

Secteur Tertiaire Informatique Filière étude - développement

Activité « Développer la persistance des données »

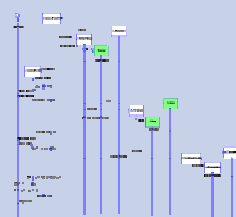
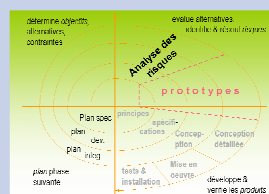
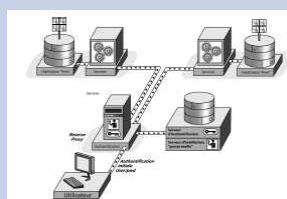
Sauvegarde et restauration

Accueil

Apprentissage

Période en
entreprise

Evaluation



Code barre

SOMMAIRE

I	DEFINITION D'UNE STRATEGIE DE SAUVEGARDE	4
I.1	Se poser les bonnes questions.....	4
I.2	Les principaux types de sauvegarde.....	4
I.3	Differentes stratégies de sauvegarde	5
II	SAUVEGARDES AVEC SQL SERVER	8
II.1	Sauvegarde des bases de données système.....	8
II.2	Mode de récupération de base de données	8
II.3	Execution des sauvegardes	9
II.3.1	Création d'unités de sauvegarde.....	9
II.3.2	Sauvegarder la base	9
II.3.3	Exemples	12
III	RESTAURATION AVEC SQL SERVER.....	13
III.1	Stratégie de restauration	13
III.2	Exécution de la restauration.....	13
III.2.1	Les opérations préliminaires	14
III.2.2	Restaurer la base.....	14
III.2.3	Exemples	16
IV	AUTOMATISATION DES SAUVEGARDES	17
IV.1	L'agent SQL Server.....	17
IV.2	Le plan de maintenance	18

L'information est le moteur de l'entreprise, et la protection contre les pertes de données est essentielle.

Des données peuvent être perdues à la suite de :

- Une panne de support
- Une panne logicielle
- Un usage inopportun d'instructions de mise à jour ou de suppression
- Des virus destructeurs
- Des catastrophes humaines
- Des catastrophes naturelles (incendie, inondation ...)

La définition d'une stratégie de sauvegarde est nécessaire pour réduire au minimum les risques de perte de données et la récupération des données avec un temps d'arrêt acceptable du système.

I DEFINITION D'UNE STRATEGIE DE SAUVEGARDE

I.1 SE POSER LES BONNES QUESTIONS

Quel type de base de données sauvegarder ?

Les bases de données système et utilisateur présentent des exigences de sauvegarde et restauration différentes.

Quelle est l'importance des données à sauvegarder dans la base?

Une base de données de développement nécessitera peut-être une sauvegarde hebdomadaire, alors qu'une base de données de production peut nécessiter une sauvegarde journalière.

Quelle est la fréquence des modifications ?

Une base de données actualisée en permanence requiert des sauvegardes continues, alors qu'une base de données mise à jour de nuit ne nécessitera qu'une sauvegarde après la mise à jour.

Quel est le meilleur moment pour effectuer les sauvegardes ?

Il est préférable de choisir les moments où l'activité de la base est la plus faible, ce qui accélère le processus de sauvegarde. Une base peut être sauvegardée même si elle est active, certaines opérations sur la base ne pouvant toutefois avoir lieu (croissance automatique, création d'index ...)

I.2 LES PRINCIPAUX TYPES DE SAUVEGARDE

La sauvegarde de base de données complète copie tous les objets, tables système, données et les lignes du journal des transactions nécessaires à la restauration dans l'état où la base se trouve en fin de sauvegarde. Elle sert de point de référence lors d'une défaillance du système.

La sauvegarde différentielle est conçue pour réduire la durée de restauration d'une base de données fréquemment modifiée. Seules les données modifiées depuis la dernière sauvegarde complète de la base de données et les lignes du

journal des transactions nécessaires à la restauration sont sauvegardées.

Attention :

- une sauvegarde différentielle requiert une sauvegarde de base de données complète.
- lors d'une succession de modifications sur une même ligne de données, seule la dernière est sauvegardée dans la sauvegarde différentielle ;

La sauvegarde du journal des transactions est effectuée pour enregistrer toutes les modifications apportées à la base de données ; la sauvegarde stocke les modifications intervenues depuis la dernière sauvegarde de ce journal, puis le vide par défaut de toutes les transactions confirmées ou annulées(tronquer).

