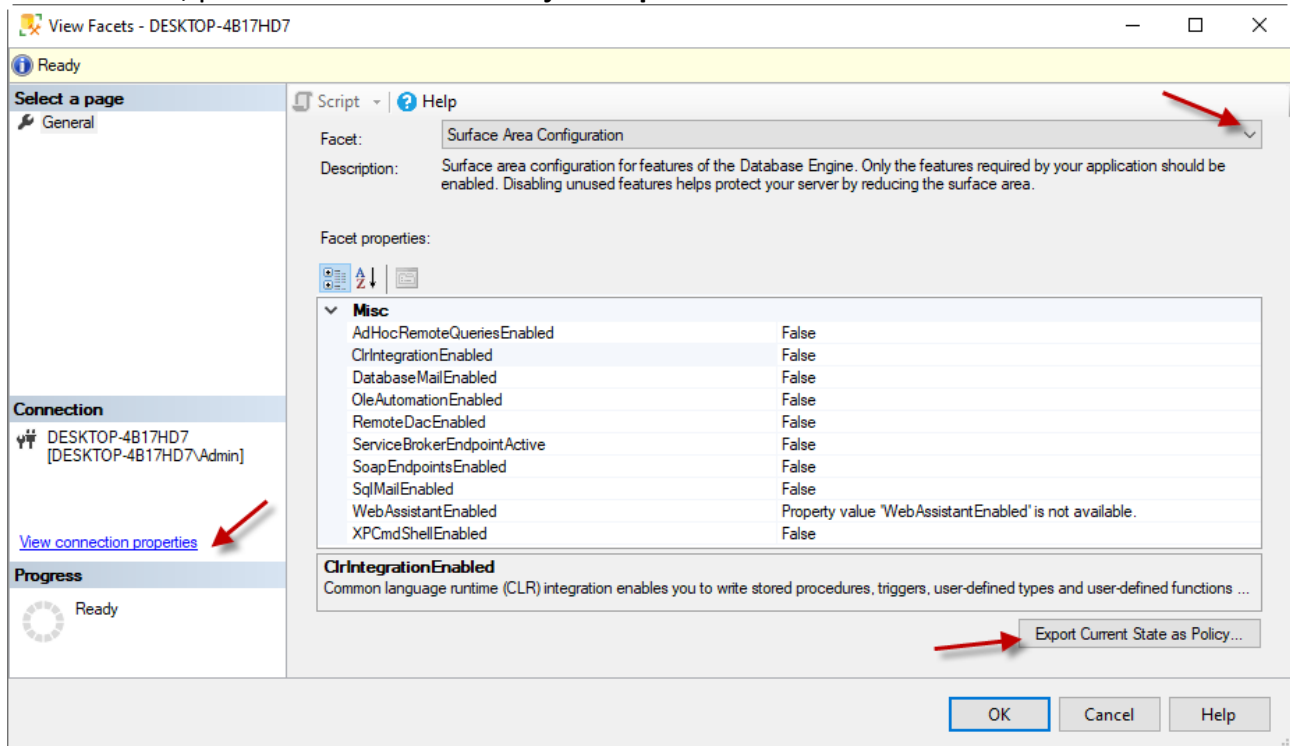
Les administrateurs (DBA) doivent :

- Utiliser **SQL Server Configuration Manager** pour :
 - Arrêter et redémarrer les services
 - Configurer les options de démarrage
 - Activer les protocoles et les autres options de connexion



- Utiliser **Facets (SSMS)** pour Activer et désactiver les fonctionnalités de SQL Server

Ouvrir SSMS, puis Bouton droit sur **Object Explorer=>Facets**



Utiliser la commande **Invoke-PolicyEvaluation** de PowerShell pour invoquer les politiques (Policy) de Surface Area Configuration Policies par exemple.

Allez dans Facets comme le montre l'image précédent et cliquez sur **Export Current State as Policy** après avoir sélectionné **Surface Area Configuration** Policies dans la liste Facet. Sauvegardez dans le dossier : **C:\Users\Admin\Documents\SQL Server Management Studio\Policies\ Surface Area Configuration_20221120.xml**

Démarrer PowerShell puis lancer les commandes suivantes :

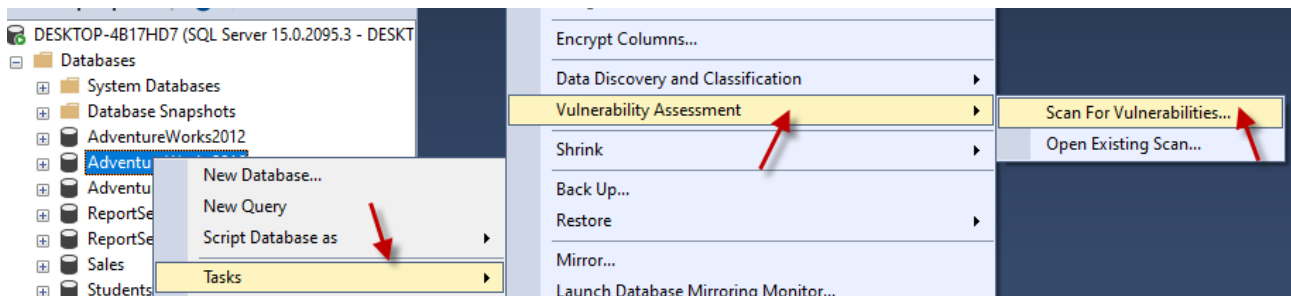
```
PS C:\Users\Admin> Invoke-PolicyEvaluation - "C:\Users\Admin\Documents\SQL Server Management Studio\Policies\Surface Area Configuration_20221120.xml" -TargetServerName 'DESKTOP-4B17HD7'
```

```
PS C:\Users\Admin> Get-ChildItem "C:\Users\Admin\Documents\SQL Server Management Studio\Policies" | Where-Object { $_.PolicyCategory -eq "Surface Area Configuration_20221120.xml" } | Invoke-PolicyEvaluation -TargetServer 'DESKTOP-4B17HD7'
```

3.1.2 Évaluation des vulnérabilités

Utilisez SSMS pour évaluer les vulnérabilités :

Bouton-Droit sur la base de données dans Object Explorer et choisir **Tasks>Vulnerability Assessment>Scan for Vulnerabilities**



Exemple de Résultat d'évaluation de vulnérabilités :

Vulnerability Assessment Results

Server: DESKTOP-4B17HD7 Database: AdventureWorks2016 Scan time: 2022-11-20T02:41:54.2845889+00:00 Export to Excel

The Vulnerability Assessment scans in SSMS and in Azure Defender for SQL both rely on independent baselines. You can set and configure the baselines in each tool. For an optimal experience and advanced capabilities, we recommend running your VA scans with Azure Defender. Learn more: <https://go.microsoft.com/fwlink/?linkid=2152847>

Total failing checks

3 ❌

Total passing checks

32 ✅

High Risk 1

Medium Risk 2

Low Risk 0

Learn more

[SQL Security Center](#)

[Best Practices for SQL Security](#)

❌ Failed (3) ✅ Passed (32)

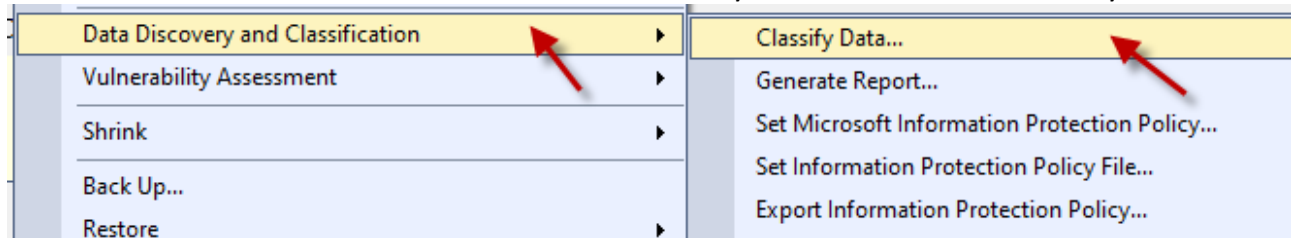
ID	Security Check	Category	Risk	Additional Information
VA1245	The database owner information in the database should match the respective dat	Surface Area Reduction	High	
VA1143	'dbo' user should not be used for normal service operation	Surface Area Reduction	Medium	
VA1219	Transparent data encryption should be enabled	Data Protection	Medium	

Remarques Importantes :

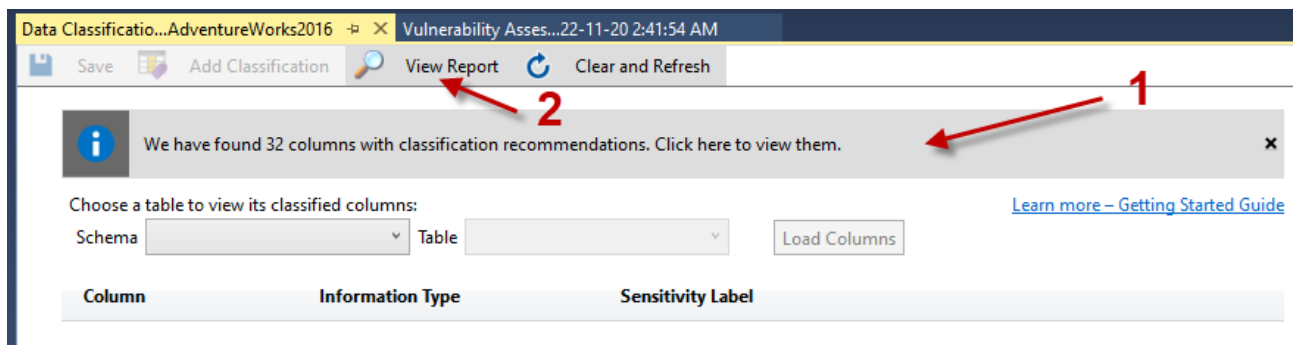
1. Une solution est proposée pour remédier aux problèmes de vulnérabilités trouvées. Il suffit de défiler jusqu'à la section **Remediation** et lire et exécuter la proposition donnée.
2. Vous pouvez aussi définir une ligne de base acceptable en cliquant sur **Approve As Baseline** dans le rapport de Résultat de Vulnérabilités

