

SQL Server

Administration

SQL Server : Admin.



Formateur : Vincent Dedeycker
v.dedeycker@wavenet.be



Plan du cours

- Chapitre 1 : Installation de SQL Server
- Chapitre 2 : Gestion des fichiers et des bases de données
- Chapitre 3 : Gestion de la sécurité
- Chapitre 4 : Backup et restauration
- Chapitre 5 : Automatisation des tâches administratives
- Chapitre 6 : Monitoring
- Chapitre 7 : Transfert des données
- Chapitre 8 : Introduction à la réplication

Chapitre 1

Installation de SQL Server

SQL Server : Admin.



Objectifs

- Connaître les configurations minimales pour installer SQL Server
- Savoir installer et configurer SQL Server
- Connaître les procédures pour installer et supprimer des composants

Plan

- Préparation de l'installation
- Installation
- Gérer SQL Server

SQL Server : Admin.

Préparation de l'installation

- Edition de SQL Server
- Configuration minimale du hardware
- Configuration minimale du software
- Différents type de licence
- Qu'est ce qu'une instance de SQL Server?
- Considérations de sécurité
- Considérations de classement
- Considérations d'upgrade

Principales éditions de SQL Server

Edition	Description
Enterprise Edition	Destinée aux entreprises ayant des besoins de haute disponibilité et Data Warehouse: L'ajout d'une licence de software assurance est nécessaire pour certaines fonctionnalités.
Standard Edition	La version Standard offre toutes les fonctionnalités basiques d'une base de données d'entreprise, avec des fonctionnalités d'analyse et de reporting très puissantes.
Express Edition	La version Express est conçue pour de très petites applications et est sous licence gratuite
Azure Sql Server	Une version 100% cloud avec une flexibilité très utile.

Principales éditions de SQL Server

Feature	Enterprise	Standard	Express
Maximum compute capacity used by a single instance - SQL Server Database Engine ¹	Operating system maximum	Limited to lesser of 4 sockets or 24 cores	Limited to lesser of 1 socket or 4 cores
Maximum compute capacity used by a single instance - Analysis Services or Reporting Services	Operating system maximum	Limited to lesser of 4 sockets or 24 cores	Limited to lesser of 1 socket or 4 cores
Maximum relational database size	524 PB	524 PB	10 GB

Configuration minimale du hardware (2017-2019)

Hardware	Configuration minimale
Processeur	Minimum : Processeur x64 : 1,4 GHz Recommandé : 2,0 GHz
Mémoire	Minimum: 1Go de RAM Recommandé: 4Go de RAM
Espace disque	Minimum: 6Go Remarque: L'espace disque nécessaire peut varier selon la configuration du système et les fonctionnalités installées.

Configuration minimale OS

	Standard	BI	Enterprise
Windows Server 2008	*	*	*
Windows Server 2012	*	*	*
Windows 7	*		
Windows 8	*		
Windows 8.1	*		
Windows 10	*		

SQL Server : Admin.

Configuration minimale du Software (2017)

Software	Requirements
.NET Framework	.NET 4.6
Windows PowerShell	PowerShell 2,0
Windows Installer	Windows Installer 4.5 or later.

SQL Server : Admin.

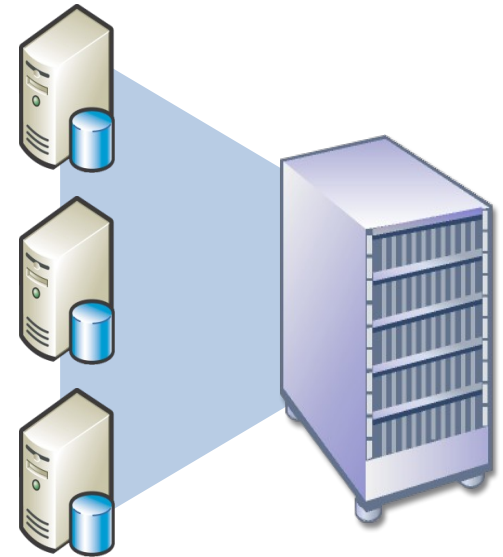
Différents type de licence

- Processor licence
 - Nécessite une licence pour chaque CPU du système qui utilise SQL Server. Cette licence ne nécessite aucun outil ou licence d'accès client.
- Serveur plus utilisateurs CALs (licence d'accès au client)
 - Nécessite une licence pour l'ordinateur qui utilise SQL Server, et un CAL pour chaque utilisateur.

<https://www.microsoft.com/fr-be/sql-server/sql-server-2017-pricing>

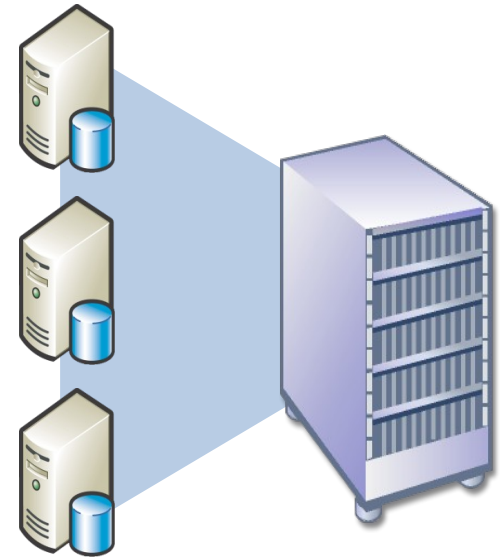
Qu'est ce qu'une instance de SQL Server?

- Une instance de la base de données SQL Server a son propre ensemble d'instances (fichiers de données, programmes spécifiques).
- Les fichiers communs sont partagés entre les instances.



Qu'est ce qu'une instance de SQL Server?

- Instance par défaut (MSSQLSERVER) : identifié par le nom de la machine sur laquelle elle fonctionne.
- Instance nommée : identifié par le format `Nom_machine\nom_instance`.
- Une instance est une installation de toute ou d'une partie des services SQL Server sur une machine (jusqu'à 50 instances différentes),
- Une base est limitée à 524 Po, mais on peut héberger 32760 bases dans une même instance.



Considérations de sécurité

- Un compte Windows est assigné aux différents services de SQL Server.
- Les services:

Services	description
SQL Server	Le moteur de base de données SQL Server
SQL Server Agent	Le service qui exécute les jobs, fait le monitoring de SQL Server, et permet l'automatisation des tâches administratives.
Analysis Services	Le moteur qui permet de réaliser une analyse OLAP et de faire du data mining
Report Server	Le moteur de reporting.
SQL Server Browser	Le service qui donne les informations de connexion à SQL aux ordinateur clients.

Considérations de sécurité

- SQL Server Service Accounts:

- Un compte utilisé par un service est un « Service Account »

- Types de « Service Account »:

Domaine User Account

Utilisé par Windows Authentication pour démarrer et se connecter à SQL Server.

Préférable quand les services doivent interagir avec des services réseau.

Local User Account

Si le serveur ne fait pas parti d'un domaine, le Local User Account est recommandé.

Considérations de classement

- Le classement (« Collation ») réfère à l'ensemble de règles qui déterminent comment les données sont comparées et classées.
- Le classement spécifie trois propriétés
 - L'ordre de tri à utiliser pour les types de données Unicode (nchar, nvarchar et ntext).
 - L'ordre de tri à utiliser pour les type non Unicode
 - La page de code utilisé pour enregistrer les caractères non Unicode.
- Le classement par défaut est Latin1_General.

Considérations de classement

- Le classement SQL
 - SQL Server cherche les correspondances entre ses règles de classement et celles qui étaient spécifiées dans les versions précédentes.
 - Le classement SQL contrôle les pages de codes utilisés pour stocker les données non-Unicode et les règles de classement des données unicodes et non-Unicode

Installation

- Processus d'installation de SQL Server 2017
- Options d'installation des composants
- Installation non assistée
- Installation à distance

Installation

1 Mise à jour des composants

- .Net Framework 4.6
- Windows installer 4.5

2 Centre d'installation de SQL Server

- Nouvelle installation ou ajouter des fonctionnalités à une installation existante

3 Outils d'analyse de configuration

- Génération de log file.
- Check la version de l'OS, les droits du user courant, restart requis, les services, clé de registre cohérent, longueur du chemin d'accès, check si la version est compatible

4 Setup Supported files

- Mise en place des composants d'installation
- Redémarrage nécessaire

Installation

5 Installation

- Outils d'analyse de configuration
- Clé du produit
- Licence
- Setup role
- Installation des fonctionnalités
- Configuration d'instance (nom)
- Server configuration (service)
- Configuration des fonctionnalités

SQL Server : Admin.

Options d'installation des composants

Composant Serveur	Notes
Moteur de base de données SQL Server	Le moteur de base de données SQL Server inclut le moteur de base de données, le service principal de stockage, de traitement et de protection des données, la réplication, la recherche en texte intégral et les outils d'administration de données relationnelles et XML.
Analysis Services	Analysis Services inclut les outils de création et de gestion d'applications de traitement analytique en ligne (OLAP, OnLine Analytical Processing) et d'exploration de données.
Reporting Services	Reporting Services inclut les composants serveur et clients permettant de créer, de gérer et de déployer des rapports tabulaires, de matrice, graphiques et de forme libre. Reporting Services est également une plateforme évolutive que vous pouvez utiliser pour développer des applications de création de rapports.
Integration Services	Integration Services propose un ensemble d'outils graphiques et d'objets programmables permettant de déplacer, de copier et de transformer les données.

Options d'installation des composants

Outils d'Administration	Notes
SQL Server Management Studio	SQL Server Management Studio est un environnement intégré permettant d'accéder aux composants de SQL Server en vue de les configurer, de les gérer, de les administrer et de les développer. Management Studio permet aux développeurs et aux administrateurs de tous niveaux de compétence d'utiliser SQL Server. Internet Explorer 6.0 SP1 ou version ultérieure est requis pour l'installation de Management Studio.
Gestionnaire de configuration SQL Server	Le Gestionnaire de configuration SQL Server permet de gérer la configuration de base des services SQL Server, des protocoles serveur et clients et des alias clients.
SQL Server Profiler	SQL Server Profiler fournit une interface utilisateur graphique qui permet d'analyser une instance du moteur de base de données ou Analysis Services.

Installation non assistée

- Se fait en créant un fichier « .ini » contenant les informations d'installation requises, puis en exécutant « setup.exe » en ligne de commandes.



```
setup.exe {/qn ou /qb} /settings <path to .ini file>
```


Installation à distance

- Peut être faite en ligne de commande en utilisant un fichier « .ini »
- Ce fichier doit inclure les paramètres suivants:

Paramètre	Description
TARGETCOMPUTER	Le nom de l'ordinateur sur lequel SQL Server sera installé
ADMINACCOUNT	Le compte avec des privilèges d'administration à utiliser pour se loguer
ADMINPASSWORD	Le mot de passe de ce compte

Gérer SQL Server

- SQL Server Management Studio
- Policy-Based Management
- Sqlcmd

SQL Server Management Studio

- Outil pour effectuer toutes les tâches d'administration de SQL Server
- Propriétés :
 - Outil de développement et de gestion intégré basé sur l'environnement de développement Visual Studio
 - Centre de gestion de tous les composants et services de SQL Server.
 - Graphique et gestion des objets
 - Gestion des scripts (*T-SQL*, *MDX*, *DMX*, *XMLA*)
 - Gestion des projets.
- Démonstration : Utiliser SQL Server Management Studio
 - Adventureworks

Introduction au T-SQL

- Transact Structured Query Language
- ajoute les fonctionnalités suivantes :
 - Éléments de programmation procédurale (boucle, conditions...) ;
 - La possibilité de créer et d'utiliser des variables locales ;
 - Des fonctions de manipulations de chaîne de caractères, de dates et de calculs mathématiques.
- Mot réservés, espaces,... à encadrer par des [].

Reprise de la base de données AdventureWorks

- Base de données d'exemple fournie par Microsoft
- Utilisée pour des démonstrations et des labos
- Représente une société multinationale de vente.
- <https://docs.microsoft.com/en-us/sql/samples/adventureworks-install-configure?view=sql-server-ver15>

Sqlcmd

- Utilisé pour exécuter des requêtes et des scripts T-SQL
- Exécute des requêtes
 - `sqlcmd -S DBSERVER1\SQLINSTANCE1`
 - `Sqlcmd -?`
 - spécifier la DB (USE), entrer la commande et l'exécuter (GO)
- Exécuter des scripts
 - Exécuter des scripts sauves dans des fichiers .sql
 - Exécuter des scripts utilisant des variables
 - Ex :
Script :
 `SELECT $(col) FROM $(tab)`
 `GO`
 `Sqlcmd -i Query.sql -o Results.out -v col="name" tab="sys.databases"`
- Exercice : Créer un fichier `cherche_info_employee.sql`.

Chapitre 2

Gestion des fichiers et des bases de données

SQL Server : Admin.

Objectifs

- Comprendre comment Microsoft SQL Server
 - Stocke les données
 - Traite les transactions
 - Utilise les fichiers et les « filegroups »
- Savoir prédire combien d'espace disque la base de données pourrait consommer.

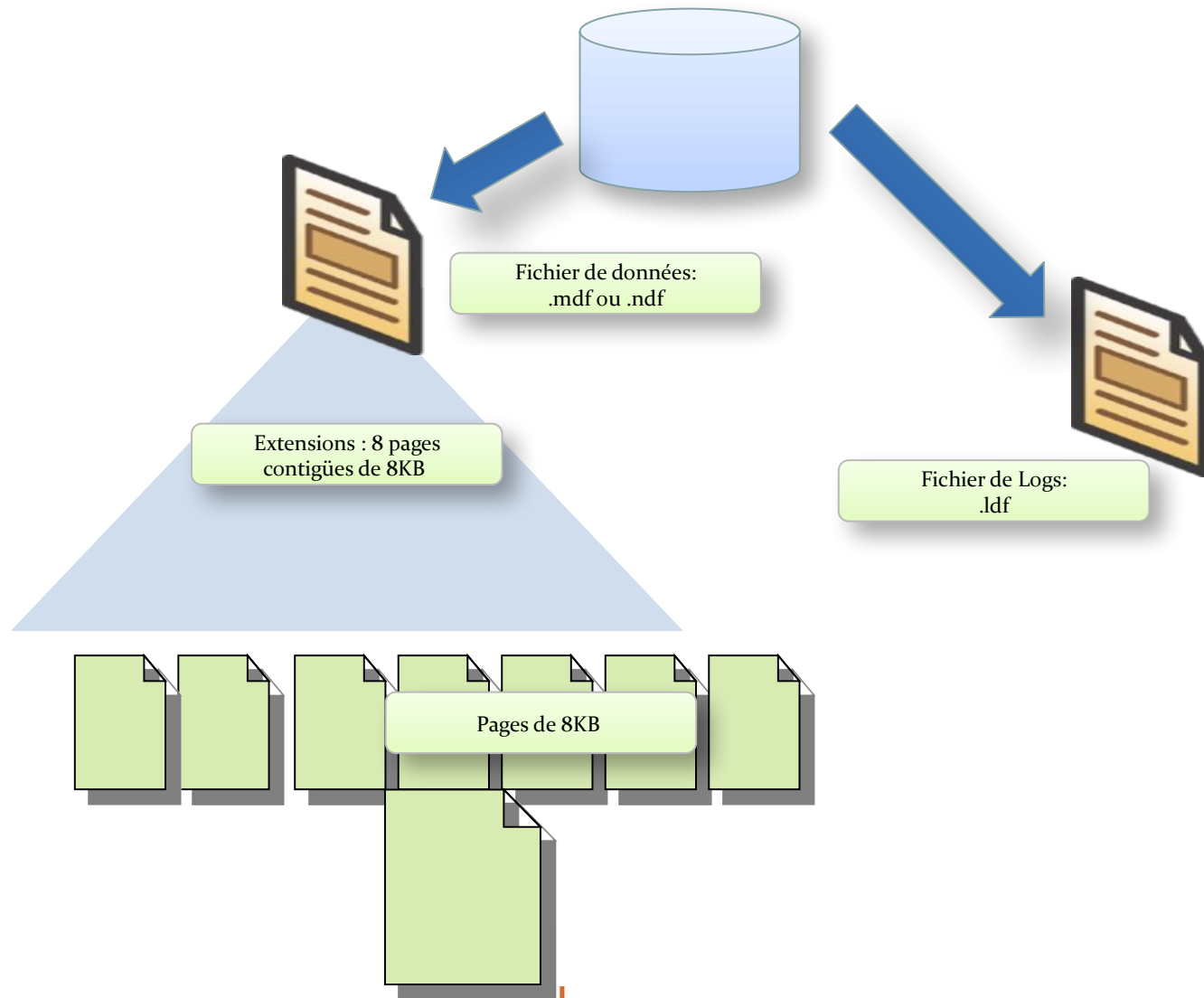
Plan

- Planification d'une base de données
- Création d'une base de données
- Gestion d'une base de données

Planification d'une base de données

- Comment les données sont-elles stockées?
- Fonctionnement du « Transaction Log »
- Emplacement des fichiers
- Base de données système
- « FileGroups »
- Capacité de planification

Comment les données sont-elles stockées?



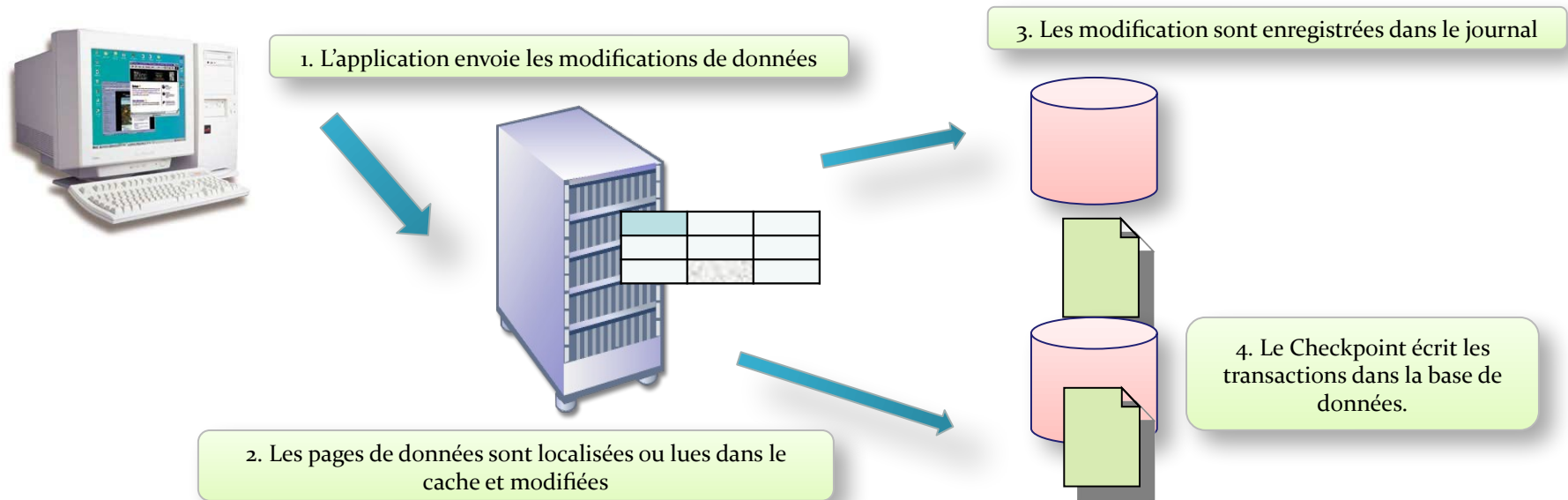
Fonctionnement du « Transaction Log »

- Une transaction est un ensemble d'une ou plusieurs instructions T-SQL traité comme une seule instruction.
- SQL Server exécute des transactions implicites et explicites
 - Implicites : quand l'une de ces instructions est exécutée:
ALTER TABLE, INSERT, CREATE, OPEN, DELETE, REVOKE, DROP, SELECT, FETCH,
TRUNCATE TABLE, GRANT, UPDATE
 - Explicites : quand le début et la fin de la transaction sont spécifiés par BEGIN TRANSACTION et COMMIT TRANSACTION
- Mode par défaut : autocommit

Fonctionnement du « Transaction Log »

- Chaque transaction est enregistrée dans le journal des transactions.
- Le « LOG », ou journal, est l'aire de stockage qui piste automatiquement les changements dans la DB.

Fonctionnement du « Transaction Log »



Emplacement des fichiers

- Types de fichiers:
 - Fichier primaire :
 - contient les informations de démarrage de la base de données et pointeurs vers ses autres fichiers.
 - Données utilisateurs et objets
 - Chaque base de données a un et un seul fichier primaire (.mdf)
 - Fichiers secondaires
 - Données utilisateurs
 - Une base de données peut avoir plusieurs fichiers secondaires
 - Ils peuvent être stockés dans des disques séparés (.ndf)

Emplacement des fichiers

- Types de fichiers:
 - Journal des transactions :

Chaque base de données doit en avoir un
Il est, en général, créé automatiquement avec un nom par défaut.
En général, 25% des fichiers de la base de données
(.ldf)

Emplacement des fichiers

- Emplacement des fichiers
 - Permet d'améliorer les performances et de tolérer les failles.
 - Pour les grandes bases de données, il faut déployer le plus possible les données dans le plus possible de disques physiques.
 - En général, il faut créer un fichier pour chaque disque physique et grouper les fichiers dans un ou plusieurs « filegroups » (voir plus loin).

Emplacement des fichiers

- Les journaux de transactions
 - Le journal de transactions peut être créé dans des disques séparés.
 - Ex: environnement de production => plusieurs bases de données dans le serveur => il faut utiliser des disques séparés pour chaque journal de transactions.
- Emplacement de la base de données **tempdb**
 - Doit être placée dans un sous système d'I/O et doit être séparée des bases de données utilisateurs pour assurer une performance optimale
 - On peut la déployer dans plusieurs disques afin d'assurer une meilleure performance.

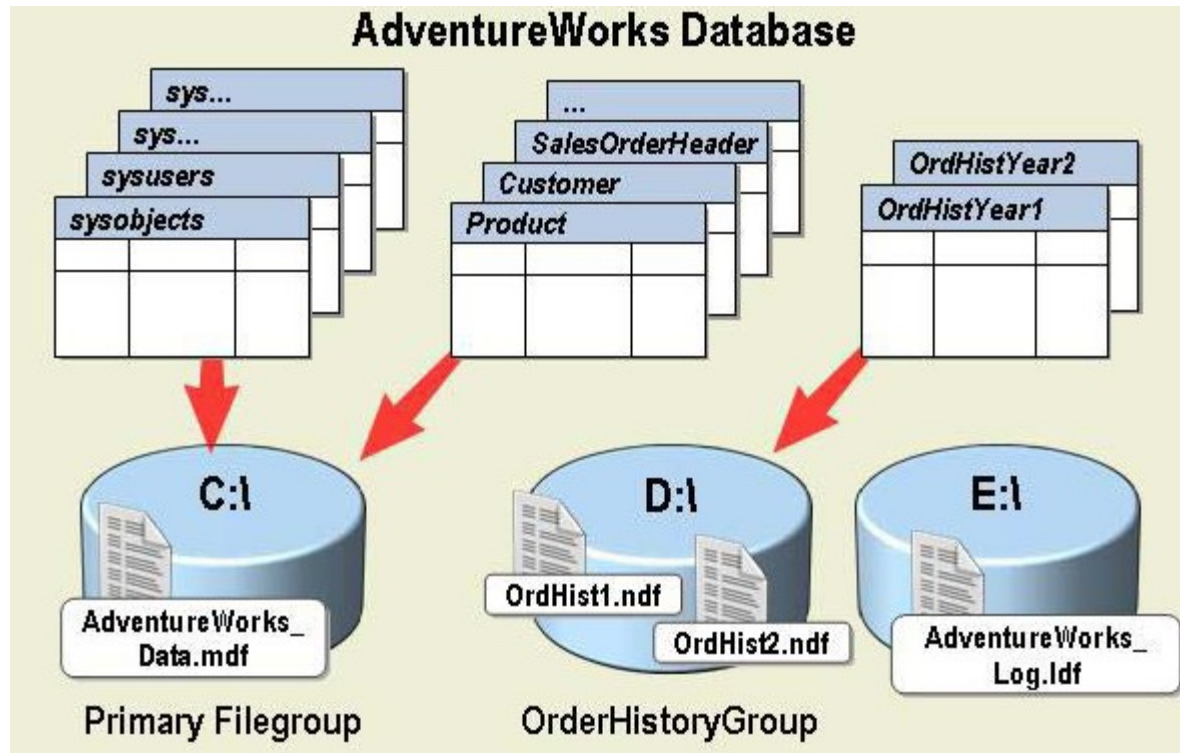
Base de données système

Database	Notes
master	Enregistre toutes les données au niveau du Server SQL. Ceci inclus les metadata d'instance comme les comptes, endpoints (service broker , SOAP), les serveurs liés et les paramètres de configuration.
model	Utilisée comme modèle pour la création des base de données dans une instance SQL. Vu que la DB tempdb est créer à chaque démarrage de SQL Server, la DB model doit toujours exister sur le serveur SQL (persistance des données)
msdb	Utilisée par SQL Agent pour organiser les alertes et les jobs, utilisé aussi par d'autres système comme Management Studio, Service Broker and Database Mail
tempdb	<p>Ressource Global disponible pour tout les utilisateurs connectés à une instance SQL Server :</p> <ul style="list-style-type: none">• Objets temporaire : tables temporaire globales ou locales, « stored procedures » temporaires, variables de tables ou cursors• Versions de lignes qui ont été générés par des modifications de données lors de transaction dans la DB.

« Filegroups »

- Les filegroups consistent en un ensemble de fichiers. Cette notion permet de simplifier le placement des fichiers.
- On peut localiser un objet ou fichier spécifique présent dans un disque en groupant les fichiers de la base de données à l'intérieur d'un ou plusieurs « filegroups ».

« Filegroups »



« Filegroups »

- Types de filegroups

- Filegroup primaire:

Chaque base de données à un seul filegroup primaire.