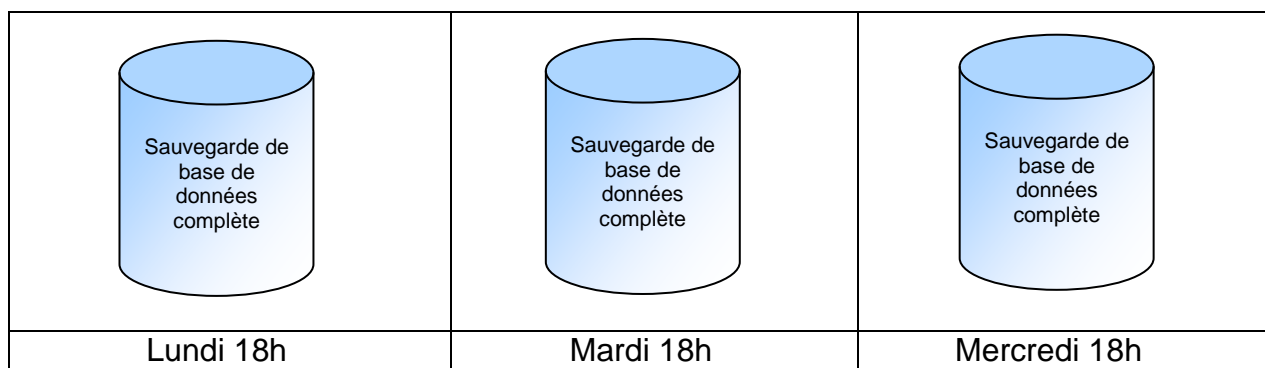
La sauvegarde d'un fichier ou groupe de fichiers est intéressante dans le cas de base de données volumineuses : elle permet de sauvegarder la base en plusieurs opérations, tout en gardant leur cohérence en sauvegardant aussi le journal des transactions.

I.3 DIFFERENTES STRATEGIES DE SAUVEGARDE

Sauvegarde complète de base de données

La base de données est petite, sa durée de sauvegarde est de l'ordre de quelques minutes ; elle est très peu modifiée, et le risque de perdre une journée de modifications est acceptable.

Une sauvegarde complète de base de données est effectuée tous les jours à 18 heures.

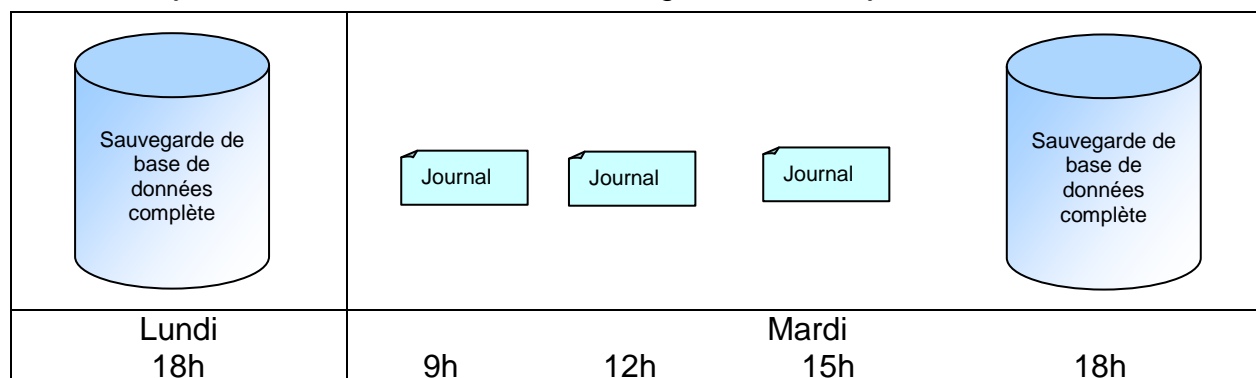


En cas de dommage sur la base mercredi à 13h, la sauvegarde complète de mardi 18h sera restaurée : toutes les modifications effectuées entre mardi 18h et mercredi 13h seront perdues.

Sauvegarde complète de base de données et sauvegarde du journal des transactions

Les modifications apportées à la base sont plus fréquentes que dans le cas précédent, le journal des transactions est stocké sur une unité physique différente de celle de la base de données.

Une sauvegarde complète de base de données est effectuée tous les jours à 18 heures, le journal des transactions est sauvegardé tous les jours à 9h, 12h et 15h.



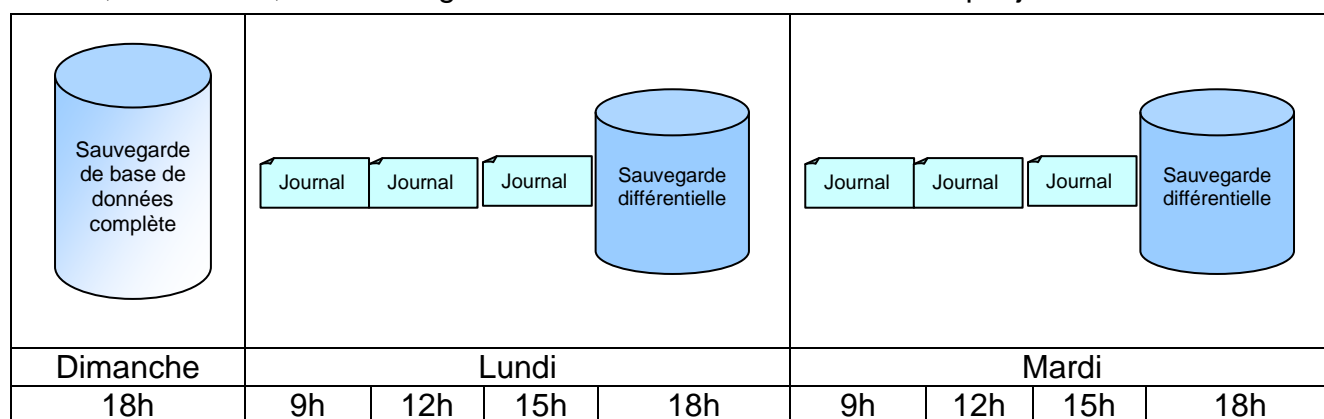
En cas de dommage sur la base mardi à 13h, la sauvegarde complète de lundi 18h sera restaurée puis les journaux successifs seront appliqués.