3.1.3 Découvrir et classer les données SQL

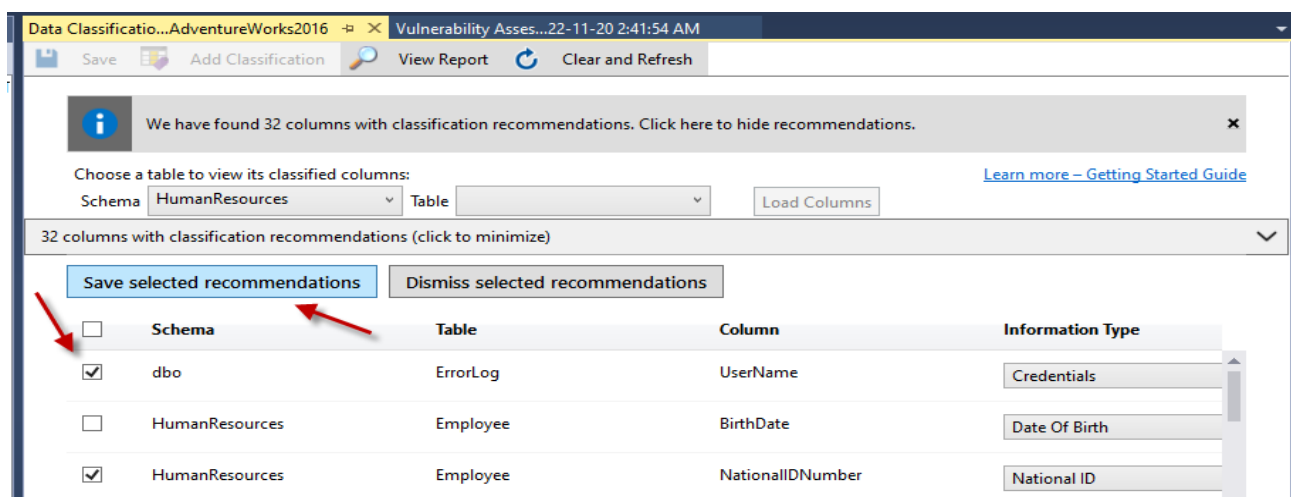
Dans SSMS, AdventureWorks >Tasks>Data Discovery and Classification>Classify Data..



L'outil propose des recommandations de classification des colonnes et la possibilité de voir le Rapport comme le montre l'écran suivant.



Et vous pouvez sauver les recommandations choisies en les cochant et en cliquant sur **Save Selected Recommendations** ou même ajouter des classifications manuellement en cliquant sur **Add Classification** dans le menu.



3.2 Gérer les accès et les authentifications

3.2.1 Déterminer d'abord les Autorisations Effectives

L'administrateur de BD devrait exécuter dans T-SQL de SSMS, les requêtes SQL suivantes:

- a) Pour déterminer qui est membre d'un rôle serveur fixe, exécutez la requête suivante :

```
SELECT SP1.name AS ServerRoleName,  
       ISNULL(SP2.name, 'No members') AS LoginName  
FROM sys.server_role_members AS SRM  
RIGHT JOIN sys.server_principals AS SP1  
    ON SRM.role_principal_id = SP1.principal_id  
LEFT JOIN sys.server_principals AS SP2  
    ON SRM.member_principal_id = SP2.principal_id  
WHERE SP1.is_fixed_role = 1 -- Remove for SQL Server 2008  
ORDER BY SP1.name;
```

- b) Pour déterminer qui est membre d'un rôle base de données fixe, exécutez la requête suivante dans chaque base de données.

```
SELECT DP1.name AS DatabaseRoleName,  
       ISNULL(DP2.name, 'No members') AS DatabaseUserName  
FROM sys.database_role_members AS DRM  
RIGHT JOIN sys.database_principals AS DP1  
    ON DRM.role_principal_id = DP1.principal_id  
LEFT JOIN sys.database_principals AS DP2  
    ON DRM.member_principal_id = DP2.principal_id  
WHERE DP1.is_fixed_role = 1  
ORDER BY DP1.name;
```

- c) Pour obtenir la liste des autorisations qui ont été accordées ou refusées au niveau du serveur, exécuter cette requête dans le Master.

```
SELECT pr.type_desc,  
       pr.name,  
       ISNULL(pe.state_desc, 'No permission statements') AS state_desc,  
       ISNULL(pe.permission_name, 'No permission statements') AS permission_name  
FROM sys.server_principals AS pr  
LEFT JOIN sys.server_permissions AS pe  
    ON pr.principal_id = pe.grantee_principal_id  
WHERE is_fixed_role = 0 -- Remove for SQL Server 2008  
ORDER BY pr.name,  
       type_desc;
```

- d) Pour obtenir la liste des autorisations qui ont été accordées ou refusées au niveau de chaque base de données, exécuter la requête qui suit:

```
SELECT pr.type_desc,  
       pr.name,  
       ISNULL(pe.state_desc, 'No permission statements') AS state_desc,  
       ISNULL(pe.permission_name, 'No permission statements') AS permission_name  
FROM sys.database_principals AS pr  
LEFT JOIN sys.database_permissions AS pe  
  ON pr.principal_id = pe.grantee_principal_id  
WHERE pr.is_fixed_role = 0  
ORDER BY pr.name,  
         type_desc;
```

- e) La requête suivante fournit le nom de l'objet de base de données qui est concerné par l'autorisation.

```
SELECT pr.type_desc,  
       pr.name,  
       pe.state_desc,  
       pe.permission_name,  
       s.name + '.' + oj.name AS OBJECT,  
       major_id  
FROM sys.database_principals AS pr  
INNER JOIN sys.database_permissions AS pe  
  ON pr.principal_id = pe.grantee_principal_id  
INNER JOIN sys.objects AS oj  
  ON oj.object_id = pe.major_id  
INNER JOIN sys.schemas AS s  
  ON oj.schema_id = s.schema_id  
WHERE class_desc = 'OBJECT_OR_COLUMN';
```

- f) Utilisez la fonction HAS_PERMS_BY_NAME pour déterminer si un utilisateur particulier (dans ce cas TestUser) dispose d'une autorisation.
Par exemple :

```
EXECUTE AS USER = 'TestUser';  
SELECT HAS_PERMS_BY_NAME ('dbo.T1', 'OBJECT', 'SELECT');  
REVERT;
```

3.2.2 Créer un compte de connexion

3.2.2.1 Créer une connexion d'authentification Windows ou SQL Server avec T-SQL

-- Authentication Windows.

```
CREATE LOGIN [<domainName>\<loginName>] FROM WINDOWS;  
GO
```

-- Exemple:

```
CREATE LOGIN [DESKTOP-4B17HD7\robert] FROM WINDOWS;  
GO
```

L'utilisateur Windows du nom de robert doit exister auparavant bien sûr.

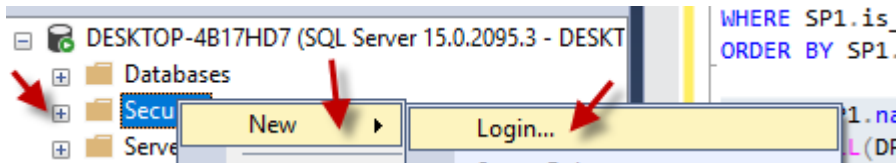
-- Authentication SQL Server.

```
CREATE LOGIN robert2 WITH PASSWORD = 'Robert123'
    MUST_CHANGE, CHECK_EXPIRATION = ON;
```

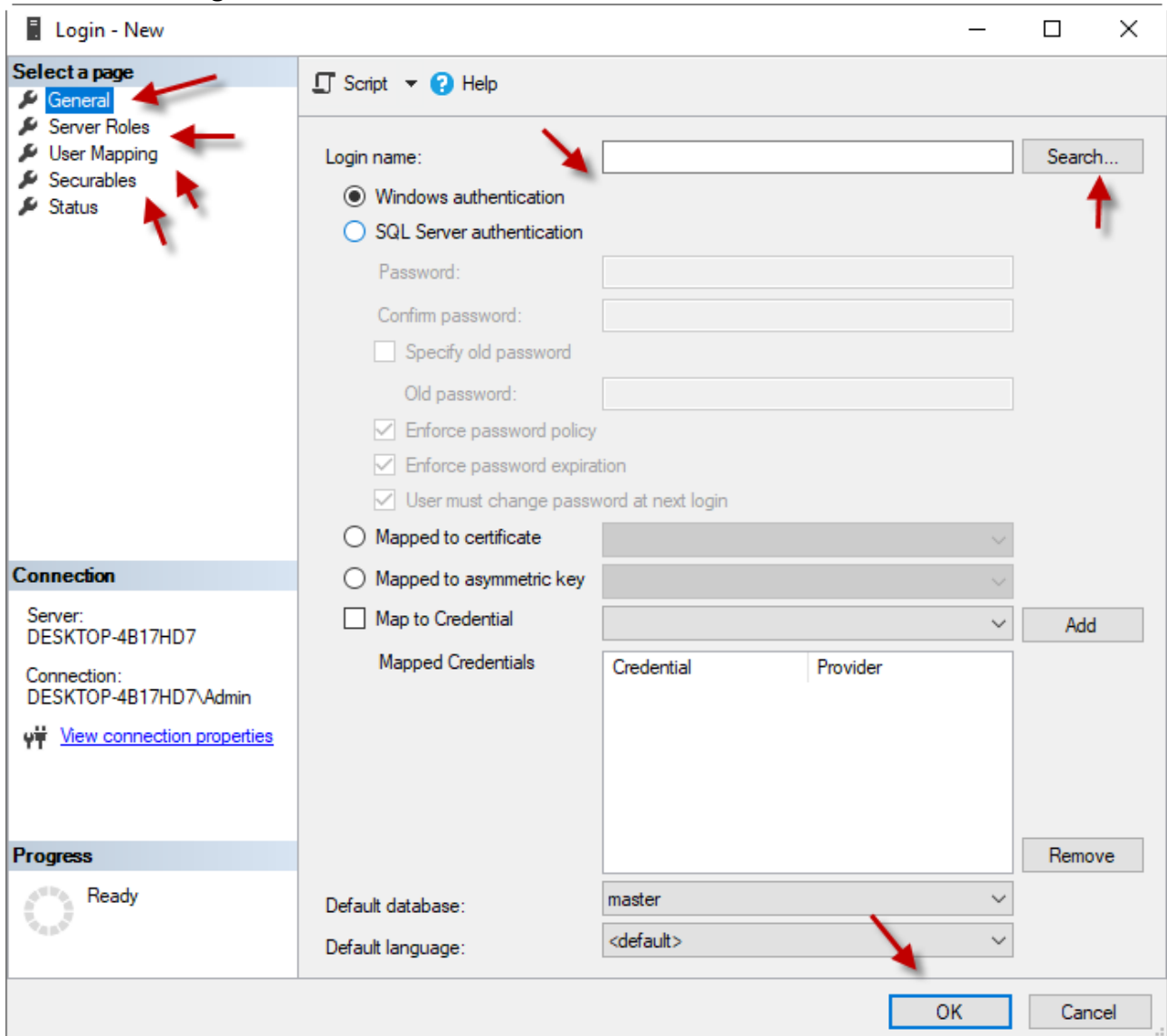
GO

Plus de détails : <https://learn.microsoft.com/en-us/sql/t-sql/statements/create-login-transact-sql?view=sql-server-ver16>

