



# Administration

SQL Server

2012

Databases



# Instance et base de données

---

- Le noyau SQL Server correspond à
  - 1 instance SQL Server
  - Ports 1433 et 1434 utilisés par défaut à l'installation
- L'instance SQL Server peut contenir plusieurs bases de données
  - Contenant elles même des objets (tables, index)
- L'Instance par défaut porte le nom du serveur
- A l'installation le mieux est de séparer sur des disques différents
  - Les bases systèmes et
  - Les bases utilisateurs



# Instance et base de données

---

- Notion de cluster SQL
  - Forcément actif/passif
- Sur un cluster de 2 nœuds on peut avoir :
  - Nœud 1 avec
    - SQL/Prod    actif
  - Nœud 2 avec
    - SQL/Test    actif
    - SQL/Prod    passif



# Instance et base de données

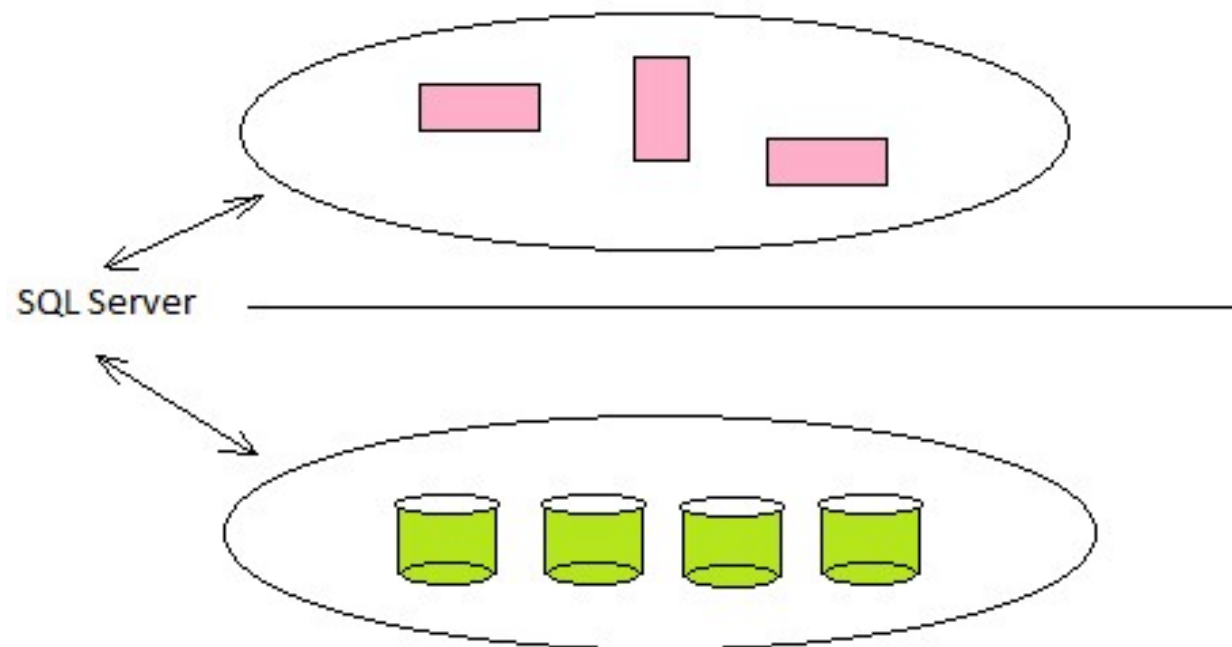
---

- La base de données est un ensemble logique d'espace de stockage permettant de stocker des objets tels que les tables sans avoir à se soucier du stockage physique
  - SQL Server permet de réaliser des associations entre les fichiers physiques et la base de données
  - Lors de la création d'une base il faut préciser au moins 2 fichiers
    - 1 pour les données
    - 1 pour le journal des transactions afin de conserver l'image avant et après modifications



