

# Programmation de bases de données: Introduction

## Plan de la séance:

- Retour sur la dernière séance:
  - Point de vue des enseignants.
  - Point de vue des étudiants.
- Où sont stockés les fichiers de la base de données ?
- L'interface SQL Server Management Studio
- Bref rappels SQL
- La propriété IDENTITY
- Début du Laboratoire 1

## Retour sur la dernière séance : Point de vue des enseignants:

- Nous avons installé deux logiciels:
  - SQL Server qui représente le serveur de bases de données de Microsoft. C'est un serveur (comme Oracle)
  - Pour exploiter un serveur de bases de données, nous avons besoin d'une interface conviviale ce qu'offre le SQL Server Management Studio. C'est le client (Comme SQL Développeur).
- Les accès à la base de données doivent être sécurisés et avec des privilèges restreints d'où la création et la configuration de connexion.
- Il y a tout un chapitre sur la sécurité des données plus loin dans la session.

## Où sont stockés les fichiers de la bases de données ?

- A ne pas confondre: Fichiers de la base de données et fichiers SQL.
- Vos fichiers SQL, vous pouvez les stocker où vous voulez: sur votre disque, clé USB, etc..
- La commande CREATE DATABASE nomBD implique la création de deux fichiers sur votre disque dur
  - Un fichier nomBD.mdf
  - Un fichier nomBD.ldf
- Les données sont stockées dans un fichier MDF, toutes les transactions sont stockées dans un fichier LDF. (journal des logs nécessaires pour récupérer la base de données à partir d'un certain point).

# Où sont stockés les fichiers de la bases de données ?

- Bouton droit sur votre Base de données, puis propriétés.
- À l'onglet fichiers, remarquez les deux fichiers et leurs types.

Sélectionner une page

- Général
- Fichiers**
- Groupes de fichiers
- Options
- Suivi des modifications
- Autorisations
- Propriétés étendues
- Magasin des requêtes

Script ? Aide

Nom de la base de données : EmpcplgDB

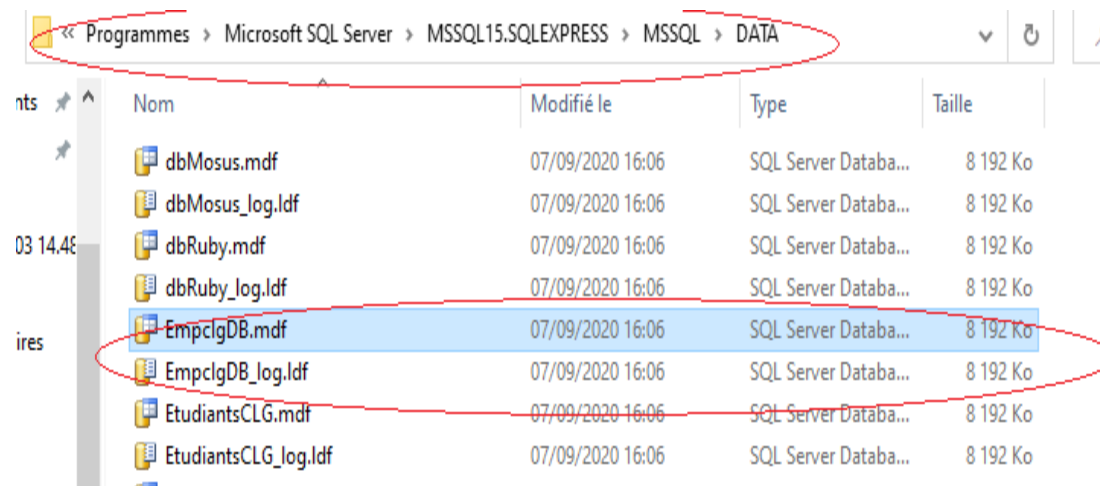
Propriétaire : Patoche

Utiliser l'indexation de texte intégral

Fichiers de la base de données :