3.2.2.2 Pour créer une connexion à l'aide de SSMS pour SQL Server :



Puis remplissez le tableau suivant en suivant les instructions tirées du site Web de Microsoft en Anglais :



Login - New

Select a page

- General
- Server Roles
- User Mapping
- Securables
- Status

Connection

Server: DESKTOP-4B17HD7

Connection: DESKTOP-4B17HD7\Admin

[View connection properties](#)

Progress

Ready

Script ? Help

Login name: Search...

☒ Windows authentication

☐ SQL Server authentication

Password:

Confirm password:

☐ Specify old password

Old password:

☒ Enforce password policy

☒ Enforce password expiration

☒ User must change password at next login

☐ Mapped to certificate

☐ Mapped to asymmetric key

☐ Map to Credential Add

Credential	Provider

Remove

Default database: master

Default language: <default>

OK Cancel

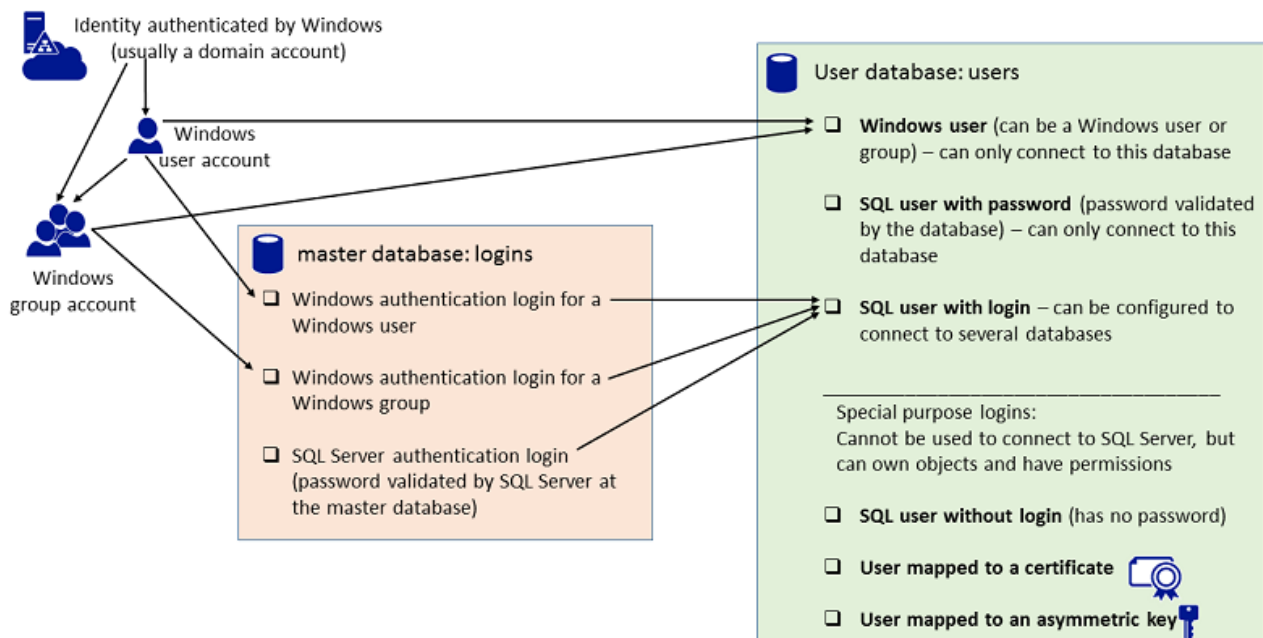
Les étapes à faire.

1. In Object Explorer, expand the folder of the server instance in which you want to create the new login.
2. Right-click the **Security** folder, point to **New**, and select **Login...**.
3. In the **Login - New** dialog box, on the **General** page, enter the name of a user in the **Login name** box. Alternately, select **Search...** to open the **Select User or Group** dialog box.
If you select **Search...**:
 - a. Under **Select this object type**, select **Object Types...** to open the **Object Types** dialog box and select any or all of the following: **Built-in security principals**, **Groups**, and **Users**. **Built-in security principals** and **Users** are selected by default. When finished, select **OK**.
 - b. Under **From this location**, select **Locations...** to open the **Locations** dialog box and select one of the available server locations. When finished, select **OK**.
 - c. Under **Enter the object name to select (examples)**, enter the user or group name that you want to find. For more information, see [Select Users, Computers, or Groups Dialog Box](#).
 - d. Select **Advanced...** for more advanced search options. For more information, see [Select Users, Computers, or Groups Dialog Box - Advanced Page](#).
 - e. Select **OK**.
4. To create a login based on a Windows principal, select **Windows authentication**. This is the default selection.
5. To create a login that is saved on a SQL Server database, select **SQL Server authentication**.
 - a. In the **Password** box, enter a password for the new user. Enter that password again into the **Confirm Password** box.
 - b. When changing an existing password, select **Specify old password**, and then type the old password in the **Old password** box.
 - c. To enforce password policy options for complexity and enforcement, select **Enforce password policy**. For more information, see [Password Policy](#). This is a default option when **SQL Server authentication** is selected.
- d. To enforce password policy options for expiration, select **Enforce password expiration**. **Enforce password policy** must be selected to enable this checkbox. This is a default option when **SQL Server authentication** is selected.
- e. To force the user to create a new password after the first time the login is used, select **User must change password at next login**. **Enforce password expiration** must be selected to enable this checkbox. This is a default option when **SQL Server authentication** is selected.
6. To associate the login with a stand-alone security certificate, select **Mapped to certificate** and then select the name of an existing certificate from the list.
7. To associate the login with a stand-alone asymmetric key, select **Mapped to asymmetric key** to, and then select the name of an existing key from the list.

8. To associate the login with a security credential, select the **Mapped to Credential** check box, and then either select an existing credential from the list or select **Add** to create a new credential. To remove a mapping to a security credential from the login, select the credential from **Mapped Credentials** and select **Remove**. For more information about credentials in general, see [Credentials \(Database Engine\)](#).
9. From the **Default database** list, select a default database for the login. **Master** is the default for this option.
10. From the **Default language** list, select a default language for the login.
11. Select **OK**.

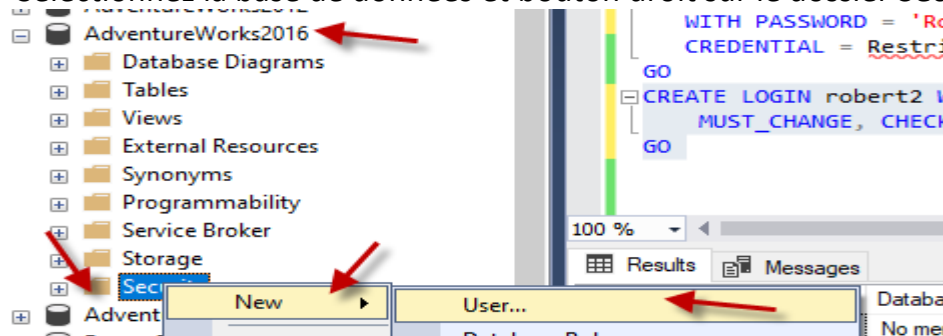
3.2.3 Créer un utilisateur de base de données

SSMS présente 6 options de création d'utilisateur (User database: users)



Pour créer donc un utilisateur avec SSMS :

- Sélectionnez la base de données et bouton droit sur le dossier **Security**



Et remplissez les informations requises à la prochaine fenêtre.