Il contient le fichier de données primaire contenant les tables système, et peut contenir les fichiers secondaire qui ne sont alloués à aucun autre filegroup.

- Filegroups définis par l'utilisateur : consistent en des fichiers de données groupés pour des raisons administratives, par exemple.

« Filegroups »

- Quand faut-il créer des filegroups définis par l'utilisateur?
 - Performance

Création de plusieurs fichiers appartenant au même filegroup dans différents disques physiques

Assigner les tables auxquelles on accède le plus fréquemment à ce filegroup
 - Faciliter la gestion

On peut utiliser différents filegroups pour séparer les données avec différents besoins de gestion.

« Filegroups »

- Filegroup par défaut :
 - Sauf indication contraire, lors de la création de la base de données, le filegroup par défaut est le filegroup primaire.
 - Il contient les pages pour toutes les tables et indexes qui n'ont pas de filegroup spécifié lors de leur création.
 - Il doit être bien dimensionné.

« Filegroups »

- Les filegroups en lecture seulement
 - Tous les filegroups, sauf le primaire, peuvent être en lecture seule.
 - Pour éviter des altérations accidentelles, placer les tables concernées dans un filegroup marqué « read-only ».
 - SQL Server supporte la compression NTFS des filegroups en lecture seulement et des base de données en lecture seulement.

Capacité de planification

- Estimer la taille de la base de données
 - Planifier une base de données = mettre en place une structure logique
 - Chaque élément de base de données doit être considéré. (tables utilisateurs et systèmes, indexes, transaction logs, LOBs ...)
 - Quand on crée une base de données, SQL Server crée une copie du modèle des objets de la base de données. Le système augmente de taille au fur et à mesure qu'on crée des objets dans la base de données.

Capacité de planification

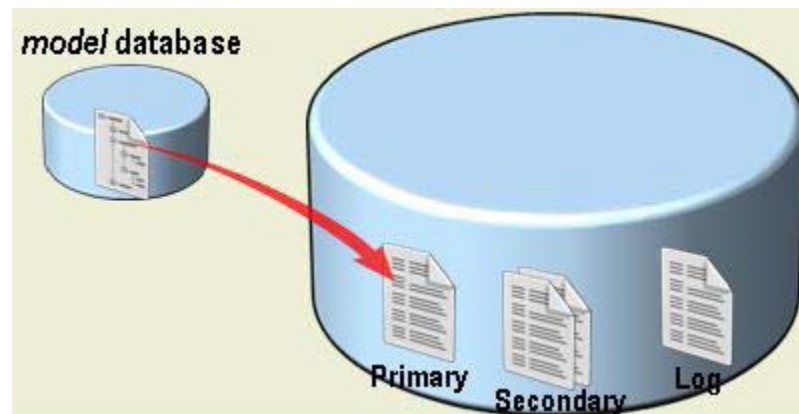
- Estimer la taille de la base de données
 - Facteurs à considérer lors de l'estimation de la taille de la base de données
 - La taille de l'objet, en incluant les projections des futures évolutions.
 - La quantité de données dans les tables, avec leurs évolutions.
 - Le nombre et la taille des indexes
 - La taille des journaux de transactions.
 - La taille des tables systèmes

Création d'une base de données

- Création d'une base de données
- Syntaxe pour créer une base de données
- Les Schémas
- Syntaxe pour créer un Schéma
- « Name Resolution » (vérification de la disponibilité du nom)

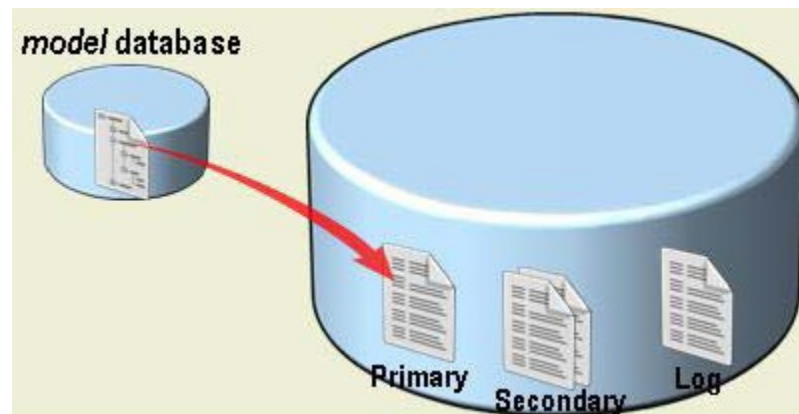
Création d'une base de données

- Une base de données peut être créée à l'aide de SQL server Management Studio ou de l'instruction T-SQL: CREATE DATABASE
- Quand on crée une base de données, SQL Server:
 - Crée le fichier de données et le journal de transactions
 - Vérifie que le créateur a toutes les permissions pour utiliser la base de données master (sysdatabases et sysaltfiles)



Création d'une base de données

- Quand on crée une base de données, SQL Server:
 - Permet de définir le nom de la base de données, ses propriétés et la localisation de ses fichiers.
 - Utilise une copie des objets dans le modèle de base de données pour initialiser la base de données et ses métadonnées.
 - Remplit le reste de la base de données par des pages vides.



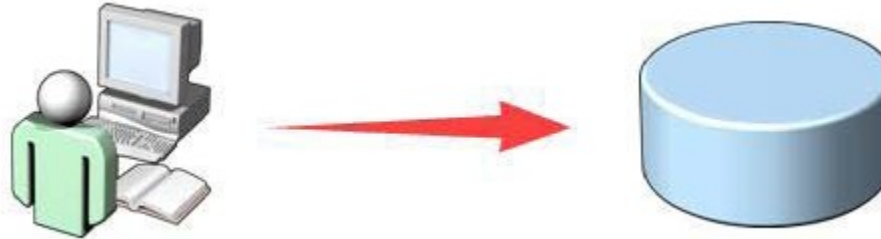
Création d'une base de données

- Options de création de la base de données:
 - Fichier primaire (nom_de_la_db.mdf) : localisé dans le répertoire des données pour cette instance de SQL Server.
 - Fichiers secondaires : .ndf est l'extension recommandée
 - Journal des transactions (nom_de_la_db.ldf) : localisé dans le répertoire des données pour cette instance de SQL Server.
 - Taille : on peut spécifier la taille pour chaque fichier de données et fichier de log. Minimum 1MB pour le fichier de données et 512MB pour les fichiers de log.

Création d'une base de données

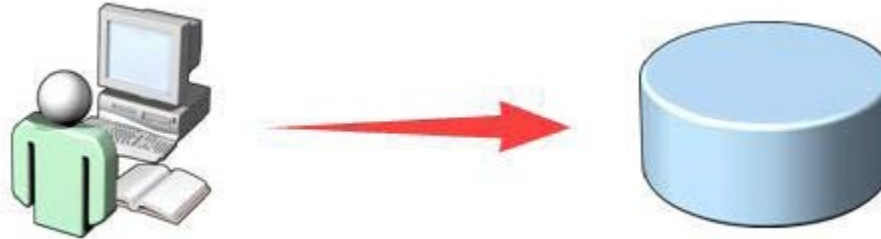
- Options de création de la base de données:
- Croissance des fichiers: on peut spécifier si le fichier va croître en taille ou pas (autogrow). On peut aussi spécifier de combien il va croître (en MB ou %).
 - Par défaut : 10%
 - Taille maximum (en MB ou %)
 - Le classement : par défaut, la base de données hérite du classement de l'instance de SQL Server dans laquelle elle est créée. Cependant, on peut toujours le redéfinir.

Syntaxe pour créer une base de données



```
CREATE DATABASE database_name
[ ON
[ PRIMARY ] [ ( NAME = logical_file_name, FILENAME = 'os_file_name'
[ , SIZE = size [ KB | MB | GB | TB ]
[ , MAXSIZE = { max_size [ KB | MB | GB | TB ] | UNLIMITED } ]
[ , FILEGROWTH = growth_increment [ KB | MB | GB | TB | % ] ) ]
[ FILEGROUP filegroup_name [DEFAULT]
( NAME = logical_file_name, FILENAME = 'os_file_name'
[ , SIZE = size [ KB | MB | GB | TB ]
[ , MAXSIZE = { max_size [ KB | MB | GB | TB ] | UNLIMITED } ]
[ , FILEGROWTH = growth_increment [ KB | MB | GB | TB | % ] ) ]
[ , ... n ] ]
[ LOG ON
[ ( NAME = logical_file_name, FILENAME = 'os_file_name'
[ , SIZE = size [ KB | MB | GB | TB ]
[ , MAXSIZE = { max_size [ KB | MB | GB | TB ] | UNLIMITED } ]
[ , FILEGROWTH = growth_increment [ KB | MB | GB | TB | % ] ) ]
[ , ... n ] ]
[ COLLATE collation_name ] ]
```

Syntaxe pour créer une base de données

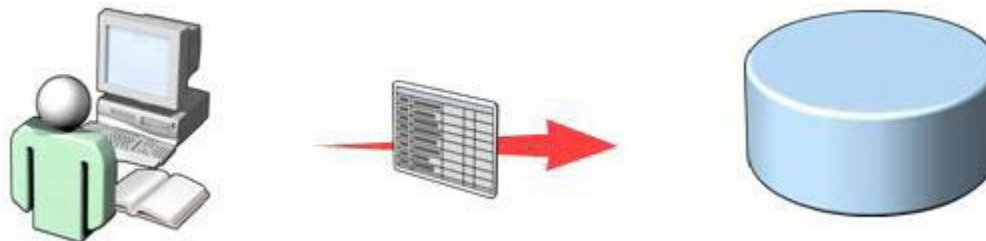


```
CREATE DATABASE Personnel
ON PRIMARY
(NAME = N'PersData1', FILENAME = N'C:\DemoProject\PersData1.mdf',
SIZE = 5 MB, MAXSIZE = 10 MB, FILEGROWTH = 0),
FILEGROUP PersDefFilegroup DEFAULT
(NAME = N'PersData2', FILENAME = N'C:\DemoProject\Persdata2.ndf',
SIZE = 100 MB, MAXSIZE = 200 MB, FILEGROWTH = 0),
(NAME = N'PersData3', FILENAME = N'C:\DemoProject\PersData3.ndf',
SIZE = 100 MB, MAXSIZE = 200 MB, FILEGROWTH = 0)
LOG ON
(NAME = N'PersLog', FILENAME = N'C:\DemoProject\PersLog.ldf',
SIZE = 25 MB, MAXSIZE = 50 MB, FILEGROWTH = 0)
```

Les Schémas

- Un schéma est un espace de noms pour les objets de la base de données.
- Le nom complet d'un objet est : serveur.DB.schema.objet.
- Les noms des schémas sont indépendants des propriétaires des bases de données.
 - Grande flexibilité
 - Gestion simplifiée des permissions
- Exemple : AdventureWorks
- Le schéma dbo : schéma par défaut pour tous les utilisateurs qui n'ont pas défini explicitement un schéma par défaut.

Syntaxe pour créer un Schéma



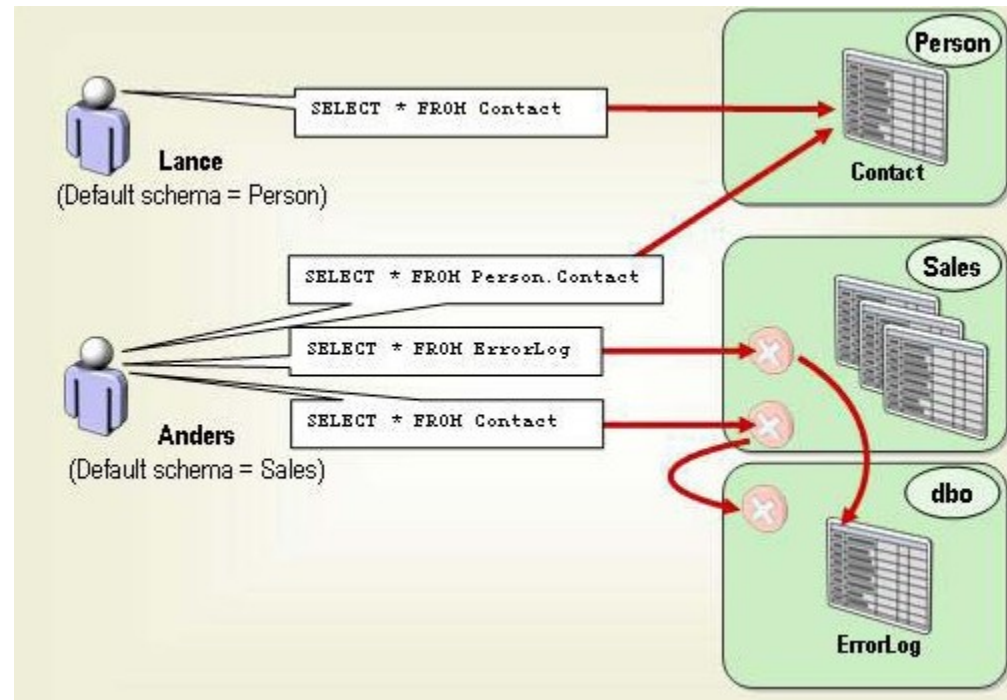
```
CREATE SCHEMA  
schema_name | AUTHORIZATION owner_name | schema_name  
AUTHORIZATION  
owner_name  
[ table_definition | view_definition | grant_statement |  
revoke_statement | deny_statement ]
```

```
ALTER SCHEMA schema_name TRANSFER securable_name
```

```
DROP SCHEMA schema_name
```

« Name Resolution »

- Si l'utilisateur a un schéma par défaut, SQL Server essaie de trouver l'objet dans ce schéma.
- Si l'objet n'est pas trouvé, ou si l'utilisateur n'a pas de schéma par défaut, SQL Server essaie de le trouver dans le schéma dbo.



« Name Resolution »

- On peut assigner un schéma par défaut à un utilisateur en utilisant l'instruction suivante :

```
ALTER USER user_name  
WITH DEFAULT_SCHEMA = schema_name
```

Labo

- Exercice
 - Créer une base de données SQLServerDBTest avec
 - Un fichier primaire : SQLServerDBTest_Data
 - Des fichiers secondaires :
 - Archive1 dans le File group en lecture seule,
 - Archive2,
 - un fichier de log :
 - SQLServerDBTest_Log1
 - Créer trois tables :
 - UserInfo, (id int , nom NVARCHAR(25)) FileGroup Primaire
 - SystemConfig, (id int , config NVARCHAR(25)) FileGroup Primaire
 - OldOrders (id int , nom NVARCHAR(25), Date Datetime2) FileGroup Archive1
 - Créer deux schémas :
 - Users avec comme membre UserInfo
 - Systems avec comme membre SystemConfig

Labo

- Exercice (Suite)

- Utiliser `databaseproperty` pour connaître le type d'accès user (UserAccess).
- Utiliser `sp_helpdb` pour connaître les détails sur la base de données créée.
- Utiliser `sp_spaceused` pour connaître les détails sur l'espace de stockage de utilisé par la DB et par une des tables de la DB

Gestion d'une base de données

- Options d'une base de données
- Sources des métadonnées d'une base de données

Options d'une base de données

- Les options des bases de données peuvent être modifiées via:
 - SQL Server Management Studio
 - L'instruction ALTER DATABASE

Options d'une base de données

Catégorie	Option	Description
Automatique	AUTO_CREATE_STATISTICS	Crée automatiquement toutes les statistiques nécessaires pour optimiser la requête (ON)
	AUTO_UPDATE_STATISTICS	modifie automatiquement toutes les anciennes statistiques nécessaires pour optimiser la requête (ON)
	AUTO_CLOSE	Ferme automatiquement la base de données quand le dernier utilisateur la quitte (OFF pour toutes les versions sauf la Express)
	AUTO_SHRINK	Rétrécissement périodique des fichiers de la base de données (OFF)

Options d'une base de données

Catégorie	Option	Description
Disponibilité	OFFLINE ONLINE EMERGENCY	Contrôle si la base de donnée est online ou offline. EMERGENCY stoppe la connexion de tous les administrateurs non système et met la base de données en mode lecture seulement. (ONLINE)
	READ_ONLY READ_WRITE	Contrôle si la base de données peut modifier les données. (READ_WRITE)
	SINGLE_USER RESTRICTED_USER MULTI_USER	Contrôle quels utilisateurs peuvent accéder à la base de données. (MULTI_USER)

Options d'une base de données

Catégorie	Option	Description
Recovery	RECOVERY	FULL : procède à une complète récupération du média en panne. BULK_LOGGED : utilise moins d'espaces pour les logs car un log utilise un minimum de données. SIMPLE : récupère la base de données seulement depuis le dernier backup complet ou référentiel. (FULL)
	PAGE_VERIFY	Permet à SQL Server de détecter les opérations d'I/O incomplètes. CHECKSUM TORN_PAGE_DETECTION (CHECKSUM)

Sources des métadonnées d'une base de données

- Les métadonnées sont accessibles via:
 - SQL Server Management Studio
 - Les vues
 - Les fonctions
 - Les procédures stockées

Sources des métadonnées d'une base de données

- SQL Server Management Studio

Outils	Description
Explorateur d'objets	Outil graphique pour localiser et gérer les serveurs, bases de données et leurs objets
Fenêtre de propriétés	Chaque objet dans l'explorateur d'objets a une fenêtre de propriétés qui lui est associée. Son contenu varie en fonction de l'objet sélectionné.
Rapports	Rapports pour différents objets dans l'explorateur d'objets.

Sources des métadonnées d'une base de données

- Les vues

Catégorie	Vue	Description
Base de données de fichiers	sys.databases	Retourne une ligne pour chaque base de données dans le serveur
	sys.database_files	Retourne une ligne pour chaque fichier dans la base de données
Objets	sys.columns	Retourne une ligne pour chaque colonne d'un objet contenant des colonnes
	sys.events	Retourne une ligne pour chaque événement pour lequel un trigger ou une notification se déclenchera
	sys.indexes	Retourne une ligne pour chaque index
	sys.tables	Retourne une ligne pour chaque table dans la base de données
	sys.views	Retourne une ligne pour chaque vue dans la base de données.
Schémas	sys.schemas	Retourne une ligne pour chaque schéma défini dans la base de données.

Sources des métadonnées d'une base de données

- Les vues

Catégorie	Vue	Description
Sécurité	Sys.database_permissions	Retourne une ligne pour chaque permission définie dans la base de données
	Sys.database_principals	Retourne une ligne pour chaque « principal » dans la base de données
	Sys.database_role_members	Retourne une ligne pour chaque membre de chaque rôle dans la base de données.

Sources des métadonnées d'une base de données

- Les fonctions

Fonction	Description
DB_ID	Retourne le nombre d'identifications de la base de données correspondant au nom de la base de données spécifié ou celui de la base de données courante si aucun nom n'est spécifié
DB_NAME	Retourne le nom de la base de données correspondant à l'ID spécifié, ou celui de la base de données courante si aucun ID n'est spécifié
FILE_ID	Retourne l'ID correspondant au fichier logique spécifié dans la base de données courante
FILE_NAME	Retourne le nom logique correspondant à l'ID spécifié du fichier logique dans la base de données courante
FILEGROUP_ID	Retourne l'ID correspondant au filegroup spécifié
FILEGROUP_NAME	Retourne le nom correspondant à l'ID spécifié du filegroup

Sources des métadonnées d'une base de données

- Les procédures

Procédures	Description
sp_databases	Liste les bases de données disponibles dans l'instance de SQL Server
sp_stored_procedures	Liste les procédures stockées présentes dans la base de données
sp_help	Donne plus d'informations sur un objet donné de la base de données.
Sp_helpfilegroup	Donne plus d'informations sur un filegroup donné.

Labo :

- Voir le rapport sur le serveur
- Voir le rapport sur la base de données
- Utiliser les procédures stockées systèmes
- Créer un script capable de récupérer les colonnes d'une table donnée.

Chapitre 3

Gestion de la sécurité

SQL Server : Admin.

Objectifs

- Connaître le modèle de sécurité de SQL Server
- Se familiariser avec la gestion de sécurité

Plan

- Introduction à la sécurité SQL Server
- Sécuriser le scope serveur
- Sécuriser le scope base de données
- Gérer les clés et les certificats

Introduction à la sécurité SQL Server

- Les « principals »
 - Les entités de sécurité
- Les « securables »
 - Les sécurisables
- Les permissions
 - GRANT/REVOKE/DENY

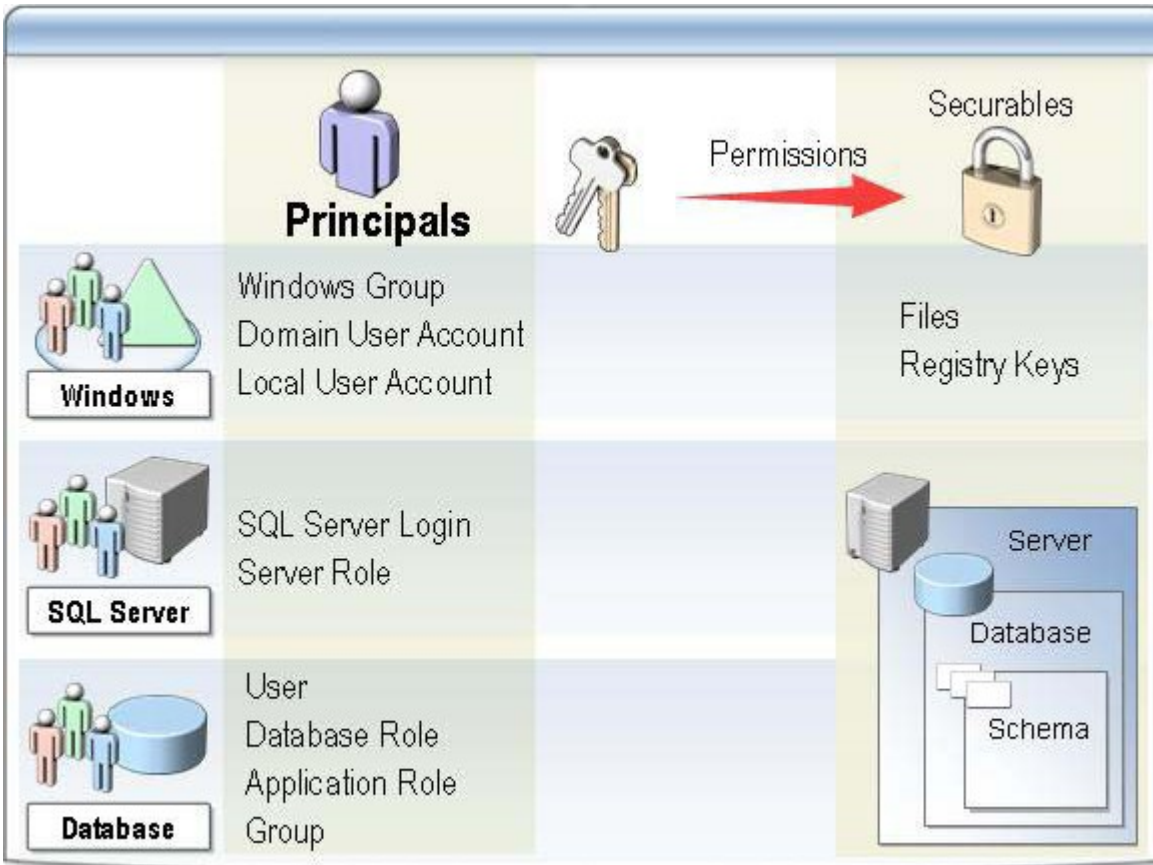
Les « principaux »



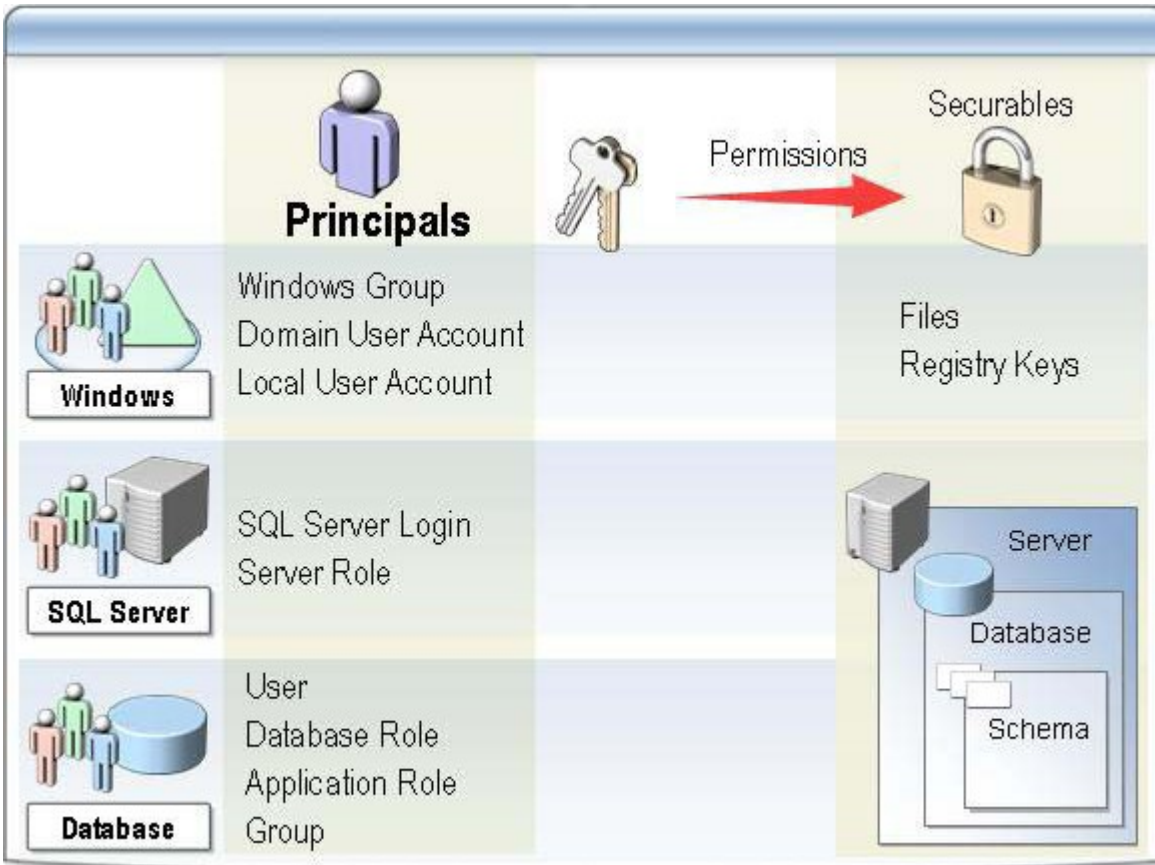
- Identité authentifié qui peut recevoir la permission d'accéder à un objet dans le système de base de données.
- Distinction entre « principaux » indivisibles et collection de « principaux »

