

**Département :** Génie Logiciel

**Filière :** Génie Logiciel

**Niveau :** 2<sup>ème</sup> année

**Module (M4.2) :** Administration des bases de données

**Elément de Module (M4.2.2) :** Administration  
des bases de données SQL Server

**A. ETTALBI**

[ahmed.ettalbi@um5.ac.ma](mailto:ahmed.ettalbi@um5.ac.ma)

[a.ettalbi@um5s.net.ma](mailto:a.ettalbi@um5s.net.ma)

# **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** (Online Analytical Processing)

- Aide à la décision (Exemple : Entreprise)

**OLTP** (Online Transactionel Processing)

- Transactions (Exemple : Banque, Billetterie)
- **Récupère** les données à un état de cohérence antérieur en cas de panne

# 1.1 Présentation de SQL Server (suite)

## *Accès 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)*

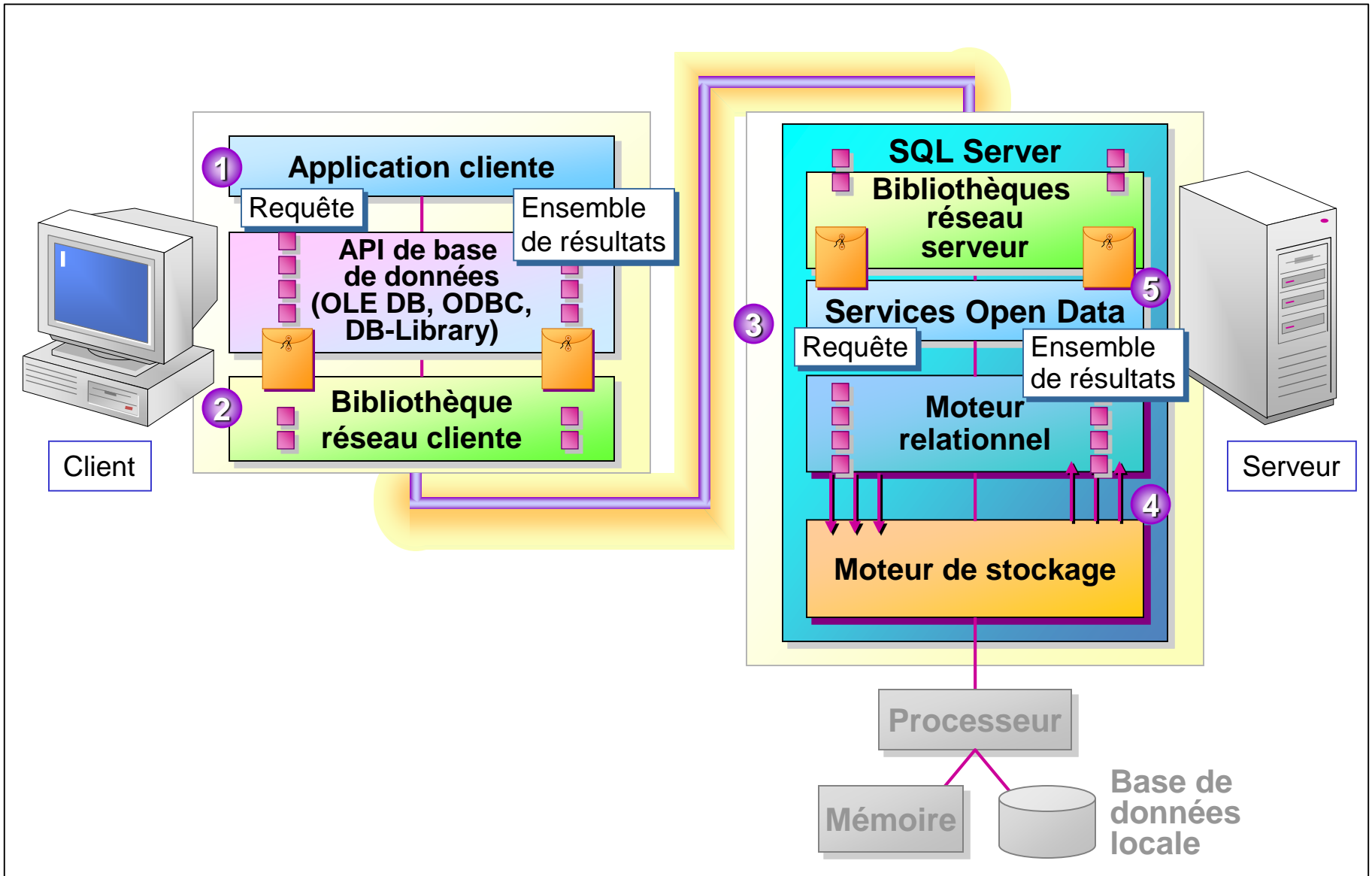
### ➤ Moteur relationnel :

- Analyse, optimise et exécute les instructions T-SQL : 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*

### **Services installés par défaut (1/4)**

#### ***MSSQLServer***

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

# 1.1 Présentation de SQL Server (suite)

## *Services SQL-Server*

### **Services installés par défaut (2/4)**

#### ***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)*

### *Services installés par défaut (3/4)*

#### *Microsoft Distributed Transaction Coordinator*

- Gère les transactions distribuées
- Transactions utilisant des sources de données différentes

# 1.1 Présentation de SQL Server (suite)

## *Services SQL-Server : (suite)*

### **Services installés par défaut (4/4)**

#### ➤ **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 <b>sauf CE</b>	W-2000, W-NT, W-Me, W-95 et W-98
SQL Server 2000 <b>CE</b>	Windows <b>CE</b>

# 1.2 Intégration de SQL Server

## *Intégration aux SE (côté Serveur)*

Version SQL Server	Version Windows
SQL Server 2008	W-2000, W-NT, W-Me, W-95, W-98 <b>et W-CE</b>

# 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

# 1.2 Intégration de SQL Server

## *Intégration de SQL Server à W-2000*

### - **Sécurité :**

Nom utilisateur et mot de passe **unique**



**Accès 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,
  - Donc, 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 :**
  - repose sur des extensions du modèle COM (Component Object Model) et de MS-Transaction Server,
  - 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**,
  - Cela 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
- 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 et les définitions de toutes les BD et de tous les objets de BD,

**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
<b>syslogins</b>	master	Contient une ligne pour chaque compte d'ouverture de session
<b>sysmessages</b>	master	Contient une ligne pour chaque avertissement ou erreur
<b>sysdatabases</b>	master	Contient une ligne pour chaque base de données
<b>sysusers</b>	toutes	Contient une ligne pour chaque utilisateur W-2000, groupe W-2000, utilisateur SQL Server ou rôle SQL Server
<b>sysobjects</b>	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
<b>sp_help</b> [objet]	Fournit des informations sur l'objet de BD spécifié
<b>sp_helpdb</b> [BD]	Fournit des informations sur la BD spécifiée
<b>sp_helpindex</b> [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
<b>DB_ID</b>	Nom	Identificateur de la BD
<b>USER_NAME</b>	ID	Nom de l'utilisateur
<b>COL_LENGTH</b>	Colonne	Largeur de la colonne
<b>STATS_DATE</b>	Index	Date de la dernière MAJ des statistiques de l'index
<b>DATALENGTH</b>	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
<b>INFORMATION_SCHEMA.TABLES</b>	Liste des tables de la BD
<b>INFORMATION_SCHEMA.COLUMNS</b>	Information sur les colonnes définies dans la BD
<b>INFORMATION_SCHEMA.TABLES_PRIVILEGES</b>	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 user 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 :

- utilisateurs ou des groupes W-2000,
- ou des comptes d'ouverture de session SQL Server,
- sont spécifiques à une BD

### Rôles :

permettent de regrouper des utilisateurs auxquels on peut attribuer les mêmes 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
<b>dbcreator</b>	Création et modification des BD
<b>diskadmin</b>	Gestion des fichiers physiques
<b>processadmin</b>	Gestion des processus SQL Server
<b>securityadmin</b>	Gestion et audit des noms de connexion serveur
<b>serveradmin</b>	Configuration des paramètres de serveur
<b>setupadmin</b>	Installation de la réplication
<b>sysadmin</b>	Exécution de toutes les activités
<b>bulkadmin</b>	Exécution de l'instruction BULK INSERT

# Rôles fixes de base de données

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

# Administration d'une BD SQL-Server

## Tâches courantes d'administration (suite) :

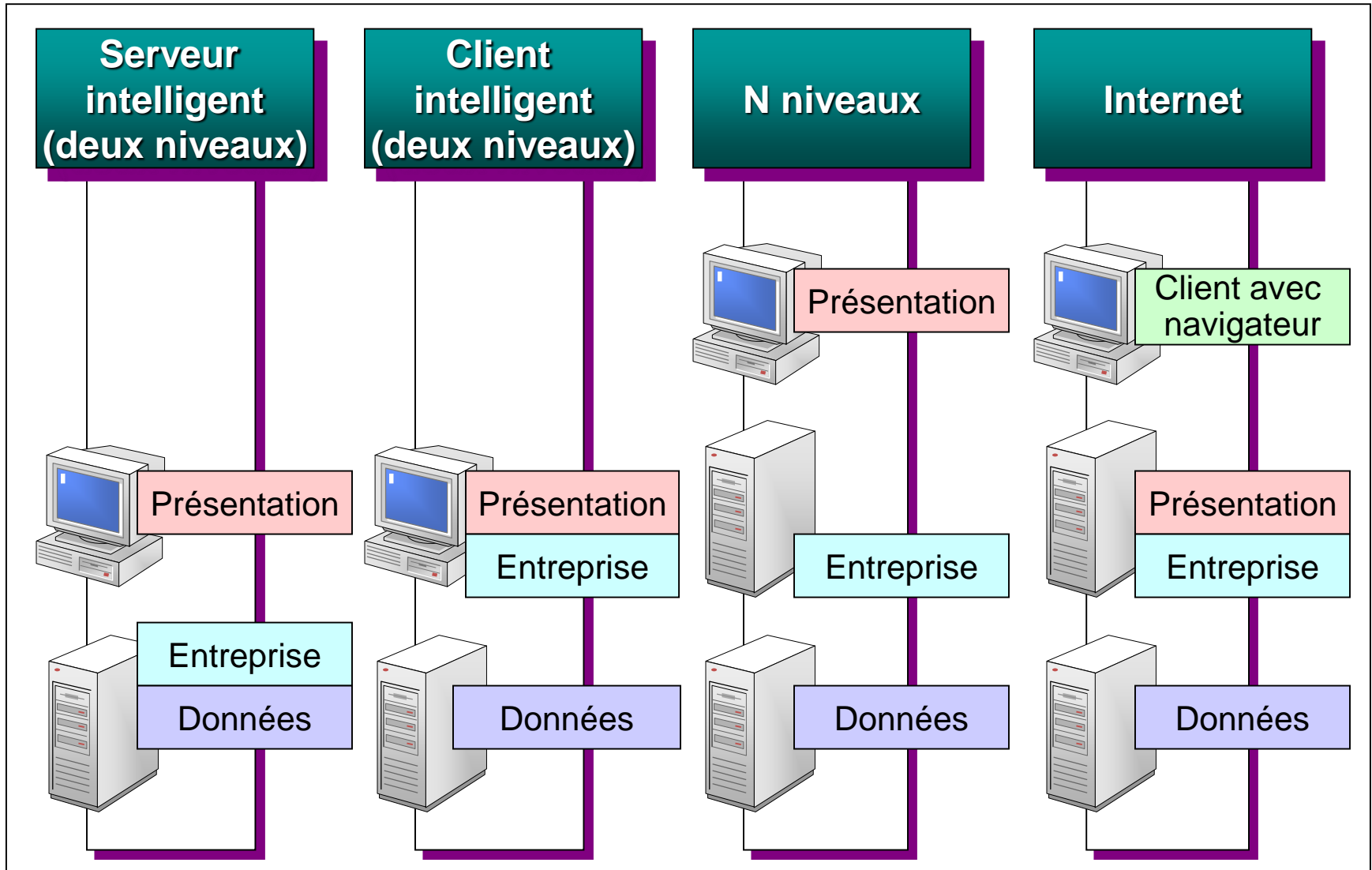
- Gestion des activités récurrentes :
  - importation et exportation des données,
  - sauvegarde et restauration de la BD,
  - surveillance et optimisation de la BD.