# Instance et base de données

- Séparation entre les structures physiques et logiques





# Instance et base de données

---

- Fichiers journaux
  - Ils permettent de stocker l'image avant et après modification des données
    - Seules les opérations DML provoquent une journalisation (INSERT, UPDATE, DELETE)
    - Les opérations importantes comme la création d'index sont stockées dans le journal
  - Il sera utilisé principalement dans les opérations de restauration ou les opérations de sauvegardes incrémentales ou lors de répliquions de base de données
    - Il garantit la cohérence des données , c'est à dire que toutes les transactions validées persistent même après un gros problème serveur



# Instance et base de données

---

- Les fichiers journaux (suite)
  - D'une façon logique toutes les informations sont enregistrées les unes à la suite des autres dans le journal
  - Chaque information est identifiées par son LSN (Log Séquence Number)
    - numéro séquentiel d'enregistrement
  - Chaque enregistrement contient l'identifiant de la transaction à l'origine de la modification des données
    - Tous les enregistrements d'une même transaction ne sont pas enregistrés de façon contigüe



# Instance et base de données

---

- Fichiers de données
  - Ils contiennent l'ensemble des données de la base de données
    - Les lignes de tables
  - Chaque fichier de données ne contiennent que les données d'une seule base de données
    - Il y a spécialisation des fichiers par rapport à la base de données
    - Ils sont structurés pour répondre de manière optimum aux sollicitations du moteur SQL
    - Ils permettent de stocker le plus de données possibles en optimisant l'espace disque utilisé





# Instance et base de données

---

- Notion de pages
  - SQL Server va structurer le fichier en blocs de 8K
    - Appelés pages
  - La page est l'unité de travail de SQL Server
    - Cette page est chargée (en entier) en mémoire
  - Une page peut contenir plusieurs ligne de tables ou d'index
    - Afin d'éviter la fragmentation une ligne doit toujours être contenue dans 1 bloc
    - Sa taille maximale est de 8060 octets