Les « securables »

- Objets auxquels le système d'autorisation de SQL Server régule l'accès.
- Notion de scope

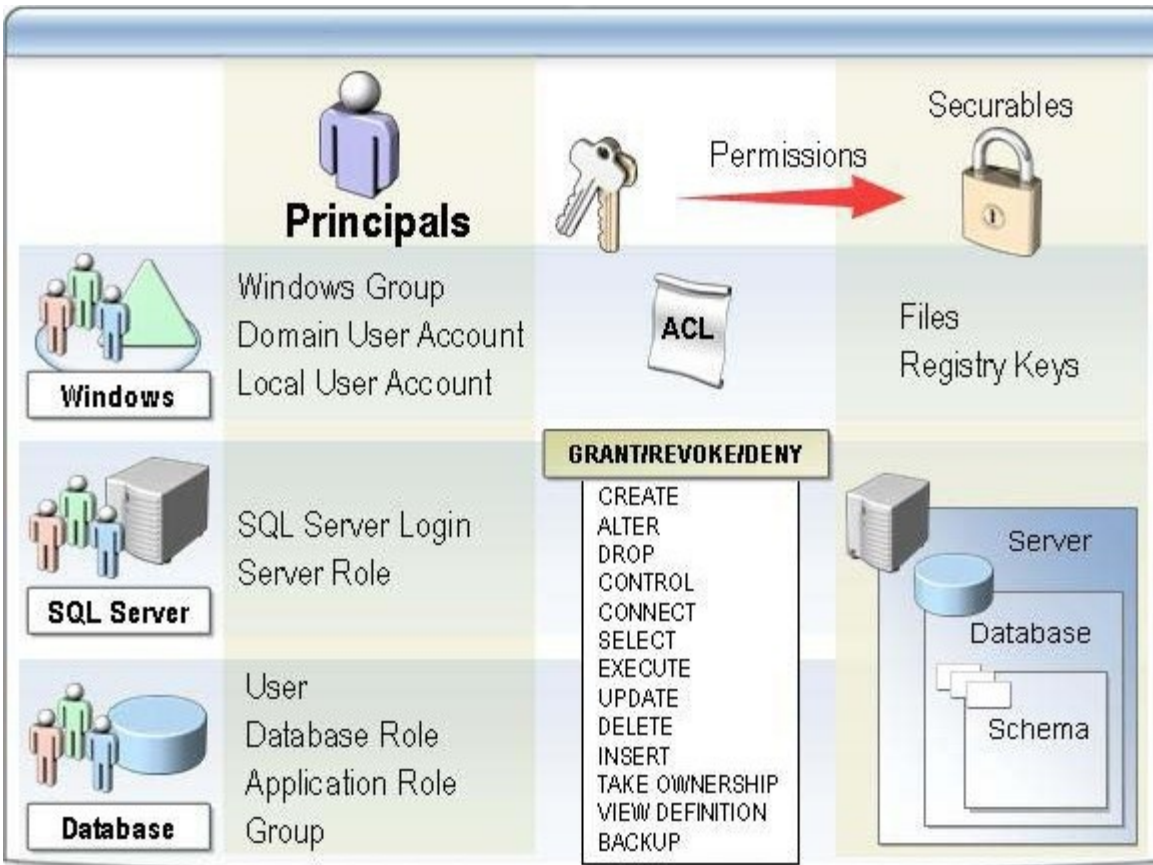


Les « securables »



Scope	« securables »
Serveur	<ul style="list-style-type: none"> - Logins - Endpoints - Databases
Base de données	<ul style="list-style-type: none"> - Users - Roles - Application roles - Certificates - Symmetric keys - Asymmetric keys - Assemblies - Full-text catalogs - DDL events - Schemas
Schéma	<ul style="list-style-type: none"> - Tables - Views - Functions - Procedures - Types - Synonyms - Aggregates

Les permissions



- Règles qui spécifient le niveau d'accès qu'ont les « principals » pour accéder aux « securables »

Les permissions

- Les permissions peuvent être gérées via
 - SQL Server Management Studio
 - Les instruction GRANT, REVOKE et DENY
- Permissions héritées
 - Certaines permissions peuvent être héritées d'un « securable » avec scope plus large
- Permissions effectives
 - Un « principal » peut exécuter une action si
 - Une permission a été explicitement accordée à ce « principal » ou à une collection de « principals » à laquelle il appartient
 - Une permission n'a pas été explicitement refusée à ce « principal » ou à une collection de « principals » à laquelle il appartient

Sécuriser le scope serveur

- Modes d'authentification SQL Server
- Les politiques de mot de passe
- Gestion des logins SQL Server
- Les rôles serveur fixes
- La délégation
- Les credentials
- Les permissions serveur

Modes d'authentification SQL Serveur

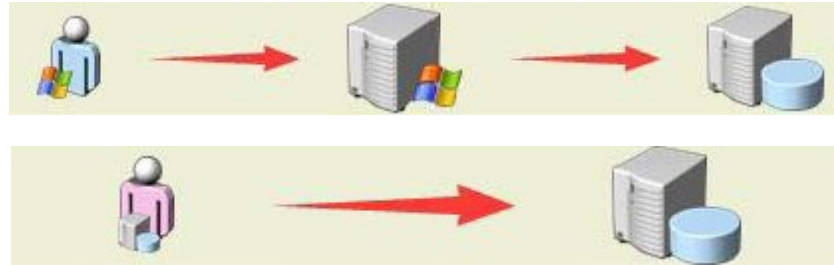
- Mode d'authentification Windows
 - Les utilisateurs sont authentifiés par Windows
 - Ils accèdent à SQL Server via un login mappé à leur compte Windows ou au groupe auquel ils appartiennent.
 - À utiliser dans un environnement réseau dans lequel tous les utilisateurs sont authentifiés via un compte Windows.



SQL Server : Admin.

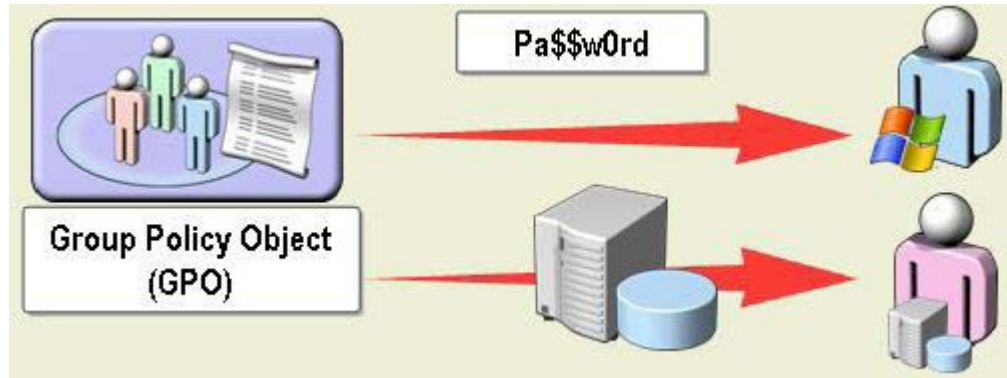
Modes d'authentification SQL Serveur

- Mode d'authentification mixte



- Les utilisateurs, connectés à SQL Serveur via une connexion fiable, accèdent à ce dernier en utilisant l'authentification Windows
- Les utilisateurs peuvent se connecter à SQL Serveur via une connexion non fiable en entrant un login et mot de passe qui sont validés par SQL Serveur indépendamment de Windows.
- À utiliser
Pour permettre aux clients ne fonctionnant pas sous Windows ou à des clients Internet d'accéder à SQL Server.

Les politiques de mot de passe



- Sous Windows Server 2003 et plus, la politique de groupes permet de définir les exigences en matière de mot de passe windows.
- SQL Server applique cette politique à ses SQL logins

Les politiques de mot de passe

- La politique de complexité des mots de passe:
 - Le mot de passe ne doit pas contenir une partie du nom de l'utilisateur du compte.
 - Il doit contenir au minimum 7 caractères
 - Les caractères doit provenir de ces 4 catégories
 - Lettres majuscules
 - Lettres miniscules
 - Chiffres
 - ! \$ # %

Les politiques de mot de passe

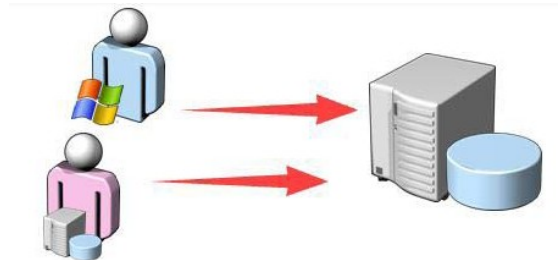
- La politique d'expiration des mots de passe:
 - Quand elle est appliquée, les utilisateurs doivent changer leurs mots de passe.
- Utilisation de cette technique après formation des utilisateurs à l'importance d'un bon mot de passe.

Gestion des logins SQL Server

- On peut gérer les logins via
 - SQL Server Management Studio
 - Les instructions CREATE LOGIN, ALTER LOGIN et DROP LOGIN

Gestion des logins SQL Server

- Création de logins
 - On peut utiliser l'instruction CREATE LOGIN pour créer un login SQL Server ou Windows.



```
CREATE LOGIN login_name  
{ WITH SQL_login_options | FROM WINDOWS [WITH  
windows_login_options ]}
```

```
CREATE LOGIN [SERVERX\SalesDBUsers]  
FROM WINDOWS  
WITH DEFAULT_DATABASE = AdventureWorks
```

```
CREATE LOGIN Alice  
WITH PASSWORD = 'Pa$$w0rd',  
DEFAULT_DATABASE = AdventureWorks
```

SQL Server : Admin.

Gestion des logins SQL Server

- SQL Server active la politique de mot de passe pour les nouveaux logins SQL s'il est installé sur Windows Server 2003 et que la politique est bien activée.
- Ce comportement peut être modifié en utilisant les options de CREATE ou ALTER LOGIN

Option	Description
HASHED	Spécifie que le mot de passe est déjà haché.
MUST_CHANGE	Demande à l'utilisateur de changer le mot de passe lors de sa première connexion.
CHECK_EXPIRATION	Spécifie que la politique d'expiration de mot de passe de WS 2003 doit être appliquée aux logins de SQL Server.
CHECK_POLICY	Met en vigueur la politique de complexité des mots de passe défini pour WS 2003 hébergeant SQL Server

Gestion des logins SQL Server

- Altérer un login
 - On peut utiliser l'instruction ALTER LOGIN pour modifier un login.

```
ALTER LOGIN Alice WITH PASSWORD = 'NewPa$$w0rd' UNLOCK
```

- Supprimer un login
 - On peut utiliser l'instruction DROP LOGIN pour supprimer un login.

```
DROP LOGIN Alice
```

Les rôles serveur fixes

- La notion de rôles permet de grouper des utilisateurs au sein d'une même unité à laquelle seront appliquées des permissions.
- SQL Server possède des rôles serveur prédéfinis couvrant les fonctions administratives les plus communes.
- Les rôles serveur fixes permettent de grouper des privilèges administratifs niveau serveur.

Les rôles serveur fixes

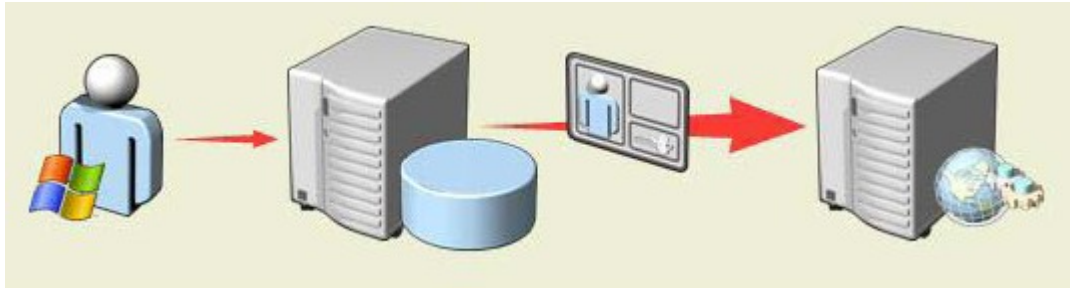


Rôle	Description
sysadmin	Exécute toutes les activités
dbcreator	Crée et altère une base de données
diskadmin	Gère les fichiers
serveradmin	Configure les propriétés serveur
securityadmin	Gère et fait l'audit des logins serveur
processadmin	Gère les processus SQL Server
bulkadmin	Exécute une instruction BULK INSERT
setupadmin	Configure la réplication et les serveurs liés

Les rôles serveur fixes

- Assigner un login à un rôle
 - On peut le faire en utilisant
SQL Server Management Studio
Utiliser la procédure stockée sp_addsrvrolemember
- À savoir
 - On ne peut pas ajouter, modifier ou supprimer un rôle serveur fixe
 - Tout membre d'un rôle serveur fixe peut ajouter un autre login à ce rôle
 - La procédure sp_addsrvrolemember ne peut pas être exécutée à l'intérieur d'une transaction définie par l'utilisateur.
 - On peut utiliser la procédure sp_dropsrvrolemember pour supprimer un membre du rôle

Les credentials



- Objets contenant des informations d'authentification et permettant aux utilisateurs connectés à SQL Server en utilisant le mode d'authentification SQL Server de se connecter à des ressources externes à SQL Server.
- Consistent habituellement en le nom de l'utilisateur Windows et son mot de passe.

Les crédits

- Création des crédits :

- Via

SQL Server Management Studio

Les instruction CREATE CREDENTIAL

- Ex :

```
CREATE CREDENTIAL credential_name WITH IDENTITY = 'identity_name'  
[ , SECRET = 'secret' ]
```

```
CREATE CREDENTIAL AlterEgo WITH IDENTITY = 'ADVENTURE-WORKS\Holly',  
SECRET = 'Pa$$w0rd'  
GO
```

Les permissions serveur

- Permissions serveurs

« Securable »	Permission	Description
Serveur	CONNECT_SQL	Connexion au serveur
	CREATE LOGIN	Créer un login
	ALTER ANY LOGIN	Altérer n'importe quel login dans le scope serveur
	CONTROLSERVER	Contrôle administratif
Login	ALTER	Altérer le login
	IMPERSONATE	Imiter le login
Base de données	CREATE TABLE	Créer des tables dans la base de données
	ALTER ANY USER	Altérer n'importe quel utilisateur de la base de données
	CONTROL	Contrôle de la base de données

Les permissions serveur

- Accorder les permission
 - Se fait via
SQL Server Management Studio
L'instruction GRANT

```
GRANT { securable_permission [ ,...n ] }  
[ON securable_type :: securable_name]  
TO login [ ,...n ]  
[ WITH GRANT OPTION ]  
[ AS { group | role } ]
```

```
USE master  
GRANT ALTER ANY DATABASE  
TO [ADVENTUREWORKS\HOLLY]
```

```
GRANT ALTER  
ON LOGIN :: AWWebApp  
TO [ADVENTUREWORKS\Holly]
```

Labo

- Gérer les logins
 - Vérifier le mode d'authentification
 - Créer un nouveau login avec une politique d'expiration de mots de passe et une politique de complexité de mots de passe.
 - Donner à ce login de droit d'altérer n'importe quel base de données.

SQL Server : Admin.

Sécuriser le scope base de données

- Gérer les utilisateurs
- Utilisateurs spéciaux
- Rôles base de données
- Rôles application
- Permissions base de données
- Permissions schéma

Gérer les utilisateurs

- Pour accéder aux bases de données, il faut créer des utilisateurs dans ces bases de données.
- On peut mapper des logins individuels ou de groupes Windows aux mêmes utilisateurs.
- Création, modification et suppression d'un utilisateur:
 - se fait via
SQL Server Management Studio
Instructions CREATE USER, ALTER USER et DROP USER
- Demo access DB permission

Gérer les utilisateurs

- Exemples :

-- Créer un utilisateur pour un login portant le même nom

```
CREATE USER Bernhard
```

-- créer un utilisateur avec un nom différent du login qu'on veut mapper

```
CREATE USER Jonas FOR LOGIN [ADVENTUREWORKS\Jonas]
```

-- créer un utilisateur avec un schéma par défaut

```
CREATE USER SalesUser FOR LOGIN [DBSERVER1\SalesUsers]  
WITH DEFAULT_SCHEMA = 'Sales'
```

Utilisateurs spéciaux

- Dbo
 - Existe par défaut dans toutes les bases de données
 - Les membres du rôle « **sysadmin** » et le login « **sa** » sont mappés à dbo
 - tout objet créé par l'administrateur système appartient automatiquement à dbo
 - Ne peut être supprimé

Utilisateurs spéciaux

- Guest

- Existe par défaut dans toutes les bases de données.
- Il permet aux logins sans compte utilisateur de se connecter à la base de données.
- Un login peut assumer l'identité Guest quand les conditions suivantes sont vérifiées :

Le login a accès à SQL Server mais pas à la base de données

La base de données contient un compte utilisateur Guest

Le compte Guest est activé en utilisant l'instruction suivante :

- Demo access_db_permission

```
USE database_name  
GRANT CONNECT TO guest;  
GO
```

Rôles base de données

- Rôles base de données fixes
 - Groupent les privilèges administratifs relatifs à la base de données pour effectuer toutes les tâches communes.

Rôle	Description
db_accessadmin	Ajoute et supprime des utilisateurs, des groupes et des rôles
db_backupoperator	Backup de la base de données
db_datareader	Lit les données de n'importe quelle table
db_datawriter	Ajoute, change et supprime les données de n'importe quelle table
db_ddladmin	Ajoute, modifie et supprime les objets de la base de données
db_denydatareader	Ne peut lire les données d'aucune table
db_denydatawriter	Ne peut écrire des données dans aucune table
db_owner	Exécute toutes les activités possibles sur toutes les bases de données
db_securityadmin	Change les rôles base de données et applications, et crée des schémas
public	Maintient les permissions par défaut

Rôles base de données

- Rôles définis par l'utilisateur
 - Ça permet de créer des groupes d'utilisateurs avec les mêmes permissions.
 - Se créent et gèrent via
 - SQL Server Management Studio
 - Les instructions CREATE ROLE, ALTER ROLE et DROP ROLE

```
USE AdventureWorks  
CREATE ROLE auditors
```

Rôles base de données

- Assigner un rôle à un « principal »
 - Se fait via
 - SQL Server Management Studio
 - La procédure stockée `sp_addrolemember`
 - Attention: on ne peut pas créer des rôles récursifs
 - Ex: Le rôle SalesManager ne peut pas être membre du rôle SeniorEmployee si SeniorEmployee est membre de SalesManager
 - Pour supprimer un membre : `sp_droprolemember`

Rôles application

- Permettent d'améliorer la sécurité pour une application particulière
- Contexte de sécurité alternatif pour l'utilisateur pour accéder aux bases de données.

Rôles application

- Caractéristiques
 - Ils n'ont pas de membres
 - Permettent aux utilisateurs d'avoir des permissions spéciales quand ils utilisent l'application.
 - Requiert un mot de passe pour être activé
 - Quand ils sont activés, les utilisateurs
 - perdent leurs permissions existantes pour la base de données
 - Héritent de toutes les permissions accordées au rôle.

Rôles application

- Création d'un rôle application
 - Se fait via
SQL Server Management Studio
L'instruction CREATE APPLICATION ROLE

```
CREATE APPLICATION ROLE weekly_receipts  
WITH PASSWORD = '987Gbv876sPYY5m23'  
GO
```

- Activation d'un rôle application
 - Se fait via
SQL Server Management Studio
La procédure stockée sp_setapprole

```
EXEC sp_setapprole 'weekly_receipts', '987Gbv876sPYY5m23'  
GO
```

Permissions base de données

- Types de permissions base de données

« Securable »	Permission	Description
USER	ALTER	Altère l'utilisateur spécifié
SCHEMA	SELECT	Sélectionne des lignes de n'importe quel objet du schéma
	Alter	Altère n'importe quel objet du schéma
	TAKE OWNERSHIP	Devenir propriétaire du schéma

Permissions base de données

- Accorder des permissions base de données
 - Se fait via
 - SQL Server Management Studio
 - L'instruction GRANT

```
GRANT { database_permission [ ,...n ] }  
TO security_account [ ,...n ]  
[ WITH GRANT OPTION ]  
[ AS { group | role } ]
```

```
USE AdventureWorks  
GRANT ALTER ANY USER  
TO HRManager
```

Permissions base de données

- Accorder des permissions base de données
 - Se fait via
 - SQL Server Management Studio
 - L'instruction GRANT

```
GRANT {securable_permission [ ,...n ] }  
ON securable_type :: <securable_name  
TO security_account [ ,...n ]  
[ WITH GRANT OPTION ]  
[ AS { group | role } ]
```

```
USE AdventureWorks  
GRANT SELECT  
ON SCHEMA :: sales  
TO SalesUser
```

Permissions schéma

- Types de permissions

« Securable »	Permission	Description
TABLE	SELECT	Sélectionner des lignes de la table
	ALTER	Altère la table
	CONTROLE	Contrôle complet sur la table

Permissions schéma

- Accorder des permissions Schéma
 - Se fait via
 - SQL Server Management Studio
 - L'instruction GRANT

```
GRANT {securable_permission [ ,...n ] }  
ON securable_type :: <securable_name  
TO security_account [ ,...n ]  
[ WITH GRANT OPTION ]  
[ AS { group | role } ]
```

```
USE AdventureWorks  
GRANT SELECT  
ON sales.orders  
TO SalesUser
```

Labo

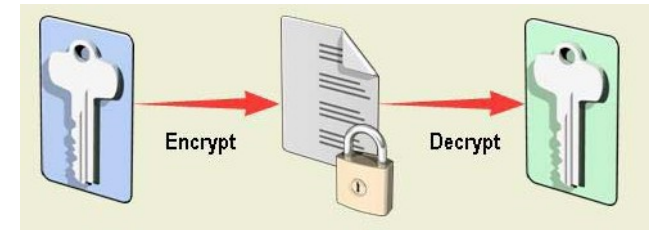
- Créer un login mappé à un groupe d'utilisateurs
- Créer un utilisateur et assigner le groupe à cet utilisateur
- Donner à cet utilisateur le droit de créer et sélectionner des tables et des vues d'une base de données.
- Refuser à cet utilisateur le droit d'altérer, de supprimer ou de sélectionner dans un schéma donné.

Gérer les clés et les certificats

- Les clés
- Les certificats
- SQL Server Cryptography Architecture
- Quand utiliser les clés et les certificats

Les clés

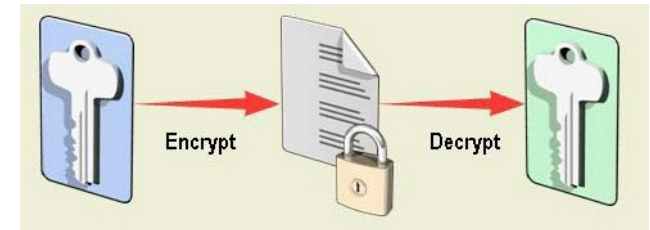
- La clé est la valeur qu'on applique à une fonction de cryptographie pour crypter ou décrypter des données sécurisées.
- Les clés symétriques
 - La clé est utilisée à la fois pour crypter et décrypter.
 - La clé doit être partagée entre le système de cryptage et de décryptage
 - La clé peut être créée via l'instruction CREATE SYMMETRIC KEY.



```
CREATE SYMMETRIC KEY SymKey  
WITH ALGORITHM = AES_256  
ENCRYPTION BY PASSWORD = 'j7%ga5c$b+  
£hdb05'
```

Les clés

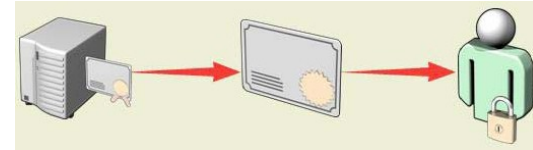
- Les clés asymétriques
 - Consistent en une paire de valeurs qui peuvent être utilisées dans les fonctions à sens unique.
 - Consistent en une clé publique, partagée par tout le monde, et une clé privée, qui doit uniquement être gardée par le propriétaire de la clé.
 - Peut être créée via l'instruction CREATE ASYMMETRIC KEY.



```
CREATE ASYMMETRIC KEY AsymKey  
WITH ALGORITHM = RSA_2048  
ENCRYPTION BY PASSWORD =  
'bmsA$dk7i82bv55'
```

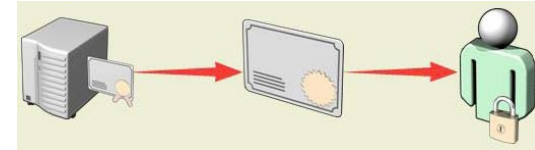
Les certificats

- Un certificat permet d'associer une clé publique avec une identité d'une personne ou d'un système qui détient la clé privée correspondante.
- Ils sont distribués par des autorités d'authentification (CA)
- Ils permettent d'authentifier un grand nombre d'utilisateurs sans devoir maintenir un mot de passe pour chaque utilisateur.