# 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
<b><i>Présentation</i></b>	Comprend la logique de présentation des données et de l'application aux utilisateurs. Elle est toujours implémentée sur un ordinateur client
<b><i>Entreprise</i></b>	Comprend la logique de l'application et les règles d'entreprise. SQL Server peut intervenir au niveau de cette couche
<b><i>Données</i></b>	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 à l'aide 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**

# Exemple d'interface API de BD

- 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

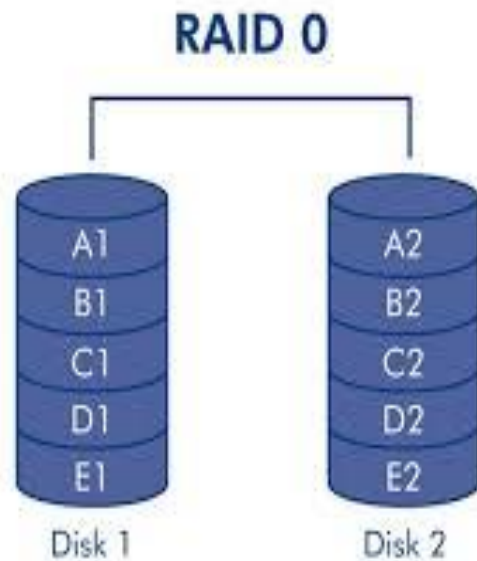
# 2.3 Système RAID

## (Redundant Arrays of Inexpensive Disks)

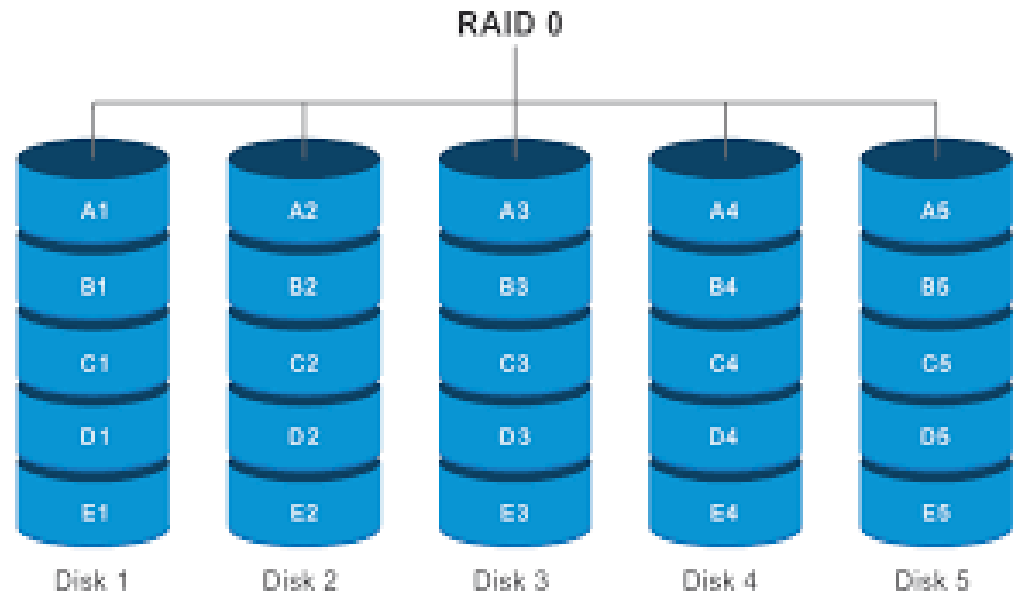
Niveau RAID	Description	Avantages/Inconvénient
<b>RAID 0</b>	Entrelacement	Offre de bonnes performances de lecture et d'écriture mais pas de disponibilité
<b>RAID 1</b>	Mise en miroir et duplexage du disque	Offre une très bonne redondance et de bonnes performances de lecture et d'écriture
<b>RAID 5</b>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

# RAID 0

## RAID 0 avec 2 disques



## RAID 0 avec plusieurs disques



**Nb minimum de disques : 2**

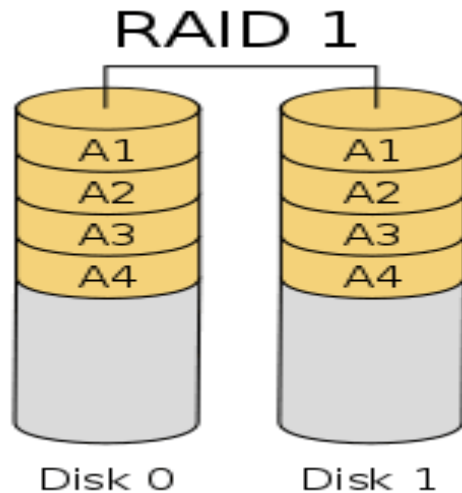
**Avantage : performance d'E/S,  
tous les disques sont exploités**

**Inconvénient : pas de disponibilité de données**

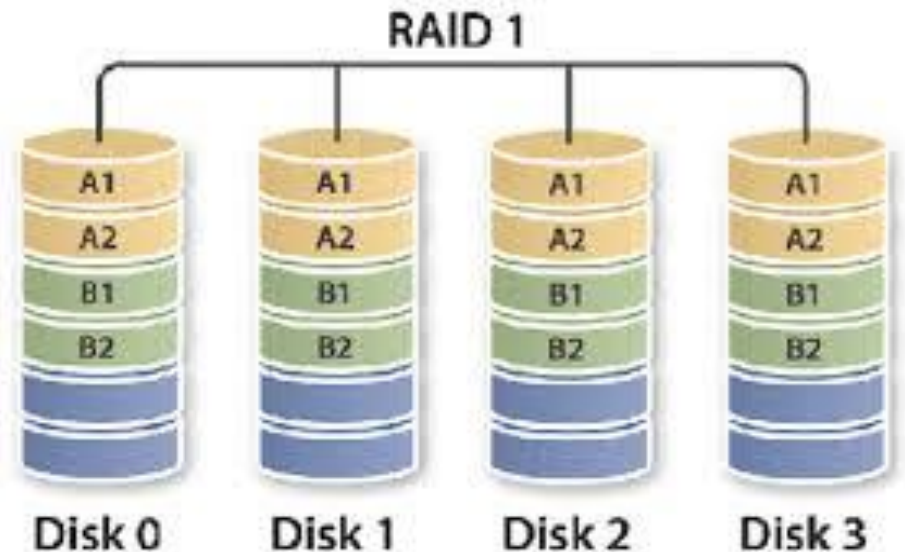
**Tolérance aux pannes : 0 disque**

# RAID 1

## RAID 1 avec 2 disques



## RAID 1 avec plusieurs disques



Nb minimum de disques : 2

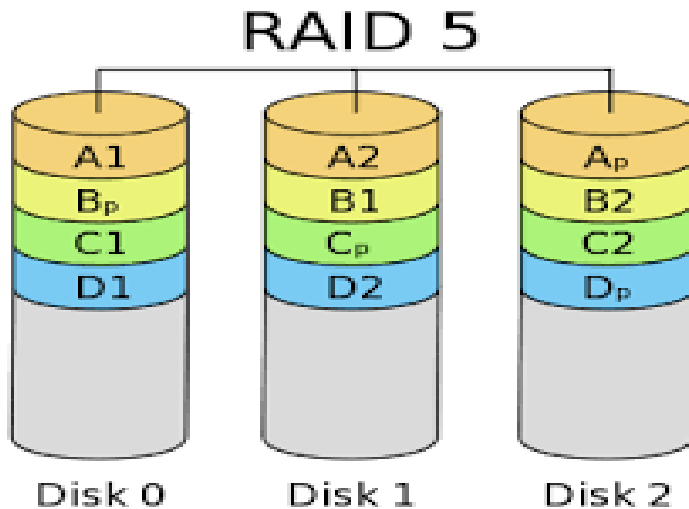
Avantage : disponibilité des données

Inconvénients : performance de S,  
(N-1) disques non exploités

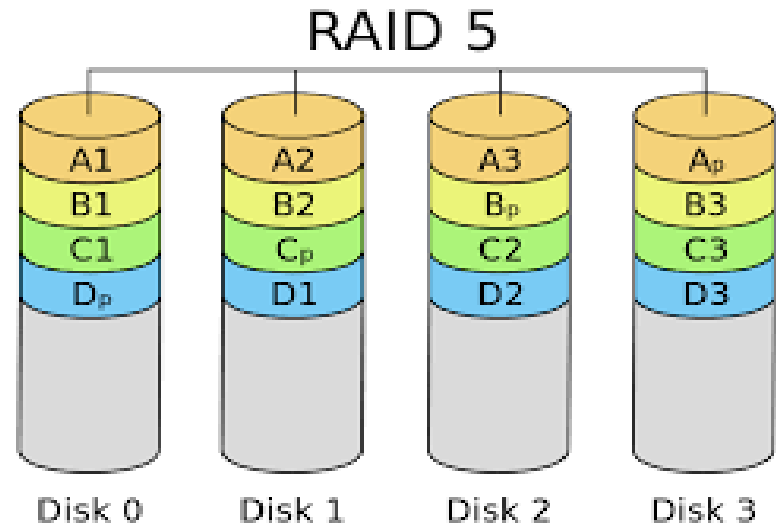
Tolérance aux pannes : (N-1) disques

# RAID 5

## RAID 5 avec 3 disques



## RAID 5 avec plusieurs disques



**Nb minimum de disques : 3**

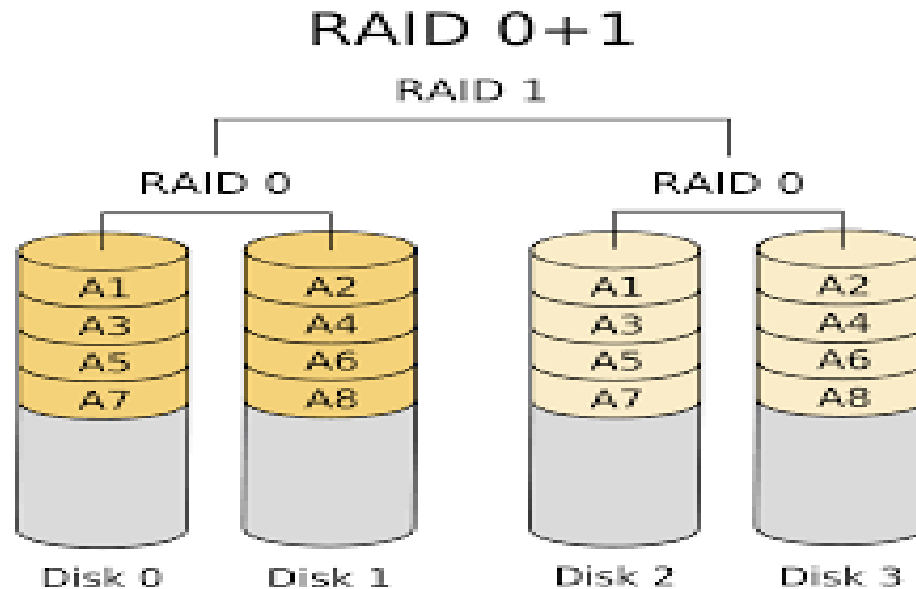
**Avantages :** performances d'E (RAID 0),  
disponibilité des données (RAID 1)

**Inconvénients :** performance de S dû à la complexité de  
la Fonction de parité

**Tolérance aux pannes : 1 disque**

# RAID 01

## RAID 01 avec 4 disques



**Nb minimum de disques : 4**

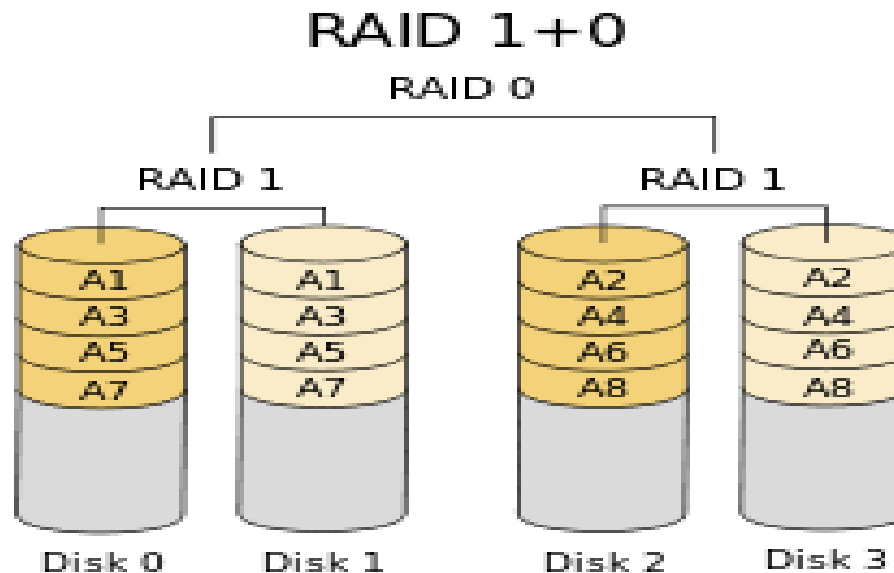
**Avantage : disponibilité des données (RAID 1),  
performance d'E (RAID0)**

**Inconvénient : performance de S, beaucoup de disques**



# RAID 10

## RAID 10 avec 4 disques



**Nb minimum de disques : 4**

**Avantage : disponibilité des données (RAID 1),  
performance d'E (RAID0)**

**Inconvénient : performance de S, beaucoup 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**),
  - un ou plusieurs fichiers journaux (**.ldf**),
  - peut comporter des fichiers data secondaires (**.ndf**)
- **Ces fichiers physiques ont à la fois :**
  - des **noms de fichiers de SE (physiques)**,
  - 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)

- BD créée → 1 copie de **model** copiée dans la BD
- les données sont stockées dans des **pages**,
- **page** : **l'unité** de travail de SQL Server
- **page** : bloc d'espace disque contigu de 8 KO
- Une BD peut stocker 128 pages par Mo
- **8060 octets** de données au maximum par ligne



# 3.1 Mode stockage des données (suite)

- Stockage des tables et index :

  - 8 pages** contiguës (64 Ko)

  - ➔ Une BD comporte 16 extensions par Mo

- Les fichiers **LOG** : informations requises pour la restauration de la BD en cas de panne du système.

