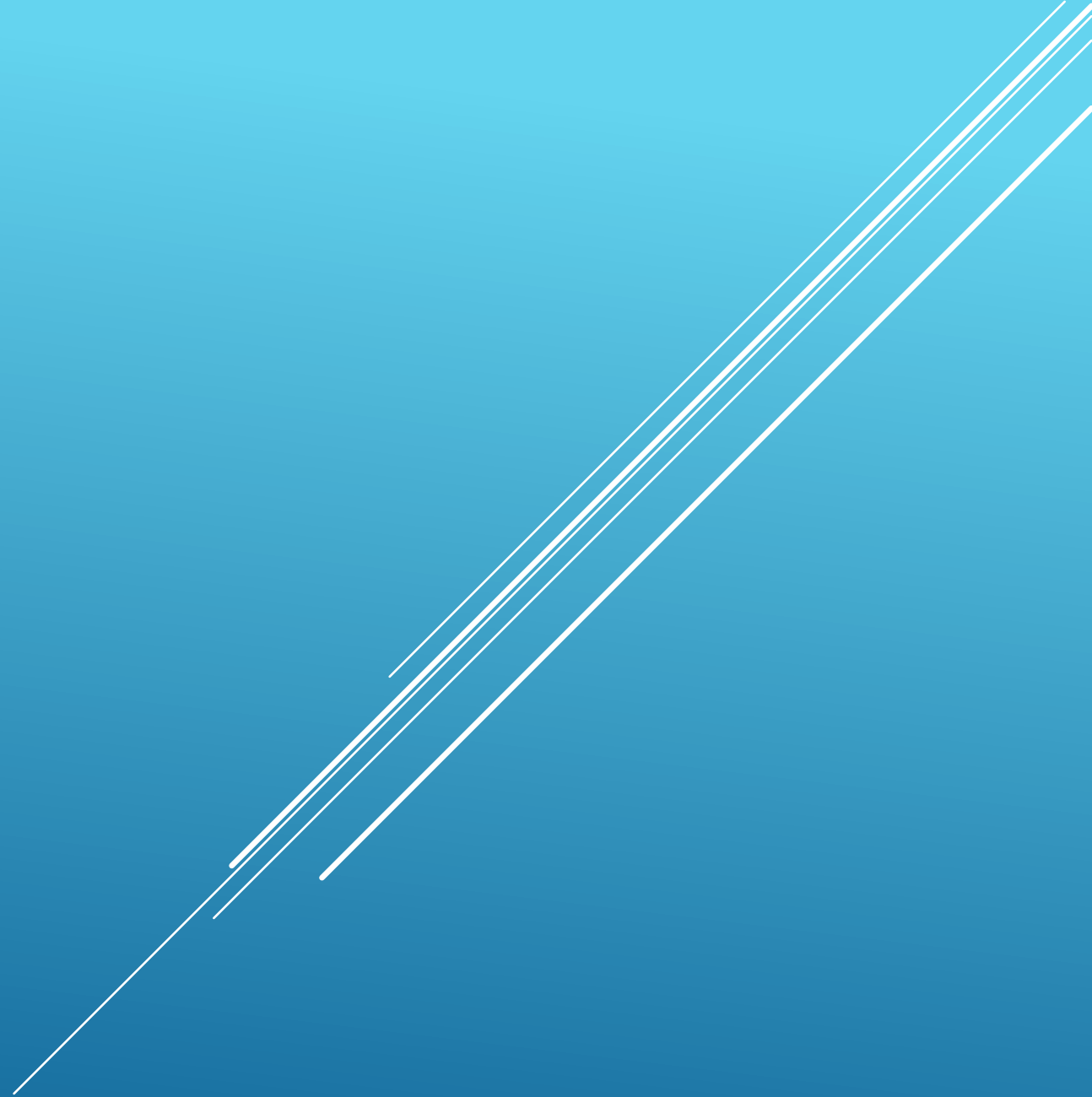


IMPORTATION ET EXPORTATION DE DONNÉES



L'importation des données consiste à récupérer des données depuis une source extérieure à SQL Server et à stocker ces données dans une ou plusieurs tables d'une base SQL Server.