Si le journal des transactions en cours peut être sauvegardé avant la restauration, la perte de données est minimum, sinon la perte de données ne concernera que les modifications apportées sur la base entre 12h et 13h.

Sauvegarde différentielle

Cette stratégie sera appliquée pour réduire la durée de restauration de la base de données endommagée : plutôt que d'appliquer plusieurs journaux de transactions volumineux, une sauvegarde différentielle sera effectuée pour appliquer les modifications apportées à la base depuis la dernière sauvegarde complète.

Une sauvegarde complète de la base est effectuée une fois par semaine, le dimanche à 18h ; le journal des transactions est sauvegardé tous les jours ouvrés à 9h, 12h et 15h ; une sauvegarde différentielle est effectuée chaque jour ouvré à 18h.



En cas de dommage sur la base mardi à 13h,

Sauvegardez si possible, le journal des transactions en cours.

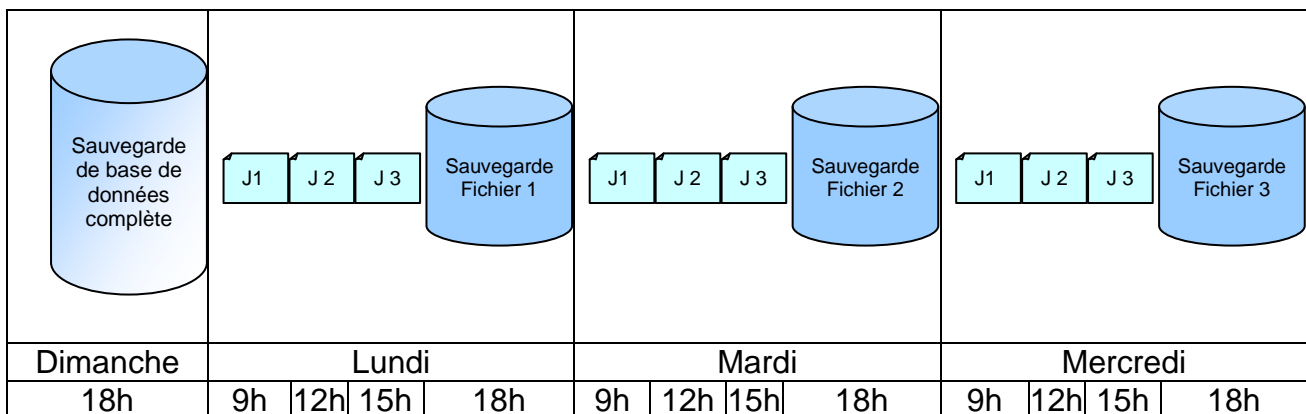
Restaurez la sauvegarde complète de dimanche 18h, la sauvegarde différentielle de lundi 18h, et les journaux de transaction de mardi 9h et 12h.

Si le journal des transactions en cours a pu être sauvegardé avant la restauration, la perte de données est minimum, sinon la perte de données ne concernera que les modifications apportées sur la base entre 12h et 13h, avec moins de manipulation de journaux.

Sauvegarde de fichiers ou de groupe de fichiers de base de données

Cette stratégie sera appliquée pour des bases de données très volumineuses partitionnées sur plusieurs fichiers, dans le but de gagner du temps de sauvegarde (éclatement d'une sauvegarde de 3 heures pendant 1 heure par jour).

Une sauvegarde complète de la base, répartie sur les fichiers 1 2 et 3 est effectuée une fois par semaine, le dimanche à 18h ; le fichier 1 est sauvegardé lundi à 18h, le fichier 2 mardi à 18h, le fichier 3 mercredi à 18h ; le journal des transactions est sauvegardé à 9h, 12h et 15h tous les jours.



En cas de dommage sur le support du fichier 2 jeudi à 13h, Sauvegardez si possible, le journal des transactions en cours.

Restaurez la sauvegarde du fichier 2 créée mardi 18h et appliquez tous les journaux de transaction créés depuis mardi 18h.

Le gain de performance réalisé vient de ce que seules les transactions concernant le fichier 2 sont appliquées.

II SAUVEGARDES AVEC SQL SERVER

II.1 SAUVEGARDE DES BASES DE DONNEES SYSTEME

Les bases de données système contiennent des données relatives à SQL Server et toutes les bases de données utilisateur ; elles doivent donc être sauvegardées à intervalles réguliers, et à chaque modification.

La base de données master doit être sauvegardée après chaque création, modification ou suppression de base de données ou d'une connexion utilisateur, à l'ajout de messages d'erreur par la procédure stockée **sp_addmessage**, ou au changement de configuration du serveur

La base de données msdb contient toutes les informations sur les travaux, alertes et opérateurs utilisés par l'agent SQL Server et doit donc être sauvegardée à chaque ajout, modification ou suppression.

La base de données model contient le paramétrage par défaut utilisé lors de la création de toute nouvelle base de données ; elle est donc à sauvegarder lors de la modification de ces informations.

II.2 MODE DE RECUPERATION DE BASE DE DONNEES

SQL Server 2005 utilise trois modes de récupération dont le choix est guidé par le type de base de données concernée et le type de sauvegarde.

Mode de récupération simple :

Les bases de données concernées par ce mode sont des bases peu volumineuses, où les données sont peu modifiées. La stratégie de sauvegarde se compose uniquement de sauvegardes complètes et différentielles, la sauvegarde du journal des transactions est impossible.

