

Administration des Bases de Données sous SQL Server

2^{ème} Année - GL

A. ETTALBI

Module M4.2

**Intitulé : ADMINISTRATION DES BASES DE
DONNEES**

Responsable : A. ETTALBI

Volume horaire : 61 Heures

Période : Semestre S3

Composition de M4.2

2 Eléments de Module :

➤ **M4.2.1 : Administration des Bases de Données Oracle (38H)**

➤ **M4.2.2 : Administration des Bases de Données SQL Server (23H)**

Caractéristiques des Eléments de Module du Module M4.2

	Nom	Volume horaire	Coeff
M4.2.1	Administration des Bases de Données Oracle	Cours : 12H TP : 24H Evaluation : 2H	65% (Ecrit+TP)
M4.2.2	Administration des Bases de Données SQL Server	Cours : 8 TP : 14H Evaluation : 1H	35% (Ecrit+TP)

PLAN GENERAL

- 1. Vue d'ensemble de SQL Server**
- 2. Planification de l'installation de SQL Server**
- 3. Gestion des fichiers de BD**
- 4. Gestion de la sécurité**
- 5. Exécution de tâches administratives**
- 6. Sauvegarde de BD**
- 7. Restauration de BD**
- 8. Surveillance des performances de SQL Server**
- 9. Transfert des données**
- 10. Présentation de la réplication**

1. Vue d'ensemble de SQL-Server

1.1 Présentation de SQL Server

1.2 Intégration de SQL Server

1.3 Bases de données SQL Server

1.4 Sécurité de SQL Server

1.5 Utilisation de SQL Server

1.1 Présentation de SQL Server

- SGBD relationnel qui gère 2 types de BD :
OLAP (Aide à la décision. Ex: Entreprise)
OLTP (Transactions. Ex: Banque, Billetterie)
- Récupère les données à un état de cohérence antérieur en cas de panne

1.1 Présentation de SQL Server (suite)

Accès des applications clientes à la BD

- Scripts **Transact-SQL**
- **XML**
- **MDX** (MultiDimensional eXpression)
- Les API **OLE DB** (Object Linking and Embedding DB) et **ODBC** (Open DB Connectivity)
- **ADO** (ActiveX Data Object)
- **ADO MD** (ADO MultiDimensionnels)
- **MS English Query**

1.1 Présentation de SQL Server (suite)

Architecture Client-Serveur

- **Client** : Chargé de la logique d'entreprise et de la présentation des données
- **Serveur** : gère et répartit les ressources disponibles du serveur (mémoire, bande passante du réseau, opérations sur les disques) entre plusieurs demandes

1.1 Présentation de SQL Server (suite)

Composantes Client

➤ **Application cliente :**

- Exécute les instructions T-SQL et reçoit les résultats
- Ne connaît pas les protocoles réseaux utilisés pour la communication,

➤ **API de la BD (OLE DB, ODBC) :**

- Utilisent un fournisseur, un pilote ou une bibliothèque de liaisons dynamiques (DLL) pour transmettre les requêtes T-SQL et recevoir le résultat.

1.1 Présentation de SQL Server (suite)

Composantes Client (Suite)

➤ **Bibliothèque réseau cliente :**

- Composant logiciel de communication qui gère les connexions réseaux et le routage sur le client
- Assemble les requêtes et les résultats pour la transmission par le protocole réseau approprié

1.1 Présentation de SQL Server (suite)

Composantes Serveur

➤ Bibliothèques réseau Serveur :

- SQL-Server peut contrôler simultanément plusieurs BR.
- La **BR Client** doit **correspondre à l'une des BRS** pour que la communication puisse se faire correctement.
- **Communication** : Protocole (TCP/IP), Canaux nommés...

➤ Services Open Data :

- Gère les connexions réseaux de façon transparente et permanente (Ecoute des BRS, Transmission des requêtes client à SQL-Server, Renvoi des résultats au Client).

1.1 Présentation de SQL Server (suite)

Composantes Serveur (suite)

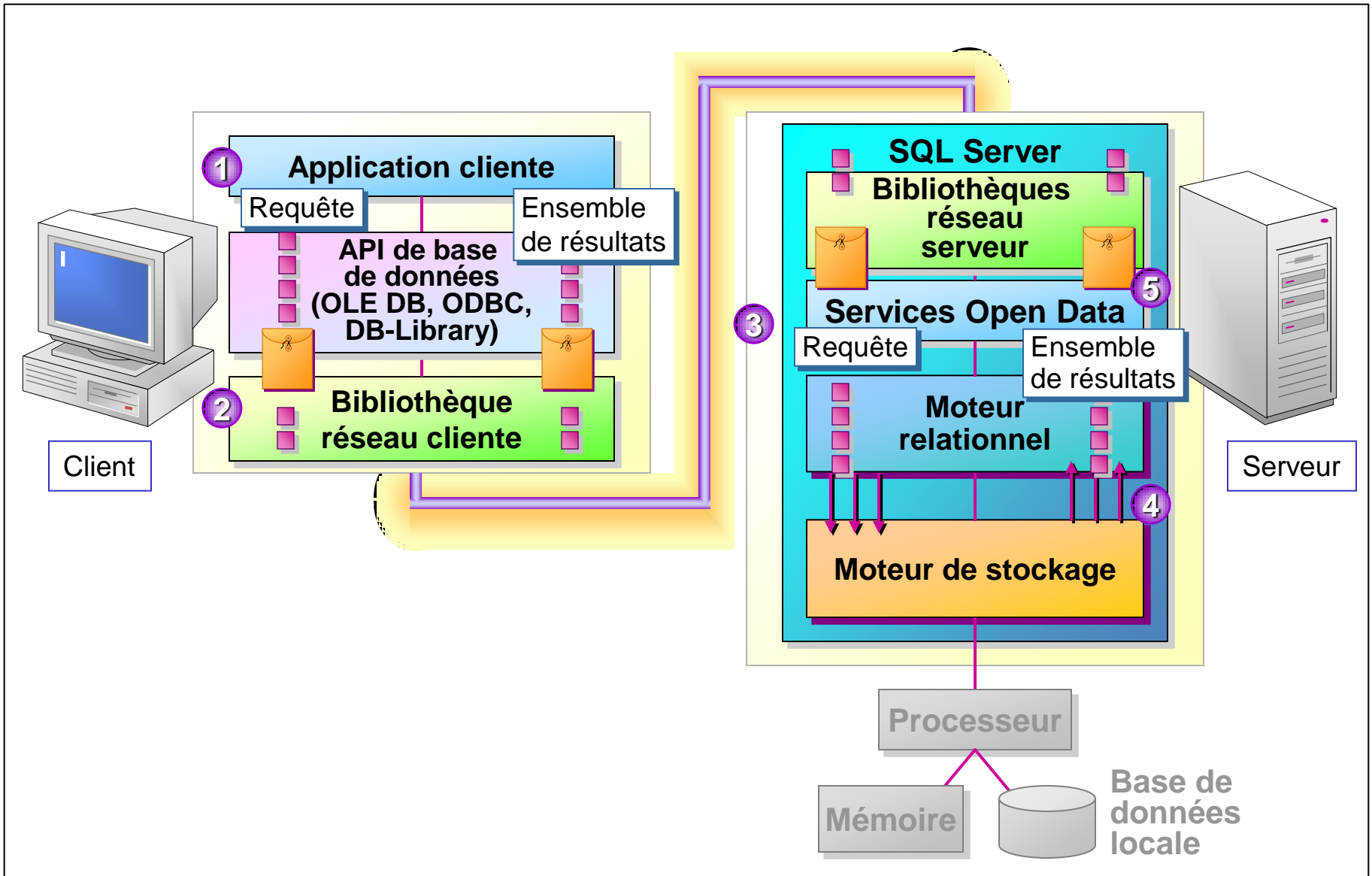
➤ Moteur relationnel :

- Analyse, optimise et exécute les instructions TSQL : LDD, LMD,
- Implémente la sécurité.

➤ Moteur de stockage :

- Gère les fichiers de la BD, les entrées/sorties physiques,
- Contrôle les conflits d'accès,
- Vérifie la cohérence des données, sauvegarde/restauration.

Processus de communication client-serveur



1.1 Présentation de SQL Server (suite)

Services SQL-Server

4 Services installés par défaut

➤ ***MSSQLServer*** :

- Est le Moteur de la BD,
- Gère les données
- Traite les requêtes et les transactions
- Assure l'intégrité et la cohérence des données.

➤ ***SQLServerAgent*** : *Gestion*

- des travaux (Création, planification,...)
- des alertes (Fin d'un travail, occurrence d'une erreur,...)
- des opérateurs (Notifications SQL-Server)

1.1 Présentation de SQL Server (suite)

Services SQL-Server : (suite)

➤ ***Microsoft Distributed Transaction Coordinator:***

- Gère les transactions distribuées (utilisant des sources de données différentes),

➤ ***Microsoft Search :***

- Prise en charge du texte intégral dans les requêtes
- Créer et gérer les index qui facilitent ces requêtes

1. Vue d'ensemble de SQL-Server

1.1 Présentation de SQL Server

1.2 Intégration de SQL Server

1.3 Bases de données SQL Server

1.4 Sécurité de SQL Server

1.5 Utilisation de SQL Server

1.2 Intégration de SQL Server

Intégration aux SE (côté Client)

Version SQL Server	Version Windows
SQL Server 2008 sauf CE	W-2000, W-NT, W-Me, W-95 et W-98
SQL Server 2000 CE	Windows CE

Intégration aux SE (côté Serveur)

Version SQL Server	Version Windows
SQL Server 2008	W-2000, W-NT, W-Me, W-95, W-98 et W-CE

1.2 Intégration de SQL Server

Intégration de SQL Server à W-2000

- **Active Directory** : Les serveurs et leurs attributs sont inscrits automatiquement dans le service d'annuaire Active Directory au démarrage du serveur,
- **Sécurité** : Avec un nom utilisateur et un mot de passe unique, il est possible d'accéder à la fois à SQL Server et à W-2000,

1.2 Intégration de SQL Server

Intégration de SQL Server à W-2000 (suite)

- **Prise en charge de plusieurs processeurs** : SQL Server prend en charge les fonctions de multitraitement symétrique (SMP) de W-2000 et tire parti de tous les processeurs,
- **Observateurs d'événements Microsoft** : SQL Server écrit des messages dans les journaux Système, Sécurité et Application de W-2000

1.2 Intégration de SQL Server

Intégration de SQL Server à W-2000 (suite)

- **W-2000 Component Services** : Component Services repose sur des extensions du modèle COM (Component Object Model) et de MS-Transaction Server. Il offre de meilleures performances dans la *gestion des threads, sécurité, gestion des transactions...*
- **Moniteur système W-2000** : SQL Server envoie au moniteur système W-2000 des informations sur les performances, ce qui permet de *surveiller les performances système de SQL Server*

1.2 Intégration de SQL Server

Intégration de SQL Server à W-2000 (suite)

- **Services Internet MS:** SQL Server *utilise IIS* pour que les navigateurs Internet puissent accéder à une BD SQL Server à l'aide du protocole HTTP.
- **Windows Clustering :** prend en charge la connexion de deux serveurs ou nœuds dans un cluster pour une disponibilité plus élevée. **SQL Server fonctionne avec WC** pour basculer automatiquement vers le nœud secondaire en cas de défaillance du nœud principal.

1. Vue d'ensemble de SQL-Server

1.1 Présentation de SQL Server

1.2 Intégration de SQL Server

1.3 Bases de données SQL Server

1.4 Sécurité de SQL Server

1.5 Utilisation de SQL Server

1.3 Bases de données SQL Server

Chaque ordinateur SQL Server possède 2 types de bases de données :

- Bases de données **système** : stockent les informations globales relatives à SQL Server,
- Bases de données **utilisateurs** : créées par l'utilisateur.

1.3 Bases de données SQL Server

A l'installation de SQL Server, le programme d'installation **crée** des BD système et des exemples de BD utilisateur.

La BD **distribution** est installée lorsqu'on configure SQL Server pour des activités de réplication.

1.3 Bases de données SQL Server

Nom BD	Description
master	contrôle les BD utilisateur et le fonctionnement global de SQL Server
model	offre un modèle pour les nouvelles BD utilisateur
tempdb	offre une zone de stockage pour les tables temporaires
msdb	offre une zone de stockage pour les informations de programmation et l'historique des travaux
distribution	stocke les données relatives à l'historique et aux transactions utilisées dans la réplication
pubs	propose un exemple de BD comme outil d'apprentissage
Northwind	propose un exemple de BD comme outil d'apprentissage
Utilisateur1	Identifie une BD définie par l'utilisateur

1.3 Bases de données SQL Server

Objets de base de données :

Table, Type de données, Contrainte, Valeurs par défaut, Règle, Index, Vue, Fonction définie par l'utilisateur, Procédure stockée, Déclencheur

Références aux objets de base de données :

Nom complet :

serveur.base_données.propriétaire.objet

Noms incomplets :

base_données.propriétaire.objet

propriétaire.objet

objet

1.3 Bases de données SQL Server

Tables système : stockent les infos sur la configuration ainsi que des définitions de toutes les BD et de tous les objets de BD dans l'instance SQL Server

Catalogue de BD : tables système stocke les informations relatives à une BD spécifique quelconque (y compris **master**)

Catalogue système: (uniquement dans la BD **master**) tables système qui stockent les métadonnées relatives au système entier et à toutes les autres BD

1.3 Bases de données SQL Server

Table système	Nom BD	Fonction
syslogins	master	Contient une ligne pour chaque compte d'ouverture de session
sysmessages	master	Contient une ligne pour chaque avertissement ou erreur
sysdatabases	master	Contient une ligne pour chaque base de données
sysusers	toutes	Contient une ligne pour chaque utilisateur W-2000, groupe W-2000, utilisateur SQL Server ou rôle SQL Server
sysobjects	toutes	Contient une ligne pour chaque objet d'une BD

1.3 Bases de données SQL Server

Extraction des métadonnées :

- Procédures stockées système
- Fonctions système et de métadonnées.