# Instance et base de données

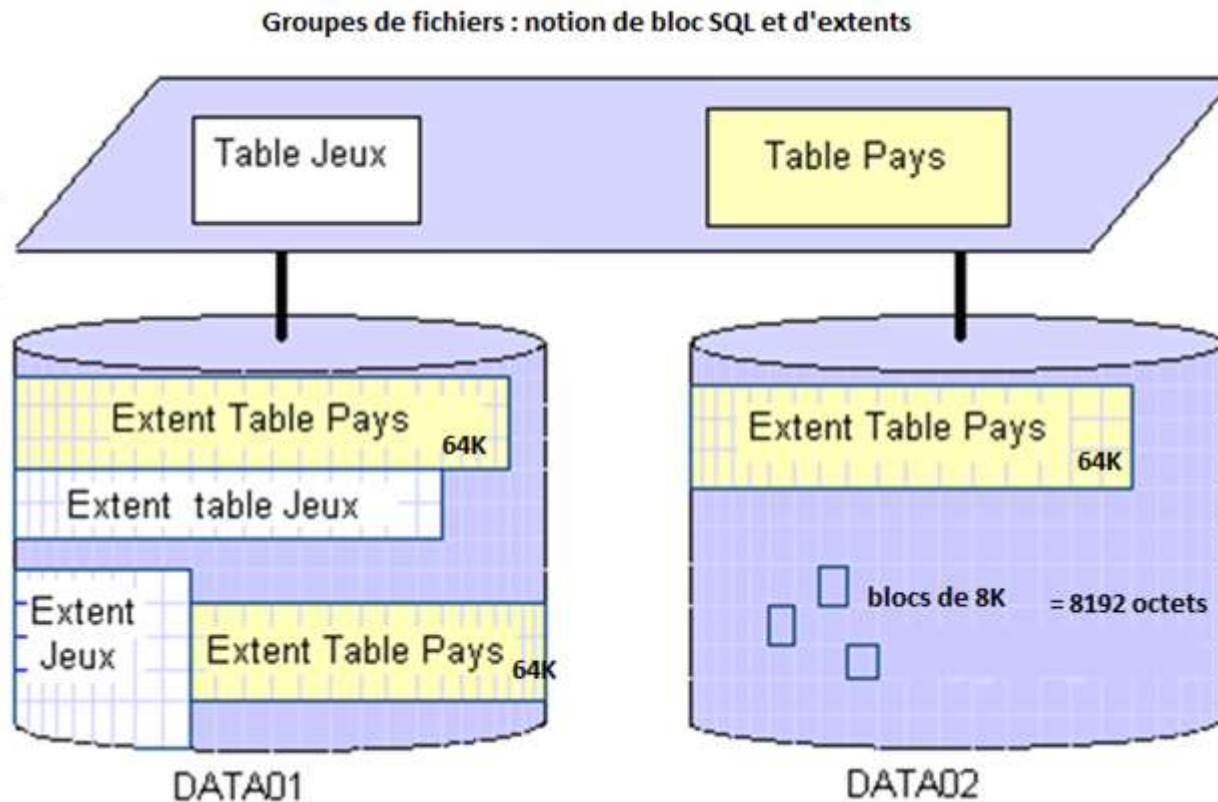
---

- Les fichiers de données
  - Sont découpés en blocs (pages)
- Les extensions sont des regroupements logiques de 8 blocs contigus
  - Elles ont une taille de 64Ko
  - Leur rôle est d'éviter une dispersion des données pour un même objet dans un fichier
  - 2 types d'extensions
    - Mixtes : contiennent des objets dont la taille est inférieure à 8 blocs
    - Uniformes : contiennent des objets de plus de 64K, elles sont spécialisées pour 1 seul objet



# Instance et base de données

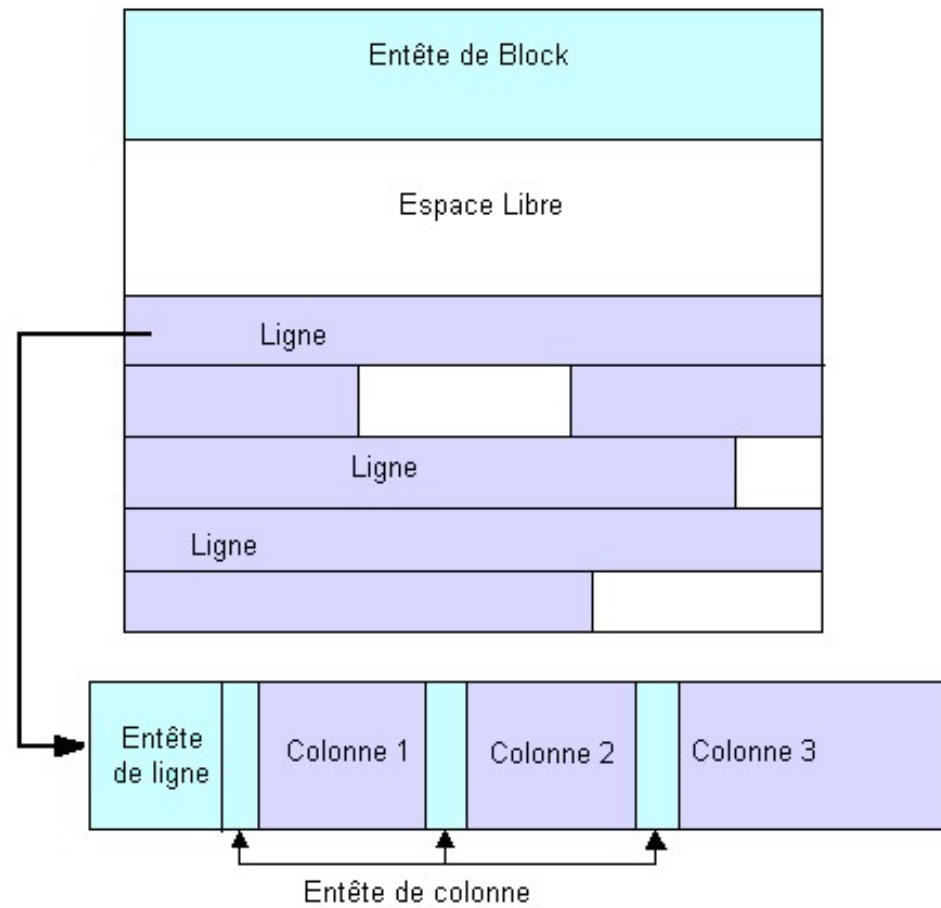
- Notion de bloc, d'extent et de segment