Les certificats

- Contient
 - Une clé publique de la personne ou du système
 - Les informations d'identification du sujet (nom, email)
 - La période de validité
 - Signature digitale
- Les certificats SQL Server sont conformes au standard X.509

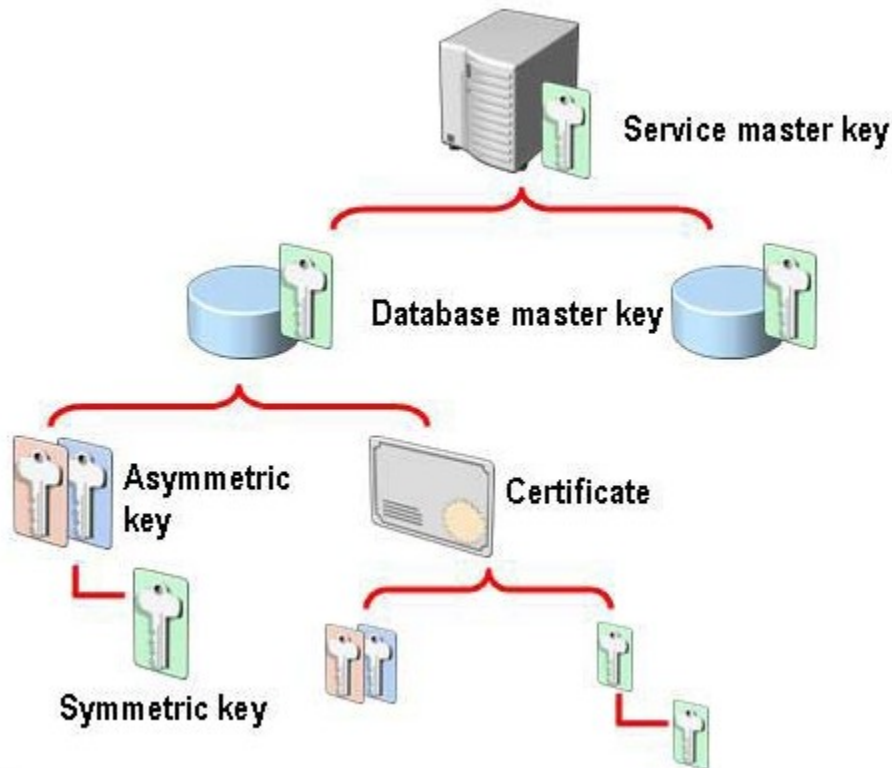


```
-- create a new certificate
CREATE CERTIFICATE AWCustRelationsCert
ENCRYPTION BY PASSWORD =
'pGFD4bb925DGvbd2439587y'
WITH SUBJECT = 'Adventure Works
Customer Relations',
EXPIRY_DATE = '10/31/2009'
```

```
-- Export the certificate
BACKUP CERTIFICATE AWCustRelationsCert
TO FILE = 'c:\certs\AWCustRelationsCert.cer'
```

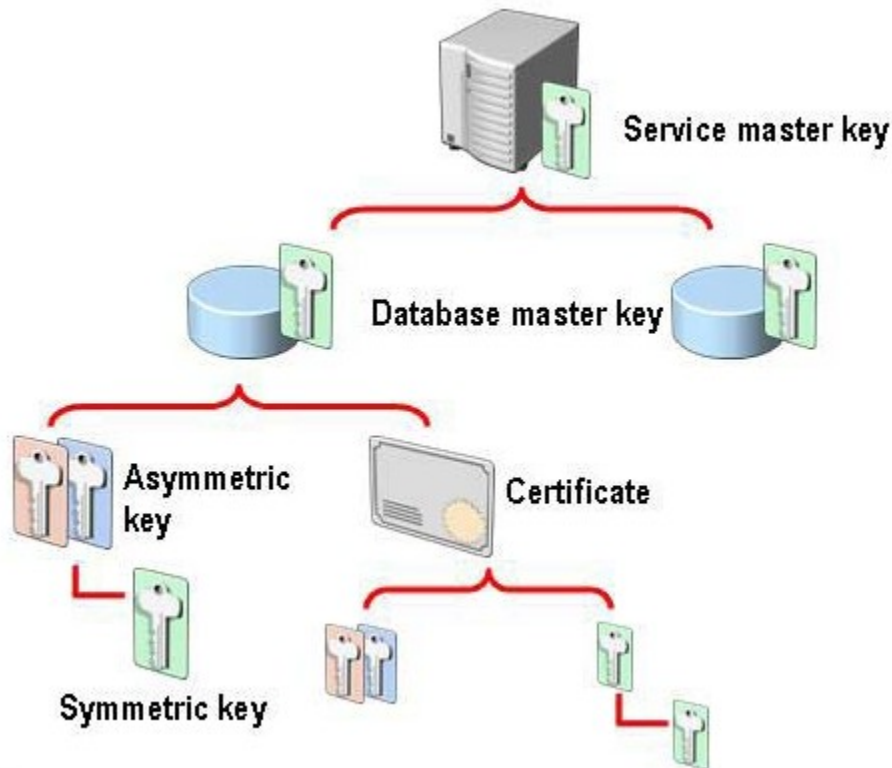
```
-- Import a certificate
CREATE CERTIFICATE SupplierCert
FROM FILE = 'c:\certs\SupplierCert.cer'
```

SQL Server Cryptography Architecture



- **Server Master Key:**
 - C'est la racine de la hiérarchie de cryptage
 - Cette clé est générée automatiquement quand on installe une instance de SQL Server
 - Elle est sécurisée par Windows Data Protection API (DPAPI)

SQL Server Cryptography Architecture



- Database master key :
 - clé symétrique qui peut être utilisée pour protéger la clé privée des certificats et les clés asymétriques dans la DB.
 - Quand cette clé est créée, elle est cryptée en utilisant un triple DES et un password donné par le user.
 - Une copie de la clé est cryptée en utilisant la Service Master Key et stockée dans la base de données elle-même et dans la Db master.

```
USE AdventureWorks
CREATE MASTER KEY ENCRYPTION BY PASSWORD =
'23987hxJ#KL95234nl0zBe'
```

Quand utiliser les clés et les certificats

- Cryptage des données

Fonction	Description
EncryptByKey et DecryptByKey	Cryptage avec une clé symétrique. Requiert l'utilisation de la fonction KeyGUID qui permet d'identifier la clé symétrique
EncryptByAsymKey et DecryptByAsymKey	Cryptage avec une clé asymétrique. Requiert l'utilisation de la fonction AsymKey_ID qui permet d'identifier la clé asymétrique
EncryptByCert et DecryptByCert	Cryptage avec un certificat. Requiert l'utilisation de la fonction Cert_ID qui permet d'identifier le certificat

Quand utiliser les clés et les certificats

- Signer les modules de code
 - On peut utiliser un certificat pour signer un module de code tels que les procédures stockées, les fonctions ou les triggers.
 - Seuls les utilisateur possédant la clé privée peuvent modifier le code.

ADD SIGNATURE TO Sales.AddOrder BY CERTIFICATE SalesDevCert

Labo

- Crypter des données avec une clé symétrique.
- Crypter des données avec une clé asymétrique.
- Crypter une clé symétrique via un certificat.

Labo

- Créer une base de données hospital
- Crypter deux logins doc1 et doc2 avec le mot de pass : ' P@ssw0rd'.
- Créer 2 utilisateurs(doc1 et doc2) sur la DB hospital associés aux logins.
- Créer une table Patient avec
 - Un id , un nom , un Docname, un SSN en varbinary(5000) et un Problem en varbinary(5000)
- Donner la possibilité de selection et insertion aux deux users sur la table
- Créer une clé assymétrique avec le mdp :P@ssw0rd pour chacun des utilisateurs avec comme algorithme RSA_2048
- « Rajouter
CREATE ASYMMETRIC KEY Doc1Key
AUTHORIZATION [user]
- ...»
- Tester l'encryption de données avec les deux utilisateurs
- Se connecter en tant qu'un autre user : EXECUTE AS LOGIN = 'Doc1';
- Pour fermer la connexion : revert ;
- ENCRYPTBYASYMKEY(ASYMKEY_ID('Doc1Key'), '111-11-1111')

Labo

Exercices 2

- Créer une DB 'Db_encryptionTest' avec en fichier primaire 'Db_encryptionTest_data' et un fichier de log 'Db_encryptionTest_log';
- Créer 'Login1' avec le password 'P@ssw0rd' et la DB précédemment créée par défaut;
- Créer 'User1' associé au 'Login1';
- Créer une table 'TAB1' avec deux colonnes (ID (int) et name (varbinary));
- Ajouter les privilèges SELECT et INSERT à l'User1 sur la TAB1;
- Créer une master key avec le password 'P@ssw0rd';
- Créer un 'Certificat1' pour l'User1;
- Créer une clé symétrique 'Key1' pour User1 avec l'algorithme AES_256 et encrypté avec le Certificat1 (+ vérifier si bien créée);
- Ouvrir la clé symétrique;
- Insérer des données cryptée dans la TAB1;
- Vérifier le cryptage;
- Sélectionner les données décryptée;

Chapitre 4

Backup et restauration

SQL Server : Admin.

Objectifs

- Savoir définir une stratégie de restauration pour les bases de données critiques

Plan

- Planification d'une stratégie de Backup
- Backup des bases de données utilisateurs
- Restauration des bases de données utilisateurs
- Récupération des données à partir d'un « Snapshot » de la base de données
- Récupération des bases de données système

Planification d'une stratégie de backup

- Types de backup
- Modèle de récupération
- Stratégie de backup complet
- Stratégie de backup de la base de données et des journaux de transactions
- Stratégie de backup différentiel
- Stratégie de backup des fichiers et des filegroups
- Opérateurs de backup

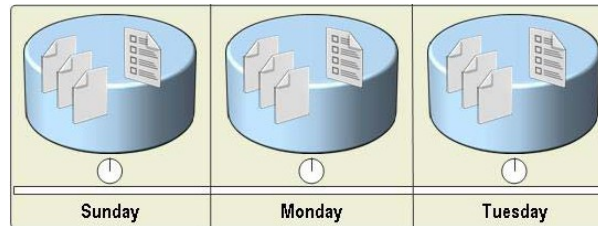
Types de backup

Types de backup	Description
Complet	Tous les fichiers de données et une partie des journaux de transactions
Journal de transactions	Tous les changements de la base de données enregistrés dans les fichiers de log.
Tail-log	La portion active du journal de transactions
Différentiel	Une partie de la base de données qui a changé depuis le dernier backup complet
Fichier ou filegroups	Fichiers et filegroups
Partiel	Le filegroup primaire, tous les filegroups L/E, les filegroups en mode lecture seulement qu'on aura spécifiés
Copie uniquement	La base de données ou le log.

Modèle de récupération

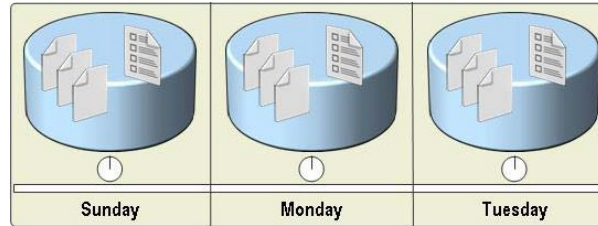
Modèle de récupération	Description
Simple	<ul style="list-style-type: none">-Utilise les copies complètes ou différentielles de la base de données-Restaure les données enregistrées avant la fin du dernier backup.-Tous les changements faits après sont perdus-Simple à implémenter et nécessite moins d'espace de stockage
Complet	<ul style="list-style-type: none">- Utilise la copie de la base de données et toutes les informations de log pour restaurer la base de données-Si les logs ne sont pas endommagés, SQL Server récupère toutes les données sauf les transactions en cours lors de l'erreur.-Nécessite un grand espace de stockage pour les fichiers de log.
Bulk-logged	<ul style="list-style-type: none">- Utilise la copie de la base de données et toutes les informations de log pour restaurer la base de données-Utilise moins d'espace que le modèle de récupération complète pour les opérations suivantes : CREATE INSERT, les opération d'insertion en bloc, SELECT INTO, WRITETEXT et UPDATETEXT.-On stocke uniquement le résultat des opérations, et non leurs détails.

Stratégie de backup complet



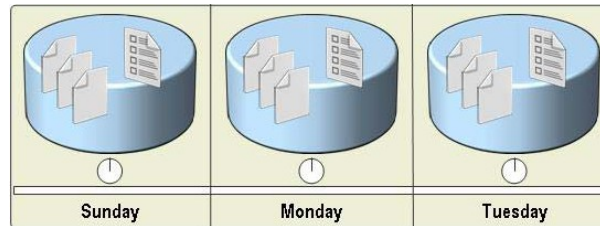
- Méthode de récupération impliquant l'exécution régulière de backup complets.
- La base de données se retrouve le même état dans lequel elle se trouvait avant l'exécution du backup.
- Le temps et les ressources pour exécuter le backup complet dépendent de la taille et de la fréquence des modifications des données de la base de données.

Stratégie de backup complet



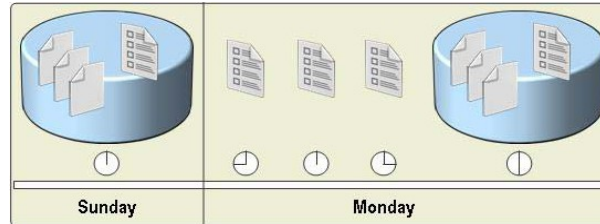
- À utiliser si
 - La base de données est petite
 - Les données de la base de données ne sont pas modifiées fréquemment ou sont en mode de lecture uniquement

Stratégie de backup complet



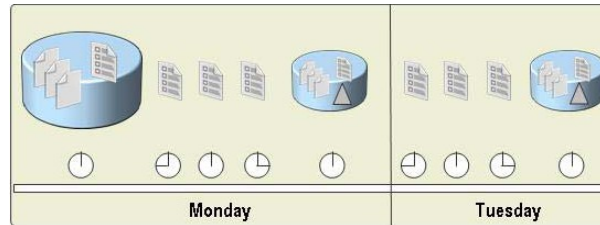
- Gestion du journal de transactions
 - Si on implémente une stratégie de backup complet, et que la base de données est configurée pour utiliser un modèle de récupération bulk-logged ou complet, ces journaux seront remplis.
 - Quand les logs sont remplis, SQL Server empêche certaines activités jusqu'à ce que le journal soit vidé.
 - Pour éviter cette situation
 - Utilise le modèle de récupération simple
 - Vider le journal de transactions périodiquement. (options NO_LOG ou TRUNCTA_ONLY avec l'instruction BACKUP LOG)

Stratégie de backup de la base de données et des journaux de transaction



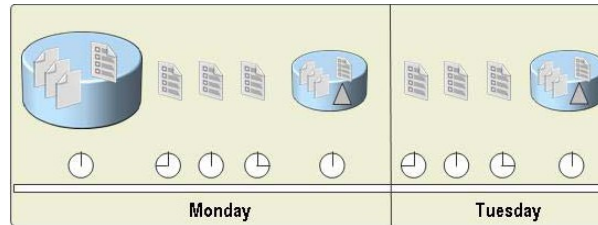
- Méthode de récupération impliquant l'exécution régulière de backup de journaux de transactions afin d'avoir un enregistrement de toutes les activités de la base de données entre deux backups complets .
- À utiliser quand
 - Les données changent fréquemment
 - Les backups complets deviennent trop lent.

Stratégie de backup différentiel



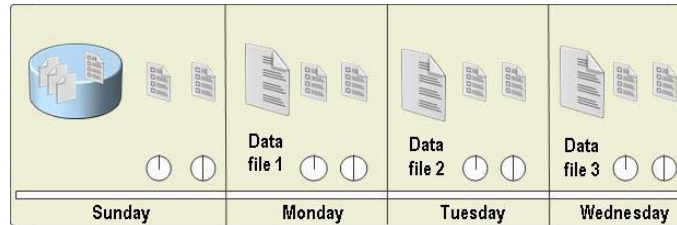
- Méthode de récupération impliquant l'exécution régulière de backup complets avec des backups référentiels
- On peut également exécuter des backup de journaux de transactions entre les deux backups
- Restauration du plus récent backup complet, puis du plus récent backup référentiel, et enfin, tous les backup de journaux de transactions exécutés entre les deux premiers backups

Stratégie de backup différentiel



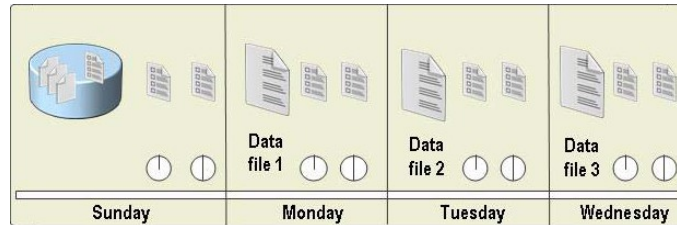
- À utiliser
 - Pour réduire le temps de récupération
 - Quand les données sont modifiées fréquemment

Stratégie de backup des fichiers et des filegroups



- Méthode de récupération impliquant le backup de fichiers et des filegroups.
- On peut également exécuter des backup de journaux de transactions entre les backup de fichiers et des filegroups.
- Stratégie très compliquée à gérer et ne maintient pas automatiquement l'intégrité référentielle.

Stratégie de backup des fichiers et des filegroups



- À utiliser quand
 - La base de données est très large
 - Le backup complet est trop lent.
- Bien faire attention à la consistance des données.

Opérateurs de backup

- Un backup peut être fait via
 - SQL Server Management Studio
 - Des instructions Transact-SQL
- Pour exécuter un backup, on doit avoir l'un des rôles suivants
 - Sysadmin
 - Db_owner
 - Db_backupoperator

Backup des bases de données utilisateurs

- Backup de base de données complet
- Backup des journaux de transactions
- Backup différentiel
- Backup des fichiers et des filegroups
- Options pour s'assurer de l'intégrité des backup

Backup de base de données complet

- Exemples

```
BACKUP DATABASE {database_name | @database_name_var}
TO <backup_device> [, ...n]
[
    WITH
        [FORMAT]
        [
            [,] {INIT | NOINIT}
        ]
]
```

```
USE master
EXEC sp_addumpdevice 'disk', 'AdventureWorksBack',
'D:\MyBackupDir\AdventureWorksBack.bak'
BACKUP DATABASE AdventureWorks TO AdventureWorksBack
```

```
BACKUP DATABASE AdventureWorks TO AdventureWorksBack WITH INIT
```

```
BACKUP DATABASE AdventureWorks TO AdventureWorksBack WITH NOINIT
```

```
BACKUP DATABASE AdventureWorks TO DISK = 'D:\Temp\MyTempBackup.bak'
```

Backup des journaux de transactions

- Exemples

```
BACKUP LOG {database_name | @database_name_var}  
TO <backup_device> [, ...n]  
[  
    WITH  
        [FORMAT]  
        [  
            [,] {INIT | NOINIT}  
        ]  
]
```

```
USE master  
EXEC sp_addumpdevice 'disk', 'AWBackLog',  
    'D:\Backup\ AWBackLog.bak'  
BACKUP LOG AdventureWorks TO AWBackLog
```

- Tail-log

```
BACKUP LOG AdventureWorks  
TO 'C:\Backup\AWTail.bak'  
WITH NORECOVERY, NO_TRUNCATE
```

Backup des journaux de transactions

- Exemples

```
BACKUP DATABASE {database_name | @database_name_var}  
TO <backup_device> [, ...n]  
[  
    WITH  
        [DIFFERENTIAL]]  
]
```

```
BACKUP DATABASE AdventureWorks TO  
DISK = 'D:\MyData\MyDiffBackup.bak'  
WITH DIFFERENTIAL
```

Backup des fichiers et des filegroups

- Exemples

```
BACKUP DATABASE {database_name | @database_name_var}
{FILE = {logical_file_name | @logical_file_name_var}
|
FILEGROUP = {logical_filegroup_name |
@logical_filegroup_name_var}} [, ...n] TO <backup_device> [, ...n]
[WITH DIFFERENTIAL]
```

```
BACKUP DATABASE PhoneOrders
FILE = Orders2 TO OrderBackup2
BACKUP LOG PhoneOrders to OrderBackupLog
```


Backup des fichiers et des filegroups

- Backup partiel
 - Inclure l'option READ_WRITE_FILEGROUPS
 - SQL Server va exécuter le backup du filegroup primaire et de tous les filegroups en mode lecture/écriture.
 - On peut également inclure les fichiers en lecture seule en les spécifiant

```
BACKUP DATABASE {database_name | @database_name_var}
{FILE = {logical_file_name | @logical_file_name_var}
|
FILEGROUP = {logical_filegroup_name |
@logical_filegroup_name_var}} [, ...n] TO <backup_device> [, ...n]
[WITH DIFFERENTIAL]
```

```
BACKUP DATABASE AdventureWorks READ_WRITE_FILEGROUPS TO AWBackup
```

Options pour s'assurer de l'intégrité des backups

- Le « mirroring »
 - SQL Server permet de faire le « mirroring » des médias de backup.
 - Les périphériques de backup et de mirroring doivent être similaires et de mêmes types.

```
BACKUP DATABASE AdventureWorks  
TO BackupOriginal  
MIRROR TO BackupMirror  
WITH  
FORMAT
```

Options pour s'assurer de l'intégrité des backups

- Checksum
 - Mécanisme de détection d'erreurs de backup consistant à créer une somme par l'opération de backup, et à la vérifier par l'opération de récupération.

```
BACKUP DATABASE [AdventureWorks] TO AWBackup WITH CHECKSUM
```

Options pour s'assurer de l'intégrité des backups

- RESTORE VERIFYONLY

- Après le backup, on peut utiliser l'instruction RESTORE VERIFYONLY pour vérifier le backup sans le restaurer.
- Vérifie que l'ensemble du backup est complet et lisible.
- L'instruction effectue les opérations suivantes
 - Vérifie que l'ensemble du backup est complet et
 - Vérifie l'intégrité de certains champs d'en-tête des pages des bases de données
 - Vérifie la validité des checksum
 - Vérifie qu'il y a suffisamment d'espace

```
RESTORE VERIFYONLY FROM AWBackup
```

Restauration des bases de données utilisateurs

- Processus de restauration
- Restauration de la base de données
- Restauration des journaux de transactions
- Restauration des pages, fichiers et filegroups
- La restauration « petit à petit »

Processus de restauration

- La restauration consiste à
 - copier les données du backup

Copier les données, les logs et les pages d'index du périphérique de backup vers les fichiers de la base de données.
 - appliquer les transactions loguées aux données

Appliquer les transactions loguées des données copiées à partir du backup à l'ensemble des données restaurées.
 - faire le roll back des transactions non complétées et rendre la base de données disponible pour les utilisateurs
 - Rendre la base de données online

Processus de restauration

- Les options NORECOVERY et RECOVERY
 - Permettent d'éviter des erreurs d'administration lors du processus de restauration
 - Quand l'option RECOVERY est utilisée, les actions suivantes sont exécutées
SQL Server fait le roll back de toutes les transactions non complètes et ré-exécute toutes les transactions complètes.
La base de données est disponible après que l'exécution complète du processus de récupération.

Processus de restauration

- Les option NORECOVERY et RECOVERY
 - L'option NORECOVERY est utilisée quand on a de multiples backup à restaurer.

Restauration de la base de données

- SQL Server recrée la base de données et tous ses fichiers associés, puis les place dans leurs places d'origine.
- Tous les objets de la base de données sont créés automatiquement.

Restauration de la base de données

- Options de restauration

- RECOVERY

Initie le processus de récupération pour que la base de données retourne à un état consistant.

À utiliser quand on implémente une stratégie de backup complet ou quand on n'exécute aucun backup différentiel ou de journaux de transactions

- NORECOVERY

Retarde le processus de récupération jusqu'à ce que le dernier backup soit restauré

À utiliser quand un backup différentiel ou de journaux de transactions existe

Restauration de la base de données

- Options de restauration

- MOVE TO

Permet de spécifier où restaurer les fichiers si on les restaure dans des emplacements différents.

- REPLACE

Permet de remplacer une base de données existante avec des données provenant du backup d'une base de données différente.

Restauration de la base de données

```
USE master
RESTORE DATABASE AdventureWorks
FROM AWBack
WITH NORECOVERY
RESTORE DATABASE AdventureWorks
FROM AWBackDiff
WITH RECOVERY
```

SQL Server : Admin.

Restauration des journaux de transactions

- Restauration des modifications de la base de données qui ont été enregistrées dans le journal de transactions.

```
USE master
RESTORE DATABASE AdventureWorks
FROM AWBack
WITH NORECOVERY
```

```
USE master
RESTORE LOG AdventureWorks
FROM AWBackLog
WITH FILE = 1,
NORECOVERY
```

```
USE master
RESTORE LOG AdventureWorks
FROM AWBackLog
WITH FILE = 2,
RECOVERY
```

Restauration des pages, fichiers et filegroups

- Permet de réduire le temps de restauration d'une grande base de données
- Récupère les données quand un fichier a été accidentellement supprimé ou endommagé
- Quand on fait une restauration des pages, fichiers ou filegroups, on doit appliquer tous les journaux de transactions qui ont été créés depuis que le backup des fichiers a été créé.

Restauration des pages, fichiers et filegroups

```
USE master
RESTORE DATABASE AdventureWorks
FILE = AW2
FROM AW2Backup
WITH NORECOVERY
```

```
USE master
RESTORE LOG AdventureWorks
FROM AWBackLog
WITH RECOVERY
```

SQL Server : Admin.