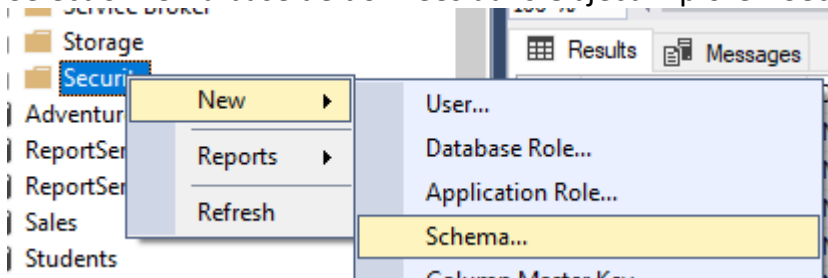
Pour Créer un utilisateur avec T-SQL

```
-- Creates the login AbolrousHazem with password '340$Uuxwp7Mcxo7Khy'.
CREATE LOGIN AbolrousHazem
    WITH PASSWORD = '340$Uuxwp7Mcxo7Khy';
GO
-- Creates a database user for the login created above.
CREATE USER AbolrousHazem FOR LOGIN AbolrousHazem;
GO
```

3.2.4 Créer un schéma de base de données

Avec SSMS :

Sélectionnez la base de données dans Object Explorer>Security>New>Schema



Avec T-SQL

L'exemple suivant crée le schéma Sprockets détenu par Annick qui contient la table NineProngs. L'instruction accorde SELECT à Mandar et refuse SELECT à Prasanna.

```
CREATE SCHEMA Sprockets AUTHORIZATION Annik
    CREATE TABLE NineProngs (source int, cost int, partnumber int)
    GRANT SELECT ON SCHEMA::Sprockets TO Mandar
    DENY SELECT ON SCHEMA::Sprockets TO Prasanna;
GO
```

```
--Pour verifier: SELECT * FROM sys.schemas;
```

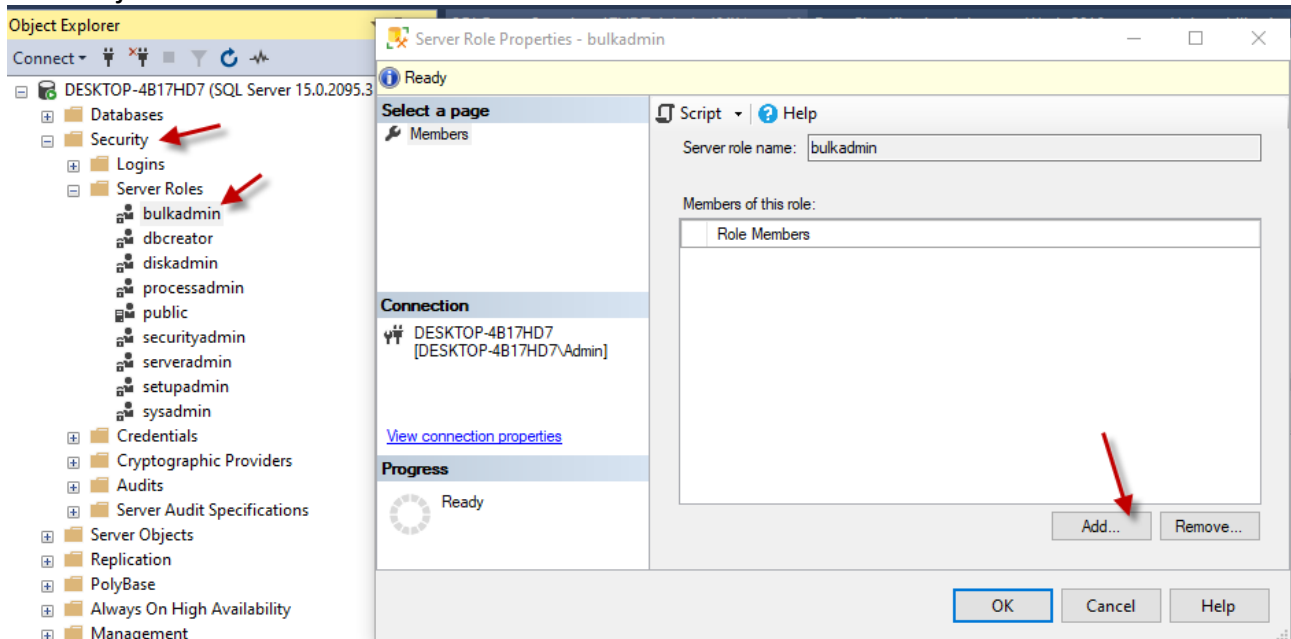
3.2.5 Joindre un rôle

Avec T-SQL

```
-- Syntaxe pour Ajouter un membre à un rôle serveur fixe
ALTER SERVER ROLE diskadmin ADD MEMBER [Domain\Juan] ;
GO
-- Exemple
ALTER SERVER ROLE diskadmin ADD MEMBER [DESKTOP-4B17HD7\robert];
GO
-- Syntaxe pour Ajouter un membre à un rôle de base de données défini par l'utilisateur
ALTER ROLE Marketing ADD MEMBER [Domain\Juan] ;
GO
-- Exemple
ALTER ROLE Marketing ADD MEMBER [DESKTOP-4B17HD7\robert];
GO
```

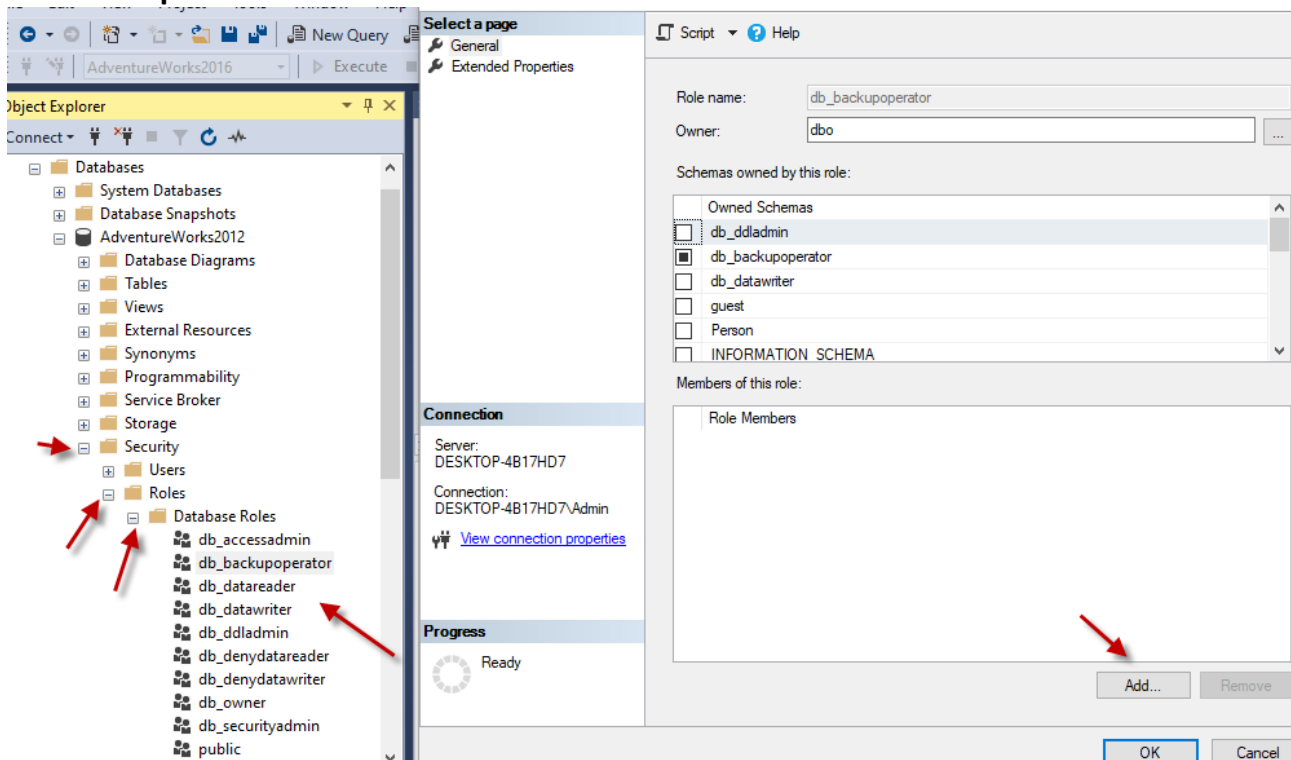
Avec SSMS

- Pour ajouter un membre à un rôle serveur fixe



Pour ajouter un membre à un rôle de base de données défini par l'utilisateur

AdventureWorks2016>Security>Roles>Database Roles>Bouton droit sur un usager puis choisir Properties.



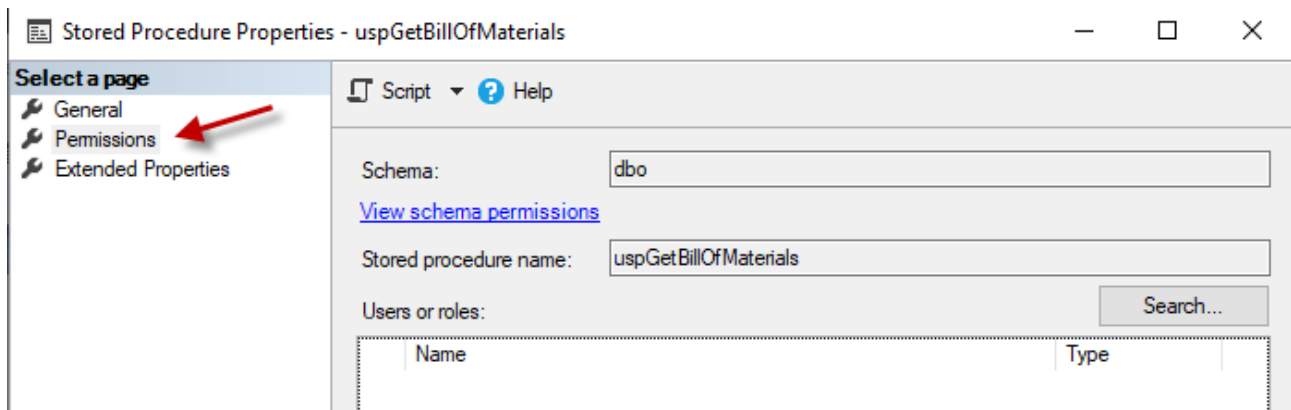
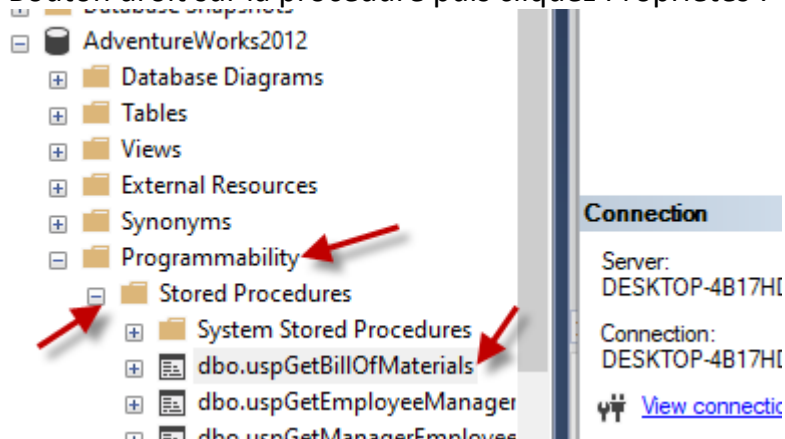
3.2.6 Accorder une autorisation à une Procédure Stockées (principal)