# Instance et base de données

- Les blocs



*Organisation du stockage dans les Blocks*



# Instance et base de données

---

- Extensions de fichiers
  - Les fichiers journaux contiennent l'extension « .ldf »
  - Les fichiers de données contiennent l'extensions :
    - Le premier fichier : « .mdf »
    - Les suivants : « .ndf »
  - Ces extensions sont recommandées mais pas obligatoires



# Instance et base de données

---

- Conseils à l'installation
  - Serveur dédié
  - Mettre le système + (.ldf) en Raid 1
  - Mettre les Datas (.mdf) en Raid 10
    - Si beaucoup de lecture et peu d'écriture mettre du Raid 5
  - Mettre la taille de la base de données à une taille correspondant à un multiple de la taille du block (64K)



# Instance et base de données

---

- Configuration proposée

Windows

A l'installation SQL :

RAID

- .ldf (log contenant les transactions)
- base system (dictionnaire de données)
- tempDB (1 par coeur ; on peut aller jusqu'à 8 tempDB)

RAID

Datas (bases de données) :

- .mdf (1er fichier)
- .mdf (2eme fichier)
- .mdf (etc ...)

*Configuration de 2 disques*



# Instance et base de données

---

- Bases Système créées par défaut
- La base Master
  - Contient la structure de toutes les bases de données
  - Si cette base est altérée aucune base ne démarre en mode simple
    - À sauvegarder
  - Au cas où
    - on peut démarrer en mode mono-utilisateur





# Instance et base de données

---

- Bases Système créées par défaut(suite)
- La base model
  - Toute base créée est une copie de la base model
  - Peut être prèt configurée avant de créer de nouvelles bases de données
  - Si elle n'est pas présente SQL Server ne démarre pas



# Instance et base de données

---

- Bases Système créées par défaut(suite)
- La base tempDB
  - Contient les données temporaires
  - Tronquée ou recrée à chaque démarrage du server SQL
  - On ne la sauvegarde pas
- La base msdb
  - Contient les plans de maintenance et travaux
- La base resource
  - Contient le dictionnaire de données
  - Recopiée en partie dans chaque base lors de sa création
  - Non visible dans SQL Server



# Instance et base de données

- Conseils à la création d'une base de données
  - Le propriétaire est le user qui crée la base
    - Si pas de propriétaire , le propriétaire par défaut sera « dbo »
    - En général on garde le schéma par défaut « dbo » pour des facilité de développement
  - Avec un jeu de caractères
    - Par défaut c'est le jeu de caractère de l'instance choisi à l'installation
  - Et un type de classement (tri)
    - Important lors d'une migration il doit être identique
  - La fonction COLLATE
    - Permet de changer de classement en cours d'un SELECT dans la clause WHERE
    - Exemple :
    - Where CIData = CSData collate Latin 1\_general\_CI\_AS ;