## 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,
- Le propriétaire et le créateur de la nouvelle BD doivent être **autorisés** à utiliser la **BD master** (Utilisation des tables système **sysdatabases** et **sysaltfiles** de la BD master)

## 3.2 Processus de création d'une BD (suite)

- SQL Server permet 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**

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

- Fichier principal (.mdf)
- Fichiers secondaires (.ndf)
- Fichier journal (.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 % (10% par défaut)
- **Classement** (par défaut une BD hérite du classement de l'instance de SQL Server dans laquelle elle a été créée)

# 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

# 3.4 Gestion de la croissance des fichiers journaux et de données

- **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 (même sur d'autres disques)

# 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,
  - envisager d'augmenter la taille **manuellement** si nécessaire,
- ➔ Choix du moment,

# Choix entre croissance automatique ou manuelle de la taille des fichiers

Dans un environnement de bureau (**production restreinte**) :

- configurer les fichiers de Base de Données pour que leurs espaces mémoire **augmentent automatiquement**,

➔ réduction des tâches d'administration.



## 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** (AUTO\_SHRINK)
- **Par défaut**, l'option de compactage automatique des fichiers est **désactivée** dans toutes les éditions **sauf SQL Server Desktop Edition**

# Activités de SQL Server au cours du compactage automatique

- **Réduit** les fichiers de données et journaux de transactions si l'espace inutilisé dépasse 25%
- **Supprime** autant d'espace libre que possible
- Le journal des transactions n'est réduit que s'il ne contient aucune partie **active**
- Exécute cette tâche en **arrière-plan**

## 3.6 Compactage manuel d'une BD

- **Intérêt** : choisir le moment
- **Méthodes de compactage** :
  - par SQL Server Enterprise Manager
  - ou ***BDCC SHRINKDATABASE*** ou ***DBCC SHRINKFILE***
- On peut alors :
  - réduire les fichiers de données et LOG en **groupe** ou **individuellement**
  - réduire **individuellement** les fichiers de données et LOG dont la **taille est inférieure** à leur taille de création d'origine

## 3.7 Compactage des fichiers LOG

Dans ce cas, SQL Server :

- **réduit les parties inactives** qui dépassent la taille souhaitée
- **si cela ne suffit pas, 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 LOG.

## 3.8 Suppression d'une BD

- Par ***DROP DATABASE*** ou sous SQL Server Entreprise 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**

## 3.9 Emplacement des fichiers et journaux de BD

### -Facteurs influencés

- **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

## 3.9 Emplacement des fichiers et journaux de BD (suite)

- **Répartir** les données sur **plusieurs disques physiques**
- SQL Server peut exécuter :
  - des **recherches parallèles** même pour une seule **table** si le groupe de fichiers de la table contient plusieurs fichiers
- utiliser une solution **RAID**, et des **groupes de fichiers** définis par l'utilisateur.

## 3.9 Emplacement des fichiers et journaux de BD (suite)

- Créer le journal **LOG** sur un **disque séparé**
- Placer la BD **tempdb** sur un sous-système d'E/S **rapide distinct** de celui utilisé pour les BD utilisateur,
- **Distribuer** la BD **tempdb** sur plusieurs disques pour améliorer les performances.



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

- Pour des performances optimales, **opter pour une solution RAID matérielle**
- RAID matérielle permet également de **remplacer un lecteur défectueux** sans arrêter le système
- **RAID1**: Disques miroirs (Redondance)
- **RAID5**: Entrelacement avec parité (Performance et Redondance)
- **RAID10 ou RAID01** : (Performances maximales)

## 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
  - en utilisant des **threads parallèles** pour le traitement des requêtes
- **facilitent** aussi la **maintenance** de la BD

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

➤ Si on dispose de **plusieurs disques**

**physiques :**

- 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

- **Groupe de fichiers principal :**

contient le fichier de données principal avec les tables système

- **Groupe de fichiers défini par l'utilisateur :**

contient des fichiers data regroupés pour des besoins d'allocation et d'administration

# Création de groupes de fichiers définis par l'utilisateur

- Lors de la création de la BD ou ultérieurement
  - Create Database, Alter Database
  - sous SQL Server Enterprise Manager)
- On peut choisir le **groupe de fichiers par défaut**  
il contient **tables** et index pour lesquels **aucun groupe** de fichiers n'a été spécifié

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

- un groupe de fichiers est désigné comme groupe de fichiers **par défaut**
- Au moment de la création de la BD, le groupe de fichiers **par défaut** est défini comme **groupe de fichiers principal**
- On peut le changer au moment de la création

# Taille du groupe de fichiers par défaut

- Si le **groupe de fichiers par défaut** reste le groupe de fichiers principal :
  - il faut lui donner une **taille appropriée**
  - ou le configurer pour que sa taille **augmente automatiquement**
- Si le groupe de fichiers principal est **saturé** :
  - **impossible d'ajouter** des informations dans les **tables système**

# Affichage des informations sur les groupes de fichiers

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

**sp\_helpfile** *[[@filename=]'nom']*

**sp\_helpfilegroup** *[groupe\_fichier]*



# Utilisation de groupes de fichiers pour améliorer les performances

## Répartition des données sur plusieurs disques

- Table créée ➔ un groupe de fichiers défini par l'utilisateur

### - Avantages :

- remplissage proportionnel

- remplissage en parallèle

- emplacement des fichiers sur un ou plusieurs disques physiques

# Utilisation de groupes de fichiers pour améliorer la maintenance

- Sauvegarder/restaurer **des fichiers** ou des groupes de fichiers **au lieu de** sauvegarder/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 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

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

## Avantages :

- **Accès parallèle** aux données (Threads séparés pour chaque fichier)
- **Répartition de la charge** sur plusieurs disques.
- Cette approche crée un **regroupement logique unique**,
- Donc **facile à configurer** et à gérer pour un Administrateur Système ou de BD.

## 3.12 Planification de la capacité

- Tâche importante d'un Admin système,
- 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,
  - déterminer la **configuration matérielle requise.**

# 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 (et prévisions de croissance),
- La **quantité de données** dans les tables (prévisions de croissance),
- Le **nombre et la taille des index** (taille de la valeur de la clé, nombre de lignes, 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**, dépend de :
  - la **quantité** et de la **fréquence** des modifications,
  - la taille de chaque **transaction**,
  - la fréquence de sauvegarde et de vidage du **journal**

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

## ➤ La taille des **tables système**

- nombre d'utilisateurs,
- nombre d'objets,
- ne constitue généralement pas un pourcentage élevé de la taille de la BD



# 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**

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

- Evaluation effectuée en déterminant :
  - le **nombre total de lignes**,
  - la **taille des lignes**,
  - le **nombre de lignes** contenues dans une page,
  - 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 (NOL)**,
- Déterminer le **nombre de lignes contenues dans chaque page** de données. ( $NLP = 8060 / NOL$ ),
- Prévoir le nombre de ligne dans la table (**NLT**)

➔ le nombre de pages requises pour stocker la table est :  $NPT = NLT / NLP$

# Remarques sur les performances

- Utiliser une solution **RAID** pour améliorer les **performances** ou assurer la **tolérance de panne**
- Placer les **fichiers** données et Log 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
- SQL Server peut authentifier lui-même les comptes de connexion

# 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,
  - le serveur SQL Server **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 **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

# 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**

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

- **Kerberos** émet un ticket d'octroi de ticket (**TGT**: Ticket-Granting Ticket)
- Le TGT contient les données cryptées **confirmant l'identité**
- Le TGT est envoyé au centre de distribution de clé (**KDC** : Key Distribution Center)

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

- Lorsqu'un client demande l'accès à un service :
  - Il envoie le **TGT** au centre KDC
  - le centre KDC envoie au client un **ticket de session** pour le service demandé
  - Le client **présente** ensuite au **service** le ticket de session qui confirme son identité

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

- Ticket de session réutilisé jusqu'à **expiration**,
- **Délai d'expiration** 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 et délégation

## 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**,
- **conserver les références** du client d'origine à chaque changement de serveur

### ➤ Pour utiliser la délégation:

- tous les serveurs auxquels on veut se connecter doivent exécuter W-2000
- avec prise en charge du protocole **Kerberos**
- 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

- A partir **d'utilisateurs** et de **groupes W-2000** déjà définis
- **ou** créer de nouveaux comptes de connexion par défaut
- Sont stockés dans la table système **sysxlogins**
- 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

- **Objectif** : permettre à un compte d'utilisateur ou de groupes W-2000 de se connecter à SQL Server
- Par SQL Server Enterprise Manager ou **sp\_grantlogin**
- Seuls les **Admins système** ou Responsables de la **sécurité** peuvent exécuter **sp\_grantlogin**

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

- 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 puis les supprimer de SQL Server

# Ajout d'un compte de connexion SQL Server

- **Objectif** : créer un compte de connexion SQL Server
- Par SQL Server Enterprise Manager ou **sp\_addlogin**
- Seuls les **Admins** système ou Responsable de la **sécurité**
- ajoute un enregistrement dans la table **sysxlogins**



# Comptes de connexion par défaut

## ➤ **BUILTIN\Administrateurs** :

- pour tous les **admins W-2000**
- possède **tous les droits** sur SQL Server et toutes les BD

## ➤ **sa** (administrateur système) :

- activé uniquement lorsque SQL Server utilise  
**l'authentification en mode mixte**
- possède **tous les droits** sur SQL Server et toutes les BD

# 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 l'ajout des comptes de connexion à SQL Server :
  - **mapper** ces comptes sur des comptes d'utilisateur ou des rôles,
  - **spécifier les BD** auxquelles les utilisateurs doivent accéder

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

- **sysusers** de chaque BD contient une ligne pour :
  - chaque utilisateur/groupe W-2000,
  - chaque utilisateur SQL Server
  - chaque rôle dans la BD
- Les autorisations de ces users sont stockées dans la table **sysprotects** de la BD active

# Attribution des comptes de connexion à des comptes d'utilisateur

- L'ajout des comptes d'utilisateur :
  - par SQL Server Entreprise Manager
  - ou par **sp\_grantdbaccess**
- L'ajout peut aussi être fait **lors de la création** des comptes de connexion
- **Qui peut faire l'ajout ?**
  - propriétaire de la BD
  - les administrateurs

# Compte d'utilisateur *dbo*

- **sa** et les membres **sysadmin** mappés sur **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
- On peut lui **attribuer** des autorisations
- On peut **supprimer/ajouter** **guest** de toutes les BD 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** d'administration au niveau du **serveur**
- Sont gérés indépendamment des BD utilisateur
- Sont stockés dans la table système **sysxlogins**

# Rôles fixes de serveur

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

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

- Par **sp\_addsrvrolemember** ou SQL Server Enterprise Manager
- **Rôles fixes de serveur figés** : pas d'ajout ni modif ni supp
- Les **membres d'un rôle** fixe de serveur **peuvent ajouter** d'autres comptes de connexions à ce rôle
- Suppression d'un membre par SQL Server Enterprise Manager ou **sp\_dropsrvrolemember**

# Rôles fixes de BD

- Permettent de regrouper des **privilèges** d'administration 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 tout user de BD
- Ne peut pas être supprimé
- Permet de maintenir les autorisations par défaut des users d'une BD
- Ne peut pas se voir attribuer des utilisateurs, des groupes ou des rôles (**par défaut** ils y appartiennent)
- Il est présent dans chaque BD (système ou user)

# Tâches liées au rôle **public**

- Exécuter des instructions qui ne requièrent pas d'autorisations (comme 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 système **sp\_addrolemember**
- Seuls les **membres** du rôle **db\_owner** peuvent exécuter **sp\_addrolemember** pour tous les rôles de la BD
- Rôles fixes de BD **figés** : pas d'ajouter, modifier ou supprimer



# 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
- Suppression par SQL Server Entreprise Manager ou la procédure 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

# Ajout de Rôles de BD définis par l'utilisateur

- Par SQL Server Enterprise Manager ou **sp\_addrole**
- Une entrée est ajoutée à la table **sysusers** de la BD active
- Seuls les membres du rôle **db\_securityadmin** ou **db\_owner** peuvent exécuter **sp\_addrole**

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

- Par SQL Server Enterprise Manager ou **sp\_addrolemember**
- Seuls les membres de **sysadmin**, **db\_securityadmin** et **db\_owner** ou le **propriétaire** du rôle peuvent exécuter **sp\_addrolemember**

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

- Ajout d'un compte de sécurité à un rôle
  - ➔ **Propagation** des autorisations
- On ne peut pas créer des **rôles rékursifs**.
  - ➔ Par exemple, *Rôle1* ne peut pas être ajouté en tant que membre de *Rôle2* si *Rôle2* est déjà membre de *Rôle1*

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

➤ **Suppression d'un rôle SQL Server de la BD active :**

- par SQL Server Enterprise Manager ou **sp\_droprole**

➤ **Suppression d'un compte de sécurité d'un rôle SQL Server :**

- par SQL Server Enterprise Manager ou **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

### Types d'autorisations

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



# Autorisations sur les instructions

- **Impliquent** la création d'objets de BD

CREATE DATABASE, CREATE TABLE, CREATE VIEW, CREATE PROCEDURE, CREATE RULE, CREATE DEFAULT, CREATE FUNCTION, BACKUP DATABASE, BACKUP LOG

- Seuls les membres de **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

### Exemples :

- utilisateur **ajouté à sysadmin**
  - ➔ hérite des autorisations associées
- utilisateur **owner** d'une table
  - ➔ peut réaliser toutes les activités liées à la table (LMD, LDD, autorisations,...)