Avec T-SQL

```
-- Grants EXECUTE permission on stored procedure
HumanResources.uspUpdateEmployeeHireInfo to an application role called
Recruiting11.
USE AdventureWorks2012;
GO
GRANT EXECUTE ON OBJECT::HumanResources.uspUpdateEmployeeHireInfo
    TO Recruiting11;
GO
```

Avec SSMS

Bouton droit sur la procédure puis cliquez Propriétés :

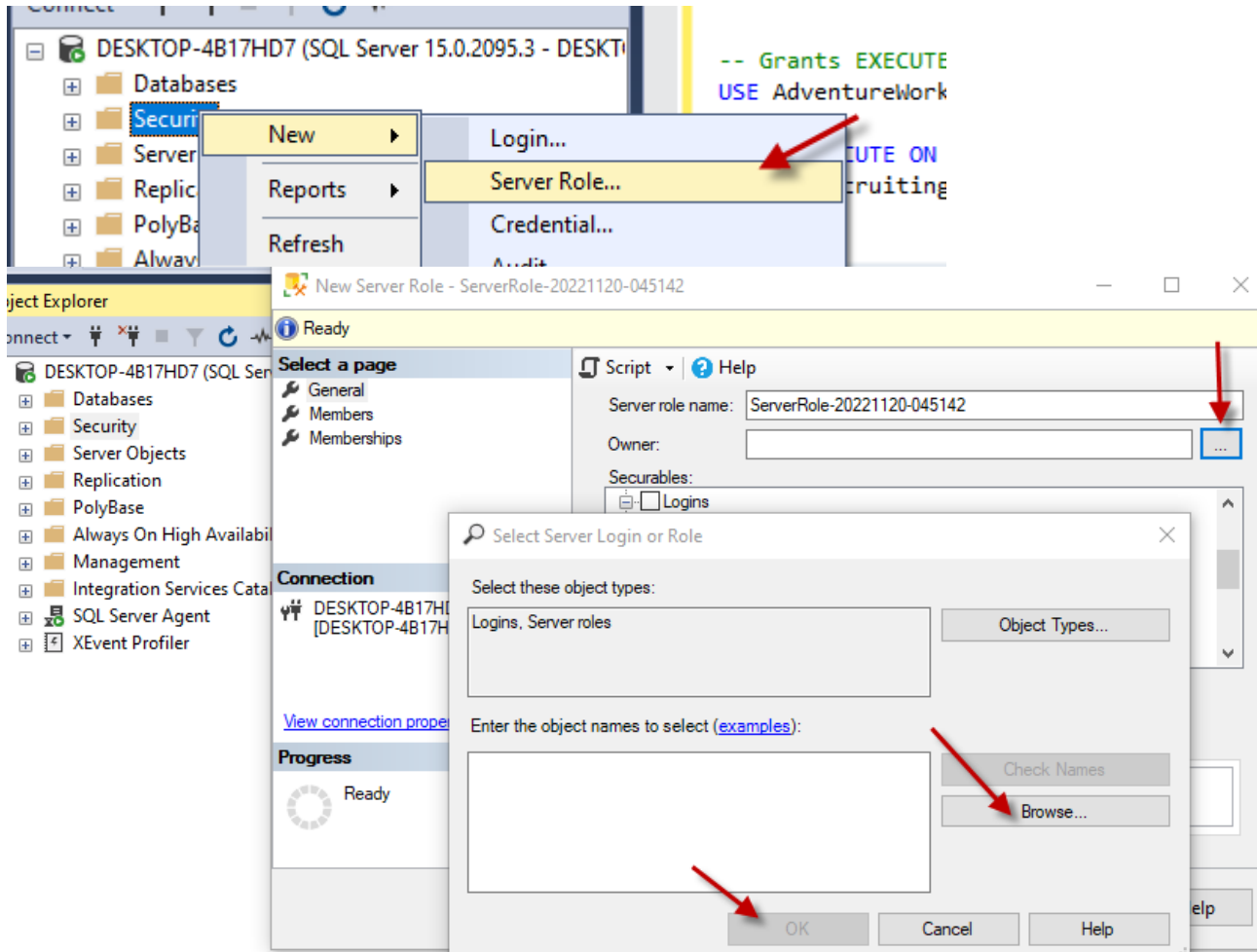


3.2.7 Créer un rôle serveur

Avec T-SQL

```
--Creates the server role auditors that is owned the securityadmin
fixed server role.
USE master;
CREATE SERVER ROLE auditors AUTHORIZATION securityadmin;
GO
```

Avec SSMS



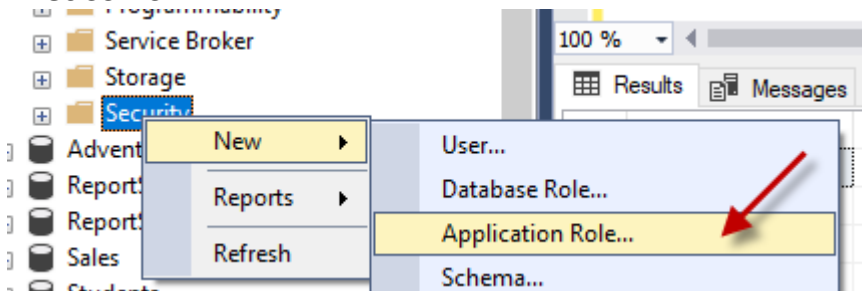
3.2.8 Créer un rôle d'application

Avec T-SQL

-- Creates an application role called "weekly_receipts" that has the password "987Gbv876sPYY5m23" and "Sales" as its default schema.

```
CREATE APPLICATION ROLE weekly_receipts
    WITH PASSWORD = '987G^bv876sPY)Y5m23'
    , DEFAULT_SCHEMA = Sales;
GO
```

Avec SSMS

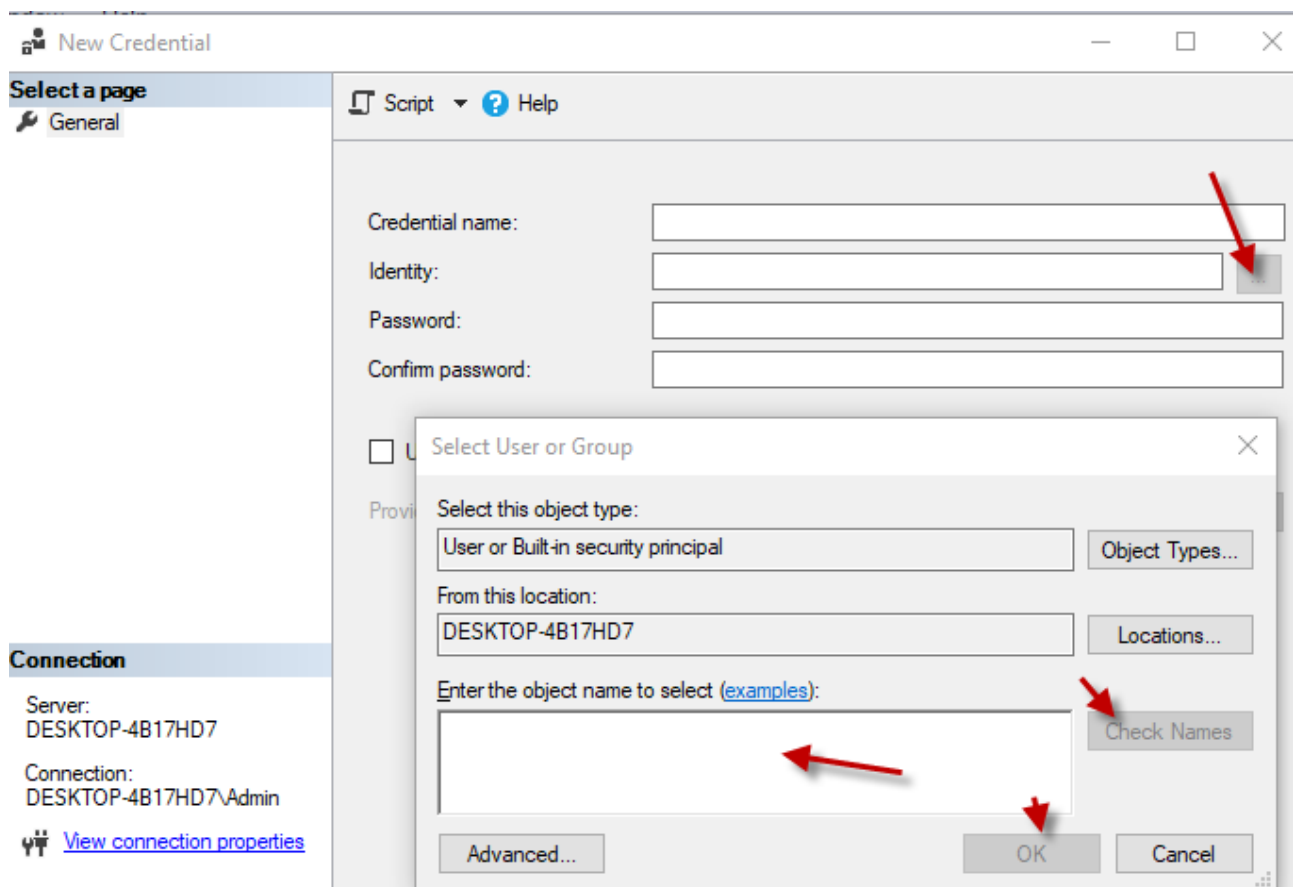
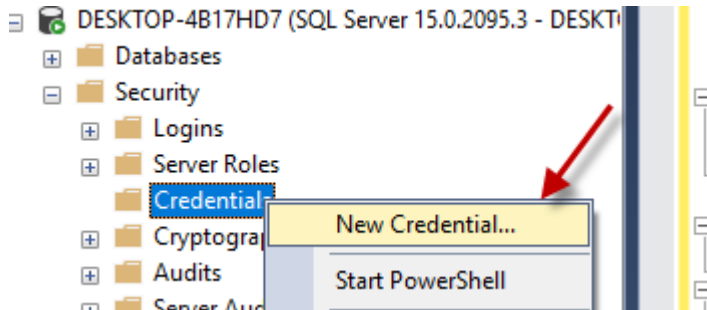