L'exportation représente l'opération inverse, le contenu d'une table, ou le résultat d'une requête est projeté dans un fichier ASCII, ou directement dans une base Access par exemple.

L'importation est en général une opération ponctuelle qui intervient entre la création et la conception de base et la mise en service de la base.

Les opérations d'exportation sont normalement moins fréquentes.

En effet, si les données doivent être manipulées depuis des outils comme Microsoft Access ou Excel, il est préférable de travailler directement sur les données stockées dans SQL Server.

L'exportation reste la solution idéale lorsqu'un utilisateur souhaite établir des calculs sur les données en travaillant en mode.

LES OUTILS

Le transfert d'informations d'une base vers une autre est une opération courante qui doit être réalisée rapidement et de façon sûre. Pour répondre aux différents cas d'utilisation qui peuvent se présenter, SQL Server propose différentes options.

Assistant d'import / export

L'utilitaire BCP

Les requêtes SELECT INTO / INSERT

L'utilitaire BCP

BCP (Bulk Copy Program) est un puissant utilitaire en ligne de commande. Il est utilisé lorsque le volume des transferts entre fichiers texte et base SQL Server est important.

BCP permet d'exporter les données d'une table ou d'une requête SQL vers un fichier texte ou bien d'importer un fichier texte dans une table.

L'utilisation de BCP ne nécessite pas de connaissance particulière en Transact SQL, sauf dans le cas où l'exportation s'appuie sur une requête SQL. Par contre, la description du fichier de données est un point de passage obligatoire.

L'utilitaire BCP

```
bcp {nom_complet_objet| "requête"}  
  {in | out | queryout| format} fichier_de_données  
  [-mmaximum_d'erreurs] [-ffichier_de_format] [-x]  
  [-efichier_d'erreurs] [-dnomBasedeDonnées]  
  [-Fpremière_ligne] [-Ldernière_ligne]  
  [-btaille_de_lot_d'instructions]  
  [-n] [-c] [-w] [-N] [-V{80|90|100}] [-q] [-n]  
  [-Cpage_de_code] [-tfin_de_champ]  
  [-rfin_de_ligne] [-ifichier_d'entrée]  
  [-ofichier_de_sortie] [-ataille_de_paquet] [-kReadOnly]  
  [-Snom_du_serveur] [-Uid_de_connexion] [-Pmot_de_passe] [-q]  
  [-T] [-v] [-R] [-k] [-E] [-h"option [, ...n]"]  
nom_complet_objet | "requête"
```

L'utilitaire BCP

Il s'agit de donner le nom complet de l'objet (table ou vue) ou bien de préciser une requête dont le résultat sera exporté dans un fichier.

Dans le cadre d'une exportation, il est possible de prendre comme objet une table, une vue ou bien de demander l'exécution d'une requête.

L'utilisateur doit posséder les droits de sélection appropriés.

L'utilitaire BCP

Dans le cas d'une importation de données, elle ne peut bien sûr être réalisée au travers d'une requête.

Il est parfois possible de réaliser des insertions au travers d'une vue si certaines précautions ont été prises lors de la définition de cette dernière.

Lors de l'étape d'importation il faut que l'utilisateur soit membre du rôle **db_owner**.

L'utilitaire BCP

`{ in | out | queryout | format } fichier_de_données`

Les options indiquent s'il s'agit d'une opération d'exportation (`out`) ou d'importation (`in`) de données.

Si une requête a été spécifiée au lieu d'un nom d'objet, il faudra utiliser **queryout**.

Enfin **format** permet de créer un fichier de format suivant les options spécifiées, l'option **f** pourra être utilisée.

Les autres options permettent de régler le type du fichier (texte...), les délimiteurs de champs et de lignes...

L'utilitaire BCP

`-S, -U, -P et -T`

Ces quatre options permettent de se connecter à un serveur SQL et d'ouvrir une connexion.

Le nom du serveur est précisé par l'option **-S**.

Par la suite, il est possible d'ouvrir une session soit en utilisant une authentification Windows (**-T**), soit en utilisant une authentification SQL Server.

Il faut alors fournir le nom de connexion (**-U**) et le mot de passe (**-P**).