# Octroi, refus et révocation d'autorisations

- **Autorisation** : soit accordée (**GRANT**), refusée (**DENY**) ou révoquée (**REVOKE**)
- Autorisation ni accordée ni refusée → neutre (révoquée)
- Les autorisations sont stockées 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 de **sysadmin**, **db\_owner** et **db\_securityadmin** et les **owners** d'objets peuvent accorder des autorisations sur ces objets
- Les autorisations **varient selon l'objet** sélectionné (EXECUTE pour une procédure stockée, SELECT, INSERT, ... pour une table ou vue)

# Refus d'une autorisation d'accès (**DENY**)

- **Restreindre** les autorisations d'un user ou d'un rôle
- **Supprime** les autorisations précédemment accordées
- **Désactive** les autorisations héritées d'un autre rôle
- On ne peut refuser des autorisations que pour la **BD active**
- Seuls les membres de **sysadmin**, **db\_owner** et **db\_securityadmin** et les **owners** d'objets peuvent refuser les autorisations

# Révocation d'autorisations accordées et refusées (**REVOKE**)

- **Désactiver** une autorisation accordée ou refusée
- 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 de **sysadmin**, **db\_owner** et **db\_securityadmin** et les **owners** 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 :
  - **sa** : attribué à **sysadmin** et ne peut être supprimé
  - **BUILTIN\Administrateurs** : auxquels sont mappés les **Administrateurs** de W-2000 et qui est un membre de **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.
- Donc :
  - **décider** si vos BD doivent posséder un compte **guest** ou non
  - **définir** dans ce cas **les autorisations** attribuées à ce compte dans ces BD

# Gestion de la sécurité dans SQL Server (Suite)

- Rôle **public** : rôle de BD spécial auquel chaque utilisateur de BD appartient.
- Donc, 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)

## Etapes à suivre :

- **Créez des rôles** définis par l'utilisateur pour votre application,
- **Appliquez des autorisations** à ces rôles,
- **Ajoutez des membres** à ces rôles

# Gestion de la sécurité dans SQL Server (Suite)

- Il **faut déterminer** les users et les rôles qui peuvent **créer des objets** dans une BD
- Il est **recommandé** d'utiliser uniquement **sysadmin**, **db\_owner** et **db\_ddladmin** pour **créer des objets** de BD
- Il est **recommandé** de **définir les objets avec dbo** en tant que **propriétaire**.
- Cela **permet aux users** de BD de **se référer à l'objet** sans mentionner le nom du propriétaire

# Gestion de la sécurité dans SQL Server (Suite)

- **dbo est propriétaire** de tous les **objets créés** à partir de **sysadmin**
- Dans les **autres rôles**, **spécifiez dbo** en tant que propriétaire lorsqu'on crée l'objet
- **sinon**, l'objet sera créé avec le compte user
- **sp\_changeobjectowner** permet de modifier le propriétaire d'un objet

# Gestion de la sécurité dans SQL Server (Suite)

## Modification du propriétaire d'objets :

- Seuls les membres de **db\_owner** et **db\_ddladmin** et **securityadmin** peuvent modifier le propriétaire d'objets
- La **modification** du propriétaire d'un objet de BD **implique** une **MAJ manuelle** de tous les scripts qui le référencent
  - ➔ 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**
- **Permettent d'exécuter** des scripts, applications compilées et fichiers de traitement par lots
- Lorsqu'un **user non membre de sysadmin** exécute ces travaux, ils sont effectués dans le contexte du **compte Windows associé à SQL Server Agent**



# Gestion de la sécurité dans SQL Server (Suite)

- Ainsi, les **users peuvent exécuter** des fonctions dans Windows auxquelles leurs comptes Windows **ne leur donnent pas accès**
- **Pour éviter cela**, on peut **configurer SQL Server Agent**
  - ➔ **permettre seulement** aux users de **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

- A l'aide de **vues** (Accès à certains champs uniquement)
- A l'aide des **procédures stockées** (Exécution de procédure stockée sans consulter les tables modifiées)
- A l'aide des **rôles 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**

# Rôles d'application

**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 Enterprise Manager ou **sp\_addapprole**
- **Personnes autorisées** : **db\_owner**, **db\_securityadmin** et **sysadmin**
- ajoute une ligne dans la table **sysusers** de la BD active

## ➤ Autorisations sur les rôles d'application

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

# Activation des rôles d'application

## Etapes :

- **Connexion** à SQL Server
- Exécution de **sp\_setapprole** pour activer les autorisations du rôle d'application
- L'utilisateur doit spécifier un **mot de passe**

# Activation des rôles d'application

## Remarques :

- 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 :

```
sp_setapprole 'SalesApp',  
{Encrypt N 'hg_7532LR'}, 'ODBC'
```

# Activation des rôles d'application

- **Suppression** d'un rôle d'application

**sp\_dropapprole**

- **Modification** du mot de passe d'un rôle

d'application : **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

- Il est chargé **d'automatiser** les tâches administratives
- **Doit être exécuté en permanence** et disposer des autorisations adéquates
- Doit être **configuré** pour qu'il **démarre automatiquement** à chaque démarrage de Windows

# Configuration de l'Agent SQL Server (suite)

- Peut être configuré dans SQL Server Entreprise Manager pour qu'il **redémarre automatiquement** en cas d'arrêt inattendu
- **Son compte** doit être membre du groupe **Administrateurs** local pour qu'il redémarre automatiquement,
- on doit lui indiquer un compte **d'ouverture de session** lors de l'installation de SQL Server,

# Configuration de l'Agent SQL Server (suite)

- Il ne pourra démarrer que si le rôle **sysadmin** est attribué à son compte de connexion
- Il peut **se connecter** à l'aide de **l'authentification** Windows ou SQL Server
- **Par défaut**, il utilise l'authentification Windows

# Configuration de SQLAgentMail et SQL Mail

- SQL Server peut établir une **connexion cliente** avec un **serveur de messagerie**
- Cela lui permet d'envoyer et recevoir du **courrier électronique**
- L'échange du courrier électronique se fait à l'aide d'une session **SQLAgentMail** ou **SQL Mail**



# Configuration de SQLAgentMail et SQL Mail (Suite)

- SQLAgentMail nécessite que SQL Server Agent utilise un compte utilisateur avec un **profile de messagerie**
- SQL Mail contient des **procédures 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 **xp\_sendmail**

# Configuration de serveurs liés

## Objectif

- Exécuter des commandes sur des **sources de données OLE DB** sur **différents serveurs**

## Condition

- **Inscrire** les informations sur la **connexion** et la **source de données** OLE DB auprès de SQL Server

# Configuration de serveurs liés (suite)

## Après l'inscription :

- on peut **référencer** cette source de donnée par un seul **nom logique**

## Ajout d'un serveur lié :

- par **sp\_addlinkedserver**
- ou sur SQL Server Enterprise Manager

# Etablissement de la sécurité des serveurs liés

- 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
- **sp\_addlinkedrvlogin** ou par SQL Server Entreprise Manager pour créer des **mappages de connexion** entre les serveurs liés
- Ajout d'un serveur lié (**sp\_addlinkedserver**)
  - ➔ **automappage** 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

L'URL peut effectuer les tâches suivantes :

- **accès** directement aux **objets** de BD comme les tables
- **exécution** des fichiers **modèles** (Document XML comprenant des instructions T-SQL)
- **exécution** 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

- **Nécessaire** pour accéder à une BD SQL Server via HTTP,
- Utiliser **IIS Virtual Directory Management** pour SQL Server pour définir et inscrire un nouveau répertoire virtuel (**racine virtuelle**) sur l'ordinateur exécutant les services Internet
- ➔ création d'une **association** entre le nouvel annuaire et une instance de SQL Server



# Accès du langage XML à SQL Server

## Dans un échange XML :

- 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
- détermine si **l'extension** de nom de fichier **DLL** (Sqlisapi.dll) ISAPI (Internet Server API) a **été inscrite** pour cette racine virtuelle

# Accès du langage XML à SQL Server (suite)

- Le serveur exécutant les services Internet **charge la bibliothèque DLL**
- Il lui **transmet la demande** de l'URL
- L'extension de nom de fichier Sqlisapi.dll **communique avec le fournisseur OLE DB** pour SQL Server
- Elle **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

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

- Identification de la quantité maximale de mémoire à configurer pour une **instance** de SQL Server
- Utilisation du **Moniteur système** Windows pour l'observateur d'effets :
- Utilisez les **statistiques** du Moniteur système Windows pour :
  - aider à évaluer comment les applications serveur s'exécutent
  - régler la valeur de mémoire si nécessaire

## 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**

## 5.3 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 :
  - Service SQL Server,
  - Agent SQL Server
  - Observateur d'événement Windows
- **Composants** de l'Agent SQL Server :
  - Les alertes, les travaux et les opérateurs
- Association de **travaux** et d'**alertes**

# Automatisation de l'administration de SQL Server

## Exemple :

- si un **travail échoue** à cause d'une erreur système,
- une **alerte** (définie pour répondre à ce numéro d'erreur) est **déclenchée**
- puis **exécution d'un autre travail** destiné à résoudre ce problème



# Création de travaux

- Par SQL Server Enterprise Manager ou **sp\_add\_job**
- stockés dans la table **sysjobs**
- Après création il faut :
  - **vérifier** qu'il est **activé** (par défaut oui)
  - indiquer le **propriétaire** (par défaut compte Windows ou SQL Server)
  - déterminer **l'emplacement de l'exécution** du travail (serveur local ou serveurs distants)
  - **créer des catégories** de travail (pour aider à organiser, filtrer et gérer)

# Vérification des autorisations

- Si le **propriétaire d'un travail** est non membre de **sysadmin**, vérifier qu'il dispose des **autorisations** pour l'exécuter
- **Exécution des travaux Transact-SQL :**
  - **Tout user** peut démarrer des travaux T-SQL liés à 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** des users membres de **sysadmin** : **exécutées**
  - User **non membre de sysadmin** : SQL Server Agent vérifie les **autorisations** pour l'exécuter
  - **Par défaut**, user non membre de **sysadmin** n'est pas autorisé à exécuter les commandes du SE ni les script ActiveX

# Vérification des autorisations (suite)

- Un **Admin peut autoriser** les users **non membres** de **sysadmin** à exécuter les commandes du SE ou les travaux de script ActiveX
- Le cas échéant, les étapes de travail sont exécutées à l'aide d'un compte user appelé ***compte proxy***
- On peut définir ce compte proxy dans SQL Server Enterprise Manager ou par la procédure stockée étendue **`xp_sqlagent_proxy_account`**

# Définition des étapes d'un travail

- Par **sp\_add\_jobstep** ou sous SQL Server Entreprise Manager
- Sont stockées dans **sysjobsteps**
- **Types** des étapes (un **seul** type par étape):
  - Instructions T-SQL
  - Commandes du SE
  - Scripts ActiveX
  - Tâches de réplication SQL Server

# 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** (pour 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
- **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

- Exécution d'une étape :
  - **Réussie** ➔ Passage à la suivante (par défaut)
  - **Echec** ➔ Arrêt de l'exécution du travail
- On peut spécifier **l'action** à exécuter en cas **d'échec** de l'exécution d'une étape d'un travail
- Une étape peut **renvoyer** à une **étape quelconque** en cas de réussite ou d'échec
- On peut indiquer le **nombre de tentatives** en cas d'échec
- On peut définir des **intervalles de reprise** en mn

# Planification des travaux

- Par **sp\_add\_jobschedule** ou sous SQL Server Entreprise Manager
- Stockées dans **sysjobschedules**
- Les travaux **s'exécutent** conformément aux **planifications** définies ou en réponse à des **alertes**
- Dans un environnement **multiserveurs**, on peut définir le travail à s'exécuter sur plusieurs serveurs cibles
- La planification doit être **activée**