3.2.9 Créer des informations d'identification

Avec T-SQL

```
-- Creates the credential called "AlterEgo.".
-- The credential contains the Windows user "Mary5" and a password.
CREATE CREDENTIAL AlterEgo WITH IDENTITY = 'Mary5',
    SECRET = '<EnterStrongPasswordHere>';
GO
```

Avec SSMS



3.3 Chiffrement (Cryptage)

3.3.1 Configurer l'outil Always Encrypted à l'aide de PowerShell

Importez le module SQLServer dans PowerShell si ce n'est déjà fait.

```
# Import the SqlServer module.
Import-Module "SqlServer"
# Navigate to the database in the remote instance.
cd SQLSERVER:\SQL\servercomputer\DEFAULT\Databases\yourdatabase
# List column master keys in the above database.
Get-SqlColumnMasterKey
```

On peut directement indiquer le chemin d'accès :

```
# Import the SqlServer module.
Import-Module "SqlServer"
# List column master keys for the specified database.
Get-SqlColumnMasterKey -Path
SQLSERVER:\SQL\servercomputer\DEFAULT\Databases\yourdatabase
```

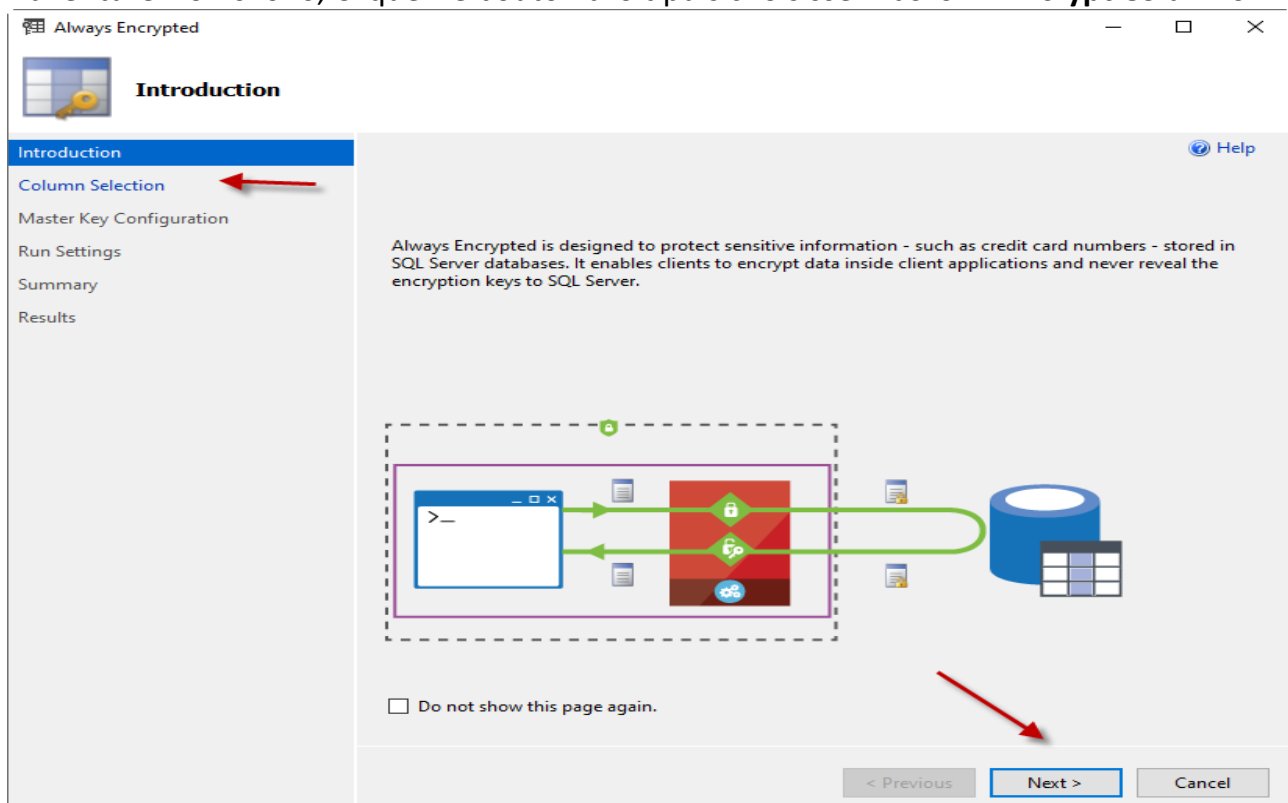
3.3.2 Configurer le chiffrement de colonne à l'aide de l'Assistant Always Encrypted

Le chiffrage peut se faire dans les 3 niveaux suivants :

- Au niveau de la base de données, si vous voulez chiffrer plusieurs colonnes de tables différentes.
- Au niveau de la table, si vous voulez chiffrer plusieurs colonnes de la même table.
- Au niveau de la colonne, si vous voulez chiffrer une colonne spécifique.

Ouvrez SSMS et selon le Niveau choisi, Par exemple la base de données

AdventureWorks2016, Cliquez le bouton droit puis choisissez **Tasks => Encrypt Columns**



3.3.3 Chiffrer une colonne de données sensible (Ex : Carte de Crédit)

a) Créer un Master Key de la base de données

Par exemple se connecter à la base de données AdventureWorks2016 et taper le code T-SQL suivant :

```
CREATE MASTER KEY ENCRYPTION BY
PASSWORD = '<complex password>';
```

b) Puis chiffrer avec le Chiffrement Symétrique et l'authentificateur.

```
CREATE CERTIFICATE Sales09
    WITH SUBJECT = 'Customer Credit Card Numbers';
GO
CREATE SYMMETRIC KEY CreditCards_Key11
    WITH ALGORITHM = AES_256
    ENCRYPTION BY CERTIFICATE Sales09;
GO
-- Create a column in which to store the encrypted data.
ALTER TABLE Sales.CreditCard
    ADD CardNumber_Encrypted varbinary(160);
GO
-- Open the symmetric key with which to encrypt the data.
OPEN SYMMETRIC KEY CreditCards_Key11
    DECRYPTION BY CERTIFICATE Sales09;
-- Encrypt the value in column CardNumber using the
-- symmetric key CreditCards_Key11.
-- Save the result in column CardNumber_Encrypted.
UPDATE Sales.CreditCard
SET CardNumber_Encrypted = EncryptByKey(Key_GUID('CreditCards_Key11')
    , CardNumber, 1, HASHBYTES('SHA2_256', CONVERT( varbinary
    , CreditCardID)));
GO
-- Verify the encryption.
-- First, open the symmetric key with which to decrypt the data.
OPEN SYMMETRIC KEY CreditCards_Key11
    DECRYPTION BY CERTIFICATE Sales09;
GO
-- Now list the original card number, the encrypted card number,
-- and the decrypted ciphertext. If the decryption worked,
-- the original number will match the decrypted number.
SELECT CardNumber, CardNumber_Encrypted
    AS 'Encrypted card number', CONVERT(nvarchar,
    DecryptByKey(CardNumber_Encrypted, 1 ,
    HASHBYTES('SHA2_256', CONVERT(varbinary, CreditCardID))))
    AS 'Decrypted card number' FROM Sales.CreditCard;
GO
```

c) Chiffrer avec le chiffrement symétrique simple

```
CREATE CERTIFICATE HumanResources037
  WITH SUBJECT = 'Employee Social Security Numbers';
GO
CREATE SYMMETRIC KEY SSN_Key_01
  WITH ALGORITHM = AES_256
  ENCRYPTION BY CERTIFICATE HumanResources037;
GO
USE [AdventureWorks2012];
GO
-- Create a column in which to store the encrypted data.
ALTER TABLE HumanResources.Employee
  ADD EncryptedNationalIDNumber varbinary(128);
GO
-- Open the symmetric key with which to encrypt the data.
OPEN SYMMETRIC KEY SSN_Key_01
  DECRYPTION BY CERTIFICATE HumanResources037;
-- Encrypt the value in column NationalIDNumber with symmetric
-- key SSN_Key_01. Save the result in column EncryptedNationalIDNumber.
UPDATE HumanResources.Employee
  SET EncryptedNationalIDNumber = EncryptByKey(Key_GUID('SSN_Key_01'),
  NationalIDNumber);
GO
-- Verify the encryption.
-- First, open the symmetric key with which to decrypt the data.
OPEN SYMMETRIC KEY SSN_Key_01
  DECRYPTION BY CERTIFICATE HumanResources037;
GO
-- Now list the original ID, the encrypted ID, and the
-- decrypted ciphertext. If the decryption worked, the original
-- and the decrypted ID will match.
SELECT NationalIDNumber, EncryptedNationalIDNumber
  AS 'Encrypted ID Number',
  CONVERT(nvarchar, DecryptByKey(EncryptedNationalIDNumber))
  AS 'Decrypted ID Number'
  FROM HumanResources.Employee;
GO
```

3.4 Création des Audits

3.4.1 Créer un Audit du Serveur et une spécification

Avec T-SQL

-- Creates a server audit called "HIPAA_Audit" with a binary file as the target and no options.

```
CREATE SERVER AUDIT HIPAA_Audit
    TO FILE ( FILEPATH = 'E:\SQLAudit\' );
```

/*Creates a server audit specification called "HIPAA_Audit_Specification" that audits failed logins for the SQL Server audit "HIPAA_Audit" created above. */

```
CREATE SERVER AUDIT SPECIFICATION HIPAA_Audit_Specification
FOR SERVER AUDIT HIPAA_Audit
    ADD (FAILED_LOGIN_GROUP);
```

