



# Administration

SQL Server

2012

Sauvegarde/Restauration



# Sauvegarde/Restauration

---

- Les sauvegardes sont réalisées pour éviter la pertes de données suite à :
  - Une panne de support
  - Un CRASH disque
  - Des erreurs utilisateurs
- Les sauvegardes peuvent se faire « à chaud » pendant que les utilisateurs travaillent, par contre certaines opérations sont impossibles pendant la sauvegarde :
  - Création ou modification de la structure d'une base de données
  - Création d'index
  - Exécution d'opérations non journalisées, car le processus de sauvegarde utilise les journaux pour garantir la cohérence des données



# Sauvegarde/Restauration

---

- Les questions à se poser pour mettre en place une stratégie de sauvegardes sont :
  - Qu'elle est la taille de chaque base de données
    - Avec le volume de modification de chaque base
  - Quelle est la criticité de chaque base
    - Et ses conséquences
  - Combien de temps la base peut être arrêtée
  - Quant peut on planifier les sauvegardes
  - Combien de temps doit on conserver les sauvegardes
    - Combien de jeu de sauvegardes doit on conserver
    - Le server SQL est il en cluster ?



# Sauvegarde/Restauration

---

- La durée des sauvegardes dépend
  - De la taille des bases SQL
  - du support sur lequel les sauvegardes sont faites
    - Fichiers sur disque
    - Bandes (à condition que le lecteur de bande soit installé sur le Serveur SQL)
  - La stratégie de sauvegarde doit être une combinaison
    - de sauvegardes complètes de chaque base de données et la sauvegarde des logs (journaux de transaction)
      - Permet de fournir un point de départ pour les restaurations
    - Il est possible de mettre en place des sauvegardes incrémentielles (pour améliorer les temps de sauvegardes)



# Sauvegarde/Restauration

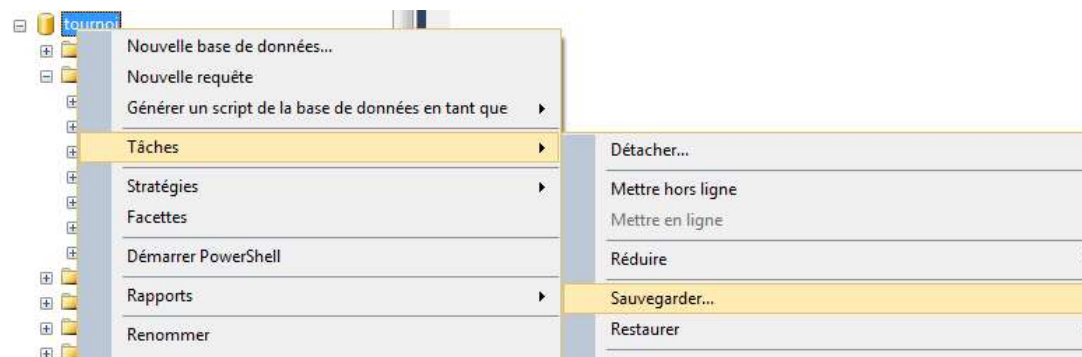
---

- Sauvegardes de copies
  - Cette sauvegarde ne rentre pas dans un plan de maintenance de sauvegardes mais compte comme un vraie sauvegarde
    - Il est possible de demander à SQLServer de vérifier la sauvegarde
  - 2 types de sauvegardes
    - Complète (propriété de la base)
    - Différentielle (extension : `.trn`)
- SQL Server utilise la base master pour les sauvegardes et les restauration



# Sauvegarde/Restauration

- Sauvegarder en via SQL Server Management Studio
  - Clic droit sur le nom de la base
  - Tâche+sauvegardes





# Sauvegarde/Restauration

Sauvegarder la base de données - MarketDev

Sélectionner une page  
Général  
Options

Script Aide

Source

Base de données : MarketDev

Mode de récupération : FULL

Type de sauvegarde : Complète

Sauvegarde de copie uniquement

Composant de sauvegarde :

Base de données

Fichiers et groupes de fichiers : [PRIMARY]