Le principal avantage de ce mode réside dans le faible espace de stockage occupé par les journaux et dans une mise en œuvre simple.

Mode de récupération complète :

SQL Server consigne toutes les modifications apportées à la base, y compris les opérations de chargement en bloc et les créations d'index. Avec ce mode de récupération, et si le dernier journal n'est pas endommagé, SQL Server est en mesure de récupérer toutes les données à l'exception des transactions en cours au moment de la défaillance.

Une récupération peut également être effectuée jusqu'à une marque précise.

Mode de récupération journalisé en bloc (Bulk_Logged):

Ce mode de récupération réduit l'espace de journalisation, car dans ce mode, le système ne stocke pas les détails des opérations suivantes : CREATE INDEX, opérations de chargement de bloc, SELECT INTO, WRITETEXT et UPDATETEXT. Dans la mesure où seul le résultat final est consigné, le journal est plus petit et les opérations de restauration sont plus rapides.

Il n'est toutefois pas possible dans ce mode, de restaurer jusqu'à une marque donnée.

Chaque base de données peut se voir appliquer un mode de récupération différent. Par défaut, **master**, **msdb** et **tempdb** appliquent le mode simple, et **model**, le mode complet.

Toutes les nouvelles bases utilisateur créées adoptent le mode de récupération de **model**.

Pour modifier le mode de récupération de la base :

Syntaxe sous Transact-SQL :

```
ALTER DATABASE nom_base  
SET RECOVERY { FULL | BULK_LOGGED | SIMPLE }
```

Sous SQL Server Management Studio, vous modifierez le mode de récupération dans l'onglet **Options** des **propriétés** de la base.

II.3 EXECUTION DES SAUVEGARDES

Les sauvegardes sur disques ou bandes se font par l'intermédiaire de fichiers de sauvegarde temporaires ou permanents pour l'automatisation et la réutilisation.

II.3.1 Création d'unités de sauvegarde

Un fichier de sauvegarde créé avant d'être utilisé est appelé unité de sauvegarde et peuvent se créer avec **Transact-SQL** à l'aide de la procédure stockée système **sp_addumpdevice**.

Syntaxe sous Transact-SQL :

```
sp_addumpdevice  
    [ @devtype = ] 'type_unité',  
    [ @logicalname = ] 'nom_logique',  
    [ @physicalname = ] 'nom-physique'
```

type_unité représente le type de support utilisé (disk, tape), *nom_logique* le nom de l'unité de sauvegarde, *nom-physique* le chemin d'accès complet au fichier de sauvegarde.

Exemple : Création d'une unité logique sur disque, de nom savVentes ; le nom du fichier de sauvegarde est ventesbk.bak et se trouve dans le répertoire C:\Backup.

Exec sp_addumpdevice 'disk', 'savVentes', 'C:\Backup\ventesbk.bak'

Sous SQL Server Management Studio, vous créerez une nouvelle unité de sauvegarde en cliquant dans le menu contextuel du dossier **Objets Serveur\Unités de Sauvegarde**.

II.3.2 Sauvegarder la base

La base de données et les journaux sont sauvegardés par l'instruction **BACKUP**, **BACKUP DATABASE** pour la base, **BACKUP LOG** pour les journaux.

Lors de la sauvegarde du journal des transactions, les opérations suivantes sont effectuées :

1. Sauvegarde du journal des transactions à partir de la dernière instruction **BACKUP LOG** jusqu'à la fin des transactions en cours
2. Vidage du journal des transactions terminées

Si les fichiers de base de données sont endommagés, il est conseillé de sauvegarder le journal actif avec l'option **NO_TRUNCATE** qui conserve toutes les transactions effectuées, ce qui peut permettre de récupérer les données.

Le contenu du journal doit être vidé régulièrement pour conserver une taille acceptable: l'option **TRUNCATE_ONLY** de **BACKUP LOG** permet de tronquer le journal, sans sauvegarde.

Syntaxe simplifiée sous Transact-SQL :

BACKUP DATABASE *nom_base*

[FILE = { *nom_fichier* | @*variable_texte* }

|

FILEGROUP = { *nom_groupefichiers* | @ *variable_texte* }

| READ_WRITE_FILEGROUPS]

TO *unite_sauvegarde*

[WITH

[[,] DESCRIPTION = { '*text*' | @*variable_texte* }]

[[,] DIFFERENTIAL]

[[,] EXPIREDATE = { *date* | @*variable_date* } | RETAINDAYS = { *jours* | @*variable_jour* }]

[[,] PASSWORD = { *motdepasse* | @*variable_pwd* }]

[[,] { FORMAT | NOFORMAT }]

[[,] { INIT | NOINIT }]

[[,] { NOSKIP | SKIP }]

[[,] MEDIADESCRIPTION = { '*text*' | @*variable_texte* }]

[[,] MEDIANAME = { *nom_media* | @*variable_texte* }]

[[,] MEDIAPASSWORD = { *motdepasse_media* | @*variable_texte* }]

[[,] NAME = { *nom_jeusauvegarde* | @*variable_texte* }]

[[,] { NOREWIND | REWIND }]

[[,] { NOUNLOAD | UNLOAD }]

]

BACKUP LOG *nom_base*

TO *unite_sauvegarde*

[WITH

[[,] DESCRIPTION = { '*text*' | @*variable_texte* }]