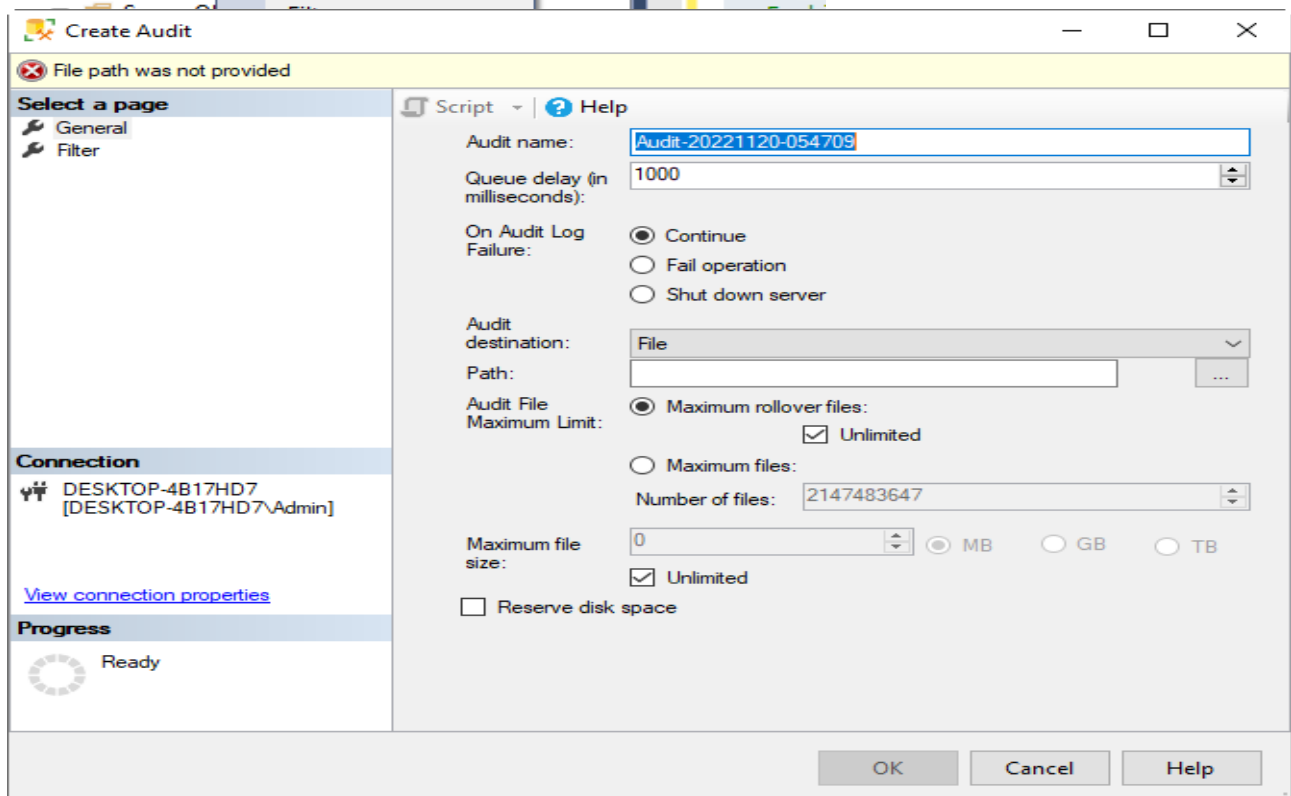
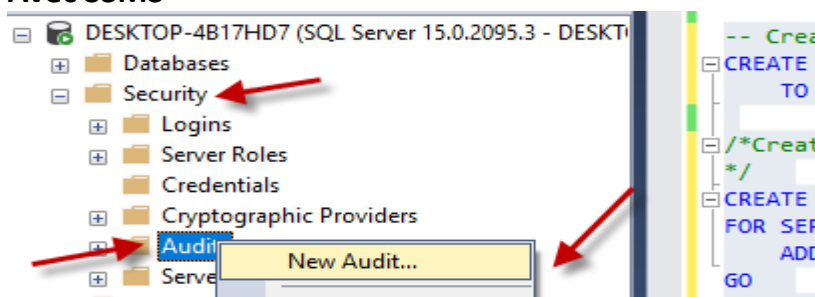
GO

-- Enables the audit.

```
ALTER SERVER AUDIT HIPAA_Audit
WITH (STATE = ON);
```

GO

Avec SSMS



3.4.2 Créer une spécification de l'audit de la Base de données

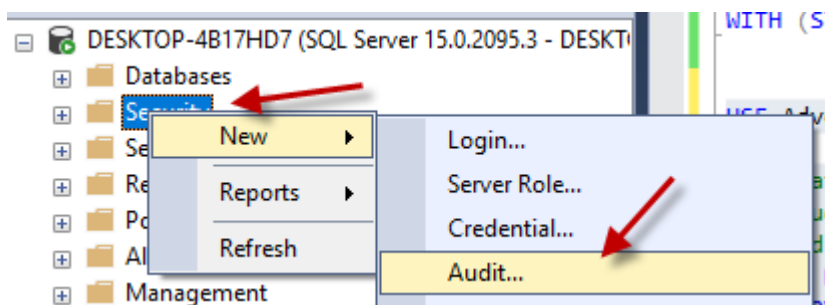
Avec T-SQL

```
USE master ;
GO
-- Create the server audit.
CREATE SERVER AUDIT Payrole_Security_Audit
    TO FILE ( FILEPATH = 'C:\Program Files\Microsoft SQL
Server\MSSQL15.MSSQLSERVER\MSSQL\DATA' ) ;
GO
-- Enable the server audit.
ALTER SERVER AUDIT Payrole_Security_Audit
WITH (STATE = ON) ;

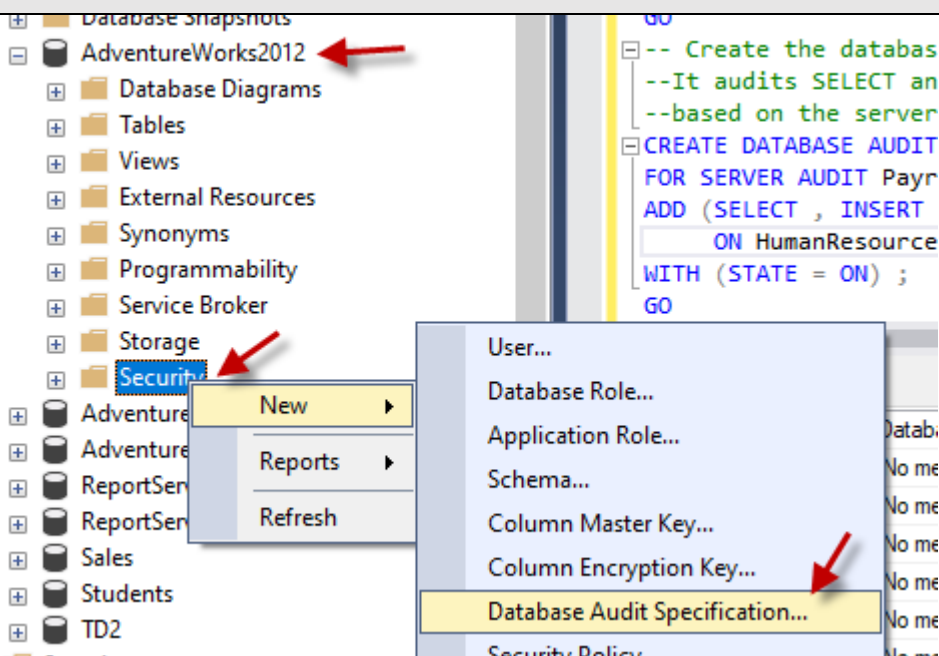
USE AdventureWorks2012 ;
GO
-- Create the database audit specification.
--It audits SELECT and INSERT statements by the dbo user for the
HumanResources.EmployeePayHistory table,
--based on the server audit defined in the previous section.
CREATE DATABASE AUDIT SPECIFICATION Audit_Pay_Tables
FOR SERVER AUDIT Payrole_Security_Audit
ADD (SELECT , INSERT
    ON HumanResources.EmployeePayHistory BY dbo )
WITH (STATE = ON) ;
GO
```

Avec SSMS

Pour créer l'audit du Serveur



Pour créer la spécification d'audit de niveau Base de données
Sélectionnez la base de données AdventureWorks2012 par exemple et dans le dossier Security (clique-droit>New>Database Audit Specification.

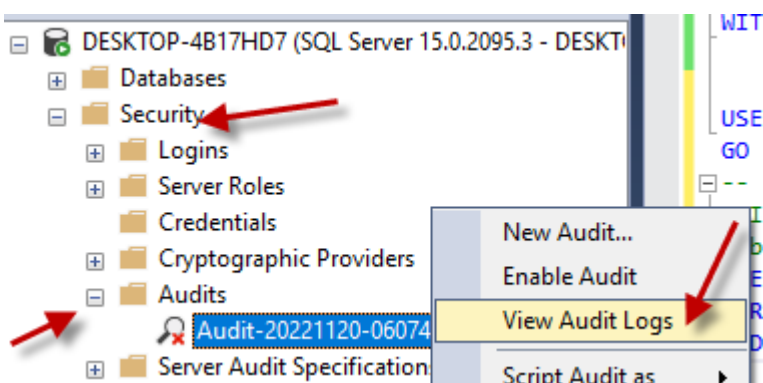


Remarque : Il faut toujours créer le **Audit Server** avant de créer le **Database Audit Specification**

3.4.3 Affichez les journaux d'audit

Pour afficher un journal d'audit SQL Server

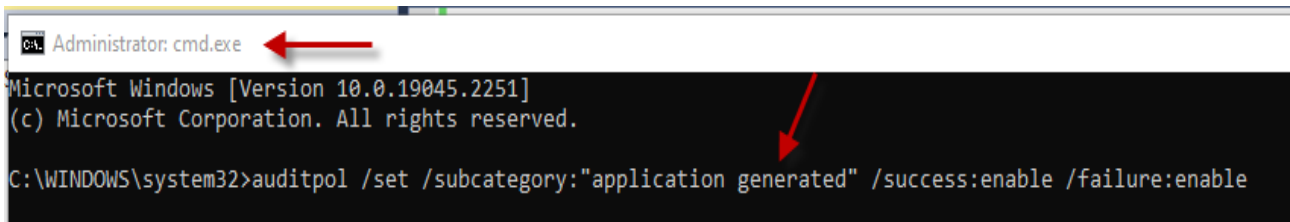
1. Dans l'Explorateur d'objets, développez le dossier **Sécurité**.
2. Développez le dossier **Audits**.
3. Cliquez avec le bouton droit sur le journal d'audit que vous souhaitez afficher et sélectionnez **Afficher les journaux d'audit**. La boîte de dialogue **Visionneuse de fichiers journaux -server_name** s'ouvre. Pour plus d'informations, consultez [Log File Viewer F1 Help](#).