L'utilitaire BCP

Exemple d'export :

```
bcp Atelier.dbo.Magiciens out c:\bd\magiciens.csv -c -t; -T
```

DÉPLACEMENT D'UNE BASE DE DONNÉES



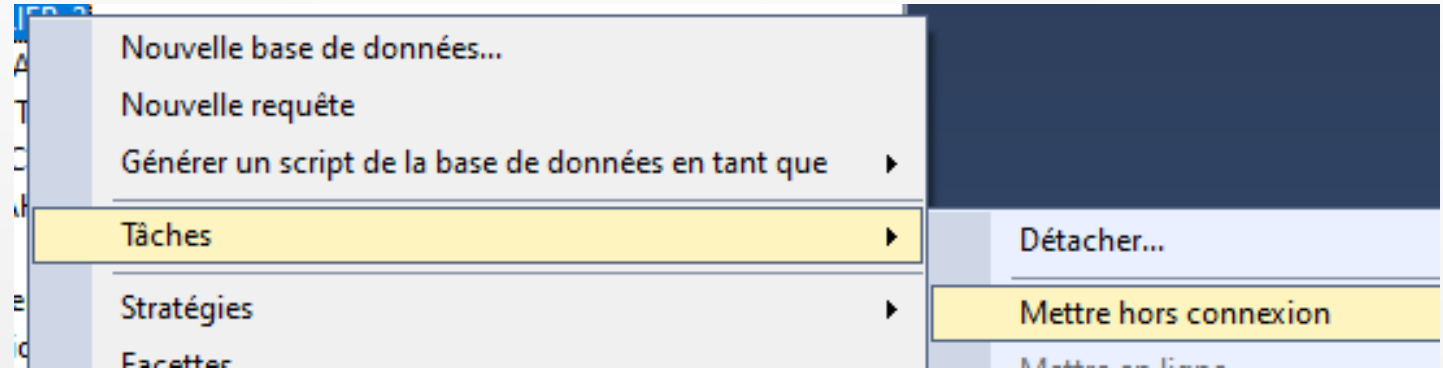
Votre instance de base de données possède déjà des chemins par défaut pour l'emplacement des fichiers pour vos bases. Cela fait partie des paramètres de votre instance.

Toutefois, on pourrait vous demander de déplacer les fichiers sur un autre emplacement.

Nous allons voir point par point quelles sont les étapes à suivre ainsi que les points d'attention à avoir

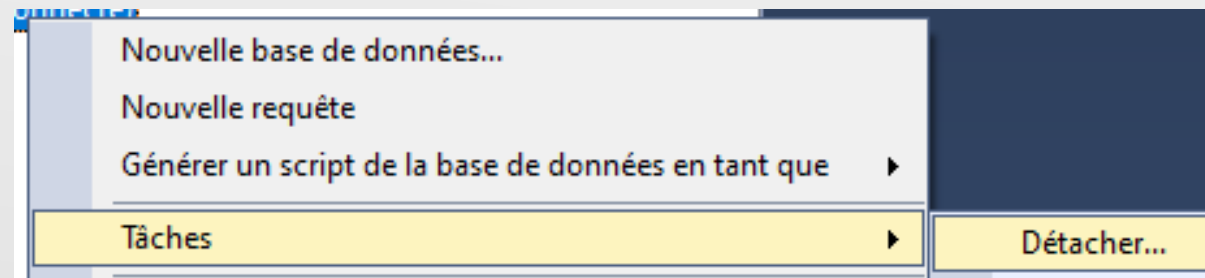
1 – Mettre sa base de données « Hors connexion »

Cette opération importante permet de s'assurer qu'aucune connexion est active, donc plus aucune opération n'aura lieu sur notre serveur.



2 – Détacher sa base de données

Cette opération permet de libérer les fichiers.



3 - Déplacer les fichiers dans le nouvel emplacement.