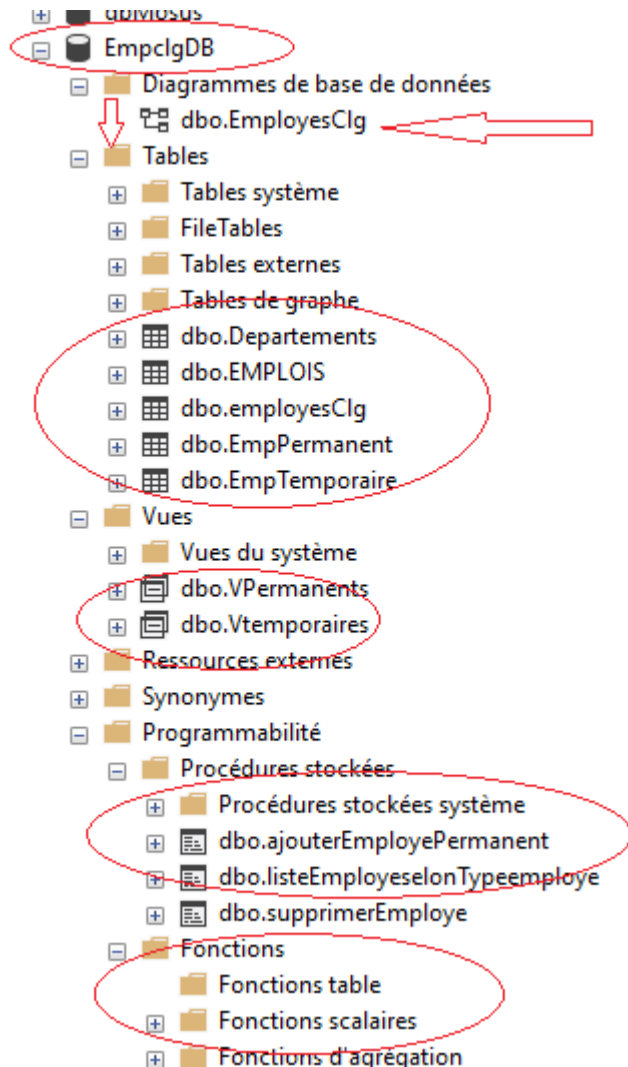
Nom logique	Type de fichier	Groupe de fichiers	Taille (Mo)	Croissance automatique/Taille maximale	Chemin d'accès
EmpcplgDB	Données de ...	PRIMARY	8	Par 64 Mo, illimitée	C:\Program Files\Microsoft SQL Server\MSSQL15.SQLEXP...
EmpcplgDB_log	JOURNAL	Non applicable	8	Par 64 Mo, limitée à 2097152 Mo	C:\Program Files\Microsoft SQL Server\MSSQL15.SQLEXP...

# Où sont stockés les fichiers de la bases de données ?



- Lorsque vous passez d'un ordinateur A à un ordinateur B , ce sera ces deux fichiers qu'il faudra copier.(de A vers B).
- Au niveau de l' ordinateur B, il faudra joindre le fichier MDF. Vous devez avoir les droits nécessaires. (on va le voir plus loin)

# Présentation de l'interface SSMS



Lorsque vous êtes connectés au serveur, l'ensemble de vos objets (tables, vues, diagrammes, procédures et fonctions stockées) se retrouvent dans la même base de données.

Remarquez :

- La BD EmpclgDB est sélectionnée.
  - Pour cette BD nous avons un diagramme de nom EmployesClg
  - Dans cette BD nous avons 5 tables: Departements, EMPLOIS, ...EmpTemporaire
  - Dans cette BD nous avons deux vues: Vpermanents et Vtemporaires
  - Dans cette BD nous avons 3 procédures stockées: ajouterEmployePermanent .....
  - Nous n'avons pas de fonctions pour le moment.
- › Pour écrire des requêtes SQL, vous devez faire: Fichier, nouveau , puis requête avec la connexion actuelle.