La restauration « petit à petit »

- Les bases de données avec plusieurs filegroups peuvent être restaurés en plusieurs étapes.
- Cette restauration fonctionne avec tous les modèles de récupération (plus flexible avec le mode complet et bulk-logged)
- On commence avec le filegroup primaire
- Ensuite les filesgroups secondaires
- Chaque filegroup devient online dès qu'il est restauré
- Les filegroups non restaurés sont marqués « offline »
- Les transactions relatives à un filegroup non encore restauré seront verrouillées jusqu'à ce que la restauration soit complète

La restauration « petit à petit »

```
BACKUP LOG AWData TO tailLogBackup WITH NORECOVERY, NO_TRUNCATE
```

```
RESTORE DATABASE AWData FILEGROUP='Primary' FROM backup1
```

```
WITH PARTIAL, NORECOVERY
```

```
RESTORE DATABASE AWData FILEGROUP='A' FROM backup2
```

```
WITH NORECOVERY
```

```
RESTORE LOG AWData FROM backup3 WITH NORECOVERY
```

```
RESTORE LOG AWData FROM backup4 WITH NORECOVERY
```

```
RESTORE LOG AWData FROM backup5 WITH NORECOVERY
```

```
RESTORE LOG AWData FROM tailLogBackup WITH RECOVERY
```

```
RESTORE DATABASE AWData FILEGROUP='C' FROM backup2a WITH NORECOVERY
```

```
RESTORE LOG AWData FROM backup3 WITH NORECOVERY
```

```
RESTORE LOG AWData FROM backup4 WITH NORECOVERY
```

```
RESTORE LOG AWData FROM backup5 WITH NORECOVERY
```

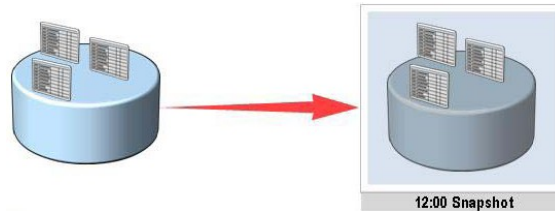
```
RESTORE LOG AWData FROM tailLogBackup WITH RECOVERY
```

```
RESTORE DATABASE AWData FILEGROUP='B' FROM backup2b WITH RECOVERY
```

Récupération des données à partir d'un « Snapshot » de la base de données

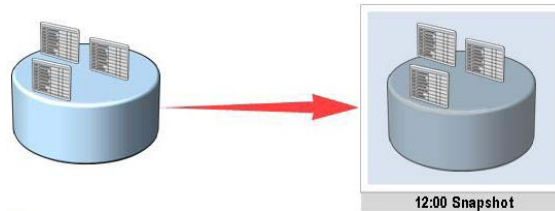
- Qu'est ce qu'un Snapshot?
- Comment ça fonctionne?
- Récupération des données

Qu'est ce qu'un Snapshot?



- Vue statique et en lecture seulement de la base de données à un moment donné.
- Peut être utile en tant que point de restauration rapide
- Ne peut être substitué aux backups car ils ne contiennent pas tous les enregistrements de la base de données.
- Peut être utilisé dans un contexte de développements, tests ou de reporting.

Qu'est ce qu'un Snapshot?



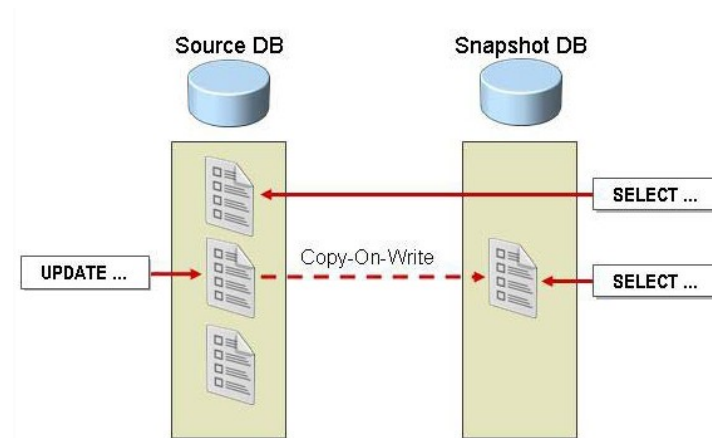
- Restrictions

- Il doit être placé sur le même serveur que la base de données source
- Il ne peut être créé pour les bases de données model, master et tempdb
- Ne peut être restauré ou subir un backup
- Ne peut être attaché ou détaché
- Doit être supprimé avant que la base de données source ne soit supprimée
- Ne peut être créé que via des instructions Transact-SQL

Comment ça fonctionne?

- Les snapshots maintiennent la vue statique de la base de données source en stockant des copies des pré-modifications des données de celle-ci.
- Création
 - Utiliser la clause AS SNAPSHOT OF dans l'instruction CREATE DATABASE

Comment ça fonctionne?



- SQL Server utilise la technologie « copy-on-write » pour implémenter le snapshot.
- Le snapshot est initialement vide.
- Quand une page de la base de données source est modifiée, l'image originale de la page est copiée dans le snapshot.

Récupération des données

- On peut utiliser le snapshot pour corriger une modification accidentelle de la base de données.
- Cependant, ce mécanisme de récupération est très léger et ne doit pas être considéré comme substitut à une vraie stratégie de récupération

Récupération des données

- Restaurer les données à partir d'un snapshot est très rapide et plus simple que l'exécution d'une opération de restauration à partir d'un backup.
- Cependant, ce mécanisme ne fonctionne pas en cas de corruption de fichiers de la base de données.
- On ne peut pas récupérer toutes les données, mais uniquement celles existantes lors de l'exécution du snapshot.

Récupération des données

- Scénarios
 - Suppression de lignes

```
ALTER TABLE Production.WorkOrderRouting  
NOCHECK CONSTRAINT CK_WorkOrderRouting_ActualEndDate  
INSERT INTO Production.WorkOrderRouting  
SELECT *  
FROM  
AdventureWorks_dbsnapshot_1800.Production.WorkOrderRouting  
ALTER TABLE Production.WorkOrderRouting  
CHECK CONSTRAINT CK_WorkOrderRouting_ActualEndDate
```

Récupération des données

- Scénarios
 - Modification accidentelle

```
UPDATE HumanResources.Department
SET Name = (
SELECT Name
FROM
AdventureWorks_dbsnapshot_1800.HumanResources.Department
WHERE DepartmentID = 1)
WHERE DepartmentID = 1
```

Récupération des données

- Scénarios

- Suppression d'objets

- Une table A a été supprimée de la base de données DB_A.
 - 1. Récupérer le script de création de l'objet à partir du snapshot DB_A_snapshot
 - 2. Exécuter le script sur la base de données DB_A
 - 3. Peupler les tables en utilisant le scénario de suppression de lignes

Récupération des données

- Scénarios

- Retourner au snapshot

- Si la base de données source est endommagé ou qu'un grand nombre de données a été accidentellement perdu, on peut restaurer la base de données à partir du snapshot le plus récent.

```
RESTORE DATABASE AdventureWorks  
FROM DATABASE_SNAPSHOT = 'AdventureWorks_dbsnapshot_1800'
```

Récupération des bases de données système

- Backup des bases de données système
- Restauration des bases de données système
- Comment restaurer la base de données master?

Backup des bases de données système

- La base de données master contient des informations sur toutes les bases de données contenues dans SQL Server.
- Il faut faire le backup de cette base de données à chaque fois qu'une base de données est créée ou modifiée.
- Après que la master est reconstruite et restaurée, on peut restaurer d'autres bases de données système.

Backup des bases de données système

- Il faut faire le backup de la base de données master à chaque fois qu'on exécute l'une des instructions suivantes:
 - CREATE DATABASE, ALTER DATABASE et DROP DATABASE
 - Les procédures stockées sp_addserver, sp_dropserver et sp_addlinkedserver
 - La procédure stockée sp_addmessage qui ajoute un message d'erreur à SQL Server

Backup des bases de données système

- La base de données msdb contient des informations sur les jobs, les alertes, et certains opérateurs utilisés par SQL Server Agent.
- Il faut faire le backup de la base de données msdb après qu'elle soit modifiée.

Backup des bases de données système

- Il faut faire le backup de la base de données model, si on la modifie, pour inclure une configuration par défaut pour toute nouvelle base de données utilisateur.

Restauration des bases de données système

- Si les fichiers contenant les bases de données système est endommagé, il faut reconstruire ou restaurer les bases de données système.

Restauration des bases de données système

- Restauration des bases de données système
 - La décision de restaurer ou reconstruire les bases de données système dépend du fait que le service SQL Server est démarré ou pas.
 - Si le service peut démarrer, on peut restaurer les bases de données à partir du backup
 - Si le service ne peut pas démarrer, on doit reconstruire les bases de données système, ajouter les informations qui ont été perdues en attachant les tables existantes et en recréant les objets.

Restauration des bases de données système

- Restauration des bases de données système
 - Si le service ne peut pas démarrer,
On doit reconstruire les bases de données système.
On doit ajouter les informations qui ont été perdues en attachant les tables existantes et en recréant les objets.
Le service SQL Server pourra démarrer
On doit restaurer les bases de données système dans l'ordre suivant
 - Master
 - Msdb
 - model

Restauration des bases de données système

- Restauration des bases de données utilisateur
 - Si la base de données master est restaurée à partir d'un backup, elle contiendra des références vers chaque base de données utilisateur, et il n'y a aucun besoin de les rattacher.
 - Si la base de données master est reconstruite, on doit restaurer les bases de données utilisateur à partir d'un backup, et attacher les fichiers de bases de données à la nouvelle base de données master.
 - Utiliser les procédures stockées suivantes
 - Sp_Attach_db
 - Sp_attach_single_file_db

Comment restaurer la base de données master?

- Si la base de données master et l'instance de SQL Server sont encore accessibles :
 - Démarrer SQL Server en mode single-user
`Sqlservr.exe -c -m`
 - Restaurer la base de données master à partir du backup le plus récent
`RESTORE DATABASE master FROM masterbackup`
 - SQL Server sera automatiquement stoppé
 - Redémarrer le serveur

Comment restaurer la base de données master?

- Si la base de données master est endommagée, l'instance de SQL Server ne peut plus démarrer:
 - Reconstruire la base de données en utilisant le programme setup.exe de SQL Server avec les options suivantes:
 - /qn (pas d'interface utilisateur)
 - REINSTALLMODE = AMUS (reconstruire proprement les DB système)
 - REINSTALL = ALL
 - Restaurer la version originale si un backup existe, sinon, recréer et configurer manuellement le système
 - Restaurer les bases de données msdb et model.

Back-up LABO

- Créer une DB DbBackupTest avec un filegroup archive
- Ajouter deux tables dans la DB : tab1 avec
 - un Id int ,
 - un name nvarchar(50)
 - une description nvarchar(255)
- et une autre table dans le filegroup archive : arch1 avec
 - un Id int ,
 - un name nvarchar(50)
 - une description nvarchar(255)
- insérer quelques lignes dans les deux tables
- effectuer un back-up de la DB
 - avec une vérification du back-up
- Insérer quelques données
- Supprimer la Base de données
- Restaurer la base de données :
 - Restaurer le back-up complet
 - Restaurer le back-up complet + le journal des transactions

Chapitre 5

Automatisation des tâches administratives

SQL Server : Admin.

Objectifs

- Automatiser les tâches administratives
- Minimiser le travail administratif

Plan

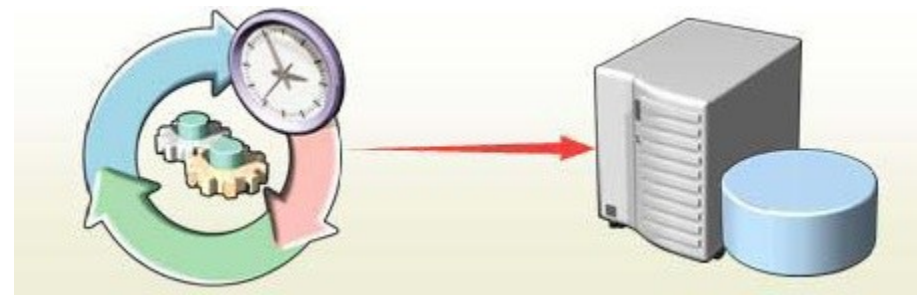
- Automatisation des tâches administratives
- Configuration de SQL Server Agent
- Création de jobs et opérateurs
- Création des alertes
- Gestion de serveurs multiples
- Gestion de la sécurité de SQL Server Agent

Automatisation des tâches administratives

- Pourquoi automatiser?
- Le wizard du plan de maintenance
- SQL Server Agent

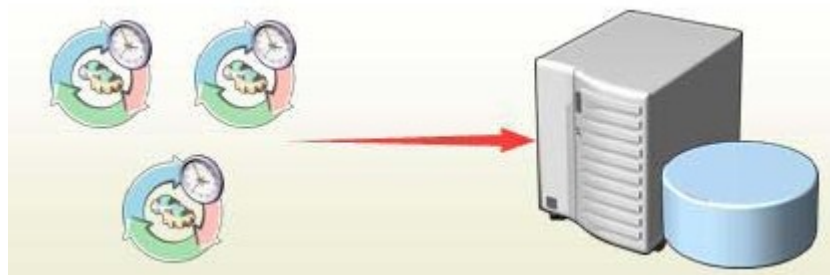
Pourquoi automatiser?

- Réduction de la charge de travail administratif.
- Réduction du risque d'erreur humaine lors de l'exécution de tâches de maintenance de la base de données
- Gestion pro-active de la base de données par des alertes indiquant que des actions doivent être prises afin de se prémunir de situations non désirées

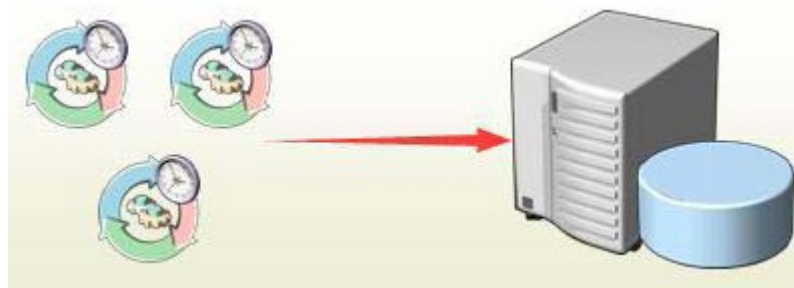


Le wizard du plan de maintenance

- Permet de programmer l'exécution des tâches de maintenance afin de s'assurer que la base de données est régulièrement sauvegardée, fonctionne bien, et est consistante.
- Permet la création d'un ou plusieurs jobs.

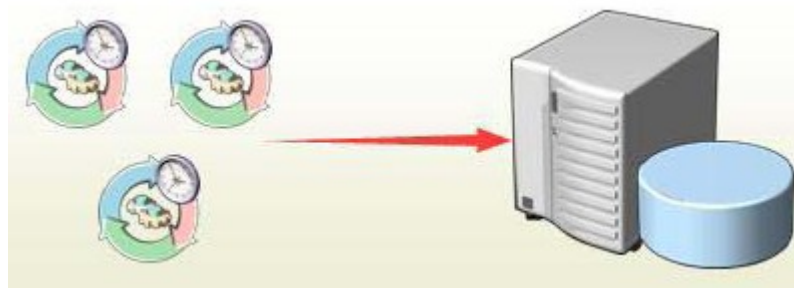


Le wizard du plan de maintenance



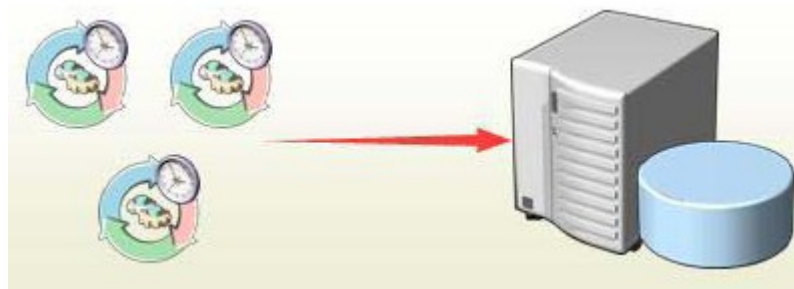
- Tâches qui peuvent être automatisées
 - Backup
 - Exécuter les SQL Server Agent Jobs
 - Compression des fichiers de données en supprimant les pages vides de la base de données
 - Exécuter une vérification interne de la consistance des données et des pages de données de la base de données afin de s'assurer que des problèmes de système et de software n'ont pas endommagé les données.

Le wizard du plan de maintenance



- Tâches qui peuvent être automatisées
 - Réorganiser les données dans les pages d'index et de données en reconstruisant les index.
 - Mise à jour des statistiques pour être sûr que l'optimisateur de requêtes possède bien des données mises à jour.

Le wizard du plan de maintenance



- Les résultats générés par les tâches administratives peuvent être écrits dans un fichier texte ou dans les tables de plan de maintenance `sysmaintplan_log` et `sysmaintplan_log_detail` de la base de données `msdb`.

SQL Server Agent

- C'est le composant de SQL Server qui est responsable de l'automatisation des tâches administratives de SQL Server.
- Il doit fonctionner tout le temps et avoir suffisamment de permissions pour exécuter les jobs et les alertes.
- Permet à l'administrateur de programmer l'exécution de tâches complexes avec plus de flexibilité que le wizard du plan de maintenance.

SQL Server Agent

- SQL Server Agent fonctionne en tant que service de Windows.
- On doit bien spécifier qu'il doit démarrer automatiquement à chaque fois que le système d'exploitation démarre.
- On peut également spécifier qu'il redémarre automatiquement si jamais il stoppe de manière imprévisible.

Configuration de SQL Server Agent

- Configuration de SQL Serve Agent
- Options de SQL Server Agent Mail

SQL Server : Admin.

Configuration de SQL Server Agent

- Rappels
 - Chaque instance de SQL Server a son propre service SQL Server Agent
 - L'instance par défaut s'appelle SQLSERVERAGENT
 - Les autres instances nommées s'appellent SQLAgent\$instance name

Configuration de SQL Server Agent

- Configuration de démarrage du service
 - Peut être configuré pour démarrer manuellement ou automatiquement.
 - Par défaut, il n'est pas configuré à fonctionner automatiquement.
 - On peut modifier cette option via
 - SQL Server Configuration Manager
 - Outils d'administration Windows -> Services
 - Quand il démarre, le service se connecte à SQL Server et exécute la procédure stockée xp_sqlagent_monitor.

Configuration de SQL Server Agent

- Le service account
 - On peut configurer le service à tourner en tant que « compte intégré » ou en tant que « compte utilisateur spécifique ».
 - Comptes intégrés
 - Local system : ce compte est membre du groupe Administrators. (à ne pas utiliser)
 - Local Service
 - Network Service
 - En général, on doit spécifier un compte d'utilisateur local ou un compte d'utilisateur domaine pour ce service, et assigner le minimum de permissions et de droits dont le service a besoin.
 - Le service requiert le droit utilisateur **log on as a service**.

Configuration de SQL Server Agent

- Le service account
 - Si un des droits requis n'est pas donné, seuls les membres du rôle serveur sysadmin peuvent créer un job.
 - Le service account de SQL Server Agent utilise l'authentification Windows pour se connecter à SQL Server, et doit être mappé à un login qui est membre du rôle sysadmin.
 - Dans le cas de la création de jobs multiserveurs, le login doit être mappé à un utilisateur qui est membre du rôle base de données TargetServersRole pour la base de données msdb.

Configuration de SQL Server Agent

- Les dépendances
 - Service SQL Server
 - Autres services Windows dont dépendront les opérations exécutées par SQL Server Agent
 - Ex : Windows Messenger Service pour envoyer des notifications en utilisant la commande **net send**

Options de SQL Server Agent Mail

- Database Mail

- Permet à SQL Server d'envoyer des e-mail via le serveur « Simple Mail Transport Protocol » (SMTP).
- Afin d'utiliser ce mode, il faut
 - Activer la database mail
 - Créer un profil Database Mail qui inclus un compte e-mail que SQL Server Agent peut utiliser. Rendre à ce profil le profil par défaut pour l'utilisateur de la base de donnée msdb qui est mappé au login correspondant au service account du service SQL Server Agent.

Options de SQL Server Agent Mail

- Database Mail

- Afin d'utiliser ce mode, il faut

Ajouter cet utilisateur au rôle de la base de données msdb : DatabaseMailUserRole
Configurer les propriétés du système d'alertes de SQL Server Agent de manière à ce qu'il utilise le Database Mail, et spécifier le profil
Redémarrer le service SQL Server Agent.

Options de SQL Server Agent Mail

- SQL Mail
 - Il s'agit d'une interface de programmation d'application de Messagerie (MAPI) qu'on peut utiliser pour configurer SQL Server pour qu'il envoie et reçoit des e-mails via Microsoft Exchange Server.
 - Afin d'utiliser ce mode, il faut
Configurer le profil MAPI pour ce service account du service SQL Server Agent.
 - Requiert l'installation d'un client e-mail tel que Microsoft office outlook.

Création de jobs et opérateurs

- Les jobs
- Les opérateurs
- Création des étapes d'un job
- Job Activity Monitor

Les jobs

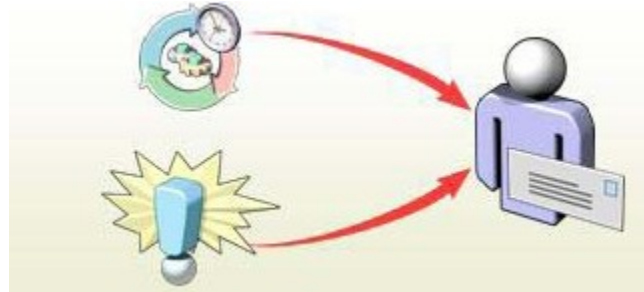
- Séries d'opérations exécutées séquentiellement par SQL Server Agent.
- Ils peuvent exécuter un grand nombre d'activités
 - Scripts Transact-SQL
 - Applications
 - Scripts ActiveX
 - Packages Integration Services
 - Commandes et requêtes Analysis Services
 - Tâches de réplication
- Ils peuvent être exécutés manuellement ou au autom (temps ou alertes)



Les jobs

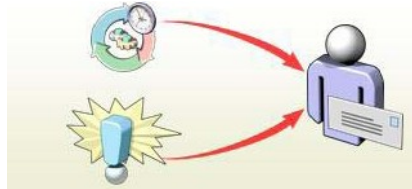
- Création de jobs
 - Se créent via
SQL Server Management Studio
La procédure stockée `sp_add_job`
 - La définition du job est stockée dans la table système `sysjob` de la base de données `msdb`.
- Quand on définit un job, on doit
 - S'assurer que le job est activé
 - Spécifier le propriétaire du job
 - Créer des catégories de jobs pour mieux organiser, filtrer et gérer les jobs

Les opérateurs



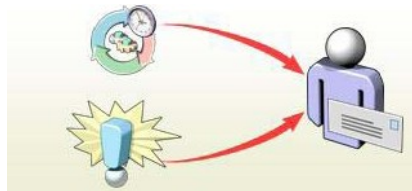
- Les opérateurs sont des alias de personnes ou groupes qui peuvent recevoir des notifications électroniques quand les jobs se sont exécutés, les étapes des jobs ont échoué, ou qu'une alerte a été déclenchée.
- Les notifications se font par pager, e-mails ou commandes net send.

Les opérateurs



- Création d'un opérateur
 - Se fait via
 - SQL Server Management Studio
 - La procédure stockée `sp_add_operator`
 - La définition de l'opérateur est stockée dans la table `sysoperators` dans la base de données `msdb`

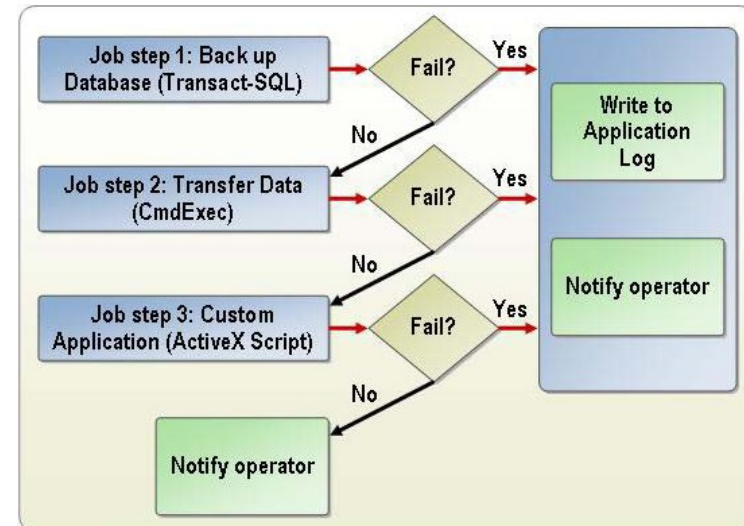
Les opérateurs



- Création d'un opérateur
 - Quand on crée un opérateur, on doit
 - Utiliser un alias pour un groupe d'e-mails pour notifier plus qu'une seule personne
 - Tester chaque méthode de notification utilisée pour notifier un opérateur pour s'assurer que l'opérateur peut recevoir la notification
 - Utiliser la commande net send pour envoyer des messages aux opérateurs faisant partie du réseau.