# Instance et base de données

---

- Conseils à la création d'une base de données
  - Limiter la taille des fichiers
    - Peut atteindre 16 Teras sous Windows avec un maximum de 32 fichiers pour une base SQL Server
  - Fichiers de la base de données
    - .mdf (fichier primaire ; contenant les données)
    - .ndf (fichier secondaire ; contenant les données)
    - .ldf (fichiers logs contenant les transactions )
  - Emplacement par défaut des fichiers visibles dans SQL Managment Studio
    - Clic droit sur le nom de BDD + options



# Instance et base de données

---

- Conseils à la création d'une base de données
  - Base TempDB
    - On a 1 Base TempDB pour 1 instance SQL Server
    - Créer 1 fichier par cœur de processeur (Max 8)
      - Clic droit sur tempDB + propriété
  - Contient les segments temporaires de l'instance SQL Serveur
    - SQL les supprime au fur et à mesure des besoins
    - TempDB remis à zéro au redémarrage du serveur SQL
      - Remet les fichiers à leur taille initiale
      - On peut également réduire la taille de la base TempDB sans redémarrage du server SQL avec les commandes SHRINKDATABASE ou SHRINKFILE



# Instance et base de données

---

- Conseils à la création d'une base de données
  - Fichiers de log (.ldf)
    - Contiennent les transactions
    - Il faut les sauvegarder régulièrement
    - Ne se vident qu'au moment de la sauvegarde
      - A condition qu'aucune transaction ne soit en cours
  - 2 modes
    - Mode Simple : journal circulaire
    - Mode complet : fichier journal grandit tout le temps, se vide au moment de la sauvegarde



# Instance et base de données

---

- Conseils à la création d'une base de données
  - Après création faire une sauvegarde de la base complète
    - Clic droit (sur BDD) + tache
      - + sauvegarde
      - + complète
  - Puis faire une sauvegarde du journal de log
    - Clic droit (sur BDD) + tache
      - + sauvegarde
      - + journal de log



# Instance et base de données

- Créer une base de données

- CREATE DATABASE nombase  
[ON  
                  [PRIMARY] [ <spécificationfichier> [ ,n]]  
                  [LOG ON]    <spécificationfichier> [ ,n]]  
]  
[ COLLATE classement]  
[;]
- Avec <spécificationfichier>
  - nom logique fichier
  - + \chemin\nomfichier
  - size [KB|MB|GB|TB]
  - MAXSIZE [KB|MB|GB|TB| unlimited] → (taille maximum)
  - FILEGROWTH [KB|MB|GB|TB| %] → (taille de l'increment)
  - ) [ ,n]
- Dans SQL Managment Studio
  - Clic droit sur base de données + Créer



# Exercice



- Utiliser Management studio pour créer une base de données appelée
  - commandes



# Instance et base de données

---

- Gérer l'espace de stockage d'une base de données
  - Concerne la gestion de l'espace disque utilisé par les fichiers de la base de données
  - Il y a un maximum de
    - 32767 groupes de fichiers
    - 32767 fichiers par groupes de fichiers
  - Si le taux d'accroissement est omis, il est de :
    - 10% pour les journaux
    - 1Mo pour les fichiers de données, multiple de 64Ko
    - En utilisant des fichiers à croissance dynamique le serveur ne sera jamais bloqué sauf saturation du disque



# Instance et base de données

- Modifier un fichier

- La commande ALTER DATABASE permet de modifier la taille d'un fichier :

- De données

- Du journal de transaction

- ALTER DATABASE nombase

- MODIFY FILE (spécificationFichier) [ ; ]

- avec NAME = nom logique

- NEWNAME = nouveau nom logique

- FILENAME = chemin et nom fichier

- size [KB|MB|GB|TB]

- MAXSIZE [KB|MB|GB|TB| unlimited] → (taille maximum)

- FILEGROWTH [KB|MB|GB|TB|%] → (taille de l'increment)

- ) [ ,n]