# Planification des travaux (suite)

## Démarrage **automatique** :

- lorsque l'Agent SQL Server est **démarré**
- A une **heure spécifique** (une seule fois)
- **régulièrement** (tous les jours, semaines ou mois)
- lorsque le **processeur est inactif** (le compte doit être membre du groupe **Administrateurs local Windows**)

# Planifications multiples

- **Plusieurs planifications** peuvent être définies pour un travail
- **Exemple** :
  - **PL1** : sauvegarde du LOG d'une BD du Lundi au Vendredi, toutes les 2 heures pendant les heures de pointes (de 08:00 à 17:00)
  - **PL2** : sauvegarde du LOG toutes les 4 heures hors des heures de pointe

# Création d'opérateurs à notifier

- Sous SQL Server Entreprise Manager ou **sp\_add\_operator**
- Stockés dans **sysoperators**
- Cas de réussite ou échec, on peut :
  - **écrire l'événement** dans le journal d'applications de Windows
  - **supprimer** le travail
  - **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 en cas de **groupe**
- **Tester** chaque méthode de notification
- Etablir une **planification de travail** pour chaque opérateur à notifier par **radiomessagerie**
- Utiliser **net send** pour envoyer des messages aux opérateurs et serveurs du réseau qui exécutent **Windows**

# Résolution des problèmes liés aux notifications d'opérateur

Si un opérateur **ne reçoit pas de notifications** :

- **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 net send
- **consulter** les **dernières tentatives** de notification pour connaître le temps de la dernière notification
- **tester** les méthodes de notification individuelles **en dehors** de SQL Server



# Analyse et configuration de l'historique des travaux

- **sysjobhistory** : contient l'état d'exécution des étapes d'un travail
- On peut afficher l'historique
- En cas d'échec, l'historique **donne des Informations** sur chaque étape du travail :
  - **Raison d'échec**
  - **Solutions** pour résoudre le problème

# Analyse de l'historique des travaux

- **L'historique** des travaux contient :
  - 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

- **Augmenter** la taille maximale de **sysjobhistory** (par défaut 1000 lignes)
- **L'historique** est automatiquement **écrasé** si la taille maximale est atteinte (selon **FIFO**)
- 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**

# 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

# Objectif

- **Répondre à un événement/erreur** dans SQL Server :
  - Erreurs SQL Server
  - Erreurs définies par l'utilisateur
  - Conditions de performance
- Si Problème survient dans SQL Server
  - Déclenchement de traitement **automatique**
  - **Notification** des **opérateurs** pour réagir rapidement
  - Donc, résolution du problème

# Principe

- Au démarrage de **SQL Server Agent** :
  - il **s'inscrit** auprès de l'Observateur d'événement Windows
  - **se connecte** au service SQL Server
- Alerte → **Événement** associé **stocké** dans le journal Windows
- SQL Server **Agent** sera **informé**
- Il cherche les caractéristiques de l'événement (**sysalerts, sysnotifications, sysoperators, sysjobs, sysmessages, ...**)
- Détermination des actions à exécuter

# Caractéristiques des alertes

- **Nom** unique
- **Activée** ou non (par défaut oui)
- **Numéro d'erreur** associé  
(système :  $\leq 50000$ , utilisateur :  $> 50000$ )
- Niveau de **gravité** (entre 0 et 25)
- **Événement** qui déclenche l'alerte
- **Actions à exécuter** par SQL Server Agent :
  - Réponse à l'alerte (Message, Notification, Travail)
  - Opérateur à notifier

# Création d'Alertes

- Sous SQL Server Entreprise Manager
- Par la procédure système **sp\_add\_alert**
- Stockées dans **sysalerts**
- Table conservée en **mémoire cache**
- Donc amélioration des performances



# Consignation d'événements dans le journal

- Alertes de niveau de gravité **entre 19 et 25**
- Messages d'erreurs **définis** (sp\_addmessage ou sp\_altermessage) **pour être consignés**
- Utilisation de RAISERROR **WITH LOG**
- Utilisation de **xp\_logevent**

# Alertes SQL Server

## Définition d'alerte pour des numéros d'erreur SQL Server

- Le numéro d'erreur **doit être consigné** dans le journal de Windows sinon pas de déclenchement de l'alerte
- Peut être définie pour un numéro d'erreur prédéfini ( $\leq 50000$ ) ou défini par l'utilisateur
- **Plusieurs alertes** peuvent être définies pour un numéro d'erreur
- Une alerte peut être associée à **une ou plusieurs BD** (Le message d'erreur doit être clair)

**Exemple** : 'Tentative de connexion par l'utilisateur %s'

# Alertes SQL Server (suite)

## Définition d'alerte pour des niveaux de gravité d'erreur

- Les niveaux de **gravité** entre 19 et 25 sont automatiquement **consignés** dans le journal Windows
- Les niveaux de **gravité** entre 20 et 25 correspondent à des **erreurs fatales**
  - ➔ Donc définir un **opérateur à notifier**
- Une alerte peut être associée à une ou plusieurs BD
- Existence d'Alertes prédéfinies qu'on peut utiliser

**Exemple** : **Espace disque (niveau 17)**

# Alertes définies par l'utilisateur

- Créer le message d'erreur
  - **sp\_addmessage** avec un **numéro** > 50000 et un **niveau de gravité**
  - stockage dans **sysmessages** de master
  - possibilité d'utiliser des **paramètres**
  - possibilité de choix **multilingue**
  - consignation dans le **journal de Windows**

## Exemple :

**sp\_addmessage** 50099,16

'Client %d supprimé par %s', 'French', 'true'

# Alertes définies par l'utilisateur (suite)

## ➤ Déclenchement de l'erreur

- À partir d'une procédure ou un trigger
- Utilisation de **RAISERROR**

## ➤ Exemple :

***RAISERROR (50099, 16, 1, @CustomerID,  
@username)***

# Alertes Performances

## ➤ Sur des **ressources SQL Server** :

- Méthodes d'accès
- Gestionnaires de tampons
- Gestionnaire de cache
- Bases de données
- Verrous
- Statistiques SQL

### Exemple :

- Alerte qui se déclenche lorsque le **LOG** d'une BD **dépasse 75%** de sa capacité

Réponse à cette alerte : **Sauvegarder** le LOG et **notifier** l'administrateur de la BD

# Transfert d'événement

## ➤ Objectif :

- Désigner **un serveur qui recevra tous les messages** de niveau de gravité  $\geq$  Valeur
- **Centraliser** le traitement
- Donc **définir la réaction** aux différentes alertes sur un **même serveur** SQL Server

# Transfert d'événement (suite)

## ➤ Avantages

- **Centralisation** donc gestion simplifiée des alertes
- **Réduire l'administration** sur les autres serveurs
- **Temps** de mise en œuvre **réduit** car les alertes ne sont définies qu'une seule fois

## ➤ Inconvénients

- Augmentation du **trafic réseau**
- Point de défaillance **unique**
- **Charge de travail** pour le serveur qui centralise



# 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 Travaux, alertes ou notifications automatisés **ne fonctionnent pas correctement** alors :

- vérifier que l'Agent SQL Server est **démarré**,
- vérifier que le rôle **sysadmin** est attribué à son compte de connexion
- vérifier que le **mot de passe** est valide

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

- 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
- vérifier 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** correspondent au nombre de lignes conservées dans **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 des problèmes liés aux alertes

➤ **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

- Rôle d'un **serveur principal** : définir, planifier et gérer les travaux sur tous les **serveurs cibles**
- Définition d'un **serveur principal** : par SQL Server Enterprise Manager ou **sp\_msx\_enlist**
- On doit lui inscrire au **moins un serveur cible**
- Une ligne sera stockée dans **systargetservers**



## 5.6 Automatisation de travaux sur plusieurs serveurs (suite)

- Un **compte et un mot de passe** d'ouverture de session SQL Server sont **automatiquement créés** pour **chaque serveur cible**
- 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

## Fonctionnement

- **Création** de travaux sur le **serveur principal**
- Le serveur principal les poste dans la table système **sysdownloadlist**
- Les serveurs **cibles consultent** cette table (travaux nouveaux ou mis à jour)
- Les serveurs **cibles exécutent** ces travaux et envoient l'état du résultat du travail à la fin
- Le serveur principal enregistre les informations relatives au résultat du travail dans **sysjobservers**

# Modification des définitions 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

# 6. Sauvegarde de Base de Données

## 6.1 Protection contre les pertes de données

6.2 Définition et changement de mode de récupération de BD

6.3 Sauvegarde de SQL Server

6.4 Moment approprié pour sauvegarder des BD

6.5 Exécution de sauvegardes

6.6 Types de méthodes de sauvegarde

6.7 Planification d'une stratégie de sauvegarde

# 6.1 Protection contre les pertes de données

- **Pertes** dues à une mauvaise utilisation de DELETE, UPDATE, propagation de virus destructeurs, catastrophes naturelles (incendie,...), vol
- Définition d'une **stratégie** de sauvegarde
  - Pour **minimiser** les pertes de données
  - Pour **recupérer** les données perdues
  - Pour **restaurer** les données avec un **impact minimal** sur la production
- Sauvegardes **régulières**

# Sauvegardes régulières

- Faire des sauvegardes **fréquentes** si le système se trouve dans un environnement **OLTP**
- Faire des sauvegardes **moins fréquentes** si le système a peu d'activités (**OLAP**)
- **Planifier** les sauvegardes à un moment où **aucune mise à jour** majeure n'est effectuée dans SQL Server
- Après avoir déterminé la stratégie de sauvegarde, on peut **automatiser** le processus

# 6. Sauvegarde de Base de Données

6.1 Protection contre les pertes de données

**6.2 Définition et changement de mode de récupération de BD**

6.3 Sauvegarde de SQL Server

6.4 Moment approprié pour sauvegarder des BD

6.5 Exécution de sauvegardes

6.6 Types de méthodes de sauvegarde

6.7 Planification d'une stratégie de sauvegarde

## 6.2 Définition et changement de mode de récupération de BD

- Définition d'un **mode de récupération** de BD
  - Mode récupération **complète**
  - Mode de récupération **Bulk\_Logged**
  - Mode de récupération **simple**
- Changement de mode de récupération de BD

*ALTER DATABASE NomBD*

*SET RECOVERY {FULL/SIMPLE/BULK\_LOGGED}*



# Mode de récupération **complète**

- Utilisé lorsque la **récupération totale** du support endommagé est la plus **prioritaire**
- Utilise des **copies de la BD** et toutes les informations des **journaux** pour restaurer la BD
- SQL Server consigne **toutes les modifications** apportées à la BD

# Mode de récupération **complète** (Suite)

- Si les journaux eux-mêmes ne sont pas endommagés, SQL Server peut récupérer toutes les données **sauf les transactions en cours** au moment de la défaillance
- Possibilité **d'insérer des marques** nommées dans le journal des transactions
- Donc **récupération jusqu'à cette marque**

# Mode de récupération **Bulk\_Logged**

- Utilise aussi les sauvegardes des BD et des journaux pour recréer la BD
- **Moins de place** dans le journal pour : CREATE INDEX, SELECT INTO, WRITETEXT, UPDATETEXT, ...
- Le journal n'enregistre que l'occurrence de ces opérations
- permet de restaurer toutes les données mais **pas partiellement** (à une marque donnée par exemple)

# Mode de récupération simple

- Utilise des copies complètes ou différentielles de la BD
- La récupération **se limite** à la restauration de la BD jusqu'au moment de la dernière sauvegarde
- Toutes les modifications effectuées après la sauvegarde **sont perdues** et doivent être recréées

# Mode de récupération simple (suite)

## Avantage :

- Utilisé pour les BD **peu volumineuses**
- Utilisé dans les BD dans lesquelles les données sont **rarement modifiées**
- **Faible espace** de stockage occupé par les journaux
- **Facile** à mettre en œuvre

# 6. Sauvegarde de Base de Données

6.1 Protection contre les pertes de données

6.2 Définition et changement de mode de  
récupération de BD

## 6.3 Sauvegarde de SQL Server

6.4 Moment approprié pour sauvegarder des BD

6.5 Exécution de sauvegardes

6.6 Types de méthodes de sauvegarde

6.7 Planification d'une stratégie de sauvegarde

## 6.3 Sauvegarde de SQL Server

- Permet de **sauvegarder une BD en cours d'utilisation**
- Sauvegarde les fichiers de BD originaux et enregistre leur emplacement
- La **sauvegarde contient** le schéma et la structure de fichiers, les données et des parties des fichiers journaux de transactions contenant les activités de BD survenues depuis le début du processus de sauvegarde
- SQL Server **utilise ces sauvegardes** pour **recréer** les fichiers à **leur emplacement d'origine** avec les objets et les données lorsqu'on restaure la BD

# Exécution et stockage des sauvegardes

## ➤ Personnes autorisées :

- Membres d'un rôle défini par l'utilisateur
- Membres de **sysadmin**, **db\_owner** et **db\_backupoperator**

## ➤ Emplacements de stockage :

- Fichier sur le **disque** dur local ou en réseau
- **Bande** (le lecteur de bande doit être connecté localement à SQL Server)
- **Canal** nommé



# 6. Sauvegarde de Base de Données

6.1 Protection contre les pertes de données

6.2 Définition et changement de mode de  
récupération de BD

6.3 Sauvegarde de SQL Server

**6.4 Moment approprié pour sauvegarder des BD**

6.5 Exécution de sauvegardes

6.6 Types de méthodes de sauvegarde

6.7 Planification d'une stratégie de sauvegarde

## 6.4 Moment approprié pour sauvegarder des BD

- Sauvegarde de BD système
- Sauvegarde de BD utilisateur
- Activités à éviter pendant la sauvegarde

# Sauvegarde de BD système

- Après modification de la BD **master**
  - Create** Database, **Alter** Database, **Drop** Database
  - Utilisation de certaines **procédures** :  
**sp\_logdevice**, **sp\_addserver**, **sp\_dropserver**,  
**sp\_addlinkedserver** et **sp\_addmessage**
- Après modification de la BD **msdb**
- Après modification de la BD **model**

# Sauvegarde de BD utilisateur

- **Après création** de BD ou son chargement en données
- Après création d'**index**
- Après **vidage du journal** de transactions :
  - Backup Log With Truncate\_only
  - Backup Log With No\_log
- Après exécution d'opérations **non journalisées**
  - Writetext, Updatetext
  - Select ... into ...