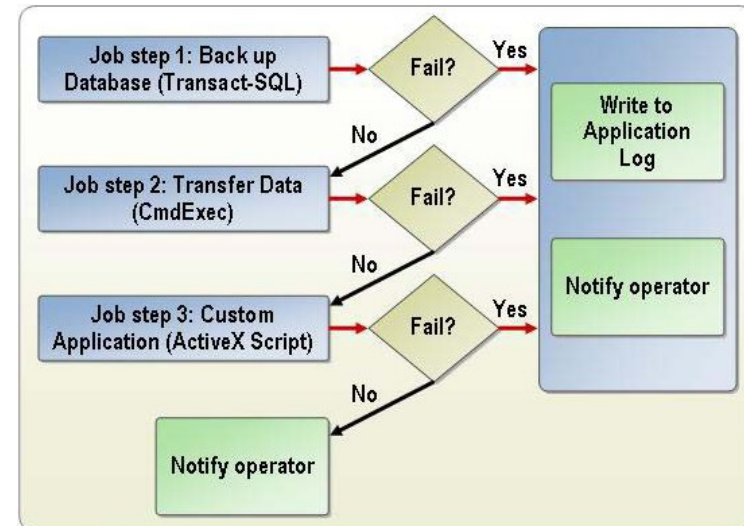
Création des étapes d'un job

- La création d'un job se fait via
 - SQL Server Management Studio
 - La procédure stockée sp_add_jobstep
- La définition des étapes d'un job est stockée dans la table sysjobsteps de la base de données msdb



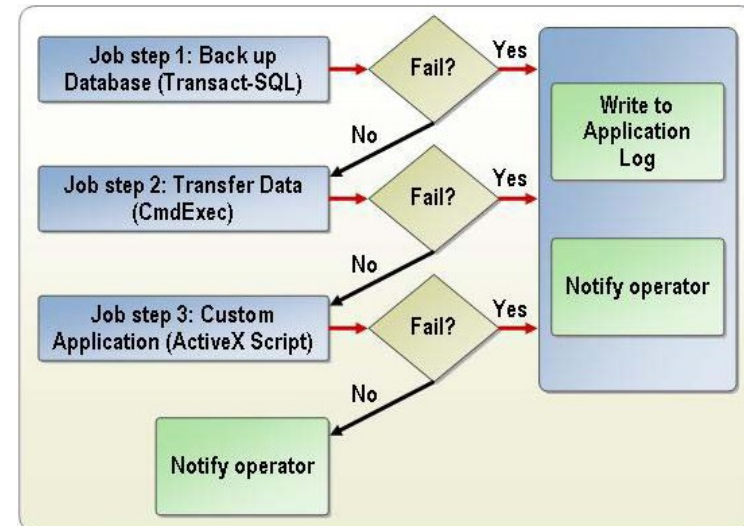
Création des étapes d'un job

- Transact-SQL
 - Identifier la base de données à utiliser
 - Inclure les variables et paramètres requis
 - On peut écrire les résultats d'une étape dans un fichier de sortie.



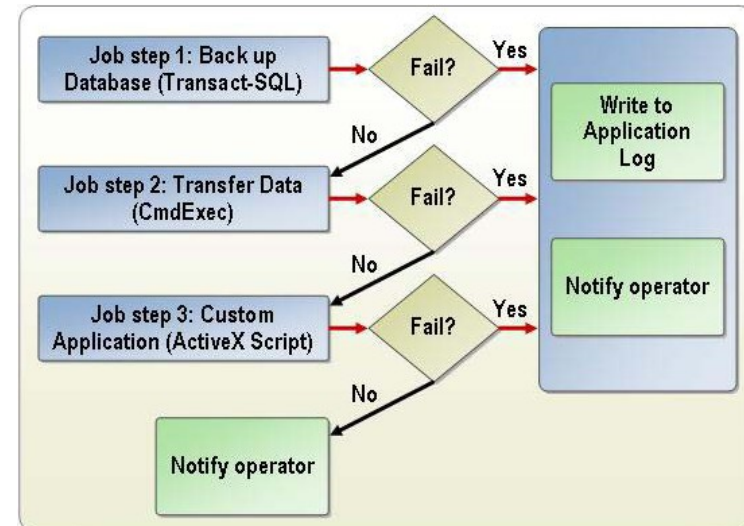
Création des étapes d'un job

- Commande du système d'exploitation
 - Identifiée par les extensions de fichiers .exe, .bat, .cmd ou .com
 - Identifier un code de sortie du processus pour indiquer que la commande a bien été exécutée avec succès
 - Inclure le chemin complet vers l'exécutable.



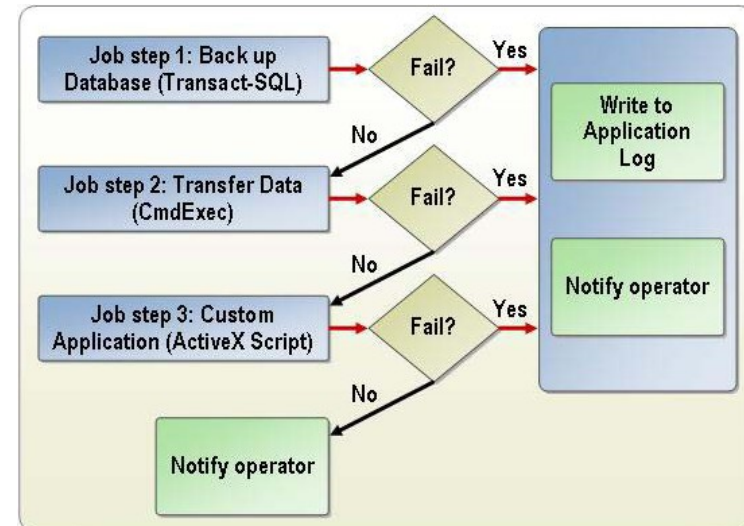
Création des étapes d'un job

- Scripts ActiveX
 - On peut écrire des étapes de job en utilisant ActiveX avec un langage spécifique (MS VB, VBScripts, MS Jscript, ou autres)
 - Identifier le langage des scripts
 - Écrire ou ouvrir le script actif.



Création des étapes d'un job

- La logique de flux
 - Lors de la création d'un job, le DBA doit spécifier l'action que SQL Server doit prendre en cas de succès ou d'échec d'une étape.
 - On peut spécifier le nombre de fois que SQL Server doit réessayer d'exécuter une étape si celle-ci échoue. On peut également spécifier l'intervalle entre deux exécutions.
 - On peut également spécifier si le job peut être supprimé après son exécution.



Job Activity Monitor

- Outil de SQL Server Management Studio permettant de voir le contenu de la table sysobjactivity.
- On peut y voir tous les jobs du serveur. On peut également les filtrer

Job Activity Monitor

- Il permet de
 - Démarrer et stopper un job
 - Voir les propriétés des jobs
 - Voir l'historique d'un job spécifique
 - Rafraichir l'information manuellement ou automatiquement.
- L'activité des jobs est également consultable via la procédure stockée `sp_help_jobactivity`

Création des alertes

- Les alertes
- Création d'une alerte

SQL Server : Admin.

Les alertes

- Les alertes sont des réponses prédéfinies pour des évènements spécifiques.
- Elles permettent d'exécuter des jobs ou de notifier un opérateur quand un évènement particulier intervient ou quand les performances atteignent un niveau critique.

Les alertes

- Les évènements sont générés par SQL Server et enregistrés dans le journal d'applications de Windows.
- SQL Server Agent lit ce journal, et compare les événements qui s'y trouvent aux alertes qu'on a définies. S'il trouve une correspondance, il déclenche l'alerte.

Les alertes

- Les erreurs de SQL Server
 - On peut spécifier un code d'erreur ou un niveau de sévérité à toutes les erreurs de SQL Server
 - L'alerte se déclenchera en fonction de l'un de ces deux indicateurs.
- Conditions de performance
 - Quand une valeur de performance atteint un seuil critique, on déclenche l'alerte.
 - ex : quand le journal de transactions dépasse 75% de sa capacité, on déclenche une alerte qui exécute un backup du journal de transactions et notifie l'administrateur

Création d'une alerte

- Se fait via
 - SQL Server Management Studio
 - La procédure stockée sp_add_alert
- Pour créer une alerte, on doit spécifier
 - Son nom
 - L'évènement ou la condition de performance qui déclenche l'alerte
 - L'action que SQL Server Agent prend en réponse à l'évènement ou à la condition de performance

Création d'une alerte

- Spécification de l'opérateur
 - On doit spécifier les informations du contact
 - Définir comment l'opérateur est notifié

Gestion de serveurs multiples

- Les serveurs master et target
- Jobs multiserveurs
- Event Forwarding

Les serveurs master et target

- Le serveur master
 - Stocke une copie centrale de la définition des jobs qui tournent dans les machines target.
 - Distribue les jobs aux serveurs target et en reçoit des évènements
- Les serveurs target
 - Sont assignés à un serveur master
 - Se connectent périodiquement au serveur master afin de mettre à jour leurs jobs programmés
 - Télécharge tout nouveau job

Jobs multiserveurs



- Pour automatiser des jobs sur des multiples serveurs, on doit définir un serveur master et un ou plusieurs serveurs target.

Jobs multiserveurs



- Définition du serveur master
 - Se fait via
 - Master Server Wizard de SQL Server Management Studio
 - La procédure stockée sp_msx_enlist
 - Le wizard effectue les opérations suivantes
 - Vérifie les propriétés de sécurité du service SQL Server Agent et SQL server dans tous les serveurs qui deviendront Target.
 - Crée un opérateur du serveur master. (MSXOperator)
 - Démarre le service SQL Server Agent du serveur master
 - Enrôle un ou plusieurs serveurs target.

Jobs multiserveurs



- Définition des serveurs target
 - Se fait via
 - Target Server Wizard de SQL Server Management Studio
 - La procédure stockée `sp_msx_enlist`
 - Ils sont assignés uniquement à un seul serveur master
 - Doivent être localisés dans le même domaine Windows que le serveur master.
 - Ne peuvent être membre d'autres serveurs master jusqu'à ce qu'ils se libèrent de leur actuel serveur master.

Jobs multiserveurs



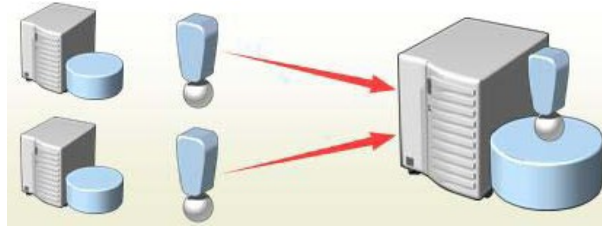
- Implémentation des jobs multiserveur
 - Ils sont créés dans le serveur master
 - 1. Le serveur master met à disposition le job aux serveurs target dans une liste de téléchargement (la table sysdownloadlist de la base de données msdb)
 - 2. Les serveurs target se connectent périodiquement au serveur master pour déterminer s'il y a un nouveau job ou si un job a été mis à jour.
 - 3. Le serveur target charge le résultat du job dans le serveur master quand le job est exécuté.

Jobs multiserveurs



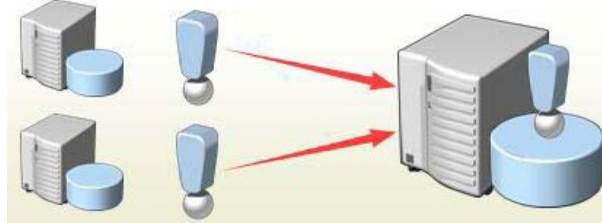
- Modification des définitions des jobs
 - Les définitions des jobs ne peuvent être modifiées dans les serveurs cible. Ils doivent uniquement être modifiés dans le serveur maître
 - SQL Server Enterprise Manager poste automatiquement les instructions nécessaires dans la liste de téléchargement.

Event Forwarding



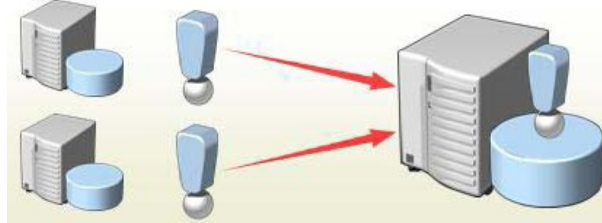
- Il s'agit d'expédier tous les messages des événements à une instance de SQL Server.
- Le serveur qui reçoit toutes les alertes est un serveur dédié qui peut jouer le rôle du serveur master.
- L'Event Forwarding permet de centraliser la gestion des alertes pour un groupe de serveurs et réduire la charge de travail dans le serveur très utilisé.

Event Forwarding



- Avantages
 - Centralisation
 - Les serveurs physiques peuvent être administrés en tant qu'un seul serveur logique.
 - Efficacité
 - Le temps de configuration est réduit.

Event Forwarding



- Inconvénient
 - Augmentation du trafic
 - Si le serveur recevant les messages devient offline, aucune alerte n'exécutera ces actions
 - Plus de temps de traitement pour le serveur recevant les alertes.

Gestion de la sécurité de SQL Server Agent

- Rôles SQL Server Agent
- Proxies SQL Server Agent

Rôles SQL Server Agent

- Pour utiliser SQL Server Agent, l'utilisateur doit être membre de l'un des rôles de la base de données msdb suivants : SQLAgentUserRole, SQLAgentReaderRole, SQLAgentOperatorRole, ou être membre du rôle sysadmin

Rôles SQL Server Agent

- SQLAgentUserRole
 - C'est le plus petit des privilèges des rôles base de données de SQL Server Agent.
 - Les membres de ce rôle ont les permissions uniquement sur les jobs locaux et les jobs qu'ils ont programmés eux-mêmes.
 - Ils ne peuvent utiliser les jobs multiserveurs ni changer les propriétaires des jobs
 - Ils peuvent voir la liste des proxies disponibles

Rôles SQL Server Agent

- SQLAgentReaderRole
 - Il inclut toutes les permissions de SQLAgentUser Role, ainsi que les permission pour voir la liste de tous les jobs multiserveurs disponibles, leurs propriétés et leurs historiques.
 - Les membres de ce rôle peuvent voir tous les jobs, ainsi que leurs propriétés
 - Ils ne peuvent changer les propriétaires des jobs

Rôles SQL Server Agent

- SQLAgentOperatorRole
 - Il s'agit du plus grand privilège.
 - Il inclut toutes les permissions des rôles SQLAgentUserRole et SQLAgentReaderRole.
 - Les membres de ce rôle peuvent également voir les propriétés pour les opérateurs et les proxies, et énumérer les proxies et les alertes disponibles dans le serveur.
 - Ils peuvent exécuter toutes les opérations possibles sur les jobs mais ne peuvent pas changer le propriétaire des jobs.

Proxies SQL Server Agent

- Définissent le contexte de sécurité pour les jobs qui n'utilisent pas le Transact-SQL et qui accèdent à un sous-système particulier. (ex : activeX)
- Chaque proxy est associé à un ou plusieurs sous-systèmes.
- Une étape d'un job qui utilise le proxy peut accéder à un sous-système spécifique en utilisant le contexte de sécurité de l'utilisateur de Windows.
- Ils sont mappés aux utilisateurs Windows en utilisant des credentials

Automatisation des processus LABO

Vérifier si il y a des indexes sur la table person.person

- Vérifier si les indexes sont fragmentés
- Mettre en place un wizard de maintenance pour maintenir les indexes de la DB
- Exécuter la maintenance et vérifier le résultat
- Mettre en place un job de maintenance quotidien

Chapitre 6

Monitoring

SQL Server : Admin.

Objectifs

- Identifier l'état actuel des ressources afin de déterminer si des changements sont nécessaires ou pas.

Plan

- Vue de l'activité en cours
- Le System Monitor
- Le Profiler SQL Server
- Les triggers DDL
- Notification d'évènement

Vue de l'activité en cours

- Activity Monitor
- Dynamic Management Objects

Activity Monitor

- Outil graphique de SQL Server Management Studio qui affiche des informations sur les processus utilisateurs et les verrous en cours.
- Composé de plusieurs parties
 - Overview
 - Processes
 - Ressource Waits
 - Data File I/O
 - Recent Expensive Queries
- Chaque colonne permet un filtrage en fonction des données présentes (dynamique)

Activity Monitor

- Composé de plusieurs parties

- Overview

Montre à l'aide de graphiques

- la charge du processeur,
- Le nombre de tâches,
- Les écritures et lectures de la base de données
- Les exécutions de requêtes

Permet de changer l'intervalle des refreshs

Activity Monitor

- Composé de plusieurs parties

- Processes

Elle affiche une liste de tous les processus connectés au système.

Chaque processus est identifié par un ProcessID

La page affiche des informations sur le processus telles que

- Les bases de données auxquelles il accède
- Le nom de l'application client
- Le login utilisé par ce processus

Cette page permet de savoir quel processus est actuellement actif, et pour mettre fin à un processus si nécessaire

Activity Monitor

- Composé de plusieurs parties

- Resource waits

Cette partie affiche les informations concernant les ressources en attente.

Activity Monitor

- Composé de plusieurs parties

- Data File I/O Pane

Affiche les informations concernant les fichiers des bases de données de l'instance courante

Activity Monitor

- Composé de plusieurs parties

- Recent Expensive Queries

Affiche les informations concernant les requêtes les plus « couteuse ».

Fusion des résultats des tables

- sys.dm_exec_request
- sys.dm_exec_query_stats

Affiche les requêtes en cours de traitement ainsi que celles terminées.

Permet d'éditer les requêtes dans une fenêtre de requête

Permet d'examiner l'exécution plan

Dynamic Management Objects

- Ce sont des vues et des objets de gestion permettant d'interroger les métadonnées de SQL Server.
- Ils donnent des informations sur l'état actuel de SQL Server.

Dynamic Management Objets

- Les vues
 - Elles sont membres du schéma sys et sont préfixées par dm.

Vue	Description
sys.dm_db_partition_stats	Retourne des informations sur les pages et les lignes pour chaque partition dans la base de données
sys.dm_exec_sessions	Retourne des informations sur toutes les sessions courantes connectées à SQL Server
sys.dm_io_pending_io_requests	Retourne des informations sur les requêtes I/O en cours
sys.dm_os_memory_pools	Retourne des informations sur chaque objet dans le cache du système
sys.dm_os_threads	Retourne des informations sur les « threads » en cours dans le système
sys.dm_broker_queue_monitors	Retourne des informations sur chaque queue de services
sys.dm_tran_locks	Retourne des informations sur tout verrou actuellement demandé

Labo

- Monitorer l'activité en cours en utilisant Activity Monitor
- Monitorer l'activité en cours en utilisant les vues de gestion dynamique

Le System Monitor

- Introduction
- SQL Server Performance Objects
- Monitoring de SQL Server

Introduction

- Il s'agit d'un composant logiciel de Microsoft Management Console (MMC) qu'on peut utiliser pour connaître les performances de l'ordinateur (utilisation du processeur, de la mémoire, des statistiques sur les activités)
- Monitoring d'objets, instances et compteurs

Introduction

- Objets
 - Les objets sont les composants majeurs, ou les sous-systèmes de l'ordinateur
 - Ils peuvent être
 - du hardware (disque dure)
 - du software (un processus)
 - Une application (instance de SQL Server)

Introduction

- Compteurs

- Les compteurs maintiennent des données sur différents aspects des objets
- Ex : pour les objets Processus, les compteurs maintiennent les données sur le pourcentage de temps processeur et les temps utilisateur
- Ils sont construits avec le système d'exploitation et lisent continuellement les données de performance.
- Si l'objet a plusieurs instances, les compteurs pistent toutes les statistiques sur chaque instance.

Introduction

- Instances
 - Il peut y avoir différentes instances pour le même objet
 - On peut choisir de voir les valeurs pour une instance spécifique d'un objet ou les valeurs combinées de toutes les instances d'un objet.

Introduction

- Les données sont affichées dans des graphes, histogrammes ou rapports
- Ces derniers peuvent être vus et imprimés.

Introduction

- Alertes et journaux
 - Performance Logs and Alert : composant logiciel de la MMC.
 - Permet de capturer les informations de performances et les écrire dans un fichier de log.
 - Les compteurs de log enregistrent les données relatives aux ressources hardware et aux services système.
 - Les fichiers de log permettent de pister les compteurs sur une longue durée de temps.

Introduction

- Alertes et journaux
 - On peut également configurer des alertes qui se déclenchent quand le compteur atteint un certain seuil.

SQL Server Performance Objects

- SQL Server contient des objets et des compteurs qui peuvent être utilisés par le System Monitor pour monitorer les activités dans un ordinateur où fonctionne SQL Server.
- Chaque objet contient des compteurs qui déterminent différents aspects de l'objet à monitorer.

SQL Server Performance Objects

Objet	Description
SQLServer:Access Methods	Allocation des objets des bases de données
SQLServer:Buffer Manager	Les buffers utilisées par SQL Server (espace mémoire libre)
SQLServer:Databases	Bases de données SQL Server (espace libre des fichiers de log , nombre de transactions actives)
SQLServer:General Statistics	Activités du serveur (nombre d'utilisateurs connectés à une instance)
SQLServer:Locks	Les verrous créés par SQL Server (les timeout et les deadlock)

SQL Server Performance Objects

Objet	Description
SQLServer:Memory Manager	Utilisation de la mémoire (nombre de structures verrouillées)
SQLServer:Plan Cache	Cache SQL Server utilisé pour stocker les objets
SQLServer:SQL Errors	Erreurs SQL Server
SQLServer:Transactions	Transactions actives (nombre moyen de transactions)

Monitoring de SQL Server

- Requiert l'analyse de certains aspects clé du système
- Activités du disque
 - SQL Server utilise les appels d'E/S du système d'exploitation pour exécuter les opérations de lecture et d'écriture
 - 2 clés à monitorer
 - Les E/S du disque
 - Isoler les activités du disque que SQL Server crée

Monitoring de SQL Server

- 2 clés à monitorer

Les E/S du disque (**PhysicalDisk**)

- Pourcentage de temps pendant lequel le disque est occupé avec des opérations de lecture/écriture : s'il est supérieur à 90%, voir le compteur responsable de la taille des queues.
- Nombre moyen de requêtes de L/E qui sont dans la queue
- Taille courante de la queue du disque

Isoler les activités du disque que SQL Server crée

Monitoring de SQL Server

- 2 clés à monitorer

Les E/S du disque

Isoler les activités du disque que SQL Server crée (**Buffer Manager**)

- Nombre de lectures sur les pages physiques de la base de données par seconde :
minimiser le nombre de lectures en utilisant un cache plus large, des index, des requêtes plus efficaces ...
- Nombre d'écritures sur les pages physiques de la base de données par seconde :
minimiser le nombre d'écritures en utilisant un cache plus large, des index, des requêtes plus efficaces ...

Monitoring de SQL Server

- Utilisation de la mémoire
 - On doit s'assurer qu'aucun processus ne consomme beaucoup de mémoire ou n'est limité par des contraintes de mémoire.
 - Objets Buffer Manager
- Utilisation du CPU
 - Une augmentation du taux d'utilisation du CPU peut indiquer un besoin de mettre à jour le CPU et des processeurs, l'utilisation de requêtes non optimisées ou de design non optimisé.

Labo

- Monitorer les performances de SQL Server en utilisant les System Monitor

Performance object	Counter	Instance
SQLServer:Databases	Active Transactions	AdventureWorks
SQLServer:Locks	Lock Requests/sec	_Total
SQL Server:Buffer Manager	Page reads/sec	n/a
SQL Server:Buffer Manager	Page writes/sec	n/a

Le Profiler SQL Server

- Le Profiler SQL Server
- Les options du Profiler
- Les catégories de traces, les événements et les colonnes

Le Profiler SQL Server



- Interface graphique pour monitorer une instance du moteur de base de données ou de Analysis Services.
- Les données capturées peuvent être sauvées dans une table ou un fichier.

Le Profiler SQL Server



- Les fonctions du Profiler
 - Créer une trace
 - Voir les résultats de la trace quand elle tourne
 - Stocker les résultats de la trace dans une table ou un fichier
 - Démarrer, stopper, mettre en pause et modifier la trace
 - Redémarrer la trace

Le Profiler SQL Server



- Que faut-il monitorer à l'aide du Profiler?
 - Les performances des requêtes
 - Activités des utilisateurs
 - Performance de la base de données tempdb
 - Problèmes de deadlock
 - Connexions de logins
 - Opérations de L/E
 - Utilisation du CPU
 - ...
- Procédure stockée `sp_trace_create`

Les options du Profiler

- Options pour déterminer pour définir les activités à enregistrer et où les enregistrer
- Le template de la trace
 - On spécifie les classes d'évènements qu'on veut monitorer et les valeurs (ou colonnes) qu'on veut enregistrer.
 - Il faut pour cela sélectionner un template qui permet de gérer les classe d'évènements, les colonnes et les filtrer pour limiter les données
 - Il existe des templates prédéfinis qu'on peut modifier, et on peut créer d'autres templates.

Les options du Profiler

- Sauvegarder les traces
 - Dans les tables
 - Spécifier l'emplacement et le nom des tables
 - Le maximum de lignes à stocker
 - Dans des fichiers
 - Spécifier l'emplacement et le nom du fichier
 - Sa taille maximum
 - Comportement de la trace quand le fichier est rempli

Les catégories de traces, les évènements et les colonnes

- L'information enregistrée par les traces est divisée en catégories
- Une catégorie contient des évènements
- Chaque évènement a ses propres attributs (colonnes)

Les catégories de traces, les évènements et les colonnes

- Catégories (groupes d'évènements)
 - Sécurité
 - Sessions
 - Procédures stockées
 - TSQL

Les catégories de traces, les évènements et les colonnes

- Evènements
 - Occurrence d'une action à l'intérieur de l'instance du moteur de base de données de SQL Server

Catégorie	Évènement	Description
Security Audit	Audit Login	L'utilisateur s'est bien connecté à SQL Server
Security Audit	Audit Logout	L'utilisateur s'est bien déconnecté de SQL Server
Sessions	ExistingConnection	Propriétés des connections actuelles
Stored Procedures	RPC: completed	Exécution d'une procédure stockée
TSQL	SQL:BatchCompleted	Exécution d'un lot de scripts
TSQL	SQL:BatchStarting	L'exécution commence

Labo

- Utilisation du profiler de SQL Server

SQL Server : Admin.