# Instance et base de données

- Ajouter un fichier de données
  - ALTER DATABASE nombase  
ADD FILE spécificationFichier [ ;]
  - Avec <spécificationfichier>
    - nom logique fichier
    - + \chemin\nomfichier
    - seize [KB|MB|GB|TB]
    - MAXSIZE [KB|MB|GB|TB| unlimited] → (taille maximum)
    - FILEGROWTH [KB|MB|GB|TB| %] → (taille de l'increment)
    - ) [,n]



# Instance et base de données

---

- Fichiers du journal
  - Comme pour les fichiers de données il est possible de redimensionner le fichier journal, ou d'ajouter un fichier au journal de transactions.
    - L'ajout nécessite l'utilisation de l'option « ADD LOGFILE »
  - Les manipulations de fichiers peuvent se faire avec
    - SQL Server Management Studio



# Instance et base de données

---

- Libérer de l'espace disque utilisé par la base de données
  - Lorsque les tables sont vidées de leur données avec les commandes DELETE ou TRUNCATE, les extensions occupées par les tables ou les index sont alors libérées
    - Mais la taille des fichiers n'est pas réduite
    - Faire un shrink de la base ou des fichiers pour réduire leur taille
    - Ou faire une sauvegarde FULL qui va faciliter le shrink, puis faire le shrink
      - La sauvegarde va valider les transactions et réorganiser les pages des fichiers de la base



# Instance et base de données

- La réduction des fichiers se fait à l'aide des commande DBCC:
  - SHRINKDATABASE  
permet de compacter l'ensemble des fichiers de la base, pour les données les extensions seront stockées en haut du fichier de façon contigues; pour les journaux SQL Server essaie de rendre aux fichiers une taille aussi proche que celle de la taille cible lorsqu'ils sont tronqués
  - SHRINKFILE  
permet de réduire la taille des fichiers, fichiers par fichier, et permet de préciser la taille finale souhaitée.
  - Avant de réaliser une opération de compactage de base il faut faire une sauvegarde de la base de données et de la base Master
    - Ces opérations s'exécutent en mode différé !



# Instance et base de données

- Syntaxe

- DBCC SHRINKDATABASE

- {nombase|id\_base|0}

- [ , pourcentage\_cible]

- [ , {NOTRUNCATE|TRUNCATEONLY} ] )

- Id\_base : identifiant de la base, peut être connu en utilisant la fonction « db\_id() », ou la vue sys.databases depuis la base Master
      - 0 : permet de préciser la base courante
      - Pourcentage\_cible : permet de préciser un pourcentage d'espace libre souhaité dans le fichier après compactage





# Instance et base de données

- **Syntaxe**

- `DBCC SHRINKFILE ([nom_fichier|id_fichier]`  
`{ [ [ ,taille_cible]`  
`[ , {NOTRUNCATE|TRUNCATEONLY| } ] ] |EMPTYFILE}`

- `Taille_cible` : permet de préciser la taille finale souhaitée, en méga octet s sou forme de nombre entier. Si aucune taille n'est précisée le fichier est réduit à son maximum.
    - `Emptyfile` : réalise la migration des données contenues dans le fichier vers les autres fichiers du même groupe, Il sera possible de supprimer ce fichier qui ne sera plus utilisé par SQL Server (avec `ALTER DATABASE`)
    - `NOTRUNCATE` : permet de ne pas rendre au système d'exploitation l'espace libre obtenu après compactage. Par défaut l'espace est libéré
    - `TRUNCATEONLY` : permet de libérer l'espace inutilisé par le fichier après compactage des données, aucune réorganisation des données n'est effectuée



# Instance et base de données

---

- Configuration de bases de données
  - On configure chaque base indépendamment les unes des autres
    - via l'instruction `ALTER DATABASE set options [;]`
    - Ou SQL Server Management Studio
  - Si toutes les bases ont les mêmes options il est préférable de configurer la base Model
    - Ainsi toutes les nouvelles bases fonctionneront avec les nouvelles options de la base Model