Exemples de Procédures stockées système

Procédure	Description
sp_help [objet]	Fournit des informations sur l'objet de BD spécifié
sp_helpdb [BD]	Fournit des informations sur la BD spécifiée
sp_helpindex [Table]	Fournit des informations sur l'index de la table spécifiée

1.3 Bases de données SQL Server

Exemple :

EXEC sp_help Employees

➔ Exécution d'une procédure stockée système afin d'obtenir des informations sur la table Employees

1.3 Bases de données SQL Server

Exemples de Fonctions système

Fonction	Paramètre transmis	Résultats
DB_ID	Nom	Identificateur de la BD
USER_NAME	ID	Nom de l'utilisateur
COL_LENGTH	Colonne	Largeur de la colonne
STATS_DATE	Index	Date de la dernière MAJ des statistiques de l'index
DATALENGTH	Type de données	Longueur réelle d'une expression de tout type de données

1.3 Bases de données SQL Server

Exemple :

SELECT USER_NAME(10)

➔ Appel d'une fonction système afin d'extraire le nom d'utilisateur correspondant à l'identificateur d'utilisateur 10

1.3 Bases de données SQL Server

Exemples de Vues de schémas d'information

Vue de schémas d'informations	Description
INFORMATION_SCHEMA.TABLES	Liste des tables de la BD
INFORMATION_SCHEMA.COLUMNS	Information sur les colonnes définies dans la BD
INFORMATION_SCHEMA.TABLES_PRIVILEGES	Informations de sécurité pour les tables de la BD

1.3 Bases de données SQL Server

Exemples:

SELECT * FROM INFORMATION_SCHEMA.TABLES

➔ interroge une vue de schémas d'informations
afin d'extraire la liste des tables d'une BD

1. Vue d'ensemble de SQL-Server

1.1 Présentation de SQL Server

1.2 Intégration de SQL Server

1.3 Bases de données SQL Server

1.4 Sécurité de SQL Server

1.5 Utilisation de SQL Server

1.4 Sécurité de SQL Server

- Authentification des comptes de connexion
- Rôles et comptes d'utilisateur de BD
- Types de rôles
- Validation des autorisations

1.4 Sécurité de SQL Server

Authentification des comptes de connexion

Authentification Windows :

- compte ou groupe Windows 2000

Authentification SQL Server :

- compte d'ouverture de session SQL Server et mot de passe

1.4 Sécurité de SQL Server

Authentification des comptes de connexion

Modes d'authentification (spécifiés par l'administrateur système):

- **Mode d'authentification Windows :**
pas de compte SQL Server
- **Mode mixte :**
authentification W2000 ou SQL Server

1.4 Sécurité de SQL Server

Rôles et comptes d'utilisateur de BD

Comptes d'utilisateur de BD : sont des utilisateurs ou des groupes W-2000 ou des comptes d'ouverture de session SQL Server. Ils sont spécifiques à une BD

Rôles : permettent de regrouper des utilisateurs au sein d'une entité unique à laquelle on peut attribuer des autorisations

1.4 Sécurité de SQL Server

Types de rôles

Rôles fixes de serveur :

- regrouper des privilèges d'administrateur au niveau du serveur

Rôles fixes de BD :

- regrouper des privilèges d'administrateur au niveau de la BD

Rôles de BD définis par l'utilisateur :

- rôles créés par l'utilisateur

Rôles fixes de serveur

Rôle	Autorisation
dbcreator	Création et modification des BD
diskadmin	Gestion des fichiers physiques
processadmin	Gestion des processus SQL Server
securityadmin	Gestion et audit des noms de connexion serveur
serveradmin	Configuration des paramètres de serveur
setupadmin	Installation de la réplication
sysadmin	Exécution de toutes les activités
bulkadmin	Exécution de l'instruction BULK INSERT

Rôles fixes de base de données

Rôle	Autorisation
public	Maintien de toutes les autorisations par défaut pour les utilisateurs d'une BD
db_owner	Exécution de toutes les activités de tous les rôles de BD
db_accessadmin	Ajout ou suppression d'utilisateurs, de groupes et de rôles de BD
db_ddladmin	Ajout, modification ou suppression d'objets de BD
db_securityadmin	Attribution d'autorisations sur les instructions et les objets
db_backupoperator	Sauvegarde des BD
db_datareader	Lecture de données de toutes les tables
db_datawriter	Ajout, modification, suppression de données de toutes les tables
db_denydatareader	Interdiction de lecture de données dans toutes les tables
db_denydatawriter	Interdiction de modification des données dans toutes les tables

1. Vue d'ensemble de SQL-Server

1.1 Présentation de SQL Server

1.2 Intégration de SQL Server

1.3 Bases de données SQL Server

1.4 Sécurité de SQL Server

1.5 Utilisation de SQL Server

1.5 Utilisation de SQL Server

- Administration d'une BD SQL Server
- Implémentation d'une BD SQL Server
- Choix d'une architecture d'application pour SQL Server
- Conception d'application à l'aide d'interfaces API de BD

Administration d'une BD SQL-Server

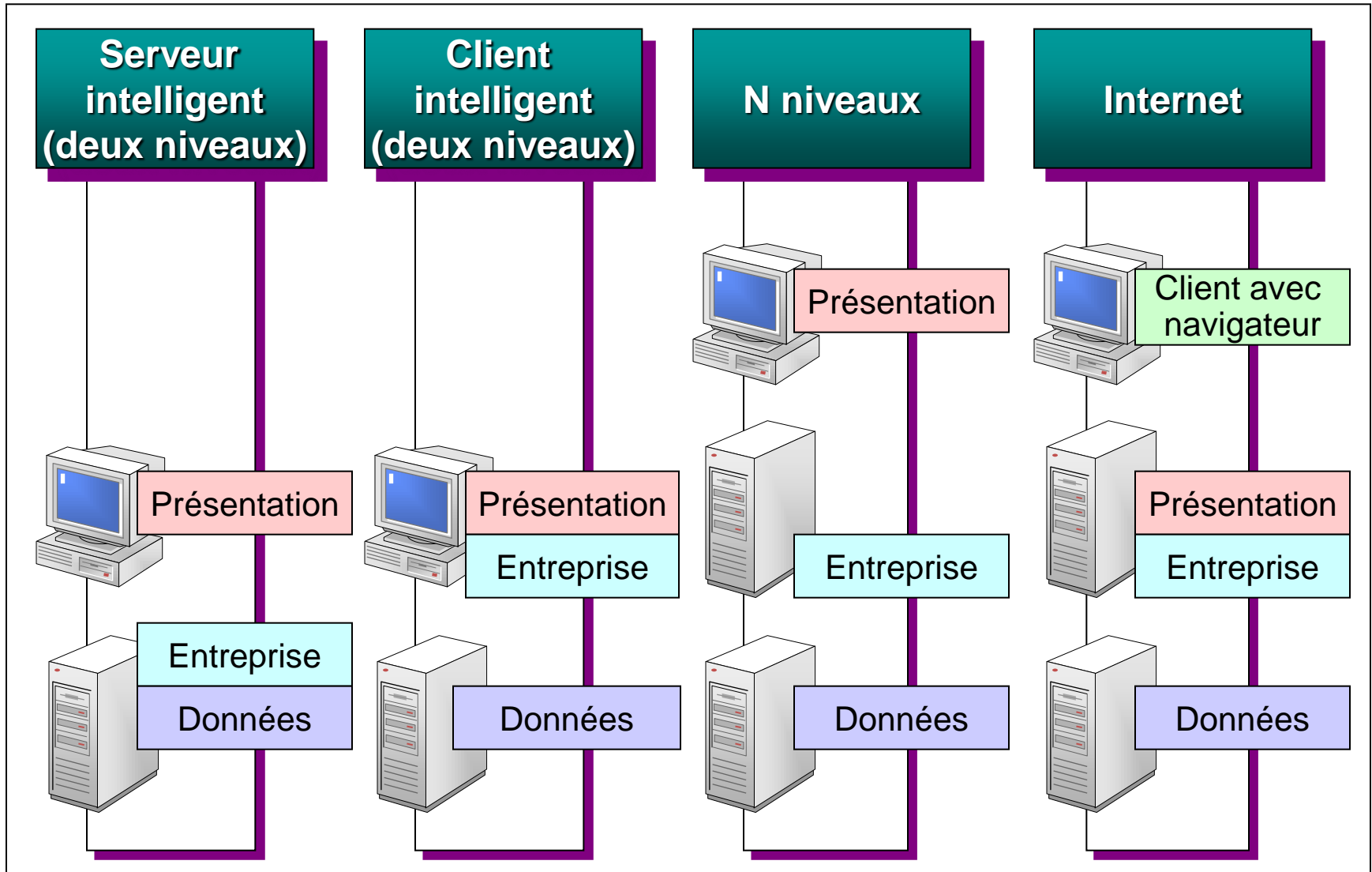
Tâches courantes d'administration :

- Installation, configuration, sécurisation,
- Création de BD (allocation d'espace disque à la BD et au journal, transfert depuis/vers la BD, définition et implémentation de la sécurité, création des travaux automatisés, configuration de la réplication),
- Gestion des activités récurrentes comme l'importation et l'exportation des données, la sauvegarde et la restauration de la BD, la surveillance et l'optimisation de la BD.

Implémentation d'une BD SQL-Server

- Concevoir la BD
- Créer la BD et ses objets
- Tester et optimiser l'application de la BD
- Planifier le déploiement

Conception d'une architecture d'application pour SQL Server



Conception d'une architecture d'application pour SQL Server

Couche logique	Description
<i>Présentation</i>	Comprend la logique de présentation des données et de l'application aux utilisateurs. Elle est toujours implémentée sur un ordinateur client
<i>Entreprise</i>	Comprend la logique de l'application et les règles d'entreprise. SQL Server peut intervenir au niveau de cette couche
<i>Données</i>	Comprend la définition de la BD, la logique d'intégrité des données, les procédures stockées et toutes les autres opérations associées aux données. SQL Server intervient au niveau de cette couche

Conception d'applications à l'aide d'interfaces API de BD

- On peut développer une application de BD qui accède à SQL Server par l'intermédiaire d'une API,
- Une API de BD se compose de 2 parties
 - Des instructions T-SQL transmises à la BD
 - Un ensemble de fonctions ou d'interfaces et de méthodes orientées objets servant à envoyer des instructions T-SQL à la BD et à traiter les résultats renvoyés par la BD

Exemple d'interface API de BD

- **OLE DB** (Object Linking and Embedding DB):
 - API de type **COM** (Component Object Model)
 - Bibliothèques d'interfaces COM permettant un **accès universel à diverses sources** de données
- SQL Server inclut un **fournisseur OLE DB natif** qui prend en charge des applications ou des API utilisant OLE DB comme ADO
- Grâce au fournisseur natif, SQL Server prend également en charge **les objets et les composants utilisant OLE DB** tels que ActiveX, ADO et MS .NET Enterprise Servers

2. Planification de l'installation de SQL Server

2.1 Configuration matérielle minimale requise

2.2 Optimisation du matériel pour SQL Server

2.3 Système RAID

2.4 Choix à l'installation du logiciel

2.5 Modes de licence

2.1 Configuration matérielle minimale requise

- **Processeur** : Intel et compatible P166 ou supérieur
- **Mémoire** : **256 Mo** min pour toutes les versions serveur de W-2000, et **64 Mo** min pour toutes les versions serveur de W-NT 4.0
- **Espace disque** :
250 Mo (installation min) **100 Mo** (outils clients)
270 Mo (installation complète) **50 à 130 Mo** (Analysis Services) **80 Mo** (English Query)

2. Planification de l'installation de SQL Server

2.1 Configuration matérielle minimale requise

2.2 Optimisation du matériel pour SQL Server

2.3 Système RAID

2.4 Choix à l'installation du logiciel

2.5 Modes de licence

2.2 Optimisation du matériel pour SQL Server

- **Processeur :**

Augmentation du nombre ou de la rapidité des processeurs → traitement plus rapide des requêtes.

Répartition sur plusieurs serveurs nécessaires pour des BD très volumineuses

- **Sous-système disque :**

Système RAID matériel et logiciel → augmenter les performances, la fiabilité, le stockage et la capacité

2. Planification de l'installation de SQL Server

2.1 Configuration matérielle minimale requise

2.2 Optimisation du matériel pour SQL Server

2.3 Système RAID

2.4 Choix à l'installation du logiciel

2.5 Modes de licence

2.3 Système RAID

Niveau RAID	Description	Avantages/Inconvénient
RAID 0	Entrelacement	Offre de bonnes performances de lecture et d'écriture mais pas de disponibilité
RAID 1	Mise en miroir et duplexage du disque	Offre une très bonne redondance et de bonnes performances de lecture et d'écriture
RAID 5	Entrelacement avec parité	Offre une redondance et des performances de lecture excellentes mais des performances d'écriture moyennes
RAID 10 / RAID 01	Mise en miroir du disque avec Entrelacement	Offre une redondance et des performances maximales mais nécessite un plus grand nombre de disques

2. Planification de l'installation de SQL Server

2.1 Configuration matérielle minimale requise

2.2 Optimisation du matériel pour SQL Server

2.3 Système RAID

2.4 Choix à l'installation du logiciel

2.5 Modes de licence

2.4 Choix à l'installation du logiciel

- Mode de licence
- Contexte de sécurité des comptes de service
- Utilisation d'instances multiples et d'instances nommées de SQL Server
- Un mécanisme de sécurité
- Classements et règles de tri SQL Server
- Bibliothèques réseau

2. Planification de l'installation de SQL Server

2.1 Configuration matérielle minimale requise

2.2 Optimisation du matériel pour SQL Server

2.3 Système RAID

2.4 Choix à l'installation du logiciel

2.5 Modes de licence

2.5 Modes de licence

➤ **Licence par processeur**

- **Licence** acquise pour **chaque processeur** exécutant SQL Server,
- **Nombre illimité** d'utilisateurs autorisés à se connecter à partir d'un réseau local ou d'un réseau étendu.