Les triggers DDL

- Les triggers DDL
- Création des triggers
- Gestion des triggers

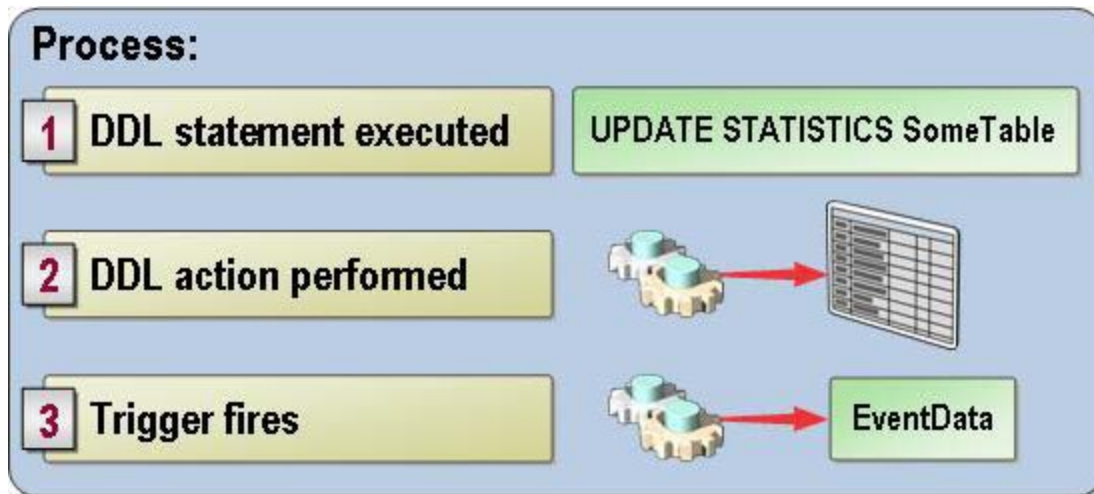
Les triggers DDL

- Les triggers DDL peuvent se déclencher lors de l'exécution d'une action sur un objet d'une base de données
- Ex : CREATE TABLE, DROP TRIGGER, or UPDATE STATISTICS.
- Ils peuvent être utiliser pour loguer les actions administratives

Les triggers DDL

- Certains triggers DDL opèrent au niveau serveur (CREATE_LOGIN), d'autres au niveau base de données (CREATE_TABLE)
- <http://technet.microsoft.com/en-us/library/bb522542> contient la liste des statement en fonction de leur scope

Les triggers DDL



Création des triggers

- La création se fait à l'aide de l'instruction CREATE TRIGGER

```
CREATE TRIGGER UpdStats  
ON DATABASE (ON ALL SERVER)  
FOR UPDATE_STATISTICS  
AS  
...
```

Création des triggers

- La fonction eventdata
 - Les Triggers standards créent les tables insérées et supprimées.
 - Les triggers DDL permettent d'utiliser la fonction eventdata pour obtenir des informations sur l'évènement qui a déclencher le trigger.
 - Cette fonction retourne un document XML EVENT_INSTANCE qui retourne des données telles que:
 - Posttime
 - Spid (id de la base de données)
 - Eventtype
 - ...

Création des triggers

```
<EVENT_INSTANCE>
<PostTime>2004-06-18T02:14:20.640</PostTime>
<SPID>58</SPID>
<EventType>UPDATE_STATISTICS</EventType>
<ServerName>SQLPC</ServerName>
<LoginName>SQLPC\Administrator</LoginName>
<UserName>SQLPC\Administrator</UserName>
<DatabaseName>AdventureWorks</DatabaseName>
<SchemaName>Production</SchemaName>
<ObjectType>STATISTICS</ObjectType>
<TargetObjectName>Product</TargetObjectName>
<TargetObjectType>TABLE</TargetObjectType>
<TSQLCommand>
<SetOptions ANSI_NULLS="ON" ANSI_NULL_DEFAULT="ON"
ANSI_PADDING="ON" QUOTED_IDENTIFIER="ON"
ENCRYPTED="FALSE"/>
<CommandText>
UPDATE STATISTICS Production.Product&#x0D;
</CommandText>
</TSQLCommand>
</EVENT_INSTANCE>
```

Création des triggers

```
CREATE TRIGGER UpdStats
ON DATABASE
FOR UPDATE_STATISTICS
AS
DECLARE @data XML
DECLARE @posttime NVARCHAR(24)
DECLARE @database NVARCHAR(100)
DECLARE @targetobject NVARCHAR(100)
SET @data = eventdata()
SET @posttime = CONVERT(NVARCHAR(24),
    @data.query('data(//PostTime)'))
SET @database = CONVERT(NVARCHAR(100),
    @data.query('data(//DatabaseName)'))
SET @targetobject = CONVERT(NVARCHAR(100),
    @data.query('data(//TargetObjectName)'))
PRINT @posttime
PRINT @database
PRINT @targetobject
```

Gestion des triggers

- Consulter les triggers

```
SELECT name  
FROM sys.triggers
```

```
SELECT definition  
FROM sys.sql_modules  
WHERE [object_id] =  
(SELECT [object_id]  
FROM sys.triggers  
WHERE name='trigger_name')
```

- Modification des triggers

```
ALTER TRIGGER UpdStats  
ON DATABASE (ON ALL SERVER)  
FOR UPDATE_STATISTICS  
AS  
...
```


Gestion des triggers

- Supprimer les triggers

```
DROP TRIGGER UpdStats  
ON DATABASE
```

Labo

- Créer une DB DbToDrop;
- Créer une table TabToDrop;
- Créer un trigger qui se déclenche lors d'une modification ou suppression de cette table, annulant la transaction (rollback) et afficher un message « d'erreur »;
- Vérifier la création du trigger;
- Supprimer la table (le trigger doit l'en empêcher et afficher votre message d'erreur);
- Désactiver le trigger;
- Supprimer la table.

Labo

Exercice 2

- Enregistrer la commande, sa date d'exécution, le login et le serveur dans une table:
 - Créer nouvelle DB;
 - Créer une table addtablelog, avec les champs:
 - logid int identity(1,1),
 - cmand nvarchar(400),
 - cmdat date,
 - login nvarchar(50),
 - servr nvarchar(50)
 - Créer un trigger qui se déclenche lors de la création de table et ira stocker dans notre table addtablelog la commande utilisée (CREATE ...), l'utilisateur ayant exécuté cette requête, la date d'exécution et le serveur;
 - Créer une table et vérifier que le slogs soit bien dans addtablelog.

Notification d'évènement

- Les Notifications
- Création d'une notification
- Traitement d'une notification
- Gestion des notifications

Les notifications

- Peuvent répondre à un certain nombre d'évènements
- Permettent d'envoyer les données des évènements à partir de différentes base de données et serveurs vers un service manipulant les évènements (Service Broker).
- L'évènement peut être une opération DDL ou DML sur un objet de la base de données, ou une trace.

Les notifications

- Pour capturer les évènements
 - Implémenter un service « Service Broker »
Procédure stockée ou application utilisant le framework Service Broker qui monitore une queue pour les messages entrants.
Il peut être écrit en utilisant du Transact-SQL ou tout autre langage
 - Créer une queue, un service et une route
- Les messages sont envoyés au service par le Service Broker qui utilise une queue où le message est stocké jusqu'à ce que le service soit prêt à accepter le message.
- SQL Server inclus des types et contrats de messages prédéfinis.

Création d'une notification

- Se fait via l'instruction CREATE EVENT NOTIFICATION
 - Donner un nom unique pour le scope
 - Spécifier le scope (SERVER, DATABASE, QUEUE)
 - Spécifier l'évènement qu'on veut capturer
 - Spécifier le Service Broker vers lequel les informations sur l'évènement doivent être envoyées.

Création d'une notification

- Exemples

```
CREATE EVENT NOTIFICATION CreateLoginEvent  
ON SERVER  
FOR CREATE_LOGIN  
TO SERVICE 'NotifyService', 'current database'
```

```
CREATE EVENT NOTIFICATION TraceSQLBatches  
ON SERVER  
FOR SQL_BatchCompleted  
TO SERVICE 'NotifyService', 'current database'
```


Traitement d'une notification

- Pour créer le service du Service Broker, utiliser une procédure stockée qui s'active à chaque fois qu'un message apparaît dans la queue.

```
CREATE QUEUE NotifyQueue  
WITH ACTIVATION (  
  PROCEDURE_NAME = dbo.EventProc,  
  MAX_QUEUE_READERS = 5,  
  EXECUTE AS SELF)  
GO
```

Traitement d'une notification

- Réception des messages
 - La procédure stockée doit retrouver le message dans la queue en utilisant la commande RECEIVE et stocker toutes les informations dans une variable.
 - Le processus doit être suspendu jusqu'à ce qu'un message soit reçu. Utiliser la commande WAITFOR

```
DECLARE @messageTypeName NVARCHAR(256),  
        @messageBody XML  
;WAITFOR(  
    RECEIVE TOP(1)  
        @messageTypeName = message_type_name,  
        @messageBody = message_body  
    FROM dbo.NotifyQueue;  
), TIMEOUT 2000 ;  
IF @@ROWCOUNT = 0  
    RETURN
```

Traitement d'une notification

- Traitement des messages
 - Le message contient une colonne « message_type_name » qui décrit le type du message.
 - Il contient aussi une colonne « message_body » qui contient toutes les informations sur l'évènement.
 - Il s'agit d'un document XML EVENT_INSTANCE.

Traitement d'une notification

```
<EVENT_INSTANCE>
<SPID>58</SPID>
<EventType>ALTER_TABLE</EventType>
<ServerName>SQLPC</ServerName>
<LoginName>SQLPC\Administrator</LoginName>
<UserName>SQLPC\Administrator</UserName>
<DatabaseName>AdventureWorks</DatabaseName>
<SchemaName>dbo</SchemaName>
<ObjectName>T1</ObjectName>
<ObjectType>TABLE</ObjectType>
<TSQLCommand>
<SetOptions ANSI_NULLS="ON" ANSI_NULL_DEFAULT="ON"
ANSI_PADDING="ON" QUOTED_IDENTIFIER="ON"
ENCRYPTED="FALSE"/>
<CommandText>ALTER TABLE T1 ADD col13 int</CommandText>
</TSQLCommand>
</EVENT_INSTANCE>
```

Traitement d'une notification

```
DECLARE @cmd NVARCHAR(1000)
DECLARE @posttime NVARCHAR(24)
DECLARE @spid NVARCHAR(6)
DECLARE @hostname NVARCHAR(100)
DECLARE @loginname NVARCHAR(100)
SET @cmd = CONVERT(NVARCHAR(100),
@messagebody.query('data(//TSQLCommand//CommandText)'))
SET @posttime = CONVERT(NVARCHAR(24),
@messagebody.query('data(//PostTime)'))
SET @spid = CONVERT(NVARCHAR(6),
@messagebody.query('data(//SPID)'))
SET @hostname = HOST_NAME()
SET @loginname = SYSTEM_USER
INSERT INTO AuditLog(Command,PostTime,HostName,LoginName)
VALUES(@cmd, @posttime, @hostname, @loginname)
GO
```

Gestion des notification

- Consulter les notifications

```
SELECT name  
FROM sys.event_notifications
```

```
SELECT name  
FROM sys.service_queues
```

- Supprimer les notifications

```
DROP EVENT NOTIFICATION UpdateStats
```

Labo

Sur la DB SQLServerDBTest:

- Activer le service broker
[`ALTER DATABASE SQLServerDBTest
SET ENABLE_BROKER
GO]`
- Créer une queue 'dbo.EventNotificationQueue' (+ Vérifier son existence);
- Créer un service /EventNotificationService avec le schémas:
([<http://schemas.microsoft.com/SQL/Notifications/PostEventNotification>])
- Créer un événement 'NotifyCreateTableEvents' lorsqu'il y a création de table;
- Créer un événement 'NotifyAlterTableEvents' lorsqu'il y une modification de table
- Vérifier la création des événements
- Créer une table
- Modifier la table
- Visualiser le message en version xml depuis dbo.EventNotificationQueue

Chapitre 7

Transfert de données

SQL Server : Admin.

Objectifs

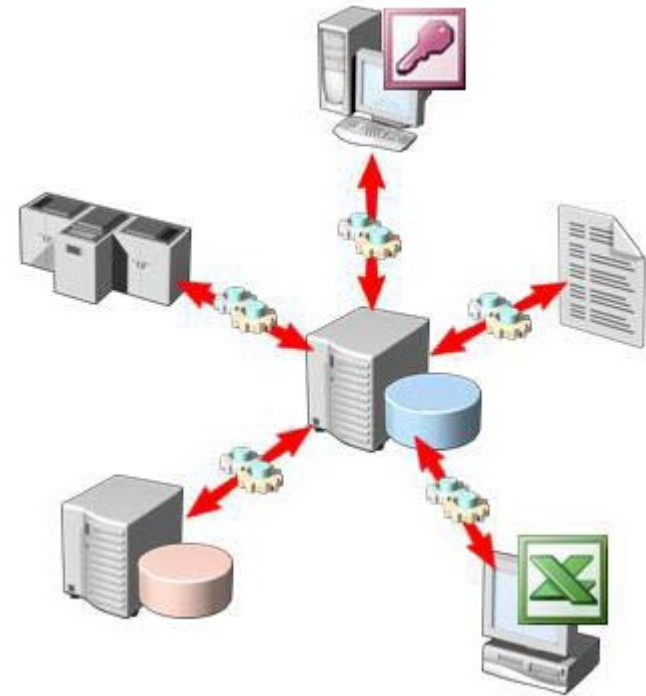
- Savoir convertir, nettoyer, standardiser, résumer et combiner les données venant de différentes sources de données
- Intégrer les données
- Se familiariser avec SQL Server Intégration Services (SSIS)

Plan

- Le transfert de données
- La transformation des données
- Outils pour le transfert des données
- Introduction à SQL Server Integration Services
- Utilisation de SQL Server Integration Services

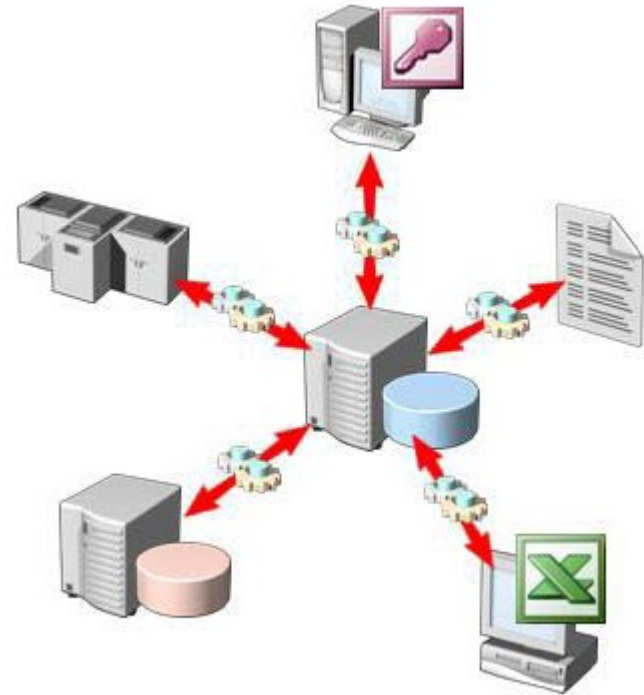
Le transfert des données

- Les transfert des données consiste à déplacer les données d'un espace de stockage vers un autre
- Théoriquement, toutes les bases de données nécessitent une solution de transfert de données



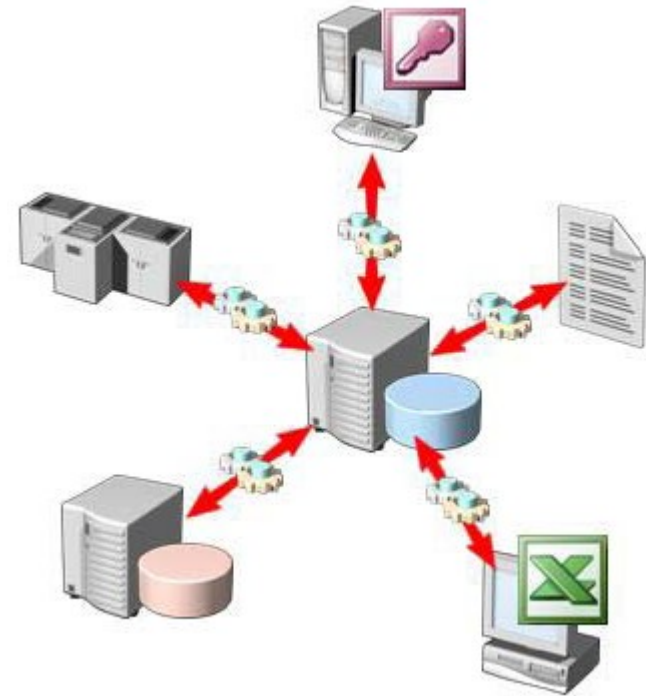
Le transfert des données

- Les étapes de transfert de données sont les suivantes :
 - Intégration
 - Transformation (optionnel)
 - Chargement
- => ETL (Extraction, Transformation, Load)



Le transfert des données

- Scénario commun de transfert de données
 - Déplacer les données vers un système de Datawarehouse
 - Consolidation des données dans un seul emplacement.
 - Déplacer les données d'un système vers un autre système
 - Déplacer les données d'un système non relationnel (Microsoft Office Excel®) vers un système relationnel (SQL Server)



La transformation des données

- Quand on importe des données dans un système, on doit s'assurer que celles-ci ont le bon format, et qu'elles respectent les règles de validations qu'on aurait instauré.
- Les transformation des données va permettre ces vérifications.
- Suppression des inconsistances

La transformation des données

- Format des données
 - On peut rencontrer des différences entre les formats des données qui proviennent de différentes sources de données
 - Exemple de modification
 - Conversion d'un char vers un varchar
 - Conversion d'un string en datetime
 - ...

La transformation des données

- Agréger les données
 - On peut devoir copier les données d'un système OLTP (OnLine Transaction Processing) vers un système décisionnel
 - Les systèmes décisionnels ne requièrent pas tout le temps de stocker les détails
 - On a le plus souvent de valeurs agrégées
 - Pour améliorer les performances, on peut calculer et stocker ces agrégations

La transformation des données

- Consistances des données
 - Quand les données proviennent de différentes sources, il peut y avoir différentes versions de la même donnée
 - Ex : les adresses, les numéro de téléphones ...

La transformation des données

- Validation des données
 - Si on a le moindre doute sur la validité des données de la source, on doit les valider avant de les stocker dans le système de destination
 - La validation peut être automatique ou manuelle

La transformation des données

- Scénarios communs

- Data Warehouse

Collection de données provenant de différentes sources de données

On y stocke des données agrégées

Les données doivent être consistantes, valides, et stockées dans une structure appropriée

- Enterprise Application Integration (EAI)

Partages de toutes les données disponibles au sein de l'organisation

La transformation des données

- Scénarios communs

- Business To Business Integration (B2B)

Les systèmes B2B ont besoin d'un mapping entre les structures des données des organisations

- ...

Outils pour le transfert des données

- Bulk Copy Program (BCP)
 - Programme en ligne de commande permettant d'importer et exporter des données de ou vers un fichier plat
- Bulk Insert
 - Version Transact-SQL du BCP
 - Utile quand on veut inclure des fonctionnalités d'import de données dans la logique T-SQL

Outils pour le transfert des données

- XML Bulk Load Composant
 - Objets COM permettant de charger des données XML dans des tables de SQL Server
- Replication
 - Quand un ou plusieurs systèmes ont la même structure, et quand la transformation des données n'est pas requise, on peut appliquer automatiquement sur un système les modifications appliquées sur un autre système
 - La réplication n'est pas conseillée quand les données doivent être transformées, ou quand celles-ci proviennent de sources de données non relationnelles.

Introduction à SQL Server Integration Services

- Qu'est ce que SSIS ?
- Comment SSIS traite-t-il les tâches?
- Les outils SSIS

Qu'est ce que SSIS ?

- Plateforme pour construire des solutions d'intégration de données
- Inclus des packages ETL
- SSIS a plusieurs composant
 - SSIS service
N'est pas obligatoire pour faire tourner un package, mais si le service est stoppé, les packages aussi.
 - SSIS runtime engine et le runtime executable
Garde le layout du package et gère le logging, débugging, la configuration, les connections et les transactions
 - SSIS data flow engine et data flow component
 - SSIS clients

Qu'est ce que SSIS ?

- Terminologie SSIS

- Package

Il s'agit d'une collection de connexions d'éléments de contrôle de flux, d'éléments de data flow, de manipulateurs d'évènement, de variables, et de configurations qu'on peut assembler graphiquement via les outils SSIS ou en les programmant. Ces packages peuvent être stockés dans le serveur d'intégration ou un système de fichiers

Le package est comme une unité de travail qu'on va chercher, exécuter et sauver.

Qu'est ce que SSIS ?

- Terminologie SSIS

- Les tâches

Tâches pour effectuer un certain nombre de fonctions

- Containers

Objets offrant une structure aux packages

Il permettent de supporter la répétition des flux de contrôle dans le package, et de grouper les tâches et d'autres containers dans des unités de travail

Comment SSIS traite-t-il les tâches?

- Un package consiste en un contrôle de flux, et optionnellement, un ou plusieurs data flows
- Contrôle de flux
 - Containers
 - Tâches
 - Contraintes de priorité

Comment SSIS traite-t-il les tâches?

- Contrôle de flux

- Containers

Les containers regroupent les tâches et d'autres containers dans des unités de travail et supportent les bouclage.

- Tâches

Effectuent un travail au sein d'un package. Ces tâches peuvent être des :

- Tâches ETL
 - Tâches de copie de données
 - Tâches communication avec d'autres processus
 - Tâches de travail avec des objets et des données de SQL Server
 - Tâches de travail et de traitement d'objets d'Analysis Services
 - Tâches administratives tels que les backups

Les tâches peuvent aussi contenir des scripts pour implémenter des fonctionnalités particulières

- Contraintes de priorité

Comment SSIS traite-t-il les tâches?

- Contrôle de flux

- Containers
- Tâches
- Contraintes de priorité

Créent une séquence ordonnée d'exécution de containers et de tâches

Comment SSIS traite-t-il les tâches?

- Data Flow
 - Tâche spéciale faisant partie des tâches de contrôle de flux
 - 3 types de composants de Data Flow
 - Sources
 - Destination
 - transformation

Comment SSIS traite-t-il les tâches?

- Data Flow

- 3 types de composants de Data Flow

- Sources

- Extrait les données d'une grande variété de sources de données (bases de données relationnelles, fichiers textes, feuilles de calculs, ...)

- Destination

- Chargement des données dans n'importe quelle source de données

- Transformation

- Modification des données
 - Validation, agrégation et mapping des données

Les outils SSIS

- Wizards
 - Assistant d'import et d'export
Peut importer et exporter de données de et vers SQL Server, des fichiers textes, Microsoft Office Access, Microsoft Office Excel et/ou des providers OLE DB et/ou ADO.NET
 - Assistant d'installation de package
Transfert les packages vers un système de fichier ou une instance de SQL Server
Avant de l'exécuter, il faut construire l'utilitaire de déploiement qui contient les packages qu'on veut déployer
 - Assistant de migration de packages

Les outils SSIS

- Designers SSIS
 - Outils graphiques pour la création des packages
 - On peut les utiliser pour créer des flux de contrôle, des data flows, des manipulateurs d'évènements, et des packages.
 - Il y a aussi des gestionnaires de connexion qu'il faut configurer

Les outils SSIS

- Utilitaires Command-prompt
 - Dtexec :
exécute un package sur la machine locale
On peut utiliser l'utilitaire graphique dtexecui.exe pour administrer les dtexec.
 - Dtutil
Copie, supprime, déplace, signe et vérifie les packages

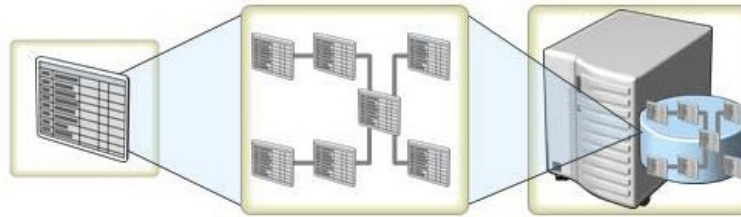
Les outils SSIS

- Labo
 - Export des données en utilisant l'assistant d'import et d'export.

Utilisation de SQL Server Integration Services

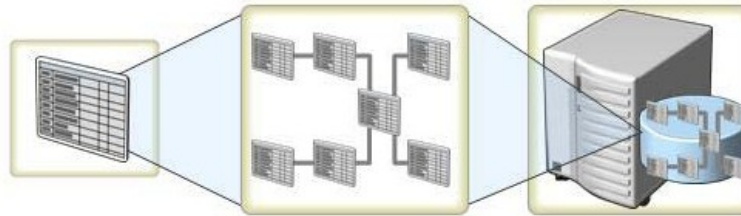
- les sources de données et les vues de la source de données
- Le gestionnaire de connexion
- Les packages de contrôle de flux
- Les tâches Data Flow
- Déploiement des packages

Les sources de données et les vues de la source de données



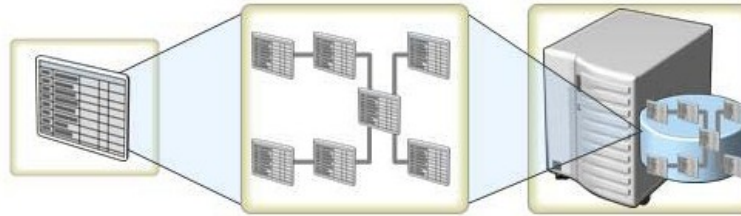
- Les sources de données et les Data Source Views (DSV) sont créés dans un projet.
- Elles sont alors accessibles par tous les packages d'un projet.
- Elles ne peuvent être déployées
- SSIS utilise leurs propriétés pour créer les gestionnaires de connexion

les sources de données et les vues de la source de données



- Sources de données
 - Il s'agit d'une référence temps réel vers la connexion vers les sources de données
 - Elle ne sont pas requises, mais peuvent rendre plus rapide le déploiement du projet ou du package.
 - Avantages
 - Les sources de données sont accessible par tout le projet
 - Les gestionnaire de connexion copie les informations de connexion de la source de donnée, et continue à fonctionner même si celle-ci n'est plus disponible
 - Si on change les propriétés de la source de données, les propriétés des gestionnaires de connexion seront également changées

les sources de données et les vues de la source de données

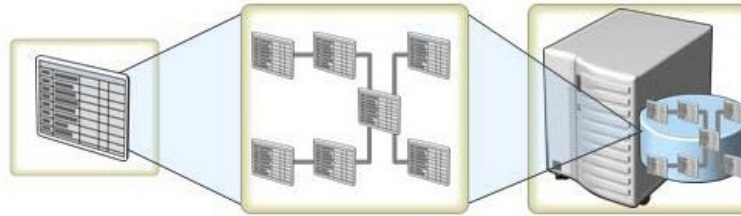


- Data Source View (DSV)
 - UDM (Unified Dimensional Model) : Modèle créant une couche logique intermédiaire entre la DB relationnel et la structure du cube et des dimensions.