# Instance et base de données

---

- Options de configuration
  - `AUTO_SHRINK {ON|OFF}`  
si activée, les fichiers sont réduits dès qu'ils disposent de plus de 25% d'espace libre, désactiver cette option car elle conduit à de nombreuses opérations de réduction de fichiers et de perte de performance
  - `READ_ONLY`  
positionne la base en lecture
  - `READ_WRITE`  
positionne la base en lecture, écriture
  - `SINGLE_USER`  
seul un utilisateur peut travailler sur la base
  - `RESTRICTED_USER`  
seuls les utilisateurs membres des rôles `db_owner`, `db_creator` et `sysadmin` peuvent se connecter à la base



# Instance et base de données

---

- Options de configuration (suite)
  - Multi\_user  
mode de fonctionnement standard de la base de données
  - Auto\_create\_statistics {ON|OFF}  
si l'option est positionnée, les statistiques manquantes lors de l'exécution de la requête sont calculées automatiquement
  - AUTO\_UPDATE\_STATISTICS {ON|OFF}  
si l'option est positionnée, les statistiques obsolètes sont calculées de façon automatique
- Dans SQL Server Management Studio les options de configuration sont visible dans :  
Propriété du menu contextuel de la base de données sélectionnée



# Instance et base de données

---

- Options via Transac SQL
  - Databasepropertyex  
permet de connaître la valeur d'un paramètre :  
`SELECT databasepropertyes ('testSQL', 'userAccess') ;`
  - Sp\_helpdb  
si aucun paramètre n'est passé cette procédure permet de connaître l'ensemble des bases qui existent sur le serveur, avec la taille, le propriétaire, l'identificateur, la date de création et les options  
cette procédure utilise la table : `sys.databases`
  - sp\_spaceused (nom\_objet)  
permet de connaître l'espace de stockage par une base ou un fichier log ou des objets de la base  
`use Gescom ;`  
`go`  
`exec sp_spaceused 'Clients'`



# Instance et base de données

---

- Rapports d'utilisation de l'espace disque
  - Dans SQL Server Management Studio, il est possible d'obtenir un rapport sur l'utilisation de l'espace disque par les différentes tables de la base sélectionnée  
clic droit sur Base + Rapport + Rapports standards
  - 3 choix sont disponibles :
    - Rapport global sur l'espace disque utilisé
    - Rapport détaillé de la consommation d'espace disque pour chaque table
    - Rapport ne détaillant que les principales tables



# Instance et base de données

---

- Supprimer une base de données
  - La suppression d'une base de données a pour effet de supprimer tous les fichiers de cette base et leurs données
    - Se fait par : `DROP DATABASE nom_base ;`
  - Cette opération est irréversible ; prévoir une sauvegarde
    - La base ne doit pas être en cours d'utilisation
    - Après suppression chaque connexion qui utilisait cette base par défaut se retrouve sans base par défaut
    - **Prévoir une sauvegarde de la base Master après suppression d'une base de données afin de pouvoir restaurer le serveur SQL si besoin**



# Instance et base de données

---

- Bases de données à relation contenant-contenu
  - SQL Server 2012 propose ce type de bases  
option : **CONTAINMENT**
    - `Create database nombase CONTAINMENT={NONE | PARTIAL}`
  - L'objectif consiste à les rendre indépendantes du serveur sur lequel elles sont créées
    - Permet de transférer facilement la base d'un serveur à un autre
    - Possible car de nombreuses méta-données sont stockées dans la base de données en plus des bases de données systèmes, ainsi les utilisateurs de la base continuent d'exister après transfert





# Instance et base de données

---

- Il est possible de détacher puis d'attacher une base de données pour la changer de serveur  
Dans SQL Management Studio
  - Ne pas oublier de cocher la case permettant la déconnexion des utilisateurs
  - Il faut être en mode administrateur
  - Attention a bien créer les répertoires d'emplacement de la base de données