# Activités à éviter pendant les sauvegardes

- **Création** ou modification de BD
- Exécution d'**opérateurs** de croissance automatique
- Création d'**index**
- Exécution d'opérations **non journalisées**
- **Compactage** d'une BD

# 6. Sauvegarde de Base de Données

6.1 Protection contre les pertes de données

6.2 Définition et changement de mode de  
récupération de BD

6.3 Sauvegarde de SQL Server

6.4 Moment approprié pour sauvegarder des BD

**6.5 Exécution de sauvegardes**

6.6 Types de méthodes de sauvegarde

6.7 Planification d'une stratégie de sauvegarde

## 6.5 Exécution de sauvegardes

- Création **d'unités** de sauvegarde
- Création de **fichiers de sauvegarde** sans unité permanente
- Utilisation de **plusieurs fichiers** de sauvegarde pour stocker les sauvegardes
- Utilisation de l'instruction **BACKUP**
- Sauvegarde sur un **lecteur de bande**

# Création d'**Unités** de sauvegarde

- **Raisons** justifiant la création d'unités de sauvegarde permanentes :
  - **Réutilisation** de fichiers de sauvegarde
  - **Automatisation** des sauvegardes
- Utilisation de **sp\_addumpdevice**
  - Spécifier le **nom logique**
  - Les noms logiques et physiques sont stockées dans **sysdevices** de la BD master



# Exemples

## Exemple 1 :

*USE master*

*EXEC sp\_addumpdevice 'disk',  
'mybackupfile',  
'C:\Buckup\MyBackupFile.bak'*

➔ création d'un fichier de sauvegarde sur le disque

## Exemple 2 :

*USE master*

*EXEC sp\_addumpdevice 'tape',  
'mytape1', '\\.\tape0'*

➔ création d'une unité de sauvegarde sur une bande

# Création de fichiers de sauvegarde sans unité de sauvegarde

**Raisons** justifiant la création d'unités de sauvegarde sans unité permanente :

- On **ne veut pas** réutiliser le fichier de sauvegarde
- Exécution d'**une seule** sauvegarde
- **Test** de l'opération de sauvegarde en vue de l'automatiser

# Utilisation de **BACKUP** DATABASE

- Spécifier un **type de support** (disque, Bande, Canal nommé)
- Spécifier un **chemin d'accès**
- Spécifier un **nom de fichier**

## Exemple

*USE master*

*BACKUP DATABASE Northwind*

*TO DISK='C:\Temp\MyCustomers.bak'*

➔ création d'un fichier et sauvegarde

# Utilisation de plusieurs fichiers de sauvegarde pour stocker les sauvegardes

- **Intérêt : réduire** la durée totale de la sauvegarde
- **Contrainte** : Toutes les unités utilisées doivent être de **même type de support** (disque, bande)
- On peut **combiner** des fichiers permanents et temporaires
- **Problème** : Si on **formate** un fichier, les données des autres fichiers ne sont **plus valides**

# Utilisation des options INIT, NOINIT et FORMAT

- **NOINIT** : ajoute les sauvegardes à un fichier (pris par défaut)
- **INIT** : écrase un fichier de sauvegarde
- **FORMAT** :
  - Ecrase le contenu d'un fichier de sauvegarde
  - Subdivise un jeu de sauvegardes réparti

# Sauvegarde sur un lecteur de bande

**Bandes** : support de stockage **pratique** car :

- **Peu** coûteuses
- Offrent un **volume** de stockage important
- Peuvent être conservées en **dehors** du site
- Donc meilleure **sécurité** des données

# Sauvegarde sur un lecteur de bande

- **Rattachement local** du lecteur de bandes à SQL Server
- **Enregistrement** des informations de sauvegarde sur l'étiquette de la bande (Nom de la BD, Heure et Date, Type de sauvegarde)
- **Stockage** de sauvegardes SQL Server et non-SQL Server (Format de sauvegarde standard : ***MS Tape Format***)

# Spécification des options de bande

Option	Description
<b>UNLOAD</b> (Par défaut)	Rembobine et extrait automatiquement la bande une fois la sauvegarde terminée
<b>NOUNLOAD</b>	Ne rembobine pas et n'extrait pas automatiquement la bande
<b>BLOCKSIZE</b>	Change la taille des blocs physiques exprimée en octets
<b>FORMAT</b>	Ecrit un en-tête sur les fichiers utilisés pour une sauvegarde
<b>SKIP</b>	Ignore les étiquettes de bande ANSI
<b>NOSKIP</b> (Par défaut)	Lit les étiquettes de bande ANSI
<b>RESTART</b>	Redémarre l'opération de sauvegarde à partir du point d'interruption



# 6. Sauvegarde de Base de Données

6.1 Protection contre les pertes de données

6.2 Définition et changement de mode de  
récupération de BD

6.3 Sauvegarde de SQL Server

6.4 Moment approprié pour sauvegarder des BD

6.5 Exécution de sauvegardes

**6.6 Types de méthodes de sauvegarde**

6.7 Planification d'une stratégie de sauvegarde

## 6.6 Types de méthodes de sauvegarde

- Sauvegarde de BD **complète**
- Sauvegarde **différentielle**
- Sauvegarde du **journal** des transactions
- Sauvegarde d'un **fichier** ou d'un groupe de fichiers de BD

# Sauvegarde de BD complète

- Sert de point de **référence** en cas de défaillance du système
- **Sauvegarde** les fichiers, objets et données d'origine
- Sauvegarde des parties du **journal** des transactions
- SQL Server sauvegarde toutes les **activités** survenues **pendant la sauvegarde**
- Il sauvegarde toutes les **transactions non validées** dans le journal des transactions

# Exemple 1

USE master

```
EXEC sp_addumpdevice 'disk', 'NwindBac',  
'C:\NwindBac.bak'
```

**BACKUP DATABASE Northwind TO NwindBac**

➔ Une unité de sauvegarde est créée et une sauvegarde complète est effectuée

## Exemple 2

BACKUP DATABASE Northwind

TO NwindBac

WITH INIT

➔ L'intégralité de la BD est sauvegardée dans le fichier *NwindBac* et toutes les sauvegardes précédentes dans le fichiers sont écrasées

# Exemple 3

```
BACKUP DATABASE Northwind  
TO NwindBac  
WITH NOINIT
```

➔ Une sauvegarde de la BD complète est ajoutée au fichier *NwindBac* et tout fichier de sauvegarde précédent est conservé

## Exemple 4

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

➔ Crée un fichier de sauvegarde sur le disque et effectue une sauvegarde complète de cette BD dans ce fichier

# Sauvegarde de BD différentielle

- Convient aux BD **souvent modifiées**
- **Requiert** une sauvegarde de BD complète
- Sauvegarde les modifications effectuées **depuis la dernière** sauvegarde de BD complète
- **Réduit** la durée des processus de sauvegarde et de restauration



# Sauvegarde de BD différentielle (suite)

## Lors d'une sauvegarde différentielle, SQL Server :

- Sauvegarde les parties de la BD **modifiées** depuis la dernière sauvegarde complète de la BD
- Sauvegarde toute **activité** survenue **pendant** la sauvegarde différentielle,
- Sauvegarde toute **transaction non validée** dans le journal des transactions

# Sauvegarde de BD différentielle (suite)

**Lors d'une sauvegarde différentielle :**

- Si une ligne de la BD a été **modifiée plusieurs fois** depuis la dernière sauvegarde de BD complète, la sauvegarde différentielle contient uniquement le **dernier** ensemble de valeurs de cette ligne
- Alors que **dans une sauvegarde du LOG**, le journal contient un **historique** de toutes les **modifications** apportée à cette ligne

# Sauvegarde de BD différentielle (suite)

- La **durée de sauvegarde** d'une BD est **réduite** car les jeux de sauvegardes sont plus petits que pour une sauvegarde complète
- La **durée de restauration** d'une BD est **réduite** car il n'est pas nécessaire d'appliquer une série de fichiers LOG
- On doit définir une **convention de nommage** pour les fichiers de sauvegarde
  - ➔ **distinguer** les fichiers contenant les différents types de sauvegardes (complètes, différentielles,...)

# Syntaxe partielle et exemple

## Syntaxe partielle :

```
BACKUP DATABASE {BD | @Var_Nom_BD}  
TO <unité_sauvegarde> [,...n]  
[WITH [DIFFERENTIAL]]
```

**Exemple :** `BACKUP DATABASE Northwind  
TO DISK='D:MyDiffBackup.bak'  
WITH DIFFERENTIAL`

➔ Une sauvegarde différentielle est créée dans un fichier de sauvegarde temporaire

# Sauvegarde du journal des transactions

- **Requiert** une sauvegarde de BD **complète**
- Sauvegarde le journal des transactions **entre** la dernière instruction **BUCKUP LOG** exécutée correctement et la fin du journal de transactions en cours
- **Tronque** le journal des transactions

# Sauvegarde du journal des transactions (suite)

## Lorsqu'on sauvegarde le LOG, SQL Server :

- Sauvegarde le LOG à partir de la dernière instruction **BACKUP LOG** exécutée avec succès et jusqu'à la **fin** du journal des transactions en cours
- Tronque** le journal des transactions jusqu'au début de la **partie active** et **élimine** la **partie inactive**

# Syntaxe partielle et exemple

## Syntaxe partielle :

```
BACKUP LOG {BD | @Var_Nom_BD}  
TO <unité_sauvegarde> [,...n]  
[WITH [,] {INIT | NOINIT}]
```

## Exemple : USE master

```
EXEC sp_addumpdevice 'disk', 'NwinBacLog',  
    'D:\NwinBacLog.bak'
```

```
BACKUP LOG Northwind TO NwinBacLog
```

➔ Crée une unité de sauvegarde sur le disque et sauvegarde le LOG de la BD Northwind

# Utilisation de l'option **NO\_TRUNCATE**

SQL Server :

- **Enregistre** le journal des transactions dans **son intégralité** même si la BD est inaccessible
- **Ne supprime pas** les transactions validées du journal des transactions
- **Permet de récupérer les données** jusqu'au moment où le système a subi la défaillance



# Vidage du journal des transactions

- Si plein**, on ne peut pas mettre à jour la BD ni restaurer entièrement la BD en cas de défaillance
- On **doit donc vider** le journal **soit** en effectuant une **sauvegarde complète** de la BD et en enregistrant les données, **soit** en **tronquant** le LOG
- Par l'instruction **BUCKUP LOG** avec l'option **TRUNCATE\_ONLY** ou **NO\_LOG**

# Syntaxe partielle et exemples

## Syntaxe :

```
BUCKUP LOG {BD/@Var_Nom_BD}  
[WITH {TRUNCATE_ONLY | NO_LOG}]
```

## Exemple 1 :

```
BACKUP LOG Northwind WITH TRUNCATE_ONLY
```

➔ Suppression de la partie inactive d'un LOG sans faire de copie de sauvegarde

## Exemple 2 :

```
BACKUP LOG Northwind WITH NO_LOG
```

➔ Suppression de la partie inactive d'un LOG plein sans faire de copie de sauvegarde

# Sauvegarde d'un fichier ou d'un groupe de fichiers de BD

- **Utilisée** lorsqu'on ne peut pas réaliser une sauvegarde complète (**BD très volumineuses**)
- SQL Server effectue les tâches suivantes :
  - Sauvegarde uniquement** les fichiers de BD spécifiés (option FILE ou FILEGROUP)
  - Permet de sauvegarder des fichiers de BD spécifiques et non toute la BD