➤ **Licence par siège**

- **Licence par siège acquise** pour chaque serveur SQL Server et licence d'accès client acquise pour chaque station de travail qui se connecte à une instance de SQL Server.

3. Gestion des fichiers de BD

3.1 Mode de stockage des données

3.2 Processus de création d'une BD

3.3 Définition des options lors de la création d'une BD

3.4 Gestion de la croissance des fichiers journaux et de données

3.5 Compactage automatique d'une BD

3.6 Compactage manuel d'une BD

3.7 Compactage des fichiers journaux de transactions

3.8 Suppression d'une BD

3.9 Emplacement des fichiers et journaux de BD

3.10 Optimisation d'une BD à l'aide d'une solution RAID matérielle

3.11 Optimisation d'une BD à l'aide de groupes de fichiers

3.12 Planification de la capacité

3.1 Mode stockage des données

- Toutes les BD possèdent un fichier de données principal (**.mdf**) et un ou plusieurs fichiers journaux de transactions (**.ldf**)
- Une BD peut aussi comporter des fichiers de données secondaires (**.ndf**)
- Ces fichiers physiques ont à la fois des **noms de fichiers de SE** et des **noms de fichiers logiques** qui peuvent être utilisés dans les instructions T-SQL
- L'emplacement par défaut est :
C:\Program Files\Microsoft SQL Server\MSSQL\Data

3.1 Mode stockage des données (suite)

- Lorsqu'on crée une BD, une copie de la BD **model** est copiée dans la BD
- les données sont stockées dans des blocs d'espace disque contigus de 8 KO appelés **pages**
 - ➔ Une BD peut stocker 128 pages par Mo
- La quantité maximale de données présente sur une seule ligne est donc de 8060 octets
- Les tables et index sont stockés dans des extensions de 8 pages contiguës (64 Ko)
 - ➔ Une BD comporte 16 extensions par Mo
- Les fichiers journaux de transactions contiennent toutes les informations requises pour la restauration de la BD en cas de panne du système.

3. Gestion des fichiers de BD

3.1 Mode de stockage des données

3.2 Processus de création d'une BD

3.3 Définition des options lors de la création d'une BD

3.4 Gestion de la croissance des fichiers journaux et de données

3.5 Compactage automatique d'une BD

3.6 Compactage manuel d'une BD

3.7 Compactage des fichiers journaux de transactions

3.8 Suppression d'une BD

3.9 Emplacement des fichiers et journaux de BD

3.10 Optimisation d'une BD à l'aide d'une solution RAID matérielle

3.11 Optimisation d'une BD à l'aide de groupes de fichiers

3.12 Planification de la capacité

3.2 Processus de création d'une BD

Lorsqu'on crée une BD, SQL Server effectue les tâches suivantes :

- Crée un **fichier de données** et un **journal de transactions** pour cette BD,
- Requier que le propriétaire et le créateur de la nouvelle BD **soient autorisés à utiliser la BD master** car toutes les informations relatives à chaque BD dans SQL Server sont enregistrées dans les tables **sysdatabases** et **sysaltfiles** de la BD master.

3.2 Processus de création d'une BD (suite)

- SQL Server permet aussi à l'utilisateur de définir le **nom et les propriétaires de la BD** et l'**emplacement** des fichiers de cette BD.
- Il utilise une copie de la BD **model** pour initialiser la BD et ses métadonnées.
- Il remplit ensuite le reste de la BD avec des pages vides, à l'exception des pages contenant des données sur la manière dont l'espace est utilisé dans la BD

3. Gestion des fichiers de BD

3.1 Mode de stockage des données

3.2 Processus de création d'une BD

3.3 Définition des options lors de la création d'une BD

3.4 Gestion de la croissance des fichiers journaux et de données

3.5 Compactage automatique d'une BD

3.6 Compactage manuel d'une BD

3.7 Compactage des fichiers journaux de transactions

3.8 Suppression d'une BD

3.9 Emplacement des fichiers et journaux de BD

3.10 Optimisation d'une BD à l'aide d'une solution RAID matérielle

3.11 Optimisation d'une BD à l'aide de groupes de fichiers

3.12 Planification de la capacité

3.3 Définition des options lors de la création d'une base de données

- Fichier principal (.**mdf**)
- Fichiers secondaires (.**ndf**)
- Fichier Journal des transactions (.**ldf**) 10 à 15% des fichiers de la BD
- **Nom logique** et emplacement physique de fichier
- Taille (minimum 512 KO)
- Croissance des fichiers
- Taille maximale (en MO ou en pourcentage : 10% par défaut)
- Classement (par défaut une BD hérite du classement de l'instance de SQL Server dans laquelle la BD a été créée)

3. Gestion des fichiers de BD

3.1 Mode de stockage des données

3.2 Processus de création d'une BD

3.3 Définition des options lors de la création d'une BD

3.4 Gestion de la croissance des fichiers journaux et de données

3.5 Compactage automatique d'une BD

3.6 Compactage manuel d'une BD

3.7 Compactage des fichiers journaux de transactions

3.8 Suppression d'une BD

3.9 Emplacement des fichiers et journaux de BD

3.10 Optimisation d'une BD à l'aide d'une solution RAID matérielle

3.11 Optimisation d'une BD à l'aide de groupes de fichiers

3.12 Planification de la capacité

3.4 Gestion de la croissance des fichiers journaux et de données

- Utilisation de la croissance automatique des fichiers
 - **Spécification de l'espace alloué**, de la **taille maximale** et de **l'incrément** de croissance de chaque fichier
 - **Optimisation de performance** en allouant suffisamment d'espace, en définissant une taille maximale et des incréments de croissance
- Croissance manuelle des données et des fichiers journaux et transactions
- **Choix entre** croissance automatique ou manuelle de la taille des fichiers
- **Création** de fichiers de données et de fichiers journaux secondaires

Choix entre croissance automatique ou manuelle de la taille des fichiers

- Dans un environnement de **production important** (espace mémoire important pour la BD) : allouer suffisamment d'espace pour les fichiers de données et envisager d'augmenter la taille **manuellement** si nécessaire,
- Dans un environnement de bureau ou un environnement de **production restreint** : configurer les fichiers de BD pour qu'ils **augmentent automatiquement**, ce qui réduit les tâches d'administration.

3. Gestion des fichiers de BD

3.1 Mode de stockage des données

3.2 Processus de création d'une BD

3.3 Définition des options lors de la création d'une BD

3.4 Gestion de la croissance des fichiers journaux et de données

3.5 Compactage automatique d'une BD

3.6 Compactage manuel d'une BD

3.7 Compactage des fichiers journaux de transactions

3.8 Suppression d'une BD

3.9 Emplacement des fichiers et journaux de BD

3.10 Optimisation d'une BD à l'aide d'une solution RAID matérielle

3.11 Optimisation d'une BD à l'aide de groupes de fichiers

3.12 Planification de la capacité

3.5 Compactage automatique BD

- **Nécessaire** si la quantité **d'espace mémoire** est trop **importante** ou les besoins en espace diminuent
- On peut **configurer** une BD ou un fichier de BD pour qu'il soit **automatiquement compacté** soit :
 - En spécifiant les options de compactage automatique dans SQL Server Enterprise Manager
 - exécuter l'instruction ALTER DATABASE AUTO_SHRINK
 - exécuter la procédure stockée système **sp_dboption**
- **Par défaut**, l'option de compactage automatique des fichiers est **désactivée** dans toutes les éditions de SQL Server sauf SQL Server Desktop Edition

Activités de SQL Server au cours du compactage automatique

- Il réduit les fichiers de données et journaux de transactions lorsque l'espace inutilisé dans le fichier dépasse 25%
- Il n'est pas possible de configurer le pourcentage d'espace libre à supprimer. SQL Server supprime autant d'espace libre que possible
- Le journal des transactions n'est réduit que s'il ne contient aucune partie active du journal logique nécessaire à la restauration de la BD
- Exécute cette tâche en arrière-plan

3. Gestion des fichiers de BD

3.1 Mode de stockage des données

3.2 Processus de création d'une BD

3.3 Définition des options lors de la création d'une BD

3.4 Gestion de la croissance des fichiers journaux et de données

3.5 Compactage automatique d'une BD

3.6 Compactage manuel d'une BD

3.7 Compactage des fichiers journaux de transactions

3.8 Suppression d'une BD

3.9 Emplacement des fichiers et journaux de BD

3.10 Optimisation d'une BD à l'aide d'une solution RAID matérielle

3.11 Optimisation d'une BD à l'aide de groupes de fichiers

3.12 Planification de la capacité

3.6 Compactage manuel d'une BD

- **Intérêt** : choisir le moment de lancer cette tâche
- **Méthodes de compactage** :
 - utiliser SQL Server Entreprise Manager
 - exécuter ***BDCC SHRINKDATABASE*** ou ***DBCC SHRINKFILE***
- On peut alors :
 - **réduire les fichiers** de données et de journal des transactions en groupe ou individuellement
 - réduire individuellement les fichiers de données et journaux de transactions dont la **taille est inférieure** à leur taille de création d'origine

3. Gestion des fichiers de BD

3.1 Mode de stockage des données

3.2 Processus de création d'une BD

3.3 Définition des options lors de la création d'une BD

3.4 Gestion de la croissance des fichiers journaux et de données

3.5 Compactage automatique d'une BD

3.6 Compactage manuel d'une BD

3.7 Compactage des fichiers journaux de transactions

3.8 Suppression d'une BD

3.9 Emplacement des fichiers et journaux de BD

3.10 Optimisation d'une BD à l'aide d'une solution RAID matérielle

3.11 Optimisation d'une BD à l'aide de groupes de fichiers

3.12 Planification de la capacité

3.7 Compactage des fichiers journaux de transactions

Dans ce cas, SQL Server :

- **réduit les parties inactives** du journal des transactions qui dépassent la taille souhaitée
- **si cela ne suffit pas** à ramener le journal des transactions à la taille souhaitée, il :
 - **renvoie un message d'erreur** indiquant qu'une partie du journal actif dépasse la taille souhaitée,
 - **indique la procédure à suivre** pour déplacer la partie active du journal de la fin du fichier journal des transactions.

3. Gestion des fichiers de BD

3.1 Mode de stockage des données

3.2 Processus de création d'une BD

3.3 Définition des options lors de la création d'une BD

3.4 Gestion de la croissance des fichiers journaux et de données

3.5 Compactage automatique d'une BD

3.6 Compactage manuel d'une BD

3.7 Compactage des fichiers journaux de transactions

3.8 Suppression d'une BD

3.9 Emplacement des fichiers et journaux de BD

3.10 Optimisation d'une BD à l'aide d'une solution RAID matérielle

3.11 Optimisation d'une BD à l'aide de groupes de fichiers

3.12 Planification de la capacité

3.8 Suppression d'une BD

- Par ***DROP DATABASE*** ou sous SQL Server Enterprise Manager
- **Limites :**
 - si elle est **en cours** d'utilisation
 - si un utilisateur y est **connecté**
 - si elle effectue la **publication** de l'une de ses tables dans le cadre de la réplication
 - s'il s'agit d'une base de données **système**

3. Gestion des fichiers de BD

3.1 Mode de stockage des données

3.2 Processus de création d'une BD

3.3 Définition des options lors de la création d'une BD

3.4 Gestion de la croissance des fichiers journaux et de données

3.5 Compactage automatique d'une BD

3.6 Compactage manuel d'une BD

3.7 Compactage des fichiers journaux de transactions

3.8 Suppression d'une BD

3.9 Emplacement des fichiers et journaux de BD

3.10 Optimisation d'une BD à l'aide d'une solution RAID matérielle

3.11 Optimisation d'une BD à l'aide de groupes de fichiers

3.12 Planification de la capacité

3.9 Emplacement des fichiers et journaux de BD

- **Performance** : vitesse des opérations d'E/S sur le disque
- **Tolérance de panne** : capacité du système à continuer de fonctionner sans perte de données lorsqu'une partie de celui-ci est défaillante
- L'utilisateur peut gérer l'espace disque pour améliorer les performance et implémenter la tolérance de panne.

Règle générale : utiliser des disques au format NTFS ayant une unité d'allocation de 64Ko (Pas de volumes compressés)

3.9 Emplacement des fichiers et journaux de BD (suite)

- **Répartir** autant de données que possible sur le plus grand nombre de disques physiques possible.
- SQL Server peut exécuter :
 - des **recherches parallèles** dans les données si l'ordinateur comporte plusieurs processeurs et plusieurs disques
 - plusieurs **recherches parallèles pour une seule table** si le groupe de fichiers de la table contient plusieurs fichiers

Pour répartir les données uniformément sur plusieurs disques, utiliser la solution **RAID**, puis des **groupes de fichiers** définis par l'utilisateur.

3.9 Emplacement des fichiers et journaux de BD (suite)

- Créer le journal de transactions sur un **disque séparé** ou utiliser une **solution RAID**
- Pour des performances optimales, placer la BD **tempdb** sur un sous-système d'E/S **rapide distinct** de celui utilisé pour les BD utilisateur,
- On peut aussi utiliser une **solution RAID** pour **distribuer** la BD **tempdb** sur plusieurs disques pour améliorer les performances.

3. Gestion des fichiers de BD

3.1 Mode de stockage des données

3.2 Processus de création d'une BD

3.3 Définition des options lors de la création d'une BD

3.4 Gestion de la croissance des fichiers journaux et de données

3.5 Compactage automatique d'une BD

3.6 Compactage manuel d'une BD

3.7 Compactage des fichiers journaux de transactions

3.8 Suppression d'une BD

3.9 Emplacement des fichiers et journaux de BD

3.10 Optimisation d'une BD à l'aide d'une solution RAID matérielle

3.11 Optimisation d'une BD à l'aide de groupes de fichiers

3.12 Planification de la capacité

3.10 Optimisation d'une BD à l'aide d'une solution RAID matérielle

- Pour des performances optimale, **opter pour une solution RAID matérielle** plutôt que pour une solution RAID reposant sur le SE
- Cette dernière solution **consomme des cycles CPU** au détriment des autres tâches système.
- Une solution RAID matérielle permet également de **remplacer un lecteur défectueux sans arrêter le système**

Types de solutions RAID

- **RAID 1: Disques miroirs** et duplexage de disques pour assurer la redondance du journal de transactions
- **RAID 5: Entrelacement avec parité** afin de combiner performance et redondance des fichiers de données et des journaux de transactions.
- **RAID 10 (ou RAID 1 + RAID 0):** pour assurer des performances maximales au niveau des fichiers de données.