Jeu de sauvegarde

Nom : MarketDev-Complète Groupe de fichiers Sauvegarde

Description :

Expiration du jeu de sauvegarde :

Après : 0 jours

Le : 24/07/2014

Destination

Sauvegarde sur :  Disque  Bande

D:\backup\marketdev.bak

Ajouter...  
Supprimer  
Sommaire

OK Annuler

Connexion

Serveur : SQL3\PROD  
Connexion : SQLFORM\clotilde  
[Afficher les propriétés de connexion](#)

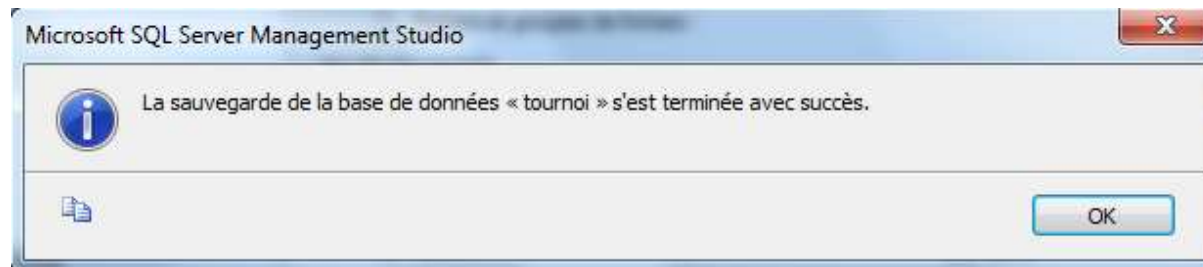
Progression

Prêt



# Sauvegarde/Restauration

```
BACKUP DATABASE [MarketDev] FILEGROUP = N'PRIMARY'  
TO DISK = N'D:\backup\marketdev.bak'  
WITH NOFORMAT, NOINIT,  
NAME = N'MarketDev-Complète Groupe de fichiers  
Sauvegarde',  
SKIP, NOREWIND, NOUNLOAD, STATS = 10  
GO
```







# Sauvegarde/Restauration

---

- Les modifications applicatives sont stockées dans les journaux (ldf)
  - 2 modes
    - Mode Simple : journal circulaire  
Si le journal est plein SQL l'épure au fur et à mesure des besoins  
le journal est vidé dès qu'il atteint sa taille maximum
    - Mode complet : fichier journal grandit tout le temps, SQL le vide au moment de la sauvegarde  
c'est la sauvegarde qui vide le journal de log



# Sauvegarde/Restauration

---

- Si les logs sont pleins et que le disque est plein
  - La base de données se bloque
- Solution
  - Démarrer SQL en mode simple
  - Clic droit + propriété + options
  - Passer en mode récupération simple
    - Arrêter la base de données
    - Redémarrer la base en mode simple
    - Faire un shrink du journal de log
    - Redémarrer la base en mode complet
    - Faire une sauvegarde



# Sauvegarde/Restauration

---

- SQL Server journalise en bloc
- Attention
  - Tout ce qui est insertions en blocs n'est pas écrit dans les logs
  - Attention aux chargements de données
- L'unité de sauvegarde
  - correspond à un emplacement de sauvegarde
  - Destination de la sauvegarde
    - » Il est possible de mettre un disque ou un répertoire distant partagé
    - » Il faut le faire à la main
      - `\\SQL1\partage\`



# Sauvegarde/Restauration

---

- Possibilité de compresser la sauvegarde (2008)
  - Dans les paramètres de la base de données
  - Ou le préciser dans la sauvegarde
- Plans de maintenances
  - Permet de faire la sauvegarde de toutes les bases du serveur SQL dans un fichier avec la date
  - Gère la suppression des anciennes sauvegardes automatiquement
- Les travaux sont stockés dans la base : msdb