# Sauvegarde d'un fichier ou d'un groupe de fichiers de BD (suite)

- Lorsqu'on effectue une telle sauvegarde :
  - On **doit spécifier** les fichiers ou les groupes de fichiers logiques
  - On **doit effectuer** des sauvegardes du LOG pour assurer la cohérence des fichiers restaurés par rapport à la BD restaurée
  - Il est recommandé d'établir une **stratégie de sauvegarde par rotation**
  - On peut spécifier jusqu'à **16** fichiers ou groupes de fichiers

# Syntaxe partielle

```
BACKUP DATABASE {BD | @Var_Nom_BD}  
[<fichier_ou_groupefichiers>[,...n]  
TO <unité_sauvegarde> [,...n]]
```

Où <fichier\_ou\_groupefichiers> représente :

```
{FILE={nom_fichier_logique | @var_nom_fichier_logique}  
| FILEGROUP={nom_groupe_fichier_logique}  
}
```

# Exemple

BACKUP DATABASE PhoneOrders

FILE=Orders2 TO OrderBackup2

BUCKUP LOG PhoneOrders to OrderLog

➔ Le fichier Orders2 d'un groupe de fichiers de BD est sauvegardé et le LOG est stocké dans le fichier OrderLog

# 6. Sauvegarde de Base de Données

6.1 Protection contre les pertes de données

6.2 Définition et changement de mode de  
récupération de BD

6.3 Sauvegarde de SQL Server

6.4 Moment approprié pour sauvegarder des BD

6.5 Exécution de sauvegardes

6.6 Types de méthodes de sauvegarde

**6.7 Planification d'une stratégie de sauvegarde**

## 6.7 Planification d'une stratégie de sauvegarde

- **Nécessité** d'une stratégie de sauvegarde
- **Choisir** la ou les méthodes adaptées à l'environnement de l'entreprise
- **Tenir compte** du processus de **restauration**
- **Tenir compte** des **exigences** liées à chaque stratégie



# Types de stratégie de sauvegarde

- **Stratégie de sauvegarde complète de BD**
- **Stratégie de sauvegarde complète de BD et de sauvegarde du LOG**
- **Stratégie de sauvegarde différentielle**
- **Stratégie de sauvegarde de fichiers ou de groupes de fichiers de BD**

# Stratégie de sauvegarde complète

- La **taille** de la BD et la **fréquence des modifications** déterminent le temps et les ressources nécessaires
- **Utile** dans le cas des **BD petites**
  - ➔ le temps de sauvegarde est raisonnable

# Stratégie de sauvegarde complète de BD et de sauvegarde du LOG

- Sauvegarde **complète**
- En plus, sauvegarde du **LOG** pour enregistrer toutes les activités de BD survenues entre deux sauvegardes complètes
- Stratégie de sauvegarde **couramment utilisée**

# Stratégie de sauvegarde différentielle

- Sauvegarde **complète**
- Des sauvegardes du **LOG**
- Sauvegarde **différentielle**
- Il suffit de restaurer la dernière sauvegarde différentielle pour récupérer une BD
- Cette dernière sauvegarde contient toutes les modifications apportées à la BD depuis la dernière sauvegarde complète

# Stratégie de sauvegarde de fichiers ou de groupes de fichiers de BD

- A utiliser pour :
  - des **BD très volumineuses**
  - **partitionnées** sur plusieurs fichiers
- A **sauvegarder aussi** le journal des transactions

# Conseils pratiques



**Définissez une stratégie de sauvegarde**



**Sauvegardez les bases de données système après les avoir modifiées**



**Planifiez des opérations de sauvegarde à un moment où l'activité de la base de données est faible**



**Créez des unités de sauvegarde**



**Testez votre stratégie de sauvegarde**

# 7. Restauration de Base de Données

## 7.1 Processus de récupération de SQL Server

7.2 Préparation de la restauration d'une BD

7.3 Restauration de sauvegardes

7.4 Restauration de BD à partir de différents  
types de sauvegardes

7.5 Restauration de BD système endommagées

# 7.1 Processus de récupération de SQL Server

- SQL Server **examine le LOG** depuis le **dernier CHECKPOINT** jusqu'au moment de la **panne**
- Si le LOG contient des **transactions validées non encore écrites** dans la BD, **les modifications sont appliquées** dans la BD
- Si le LOG contient des **transactions non validées**, elles sont **annulées**



# Activités exécutées par SQL Server pendant le processus de restauration

- SQL Server effectue un **contrôle** de sécurité et ne restaure pas si :
  - **BD existe** déjà
  - **Fichiers** de BD sont **différents**
  - **Fichiers** de BD sont **incomplets**
- Lors d'une restauration de BD à partir d'une sauvegarde complète, SQL Server **reconstitue le schéma** complet de la BD

# 7. Restauration de Base de Données

7.1 Processus de récupération de SQL Server

**7.2 Préparation de la restauration d'une BD**

7.3 Restauration de sauvegardes

7.4 Restauration de BD à partir de différents  
types de sauvegardes

7.5 Restauration de BD système endommagées

## 7.2 Préparation de la restauration d'une BD

- On **doit vérifier** les sauvegardes pour **s'assurer** que :
  - on **restaure** les données et les objets **voulus**
  - la sauvegarde contient les **informations valides**
- On doit exécuter des tâches nécessaires au lancement du processus de restauration

# Vérification des sauvegardes

Instruction **RESTORE HEADERONLY** :

- Renvoie les **informations d'en-tête d'un fichier de sauvegarde** ou d'un jeu de sauvegardes :

*(nom et description de fichier de sauvegarde, type de support, méthode de sauvegarde, date et heure de la sauvegarde, taille de la sauvegarde)*

# Vérification des sauvegardes (suite)

Instruction **RESTORE FILELISTONLY** :

- Renvoie des **informations sur les fichiers de BD ou de LOG d'origine** : *(noms logiques et physiques des fichiers de BD et de LOG, type de fichier (de BD ou de LOG), appartenance à un groupe de fichiers, taille du jeu de sauvegarde en MO, taille de fichier maximale autorisée en MO)*

# Vérification des sauvegardes (suite)

Instruction **RESTORE LABELONLY** :

- Renvoie des **informations sur le support de sauvegarde** contenant le fichier de sauvegarde

Instruction **RESTORE VERIFYONLY** :

- Vérifie que les **fichiers** constituant le jeu de sauvegardes sont **complets**,
- Vérifie que toutes les **sauvegardes** sont **lisibles**

# Exécution de tâches spécifiques avant la restauration de sauvegardes

## ➤ **Restriction de l'accès** à la BD

- Limitez l'accès aux membres du rôle **db\_owner**, **dbcreator** ou **sysadmin**

## ➤ **Sauvegarde du LOG** pour :

- Assurer la cohérence de la BD
- Capturer les modifications effectuées entre la dernière sauvegarde du LOG et le moment auquel la BD a été déconnectée

# 7. Restauration de Base de Données

7.1 Processus de récupération de SQL Server

7.2 Préparation de la restauration d'une BD

**7.3 Restauration de sauvegardes**

7.4 Restauration de BD à partir de différents  
types de sauvegardes

7.5 Restauration de BD système endommagées



## 7.3 Restauration de sauvegardes

- Par SQL Server Enterprise Manager ou l'instruction **RESTORE**
- On peut spécifier des **options** propres au type de sauvegarde qu'on veut restaurer
- On peut aussi **décider** si on désire **lancer** le processus de récupération **après** chaque opération de **restauration**

# Utilisation de l'instruction RESTORE

- On **n'a pas à supprimer la BD** endommagée
- SQL Server **crée automatiquement** les fichiers et les objets de BD
- Les **options** de restauration permettent de **détailler la façon de restaurer** les sauvegardes

# Syntaxe partielle

```
RESTORE DATABASE {BD | @var_nom_bd}  
[FROM <unité_sauvegarde> [,...n]]  
[WITH  
    [FILE=numéroofichier]  
    [[,]MOVE 'nom_fichier_logique' TO 'nom_fichier_SE']  
    [[,]REPLACE]  
    [[,] {NORECOVERY | RECOVERY | STANDBY=  
        nom_fichier_annulation}]]  
    [[,]RESTART]]
```

Où <unité\_sauvegarde> représente :

```
{ {unité_sauvegarde | @var_nom_unité_sauvegarde} |  
{DISK | TAPE | PIPE}='unité_sauvegarde_temporaire' |  
@var_nom_unité_sauvegarde_temporaire}
```

# Exemple

USE master

RESTORE DATABASE Northwind

FROM NwinBac

➔ Restauration de la BD Northwind à partir  
d'un fichier de sauvegarde permanent

# Lancement du processus de récupération

Utilisation de l'option **RECOVERY** (par défaut) :

- **A utiliser lors** de la restauration du **dernier LOG** ou dans le cadre d'une restauration complète de BD
- **A ne pas utiliser si** d'autres journaux de transactions ou d'autres sauvegardes différentielles doivent être restaurées

# Lancement du processus de récupération (suite)

Utilisation de l'option **RECOVERY** (suite) :

- SQL Server **annule** les **transactions non validées** dans le LOG
- **transmet** les **transactions validées**
- Permet **d'accéder** à la BD

# Lancement du processus de récupération (suite)

Utilisation de l'option **NORECOVERY** :

- **A utiliser** pour restaurer **toutes** les sauvegardes **sauf la dernière**
- SQL Server **n'annule pas** les **transactions non validées** dans le LOG
- **ne transmet pas** les transactions **validées**
- La **BD ne peut pas être utilisée** tant qu'elle n'est pas récupérée

# Définition des options de restauration

Option RESTORE	Description
<b>FILE</b>	<ul style="list-style-type: none"><li>- Restaure une sauvegarde particulière</li><li>- On doit indiquer un numéro de fichier</li></ul>
<b>RESTART</b>	<ul style="list-style-type: none"><li>- Reprend une opération de récupération interrompue</li></ul>
<b>MOVE ... TO</b>	<ul style="list-style-type: none"><li>- Indique un emplacement de restauration des fichiers de sauvegarde</li><li>- A utiliser pour effectuer une restauration sur un <b>autre disque</b>, serveur ou ordinateur SQL Server de secours</li></ul>
<b>REPLACE</b>	<ul style="list-style-type: none"><li>- Remplace une BD existante</li><li>- SQL Server n'effectue pas de contrôle de sécurité</li></ul>



# 7. Restauration de Base de Données

7.1 Processus de récupération de SQL Server

7.2 Préparation de la restauration d'une BD

7.3 Restauration de sauvegardes

**7.4 Restauration de BD à partir de différents  
types de sauvegardes**

7.5 Restauration de BD système endommagées

## 7.4 Restauration de BD à partir de différents types de sauvegardes

- Restauration à partir d'une **sauvegarde complète** de BD
- Restauration à partir d'une **sauvegarde différentielle**
- Restauration à partir d'une **sauvegarde de LOG**
- Restauration à partir d'une **sauvegarde de fichier ou de groupe de fichiers**

# Restauration à partir d'une sauvegarde complète de BD

On l'utilise dans les **cas suivants** :

- Le **disque** physique est **endommagé**
- **Toute la BD** est **endommagée**, détériorée ou supprimée
- Pour **conserver une copie** identique de la BD sur un **autre ordinateur SQL Server**

# Restauration à partir d'une sauvegarde complète de BD (suite)

- Démarrer** la récupération avec **RECOVERY**  
si on n'a aucun **LOG** ni sauvegarde différentielle
- Retarder** la récupération (**NORECOVERY**)  
si on a des **LOG** ou sauvegardes différentielles

# Exemple

USE master

RESTORE DATABASE Northwind

FROM NwinBac

WITH FILE=2, RECOVERY

➔ Sauvegarde complète stockée dans le fichier de sauvegarde **NwinBac** et 2 sauvegardes ont été ajoutées à ce fichier

➔ La BD **Northwind** est entièrement remplacée par la 2<sup>ème</sup> sauvegarde figurant dans l'unité de sauvegarde **NwinBac**

➔ Enfin, le processus de récupération rétablit la cohérence de la BD (transmet les modifications validées et annule celles non validées)

# Restauration à partir d'une sauvegarde différentielle

- Restaure uniquement les **parties** de la BD qui ont été **modifiées depuis la dernière sauvegarde complète** de BD
- **Rétablit la BD à son état initial** au moment où la sauvegarde différentielle a été effectuée
- Dure **moins longtemps** que l'application d'une série de **LOG**

# Éléments à prendre en compte pour la restauration de sauvegardes différentielles

- Restaurer la sauvegarde complète de BD avant la sauvegarde différentielle
- Spécifier le fichier de sauvegarde contenant la sauvegarde différentielle
- Spécifier **NORECOVERY** si d'autres LOG doivent être restaurés;
- sinon, utiliser l'option **RECOVERY**