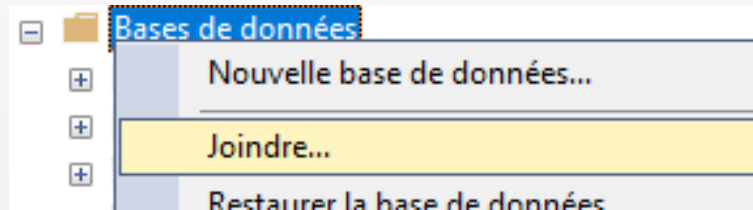
Les deux fichiers sont à prendre en compte :

- Le fichier de données `<database>.mdf`
- Le fichier de transaction `<database>_log.ldf`

Il est conseillé, dans une configuration de production, de placer ces fichiers dans deux emplacements différents.

4 – Rattacher la base de données

Cette opération consiste à « recréer » notre base de données à partir des fichiers situés dans le nouvel emplacement.



Dans l'assistant qui s'ouvre, cliquer sur « Ajouter » puis sélectionner le fichier de données.

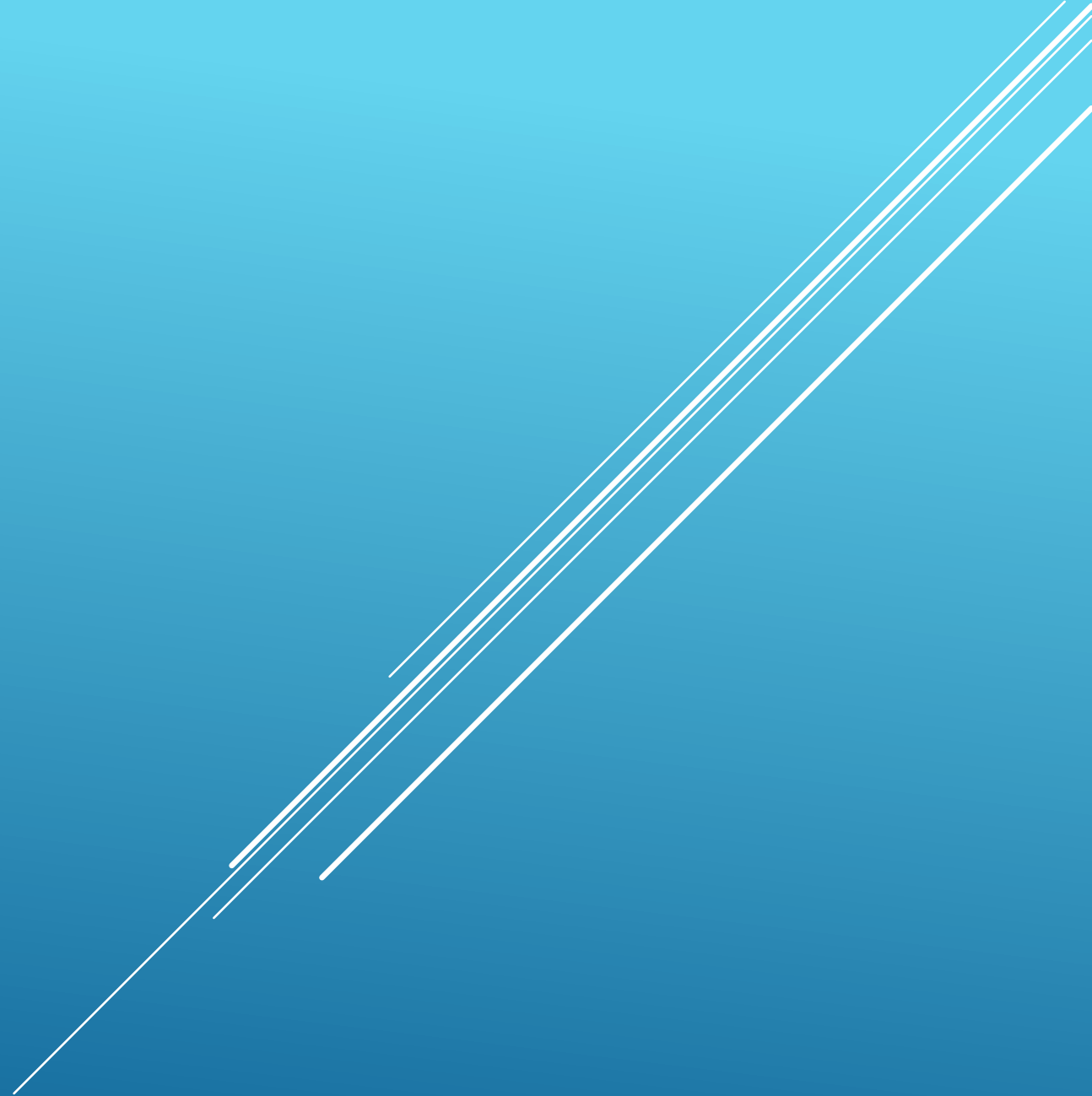
Si l'assistant ne récupère pas automatiquement le fichier de transactions associé, Il faudra renseigner cet emplacement.