[[,] EXPIREDATE = { *date* | @*variable_date* } | RETAINDAYS = { *jours* | @*variable_jour* }]

[[,] PASSWORD = { *motdepasse* | @*variable_pwd* }]

[[,] { FORMAT | NOFORMAT }]

[[,] { INIT | NOINIT }]

[[,] { NOSKIP | SKIP }]

[[,] MEDIADESCRIPTION = { '*text*' | @*variable_texte* }]

[[,] MEDIANAME = { *nom_media* | @*variable_texte* }]

[[,] MEDIAPASSWORD = { *motdepasse_media* | @*variable_texte* }]

[[,] NAME = { *nom_jeusauvegarde* | @*variable_texte* }]

[[,] NO_TRUNCATE]

[[,] NORECOVERY]

[[,] { NOREWIND | REWIND }]

[[,] { NOUNLOAD | UNLOAD }]

]

FILE = { *nom_fichier* | @*variable_texte* }

Nom d'un ou de plusieurs fichiers à sauvegarder.

FILEGROUP = { *nom_groupefichiers* | @ *variable_texte* }

Nom d'un ou de plusieurs groupe de fichiers à sauvegarder.

READ_WRITE_FILEGROUPS

Nom d'un ou de plusieurs groupe de fichiers à sauvegarder.

DESCRIPTION

Texte servant à décrire le jeu de sauvegarde.

DIFFERENTIAL

Indique que la sauvegarde est une sauvegarde différentielle.

EXPIREDATE = { *date* | *@variable_date* }

Indique la date à laquelle le jeu de sauvegarde expire et peut par conséquent être écrasé (variable chaîne ou date).

RETAIN_DAYS = { *jours* | *@variable_jour* }

Indique le nombre de jours qui doivent s'écouler avant que le jeu de supports de sauvegarde puisse être écrasé (variable de type entier).

PASSWORD = { *motdepasse* | *@variable_pwd* }

Définit le mot de passe utilisé avec le jeu de sauvegarde (chaîne de caractères).
Si un mot de passe est défini pour le jeu de sauvegarde, il doit être fourni lors de chaque opération de restauration à partir du jeu.

FORMAT | **NOFORMAT**

FORMAT indique qu'un nouveau support de sauvegarde est créé et invalide le contenu du support existant : un nouvel en-tête de support est créé.

INIT | **NOINIT**

INIT indique que tous les jeux de sauvegardes doivent être écrasés mais préserve l'en-tête de support. Le support de sauvegarde ne sera écrasé que si le jeu de sauvegarde a expiré.

NOSKIP | **SKIP**

Avec **NOSKIP**, SQL Server vérifie la date d'expiration de tous les jeux de sauvegarde avant d'autoriser leur écrasement.

MEDIADESCRIPTION = { *'text'* | *@variable_texte* }

Texte de description du jeu de supports (255 caractères).

MEDIA_NAME = { *nom_media* | *@variable_texte* }

Nom du support de sauvegarde complet (128 caractères).

MEDIA_PASSWORD = { *motdepasse_media* | *@variable_texte* }

Mot de passe utilisé avec le jeu de supports.

NAME = { *nom_jeusauvegarde* | *@variable_texte* }

Nom du jeu de sauvegarde (128 caractères).

NOREWIND | **REWIND**

Libération et rembobinage de la bande dans le cas d'une sauvegarde sur bande.

NOUNLOAD | **UNLOAD**

Déchargement de la bande dans le cas d'une sauvegarde sur bande.

NORECOVERY

Sauvegarde la fin du journal et laisse la base de données en état de restauration.

Sous SQL Server Management Studio, vous pourrez sauvegarder une base par le menu contextuel, en sélectionnant l'option **Tâches**, puis **Sauvegarder...** sur l'onglet **Général**, vous choisirez le type de sauvegarde, l'unité ou le fichier de sauvegarde, et les options d'expiration ; sur l'onglet Options, vous choisirez de créer un nouveau support de sauvegarde ou d'ajouter au support existant, et dans le cas d'une sauvegarde du journal, vous déciderez de le tronquer ou non .

II.3.3 Exemples

Exemple 1: Sauvegarde complète de la base de données Ventes sur l'unité logique. savVentes précédemment créée.

```
BACKUP DATABASE Ventes TO savVentes
```

Exemple 2: Sauvegarde différentielle de la base de données Ventes sur l'unité logique. savVentesDiff précédemment créée.

```
BACKUP DATABASE Ventes TO savVentesDiff With DIFFERENTIAL
```

Exemple 3: Sauvegarde du journal des transactions sur l'unité logique. savVentesLog précédemment créée.

```
BACKUP LOG Ventes TO savVentesLog
```

Exemple 4: Sauvegarde du fichier Ventes_Data de la base de données Ventes sur l'unité logique. savVentes précédemment créée.

```
BACKUP DATABASE Ventes FILE = 'Ventes_data' TO savVentes
```

III RESTAURATION AVEC SQL SERVER

Avant de démarrer une restauration de base de données en cas de défaillance, il peut être intéressant de vérifier que la base est réellement endommagée et ne peut pas être récupérée par un autre moyen.