# Exemple

USE master

**RESTORE DATABASE** Northwind

FROM NwinBacDiff

**WITH NORECOVERY**

➔ Restaure une sauvegarde différentielle **sans récupérer** la BD

➔ Le fichier **NwinBacDiff** contient une sauvegarde différentielle

➔ SQL Server permet de **restaurer des LOG** avant de rétablir la cohérence de la BD **Northwind** en spécifiant **NORECOVERY**



# Restauration à partir d'une sauvegarde de LOG

- **Restaure les modifications** apportées à la BD **enregistrées dans le LOG**
- Généralement utilisée pour permettre **l'application des modifications** apportées à la BD depuis la **dernière sauvegarde complète ou différentielle**
- Permet aussi de **recupérer** une BD **jusqu'à un certain point dans le temps**

# Éléments à prendre en compte pour la restauration des journaux de transactions

- **Restaurer d'abord** la sauvegarde **complète**
- **Restaurer** les sauvegardes des **LOG créées** **après** une sauvegarde **différentielle** pour préserver la cohérence des données
- Spécifier **NORECOVERY** pour tous les LOG sauf le dernier
- **BD non récupérée** jusqu'à ce que le **dernier LOG** soit restauré

# Exemple

- Une **sauvegarde complète** de BD a été effectuée et enregistrée dans un fichier de sauvegarde
- **Deux sauvegardes de LOG** ont été créées dans un autre fichier de sauvegardes
- ➔ **3 opérations de restaurations** distinctes doivent être effectuées pour garantir la cohérence de la BD:

# Exemple (suite)

**1-**     *RESTORE DATABASE Northwind*  
          *FROM NwindBac*  
          *WITH NORECOVERY*

➔ **restauration à partir d'une sauvegarde complète**  
sans récupérer la BD

**2-**     *RESTORE LOG Northwind*  
          *FROM NwindBacLog*  
          *WITH FILE=1, STATS, NORECOVERY*

➔ **restauration du 1<sup>er</sup> LOG** sans récupérer la BD en  
affichant la progression du processus de restauration

# Exemple (suite)

**3-**     *RESTORE LOG Northwind*  
*FROM NwindBacLog*  
*WITH FILE=2, RECOVERY*

➔ restaure le 2<sup>ème</sup> **LOG**, transmet les transactions validées et annule les transactions non validées

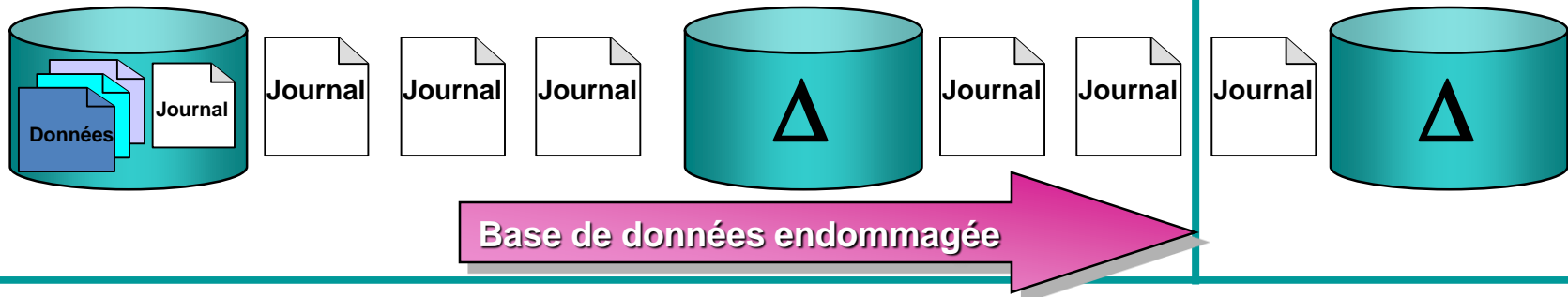
# Restauration à partir d'une sauvegarde de LOG (suite)

## Sauvegardes de la base de données Northwind

Base de données complète

Différentielle

Différentielle

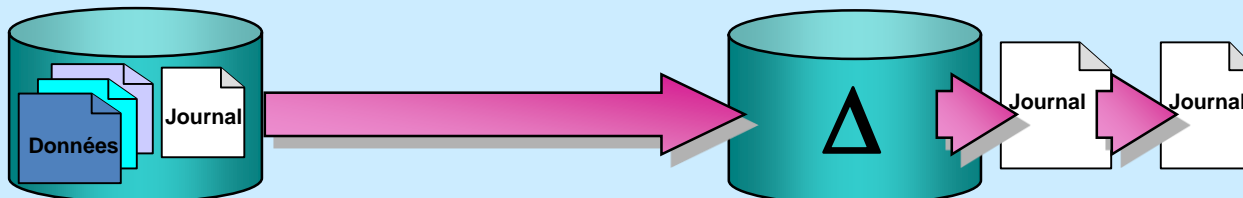


```
USE master
RESTORE LOG Northwind
FROM NwindBacLog
WITH FILE = 2, RECOVERY
```

## Restauration de la base de données Northwind

Base de données complète

Différentielle



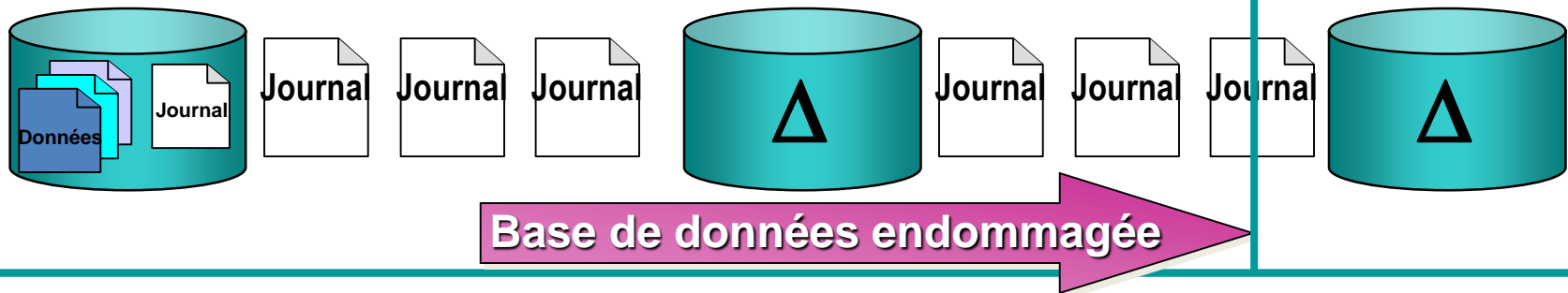
# Spécification d'un point dans le temps (exemple)

## Sauvegardes de la base de données Northwind

Base de données complète

Différentielle

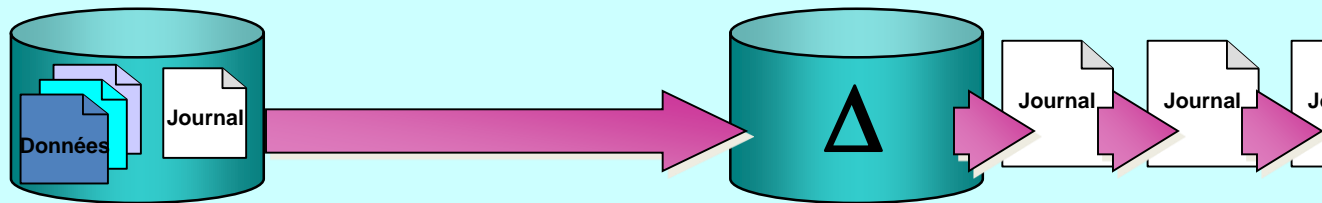
Differentielle



## Restauration de la base de données Northwind

Base de données complète

Différentielle



# Spécification d'un point dans le temps (exemple)

1- *RESTORE DATABASE Northwind*  
*FROM NwindBac*  
*WITH NORECOVERY*

→ restauration à partir d'une sauvegarde complète sans récupérer la BD

2- *RESTORE LOG Northwind*  
*FROM NwindBacLog*  
*WITH FILE=1, NORECOVERY*

→ restauration du 1<sup>er</sup> LOG sans récupérer la BD



# Spécification d'un point dans le temps (exemple-suite)

3- *RESTORE LOG Northwind*

*FROM NwindBacLog*

*WITH FILE=2, RECOVERY,*

*STOPAT='3 janvier 2020, 01:00'*

➔ restaure le 2<sup>ème</sup> LOG, applique les modifications apportées avant le 3 janvier 2020 à 01:00 et récupère la BD

# Restauration à partir d'une sauvegarde de fichier ou de groupe de fichiers

## ➤ Rappel de l'intérêt :

- **Réduire le temps** nécessaire de restauration
- Récupérer des données lorsqu'un **fichier particulier a été endommagé** ou supprimé accidentellement

## ➤ On peut restaurer un fichier de BD à partir de :

- une **sauvegarde complète** de BD
- une **sauvegarde de fichier individuel**

# Restauration à partir d'une sauvegarde de fichier ou de groupe de fichiers (suite)

- **Appliquer tous les LOG** créés depuis la sauvegarde du fichier
  - ➔ **rétablir la cohérence** du fichier restauré par rapport au reste de la BD
- **Restaurer les sauvegardes** de groupes de fichiers **conjointement** si une table et ses index sont sur deux groupes de **fichiers différents**

# Syntaxe partielle

```
RESTORE DATABASE {BD | @var_nom_BD}  
<fichier_ou_groupe_fichiers> [,...m]  
[FROM <unité_sauveragdre> [,...n]]
```

Où <fichier\_ou\_groupe\_fichiers> est :

-> {**FILE**=nom\_fichier\_logique

-> **FILEGROUP**=nom\_groupe\_fichier\_logique}

# Exemple

- La BD **B** utilise **3 fichiers** distincts : **F1, F2 et F3**
- **F2** contient une seule **table** et ses **index**
- **F2 sauvegardé** sur le fichier de sauvegarde

## **F2Bac**

- Une **sauvegarde du LOG** a été effectuée depuis la dernière sauvegarde de F2
- Le **support physique** contenant **F2** a été **endommagé**

# Exemple (suite)

**Etape 1** : on restaure la sauvegarde du fichier de BD F2 **sans transmettre** les transactions validées ni annuler celles non validées

USE master

RESTORE DATABASE B

FILE=F2

WITH NORECOVERY

# Exemple (suite)

**Etape 2** : on restaure la sauvegarde du LOG, on transmet les transactions validées et on annule celles non validées

USE master

RESTORE LOG B

FILE=FBacLog

WITH RECOVERY

# 7. Restauration de Base de Données

7.1 Processus de récupération de SQL Server

7.2 Préparation de la restauration d'une BD

7.3 Restauration de sauvegardes

7.4 Restauration de BD à partir de différents types de sauvegardes

**7.5 Restauration de BD système endommagées**



## 7.5 Restauration de BD système endommagées

- **Obligatoire** si le support physique contenant les **BD système** est **endommagé**
- **Si on peut démarrer SQL Server**, on doit utiliser l'instruction **RESTORE DATABASE** ou SQL Server Enterprise Manager pour restaurer les **BD système**

## 7.5 Restauration de BD système endommagées (suite)

- Si la BD **master est endommagée** et **on ne peut pas démarrer SQL Server** alors :
  - **Reconstruire** les BD système à l'aide de l'utilitaire de ligne de commande **Rebuildm.exe** situé dans le dossier 80\Tools\Bin (2000)
  - **Redémarrer** le service SQL Server
  - Restaurer les sauvegardes des BD système **master, msdb, model** à partir des sauvegarde

# Rattachement ou restauration de BD utilisateur

## ➤ Cas 1 :

la BD **master a été restaurée** à partir  
d'une sauvegarde valide :

- elle **contient des références** à chaque à  
chaque BD utilisateur
- **Donc**, **aucune autre action** n'est requise

# Rattachement ou restauration de BD utilisateur (suite)

## ➤ Cas 2 :

la BD **master a été reconstruite** et qu'aucune sauvegarde valide n'a été appliquée :

- **Soit restaurer les BD utilisateurs** à partir d'une sauvegarde
- **Soit rattacher les fichiers des BD utilisateur** existantes à la **nouvelle BD master** par le procédure stockée système **sp\_attach\_db** ou **sp\_attach\_single\_file\_db**

# Exemple

```
USE master
```






```
EXEC sp_attach_single_file_db
```

```
@dbname='Northwind',
```

```
@physname='Mssql\Data\Northwind.mdf'
```

**➔ Rattachement de la BD Northwind à la BD master**

# Conseils pratiques

-  Rassemblez des informations sur les sauvegardes que vous envisagez de restaurer
-  Utilisez l'option NORECOVERY si vous avez d'autres sauvegardes à restaurer
-  Utilisez l'option RECOVERY sur la dernière sauvegarde pour rétablir la cohérence de la base de données
-  Ajouter une marque dans le journal afin que la base de données soit restaurée jusqu'à un point antérieur au début de l'opération
-  Testez régulièrement vos fichiers de sauvegarde à l'aide de l'instruction RESTORE VERIFYONLY