# Rappels SQL

- Toutes les commandes que vous avez apprises lors de votre cours de la session 2 sont valables.
- Les types de données de MS SQL server sont différents. **Voir les détails sur le site du cours**

Types	Explications
Varchar(n)	Les types caractères de longueur variable. Lorsque n'est pas précisé il vaut 1
Char(n)	Les types caractères de longueur fixe. Lorsque n'est pas précisé il vaut 1
Date	Définit une date dans SQL Server
Bigint, int, smallint, tinyint	représentent des entiers signés stockés respectivement dans 8,4,2 et un octet.
decimal[ ( p,s ) ]	p, Nombre total maximal (incluant le s)de chiffres décimaux à stocker. s est le nombre de chiffres après la virgule
Money	Données monétaires

# La propriété IDENTITY d'une colonne

- Vous pouvez mettre en œuvre des colonnes d'identification à l'aide de la propriété IDENTITY. Ce qui permet de réaliser un auto-incrément sur une colonne.
- En général, la propriété IDENTITY se définit sur la clé primaire.
- Si la propriété IDENTITY est définie sur une colonne, alors c'est le système qui insère des données dans cette colonne (pas l'utilisateur).
- La propriété IDENTITY se fait à la création de table.
- L'avantage d'avoir IDENTITY est que la clé primaire ne sera jamais dupliquée. Ce qui est utile lorsque les insertions se font par une interface C#, JAVA, PHP ...
- Exemples (au tableau ou voir le cours théorique)



## La propriété IDENTITY d'une colonnes

Parfois, nous souhaitons récupérer le dernier numéro inséré, dans ce cas on utilise:

- `SELECT @@IDENTITY`: Permet de retourner la dernière valeur insérée **pour l'ensemble** de la base de données durant la session en cours.
- `SELECT IDENT_CURRENT ('nomTable')` Permet de retourner le dernier numéro inséré d'une table.
- Exemples (au tableau ou voir le cours théorique)

## La propriété IDENTITY d'une colonnes

- SET IDENTITY\_INSERT nomTable ON: permet de faire une insertion manuelle de la colonne dans la table. Dans ce cas vous **perdez la séquence**.
- SET IDENTITY\_INSERT nomTable OFF permet de revenir à l'insertion automatique dans la colonne. On reprend la séquence après IDENTITY\_INSERT ON
- Exemples

# Rappels SQL

- À des exceptions près, Les commandes ALTER Table sont presque identiques avec ORACLE
- Les commandes DML sont identiques avec ORACLE. Pour les insertions, SQL SERVER offre une insertion multiples

**INSERT INTO Departements VALUES**

**('inf', 'Informatique')**

**('rsh', 'Ressources humaines')**

**('ges', 'Gestion')**

**('rec', 'Recherche et developpement');**

- Le COMMIT et le ROLLBACK ont le même rôle, mais n'ont pas la même syntaxe. Pour faire un COMMIT il faut d'abord faire un: **BEGIN TRANSACTION**
- Pour le SELECT, SQL Server offre: SELECT TOP N pour chercher les N premières lignes. Avec ORACLE il fallait faire une sous-requête avec un ROWNUM

# Points clés

- › Les commandes DDL (Data Definition Language) sont les mêmes qu'avec Oracle que vous avez vu en session 2. Seul les types de données changent. Dans la commande DROP TABLE vous n'avez pas CASCADE CONSTRAINTS
- › MS SQL Server permet de faire une auto-incrémentation de la clé primaire avec la propriété IDENTITY.
- › Les commandes DML (Data Manipulation Language) sont les mêmes qu'avec Oracle.
- › Les transactions : Pour faire un COMMIT vous avez besoin de d'abord un BEGIN TRANSACTION. Le ROLLBACK se fait jusqu'au dernier BEGIN TRANSACTION.
- › Il y a un chapitre sur les transactions plus loin.
- › La Commande SELECT se fait de la même façon qu'avec ORACLE. Les jointures se font avec INNER JOIN (ou RIGHT /LEFT OUTER JOIN)
- › Pour afficher les N premières lignes utiliser SELECT **TOP N**



CONCLUSION



QUESTIONS ??