3. Gestion des fichiers de BD

3.1 Mode de stockage des données

3.2 Processus de création d'une BD

3.3 Définition des options lors de la création d'une BD

3.4 Gestion de la croissance des fichiers journaux et de données

3.5 Compactage automatique d'une BD

3.6 Compactage manuel d'une BD

3.7 Compactage des fichiers journaux de transactions

3.8 Suppression d'une BD

3.9 Emplacement des fichiers et journaux de BD

3.10 Optimisation d'une BD à l'aide d'une solution RAID matérielle

3.11 Optimisation d'une BD à l'aide de groupes de fichiers

3.12 Planification de la capacité

3.11 Optimisation d'une BD à l'aide de groupes de fichiers

- Les groupes de fichiers améliorent les performances en **répartissant les données sur plusieurs disques** et en utilisant des **threads parallèles** pour le traitement des requêtes
- Ils facilitent aussi la **maintenance** de la BD
- Si la configuration matérielle comporte **plusieurs disques**, on peut placer des objets et des fichiers spécifiques sur certains disques en **regroupant** les fichiers de la BD dans **un ou plusieurs groupes de fichiers**

Types de groupes de fichiers

SQL Server possède un **groupe de fichiers principal** et peut également comprendre des groupes de fichiers définis par l'utilisateur.

- **Groupe de fichiers principal** : contient le fichier de données principal avec les tables système
- **Groupe de fichiers défini par l'utilisateur** : se compose de fichiers de données qui sont regroupés pour des besoins d'allocation et d'administration

Création de groupes de fichiers définis par l'utilisateur

- On peut créer un groupe de fichiers défini par l'utilisateur **lors de la création** de la base de données **ou ultérieurement**. (**Create Database**, **Alter Database** ou sous SQL Server Entreprise Manager)
- On peut choisir le **groupe de fichiers par défaut** (qui va contenir les pages de toutes les tables et tous les index pour lesquels aucun groupe de fichiers n'a été spécifié au moment de la création)

Création de groupes de fichiers définis par l'utilisateur (suite)

- SQL Server désigne un groupe de fichiers comme groupe de fichiers par défaut
- Le groupe de fichiers par défaut est défini comme groupe de fichiers principal au moment de la création de la BD, sauf indication contraire

Taille du groupe de fichiers par défaut

- Si le groupe de fichiers par défaut reste le groupe de fichiers principal, on doit lui donner une taille appropriée ou le configurer pour que sa taille **augmente automatiquement**,
- Si le groupe de fichiers principal est **saturé**, il sera **impossible d'ajouter** des informations dans les **tables système**,
- Par contre, lorsqu'un groupe de fichiers défini par l'utilisateur est saturé, seuls les fichiers utilisateurs spécifiquement alloués à ce groupe de fichiers sont concernés.

Affichage des informations sur les groupes de fichiers

- A l'aide de SQL Server Enterprise Manager ou par des procédures stockées système :

sp_helpfile *[@filename=]'nom'* : renvoie les noms physiques et les attributs des fichiers associés à la BD en cours

sp_helpfilegroup *[groupe_fichier]* : renvoie les noms et les attributs des groupes de fichiers associés à la BD en cours

Utilisation de groupes de fichiers pour améliorer les performances

Répartition de la charge des données sur plusieurs disques

- Lorsqu'on crée **une table**, on peut l'attribuer à un **groupe de fichiers** défini par l'utilisateur
- Les groupes de fichiers obéissent à une stratégie de **remplissage proportionnel** réparti sur tous les fichiers du groupe de fichiers
- Chaque fichier est **rempli en parallèle**
- Chaque fichier est placé physiquement sur **un ou plusieurs disques**
- SQL Server conserve un plan de fichier qui associe chaque **objet de la BD avec son emplacement** sur le disque

Utilisation de groupes de fichiers pour améliorer les performances (suite)

Utilisation de threads parallèles pour améliorer l'accès aux données

- A chaque **accès séquentiel** à une table, le système crée **un thread** séparé pour chaque fichier en parallèle,
- Lorsque le système exécute une recherche de table dans un groupe de fichiers composé de 4 fichiers, il utilise 4 threads séparés pour **lire les données en parallèle**

Utilisation de groupes de fichiers pour améliorer la maintenance

- Sauvegarder ou restaurer **des fichiers** ou des groupes de fichiers plutôt que sauvegarder ou restaurer **toute** la BD,
- **Regrouper** les tables et les index ayant les **mêmes exigences** en matière de maintenance dans les **mêmes groupes** de fichiers,
- **Attribuer** une table demandant un niveau de **maintenance élevé** à son **propre groupe** de fichiers.

Optimisation d'une BD en utilisant des groupes de fichiers avec une solution RAID matérielle

- On peut combiner des groupes de fichiers avec des solutions RAID matérielles :
 - **configurer l'entrelacement matériel,**
 - utiliser des **groupes de fichiers** pour **répartir les données** sur plusieurs agrégats par bandes
- Cette solution permet un **accès parallèle** aux données à l'aide de threads séparés pour chaque fichier et **répartit la charge** sur plusieurs disques.
- Cette approche crée un **regroupement logique unique**, donc elle est facile à configurer et à gérer pour un Administrateur Système ou de BD.

3. Gestion des fichiers de BD

3.1 Mode de stockage des données

3.2 Processus de création d'une BD

3.3 Définition des options lors de la création d'une BD

3.4 Gestion de la croissance des fichiers journaux et de données

3.5 Compactage automatique d'une BD

3.6 Compactage manuel d'une BD

3.7 Compactage des fichiers journaux de transactions

3.8 Suppression d'une BD

3.9 Emplacement des fichiers et journaux de BD

3.10 Optimisation d'une BD à l'aide d'une solution RAID matérielle

3.11 Optimisation d'une BD à l'aide de groupes de fichiers

3.12 Planification de la capacité

3.12 Planification de la capacité

- Est l'une des tâches principales d'un Administrateur système qui consiste à **allouer, gérer et contrôler** les **besoins d'espace** et de stockage de SQL Server et de ses BD,
- L'évaluation de l'espace requis par une BD peut **aider à planifier** l'agencement du stockage et à déterminer la configuration matérielle requise.

Evaluation de la taille d'une BD

- Lors de la planification d'une BD, on **configure** la **structure logique**,
- Sous-jacents à cette structure se trouvent des **fichiers et objets physiques** qui occupent de l'espace disque : journal des transactions, tables et index qui forment les fichiers de données,
- Lorsqu'on crée une BD, SQL Server **crée une copie** de la BD **model**, notamment des tables systèmes qui contiennent des informations sur les fichiers, les objets, les autorisations et les contraintes.

Facteurs à prendre en compte lors de l'évaluation de la taille d'une BD

- La **taille** de la BD **model** et de ses tables système en tenant compte des prévisions de croissance,
- La **quantité de données** dans les tables en tenant compte des prévisions de croissance,
- Le **nombre et la taille des index**, notamment la taille de la valeur de la clé, le nombre de lignes et le taux de remplissage.

Facteurs à prendre en compte lors de l'évaluation de la taille d'une BD (Suite)

- **La taille du journal des transactions**, qui dépend de la quantité et de la fréquence des modifications, de la taille de chaque transaction et de la fréquence de sauvegarde et de vidage du journal
- **La taille des tables système**, tel que le nombre d'utilisateurs, d'objets, etc., qui ne constitue généralement pas un pourcentage élevé de la taille de la base de données

Evaluation de la quantité de données contenues dans les tables

- Après avoir examiné la quantité d'espace alloué à la BD **model**, on doit évaluer la quantité de données qui sera stockée dans les tables, en tenant compte des prévisions de croissance
- Cette évaluation peut être effectuée en déterminant le **nombre total de lignes**, la **taille des lignes**, le **nombre de ligne** contenues dans une page et le **nombre total de pages** requises pour chaque table de la BD

Evaluation de la quantité de données contenues dans les tables (suite)

Démarche à suivre :

- Calculer le **nombre d'octets par ligne**. Effectuer la moyenne des colonnes à longueur variable,
- Déterminer le **nombre de lignes contenues dans chaque page** de données. Pour cela diviser 8060 par le nombre d'octets par ligne. Arrondir le résultat à l'entier le plus proche,
- **Diviser** le nombre approximatif de lignes dans la tables par le nombre de lignes contenues dans chaque page de données. Le résultat donne le **nombre de pages requises** pour stocker la table,

Remarques sur les performances

- Utiliser la solution **RAID** pour améliorer les performances ou assurer la tolérance de panne
- Placer les fichiers de données et les journaux des transactions sur des disques **physiques différents**
- Utiliser des **groupes de fichiers** définis par l'utilisateur pour simplifier les stratégies de sauvegarde des BD très volumineuses

4. Gestion de la sécurité

4.1 Implémentation d'un mode d'authentification

4.2 Attribution de comptes de connexion à des utilisateurs et des rôles

4.3 Attribution d'autorisations à des utilisateurs et des rôles

4.4 Gestion de la sécurité dans SQL Server

4.5 Gestion de la sécurité des applications

4.6 Gestion de la sécurité de SQL Server dans l'entreprise

4.1 Implémentation d'un mode d'authentification

- Traitement de l'authentification
- Choix d'un mode d'authentification
- Authentification mutuelle à l'aide du protocole **Kerberos**
- Emprunt d'identité et délégation
- Cryptage
- Etapes de l'implémentation du mode d'authentification
- Création de comptes de connexion

Traitement de l'authentification

SQL Server peut faire appel à Windows 2000 pour authentifier les comptes de connexion comme il peut les authentifier lui-même

Traitement des comptes de connexion authentifiés par Windows 2000

- Lorsqu'un utilisateur se connecte à SQL Server, le client ouvre une connexion approuvée avec le serveur SQL Server, qui transmet les références de sécurité W-2000 de l'utilisateur à SQL Server. Comme le client a ouvert une connexion approuvée, **SQL Server sait que W-2000 a déjà validé le compte de connexion**

Traitement des comptes de connexion authentifiés par Windows 2000 (Suite)

-Si SQL Server trouve le compte d'utilisateur ou de groupe W-2000 de l'utilisateur dans la liste des comptes de connexion SQL Server de la table système **sysxlogins**, il accepte la connexion. Il n'est pas nécessaire que SQL Server valide de nouveau le mot de passe **car W-2000 l'a déjà validé**

Traitement des comptes de connexion authentifiés par Windows 2000 (Suite)

Si plusieurs ordinateurs SQL Server participent à un domaine ou un groupe de domaines approuvés, la connexion à un domaine de réseau unique est **suffisante** pour permettre l'accès à tous les ordinateurs SQL Server

Traitement des comptes de connexion authentifiés par SQL Server

- Lorsqu'un utilisateur se connecte avec un compte de connexion et un mot de passe SQL Server, SQL Server vérifie qu'un compte de connexion existe dans la table **sysxlogins** et que le mot de passe spécifié correspond à celui enregistré au préalable
- Si aucun compte de connexion n'est défini pour l'utilisateur, l'authentification échoue et la connexion est refusée

Choix d'un mode d'authentification

Avantages du mode d'authentification Windows

- Fonctionnalités de sécurités avancées
- Ajout de groupes à l'aide d'un seul compte
- Accès rapide

Avantages du mode mixte

- Les clients non-Windows 2000 et Internet peuvent l'utiliser pour se connecter

Authentification mutuelle à l'aide du protocole **Kerberos**

- Principal protocole de sécurité pour l'authentification Windows 2000,
- Vérifie à la fois l'identité de l'utilisateur et des services réseau (Authentification mutuelle)
- SQL Server 2000 utilise **Kerberos** pour prendre en charge l'authentification mutuelle entre client et serveur

Authentification mutuelle à l'aide du protocole **Kerberos** (Suite)

- Lorsqu'on ouvre une session, W-2000 recherche un serveur exécutant le service d'annuaire Active Directory et le service d'authentification **Kerberos**
- **Kerberos** émet un ticket d'octroi de ticket (**TGT**: Ticket-Granting Ticket) qui contient les données cryptées confirmant l'identité au centre de distribution de clé (**KDC** : Key Distribution Center)

Authentification mutuelle à l'aide du protocole **Kerberos** (Suite)

- Lorsqu'on demande l'accès à un service, on envoie ce ticket au centre KDC qui envoie un ticket de session pour le service demandé. On présente ensuite le ticket de session qui confirme au service l'identité
- Si une authentification mutuelle est nécessaire, le serveur répond alors par un message crypté pour s'identifier

Authentification mutuelle à l'aide du protocole **Kerberos** (Suite)

- Le ticket de session peut être réutilisé pour accéder au même service jusqu'à son expiration, ce qui évite au centre KDC d'émettre constamment des tickets
- Le délai d'expiration est déterminé par le centre **KDC**, en général inférieur à 8H (durée d'une session de connexion standard)

Emprunt d'identité et délégation

Permettent à SQL Server d'utiliser les références de sécurité du client d'origine pour accéder à des ressources situées sur les serveurs locaux ou distants

Emprunt d'identité:

SQL Server peut avoir besoin de présenter l'identité d'un client à des ressources auxquelles il accède de la part du client (Système de fichiers par exemple)

Emprunt d'identité et délégation (Suite)

Délégation:

Permet de se connecter à plusieurs serveurs, et, à chaque changement de serveur, de conserver les références du client d'origine

Pour utiliser la délégation, tous les serveurs auxquels on veut se connecter doivent exécuter W-2000 avec prise en charge du protocole **Kerberos** et on doit configurer **Active Directory**

Etapes d'implémentation du mode d'authentification

- 1- Définir le mode d'authentification
- 2- Arrêter puis redémarrer le service **MSSQLServer**
- 3- Créer des groupes et des utilisateurs Windows 2000
- 4- Autoriser les groupes et les utilisateurs Windows 2000 à accéder à SQL Server
- 5- Créer des comptes de connexions SQL Server pour les utilisateurs qui se connectent à l'aide de connexions non approuvées

Création de comptes de connexion

- On peut créer des comptes de connexion à partir d'utilisateurs et de groupes W-2000 déjà définis ou créer de nouveaux comptes de connexion par défaut
- Les comptes de connexions sont stockés dans la table système **master..sysxlogins**
- Lorsqu'un compte de connexion est ajouté à un serveur SQL Server, il se voit attribuer une BD par défaut. Cette opération ne crée pas de compte d'utilisateur dans cette BD

Ajout d'un compte de connexion Windows 2000 à SQL Server

- On peut utiliser SQL Server Enterprise Manager ou la procédure stockée système **sp_grantlogin** pour permettre à un compte d'utilisateur ou de groupes W-2000 de se connecter à SQL Server
- Seuls les administrateurs système ou ceux chargés de la sécurité peuvent exécuter la procédure **sp_grantlogin**
- La suppression d'un utilisateur ou d'un groupe W-2000 ne supprime pas l'utilisateur ou le groupe correspondant dans SQL Server. Donc, on doit supprimer les utilisateurs ou les groupes W-2000 avant de les supprimer de SQL Server

Ajout d'un compte de connexion SQL Server

- On peut utiliser SQL Server Enterprise Manager ou la procédure stockée système **sp_addlogin** pour créer un compte de connexion SQL Server.
- Seuls les administrateurs système ou ceux chargés de la sécurité peuvent exécuter la procédure **sp_addlogin**
- La création d'un compte de connexion SQL Server ajoute un enregistrement dans la table **sysxlogins** de la BD **master**.

