Memo SQL Server

*Par Jean-Baptiste Cochinard*

Table des matières

[Les type de données SQL server 2](#_Toc69467297)

[DDL : CREATE ALTER DROP 3](#_Toc69467298)

[TABLE 3](#_Toc69467299)

[Contraintes 4](#_Toc69467300)

[La gestion des index 4](#_Toc69467301)

[DML : INSERT UPDATE DELETE 5](#_Toc69467302)

[Transac SQL 5](#_Toc69467303)

[Opérateurs et fonctions Transac SQL 5](#_Toc69467304)

[Instruction DML 5](#_Toc69467305)

[Gestion des transaction 5](#_Toc69467306)

[L’algebre relationnelle 6](#_Toc69467307)

[DML : Select 6](#_Toc69467308)

[Sous requête, table CTE et VIEW 8](#_Toc69467309)

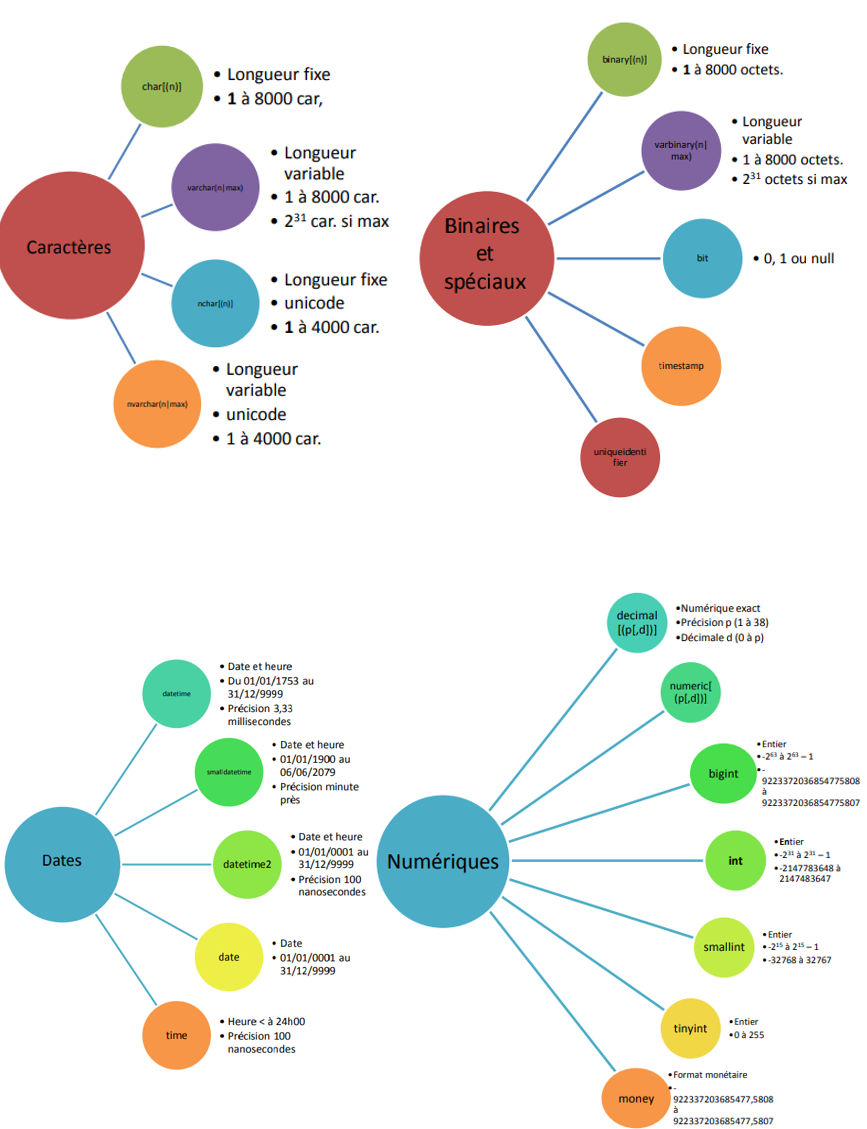
[Sous-requete 8](#_Toc69467310)

[Tables CTE 8](#_Toc69467311)

[Objet View 9](#_Toc69467312)

[Opérateur enssembliste 9](#_Toc69467313)

# Les type de données SQL server

Exemple : DECIMAL(5,2) permet un nombre à 5 chiffres dont l’espace après la virgule est de 2

# DDL : CREATE ALTER DROP

## TABLE

|  |  |
| --- | --- |
| **USE** |  |
| **CREATE**  **TABLE** |  |
| **ALTER**  **TABLE** |  |
| **DROP**  **TABLE** |  |

## Contraintes

|  |  |
| --- | --- |
| **CHECK**  **DEFAULT** |  |
| **KEY** |  |
| **IDENTITY** |  |

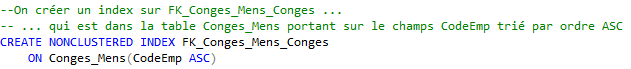
*NOTE : une clé UNIQUE veut dire qu’une seule valeur est possible par table*

## La gestion des index

Les index permettent une recherche plus rapide lors de l’interrogation des bases de données. Il ne faut pas en créer de trop sinon ça alourdira le processus, il faut viser les colonnes avec le plus de recherches.

Un index est automatiquement créé à la déclaration de la **PRIMARY KEY** en **CLUSTERED** et en en **NONCLUSTERED** sur les clé **UNIQUE.** Une bonne pratique de définir un index sur les **FOREIGN KEY**

* Le type **CLUSTERED** est un type unique, il va organiser physiquement l’ordre de la table, elle est définie sur la clé primaire par défaut.
* Le type **NONCLUSTERED** peut être définie sur plusieurs colonne et permet d’accélérer les recherches portant sur celle-ci.