3.4.4 Écrire les événements d'audit dans le journal Windows (Events Viewer)

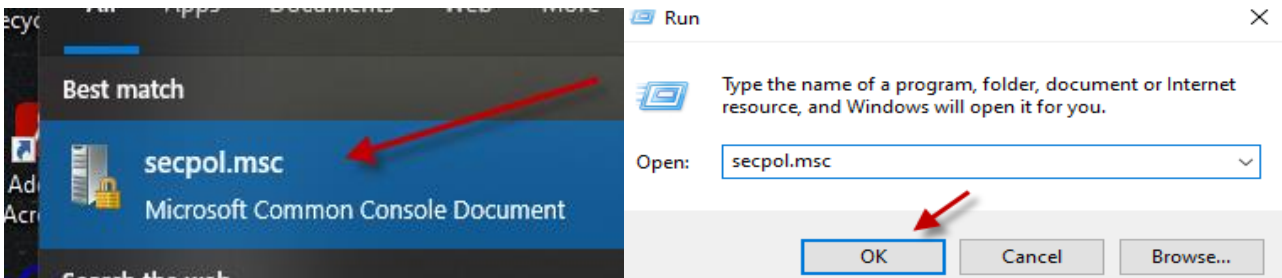
a) Pour configurer le paramètre Auditer l'accès aux objets dans Windows à l'aide de l'outil **auditpol** :

Ouvrir l'invite de commande Windows en tant qu'Administrateur puis taper la commande suivante : `auditpol /set /subcategory:"application generated" /success:enable /failure:enable`

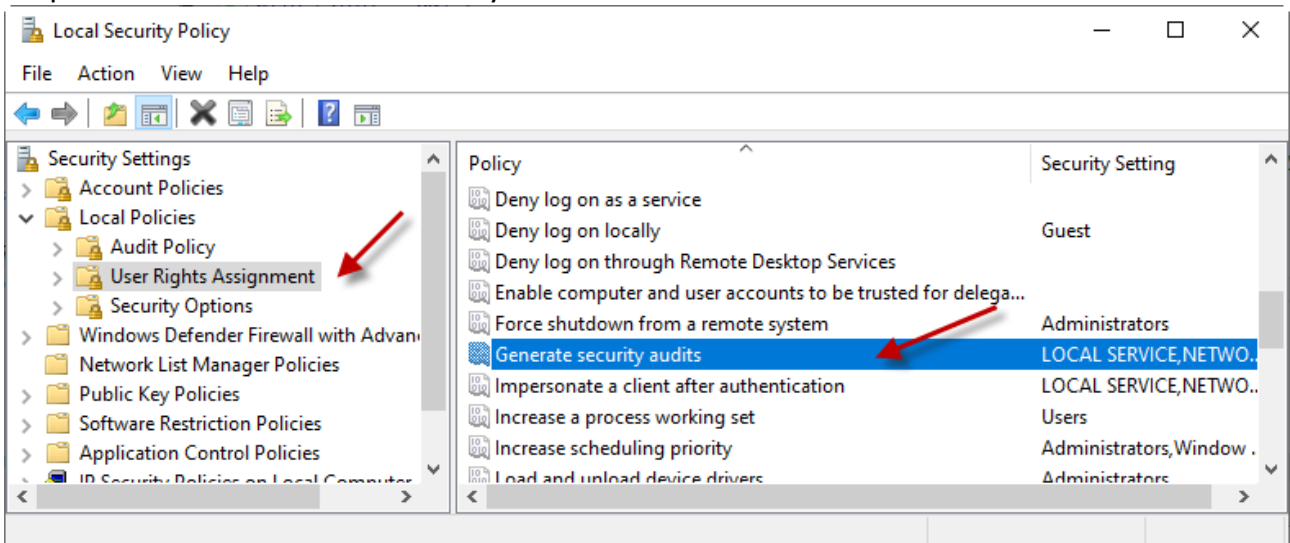


b) Pour octroyer l'autorisation Générer des audits de sécurité à un compte à l'aide de l'outil **secpol**

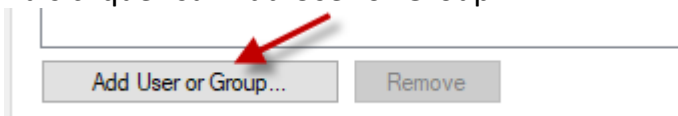
Dans le menu **SEARCH** ou **Start>Run** de Windows, tapez **SECPOL.MSC** puis **OK**



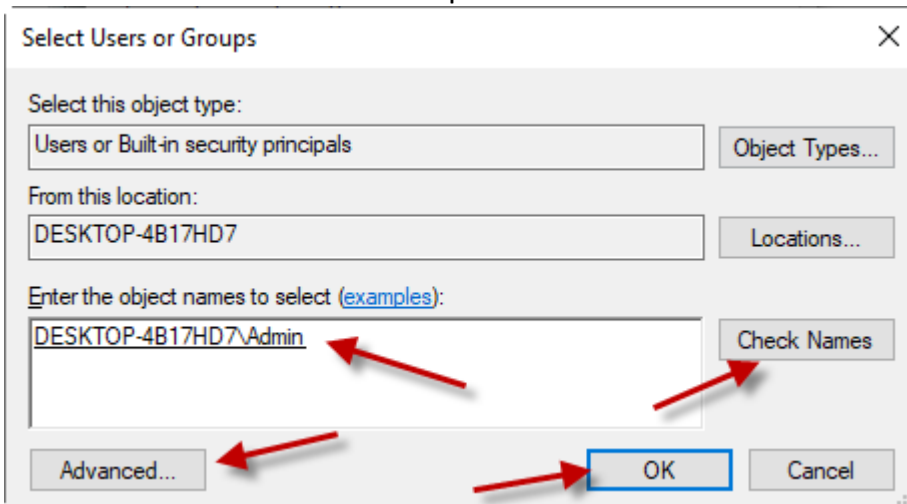
Cliquez 2 fois sur **Generate Security Audits**



Puis cliquez sur **Add User or Group**

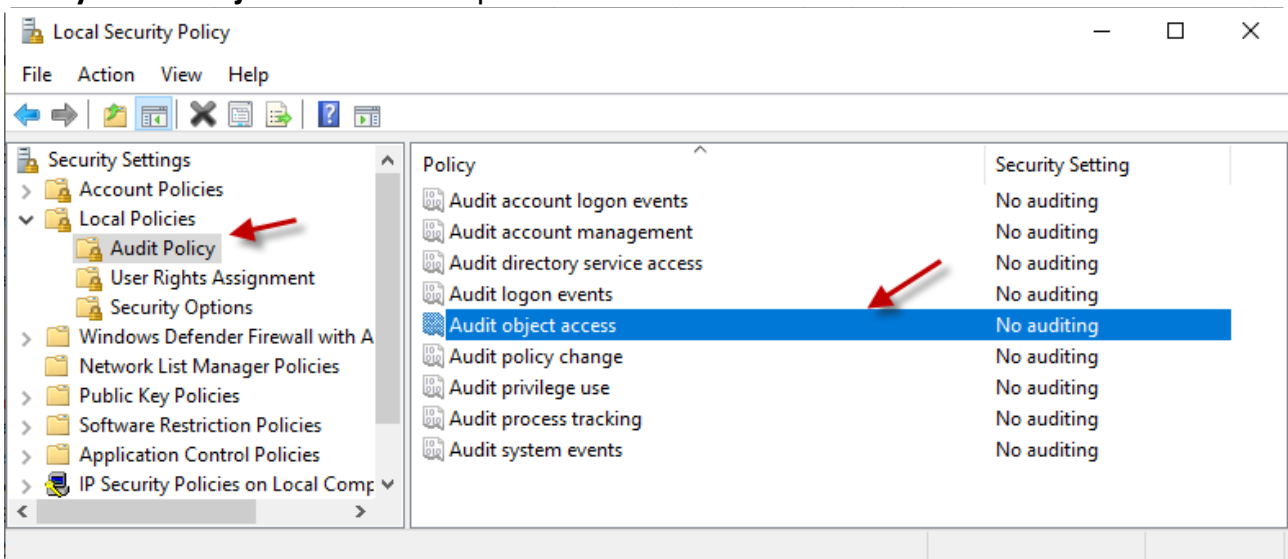


Dans la boîte de dialogue **Select Users or Groups**, tapez le nom du compte d'utilisateur, tel que **domaine1\utilisateur1** puis **Check Names**, puis cliquez sur **OK**, ou cliquez sur **Advanced** et recherchez le compte.



c) Pour configurer le paramètre Auditer l'accès aux objets dans Windows à l'aide de l'outil **secpol**

Lancez l'outil SECPOL de Windows comme dans la partie b) puis allez dans **Audit Policy>Audit Object Access** et cliquez 2 fois dessus.



Ensuite cochez Success et Failure puis cliquez sur Apply et OK

Audit these attempts:

☒ Success

☒ Failure



This setting might not be enforced if other policy is configured to override category level audit policy.
For more information, see [Audit object access](#). (Q921468)

4. Références

- <https://learn.microsoft.com/en-us/sql/relational-databases/security/security-center-for-sql-server-database-engine-and-azure-sql-database?view=sql-server-ver16>
- <https://learn.microsoft.com/en-us/sql/relational-databases/security/sql-server-security-best-practices?view=sql-server-ver16>
- <https://www.w3schools.com/sql>
- https://www.w3schools.com/sql/sql_ref_sqlserver.asp
- <https://learn.microsoft.com/en-us/sql/relational-databases/security/authentication-access/create-a-login?view=sql-server-ver16>