Comptes de connexion par défaut

SQL Server possède 2 comptes de connexions par défaut :

-BUILTIN\Administrateurs : compte de connexion par défaut pour tous les administrateurs W-2000. Il possède tous les droits sur le serveur SQL Server et dans toutes les BD

-sa (administrateur système) : compte de connexion spécial qui possède tous les droits sur le serveur SQL Server et dans toutes les BD. Il est activé uniquement lorsque SQL Server utilise l'authentification en mode mixte

4. Gestion de la sécurité

4.1 Implémentation d'un mode d'authentification

4.2 Attribution de comptes de connexion à des utilisateurs et des rôles

4.3 Attribution d'autorisations à des utilisateurs et des rôles

4.4 Gestion de la sécurité dans SQL Server

4.5 Gestion de la sécurité des applications

4.6 Gestion de la sécurité de SQL Server dans l'entreprise

4.2 Attribution des comptes de connexion à des utilisateurs et des rôles

- Après avoir ajouté des comptes de connexion à SQL Server, on peut les **mapper** sur des comptes d'utilisateur ou des rôles dans toutes les BD auxquelles les utilisateurs doivent accéder
- La table système **sysusers** d'une BD contient une ligne pour chaque utilisateur W-2000, groupe W-2000, utilisateur SQL Server ou rôle dans la BD
- Les autorisations sont attribuées aux entrées dans la table **sysusers** et stockées dans la table **sysprotects** de la BD active

Attribution des comptes de connexion à des comptes d'utilisateur

- On peut ajouter des comptes d'utilisateur par SQL Server Enterprise Manager ou par la procédure stockée système **sp_grantdbaccess**.
- On peut aussi ajouter des comptes d'utilisateurs aux BD lors de la création des comptes de connexion
- Seuls les propriétaires d'une BD ou les administrateurs d'accès aux BD peuvent exécuter la procédure **sp_grantdbaccess**

Compte d'utilisateur *dbo*

- Le nom d'ouverture de session **sa** et les membres du rôle **sysadmin** sont mappés sur le compte d'utilisateur **dbo** dans toutes les BD
- Tous les objets créés par un administrateur système font partie de **dbo**
- C'est un compte par défaut et ne peut pas être supprimé

Compte d'utilisateur *guest*

- Autorise les connexions sans compte d'utilisateur associé à accéder à une BD
- Des autorisations peuvent être attribuées à **guest** comme à tout autre compte d'utilisateur
- On peut supprimer l'utilisateur **guest** de toutes les BD et l'y ajouter sauf les BD **master** et **tempdb**

Attribution des comptes de connexion à des rôles

- Servent à **regrouper** des utilisateurs en une seule unité à laquelle on peut attribuer des droits
- **3 types de rôles :**
 - Rôles fixes de **serveur**
 - Rôles fixes de **BD**
 - Rôles de BD **définis par l'utilisateur**

Rôles fixes de serveur

- Permettent de regrouper des privilèges administrateur au niveau du serveur
- Sont gérés indépendamment des BD utilisateur
- Sont stockés dans la table système **master..sysxlogins**

Rôles fixes de serveur

Nom du Rôle	Autorisations
sysadmin	Effectuer n'importe quelle activité
dbcreator	Gérer et modifier des BD
diskadmin	Gérer des fichiers disque
processadmin	Gérer les processus SQL Server
serveradmin	Configurer les paramètres du serveur
setupadmin	Installer la réplication
securityadmin	Gérer et contrôler les connexions au serveur
bulkadmin	Exécuter des instructions BULK INSERT

Attribution d'un compte de connexion à un rôle fixe de serveur

- Par SQL Server Enterprise Manager ou la procédure stockée système **sp_addsrvrolemember**
- On ne peut pas ajouter, modifier ou supprimer des rôles fixes de serveur
- Tous les membres d'un rôle fixe de serveur peuvent ajouter d'autres comptes de connexions à ce rôle
- On peut supprimer un membre d'un rôle fixe de serveur par SQL Server Enterprise Manager ou la procédure stockée système **sp_dropsrvrolemember**

Rôles fixes de BD

- Permettent de regrouper des privilèges administrateur au niveau de la BD
- Sont stockés dans la table système **sysusers** de chaque BD

Rôles fixes de BD

Nom du Rôle	Autorisations
public	Gérer toutes les autorisations attribuées par défaut
db_owner	Effectuer toutes les activités de rôle de BD
db_accessadmin	Ajouter ou supprimer des utilisateurs, des groupes et des rôles d'une BD
db_ddladmin	Ajouter, modifier ou supprimer les objets d'une BD
db_securityadmin	Attribuer des autorisations sur les objets et les instructions
db_backupoperator	Sauvegarder la BD
db_datareader	Lire les données de toutes les tables
db_datawriter	Ajouter, modifier ou supprimer les données de toutes les tables
db_denydatareader	Refuser la lecture des données de toutes les tables
db_denydatawriter	Refuser la modification des données de toutes les tables

Rôle public

- Rôle spécial auquel appartient chaque utilisateur de BD
- Ne peut pas être supprimé
- Permet de maintenir toutes les autorisations par défaut des utilisateurs d'une BD
- Ne peut pas se voir attribuer des utilisateurs, des groupes ou des rôles car ces derniers lui appartiennent déjà par défaut
- Il est présent dans chaque BD, y compris les BD **master**, **msdb**, **tempdb**, **model** et dans toutes les BD utilisateur

Tâches liées au rôle **public**

- Exécuter des instructions qui ne requièrent pas d'autorisations telles que l'instruction PRINT,
- Afficher les informations des tables système et exécuter certaines procédures stockées système afin d'extraire des informations de la BD **master** des BD utilisateur auxquelles il a accès,
- Avoir accès à toutes BD avec un compte **guest**

Attribution d'un compte de sécurité à un rôle fixe de BD

- Par SQL Server Enterprise Manager ou la procédure stockée système **sp_addrolemember** pour ajouter un compte de sécurité en tant que membre d'un rôle fixe de BD
- Seuls les membres du rôle **db_owner** peuvent exécuter la procédure stockée système **sp_addrolemember** pour tous les rôles de la BD
- **On ne peut pas** ajouter, modifier ou supprimer des rôles fixes de BD

Attribution d'un compte de sécurité à un rôle fixe de BD

- Tous les **membres d'un rôle fixe** de BD peuvent **ajouter d'autres comptes de connexions à ce rôle**
- On peut supprimer un compte de sécurité d'un rôle par SQL Server Enterprise Manager ou la procédure stockée système **sp_droprolemember**

Rôles de BD définis par l'utilisateur

➤ Ajoutez un rôle :

- Lorsqu'un groupe de personnes doit effectuer un ensemble spécifique d'activités dans SQL Server
- Si vous ne disposez pas des autorisations pour gérer les comptes d'utilisateur W-2000

Rôles de BD définis par l'utilisateur

- Utilisez SQL Server Enterprise Manager ou la procédure stockée système **sp_addrole** pour créer un rôle de BD
- Une entrée est ajoutée à la table **sysusers** de la BD active pour chaque rôle défini par l'utilisateur
- Seuls les membres du rôle **db_securityadmin** ou **db_owner** peuvent exécuter la procédure **sp_addrole**

Attribution d'un compte de sécurité à un rôle de BD défini par l'utilisateur

- Par SQL Server Enterprise Manager ou la procédure stockée système **sp_addrolemember** pour ajouter des utilisateurs ou des rôles en tant que membre du rôle
- Seuls les membres du rôle fixe de serveur **sysadmin**, des rôles fixes de BD **db_securityadmin** et **db_owner** ou le propriétaire du rôle peuvent exécuter la procédure **sp_addrolemember** pour ajouter un membre à un rôle de BD défini par l'utilisateur

Attribution d'un compte de sécurité à un rôle de BD défini par l'utilisateur (Suite)

- Lorsqu'on ajoute un compte de sécurité à un rôle, toutes les **autorisations appliquées** au rôle sont appliquées au nouveau membre
- Lorsqu'on ajoute un rôle SQL Server en tant que membre d'un autre rôle SQL Server, on ne peut pas créer des rôles récursifs. **Par exemple, *Compte_de_sécurité*** ne peut pas être ajouté en tant que membre de ***Rôle*** si ***Rôle*** est déjà membre de ***Compte_de_sécurité***

Attribution d'un compte de sécurité à un rôle de BD défini par l'utilisateur (Suite)

- On peut supprimer un rôle SQL Server de la BD active par SQL Server Enterprise Manager ou la procédure stockée système **sp_droprole**
- On peut supprimer un compte de sécurité d'un rôle SQL Server par SQL Server Enterprise Manager ou la procédure stockée système **sp_droprolemember**

4. Gestion de la sécurité

4.1 Implémentation d'un mode d'authentification

4.2 Attribution de comptes de connexion à des utilisateurs et des rôles

4.3 Attribution d'autorisations à des utilisateurs et des rôles

4.4 Gestion de la sécurité dans SQL Server

4.5 Gestion de la sécurité des applications

4.6 Gestion de la sécurité de SQL Server dans l'entreprise

4.3 Attribution d'autorisations à des utilisateurs et des rôles

- Après avoir attribué des comptes de connexion à des comptes d'utilisateur et des rôles, on **doit octroyer des autorisations** pour mettre en place la sécurité de la BD
- Les autorisations spécifient les objets de la BD que les utilisateurs sont autorisés à utiliser et ce que ces derniers peuvent faire de ces objets
- Les autorisations dont dispose un utilisateur dans une BD dépendent de celles du compte d'utilisateur et des rôles dont l'utilisateur est membre

Types d'autorisations

- Autorisations sur les instructions
- Autorisations sur les objets
- Autorisations prédéfinies

Autorisations sur les instructions

- Impliquent la création de BD ou d'éléments d'une BD
- CREATE DATABASE, CREATE TABLE, CREATE VIEW, CREATE PROCEDURE, CREATE RULE, CREATE DEFAULT, CREATE FUNCTION, BACKUP DATABASE, BACKUP LOG
- Seuls les membres du rôle **sysadmin**, **db_owner** ou **db_securityadmin** peuvent accorder des autorisations sur les instructions

Autorisations sur les objets

- Impliquent l'utilisation des données ou l'exécution de procédures
- SELECT, INSERT, UPDATE, DELETE, REFERENCES sur les vues de la table
- SELECT, UPDATE, REFERENCES sur les colonnes
- EXEC sur les procédures stockées

Autorisations prédéfinies attribuées aux rôles fixes

- Administration implicite

Exemple : un utilisateur ajouté en tant que membre du rôle **sysadmin** hérite automatiquement de toutes les autorisations nécessaires à la réalisation de toute tâche liée à l'installation de SQL Server.

Autorisations prédéfinies attribuées aux propriétaires d'objets

Exemple : un utilisateur propriétaire d'une table ou un membre d'un groupe désigné comme propriétaire de la table peut réaliser toutes les activités liées à la table. Il peut afficher, ajouter ou supprimer des données, modifier la définition de la table et contrôler les autorisations qui permettent à d'autres utilisateurs de travailler avec la table.