Avantages :

- Manipulation des données plus facile pour les end-users
- Toute l'information utile est contenue dans le modèle
- Consolidation des données
- Règles du business définies dans le modèle

les sources de données et les vues de la source de données



- Data Source View (DSV)
 - DSV : représentation logique de la DB relationnelle

Avantages

- Création de clés logiques
- Création de requêtes nommées permettant de construire des vues logiques
- Création de calculs nommés permettant d'ajouter des colonnes logiques

Le gestionnaire de connexion

- Représentation logique de la connexion
- SSI crée les connexions physiques en utilisant les informations de connexions contenues dans le gestionnaire de connexion
- Un package peut avoir plusieurs gestionnaires de connexions. Chacun d'eux peut avoir différents attributs

Le gestionnaire de connexion

- Types de gestionnaires de connexions

Type	Description
ADO	Sources de données ActiveX Data Objects (ADO)
ADO.NET	Se connecte aux sources de données en utilisant les providers .NET
EXCEL	Workbooks Excel
FILE	Fichiers ou répertoires
FLATFILE	Fichiers plats
FTP	Serveur FTP
HTTP	Serveur Web

Le gestionnaire de connexion

- Types de gestionnaires de connexions

Type	Description
MSMQ	Queues de messages
MSOLAP90	Instances de SQL Server Analysis Services (SSAS) et projets Analysis Services
MULTIFILE	Multiples fichiers et répertoires
MULTIFLATFILE	Plusieurs fichiers de données et répertoires
OLEDB	Se connecte aux sources de données en utilisant les providers OLE DB
ODBC	Se connecte aux sources de données en utilisant les providers ODBC
SMO Server	Serveur SQL Management Object (SMO)

Le gestionnaire de connexion

- Types de gestionnaires de connexions

Type	Description
SMTP	Serveur mail SMTP
SQLMOBILE	Bases de données SQL Server Mobile
WMI	Se connecte à un serveur et spécifie le scope de la gestion de Microsoft Windows® Management Instrumentation (WMI) dans le serveur

Les packages de contrôle de flux

- Processus consistant à contrôler les chemins d'exécution du package
- On peut placer des conditions pour contrôler la séquence d'exécution des tâches en se basant sur le succès de l'exécution des tâches ou sur les résultats de certaines expressions
- On peut grouper ou boucler les tâches

Les packages de contrôle de flux

- Les tâches
 - Exécutent les travaux du package
 - La tâche la plus importante est la tâche de data flow
 - Il peut y avoir plusieurs data flows dans un même package => plusieurs opérations ETL
 - Les autres tâches peuvent effectuer plusieurs opérations allant de l'envoi d'e-mail à la réception de données provenant d'un service Web

Les packages de contrôle de flux

- Contraintes de priorité
 - Permettent de rassembler les tâches
 - Structurent le flux des opérations dans le package
 - Elle peuvent dépendre de certaines conditions

Ex :

- Succès d'une tâche
- Évaluation d'une expression

Les packages de contrôle de flux

- Containers

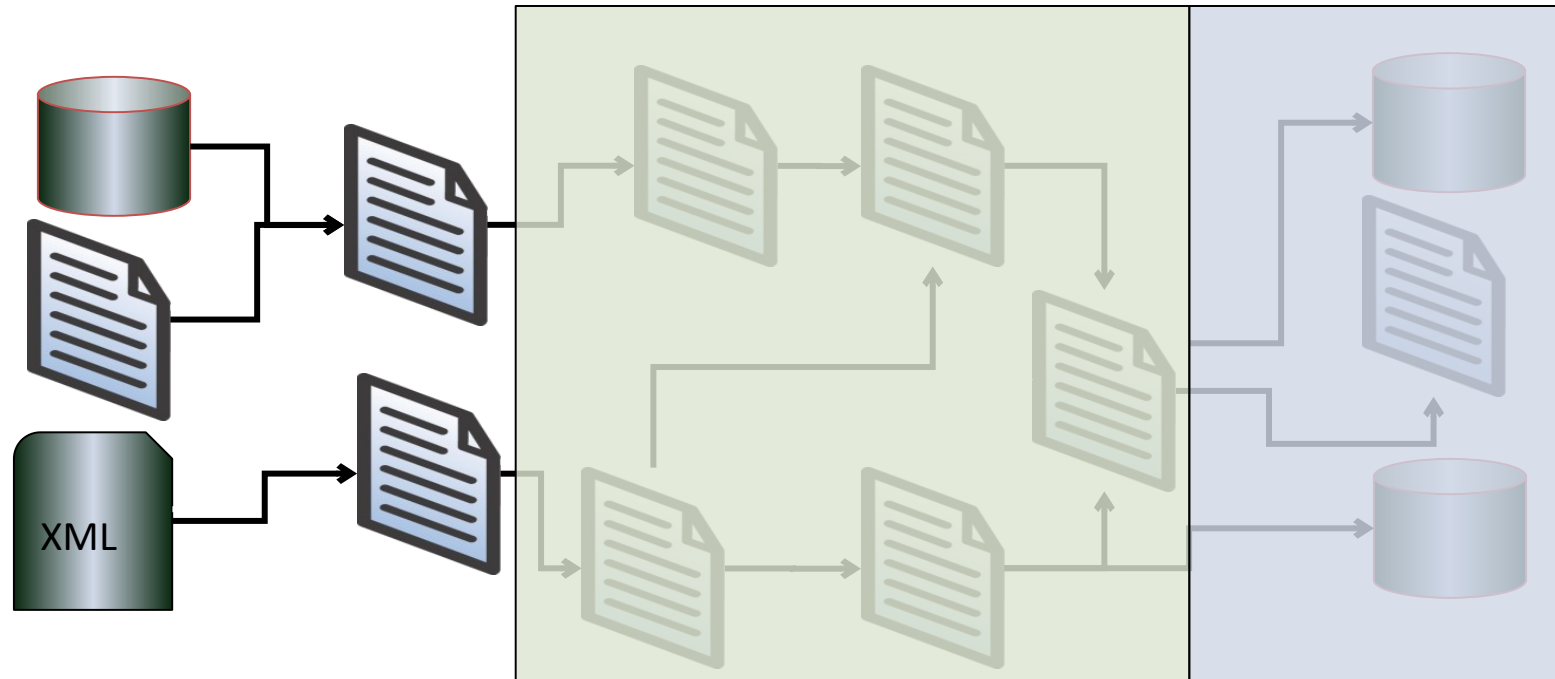
- Permettent de grouper et boucler certaines tâches
- On peut également les utiliser en tant que scope pour certaines variables, pour rendre certaines valeur privées et non accessibles à certaines opérations
- Deux types de containers de bouclage

Foreach : utilise un énumérateur pour effectuer le bouclage

- des lignes du jeu d'enregistrements ADO et des informations de schéma ;
- des structures de fichiers et de répertoires ;
- des variables système, package et utilisateur.
- ...

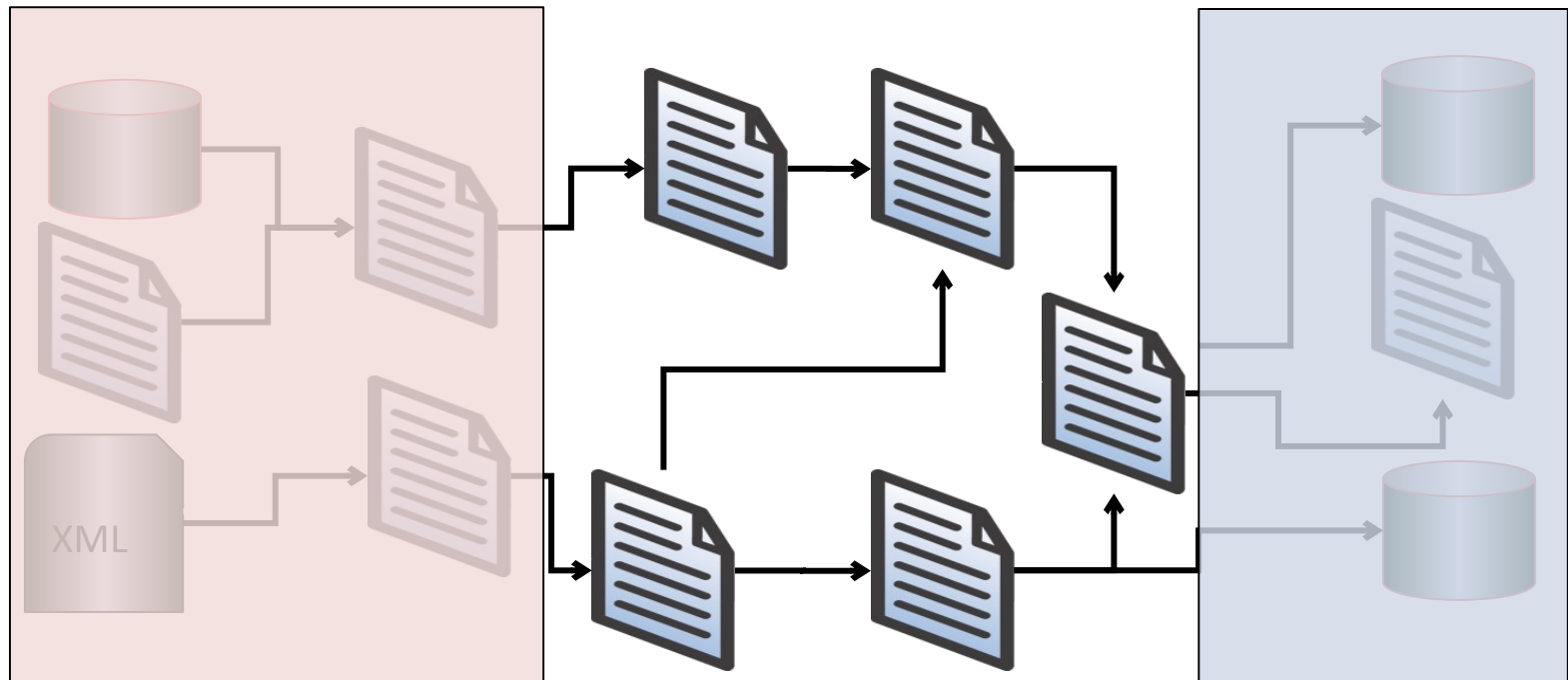
For : utilise une expression variable

Les tâches Data Flow



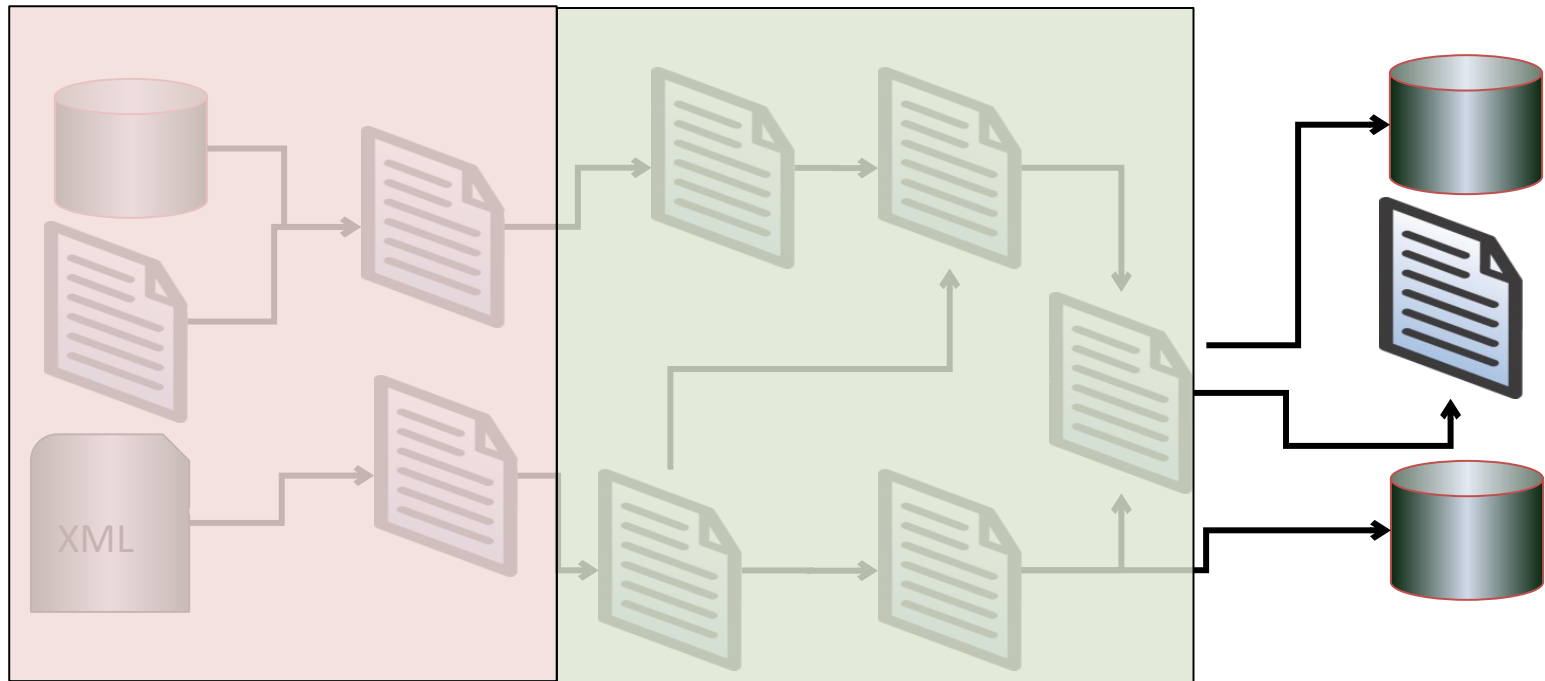
Extraire : Accéder aux systèmes, internes et externes, de stockage des données afin de collecter les données utiles.

Les tâches Data Flow



Transformer : Les données doivent être vérifiées, formatées, nettoyées et consolidées.

Les tâches Data Flow



Charger : Alimenter les entrepôts de données.

Déploiement des packages

- Le processus de déploiement des packages consiste en 3 étapes
 - Créer et définir les configurations du package
 - Configurer l'utilitaire de déploiement et construire le projet
 - Copier et déployer

Déploiement des packages

- Le processus de déploiement des packages consiste en 3 étapes

- Créer et définir les configurations du package

Les configurations permettent de mettre à jour les valeurs des propriétés lors de l'exécution

- Ex : quand on déploie les packages vers la production
 - » Il faut modifier les informations de connexion vers les serveurs

Différents types de configuration:

- Fichier de configuration XML
 - Variable d'environnement
 - Entrée de Registre
 - Variable de package parent
 - Table SQL Server

Déploiement des packages

- Le processus de déploiement des packages consiste en 3 étapes
 - Configurer l'utilitaire de déploiement et construire le projet

La configuration de l'utilitaire de déploiement permet d'activer la construction de l'utilitaire lors de la construction de projet, et la spécification du chemin de création.
 - Copier et déployer

La manifeste de déploiement créer lors la construction du projet aura un nom de fichier nom_du_projet.SSISDeploymentManifest.

Labo

- Transfert de données en utilisant SQL Server Integration Services
 - Création d'un package SSIS
 - Déploiement d'un package SSIS

Chapitre 8

Introduction à la réplication

SQL Server : Admin.

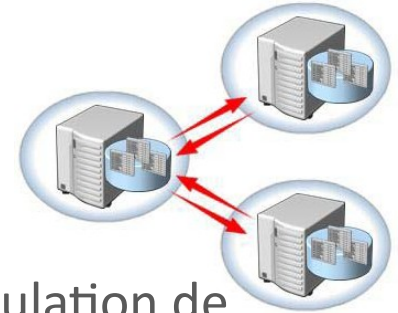
Objectifs

- Concept de distribution et disponibilité des données
- Connaitre les différents types de réplication, les rôles serveurs et les scénarios de réplication
- Implémenter un scénarios de réplication

Plan

- Définition
- Types de réplication
- Les rôles des serveurs
- Les articles, les publications et les abonnements

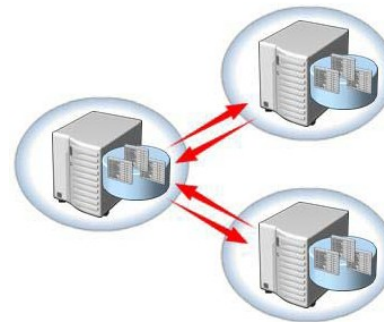
Définition



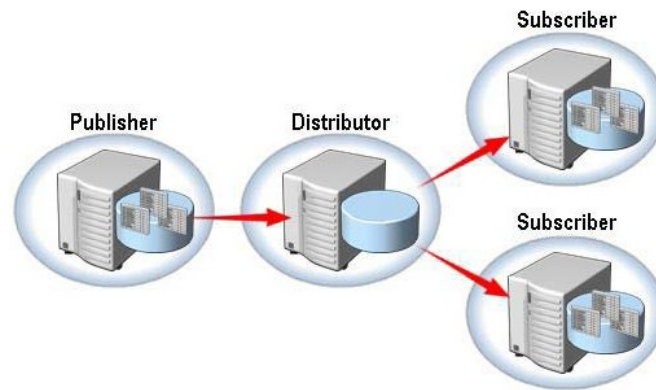
- La réplication est le mécanisme de création et de manipulation de plusieurs copies de la même donnée.
- Les données sont distribuées sur le même site et leurs modifications sont répliquées dans des temps raisonnables

Définition

- Avantages
 - Les données sont proches géographiquement des utilisateurs
 - Permettent à des sites autonomes de ne pas être continuellement connectés aux autres sites
 - ...

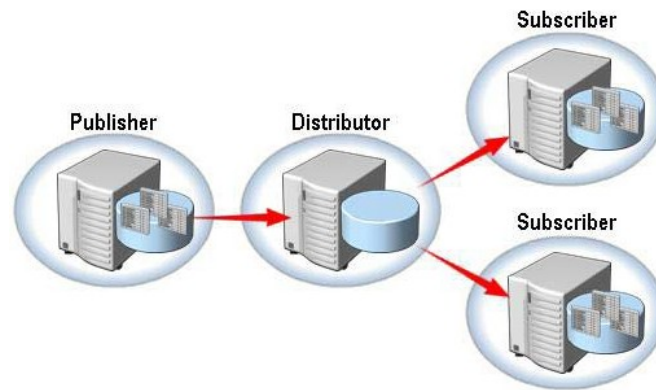


Les rôles des serveurs



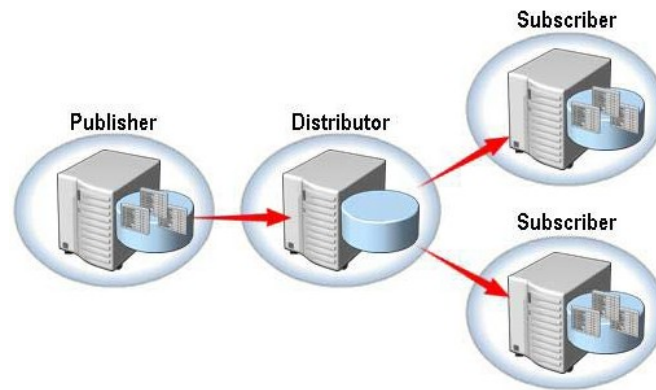
- Une instance de SQL Server peut être un publieur, un distributeur, un abonné, ou une combinaison des trois.
- La topographie représente la combinaison utilisée

Les rôles des serveurs



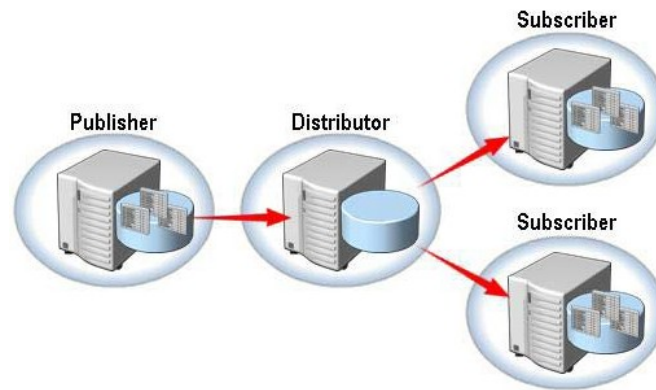
- Publieur
 - Possède la copie originale des données et la rend disponible aux abonnés.
 - Les données sont envoyées au distributeur qui les transfère aux abonnés

Les rôles des serveurs



- Distributeur
 - Stocke les informations sur le statut de la réplication, les métadonnées de la publication, et parfois, les données.
 - Le publieur peut faire office de distributeur, mais quand il y a une grande quantité de données, un distributeur séparé est créé.

Les rôles des serveurs



- Abonnés
 - Possède les copies de données
 - On peut lui
 - Interdire toute modification des données
 - Permettre de modifier ces copies de données
 - Permettre de modifier ces copies de données et les fusionner après

Types de réplication

- Snapshot Replication

- La réplication de capture instantanée transmet les données telles qu'elles apparaissent à un moment précis, sans contrôler les mises à jour des données
- Lors de la synchronisation, l'intégralité de la capture instantanée est générée et envoyée aux abonnés
- L'utilisation de la réplication de capture instantanée seule est la plus appropriée lorsqu'une ou plusieurs des conditions suivantes est vraie :
 - Les données sont peu modifiées ;
 - il est permis de posséder des copies de données non mises à jour par rapport au serveur de publication pendant un certain temps ;
 - de faibles volumes de données sont répliqués ;
 - un volume important de modifications se produit sur un court laps de temps.

Types de réplication

- Transactional Replication

- La réplication transactionnelle commence en général avec la capture instantanée des objets et des données de la base de données de publication
- Dès que la première capture est effectuée, les changements de données et les modifications de schémas effectués ensuite au niveau du serveur de publication sont en général transmis à l'Abonné à mesure qu'ils se produisent (presque en temps réel)
- Les changements de données sont appliqués à l'Abonné dans le même ordre et dans les mêmes limites de transaction que sur le serveur de publication

Types de réplication

- Transactional Replication

- La réplication transactionnelle est en général utilisée dans les environnements serveur à serveur, et convient pour chacun des cas suivants :

Vous souhaitez propager les modifications incrémentielles vers les Abonnés, au fur et à mesure qu'elles s'exécutent.

L'application nécessite une faible latence entre le moment où les changements sont effectués sur le serveur de publication et le moment où ils parviennent à l'Abonné.

L'application a besoin d'un accès aux états de données intermédiaires. Par exemple, si une ligne change cinq fois, la réplication transactionnelle permet à une application de réagir à chaque changement (par exemple l'activation d'un déclencheur), et non pas seulement au dernier changement effectué sur la ligne. Le serveur de publication a un volume très élevé d'activité d'insertion, de mise à jour et de suppression.

Le serveur de publication ou l'Abonné est une base de données non-SQL Server, Oracle par exemple.

Types de réplication

- Merge Replication

- La réplication de fusion, comme la réplication transactionnelle, démarre généralement avec une capture instantanée des objets et des données de la base de données de publication.
- Les modifications de données et de schéma ultérieures qui sont effectuées sur le serveur de publication et sur les Abonnés sont suivies avec des déclencheurs.
- L'abonné est synchronisé avec l'éditeur lorsqu'il est connecté au réseau et il échange toutes les lignes qui ont changé entre l'éditeur et l'abonné depuis la dernière synchronisation.

Types de réplication

- Merge Replication

- La réplication de fusion est généralement utilisée dans des environnements de serveur à client. La réplication de fusion est appropriée dans les situations suivantes :

Plusieurs abonnés peuvent mettre à jour les mêmes données à différents moments et propager ces modifications au serveur de publication et à d'autres Abonnés. des abonnés doivent recevoir des données, apporter des modifications hors connexion et synchroniser ultérieurement ces modifications avec l'éditeur et d'autres abonnés ;

Chaque Abonné requiert une partition de données différente.

Des conflits peuvent se produire et, le cas échéant, vous devez pouvoir les détecter et les résoudre.

L'application requiert le résultat des modifications des données au lieu de devoir accéder aux états intermédiaires des données. Par exemple, si une ligne change cinq fois sur un Abonné avant qu'il se synchronise avec un serveur de publication, la ligne ne change qu'une seule fois sur le serveur de publication pour refléter le résultat final des modifications (c'est-à-dire la cinquième valeur).

Types de réplication

- Réplication hétérogène
 - On peut répliquer les données provenant de SQL Server vers d'autres systèmes de gestion de bases de données tels que IBM DB2, Oracle ou SyBase.

Les articles, les publications et les abonnements

- Les données répliquées sont organisées en articles et publications
- Article
 - Il peut représenter toute, ou une partie, de la table ou de l'objet de la base de données.
 - Il peut être filtrer horizontalement (les lignes) ou verticalement (les colonnes)

Les articles, les publications et les abonnements

- Publication
 - Peut contenir un ou plusieurs articles
 - Elle peut uniquement avoir des articles provenant de la même base de données,
 - il peut y avoir plusieurs publications au sein de la même base de données

Les articles, les publications et les abonnements

- Les abonnements
 - Ils sont créés pour les publications et non les articles.
 - On peut ajouter ou supprimer des abonnements
 - L'ajout des abonnements se fait au niveau du publieur.
 - La suppression d'un abonnement se fait au niveau des abonnés.

Les articles, les publications et les abonnements

- Les agents de réplication
 - Le processus de réplication est exécuté par les agent de répliations
 - Ces agent sont configurer lorsqu'on construit une solution de réplication
 - Quand on implémente un réplication, on doit spécifier dans quelle instance de SQL Server les agent de répliations vont fonctionner.

Les articles, les publications et les abonnements

- Les agents de réplication
 - Il existe différents types d'agents de réplication
 - SQL Server Agent
 - Snapshot Agent
 - Log Reader Agent
 - Queue Reader Agent
 - Distribution Agent
 - Merge Agent

Labo

- Implémentation d'une réplication
 - Créer une réplication
 - Créer un abonnement

SQL Server : Admin.