# Sauvegarde/restauration

- -- liste et verification des sauvegardes effectuées
  - SELECT s.database\_name,
  - m.physical\_device\_name,
  - cast(s.backup\_size/1000000 as varchar(14))+ ' '+ 'MB' as bkSize,
  - CAST (DATEDIFF(second,s.backup\_start\_date , s.backup\_finish\_date)AS  
VARCHAR(4))+ ' '+ 'Seconds' TimeTaken,
  - s.backup\_start\_date,
  - CASE s.[type]
  - WHEN 'D' THEN 'Full'
  - WHEN 'I' THEN 'Differential'
  - WHEN 'L' THEN 'Transaction Log'
  - END as BackupType,
  - s.server\_name,
  - s.recovery\_model
  - FROM msdb.dbo.backupset s
  - inner join msdb.dbo.backupmediafamily m
  - ON s.media\_set\_id = m.media\_set\_id
  - WHERE s.backup\_start\_date > '18/03/2012'
  - --and s.type='D'
  - ORDER BY database\_name, backup\_start\_date desc, backup\_finish\_date
  - -- ceci est un commm
  - --restore headeronly
  - --from disk='c:\backup\temp.bak'



# Sauvegarde/Restauration

- La commande RESTORE

- Restore headeronly from disk='D:\\adv.back' ;  
go
  - Restore headeronly => Ce que contient la sauvegarde
- Restore filelistonly from disk='D:\\adv.back' ;  
go
  - Restore filelistonly => détaille BDD qui a été sauvegardée
- Restore labelonly from disk='D:\\adv.back' ;  
go
  - Restore labelonly => dit si la sauvegarde est faite avec SQL Server ou avec un autre outil



# Sauvegarde/Restauration

---

- Avant la restauration
  - Commencer par sauvegarder la fin du journal de transaction avant de restaurer
    - Pour retrouver les dernières transactions
  - Sauvegarder les logs avec l'option:
    - Sauvegarder la fin du journal de transactions et laisser la base en l'état de restauration
  - Déconnecter les utilisateurs avant de restaurer
    - On ne peut pas restaurer s'il y a des utilisateurs connectés à la base plantée
    - Vérifier s'il y a des utilisateurs connectés avec la commande
      - Sp Who => on regarde ceux qui utilisent la base plantée



# Sauvegarde/Restauration

---

- Restauration
  - Arrêter l'instance SQL
  - Recopier les fichiers '.mdf' à l'emplacement attendu
  - Redémarrer l'instance SQL
  - Supprimer la base cassée
    - Labs 22462A\_07\_PRJ
    - `Select top 1000 restore_history_id`
    - `Desc restorehistory`
    - `Sp_columns [restorehistory]`
    - `Select * from sys.columns`
    - `Select * from sys.tables`





# Sauvegarde/Restauration

- Exemple de restauration d'un fichier « .mdf »
  - -- restauration de la base à un autre emplacement
  - -- le backup log est à supprimé car déjà fait
  - --
  - USE [master]
  
  - -- sauvegarde des journaux de log avant restauration
  - -- supprimer l'option NORECOVERY car il y a des sessions utilisateur connectees
  - --
  - --BACKUP LOG [MarketDev] TO DISK =  
N'D:\backup\marketdev.bak' WITH NO\_TRUNCATE ,
  - --NOFORMAT, NOINIT,
  - --NAME = N'MarketDev-Journal des transactions  
Sauvegarde',
  - --SKIP, NOREWIND, NOUNLOAD, STATS = 10
  - --GO



# Sauvegarde/Restauration

```
- -- backup log réussi précédemment
- use master
- go
- sp_who
- -- mettre use master avant chaque commande de
  restauration SQL
- use master
- Alter database [MarketDev] set single_user with
  rollback immediate
- GO
- -- restauration lere base complete effectuée
- use master
- RESTORE DATABASE [MarketDev] FROM DISK =
  N'D:\backup\marketdev.bak' WITH FILE = 2,
- MOVE N'MarketDev' TO
  N'D:\datasql\marketdev.mdf',
- NORECOVERY, NOUNLOAD, REPLACE, STATS = 5
```