Octroi, refus et révocation d'autorisations

- **Autorisation** : soit accordée (**GRANT**), refusée (**DENY**) ou révoquée (**REVOKE**)
- Autorisation ni accordée ni refusée → neutre (révoquée)
- Les autorisations font l'objet d'entrée dans la table système **sysprotects** de chaque BD
- Les autorisations accordées peuvent être **cumulées**
- L'instruction **DENY annule** une autorisation attribuée à un rôle auquel appartient l'utilisateur

Octroi d'une autorisation d'accès (GRANT)

- On ne peut accorder une autorisation que dans la BD active
- Seuls les membres des rôles **sysadmin**, **db_owner** et **db_securityadmin** et les propriétaires d'objets peuvent accorder des autorisations sur ces objets
- Les autorisations disponibles **varient** selon l'objet sélectionné (EXECUTE pour une procédure stockée, SELECT, INSERT, ... pour une table ou vue)

Refus d'une autorisation d'accès (DENY)

- On peut restreindre les autorisations d'un utilisateur ou d'un rôle spécifique en refusant les autorisations pour ce compte de sécurité
- Supprime les autorisations précédemment accordées à l'utilisateur ou au rôle
- Désactive les autorisations héritées d'un autre rôle
- On ne peut refuser des autorisations que pour la BD active
- Seuls les membres des rôles **sysadmin**, **db_owner** et **db_securityadmin** et les propriétaires d'objets peuvent refuser les autorisations

Révocation d'autorisations accordées et refusées (REVOKE)

- On peut désactiver une autorisation accordée ou refusée en la révoquant
- La révocation d'une autorisation supprime les entrées de la table système **syspermissions** créées lors de l'octroi et du refus de l'autorisation
- On ne peut révoquer que les autorisations de la BD active
- Seuls les membres des rôles **sysadmin**, **db_owner** et **db_securityadmin** et les propriétaires d'objets peuvent révoquer les autorisations

4. Gestion de la sécurité

4.1 Implémentation d'un mode d'authentification

4.2 Attribution de comptes de connexion à des utilisateurs et des rôles

4.3 Attribution d'autorisations à des utilisateurs et des rôles

4.4 Gestion de la sécurité dans SQL Server

4.5 Gestion de la sécurité des applications

4.6 Gestion de la sécurité de SQL Server dans l'entreprise

4.4 Gestion de la sécurité dans SQL Server

- Tenir compte des recommandations suivantes lors de la planification de la sécurité :
- Utilisation des comptes de connexion par défaut:
 - Compte de connexion **sa*** : attribué au rôle fixe de serveur **sysadmin** et ne peut être supprimé
 - Compte de connexion **BUILTIN\Administrateurs*** : Les administrateurs système doivent être membres du rôle fixe de serveur **sysadmin**. Le groupe local **Administrateurs** de W-2000 est mappé sur le compte de connexion **BUILTIN\Administrateurs** qui est un membre du rôle **sysadmin**

Gestion de la sécurité dans SQL Server (Suite)

- Le compte d'utilisateur **guest** permet à un compte de connexion dépourvu de compte d'utilisateur d'accéder à une BD. - Vous devez donc **décider** si vos BD doivent posséder un compte **guest**, et si tel est le cas, **définir les autorisations** attribuées à ce compte dans vos BD
- Le rôle **public** est un rôle de BD spécial auquel chaque utilisateur de BD appartient. On doit définir les autorisations attribuées au rôle **public** dans chaque BD (Par défaut, le rôle **public** ne dispose d'aucune autorisation)

Gestion de la sécurité dans SQL Server (Suite)

Lors de l'application d'autorisations aux rôles, tenez compte des instructions suivantes :

- Créez des rôles** définis par l'utilisateur pour votre application,
- Appliquez des autorisations** aux rôles définis par l'utilisateur que vous avez créés pour votre application,
- Ajoutez des membres** aux rôles définis par l'utilisateur que vous avez créés.

Gestion de la sécurité dans SQL Server (Suite)

- Il est primordial de déterminer les utilisateurs et les rôles qui peuvent **créer des objets** dans une BD
- Il est recommandé d'utiliser uniquement les rôles fixes de BD **sysadmin**, **db_owner** et **db_ddladmin** pour créer des objets de BD
- Il est aussi recommandé de définir tous les objets avec l'utilisateur **dbo** spécifié en tant que propriétaire. Cela permet à tous les utilisateurs de BD de se référer à l'objet sans avoir à inclure le nom du propriétaire

Gestion de la sécurité dans SQL Server (Suite)

- **dbo** est propriétaire de tous les objets créés à partir du rôle **sysadmin**
- Dans tous les autres rôles, spécifiez toujours l'utilisateur **dbo** en tant que nom de propriétaire lorsque vous créez l'objet ; sinon, l'objet sera créé avec votre nom d'utilisateur
- Si un objet n'a pas été créé avec l'utilisateur **dbo** en tant que propriétaire de l'objet, on peut modifier ce dernier à l'aide de la procédure stockée système **sp_changeobjectowner**

Gestion de la sécurité dans SQL Server (Suite)

Lors de la modification de propriétaires d'objets de BD, tenez compte des points ci-dessous :

- Seuls les membres des rôles fixes de BD **db_owner** et **db_ddladmin** et les membre du rôle de serveur **securityadmin** peuvent modifier les propriétaires d'objets de BD
- La modification du propriétaire d'un objet de BD implique une **MAJ manuelle** de tous les scripts et fichiers de traitement par lots faisant référence à cet objet afin d'inclure les nouvelles informations de propriété. SQL Server **ne peut pas exécuter** cette MAJ **automatiquement**

Gestion de la sécurité dans SQL Server (Suite)

- Les travaux SQL Server peuvent contenir des étapes de types **CmdExec** et **ActiveScripting**
- Elles permettent d'exécuter des scripts, des applications compilées et des fichiers de traitement par lots
- Lorsqu'un utilisateur qui n'est pas membre du rôle **sysadmin** exécute ces travaux, ils sont effectués dans le contexte du compte d'utilisateur W-2000 associé au service de **l'Agent SQL Server**

Gestion de la sécurité dans SQL Server (Suite)

- Ainsi, les utilisateurs peuvent être en mesure d'exécuter des fonctions dans l'environnement W auxquelles leurs comptes W ne leur donnent normalement pas accès
- Pour éviter cela, vous pouvez configurer le service de **l'Agent SQL Server** pour qu'il permette seulement aux utilisateurs disposant de droits **sysadmin** d'exécuter des étapes de travail **CmdExec** et **ActiveScripting**

4. Gestion de la sécurité

4.1 Implémentation d'un mode d'authentification

4.2 Attribution de comptes de connexion à des utilisateurs et des rôles

4.3 Attribution d'autorisations à des utilisateurs et des rôles

4.4 Gestion de la sécurité dans SQL Server

4.5 Gestion de la sécurité des applications

4.6 Gestion de la sécurité de SQL Server dans l'entreprise

4.5 Gestion de la sécurité des applications

- Gestion de la sécurité à l'aide de **vues** (Accès à certains champs uniquement) et de **procédures stockées** (Exécution de procédure stockée sans consulter les tables modifiées)
- Gestion de la sécurité des applications clientes à l'aide de **rôle d'application**

Rôles d'application

➤ **Caractéristiques :**

- Ne contiennent pas de membres
- Sont activés pour les utilisateurs lorsqu'ils exécutent l'application
- Nécessitent l'activation d'un mot de pass

➤ **Lors de l'activation d'un rôle d'application :**

- L'utilisateur **perd** toutes les autorisations existantes de la BD active sauf celles du rôle **public**
- **Hérite** de toutes les autorisations liées au rôle de l'application dans la BD active

Création des rôles d'application

➤ Création d'un rôle d'application :

- Par SQL Server Entreprise Manager ou la procédure stockée système **sp_addapprole**
- Seuls les membres des rôles **db_owner**, **db_securityadmin** et **sysadmin** peuvent exécuter **sp_addapprole**
- La création d'un rôle d'application ajoute un compte de sécurité pour le nouveau rôle en insérant une ligne dans la table **sysusers** de la BD active

➤ Gestion des autorisations sur les rôles d'application:

- Par SQL Server Entreprise Manager
- Ou les instructions **GRANT**, **DENY** et **REVOKE**

Activation des rôles d'application

- Une fois qu'un client s'est connecté à SQL Server au moyen d'un compte de connexion, il doit exécuter la procédure stockée système **sp_setapprole** pour activer les autorisations associées à un rôle d'application
- L'utilisateur doit spécifier un mot de passe
- La portée d'un rôle d'application est limitée à la BD active. Si l'utilisateur change de BD, il dispose des autorisations du compte **guest** dans cette BD
- Le rôle ne peut pas être désactivé tant que l'utilisateur ne se déconnecte pas

Activation des rôles d'application

➤ Syntaxe :

```
sp_setapprole [@rolename=] 'role',  
[@password=] {Encrypt N 'mot_pass'} | 'mot_pass'  
[,[@encrypt=] 'style_cryptage']
```

➤ **Exemple :** EXEC **sp_setapprole** 'SalesApp',
{Encrypt N 'hg_7532LR'}, 'ODBC'

- On peut supprimer un rôle d'application par la procédure stockée système **sp_dropapprole**
- On peut modifier le mot de passe d'un rôle d'application par la procédure stockée système **sp_approlepassword**

5. Exécution de tâches administratives

5.1 Tâches liées à la configuration

5.2 Tâches de routine liées à l'administration de SQL Server

5.3 Automatisation des tâches de maintenance de routine

5.4 Création d'alertes

5.5 Résolution des problèmes liés à l'automatisation de SQL Server

5.6 Automatisation de travaux sur plusieurs serveurs

5.1 Tâches liées à la configuration

- Configuration de l'Agent SQL Server
- Configuration de SQLAgentMail et de SQLMail
- Configuration de serveurs liés
- Configuration des noms de sources de données
- Configuration de la prise en charge du langage XML de SQL Server dans les services Internet
- Configuration de SQL Server pour partager les ressources de mémoire avec les autres applications serveur

Configuration de l'Agent SQL Server

- L'Agent SQL Server est le composant de SQL Server **chargé** d'automatiser les **tâches** administratives de SQL Server
- Pour qu'il exécute des travaux et déclenche des alertes, il **doit être exécuté en permanence** et disposer des autorisations adéquates
- Doit être **configuré** pour qu'il **démarre automatiquement** à chaque démarrage de Windows 2000
- Peut être configuré dans SQL Server Entreprise Manager pour qu'il **redémarre automatiquement** en cas d'arrêt inattendu
- Pour qu'il redémarre automatiquement, son compte doit être membre du groupe **Administrateurs** local

Configuration de l'Agent SQL Server (suite)

- Lorsqu'on installe SQL Server, on doit indiquer un compte **d'ouverture de session** pour le service SQL Server Agent
- L'Agent SQL Server ne pourra démarrer que si le rôle **sysadmin** de SQL Server est attribué au compte de connexion
- L'Agent SQL Server peut **se connecter** à l'ordinateur local SQL Server à l'aide de **l'authentification** Windows ou SQL Server
- Par défaut, l'Agent SQL Server utilise l'authentification Windows à l'aide du compte de connexion défini comme compte de démarrage du service SQL Server Agent

Configuration de SQLAgentMail et SQL Mail

- SQL Server peut envoyer et recevoir du **courrier électronique** en établissant une connexion cliente avec un serveur de messagerie
- Il envoie et reçoit du courrier électronique à l'aide d'une session **SQLAgentMail** ou **SQL Mail**
- SQLAgentMail nécessite que le service SQL Server Agent utilise un compte d'utilisateur de domaine avec un **profile de messagerie** (à indiquer avec SQL Server Enterprise Manager)
- SQL Mail contient des **procédure stockées pour traiter les messages électroniques** entrants et renvoyer des résultats
- Permet d'envoyer des messages électroniques à partir de l'application de BD en exécutant par exemple la procédure stockée **xp_sendmail** à partir d'un déclencheur

Configuration de SQLAgentMail et SQL Mail (suite)

Pour que SQL Mail et SQLAgentMail démarrent des sessions de messagerie, on doit effectuer les tâches suivantes:

- Utiliser un serveur de messagerie **compatible** MAPI-1 (*Messaging Application Programming Interface*)
- **Configurer** un client de messagerie sur l'ordinateur qui exécute SQL Server
- Utiliser un compte de **connexion d'utilisateur de domaine** pour les services SQL Server et SQL Server Agent
- Configurer un **profil de messagerie** pour chaque compte d'utilisateur de domaine en vue d'une connexion au serveur de messagerie
- Utiliser SQL Server Enterprise Manager pour **spécifier un profil de messagerie** pour chaque session SQL Mail et SQLAgentMail

Configuration de serveurs liés

- La configuration d'un serveur lié permet à SQL Server d'exécuter des commandes sur des **sources de données OLE DB** sur **différents serveurs**
- Pour configurer la définition d'un serveur lié, on **inscrit les informations sur la connexion et la source de données OLE DB** auprès de SQL Server
- Une fois l'inscription effectuée, on peut toujours faire **référence** à cette source de donnée par un seul **nom logique**
- On peut créer la définition d'un serveur lié par la procédure stockée système **sp_addlinkedserver** ou sur SQL Server Enterprise Manager

Etablissement de la sécurité des serveurs liés

- Lors d'une connexion à un serveur lié, le serveur émetteur fournit **un nom et un mot de passe** d'ouverture de session pour se connecter au serveur récepteur
- On crée des **mappages de connexion** entre les serveurs liés par la procédure stockée système **sp_addlinkedrvlogin** ou par SQL Server Entreprise Manager
- Lorsqu'on ajoute un serveur lié par **sp_addlinkedserver**, un automappage par défaut est ajouté à tous les comptes de connexion locaux

Configuration des noms de sources de données

Un nom de source de données définit:

- Le **pilote ODBC** à utiliser les informations de connexion (nom et emplacement de la source de données, compte de connexion et mot de passe)
- Options propres au pilote pour la connexion

Configuration de la prise en charge du langage XML de SQL Server dans les services Internet

- On peut configurer les services Internet (Microsoft Internet Information Services) pour permettre l'accès à SQL Server
- On peut accéder à SQL Server par le biais du protocole HTTP à l'aide d'une URL

Configuration de la prise en charge du langage XML de SQL Server dans les services Internet

On peut préciser que cette URL effectue les tâches suivantes :

- Elle accède directement aux objets de BD comme les tables
- Elle exécute les fichiers modèles. Un modèle est un document valide XML comprenant des instructions T-SQL
- Elle exécute des requêtes XPath (XML Path Language). Les requêtes XPath sont exécutées sur des fichiers de schéma de mappage annotés, spécifiés dans l'URL.