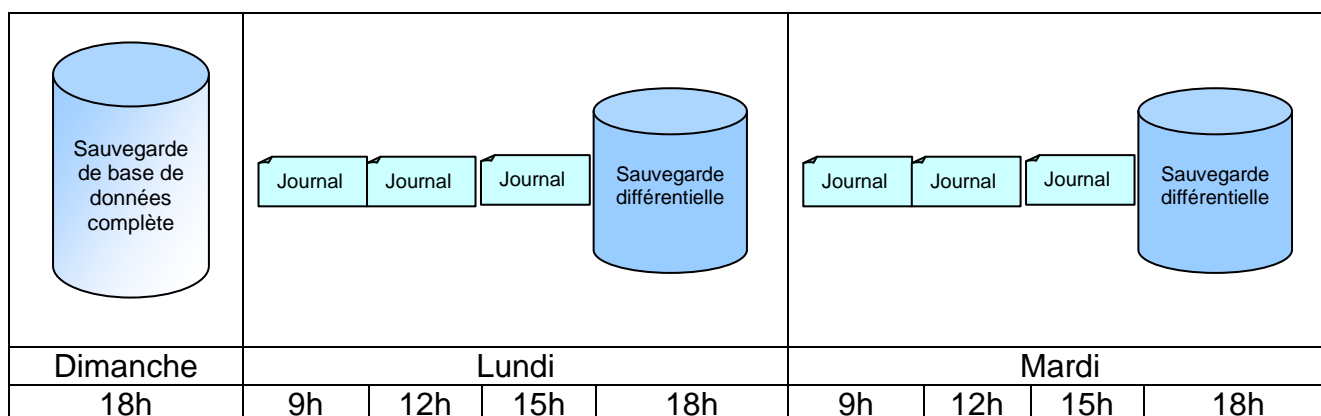
1. en étudiant dans les journaux d'erreur de SQL Server les types d'erreur rencontrés
2. en vérifiant l'état de la base de données par les propriétés de la fonction databaseproperty (IsShutDown, IsEmergencyMode, IsSuspect ...)
3. en utilisant certaines commandes DBCC qui peuvent effectuer des réparations mineures sans perte de données.

La dernière issue est la restauration depuis la dernière sauvegarde.

III.1 STRATEGIE DE RESTAURATION

Sous Management studio, en affichant la boîte de dialogue **Restaurer la base de données**, vous constatez que SQL Server propose une stratégie de restauration dépendante de la stratégie de sauvegarde appliquée : sauvegarde complète, différentielle ou sauvegarde des journaux de transactions.

Dans cette configuration,



En cas de dommage sur la base mardi à 13h,

1. Mettre la base de données en accès restreint pendant la durée de la restauration.
2. Le journal des transactions en cours sera sauvegardé avec l'option NO_TRUNCATE, si il est accessible.
3. La sauvegarde complète de dimanche 18h, la sauvegarde différentielle de lundi 18h, les journaux de transaction de mardi 9h et 12h puis le dernier journal seront successivement restaurés.

III.2 EXECUTION DE LA RESTAURATION

Dans le cas de restauration de bases de données utilisateur endommagées, la base n'a pas besoin d'être supprimée : SQL Server recrée automatiquement les fichiers et objets de base de données

III.2.1 Les opérations préliminaires

Avant de restaurer un fichier de sauvegarde, on vérifiera qu'il contient les sauvegardes attendues, soit avec Management Studio, soit avec l'instruction **RESTORE HEADERONLY** qui renvoie des informations concernant la sauvegarde (nom et description, méthode, date et heure, taille,).

Pour obtenir des informations sur les fichiers de base de données ou journal des transactions contenus dans un fichier de sauvegarde, on exécutera l'instruction **RESTORE FILELISTONLY**.

Pour mettre la base de données en accès restreint pendant la durée de la restauration, on exécutera :

```
ALTER DATABASE Ventes SET RESTRICTED_USER WITH ROLLBACK IMMEDIATE
```

Toutes les transactions incomplètes seront annulées et les autres connexions à la base de données immédiatement déconnectées. Suivant le problème, il se peut que cette instruction soit un échec.

Le journal des transactions en cours sera sauvegardé avec l'option NO_TRUNCATE

```
BACKUP LOG Ventes TO savVentesLog WITH NO_TRUNCATE
```

III.2.2 Restaurer la base

La base de données et les journaux sont restaurés par l'instruction **RESTORE**. **RESTORE DATABASE** pour la base, **RESTORE LOG** pour les journaux.