# Sauvegarde/Restauration

- -- restauration sauvegarde différentielle
- RESTORE DATABASE [MarketDev] FROM DISK =  
N'D:\backup\marketdev.bak' WITH FILE = 4,
- NORECOVERY, NOUNLOAD, STATS = 10
- -- restauration des logs faites avant le crash
- -- mettre l'option NORECOVERY car il y a encore des  
restauration à faire après celle la
- -- si on met RECOVERY on doit tout recommencer
- RESTORE LOG [MarketDev] FROM DISK = N'D:\backup\marketdev.bak'  
WITH FILE = 5,
- NOUNLOAD, NORECOVERY, STATS = 5
- RESTORE LOG [MarketDev] FROM DISK = N'D:\backup\marketdev.bak'  
WITH FILE = 6,
- NOUNLOAD, NORECOVERY, STATS = 5
- RESTORE DATABASE [MarketDev] with RECOVERY;
- Alter database [MarketDev] set multi\_user ;
- GO



# Sauvegarde/Restauration

- Exemple de restauration d'un fichier « .ldf »
  - -- Perte du fichier .ldf
  - --
  - --arreter l'instance SQL
  - --recopier les fichiers .mdf
  - --redémarrer l'instance SQL
  - --supprimer la base cassée
  - --creer une base du meme nom que la base cassée.....
  - --CREATE DATABASE marketdev2 ON
  - --( NAME = N'marketdev',FILENAME =  
N'd:\Datasql\marketdev2.mdf' )
  - --LOG ON
  - --( NAME = N'marketdev\_log',FILENAME =  
N'c:\logsql\marketdev2.ldf' )
  - --arret sql
  - --remplacer le mdf
  - --demarrer SQL



# Sauvegarde/Restauration

- Exemple de restauration d'un fichier « .ldf »
  - -- emergency previent la base qu'elle est en état d'urgence exceptionnelle et en recuperation
  - alter database marketdev2 set emergency;
  - -- mettre la base en acces simple user
  - alter database marketdev2 set single\_user with rollback immediate;
  - -- prévenir que les fichiers ne sont pas cohérents
  - -- il verifie et essaie de reparer en autorisant une perte de données
  - DBCC CHECKDB (N'marketdev2', REPAIR\_ALLOW\_DATA\_LOSS)
  - --mettre la base en multi user
  - alter database marketdev2 set multi\_user;



# Sauvegarde/Restauration

- Exemple de restauration d'un fichier « .ldf »
  - -- change le maniere de fonctionner de sqlserver change les option
  - -- allow-update met à jour les bases en mode emergency et récupération
  - -- pas besoin de reconfigure après
  - sp\_configure 'allow updates', 1 reconfigure with override ;
  
  - --arret sql
  - --suppr log
  - -- redemarrer SQL
  
  - alter database marketdev2 set multi\_user ;
  - alter database marketdev2 set emergency ;
  - alter database marketdev2 set single\_user ;



# Sauvegarde/Restauration

- Exemple de restauration d'un fichier « .ldf »
  - -- après la commande ci-dessous on a le message
  - --Avertissement : le journal de la base de données 'marketdev2' a été reconstruit.
  - --La cohérence transactionnelle a été perdue.
  - --La chaîne RESTORE a été rompue et le serveur n'a plus de contexte sur les fichiers journaux précédents,
  - --aussi devez-vous connaître leur nature. Vous devez exécuter DBCC CHECKDB pour valider la cohérence physique.
  - --La base de données a été mise en mode dbo uniquement.
  - --Au moment de mettre la base de données à la disposition des utilisateurs, vous devrez rétablir
  - --ses options et supprimer tous les fichiers journaux en surplus.
  - alter database marketdev2 rebuild log on (name='marketdev\_log',filename='c:\logsql\marketdev2.ldf') ;
  - alter database marketdev2 set multi\_user ;



# Sauvegarde/Restauration

---

- Exemple de restauration d'un fichier « .ldf »
  - -- pas besoin de reconfigure après
  - `sp_configure 'allow updates', 0` reconfigure with override ;
  
  - redémarrer l'agent SQL (clic droit sur agent SQL Server + redémarrer)
  - redémarrer l'agent SQL (clic droit sur agent SQL Server + redémarrer)
  - L'agent dépend de SQL serveur