Configuration d'un annuaire virtuel dans les services Internet

- Pour accéder à une BD SQL Server par le biais du protocole HTTP, on doit configurer un annuaire virtuel approprié
- Utiliser l'utilitaire **IIS Virtual Directory Management** pour SQL Server pour définir et inscrire un nouveau répertoire virtuel appelé également **racine virtuelle** sur l'ordinateur exécutant les services Internet
- Cet utilitaire ordonne aux services Internet de créer une **association** entre le nouvel annuaire et une instance de SQL Server

Accès du langage XML à SQL Server

Dans un échange XML, les opérations suivantes de produisent:

- Le **nom du serveur** exécutant les services Internet est **indiqué à l'URL**
- Le serveur exécutant les services Internet **examine la racine virtuelle** indiquée dans l'URL et détermine si **l'extension de nom de fichier DLL** (Sqlisapi.dll) ISAPI (Internet Server API) **a été inscrite** pour cette racine virtuelle
- Le serveur exécutant les services Internet **charge la bibliothèque DLL** et lui **transmet la demande d'URL**
- L'extension de nom de fichier Sqlisapi.dll **communique avec le fournisseur OLE DB** pour SQL Server et **établit une connexion avec l'instance de SQL Server** identifiée dans la racine virtuelle

Configuration de SQL Server pour partager les ressources de mémoire avec les autres applications serveur

- Configuration des options de mémoire :
 - **min server memory** : définit un niveau sous lequel SQL Server ne libère pas de mémoire
 - **max server memory** : empêche SQL Server d'utiliser plus de mémoire que la quantité spécifiée
- Identification de la quantité maximale de mémoire à configurer pour une instance de SQL Server
- Utilisation du Moniteur système Windows 2000 pour l'observateur d'effets : Utilisez les statistiques du Moniteur système Windows 2000 pour vous aider à évaluer comment les applications serveur s'exécutent et pour régler la valeur de mémoire.

5. Exécution de tâches administratives

5.1 Tâches liées à la configuration

5.2 Tâches de routine liées à l'administration de SQL Server

5.3 Automatisation des tâches de maintenance de routine

5.4 Création d'alertes

5.5 Résolution des problèmes liés à l'automatisation de SQL Server

5.6 Automatisation de travaux sur plusieurs serveurs

5.2 Tâches de routine liées à l'administration de SQL Server

- Exécution de tâches planifiées régulièrement
 - Sauvegarde de BD
 - Importation et exportation de données
- Identification et résolution de problèmes potentiels
 - Surveillance de l'espace des BD et des journaux
 - Surveillance des performances

5. Exécution de tâches administratives

5.1 Tâches liées à la configuration

5.2 Tâches de routine liées à l'administration de SQL Server

5.3 Automatisation des tâches de maintenance de routine

5.4 Création d'alertes

5.5 Résolution des problèmes liés à l'automatisation de SQL Server

5.6 Automatisation de travaux sur plusieurs serveurs

5.3 Automatisation des tâches de maintenance de routine

- Automatisation de l'administration de SQL Server
- Création de travaux
- Vérification des autorisations
- Définition des étapes d'un travail
- Création d'un organigramme des actions par étape de travail
- Planification de travaux
- Création d'opérateurs à notifier
- Analyse et configuration de l'historique des travaux

Automatisation de l'administration de SQL Server

- Services utilisés pour automatiser SQL Server :
 - Agent SQL Server, SQL Server et Observateur d'événement Windows 2000
- Composants de l'Agent SQL Server :
 - Les alertes, les travaux et les opérateurs
- Association de travaux et d'alertes

Exemple : si un travail échoue à cause d'une erreur système, une **alerte** définie pour répondre à ce numéro d'erreur peut **exécuter un autre travail** destiné à résoudre ce problème

Création de travaux

- Définir un travail par SQL Server Enterprise Manager ou la procédure stockée système **sp_add_job**
- La définition du travail est stockée dans la table système **msdb..sysjobs**
- Quand on définit un travail, on doit :
 - **Vérifier** qu'il est activé (par défaut oui)
 - Indiquer le **propriétaire** chargé de l'exécution du travail (par défaut compte d'ouverture de session Windows 2000 ou SQL Server)
 - Déterminer **l'emplacement de l'exécution** du travail (serveur local ou plusieurs serveurs distants)
 - **Créer des catégories** de travail (pour aider à organiser, filtrer et gérer)

Vérification des autorisations

- Si un compte d'ouverture de session qui n'est pas membre du rôle **sysadmin** possède un travail, on doit vérifier que le propriétaire du travail dispose des autorisations appropriées pour l'exécuter
- **Exécution des travaux Transact-SQL :**
 - Tous les utilisateurs peuvent démarrer des travaux T-SQL qui opèrent dans le cadre de la sécurité du propriétaire du travail ou de l'utilisateur indiqué

Vérification des autorisations (suite)

➤ Exécution de commandes du SE ou de travaux de script ActiveX :

- Les étapes qui appartiennent aux utilisateurs membres du rôle **sysadmin** sont exécutées dans le cadre de la sécurité du compte de connexion du service SQL Server.
- Si un compte d'ouverture de session qui n'est pas membre du rôle **sysadmin** possède un travail, l'Agent SQL Server doit vérifier que son propriétaire dispose des autorisations appropriées pour l'exécuter
- **Par défaut**, les utilisateurs qui ne sont pas membres du rôle **sysadmin** **ne sont pas autorisés** à exécuter les commandes du SE ni les travaux de script ActiveX

Vérification des autorisations (suite)

- Un administrateur peut autoriser les utilisateurs qui ne sont pas membre du rôle **sysadmin** à exécuter les commandes du SE ou les travaux de script ActiveX
- Le cas échéant, les étapes de travail sont exécutées dans le cadre de la sécurité d'un compte d'utilisateur de domaine défini appelé **compte proxy**
- On peut définir ce compte proxy dans SQL Server Entreprise Manager ou par la procédure stockée étendue **xp_sqlagent_proxy_account**

Définition des étapes d'un travail

- Sous SQL Server Entreprise Manager ou par la procédure stockée système **sp_add_jobstep**
- Les définitions des étapes d'un travail sont stockées dans la table système **msdb..sysjobsteps**
- On peut définir les étapes d'un travail pour exécuter des instructions T-SQL, des commandes du SE, des scripts ActiveX ou des tâches de réplication SQL Server (**un seul type d'exécution par étape de travail**)

Utilisation d'instructions T-SQL

- On doit identifier la BD à utiliser
- On doit inclure les variables et les paramètres requis dans l'étape de travail
- On doit envoyer l'ensemble des résultats d'une étape de travail vers un fichier de sortie (capturer les messages d'erreur)
- Le fichier de sortie d'une étape de travail ne peut servir d'entrée dans l'étape suivante

Utilisation de commandes du SE (Fichiers .exe, .bat, .cmd ou .com)

- On doit identifier un code de sortie du processus pour indiquer que la commande a réussi
- On doit indiquer le chemin d'accès complet de l'application exécutable dans la zone de texte

Commande

- Le chemin permet à l'Agent SQL Server de rechercher l'application source

Utilisation de scripts ActiveX

- On doit identifier le langage de script dans lequel est écrite l'étape du travail
- Ecrire ou ouvrir le script actif
- On peut utiliser l'objet SQLActiveScriptHost pour imprimer la sortie vers l'historique des étapes de travail ou créer des objets
- On peut aussi compiler de façon externe des scripts ActiveX (à l'aide par exemple de VB) puis les exécuter en tant que commandes du SE

Utilisation de la réplication

- les processus de réplication sont appelés

Agents

- Ils sont implémentés sous forme de travaux de l'Agent SQL Server

Création d'un organigramme des actions par étape de travail

- Lors de la création de travaux, un administrateur de BD peut être amené à spécifier **l'action appropriée** que doit exécuter SQL Server en cas **d'échec** de l'exécution de travail
- **Par défaut**, SQL Server **passse à l'étape suivante** du travail en cas de réussite d'une étape et s'arrête en cas d'échec
- Cependant, les étapes de travail peuvent renvoyer à une étape quelconque définie dans le travail en cas de réussite ou d'échec
- On peut indiquer le **nombre de tentatives** en cas d'échec
- On peut également définir des **intervalles de reprise** en mn

Planification des travaux

- Sous SQL Server Entreprise Manager ou par la procédure stockée système **sp_add_jobschedule**
- Les planifications des travaux sont stockées dans la table système **msdb..sysjobshedsules**
- Les travaux **s'exécutent** conformément aux **planifications** définies ou en réponse à des alertes
- Si on travaille dans un environnement multiserveurs, on peut définir le travail à s'exécuter sur plusieurs serveurs cibles
- Un travail est exécuté conformément à la planification uniquement lorsque cette dernière est activée

Planification des travaux (suite)

- On peut planifier le **démarrage automatique** des travaux dans les situations suivantes :
 - lorsque **l'Agent SQL Server est démarré**
 - **A une heure spécifique** (une seule fois)
 - **régulièrement** (tous les jours, toutes les semaines ou tous les mois)
 - lorsque le **processeur est inactif**
- Le compte d'utilisateur de domaine utilisé par le service SQL Server doit être **membre du groupe Administrateurs local** Windows pour exécuter un travail lorsque le processeur est inactif

Planifications multiples

- **Plusieurs planifications** peuvent être définies **par travail**
- **Par exemple**, on peut planifier un travail qui sauvegarde le journal des transactions d'une BD du Lundi au Vendredi, toutes les 2 heures pendant les heures de pointes (de 08:00 à 17:00), puis établir une autre planification qui exécute le travail toutes les 4 heures hors des heures de pointe

Création d'opérateurs à notifier

- Sous SQL Server Entreprise Manager ou par la procédure stockée système **sp_add_operator**
- La définition de l'opérateur est stockée dans la table système **msdb..sysoperators**
- Une fois le travail exécuté ou en cas d'échec d'une de ses étapes, on peut écrire l'événement dans le journal d'applications de Windows, supprimer le travail ou **notifier** un opérateur par radiomessagerie, courrier électronique ou une commande **net send**

Tâches à effectuer lors de la création d'opérateurs

- Utiliser un **alias de messagerie** de groupe pour demander à plusieurs personnes de résoudre des problèmes potentiels
- **tester chaque méthode de notification** utilisée pour s'assurer que l'opérateur peut recevoir le message
- établir une **planification de travail** pour chaque opérateur à notifier par radiomessagerie
- Utiliser une commande **net send** pour envoyer des messages aux opérateurs et serveurs du réseau qui exécutent Windows 2000 ou Windows NT

Résolution des problèmes liés aux notifications d'opérateur

Si un opérateur ne reçoit pas de notifications, on doit:

- **vérifier** que l'**opérateur** est **disponible** pour en recevoir
- **s'assurer** que le **service d’Affichage** des messages s'exécute sur l'ordinateur de l'opérateur à notifier par une commande **net send**
- **consulter les dernières tentatives de notification** pour connaître la date et l'heure auxquelles la dernière notification s'est produite
- **tester des méthodes de notification individuelles** en dehors de SQL Server en vérifiant qu'on peut envoyer des messages électroniques, avertir un opérateur par radiomessagerie ou exécuter une commande **net send**

Analyse et configuration de l'historique des travaux

- l'Agent SQL Server capture l'état d'exécution des étapes du travail et stocke les informations dans la table système msdb..sysjobhistory
- On peut afficher l'historique des travaux dans SQL Server Enterprise Manager

Analyse de l'historique des travaux

- Si un travail échoue, on peut afficher le **journal d'historique des travaux** pour obtenir des informations sur chaque étape du travail, la raison de son échec et les solutions permettant de résoudre le problème
- **L'historique** des travaux enregistre les éléments suivants:
 - La **date et l'heure** auxquelles l'étape du travail a eu lieu
 - **L'échec ou la réussite** de l'étape du travail
 - **L'opérateur notifié et la méthode de notification**
 - La **durée de l'étape** du travail
 - Les **erreurs ou messages**

Configuration de la taille de l'historique des travaux

- Pour conserver des informations sur chaque travail, on doit augmenter la taille maximale de la table système **sysjobhistory**
- L'historique des travaux est automatiquement écrasé lorsque la taille maximale des lignes est atteinte
- Par défaut, l'extension automatique est définie pour les propriétés des fichiers de la BD **msdb**
- Par défaut, l'option de BD **Vider le journal au point de contrôle** est activée
- Par défaut, la taille maximale de l'historique des travaux est de 1000 lignes
- Par défaut, la taille maximale de l'historique des travaux par travail est de 100 lignes
- Lorsque les **limites sont atteintes**, les lignes sont **supprimées** de la table système **sysjobhistory** selon le mode FIFO

5. Exécution de tâches administratives

5.1 Tâches liées à la configuration

5.2 Tâches de routine liées à l'administration de SQL Server

5.3 Automatisation des tâches de maintenance de routine

5.4 Création d'alertes

5.5 Résolution des problèmes liés à l'automatisation de SQL Server

5.6 Automatisation de travaux sur plusieurs serveurs

5.4 Création d'alertes

- Utilisation d'alertes en réponse à des problèmes potentiels
- Consignation d'événements dans le journal d'applications
- Création d'alertes en réponse à des erreurs SQL Server
- Création d'alertes pour une erreur définie par l'utilisateur
- Réponse à des alertes de condition de performances
- Attribution d'un opérateur de prévention de défaillance

Utilisation d'alertes en réponse à des problèmes potentiels

- Les alertes répondent aux erreurs SQL Server ou aux erreurs définies par l'utilisateur (événements) qui ont été consignées dans le journal d'application de Windows
- Les erreurs SQL Server apparaissent en réponse à des problèmes prédéfinis notamment lorsque l'utilisateur ne dispose pas des autorisations requises pour modifier une table ou lorsque le journal des transactions est saturé
- Pour faire apparaître des messages définis par l'utilisateur, l'application de BD doit appeler l'instruction **RAISERROR**

Exemple de scénario

- Une gestionnaire de comptes souhaite être **notifiée par courrier électronique** chaque fois **qu'un client est supprimé de la BD**
- Elle veut également connaître le nom de l'employé **qui a effectué cette procédure** dans l'éventualité où cette suppression impliquerait une action

Procédure d'alertes pour répondre à la demande de la gestionnaire des comptes

1. **Emp1**, employé du service clientèle, **supprime** le client **Cl1** de la table Client. La procédure stockée **RemoveCustomer** est exécutée et **l'erreur** numéro 50099 s'affiche
2. Cette **erreur** (Evt) est **consignée** dans le **journal d'applications** Windows
3. Le **journal d'applications** Windows **prévient** l'**Agent SQL Server** qu'un événement s'est produit
4. L'**Agent SQL Server** **compare** alors **l'erreur** aux **alertes définies** dans la table système **msdb..sysalerts** qui est conservée en mémoire cache
5. L'**Agent SQL Server** **traite** alors **l'alerte** :
 - a. en consultant la table système **msdb..notifications** afin **d'envoyer un message électronique**
 - b. en consultant la table système **msdb..operators** qui **identifie la personne** à laquelle envoyer la notification