5 – Vérifier le bon fonctionnement de la base

Tout s'est bien passé, la base de données apparaît à nouveau dans la liste des Bases de données.

Il reste à contrôler le bon fonctionnement en vérifiant si les connexions fonctionnent toujours sur cette base.

SAUVEGARDES



La gestion des sauvegardes reste l'une des tâches les plus importantes qui doivent être réalisées par l'administrateur de bases de données.

Si les opérations de sauvegardes sont planifiées avec exactitude et rigueur, il est tout à fait envisageable de réaliser les sauvegardes sous forme de travaux automatisés et le responsable sera prévenu par e-mail du bon déroulement des opérations.

Avec des sauvegardes valides d'une base de données, vous pouvez récupérer vos données suite à de nombreuses défaillances, par exemple :

- défaillance du support ;
- erreurs utilisateur, telles que la suppression d'une table par inadvertance ;
- défaillances matérielles, telles qu'un lecteur de disque endommagé ou la perte permanente d'un serveur ;
- catastrophes naturelles.

Pensez à stocker vos sauvegardes hors site.

Les sauvegardes SQL Server permettent de sauvegarder la base de données alors que des utilisateurs y sont connectés.

Cette sauvegarde va prendre en compte tous les fichiers constituant la base de données et va enregistrer leur emplacement.

Le processus de sauvegarde assure la cohérence des données et des journaux en capturant toutes les activités survenant durant le processus de sauvegarde.

Sauvegarde d'une base de données

La sauvegarde totale d'une base de données permet de fournir un point de départ pour les restaurations.

Si uniquement des sauvegardes complètes de base de données sont effectuées, en cas de problème, les transactions validées depuis la dernière sauvegarde complète seront perdues.

Les sauvegardes complètes nécessitent un temps relativement long et occupent sur le support de sauvegarde un espace conséquent.

Sauvegarde du journal des transactions

En complément des sauvegardes complètes, il est toujours possible de mettre en place une politique de sauvegarde des journaux de transactions. La sauvegarde des journaux présente deux avantages majeurs.

Il est possible de récupérer la totalité des transactions validées depuis la dernière sauvegarde complète de la base.

La taille du fichier journal ne risque pas de grandir de façon anarchique car lors de chaque sauvegarde du journal, celui-ci sera tronqué. Le risque de saturer le disque suite à l'extension du fichier journal est ainsi diminué. La sauvegarde des journaux peut être réalisée par un travail qui est planifié pour une exécution régulière.

Sauvegarde du journal des transactions

SQL Server génère des points de contrôle synchronisation (CHECKPOINT) automatiques.

Le but de ces points de synchronisation est de stocker sur disque l'ensemble des transactions validées pour lesquelles les modifications sont encore en mémoire.

Le point de synchronisation peut être déclenché ponctuellement par l'intermédiaire de l'instruction Transact SQL CHECKPOINT.

```
CHECKPOINT [valeur] [;]
```


Sauvegarde du journal des transactions

Le point de synchronisation peut également être réalisé de façon automatique si le moteur de base de données est arrêté.

La fréquence des points de synchronisation est fixée par SQL Server en fonction de l'activité de mise à jour sur la base.

Sauvegardes différentielles

Si les sauvegardes complètes de base de données sont des opérations lourdes sur les très grosses bases, une solution intéressante peut être mise en place avec les sauvegardes différentielles.