Syntaxe simplifiée sous Transact-SQL :

```
RESTORE DATABASE nom_base
```

```
[ FROM unite_sauvegarde [ ,...n ] ]
```

```
[ WITH
```

```
    [ [ , ] FILE = { numero_fichier | @numero_fichier } ]
```

```
    [ [ , ] MEDIANAME = { nom_media | @variable_texte } ]
```

```
    [ [ , ] MEDIAPASSWORD = { motdepasse_media | @variable_texte } ]
```

```
    [ [ , ] MOVE 'logical_file_name' TO 'operating_system_file_name' ]
```

```
    [ ,...n ]
```

```
    [ [ , ] PASSWORD = { motdepasse | @variable_pwd } ]
```

```
    [ [ , ] { RECOVERY | NORECOVERY } ]
```

```
    [ [ , ] REPLACE ]
```

```
    [ [ , ] RESTART ]
```

```
    [ [ , ] RESTRICTED_USER ]
```

```
    [ [ , ] { REWIND | NOREWIND } ]
```

```
    [ [ , ] { STOPAT = { date_time | @date_time_var }
```

```
    | STOPATMARK = { 'mark_name' | 'lsn:lsn_number' }
```

```
        [ AFTER datetime ]
```

```
    | STOPBEFOREMARK = { 'mark_name' | 'lsn:lsn_number' }
```

```
        [ AFTER datetime ] } ]
```

```
    [ [ , ] { UNLOAD | NOUNLOAD } ]
```

```
]
```

```

RESTORE LOG nom_base
[ FROM unite_sauvegarde [ ,...n ] ]
[ WITH
    [ [ , ] FILE = { numero_fichier | @numero_fichier } ]
    [ [ , ] MEDIANAME = { nom_media | @variable_texte } ]
    [ [ , ] MEDIAPASSWORD = { motdepasse_media | @variable_texte } ]
    [ [ , ] MOVE 'logical_file_name' TO 'operating_system_file_name' ]
    [ ,...n ]
    [ [ , ] PASSWORD = { motdepasse | @variable_pwd } ]
    [ [ , ] { RECOVERY | NORECOVERY } ]
    [ [ , ] REPLACE ]
    [ [ , ] RESTART ]
    [ [ , ] RESTRICTED_USER ]
    [ [ , ] { REWIND | NOREWIND } ]
    [ [ , ] STATS [ = percentage ] ]
    [ [ , ] { STOPAT = { date_time | @date_time_var }
    | STOPATMARK = { 'mark_name' | 'lsn:lsn_number' }
    [ AFTER datetime ]
    | STOPBEFOREMARK = { 'mark_name' | 'lsn:lsn_number' }
    [ AFTER datetime ] } ]
    [ [ , ] { UNLOAD | NOUNLOAD } ]
]

```

FILE = { *numero_fichier* | @*numero_fichier* }

Identifie le jeu de sauvegardes à restaurer. Ainsi, la valeur **1** peut indiquer le premier jeu de sauvegarde sur le support, et la valeur **2** le second jeu.

MEDIANAME = { *nom_media* | @*variable_texte* }

Indique le nom du support.

MEDIAPASSWORD = { *motdepasse_media* | @*variable_texte* }

Fournit le mot de passe du support de sauvegarde.

MOVE '*logical_file_name*' TO '*operating_system_file_name*'

Indique que le fichier est transféré vers un autre emplacement que son emplacement d'origine.

RECOVERY | NORECOVERY

L'option NORECOVERY est à utiliser si d'autres restaurations doivent être appliquées. L'option RECOVERY sera utilisée pour la dernière restauration.

REPLACE

Indique que SQL Server doit créer la base de données spécifiée et ses fichiers même s'il en existe une autre de même nom. Lorsque l'option REPLACE n'est pas précisée, une vérification a lieu, pour éviter le remplacement accidentel d'une autre base de données.

RESTART

Indique la reprise d'une opération de restauration interrompue.

RESTRICTED_USER

Restreint l'accès à la base aux membres des rôles db_owner, dbcreator ou sysadmin

{ STOPAT | STOPATMARK | STOPBEFOREMARK }

Indique que la base de données doit être restaurée dans l'état qui était le sien à la date et à l'heure spécifiées ou jusqu'à une transaction marquée (*mark_name*) ou un numéro de séquence d'enregistrement (*lsn_number*) ou jusqu'à une transaction marquée (*mark_name*) ou un numéro de séquence de journal (*lsn_number*).

III.2.3 Exemples

Exemple 1: Restore complète de la base de données Ventes à partir de l'unité logique. savVentes.

```
RESTORE DATABASE Ventes FROM savVentes with NORECOVERY
```

Exemple 2: Restore de la sauvegarde différentielle Ventes à partir de l'unité logique. savVentesDiff.

```
RESTORE DATABASE Ventes FROM savVentesDiff with NORECOVERY
```

Exemple 3: Restore des journaux des transactions à partir de l'unité logique. savVentesLog.

```
RESTORE LOG Ventes FROM savVentesLog with FILE=1, NORECOVERY
```

```
RESTORE LOG Ventes FROM savVentesLog with FILE=2, RECOVERY
```

IV AUTOMATISATION DES SAUVEGARDES

SQL Server 2005 permet l'exécution de tâches planifiées, telles la sauvegarde de base de données ou l'import et l'export de données.

SQL Server peut envoyer et recevoir du courrier électronique en établissant une connexion avec un serveur de messagerie.

- Pour envoyer un message en cas de réussite ou d'échec d'un travail planifié
- En cas de déclenchement d'une alerte

IV.1 L'AGENT SQL SERVER

L'agent SQL Server permet de créer de travaux et d'en planifier leur exécution :

- Création de la définition du travail
- Découpage du travail en étapes
- Configuration de la planification de ce travail
- Gestion des messages de notification d'achèvement, succès ou échec.