Consignation d'événement dans le journal applications

- Lorsque le service **SQL Server Agent** démarre, il **s'inscrit** auprès de l'Observateur d'événement et **se connecte** au service SQL Server
- Ainsi, **l'Agent SQL Server** peut être informé lorsque des événements SQL Server sont consignés dans le journal d'applications Windows
- L'Agent SQL Server compare alors ces événements aux travaux et alertes mis en cache pour déterminer si une action définie doit être exécutée

Consignation d'événement dans le journal applications (suite)

SQL Server **consigne des événements** dans le journal d'applications Windows **dans les cas suivants :**

- Des **erreurs** SQL Server ayant un **niveau de gravité** compris entre 19 et 25 se produisent
- Des **messages d'erreur** sont **définis pour être consignés** dans le journal d'applications Windows à l'aide de la procédure stockée système **sp_addmessage** ou **sp_altermessage**
- L'instruction **RAISERROR WITH LOG** est exécutée
- La procédure stockée étendue **xp_logevent** est exécutée

Création d'alertes en réponse à des erreurs SQL Server

- Définition d'alertes pour des **numéros d'erreur SQL Server**
- Définition d'alertes pour des **niveaux de gravité d'erreur**

Définition d'alertes pour des numéros d'erreurs SQL Server

- Sous SQL Server Entreprise Manager ou par la procédure stockée système **sp_add_alert**
- La définition de l'alerte est stockée dans la table système **msdb..sysalerts**
- Cette table est conservée en mémoire cache pour améliorer les performances

Définition d'alertes pour des numéros d'erreurs SQL Server (suite)

Lorsqu'on définit des alertes pour un numéro d'erreur SQL Server, on doit tenir compte de :

- Les numéros d'erreurs doivent être **consignés** dans le **journal d'application Windows**
- On peut définir des alertes pour n'importe quel numéro d'erreur fourni par le système SQL Server ou défini par l'utilisateur et stocké dans la table système **master..sysmessages**
- On peut définir **plusieurs alertes pour un numéro d'erreur** SQL Server. Toutefois, chaque alerte doit être limitée à une BD particulière ou s'appliquer à toutes les BD
- Lorsqu'on crée une alerte qui s'applique à toutes les BD, s'assurer que le **message d'erreur est suffisamment clair**
- On peut définir une chaîne particulière dans le texte du message d'erreur outre le numéro d'erreur (**Exemple** : Tentative de connexion par l'utilisateur '%s')

Définition d'alertes pour des niveaux de gravité d'erreurs

Lorsqu'on définit des alertes en utilisant des niveaux de gravité d'erreur comme condition, **on doit tenir compte de :**

- Les erreurs SQL Server pourvues des niveaux de gravité situés **entre 19 et 25** sont **automatiquement consignées** dans le journal d'applications Windows
- Les niveaux de gravité situés **entre 20 et 25** correspondent à des **erreurs fatales**. On doit toujours définir l'opérateur à notifier en cas d'apparition d'erreurs SQL Server à ces niveaux de gravité
- SQL Server fournit des **alertes prédéfinies** qu'on peut utiliser. On doit modifier les alertes prédéfinies sur des erreurs fatales en définissant l'opérateur à notifier et en supprimant le terme « demo » du nom de l'alerte

Définition d'alertes pour des niveaux de gravité d'erreurs (suite)

- On peut **créer une alerte qui se déclenche** lorsqu'une erreur d'un niveau de gravité donné apparaît dans toutes les BD ou dans une BD spécifique
- On peut définir une chaîne particulière dans le texte du message d'erreur outre le niveau de gravité (**Exemple** : créer une alerte avertit, en affichant « espace disque », qu'une erreur de niveau 17 se produit dans une BD quelconque)

Configuration du transfert d'événement

- On peut **configurer l'Agent SQL Server** pour désigner un **serveur qui recevra tous** les messages d'événement dont le niveau de gravité est \geq à une valeur donnée
- On **peut transférer des événements vers un serveur** dont le trafic est $<$ à celui des autres serveurs du domaine

Exemple : On peut configurer le transfert des erreurs ayant un niveau de gravité 18 au minimum vers le serveur Comptabilité. Si une erreur de niveau de gravité 19 se produit sur le serveur, l'événement est transféré automatiquement sur le serveur Comptabilité qui devra résoudre le problème

Création d'alertes pour une erreur définie par l'utilisateur

- Création du message d'erreur
- Déclenchement de l'erreur à partir de l'application de BD
- Définition d'une alerte pour le message d'erreur

Création du message d'erreur

- Sous SQL Server Entreprise Manager ou par la procédure stockée système **sp_addmessage**
- Les **numéros des erreurs** définies par l'utilisateur doivent être **> à 50000**. Les numéros d'erreur **< à 50000** sont réservés aux erreurs système SQL Server prédéfinies
- Toutes les erreurs définies par l'utilisateur sont **stockées** dans la table système **sysmessages** de la BD **master**
- Les messages d'erreur peuvent comporter des **paramètres pour capturer des détails spécifiques** afin que les informations appropriées soient fournies pour le statut de détail ou pour résoudre le problème

Création du message d'erreur (suite)

- Les messages d'erreurs SQL Server s'affichent dans la **langue sélectionnée lors de l'installation**. Si vous administrez un environnement SQL Server multilingue, vous pouvez également créer des messages définis par l'utilisateur pour d'autres langues
- On **doit consigner le message** d'erreur dans le journal d'applications Windows si on envisage de déclencher une alerte pour ce message

Exemple de création du message d'erreur

```
EXEC sp_addmessage 50099, 16,  
    'Client %d supprimé par %s', 'French', 'true'
```

- Cet exemple crée un message d'erreur défini par l'utilisateur numéro 50099, de niveau de gravité 16 et qui est consigné dans le journal d'applications Windows (**true**) lorsque l'erreur se produit
- Le paramètre %d est remplacé par le numéro du client supprimé et le paramètre %s par le nom de l'utilisateur qui a exécuté l'instruction DELETE sur la table **Client**

Déclenchement de l'erreur à partir de l'application de BD

- Utiliser l'instruction **RAISERROR** dans une procédure stockée ou un déclencheur
- RAISERROR renvoie un message d'erreur défini par l'utilisateur et définit un indicateur système (dans la fonction système @@error) pour enregistrer une erreur survenue

Déclenchement de l'erreur à partir de l'application de BD (Exemple)

```
CREATE PROCEDURE removecustomer  
@CustomerID varchar(5)=NULL  
AS  
.  
DECLARE @username varchar(60)  
SELECT @username=suser_sname()  
BEGIN TRANSACTION  
    DELETE Customers  
    WHERE CustomerID=@CustomerID  
    RAISERROR (50099, 16, 1, @CustomerID, @username)  
COMMIT TRANSACTION
```

Définition d'une alerte pour le message d'erreur

- Une alerte est créée pour l'erreur numéro 50099 afin qu'un message électronique contenant le texte du message d'erreur soit envoyé au gestionnaire des comptes
- Lorsqu'un utilisateur exécute la procédure stockée **RemoveCustomer**, l'erreur 50099 est déclenchée, elle est consignée dans le journal d'applications Windows
- L'alerte pour le numéro d'erreur est déclenchée et envoie au gestionnaire des comptes un message électronique contenant le texte du message d'erreur

Résultats : Error: 50099, Severity: 16, State 1

A. ETTALBI Client 732 supprimé par ACCOUNTING\evacorets

Réponse à des alertes de condition de performance

- On peut utiliser des **alertes à la fois en réponse** à des erreurs **SQL Server** et à des **conditions de performances** SQL Server comme celles définies dans le Moniteur système Windows
- Par exemple**, on peut créer une alerte de condition de performances qui se déclenche lorsque le journal des transactions de la BD dépasse 75% de sa capacité
- La réponse à l'alerte peut exécuter un travail qui sauvegarde le journal des transactions et notifie l'administrateur de la BD
- On peut créer des alertes de condition de performances sur des ressources SQL Server comportant certains des objets suivants : ***Méthodes d'accès, Gestionnaires de tampons, Gestionnaire de cache, Bases de données, Verrous, Statistiques SQL***

Attribution d'un opérateur de prévention de défaillance

-On peut attribuer un opérateur de prévention de défaillance pour répondre à une alerte déclenchée par l'échec des notifications envoyées par radiomessagerie à des opérateurs définis

-**Par exemple**, Si un utilisateur n'est pas de service le Jeudi et qu'une alerte s'est déclenchée ce jour-là à 01:30, l'opérateur de prévention de défaillance est contacté

Un **opérateur de prévention** de défaillance **est contacté** lorsque **toutes ces conditions sont réunies** :

-Des **notifications** par radiomessagerie ont été **créées** pour la réponse à l'alerte

-Aucun des opérateurs à prévenir par radiomessagerie n'est de service

-Un opérateur de prévention de défaillance est défini

-La session de messagerie de **l'Agent SQL Server est démarrée**

Attribution d'un opérateur de prévention de défaillance (suite)

Lorsqu'on attribue un opérateur de prévention de défaillance, on doit tenir compte de :

- Les informations de l'opérateur de prévention de défaillance sont mises en mémoire cache, elles ne dépendent pas donc de la connexion à la BD **msdb**
- On ne peut avoir qu'un seul opérateur de prévention de défaillance
- **On ne peut pas supprimer un opérateur qui a été désigné comme opérateur de prévention de défaillance.** Mais, on peut supprimer l'attribution d'opérateur de prévention de défaillance puis supprimer l'opérateur

5. Exécution de tâches administratives

5.1 Tâches liées à la configuration

5.2 Tâches de routine liées à l'administration de SQL Server

5.3 Automatisation des tâches de maintenance de routine

5.4 Création d'alertes

5.5 Résolution des problèmes liés à l'automatisation de SQL Server

5.6 Automatisation de travaux sur plusieurs serveurs

5.5 Résolution des problèmes liés à l'automatisation de SQL Server

Si les travaux, alertes ou notifications automatisés ne fonctionnent pas correctement alors :

- Vérifier que l'Agent SQL Server est **démarré**, que le rôle **sysadmin** est attribué à son compte de connexion et que le mot de passe est valide
- Vérifier que le travail, la planification, l'alerte et l'opérateur sont **activés**
- Vérifier que les utilisateurs disposent des autorisations pour exécuter ces types de travaux et que le compte d'utilisateur de domaine utilisé en tant que compte proxy dispose des autorisations pour exécuter les travaux

5.5 Résolution des problèmes liés à l'automatisation de SQL Server (suite)

- **Consulter les messages d'erreurs** dans le journal d'applications Windows et les journaux d'erreurs de l'Agent SQL Server pour déterminer l'origine du problème et le résoudre
- **Vérifier que la taille du fichier** et la taille de croissance de la BD **msdb** correspond au nombre de lignes conservées dans la table système **sysjobhistory**
- **Vérifier le fonctionnement du client de messagerie**

Résolution des problèmes liés aux alertes

➤ Facteurs pouvant entraîner un retard de traitement des alertes

- Le journal d'application Windows est plein
- L'utilisation du processeur est inhabituellement élevée
- Le nombre de réponses aux alertes est élevé

➤ Résolution du retard de traitement des alertes

- Désactivez temporairement l'alerte
- Augmenter le délai entre les réponses
- Corriger le problème des ressources globales
- Vider le journal d'applications Windows

5. Exécution de tâches administratives

5.1 Tâches liées à la configuration

5.2 Tâches de routine liées à l'administration de SQL Server

5.3 Automatisation des tâches de maintenance de routine

5.4 Création d'alertes

5.5 Résolution des problèmes liés à l'automatisation de SQL Server

5.6 Automatisation de travaux sur plusieurs serveurs

5.6 Automatisation de travaux sur plusieurs serveurs

- Définition d'un **serveur principal** qui définit, planifie et gère les travaux sur tous les serveurs cibles et de **serveurs cibles**
- Utiliser SQL Server Enterprise Manager ou la procédure stockée système **sp_msx_enlist** pour définir un serveur principal
- On doit lui inscrire au **moins un serveur cible**
- Une ligne sera stockée dans la table système **systargetservers**
- Un **compte et un mot de passe** d'ouverture de session SQL Server sont **automatiquement créés** pour **chaque serveur cible** avec le suffixe **msx_probe**

5.6 Automatisation de travaux sur plusieurs serveurs (suite)

- L'Assistant crée un opérateur **MSXOperator** sur le serveur principal et sur chaque serveur cible
- Le serveur principal représente généralement le serveur d'un service principal ou d'une unité d'entreprise principale. Dans une structure plus petite, il dessert toute l'entreprise
- On peut aussi désigner le serveur principal comme serveur de transfert d'événements
- Les **serveurs cibles** sont attribués à un **seul serveur principal**
- Ils doivent être situés dans le **même domaine Windows** que le serveur principal

Définition des travaux sur plusieurs serveurs

- Après avoir défini le serveur principal et les serveurs cibles, on peut **créer sur le serveur principal des travaux** qui seront **exécutés sur un ou plusieurs serveurs cibles**
- Pour traiter les travaux dans un environnement multiserveurs, SQL Server exécute la procédure suivantes :
 - Le serveur principal poste des travaux pour les serveurs cibles dans une liste de téléchargement de la table système **msdb..sysdownloadlist**
 - Les serveurs cibles se connectent périodiquement au serveur principal pour déterminer si des travaux nouveaux ou mis à jour ont été postés
 - A la fin du travail, le serveur cible envoie l'état du résultat du travail

Modification des définition de travaux sur plusieurs serveurs

- Les définitions des travaux ne peuvent être modifiés sur le serveur cible ; les **modifications doivent être effectuées sur le serveur principal**
- SQL Server Entreprise Manager **poste automatiquement** les instructions nécessaires dans la liste de téléchargement