# DML : INSERT UPDATE DELETE

## Transac SQL

**Transac SQL** Est un langage procédural associé à la base de données Microsoft SQL Server, celui-ci n’appartient pas à la norme et n’est donc pas compatible avec les autres gestionnaires de BDD.

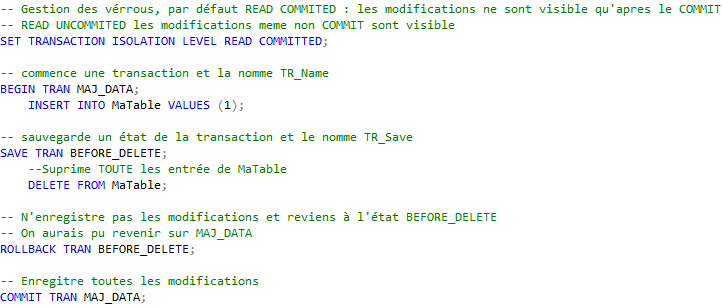
## Opérateurs et fonctions Transac SQL

|  |  |
| --- | --- |
|  |  |

## Instruction DML

|  |  |
| --- | --- |
| **INSERT INTO … VALUES** |  |
| **UPDATE… SET** |  |
| **DELETE FROM** |  |

## Gestion des transaction

Les transactions permettent d’améliorer les performances, assurer l’intégrité des données, tester les effets des MAJ et de gérer les verrous sur les lignes (aucun autre utilisateur ne peut modifier les ligne impacter par la transaction en cours). Si il y as la moindre erreur lors de l’enregistrement, le COMMIT ne sera pas pris en compte.

# L’algebre relationnelle

L’algèbre relationnelle est une sorte d’algorithme pour les bases de données, les différentes opérations disponibles pour parvenir au résultat que l’ont souhaite on dispose de plusieurs outils :

* Restriction : permet de restreindre les résultats d’une requête avec des condition **WHERE,** c’est la première opération à mettre en place pour optimiser notre requête (faire les jointures après etc.)
* Projection : permet de ne sélectionner que quelque colonne d’une table
* Calcul associé : permet de faire des calcule (agrégat etc.) et de créer des nouvelles colonnes avec ces calculs
* Produit cartésien : relation entre 2 tables contenant l’ensemble des attributs des table (**CROSS JOIN**)
* Jointure : idem qu’un produit cartésien mais on exprime une condition (**INNER JOIN** et **WHERE**)
* Union, intersection, différence et division : Ces 4 opérateurs permettent de mettre en relation 2 structures équivalentes (même nombre de colonne et même type). **UNION** récupère l’intégralité des 2 ensembles, **INTERSECTION** ne récupère que les éléments communs, **DIFFERENCE** récupère les enregistrements du premier ensemble qui n’est pas dans le deuxième ensembles et **DIVISION** n’a pas d’implémentation en SQL

# DML : Select

|  |  |
| --- | --- |
| **Ordre** |  |
| **Projection** | *Un alias n’est pas utilisable, on ne peut pas faire de WHERE dessus par exemple*  *Si ont as un espace dans le nom après AS ont peut utiliser des crochets : [Le Nom]* |
| **WHERE** | *Tips : utiliser* ***LIKE*** *‘%D’ pour rechercher un mot finissants par D, ‘D%’ = commence,* ***NOT LIKE*** *= inverse* |
| **INTO** |  |
| **Agrégat** | *Tip :* ***COALESCE****(expression, defaullt) nous renvois la valeur dans default si expression = NULL* |
| **ORDER BY** |  |
| **CROSS JOIN** |  |
| **INNER JOIN** |  |
| **OUTER JOIN** |  |

# Sous requête, table CTE et VIEW

## Sous-requete

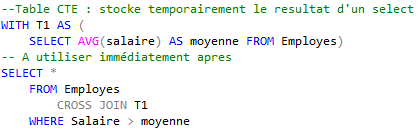
Imbriquer une requête **SELECT** dans une autre requête SELECT, INSERT, UPDATE ou DELETE est plus performante que dans une table temporaire, à utiliser donc si on privilégie la performance à la lisibilité

|  |  |
| --- | --- |
| **DELETE**  **UPDATE**  **INSERT**  **INTO**  **SELECT** |  |

## Tables CTE

Simplifie l’écriture et plus performant que les tables temporaires, mais elle doit être utiliser immédiatement contrairement aux tables temporaires.

Contrainte : on ne peut pas utiliser de clauses **INTO**, **ORDER BY** dans le **SELECT** de la table CTE.



## Objet View

L’objet **VIEW** permettent de stocker des requêtes SELECT, *mais pas ses données*. L’objet **VIEW** est utilisé comme une table. Il permet de masquer la complexité d’écriture d’une requête, de sécurisé l’accès à certaines données pour certain profil etc.

*Comme pour les tables CTE, il n’est pas possible d’utiliser de clause* ***INTO*** *et* ***ORDER BY****.*

|  |  |
| --- | --- |
| **Création**  **Manipulation**  **Suppression** |  |

# Opérateur enssembliste

Les opérateurs ensemblistes peuvent parfois simplifier des requêtes complexes, ils est le plus souvent traduisible avec les tables temporaire et les CTE, mais il est à noté toutefois son existence, dont l’utilisation peut être pertinente dans certain cas.

* **UNION**, Unir un ensemble de lignes provenant de plusieurs requêtes
* **EXCEPT**, Extrait un ensemble de lignes présentes dans un jeu de résultat qui ne sont pas dans un autre
* **INTERSECT**, extrait un ensemble de lignes présentes simultanément dans deux jeux de résultats