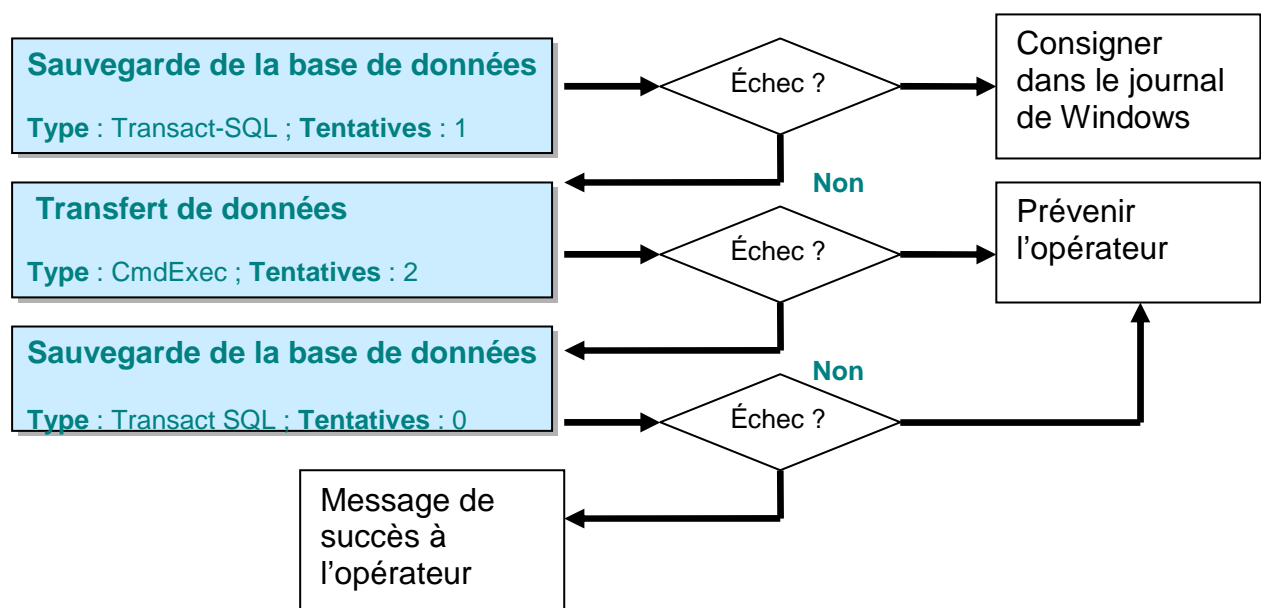
Une étape de travail peut être définie par :

- Un lot d'instructions Transact SQL
- Une commande du système d'exploitation (.exe, .bat, .com, .cmd ...)
- Des scripts Active X (langage VbScript, JScript)

Exemple : Un travail d'import de données peut être défini par :

- Une étape de sauvegarde de la base de données (type Transact SQL)
- Une étape d'import des données (Type Cmdexec)
- Une étape de sauvegarde de la base après transfert (type Transact SQL)

Chaque résultat d'étape peut être évalué (succès, échec) et un traitement particulier peut être associé.



Ce travail peut être planifié au lancement du Serveur SQL, lorsque les processus sont inactifs, ou périodiquement : on définira alors une périodicité (quotidienne, hebdomadaire ou mensuelle), la fréquence, les dates de début et de fin.

On pourra également paramétrer des opérateurs à notifier, soit par réseau, soit par radiomessagerie, soit par courrier électronique.

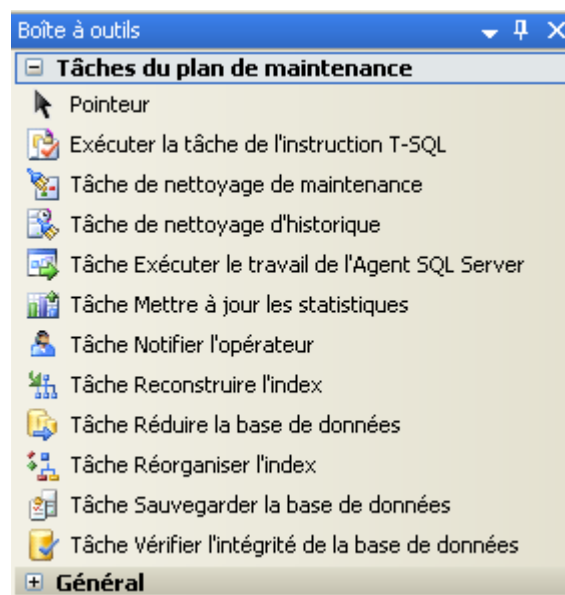
Des alertes peuvent également répondre à des erreurs consignées dans le journal de Windows.

IV.2 LE PLAN DE MAINTENANCE

SQL Server 2005 permet de bâtir un plan de maintenance des bases de données.

Sous l'onglet Gestion, en affichant le menu contextuel du dossier Plan de maintenance, vous pourrez définir un nouveau plan de maintenance, qui propose entre autre une tâche de sauvegarde de la base de données :

- soit manuellement et de manière graphique, à partir de la boîte à outils,
- soit en se laissant guider par l'assistant.



Etablissement référent

Marseille Saint Jérôme

Equipe de conception

Elisabeth Cattaneo

Remerciements :

Reproduction interdite

Article L 122-4 du code de la propriété intellectuelle.
« toute représentation ou reproduction intégrale ou partielle faite sans le
consentement de l'auteur ou de ses ayants droits ou ayants cause est
illicite. Il en est de même pour la traduction, l'adaptation ou la reproduction
par un art ou un procédé quelconques. »

Date de mise à jour 05/05/2008
afpa © Date de dépôt légal mai 08



**fpa / Direction de l'Ingénierie 13 place du Générale de Gaulle / 93108 Montreuil
Cedex
association nationale pour la formation professionnelle des
adultes
Ministère des Affaires sociales du Travail et de la
Solidarité**