Les sauvegardes différentielles ne vont prendre en compte que les données modifiées depuis la dernière sauvegarde complète.

Les sauvegardes différentielles sont plus rapides et moins volumineuses que les sauvegardes complètes, et associées aux sauvegardes du journal des transactions, elles peuvent constituer une solution de sauvegarde à la fois rapide et performante.

Les combinaisons possibles

La bonne solution pour réaliser des sauvegardes passe par une combinaison des différentes méthodes de sauvegarde.

Par exemple, il est possible de combiner les sauvegardes complètes, les sauvegardes différentielles et les sauvegardes du journal des transactions pour minimiser à la fois le volume et le temps des sauvegardes, et la perte des transactions validées.

En mettant en place ces opérations sous forme de travaux planifiés avec notification en fin d'exécution, les opérations de sauvegardes peuvent se dérouler automatiquement sans effort de la part de l'administrateur.

Bonnes Pratiques :

Utiliser un stockage distinct

Veillez à placer vos sauvegardes de bases de données sur un appareil ou à un emplacement physique distinct des fichiers de base de données.

Bonnes Pratiques :

Choisir le mode de récupération approprié

Les opérations de sauvegarde et de restauration interviennent dans le cadre d'un mode de récupération.

Ainsi, le mode de récupération d'une base de données détermine les types de sauvegardes et les scénarios de restauration pris en charge pour la base de données

En règle générale, une base de données utilise le mode de récupération complète ou le mode de récupération simple.

Bonnes Pratiques :

Concevoir votre stratégie de sauvegarde

Le choix de la meilleure stratégie de sauvegarde possible dépend d'un éventail de facteurs, notamment des facteurs primordiaux suivants :

- Combien d'heures par jour les applications ont-elles besoin d'accéder à la base de données ?
- Quelle est la fréquence de modification et de mise à jour possible ?
- Ces modifications risquent-elles de porter sur une petite ou sur une grande partie de la base de données ?
- Pendant combien de temps votre entreprise doit-elle conserver les sauvegardes ?

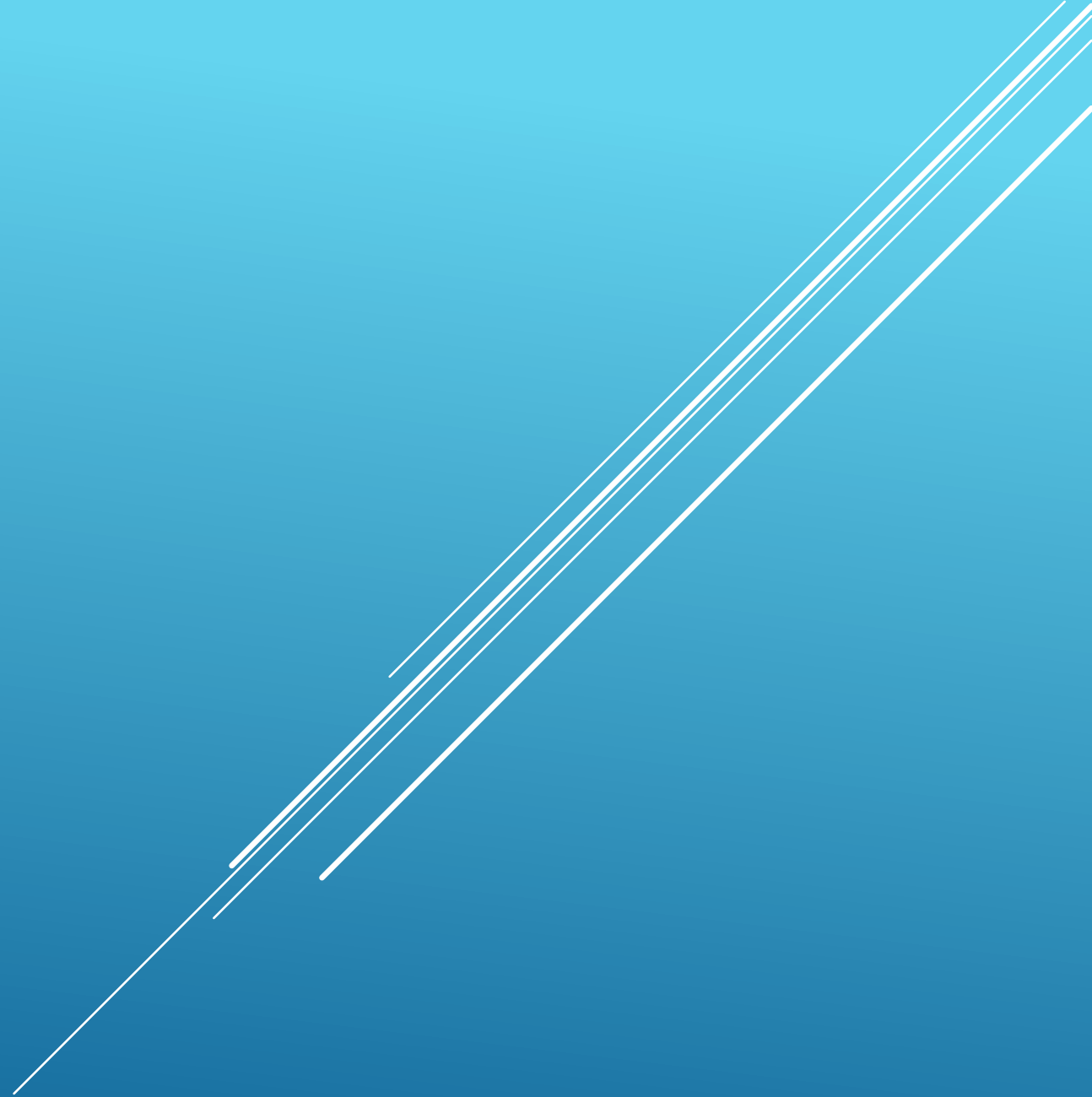
Bonnes Pratiques :

NE PAS OUBLIER :

Toutes les stratégies de sauvegarde commencent **toujours** par une sauvegarde complète de la base.

Mise en pratique !

RESTAURATIONS



La restauration représente l'opération inverse d'une sauvegarde. Le processus de restauration peut également être utilisé pour réaliser des migrations de bases de données utilisateurs.

Cette opération peut être réalisée soit à l'aide des commandes Transact SQL (RESTORE) soit par l'intermédiaire de la console d'administration SQL Server Management Studio.

Le simple fait de remplacer les fichiers constituant les données et le journal ne constitue en aucun cas une restauration, puisque SQL Server n'est pas capable d'accéder à ces fichiers tant qu'ils ne sont pas référencés dans la base Master.

SQL Server réalise automatiquement un certain nombre d'opérations afin d'accélérer le processus de restauration et de réduire au maximum le temps d'indisponibilité du serveur.

- **Le contrôle de sécurité**

L'intérêt principal de ce contrôle de sécurité est de se prémunir des restaurations accidentelles, qui peuvent écraser une base existante pour remonter une version précédente de la base. De même, le contrôle de sécurité va s'assurer qu'il possède tous les fichiers participant à un jeu de sauvegarde.

Le contrôle de sécurité interdira également la restauration de la base si le jeu des fichiers constituant la base est différent de celui enregistré par le jeu de sauvegarde.

SQL Server réalise automatiquement un certain nombre d'opérations afin d'accélérer le processus de restauration et de réduire au maximum le temps d'indisponibilité du serveur.

- **Reconstruction de la base de données et des fichiers associés**

Dans le cadre d'une restauration à partir d'une sauvegarde complète de la base, SQL Server se charge de recréer la base et les fichiers qui la composent.

Les objets sont créés et les données sont transférées.

Le schéma de la base de données est donc reconstruit automatiquement durant la procédure de restauration et ne nécessite pas d'opérations manuelles.

Opérations préliminaires

Avant de réaliser une restauration, il est important de réaliser quelques opérations préliminaires afin de s'assurer du bon déroulement du processus.

- La vérification des supports
- S'assurer qu'aucun utilisateur ne travaille sur la base

```
select login_name,program_name,database_id,host_name
from sys.dm_exec_sessions where is_user_process=1
```
- Sauvegarde du journal des transactions

Mise en pratique !